



Generalised tangent stabilised nonlinear elasticity: An automated framework for controlling material and geometric instabilities

Roman Poya ^a*, Rogelio Ortigosa ^b*, Antonio J. Gil ^c, Theodore Kim ^d, Javier Bonet ^e

^a Meshing Framework, Simulation & Test Solutions, Siemens Digital Industries Software, Cambridge, United Kingdom

^b Multiphysics Simulation and Optimisation Lab, Technical University of Cartagena, Cartagena (Murcia), Spain

^c Zienkiewicz Centre for Modelling, Data and AI, Faculty of Science and Engineering, Swansea University, Bay Campus, United Kingdom

^d Computer Graphics Group, Yale School of Engineering & Applied Science, Yale University, New Haven, CT, United States

^e Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Barcelona, Spain



ARTICLE INFO

Keywords:

Nonlinear elasticity
Large strain
Material instability
Geometric instability
Tangent stabilisation
Finite elements

ABSTRACT

Tangent stabilised large strain isotropic elasticity was recently proposed by Poya et al. (2023) wherein by working directly with principal stretches the entire eigenstructure of constitutive and geometric/initial stiffness terms were found in closed-form, giving fresh insights into exact convexity conditions of highly non-convex functions in discrete settings. Consequently, owing to these newly found tangent eigenvalues an analytic tangent stabilisation was proposed (for common non-convex strain energies that exhibit material and/or geometric instabilities) bypassing incumbent numerical approaches routinely used in nonlinear finite element analysis. This formulation appears to be extremely robust for quasi-static simulation of complex deformations even with no load increments and time stepping while still capturing instabilities (similar to dynamic analysis) automatically in ways that are infeasible for path-following techniques in practice. In this work, we generalise the notion of tangent stabilised elasticity to virtually all known invariant formulations of nonlinear elasticity. We show that, closed-form eigen-decomposition of tangents is easily available irrespective of invariant formulation or integrity basis. In particular, we work out closed-form tangent eigensystems for isotropic Total Lagrangian deformation gradient (*F*)-based and right Cauchy–Green (*C*)-based as well as Updated Lagrangian left Cauchy–Green (*b*)-based formulations and present their exact convexity conditions postulated in terms of their corresponding tangent and geometric stiffness eigenvalues. In addition, we introduce the notion of geometrically stabilised polyconvex large strain elasticity for models that are materially stable but exhibit geometric instabilities for whom we construct their geometric stiffness in a spectrally-decomposed form analytically. We further extend this framework to the case of transverse isotropy where once again, closed-form tangent eigensystems are found for common transversely isotropic invariants. In this context, we augment the recent work on mixed variational formulations in principal stretches for deformable and rigid bodies, by presenting a mixed variational formulation for models with arbitrarily directed inextensible fibres. Since, tangent stabilisation unleashes an unparalleled capability for extreme deformations new numerical techniques are required to guarantee element-inversion-safe analysis. To this end, we propose a discretisation-aware load-stepping together with a line search scheme for a robust industry-grade implementation of tangent stabilised elasticity over general polyhedral meshes. Extensive comparisons with path-following

* Corresponding authors.

E-mail addresses: roman.poya@siemens.com (R. Poya), rogelio.ortigosa@upct.es (R. Ortigosa).

techniques provide conclusive evidence that utilising tangent stabilised elasticity can offer both faster and automated results.

1. Introduction

Simulating large deformation material and structural response robustly in the context of second order solvers remains a challenging problem. In recent years, nonlinear elastic models have been extensively used for newer applications where extreme deformations are encountered to a degree seldom seen in conventional engineering structures. Non-exhaustive examples of this include, designing meta-materials governed by their geometry rather than composition with multi-stable configurations [1], structural topology optimisation wherein the underlying strain energy function is forced to behave in compliance with shapes dictated by void regions and hard physical constraints [2–4], and physics-based rendering and animation wherein volumetric objects are pressed to a plane, line or even a point. Often, for these applications, obtaining the final deformation profile is more important than the yield or limit points. There is substantial evidence that current numerical approaches in the engineering simulation community fall short of modelling such phenomena [2,3,5].

Polyconvex large strain elastic models [6–10] are the most recent incarnations in robust simulation of nonlinear deformations and facilitate a systematic way for describing materially stable models (i.e. models with Positive Semi-Definite (PSD) constitutive tangent operator) while also circumventing numerical phenomena such as shear and volumetric lockings due to their enhanced albeit substantially expensive numerical treatment. However, these models by nature leave out or do not preclude geometric instabilities. Here, it serves the purposes to illustrate what is implied by these two forms of instabilities. Mathematically speaking, given a composite strain energy function $e(I(f))$ where f is in general a deformation measure and I the integrity basis [11,12], we can obtain the first and second derivatives of e with respect to f through the chain rule as

$$\frac{\partial e}{\partial f} = \frac{\partial e}{\partial I} \frac{\partial I}{\partial f}, \quad \frac{\partial^2 e}{\partial f^2} = \frac{\partial I}{\partial f} \left(\frac{\partial^2 e}{\partial I \partial I} \right) \frac{\partial I}{\partial f} + \frac{\partial e}{\partial I} \frac{\partial^2 I}{\partial f \partial f}.$$

In the above expression the first (green) term comprises of second derivative of e with respect to I and hence, represents the constitutive tangent contribution encoding sources of [material instability](#), if any, while the second (red) term only has the first derivative of e with respect to I which is usually called the geometric or initial component responsible for [geometric instability](#), if any. Virtually, all formulations of nonlinear elasticity (in fact, all metric-based optimisation formulations) possess these two components. Physically, the names material and geometric instabilities come from materials that exhibit strain softening such as brittle failure in geomechanics and snap-through and snap-back behaviour of thin-walled and engineering structures in structural analysis. As mentioned, for polyconvex energy functions the constitutive component of the tangent is always PSD hence, they are called “materially-stable” models. For convex energy functions both constitutive and geometric (if any) components are PSD however, such models are often too limiting [13]. In the general 3-dimensional setting, eliminating such sources of instability from strain energy functions requires an *a posteriori* *numerical* eigen-decomposition of both constitutive tangent and geometric stiffness operators and subsequently clamping the negative eigenvalues to zero (or taking their absolute value as in [14] for a bigger stabilisation). This is often called positive semi-definite projection of tangents [12,15,16]. Interestingly, aside from specialised applications, this approach has seldom or rather never been followed in nonlinear analysis of solids in the engineering community as it amounts to “approximation” of the two derivatives which in turn deteriorates convergence of second order solvers such as Newton-Raphson (NR). In engineering simulations, Newton-Raphson with continuation (i.e. applying the load in increments) remains the most well-established nonlinear solution procedure.

Recently, in Poya et al. [13] the authors introduced a new way of looking at large strain elasticity by using the principal stretch information obtained via the Singular-Value-Decomposition (SVD) of the deformation gradient. Using this information alone the complete eigensystem of the tangent operators (constitutive and geometric) was worked out explicitly and in *closed-form*. Consequently, a new tangent stabilisation technique was proposed to circumvent indefiniteness of the global tangent operator and in turn non-convergence of Newton-Raphson when faced with material and/or geometric instabilities. This technique involves fast per quadrature-point projection of the local tangents to positive semi-definite cone using analytically constructed tangent eigenvalues. The resulting second order minimisation approach is the so-called analytically Projected Newton-Raphson (PN or P-NR) [13,17]. In the following sections, you may notice that the terms P-NR and tangent stabilisation are used interchangeably. P-NR even in quasi-static regimes, behaves like dynamic analysis without adding any inertia effect. While the quadratic convergence of NR is compromised this technique enables handling large deformations even with a single pass of Newton i.e. a single load increment with no manual tuning or parameter adjustment required. Fig. 1 shows such an example of Projected Newton with a single load increment and the evolution of wrinkles at various iterations. To verify the robustness of this technique, a large-scale distortion optimisation benchmark with extreme deformations over 10000 computational meshes was carried in Poya et al. [18] where the proposed framework passed the entire benchmark with 100% success rate with no exception. To best of the our knowledge, no incumbent approach in structural analysis has ever undergone such a comprehensive and successful benchmarking. Moreover, blending this technique with continuation approaches specially for displacement-driven problems is straight-forward but often not required. Under some pathological conditions applying the entire imposed displacements in one increment can result in element inversion or self-intersection for which continuation is often helpful. In this work, we present a continuation technique that achieves exactly this goal.

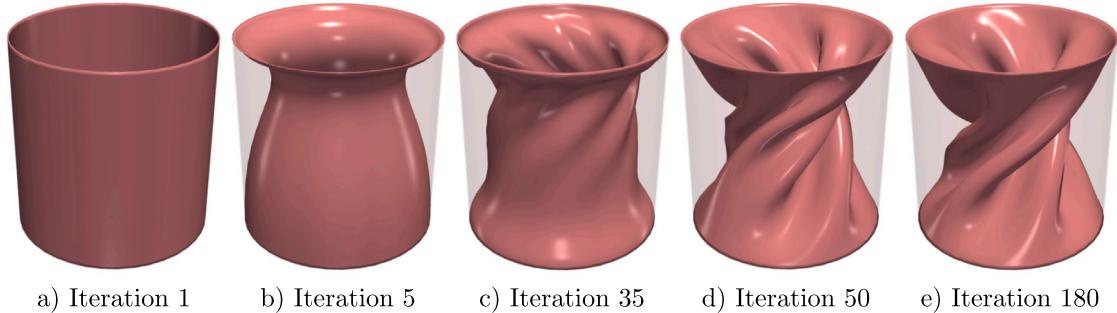


Fig. 1. Quasi-static simulation with tangent stabilised elasticity with a single step (1 load increment) utilising Newton–Raphson (i.e. Newton Raphson with tangents projected to positive semi-definite cone — P-NR) showing evolution of coarse wrinkles on a cylindrical twist.

Incumbent structural mechanics approaches for simulations beyond the onset of material and geometric instabilities is based on path-following techniques such as the popular arc-length method for quasi-static analysis [19–25] or alternatively resorting to dynamic analysis [26–28]. The arc length method in particular, is a curve parametrisation strategy that restricts the converged solution to follow a certain path. For quasi-static analysis, arc length is currently the only powerful method used for overcoming non-convergence of NR. However, despite receiving considerable attention since the early 1980s, the method is still far from being either robust and/or automatic on problems of industrial complexity [5,29]. Back-tracing an already traced path due to lack of a globally-accepted appropriate load factor choice, the dependence of the method on increment size, and no globally well-established strategy for Dirichlet-driven problems (or when external forces are absent) are amongst outstanding issues of the method not fully resolved yet [26]. However, the more pressing issue entails the non-convergence of the method under extreme deformations in emerging applications as repeatedly documented [2–4,30], forcing researchers to adopt simpler consistently linearised elasticity formulations instead [3,30–33]. In this work, we make the case that, tangent stabilised elasticity can be a powerful alternative for such applications with excellent automation capacity requiring no tuning and parameter input.

It is worth noting that, in the context of nonlinear finite elements, many stabilisation techniques have been proposed throughout the years specially for mixed finite elements [34–41] which are specific to the treatment of instabilities arising from certain families of mixed methods. Some of these instabilities are numerical such as artificial hourgassing nevertheless, they still emanate from indefiniteness of tangent operators [42,43]. These approaches have more of a case-by-case treatment depending on the mixed finite element spaces used. Our work is distinct from these works and does not deal with such artificial instabilities. Instead, we tackle the general problem of physical instabilities in hyperelastic solids and their subsequent tangent stabilisation. However, the proposed stabilisation can be used in the context of mixed methods (as shown in [13] for the novel class of mixed finite elements in principal stretches) eliminating sources of artificial instability too. In [Appendix A](#), we describe a mixed variational principle for transversely isotropic solids with inextensible fibres wherein we perform a similar stabilisation. We adhere to the most prudent way of eliminating tangent indefiniteness favouring automation and robustness while compromising quadratic convergence.

In this work, we lay out the groundwork for generalised tangent stabilised elasticity for virtually all known formulations of large strain elasticity by expanding the recently proposed framework of Poya et al. [13]. Intuitively, this entails finding analytical eigensystems of tangent elasticity operators emanating from all of these formulations. In particular, the key contributions of this work can be summarised as follows:

1. Closed-form tangent eigensystems for Total Lagrangian deformation gradient (F)-based and right Cauchy–Green (C)-based as well as Updated Lagrangian left Cauchy–Green (b)-based formulations of nonlinear elasticity.
2. Exact convexity conditions for all of the above formulations postulated in terms of their corresponding tangent and geometric stiffness eigenvalues.
3. Introducing geometrically stabilised polyconvex large strain elasticity, for polyconvex formulations and presenting compact expressions for their geometric stiffness eigensystem.
4. Extension to the case of transverse isotropy and once again, working out closed-form tangent eigensystems for common transversely isotropic invariants.
5. Discretisation-aware load stepping scheme for inversion-safe simulation with tangent stabilised elasticity, and finally
6. Mixed Hu–Washizu variational formulation and finite element implementation for large strain elastic models with inextensible fibres augmenting the mixed finite element framework of Poya et al. [13] for deformable and rigid models.

Complete implementation of all formulations is available for prototyping via the open-source Florence finite element package <https://github.com/romeric/florence> while a performance-oriented and more thorough implementation is part of [Simcenter3DPre/Post\(Meshing\)](#) suite.

The structure of the paper is as follows. In Section 2, the mapping and kinematics of motion is introduced, followed by the ingredients that make up our formulation. In Section 3, we present large strain elasticity, the general convexity conditions and transverse isotropy restrictions imposed on the strain energy densities. In Section 4, we briefly review the work of [13] and explicitly

work out eigensystems for various isotropic invariant formulations of large strain elasticity. In Section 5, we present the same for transversely isotropic invariants. In 6, we introduce a new continuation technique with Newton–Raphson procedure and a line search scheme that guarantees element-inversion free simulations through safe load stepping. Various numerical examples are presented in Section 7. Finally Section 8 concludes the paper. In Appendices A–B, variational principles including a mixed Hu–Washizu variational formulation for transversely isotropic large strain elasticity with inextensible fibres is presented and its finite element discretisation is discussed.

2. Continuum mechanics preliminaries

Consider the 3-dimensional deformation of an elastic medium from its initial configuration occupying a volume V , of boundary ∂V , into a final configuration at volume v , of boundary ∂v (see Fig. 2). Following Bonet et al. [7,8] the standard notation and definitions for fundamental kinematic measures namely, the deformation gradient F , its cofactor H and its determinant J are used¹

$$F = \frac{\partial x}{\partial X} = \nabla_0 x, \quad H = J F^{-T} = \frac{1}{2} F \times F, \quad J = \det F. \quad (1)$$

where x represents the current position of a particle originally at X and ∇_0 denotes the gradient with respect to material coordinates and \times is the tensor cross product operator $[A \times B]_{ij} = \epsilon_{ijk} \epsilon_{IJK} A_{jJ} B_{kK}$ where ϵ is the third order Levi-Civita tensor [8,44]. Virtual and incremental variations of x will be denoted by δu and Δu , respectively. It is assumed that $x, \delta u$ and Δu satisfy appropriate displacement based boundary conditions in $\partial_u V$. Additionally, the body is under the action of certain body forces per unit undeformed volume f_0 and traction per unit undeformed area t_0 in $\partial_t V$, where $\partial_t V \cup \partial_u V = \partial V$ and $\partial_t V \cap \partial_u V = \emptyset$. Crucial to our development is the left polar decomposition of the deformation gradient tensor

$$F = RU, \quad (2)$$

into a rotation tensor R and a stretch tensor U . This decomposition is often obtained from the Singular-Value-Decomposition (SVD) of F as follows

$$F = \hat{U} \Lambda \hat{V}^T, \quad R = \hat{U} \hat{V}^T, \quad U = \hat{V} \Lambda \hat{V}^T, \quad (3)$$

where \hat{U} and \hat{V} are orthogonal tensors namely left and right singular-matrices, respectively and the tensor Λ encodes the singular-values of F i.e. the principal stretches namely $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ such that $\lambda_i = \Lambda_{ii}$ (we have used the symbol \hat{U} for left singular matrix (and accordingly \hat{V} for right singular matrix) to distinguish with the stretch tensor U). Note that, in mechanics literature, the more standard definitions for the stretch tensor U and the deformation gradient tensor F in principal direction appear as

$$U = \sum_{i=1}^d \lambda_i [N]_i \otimes [N]_i, \quad F = \sum_{i=1}^d \lambda_i [\mathbf{n}]_i \otimes [\mathbf{N}]_i, \quad (4)$$

where $[N]_i$ are the material principal directions in fact corresponding to the columns of \hat{V} which is obtained from the SVD and d denotes the Euclidean space dimension (i.e. 1,2,3). Similarly, vectors $[\mathbf{n}]_i$ represent the spatial principal directions $[\mathbf{n}]_i = R[N]_i$.

3. Large strain elasticity

There is often a set of requirements that the internal strain energy density e has to fulfil. In this section we briefly review two such requirements most pertinent to our work.

The first requirement is that of convexity. Convexity implies that for energy function $e(F)$ the following condition should be met [10,45,46]

$$e(\theta F_1 + (1 - \theta) F_2) \leq \theta e(F_1) + (1 - \theta) e(F_2); \quad \forall F_1, F_2 \in \mathbb{R}^{d \times d}; \quad \theta \in [0, 1]. \quad (5)$$

which for differentiable functions can be expressed alternatively as

$$\left(\frac{\partial e(F)}{\partial F} \Big|_{F_1} - \frac{\partial e(F)}{\partial F} \Big|_{F_2} \right) : (F_1 - F_2) \geq 0; \quad \forall F_1, F_2 \in \mathbb{R}^{d \times d}, \quad (6)$$

and for twice differentiable functions can be expressed alternatively as

$$D^2 e(F)[\delta F; \delta F] = \delta F \bullet C \bullet \delta F \geq 0; \quad \forall F, \delta F \in \mathbb{R}^{d \times d}, \quad (7)$$

where \bullet is the dual product. Eq. (7) essentially mandates positive semi-definiteness of the elasticity tensor C in (17).

¹ Unless stated otherwise, throughout this work, lower case bold letters (a) denote vectors, capital bold letters (A) second order tensors and capital blackboard-bold letters (\mathbb{A}) fourth order tensors. Calligraphic letters (\mathcal{A}) are used to represent tangent elasticity operators, λ s are used to denote the singular-values of the deformation gradient tensor (i.e. principal stretches) and $\bar{\lambda}$ s are used to denote the eigenvalues of Hessian and tangent elasticity operators. Principal and non-principal directions are represented with N and M , respectively. Square brackets around bold letters with subscripts such as $([N]_i)$ indicate distinct vectors/tensors spanning in d -dimensional space as opposed to indices of the stated vector/tensor (N_i).

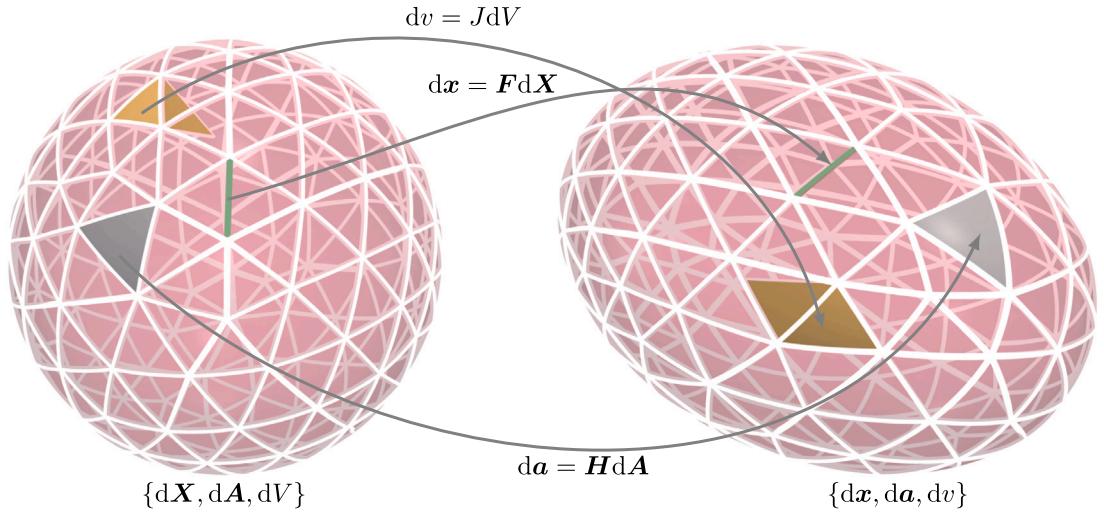


Fig. 2. Mapping original and deformed configurations and fundamental kinematic measures of motion in \mathbb{R}^3 Euclidean space. $\{F, H, J\}$ map edges, areas and volumes from original $\{dX, dA, dV\}$ to deformed $\{dx, da, dv\}$ configuration, respectively.

The second requirement is that of objectivity or frame invariance which implies independence of the energy density with respect to arbitrary rotations Q of the spatial configuration, which can be formulated as

$$e(QF) = e(F); \quad \forall Q \in SO(3) \quad (8)$$

with $SO(3)$ the special (proper) orthogonal group [47,48]. This requirement dictates that the strain energy e should be expressed in terms of symmetric strain measures such as the right Cauchy–Green strain tensor $C = F^T F$ as $e(F) = \tilde{e}(C)$. For the anisotropic energies further restrictions, reflecting the consequences of the principle of material symmetry, have to be taken into account. In particular, the energy e must be independent of any rotations/reflections $Q \in G_m \subset O(3)$ of the material configuration, where G_m denotes the corresponding symmetry group and $O(3)$ the full orthogonal group [10,49,50]. The anisotropic restriction is formally formulated as

$$\tilde{e}(C) = \tilde{e}(QCQ^T); \quad \forall Q \in O(3) \quad (9)$$

Similar to [10], in this work, we restrict ourselves to the symmetry $D_{\infty h}$, which corresponds to the usual definition of transverse isotropy and the symmetry C_∞ , the so-called rotational symmetry. Moreover, these two symmetry groups can be characterised by appropriate *structural tensors* which encapsulate the symmetry attributes of their corresponding groups, which in are given by the second-order tensor $M \otimes M$ (for $D_{\infty h}$) and the first-order tensor (vector) M (for C_∞). It is now straightforward to obtain an irreducible list of isotropic invariants (the so-called *integrity basis*) for the characterisation of energy densities.

$$\text{isotropic elasticity : } \text{tr}(C), \text{tr}(C^2), \text{tr}(C^3), \quad (10a)$$

$$\text{transversely-isotropic elasticity : } \text{tr}(C(M \otimes M)), \text{tr}(C^2(M \otimes M)), \quad (10b)$$

where tr represents the trace operator. We denote the above 5 invariants as

$$J_1 = \text{tr}(C), \quad J_2 = \text{tr}(C^2), \quad J_3 = \text{tr}(C^3), \quad J_4 = \text{tr}(C(M \otimes M)), \quad J_5 = \text{tr}(C^2(M \otimes M)), \quad (11)$$

and introduce the internal energy \tilde{e} as

$$e(\nabla_0 x) = \tilde{e}(J_1, J_2, J_3, J_4, J_5) \quad (12)$$

Many particularisations of the above integrity bases are possible and have surfaced in the literature. In the next two sections, we review virtually all commonly used formulations of isotropic and transversely isotropic large strain elasticity. We proceed to obtain closed-form eigen-decompositions for their constitutive tangent and geometric stiffness operators. This enables postulating convexity in terms of these newly found eigenvalues which is used to devise the corresponding tangent stabilisations for each formulation.

