Implementation of displacement-based linear and nonlinear elastostatic user element subroutine (UEL) in Abaqus/Standard

Bibekananda Datta

PhD student, Mechanical Engineering
Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218

Preface

Finite element analysis/method (FEA/M) is a widely adopted numerical method to solve partial differential equations that originated in the field of structural mechanics. Eventually, FEM transcended beyond structural mechanics and established itself as one of the primary methods in continuum mechanics and multiphysics modeling. One of the most popular and user-friendly commercial finite element programs for nonlinear mechanics problems is Abaqus/ Standard which provides a plethora of built-in element technologies, constitutive models, and contact modeling features. Additionally, it provides user programming capability through which users can enhance the existing features or finite element capabilities known as user subroutines. User-defined element subroutine (UEL) is the most high-level interface for users to program new element technology for nonlinear solid mechanics and multiphysics modeling.

This document is a brief overview of the theoretical background, algorithm, and implementation of the user element subroutine (UEL) in Abaqus/ Standard using isotropic linear elasticity and hyperelasticity as examples. This document assumes that the users are familiar with the fundamentals of continuum mechanics, finite element analysis, common features of Abaqus/ Standard, and Fortran programming. Fortran-based implementations and corresponding Abaqus input files accompanying those implementations are available in the following GitHub repositories:

- Isotropic linear elasticity: https://github.com/bibekananda-datta/Abaqus-UEL-Elasticity
- Hyperelasticity: https://github.com/bibekananda-datta/Abaqus-UEL-Hyperelasticity

For both cases, the implementation includes 3D continuum implementation and 2D plane strain implementation. I should warn the users that, the implementations are based on the standard displacement-based finite element formulation and do not include any special techniques to circumvent advanced numerical issues such as shear or volumetric locking, hourglass modes, etc. often observed in finite element simulations. Using these implementations or algorithms as templates, users can extend the codes to include more capabilities. It should be straightforward to include axisymmetric elements in these templates but including plane stress case, especially for the hyperelastic problem, will require some extra effort.

Contents

1 Brief review of isotropic linear elastostatics

Let $\Omega(\mathbf{x}) \in \mathbb{R}^3$ be an isotropic linear elastic body undergoing small deformation under mechanical loading. For small deformation, the infinitesimal strain tensor can be defined as,

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left[\mathrm{grad}(\mathbf{u}) + (\mathrm{grad}(\mathbf{u}))^\top\right] = \mathrm{sym}[\mathrm{grad}(\mathbf{u})], \tag{1.1}$$

where, $\mathbf{u}$ is the displacement field. The strain tensor is often decomposed into a deviatoric, $\varepsilon_{\mathrm{dev}}$, and a volumetric part, $\varepsilon_{\mathrm{vol}}$, as,

$$\boldsymbol{\varepsilon} = \underbrace{\left(\boldsymbol{\varepsilon} - \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\varepsilon})\mathbb{1}\right)}_{\varepsilon_{\mathrm{dev}}} + \underbrace{\frac{1}{3}\,\mathrm{tr}(\boldsymbol{\varepsilon})\mathbb{1}}_{\varepsilon_{\mathrm{vol}}} \qquad \Rightarrow \varepsilon_{ij} = \underbrace{\left(\varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}\delta_{ij}\right)}_{\varepsilon_{\mathrm{dev}}} + \underbrace{\frac{1}{3}\varepsilon_{kk}\delta_{ij}}_{\varepsilon_{\mathrm{vol}}} \tag{1.2}$$

The constitutive relation for linear elastic material is given by Hooke's law as,

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}, \qquad \Rightarrow \boldsymbol{\sigma} = \lambda\,\mathrm{tr}(\boldsymbol{\varepsilon})\mathbb{1} + 2\mu\boldsymbol{\varepsilon}, \qquad \Rightarrow \sigma_{ij} = \mathbb{C}_{ijkl}\varepsilon_{kl} = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij}, \tag{1.3}$$

where, $\boldsymbol{\sigma}$ is known as the Cauchy stress. Here, $\mathbb{C}$ is known as the fourth-order elasticity or modulus tensor, and μ and λ are known as Lamé constants. These parameters can be expressed in terms of more familiar parameters, *i.e.*, Young's modulus, E, and Poisson's ratio, ν, as,

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}. \tag{1.4}$$

For three-dimensional isotropic linear elastic materials, using Voigt notation, the Hooke's law can be written as,

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{13}\\ \sigma_{12}\end{bmatrix} = \begin{bmatrix}2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0\\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0\\ 0 & 0 & 0 & \mu & 0 & 0\\ 0 & 0 & 0 & 0 & \mu & 0\\ 0 & 0 & 0 & 0 & 0 & \mu\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{23}\\ 2\varepsilon_{13}\\ 2\varepsilon_{12}\end{bmatrix} \tag{1.5}$$

For two-dimensional (plane stress and plane strain) cases,

$$\underbrace{\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ 2\varepsilon_{12}\end{bmatrix}}_{\text{plane stress}}, \quad \underbrace{\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix}1-\nu & \nu & 0\\ \nu & 1-\nu & 0\\ 0 & 0 & \frac{1-2\nu}{2}\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ 2\varepsilon_{12}\end{bmatrix}}_{\text{plane strain}} \tag{1.6}$$

Remark 1. The elasticity tensor (sometimes referred to as the tangent modulus, especially in the context of finite elasticity), $\mathbb{c}$, is a fourth-order tensor of dimension $3 \times 3 \times 3 \times 3$. Similarly, stress and strain tensors are second-order tensors of dimension 3×3. The dimension mentioned here corresponds to the three-dimensional cases since two-dimensional plane stress and plane strain cases are somewhat idealizations of reality. However, because of the symmetry of the stress and strain tensors, it is possible to reduce the order of stress and strain tensors and represent them as a one-dimensional tensor (*i.e.*, vector) form of dimension 6×1. Consequently, the elasticity tensor, $\mathbb{c}$, can be reduced to a second-order tensor, $\mathbf{D}$, of dimension 6×6. Following this procedure, the elasticity tensor is represented in its reduced second-order form in (1.5) and (1.6). This is known as Voigt notation.

2 Displacement-based finite element formulation

2.1 Strong form of the governing equations of boundary value problem

The governing partial differential equation for stress equilibrium as well as the boundary conditions in terms of the displacement vector, $\mathbf{g}$, on Γ_g and the Cauchy traction, $\mathbf{t}$, on Γ_t are given by,

$$\begin{aligned} \operatorname{div}(\boldsymbol{\sigma}) + \rho\mathbf{b} &= 0 \qquad && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_g, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t} && \text{on } \Gamma_t, \end{aligned} \tag{2.1}$$

where, $\mathbf{b}$ is the body force per mass.

2.2 Weighted residual based weak formulation of the governing equations

Let $\mathbf{u} \in \mathscr{U}$ be the trial solution which satisfies $u_i = g_i$, and $\mathbf{w} \in \mathscr{W}$ be a vector test (or weight) function which satisfies $w_i = 0$ on Γ_g, I can write the weak form of momentum balance equation as

$$\int\limits_{\Omega} \operatorname{div}(\boldsymbol{\sigma}) \cdot \mathbf{w} \; dv + \int\limits_{\Omega} \rho\mathbf{b} \cdot \mathbf{w} \; dv = 0. \tag{2.2}$$

Let me use the following identity and divergence theorem on the above weak form,

$$\begin{aligned} &\operatorname{div}(\mathbf{A}^\top\mathbf{u}) = \operatorname{div}(\mathbf{A}) \cdot \mathbf{u} + \mathbf{A} : \operatorname{grad}(\mathbf{u}), \\ &\text{and} \int\limits_{\Omega} \operatorname{div}(\mathbf{A}^\top\mathbf{u}) \; dv = \int\limits_{\partial\Omega} (\mathbf{A}\mathbf{n}) \cdot \mathbf{u} \; ds. \end{aligned} \tag{2.3}$$

Thus, the global weak form, $\mathcal{W}_{\mathbf{u}}(\mathbf{u})$, for a linear elastic material can be written as,

$$\mathcal{W}_{\mathbf{u}}(\mathbf{u}) = -\int\limits_{\Omega} \boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u})\right] : \operatorname{grad}(\mathbf{w}) \; dv + \int\limits_{\Omega} \rho\mathbf{b} \cdot \mathbf{w} \; dv + \int\limits_{\Gamma_t} \mathbf{t} \cdot \mathbf{w} \; ds = 0, \tag{2.4}$$

The above statement is equivalent to the variational principle (or principle of virtual work) where the first term represents internal work and the second and third terms combined represent external work.

2.3 Discretization and Buvnov-Galerkin finite element approximation

The computational domain and its boundary are discretized using finite elements, *i.e.*, $\Omega_0 = \cup\, \Omega_0^e$ and $\partial\Omega_0 = \cup\, \partial\Omega_0^e$. This approximation allows me to write the system of residuals for each element as (Zienkiewicz et al., 2014, Chapter 2). Thus, the element-level weak form, $\mathcal{W}_{\mathbf{u}}^e(\mathbf{u})$, can be written as,

$$\mathcal{W}_{\mathbf{u}}^e(\mathbf{u}) = -\int\limits_{\Omega^e} \boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u})\right] : \mathrm{grad}(\mathbf{w})\ dv + \int\limits_{\Omega^e} \rho\mathbf{b}\cdot\mathbf{w}\ dv + \int\limits_{\Gamma_t^e} \mathbf{t}\cdot\mathbf{w}\ ds = 0, \tag{2.5}$$

where, $\boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u})\right]$ is the Cauchy stress, and $\mathbf{t} = \boldsymbol{\sigma}\cdot\mathbf{n}$ is the Cauchy traction. The infinitesimal strain tensor is given by, $\boldsymbol{\varepsilon}(\mathbf{u}) = \mathrm{sym}[\mathrm{grad}(\mathbf{u})]$.

As is standard in finite element literature, I employ the following discretization for the coordinate s of an element, $\mathbf{x}$, nodal displacement, $\mathbf{u}$, and weight functions, $\mathbf{w}$,

$$\begin{aligned}
\mathbf{x} = \mathbf{N_u}(\mathbf{x})\mathbf{x}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}_{\mathbf{u}}^a(\mathbf{x})\mathbf{x}^a, \quad &\Rightarrow \quad x_i &= \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{x})x_i^a \\
\mathbf{u}(\mathbf{x}) = \mathbf{N_u}(\mathbf{x})\mathbf{u}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}_{\mathbf{u}}^a(\mathbf{x})\mathbf{u}^a, \quad &\Rightarrow \quad u_i(\mathbf{x}) &= \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{x})u_i^a, \\
\mathbf{w}(\mathbf{x}) = \mathbf{N_u}(\mathbf{x})\mathbf{w}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}_{\mathbf{u}}^a(\mathbf{x})\mathbf{w}^a, \quad &\Rightarrow \quad w_i(\mathbf{x}) &= \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{x})w_i.
\end{aligned} \tag{2.6}$$

Here, $\mathbf{x}_e$ and $\mathbf{u}_e$ are the vectors containing nodal coordinates and nodal displacement of an element. $\mathbf{w}_e$ is the corresponding arbitrary test functions. Here, $\mathbf{N_u}$ is known as the interpolation (or shape) function matrix corresponding to all the nodes and $\mathbf{N}_{\mathbf{u}}^a$ is the repeating sub-matrix corresponding to each node, a.

