

Lie-group interpolation and variational recovery for internal variables

Alejandro Mota · WaiChing Sun · Jakob T. Ostien · James W. Foulk III · Kevin N. Long

Received: 20 August 2012 / Accepted: 10 May 2013
© Springer-Verlag Berlin Heidelberg (outside the USA) 2013

Abstract We propose a variational procedure for the recovery of internal variables, in effect extending them from integration points to the entire domain. The objective is to perform the recovery with minimum error and at the same time guarantee that the internal variables remain in their admissible spaces. The minimization of the error is achieved by a three-field finite element formulation. The fields in the formulation are the deformation mapping, the *target* or mapped internal variables and a Lagrange multiplier that enforces the equality between the *source* and target internal variables. This formulation leads to an L_2 projection that minimizes the distance between the source and target internal variables as measured in the L_2 norm of the internal variable space. To ensure that the target internal variables remain in their original space, their interpolation is performed by recourse to Lie groups, which allows for direct polynomial interpolation of the corresponding Lie algebras by means of the logarithmic map. Once the Lie algebras are interpolated, the mapped variables are recovered by the exponential map, thus guaranteeing that they remain in the appropriate space.

Keywords Variational methods · Internal variables · Recovery · Interpolation · Lie groups

A. Mota (✉) · W. Sun · J. T. Ostien · J. W. Foulk III
Mechanics of Materials Department, Sandia National Laboratories,
Livermore, CA 94550, USA
e-mail: amota@sandia.gov

K. N. Long
Solid Mechanics Department, Sandia National Laboratories,
Albuquerque, NM 87185, USA

1 Introduction

The transfer of field data from one mesh to another is a need that arises frequently within the context of mesh adaption in the finite element method [5, 14, 26, 27, 31, 33, 34]. Fields that are available at the nodes may be directly mapped by using the corresponding interpolation functions. The situation is more complicated, however, in simulations that carry state information in internal variables, as these are normally available only at integration points.

1.1 Previous work

Related work to our proposed technique can be divided into two classes of methods: the mapping of internal variables from one mesh to another, and stress recovery techniques.

Different methods have been devised to map internal variables from one mesh to another. Ortiz and Quigley [27] propose transfer operators based on a three-field Hu-Washizu formulation. By choosing discontinuous interpolation functions for the deformation gradient and first Piola-Kirchhoff stress fields, the authors derive transfer operators that are local to each element. Radovitzky and Ortiz [33] use a transfer operator which reduces to extrapolation of the internal variables from the integration points in the source mesh to the integration points in the target mesh. Rashid [34] introduces a method that assumes constant values for the internal variables within the cells of a Voronoi tessellation based on the integration points of the original mesh. These values are then transferred to the final mesh using another Voronoi tessellation and an algorithm for finding the intersecting volumes of the cells corresponding to the source and target meshes. Jiao and Heath [14] propose a technique for transferring information in surface meshes based on a third mesh that they term common refinement. This mesh is conformal to both the source

and target surface meshes. The authors then proceed to define an L_2 minimization between the source and target internal variable fields for the transfer. They also note the desirable properties of the L_2 minimization for mapping, such as that it requires solving a linear system of equations with a corresponding matrix that is positive-definite and sparse. Bucher et al. [5] derive transfer operators that extrapolate the internal variables at integration points to nodes in the source mesh using serendipity interpolation functions.

The proposed mixed finite element formulation is also related to classical stress recovery techniques such as interpolation-extrapolation from superconvergent points [2], L_2 stress projection [25], stress smoothing [13], superconvergent patch recovery [43] and variational stress recovery [23], among others. All of these start from a multi-field variational principle, although their objectives are different [42, chap. 8].

1.2 Admissible spaces

It is common for finite deformation constitutive models to be endowed with internal variables that do not belong to linear spaces. Nevertheless, a problem that is prevalent to the mapping procedures hitherto described is that they transfer internal variables with the addition operator, typically via Lagrange interpolation. This operator, however, may not be admissible in the spaces in which these variables exist, and as a consequence such procedures do not, in general, guarantee that transferred internal variables remain in their appropriate spaces. For example, a scalar isotropic damage parameter might be extrapolated outside of its admissible range, $[0, 1)$.

1.3 Proposed method

We advance a three-field finite element formulation as a method for recovering internal variables. The additional fields in the formulation are the *target* field of internal variables and a Lagrange multiplier that enforces the equality between the *source* and target internal variables. The Lagrange multiplier is subsequently identified as the corresponding conjugate thermodynamic force. The formulation leads to an expression for the target internal variables that is an L_2 projection of the source internal variables onto the space spanned by the interpolation functions selected for the extra fields. In effect, the projection extends the field of internal variables from the integration points to the entire domain. Once extended in this manner, values for the internal variables may be computed at any arbitrary point as a means of transferring the variables to a different mesh. By using a variational approach, the distance between the source and target internal variable fields is minimized in the L_2 norm of their space, and thus the projection is orthogonal and the corresponding operator is self-adjoint. The matrices that result

from the operator are symmetric positive definite, sparse and of narrow band.

Next it is shown that for a set of common internal variables, the Lie algebras corresponding to these variables lie in linear spaces and are therefore suitable for polynomial interpolation. The proposed recovery scheme preserves the constraints imposed on the internal variables as long as they belong to a Lie group. We adhere to the classical definition of internal variables as fields that are endowed with their own evolution equations as to eliminate the explicit dependence of the stored energy function from the history of deformation [1, 24]. Thus, the proposed variational principle states that thermodynamic forces conjugate to the internal variables (such as stresses) are not mapped, but are computed instead based on the target internal variables, thereby satisfying their own constraints (such as that stresses be within the elastic domain) and balance laws.

Assuming that internal variables belong to Lie groups, the corresponding Lie algebras may be obtained using the logarithmic map. Upon interpolation in the Lie algebra, the internal variable of interest is recovered by using the exponential map [9, 16, 19, 37]. The computation of the logarithmic map may be effected by using explicit formulas (if the eigenvalues are readily available) [15] or the inverse scaling and squaring algorithm with Padé approximants [6, 7]. A fast and accurate method for the computation of the exponential map is provided by the scaling and squaring algorithm with Padé approximants [10, 11]. Most tensor fields of interest, however, admit a polar decomposition and thus it proves convenient to interpolate the rotational and stretch components separately. An added benefit of this method is that the computation of logarithmic maps for these components is relatively straightforward.

Finally, the L_2 projection and Lie algebra interpolation are applied first separately and then concurrently to selected numerical examples to demonstrate their effect on the recovery of internal variables.

1.4 General framework

We are concerned with a general class of materials that admit incremental variational constitutive updates. Within this framework, an incremental stress potential embodies the constitutive behavior of the material during a time increment, including elasticity, viscoelasticity, viscoplasticity, and rate dependence [3, 8, 18, 20, 22, 28, 41]. The mechanical response of solids of this type is characterized by a dissipation potential of the form

$$D(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}, \dot{\mathbf{z}}) = \dot{A}(\mathbf{F}, \mathbf{z}) + \phi(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}) + \psi^*(\mathbf{z}, \dot{\mathbf{z}}), \quad (1.1)$$

in which $A(\mathbf{F}, \mathbf{z})$ is the Helmholtz free-energy density, $\phi(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z})$ is a viscous potential, $\psi^*(\mathbf{z}, \dot{\mathbf{z}})$ is a dual kinetic potential or dissipation function, \mathbf{F} is the deformation gra-

dent and \mathbf{z} is a collection of suitable internal variables that describe the state of the material at a given point. The first Piola-Kirchhoff stress and the conjugate thermodynamic forces to \mathbf{z} are given by

$$\begin{aligned} \mathbf{P} &= \frac{\partial D}{\partial \dot{\mathbf{F}}}(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}, \dot{\mathbf{z}}) = \frac{\partial A}{\partial \mathbf{F}}(\mathbf{F}, \mathbf{z}) + \frac{\partial \phi}{\partial \dot{\mathbf{F}}}(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}), \\ \mathbf{y} &= -\frac{\partial A}{\partial \mathbf{z}}(\mathbf{F}, \mathbf{z}), \end{aligned} \tag{1.2}$$

respectively. In order to ensure a variational structure, we have postulated the existence of a dual kinetic potential or dissipation function $\psi^*(\mathbf{z}, \dot{\mathbf{z}})$ such that

$$\mathbf{y} = \frac{\partial \psi^*}{\partial \dot{\mathbf{z}}}(\mathbf{z}, \dot{\mathbf{z}}). \tag{1.3}$$

Next, we minimize the dissipation potential (1.1) with respect to the internal variable rates as

$$\begin{aligned} \inf_{\dot{\mathbf{z}}} [D(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}, \dot{\mathbf{z}})] &= \inf_{\dot{\mathbf{z}}} \left[\dot{A}(\mathbf{F}, \mathbf{z}) + \psi^*(\mathbf{z}, \dot{\mathbf{z}}) \right. \\ &\quad \left. + \phi(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}), \right. \\ &= \inf_{\dot{\mathbf{z}}} \left[\frac{\partial A}{\partial \mathbf{F}}(\mathbf{F}, \mathbf{z}) : \dot{\mathbf{F}} + \frac{\partial A}{\partial \mathbf{z}}(\mathbf{F}, \mathbf{z}) \cdot \right. \\ &\quad \left. \dot{\mathbf{z}} + \psi^*(\mathbf{z}, \dot{\mathbf{z}}) \right] + \phi(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}), \end{aligned} \tag{1.4}$$

which in turn leads to Biot’s equation for standard dissipative systems

$$\frac{\partial A}{\partial \mathbf{z}}(\mathbf{F}, \mathbf{z}) + \frac{\partial \psi^*}{\partial \dot{\mathbf{z}}}(\mathbf{z}, \dot{\mathbf{z}}) = \mathbf{0}. \tag{1.5}$$

Approximate solutions to this equation may be found by recourse to the incremental energy density function for a time increment $t \in [t_n, t_{n+1}]$

$$w(\mathbf{F}_{n+1}, \mathbf{z}_{n+1}) := \int_{t_n}^{t_{n+1}} [\dot{A}(\mathbf{F}, \mathbf{z}) + \phi(\mathbf{F}, \dot{\mathbf{F}}, \mathbf{z}) + \psi^*(\mathbf{z}, \dot{\mathbf{z}})] dt, \tag{1.6}$$

in which the integral is evaluated using a midpoint-like rule as follows

$$\begin{aligned} w(\mathbf{F}_{n+1}, \mathbf{z}_{n+1}) &\approx A(\mathbf{F}_{n+1}, \mathbf{z}_{n+1}) - A(\mathbf{F}_n, \mathbf{z}_n) \\ &\quad + \Delta t \left[\phi \left(\mathbf{F}_{n+\alpha}, \frac{\Delta \mathbf{F}}{\Delta t}, \mathbf{z}_{n+\alpha} \right) \right. \\ &\quad \left. + \psi^* \left(\mathbf{z}_{n+\alpha}, \frac{\Delta \mathbf{z}}{\Delta t} \right) \right], \end{aligned} \tag{1.7}$$

with

$$\Delta t := t_{n+1} - t_n, \Delta \mathbf{F} := \mathbf{F}_{n+1} \mathbf{F}_n^{-1}, \Delta \mathbf{z} := \mathbf{z}_{n+1} - \mathbf{z}_n, \tag{1.8}$$

and

$$\begin{aligned} \mathbf{F}_{n+\alpha} &:= \exp[(1 - \alpha) \log \mathbf{F}_n + \alpha \log \mathbf{F}_{n+1}], \\ \mathbf{z}_{n+\alpha} &:= (1 - \alpha)\mathbf{z}_n + \alpha\mathbf{z}_{n+1}, \end{aligned} \tag{1.9}$$

where α is an algorithmic parameter. In order to obtain an explicit scheme, we choose $\alpha = 0$. Next, we define the incremental stress potential as

$$\begin{aligned} W(\mathbf{F}_{n+1}) &:= \inf_{\mathbf{z}_{n+1}} [w(\mathbf{F}_{n+1}, \mathbf{z}_{n+1})] \\ &= \inf_{\mathbf{z}_{n+1}} \left[A(\mathbf{F}_{n+1}, \mathbf{z}_{n+1}) - A(\mathbf{F}_n, \mathbf{z}_n) \right. \\ &\quad \left. + \Delta t \psi^* \left(\mathbf{z}_n, \frac{\Delta \mathbf{z}}{\Delta t} \right) \right] \\ &\quad + \Delta t \phi \left(\mathbf{F}_n, \frac{\Delta \mathbf{F}}{\Delta t}, \mathbf{z}_n \right). \end{aligned} \tag{1.10}$$

This minimization provides an optimal path for the internal variables \mathbf{z} in the time increment $t \in [t_n, t_{n+1}]$. Furthermore, the Euler-Lagrange equation corresponding to (1.10) is

$$\frac{\partial A}{\partial \mathbf{z}_{n+1}}(\mathbf{F}_{n+1}, \mathbf{z}_{n+1}) + \Delta t \frac{\partial \psi^*}{\partial \mathbf{z}_{n+1}} \left(\mathbf{z}_n, \frac{\Delta \mathbf{z}}{\Delta t} \right) = \mathbf{0}. \tag{1.11}$$

which is a discrete version of Biot’s equation (1.5) [21]. The incremental first Piola-Kirchhoff stress and tangent moduli can be computed in turn as

$$\mathbf{P}_{n+1} := \frac{\partial W}{\partial \mathbf{F}}(\mathbf{F}_{n+1}), \quad \mathbb{C}_{n+1} := \frac{\partial^2 W}{\partial \mathbf{F}^2}(\mathbf{F}_{n+1}), \tag{1.12}$$

respectively. Thus, by using variational constitutive updates, the stress and the tangent moduli can be derived from the hyperelastic-like potential (1.10) for a very general class of constitutive behavior that may include viscosity and rate dependence.