4. Stabilised large strain isotropic elasticity

4.1. Large strain isotropic elasticity in principle stretches: The new way

We start with large strain isotropic elasticity formulation of Poya et al. [13] which will serve as the basis for our subsequent formulations. Following [13], we express the internal strain energy density e dependent on the state of deformation $\nabla_0 x$ and

encapsulating the constitutive information of an isotropic hyperelastic material necessary to close the system of governing equations, in terms of principal stretches of the deformation gradient as follows

$$e(\nabla_0 \mathbf{x}) = W(\lambda_1, \lambda_2, \lambda_3) = W(\mathcal{A}_F), \quad (13)$$

where for brevity we have defined the set of principal stretches as $\mathcal{A}_F = \{\lambda_1, \dots, \lambda_d\}$. Furthermore, a set of work conjugate stresses $\Sigma_F = \{\Sigma_{\lambda_1}, \dots, \Sigma_{\lambda_d}\}$ can be introduced corresponding to each principal stretch such that

$$\Sigma_{\lambda_1} = \frac{\partial W}{\partial \lambda_1}, \quad \Sigma_{\lambda_2} = \frac{\partial W}{\partial \lambda_2}, \quad \Sigma_{\lambda_3} = \frac{\partial W}{\partial \lambda_3}, \quad (14)$$

which can be used to obtain the more standard first Piola–Kirchhoff stress tensor \mathbf{P} by first recalling that

$$De[\delta \mathbf{u}] = \mathbf{P} : \nabla_0 \delta \mathbf{u}; \quad \mathbf{P} = \left. \frac{\partial e}{\partial \mathbf{F}} \right|_{\mathbf{F}=\nabla_0 \mathbf{x}}. \quad (15)$$

which leads to the evaluation of \mathbf{P} as [13]

$$\mathbf{P} = \sum_{i=1}^d \Sigma_{\lambda_i} \frac{\partial \lambda_i}{\partial \mathbf{F}} = \sum_{i=1}^d \Lambda_{P_{ii}} [\mathbf{n}]_i \otimes [\mathbf{N}]_i = \hat{\mathbf{U}} \mathbf{\Lambda}_P \hat{\mathbf{V}}^T, \quad (16)$$

where $\Lambda_{P_{ii}} = \Sigma_{\lambda_i}$ is the diagonal principal stress tensor revealing that, the first Piola Kirchhoff stress tensor \mathbf{P} is coaxial with \mathbf{F} . Consequently, we term this formulation as “ \mathbf{F} -based formulation in principal stretches” in sharp contrast with the classical right Cauchy–Green or “ \mathbf{C} -based formulation in principal stretches” that uses the squares of the singular-values. The total tangent elasticity operator \mathbf{C} needed in the context of second order minimisation techniques such as Newton–Raphson can be obtain in a similar fashion by first recalling that

$$D^2 e[\delta \mathbf{u}; \Delta \mathbf{u}] = \nabla_0 \delta \mathbf{u} : D\mathbf{P}[\Delta \mathbf{u}] = \nabla_0 \delta \mathbf{u} : \mathbf{C} : \nabla_0 \Delta \mathbf{u}; \quad \mathbf{C} = \left. \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \right|_{\mathbf{F}=\nabla_0 \mathbf{x}} = \left. \frac{\partial^2 e}{\partial \mathbf{F} \partial \mathbf{F}} \right|_{\mathbf{F}=\nabla_0 \mathbf{x}}, \quad (17)$$

which leads to the evaluation of \mathbf{C} as an algebraic summation of the constitutive \mathbf{C}_k and the geometric stiffness \mathbf{C}_p components as [13]

$$\begin{aligned} \mathbf{C} = \mathbf{C}_k + \mathbf{C}_p &= \left[\frac{\partial \lambda_1}{\partial \mathbf{F}}, \dots, \frac{\partial \lambda_d}{\partial \mathbf{F}} \right] [\mathbf{H}_W] \begin{bmatrix} \frac{\partial \lambda_1}{\partial \mathbf{F}} \\ \vdots \\ \frac{\partial \lambda_d}{\partial \mathbf{F}} \end{bmatrix} + \sum_{i=1}^d \Sigma_{\lambda_i} \frac{\partial^2 \lambda_i}{\partial \mathbf{F} \partial \mathbf{F}} \\ &= \underbrace{\sum_{i=1}^d \sum_{j=1}^d \frac{\partial^2 W}{\partial \lambda_i \partial \lambda_j} \frac{\partial \lambda_i}{\partial \mathbf{F}} \otimes \frac{\partial \lambda_j}{\partial \mathbf{F}}}_{\mathbf{C}_k} + \underbrace{\sum_{i=1}^d \Sigma_{\lambda_i} \frac{\partial^2 \lambda_i}{\partial \mathbf{F} \partial \mathbf{F}}}_{\mathbf{C}_p}, \end{aligned} \quad (18)$$

where \mathbf{H}_W denotes the symmetric but not necessarily PSD $d \times d$ Hessian operator containing the second derivatives of W with respect to principal stretches

$$\mathbf{H}_W = \begin{bmatrix} \frac{\partial^2}{\partial \lambda_1 \partial \lambda_1} W & \cdots & \frac{\partial^2}{\partial \lambda_1 \partial \lambda_d} W \\ \ddots & \ddots & \vdots \\ sym & & \frac{\partial^2}{\partial \lambda_d \partial \lambda_d} W \end{bmatrix}, \quad (19)$$

where $\frac{\partial^2}{\partial \lambda_i \partial \lambda_j} \hat{W} = \frac{\partial^2 \hat{W}}{\partial \lambda_j \partial \lambda_i}$ and the last term in (18) is the “initial stress” or “geometric stiffness” term. As described in [13], convexity mandates positive semi-definiteness of (18) and hence, of both constitutive \mathbf{C}_k (via the Hessian \mathbf{H}_W) and geometric \mathbf{C}_p contributions. The derivatives of principal stretches with respect to the deformation gradient $\frac{\partial \lambda_i}{\partial \mathbf{F}}$ needed for consistent linearisation can be obtained by taking directional derivatives which in 2-dimensions yields

$$D\lambda_1[\delta \mathbf{u}] = \frac{\partial \lambda_1}{\partial \mathbf{F}} : \nabla_0 \delta \mathbf{u}, \quad [\mathbf{D}]_1 = \frac{\partial \lambda_1}{\partial \mathbf{F}} = [\mathbf{n}]_1 \otimes [\mathbf{N}]_1 = \hat{\mathbf{U}} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (20a)$$

$$D\lambda_2[\delta \mathbf{u}] = \frac{\partial \lambda_2}{\partial \mathbf{F}} : \nabla_0 \delta \mathbf{u}, \quad [\mathbf{D}]_2 = \frac{\partial \lambda_2}{\partial \mathbf{F}} = [\mathbf{n}]_2 \otimes [\mathbf{N}]_2 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (20b)$$

and similarly in 3-dimensions

$$D\lambda_1[\delta \mathbf{u}] = \frac{\partial \lambda_1}{\partial \mathbf{F}} : \nabla_0 \delta \mathbf{u}, \quad [\mathbf{D}]_1 = \frac{\partial \lambda_1}{\partial \mathbf{F}} = [\mathbf{n}]_1 \otimes [\mathbf{N}]_1 = \hat{\mathbf{U}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (21a)$$

$$D\lambda_2[\delta \mathbf{u}] = \frac{\partial \lambda_2}{\partial \mathbf{F}} : \nabla_0 \delta \mathbf{u}, \quad [\mathbf{D}]_2 = \frac{\partial \lambda_2}{\partial \mathbf{F}} = [\mathbf{n}]_2 \otimes [\mathbf{N}]_2 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (21b)$$

$$D\lambda_3[\delta \mathbf{u}] = \frac{\partial \lambda_3}{\partial \mathbf{F}} : \nabla_0 \delta \mathbf{u}, \quad [\mathbf{D}]_3 = \frac{\partial \lambda_3}{\partial \mathbf{F}} = [\mathbf{n}]_3 \otimes [\mathbf{N}]_3 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (21c)$$

where we have denoted $[\mathbf{D}]_i = \frac{\partial \lambda_i}{\partial F}$ for the scaling rank-one tensors to ease further developments down the line. The elegance of this formulation is that both the constitutive tangent \mathbf{C}_k and geometric stiffness \mathbf{C}_p can be represented in spectrally-decomposed form. The Hessian operator \mathbf{H}_W is a small $d \times d$ matrix and simple to find its analytical eigenvalues both in 2 and 3 dimensions; refer to [13] for analytical formulae. The geometric stiffness on the other hand, has a null space of 2 in 2-dimensions (only two eigenvalues) and a null space of 3 in 3-dimensions (only 6 eigenvalues) and can be written as an algebraic summation of eigenvalues and eigenmatrices as [13]

$$\mathbf{C}_p = \sum_{i=1}^k \bar{\lambda}_i^{C_p} [\mathbf{L}]_i \otimes [\mathbf{L}]_i + \sum_{i=1}^k \bar{\lambda}_{i+k}^{C_p} [\mathbf{T}]_i \otimes [\mathbf{T}]_i, \quad \text{where } k = 1 \text{ in 2D; } k = 3 \text{ in 3D}, \quad (22)$$

where $\bar{\lambda}_i^{C_p}$ are the eigenvalues of the geometric stiffness term which in 2-dimensions are found concisely as

$$\bar{\lambda}_1^{C_p} = \frac{\Sigma_{\lambda_1} - \Sigma_{\lambda_2}}{\lambda_1 - \lambda_2}, \quad \bar{\lambda}_2^{C_p} = \frac{\Sigma_{\lambda_1} + \Sigma_{\lambda_2}}{\lambda_1 + \lambda_2}, \quad (23)$$

and similarly in 3-dimensions the 6 eigenvalues of the geometric stiffness are found concisely as

$$\begin{aligned} \bar{\lambda}_1^{C_p} &= \frac{\Sigma_{\lambda_2} - \Sigma_{\lambda_3}}{\lambda_2 - \lambda_3}, & \bar{\lambda}_2^{C_p} &= \frac{\Sigma_{\lambda_1} - \Sigma_{\lambda_3}}{\lambda_1 - \lambda_3}, & \bar{\lambda}_3^{C_p} &= \frac{\Sigma_{\lambda_1} - \Sigma_{\lambda_2}}{\lambda_1 - \lambda_2}, \\ \bar{\lambda}_4^{C_p} &= \frac{\Sigma_{\lambda_2} + \Sigma_{\lambda_3}}{\lambda_2 + \lambda_3}, & \bar{\lambda}_5^{C_p} &= \frac{\Sigma_{\lambda_1} + \Sigma_{\lambda_3}}{\lambda_1 + \lambda_3}, & \bar{\lambda}_6^{C_p} &= \frac{\Sigma_{\lambda_1} + \Sigma_{\lambda_2}}{\lambda_1 + \lambda_2}, \end{aligned} \quad (24)$$

where $[\mathbf{L}]_s, [\mathbf{T}]_s$ are the eigenmatrices or the so-called *flip* and *twist* tensors (in the sense of Smith et al. [17]) which in 2-dimensions are given by

$$[\mathbf{L}]_1 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad [\mathbf{T}]_1 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (25)$$

and similarly in 3-dimensions

$$[\mathbf{L}]_1 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad [\mathbf{T}]_1 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (26a)$$

$$[\mathbf{L}]_2 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad [\mathbf{T}]_2 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (26b)$$

$$[\mathbf{L}]_3 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad [\mathbf{T}]_3 = \frac{1}{\sqrt{2}} \hat{\mathbf{U}} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T. \quad (26c)$$

It is important to note that, Eqs. (23)–(24) and (25)–(26) describe the eigen-decomposition of the geometric stiffness operator for any isotropic hyperelastic energy. Given that, (23) and (24) are eigenvalues hence, their positiveness coupled with positive semi-definiteness of (19) describes the convexity condition of the strain energy. In other words, convexity can be postulated in terms of these newly found tangent eigenvalues concisely as [13].

Corollary 1. A sufficiently smooth and twice differentiable two-dimensional energy function $e(\mathbf{F})$ is convex in \mathbf{F} if and only if there exists a function $W(\lambda_1, \lambda_2) = e(\mathbf{F})$ that satisfies

$$\frac{\Sigma_{\lambda_1} + \Sigma_{\lambda_2}}{\lambda_1 + \lambda_2} \geq 0, \quad \frac{\Sigma_{\lambda_1} - \Sigma_{\lambda_2}}{\lambda_1 - \lambda_2} \geq 0, \quad \partial_{\lambda_1 \lambda_1}^2 W \partial_{\lambda_2 \lambda_2}^2 W \geq \left(\partial_{\lambda_1 \lambda_2}^2 W \right)^2. \quad (27)$$

Corollary 2. A sufficiently smooth and twice differentiable three-dimensional energy function $e(\mathbf{F})$ is convex in \mathbf{F} if and only if there exists a function $W(\lambda_1, \lambda_2, \lambda_3) = e(\mathbf{F})$ that satisfies

$$\begin{aligned} \frac{\Sigma_{\lambda_1} + \Sigma_{\lambda_2}}{\lambda_1 + \lambda_2} &\geq 0, & \frac{\Sigma_{\lambda_1} + \Sigma_{\lambda_3}}{\lambda_1 + \lambda_3} &\geq 0, & \frac{\Sigma_{\lambda_2} + \Sigma_{\lambda_3}}{\lambda_2 + \lambda_3} &\geq 0, \\ \frac{\Sigma_{\lambda_1} - \Sigma_{\lambda_2}}{\lambda_1 - \lambda_2} &\geq 0, & \frac{\Sigma_{\lambda_1} - \Sigma_{\lambda_3}}{\lambda_1 - \lambda_3} &\geq 0, & \frac{\Sigma_{\lambda_2} - \Sigma_{\lambda_3}}{\lambda_2 - \lambda_3} &\geq 0, \\ \partial_{\lambda_1 \lambda_1}^2 W \partial_{\lambda_2 \lambda_2}^2 W \partial_{\lambda_3 \lambda_3}^2 W + 2\partial_{\lambda_1 \lambda_2}^2 W \partial_{\lambda_1 \lambda_3}^2 W \partial_{\lambda_2 \lambda_3}^2 W &\geq \\ \partial_{\lambda_1 \lambda_1}^2 W \left(\partial_{\lambda_2 \lambda_3}^2 W \right)^2 + \partial_{\lambda_2 \lambda_2}^2 W \left(\partial_{\lambda_1 \lambda_3}^2 W \right)^2 + \partial_{\lambda_3 \lambda_3}^2 W \left(\partial_{\lambda_1 \lambda_2}^2 W \right)^2. \end{aligned} \quad (28)$$

The proof of above corollaries can be found in [13]. The above convexity conditions are often too restrictive to impose on an energy function. The elegance of this formulation however is that we have obtained the tangent eigenvalues *and* their linkage to convexity essentially cost-free. Hence, alternatively we can proceed to work with the common (typically non-convex) strain energy functions and instead stabilise the resulting tangents. In other words, to guarantee positive semi-definiteness of \mathbf{H}_W , a tangent stabilisation procedure can be devised via performing a projection to PSD cone which essentially entails pruning tangent and geometric stiffness eigenvalues which are not positive. This has traditionally been achieved numerically by performing eigen-decomposition on the much larger tangent elasticity tensor \mathbf{C}_k and clamping the negative eigenvalues to zeros [15,16]. For the Hessian operator \mathbf{H}_W this can be achieved as

$$\mathbf{H}_W^{\text{PSD}} = \sum_{i=1}^d \max(\bar{\lambda}_i^{\mathbf{H}_W}, 0) [\mathbf{e}]_i \otimes [\mathbf{e}]_i, \quad (29)$$

where $\bar{\lambda}_i^{\mathbf{H}_W}$ are the eigenvalues and $[\mathbf{e}]_i$ are the eigenvectors of the Hessian \mathbf{H}_W . Fortunately, the eigenvalues of the constitutive part of the elasticity tensor \mathbf{C}_k can be found in closed-form solely through analytic eigen-decomposition of \mathbf{H}_W since \mathbf{C}_k has a null space of 2 in 2-dimensions and 6 in 3-dimensions. Hence, once $\mathbf{H}_W^{\text{PSD}}$ has been obtained the positive semi-definite constitutive tangent elasticity tensor $\mathbf{C}_k^{\text{PSD}}$ can be built in d -dimensions simply as

$$\mathbf{C}_k^{\text{PSD}} = \left[\frac{\partial \lambda_1}{\partial \mathbf{F}}, \dots, \frac{\partial \lambda_d}{\partial \mathbf{F}} \right] [\mathbf{H}_W^{\text{PSD}}] \begin{bmatrix} \frac{\partial \lambda_1}{\partial \mathbf{F}} \\ \vdots \\ \frac{\partial \lambda_d}{\partial \mathbf{F}} \end{bmatrix}. \quad (30)$$

From the numerical implementation point of view, it is more convenient to build \mathbf{C}_k and $\mathbf{C}_k^{\text{PSD}}$ in a spectrally decomposed form as follows

$$\mathbf{C}_k = \sum_{i=1}^d \sum_{j=1}^d \bar{\lambda}_i^{\mathbf{H}_W} [\mathbf{e}]_i^j [\mathbf{D}]_j \otimes [\mathbf{e}]_i^j [\mathbf{D}]_j, \quad \mathbf{C}_k^{\text{PSD}} = \sum_{i=1}^d \sum_{j=1}^d \max(\bar{\lambda}_i^{\mathbf{H}_W}, 0) [\mathbf{e}]_i^j [\mathbf{D}]_j \otimes [\mathbf{e}]_i^j [\mathbf{D}]_j, \quad (31)$$

where $[\mathbf{e}]_i^j$ implies the j th component of the eigenvector $[\mathbf{e}]_i$. A similar PSD projection can be performed on the geometric stiffness term. This is an easier case as the eigensystem of geometric stiffness operator is already obtained in a decomposed form as presented in (22). Consequently, PSD projection is obtained via

$$\mathbf{C}_p^{\text{PSD}} = \sum_{i=1}^k \max(\bar{\lambda}_i^{\mathbf{C}_p}, 0) [\mathbf{L}]_i \otimes [\mathbf{L}]_i + \sum_{i=1}^k \max(\bar{\lambda}_{i+k}^{\mathbf{C}_p}, 0) [\mathbf{T}]_i \otimes [\mathbf{T}]_i. \quad (32)$$

4.2. Large strain isotropic elasticity using Cauchy–Green strain

In practice, perhaps the most widely used and commonly implemented formulation of large strain elasticity is based on Cauchy–Green strain [48,51–54]. In this section, we demonstrate that the technique presented in Poya et al. [13] for spectrally-decomposed tangent operator construction is generic and can be applied to virtually any family of invariants in particular, to strain energies expressed in terms of Cauchy–Green and its invariants. To this end, as customary in nonlinear mechanics literature, we first express the strain energy e as

$$e(\nabla_0 \mathbf{x}) = \check{W}(\mathbf{C}) = \check{W}(\lambda_1^2, \lambda_2^2, \lambda_3^2) = \check{W}(\mathcal{A}_C), \quad (33)$$

where once again, as customary, the strain energy e has been further expressed in terms of the set of squares of principal stretches \mathcal{A}_C . The Cauchy–Green strain tensor \mathbf{C} can now be described in terms of principal information as

$$\mathbf{C} = \sum_{i=1}^d \lambda_i^2 [\mathbf{N}]_i \otimes [\mathbf{N}]_i = \hat{\mathbf{V}} \Lambda_C \hat{\mathbf{V}}^T, \quad (34)$$

where $\Lambda_{C_{ii}} = \lambda_i^2$. For brevity and without going into the details, the first and second directional derivatives of the energy \check{W} needed in the context of second order minimisation methods such as Newton–Raphson can be computed as; c.f. [52,53]

$$D\check{W}[\delta \mathbf{u}] = 2 \frac{\partial \check{W}}{\partial \mathbf{C}} : \frac{1}{2} DC[\delta \mathbf{u}] \quad (35a)$$

$$D^2 \check{W}[\delta \mathbf{u}; \Delta \mathbf{u}] = \frac{1}{2} DC[\delta \mathbf{u}] : 4 \frac{\partial^2 \check{W}}{\partial \mathbf{C} \partial \mathbf{C}} : \frac{1}{2} DC[\Delta \mathbf{u}] + 2 \frac{\partial \check{W}}{\partial \mathbf{C}} : \frac{1}{2} D^2 C[\delta \mathbf{u}; \Delta \mathbf{u}], \quad (35b)$$

and noting that, the directional derivatives of \mathbf{C} are given by (see for instance, [52] page 118 or [53], page 77), (35) simplifies to

$$D\check{W}[\delta \mathbf{u}] = \mathbf{S} : \frac{1}{2} DC[\delta \mathbf{u}] \quad (36a)$$

$$D^2 \check{W}[\delta \mathbf{u}; \Delta \mathbf{u}] = \frac{1}{2} DC[\delta \mathbf{u}] : \check{\mathbf{C}}_k : \frac{1}{2} DC[\Delta \mathbf{u}] + \nabla_0 \delta \mathbf{u} : \check{\mathbf{C}}_p : \nabla_0 \Delta \mathbf{u} \quad (36b)$$

leading to the evaluation of the second Piola–Kirchhoff stress tensor \mathbf{S} , the constitutive tangent $\check{\mathbf{C}}_k$ and the geometric stiffness $\check{\mathbf{C}}_p$ operators

$$\mathbf{S} = 2 \frac{\partial \check{W}}{\partial \mathbf{C}}, \quad \check{\mathbf{C}}_k = 4 \frac{\partial^2 \check{W}}{\partial \mathbf{C} \partial \mathbf{C}}, \quad \check{\mathbf{C}}_p = \mathbf{I} \otimes \mathbf{S}, \quad (37)$$

Similar to (18), a sufficient condition for convexity is then the positive semi-definiteness of (36b) via the constitutive $\check{\mathbf{C}}_k$ and geometric $\check{\mathbf{C}}_p$ contributions.

Our goal now is to find the closed-form eigensystem of both $\check{\mathbf{C}}_k$ and $\check{\mathbf{C}}_p$. The recipe here is to resort once again to principal stretch-based approach (which we term as the \mathbf{C} -based formulation in principal stretches) and similar to before introduce a set of work conjugate stresses $S_C = \{\bar{\Sigma}_{\lambda_1}, \dots, \bar{\Sigma}_{\lambda_d}\}$ first but this time corresponding to the squares of principal stretches such that

$$\bar{\Sigma}_{\lambda_1} = 2 \frac{\partial \bar{W}}{\partial \lambda_1^2}, \quad \bar{\Sigma}_{\lambda_2} = 2 \frac{\partial \bar{W}}{\partial \lambda_2^2}, \quad \bar{\Sigma}_{\lambda_3} = 2 \frac{\partial \bar{W}}{\partial \lambda_3^2}, \quad (38)$$

which can be used to define the second Piola–Kirchhoff stress tensor \mathbf{S} similar to (16)

$$\mathbf{S} = \sum_{i=1}^d \bar{\Sigma}_{\lambda_i} \frac{\partial \lambda_i^2}{\partial \mathbf{C}} = \sum_{i=1}^d \bar{\Sigma}_{\lambda_i} [\mathbf{N}]_i \otimes [\mathbf{N}]_i = \hat{\mathbf{V}} \mathbf{A}_S \hat{\mathbf{V}}^T. \quad (39)$$

Similar to (20)–(21), the derivatives of squares of principal stretches with respect to \mathbf{C} needed for consistent linearisation can be calculated by taking directional derivatives which in 2-dimensions leads to

$$[\bar{\mathbf{D}}]_1 = \frac{\partial \lambda_1^2}{\partial \mathbf{C}} = [\mathbf{N}]_1 \otimes [\mathbf{N}]_1 = \hat{\mathbf{V}} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (40a)$$

$$[\bar{\mathbf{D}}]_2 = \frac{\partial \lambda_2^2}{\partial \mathbf{C}} = [\mathbf{N}]_2 \otimes [\mathbf{N}]_2 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (40b)$$

and similarly in 3-dimensions

$$[\bar{\mathbf{D}}]_1 = \frac{\partial \lambda_1^2}{\partial \mathbf{C}} = [\mathbf{N}]_1 \otimes [\mathbf{N}]_1 = \hat{\mathbf{V}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (41a)$$

$$[\bar{\mathbf{D}}]_2 = \frac{\partial \lambda_2^2}{\partial \mathbf{C}} = [\mathbf{N}]_2 \otimes [\mathbf{N}]_2 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (41b)$$

$$[\bar{\mathbf{D}}]_3 = \frac{\partial \lambda_3^2}{\partial \mathbf{C}} = [\mathbf{N}]_3 \otimes [\mathbf{N}]_3 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (41c)$$

where we have denoted $[\bar{\mathbf{D}}]_i = \frac{\partial \lambda_i^2}{\partial \mathbf{C}}$ for the scaling rank-one tensors to highlight their similarities with $[\mathbf{D}]_i$ found earlier. Hence, following a procedure similar to Poya et al. [13] for finding the eigensystem of $\frac{\partial^2 W}{\partial F \partial F}$, the constitutive tangent $\check{\mathbf{C}}_k$ operator emerges as an algebraic summation of two further terms

$$\begin{aligned} \check{\mathbf{C}}_k &= \frac{\partial^2 \bar{W}}{\partial \mathbf{C} \partial \mathbf{C}} = 4 \left[\frac{\partial \lambda_1^2}{\partial \mathbf{C}}, \dots, \frac{\partial \lambda_d^2}{\partial \mathbf{C}} \right] [\mathbf{H}_{\bar{W}}] \begin{bmatrix} \frac{\partial \lambda_1^2}{\partial \mathbf{C}} \\ \vdots \\ \frac{\partial \lambda_d^2}{\partial \mathbf{C}} \end{bmatrix} + \sum_{i=1}^d \bar{\Sigma}_{\lambda_i} \frac{\partial^2 \lambda_i^2}{\partial \mathbf{C} \partial \mathbf{C}} \\ &= \sum_{i=1}^d \sum_{j=1}^d 4 \frac{\partial^2 \bar{W}}{\partial \lambda_i^2 \lambda_j^2} \frac{\partial \lambda_i^2}{\partial \mathbf{C}} \otimes \frac{\partial \lambda_j^2}{\partial \mathbf{C}} + \sum_{i=1}^d \bar{\Sigma}_{\lambda_i} \frac{\partial^2 \lambda_i^2}{\partial \mathbf{C} \partial \mathbf{C}}. \end{aligned} \quad (42)$$

where $\mathbf{H}_{\bar{W}}$ denotes the symmetric but not necessarily PSD $d \times d$ Hessian operator containing the second derivatives of \bar{W} with respect to squares of principal stretches

$$\mathbf{H}_{\bar{W}} = \begin{bmatrix} \frac{\partial^2}{\lambda_1^2 \lambda_1^2} \bar{W} & \cdots & \frac{\partial^2}{\lambda_1^2 \lambda_d^2} \bar{W} \\ \ddots & \ddots & \vdots \\ sym & & \frac{\partial^2}{\lambda_d^2 \lambda_d^2} \bar{W} \end{bmatrix}. \quad (43)$$

The similarity between (18), (19), (42) and (43) is apparent. In fact, the structure of (42) reveals a further decomposition of $\check{\mathbf{C}}_k$ into a constitutive and a geometric-type part when expressed in terms of principal stretches. Unlike the \mathbf{F} -based formulation of (18), this \mathbf{C} -based formulation mixes up first and second order derivatives. This can be demonstrated by looking at the explicit

structure of $\mathbf{H}_{\bar{W}}$ in (43) which can be expanded in terms of derivatives of principal stretches directly (rather than their squares)

$$\mathbf{H}_{\bar{W}} = \begin{bmatrix} \frac{1}{\lambda_1^2} \partial_{\lambda_1 \lambda_1}^2 \bar{W} - \frac{1}{2\lambda_1^3} \bar{\Sigma}_{\lambda_1} & \cdots & \frac{1}{\lambda_1 \lambda_d} \partial_{\lambda_1 \lambda_d}^2 \bar{W} \\ \vdots & \ddots & \vdots \\ sym & \frac{1}{\lambda_d^2} \partial_{\lambda_d \lambda_d}^2 \bar{W} - \frac{1}{2\lambda_d^3} \bar{\Sigma}_{\lambda_d} \end{bmatrix}. \quad (44)$$

Following the same procedure as Poya et al. [13] after some tedious but straight-forward algebra we can show that the analytic eigensystem of \mathbf{C} -based formulations has a strikingly similar structure to \mathbf{F} -based formulation with minor differences. The first d eigenvalues and eigenmatrices are found from the matrix $\mathbf{H}_{\bar{W}}$ in (43) (or the first term in (42)). The next few constitutive tangent eigenvalues come from the second term in (42). This term has a null space of 3 in 2-dimensions (only 1 eigenvalue) and a null space of 6 in 3-dimensions (only 3 eigenvalues). Its only eigenvalue in 2-dimensions can be written concisely as

$$\bar{\lambda}_3^{\check{C}_k} = \frac{\bar{\Sigma}_{\lambda_1} - \bar{\Sigma}_{\lambda_2}}{\lambda_1^2 - \lambda_2^2}, \quad (45)$$

and similarly in 3-dimensions the 3 eigenvalues of the geometric stiffness are found concisely as

$$\bar{\lambda}_4^{\check{C}_k} = \frac{\bar{\Sigma}_{\lambda_2} - \bar{\Sigma}_{\lambda_3}}{\lambda_2^2 - \lambda_3^2}, \quad \bar{\lambda}_5^{\check{C}_k} = \frac{\bar{\Sigma}_{\lambda_1} - \bar{\Sigma}_{\lambda_3}}{\lambda_1^2 - \lambda_3^2}, \quad \bar{\lambda}_6^{\check{C}_k} = \frac{\bar{\Sigma}_{\lambda_1} - \bar{\Sigma}_{\lambda_2}}{\lambda_1^2 - \lambda_2^2}. \quad (46)$$

The only eigenmatrix in 2-dimensions take the form

$$[\bar{\mathbf{L}}]_1 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad (47)$$

and similarly in 3-dimensions the 3 eigenmatrices take the form

$$[\bar{\mathbf{L}}]_1 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad [\bar{\mathbf{L}}]_2 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T, \quad [\bar{\mathbf{L}}]_3 = \hat{\mathbf{V}} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{V}}^T. \quad (48)$$

which are essentially the \mathbf{C} -based *flip* tensors similar to (24). Interestingly due to material frame invariance of \mathbf{C} -based formulation the corresponding \mathbf{C} -based *twist* tensors do not exist. Finally, the last 3 eigenvalues come from the geometric stiffness operator \check{C}_p . Noting its structure in (37) (that is the appearance of the identity tensor I), its eigensystem is trivial and in fact already available. The geometric stiffness operator \check{C}_p has 3 eigenvalues with a multiplicity of 3 and they are indeed the principal conjugate stresses $\bar{\Sigma}_{\lambda_i}$ s i.e.