For a three-dimensional case, the matrix representation of interpolation functions, $\mathbf{N}_{\mathbf{u}}^a$, and the vector representation of coordinates of a node, $\mathbf{x}^a$, can be written as,

$$\mathbf{N}_{\mathbf{u}}^a = \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix}_{\mathrm{nsd}\times\mathrm{nsd}} , \quad \mathbf{x}^a = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_{\mathrm{nsd}\times 1} , \tag{2.7}$$

For a two-dimensional case, $[\mathbf{N}_{\mathbf{u}}^a]$ can be reduced by eliminating 3rd row and 3rd column, and $\mathbf{x}^a$ is accordingly modified by eliminating 3rd row. Displacement $\mathbf{u}$ and test function $\mathbf{w}$ also can be represented using a similar form. Using $[\mathbf{N}_{\mathbf{u}}^a]$ as the repeating sub-matrix block for the shape function matrix, $[\mathbf{N_u}]$, I can write, $\mathbf{N_u}$, and nodal coordinate of the element,

$\mathbf{x}_e$, as,

$$\mathbf{N_u} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \cdots & N_{\text{nen}} & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \cdots & 0 & N_{\text{nen}} & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \cdots & 0 & 0 & N_{\text{nen}} \end{bmatrix}_{\text{nsd}\times\text{nsd}*\text{nen}}, \tag{2.8}$$
$$\text{and} \quad \mathbf{x}_e = \begin{bmatrix} x_1 & y_1 & z_1 & x_2 & y_2 & z_2 & \cdots & \cdots & \cdots & x_{\text{nen}} & y_{\text{nen}} & z_{\text{nen}} \end{bmatrix}^{\top}_{\text{nsd}*\text{nen}\times 1}.$$

Substituting the discretization of $\mathbf{w} = \mathbf{N_u}\mathbf{w}_e$ in the corresponding global form (2.5), for any arbitrary test functions, $\mathbf{w}_e$, I have the residual as,

$$\mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) = \int_{\Omega^e} \boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u}_e)\right] : \frac{\partial \mathbf{N_u}}{\partial \mathbf{x}}\, dv - \int_{\Omega^e} \rho \mathbf{N}^{\top}_{\mathbf{u}} \mathbf{b}\, dv - \int_{\Gamma^e_t} \mathbf{N}^{\top}_{\mathbf{u}} \mathbf{t}^e\, ds = 0, \tag{2.9}$$

Recognizing

$$\boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u}_e)\right] : \frac{\partial \mathbf{N_u}}{\partial \mathbf{x}} = \frac{\partial \mathbf{N_u}}{\partial \mathbf{x}} : \boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u}_e)\right] = \mathbf{B}^{\top}_{\mathbf{u}} \boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u}_e)\right]. \tag{2.10}$$

Here, $[\mathbf{B_u}]$ is known as the strain-displacement matrix. I should note that, since the Cauchy stress is symmetric, *i.e.*, $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\top}$, for a two-dimensional case, $[\mathbf{B_u}]$ has a dimension of $[\mathbf{B_u}]_{3\times\text{nsd}*\text{nen}}$, and for three-dimensional case, the dimension is $[\mathbf{B_u}]_{6\times\text{nsd}*\text{nen}}$. Similar to the shape function matrix, $[\mathbf{N_u}]$, the strain-displacement matrix, $[\mathbf{B_u}]$, is also composed of repetitive sub-matrix, $[\mathbf{B}^a_{\mathbf{u}}]$. The matrix form of $[\mathbf{B_u}]$ and $[\mathbf{B}^a_{\mathbf{u}}]$ are given by,

$$\mathbf{B_u} = \begin{bmatrix} \mathbf{B}^1_{\mathbf{u}} & \mathbf{B}^2_{\mathbf{u}} & \mathbf{B}^3_{\mathbf{u}} & \cdots & \cdots & \mathbf{B}^{\text{nen}}_{\mathbf{u}} \end{bmatrix}_{\text{nstress}\times\text{nsd}*\text{nen}}, \quad \text{where,} \quad \mathbf{B}^a_{\mathbf{u}} = \begin{bmatrix} N^a_{,1} & 0 & 0 \\ 0 & N^a_{,2} & 0 \\ 0 & 0 & N^a_{,3} \\ 0 & N^a_{,3} & N^a_{,2} \\ N^a_{,3} & 0 & N^a_{,1} \\ N^a_{,2} & N^a_{,1} & 0 \end{bmatrix}_{\text{nstress}\times\text{nsd}}. \tag{2.11}$$

For two-dimensional plane strain and plane stress cases, the matrix form of $[\mathbf{B}^a_{\mathbf{u}}]$ and as well as the vector form of $\boldsymbol{\sigma}$ can be easily reduced. The dimension for $[\mathbf{B}^a_{\mathbf{u}}]$ is $[\mathbf{B}^a_{\mathbf{u}}]_{3\times\text{nsd}}$. Now, with these definitions, I can write the residual as,

$$\mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) = -\int_{\Omega^e} \mathbf{B}^{\top}_{\mathbf{u}} \boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u}_e)\right]\, dv + \int_{\Omega^e} \rho \mathbf{N}^{\top}_{\mathbf{u}} \mathbf{b}\, dv + \int_{\Gamma^e_t} \mathbf{N}^{\top}_{\mathbf{u}} \mathbf{t}^e\, ds = 0, \tag{2.12}$$

In the element residual, the first term corresponds to internal force, f_{int}, and the second and third term combined corresponds to external force, f_{ext}. At equilibrium, the difference between external and internal force should be zero and this is what will give us the displacement solution. Now, by using the discretized constitutive relation for the Cauchy stress can be written as,

$$\boldsymbol{\sigma}\left[\boldsymbol{\varepsilon}(\mathbf{u}_e)\right] = \mathbf{D}\mathbf{B}_{\mathbf{u}}\mathbf{u}_e, \tag{2.13}$$

where $\mathbf{D}$ is the stiffness tensor for Hookean materials with a dimension of $[\mathbf{D}]_{\text{nStress}\times\text{nStress}}$ where nStress = 6 for three-dimensional cases and nStress = 3 for two-dimensional plane strain and plane stress cases. For standard compressible isotropic linear elastic materials, $\mathbb{C}$ can mapped from $\mathbf{D}$ using Voigt notation. By substituting the discretized constitutive relation, the mechanical residual for an element can be written as,

$$\mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) = -\int\limits_{\Omega^e} \mathbf{B}_{\mathbf{u}}^\top\mathbf{D}\mathbf{B}_{\mathbf{u}}\mathbf{u}\ dv + \int\limits_{\Omega^e} \rho\mathbf{N}_{\mathbf{u}}^\top\mathbf{b}\ dv + \int\limits_{\Gamma^e_t} \mathbf{N}_{\mathbf{u}}^\top\mathbf{t}^e\ ds = 0 \tag{2.14}$$

Following standard finite element literature, for linear elastic materials, I can write,

$$\underbrace{\int\limits_{\Omega^e} \mathbf{B}_{\mathbf{u}}^\top\mathbf{D}\mathbf{B}_{\mathbf{u}}\ dv}_{\mathbf{k}^e}\ \ \mathbf{u}_e = \underbrace{\int\limits_{\Omega^e} \rho\mathbf{N}_{\mathbf{u}}^\top\mathbf{b}\ dv - \int\limits_{\Gamma^e_t} \mathbf{N}_{\mathbf{u}}^\top\mathbf{t}\ ds}_{\mathbf{f}^e}, \qquad \Rightarrow [\mathbf{k}^e]\{\mathbf{u}\} = \{\mathbf{f}^e\}. \tag{2.15}$$

where $[\mathbf{k}^e]$ is the element stiffness matrix, $\mathbf{u}$ is the nodal displacement of the element, and $\mathbf{f}^e$ is external forces. The dimension of element stiffness matrix is $[\mathbf{k}^e]_{\text{nsd}*\text{nen}\times\text{nsd}*\text{nen}}$, and the dimension of force vector is $\{\mathbf{f}^e\}_{\text{nsd}*\text{nen}\times 1}$.

The global stiffness matrix, $[\mathbf{k}]$, and the global force vector, $\mathbf{f}$, can be obtained by assembling the element-level quantities as,

$$\mathbf{K} = \mathop{\mathsf{A}}_{e=1}^{\text{nel}} \mathbf{k}^e \qquad \text{and} \qquad \mathbf{f} = \mathop{\mathsf{A}}_{e=1}^{\text{nel}} \mathbf{f}^e \tag{2.16}$$

Thus, the final global matrix form is,

$$[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{f}\}, \tag{2.17}$$

which can be solved for nodal displacements, $\mathbf{u}$, with appropriate boundary conditions being imposed on the linear system of equations.

3 Isoparametric element formulation and numerical integration

So far, I have kept the form of the shape functions abstract. Shape functions are typically defined using the Lagrangian interpolation procedure as a function of the nodal coordinates $\mathbf{N}(\mathbf{x})$. However, it is possible to use different interpolation procedures to define the shape functions. For a large computational model with multiple elements, defining individual shape functions becomes computationally challenging. Thus, in practice, the shape functions are defined in a normalized local iso-parametric coordinate system, $\mathbf{N}(\boldsymbol{\xi})$, where $\boldsymbol{\xi} \in [-1, 1]$. These elements are called master or parent elements. Thus, a mapping from the local coordinates to the global coordinate system is necessary.

Typically during implementation, $[\mathbf{N}(\boldsymbol{\xi})]_{\text{nen}\times 1}$ is defined as a vector containing the shape functions corresponding to the element nodes. At the same time, $\left[\frac{\partial \mathbf{N}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]_{\text{nen}\times\text{nsd}}$ is defined as a matrix containing the derivatives of shape function which is essentially used to form the strain-displacement matrix, $[\mathbf{B_u}]$. Thus, coordinate mapping for the shape functions is,

$$\begin{aligned}\left[\frac{\partial \mathbf{N}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]_{\text{nen}\times\text{nsd}} &= \left[\frac{\partial \mathbf{N}(\boldsymbol{\xi})}{\partial \mathbf{x}}\right]_{\text{nen}\times\text{nsd}} \left[\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right]_{\text{nsd}\times\text{nsd}}, \\ \Rightarrow \left[\frac{\partial \mathbf{N}(\boldsymbol{\xi})}{\partial \mathbf{x}}\right]_{\text{nen}\times\text{nsd}} &= J_{\boldsymbol{\xi}}^{-1} \left[\frac{\partial \mathbf{N}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]_{\text{nen}\times\text{nsd}}.\end{aligned} \tag{3.1}$$

Here, $J_{\boldsymbol{\xi}} = \left[\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right]$ is known as element jacobian.

3.1 Evaulation of volume integrals

Using the definition of iso-parametric coordinate system, any generic volume integral can be mapped as,

$$\int_{\Omega^e} f(\mathbf{x})\ dv = \int_{\square} \det(J_{\boldsymbol{\xi}}) \hat{f}(\boldsymbol{\xi})\ d\,\square \tag{3.2}$$

Since $\boldsymbol{\xi} \in [-1, 1]$, it is straightforward and convenient to apply the Gauss quadrature approach to evaluate the integrals within the element stiffness matrix and force vectors. For volume integrals, it can be done as,

$$\begin{aligned}\int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \det(J_{\boldsymbol{\xi}})\hat{f}(\boldsymbol{\xi})\ d\xi_1 d\xi_2 d\xi_3 &= \sum_i\sum_j\sum_k \det(J_{\boldsymbol{\xi}}) W_i W_j W_k \hat{f}(\xi_1^i, \xi_2^j, \xi_3^k), \\ &= \sum_{i=1}^{\text{nip}} \det(J_{\boldsymbol{\xi}}) W \hat{f}(\xi_1^i, \xi_2^j, \xi_3^k).\end{aligned} \tag{3.3}$$

Here, $W = W_i W_j W_k$. For standard Lagrangian elements, the values of Gauss weights and local coordinates of integration points are available in any standard finite element textbook (Zienkiewicz et al., 2013), thus, I did not list them here.

Remark 2. To implement surface traction or pressure boundary conditions, it is also required to calculate the surface integral. However, for user-defined elements, these boundary conditions can be implemented using built-in Abaqus elements overlaid on the user elements which I will refer to as the "dummy" element technique. The same "dummy" elements are used for visualizing the results in Abaqus (to be discussed later).

4 Implementation of displacement element in ABAQUS/Standard

Since the user-element subroutine (UEL) in ABAQUS/Standard allows programming both linear and nonlinear physical and material behavior, to maintain generality of the programming interface, it asks the user to program the element stiffness matrix, `AMATRX` and element residual vector `RHS` instead of the force vector that appeared in our formulation. If I consider the element residual, $\mathbf{R}^e_{\mathbf{u}}(\mathbf{u})$ to be a generic nonlinear function of the displacement field, $\mathbf{u}$, I need to linearize the element residual first as follows,

$$\mathbf{R}_{\mathbf{u}}^{e}\left(\mathbf{u}_{e}+\Delta\mathbf{u}_{e}\right)=\mathbf{R}_{\mathbf{u}}^{e}(\mathbf{u})+\frac{\partial\mathbf{R}_{\mathbf{u}}^{e}(\mathbf{u})}{\partial\mathbf{u}}\Delta\mathbf{u} \tag{4.1}$$

To obtain a solution of $\Delta\mathbf{u}$, the perturbed residual has to vanish, thus giving me

$$-\frac{\partial\mathbf{R}_{\mathbf{u}}^{e}}{\partial\mathbf{u}_{e}}\Delta\mathbf{u}_{e}=\mathbf{R}_{\mathbf{u}}^{e},\quad\Rightarrow\mathbf{k}^{e}\Delta\mathbf{u}_{e}=\mathbf{R}_{\mathbf{u}}^{e} \tag{4.2}$$

It is evident that the definition of the element stiffness matrix is as follows,

$$\mathbf{k}^{e}=-\frac{\partial\mathbf{R}_{\mathbf{u}}^{e}}{\partial\mathbf{u}_{e}}. \tag{4.3}$$

According to ABAQUS documentation, for user-defined element subroutines (UEL), the element stiffness matrix is called, `AMATRX`, and the residual is called the right-hand side vector, `RHS`. The user is responsible for programming the element stiffness matrix and residual vector. Once the element stiffness matrix is available, ABAQUS performs the assembly and application of displacement (Dirichlet or kinematic type) boundary conditions to the discretized global system. Consequently, by employing an initial guess for the nodal displacement field, $\Delta\mathbf{u}$ is solved at each iteration until the residual falls below a certain tolerance (close to zero). For linear material, it takes a single iteration to obtain the solution. However, for nonlinear cases, it takes more than a few iterations to achieve convergence.