2 Finite element formulation

Consider a body $B \subset \mathbb{R}^3$ undergoing a motion described by the mapping $\mathbf{x} = \boldsymbol{\varphi}(\mathbf{X}, t) : B \times [t_1, t_2] \rightarrow \mathbb{R}^3$, with the deformation gradient defined by $\mathbf{F} := \text{Grad } \boldsymbol{\varphi}$.

Assume that the boundary ∂B , with unit normal \mathbf{N} , is the union of a displacement boundary $\partial_\varphi B$, where boundary displacements $\boldsymbol{\chi} : \partial_\varphi B \times [t_1, t_2] \rightarrow \mathbb{R}^3$ are prescribed, and a traction boundary $\partial_T B$, where tractions $\mathbf{T} : \partial_T B \times [t_1, t_2] \rightarrow \mathbb{R}^3$ are applied ($\partial_\varphi B \cap \partial_T B = \emptyset$). Let also $R\mathbf{B} : B \times [t_1, t_2] \rightarrow \mathbb{R}^3$ be the body force, with R the mass density in the reference configuration. Furthermore, for

every $t \in [t_1, t_2]$ introduce the energy functional

$$\begin{aligned} \Phi_0[\boldsymbol{\varphi}] := & \int_B A(\mathbf{F}, \mathbf{z}) \, dV - \int_B \mathbf{R}\mathbf{B} \cdot \boldsymbol{\varphi} \, dV \\ & - \int_{\partial_T B} \mathbf{T} \cdot \boldsymbol{\varphi} \, dS, \end{aligned} \tag{2.1}$$

in which $A(\mathbf{F}, \mathbf{z})$ is the Helmholtz free-energy density and \mathbf{z} is a collection of internal variables.

2.1 Three-field functional

The functional (2.1) is modified by introducing a constraint as

$$\begin{aligned} \Phi[\boldsymbol{\varphi}, \bar{\mathbf{z}}, \bar{\mathbf{y}}] := & \int_B A(\mathbf{F}, \bar{\mathbf{z}}) \, dV + \int_B \bar{\mathbf{y}} \cdot (\bar{\mathbf{z}} - \mathbf{z}) \, dV \\ & - \int_B \mathbf{R}\mathbf{B} \cdot \boldsymbol{\varphi} \, dV - \int_{\partial_T B} \mathbf{T} \cdot \boldsymbol{\varphi} \, dS, \end{aligned} \tag{2.2}$$

in which $\bar{\mathbf{z}}$ is the field of *target* internal variables that is constrained to be equal to the *source* internal variables \mathbf{z} by means of the Lagrange multiplier $\bar{\mathbf{y}}$. Although the Helmholtz free-energy density A is now evaluated using $\bar{\mathbf{z}}$ instead of \mathbf{z} , the functionals in (2.1) and (2.2) are equivalent at this stage due to this constraint.

Assume that $\boldsymbol{\varphi} \in U := (W_2^1(B))^3$, $\bar{\mathbf{z}}, \bar{\mathbf{y}} \in V := (W_2^1(B))^q$, in which $W_2^1(B)$ is the Sobolev space of square-integrable functions with square-integrable first derivatives, and q is the number of scalar parameters used to represent the collection of internal variables. The functional (2.2) is optimized by applying variations with respect to the independent fields $\boldsymbol{\varphi}$, $\bar{\mathbf{z}}$ and $\bar{\mathbf{y}}$. Define test functions corresponding to these fields as $\boldsymbol{\xi} \in U$, $\boldsymbol{\eta}, \boldsymbol{\zeta} \in V$, with $\boldsymbol{\xi} = \mathbf{0}$ on $\partial_\varphi B$. The variations follow as

$$\begin{aligned} D\Phi[\boldsymbol{\varphi}, \bar{\mathbf{z}}, \bar{\mathbf{y}}](\boldsymbol{\xi}) = & \int_B \mathbf{P} : \text{Grad } \boldsymbol{\xi} \, dV - \int_B \mathbf{R}\mathbf{B} \cdot \boldsymbol{\xi} \, dV \\ & - \int_{\partial_T B} \mathbf{T} \cdot \boldsymbol{\xi} \, dS = 0, \end{aligned} \tag{2.3}$$

$$D\Phi[\boldsymbol{\varphi}, \bar{\mathbf{z}}, \bar{\mathbf{y}}](\boldsymbol{\eta}) = \int_B (\bar{\mathbf{y}} - \mathbf{y}) \cdot \boldsymbol{\eta} \, dV = 0, \tag{2.4}$$

$$D\Phi[\boldsymbol{\varphi}, \bar{\mathbf{z}}, \bar{\mathbf{y}}](\boldsymbol{\zeta}) = \int_B (\bar{\mathbf{z}} - \mathbf{z}) \cdot \boldsymbol{\zeta} \, dV = 0, \tag{2.5}$$

where $\mathbf{P} := \partial A / \partial \mathbf{F}$ is the first Piola-Kirchhoff stress and $\mathbf{y} := -\partial A / \partial \bar{\mathbf{z}}$ is the thermodynamic force conjugate to $\bar{\mathbf{z}}$.

The corresponding Euler–Lagrange equations are

$$\begin{aligned} \text{Div } \mathbf{P} + \mathbf{R}\mathbf{B} &= \mathbf{0} \text{ in } B, \mathbf{P}\mathbf{N} = \mathbf{T} \text{ on } \partial_T B, \\ \bar{\mathbf{y}} &= \mathbf{y} \text{ in } B, \\ \bar{\mathbf{z}} &= \mathbf{z} \text{ in } B, \end{aligned} \tag{2.6}$$

as expected. Note that the equilibrium condition (2.6) is evaluated using the target internal variable field $\bar{\mathbf{z}}$, and therefore equilibrium and constitutive constraints are satisfied using this field. Next, introduce discretizations for the fields and test functions as

$$\begin{aligned} \boldsymbol{\varphi}_h(\mathbf{X}) &:= N_a(\mathbf{X})\boldsymbol{\varphi}_a \in U_h, \quad \boldsymbol{\xi}_h(\mathbf{X}) := N_b(\mathbf{X})\boldsymbol{\xi}_b \in U_h, \\ \bar{\mathbf{z}}_h(\mathbf{X}) &:= \lambda_\alpha(\mathbf{X})\bar{\mathbf{z}}_\alpha \in V_h, \quad \boldsymbol{\eta}_h(\mathbf{X}) := \lambda_\beta(\mathbf{X})\boldsymbol{\eta}_\beta \in V_h, \\ \bar{\mathbf{y}}_h(\mathbf{X}) &:= \lambda_\alpha(\mathbf{X})\bar{\mathbf{y}}_\alpha \in V_h, \quad \boldsymbol{\zeta}_h(\mathbf{X}) := \lambda_\beta(\mathbf{X})\boldsymbol{\zeta}_\beta \in V_h, \end{aligned} \tag{2.7}$$

where N_a and N_b are interpolation functions for $\boldsymbol{\varphi}$ and $\boldsymbol{\xi}$, λ_α and λ_β are interpolation functions for $(\bar{\mathbf{z}}, \bar{\mathbf{y}})$ and $(\boldsymbol{\eta}, \boldsymbol{\zeta})$, and $(a, b) \in [1 \dots N]$ and $(\alpha, \beta) \in [1 \dots M]$, in which N is the number of nodes for $\boldsymbol{\varphi}$, M is the number of nodes for $\bar{\mathbf{z}}$ and $\bar{\mathbf{y}}$ respectively. $U_h \subset U$ and $V_h \subset V$ are finite-dimensional subspaces of U and V respectively. Introducing these discretizations into the variational statements (2.3)–(2.5) gives

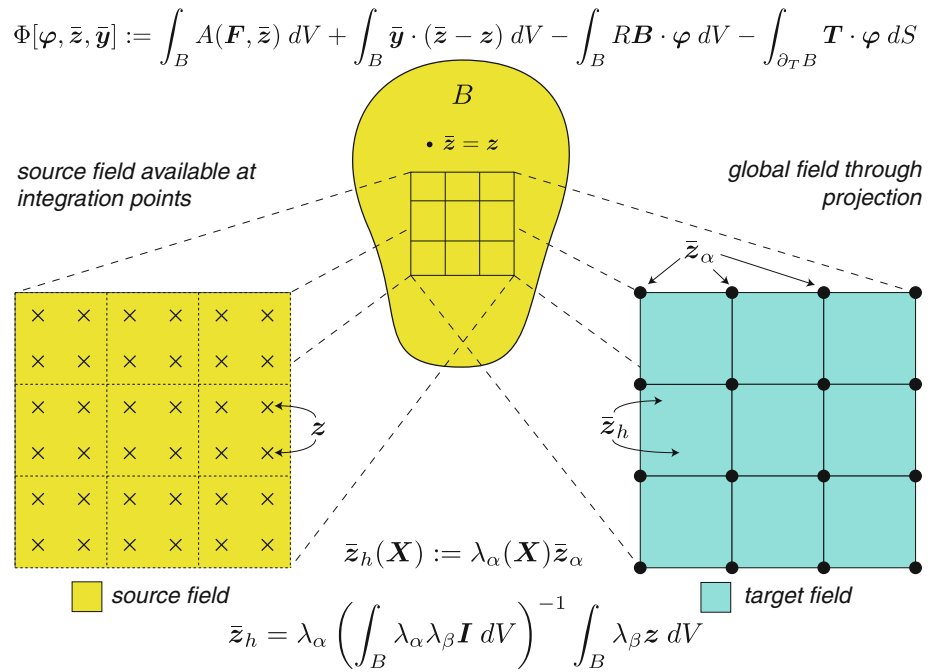
$$\int_B \mathbf{P} \cdot \text{Grad } N_a \, dV - \int_B \mathbf{R}\mathbf{B} N_a \, dV - \int_{\partial_T B} \mathbf{T} N_a \, dS = \mathbf{0}, \tag{2.8}$$

$$\bar{\mathbf{y}}_h = \lambda_\alpha \left(\int_B \lambda_\alpha \lambda_\beta \mathbf{I} \, dV \right)^{-1} \int_B \lambda_\beta \mathbf{y} \, dV, \tag{2.9}$$

$$\bar{\mathbf{z}}_h = \lambda_\alpha \left(\int_B \lambda_\alpha \lambda_\beta \mathbf{I} \, dV \right)^{-1} \int_B \lambda_\beta \mathbf{z} \, dV, \tag{2.10}$$

which are the discrete statements of equilibrium, the discrete thermodynamic forces and discrete target internal variables, respectively, and with \mathbf{I} being the $q \times q$ identity. Note that λ_α and λ_β form a basis for the space V_h , therefore (2.9) and (2.10) are projections of the fields \mathbf{y} and \mathbf{z} onto V_h . Effective computation of the integrals in these expressions readily suggests the use of interpolation functions λ_α and λ_β that are of the same order or less than N_a and N_b . This allows the application of the same integration scheme for (2.9) and (2.10) as for (2.8), ensuring that the projections will be fully integrated (provided that full integration is used for the equilibrium statement as well) using the same integration points, thus avoiding any unnecessary transfer of variables at this stage. Furthermore, if the interpolation functions λ_α and λ_β are discontinuous across element boundaries, the projection (2.10) reduces to extrapolation of values from integration points to nodes (cf. Ortiz and Quigley [27], Radovitzky and

Fig. 1 The formulation (2.2) leads to a projection (2.10) of integration-point values of the source field z into nodal values of the discrete target field \bar{z}_α (2.11). The discrete target field \bar{z}_h covers the entire domain B and is recovered by using the nodal values \bar{z}_α and the interpolation functions λ_α



Ortiz [33]). For convenience, introduce

$$\mathbf{H}_{\alpha\beta} := \int_B \lambda_\alpha \lambda_\beta \mathbf{I} dV, \quad \boldsymbol{\Sigma}_\beta := \int_B \lambda_\beta z dV,$$

$$\bar{z}_\alpha := \mathbf{H}_{\alpha\beta}^{-1} \boldsymbol{\Sigma}_\beta, \tag{2.11}$$

in which \bar{z}_α are the nodal values of the discrete target field shown in Fig. 1. Then the projection (2.10) can also be expressed as

$$\bar{z}_h(\mathbf{X}) = \lambda_\alpha(\mathbf{X}) \bar{z}_\alpha. \tag{2.12}$$

The proof that the projection is optimal is simple and shown in the Appendix.