$$\bar{\lambda}_i^{\check{C}_p} = \bar{\Sigma}_{\lambda_i} \quad i = 1, \dots, d \quad (49)$$

and the corresponding eigenmatrices are simply $\hat{\mathbf{V}}$. Consequently, the constitutive tangent elasticity tensor \check{C}_k can be constructed in a spectrally decomposed form as

$$\check{C}_k = \frac{\partial^2 \bar{W}}{\partial \mathbf{C} \partial \mathbf{C}} = 4 \left[\frac{\partial \lambda_1^2}{\partial \mathbf{C}}, \dots, \frac{\partial \lambda_d^2}{\partial \mathbf{C}} \right] [\mathbf{H}_{\bar{W}}] \begin{bmatrix} \frac{\partial \lambda_1^2}{\partial \mathbf{C}} \\ \vdots \\ \frac{\partial \lambda_d^2}{\partial \mathbf{C}} \end{bmatrix} + \sum_{i=1}^k \bar{\lambda}_{i+d}^{\check{C}_k} [\bar{\mathbf{L}}]_i \otimes [\bar{\mathbf{L}}]_i. \quad (50)$$

To guarantee positive semi-definiteness of $\mathbf{H}_{\bar{W}}$ a similar tangent stabilisation procedure described before can be performed via a projection to PSD cone. For the Hessian operator this can be achieved as

$$\mathbf{H}_{\bar{W}}^{\text{PSD}} = \sum_{i=1}^d \max(\bar{\lambda}_i^{\mathbf{H}_{\bar{W}}}, 0) [\bar{\mathbf{e}}]_i \otimes [\bar{\mathbf{e}}]_i, \quad (51)$$

where $\bar{\lambda}_i^{\mathbf{H}_{\bar{W}}}$ are the eigenvalues and $[\bar{\mathbf{e}}]_i$ are the eigenvectors of the Hessian $\mathbf{H}_{\bar{W}}$. Similar to \mathbf{F} -based formulation, the eigenvalues of the constitutive part of the elasticity tensor \check{C}_k can be found in closed-form solely through analytic eigen-decomposition of $\mathbf{H}_{\bar{W}}$ can be built in d -dimensions simply as

$$\check{C}_k^{\text{PSD}} = 4 \left(\frac{\partial^2 \bar{W}}{\partial \mathbf{C} \partial \mathbf{C}} \right)^{\text{PSD}} = 4 \left[\frac{\partial \lambda_1^2}{\partial \mathbf{C}}, \dots, \frac{\partial \lambda_d^2}{\partial \mathbf{C}} \right] [\mathbf{H}_{\bar{W}}^{\text{PSD}}] \begin{bmatrix} \frac{\partial \lambda_1^2}{\partial \mathbf{C}} \\ \vdots \\ \frac{\partial \lambda_d^2}{\partial \mathbf{C}} \end{bmatrix} + \sum_{i=1}^k \max(\bar{\lambda}_{i+d}^{\check{C}_k}, 0) [\bar{\mathbf{L}}]_i \otimes [\bar{\mathbf{L}}]_i. \quad (52)$$

Alternatively, for numerical implementation a more convenient format of building \check{C}_k^{PSD} is

$$\check{C}_k^{\text{PSD}} = 4 \sum_{i=1}^d \sum_{j=1}^d \max(\bar{\lambda}_i^{\mathbf{H}_{\bar{W}}}, 0) [\bar{\mathbf{e}}]_i^T [\bar{\mathbf{D}}]_j \otimes [\bar{\mathbf{e}}]_i^T [\bar{\mathbf{D}}]_j + \sum_{i=1}^k \max(\bar{\lambda}_{i+d}^{\check{C}_k}, 0) [\bar{\mathbf{L}}]_i \otimes [\bar{\mathbf{L}}]_i. \quad (53)$$

A similar PSD projection can be performed on the geometric stiffness term. This is a straight-forward case. Consequently, PSD projection is obtained via

$$\check{\mathbf{C}}_p^{\text{PSD}} = \mathbf{I} \otimes \mathbf{S}^{\text{PSD}} = \mathbf{I} \otimes (\hat{\mathbf{V}} \mathbf{A}_S^{\text{PSD}} \hat{\mathbf{V}}^T). \quad (54)$$

From (52)–(54) it is evident that the \mathbf{C} -based formulation in total comprises of 5 eigenvalues in 2-dimensions and 9 eigenvalues in 3-dimensions (as opposed to 4 and 9 for \mathbf{F} -based formulation). Given that, (45) and (46) are eigenvalues hence their positiveness coupled with positive semi-definiteness of (19) and (54) describes the convexity condition of the strain energy. In other words, convexity can be postulated in terms of these newly found eigenvalues as [13].

Corollary 3. For a smooth and twice differentiable two-dimensional energy function $\bar{W}(\mathbf{C})$ to be convex in \mathbf{C} , it is sufficient that there exists a function $\bar{W}(\lambda_1^2, \lambda_2^2) = \bar{W}(\mathbf{C})$ that satisfies

$$\bar{\Sigma}_{\lambda_1} \geq 0, \quad \bar{\Sigma}_{\lambda_2} \geq 0, \quad \frac{\bar{\Sigma}_{\lambda_1} - \bar{\Sigma}_{\lambda_2}}{\lambda_1^2 - \lambda_2^2} \geq 0, \quad \partial_{\lambda_1^2 \lambda_1^2}^2 \bar{W} \partial_{\lambda_2^2 \lambda_2^2}^2 \bar{W} \geq \left(\partial_{\lambda_1^2 \lambda_2^2}^2 \bar{W} \right)^2. \quad (55)$$

Corollary 4. For a smooth and twice differentiable three-dimensional energy function $\bar{W}(\mathbf{C})$ to be convex in \mathbf{C} , it is sufficient that there exists a function $\bar{W}(\lambda_1^2, \lambda_2^2, \lambda_3^2) = \bar{W}(\mathbf{C})$ that satisfies

$$\begin{aligned} \bar{\Sigma}_{\lambda_1} &\geq 0, \quad \bar{\Sigma}_{\lambda_2} \geq 0, \quad \bar{\Sigma}_{\lambda_3} \geq 0, \\ \frac{\bar{\Sigma}_{\lambda_1} - \bar{\Sigma}_{\lambda_2}}{\lambda_1^2 - \lambda_2^2} &\geq 0, \quad \frac{\bar{\Sigma}_{\lambda_1} - \bar{\Sigma}_{\lambda_3}}{\lambda_1^2 - \lambda_3^2} \geq 0, \quad \frac{\bar{\Sigma}_{\lambda_2} - \bar{\Sigma}_{\lambda_3}}{\lambda_2^2 - \lambda_3^2} \geq 0, \\ \partial_{\lambda_1^2 \lambda_1^2}^2 \bar{W} \partial_{\lambda_2^2 \lambda_2^2}^2 \bar{W} \partial_{\lambda_3^2 \lambda_3^2}^2 \bar{W} + 2\partial_{\lambda_1^2 \lambda_2^2}^2 \bar{W} \partial_{\lambda_1^2 \lambda_3^2}^2 \bar{W} \partial_{\lambda_2^2 \lambda_3^2}^2 \bar{W} &\geq \\ \partial_{\lambda_1^2 \lambda_1^2}^2 \bar{W} \left(\partial_{\lambda_2^2 \lambda_3^2}^2 \bar{W} \right)^2 + \partial_{\lambda_2^2 \lambda_2^2}^2 \bar{W} \left(\partial_{\lambda_1^2 \lambda_3^2}^2 \bar{W} \right)^2 + \partial_{\lambda_3^2 \lambda_3^2}^2 \bar{W} \left(\partial_{\lambda_1^2 \lambda_2^2}^2 \bar{W} \right)^2. \end{aligned} \quad (56)$$

Remark 1. It is often convenient and rather customary to express the strain energy in terms of invariants of Cauchy–Green tensor \mathbf{C} such as

$$e(\nabla_0 \mathbf{x}) = \psi(I_C, II_C, III_C) \quad (57)$$

where the set of invariants is defined as

$$I_C = \text{tr}(\mathbf{C}) = \sum_{i=1}^d \lambda_i^2, \quad II_C = \mathbf{C} : \mathbf{C} = \sum_{i=1}^d \lambda_i^4, \quad III_C = \det(\mathbf{C}) = \prod_{i=1}^d \lambda_i^2, \quad (58)$$

where C is determinant of \mathbf{C} . Furthermore, we can introduce $\mathbf{G} = \mathbf{C}\mathbf{C}^{-1} = \frac{1}{2}\mathbf{C} \times \mathbf{C}$ to represent the cofactor of \mathbf{C} . Finding analytical eigensystems emanating from this integrity basis is now straight-forward using the algebra developed so far. Specifically noting that, consistent linearisation of ψ yields

$$\frac{\partial \psi}{\partial \mathbf{C}} = \frac{\partial \psi}{\partial I_C} \mathbf{I} + 2 \frac{\partial \psi}{\partial II_C} \mathbf{C} + \frac{\partial \psi}{\partial III_C} \mathbf{G}, \quad (59a)$$

$$\frac{\partial^2 \psi}{\partial \mathbf{C} \partial \mathbf{C}} = [\mathbf{I}, \quad 2\mathbf{C}, \quad \mathbf{G}] \underbrace{[\mathbf{H}_W^\psi] \begin{bmatrix} \mathbf{I} \\ 2\mathbf{C} \\ \mathbf{G} \end{bmatrix}}_{\mathbf{C}_k^\psi} + \underbrace{\left(2 \frac{\partial \psi}{\partial II_C} \mathbb{I} + \frac{\partial \psi}{\partial III_C} (\mathbb{I} \times \mathbf{C}) \right)}_{\mathbf{C}_p^\psi}, \quad (59b)$$

where \mathbf{H}_W^ψ represents the symmetric (but not necessarily positive definite) $d \times d$ Hessian operator

$$\mathbf{H}_W^\psi = \begin{bmatrix} \frac{\partial^2 \psi}{\partial I_C \partial I_C} & \frac{\partial^2 \psi}{\partial I_C \partial II_C} & \frac{\partial^2 \psi}{\partial I_C \partial III_C} \\ \frac{\partial^2 \psi}{\partial II_C \partial I_C} & \frac{\partial^2 \psi}{\partial II_C \partial II_C} & \frac{\partial^2 \psi}{\partial II_C \partial III_C} \\ \frac{\partial^2 \psi}{\partial III_C \partial I_C} & \frac{\partial^2 \psi}{\partial III_C \partial II_C} & \frac{\partial^2 \psi}{\partial III_C \partial III_C} \end{bmatrix}. \quad (60)$$

Similar to (18), a sufficient condition for convexity is then positive semi-definiteness of (59b) via the constitutive \mathbf{C}_k^ψ and geometric \mathbf{C}_p^ψ contributions. The constitutive part once again, has a $d \times d$ Hessian whose eigen-decomposition can be found analytically similar to (19) and (43) using the formula presented in [13]. The eigensystem of the geometric stiffness part can be found by the algebra developed earlier for \mathbf{C} -based formulation in particular, the term $\mathbb{I} \times \mathbf{C}$ represents the Hessian of $\mathbf{C} = J^2$ which can be easily obtained using our earlier arrangements. Later in Section 4.4, we undertake determining the eigensystem of the geometric stiffness term emerging from polyconvex energies which has the exact same structure. Ultimately, this formulation creates another level of “indirection” but its eigensystem can always be worked out using principal information alone. This in general holds true for all isotropic invariant formulations.

4.3. Updated Lagrangian formulation for large strain isotropic elasticity

In the engineering simulation community often stress and strain measures are used that are physically more meaningful such as the Cauchy (σ) or the Kirchhoff (τ) stress and engineering (ϵ) or alternatively the corresponding work-conjugate Almansi or left Cauchy–Green ($b = FF^T$) strain. We will henceforth call this formulation the b -based formulation and introduce $b = \det(b)$ and $g = bb^{-T} = \frac{1}{2}b \times b$. The above mentioned stresses can be obtained using the standard push-forward relationships [48,52–54]

$$\tau = J\sigma = FSF^T = PF^T. \quad (61)$$

Note that, the structure of FSF^T reveals a crucial identity. That is, if the second Piola–Kirchhoff stress tensor S is PSD then from elementary linear algebra [11] it follows that both τ , and σ are PSD (subject to positiveness of J which is always assumed). This is important as the eigenvalues of the geometric stiffness operator $\check{\mathcal{C}}_p$ in the C -based formulation presented in the previous section in (49) were precisely the singular-values of S . This implies that, we only need to perform PSD projection in the material or Lagrangian configuration and hence, reveals that push-forward operations involving C -based formulation retain convexity as well as any stabilisation applied. In particular, noting the standard push-forward operation between material $\check{\mathcal{C}}_k$ and spatial $\bar{\check{\mathcal{C}}}_k$ elasticity tensors [52,53]

$$\bar{\check{\mathcal{C}}}_k = [\mathbf{F}]_{iI}[\mathbf{F}]_{jJ}[\mathbf{F}]_{pP}[\mathbf{F}]_{qQ}\check{\mathcal{C}}_k, \quad (62)$$

the updated Lagrangian versions of the *scaling* (40)–(41) and *flip* (47)–(48) eigenmatrices are obtained as follows in 2-dimensions

$$[\bar{\check{\mathcal{D}}}]_1 = \hat{\mathbf{U}} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{U}}^T, \quad [\bar{\check{\mathcal{D}}}]_2 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \hat{\mathbf{U}}^T, \quad [\bar{\check{\mathcal{L}}}]_1 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{U}}^T, \quad (63a)$$

and similarly in 3-dimensions the 3 eigenmatrices take the form

$$[\bar{\check{\mathcal{D}}}]_1 = \hat{\mathbf{U}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{U}}^T, \quad [\bar{\check{\mathcal{D}}}]_2 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{U}}^T, \quad [\bar{\check{\mathcal{D}}}]_3 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{U}}^T, \quad (64a)$$

$$[\bar{\check{\mathcal{L}}}]_1 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \hat{\mathbf{U}}^T, \quad [\bar{\check{\mathcal{L}}}]_2 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \hat{\mathbf{U}}^T, \quad [\bar{\check{\mathcal{L}}}]_3 = \hat{\mathbf{U}} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{U}}^T. \quad (64b)$$

Furthermore, the spatial Hessian $\mathbf{H}_{\bar{W}}$ is related to the material Hessian \mathbf{H}_W in (43) via the following push-forward operation

$$\mathbf{H}_{\bar{W}} = \Lambda^2 \mathbf{H}_W \Lambda^2, \quad (65)$$

where Λ^2 is the diagonal tensor containing the squares of principal stretches $\Lambda_{ii}^2 = \lambda_i^2$. The eigenvalues of the spatial constitutive operator corresponding to *flip* modes are related to their material counterparts in (52) via

$$\bar{\lambda}_3^{\check{\mathcal{C}}_k} = (\lambda_1 \lambda_2)^2 \bar{\lambda}_3^{\check{\mathcal{C}}_k}, \quad \text{in 2D} \quad (66a)$$

$$\bar{\lambda}_4^{\check{\mathcal{C}}_k} = (\lambda_2 \lambda_3)^2 \bar{\lambda}_4^{\check{\mathcal{C}}_k}, \quad \bar{\lambda}_5^{\check{\mathcal{C}}_k} = (\lambda_1 \lambda_3)^2 \bar{\lambda}_5^{\check{\mathcal{C}}_k}, \quad \bar{\lambda}_6^{\check{\mathcal{C}}_k} = (\lambda_1 \lambda_2)^2 \bar{\lambda}_6^{\check{\mathcal{C}}_k}, \quad \text{in 3D} \quad (66b)$$

Similarly, the corresponding eigenvalues for spatial geometric stiffness operator $\bar{\check{\mathcal{C}}}_p$ is related to the eigenvalues of its material counterpart $\check{\mathcal{C}}_p$ in (37) in 2-dimensions via

$$\bar{\lambda}_4^{\check{\mathcal{C}}_p} = \lambda_1^2 \bar{\lambda}_4^{\check{\mathcal{C}}_p}, \quad \bar{\lambda}_5^{\check{\mathcal{C}}_p} = \lambda_2^2 \bar{\lambda}_5^{\check{\mathcal{C}}_p}, \quad \text{in 2D} \quad (67a)$$

$$\bar{\lambda}_7^{\check{\mathcal{C}}_p} = \lambda_1^2 \bar{\lambda}_7^{\check{\mathcal{C}}_p}, \quad \bar{\lambda}_8^{\check{\mathcal{C}}_p} = \lambda_2^2 \bar{\lambda}_8^{\check{\mathcal{C}}_p}, \quad \bar{\lambda}_9^{\check{\mathcal{C}}_p} = \lambda_3^2 \bar{\lambda}_9^{\check{\mathcal{C}}_p}, \quad \text{in 3D.} \quad (67b)$$

Finally, the convexity of updated Lagrangian formulation for b -based invariants can be postulated in terms of the above eigenvalues exactly in a similar fashion to (55)–(56) in that, as mentioned before, in this setting, convexity does not change with push-forward operations.

Remark 2. One difference of the current C -based (and the updated Lagrangian) formulation from classic formulations of elasticity is that, the resulting tangent operators are $d^2 \times d^2$ (assuming that, we flatten fourth order tensors column-wise) instead of their usual Voigt notations (3×3 in 2-dimensions and 6×6 in 3-dimensions). However, Voigt matrices can easily be retrieved once PSD projection has been performed using the standard Voigt transformations; c.f. Bonet et al. [52] (page 245). From computer implementation standpoint, many performance-oriented optimisations are possible in order to directly create stabilised Voigt matrices; see Appendix C.

4.4. Polyconvex large strain elasticity

An alternative formulation for large strain elasticity that has surfaced in recent years is the polyconvex formulations which is based on the approximation of the minors of the deformation gradient or the fundamental kinematic measures namely $\{\mathbf{F}, \mathbf{H}, \mathbf{J}\}$, mainly attributed to the work of Schröder et al. [6] and Bonet et al. [7]. These formulations have a compelling property in that, they are by-design (*ab-initio*) materially stable i.e. their corresponding constitutive tangent operators are guaranteed to be PSD.

Moreover, polyconvexity ensures the well-posedness of the boundary-value problem in the large strain regime [6–8,50,55–57]. However, polyconvex models do not preclude geometric instabilities implying that the geometric stiffness operators obtained using these formulations can still be indefinite. In this section, we extend the tangent stabilisation of [13] to the case of polyconvex formulations wherein once again closed-form eigensystems can be obtained for the geometric stiffness term emerging from these formulations. Hence, giving rise to the notion of geometrically stable large strain polyconvex elasticity.

Following Bonet et al. [7], the internal energy density e can be written in terms of the fundamental kinematics as

$$e(\nabla_0 \mathbf{x}) = \tilde{W}(\mathbf{F}, \mathbf{H}, \mathbf{J}), \quad (68)$$

where \tilde{W} represents a polyconvex energy function in terms of the extended set of arguments $\mathcal{A}_{FHJ} = \{\mathbf{F}, \mathbf{H}, \mathbf{J}\}$. Similar to our earlier development, it is once again, convenient to introduce a set of work-conjugate stresses $S_{FHJ} = \{\Sigma_F, \Sigma_H, \Sigma_J\}$

$$\Sigma_F = \frac{\partial \tilde{W}}{\partial \mathbf{F}}, \quad \Sigma_H = \frac{\partial \tilde{W}}{\partial \mathbf{H}}, \quad \Sigma_J = \frac{\partial \tilde{W}}{\partial \mathbf{J}}, \quad (69)$$

which can now be used to obtain the more standard first Piola–Kirchhoff stress tensor \mathbf{P} [7]

$$\mathbf{P} = \Sigma_F + \Sigma_H \times \mathbf{F} + \Sigma_J \mathbf{H}. \quad (70)$$

In the context of second order optimisation methods, such as Newton–Raphson-like schemes, the tangent elasticity operator $\tilde{\mathbf{C}}$ is also often required which can be obtained through similar directional derivatives as an algebraic summation of constitutive $\tilde{\mathbf{C}}_k$ and geometric stiffness $\tilde{\mathbf{C}}_p$ components [7]

$$\begin{aligned} \tilde{\mathbf{C}} &= \tilde{\mathbf{C}}_k + \tilde{\mathbf{C}}_p = [\mathbb{I}, \quad \mathbb{I} \times \mathbf{F}, \quad \mathbf{H}] [\mathbb{H}_W] \underbrace{\begin{bmatrix} \mathbb{I} \\ \mathbb{I} \times \mathbf{F} \\ \mathbf{H} \end{bmatrix}}_{\tilde{\mathbf{c}}_k} + \underbrace{\mathbb{I} \times (\Sigma_H + \Sigma_J \mathbf{F})}_{\tilde{\mathbf{c}}_p} \\ &= \tilde{W}_{FF} + \mathbf{F} \times (\tilde{W}_{HH} \times \mathbf{F}) + \tilde{W}_{JJ} \mathbf{H} \otimes \mathbf{H} + 2(\tilde{W}_{FH} \times \mathbf{F})^{\text{sym}} \\ &\quad + 2(\tilde{W}_{FJ} \otimes \mathbf{H})^{\text{sym}} + 2((\mathbf{F} \times \tilde{W}_{HJ}) \otimes \mathbf{H})^{\text{sym}} + \tilde{\mathbf{C}}_p, \end{aligned} \quad (71)$$

where \mathbb{H}_W represents the symmetric positive definite Hessian operator given by

$$[\mathbb{H}_W] = \begin{bmatrix} \partial_{FF}^2 \tilde{W} & \partial_{FH}^2 \tilde{W} & \partial_{FJ}^2 \tilde{W} \\ \partial_{HF}^2 \tilde{W} & \partial_{HH}^2 \tilde{W} & \partial_{HJ}^2 \tilde{W} \\ \partial_{JF}^2 \tilde{W} & \partial_{JH}^2 \tilde{W} & \partial_{JJ}^2 \tilde{W} \end{bmatrix}, \quad (72)$$

where $\partial_{AB}^2 \tilde{W} = \frac{\partial^2 \tilde{W}}{\partial A \partial B}$, where \mathbf{A} and \mathbf{B} can represent any two elements from the set \mathcal{A}_{FHJ} . Similar to (18) and (42), a sufficient condition for convexity is then positive semi-definiteness of (71) via the constitutive $\tilde{\mathbf{C}}_k$ (via the Hessian \mathbb{H}_W) and geometric $\tilde{\mathbf{C}}_p$ contributions.

It can be proven that for polyconvex models the constitutive tangent operator $\tilde{\mathbf{C}}_k$ emanating from \mathbb{H}_W is always PSD; see [7,13]. As a result, polyconvex formulation remains the only formulation that truly separates material instabilities encapsulated in the constitutive term $\tilde{\mathbf{C}}_k$ from geometric instabilities included in the initial stress term $\tilde{\mathbf{C}}_p$. It is however, well-known that polyconvexity does not entail convexity with respect to \mathbf{F} [13] that is, the term $\tilde{\mathbf{C}}_p$ is not necessarily PSD.

4.5. Stabilisation of geometric stiffness in polyconvex large strain elasticity

Our goal now is to come up with a stabilisation strategy for the geometric stiffness term $\tilde{\mathbf{C}}_p$. To do so, it is necessary to recall that, in polyconvex formulations the kinematics measures \mathbf{F} , \mathbf{H} and J are often treated as separate variables whose information typically cannot be retrieved solely from \mathbf{F} . In particular, in the context of variational formulations two separate approaches can be followed. The mixed Hu–Washizu variational formulation [6,7] and the standard displacement-based formulation [7,32]. Although we are yet to discuss variational formulations, it is nevertheless important to keep this in mind. We consider the mixed Hu–Washizu variational formulation first where the kinematic measures are treated as separate variables. The stabilisation then follows closely the algebra developed so far by choosing to split the geometric stiffness $\tilde{\mathbf{C}}_p$ term in (71) into two as

$$\tilde{\mathbf{C}}_p = \tilde{\mathbf{C}}_{p_1} + \tilde{\mathbf{C}}_{p_2} = \mathbb{I} \times \Sigma_H + \mathbb{I} \times \Sigma_J \mathbf{F}. \quad (73)$$

This split is mathematically accurate as discussed shortly. Considering the second term first

$$\tilde{\mathbf{C}}_{p_2} = \mathbb{I} \times \Sigma_J \mathbf{F}, \quad (74)$$

which is in fact the Hessian of J i.e. $\mathbb{I} \times \mathbf{F}$ scaled with Σ_J . The analytical eigensystem of J has already been described in [13] (i.e. via our \mathbf{F} -based formulation) and can be obtained in a straight-forward fashion using the same arrangement. In 3-dimensions all

the 9 eigenvalues of the geometric stiffness can be written concisely as

$$\bar{\lambda}_{1,2,3}^{\tilde{C}_{p_2}} = \Sigma_J \left(\sqrt[3]{J + \sqrt{J^2 - \frac{(\mathbf{F} : \mathbf{F})^3}{27}}} + \sqrt[3]{J - \sqrt{J^2 - \frac{(\mathbf{F} : \mathbf{F})^3}{27}}} \right), \quad (75a)$$

$$\bar{\lambda}_4^{\tilde{C}_{p_2}} = -\Sigma_J \lambda_1, \quad \bar{\lambda}_5^{\tilde{C}_{p_2}} = -\Sigma_J \lambda_2, \quad \bar{\lambda}_6^{\tilde{C}_{p_2}} = -\Sigma_J \lambda_3, \quad \bar{\lambda}_7^{\tilde{C}_{p_2}} = \Sigma_J \lambda_1, \quad \bar{\lambda}_8^{\tilde{C}_{p_2}} = \Sigma_J \lambda_2, \quad \bar{\lambda}_9^{\tilde{C}_{p_2}} = \Sigma_J \lambda_3, \quad (75b)$$

where the first 3 eigenvalues are the 3 roots of the cubic expression above which always compute to real numbers.²

Remark 3. In terms of singular-value formulation of Poya et al. [13] the term $\Sigma_J \mathbb{I} \times \mathbf{F}$ also contributes in the \mathbf{H}_W operator of (19) which can be written as

$$\mathbf{H}_W^{\tilde{C}_{p_2}} = \Sigma_J \begin{bmatrix} 0 & \lambda_3 & \lambda_2 \\ 0 & \lambda_1 & \\ sym & 0 \end{bmatrix}. \quad (76)$$

The first 3 eigenvalues of $\Sigma_J \mathbb{I} \times \mathbf{F}$ in (75) are indeed the eigenvalues of above $\mathbf{H}_W^{\tilde{C}_{p_2}}$ matrix or alternatively the roots of its characteristic equation

$$(\bar{\lambda}_i^{\tilde{C}_{p_2}})^3 - \Sigma_J^2 (\mathbf{F} : \mathbf{F}) (\bar{\lambda}_i^{\tilde{C}_{p_2}}) + 2J \Sigma_J^3 = 0, \quad \text{for } i = 1, 2, 3. \quad (77)$$

Using this approach, the eigenvectors of $\mathbf{H}_W^{\tilde{C}_{p_2}}$ are also needed which can be found analytically [13] exactly similar to (29) and (51) for PSD projection

$$\mathbf{H}_W^{\tilde{C}_{p_2}^{\text{PSD}}} = \sum_{i=1}^d \max(\bar{\lambda}_i^{\mathbf{H}_W^{\tilde{C}_{p_2}}}, 0) [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i \otimes [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i, \quad (78)$$

where $\bar{\lambda}_i^{\mathbf{H}_W^{\tilde{C}_{p_2}}}$ are the eigenvalues and $[\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i$ are the eigenvectors of the tensor $\mathbf{H}_W^{\tilde{C}_{p_2}}$.