4.1 Algorithmic procedure for small strain displacement element

The following algorithmic procedure illustrates a simple implementation of displacement-based finite element formulation of isotropic linear elasticity. However, the implementation can be easily extended to include different constitutive responses such as hypoelasticity, elasto-plasticity, viscoelasticity, etc. by changing the subroutine `umatElastic` with an appropriate set of input and output arguments. This subroutine calculates the constitutive behavior of the material, *i.e.*, stress, and elasticity tensor, in the tensor form and then maps it to vector-matrix form using Voigt convention.

Procedure 1: ABAQUS/ Standard user element subroutine (UEL) implementation for isotropic linear elasticity

```
Input  : PROPS, COORDS, JELEM, JTYPE, NNODE, NDOFEL, TIME, DTIME, U,
         DU, V, A, PREDEF, JDLTYP, NDLOAD, MDLOAD, DDLMAG, ALMAG
Output: AMATRX, RHS, PNEWDT, ENERGY, SVARS

Get nInt ← PROPS and nDim, nTens ← JTYPE
Initialize $\{\mathbf{N}^a_{\mathbf{u}}, \mathbf{B}^a_{\mathbf{u}}, \mathbf{N}_{\mathbf{u}}, \mathbf{B}_{\mathbf{u}}, \mathbf{k}^e_{\mathbf{uu}}\} = 0$
Get nodal displacement vector of the element, $\mathbf{u}_e, \Delta\mathbf{u}_e$
Get $w_{\text{int}}, \boldsymbol{\xi}_{\text{int}}$ ← SUBROUTINE gaussQuadrtr(nDim, nNode)
for $k = 1$ to nInt do
    Get $[\mathbf{N}_{\mathbf{u}}], \left[\frac{\partial \mathbf{N}_{\mathbf{u}}}{\partial \boldsymbol{\xi}}\right]$ ← SUBROUTINE interpFunc(nDim, nNode)
    Calculate $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = [\overline{\mathbf{x}}_e]\left[\frac{\partial \mathbf{N}_{\mathbf{u}}}{\partial \boldsymbol{\xi}}\right]$
    Calculate $\frac{\partial \mathbf{N}_{\mathbf{u}}}{\partial \mathbf{x}} = \frac{\partial \mathbf{N}_{\mathbf{u}}}{\partial \boldsymbol{\xi}}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right)^{-1}$ and $J_{\xi}$
    for $i = 1$ to nNode do
        Form $\mathbf{N}^a_{\mathbf{u}}$(nDim, NDOFEL) and $\mathbf{B}^a_{\mathbf{u}}$(nTens, nDim) matrix
        $\mathbf{N}_{\mathbf{u}}$(1 : nDim, nDim * (i − 1) + 1 : nDim * i) = $\mathbf{N}^a_{\mathbf{u}}$(1 : nDim, 1 : nDim)
        $\mathbf{B}_{\mathbf{u}}$(1 : nTens, nDim * (i − 1) + 1 : nDim * i) = $\mathbf{B}^a_{\mathbf{u}}$(1 : nTens, 1 : nDim)
    end
                                                        // end of nodal loop
    Calculate $\boldsymbol{\varepsilon} = \mathbf{B}_{\mathbf{u}}\mathbf{u}_e$
    Get $\mathbf{D}, \boldsymbol{\sigma}$ ← SUBROUTINE umatElastic(PROPS, $\boldsymbol{\varepsilon}$)      // UMAT returns Cauchy
     stress and material tangent in voigt form
    Store $\boldsymbol{\sigma}, \boldsymbol{\varepsilon}$ → SVARS $\mathbf{k}^e_{\mathbf{uu}} = \mathbf{k}^e_{\mathbf{uu}} + w_{\text{int}}(k)\det(J_{\xi})\left(\mathbf{B}^{\top}_{\mathbf{u}}\mathbf{D}\mathbf{B}_{\mathbf{u}}\right)$
    $\mathbf{R}_{\mathbf{u}} = \mathbf{R}_{\mathbf{u}} - w_{\text{int}}(k)\det(J_{\xi})\left(\mathbf{B}^{\top}_{\mathbf{u}}\boldsymbol{\sigma} - \rho\mathbf{N}^{\top}_{\mathbf{u}}\mathbf{b}\right)$
end
                                              // end of integration point loop
Assign AMATRX = $\mathbf{k}^e_{\mathbf{uu}}$ and RHS = $\mathbf{R}_{\mathbf{u}}$
```

4.2 Brief description of the user element subroutine in GitHub repository

Warning: This is a very brief description of the subroutine and example input file. Interested users should go through the subroutine to understand the implementation and consult Abaqus documentation on the user element to understand the keywords used in the input files.

The user element subroutine (UEL) available on the GitHub repository for isotropic linear elasticity, `uel_mech.for` implemented 3D continuum solid elements and 2D plane strain elements by making multiple options available to the user. The following table lists the available user element types with corresponding tags and the number of integration points (reduced and/or full) to be used as a property.

For the set of elements defined as `user element` in the Abaqus input file, Abaqus solver

Type of element	UEL element tag	Integration points `(nInt)`
4-node tetrahedral	`U1`	1 (full)
8-node hexahedral	`U2`	8 (full) and 1 (reduced)
3-node triangular (plane-strain)	`U3`	1 (full)
4-node quadrilateral (plane-strain)	`U4`	4 (full) and 1 (reduced)
10-node tetrahedral	`U5`	4 (full)
20-node hexahedral	`U6`	27 (full) and 8 (reduced)
6-node triangular (plane-strain)	`U7`	3 (full)
8-node quadrilateral (plane-strain)	`U8`	9 (full) and 4 (reduced)

Table 1: Types of available small displacement mechanical elements in Abaqus UEL.

calls the subroutine `UEL(...)` at each iteration (for Newton-Raphson solver) or every few iterations (for modified Newton-Raphson solver) which returns `AMATRX` and `RHS` for global assembly and solution. To enhance the capabilities of the default `UEL(...)` subroutine from Abaqus, an additional subroutine is used with an extended set of input arguments. This subroutine requires the user to enter 4 properties as input; two of which are `real` properties and rest two are `integer` properties.

Name of the property	Property location	UEL variable
Elastic modulus, E	`props(1)`	`E`
Poisson ratio, ν	`props(2)`	`nu`
No of integration points	`jprops(1)`	`nInt`
No of post-processing variables	`jprops(2)`	`nPostVars`

Table 2: List of properties for the small displacement-based isotropic linear elastic element.

One of the main caveats of the Abaqus user subroutine is that the visualization of the element output is not natively supported on Abaqus/ Viewer. If the user subroutine shares element connectivity and integration points with any built-in Abaqus element, users can overlay the built-in Abaqus elements with negligible elastic properties on the user elements. Henceforth, the user can store the element output results using the global variable technique and transfer them to the built-in elements using `UVARM(...)` from Abaqus. This subroutine is called at each integration point for the built-in Abaqus element. In this implementation, I leveraged this technique following Chester et al., 2015.

4.3 Brief description of keywords in Abaqus input file

Abaqus/CAE does not support user element features as well to build the finite element model. However, if the `user element` has the same element topology and nodal connectivity as any of the built-in Abaqus elements, then the user can use Abaqus/CAE to build the model and export the model as a keyword-based Abaqus input file. To invoke the user element from the input file, the following keywords are required.

```
*User Element,Type=< >,Nodes=< >,Coordinates=< >,Properties=< >,Iproperties=< >
<list of degrees of freedom>
*Element, type=< >
<list of element connectivity>
```

The above block defines the element with an element tag, number of nodes, coordinates (or dimensions), and properties for that element. to enter the properties, the following keyword is required.

```
*uel property,elset= < >
** rho, E, nu, nInt, nPostVar
< >, < >, < >, < >
```

As previously mentioned, to visualize the result, an additional set of the built-in Abaqus element can be overlaid on the user element which shares the same element connectivity but the element number is offset by a certain value. Additionally, to visualize the element output (evaluated at the integration points) available via `UVARM` subroutine, the following keyword needs to be used

```
*User output variables
< >
```

Finally, the user has to request the output in the `*STEP` section of the input file.

```
*element output, elset=< >
uvarm
```

The number of `User output variables` is the same as `nPostVars`. The reason it needs to be defined twice is because `UEL(...)` subroutine does not have any access to post-processing-related keywords.

5 A brief review of finite elasticity (hyperelasticity)

Hyperelastic materials are a special class of elastic materials for which the stress is determined from a representative strain energy density function. Natural and synthetic rubbers and biological tissues are common examples of hyperelastic behavior. This class of materials is accompanied by geometrically nonlinear deformation. Thus, unlike infinitesimal linear elasticity, a simple kinematic description of deformation is deemed to be insufficient.

We consider a macroscopically homogeneous continuous body within the space which occupies a volume $\Omega_0(\mathbf{X}) \in \mathbb{R}^3$. This is known as the hydrated reference state of the body with material point coordinate $\mathbf{X} \in \Omega_0 \cup \partial\Omega_0$, where $\partial\Omega_0$ is the boundary of Ω_0. Material point coordinate, $\mathbf{X}$, can be mapped to the current deformed state with spatial point coordinate, $\mathbf{x}$, via a twice continuously differentiable, smooth one-to-one mapping such that $\mathbf{x} = \varphi(\mathbf{X}, t)$. Let, Ω and $\partial\Omega$ denote the body and the boundary in the current deformed state such that $\mathbf{x} \in \Omega \cup \partial\Omega$. The deformation gradient, $\mathbf{F}$, between the reference state and the current state can be defined as,

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbb{1} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \quad \Rightarrow \mathbf{F} = F_{iI}\, \mathbf{e}_i \otimes \mathbf{e}_I. \tag{5.1}$$

I should note that $\mathbf{F}$ is known as the so-called two-point tensor and its determinant represents the change in volume, *i.e.*,

$$J = \frac{dv}{dV} = \det(\mathbf{F}). \tag{5.2}$$

For incompressible materials, $J = \det(\mathbf{F}) = 1$. The relation between the undeformed area, dA, and the deformed area, da, is given by Nanson's formula as,

$$da\ \mathbf{n} = J\mathbf{F}^{-\top} dA\ \mathbf{N}. \tag{5.3}$$

where, $\mathbf{N}$ and $\mathbf{n}$ are the unit normals to the undeformed and deformed area, respectively. The deformation gradient, $\mathbf{F}$, can be decomposed into rotation and stretch deformation as follows,

$$\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}. \tag{5.4}$$

where, $\mathbf{R}$ is known as the rotation tensor with properties $\mathbf{R}^\top = \mathbf{R}^{-1}$ and $\det(\mathbf{R}) = 1$. $\mathbf{U}$ and $\mathbf{V}$ are known as the right and left stretch tensors, respectively. Since $\mathbf{F}$ is unsymmetric as well as contains rigid body rotation, it is less of a useful measure for a deformable body. Thus, as is standard in continuum mechanics,

$$\begin{aligned} \mathbf{b} = \mathbf{F}\mathbf{F}^\top = \mathbf{V}^2 \quad &\Rightarrow B_{ij} = F_{iI}F_{jI}, \\ \mathbf{C} = \mathbf{F}^\top\mathbf{F} = \mathbf{U}^2 \quad &\Rightarrow C_{IJ} = F_{iI}F_{iJ}, \end{aligned} \tag{5.5}$$

denote the left and right Cauchy-Green deformation tensors. Using spectral decomposition, I can further write,

$$\mathbf{b} = \sum_{i=1}^{3} \lambda_i^2\ \mathbf{n}_i \otimes \mathbf{n}_i, \quad \mathbf{C} = \sum_{i=1}^{3} \lambda_i^2\ \mathbf{N}_i \otimes \mathbf{N}_i, \tag{5.6}$$

where, λ_i is the eigen values of $\mathbf{b}$ and $\mathbf{C}$, and $\mathbf{N}_i$ and $\mathbf{n}_i$ are the eigen vector of $\mathbf{C}$ and $\mathbf{b}$, respectively. The same quantities can be used to write the stretch tensors, **U** and **V**, rotation tensor, **R**, and deformation gradient, **F**, as below,

$$\mathbf{U} = \sum_{i=1}^{3} \lambda_i \, \mathbf{N}_i \otimes \mathbf{N}_i, \quad \mathbf{V} = \sum_{i=1}^{3} \lambda_i \, \mathbf{n}_i \otimes \mathbf{n}_i, \quad \mathbf{R} = \sum_{i=1}^{3} \lambda_i \, \mathbf{n}_i \otimes \mathbf{N}_i, \quad \mathbf{F} = \sum_{i=1}^{3} \lambda_i \, \mathbf{n}_i \otimes \mathbf{N}_i. \tag{5.7}$$