3 Interpolation of internal variables

The interpolation of internal variables using standard finite element interpolation functions may lead to values that are outside their admissible spaces. To ensure that polynomial interpolation yields reasonable results, the internal variables must belong to a *vector space* over the field of scalars.

3.1 Groups

A vector space is defined as an *abelian group* together with the field of scalars and scalar multiplication. A *group* in turn is defined as a set G and a binary operation that satisfy the following group axioms:

1. The operation is closed on the set G .

2. The operation is associative.
3. The operation admits the identity element.
4. The operation admits the inverse for every element of G .

If the operation is also commutative, the group is said to be *abelian* [9,16,32,37]. Polynomial interpolation requires addition as the group operation, but many internal variables of interest, such as scalar damage, tensor fields like decompositions of the deformation gradient and others do not belong to additive abelian groups. Polynomial interpolation, however, may still be applied if these variables are transformed to additive abelian groups, and this may be effected by recourse to *Lie groups* and *Lie algebras*.

3.2 Lie groups and Lie algebras

A Lie group G is defined as a multiplicative group, which is also a manifold, that has the following properties:

1. The multiplication operation is smooth on the set G .
2. The inverse operation is smooth on the set G .

A Lie algebra is a vector space g with a bilinear operation $[u, v] := uv - vu \forall u, v \in g$, called the *Lie bracket*, that satisfies:

1. Antisymmetry: $[u, v] = -[v, u] \forall u, v \in g$.
2. The Jacobi identity: $[u, [v, w]] + [v, [w, u]] + [w, [u, v]] = 0 \forall u, v, w \in g$.

Table 1 Some common Lie groups and their corresponding Lie algebras

Lie group	Lie algebra
Positive reals $\mathbb{R}^+ = \{a \in (0, +\infty)\}$	Reals $\mathbb{R} = \{b \in (-\infty, +\infty)\}$
General linear group $GL(n) = \{A \in M(n) \det A \neq 0\}$	$n \times n$ matrices $gl(n) = M(n)$
Identity component of $GL(n)$ $GL^+(n) = \{A \in GL(n) \det A > 0\}$	$n \times n$ matrices $gl(n) = M(n)$
Special linear group $SL(n) = \{A \in GL(n) \det A = 1\}$	Traceless matrices $sl(n) = \{B \in gl(n) \text{tr } B = 0\}$
Orthogonal group $O(n) = \{A \in GL(n) AA^T = I\}$	Skew-symmetric matrices $so(n) = \{B \in gl(n) B = -B^T\}$
Special orthogonal group (rotations) $SO(n) = \{A \in O(n) \det A = 1\}$	Skew-symmetric matrices $so(n) = \{B \in gl(n) B = -B^T\}$
Symplectic group ^a $Sp(n) = \{A \in GL(2n) A^T J A = J\}$	$sp(n) = \{B \in gl(2n) B^T J = -JB\}$
Affine isometries (rigid body motions) ^b $SE(n) = \left\{ \begin{pmatrix} A & x \\ 0^T & 1 \end{pmatrix} \middle A \in SO(n) \right\}$	$se(n) = \left\{ \begin{pmatrix} B & x \\ 0^T & 0 \end{pmatrix} \middle B \in so(n) \right\}$

The Lie bracket for all these Lie algebras is the commutator $[u, v] := uv - vu$, except for the reals, for which it is zero
^a $J := [0, I; -I, 0]; 0, I \in M(n)$
^b $x, 0 \in \mathbb{R}^n$

A Lie algebra may be interpreted as the infinitesimal version of its corresponding Lie group, and the transformations that relate them are the exponential (from Lie algebra to Lie group) and logarithmic (from Lie group to Lie algebra) maps [9, 16, 32, 37]. Note that the operation for the Lie algebras is addition, which makes them amenable to polynomial interpolation. Some classical Lie groups and their corresponding Lie algebras are shown in Table 1.

3.3 Interpolation and extrapolation

Implicit in the formulation of Sects. 1 and 2 is the assumption that the source field z belongs to a vector space over the scalars g with addition as the group operation. This ensures that polynomial interpolation or linear combinations in general yield values of z that belong to g . The source field z is the result of applying the logarithmic map to an *original* field of internal variables Z which cannot be interpolated or extrapolated directly by polynomials or enter a linear combination inasmuch as its group operation is multiplication.

Assuming that an original field Z is given as a collection of M point values Z_α that belong to a Lie group G , with g its corresponding Lie algebra, it may be interpolated or extrapolated indirectly by

$$\begin{aligned}
 z_\alpha &:= \log(Z_\alpha), \quad Z_\alpha \in G, \quad z_\alpha \in g, \quad \alpha \in [1 \dots M], \\
 z(X) &= \sum_{\alpha=1}^M \phi_\alpha(X) z_\alpha, \quad z(X) \in g, \\
 Z(X) &:= \exp(z(X)), \quad Z(X) \in G,
 \end{aligned}
 \tag{3.1}$$

in which X is an arbitrary position within the domain and ϕ_α are suitable interpolation functions. Likewise, any other expression that involves discrete values of Z_α may be evaluated in this manner. For example, the projection (2.10) requires values of the *source* field z at integration points. These are computed by applying the logarithmic map to the corresponding integration point values of the *original* field Z . Once the projection is performed, mapped values of the original field are recovered by applying the exponential map to the *target* field.

Failure to perform interpolation, extrapolation or linear combinations with a scheme that preserves the variables in their original space may lead to severe error and even non-physical results. Tables 2 and 3 illustrate with some examples the significant difference in the results of interpolation and extrapolation with and without the use of Lie algebras. Examples of internal variables and their transformation to Lie algebras are shown in Table 4.

4 Time derivatives and incremental updates

For completeness, in this section we address the issue of time derivatives and incremental updates of internal variables within the framework of Lie groups and Lie algebras.

4.1 Evolution of internal variables

Assume that the original internal variable $Z(t)$ belongs to a Lie group G from Table 1, with g its corresponding Lie algebra. The evolution of $Z(t)$ is described by suitable kinetic equations of the general form

Table 2 Interpolation and extrapolation without Lie algebras

Lie group G	\mathbf{Z}_1	\mathbf{Z}_2	$\mathbf{Z}(0)$	$\in G$	$\mathbf{Z}(2)$	$\in G$
\mathbb{R}^+	0.90	0.10	0.50	Yes	-0.30	No
$GL^+(3)$	$\begin{pmatrix} 2 & 0 & 4 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 4 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 2 \end{pmatrix}$	No	$\begin{pmatrix} 2 & 0 & -2 \\ 0 & 2 & 0 \\ 6 & 0 & 2 \end{pmatrix}$	Yes
$SL(3)$	$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	No	$\begin{pmatrix} 1 & -1 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	No
$SO(3)$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.00 & 0.50 \\ 0.00 & 0.50 & -0.50 \\ -0.50 & 0.50 & 0.00 \end{pmatrix}$	No	$\begin{pmatrix} -0.50 & 0.00 & 1.50 \\ 0.00 & 1.50 & 0.50 \\ -1.50 & -1.50 & 0.00 \end{pmatrix}$	No

$\mathbf{Z}_1 = \mathbf{Z}(-1)$ and $\mathbf{Z}_2 = \mathbf{Z}(1)$ are given data and belong to the corresponding Lie group. $\mathbf{Z}(\xi) := N_1(\xi)\mathbf{Z}_1 + N_2(\xi)\mathbf{Z}_2$. $N_1(\xi) := \frac{1}{2}(1 - \xi)$ and $N_2(\xi) := \frac{1}{2}(1 + \xi)$

Table 3 Interpolation and extrapolation with Lie algebras

Lie group G	\mathbf{Z}_1	\mathbf{Z}_2	$\mathbf{Z}(0)$	$\in G$	$\mathbf{Z}(2)$	$\in G$
\mathbb{R}^+	0.90	0.10	0.30	Yes	0.03	Yes
$GL^+(3)$	$\begin{pmatrix} 2 & 0 & 4 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 4 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 3.09 & 0.00 & 2.35 \\ 0.00 & 2.00 & 0.00 \\ 2.35 & 0.00 & 3.09 \end{pmatrix}$	Yes	$\begin{pmatrix} -0.32 & 0.00 & -1.14 \\ 0.00 & 2.00 & 0.00 \\ 3.42 & 0.00 & -0.32 \end{pmatrix}$	Yes
$SL(3)$	$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.54 & 1.18 & 0.00 \\ 1.18 & 1.54 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}$	Yes	$\begin{pmatrix} -0.16 & -0.57 & 0.00 \\ 1.71 & -0.16 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}$	Yes
$SO(3)$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0.72 & 0.28 & 0.63 \\ 0.28 & 0.72 & -0.63 \\ -0.63 & 0.63 & 0.44 \end{pmatrix}$	Yes	$\begin{pmatrix} -0.61 & -0.54 & 0.58 \\ -0.54 & 0.82 & 0.19 \\ -0.58 & -0.19 & -0.79 \end{pmatrix}$	Yes

$\mathbf{Z}_1 = \mathbf{Z}(-1)$ and $\mathbf{Z}_2 = \mathbf{Z}(1)$ are given data and belong to the corresponding Lie group. $\mathbf{Z}(\xi) := \exp \mathbf{z}(\xi)$, $\mathbf{z}(\xi) := N_1(\xi)\mathbf{z}_1 + N_2(\xi)\mathbf{z}_2$, $\mathbf{z}_1 := \log \mathbf{Z}_1$, $\mathbf{z}_2 := \log \mathbf{Z}_2$. $N_1(\xi) := \frac{1}{2}(1 - \xi)$ and $N_2(\xi) := \frac{1}{2}(1 + \xi)$

Table 4 Examples of internal variables and their corresponding transformation to Lie algebras

Internal variable	Transformation
Scalar damage	$D \in [0, 1)$ with an independent evolution equation. Define $H := 1 - D$ and assume $H \in \mathbb{R}^+$, a Lie group from Table 1. Then $h := \log(H) \in \mathbb{R}$ is in the corresponding Lie algebra and may be interpolated directly. The damage is recovered with $D = 1 - \exp(h)$ as in (3.1), ensuring that $D \in [0, 1)$, provided that the evolution equation satisfies the constraint
Tensor damage	det $\mathbf{D} \in [0, 1)$ with an independent evolution equation for anisotropic damage. Define $\mathbf{H} := \mathbf{I} - \mathbf{D}$, thus $\mathbf{H} \in GL^+(3)$. Then $\mathbf{h} := \log(\mathbf{H}) \in gl(3)$ is in the corresponding Lie algebra, and the interpolated damage tensor may be recovered using the same procedure as above
Tensor variables	Belong to $GL(3)$, interpolated by $gl(3)$
Isochoric deformations	Belong to $SL(3)$, interpolated by $sl(3)$
Rotations	Belong to $SO(3)$, interpolated by $so(3)$
Volumetric deformations	Belong to \mathbb{R}^+ , interpolated by \mathbb{R}

$$\frac{\partial \mathbf{Z}}{\partial t}(t) = \mathbf{f}(\mathbf{Z}(t)), \quad \mathbf{Z}(0) = \mathbf{I}, \quad \frac{\partial \mathbf{Z}}{\partial t}(0) = \mathbf{z} \in g. \quad (4.1)$$

It can be shown that a one-parameter curve such as (4.1) satisfies

$$\frac{\partial \mathbf{Z}}{\partial t}(t)\mathbf{Z}^{-1}(t) \in g, \quad (4.2)$$

in which the inverse function $\mathbf{Z}^{-1}(t)$ exists as that is one of the requirements of Lie groups [9, 16, 32, 37]. Examples of kinetic equations that may be written in the form (4.2) are shown in Table 5.