In 2-dimensions much simpler expressions are found for the eigenvalues of \tilde{C}_{p_2} . There are 4 with multiplicity of 2 and the first 2 are eigenvalues of $\mathbf{H}_W^{\tilde{C}_{p_2}} = \Sigma_J [0, 1; 1, 0]$, (with eigenvectors $[\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_1 = \frac{1}{2}[1, 1]^T$ and $[\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_2 = \frac{1}{2}[-1, 1]^T$)

$$\bar{\lambda}_1^{\tilde{C}_{p_2}} = \Sigma_J, \quad \bar{\lambda}_2^{\tilde{C}_{p_2}} = -\Sigma_J, \quad \bar{\lambda}_3^{\tilde{C}_{p_2}} = -\Sigma_J, \quad \bar{\lambda}_4^{\tilde{C}_{p_2}} = \Sigma_J. \quad (79)$$

The term \tilde{C}_{p_2} can finally be reconstructed from its eigen-decomposition in 2 and 3 dimensions, respectively by noting the eigenmatrices presented for the \mathbf{F} -based formulation as

$$\tilde{C}_{p_2} = \sum_{i=1}^2 \sum_{j=1}^2 \bar{\lambda}_i^{\tilde{C}_{p_2}} [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j \otimes [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j + \bar{\lambda}_3^{\tilde{C}_{p_2}} [\mathbf{L}]_1 \otimes [\mathbf{L}]_1 + \bar{\lambda}_4^{\tilde{C}_{p_2}} [\mathbf{T}]_1 \otimes [\mathbf{T}]_1, \quad \text{in 2D}, \quad (80a)$$

$$\tilde{C}_{p_2} = \sum_{i=1}^3 \sum_{j=1}^3 \bar{\lambda}_i^{\tilde{C}_{p_2}} [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j \otimes [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j + \sum_{i=4}^6 \bar{\lambda}_i^{\tilde{C}_{p_2}} [\mathbf{L}]_i \otimes [\mathbf{L}]_i + \sum_{i=7}^9 \bar{\lambda}_i^{\tilde{C}_{p_2}} [\mathbf{T}]_i \otimes [\mathbf{T}]_i, \quad \text{in 3D}. \quad (80b)$$

Alternatively, the PSD (stabilised) version of \tilde{C}_{p_2} can be constructed in 2 and 3 dimensions, respectively as

$$\begin{aligned} \tilde{C}_{p_2}^{\text{PSD}} = & \sum_{i=1}^2 \sum_{j=1}^2 \max(\bar{\lambda}_i^{\tilde{C}_{p_2}}, 0) [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j \otimes [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j \\ & + \max(\bar{\lambda}_3^{\tilde{C}_{p_2}}, 0) [\mathbf{L}]_1 \otimes [\mathbf{L}]_1 + \max(\bar{\lambda}_4^{\tilde{C}_{p_2}}, 0) [\mathbf{T}]_1 \otimes [\mathbf{T}]_1, \quad \text{in 2D}, \end{aligned} \quad (81a)$$

$$\begin{aligned} \tilde{C}_{p_2}^{\text{PSD}} = & \sum_{i=1}^3 \sum_{j=1}^3 \max(\bar{\lambda}_i^{\tilde{C}_{p_2}}, 0) [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j \otimes [\mathbf{e}^{\mathbf{H}_W^{\tilde{C}_{p_2}}}]_i^j [\mathbf{D}]_j \\ & + \sum_{i=4}^6 \max(\bar{\lambda}_i^{\tilde{C}_{p_2}}, 0) [\mathbf{L}]_i \otimes [\mathbf{L}]_i + \sum_{i=7}^9 \max(\bar{\lambda}_i^{\tilde{C}_{p_2}}, 0) [\mathbf{T}]_i \otimes [\mathbf{T}]_i, \quad \text{in 3D}. \end{aligned} \quad (81b)$$

To obtain the eigensystem of the first term in the geometric stiffness namely

$$\tilde{C}_{p_1} = \mathbb{I} \times \Sigma_H, \quad (82)$$

² The 3 cubic roots of a complex number $z = x + yi$ is given by

$$r_k = \sqrt[3]{|z|} (\cos(\phi_k) + \sin(\phi_k)i), \quad \phi_k = \frac{1}{3}\text{atan}2(y, x) + \frac{2\pi k}{3}, \quad k = 1, 2, 3.$$

it is important to note that, in polyconvex elasticity, the term Σ_H is also independent variable whose information cannot be retrieved from the geometry i.e. the deformation gradient. This is also the reason why it is possible to treat the two terms in the geometric stiffness separately as they do not span the same eigen-space. Hence, intuitively we need the Singular-Value-Decomposition (SVD) of Σ_H first

$$\Sigma_H = \tilde{U} \Lambda_{\Sigma_H} \tilde{V}^T, \quad (83)$$

where \tilde{U} and \tilde{V} are orthogonal tensors namely left and right singular-matrices, respectively and the tensor Λ_{Σ_H} encodes the singular values of Σ_H i.e. the principal work-conjugate stresses namely $\lambda_1^{\Sigma_H} \geq \lambda_2^{\Sigma_H} \geq \lambda_3^{\Sigma_H} \geq 0$ such that $\lambda_i^{\Sigma_H} = \Lambda_{\Sigma_H}_{ii}$. It follows that, the eigensystem of $\tilde{C}_{p_1} = \tilde{U} \otimes \Sigma_H$ is indeed the eigensystem of Hessian of $\det(\Sigma_H) = \tilde{J}$ which in 3-dimensions can be written as

$$\tilde{\lambda}_{1,2,3}^{\tilde{C}_{p_1}} = \sqrt[3]{\tilde{J}} + \sqrt[3]{\tilde{J}^2 - \frac{(\Sigma_H : \Sigma_H)^3}{27}}, \quad (84a)$$

$$\tilde{\lambda}_4^{\tilde{C}_{p_1}} = -\lambda_1^{\Sigma_H}, \quad \tilde{\lambda}_5^{\tilde{C}_{p_1}} = -\lambda_2^{\Sigma_H}, \quad \tilde{\lambda}_6^{\tilde{C}_{p_1}} = -\lambda_3^{\Sigma_H}, \quad \tilde{\lambda}_7^{\tilde{C}_{p_1}} = \lambda_1^{\Sigma_H}, \quad \tilde{\lambda}_8^{\tilde{C}_{p_1}} = \lambda_2^{\Sigma_H}, \quad \tilde{\lambda}_9^{\tilde{C}_{p_1}} = \lambda_3^{\Sigma_H}. \quad (84b)$$

Once again, in terms of singular-value formulation of Poya et al. [13] the term $\tilde{U} \otimes \Sigma_H$ contributes a $d \times d$ whose eigenvalues are indeed the first 3 above eigenvalues

$$H_W^{\tilde{C}_{p_1}} = \begin{bmatrix} 0 & \lambda_3^{\Sigma_H} & \lambda_2^{\Sigma_H} \\ & 0 & \lambda_1^{\Sigma_H} \\ sym & & 0 \end{bmatrix}, \quad H_W^{\tilde{C}_{p_1}^{PSD}} = \sum_{i=1}^d \max(\tilde{\lambda}_i^{H_W^{\tilde{C}_{p_1}}}, 0) [e^{H_W^{\tilde{C}_{p_1}}}]_i \otimes [e^{H_W^{\tilde{C}_{p_1}}}]_i, \quad (85)$$

where $\tilde{\lambda}_i^{H_W^{\tilde{C}_{p_1}}}$ are the eigenvalues and $[e^{H_W^{\tilde{C}_{p_1}}}]_i$ are the eigenvectors of the tensor $H_W^{\tilde{C}_{p_1}}$.

In 2-dimensions much simpler expressions are found for the eigenvalues of \tilde{C}_{p_1} . There are 4 with multiplicity of 2 and the first 2 are eigenvalues of $H_W^{\tilde{C}_{p_1}} = [0, 1; 1, 0]$, (with eigenvectors $[e^{H_W^{\tilde{C}_{p_1}}}]_1 = \frac{1}{2}[1, 1]^T$ and $[e^{H_W^{\tilde{C}_{p_1}}}]_2 = \frac{1}{2}[-1, 1]^T$)

$$\tilde{\lambda}_1^{\tilde{C}_{p_1}} = 1, \quad \tilde{\lambda}_2^{\tilde{C}_{p_1}} = -1, \quad \tilde{\lambda}_3^{\tilde{C}_{p_1}} = -1, \quad \tilde{\lambda}_4^{\tilde{C}_{p_1}} = 1. \quad (86)$$

Finally, the eigenmatrices have exactly similar structure as before namely *scaling*, *flip*, *twist* but emanating from the SVD of Σ_H which in 2-dimensions can be written as

$$[\tilde{D}]_1 = \tilde{U} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{D}]_2 = \tilde{U} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tilde{V}^T, \quad (87a)$$

$$[\tilde{L}]_1 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{T}]_1 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tilde{V}^T, \quad (87b)$$

and similarly in 3-dimensions

$$[\tilde{D}]_1 = \tilde{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{L}]_1 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{T}]_1 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tilde{V}^T, \quad (88a)$$

$$[\tilde{D}]_2 = \tilde{U} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{L}]_2 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{T}]_2 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tilde{V}^T, \quad (88b)$$

$$[\tilde{D}]_3 = \tilde{U} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{V}^T, \quad [\tilde{L}]_3 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tilde{V}^T, \quad [\tilde{T}]_3 = \frac{1}{\sqrt{2}} \tilde{U} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tilde{V}^T. \quad (88c)$$

The term \tilde{C}_{p_1} can then be reconstructed from its eigen-decomposition in 2 and 3 dimensions respectively as

$$\tilde{C}_{p_1} = \sum_{i=1}^2 \sum_{j=1}^2 \tilde{\lambda}_i^{\tilde{C}_{p_1}} [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \otimes [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \tilde{\lambda}_3^{\tilde{C}_{p_1}} [\tilde{L}]_1 \otimes [\tilde{L}]_1 + \tilde{\lambda}_4^{\tilde{C}_{p_1}} [\tilde{T}]_1 \otimes [\tilde{T}]_1, \quad \text{in 2D}, \quad (89a)$$

$$\tilde{C}_{p_1} = \sum_{i=1}^3 \sum_{j=1}^3 \tilde{\lambda}_i^{\tilde{C}_{p_1}} [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \otimes [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j + \sum_{i=4}^6 \tilde{\lambda}_i^{\tilde{C}_{p_1}} [\tilde{L}]_i \otimes [\tilde{L}]_i + \sum_{i=7}^9 \tilde{\lambda}_i^{\tilde{C}_{p_1}} [\tilde{T}]_i \otimes [\tilde{T}]_i, \quad \text{in 3D}. \quad (89b)$$

Alternatively, the PSD (stabilised) version of \tilde{C}_{p_1} can be constructed as

$$\begin{aligned} \tilde{C}_{p_1}^{\text{PSD}} &= \sum_{i=1}^2 \sum_{j=1}^2 \max(\tilde{\lambda}_i^{\tilde{C}_{p_1}}, 0) [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \otimes [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \\ &\quad + \max(\tilde{\lambda}_3^{\tilde{C}_{p_1}}, 0) [\tilde{L}]_1 \otimes [\tilde{L}]_1 + \max(\tilde{\lambda}_4^{\tilde{C}_{p_1}}, 0) [\tilde{T}]_1 \otimes [\tilde{T}]_1, \quad \text{in 2D}, \end{aligned} \quad (90a)$$

$$\begin{aligned} \tilde{C}_{p_1}^{\text{PSD}} &= \sum_{i=1}^3 \sum_{j=1}^3 \max(\tilde{\lambda}_i^{\tilde{C}_{p_1}}, 0) [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \otimes [e^{H_W^{\tilde{C}_{p_1}}}]_i^j [\tilde{D}]_j \end{aligned}$$

$$+ \sum_{i=4}^6 \max(\tilde{\lambda}_i^{\tilde{C}_{p_1}}, 0) [\tilde{L}]_i \otimes [\tilde{L}]_i + \sum_{i=7}^9 \max(\tilde{\lambda}_i^{\tilde{C}_{p_1}}, 0) [\tilde{T}]_i \otimes [\tilde{T}]_i, \quad \text{in 3D.} \quad (90\text{b})$$

Remark 4. In the context of displacement-based variational formulation Σ_H and F are co-axial and hence SVD of Σ_H is not necessary but the structure of \tilde{C}_{p_1} remains the same. In fact, for displacement-based variational formulation the eigensystem of \tilde{C}_{p_1} is structurally exactly the same and simply the eigenmatrices change from $[\tilde{D}]_i$, $[\tilde{L}]_i$ and $[\tilde{T}]_i$ to $[D]_i$, $[L]_i$ and $[T]_i$. The singular-values of Σ_H namely $\Lambda_{\Sigma_{Hii}} = \lambda_i^{\Sigma_H}$ from (83) needed in (84)–(86) are also straightforward to compute without performing an SVD as they are cross terms. In 2-dimensions they are simply

$$\lambda_1^{\Sigma_H} = \frac{\partial \tilde{W}}{\partial \lambda_2}, \quad \lambda_2^{\Sigma_H} = \frac{\partial \tilde{W}}{\partial \lambda_1} \quad (91)$$

and in 3-dimensions

$$\lambda_1^{\Sigma_H} = \frac{\partial \tilde{W}}{\partial (\lambda_2 \lambda_3)}, \quad \lambda_2^{\Sigma_H} = \frac{\partial \tilde{W}}{\partial (\lambda_1 \lambda_3)}, \quad \lambda_3^{\Sigma_H} = \frac{\partial \tilde{W}}{\partial (\lambda_1 \lambda_2)}. \quad (92)$$

Finally, in the case of displacement-based variational formulation it is possible to add the eigenvalues of \tilde{C}_{p_1} and \tilde{C}_{p_2} together (since their eigenmatrices are the same) to form the total geometric stiffness \tilde{C}_p in 2 and 3 dimensions, respectively as

$$\tilde{C}_p = \sum_{i=1}^2 \sum_{j=1}^2 \tilde{\lambda}_i^{\tilde{C}_p} [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j \otimes [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j + \tilde{\lambda}_3^{\tilde{C}_p} [L]_1 \otimes [L]_1 + \tilde{\lambda}_4^{\tilde{C}_p} [T]_1 \otimes [T]_1, \quad \text{in 2D,} \quad (93\text{a})$$

$$\tilde{C}_p = \sum_{i=1}^3 \sum_{j=1}^3 \tilde{\lambda}_i^{\tilde{C}_p} [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j \otimes [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j + \sum_{i=4}^6 \tilde{\lambda}_i^{\tilde{C}_p} [L]_i \otimes [L]_i + \sum_{i=7}^9 \tilde{\lambda}_i^{\tilde{C}_p} [T]_i \otimes [T]_i, \quad \text{in 3D.} \quad (93\text{b})$$

Alternatively, the PSD (stabilised) version of \tilde{C}_p can be constructed as

$$\begin{aligned} \tilde{C}_p^{\text{PSD}} = & \sum_{i=1}^2 \sum_{j=1}^2 \max(\tilde{\lambda}_i^{\tilde{C}_p}, 0) [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j \otimes [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j \\ & + \max(\tilde{\lambda}_3^{\tilde{C}_p}, 0) [L]_1 \otimes [L]_1 + \max(\tilde{\lambda}_4^{\tilde{C}_p}, 0) [T]_1 \otimes [T]_1, \quad \text{in 2D,} \end{aligned} \quad (94\text{a})$$

$$\begin{aligned} \tilde{C}_p^{\text{PSD}} = & \sum_{i=1}^3 \sum_{j=1}^3 \max(\tilde{\lambda}_i^{\tilde{C}_p}, 0) [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j \otimes [e^{H_W^{\tilde{C}_p}}]_i^j [D]_j \\ & + \sum_{i=4}^6 \max(\tilde{\lambda}_i^{\tilde{C}_p}, 0) [L]_i \otimes [L]_i + \sum_{i=7}^9 \max(\tilde{\lambda}_i^{\tilde{C}_p}, 0) [T]_i \otimes [T]_i, \quad \text{in 3D.} \end{aligned} \quad (94\text{b})$$

where $\tilde{\lambda}_i^{\tilde{C}_p} = \tilde{\lambda}_i^{\tilde{C}_{p_1}} + \tilde{\lambda}_i^{\tilde{C}_{p_2}}$ and $H_W^{\tilde{C}_p} = H_W^{\tilde{C}_{p_1}} + H_W^{\tilde{C}_{p_2}}$ with $[e^{H_W^{\tilde{C}_p}}]_i$ its corresponding eigenvectors.

Remark 5. It is also possible to consider yet another isotropic integrity basis or set of invariants based on the symmetric/objective strain measures for large strain isotropic polyconvex elasticity [7,9]. This in particular, entails the Cauchy–Green strain tensor C , its cofactor G , and its determinant C (presented earlier in Remark 1). To this end, we can express the polyconvex strain energy \tilde{W} as

$$e(\nabla_0 \mathbf{x}) = \tilde{W}(F, H, J) = \overline{\overline{W}}(C, G, C). \quad (95)$$

The procedure to find the analytic eigensystem for this set of invariants follows exactly the same as that of polyconvex formulation in terms of \mathcal{A}_{FHJ} bases. In particular, the geometric stiffness operator emerging from this formulations is $\mathbb{I} \times \Sigma_G + \Sigma_C \mathbb{I} \times C$ which is similar to (71).

4.6. Large strain isotropic elasticity using stretch tensor invariants

One of the most recent formulations for large strain isotropic elasticity presented in Smith et al. [17] uses the “lower order” stretch tensor (U) invariants (compared to Cauchy–Green invariants) as a new integrity basis defined as

$$e(\nabla_0 \mathbf{x}) = \phi(I_U, II_U, III_U) \quad (96)$$

where the set of invariants is defined as

$$I_U = \text{tr}(U) = \sum_{i=1}^d \lambda_i, \quad II_U = U : U = \sum_{i=1}^d \lambda_i^2, \quad III_U = J = \prod_{i=1}^d \lambda_i. \quad (97)$$

The analytic eigensystem of these invariants have already surfaced in [17,18,58] and in fact were the inspiration behind the work in Poya et al. [13]. We can however, find analytical eigensystems emanating from this integrity basis with the algebra developed thus far. Specifically noting that, consistent linearisation of ϕ yields

$$\frac{\partial \phi}{\partial \mathbf{F}} = \frac{\partial \phi}{\partial I_U} \mathbf{R} + 2 \frac{\partial \phi}{\partial II_U} \mathbf{F} + \frac{\partial \phi}{\partial III_U} \mathbf{H}, \quad (98a)$$

$$\frac{\partial^2 \phi}{\partial \mathbf{C} \partial \mathbf{C}} = [\mathbf{R}, \quad 2\mathbf{F}, \quad \mathbf{H}] [\mathbf{H}_W^\phi] \underbrace{\begin{bmatrix} \mathbf{R} \\ 2\mathbf{F} \\ \mathbf{H} \end{bmatrix}}_{\mathbf{C}_k^\phi} + \underbrace{\left(\frac{\partial \phi}{\partial I_U} \mathbb{T}_R + 2 \frac{\partial \phi}{\partial II_U} \mathbb{I} + \frac{\partial \phi}{\partial III_U} (\mathbb{I} \times \mathbf{F}) \right)}_{\mathbf{C}_p^\phi}. \quad (98b)$$

One clear advantage of these invariants is their ability to express rotation gradients in a spectrally-decomposed form. In the above expression \mathbb{T}_R is the rotation gradient which just is a scaled *twist* tensor as presented in [13,17] and \mathbf{H}_W^ϕ represents the symmetric (but not necessarily positive definite) $d \times d$ Hessian operator

$$\mathbf{H}_W^\phi = \begin{bmatrix} \frac{\partial^2 \phi}{\partial I_U \partial I_U} & \frac{\partial^2 \phi}{\partial I_U \partial II_U} & \frac{\partial^2 \phi}{\partial I_U \partial III_U} \\ & \frac{\partial^2 \phi}{\partial II_U \partial II_U} & \frac{\partial^2 \phi}{\partial II_U \partial III_U} \\ sym & & \frac{\partial^2 \phi}{\partial III_U \partial III_U} \end{bmatrix}. \quad (99)$$

Similar to (18), a sufficient condition for convexity is then positive semi-definiteness of (98b) via the constitutive \mathbf{C}_k^ϕ and geometric \mathbf{C}_p^ϕ contributions. The constitutive part once again, comprises of a $d \times d$ Hessian (\mathbf{H}_W^ϕ) whose eigen-decomposition is simple to find [13,17,58]. The eigensystem of the geometric stiffness once again contains $\mathbb{I} \times \mathbf{F}$ which represents the Hessian of J and was described earlier.

5. Stabilised large strain transversely isotropic elasticity

In this section we briefly consider the case of large strain transversely isotropic elasticity presented earlier in Section 3. It should be emphasised in advance that, transversely isotropic invariants of type considered in Section 3 are in general convex by themselves and do not require tangent stabilisation. However, functions of such invariants may not necessarily be convex. Furthermore, as will be shown in Appendix A depending on the variational formulation transversely isotropic invariants still require tangent stabilisation. In such cases, if necessary, their analytic eigensystem will have to be obtained on case by case basis. In the next subsections we consider different formulations for large strain transversely isotropic energies and show that analytic eigensystems can be found for common transversely isotropic energies without relying on SVD informations or SVD differentials unlike some prior work [59,60].

5.1. Large strain transversely isotropic elasticity based on Cauchy–Green invariants

Let us consider the two transversely invariants introduced in Section 3.

$$J_4 = \text{tr}(\mathbf{C}(\mathbf{M} \otimes \mathbf{M})), \quad J_5 = \text{tr}(\mathbf{C}^2(\mathbf{M} \otimes \mathbf{M})), \quad (100)$$

which are certainly expressed in terms of \mathbf{C} due to the anisotropic restriction imposed. Consistent linearisation of these invariants yield the first derivatives as

$$\frac{\partial J_4}{\partial \mathbf{C}} = \mathbf{M} \otimes \mathbf{M}, \quad \frac{\partial J_5}{\partial \mathbf{C}} = 2\mathbf{C}(\mathbf{M} \otimes \mathbf{M}) \quad (101a)$$

$$\frac{\partial^2 J_4}{\partial \mathbf{C} \partial \mathbf{C}} = \mathbf{0}, \quad \frac{\partial^2 J_5}{\partial \mathbf{C} \partial \mathbf{C}} = 2[\mathbb{I}]_{ILJM} [\mathbf{M} \otimes \mathbf{M}]_{JK} \quad (101b)$$

From (101) it is clear that both of these invariants are convex in \mathbf{C} in that, J_4 does not contribute in the tangent elasticity operator and J_5 has a constant tangent. Nevertheless, for our subsequent development, it is useful to obtain an analytical formula for the eigen-decomposition of tangent of J_5 . This can be found in a straight-forward fashion

$$\frac{\partial^2 J_5}{\partial \mathbf{C} \partial \mathbf{C}} = \sum_{i=1}^d 2[\mathbf{E}]_i \otimes [\mathbf{E}]_i \quad (102)$$

revealing that there are d equal and constant eigenvalues (all 2s) and the eigenmatrices namely the second order tensors $[\mathbf{E}]_i$ contain the transversely isotropic fibre orientation or rather the structural tensor \mathbf{M} as their i^{th} row which in 2-dimensions can be written as

$$[\mathbf{E}]_1 = \begin{bmatrix} \mathbf{M}^T \\ \mathbf{0}_{1 \times 2} \end{bmatrix}, \quad [\mathbf{E}]_2 = \begin{bmatrix} \mathbf{0}_{1 \times 2} \\ \mathbf{M}^T \end{bmatrix}, \quad (103)$$

and similarly in 3-dimensions

$$[E]_1 = \begin{bmatrix} \mathbf{M}^T \\ \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{1 \times 3} \end{bmatrix}, \quad [E]_2 = \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ \mathbf{M}^T \\ \mathbf{0}_{1 \times 3} \end{bmatrix}, \quad [E]_3 = \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{1 \times 3} \\ \mathbf{M}^T \end{bmatrix}. \quad (104)$$

Finally, we consider a generalised class of transversely isotropic energies expressed in terms of these two invariants namely, $\bar{W}^{ti}(J_4, J_5)$. Consistent linearisation of the energy yields

$$\frac{\partial \bar{W}^{ti}}{\partial \mathbf{C}} = \frac{\partial \bar{W}_{ti}}{\partial J_4} (\mathbf{M} \otimes \mathbf{M}) + 2 \frac{\partial \bar{W}^{ti}}{\partial J_5} \mathbf{C} (\mathbf{M} \otimes \mathbf{M}), \quad (105a)$$

$$\frac{\partial^2 \bar{W}^{ti}}{\partial \mathbf{C} \partial \mathbf{C}} = [(\mathbf{M} \otimes \mathbf{M}), \quad 2\mathbf{C} (\mathbf{M} \otimes \mathbf{M})] [\mathbf{H}_{\bar{W}^{ti}}] \begin{bmatrix} (\mathbf{M} \otimes \mathbf{M}) \\ 2\mathbf{C} (\mathbf{M} \otimes \mathbf{M}) \end{bmatrix} + 2 \frac{\partial \bar{W}^{ti}}{\partial J_5} [\mathbb{I}]_{ILJM} [\mathbf{M} \otimes \mathbf{M}]_{JK} \quad (105b)$$

where $\mathbf{H}_{\bar{W}^{ti}}$ represents the 2×2 symmetric (but necessarily positive definite) Hessian operator

$$\mathbf{H}_{\bar{W}^{ti}} = \begin{bmatrix} \frac{\partial^2 \bar{W}^{ti}}{\partial J_4 \partial J_4} & \frac{\partial^2 \bar{W}^{ti}}{\partial J_4 \partial J_5} \\ sym & \frac{\partial^2 \bar{W}^{ti}}{\partial J_5 \partial J_5} \end{bmatrix}, \quad (106)$$

whose eigen-decomposition can be easily obtained analytically using the formula presented in [13].