The Cauchy-Green deformation tensors, **B** and **C**, share the same identities as,

$$\begin{aligned} I_1 &= \mathrm{tr}(\mathbf{C}) = \mathrm{tr}(\mathbf{B}) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \\ I_2 &= \frac{1}{2}[(\mathrm{tr}(\mathbf{C}))^2 - \mathrm{tr}(\mathbf{C}^2)] = \frac{1}{2}[(\mathrm{tr}(\mathbf{b}))^2 - \mathrm{tr}(\mathbf{b}^2)] = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2, \\ I_3 &= \det(\mathbf{C}) = \det(\mathbf{B}) = \lambda_1^2\lambda_2^2\lambda_3^2. \end{aligned} \tag{5.8}$$

Using the definition of the Cauchy-Green deformation tensors, the Green-Lagrange strain tensor, **E**, and the Euler-Almansi strain tensor, **e**, are defined as,

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbb{1}), \quad \mathbf{e} = \frac{1}{2}(\mathbb{1} - \mathbf{b}^{-1}). \tag{5.9}$$

The Cauchy traction, **t**, on the deformed body and the Piola-Kirchoff traction, **T**, on the undeformed body are defined as,

$$\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}, \quad \text{and} \quad \mathbf{T} = \mathbf{P} \cdot \mathbf{N}. \tag{5.10}$$

where, $\boldsymbol{\sigma}$ is the Cauchy stress, and **P** is the first Piola-Kirchhoff stress. Using Nanson's formula, these two stress tensors can be related as,

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-\top} \quad \Rightarrow \boldsymbol{\sigma} = J^{-1}\mathbf{P}\mathbf{F}^{\top} \tag{5.11}$$

The symmetry conditions for the Cauchy stress, $\boldsymbol{\sigma}$, and the first Piola-Kirchhoff stress, **P**, are given by,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\top}, \quad \text{and} \quad \mathbf{P}\mathbf{F}^{\top} = \mathbf{F}\mathbf{P}^{\top}. \tag{5.12}$$

Since, the first Piola-Kirchhoff stress, **P** is not symmetric, for mathematical and computational convenience, a symmetric second Piola-Kirchhoff stress, **S**, is defined as,

$$\mathbf{S} = \mathbf{F}^{-1}\mathbf{P} = J\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-\top}. \tag{5.13}$$

For quasi-static and static cases where mass inertia can be ignored, the governing partial differential equation for linear momentum balance is given by,

$$\begin{gathered} \mathrm{div}(\boldsymbol{\sigma}) + \rho\mathbf{B} = 0, \\ \mathrm{Div}(\mathbf{P}) + \rho_{\mathrm{R}}\mathbf{b} = 0, \quad \Rightarrow \mathrm{Div}(\mathbf{F}\mathbf{S}) + \rho_{\mathrm{R}}\mathbf{b} = 0. \end{gathered} \tag{5.14}$$

where, **b** is the body force per unit mass, and ρ and ρ_{R} are the mass density in the current and reference coordinate, respectively, and related to each other by $\rho_{\mathrm{R}} = J\rho$. The rate change of strain energy density is given by,

$$\dot{\Psi} = J\sigma : \mathbf{d} = \mathbf{S} : \dot{\mathbf{E}} = \mathbf{P} : \dot{\mathbf{F}} \tag{5.15}$$

where Ψ is the referential Helmholtz free energy density (also known as the strain energy density) for mechanical deformation.

Application of the first (balance of energy) and the second law (entropy inequality and maximum dissipation criteria) to a mechanical system in conjunction with the Coleman-Noll argument gives us,

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}}, \quad \Rightarrow \mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} = 2\frac{\partial \Psi}{\partial \mathbf{C}}, \quad \Rightarrow \boldsymbol{\sigma} = J^{-1}\mathbf{F}\left(2\frac{\partial \Psi}{\partial \mathbf{C}}\right)\mathbf{F}^{\top} \tag{5.16}$$

To ensure objectivity and material frame-indifference, the strain energy density functions are often expressed in terms of the invariants of the Cauchy-Green deformation tensors, *i.e.*, $\Psi = \hat{\Psi}_{\mathrm{R}}(I_1, I_2, I_3)$. Thus, it becomes convenient to calculate stress in terms of the invariants,

$$\begin{aligned} \boldsymbol{\sigma} &= 2J^{-1}\left[\left(\frac{\partial \Psi}{\partial I_1} + I_1\frac{\partial \Psi}{\partial I_2}\right)\mathbf{b} - \frac{\partial \Psi}{\partial I_2}\mathbf{b}^2 + I_3\frac{\partial \Psi}{\partial I_3}\mathbb{1}\right], \\ \mathbf{S} &= 2\left(\frac{\partial \Psi}{\partial I_1} + I_1\frac{\partial \Psi}{\partial I_2}\right)\mathbb{1} - \frac{\partial \Psi}{\partial I_2}\mathbf{C} + I_3\frac{\partial \Psi}{\partial I_3}\mathbf{C}^{-1} \quad \Rightarrow \mathbf{S}_i = \frac{1}{\lambda_i}\frac{\partial \Psi}{\partial \lambda_i}. \end{aligned} \tag{5.17}$$

Remark 3. Hyperelastic materials are often considered to be incompressible, $J = 1$. For purely incompressible materials, the third term in the stress expressions is substituted by Lagrange multiplier, $\mathcal{P}$, such that,

$$\mathcal{P} = I_3\frac{\partial \Psi}{\partial I_3} \tag{5.18}$$

$\mathcal{P}$ is evaluated using problem-specific boundary condition(s).

6 Finite element formulation for finite strain eleasticity

Truly incompressible or quasi-incompressible materials often demonstrate a well-known numerical issue called *volumetric locking* which requires special treatment of the formulation. First, I will derive the finite element formulation for a general nonlinear elastic case without employing any special treatment. Special numerical treatments for incompressibility and quasi-incompressibility are discussed in the later section.

For finite deformation, the governing equation for linear momentum balance be written in two different ways, either in terms of the material coordinate and the Piola-Kirchhoff stress or in terms of the spatial coordinate and the Cauchy stress. In referential (or material) configuration, the governing partial differential equation for stress equilibrium as well as the boundary conditions are given by,

$$\begin{aligned} \operatorname{Div}\mathbf{P} + \rho_R\mathbf{B} &= 0 \qquad && \text{in } \Omega_0, \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_g, \\ \mathbf{P}\cdot\mathbf{N} &= \mathbf{T} && \text{on } \Gamma_T. \end{aligned} \tag{6.1}$$

where, $\partial\Omega_0 = \Gamma_g \cup \Gamma_T$ is the boundary of the referential domain Ω_0, and Γ_g and Γ_T are two complementary subsurfaces. Similarly, in the current configuration, the governing partial differential equation for stress equilibrium as well as the boundary conditions are given by,

$$\begin{aligned} \operatorname{div}\boldsymbol{\sigma} + \rho\mathbf{B} &= 0 \qquad && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_g, \\ \boldsymbol{\sigma}\cdot\mathbf{n} &= \mathbf{t} && \text{on } \Gamma_t. \end{aligned} \tag{6.2}$$

where, $\partial\Omega = \Gamma_g \cup \Gamma_t$ is the boundary of the referential domain Ω, and Γ_g and Γ_t are two complementary subsurfaces.

Similar to the isotropic linear elastic case described in the previous section, considering the referential computational domain to be discretized finite elements, *i.e.*, $\Omega_0 = \cup\Omega_0^e$, I can write weak form for an element in the reference configuration as,

$$\mathcal{W}^e_{\mathbf{u}}(\mathbf{u}) = -\int\limits_{\Omega_0^e} \mathbf{P} : \mathrm{Grad}(\mathbf{W})\ dV + \int\limits_{\Omega_0^e} \rho_{\mathrm{R}}\mathbf{B}\cdot\mathbf{W}\ dV + \int\limits_{\Gamma_T^e} \mathbf{T}_e\cdot\mathbf{W}\ dS = 0, \tag{6.3}$$

where, $\mathbf{W}$ is the weight function that vanishes on $\Gamma_g \subset \partial\Omega_0$.

Using a similar rationale, the weak form for the element in the current configuration can be written as,

$$\mathcal{W}^e_{\mathbf{u}}(\mathbf{u}) = -\int\limits_{\Omega^e} \boldsymbol{\sigma} : \mathrm{grad}(\mathbf{w})\ dv + \int\limits_{\Omega^e} \rho\mathbf{B}\cdot\mathbf{w}\ dv + \int\limits_{\Gamma_t^e} \mathbf{T}_e\cdot\mathbf{w}\ ds = 0, \tag{6.4}$$

where, $\mathbf{w}$ is the weight function that vanishes on $\Gamma_g \subset \partial\Omega$. By definition, the weight functions, $\mathbf{W}$ and $\mathbf{w}$, and their gradients are related to each other as,

$$\begin{gathered} \mathbf{W}(\mathbf{X}) = \mathbf{w}(\mathbf{x}), \quad \Rightarrow \mathbf{W} = \mathbf{w}\circ\varphi, \\ \mathrm{Grad}(\mathbf{W}) = \mathrm{grad}(\mathbf{w})\mathbf{F}. \end{gathered} \tag{6.5}$$

Using standard vector and tensor algebraic operations, I can show that both of the above residual statements are equal.

6.1 PK-I stress-based total Lagrangian finite element formulation

The nodal coordinate, $\mathbf{X}$, and the trial solutions for the displacement, $\mathbf{u}$, and the vector weight function, $\mathbf{W}$, can be approximated using the interpolation function matrix, $\mathbf{N_u}$ as

$$\begin{aligned}\mathbf{X} = \mathbf{N_u}(\mathbf{X})\mathbf{X}_e = \sum_{a=1}^{\text{nen}} \mathbf{N}_\mathbf{u}^a(\mathbf{X})\mathbf{X}^a, \quad &\Rightarrow \quad x_i = \sum_{a=1}^{\text{nen}} N^a(\mathbf{X})x_i^a \\ \mathbf{u}(\mathbf{X}) = \mathbf{N_u}(\mathbf{X})\mathbf{u}_e = \sum_{a=1}^{\text{nen}} \mathbf{N}_\mathbf{u}^a(\mathbf{X})\mathbf{u}^a \quad &\Rightarrow \quad u_i(\mathbf{X}) = \sum_{a=1}^{\text{nen}} N^a(\mathbf{X})u_i^a, \\ \mathbf{W}(\mathbf{X}) = \mathbf{N_u}(\mathbf{X})\mathbf{W}_e = \sum_{a=1}^{\text{nen}} \mathbf{N}_\mathbf{u}^a(\mathbf{X})\mathbf{W}^a \quad &\Rightarrow \quad W_i(\mathbf{X}) = \sum_{a=1}^{\text{nen}} N^a(\mathbf{X})W_i.\end{aligned} \tag{6.6}$$

The definition and matrix form of $[\mathbf{N_u}]$ remains the same as the case of linear elasticity described in the previous section. Using that definition the deformation gradient can be calculated as,

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbb{1} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbb{1} + [\overline{\mathbf{u}_e}]\left[\frac{\partial \mathbf{N_u}}{\partial \mathbf{X}}\right], \quad \Rightarrow F_{iJ} = \delta_{ij} + \sum_{a=1}^{\text{nen}} \frac{\partial N^a}{\partial X_J} u_i^a \tag{6.7}$$

where, $\frac{\partial \mathbf{N_u}}{\partial \mathbf{X}}$ is a matrix which has a dimension of $\left[\frac{\partial \mathbf{N_u}}{\partial \mathbf{X}}\right]_{\text{nen}\times\text{nsd}}$, and $[\overline{\mathbf{u}_e}]$ has a dimension of $[\overline{\mathbf{u}_e}]_{\text{nsd}\times\text{nen}}$. I should note that, when written in index notation, N^a is not the matrix form, but rather a scalar interpolation function corresponding to the node, a.