A one-parameter subgroup of the group G is a continuous map $\hat{\mathbf{Z}} : \mathbb{R} \rightarrow G$ such that $\forall t \in \mathbb{R}$ and $\forall s \in \mathbb{R}$ it follows that $\hat{\mathbf{Z}}(t + s) = \hat{\mathbf{Z}}(t)\hat{\mathbf{Z}}(s)$. It can be shown that this map is differentiable and that there is a unique $\hat{\mathbf{z}} \in g$ independent of t such that

$$\hat{\mathbf{Z}}(t) = \exp(t\hat{\mathbf{z}}), \quad (4.3)$$

and

$$\frac{\partial \hat{\mathbf{Z}}}{\partial t}(t) = \hat{\mathbf{z}} \exp(t\hat{\mathbf{z}}) = \exp(t\hat{\mathbf{z}})\hat{\mathbf{z}}, \quad (4.4)$$

Table 5 Examples of kinetic equations for internal variables that adopt the form 4.2

Kinetic equation	G	g	Description
$\dot{D}D^{-1} = f(\mathbf{F}, \dots) \in g$ $\dot{H}H^{-1} \in g$	\mathbb{R}^+	\mathbb{R}	Scalar damage evolution. See corresponding note on transforming $D = 1 - H$ in Table 4
$\dot{\mathbf{D}}\mathbf{D}^{-1} = \mathbf{f}(\mathbf{F}, \dots) \in g$ $\dot{\mathbf{H}}\mathbf{H}^{-1} \in g$	$GL^+(3)$	$gl(3)$	Tensor damage evolution. See corresponding note on transforming $\mathbf{D} = \mathbf{I} - \mathbf{H}$ in Table 4
$\dot{\mathbf{F}}^p \mathbf{F}^{p-1} = \dot{\varepsilon}^p \mathbf{M} \in g$	$SL(3)$	$sl(3)$	Flow rule for J_2 plasticity extended to finite deformations. $\dot{\varepsilon}^p$: equivalent plastic strain rate; \mathbf{M} : direction of plastic flow with $\text{tr } \mathbf{M} = 0, \mathbf{M} : \mathbf{M} = \frac{3}{2}$ [28]
$\dot{\mathbf{F}}^p \mathbf{F}^{p-1} = \dot{\varepsilon}^p \mathbf{M} + \dot{\theta}^p \mathbf{N} \in g$	$GL^+(3)$	$gl(3)$	Flow rule for a porous metal plasticity constitutive model for the simulation of ductile damage. $\dot{\varepsilon}^p$ and \mathbf{M} as above; $\dot{\theta}^p$: volumetric plastic deformation rate; $\mathbf{N} := \pm \frac{1}{3} \mathbf{I}$ [41]
$\dot{\mathbf{R}}\mathbf{R}^T = \mathbf{W} \in g$	$SO(3)$	$so(3)$	Kinetic equation that describes the evolution of a rotation

G and g are the corresponding Lie group and Lie algebra

in which \hat{z} is known as the *infinitesimal generator* of the subgroup $\hat{\mathbf{Z}}$. For the proofs of (4.2), (4.3) and (4.4), and further discussion see Gallier [9], Kosmann-Schwarzbach [16], Procesi [32], Sepanski [37].

4.2 Incremental updates

Kinetic equations of the form (4.1) are generally integrated in time using an incremental solution procedure with time intervals $[s, t]$, where s and t are assumed to be independent. The state of the problem is known at time s and may be updated to time t by

$$\mathbf{Z}(t) = \Delta \mathbf{Z}(\Delta t) \mathbf{Z}(s), \tag{4.5}$$

in which $\Delta t := t - s$ and $\Delta \mathbf{Z}(\Delta t)$ is an increment of the internal variable, with $\Delta \mathbf{Z}(0) = \mathbf{I}$ and $(\partial \Delta \mathbf{Z} / \partial \Delta t)(0) = \mathbf{z}(s) \in g$. The fact that (4.5) is a reasonable approximation can be seen by differentiating it with respect to t and evaluating at $t = s$, leading to

$$\frac{\partial \mathbf{Z}}{\partial t}(s) = \frac{\partial \Delta \mathbf{Z}}{\partial \Delta t}(0) \mathbf{Z}(s) \Rightarrow \frac{\partial \mathbf{Z}}{\partial t}(s) \mathbf{Z}^{-1}(s) = \mathbf{z}(s), \tag{4.6}$$

which is a kinetic equation of the form (4.2).

The increment $\Delta \mathbf{Z}(\Delta t)$ may be represented with a Taylor series as

$$\Delta \mathbf{Z}(\Delta t) = \mathbf{I} + \Delta t \mathbf{z}(s) + O[(\Delta t)^2]. \tag{4.7}$$

On the other hand, the following exponential may be expanded in a power series as

$$\exp[\Delta t \mathbf{z}(s)] = \mathbf{I} + \Delta t \mathbf{z}(s) + O[(\Delta t)^2]. \tag{4.8}$$

Thus, one possible approximation for the increment is to assume that it is a one-parameter subgroup as in (4.3)

$$\Delta \mathbf{Z}(\Delta t) \approx \exp[\Delta t \mathbf{z}(s)]. \tag{4.9}$$

As examples, the incremental updates corresponding to the kinetic equations of Table 5 are shown in Table 6.

5 Computation of the exponential and logarithmic maps

There exist different algorithms for the computation of the exponential and logarithmic maps. In the case of scalar fields, the problem is trivial. For tensors, Ortiz et al. [29] advocate the use of power series or a spectral decomposition, depending on the norm of the operand. This approach, however, may converge slowly (exponential) or not at all (logarithm) when using the power series, or may require the use of complex arithmetic when using the spectral decomposition.

Computer linear algebra systems such as Matlab [40] use the scaling and squaring method combined with Padé approximants for the computation of the exponential map. This algorithm does not require the use of complex arithmetic for real tensors and is optimized for fast convergence. See Higham [11] and references therein.

Similar algorithms exist for the logarithmic map, such as the inverse square and scaling method with Padé approximants [10]. This algorithm, however, relies on the computation of the square root of a tensor. This in turn requires the computation of the Schur decomposition, which in general is complex.

Another complication arises from the fact that a real matrix with real negative eigenvalues has a real logarithm if and only if it has an even number of Jordan blocks of each size for every real negative eigenvalue. See Higham [12, Chapters 1 & 11]. A tensor $\mathbf{A} \in GL^+(3)$ with real eigenvalues

Table 6 Examples of kinetic equations for internal variables and their corresponding incremental updates

Increments take the form of one-parameter subgroups (4.3)

Kinetic equation	Incremental update
$\dot{D}D^{-1} = f(\mathbf{F}, \dots), \quad D = 1 - H$	$H(t_{n+1}) = -\exp[\Delta f(\mathbf{F}_{n+1}, \dots)][1 - H(t_n)]$
$\dot{D}D^{-1} = f(\mathbf{F}, \dots), \quad D = \mathbf{I} - \mathbf{H}$	$\mathbf{H}(t_{n+1}) = -\exp[\Delta f(\mathbf{F}_{n+1}, \dots)][\mathbf{I} - \mathbf{H}(t_n)]$
$\dot{\mathbf{F}}^p \mathbf{F}^{p-1} = \dot{\varepsilon}^p \mathbf{M}$	$\mathbf{F}_{n+1}^p = \exp(\Delta \varepsilon^p \mathbf{M}) \mathbf{F}_n^p$
$\dot{\mathbf{F}}^p \mathbf{F}^{p-1} = \dot{\varepsilon}^p \mathbf{M} + \dot{\theta}^p \mathbf{N}$	$\mathbf{F}_{n+1}^p = \exp(\Delta \varepsilon^p \mathbf{M} + \Delta \theta^p \mathbf{N}) \mathbf{F}_n^p$
$\dot{\mathbf{R}} \mathbf{R}^T = \mathbf{W}$	$\mathbf{R}_{n+1} = \exp(\Delta \mathbf{W}) \mathbf{R}_n$

has either three positive eigenvalues or one positive and two negative eigenvalues. Thus, for the latter case there is no real logarithm.

It is highly desirable to use algorithms that do not require complex arithmetic. One way to achieve this is to compute the polar decomposition of the tensor and obtain the logarithms of the rotation and stretch components separately, interpolate these, and then apply the exponential and recombine.

5.1 Logarithmic map

There exist explicit formulas for computing the logarithm of a rotation [30]. The angle of rotation is given by

$$\theta := \cos^{-1} \left[\frac{1}{2} (\text{tr } \mathbf{R} - 1) \right], \quad \mathbf{R} \in SO(3), \quad \theta \in [0, \pi], \tag{5.1}$$

and the logarithm of the rotation is determined by

$$\log \mathbf{R} = \begin{cases} \mathbf{0} \in so(3), & \text{if } \theta = 0; \\ \frac{\theta}{2 \sin \theta} (\mathbf{R} - \mathbf{R}^T) \in so(3), & \text{if } \theta \in (0, \pi); \\ \pm \pi \check{\mathbf{v}} \in so(3), & \text{if } \theta = \pi; \end{cases} \tag{5.2}$$

in which $\check{\mathbf{v}} \in so(3)$ is the skew-symmetric tensor such that $\check{\mathbf{v}} \cdot \mathbf{u} \equiv \mathbf{v} \times \mathbf{u} \forall \mathbf{u} \in \mathbb{R}^3$, \mathbf{v} is the eigenvector corresponding to the eigenvalue of 1 of \mathbf{R} , and the sign is selected according to continuity conditions from the field in the neighborhood.

It proves convenient to compute the polar decomposition by means of the singular value decomposition (SVD) as follows: given $\mathbf{A} \in GL^+(3)$, then

$$\begin{aligned} \mathbf{A} &= \mathbf{U} \mathbf{D} \mathbf{V}^T, & \mathbf{U}, \mathbf{V} &\in SO(3), \quad \mathbf{D} = \text{diag}(s_1, s_2, s_3), \\ \mathbf{R} &:= \mathbf{U} \mathbf{V}^T, & \mathbf{R} &\in SO(3), \\ \mathbf{S} &:= \mathbf{V} \mathbf{D} \mathbf{V}^T, & \mathbf{S} &\in SPD(3) \\ \mathbf{A} &= \mathbf{R} \mathbf{S}, & &\text{polar decomposition,} \end{aligned} \tag{5.3}$$

in which $s_1, s_2, s_3 \in \mathbb{R}^+$ are the singular values of \mathbf{A} , and $SPD(3)$ is the space of three-dimensional symmetric positive-definite tensors. This is a vector space but not a Lie group, since the product of two symmetric positive-definite tensors is not necessarily symmetric. Nevertheless, one can

still use the logarithmic map as $\log: SPD(3) \rightarrow S(3)$, where $S(3)$ is the vector space of three-dimensional symmetric tensors. It follows that the exponential map can be used as well as $\exp: S(3) \rightarrow SPD(3)$ [9]. The logarithm of the rotation \mathbf{R} can be computed using (5.2) and the logarithm of the stretch tensor \mathbf{S} is readily obtained as the SVD yields its eigenvalue decomposition, *i.e.*

$$\log \mathbf{S} = \mathbf{V} (\log \mathbf{D}) \mathbf{V}^T, \tag{5.4}$$

where $\log \mathbf{D} := \text{diag}(\log s_1, \log s_2, \log s_3)$.

5.2 Exponential map

The exponential map may be treated in a similar manner as the logarithmic map. Assume the existence of two tensors $\mathbf{W} \in so(3)$ and $\mathbf{H} \in S(3)$ that are the result of interpolating, extrapolating or otherwise manipulating the polar components of a tensor field as presently described. Define

$$\theta := \left(\frac{\mathbf{W} : \mathbf{W}}{2} \right)^{\frac{1}{2}} \tag{5.5}$$

as the angle of rotation. Then the exponential map for the skew-symmetric tensor \mathbf{W} is given by the expression

$$\exp \mathbf{W} = \begin{cases} \mathbf{I} \in SO(3), & \text{if } \theta = 0; \\ \mathbf{I} + \frac{\sin \theta}{\theta} \mathbf{W} + \frac{(1 - \cos \theta)}{\theta^2} \mathbf{W}^2 \in SO(3), & \text{if } \theta > 0; \end{cases} \tag{5.6}$$

which is often accredited to Rodrigues [9]. The exponential map for the symmetric tensor \mathbf{H} may be computed by using the SVD, as in this case the eigenvalue decomposition can be readily obtained from it. Start with

$$\mathbf{H} = \mathbf{U} \mathbf{D} \mathbf{V}^T, \quad \mathbf{U}, \mathbf{V} \in SO(3), \quad \mathbf{D} = \text{diag}(s_1, s_2, s_3), \tag{5.7}$$

then the eigenvalue decomposition of \mathbf{H} is given by

$$\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T, \quad \mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3), \tag{5.8}$$

with

$$\lambda_i = \begin{cases} s_i, & \text{if } \mathbf{U} \mathbf{e}_i = \mathbf{V} \mathbf{e}_i; \\ -s_i, & \text{if } \mathbf{U} \mathbf{e}_i = -\mathbf{V} \mathbf{e}_i; \end{cases} \tag{5.9}$$

in which e_i is the canonical basis. Equivalently, the eigenvalues of H are equal to its singular values, up to the sign, which is determined by inspecting the corresponding left and right singular vectors. If the singular vectors corresponding to a singular value have the same sign, the eigenvalue is positive, otherwise it is negative. The eigenvectors are chosen to be the same as the left singular vectors. The exponential map for H then follows immediately as

$$\exp H = U(\exp \Lambda)U^T, \tag{5.10}$$

where $\exp \Lambda := \text{diag}(\exp \lambda_1, \exp \lambda_2, \exp \lambda_3)$, and which confirms that the exponential map maps symmetric tensors to symmetric positive-definite tensors. Finally, the original field is recovered by $A = \exp W \exp H$.

6 Numerical examples

In this section we demonstrate the performance of the aforementioned techniques by means of three numerical examples. The first case pertains to the L_2 projection alone, in which we compare the recovered internal variable fields obtained from both global and element-by-element projections to exact solutions. The second example concerns interpolation and extrapolation alone applied to the bending of a straight beam into a ring, for which the exact solution is available. Comparisons are made of fields recovered using direct and Lie algebra interpolation and extrapolation. Finally, both the projection and interpolation are combined and used in simulations of a large deformation J_2 elasto-plasticity problem to demonstrate their performance with respect to a number of other commonly-used internal variable recovery schemes.