5.2. Large strain transversely isotropic elasticity based on polyconvex invariants

Following Horák et al. [10], similar to J_4 and J_5 we can introduce their transversely isotropic polyconvex counterparts I_4 and I_5 as

$$I_4 = J_4 = (\mathbf{F}\mathbf{M}) \cdot (\mathbf{F}\mathbf{M}), \quad I_5 = (\mathbf{H}\mathbf{M}) \cdot (\mathbf{H}\mathbf{M}), \quad (107)$$

Similarly, if we consider a generalised class of transversely isotropic energy expressed in terms of these two invariants namely, $\bar{W}^{ti}(I_4, I_5)$. Keeping in mind that, \mathbf{F} and \mathbf{H} are independent polyconvex variables, consistent linearisation of the energy yields [10, 61]

$$\frac{\partial \bar{W}^{ti}}{\partial \mathbf{F}} = 2\mathbf{F} (\mathbf{M} \otimes \mathbf{M}), \quad \frac{\partial \bar{W}^{ti}}{\partial \mathbf{H}} = 2\mathbf{H} (\mathbf{M} \otimes \mathbf{M}), \quad (108a)$$

$$\frac{\partial^2 \bar{W}^{ti}}{\partial \mathbf{F} \partial \mathbf{F}} = 2[\mathbb{I}]_{ILJM} [\mathbf{M} \otimes \mathbf{M}]_{JK}, \quad \frac{\partial^2 \bar{W}^{ti}}{\partial \mathbf{H} \partial \mathbf{H}} = 2[\mathbb{I}]_{ILJM} [\mathbf{M} \otimes \mathbf{M}]_{JK}, \quad (108b)$$

which are all PSD and further do not contribute in the geometric stiffness operator in (71). As we will see in Appendix A however, depending on the variational formulation they still require tangent stabilisation. Noting the Hessian operator in (72) we can collapse the terms relating to \mathbf{F} and \mathbf{H} in (71) we obtain the constitutive tangent operator in a decomposed form

$$\tilde{\mathcal{C}}_k^{ti} = \sum_{i=1}^d 2[E]_i \otimes [E]_i + (\mathbb{I} \times \mathbf{F}) \left(\sum_{i=1}^d 2[E]_i \otimes [E]_i \right) (\mathbb{I} \times \mathbf{F}). \quad (109)$$

Before turning to the next section, it is worth noting that, all the algebra developed to handle analytic construction of tangents and their subsequent stabilisation is not computationally demanding, does not necessarily add extra overhead compared to implementations of classical elasticity and can be implemented extremely efficiently via static tensor types such as [62] as presented in [13, 18].

6. Discretisation-aware load stepping

In solid mechanics, usually coercive strain energies are used for elastic deformations [13, 18, 45] in order to ensure the assumption of no interpenetration of matter. Coercive strain energies are barrier energies that implicitly encode positivity of Jacobian (J). In other words, they disallow element inversion. However, under extreme scenarios element inversion still occurs since, in discrete setting, minimisation schemes such as Newton–Raphson minimise the deformation over the entire computational mesh in an “average” sense. This phenomenon becomes more pronounced with tangent stabilised elasticity as it unleashes extreme deformations way beyond conventional Newton–Raphson with full tangents/Hessians. A detailed analysis of element inversion and the possible remedies in the context of Newton–Raphson can be found in Poya et al. [18]. Hence, often a specialised and rather sophisticated numerical treatment is required to circumvent this issue and perform inversion-free simulations. One way of achieving this is through a line search scheme. However, conventional line search schemes that merely satisfy Wolfe’s condition are also incapable of guaranteeing element inversion when deformations are too large [63]. Inspired by the work of [63, 64], we can devise a discretisation-aware line search scheme that explicitly checks for geometric validity. However, unlike these earlier works that apply only to simplicial meshes we provide an approach for general polyhedral finite elements of arbitrary order. Assuming that the variational formulations and finite element discretisation have been carried out (see Appendices A and B) the update rule for Newton–Raphson procedure equipped with line search can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \Delta \mathbf{u}, \quad (110)$$

where k is the iteration counter and α the line search parameter or the step size. If we write this equation for a single quadrature point (q) of a finite element, we obtain

$$\mathbf{x}_{k+1}^q = \mathbf{x}_k^q + \alpha^q \Delta \mathbf{u}^q. \quad (111)$$

Taking derivatives yields

$$\nabla_0 \mathbf{x}_{k+1}^q = \nabla_0 (\mathbf{x}_k^q + \alpha^q \Delta \mathbf{u}^q) = \nabla_0 \mathbf{x}_k^q + \alpha^q \nabla_0 \Delta \mathbf{u}^q \quad (112)$$

Noting that, $\mathbf{F}_k = \nabla_0 \mathbf{x}_k^q$ is the deformation gradient at previous iteration (i.e. iteration k) and $\mathbf{F}_{k+1} = \nabla_0 \mathbf{x}_{k+1}^q$ the deformation gradient at current iteration (i.e. iteration $k+1$) and further denoting $\bar{\mathbf{F}} = \nabla_0 \Delta \mathbf{u}^q$ and its determinant as $\det(\Delta \mathbf{u}^q) = \bar{J}$ we have

$$\mathbf{F}_{k+1} = \mathbf{F}_k + \alpha^q \bar{\mathbf{F}}. \quad (113)$$

The goal now is to find an α^q that drives the current iteration Jacobian to zero i.e. $J_{k+1} = \det(\mathbf{F}_{k+1}) = 0$. In other words, the problem of finding the right step size can be posed as a maximisation problem on per-quadrature basis as

$$(\alpha^q)^* = \arg \max_{\alpha^q} \left\{ \det (\mathbf{F}_k + \alpha^q \bar{\mathbf{F}}) \right\}, \quad (114)$$

where $(\alpha^q)^*$ is the optimal step size. In the context of solid mechanics, this maximisation problem is well-defined as the starting point is always within the feasible region (geometric configuration/mesh (or the one at previous iteration) is always assumed inversion-free). Fortunately, the above functional involves either 2×2 or 3×3 matrices for which (i.e. for the determinant of sums) simple closed-form expressions exist, in particular

$$\det (\mathbf{F}_k + \alpha^q \bar{\mathbf{F}}) = J_k + \alpha^q (\bar{J} + \mathbf{H}_k : \bar{\mathbf{F}}), \quad \text{in 2D} \quad (115a)$$

$$\det (\mathbf{F}_k + \alpha^q \bar{\mathbf{F}}) = J_k + \alpha^q (\bar{J} + \mathbf{H}_k : \bar{\mathbf{F}}) + (\alpha^q)^2 (\bar{\mathbf{H}} : \mathbf{F}_k), \quad \text{in 3D} \quad (115b)$$

where $J_k = \det(\mathbf{F}_k)$ and $\mathbf{H}_k = \mathbf{J}_k \mathbf{F}_k^{-T} = \frac{1}{2} \mathbf{F}_k \times \mathbf{F}_k$ are the Jacobian and the cofactor at previous iteration, respectively. The above expressions can be solved for α^q directly as they are linear and quadratic functions. The maximum affordable step sizes (α^q s) are computed for all quadrature points and their minimum over the entire computational mesh is chosen as the safe global step size (α). Once determined, α is further passed as the maximum allowed step size to a standard line search technique to further check for Wolfe's condition. The line search algorithm might further decrease α . Note that, when determining α^q in 3-dimension, the quadratic equation above can have complex roots. Fortunately, unlike the arc-length methods, one is not spoilt for choice here. A complex root simply implies that $J_{k+1} = 0$ or collapsed configuration is infeasible. Such values including negative values of α are simply discarded. It is worth mentioning that, in sharp contrast to [63,64] techniques which are applicable to only linear/simplicial meshes our technique can be applied to general polyhedral finite elements and further requires only a linear functional in 2-dimensions and a quadratic in 3-dimensions (as opposed to quadratic in 2-dimensions and cubic in 3-dimensions).

In engineering analysis, continuation is a well-established practice as often it is not possible to apply the full load in one step within a nonlinear solution [19,20,53]. Tangent stabilised elasticity however, facilitates applying external loads (Neumann boundary conditions) in a single step by further relying on the above discretisation-aware line search scheme. For imposed displacements via Dirichlet boundary conditions however, situations can arise where applying the full imposed displacements in a single step might cause element-inversion and mesh invalidity. In such cases, it is understandable to rely on increments. Fortunately, an automatic way to determine the number of load increments (or rather the load factor) can be established using the same approach followed for the line search scheme. Recall that, the geometry update via imposed Dirichlet boundary condition at any given increment of the continuation can be written

$$\mathbf{x}^n = \mathbf{x}^{n-1} + \Delta \lambda_d \mathbf{u}_p, \quad (116)$$

where n is the increment counter, $\Delta \lambda_d$ the desired user-defined load factor increment (which is constant with equal load increments) and \mathbf{u}_p the total prescribed displacement. Instead of working with a constant $\Delta \lambda_d$ which might be infeasible, similar to the above line search procedure, we can choose to explicitly solve for the load factor increment $\Delta \lambda^n$ by re-writing the above equation on a quadrature point level as

$$(\mathbf{x}^n)^q = (\mathbf{x}^{n-1})^q + (\Delta \lambda^n)^q \mathbf{u}_p^q, \quad (117)$$

where the solution to the right load factor increment can be posed as a maximisation problem resulting in a similar set of equations

$$\det (\mathbf{F}^n + (\Delta \lambda^n)^q \bar{\mathbf{F}}) = J^n + (\Delta \lambda^n)^q (\bar{J} + \mathbf{H}^n : \bar{\mathbf{F}}), \quad \text{in 2D} \quad (118a)$$

$$\det (\mathbf{F}^n + (\Delta \lambda^n)^q \bar{\mathbf{F}}) = J^n + (\Delta \lambda^n)^q (\bar{J} + \mathbf{H}^n : \bar{\mathbf{F}}) + ((\Delta \lambda^n)^q)^2 (\bar{\mathbf{H}} : \mathbf{F}^n), \quad \text{in 3D} \quad (118b)$$

where $\mathbf{F}^n = \nabla_0(\mathbf{x}^n)^q$, $J^n = \det(\mathbf{F}^n)$ and $\mathbf{H}^n = J^n(\mathbf{F}^n)^{-T} = \frac{1}{2}\mathbf{F}^n \times \mathbf{F}^n$ are the deformation gradient, Jacobian and the cofactor at previous increment, respectively, and with slight abuse of notation $\bar{\mathbf{F}} = \nabla_0\mathbf{u}_p^q$. Like the *non-consistent* arc-length method, the value of global $(\Delta\lambda^n)^q$ is solved for explicitly but now by taking the minimum of their local per-quadrature points values. The right global $\Delta\lambda^n$ then limits the amount of load/Dirichlet BC at any given increment to the maximum discretisation/inversion-safe load factor. Certainly, cost-savings can be applied here by only traversing over elements which are on the boundary/interfaces where the loading is applied. This discretisation-aware continuation technique coupled with Projected Newton–Raphson is shown in Algorithms 1 and 2. For problems with no prestressing where assumed stored energy is continuously increasing during the course of minimisation, a simplified version of the line search presented in [13] can be articulated as shown in Algorithm 3.

Newton-Raphson with automatic safe load stepping

```

Data:  $X, \Delta\lambda_d, \lambda_{\max}, \text{thresholdr}, \text{thresholdc}$ 
Result:  $x$ 
1 Initialise  $n = 0$ 
2 while  $\lambda < \lambda_{\max}$  do
3   Initialise  $k = 0$ 
4   Assemble external forces  $\mathbf{F}$  using desired load factor  $\Delta\lambda_d$ 
5   Impose incremental Dirichlet BC using desired load factor  $\Delta\lambda_d$ 
6    $\Delta\lambda^n = \text{ComputeSafeLoadFactor}(\mathbf{x}^n, \Delta\lambda_d, \lambda_{\max})$ 
7   Re-assemble forces  $\mathbf{F}, \mathbf{R}$  with safe load factor  $\Delta\lambda^n$ 
8   while  $\|\mathbf{R}\| > \text{thresholdr}$  or  $k < \text{maxIter}$  do
9     Assemble internal forces and tangent stiffness  $\mathbf{R}, \mathbf{K}$ 
10    Add external forces  $\mathbf{R} = \mathbf{R} + \mathbf{F}$ 
11    Solve for displacements  $\mathbf{K}\Delta\mathbf{u} = -\mathbf{R}$ 
12    if  $|\mathbf{R} \cdot \Delta\mathbf{u}| < \text{thresholdc}$  then
13      | break
14    end if
15     $\alpha = \text{ComputeSafeStepSize}(\mathbf{x}_k, \Delta\mathbf{u})$ 
16     $\alpha = \text{LineSearch}(\mathbf{x}_k, \mathbf{R}, \Delta\mathbf{u}, \alpha)$ 
17     $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\Delta\mathbf{u}$ 
18     $k = k + 1$ 
19  end while
20   $\lambda = \lambda + \Delta\lambda^n$ 
21   $n = n + 1$ 
22 end while

```

Compute safe load factor

```

Data:  $\mathbf{x}^n, \Delta\lambda_d, \lambda_{\max}$ 
Result:  $\Delta\lambda^n$ 
1 Initialise  $\Delta\lambda^n = \text{DBLMAX}$ ,  $c = 0.9$ 
2 for  $e$  in elements do
3   for  $q$  in quadratures do
4     if  $\text{ndim} = 2$  then
5       | Solve  $J^n + (\Delta\lambda^n)^q (\bar{\mathbf{J}} + \mathbf{H}^n : \bar{\mathbf{F}})$  for  $(\Delta\lambda^n)^q$ 
6     else if  $\text{ndim} = 3$  then
7       | Solve  $J^n + (\Delta\lambda^n)^q (\bar{\mathbf{J}} + \mathbf{H}^n : \bar{\mathbf{F}}) + ((\Delta\lambda^n)^q)^2 (\bar{\mathbf{H}} : \mathbf{F}^n)$  for  $(\Delta\lambda^n)^q$ 
8     if  $\Delta\lambda^n > (\Delta\lambda^n)^q$  then
9       |  $\Delta\lambda^n = (\Delta\lambda^n)^q$ 
10      end if
11    end for
12 end for
13  $\Delta\lambda^n = c\Delta\lambda^n$ 

```

Line search procedure

```

Data:  $x_k, R, \Delta u, \alpha$ 
Result:  $\alpha$ 
1 Initialise  $m = 0, \text{maxIter} = 50, \alpha_{\min} = 10^{-16}, \rho = 0.5, c = 0.95$ 
2 Compute curvature at  $x_k$ :  $R_0 = R \cdot \Delta u$ 
3 while  $\alpha > \alpha_{\min}$  or  $m < \text{maxIter}$  do
4   Assemble residual at  $x_k + \alpha \Delta u$ :  $R_\alpha = R(x_k + \alpha \Delta u)$ 
5   Compute curvature at  $x_k + \alpha \Delta u$ :  $R_\alpha = R_\alpha \cdot \Delta u$ 
6   if  $R_\alpha \leq c R_0$  then
7     | break
8   end if
9    $\alpha = \rho \alpha$ 
10   $m = m + 1$ 
11 end while

```

7. Numerical examples

7.1. Comparison of all presented formulations of nonlinear elasticity in 2 and 3 dimensions

The goal of this example is to

1. Compare all classic implementations of nonlinear elasticity with the new ones proposed in this work in particular, for F -based, C -based and b -based formulations
2. Show that, the proposed spectrally-decomposed tangent constructions are not “approximation” and achieve the same quadratic Newton–Raphson convergence of classic implementations

To facilitate these comparisons, we have implemented the standard F -based, C -based and b -based formulations of nonlinear elasticity too. This is followed by their new implementations based on spectrally-decomposed tangent construction. With tangent stabilised elasticity, we choose to work with coercive (i.e. barrier) polyconvex energies of quasi-conformal type [see Lemma C.7 in Hartmann and Neff [45]]

$$e(\mathbf{F}) = \mu_1 \left(\frac{\mathbf{F} : \mathbf{F}}{3J^{2/3}} - 1 \right) + \mu_2 \left(\frac{\mathbf{H} : \mathbf{H}}{3J^{2/3}} - 1 \right) + \kappa (J + J^{-1} - 2), \quad (119)$$

which is a nearly-incompressible Mooney–Rivlin type model. The reason for choosing coercive energies in the context of tangent stabilised elasticity was laid out in [13], i.e. to guarantee inversion-free simulations. This is in addition to safe load stepping scheme presented earlier in Algorithm 1 and in fact complementary to it. In 2-dimensions, a well-known particularisation of this energy is available which we can write in terms of F -based, C -based and b -based formulations as

$$e(\mathbf{F}) = \mu \left(\frac{\mathbf{F} : \mathbf{F}}{2J} - 1 \right) + \kappa (J + J^{-1} - 2), \quad (120a)$$

$$\check{W}(\mathbf{C}) = \mu \left(\frac{\text{tr}(\mathbf{C})}{2\sqrt{C}} - 1 \right) + \kappa \left(\sqrt{C} + \sqrt{C}^{-1} - 2 \right), \quad (120b)$$

$$\bar{W}(\mathbf{b}) = \mu \left(\frac{\text{tr}(\mathbf{b})}{2\sqrt{b}} - 1 \right) + \kappa \left(\sqrt{b} + \sqrt{b}^{-1} - 2 \right), \quad (120c)$$

The analytic eigensystem of this energy in 2-dimensions was already worked out in [13] for F -based formulations and eigen-spectrum was extensively analysed; see also [18]. We show the analytic eigensystem for the C -based counterpart which can be obtained easily using the algebra developed in the previous sections. The first two eigenvalues of constitutive tangent namely $\bar{\lambda}_1^{\check{C}_k}$ and $\bar{\lambda}_2^{\check{C}_k}$ are found from the Hessian matrix

$$\mathbf{H}_{\bar{W}} = \begin{bmatrix} \frac{3\text{tr}(\mathbf{C})\mu}{2\sqrt{C}\lambda_1^4} - \frac{2\mu}{\sqrt{C}\lambda_1^2} - \frac{\kappa}{\lambda_1^4} \left(\sqrt{C} - 3\sqrt{C}^{-1} \right) & \kappa \left(\frac{1}{\sqrt{C}} + \frac{1}{(\sqrt{C})^3} \right) - \frac{\text{tr}(\mathbf{C})\mu}{2(\sqrt{C})^3} \\ \text{sym} & \frac{3\text{tr}(\mathbf{C})\mu}{2\sqrt{C}\lambda_2^4} - \frac{2\mu}{\sqrt{C}\lambda_2^2} - \frac{\kappa}{\lambda_2^4} \left(\sqrt{C} - 3\sqrt{C}^{-1} \right) \end{bmatrix}, \quad (121)$$

and the third one corresponds to flip mode which is

$$\bar{\lambda}_3^{\check{C}_k} = \frac{\text{tr}(\mathbf{C})\mu}{2(\sqrt{C})^3} - \kappa \frac{C - 1}{(\sqrt{C})^3}, \quad (122)$$

and finally the geometric stiffness eigenvalues are indeed the principal components of PK2

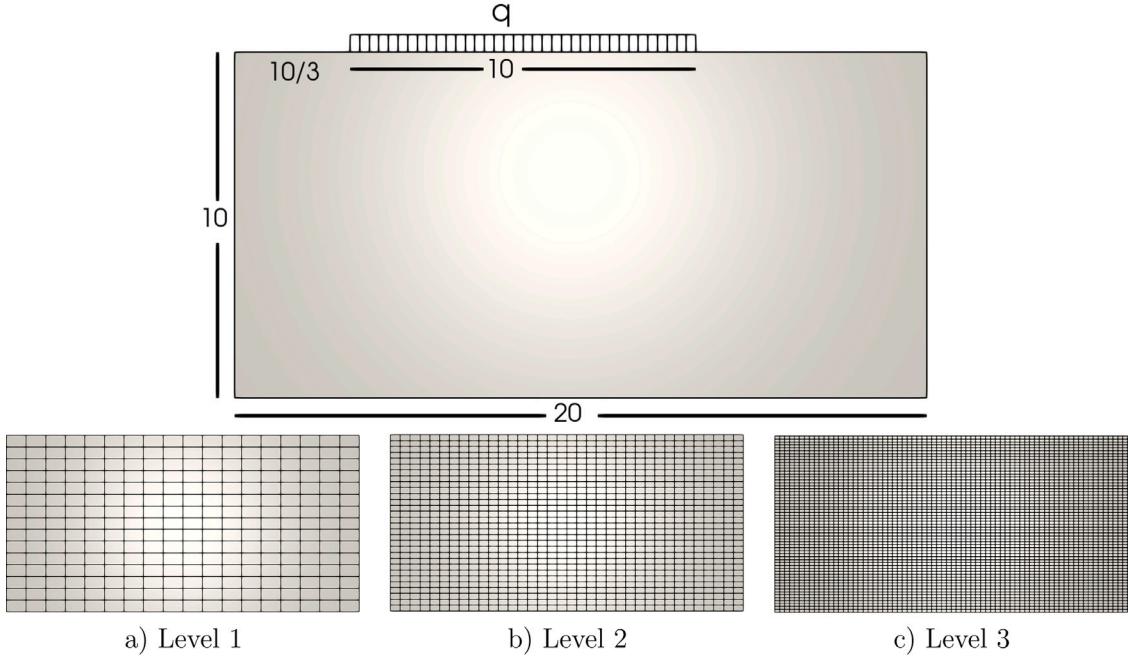


Fig. 3. Geometry and loading setup for the rectangular block specimen and the computational mesh with 3 refinement levels.

Table 1

Quadratic convergence of Newton–Raphson for the first increment for the 2-dimensional compression of rectangular block using new (spectrally-decomposed tangent) and classic implementations of *F*-based, *C*-based and *b*-based formulations. Results are in solid agreement validating all numerical implementations.

Formulation	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6
<i>F</i> -based Classic	1.00000e+00	1.31091e-01	3.58628e-02	7.82149e-04	4.75692e-07	6.46615e-11
<i>F</i> -based PS	1.00000e+00	1.31091e-01	3.58628e-02	7.82149e-04	4.75673e-07	6.89650e-11
<i>C</i> -based Classic	1.00000e+00	1.31091e-01	3.58628e-02	7.82149e-04	4.75706e-07	6.38668e-11
<i>C</i> -based PS	1.00000e+00	1.31091e-01	3.58628e-02	7.82149e-04	4.75707e-07	6.35543e-11
<i>b</i> -based Classic	1.00000e+00	1.31091e-01	3.58628e-02	7.82149e-04	4.75690e-07	5.56428e-11
<i>b</i> -based PS	1.00000e+00	1.31091e-01	3.58628e-02	7.82149e-04	4.75701e-07	5.87015e-11

$$\Lambda_{S_{11}} = \frac{\mu}{\sqrt{C}} - \frac{\text{tr}(C)\mu}{2\sqrt{C}\lambda_1^2} + \frac{\kappa}{\lambda_1^2} \left(\sqrt{C} - \sqrt{C}^{-1} \right), \quad (123a)$$

$$\Lambda_{S_{22}} = \frac{\mu}{\sqrt{C}} - \frac{\text{tr}(C)\mu}{2\sqrt{C}\lambda_2^2} + \frac{\kappa}{\lambda_2^2} \left(\sqrt{C} - \sqrt{C}^{-1} \right). \quad (123b)$$

Similarly, the eigensystem for *b*-based formulation can be obtained using the push-forward relationships from (65), (66) and (67). The 3-dimensional tangent eigensystem is longer but can be obtained as easily from the symbolic code provided in Fig. D.14.

For the first example, we consider compression of a rectangular block using asymmetric loading as shown Fig. 3. This example has been analysed in the past by others and is known to show a mild degree of localised failure [43] due to geometric instability. For the purpose of verifications, we first start with bilinear finite element ansatz functions for quadrilaterals. Three refinement levels are considered as will be shown in Fig. 3. Material properties used for the above coercive energy are: Shear modulus $\mu = 1000$ and Poisson's ratio $\nu = 0.49$.

We start the comparisons with applying a compressive load on the rectangular block and use the standard NR with 5 increments and check that all implementations achieve quadratic convergence by using a residual norm of 10^{-8} as stopping criterion. Tables 1 and 2 show that this is certainly the case. All implementations with spectrally-decomposed tangents (denoted as PS) achieve quadratic convergence and are in excellent agreement with classic implementations. Further, in Fig. 4 we plot the distribution of derived quantities namely, the deformation gradient component F_{xx} and the hydrostatic pressure p_{hyd} and verify that these quantities match for all formulations. Fig. 4 also shows the finest mesh (Level 3) for the geometry shown in Fig. 3 with 4356 quadrilateral elements 4490 nodes.

We then use the (Level 2) mesh and extrude it in the *Z* direction by unit length and re-perform the same experiment but now in 3-dimensions. Once again, Tables 3 and 4 show that, all implementations with spectrally-decomposed tangents achieve quadratic convergence and are in excellent agreement with classic implementations. Once again, in Fig. 5 we plot the distribution of derived

Table 2

Quadratic convergence of Newton–Raphson for last (5th) increment for 2-dimensional compression of rectangular block using new (spectrally-decomposed tangent) and classic implementations of F -based, C -based and b -based formulations. Results are in solid agreement validating all numerical implementations.

Formulation	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
F -based Classic	1.00000e+00	2.38769e-02	2.30346e-03	1.07726e-06	6.96135e-11
F -based PS	1.00000e+00	2.38769e-02	2.30346e-03	1.07722e-06	7.88531e-11
C -based Classic	1.00000e+00	2.38769e-02	2.30346e-03	1.07725e-06	7.31756e-11
C -based PS	1.00000e+00	2.38769e-02	2.30346e-03	1.07727e-06	7.85511e-11
b -based Classic	1.00000e+00	2.38769e-02	2.30346e-03	1.07726e-06	8.63721e-11
b -based PS	1.00000e+00	2.38769e-02	2.30346e-03	1.07730e-06	5.98287e-11

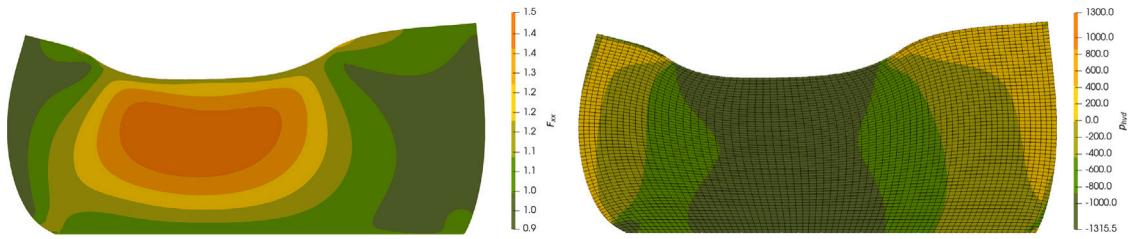


Fig. 4. F_{xx} and hydrostatic pressure p_{hyd} distribution for the 2-dimensional rectangular block from Tables 1 and 2. All implementations result in identical distribution hence, only b -based result is shown here.