Unlike the second Piola-Kirchhoff stress, $\mathbf{S}$, and the Cauchy stress, $\boldsymbol{\sigma}$, the first Piola-Kirchhoff stress is not symmetric ($\mathbf{P} \neq \mathbf{P}^\top$). Thus, I will have,

$$\mathbf{P} : \text{Grad}\,\mathbf{W} = \text{Grad}\,(\mathbf{N}) : \mathbf{P}\mathbf{W}_e = \mathbf{G}_\mathbf{u}^\top \mathbf{P}\mathbf{W}_e. \tag{6.8}$$

Using these definitions, I can write the element-level residual in the reference configuration as,

$$\mathbf{R}_\mathbf{u}^e(\mathbf{u}_e) = -\int\limits_{\Omega_0^e} \mathbf{G}_\mathbf{u}^\top \mathbf{P}(\mathbf{u}_e)\, dV + \int\limits_{\Omega_0^e} \rho_\text{R} \mathbf{N}_\mathbf{u}^\top \mathbf{b}\, dV + \int\limits_{\Gamma_T^e} \mathbf{N}_\mathbf{u}^\top \mathbf{T}_e\, dS = 0. \tag{6.9}$$

where, $[\mathbf{G_u}]$ is the non-symmetric gradient matrix with a dimension of $[\mathbf{G_u}]_{\text{nstress}\times\text{nsd}*\text{nen}}$. For two- and three-dimensional cases. $[\mathbf{G_u}]$ consists of repeating sub-matrices, $[\mathbf{G}_\mathbf{u}^a]$, for each node a which are defined as,

$$\mathbf{G}^a_{\mathbf{u}} = \begin{bmatrix} N^a_{,1} & 0 \\ 0 & N^a_{,1} \\ N^a_{,2} & 0 \\ 0 & N^a_{,2} \end{bmatrix}_{\text{nsd}^2 \times \text{nsd}}, \qquad \mathbf{G}^a_{\mathbf{u}} = \begin{bmatrix} N^a_{,1} & 0 & 0 \\ 0 & N^a_{,1} & 0 \\ 0 & 0 & N^a_{,1} \\ N^a_{,2} & 0 & 0 \\ 0 & N^a_{,2} & 0 \\ 0 & 0 & N^a_{,2} \\ N^a_{,3} & 0 & 0 \\ 0 & N^a_{,3} & 0 \\ 0 & 0 & N^a_{,3} \end{bmatrix}_{\text{nsd}^2 \times \text{nsd}}, \tag{6.10}$$

The non-symmetric gradient matrix, $[\mathbf{G_u}]$, will be composed of by repeating the sub-matrices, $[\mathbf{G}^a_{\mathbf{u}}]$ for each node a.

$$\mathbf{G_u} = \begin{bmatrix} \mathbf{G}^1_{\mathbf{u}} & \mathbf{G}^2_{\mathbf{u}} & \mathbf{G}^3_{\mathbf{u}} & \cdots & \cdots & \mathbf{G}^{\text{nen}}_{\mathbf{u}} \end{bmatrix}_{\text{nsd}^2 \times \text{nsd}*\text{nen}}, \tag{6.11}$$

For two- and three-dimensional cases, the first Piola-Kirchhoff stress vector is defined as follows,

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} P_{11} & P_{21} & P_{12} & P_{22} \end{bmatrix}^{\top}_{\text{nstress} \times 1}, \\ \text{and} \quad \mathbf{P} &= \begin{bmatrix} P_{11} & P_{21} & P_{31} & P_{12} & P_{22} & P_{32} & P_{13} & P_{23} & P_{33} \end{bmatrix}^{\top}_{\text{nstress} \times 1}. \end{aligned} \tag{6.12}$$

Since the first Piola-Kirchhoff stress, $\mathbf{P}$, is a nonlinear function of the displacement field, $\mathbf{u}_e$, I need to linearize the element residual first.

$$\mathbf{R}^e_{\mathbf{u}}\left(\mathbf{u}_e + \Delta \mathbf{u}_e\right) = \mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) + \frac{\partial \mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e)}{\partial \mathbf{u}} \Delta \mathbf{u} \tag{6.13}$$

To obtain the solution of a nonlinear equation, the perturbed residual, $\mathbf{R}^e_{\mathbf{u}}\left(\mathbf{u} + \Delta \mathbf{u}\right) = 0$, which gives me,

$$-\frac{\partial \mathbf{R}^e_{\mathbf{u}}}{\partial \mathbf{u}} \Delta \mathbf{u} = \mathbf{R}^e_{\mathbf{u}}. \tag{6.14}$$

The element stiffness matrix, $\mathbf{k}^e$, is then given by,

$$\begin{aligned}\mathbf{k}^e = -\frac{\partial \mathbf{R}^e_{\mathbf{u}}}{\partial \mathbf{u}_e} &= \int\limits_{\Omega^e_0} \mathbf{G}^\top_{\mathbf{u}} \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{u}_e} \; dV, \\ &= \int\limits_{\Omega^e_0} \mathbf{G}^\top_{\mathbf{u}} \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \frac{\partial \mathbf{N}_{\mathbf{u}}}{\partial \mathbf{X}} \; dV, \\ &= \int\limits_{\Omega^e_0} \mathbf{G}^\top_{\mathbf{u}} \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \mathbf{G}_{\mathbf{u}} \; dV, \\ &= \int\limits_{\Omega^e_0} \mathbf{G}^\top_{\mathbf{u}} \mathbb{A} \mathbf{G}_{\mathbf{u}} \; dV. \end{aligned} \tag{6.15}$$

In the context of nonlinear finite element analyses, $[\mathbf{k}^e]$ is often referred to as the tangent matrix. In some finite element literature, $\mathbb{A} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}}$ is defined as the first elasticity tensor. The dimension of the element stiffness (tangent) matrix, $[\mathbf{k}^e]_{\text{nsd}*\text{nen}\times\text{nsd}*\text{nen}}$.

The element residual vector, $\mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e)$, is given by,

$$\mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) = -\int\limits_{\Omega^e_0} \mathbf{G}^\top_{\mathbf{u}} \mathbf{P} \; dV + \int\limits_{\Omega^e_0} \rho_{\mathrm{R}} \mathbf{N}^\top_{\mathbf{u}} \mathbf{B} \; dV + \int\limits_{\Gamma^e_T} \mathbf{N}^\top_{\mathbf{u}} \mathbf{T}_e \; dS \tag{6.16}$$

Similar to the previous case of isotropic linear elasticity, the residual is again the difference between external and internal force. For nonlinear problems, it is convenient to divide the loading step into multiple sub-steps and solve for the displacement change, $\Delta\mathbf{u}$. Assuming an initial guess for the displacement, the procedure is iteratively repeated until convergence is achieved. This is known as the Newton-Raphson procedure for nonlinear systems of equations.

Although the PK-I stress-based total Lagrangian formulation appears to be simple, it requires evaluating the first elasticity tensor, $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}$, with a dimension of $\left(\frac{\partial \mathbf{P}}{\partial \mathbf{F}}\right)_{\text{nsd}^2\times\text{nsd}^2}$ is often not that straight-forward. Thus, to make the implementation process simple, the total Lagrangian formulation is often done in terms of the second Piola-Kirchhoff stress, $\mathbf{S}$, since it is symmetric $(\mathbf{S} = \mathbf{S}^\top)$.

Remark 4. In some finite element literature, $\mathbb{A}$ is also called material tangent tensor. However, I will reserve this terminology for the tangent that emerges in the second Piola-Kirchhoff-based formulation.

6.2 PK-II stress-based total Lagrangian finite element formulation

By substituting $\mathbf{P} = \mathbf{FS}$ into the weak form for an element in the reference configuration, for an arbitrary test function, I have,

$$\mathcal{W}^e_{\mathbf{u}}(\mathbf{u}) = -\int\limits_{\Omega^e_0} \mathbf{FS} : \mathrm{Grad}(\mathbf{W})\ dV + \int\limits_{\Omega^e_0} \rho_{\mathrm{R}} \mathbf{B} \cdot \mathbf{W}\ dV + \int\limits_{\Gamma^e_T} \mathbf{T}_e \cdot \mathbf{W}\ dS = 0. \tag{6.17}$$

Following the standard Galerkin procedure, let me consider the following discretization for the nodal coordinates of an element, $\mathbf{x}_e$, nodal displacement, $\mathbf{u}_e$, and weight functions, $\mathbf{W}$, to write the discretized version of the element residual.

$$\begin{aligned}
\mathbf{X} = \mathbf{N}_{\mathbf{u}}(\mathbf{X})\mathbf{X}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}^a_{\mathbf{u}}(\mathbf{X})\mathbf{X}^a, \quad \Rightarrow \quad x_i = \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{X}) x^a_i \\
\mathbf{u}(\mathbf{X}) = \mathbf{N}_{\mathbf{u}}(\mathbf{X})\mathbf{u}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}^a_{\mathbf{u}}(\mathbf{X})\mathbf{u}^a, \quad \Rightarrow \quad u_i(\mathbf{X}) = \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{X}) u^a_i, \\
\mathbf{W}(\mathbf{X}) = \mathbf{N}_{\mathbf{u}}(\mathbf{X})\mathbf{W}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}^a_{\mathbf{u}}(\mathbf{X})\mathbf{W}^a, \quad \Rightarrow \quad W_i(\mathbf{X}) = \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{X}) W_i.
\end{aligned} \tag{6.18}$$

With the Galerkin discretization, for arbitrary test function, $\mathbf{W}$, the element residual for the linear momentum balance equation can be written as,

$$\mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) = -\int\limits_{\Omega^e_0} \mathbf{FS} : \mathrm{Grad}(\mathbf{N}_{\mathbf{u}})\ dV + \int\limits_{\Omega^e_0} \rho_{\mathrm{R}} \mathbf{N}^{\top}_{\mathbf{u}} \mathbf{B}\ dV + \int\limits_{\Gamma^e_T} \mathbf{N}^{\top}_{\mathbf{u}} \mathbf{T}_e\ dS = 0. \tag{6.19}$$

The element residual vector, $\mathbf{R}^e_{\mathbf{u}}$ corresponds to all of the degrees of freedom (DOFs) in an element. To facilitate the calculation of the element tangent matrix, it would be wise to write the residual for each node as follows,

$$\begin{aligned}
\mathbf{R}^a_{\mathbf{u}}(\mathbf{u}_e) &= -\int\limits_{\Omega^e_0} \mathbf{FS} : \mathrm{Grad}(\mathbf{N}^a_{\mathbf{u}})\ dV + \int\limits_{\Omega^e_0} \rho_{\mathrm{R}} \mathbf{N}^{a\top}_{\mathbf{u}} \mathbf{B}\ dV + \int\limits_{\Gamma^e_T} \mathbf{N}^{a\top}_{\mathbf{u}} \mathbf{T}_e\ dS = 0, \\
\Rightarrow \mathbf{R}^a_{\mathbf{u}}(u^a_i) &= -\int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial X_I} F_{iJ} S_{JI}\ dV + \int\limits_{\Omega^e_0} \rho_{\mathrm{R}} N^a B_I\ dV + \int\limits_{\Gamma^e_T} N^a T^e_i\ dS = 0
\end{aligned} \tag{6.20}$$

Following the standard procedure of the Newton-Raphson method, the component of the element tangent matrix is given by,

$$\mathbf{k}^{ab} = -\frac{\mathbf{R}^a_{\mathbf{u}}}{\partial \mathbf{u}^b} \tag{6.21}$$