6.1 Projection of discontinuous scalar fields

We consider a simple example to demonstrate the effects of the L_2 projection separate from interpolation and extrapolation. A cubic body of size $2 \times 2 \times 2$ is centered at the origin of a Cartesian coordinate system $\{XYZ\}$. The cube is discretized by 8 hexahedral trilinear elements of equal size. The following scalar field is prescribed at the eight standard Gauss quadrature points in each finite element,

$$z(X) = \begin{cases} X + 1, & \text{if } X \in [-1, 0); \\ X, & \text{if } X \in [0, 1]. \end{cases} \tag{6.1}$$

The values of the field at the quadrature points are projected onto the nodes to obtain a mapped field $\bar{z}_h(X)$. Two different projection methods are used: the first is to apply the global L_2 projection as expressed in (2.10); the second is to restrict the application of this projection to a single element, thus

recovering a local projection field where inter-element discontinuities may exist.

The projected field is shown in Fig. 2. The global L_2 projection leads to a smooth linear field $\bar{z}_h(X) = \frac{X}{4} + \frac{1}{2}$, which is identical to the linear least squares regression of the nodal values of the original field $z(X)$. This confirms that the projection defined in (2.10) minimizes the error measured by the L_2 norm $\|\bar{z}_h - z\|$ and is therefore optimal in this norm. Note that the jump condition is not preserved when the global L_2 projection method is used, as both the trial and weighing space are C^0 -continuous globally.

The local element-by-element approach, on the other hand, preserves the jump condition and reproduces the exact field in this example, as the local projection minimizes the L_2 norm as measured in each element, and the original linear scalar field can be reproduced exactly with trilinear interpolation functions.

This introductory example illustrates that different projection strategies may be derived from the same variational statement. The choice of the trial and weighting space should be made by considering the properties and nature of the field under consideration.

6.2 Bending of a beam into a ring

This example demonstrates the use of Lie groups and their corresponding Lie algebras as an aid to improve the interpolation of a tensor field. An initially straight beam of dimensions $L \times \frac{L}{16} \times \frac{L}{16}$ is bent into a ring with neutral-axis radius $R = \frac{L}{2\pi}$, with the height and depth remaining constant throughout the deformation. In the deformed configuration, the original centroid of the beam remains at the same position and the two ends meet, forming a ring. The exact deformation gradient F expressed in matrix notation with respect to the canonical basis e_i reads

$$[F(X, Y)]_{e_i} = \begin{pmatrix} \frac{R-Y}{R} \cos(\frac{X}{R}) & -\sin(\frac{X}{R}) & 0 \\ \frac{R-Y}{R} \sin(\frac{X}{R}) & \cos(\frac{X}{R}) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.2}$$

The right polar decomposition of the deformation gradient is

$$F(X, Y) = R(X) S(Y), \tag{6.3}$$

where

$$[R(X)]_{e_i} = \begin{pmatrix} \cos(\frac{X}{R}) & -\sin(\frac{X}{R}) & 0 \\ \sin(\frac{X}{R}) & \cos(\frac{X}{R}) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad [S(Y)]_{e_i} = \begin{pmatrix} \frac{R-Y}{R} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.4}$$

The beam is discretized with trilinear hexahedra into two meshes: a coarse one of 16 elements and a fine one of 2,500

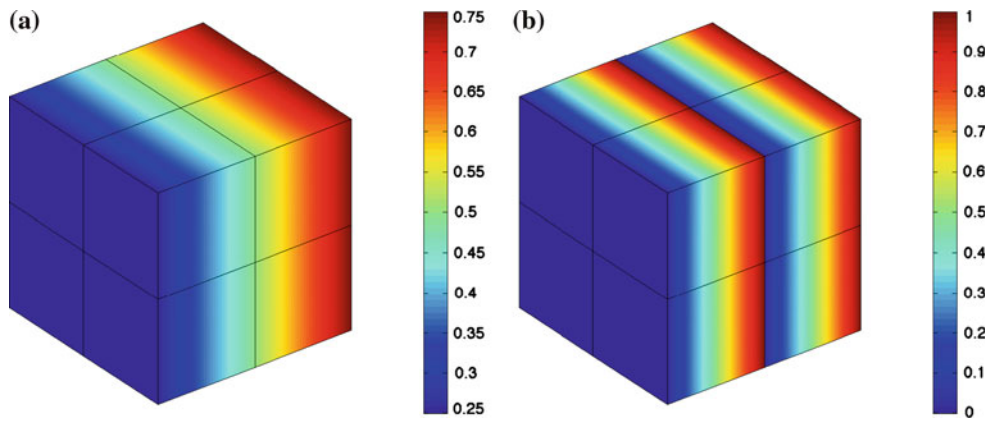


Fig. 2 Projected scalar internal variable field: **a** global L_2 projection, **b** local L_2 projection

elements. The deformation gradient $\mathbf{F}(X, Y)$ is first computed via (6.2) at the nodes of the coarse mesh. Using the element interpolation functions of the coarse mesh, we obtain values of the deformation gradient $\mathbf{F}^h(X, Y)$ at the position of the nodes of the fine mesh and compare them with the exact deformation gradient $\mathbf{F}(X, Y)$ given by (6.2).

Four different schemes are used: (1) direct interpolation of both the rotation and the stretch; (2) Lie algebra interpolation of the rotation and direct interpolation of the stretch; (3) Lie algebra interpolation of both the rotation and the stretch; and (4) Lie algebra interpolation of the deformation gradient without polar decomposition. In the first three schemes, the polar decomposition is applied to the deformation gradient and thus the rotation and stretch are interpolated separately. The polar decomposition is not used in the last scheme and therefore the logarithmic map is applied directly to the deformation gradient. This serves to illustrate the difficulties encountered when applying the logarithmic map to tensor fields with negative eigenvalues.

The exact form of the deformation gradient is available, therefore it is possible to quantify the distribution of the interpolation error by comparing directly the interpolated deformation gradient $\mathbf{F}^h(X, Y)$ with the exact solution $\mathbf{F}(X, Y)$ via the Frobenius norm, *i.e.*

$$\begin{aligned}
 e(X, Y) &= \|\mathbf{F}^h(X, Y) - \mathbf{F}(X, Y)\|_F \\
 &= \left\{ [\mathbf{F}^h(X, Y) - \mathbf{F}(X, Y)] : [\mathbf{F}^h(X, Y) \right. \\
 &\quad \left. - \mathbf{F}(X, Y)] \right\}^{\frac{1}{2}}. \tag{6.5}
 \end{aligned}$$

We now show that the use of Lie groups and Lie algebras together with the polar decomposition in the interpolation reduce the error significantly.

The results of using direct interpolation for both the rotation and the stretch are shown in Fig. 3a. The interpolation error $e(X, Y)$ is zero at the locations where the nodes of the coarse and fine meshes coincide, as expected. By con-

trast, there is substantial discrepancy between $\mathbf{F}^h(X, Y)$ and $\mathbf{F}(X, Y)$ away from these locations and the error reaches a maximum $e_{\max} = 0.1185$ at radial planes that pass through the center of each coarse hexahedral element.

Next are the results of using Lie algebra interpolation for the rotation and direct interpolation for the stretch, shown in Fig. 3b. There is a significant improvement of accuracy when using this scheme, as the maximum error $e_{\max} = 1.21 \times 10^{-15}$ is reduced by 14 orders of magnitude.

Figure 3c shows the results of applying Lie algebra interpolation for both the rotation and the stretch. The maximum value of the interpolation error now becomes $e_{\max} = 0.0177$, which falls in between the values of the two cases discussed above. This is due to the fact that the stretch given by (6.4) varies linearly along the Y direction. Thus, direct linear interpolation can reproduce the stretch exactly, while Lie algebra interpolation cannot. This example demonstrates that the appropriate choice of interpolation scheme depends on the nature of the field. Note that in general, however, the stretch is unlikely to have this simple form and linear extrapolation may result in stretches that are not positive-definite.

Finally, the results of interpolation without the use of the polar decomposition are shown in Fig. 3d. The maximum value of the error is $e_{\max} = 2.2$, which is located near the top of the ring. The reason for this severe error is that the large rotations encountered near the top of the ring lead to deformation gradients with negative eigenvalues, for which no real logarithm exists. The algorithm for the computation of the logarithmic map returns a complex, non-principal logarithm, resulting in the loss of the bijection between the corresponding Lie group and Lie algebra [12].

6.2.1 The effect of mesh refinement

We investigate the effect of mesh refinement on the overall accuracy of the three interpolation schemes that use the polar decomposition mentioned above. The error measure defined

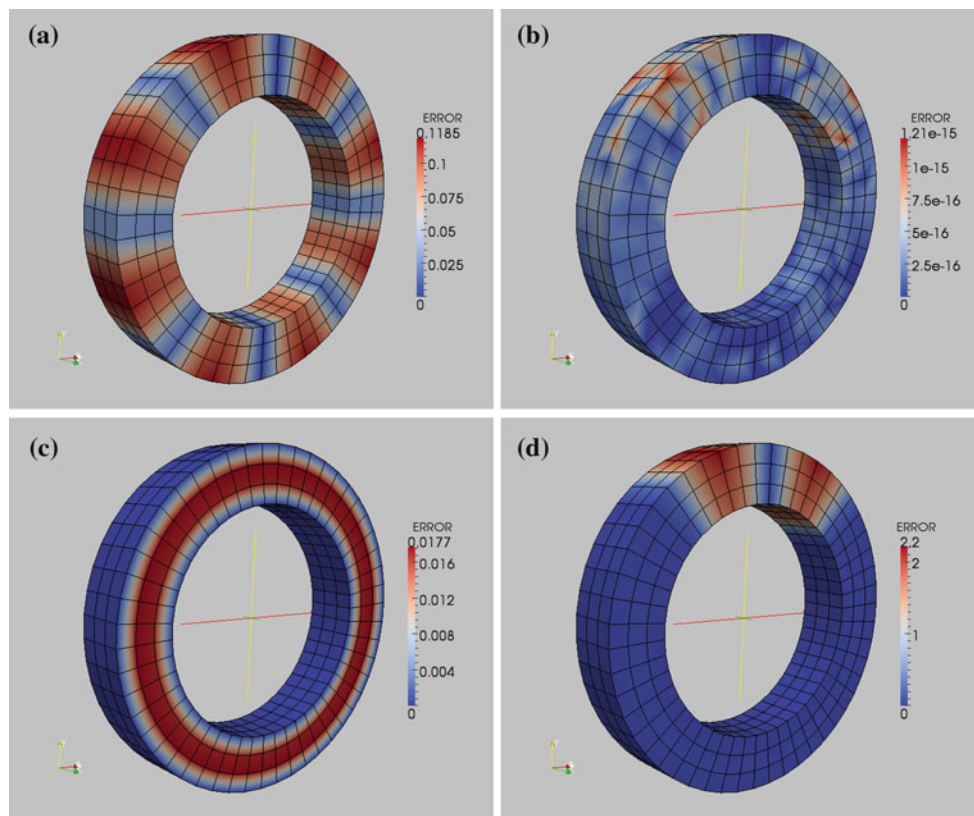


Fig. 3 Interpolation error (6.5) for different interpolation schemes: **a** direct interpolation of rotation and stretch. **b** Lie algebra interpolation of rotation and direct interpolation of stretch. **c** Lie algebra interpolation of rotation and stretch. **d** Lie algebra without polar decomposition

by the norm (9.2) on the tensor field A is evaluated numerically at a variety of refinement levels, *i.e.*

$$\begin{aligned}
 E_A &= \left(\int_B \|A^h(\mathbf{X}) - A(\mathbf{X})\|_F^2 dV \right)^{\frac{1}{2}} \\
 &\approx \left(\sum_{j=1}^{N_E} \sum_{i=1}^{N_G} \|\lambda_\alpha(\xi_i) A_\alpha - A(\xi_i)\|_F^2 w(\xi_i) J(\xi_i) \right)^{\frac{1}{2}}
 \end{aligned}
 \tag{6.6}$$

where A can be any of F , R or S , E_A is the error corresponding to the interpolated tensor field A , ξ_i is the integration point in the parametric domain, $w(\xi_i)$ is the weight of the Gauss quadrature, $J(\xi_i)$ is the Jacobian determinant of the isoparametric mapping, N_G is the number of integration points per element, and N_E is the number of elements in the mesh.

Initially, the domain of size $L \times \frac{L}{16} \times \frac{L}{16}$ is meshed by four elements along the X (length) direction and one element along the Y (height) and Z (width) directions. The aspect ratio of length divided by height of all the elements is therefore $\frac{l}{h} = 4$. Then, we refine the mesh by increasing the number of elements in both the X and Y directions, but maintain the number of elements along the Z direction equal to

one. In each refined mesh, the aspect ratio remains constant and equal to four for all elements.