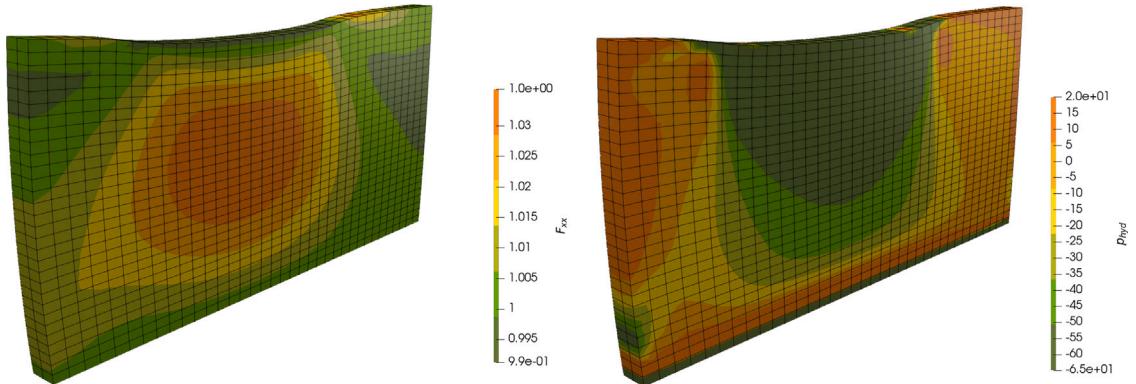


Fig. 5. F_{xx} and hydrostatic pressure p_{hyd} distribution for the 3-dimensional rectangular block from Tables 3 and 4. All implementations result in identical distribution hence, only b -based result is shown here.

quantities namely, the deformation gradient component F_{xx} and the hydrostatic pressure p_{hyd} and verify that these quantities match for all formulations in 3-dimensions too.

7.2. Verification of tangent stabilised elasticity

The goal of this example is to:

1. Verify the result of tangent stabilised elasticity with Projected Newton–Raphson (P-NR), with classical Newton–Raphson (NR) and arc-length Newton–Raphson (arc-length NR)
2. Validate the result and accuracy of tangent stabilised elasticity with external finite element suite

While, as shown in the previous example, the spectrally-decomposed tangent constructions achieve identical convergence to classic implementations, they come with an added benefit of analytically and easily being stabilised. Hence, we now shift the focus to clearly showcase the convergence characteristics of different formulations with P-NR using a single increment. To this end, we use the already described rectangular block example under compression with (Level 1) mesh and increase the loading

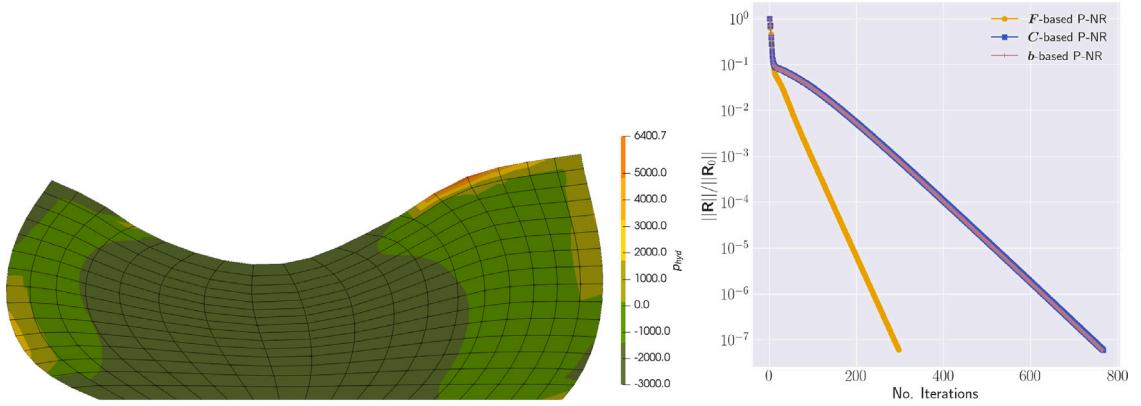


Fig. 6. Convergence of P-NR with a load factor for which NR with constant load factor fails (NR was tested up to 1000 increments). Figure confirms that, PSD projection does not work the same way between different formulations. In C - and b -based formulations constitutive and geometric components need to be projected separately resulting in a less accurate approximation of tangents. This is not the case with F -based formulation as there the geometric and constitutive operators are all “collapsed” into a single tangent operator. The performance of C - and b -based in general varies significantly depending on which eigen-modes are pruned.

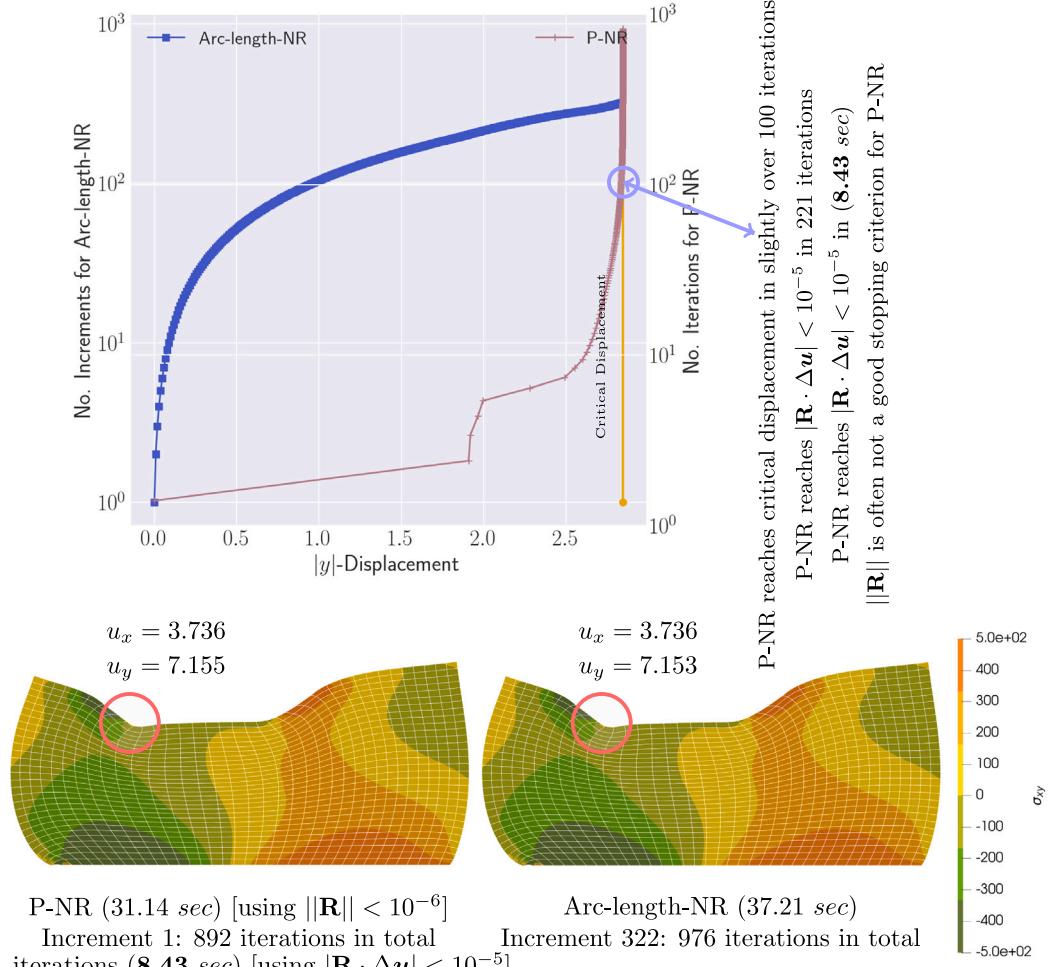


Fig. 7. Side-by-side comparison Arc length-NR and P-NR: P-NR uses the discretisation-aware line search from Algorithm 1 and reaches the critical displacement faster by over 4x. Arc-length was manually tuned to reach the critical load factor in the least number of increments. Both results are in good agreement.

Table 3

Quadratic convergence of Newton–Raphson for the first increment for the 3-dimensional compression of block using new (spectrally-decomposed tangent) and classic implementations of *F*-based, *C*-based and *b*-based formulations. Results are in solid agreement validating all numerical implementations.

Formulation	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
<i>F</i> -based Classic	1.00000e+00	2.78701e-02	8.52802e-03	2.66989e-05	1.75363e-09
<i>F</i> -based PS	1.00000e+00	2.78701e-02	8.52802e-03	2.66989e-05	1.74557e-09
<i>C</i> -based Classic	1.00000e+00	2.78701e-02	8.52802e-03	2.66989e-05	1.75040e-09
<i>C</i> -based PS	1.00000e+00	2.78701e-02	8.52802e-03	2.63867e-05	1.34560e-09
<i>b</i> -based Classic	1.00000e+00	2.78701e-02	8.52802e-03	2.66989e-05	1.75855e-09
<i>b</i> -based PS	1.00000e+00	2.78701e-02	8.52802e-03	2.63867e-05	1.34560e-09

Table 4

Quadratic convergence of Newton–Raphson for the last (5th) increment for the 3-dimensional compression of block using new (spectrally-decomposed tangent) and classic implementations of *F*-based, *C*-based and *b*-based formulations. Results are in solid agreement validating all numerical implementations.

Formulation	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6
<i>F</i> -based Classic	1.00000e+00	4.76957e-02	2.01982e-02	1.96365e-04	9.63069e-08	2.78094e-11
<i>F</i> -based PS	1.00000e+00	4.76957e-02	2.01982e-02	1.96365e-04	9.62955e-08	2.86818e-11
<i>C</i> -based Classic	1.00000e+00	4.76957e-02	2.01982e-02	1.96365e-04	9.63116e-08	3.08771e-11
<i>C</i> -based PS	1.00000e+00	4.76957e-02	2.01721e-02	1.89533e-04	7.22232e-08	1.61624e-10
<i>b</i> -based Classic	1.00000e+00	4.76957e-02	2.01982e-02	1.96365e-04	9.63003e-08	2.70922e-11
<i>b</i> -based PS	1.00000e+00	4.76957e-02	2.01723e-02	1.91047e-04	8.68032e-08	3.11638e-10

enough for standard NR (with constant load factor) to fail all the way up to 1000 increments and instead use P-NR to perform the simulation. Fig. 6 shows the deformed configuration and distribution of F_{xx} obtained with P-NR. The convergence of P-NR is shown in Fig. 6 which as can be observed is certainly not quadratic. However, it is worth noting that, standard NR fails for this case irrespective of number of load increments hence, this is more of a stress test for P-NR. If load increments and P-NR are combined the performance is certainly better as shown in [13]. Interestingly, as shown earlier, while spectrally-decomposed tangent constructions for all formulations produce the same convergence patterns the situation is not the same when tangents are stabilised. The answer to this is straightforward. In *C*- and *b*-based formulations constitutive and geometric components need to be projected separately resulting in a more-than-necessary stabilisation and hence, lesser accurate approximation of tangents. This is not the case with *F*-based formulation, as there, the geometric and constitutive tangents are all “collapsed” into one resulting in smaller stabilisation. The performance of *C*- and *b*-based in general varies significantly from example to example depending on the loading, geometry and material models and depending on which eigen-modes are pruned. The *F*-based P-NR on the other hand, has more of a uniform convergence pattern and remains the most competitive.

As noted in [43], the localised failure profile is only obtained with Q2 elements for this problem. To this end, we use the (Level 2) mesh and increase the finite element polynomial degree to 2 and redo the experiment with both arc length NR and P-NR. Fig. 7 shows a mild kink (highlighted in red circles) using both arc length NR and P-NR. The result of arc length NR and P-NR are in good agreements. Arc length-NR ultimately reaches a load factor of $\lambda \approx 1.1$ similar to [43] which is then used as a total load for P-NR. It can be observed that P-NR has a much faster convergence towards the solution through the course of iterations and reaches the expected result within a crude accuracy in slightly over a 100 iterations. For a fixed residual tolerance of 10^{-6} P-NR was still slightly faster than arc length (both over 30 s). Alternatively, if a curvature based stopping criterion is used as shown in Algorithm 1, 221 iterations of P-NR is sufficient to reach a tolerance of 10^{-5} , making it 4.4x faster than the arc-length-NR. The authors experience suggest that, for P-NR style algorithms residual based tolerances are often too tight implying that the stopping criterion of standard NR must be revisited for P-NR since through the course of iterations displacements drop rapidly. This makes curvature based stopping criterion more suited for P-NR. This example uses the discretisation-aware line search and the *F*-based formulation. The arc-length method uses a cylindrical constraint with secant path correction [22,23]. Arc-length was repeatedly tuned (with different user-defined starting load factor and number of increments) to reach the critical displacement in the least number of increments. P-NR needed no tuning.

The examples presented above demonstrate that P-NR captures the same deformation pattern as NR and arc length-NR with strong agreement. We now consider a simple example to emphasise the correctness of tangent stabilised elasticity in accurately capturing stable configurations. Fig. 8 shows example of a bar with aspect ratio 2:1 clamped on the left side, discretised with Q2 mesh under two loading scenarios namely, centric, and eccentric loading. The first three deformation modes are first obtained by performing linear modal analysis by evaluating the previously described nearly-incompressible Mooney–Rivlin energy (with $\mu = 1000$ and $\nu = 0.49$) at the origin. These deformation modes are shown in Fig. 8(a,b,c). When subjected to eccentric loading, both P-NR and arc length-NR accurately capture the well-known dominant buckling mode. However, under centric loading, arc length-NR produces a different profile that corresponds to the third dominant mode of the bar, while P-NR achieves the same buckling profile. With NR, in the presence of multiple potential solutions, the bar may randomly buckle in one direction or remain compressed in the central region. P-NR, on the other hand, effectively eliminates such bifurcations through stabilisation, resulting in a more deterministic simulation that captures the most dominant mode. This stabilising effect was also observed in a study by Poya et al. [13] (refer to

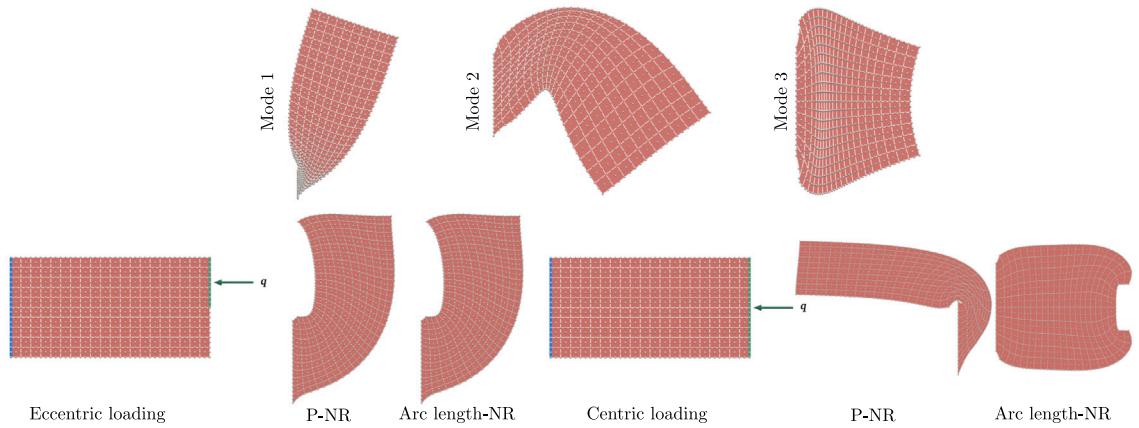


Fig. 8. Dominant buckling modes (top) and quasi-static simulations (bottom) of bar clamped on the left and under the action of centric (bottom left) and eccentric (bottom right) loading. Under eccentric loading, both P-NR and arc length-NR accurately capture the dominant buckling mode. Under centric loading, arc length-NR produces a different profile that corresponds to the third dominant mode of the bar, while P-NR captures the same dominant buckling mode by effectively eliminating bifurcation, resulting in a more deterministic simulation.

Figure 31), where P-NR consistently yielded the same buckling profile regardless of refinements in h (and p) and variations in finite element formulations (displacement vs mixed), whereas NR produced different profiles. Despite this, it should be mentioned that, due to complexity of loading scenarios, in general, it is not guaranteed that P-NR would *always* be the most dominant mode.

We finally consider a standard benchmark problem recommended by the National Agency for Finite Element Methods and Standards (NAFEMS) [65] namely the Z-section cantilever under torsional loading shown in Fig. 10. We perform this to validate the results of the new formulations with other established finite element suites. The cantilever is 10m long, the flanges are 1m wide, the web 2m wide and the thickness is uniform, 0.1m. The benchmark entails subjecting the structure to torsional moment of 2MNm, generated by the two forces uniformly distributed across the flanges, $S = 0.6\text{MN}$; see specification in [66]. The Young's modulus is taken as 210 GPa, and the Poisson ratio is 0.3. The target quantity is the mid-surface axial stress (σ_{xx}) at point A (at the edge of a flange, 2.5 m from the clamped end) of value -108 MPa (compressive) [65]. After calibrating material parameters, using the material model in (119) using Q2 hexahedral elements with 20 elements along the length, 10 across the flange and 1 across the thickness, we obtain $\sigma_{xx}^A = -107.113 \text{ MPa}$ (for all formulations), closely matching the reference value and hence validating the implementation and accuracy of tangent stabilised elasticity. Although not shown, we further refined the mesh with 100 elements along the length (keeping the refinement fixed in other directions) and compared this value with Abaqus results reported in [66]. We obtain, a value of $\sigma_{xx}^A = -111.048 \text{ MPa}$ vs Abaqus's shear flexible quadratic element (S8R) $\sigma_{xx}^A = -110.767 \text{ MPa}$ and shear rigid quadratic element (S8R5) $\sigma_{xx}^A = -111.103 \text{ MPa}$ further validating our implementations.

7.3. Automatic discretisation-safe load stepping

The goal of this example is to showcase the robustness of the discretisation-aware load stepping and line search from Algorithms 1 and 2. For this, we consider the case of displacement-driven loading. Fig. 9 shows the same rectangular block considered early but now with two $p = 2$ tetrahedral meshes that is pressed with imposed displacements applied in one increment. Such excess displacement is too large to be applied in one increment and causes element inversion near the boundary. As can be seen, the proposed discretisation-aware load stepping, described in Algorithms 1 and Algorithm 2, ensures that the loading is automatically adjusted to maximum inversion-safe displacements per increment. Further, it is noticeable that automatic load incrementation also improves the convergence of P-NR on a per-increment basis. From the bottom rows in Fig. 9 we can observe once again that, curvature drops much faster compared to pure residual since displacements disappear. An engineering accuracy is reached in all increments within 15 iterations. Top row in Fig. 9 shows distribution of Cauchy stress σ_{yz} for both refinements.

To showcase the robustness of discretisation-aware load stepping, we finally, consider an extreme hypothetical loading scenario where the NAFEM Z-section cantilever is pressed to half its length via displacements, imposed on the lower flange. This is too extreme of a displacement which causes buckling, structural collapse and element inversion if the load increments are not sufficiently small. As can be seen in Fig. 10, the discretisation-aware load stepping, described in Algorithms 1 and 2, ensures that the loading is automatically adjusted to maximum inversion-safe displacements per increment. The final result is inversion-free Q2 elements i.e. $J > 0$ when sampled with 27, 64, 125 point quadrature rules for hexahedral elements. The right plot in Fig. 10 shows the per-increment convergence of P-NR for all load factors determined by the algorithm. Convergence plots in Fig. 10 was obtained with C-based formulation.

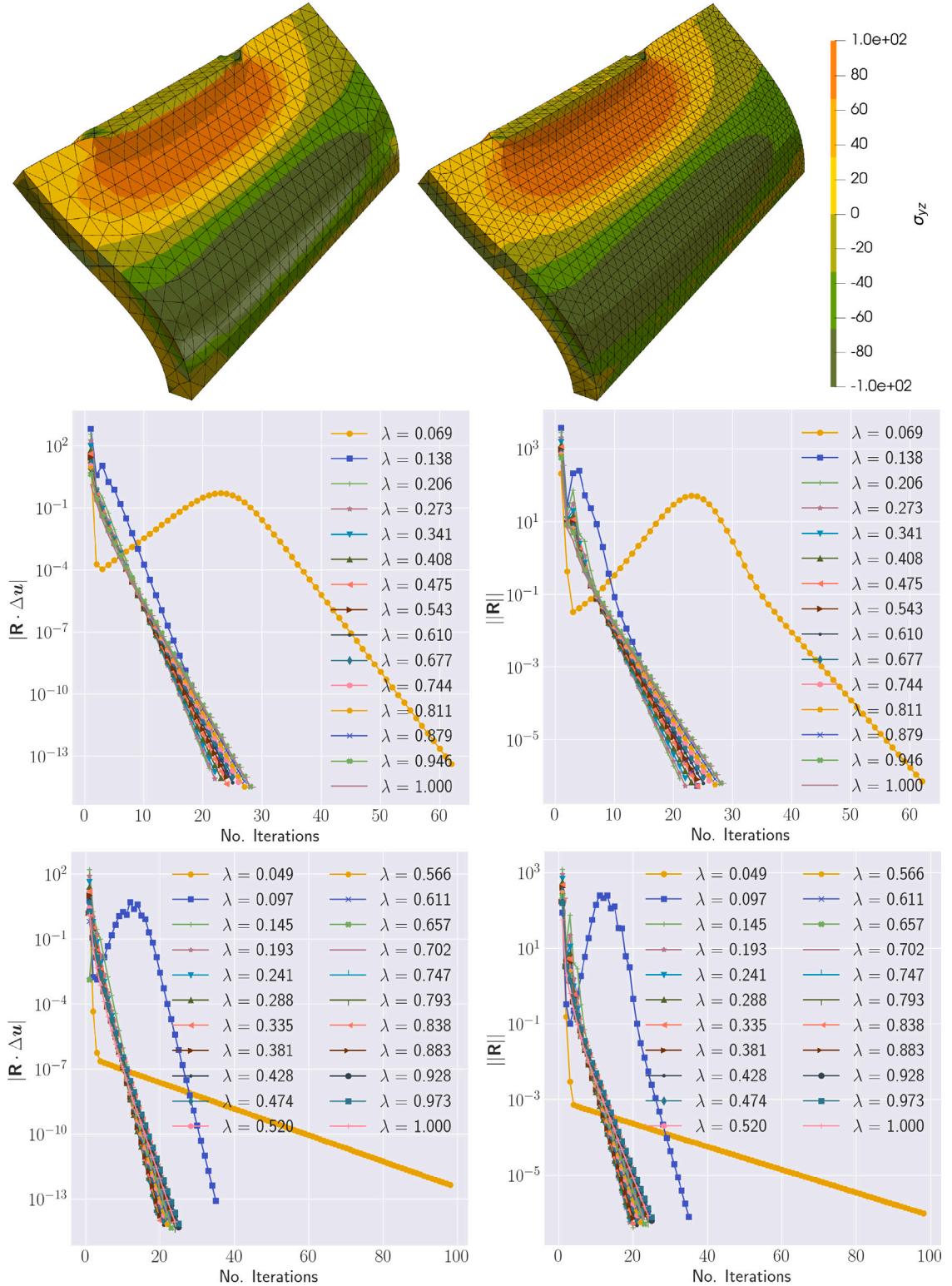


Fig. 9. Automatic discretisation-aware load stepping with P-NR: A block (with two $p = 2$ tetrahedral meshes) is pressed with imposed displacements applied in full that creates element inversion. Discretisation-aware load stepping, described in Algorithms 1 and 2, ensures that the loading is automatically adjusted to maximum inversion-safe displacements per increment. Further, load incrementation improves the convergence of P-NR on a per-increment basis. Two bottom rows show per-increment convergence (with accumulative load factor λ in legends) of curvature $|R \cdot \Delta u|$, and residual $\|R\|$ for coarse and fine meshes, respectively. Curvature drops rapidly within 15 iterations.

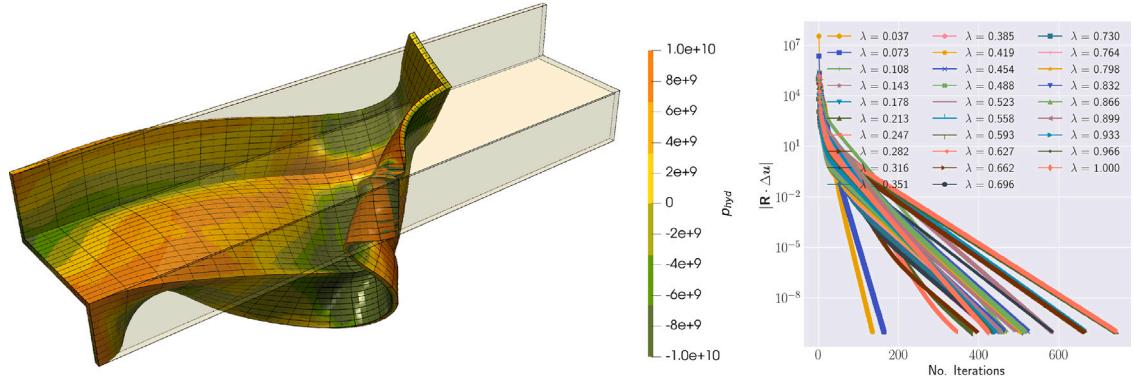


Fig. 10. Guaranteed inversion-free deformation of a Z-section cantilever through imposed displacements on the lower flange by half the section length using tangent stabilised C-based formulation. The level of displacements causes structural collapse and hence element inversion during simulation when imposed in one increment. Discretisation-aware load stepping, described in Algorithms 1 and 2, ensures that the loading is automatically adjusted to maximum inversion-safe displacements per increment. Despite the level of deformation and high element aspect ratio, the final result is inversion-free Q2 elements i.e. $J > 0$ when sampled at 27, 64 and 125 quadrature point. The right figure shows the maximum load factors determined by the algorithm and their corresponding convergence.