Now, the mechanical tangent, $k^{ab}_{u_i u_k}$, is given by,

$$\begin{aligned}
k^{ab}_{u_i u_k} = -\frac{\partial \mathbf{R}^a_{\mathbf{u}}}{\partial u^b_k} &= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(\frac{\partial F_{iJ}}{\partial u^b_k} S_{JI} + F_{iJ}\frac{\partial S_{JI}}{\partial u^b_k}\right)\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(\frac{\partial F_{iJ}}{\partial u^b_k} S_{JI} + F_{iJ}\frac{\partial S_{JI}}{\partial C_{KL}}\frac{\partial C_{KL}}{\partial F_{mN}}\frac{\partial F_{mN}}{\partial u^b_k}\right)\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(\frac{\partial F_{iJ}}{\partial u^b_k} S_{JI} + F_{iJ}\left(2\frac{\partial S_{JI}}{\partial C_{KL}}\right) F_{mL}\delta_{KN}\frac{\partial F_{mN}}{\partial u^b_k}\right)\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(\frac{\partial F_{iJ}}{\partial u^b_k} S_{JI} + F_{iJ}\mathbb{C}_{IJKL}F_{mL}\frac{\partial F_{mK}}{\partial u^b_k}\right)\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(\frac{\partial N^b}{\partial X_J}\delta_{ik} S_{JI} + F_{iJ}\mathbb{C}_{IJKL}F_{mL}\frac{\partial N^b}{\partial X_K}\delta_{mk}\right)\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(\frac{\partial N^b}{\partial X_J}\delta_{ik} S_{JI} + F_{iJ}\mathbb{C}_{IJKL}F_{kL}\frac{\partial N^b}{\partial X_K}\right)\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I} S_{JI}\delta_{ik}\frac{\partial N^b}{\partial X_J}\, dV + \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_I}\left(F_{iJ}\mathbb{C}_{IJKL}F_{kL}\right)\frac{\partial N^b}{\partial X_K}\, dV, \\
&= \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_J} S_{JL}\delta_{ik}\frac{\partial N^b}{\partial X_L}\, dV + \int\limits_{\Omega_0^e} \frac{\partial N^a}{\partial X_J}\left(F_{iI}\mathbb{C}_{IJKL}F_{kK}\right)\frac{\partial N^b}{\partial X_L}\, dV.
\end{aligned} \tag{6.22}$$

Here,

$$\mathbb{C}_{IJKL} = 2\frac{\partial S_{JI}}{\partial C_{KL}} = \frac{\partial S_{JI}}{\partial E_{KL}} \tag{6.23}$$

is defined as the material tangent tensor (or second elasticity tensor). Owing to the symmetric nature of the second Piola-Kirchhoff stress, $\mathbf{S} = \mathbf{S}^\top \Rightarrow S_{IJ} = S_{JI}$, and the right Cauchy-Green deformation tensor, $\mathbf{C} = \mathbf{C}^\top \Rightarrow C_{KL} = C_{LK}$, the material tangent tensor, $\mathbb{C}$ possesses minor symmetry, *i.e.*, $\mathbb{C}_{IJKL} = \mathbb{C}_{JILK}$.

I used the property of summation indices (or dummy indices) to obtain the last expression for the element tangent matrix. Additionally, the following intermediate results were used when performing algebraic calculation of $k^{ab}_{u_i u_k}$.

$$\begin{aligned}
\frac{\partial C_{KL}}{\partial F_{mN}} &= F_{mL}\delta_{KN} + F_{mK}\delta_{Lm} = 2F_{mL}\delta_{KN} \quad (\text{since } C_{KL} = C_{LK}), \\
\text{and,} \quad F_{iJ} &= \delta_{ij} + \sum_{b=1}^{\text{nen}} \frac{\partial N^b}{\partial X_J} u^b_i \qquad \Rightarrow \quad \frac{\partial F_{iJ}}{\partial u^b_k} = \frac{\partial N^b}{\partial X_J}\delta_{ik}.
\end{aligned} \tag{6.24}$$

The matrix form of the mechanical element tangent matrix is given by,

$$[\mathbf{k}^e_{\mathbf{uu}}]_{\text{nsd}*\text{nen}\times\text{nsd}*\text{nen}} = \int\limits_{\Omega_0^e} \Big(\mathbf{G}_\mathbf{u}^\top \Sigma_\mathbf{S} \mathbf{G}_\mathbf{u} + (\mathbf{B}_\mathbf{u}\boldsymbol{\Sigma}_\mathbf{F})^\top \mathbf{D}_\mathbb{C}(\mathbf{B}_\mathbf{u}\Sigma_\mathbf{F})\Big)\ dV \tag{6.25}$$

where, $\mathbf{D}_\mathbb{C}$ is the Voigt matrix form of the referential mechanical tangent $\mathbb{C}_{IJKL}$. The first term in the element tangent matrix is known as geometric stiffness and is a result of initial stresses and the second term is known as material stiffness. The second term is the standard term as appeared in the case of the linear elastic finite element formulation with $\mathbf{D}_\mathbb{C}$ corresponding to the material tangent, $\mathbb{C}$. Often in finite element literature, $\mathbf{B}_\mathbf{u}\Sigma_\mathbf{F}$ is represented as an additive decomposition of $\mathbf{B}_\mathbf{u} + \mathbf{B}_\mathbf{u}^\mathrm{L}$.

The matrix form of element residual vector, $\mathbf{R}^e_\mathbf{u}(\mathbf{u})$, is then given by,

$$\mathbf{R}^e_\mathbf{u}(\mathbf{u}_e) = -\int\limits_{\Omega_0^e} (\mathbf{B}_\mathbf{u}\Sigma_\mathbf{F})^\top \mathbf{S}\ dV + \int\limits_{\Omega_0^e} \rho_\mathrm{R} \mathbf{N}_\mathbf{u}^\top \mathbf{B}\ dV + \int\limits_{\Gamma_T^e} \mathbf{N}_\mathbf{u}^\top \mathbf{T}_e\ dS \tag{6.26}$$

Here, $\mathbf{S}$ is the second Piola-Kirchhoff stress vector in Voigt form.

$[\mathbf{G}_\mathbf{u}]$ in the geometric stiffness term is the non-symmetric gradient matrix of the shape functions which appears as,

$$\mathbf{G}_\mathbf{u} = \Big[\mathbf{G}_\mathbf{u}^1 \quad \mathbf{G}_\mathbf{u}^2 \quad \mathbf{G}_\mathbf{u}^3 \quad \cdots \quad \cdots \quad \mathbf{G}_\mathbf{u}^\text{nen}\Big]_{\text{nsd}^2\times\text{nsd}*\text{nen}}, \tag{6.27}$$

where the sub-matrix, $[\mathbf{G}_\mathbf{u}^a]$, for two-dimensional and three-dimensional cases are given by,

$$\mathbf{G}_\mathbf{u}^a = \begin{bmatrix} N^a_{,1} & 0 \\ 0 & N^a_{,1} \\ N^a_{,2} & 0 \\ 0 & N^a_{,2} \end{bmatrix}_{\text{nsd}^2\times\text{nsd}}, \qquad \mathbf{G}_\mathbf{u}^a = \begin{bmatrix} N^a_{,1} & 0 & 0 \\ 0 & N^a_{,1} & 0 \\ 0 & 0 & N^a_{,1} \\ N^a_{,2} & 0 & 0 \\ 0 & N^a_{,2} & 0 \\ 0 & 0 & N^a_{,2} \\ N^a_{,3} & 0 & 0 \\ 0 & N^a_{,3} & 0 \\ 0 & 0 & N^a_{,3} \end{bmatrix}_{\text{nsd}^2\times\text{nsd}}. \tag{6.28}$$

$[\Sigma_{\mathbf{S}}]$ is the stress matrix. For a two-dimensional and a three-dimensional case, it is given by,

$$\Sigma_{\mathbf{S}} = \begin{bmatrix} \mathbf{S}_{11} & 0 & \mathbf{S}_{12} & 0 \\ 0 & \mathbf{S}_{11} & 0 & \mathbf{S}_{12} \\ \mathbf{S}_{12} & 0 & \mathbf{S}_{22} & 0 \\ 0 & \mathbf{S}_{12} & 0 & \mathbf{S}_{22} \end{bmatrix}, \quad \Sigma_{\mathbf{S}} = \begin{bmatrix} \mathbf{S}_{11} & 0 & 0 & \mathbf{S}_{12} & 0 & 0 & \mathbf{S}_{13} & 0 & 0 \\ 0 & \mathbf{S}_{11} & 0 & 0 & \mathbf{S}_{12} & 0 & 0 & \mathbf{S}_{13} & 0 \\ 0 & 0 & \mathbf{S}_{11} & 0 & 0 & \mathbf{S}_{12} & 0 & 0 & \mathbf{S}_{13} \\ \mathbf{S}_{12} & 0 & 0 & \mathbf{S}_{22} & 0 & 0 & \mathbf{S}_{23} & 0 & 0 \\ 0 & \mathbf{S}_{12} & 0 & 0 & \mathbf{S}_{22} & 0 & 0 & \mathbf{S}_{23} & 0 \\ 0 & 0 & \mathbf{S}_{12} & 0 & 0 & \mathbf{S}_{22} & 0 & 0 & \mathbf{S}_{23} \\ \mathbf{S}_{13} & 0 & 0 & \mathbf{S}_{23} & 0 & 0 & \mathbf{S}_{33} & 0 & 0 \\ 0 & \mathbf{S}_{13} & 0 & 0 & \mathbf{S}_{23} & 0 & 0 & \mathbf{S}_{33} & 0 \\ 0 & 0 & \mathbf{S}_{13} & 0 & 0 & \mathbf{S}_{23} & 0 & 0 & \mathbf{S}_{33} \end{bmatrix}. \tag{6.29}$$

$[\Sigma_{\mathbf{S}}]$ has a dimension of $[\Sigma_{\mathbf{S}}]_{\mathrm{nsd}^2 \times \mathrm{nsd}^2}$. It is also possible to represent $\mathbf{G}^a_{\mathbf{u}}$ and consequently $\mathbf{G}_{\mathbf{u}}$ and $\Sigma_{\mathbf{S}}$ matrices in alternative matrix forms which will essentially give the same result (de Borst et al., 2012; Reddy, 2015).

$[\Sigma_{\mathbf{F}}]_{\mathrm{nsd}*\mathrm{nen} \times \mathrm{nsd}*\mathrm{nen}}$ is a square banded diagonal matrix of dimension $\mathrm{nsd} * \mathrm{nen} \times \mathrm{nsd} * \mathrm{nen}$, and for a three-dimensional case, it appears as,

$$\Sigma_{\mathbf{F}} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ & & & & \cdots & & & & & \\ & & & & \cdots & & & & & \\ & & & & \cdots & & & & & \\ & & & & \cdots & & & & & \\ & & & & \cdots & & & & & \\ & & & & \cdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & F_{11} & F_{12} & F_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & F_{21} & F_{22} & F_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & F_{31} & F_{32} & F_{33} \end{bmatrix}_{\mathrm{nsd}*\mathrm{nen} \times \mathrm{nsd}*\mathrm{nen}} \tag{6.30}$$

For a two-dimensional case, I can eliminate the rows and columns related to the third dimension and reduce the size of $\Sigma_{\mathbf{F}}$.

Use of the second Piola-Kirchhoff stress, $\mathbf{S}$, in formulating the finite element, is immediately evident. I can easily take advantage of an existing linear finite element program. By adding a few additional lines to the program, I can modify the stiffness (tangent) matrix for finite strain. Additionally, the fourth-order material tangent matrix, $\mathbb{C}$, needs to be mapped to $\mathbf{D}_{\mathbb{C}}$ using Voigt notation. Finally, I should note that the deformation-dependent body force and traction (such as pressure) will require further treatment of the element stiffness (tangent) matrix.