Figures 4a, b, c show the convergence curves for the three interpolation schemes. The curves correspond to: direct interpolation for both rotation and stretch (red), Lie algebra interpolation for the rotation and direct interpolation for the stretch (blue), and Lie algebra interpolation for both the rotation and the stretch (green).

The behavior exhibited by the curves reflects the ability of each interpolation scheme to reproduce the corresponding exact field. For the ring configuration, the logarithm of the rotation can be readily computed using (5.2), while the logarithm of the stretch is trivial to compute in the canonical basis e_i . Thus, from (6.4) we obtain

$$[\log R(X)]_{e_i} = \begin{pmatrix} 0 & -\frac{X}{R} & 0 \\ \frac{X}{R} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad [\log S(Y)]_{e_i} = \begin{pmatrix} \log(\frac{R-Y}{R}) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.
 \tag{6.7}$$

It is immediately apparent from (6.4) and (6.7) that, as the logarithm of the rotation and the stretch are both linear with respect to the reference coordinates, the best scheme for this case is to use Lie algebra interpolation for the rotation and direct interpolation for the stretch. This is clearly confirmed by Fig. 4a, b, c. The interpolation error of the

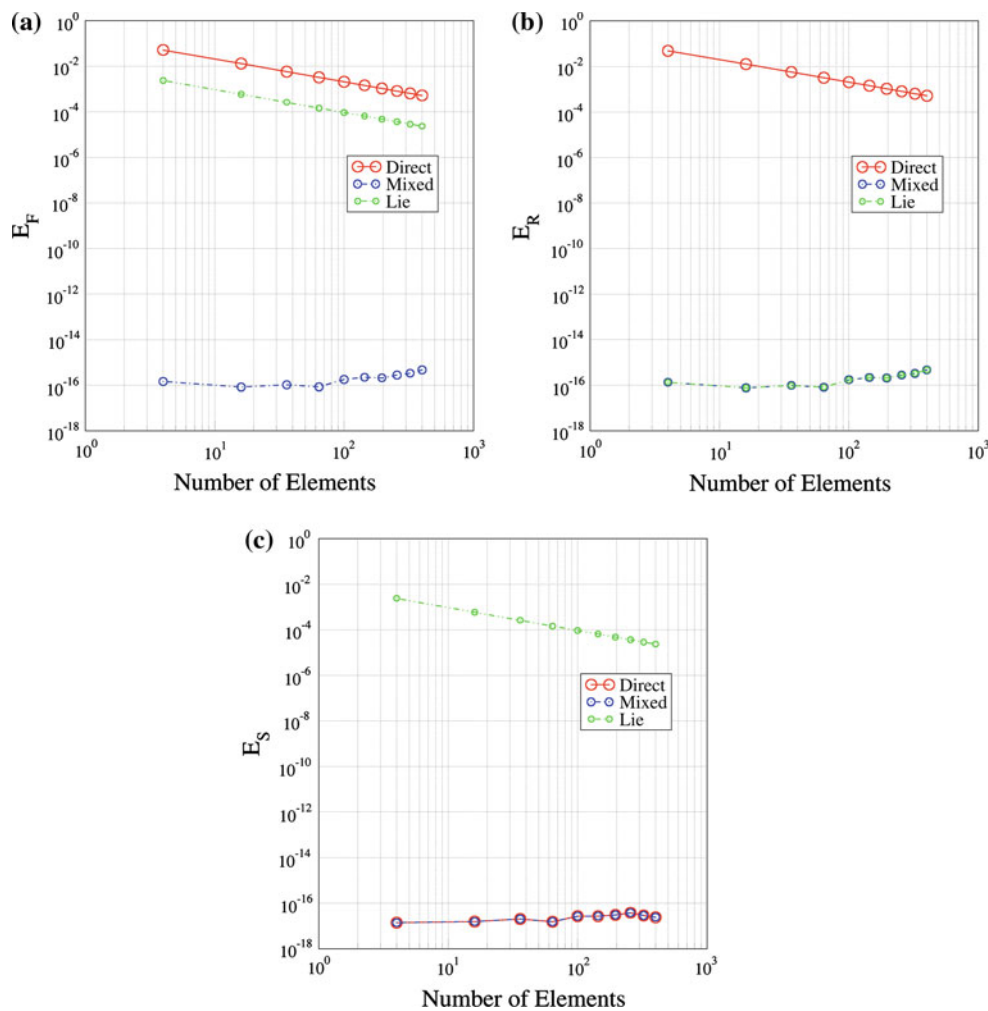


Fig. 4 Convergence plots for the ring bending problem: error in **a** deformation gradient, **b** rotation and **c** right stretch

deformation gradient is shown in Fig. 4a, which shows linear convergence for direct interpolation, linear convergence for Lie algebra interpolation, and exact representation, up to machine precision, for mixed interpolation. Convergence on the rotation is examined further in Fig. 4b, in which both schemes that use Lie algebra interpolation for the rotation are exact, up to machine precision. Conversely, convergence for the stretch is exact up to machine precision when using direct interpolation, as shown in Fig. 4c.

Note that Lie algebra interpolation for both rotation and stretch has less error than direct interpolation for both rotation and stretch, despite the fact that the stretch is linear with respect to the reference coordinates.

6.2.2 Comparison with other recovery schemes

Our objective is to compare the recovery of internal variables in three dimensions; relevant candidate methods for comparison, however, do not address this situation [5, 14, 34]. Therefore, we now compare the performance of the variational

projection with respect to other simpler, customary transfer schemes. To this end, the values of the exact deformation gradient (6.2) are computed at the integration points of a 16-element coarse mesh and then transferred to the nodes by means of various recovery methods. The assumption is that once nodal values are obtained, the element interpolation functions are used to extend the field over the entire domain for the purpose of mapping it to a different mesh. The schemes are summarized as follows:

1. Direct averaging that employs a searching scheme to determine the closest integration point to a node in each element. The value of the closest integration point is then assigned to the node. If the same node is shared by more than one element, then the integration point values are averaged among the elements attached to that node.
2. Extrapolation. Additional interpolation functions are established using the integration points as local nodes, then the fields at the integration points are extrapolated

Table 7 Minimum and maximum error (6.5) at the nodes and global error (6.6) of various recovery methods for the bending ring problem

Method	$e(\mathbf{X}_\alpha)$		E_F
	min	max	
1. Direct averaging	6.34×10^{-2}	1.41×10^{-1}	8.70×10^{-3}
2. Extrapolation	6.62×10^{-2}	6.96×10^{-1}	5.08×10^{-2}
3. Variational (polar, direct)	1.68×10^{-2}	1.99×10^{-2}	2.01×10^{-4}
4. Variational (no polar, direct)	1.68×10^{-2}	1.99×10^{-2}	2.01×10^{-4}
5. Variational (polar, mixed)	5.38×10^{-16}	2.45×10^{-13}	3.30×10^{-16}
6. Variational (polar, both Lie)	7.80×10^{-3}	8.70×10^{-3}	3.35×10^{-16}

to the actual nodes of each finite element. The nodal values are then averaged as in the previous scheme.

3. Variational projection with polar decomposition and direct interpolation.
4. Variational projection with no polar decomposition and direct interpolation.
5. Variational projection with polar decomposition, Lie algebra interpolation for the rotation and direct interpolation for the stretch.
6. Variational projection with polar decomposition and Lie algebra interpolation for both the rotation and stretch.

The minimum and maximum error (6.5) at the nodes as well as the global error (6.6) of the six recovery procedures are compared in Table 7.

The table shows that the four variational projection cases result in less error than both the direct averaging and extrapolation methods, regardless of whether Lie algebra interpolation is employed. This is consistent with the proposition in Sect. 2 in which the variational projection is proved to minimize the L_2 norm. Furthermore, the global error measured by E_F is reduced by a minimum of 12 orders of magnitude when the interpolation for the rotation is conducted in $so(3)$ instead of $SO(3)$.

Even though the local error measure at the nodes $e(\mathbf{X}_\alpha)$ is lower in the variational approach with mixed interpolation than in the variational approach with full Lie algebra interpolation, the reduction in global error E_F is within the same order. This is explained by recalling that the values of the field at the integration points are computed using the exact expression (6.2). When projecting to the nodes using Lie algebra interpolation, some error is introduced into the stretch, as discussed before. This error is manifested in the values of the nodal local error $e(\mathbf{X}_\alpha)$. In the computation of the global error E_F , however, the nodal quantities are used to compute new values at the integration points. These new integration point values are very close to the original ones computed using the exact expression, hence the low values for the global error.

Table 8 Material parameters for cylindrical billet

Young's modulus	$E = 1.0$ GPa
Poisson's ratio	$\nu = 0.3$
Yield stress	$\sigma_y = 1.0$ MPa
Hardening modulus	$H = 3.0$ MPa

6.3 Elasto-plastic upsetting of an axisymmetric billet

This example demonstrates the ability of the variational projection to maintain internal variables in their admissible space. The variational projection operator and the Lie algebra interpolation techniques are integrated into a unified framework. The simulation is a severe deformation test problem proposed by Krieg and Krieg [17] and further examined by Taylor and Becker [39] and Simo and Hughes [38]. For completeness, we briefly review the statement of the problem below.

Consider a cylindrical billet with initial radius $r = 10$ mm and initial height $h = 30$ mm. Due to radial symmetry, only one eighth of the domain is included in the calculation. The top of the cylinder is fixed by a roller such that it does not move vertically but is free to expand horizontally. The bottom of the cylinder is fixed horizontally, and a prescribed vertical displacement is applied to it. The constitutive response of the specimen is simulated via a J_2 elasto-plastic model with linear isotropic hardening. The material parameters used for this calculation are those listed in Taylor and Becker [39] and Simo and Hughes [38] (see Table 8).

Four finite element simulations are performed using the Albany code developed at Sandia National Laboratories as part of the LCM (Laboratory for Computational Mechanics) project [36]. To preserve the radial symmetry of the cylinder, hexahedral meshes are generated via a radial meshing algorithm in Sandia's CUBIT [35] mesh-generation software in such a manner that node distribution is unbiased along the radial direction. 8-node and 27-node Lagrange hexahedral elements are both used on two meshes consisting of 693 and 2175 elements. The final configuration in all four simulations is attained using adaptive loading steps. Figure 5

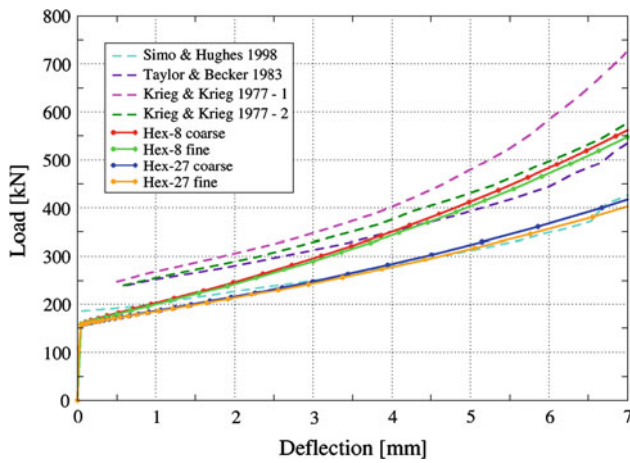


Fig. 5 Load-deflection curve of the elastic-plastic upsetting of an axisymmetric billet

shows the resulting load-deflection curves and their good agreement with existing curves [17,38,39]. The results of Krieg and Krieg [17], Taylor and Becker [39] and Simo and Hughes [38] are shown in dashed lines without markers as the corresponding curves were simply digitized from these references. The results of our simulations are plotted with continuous lines and markers for the data points for clarity.

6.3.1 Admissible range of isochoric plastic response

The simulations are advanced until the billet has reached an upsetting of 53%, just prior to contact between the deformed shaft of the cylinder and the plane of the prescribed displacement boundary condition. The plastic deformation in this configuration is significant, and thus the plastic part of the deformation gradient $F^P = (F^e)^{-1}F$ computed at each integration point is extended to the entire domain by means of the variational projection. Figure 6 shows the extended F^P

fields of the two 8-node element meshes using direct interpolation. The color in the figure represents the determinant of F^P .

The flow rule for J_2 plasticity is

$$\dot{F}^P F^{P-1} = \dot{\epsilon}^P M \in sl(3), \tag{6.8}$$

in which $\dot{\epsilon}^P$ is the equivalent plastic strain rate and M is the direction of plastic flow, with $\text{tr } M = 0$ and $M : M = \frac{3}{2}$ [28]. This flow rule is a kinetic equation of the form (4.2), and therefore it is required that the plastic part of the deformation gradient be isochoric, or equivalently that its determinant be equal to one, i.e. $F^P \in SL(3)$ as stated in Table 5.

The simulation results indicate, however, that this is not necessarily the case when the L_2 projection is performed without transforming F^P to its corresponding Lie algebra for interpolation, thus violating one of the fundamental assumptions of J_2 plasticity. By using direct interpolation for F^P , spurious volumetric plastic deformations are introduced into the extended field. This is evident in Fig. 6, in which the determinant of the extended plastic part of the deformation gradient is in the range $\det F^P \in [0.83, 1.24]$ for the coarse mesh, and $\det F^P \in [0.64, 1.38]$ for the fine mesh.