7.4. Comparison of arc length with P-NR under extreme loading scenarios and structural collapse

The goal of this example is to:

1. Perform a thorough comparison of P-NR with the arc length method in 3-dimensional settings and under extreme loading scenarios and highlight the differences in performance and characteristics of the two approaches
2. Discuss performance of different tangent stabilisation techniques including geometrically stabilised polyconvexity and their merit/usefulness in solving such extreme problems

To this end, we choose a cylindrical shell under compression as shown in Fig. 11. The base and four symmetrical points at the top of cylinder are fixed. A nominal load of $q = -0.224$ is applied to the rest of the nodes on the top face and a load controlled arc length is used to monitor the percentage of applied load that the structure can take. The final load factor achieved $\lambda \approx 0.56$ is then given to P-NR with a single increment. The same nearly incompressible Mooney–Rivlin material with reference parameters (shear modulus $\mu = 1$ and Poisson's ratio $\nu = 0.45$) is used to keep the rest of the setting simplified for comparisons. Further, for the purpose of demonstrating differences between the arc-length-NR and P-NR it suffices to choose a Q1 hexahedral mesh as the same concepts hold for higher order finite elements too. Fig. 11 shows the final deformed configuration obtained with NR with constant load factor, NR with arc-length and NR with tangent stabilised elasticity (P-NR) when the cylinder is pressed to the failure point. This example illustrates a few key points. For this example, NR with constant load factor fails regardless of the number of increments. Arc-length NR succeeds but requires a few attempts to get its parameters (i.e. the starting arc-length and number of increments) tuned correctly but even with best effort it is still $14.5\times$ slower than P-NR. The load-deflection curves at point A in Fig. 11 obtained with arc-length NR is shown in Fig. 12 for both y and z directions. On the same plots, the displacement vs iteration count of P-NR is shown where the final displacement of point A agrees well with the arc-length method. The arc-length-NR employed here uses a cylindrical constraint with secant path correction and the key to its success is often in getting the parameters correct, as has been well-known in the literature [22–24]. This certainly illustrates the lack of automation inherent in arc length based approaches. On the other hand, for this example, P-NR succeeds with a single load increment and 53 iterations while giving similar result to arc-length NR. Furthermore, it requires no tuning and extra parameter input. For many applications mentioned earlier such automation remains key.

Finally, we would like to assess the performance of different tangent stabilisation techniques presented for this complex problem. Fig. 13 shows convergence of P-NR with different tangent stabilisation technique for both residual and curvature. First of all, it is clear that all techniques converge to the same solution without getting stuck. Unlike Fig. 6 however, we see that the C- and b-based formulations is now closer in performance to F-based. As pointed out earlier, the convergence characteristics of these formulations are problem dependent although the F-based formulation still converges quicker. The more interesting insight however comes from the geometrically stabilised polyconvex elasticity formulation whose performance is considerably worse than the former 3 formulations. The explanation for this follows the same logic discussed earlier that is, while the geometric stiffness (\tilde{C}_p) that emerges in polyconvex elasticity might not be (and often cannot be as clear from (123), for instance) PSD, the total tangent stiffness (\tilde{C}) could very well be PSD owing to the contribution of always PSD constitutive tangent (\tilde{C}_k). Hence, PSD projection of the geometric stiffness (\tilde{C}_p) alone truncates second-order information (i.e. Hessian terms) unnecessarily which results in damped and rather inferior convergence. Finally, it makes sense to include in the results the case of polyconvex elasticity by completely discarding the geometric stiffness. This is a rather popular Gauss–Newton approximation in numerical optimisation. We see from the figures that discarding the geometric stiffness results in significant oscillation in convergence however, the total number of iterations remains the same as that of geometrically stabilised polyconvex elasticity, for this problem. In sum, F-based formulation remains the least intrusive and most competitive way of tangent stabilisation.

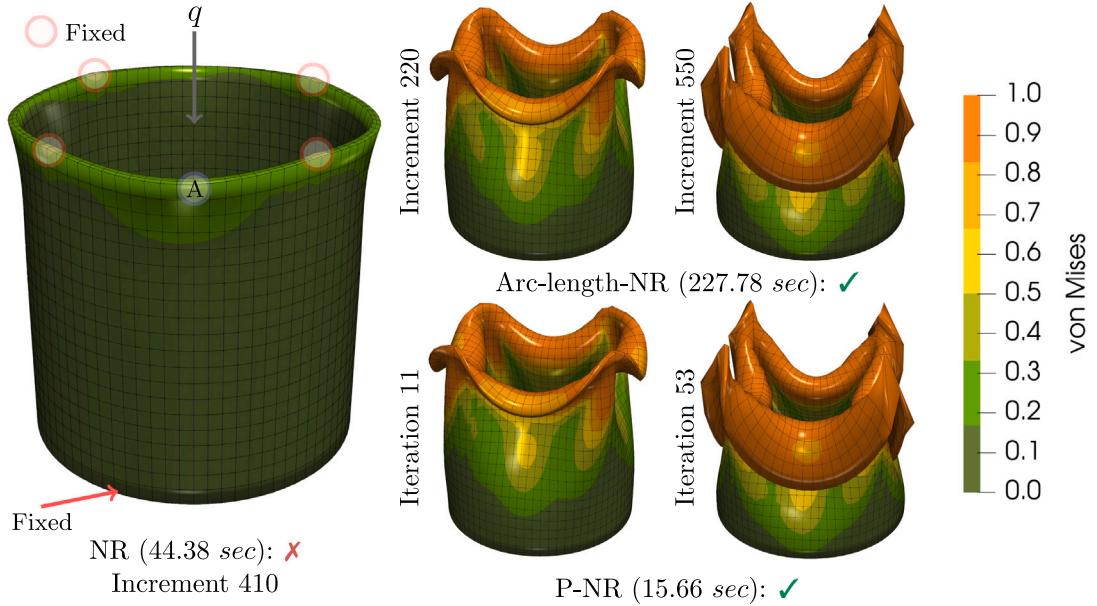


Fig. 11. Press simulation of a cylindrical column using different methods: Newton–Raphson [NR] (1000 increments), Arc length based Newton–Raphson [Arc length NR] (550 increments: 1655 iterations in total), and Tangent stabilised/Projected Newton–Raphson [P-NR] (1 increment: 53 iterations in total). Standard NR failed regardless of increments, while Arc length NR was 14.5 times slower than P-NR. Arc length NR required parameter adjustments to work correctly. P-NR used the load factor achieved by Arc length NR. Similar final results were obtained for both methods.

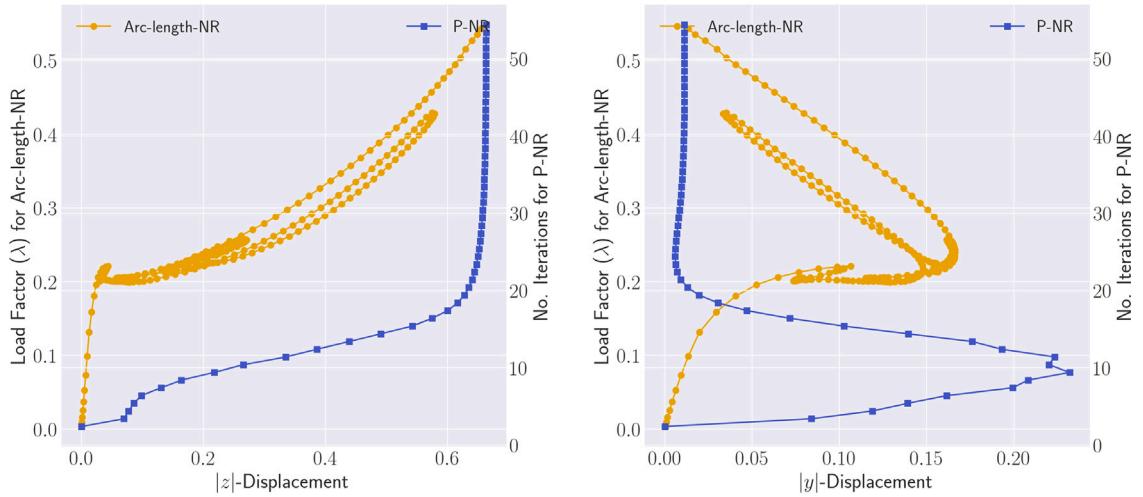


Fig. 12. Press simulation of a cylindrical column from Fig. 11: Load–deflection curve for point A in Fig. 11 for arc length vs convergence of F -based P-NR with iterations. The final y and z displacements for point A obtained with P-NR match well with arc-length NR.

8. Concluding remarks

In this work, we have generalised the notion of tangent stabilised elasticity to virtually all known invariant formulations of nonlinear elasticity. We have shown that, closed-form eigen-decomposition of tangents is easily available irrespective of invariant formulation. In particular, we have worked out closed-form tangent eigensystems for Total Lagrangian deformation gradient (F)-based and right Cauchy–Green (C)-based as well as Updated Lagrangian left Cauchy–Green (b)-based formulations and present their exact convexity conditions postulated in terms of their corresponding tangent and geometric stiffness eigenvalues. In addition, we have introduced the notion of geometrically stabilised polyconvex large strain elasticity for models that are materially stable but exhibit geometric instabilities for whom we have constructed their geometric stiffness in a spectrally-decomposed form analytically. Numerical experiments however show that compared to F -based formulation, geometrically stabilised polyconvex elasticity might not be competitive as it leads to over-truncation of tangent elasticity. We have further extended the framework to the case of

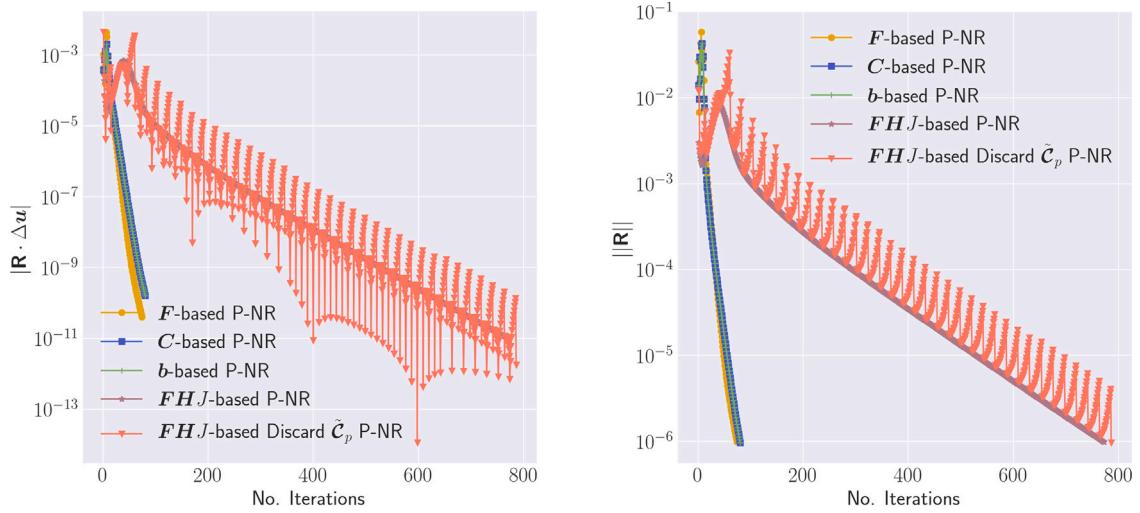


Fig. 13. Press simulation of a cylindrical column from Fig. 11 : Convergence of different tangent stabilisation techniques with P-NR. The performance of C- and b-based formulations is close to F-based although F-based formulation is still marginally faster and requires fewer iterations. PSD projection of the geometric stiffness ($\tilde{\mathcal{C}}_p$) alone in polyconvex elasticity leads to unnecessarily truncation of second-order information (i.e. Hessian terms) which results in inferior convergence pattern. Discarding this geometric stiffness altogether results in oscillation in residual/gradient.

transverse isotropy where once again, closed-form tangent eigensystems are found for common transversely isotropic invariants. We have further presented a mixed variational formulation for models with arbitrarily directed inextensible fibres. Finally, we have proposed a discretisation-aware load-stepping together with a line search scheme for a robust implementation of tangent stabilised elasticity over general polyhedral meshes. Multiple examples have been presented that show the robustness and performance characteristics of tangent stabilised elasticity specifically when compared to the path-following approaches such as arch-length. In this regard Newton–Raphson with Projected (stabilised) tangents i.e. P-NR is shown to be competitive to arc-length requiring no manual user intervention. The automatic nature of specific P-NR flavours showcased in this work is due in part to the load stepping scheme that is sensitive to discretisation. Although we have introduced curvature as a different way to measure convergence, we do not claim that it is the best measure. Additionally, although P-NR seems highly robust, it may still encounter convergence issues. This can occur when the line search becomes too small, causing the simulations to become unresponsive. There may be other criteria for tracking convergence in P-NR that are superior. We note that, an analysis of performance of P-NR with implicit and explicit transient dynamic analysis algorithms would be indeed an interesting future piece of work.

CRediT authorship contribution statement

Roman Poya: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Rogelio Ortigosa:** Writing – review & editing, Validation, Methodology, Funding acquisition. **Antonio J. Gil:** Writing – review & editing, Validation, Funding acquisition. **Theodore Kim:** Writing – review & editing, Validation, Methodology, Conceptualization. **Javier Bonet:** Writing – review & editing, Validation, Methodology.

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.

Acknowledgements

The first author thanks Brent Meranda manager of the Meshing Framework, Discretisation, Simulation and Test Solutions, Siemens Digital Industries Software. The second author acknowledges the support of grant PID2022-141957OA-C22 funded by MICIU/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”. The second author also acknowledges the support provided by the Autonomous Community of the Region of Murcia, Spain through the programme for the development of scientific and technical research by competitive groups (21996/PI/22), included in the Regional Program for the Promotion of Scientific and Technical Research of Fundacion Seneca - Agencia de Ciencia y Tecnología de la Región de Murcia. The third author acknowledges the financial support received through the UK Defence, Science and Technology Laboratory, grants: DSTLX1000160466R and RQ0000028640.

Appendix A. Variational formulations

A.1. Standard displacement-based variational principle

The solution of large strain elastic problems is often expressed by means of the total energy minimisation variational principle as

$$\Pi(\mathbf{x}^*) = \min_{\mathbf{x}} \int_V W(\mathcal{A}_{\mathbf{x}}) dV - \int_V \mathbf{f}_0 \cdot \mathbf{x} dV - \int_{\partial_V} \mathbf{t}_0 \cdot \mathbf{x} dA, \quad (\text{A.1})$$

where \mathbf{x}^* denotes the exact solution. The stationary condition of this functional leads to the principle of virtual work, commonly written as

$$D\Pi[\delta\mathbf{u}] = \int_V \mathbf{P}^x : \delta\nabla_0 \mathbf{u} dV - \int_V \mathbf{f}_0 \cdot \delta\mathbf{u} dV - \int_{\partial_V} \mathbf{t}_0 \cdot \delta\mathbf{u} dA = 0. \quad (\text{A.2})$$

In this expression, the first Piola–Kirchhoff tensor \mathbf{P}^x is evaluated in the standard fashion using (16) in terms of the gradient of the deformation $\nabla_0 \mathbf{x}$ and in our setting its singular-values. For better clarification of the notation, it is useful to introduce the definition of the geometrically compatible strain and stretch measures as

$$\mathbf{F}_x = \hat{\mathbf{U}}_x \mathbf{A}_x \hat{\mathbf{V}}_x^T = \nabla_0 \mathbf{x}, \quad \lambda_{x_i} = \Lambda_{x_{ii}}, \quad \mathbf{P}^x = \sum_{i=1}^d \Sigma_{\lambda_i}^x \frac{\partial \lambda_{x_i}}{\partial F_x} = \hat{\mathbf{U}}_x \mathbf{A}_{Px} \hat{\mathbf{V}}_x^T, \quad \Sigma_{\lambda_i}^x = \Sigma_{\lambda_i}(\lambda_{x_i}), \quad (\text{A.3})$$

where the superscript (subscript) x is used for stresses (strains) to indicate that they are evaluated in terms of geometry. An iterative Newton–Raphson process to converge towards the solution is usually established by solving a linearised system for the increment $\Delta\mathbf{u}$ as

$$D^2\Pi[\delta\mathbf{u}; \Delta\mathbf{u}] = -D\Pi(\mathbf{x})[\delta\mathbf{u}], \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \Delta\mathbf{u}, \quad (\text{A.4})$$

where α is the line search parameter and the second derivative of the total energy functional is given by

$$D^2\Pi[\delta\mathbf{u}; \Delta\mathbf{u}] = \int_V D^2e[\delta\mathbf{u}, \Delta\mathbf{u}] dV. \quad (\text{A.5})$$

The tangent operator is evaluated taking $\mathbf{F}_x = \nabla_0 \mathbf{x}$ and λ_{x_i} as its singular-values.

A.2. Mixed variational principle for isotropic elasticity

A series of novel mixed variational principles for principal stretch based formulations were presented in Poya et al. [13] for general analysis of isotropic compressible and nearly incompressible elastica which for completeness we briefly summarise. For isotropic elasticity, a variational principle can be written in terms of the geometry and principal stretches as a constrained minimisation problem using a standard Lagrange multiplier approach

$$\Pi_M(\mathbf{x}^*, \mathcal{A}^*, S^*) = \min_{\mathbf{x}, \mathcal{A}} \left\{ \max_S \left\{ \int_V W(\mathcal{A}) dV + \sum_{i=1}^d \int_V \Sigma_{\lambda_i}(\lambda_{x_i} - \lambda_i) dV - \int_V \mathbf{f}_0 \cdot \mathbf{x} dV - \int_{\partial_V} \mathbf{t}_0 \cdot \mathbf{x} dA \right\} \right\}. \quad (\text{A.6})$$

This 7-field formulation belongs to the general class of Hu–Washizu type of mixed variational principles. Note that, the conjugate variables Σ_{λ_i} in this expression, are simply Lagrange multipliers.

A.3. Mixed variational principle for isotropic rigid and stiff models

A variational principle can be easily established as particular case of the above one for the case of nearly rigid i.e. stiff materials or the practically relevant case, the truly rigid formulation which do not have a deformable part

$$\Pi_M^R(\mathbf{x}^*, S^*) = \min_{\mathbf{x}} \left\{ \max_S \left\{ \sum_{i=1}^d \int_V \Sigma_{\lambda_i}(\lambda_{x_i} - 1) - \frac{1}{2C_c} \Sigma_{\lambda_i}^2 dV - \int_V \mathbf{f}_0 \cdot \mathbf{x} dV - \int_{\partial_V} \mathbf{t}_0 \cdot \mathbf{x} dA \right\} \right\}, \quad (\text{A.7})$$

where $C_c > 0$ is an additional penalty term to avoid saddle point problem and the formulation above falls under the general category of Perturbed Lagrangian formulation.

A.4. Mixed variational principle for transversely isotropic elasticity with inextensible fibres

Poya et al. [13]’s approach is essentially an F -based formulation in terms of principal stretches. As seen in the above $\tilde{\mathcal{C}}_k^{ij}$ it is already positive definite. Hence, it requires no special treatment. In general, following this approach all isotropic invariants are resolved using the SVD but no SVD is performed for transversely isotropic cases. Following [67,68], we introduce the following variational principle

$$\Pi_M^I(\mathbf{x}^*, \Sigma_{c_i}^*) = \min_{\mathbf{x}} \left\{ \max_{\Sigma_{c_i}} \left\{ \int_V W(\mathcal{A}_x) dV + \sum_{i=1}^m \int_V \Sigma_{c_i}(I_4^i - 1) - \frac{1}{2C_i} \Sigma_{c_i}^2 dV - \int_V \mathbf{f}_0 \cdot \mathbf{x} dV - \int_{\partial_V} \mathbf{t}_0 \cdot \mathbf{x} dA \right\} \right\}, \quad (\text{A.8})$$

where m is the number of specified fibre directions and $I_4^i = \mathbf{F}[\mathbf{M}]^i \cdot \mathbf{F}[\mathbf{M}]^i$ denotes the inextensibility constraint condition in the i th direction with $[\mathbf{M}]^i$ the corresponding i th fibre direction. The stationary conditions of this hybrid functional is evaluated in the same fashion as previous formulation. For instance, the first derivative with respect to geometry gives the principle of virtual work as

$$D_1 \Pi_M^I [\delta \mathbf{u}] = \int_V \mathbf{P}^I : \nabla_0 \delta \mathbf{u} \, dV - \int_V \mathbf{f}_0 \cdot \delta \mathbf{u} \, dV - \int_{\partial_V} \mathbf{t}_0 \cdot \delta \mathbf{u} \, dA = 0, \quad (\text{A.9})$$

where the first Piola–Kirchhoff stress tensor is now evaluated as

$$\mathbf{P}^I = \sum_{i=1}^d \Sigma_{\lambda_{x_i}} \frac{\partial \lambda_{x_i}}{\partial \mathbf{F}_x} + 2 \sum_{i=1}^m \Sigma_{c_i} \mathbf{F} ([\mathbf{M}]^i \otimes [\mathbf{M}]^i). \quad (\text{A.10})$$

The first derivative with respect to Σ_{c_i} enforces the inextensibility constraint

$$D_2 \Pi_M^I [\delta \Sigma_{c_i}] = \sum_{i=1}^m \int_V \delta \Sigma_{c_i} \left((I_4^i - 1) - \frac{1}{C_i} \Sigma_{c_i} \right) \, dV. \quad (\text{A.11})$$

The evaluation of second derivatives required for a Newton–Raphson process proceeds along the same lines. For instance, the second derivative with respect to geometry leads to the geometric stiffness operator

$$D_{1;1}^2 \Pi_M^I [\delta \mathbf{u}; \Delta \mathbf{u}] = \int_V \nabla_0 \delta \mathbf{u} : \mathbf{C}^I : \nabla_0 \Delta \mathbf{u} \, dV, \quad (\text{A.12})$$

where the total tangent stiffness is now evaluated as

$$\mathbf{C}^I = \sum_{i=1}^d \sum_{j=1}^d \frac{\partial^2 W}{\partial \lambda_{x_i} \partial \lambda_{x_j}} \frac{\partial \lambda_{x_i}}{\partial \mathbf{F}} \otimes \frac{\partial \lambda_{x_j}}{\partial \mathbf{F}} + \sum_{i=1}^d \Sigma_{\lambda_{x_i}} \frac{\partial^2 \lambda_{x_i}}{\partial \mathbf{F}_x \partial \mathbf{F}_x} + 2 \sum_{i=1}^m \Sigma_{c_i} \left(\sum_{j=1}^d [\mathbf{E}]_j^i \otimes [\mathbf{E}]_j^i \right), \quad (\text{A.13})$$

where once again, $[\mathbf{E}]_j^i$ represents the j th eigenmatrix for the i th fibre. Alternatively, the PSD (stabilised) version of the tangent can be constructed as

$$\begin{aligned} \mathbf{C}^{I,\text{PSD}} &= \left[\frac{\partial \lambda_{x_1}}{\partial \mathbf{F}}, \dots, \frac{\partial \lambda_{x_d}}{\partial \mathbf{F}} \right] [\mathbf{H}_W^{\text{PSD}}] \begin{bmatrix} \frac{\partial \lambda_{x_1}}{\partial \mathbf{F}} \\ \vdots \\ \frac{\partial \lambda_{x_d}}{\partial \mathbf{F}} \end{bmatrix} \\ &+ \sum_{i=1}^k \max(\bar{\lambda}_{x_i}^{C_p}, 0) [\mathbf{L}]_i \otimes [\mathbf{L}]_i + \max(\bar{\lambda}_{x_{i+k}}^{C_p}, 0) [\mathbf{T}]_i \otimes [\mathbf{T}]_i \\ &+ 2 \sum_{i=1}^m \max(\Sigma_{c_i}, 0) \left(\sum_{j=1}^d [\mathbf{E}]_j^i \otimes [\mathbf{E}]_j^i \right), \end{aligned} \quad (\text{A.14})$$

where $\bar{\lambda}_{x_i}^{C_p}$ indicates the evaluation tangent operator with respect to geometry. Crucially, PSD projection (stabilisation) requires pruning the contribution of transversely isotropic invariant to tangent operator when the inextensibility constraint Σ_{c_i} becomes negative. The second derivative with respect to Σ_{c_i} has a standard Galerkin mass matrix format

$$D_{2;2}^2 \Pi_M^I [\delta \Sigma_{c_i}; \Delta \Sigma_{c_i}] = - \sum_{i=1}^m \frac{1}{C_i} \int_V \delta \Sigma_{c_i} \cdot \Delta \Sigma_{c_i} \, dV, \quad (\text{A.15})$$

and finally, the cross terms emerging from derivatives of geometry and inextensibility constraint are obtained as

$$D_{1;2}^2 \Pi_M^I [\delta \mathbf{u}; \Delta \Sigma_{c_i}] = 2 \sum_{i=1}^m \int_V (\mathbf{F}[\mathbf{M}]^i : \nabla_0 \delta \mathbf{u}) \cdot \Delta \Sigma_{c_i} \, dV, \quad (\text{A.16a})$$

$$D_{2;1}^2 \Pi_M^I [\delta \Sigma_{c_i}; \Delta \mathbf{u}] = 2 \sum_{i=1}^m \int_V \delta \Sigma_{c_i} \cdot (\mathbf{F}[\mathbf{M}]^i : \nabla_0 \Delta \mathbf{u}) \, dV. \quad (\text{A.16b})$$

Appendix B. Finite element discretisation

The implementation of the various variational principles described in the previous section is based on a finite element partition of the domain into a set of elements. Inside each element the problem variables are interpolated in terms of a set of shape functions N_a as

$$\mathbf{x} = \sum_{a=1}^{n_x} \mathbf{x}_a N_a^x, \quad \Sigma_{c_i} = \sum_{a=1}^{n_{\Sigma_{c_i}}} \Sigma_{c_{i_a}} N_a^{\Sigma_{c_i}}, \quad (\text{B.1})$$

where a denotes the nodes or other degrees of freedom used in the interpolation of the above variables. In general, different interpolations are often used to describe different variables. However, the same interpolation space will invariably be used for

strain–stress conjugate pairs; that is, $N_a^{\lambda_i} = N_a^{\Sigma_{\lambda_i}}$, etc. The virtual and incremental variables are also interpolated using the same spaces as

$$\delta\mathbf{u} = \sum_{a=1}^{n_x} \delta u_a N_a^x, \quad \delta\Sigma_{\lambda_i} = \sum_{a=1}^{n_{\Sigma_{\lambda_i}}} \delta\Sigma_{\lambda_{i_a}} N_a^{\Sigma_{\lambda_i}}, \quad \Delta\mathbf{u} = \sum_{a=1}^{n_x} \Delta u_a N_a^x, \quad \Delta\Sigma_{c_i} = \sum_{a=1}^{n_{\Sigma_{c_i}}} \Delta\Sigma_{c_{i_a}} N_a^{\Sigma_{c_i}}. \quad (\text{B.2})$$

Finite element equations are derived by simply substituting the above expressions into the functional expressions provided in the previous section. For instance, substituting the above interpolation for the virtual displacements into any of the virtual work statements given in the previous section leads to residual forces as

$$D_1\Pi[\delta\mathbf{u}] = \sum_a R_a^x \cdot \delta\mathbf{u}, \quad R_a^x = \int_{V^e} \mathbf{P}\nabla_0 N_a^x \, dV^e - \int_{V^e} \mathbf{f}_0 N_a^x \, dV^e - \int_{\partial_e V^e} t_0 N_a^x \, dA^e, \quad (\text{B.3})$$

where the first Piola–Kirchhoff stress tensor above will be evaluated in accordance with each of the formulations presented in the previous section and V^e and A^e denote the volume and area of element e in the original configuration. We only present the finite element discretisation for the transversely isotropic elasticity with inextensible fibres as the discretisation for other formulations have already been presented in Poya et al. [13]. For instance, the geometric compatibility residuals $\mathbf{R}_a^{\Sigma_{c_i}}$ emerge from the discretisation of (A.11).

$$\mathbf{R}_a^{\Sigma_{c_i}} = \sum_{i=1}^m \int_{V^e} \left(I_4^i - 1 - \frac{\Sigma_{c_i}}{C_i} \right) N_a^{\Sigma_{c_i}} \, dV^e. \quad (\text{B.4})$$

In order to complete the finite element formulation it is necessary to derive equations for the components of the tangent matrix by discretising the tangent operators. For the case of mixed inextensible formulation, the resulting tangent operator expands to

$$D^2\Pi_M[\delta\mathbf{u}, \delta\Sigma_{c_i}; \Delta\mathbf{u}, \Delta\Sigma_{c_i}] = [\delta\mathbf{u} \quad \delta\Sigma_{c_i}] \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{x\Sigma_{c_i}} \\ \mathbf{K}_{\Sigma_{c_i}x} & \mathbf{K}_{\Sigma_{c_i}\Sigma_{c_i}} \end{bmatrix} [\Delta\mathbf{u} \quad \Delta\Sigma_{c_i}], \quad (\text{B.5})$$

where $\mathbf{K}_{x\Sigma_{c_i}}$ and $\mathbf{K}_{\Sigma_{c_i}\Sigma_{c_i}}$ emerge from the discretisation of (A.15) and (A.16), respectively

$$\mathbf{K}_{x\Sigma_{c_i}}^{ab} = 2 \sum_{i=1}^m \int_{V^e} (\mathbf{F}[\mathbf{M}]^i : \nabla_0 N_a^x) N_b^{\Sigma_{c_i}} \, dV^e, \quad (\text{B.6a})$$

$$\mathbf{K}_{\Sigma_{c_i}\Sigma_{c_i}}^{ab} = - \sum_{i=1}^m \frac{1}{C_i} \int_{V^e} N_a^{\Sigma_{c_i}} \cdot N_b^{\Sigma_{c_i}} \, dV^e. \quad (\text{B.6b})$$

These equations can be implemented using a variety of finite element spaces but, not all choices will lead to effective or valid finite element formulations [35,36,41,69–73]. Some choices of ansatz spaces are given in [67,68] for continuous interpolation of all fields and in [13] for discontinuous interpolation of secondary (dual) variables.