6.3 Cauchy stress-based updated Lagrangian finite element formulation

Recalling the weak form for the element in terms of Cauchy stress given by,

$$\mathcal{W}^e_{\mathbf{u}}(\mathbf{u}) = -\int\limits_{\Omega^e} \boldsymbol{\sigma} : \mathrm{grad}(\mathbf{w})\ dv + \int\limits_{\Omega^e} \rho\mathbf{B}\cdot\mathbf{w}\ dv + \int\limits_{\Gamma^e_t} \mathbf{T}_e\cdot\mathbf{w}\ ds = 0. \tag{6.31}$$

I can follow the same procedure as the total Lagrangian approach (employing the Galerkin discretization and Newton-Raphson method) to obtain the element tangent matrix in terms of Cauchy stress and spatial tangent tensor. The discretization will be as follows,

$$\begin{aligned}
\mathbf{x} = \mathbf{N}_{\mathbf{u}}(\mathbf{x})\mathbf{x}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}^a_{\mathbf{u}}(\mathbf{x})\mathbf{x}^a, \quad \Rightarrow \quad x_i = \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{x})x^a_i \\
\mathbf{u}(\mathbf{x}) = \mathbf{N}_{\mathbf{u}}(\mathbf{x})\mathbf{u}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}^a_{\mathbf{u}}(\mathbf{x})\mathbf{u}^a, \quad \Rightarrow \quad u_i(\mathbf{x}) = \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{x})u^a_i, \\
\mathbf{w}(\mathbf{x}) = \mathbf{N}_{\mathbf{u}}(\mathbf{x})\mathbf{w}_e &= \sum_{a=1}^{\mathrm{nen}} \mathbf{N}^a_{\mathbf{u}}(\mathbf{x})\mathbf{w}^a, \quad \Rightarrow \quad w_i(\mathbf{x}) = \sum_{a=1}^{\mathrm{nen}} N^a(\mathbf{x})w_i.
\end{aligned} \tag{6.32}$$

However, here I will follow an alternative approach in which I will push forward the element tangent matrix from the total Lagrangian approach to the current configuration to obtain the mechanical tangent, $k^{ab}_{u_iu_k}$, in updated Lagrangian form,

$$\begin{aligned}
k^{ab}_{u_iu_k} &= \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial X_J} S_{JL}\delta_{ik}\frac{\partial N^b}{\partial X_L}\ dV + \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial X_J}\left(F_{iI}\mathbb{C}_{IJKL}F_{kK}\right)\frac{\partial N^b}{\partial X_L}\ dV, \\
&= \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial x_j}\frac{\partial x_j}{\partial X_J} S_{JI}\delta_{ik}\frac{\partial N^b}{\partial x_l}\frac{\partial x_l}{X_L}\ dV + \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial x_j}\frac{\partial x_j}{X_J}\left(F_{iI}\mathbb{C}_{IJKL}F_{kK}\right)\frac{\partial N^b}{\partial x_l}\frac{\partial x_l}{X_L}\ dV, \\
&= \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial x_j}\left(J^{-1}F_{jJ}S_{JL}F_{lL}\right)\delta_{ik}\frac{\partial N^b}{\partial x_l}\ dv + \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial x_j}\left(J^{-1}F_{iI}F_{jJ}\mathbb{C}_{IJKL}F_{kK}F_{lL}\right)\frac{\partial N^b}{\partial x_l}\ dv, \\
&= \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial x_j}\sigma_{jl}\delta_{ik}\frac{\partial N^b}{\partial x_l}\ dv + \int\limits_{\Omega^e_0} \frac{\partial N^a}{\partial x_j}\mathbb{c}_{ijkl}\frac{\partial N^b}{\partial x_l}\ dv.
\end{aligned} \tag{6.33}$$

where,

$$\mathbb{c}_{ijkl} = J^{-1}F_{iI}F_{jJ}\mathbb{C}_{IJKL}F_{kK}F_{lL} \tag{6.34}$$

is defined as the spatial tangent tensor (or fourth elasticity tensor). Similar to the material tangent tensor, $\mathbb{c}$, also possesses minor symmetry.

In matrix form, the element tangent matrix can be written as,

$$\mathbf{k}^e = \int\limits_{\Omega^e} \left(\mathbf{G}^{\top}_{\mathbf{u}}\Sigma_{\boldsymbol{\sigma}}\mathbf{G}_{\mathbf{u}} + \mathbf{B}^{\top}_{\mathbf{u}}\mathbf{D}\mathbf{B}_{\mathbf{u}}\right)\ dv. \tag{6.35}$$

Here, the shape function gradient matrix, $\mathbf{G}_{\mathbf{u}}$, and the stress matrix, $\Sigma_{\boldsymbol{\sigma}}$ has the same form as the PK-II stress-based Lagrangian formulation. However, the shape function, gradient matrix, and strain-displacement matrix are now evaluated at the current coordinate system. Additionally, $\mathbf{D}$ is mapped from the fourth-order spatial tangent tensor, $\mathbb{c}$, using Voigt notation convention.

Similarly, I can push-forward the nodal residual (in index notation) in total Lagrangian approach to obtain the nodal residual (in index notation) updated Lagrangian form and consequently the element residual vector (in matrix form) in updated Lagrangian form as follows,

$$\begin{aligned} \mathbf{R}^a_{\mathbf{u}}(u_i) &= -\int\limits_{\Omega^e} \frac{\partial N^a}{\partial x_j}\sigma_{ij}\ dv + \int\limits_{\Omega^e} \rho N^a B_i\ dv + \int\limits_{\Gamma^e_t} N^a t^e_i\ ds, \\ \mathbf{R}^e_{\mathbf{u}}(\mathbf{u}_e) &= -\int\limits_{\Omega^e} \mathbf{B}^\top_{\mathbf{u}}\boldsymbol{\sigma}\ dv + \int\limits_{\Omega^e} \rho \mathbf{N}^\top_{\mathbf{u}}\mathbf{B}\ dv + \int\limits_{\Gamma^e_t} \mathbf{N}^\top_{\mathbf{u}}\mathbf{T}_e\ ds. \end{aligned} \tag{6.36}$$

Cauchy stress-based finite element formulation resembles similarity with linear elastic finite element formulation except for the geometric stiffness term. Since these matrices are mostly sparse in the case of updated Lagrangian formulation, it is computationally efficient, however, it requires updating the geometric configuration at each iteration to calculate the shape functions and their derivatives.

6.4 Relation between different elasticity tensors (or tangents)

The first Piola-Kirchhoff stress, $\mathbf{P}$, and the first elasticity tensor, $\mathbb{A}$, are defined as,

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}}, \quad \text{and} \quad \mathbb{A} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}} = \frac{\partial^2 \Psi}{\partial \mathbf{F} \partial \mathbf{F}} \tag{6.37}$$

The second Piola-Kirchhoff stress, $\mathbf{S}$, and the second elasticity tensor (also called material tangent tensor), $\mathbb{A}$, are defined as,

$$\mathbf{S} = 2\frac{\partial \Psi}{\partial \mathbf{C}}, \quad \text{and} \quad \mathbb{C} = 2\frac{\partial \mathbf{S}}{\partial \mathbf{C}} = 4\frac{\partial^2 \Psi}{\partial \mathbf{C} \partial \mathbf{C}} \tag{6.38}$$

By recalling the element tangent matrices for PK-I and PK-II stress based total Lagrangian formulation, following relation holds true between first and second elasticity tensor,

$$\mathbb{A}_{iJkL} = S_{JL}\delta_{ik} + F_{iI}\mathbb{C}_{IJKL}F_{kK} \tag{6.39}$$

As defined before, the spatial tangent (or fourth elasticity tensor) is as follows,

$$\mathbb{c}_{ijkl} = J^{-1}F_{iI}F_{jJ}\mathbb{C}_{IJKL}F_{kK}F_{lL}. \tag{6.40}$$

The relation between the first and fourth elasticity tensor (spatial tangent) is as follows,

$$\begin{gathered} \mathbb{A}_{iJkL} = JF^{-1}_{Jj}\left(\mathbb{c}_{ijkl} + \sigma_{jl}\delta_{ik}\right)F^{-1}_{Ll}, \\ \text{where, } \mathbb{a}_{ijkl} = \mathbb{c}_{ijkl} + \sigma_{jl}\delta_{ik}. \end{gathered} \tag{6.41}$$

Sometimes a so-called third elasticity tensor, $\underline{\mathbb{a}}_{ijkl}$, (Bower, 2009) is defined in terms of Kirchhoff stress, $\boldsymbol{\tau} = J\boldsymbol{\sigma}$, as follows,

$$\begin{aligned} \underline{\mathbb{a}}_{ijkl} &= \underline{\mathbb{c}}_{ijkl} + \tau_{jl}\delta_{ik}, \\ \text{where, } \underline{\mathbb{c}}_{ijkl} &= F_{iI}F_{jJ}\mathbb{C}_{IJKL}F_{kK}F_{lL} = J^{-1}\mathbb{c}_{ijkl}. \end{aligned} \tag{6.42}$$

Finally, the relation between the first and third elasticity tensor is,

$$\mathbb{A}_{iJkL} = F^{-1}_{Jj}\underline{\mathbb{a}}_{ijkl}F^{-1}_{Ll}. \tag{6.43}$$

6.5 Some notes on fourth-order tensors

A generic fourth tensor is defined by,

$$\mathbb{A} = \mathbf{B} \otimes \mathbf{C} = B_{ij}C_{kl}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l = A_{ijkl}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l \tag{6.44}$$

Two fourth-order unit tensors are defined as,

$$\begin{gathered} \mathbb{I} : \mathbf{A} = \mathbf{A} \quad \text{and} \quad \bar{\mathbb{I}} : \mathbf{A} = \mathbf{A}^\top, \\ \text{where, } \mathbb{I} = \delta_{ik}\delta_{jl}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l, \\ \text{and } \bar{\mathbb{I}} = \delta_{il}\delta_{jk}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l. \end{gathered} \tag{6.45}$$

For convenience, two additional fourth-order tensors, $\mathbb{S}$ and $\mathbb{W}$, are given by,

$$\begin{aligned} \mathbb{S} &= \frac{1}{2}\left(\mathbb{I} + \bar{\mathbb{I}}\right) \quad \Rightarrow \mathbb{S}_{ijkl} = \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right), \\ \mathbb{W} &= \frac{1}{2}\left(\mathbb{I} - \bar{\mathbb{I}}\right), \quad \Rightarrow \mathbb{W}_{ijkl} = \frac{1}{2}\left(\delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}\right) \end{aligned} \tag{6.46}$$

With these definitions, I can write,

$$\text{sym}(\mathbf{A}) = \mathbb{S} : \mathbf{A}, \quad \text{skw}(\mathbf{A}) = \mathbb{W} : \mathbf{A}. \tag{6.47}$$

Finally, another important fourth-order tensor is given by the tensor multiplication of the second-order identity tensor,

$$\mathbb{1} \otimes \mathbb{1} = \delta_{ij}\delta_{kl}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l \tag{6.48}$$

If $\mathbb{A}$ is a fourth-order tensor and $\mathbf{B}$ is a second-order tensor, then their double contraction gives,

$$\mathbf{C} = \mathbb{A} : \mathbf{B} = A_{ijkl}B_{kl}\mathbf{e}_i \otimes \mathbf{e}_j. \tag{6.49}$$

7 Constitutive model specific material and spatial tangents

7.1 Quasi-incompressible Neo-Hookean model

The strain energy density, Ψ, for a quasi-incompressible Neo-Hookean type material is given by,

$$\Psi = \frac{\mu}{2}(I_1 - 3 - 2\ln J) + \frac{\kappa}{2}(\ln J)^2. \tag{7.1}$$

where, $I_1 = \mathrm{tr}(\mathbf{C})$ and $J = \det(\mathbf{F})$, and μ and κ are the material parameters representing the shear modulus and bulk modulus. The second Piola-Kirchhoff stress, $\mathbf{S}$, is given by,

$$\mathbf{S} = \mu(\mathbb{1} - \mathbf{C}^{-1}) + \kappa(\ln J)\mathbf{C}^{-1}. \tag{7.2}$$

Let me recall some useful tensor calculus identities,

$$\begin{gathered}
\frac{\partial \mathbf{C}}{\partial \mathbf{C}} = \mathbb{1}, \quad \frac{\partial I_1}{\partial \mathbf{C}} = \mathbb{1}, \quad \frac{\partial I_2}{\partial \mathbf{C}} = 2\mathbf{C}, \quad \frac{\partial I_3}{\partial \mathbf{C}} = J^2\mathbf{C}, \quad \frac{\partial J}{\partial \mathbf{C}} = \frac{J}{2}\mathbf{C}^{-1}, \quad \frac{\partial \ln J}{\partial \mathbf{C}} = \frac{1}{2}\mathbf{C}^{-1}, \\
\left(\frac{\partial \mathbf{C}^{-1}}{\partial \mathbf{C}}\right)_{IJKL} = -\frac{1}{2}\left(C^{-1}_{IK}C^{-1}_{JL} + C^{-1}_{JK}C^{-1}_{IL}\right) = \mathbb{I}_{\mathbf{C}^{-1}}.
\end{gathered} \tag{7.3}$$