This severe error occurs when the mesh is subjected to a significant amount of distortion at the edge of the bottom face, as shown in Fig. 6a. The spurious plastic volumetric strain is not eliminated through refinement, as the significant distortion of the finite element mesh is present regardless of mesh size.

If the projection of the plastic part of the deformation gradient F^P is conducted through Lie algebra interpolation, then it remains isochoric as illustrated in Fig. 7. This projection is obtained by first decomposing F^P into rotation and right stretch tensors via the singular value decomposition as described in (5.3). Then, the rotation tensor is mapped from $SO(3)$ into $so(3)$ using the logarithmic map (5.2) and the stretch is mapped from $GL^+(3)$ to $gl(3)$ by means of the

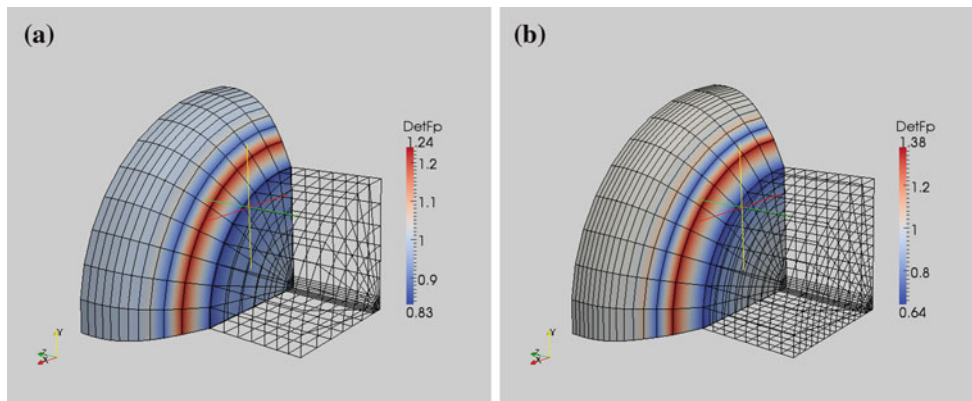


Fig. 6 Plastic part of the deformation gradient by direct interpolation for 8-node hexahedral elements: **a** 693-element mesh and **b** 2175-element mesh

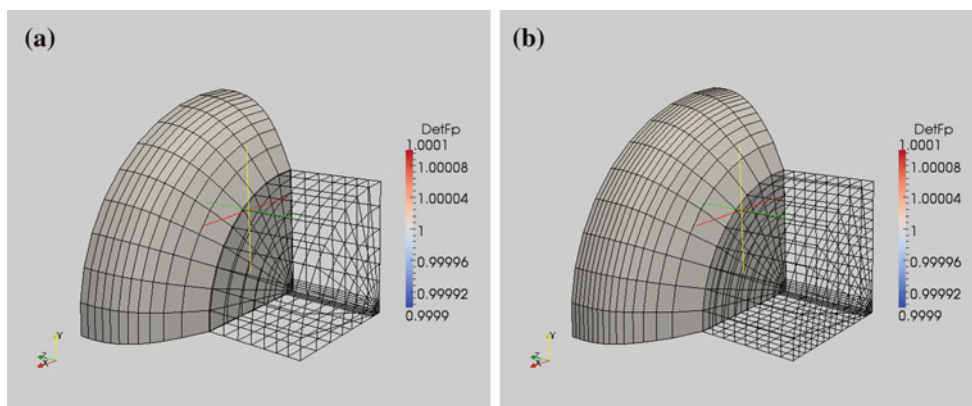


Fig. 7 Plastic part of the deformation gradient by Lie group and Lie algebra interpolation for 8-node hexahedral elements: **a** 693-element mesh and **b** 2175-element mesh

logarithmic map (5.4). Note that $SPD(3) \subset GL^+(3)$ and $S(3) \subset gl(3)$.

By applying the projection (2.10) on $so(3)$ and $gl(3)$ where the addition operation is valid, the projected field is guaranteed to remain in the admissible space. The projected fields are then transformed from their Lie algebras $so(3)$ and $gl(3)$ back to their Lie groups $SO(3)$ and $GL(3)$ by means of (5.6) and (5.8), respectively. The extended plastic part of the deformation gradient is then recovered by $F^P = RS$ at any point of the finite element mesh. The results obtained from this calculation show that the determinant of the plastic part of the deformation gradient is equal to one (up to machine precision) everywhere in the domain. No spurious dilation or contraction is introduced due to the projection from integration points to nodes. This in effect prevents the injection of non-physical data into the simulation when remeshing or post-processing take place. Figure 7 shows that the isochoric constraint is preserved for both the coarse and the fine meshes.

6.3.2 Comparisons with other internal variable recovery schemes

We compare the performance of the same set of internal variable recovery schemes discussed in Sect. 6.2.2 but now using the upsetting billet example. The plastic part of the deformation gradient F^P at the integration points of the coarse mesh is first projected to the nodes. Using these nodal values and the element interpolation functions, the field is extended over the entire domain, and this extended field is used to compute the global error as defined in (6.6). Table 9 shows the results. Among the six internal variable recovery schemes, only the ones using Lie algebras are able to preserve the isochoric constraint of the J_2 plasticity model. As before, in the computation of the global error E_F , the nodal quantities are used to compute new values at the integration points. The new integration point values are very close

Table 9 Comparison of various internal variable recovery methods for the upsetting billet problem

Method	$\det F^P$		E_F
	min	max	
1. Direct averaging	1.00	1.24	9.35×10^{-9}
2. Extrapolation	0.95	1.10	1.56×10^{-8}
3. Variational (polar, direct)	0.89	1.10	7.36×10^{-9}
4. Variational (no polar, direct)	0.83	1.24	7.40×10^{-9}
5. Variational (polar, mixed)	1 ^a	1 ^a	7.37×10^{-9}
6. Variational (polar, both Lie)	1 ^a	1 ^a	7.07×10^{-9}

^a Up to machine precision

to the original ones, resulting the low values for the global error.

Despite low values of the global error, it is clear that using direct interpolation leads to large errors in the range of $\det F^P$, and therefore any mapping of F^P to a new mesh that relies in direct interpolation will introduce large errors into the field. Only the use of Lie algebra interpolation guarantees that the isochoric constraint will be preserved when mapping the field to a new mesh.

7 Discussion

As mentioned in the Appendix, the three-field finite element formulation (2.2) leads to an expression for the discrete target field \bar{z}_h that is a projection of the source field z onto the space V_h spanned by the basis functions λ_α . As with all projections, it is then evident that the error (9.11) can only be zero if and only if the source field is already an element of this space, *i.e.* $z \in V_h$. Conversely, by the idempotence property of projection operators, projecting the discrete target field \bar{z}_h results in zero error. This is illustrated by the cube example of Sect. 6.1, where it is shown that a discontinuous field cannot be recovered by projecting it to a C^0 continuous space.

The projection operator in effect extends the field of source internal variables \mathbf{z} , which only exists at integration points, to the entire domain. It does this by computing nodal values of the internal variables (2.12) and using the element interpolation functions, thus allowing the computation of the field at any point in the domain, for example the location of the integration points of a different discretization.

Once the source internal variables have been projected by (2.10), the three-field functional (2.2) requires that the energy density be evaluated using the discrete target field. This implies that the equilibrium condition (2.8) must be evaluated with the discrete target field as well. Assume that the equilibrium condition is satisfied with the source field. In general the projection error (9.11) is not zero and therefore it is expected that there will be a residual force when evaluating (2.8) with the discrete target field. Provided that (2.8) is stable, however, this residual force will be small if the error (9.11) is also small.

If this residual force is unacceptable, equilibrium can be restored by applying whichever solution scheme is being employed for the simulation. Two options are available: maintain the internal variables constant and iterate on the deformation mapping or vice versa. The use of any of these options and their effects on the residual, however, is beyond the scope of this work.

The bending ring example of Sect. 6.2 demonstrates that the appropriate use of Lie algebras in the interpolation results in a significant reduction of the interpolation error. For this particular example, however, the stretch is linear with respect to the reference height, and therefore direct interpolation is able to reproduce the stretch exactly. This example illustrates the importance of the proper selection of interpolation scheme, in particular if further information is available about the nature of the field, as is the case for the ring. In general, however, the stretch is unlikely to have this simple linear form, and direct extrapolation may lead to tensors that are not positive-definite.

There are non-smooth fields of internal variables for which smooth mapping methods, such as the present one, do not apply directly. The projected discrete target field $\bar{\mathbf{z}}_h(\mathbf{X})$ in (2.12) has the same continuity conditions as the interpolation functions $\lambda_\alpha(\mathbf{X})$ in (2.7). Consider for example the flow rule for J_2 plasticity (6.8). This is a non-holonomic constraint imposed on the plastic part of the deformation gradient \mathbf{F}^P , and thus the equivalent plastic strain ε^P can only be found by means of an incremental update of the type shown in Table 6. The Kuhn-Tucker optimality conditions require that $\dot{\varepsilon}^P \geq 0$ and therefore $\varepsilon^P \geq 0$ [38]. Thus, the equivalent plastic strain ε^P is a non-smooth (non-analytic, in fact) field that does not belong to a group, as one of the requirements for groups is smoothness. It can still be projected and interpolated directly, but there is no guarantee that the projection will give rise to

values of the field greater or equal to zero everywhere in the domain.

8 Conclusion

The proposed recovery operator is composed of two parts: the variational L_2 projection that minimizes the error; and the mapping of fields to Lie algebras for interpolation and mapping back to Lie groups.

We prove that the L_2 projection can be derived naturally from a variational formulation and that it minimizes the error between the *source* and *target* fields. We also show that by using Lie algebras many common tensor fields can be interpolated using standard finite element interpolation functions while at the same time remaining in their admissible spaces. This is of fundamental importance if the fields are required to satisfy their own constraints. Among the well-known virtues of the L_2 projection is that it does not require search algorithms and can be implemented with relative ease.

The computation of the logarithmic map leads to complex arithmetic for tensors fields with negative eigenvalues. One way to completely bypass complex arithmetic is the proposed technique of using the singular value decomposition to compute the polar decomposition, and then apply the logarithmic map separately for the rotational and stretch components.

The numerical examples emphasize different aspects of the method. First, the effect of the local and global L_2 projections on continuous and discontinuous fields is investigated. As expected, a global continuous projection approximates better a global continuous field. Next, the effect of direct and Lie algebra interpolation is considered by extending the deformation gradient field of a bending ring. The results show that avoiding the polar decomposition and using direct interpolation instead of Lie algebra interpolation introduces severe error. Finally, both the projection and interpolation are used together in a billet upsetting example, where the field of interest is the plastic part of the deformation gradient \mathbf{F}^P . It is shown here that using direct interpolation gives rise to severe error when the fields have constraints, such as the isochoric condition for J_2 plasticity. The use of Lie algebras for interpolation avoids this situation altogether, preserving the isochoric constraint up to machine precision.

We conclude that the use of these techniques provide an integrated and powerful method for projecting source fields of internal variables from integration points to the entire domain as target fields, which in turn serves the purpose of mapping them to a different discretization.

Acknowledgments Support for this work was received through the U.S. Department of Energy's (DOE) Advanced Simulation and Computing (ASC) Program at Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin

Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000. This work was also supported in part by the Joint U.S. Department of Defense (DoD)/DOE Munitions Technology Development Program.