Appendix C. Analytic eigensystem of neo-Hookean type energies in Voigt notation

A practically relevant case of Voigt notation arises for neo-Hookean-style energies of type

$$\check{W} = \frac{\mu}{2}(\text{tr}(\mathbf{C}) - N) + f(J, \kappa), \quad (\text{C.1})$$

where μ and κ are Lamé parameters. For such energy functions the constitutive tangent takes the following form

$$\check{\mathbf{C}}_k = \alpha C_{IJ}^{-1} C_{KL}^{-1} + \beta(C_{IK}^{-1} C_{JL}^{-1} + C_{IL}^{-1} C_{JK}^{-1}) \quad (\text{C.2})$$

where α and β are in general variables depending on material parameters and $\mathbf{C} = J^2$. The explicit constitutive tangent of these energies in 3-dimensions can be written as

$$\check{\mathbf{C}}_k^{Voigt} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & 0 & 0 & 0 \\ & C_{2222} & C_{2233} & 0 & 0 & 0 \\ & & C_{3333} & 0 & 0 & 0 \\ & & & C_{1212} & 0 & 0 \\ & & & & C_{1313} & 0 \\ & & & & & C_{2323} \end{bmatrix}_{sym}, \quad (\text{C.3})$$

whose analytic eigensystem separates again into *scaling* and *flip* modes similar to (42) but now in simpler vector form due to Voigt notation

$$\check{\mathbf{C}}_k^{Voigt} = \sum_{i=1}^3 \sum_{j=1}^3 \bar{\lambda}_i^{Voigt} [\bar{\mathbf{e}}]_i^j [\mathbf{d}]_j \otimes [\bar{\mathbf{e}}]_i^j [\mathbf{d}]_j + \sum_{i=1}^3 \bar{\lambda}_{i+3}^{Voigt} [\mathbf{I}]_i \otimes [\mathbf{I}]_i \quad (\text{C.4})$$

where the *scaling* and *flip* eigenvectors are simply unit vectors

$$[\mathbf{d}]_1 = [1, 0, 0, 0, 0, 0]^T, \quad [\mathbf{d}]_2 = [0, 1, 0, 0, 0, 0]^T, \quad [\mathbf{d}]_3 = [0, 0, 1, 0, 0, 0]^T, \quad (\text{C.5})$$

$$[\mathbf{I}]_1 = [0, 0, 0, 1, 0, 0]^T, \quad [\mathbf{I}]_2 = [0, 0, 0, 0, 1, 0]^T, \quad [\mathbf{I}]_3 = [0, 0, 0, 0, 0, 1]^T \quad (\text{C.6})$$

```

from sympy import *

def GetFirstPiolaAndTangentsEigenSystemSymbolically(W):
    """Given energy W expressed in terms of squares of singular values: W(l1^2, l2^2, l3^2) returns principal components of second Piola-Kirchhoff and analytic eigenvalues of constitutive and initial stiffness operators"""

# -----#
# Second Piola-Kirchhoff stress tensor [S = V * sigmaS * V^T]
sigmaS = zeros(2,1)
sigmaS[0] = diff(W, 11) / 11
sigmaS[1] = diff(W, 12) / 12
# -----#

# Flip mode eigenvalues of constitutive operator
lambdas = zeros(1,1)
lambdas[0] = (sigmaS[0] - sigmaS[1]) / (11**2 - 12**2)
# -----#

# Hessian matrix
Hw = zeros(2,2)

Hw[0,0] = 1 / 11**2 * diff(W, 11, 2) - 1 / 11**3 * diff(W, 11)
Hw[1,1] = 1 / 12**2 * diff(W, 12, 2) - 1 / 12**3 * diff(W, 12)

Hw[0,1] = diff(diff(W, 11), 12) / 11 / 12
Hw[1,0] = Hw[0,1]

# Get its eigenvalues
lambdasHw = Hw.eigenvals().items()
# -----#

return sigmaS, lambdas, lambdasHw

```



```

from sympy import *

def GetFirstPiolaAndTangentsEigenSystemSymbolically(W):
    """Given energy W expressed in terms of squares of singular values: W(l1^2, l2^2, l3^2) returns principal components of second Piola-Kirchhoff and analytic eigenvalues of constitutive and initial stiffness operators"""

# -----#
# Second Piola-Kirchhoff stress tensor [S = V * sigmaS * V^T]
sigmaS = zeros(2,1)
sigmaS[0] = diff(W, 11) / 11
sigmaS[1] = diff(W, 12) / 12
sigmaS[2] = diff(W, 13) / 13
# -----#

# Flip mode eigenvalues of constitutive operator
lambdas = zeros(3,1)
lambdas[0] = (sigmaS[1] - sigmaS[2]) / (12**2 - 13**2)
lambdas[1] = (sigmaS[0] - sigmaS[2]) / (11**2 - 13**2)
lambdas[2] = (sigmaS[0] - sigmaS[1]) / (11**2 - 12**2)
# -----#

# Hessian matrix
Hw = zeros(3,3)

Hw[0,0] = 1 / 11**2 * diff(W, 11, 2) - 1 / 11**3 * diff(W, 11)
Hw[1,1] = 1 / 12**2 * diff(W, 12, 2) - 1 / 12**3 * diff(W, 12)
Hw[2,2] = 1 / 13**2 * diff(W, 13, 2) - 1 / 13**3 * diff(W, 13)

Hw[0,1] = diff(diff(W, 11), 12) / 11 / 12
Hw[1,2] = diff(diff(W, 11), 13) / 11 / 13
Hw[1,2] = diff(diff(W, 12), 13) / 12 / 13

Hw[1,0] = Hw[0,1]
Hw[2,0] = Hw[0,2]
Hw[2,1] = Hw[1,2]

# Get its eigenvalues
lambdasHw = Hw.eigenvals().items()
# -----#

return sigmaS, lambdas, lambdasHw

```

Fig. D.14. Symbolic code for obtaining principal components of Second Piola–Kirchhoff (which are also eigenvalues of geometric stiffness) and eigensystem of constitutive tangent operator in 2 and 3 dimensions.

and the last 3 eigenvalues are already decoupled and appear as

$$\check{\lambda}_4^{C_k^{Voigt}} = C_{1212}, \quad \check{\lambda}_5^{C_k^{Voigt}} = C_{1313}, \quad \check{\lambda}_6^{C_k^{Voigt}} = C_{2323}. \quad (\text{C.7})$$

For neo-Hookean style energies often $C_{1111} = C_{2222} = C_{3333}$, $C_{1122} = C_{1133} = C_{2233}$ and $C_{1212} = C_{1313} = C_{2323}$ and, the first 3 eigenvalues (and eigenvectors $[\bar{e}]_i$) are obtained from the eigen-decomposition of the upper 3×3 block of \check{C}_k^{Voigt} which actually corresponds to H_W from (43)

$$\check{\lambda}_1^{C_k^{Voigt}} = C_{1111} + 2C_{1122}, \quad \check{\lambda}_2^{C_k^{Voigt}} = C_{1111} - C_{1122}, \quad \check{\lambda}_3^{C_k^{Voigt}} = C_{1111} - C_{1122}. \quad (\text{C.8})$$

In 3-dimensional Voigt format the eigenvectors of H_W can always be chosen as $[\bar{e}]_1 = \frac{1}{\sqrt{3}}[1, 1, 1]^T$, $[\bar{e}]_2 = \frac{1}{\sqrt{6}}[-2, 1, 1]^T$ and $[\bar{e}]_3 = \frac{1}{\sqrt{2}}[0, 1, -1]^T$ to give

$$\boxed{\check{C}_k^{Voigt} = \frac{C_{1111} + 2C_{1122}}{3}[\mathbf{q}]_1 \otimes [\mathbf{q}]_1 + \frac{C_{1111} - C_{1122}}{6}[\mathbf{q}]_2 \otimes [\mathbf{q}]_2 + \frac{C_{1111} - C_{1122}}{2}[\mathbf{q}]_3 \otimes [\mathbf{q}]_3 + C_{1212}[I]_1 \otimes [I]_1 + C_{1212}[I]_2 \otimes [I]_2 + C_{1212}[I]_3 \otimes [I]_3,} \quad (\text{C.9})$$

with $[\mathbf{q}]_1 = [1, 1, 1, 0, 0, 0]^T$, $[\mathbf{q}]_2 = [-2, 1, 1, 0, 0, 0]^T$ and $[\mathbf{q}]_3 = [0, 1, -1, 0, 0, 0]^T$. Similar but simpler expressions are found in 2-dimensions

$$\check{C}_k^{Voigt} = \begin{bmatrix} C_{1111} & C_{1122} & 0 \\ C_{2222} & 0 & \\ sym & & C_{1212} \end{bmatrix} = \sum_{i=1}^2 \sum_{j=1}^2 \check{\lambda}_i^{C_k^{Voigt}} [\bar{e}]_i^j [\mathbf{d}]_j \otimes [\bar{e}]_i^j [\mathbf{d}]_j + \check{\lambda}_3^{C_k^{Voigt}} [I]_1 \otimes [I]_1 \quad (\text{C.10})$$

where the scaling and flip eigenvectors are simply unit vectors

$$[\mathbf{d}]_1 = [1, 0, 0]^T, \quad [\mathbf{d}]_2 = [0, 1, 0]^T, \quad [I]_1 = [0, 0, 1]^T \quad (\text{C.11})$$

and, the eigenvalues for neo-Hookean style energies are given by

$$\check{\lambda}_1^{C_k^{Voigt}} = C_{1111} + C_{1122}, \quad \check{\lambda}_2^{C_k^{Voigt}} = C_{1111} - C_{1122}, \quad \check{\lambda}_3^{C_k^{Voigt}} = C_{1212}. \quad (\text{C.12})$$

Similarly, in 2-dimensional Voigt format the eigenvectors of H_W can always be chosen as $[\bar{e}]_1 = \frac{1}{\sqrt{2}}[1, 1]^T$ and $[\bar{e}]_2 = \frac{1}{\sqrt{2}}[-1, 1]^T$ to give

$$\boxed{\check{C}_k^{Voigt} = \frac{C_{1111} + C_{1122}}{2}[\mathbf{q}]_1 \otimes [\mathbf{q}]_1 + \frac{C_{1111} - C_{1122}}{2}[\mathbf{q}]_2 \otimes [\mathbf{q}]_2 + C_{1212}[I]_1 \otimes [I]_1,} \quad (\text{C.13})$$

with $[q]_1 = [1, 1, 0]^T$ and $[q]_2 = [1, -1, 0]^T$. Note that, the above eigen-decompositions hold true for both C - and b -based formulations.

Appendix D. Symbolic code for computing tangent eigensystems

Symbolic code for the analytic eigensystems of F -based formulation was given in Poya et al. [13]. In Fig. D.14 we provide the analytic eigensystems of C -based formulation from which the analytic eigensystem of b -based formulation can also be obtained using the push-forward operations defined in (65), (66) and (67).

Data availability

Data will be made available on request.

References

- [1] E. Medina, P.E. Farrell, K. Bertoldi, C.H. Rycroft, Navigating the landscape of nonlinear mechanical metamaterials for advanced programmability, *Phys. Rev. B* 101 (2020) 064101.
- [2] R.D. Lahuerta, E.T. Simões, E.M.B. Campello, P.M. Pimenta, E.C.N. Silva, Towards the stabilization of the low density elements in topology optimization with large deformation, *Comput. Mech.* 52 (4) (2013) 779–797.
- [3] F. Wang, B.S. Lazarov, O. Sigmund, J.S. Nergaard Jensen, Interpolation scheme for fictitious domain techniques and topology optimization of finite strain elastic problems, *Comput. Methods Appl. Mech. Engrg.* 276 (2014) 453–472.
- [4] R. Ortigosa, J. Martínez-Frutos, A. Gil, A new stabilisation approach for level-set based topology optimisation of hyperelastic materials, *Struct. Multidiscip. Optim.* 60 (2019).
- [5] P.E. Farrell, Á. Birkisson, S.W. Funke, Deflation techniques for finding distinct solutions of nonlinear partial differential equations, *SIAM J. Sci. Comput.* 37 (4) (2015) A2026–A2045.
- [6] J. Schröder, P. Wriggers, D. Balzani, A new mixed finite element based on different approximations of the minors of deformation tensors, *Comput. Methods Appl. Mech. Engrg.* 200 (49–52) (2011) 3583–3600.
- [7] J. Bonet, A.J. Gil, R. Ortigosa, A computational framework for polyconvex large strain elasticity, *Comput. Methods Appl. Mech. Engrg.* 283 (2015) 1061–1094.
- [8] J. Bonet, A.J. Gil, R. Ortigosa, On a tensor cross product based formulation of large strain solid mechanics, *Int. J. Solids Struct.* 84 (2016) 49–63.
- [9] A. Kraus, P. Wriggers, N. Viebahn, J. Schröder, Low order locking-free mixed finite element formulation with approximation of the minors of the deformation gradient, *Internat. J. Numer. Methods Engrg.* 120 (2019) 1011–1026.
- [10] M. Horák, A.J. Gil, R. Ortigosa, M. Kružík, A polyconvex transversely-isotropic invariant-based formulation for electro-mechanics: Stability, minimisers and computational implementation, *Comput. Methods Appl. Mech. Engrg.* 403 (2023) 115695.
- [11] G.H. Golub, C.F. Van Loan, *Matrix Computations*, fourth ed., Johns Hopkins Studies in Mathematical Sciences, Baltimore, USA, 2012.
- [12] J. Nocedal, S.J. Wright, *Numerical Optimization*, second ed., Springer, New York, NY, 2006.
- [13] R. Poya, R. Ortigosa, A.J. Gil, Variational schemes and mixed finite elements for large strain isotropic elasticity in principal stretches: Closed-form tangent eigensystems, convexity conditions, and stabilised elasticity, *Internat. J. Numer. Methods Engrg.* 124 (16) (2023) 3436–3493.
- [14] H. Chen, H.-T.D. Liu, D.I. Levin, C. Zheng, A. Jacobson, Stabler neo-hookean simulation: Absolute eigenvalue filtering for projected Newton, in: ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA, 2024.
- [15] J. Teran, E. Sifakis, G. Irving, R. Fedkiw, Robust quasistatic finite elements and flesh simulation, SCA '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 181–190.
- [16] A. Stomakhin, R. Howes, C. Schroeder, J.M. Teran, Energetically consistent invertible elasticity, in: J. Lee, P. Kry (Eds.), *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, The Eurographics Association, 2012.
- [17] B. Smith, F.D. Goes, T. Kim, Analytic eigensystems for isotropic distortion energies, *ACM Trans. Graph.* 38 (1) (2019).
- [18] R. Poya, R. Ortigosa, T. Kim, Geometric optimisation via spectral shifting, *ACM Trans. Graph.* 42 (3) (2023).
- [19] E. Riks, An incremental approach to the solution of snapping and buckling problems, *Int. J. Solids Struct.* 15 (7) (1979) 529–551.
- [20] M. Crisfield, A fast incremental/iterative solution procedure that handles “snap-through”, *Comput. Struct.* 13 (1) (1981) 55–62.
- [21] M.A. Crisfield, An arc-length method including line searches and accelerations, *Internat. J. Numer. Methods Engrg.* 19 (9) (1983) 1269–1289.
- [22] Y. Feng, D. Perić, D. Owen, Determination of travel directions in path-following methods, *Math. Comput. Modelling* 21 (7) (1995) 43–59.
- [23] Y. Feng, D. Perić, D. Owen, A new criterion for determination of initial loading parameter in arc-length methods, *Comput. Struct.* 58 (3) (1996) 479–485.
- [24] E. de Souza Neto, Y. Feng, On the determination of the path direction for arc-length methods in the presence of bifurcations and ‘snap-backs’, *Comput. Methods Appl. Mech. Engrg.* 179 (1) (1999) 81–89.
- [25] E. Carrera, A study on arc-length-type methods and their operation failures illustrated by a simple model, *Comput. Struct.* 50 (2) (1994) 217–229.
- [26] R. de Borst, M.A. Crisfield, J.J.C. Remmers, C.V. Verhoosel, *Nonlinear Finite Element Analysis of Solids and Structures*, second ed., John Wiley & Sons, Chichester, West Sussex, United Kingdom, 2012.
- [27] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, *Nonlinear Finite Elements for Continua and Structures*, second ed., Wiley, 2013.
- [28] J. Bonet, A.J. Gil, R.D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Dynamics*, first ed., Cambridge University Press, Cambridge, UK, 2021.
- [29] G. Petti, W.M. Coombs, C.E. Augarde, A displacement-controlled arc-length solution scheme, *Comput. Struct.* 258 (2022) 106674.
- [30] R. Ortigosa, D. Ruiz, A. Gil, A. Donoso, J. Bellido, A stabilisation approach for topology optimisation of hyperelastic structures with the SIMP method, *Comput. Methods Appl. Mech. Engrg.* 364 (2020) 112924.
- [31] R. Poya, R. Sevilla, A.J. Gil, A unified approach for a posteriori high-order curved mesh generation using solid mechanics, *Comput. Mech.* 58 (3) (2016) 457–490.
- [32] R. Poya, A.J. Gil, R. Ortigosa, R. Sevilla, J. Bonet, W.A. Wall, A curvilinear high order finite element framework for electromechanics: From linearised electro-elasticity to massively deformable dielectric elastomers, *Comput. Methods Appl. Mech. Engrg.* 329 (2018) 75–117.
- [33] R. Poya, A.J. Gil, R. Ortigosa, R. Palma, On a family of numerical models for couple stress based flexoelectricity for continua and beams, *J. Mech. Phys. Solids* 125 (2019) 613–652.
- [34] F. Brezzi, K.-J. Bathe, A discourse on the stability conditions for mixed finite element formulations, *Comput. Methods Appl. Mech. Engrg.* 82 (1) (1990) 27–57, Proceedings of the Workshop on Reliability in Computational Mechanics.

- [35] J. Schröder, N. Viebahn, P. Wriggers, F. Auricchio, K. Steeger, On the stability analysis of hyperelastic boundary value problems using three- and two-field mixed finite element formulations, *Comput. Mech.* 60 (3) (2017) 479–492.
- [36] F. Auricchio, L. Beirao da Veiga, C. Lovadina, A. Reali, A stability study of some mixed finite elements for large deformation elasticity problems, *Comput. Methods Appl. Mech. Engrg.* 194 (9) (2005) 1075–1092.
- [37] F. Auricchio, L. Beirao da Veiga, C. Lovadina, A. Reali, The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: mixed FEMs versus NURBS-based approximations, *Comput. Methods Appl. Mech. Engrg.* 199 (5) (2010) 314–323, Computational Geometry and Analysis.
- [38] W.A. Wall, M. Bischoff, E. Ramm, A deformation dependent stabilization technique, exemplified by EAS elements at large strains, *Comput. Methods Appl. Mech. Engrg.* 188 (4) (2000) 859–871.
- [39] J. Simo, F. Armero, R. Taylor, Improved versions of assumed enhanced strain tri-linear elements for 3D finite deformation problems, *Comput. Methods Appl. Mech. Engrg.* 110 (3) (1993) 359–386.
- [40] F. Armero, On the locking and stability of finite elements in finite deformation plane strain problems, *Comput. Struct.* 75 (3) (2000) 261–290.
- [41] M. Cervera, M. Chiumenti, R. Codina, Mixed stabilized finite element methods in nonlinear solid mechanics: Part I: Formulation, *Comput. Methods Appl. Mech. Engrg.* 199 (37) (2010) 2559–2570.
- [42] R. Pfefferkorn, P. Betsch, On transformations and shape functions for enhanced assumed strain elements, *Internat. J. Numer. Methods Engrg.* 120 (2) (2019) 231–261.
- [43] S. Bieber, F. Auricchio, A. Reali, M. Bischoff, Artificial instabilities of finite elements for nonlinear elasticity: Analysis and remedies, *Internat. J. Numer. Methods Engrg.* 124 (11) (2023) 2638–2675.
- [44] R. de Boer, *Vektor- und Tensorrechnung für Ingenieure*, Springer, 1982.
- [45] S. Hartmann, P. Neff, Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility, *Int. J. Solids Struct.* 40 (2003) 2767–2791.
- [46] B. Dacorogna, second ed., *Direct Methods in the Calculus of Variations*, vol. 78, Springer, 2008.
- [47] M. Gurtin, *An Introduction to Continuum Mechanics*, Academic Press, 1981.
- [48] R.W. Ogden, *Non-Linear Elastic Deformations*, Dover Civil and Mechanical Engineering, Dover Publications, 1997.
- [49] Q.-S. Zheng, Theory of representations for tensor functions - a unified invariant approach to constitutive equations, *Appl. Mech. Rev.* 47 (11) (1994) 545–587.
- [50] J. Schröder, P. Neff, V. Ebbing, Anisotropic polyconvex energies on the basis of crystallographic motivated structural tensors, *J. Mech. Phys. Solids* 56 (12) (2008) 3486–3506.
- [51] J.E. Marsden, T.J.R. Hughes, *Mathematical Foundations of Elasticity*, Dover Publications, Inc, New York, 1983.
- [52] J. Bonet, A.J. Gil, R.D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*, third ed., Cambridge University Press, Cambridge, UK, 2016.
- [53] P. Wriggers, *Nonlinear Finite Element Methods*, Springer, 2008.
- [54] E.A. de Souza Neto, D. Perić, D.R.J. Owen, *Computational Methods for Plasticity: Theory and Applications*, First, John Wiley & Sons, Chichester, West Sussex, United Kingdom, 2008.
- [55] J.M. Ball, Convexity conditions and existence theorems in nonlinear elasticity, *Arch. Ration. Mech. Anal.* 63 (4) (1976) 337–403.
- [56] J.M. Ball, Energy-minimising configurations in nonlinear elasticity, in: *Proceedings of the International Congress of Mathematicians*, Warsaw, 1983.
- [57] J. Bonet, A.J. Gil, C.H. Lee, M. Aguirre, R. Ortigosa, A first order hyperbolic framework for large strain computational solid dynamics. Part I: Total Lagrangian isothermal elasticity, *Comput. Methods Appl. Mech. Engrg.* 283 (2015) 689–732.
- [58] T. Kim, D. Eberle, Dynamic deformables: Implementation and production practicalities (now with code!), in: *ACM SIGGRAPH 2022 Courses*, SIGGRAPH '22, Association for Computing Machinery, New York, NY, USA, 2022.
- [59] H. Xu, F. Sin, Y. Zhu, J. Barbić, Nonlinear material design using principal stretches, *ACM Trans. Graph.* 34 (4) (2015).
- [60] Y. Zhu, Eigen space of mesh distortion energy hessian, 2021.
- [61] T. Kim, F. De Goes, H. Iben, Anisotropic elasticity for inversion-safety and element rehabilitation, *ACM Trans. Graph.* 38 (4) (2019).
- [62] R. Poya, A.J. Gil, R. Ortigosa, A high performance data parallel tensor contraction framework: Application to coupled electro-mechanics, *Comput. Phys. Comm.* 216 (2017) 35–52.
- [63] J. Smith, S. Schaefer, Bijective parameterization with free boundaries, *ACM Trans. Graphs.* 34 (2015) 1–9.
- [64] M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin, D. Panozzo, C. Jiang, D.M. Kaufman, Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics, *ACM Trans. Graph.* 39 (4) (2020).
- [65] The Standard NAFEMS Benchmarks. National Agency for Finite Element Methods and Standards (U.K.), Z-section cantilever - analysis type: Linear elastic, 1990.
- [66] P. Krysl, J.-S. Chen, Benchmarking computational shell models, *Arch. Comput. Methods Eng.* 30 (1) (2022) 301–315.
- [67] F. Auricchio, G. Scalet, P. Wriggers, Fiber-reinforced materials: Finite elements for the treatment of the inextensibility constraint, *Comput. Mech.* 60 (6) (2017) 905–922.
- [68] P. Wriggers, F. Auricchio, J. Schröder, Finite element formulations for large strain anisotropic material with inextensible fibers, *Adv. Model. and Simul. in Eng. Sci.* 3 (25) (2016).
- [69] D. Chapelle, K. Bathe, The inf-sup test, *Comput. Struct.* 47 (4) (1993) 537–545.
- [70] F. Auricchio, L.B. da Veiga, F. Brezzi, C. Lovadina, Mixed finite element methods, in: *Encyclopedia of Computational Mechanics Second Edition*, John Wiley and Sons, Ltd, 2017, pp. 1–53.
- [71] A.N. Brooks, T.J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 32 (1) (1982) 199–259.
- [72] T.J. Hughes, L.P. Franca, M. Balestra, A new finite element formulation for computational fluid dynamics: V. circumventing the babuška-brezzini condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations, *Comput. Methods Appl. Mech. Engrg.* 59 (1) (1986) 85–99.
- [73] T. Elguedj, T. Hughes, Isogeometric analysis of nearly incompressible large strain plasticity, *Comput. Methods Appl. Mech. Engrg.* 268 (2014) 388–416.