9 Appendix: error minimization

Proposition *The projection defined by (2.10) is orthogonal with respect to V_h and therefore the distance between source and target fields is minimal in the L_2 norm of V .*

Proof Define the inner product in V as

$$(\mathbf{u}, \mathbf{v}) := \int_B \mathbf{u} \cdot \mathbf{v} \, dV, \quad \forall \mathbf{u}, \mathbf{v} \in V, \tag{9.1}$$

and the corresponding L_2 norm as

$$\|\mathbf{u}\| := \left(\int_B \mathbf{u} \cdot \mathbf{u} \, dV \right)^{\frac{1}{2}} \geq 0, \quad \forall \mathbf{u} \in V. \tag{9.2}$$

We follow the approach outlined by Brenner and Scott [4]. The variational statement (2.5) may be written after discretization as

$$\int_B (\bar{\mathbf{z}}_h - \mathbf{z}) \cdot \boldsymbol{\zeta}_h \, dV = 0 \tag{9.3}$$

where $\mathbf{z} \in V$ and $\bar{\mathbf{z}}_h, \boldsymbol{\zeta}_h \in V_h$. Using the inner product notation introduced in (9.1) this discrete statement takes the form

$$(\bar{\mathbf{z}}_h - \mathbf{z}, \boldsymbol{\zeta}_h) = 0, \quad \forall \boldsymbol{\zeta}_h \in V_h \tag{9.4}$$

which shows the fundamental orthogonality relation between the projection error $\bar{\mathbf{z}}_h - \mathbf{z}$ and the space V_h , therefore proving that (2.10) is an orthogonal projection. Introduce the Cauchy-Schwarz inequality

$$(\mathbf{u}, \mathbf{v}) \leq \|\mathbf{u}\| \|\mathbf{v}\|, \quad \forall \mathbf{u}, \mathbf{v} \in V, \tag{9.5}$$

then for any $\mathbf{u}_h \in V_h$

$$\begin{aligned} \|\bar{\mathbf{z}}_h - \mathbf{z}\|^2 &= (\bar{\mathbf{z}}_h - \mathbf{z}, \bar{\mathbf{z}}_h - \mathbf{z}) \\ &= (\bar{\mathbf{z}}_h - \mathbf{z}, \mathbf{u}_h - \mathbf{z}) + (\bar{\mathbf{z}}_h - \mathbf{z}, \bar{\mathbf{z}}_h - \mathbf{u}_h) \\ &= (\bar{\mathbf{z}}_h - \mathbf{z}, \mathbf{u}_h - \mathbf{z}) \quad \text{from (9.4) with } \boldsymbol{\zeta}_h = \bar{\mathbf{z}}_h - \mathbf{u}_h \\ &\leq \|\bar{\mathbf{z}}_h - \mathbf{z}\| \|\mathbf{u}_h - \mathbf{z}\| \quad \text{from (9.5)}. \end{aligned} \tag{9.6}$$

If $\|\bar{\mathbf{z}}_h - \mathbf{z}\| = 0$ then (9.6) is satisfied trivially and (2.10) is optimal. If $\|\bar{\mathbf{z}}_h - \mathbf{z}\| > 0$ then it follows that $\|\bar{\mathbf{z}}_h - \mathbf{z}\| \leq \|\mathbf{u}_h - \mathbf{z}\|$. As \mathbf{u}_h is any element in V_h , this inequality is also satisfied when taking the infimum

$$\|\bar{\mathbf{z}}_h - \mathbf{z}\| \leq \inf\{\|\mathbf{u}_h - \mathbf{z}\| : \mathbf{u}_h \in V_h\}, \tag{9.7}$$

and by the definition of the infimum,

$$\inf\{\|\mathbf{u}_h - \mathbf{z}\| : \mathbf{u}_h \in V_h\} \leq \|\bar{\mathbf{z}}_h - \mathbf{z}\|, \tag{9.8}$$

therefore to satisfy both (9.7) and (9.8) it follows that

$$\|\bar{\mathbf{z}}_h - \mathbf{z}\| = \inf\{\|\mathbf{u}_h - \mathbf{z}\| : \mathbf{u}_h \in V_h\}. \tag{9.9}$$

The infimum exists for some $\mathbf{u}_h \in V_h$, thus (9.9) is actually a minimum

$$\|\bar{\mathbf{z}}_h - \mathbf{z}\| = \min\{\|\mathbf{u}_h - \mathbf{z}\| : \mathbf{u}_h \in V_h\}, \tag{9.10}$$

which shows that the projection (2.10) is optimal.

An alternate approach to determine that (2.10) is optimal is to introduce the difference or error between $\bar{\mathbf{z}}_h$ and \mathbf{z} into a variational principle as

$$\Pi[\bar{\mathbf{z}}_h] := \|\bar{\mathbf{z}}_h - \mathbf{z}\|^2 = \int_B (\bar{\mathbf{z}}_h - \mathbf{z}) \cdot (\bar{\mathbf{z}}_h - \mathbf{z}) \, dV. \tag{9.11}$$

Introducing as before the test function $\boldsymbol{\zeta}_h$, and upon minimization by applying variations, this leads to

$$D\Pi[\bar{\mathbf{z}}_h](\boldsymbol{\zeta}_h) = 2 \int_B (\bar{\mathbf{z}}_h - \mathbf{z}) \cdot \boldsymbol{\zeta}_h \, dV = 0, \tag{9.12}$$

which is equivalent to (9.3) and which implies that (2.10) is optimal as it minimizes the error.

These two methods show that the variational principle (2.2) results in a target field $\bar{\mathbf{z}}_h$ that minimizes the error in the norm (9.2) with respect to the source field of internal variables \mathbf{z} . Note that the error (9.11) is zero if and only if the source field \mathbf{z} is already a member of the discrete space V_h . Thus, in general $\bar{\mathbf{z}}_h$ is not equal to \mathbf{z} , and in the evaluation of the equilibrium condition (2.6) with the target internal variables in a new mesh, there are two options: satisfy equilibrium by changing the deformation mapping and maintaining the target internal variables constant, or satisfy equilibrium by evolving the target internal variables and maintaining the deformation mapping constant.

References

1. Antman S (2005) *Nonlinear Problems in Elasticity*, 2nd edn. Springer, New York
2. Barlow J (1976) Optimal stress locations in finite element models. *Int J Numer Methods Eng* 10:243–251
3. Bleier N, Mosler J (2012) Efficient variational constitutive updates by means of a novel parameterization of the flow rule. *Int J Numer Methods Eng* 89(9):1120–1143. doi:10.1002/nme.3280. ISSN 0029–5981
4. Brenner SC, Scott LR (2002) *The mathematical theory of finite element methods*. Springer, New York. ISBN 9780387954516
5. Bucher A, Meyer A, Goerke UJ, Kreissig R (2007) A comparison of mapping algorithms for hierarchical adaptive FEM in finite elasto-plasticity. *Comput Mech* 39(4): 521–536. doi:10.1007/s00466-006-0051-z. ISSN 0178–7675
6. Cheng SH, Higham NJ, Kenney CS, Laub AJ (2001) Approximating the logarithm of a matrix to specified accuracy. *SIAM J Matrix Anal Appl* 22(4):1112–1125. ISSN 0895–4798

7. Davies PI, Higham NJ (2003) A Schur–Parlett algorithm for computing matrix functions. *SIAM J Matrix Anal Appl* 25(2):464–485. doi:[10.1137/S0895479802410815](https://doi.org/10.1137/S0895479802410815). ISSN 0895–4798
8. Fancello E, Ponthot JP, Stainier L (2006) A variational formulation of constitutive models and updates in non-linear finite viscoelasticity. *Int J Numer Methods Eng* 65(11):1831–1864. doi:[10.1002/nme.1525](https://doi.org/10.1002/nme.1525). ISSN 0029–5981
9. Gallier J (2011) *Geometric methods and applications: for Computer science and engineering*. Springer, New York. ISBN 9781441999603
10. Higham NJ (2001) Evaluating Padé approximants of the matrix logarithm. *SIAM J Matrix Anal Appl* 22(4):1126–1135. ISSN 0895–4798
11. Higham NJ (2005) The scaling and squaring method for the matrix exponential revisited. *SIAM J Matrix Anal Appl* 26(4):1179–1193. doi:[10.1137/04061101X](https://doi.org/10.1137/04061101X). ISSN 0895–4798
12. Higham NJ (2008) *Functions of matrices, theory and computation*. Society for Industrial and Applied Mathematics, Philadelphia
13. Hinton E, Campbell JS (1974) Local and global smoothing of discontinuous finite element functions using a least squares method. *Int J Numer Methods Eng* 8:461–480
14. Jiao XM, Heath MT (2004) Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *Int J Numer Methods Eng* 61(14):2402–2427. doi:[10.1002/nme.1147](https://doi.org/10.1002/nme.1147). ISSN 0029–5981
15. Jog CS (2008) The explicit determination of the logarithm of a tensor and its derivatives. *J Elast* 93(2):141–148. doi:[10.1007/s10659-008-9169-x](https://doi.org/10.1007/s10659-008-9169-x). ISSN 0374–3535
16. Kosmann-Schwarzbach Y (2009) *Groups and symmetries: from finite groups to lie groups*. Springer, New York. ISBN 9780387788654
17. Krieg RD, Krieg DB (1977) Accuracies of numerical solution methods for the elastic-perfectly plastic model. *ASME J Pres Vesel Technol* 99:510–515
18. Lambrecht M, Miehe C, Dettmar J (2003) Energy relaxation of non-convex incremental stress potentials in a strain-softening elastic-plastic bar. *Int J Solids Struct* 40(6):1369–1391. doi:[10.1016/S0020-7683\(02\)00658-3](https://doi.org/10.1016/S0020-7683(02)00658-3). ISSN 0020–7683
19. Marsden JE, Ratiu TS (1999) *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Springer, New York. ISBN 9780387986432
20. Miehe C, Lambrecht M, Gürses E (2004) Analysis of material instabilities in inelastic solids by incremental energy minimization and relaxation methods: evolving deformation microstructures in finite plasticity. *J Mech Phys Solids* 52(12): 2725–2769. doi:[10.1016/j.jmps.2004.05.011](https://doi.org/10.1016/j.jmps.2004.05.011). ISSN 0022–5096
21. Miehe C, Schotte J, Lambrecht M (2002) Homogenization of inelastic solid materials at finite strains based on incremental minimization principles. Application to the texture analysis of polycrystals. *J Mech Phys Solids* 50(10): 2123–2167. doi:[10.1016/S0022-5096\(02\)00016-9](https://doi.org/10.1016/S0022-5096(02)00016-9). ISSN 0022–5096
22. Mosler J, Bruhns OT (2010) On the implementation of rate-independent standard dissipative solids at finite strain—Variational constitutive updates. *Comput Methods Appl Mech Eng* 199(9–12):417–429. doi:[10.1016/j.cma.2009.07.006](https://doi.org/10.1016/j.cma.2009.07.006). ISSN 0045–7825
23. Mota A, Abel JF (2005) Stress recovery in geometrically nonlinear membranes by a Hu–Washizu finite element formulation. *Comput Mech* 36(3): 171–181. doi:[10.1007/s00466-004-0650-5](https://doi.org/10.1007/s00466-004-0650-5). ISSN 0178–7675
24. Muschik W, Papenfuss C, Ehrentraut H (2001) A sketch of continuum thermodynamics. *J Non-Newton Fluid Mech* 96(1–2, Sp. Iss. SI): 255–290. ISSN 0377–0257 (2nd International Workshop on Nonequilibrium Thermodynamics, Oxford, England, Aug, 2000)
25. Oden JT, Brauchli HJ (1971) On the calculation of consistent stress distributions in finite element approximations. *Int J Numer Methods Eng* 3:317–325
26. Orlando A, Peric D (2004) Analysis of transfer procedures in elastoplasticity based on the error in the constitutive equations: Theory and numerical illustration. *Int J Numer Methods Eng* 60(9): 1595–1631. ISSN 0029–5981
27. Ortiz M, Quigley JJ (1991) Adaptive mesh refinement in strain localization problems. *Comput Methods Appl Mech Eng* 90(1–3):781–804. ISSN 0045–7825
28. Ortiz M, Stainier L (1999) The variational formulation of viscoplastic constitutive updates. *Comput Methods Appl Mech Eng* 171(3–4):419–444
29. Ortiz M, Radovitzky RA, Repetto EA (2001) The computation of the exponential and logarithmic mappings and their first and second linearizations. *Int J Numer Methods Eng* 52(12):1431–1441
30. Park FC, Ravani B (1997) Smooth invariant interpolation of rotations. *ACM Trans Graph* 16(3):277–295. doi:[10.1145/256157.256160](https://doi.org/10.1145/256157.256160). ISSN 0730–0301
31. Peric D., Hochard C, Dutko M, Owen DRJ (1996) Transfer operators for evolving meshes in small strain elasto-plasticity. *Comput Methods Appl Mech Eng* 137(3–4):331–344. ISSN 0045–7825
32. Procesi C (2006) *Lie groups: an approach through invariants and representations* (Universitext), 1st edn. Springer, New York. ISBN 0387260404
33. Radovitzky R, Ortiz M (1999) Error estimation and adaptive meshing in strongly nonlinear dynamic problems. *Comput Methods Appl Mech Eng* 172(1–4):203–240. ISSN 0045–7825
34. Rashid MM (2002) Material state remapping in computational solid mechanics. *Int J Numer Methods Eng* 55(4):431–450. doi:[10.1002/nme.508](https://doi.org/10.1002/nme.508). ISSN 0029–5981
35. Sandia National Laboratories (2013a) CUBIT: Geometry and mesh generation toolkit. URL <http://cubit.sandia.gov/>
36. Sandia National Laboratories (2013b) LCM: Laboratory for computational mechanics. URL <https://software.sandia.gov/albany/>
37. Sepanski MR (2007) *Compact lie groups*. Springer, New York. ISBN 9780387302638
38. Simo JC, Hughes TJR (1998) *Computational inelasticity*. Springer, New York
39. Taylor LM, Becker EB (1983) Some computational aspects of large deformation, rate-dependent plasticity problems. *Comput Methods Appl Mech Eng* 41:251–278
40. The Mathworks, Inc. Matlab (2011) URL <http://www.mathworks.com/>. Natick, Massachusetts
41. Weinberg K, Mota A, Ortiz M (2006) A variational constitutive model for porous metal plasticity. *Comput Mech* 37:142–152
42. Wriggers P (2008) *Nonlinear finite element methods*. Springer, Berlin
43. Zienkiewicz OC, Zhu JZ (1992) The superconvergent patch recovery and a-posteriori error estimates. 1. The recovery technique. *Int J Numer Methods Eng* 33:1331–1364