Now, the material tangent, $\mathbb{C}$, is given by,

$$\begin{aligned}
\mathbb{C} = 2\frac{\partial \mathbf{S}}{\partial \mathbf{C}} &= \kappa \mathbf{C}^{-1} \otimes \mathbf{C}^{-1} - 2(\mu - \kappa \ln J)\mathbb{I}_{\mathbf{C}^{-1}}, \\
\Rightarrow \mathbb{C}_{IJKL} &= \kappa C^{-1}_{IJ}C^{-1}_{KL} + (\mu - \kappa \ln J)\left(C^{-1}_{IK}C^{-1}_{JL} + C^{-1}_{JK}C^{-1}_{IL}\right).
\end{aligned} \tag{7.4}$$

The Cauchy stress, $\boldsymbol{\sigma}$, is given by,

$$\boldsymbol{\sigma} = \frac{\mu}{J}(\mathbf{b} - \mathbb{1}) + \frac{\kappa}{J}(\ln J)\mathbb{1}. \tag{7.5}$$

The spatial tangent, $\mathbb{c}$, is given by,

$$\begin{aligned}
\mathbb{c}_{ijkl} &= J^{-1}F_{iI}F_{jJ}F_{kK}F_{lL}\mathbb{C}_{IJKL}, \\
&= \frac{\kappa}{J}\delta_{ij}\delta_{kl} + \frac{\mu - \kappa \ln J}{J}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \\
\Rightarrow \mathbb{c} &= \frac{\kappa}{J}\mathbb{1} \otimes \mathbb{1} - \frac{2}{J}(\mu - \kappa \ln J)\mathbb{i},
\end{aligned} \tag{7.6}$$

where,

$$\begin{aligned}
\mathbb{i} &= \mathbf{F} \oslash \mathbf{F} : \mathbb{I}_{\mathbf{C}^{-1}} : \mathbf{F}^{\top} \oslash \mathbf{F}^{\top}, \\
&= \frac{-1}{2}F_{iI}F_{jJ}F_{kK}F_{lL}\left(C^{-1}_{IK}C^{-1}_{JL} + C^{-1}_{JK}C^{-1}_{IL}\right), \\
&= -\frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right).
\end{aligned} \tag{7.7}$$

7.2 Quasi-incompressible Arruda-Boyce model

The strain energy density, Ψ, for a quasi-incompressible Arruda-Boyce type model is given by,

$$\begin{gathered}
\Psi = \mu\left[\lambda_{\mathrm{L}}^2\left(\frac{\lambda_c\beta_c}{\lambda_{\mathrm{L}}} + \ln\frac{\beta_c}{\sinh\beta_c}\right) - \left(\frac{\lambda_{\mathrm{L}}}{3}\right)\ln J\right] + \frac{\kappa}{2}(\ln J)^2, \\
\text{where, } \beta_c = \mathcal{L}^{-1}\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right), \quad \text{and} \quad \lambda_c = \sqrt{\frac{I_1}{3}},
\end{gathered} \tag{7.8}$$

The second Piola-Kirchhoff stress, $\mathbf{S}$, is given by,

$$\mathbf{S} = \mu\left(\frac{\lambda_{\mathrm{L}}}{3\lambda_c}\beta_c\right)\mathbb{1} - \left[\frac{\mu\lambda_{\mathrm{L}}}{3} - \kappa(\ln J)\right]\mathbf{C}^{-1}. \tag{7.9}$$

The Cauchy stress, $\boldsymbol{\sigma}$, is given by,

$$\boldsymbol{\sigma} = \frac{\mu}{J}\left(\frac{\lambda_{\mathrm{L}}}{3\lambda_c}\beta_c\right)\mathbf{b} - \left[\frac{\mu\lambda_{\mathrm{L}}}{3J} - \frac{\kappa}{J}(\ln J)\right]\mathbb{1}. \tag{7.10}$$

I will use the following intermediate results to derive the material tangent for the Arruda-Boyce model.

$$\begin{gathered}
\frac{\partial\lambda_c}{\partial\mathbf{C}} = \frac{\partial}{\partial\mathbf{C}}\left(\sqrt{\frac{I_1}{3}}\right) = \frac{1}{2\sqrt{3}}I_1^{-1/2}\mathbb{1} = \frac{1}{6\lambda_c}\mathbb{1}, \\
\frac{\partial\lambda_c^{-1}}{\partial\mathbf{C}} = \frac{-1}{\lambda_c^2}\frac{\partial\lambda_c}{\partial\mathbf{C}} = \frac{-1}{6\lambda_c^3}\mathbb{1}, \\
\frac{\partial\beta_c}{\partial\mathbf{C}} = \frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)}\frac{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)}{\partial\mathbf{C}} = \frac{1}{2\sqrt{3}\lambda_{\mathrm{L}}}\frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)}I_1^{-1/2}\mathbb{1} = \frac{1}{6\lambda_{\mathrm{L}}\lambda_c}\frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)}\mathbb{1}, \\
\frac{\partial\ln J}{\partial\mathbf{C}} = \frac{1}{2}\mathbf{C}^{-1}
\end{gathered} \tag{7.11}$$

The material tangent, $\mathbb{C}$, is given by,

$$\begin{aligned}
\mathbb{C} &= 2\frac{\partial\mathbf{S}}{\partial\mathbf{C}} \\
&= \frac{2\mu\lambda_{\mathrm{L}}}{3}\left(\frac{1}{\lambda_c}\frac{\partial\beta_c}{\partial\mathbf{C}} + \beta_c\frac{\partial\lambda_c^{-1}}{\partial\mathbf{C}}\right)\mathbb{1} + 2\kappa\left(\frac{\partial\ln J}{\partial\mathbf{C}}\right)\mathbf{C}^{-1} - 2\left[\frac{\mu\lambda_{\mathrm{L}}}{3} - \kappa(\ln J)\right]\frac{\partial\mathbf{C}^{-1}}{\partial\mathbf{C}}, \\
&= \frac{\mu}{9\lambda_c^2}\left(\frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)} - \frac{\lambda_{\mathrm{L}}}{\lambda_c}\beta_c\right)\mathbb{1}\otimes\mathbb{1} + \kappa\mathbf{C}^{-1}\otimes\mathbf{C}^{-1} - 2\left[\frac{\mu\lambda_{\mathrm{L}}}{3} - \kappa(\ln J)\right]\mathbb{I}_{\mathbf{C}^{-1}}.
\end{aligned} \tag{7.12}$$

Using index notation, I can write,

$$\begin{aligned}\mathbb{C}_{IJKL} = &\frac{\mu}{9\lambda_c^2}\left(\frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)} - \frac{\lambda_{\mathrm{L}}}{\lambda_c}\beta_c\right)\delta_{IJ}\delta_{KL} + \kappa C^{-1}_{IJ}C^{-1}_{KL} \\ &+ \left[\mu\left(\frac{\lambda_{\mathrm{L}}}{3\lambda_c}\right) - \kappa(\ln J)\right]\left(C^{-1}_{IK}C^{-1}_{JL} + C^{-1}_{JK}C^{-1}_{IL}\right)\end{aligned} \tag{7.13}$$

The spatial tangent, $\mathbb{c}$, can be calculated as,

$$\begin{aligned}\mathbb{c}_{ijkl} &= J^{-1}F_{iI}F_{jJ}F_{kK}F_{lL}\mathbb{C}_{IJKL}, \\ &= \frac{\mu}{9J\lambda_c^2}\left[\frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)} - \frac{\lambda_{\mathrm{L}}}{\lambda_c}\beta_c\right]b_{ij}b_{kl} + \frac{\kappa}{J}\delta_{ij}\delta_{kl} + J^{-1}\left[\frac{\mu\lambda_{\mathrm{L}}}{3} - \kappa(\ln J)\right](\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \\ \Rightarrow \mathbb{c} &= \frac{\mu}{9J\lambda_c^2}\left[\frac{\partial\beta_c}{\partial\left(\frac{\lambda_c}{\lambda_{\mathrm{L}}}\right)} - \frac{\lambda_{\mathrm{L}}}{\lambda_c}\beta_c\right]\mathbf{b}\otimes\mathbf{b} + \frac{\kappa}{J}\mathbb{1}\otimes\mathbb{1} + 2J^{-1}\left[\frac{\mu\lambda_{\mathrm{L}}}{3} - \kappa(\ln J)\right]\mathbb{i}.\end{aligned} \tag{7.14}$$

In evaluating the stress and material (or spatial) tangent for the Arruda-Boyce model, the inverse Langevin function, $\mathcal{L}^{-1}(x)$ needs to be evaluated. No exact formula is available for this function, and it is often approximated using Padé approximation or Taylor series. Padé approximation often behaves poorly in the limit of $x = 1$ and Taylor series requires additional terms to be included. A simple piecewise approximation is also possible (Bergström, 1999) to evaluate the Inverse Langevin function as given below,

$$\mathcal{L}^{-1}(x) = \begin{cases} 1.31446\tan(1.58986x) + 0.91209x & \text{if } |x| < 0.84136, \\ 1/(\operatorname{sign}(x) - x) & \text{if } 0.84136 \leq |x| < 1 \end{cases} \tag{7.15}$$

It is straightforward to calculate the derivative of this piecewise approximation of the Langevin function.

$$\mathrm{D}\mathcal{L}^{-1}(x) = \begin{cases} 2.0898073756\tan^2(1.58986x) + 3.0018973756 & \text{if } |x| < 0.84136, \\ 1/(\operatorname{sign}(x) - x)^2 & \text{if } 0.84136 \leq |x| < 1 \end{cases} \tag{7.16}$$

8 Implementation of displacement element in ABAQUS/Standard

This section will list the total Lagrangian and updated Lagrangian finite element formulation procedures for finite strain hyperelastic materials. By changing the material point constitutive calculation, these procedures can be used for finite viscoelasticity and plasticity. It is rare to use PK-I stress-based total Lagrangian formulation, hence I will skip it here. However, I should note that the procedure is the same as linear elasticity except for the form of the strain-displacement matrix and constitutive relation.

Compared to the PK-II stress-based total Lagrangian formulation, I mapped the parent element to the current configuration and the reference configuration to calculate the deformation gradient, $\mathbf{F}$. However, it is possible to store the deformation gradient, $\mathbf{F}$, as a state variable and use the current configuration mapping to calculate the incremental deformation gradient. I should mention the element jacobian (J_ξ vs. j_ξ) being used during the numerical integration of the element stiffness (tangent) matrix and residual vector.

8.1 Algorithmic procedure for total and updated Lagrangian element

Procedure 2: PK-II stress-based total Lagrangian user element subroutine (UEL) implementation of hyperelasticity in Abaqus/ Standard

```
Input  : PROPS, COORDS, JELEM, JTYPE, NNODE, NDOFEL, TIME, DTIME, U,
         DU, V, A, PREDEF, JDLTYP, NDLOAD, MDLOAD, DDLMAG, ALMAG
Output: AMATRX, RHS, PNEWDT, ENERGY, SVARS

Get nInt ← PROPS and nDim, nTens ← JTYPE
Initialize: $\{\mathbf{N}^a_\mathbf{u}, \mathbf{B}^a_\mathbf{u}, \mathbf{G}^a_\mathbf{u}, \mathbf{N}_\mathbf{u}, \mathbf{B}_\mathbf{u}, \mathbf{G}_\mathbf{u}, \Sigma_\mathbf{F}, \Sigma_\mathbf{S}, \mathbf{k}^e_\mathbf{uu}, \mathbf{R}^e_\mathbf{u}\} = 0$
Get nodal displacement vectors of the element: $\mathbf{u}_e, \Delta\mathbf{u}_e$
Reshape nodal coordinate and displacement vector in matrix form: $\overline{\mathbf{x}_e}, [\overline{\mathbf{u}_e}], \overline{\Delta\mathbf{u}_e}$
Get $w_\text{int}, \boldsymbol{\xi}_\text{int}$ ← SUBROUTINE gaussQuadrtr(nDim, nNode)
for $k = 1$ to nInt do
    Get $\mathbf{N}, \frac{\partial \mathbf{N}_\mathbf{u}}{\partial \boldsymbol{\xi}}$ ← SUBROUTINE interpFunc(nDim, nNode)
    Calculate: $\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = \overline{\mathbf{x}_e}\frac{\partial \mathbf{N}_\mathbf{u}}{\partial \boldsymbol{\xi}}$        // map to the reference configuration to calculate F
    Calculate: $\frac{\partial \mathbf{N}_\mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{N}_\mathbf{u}}{\partial \boldsymbol{\xi}} \left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}}\right)^{-1}$ and $J_\xi$
    Calculate deformation gradient: $\mathbf{F} = \mathbb{1} + [\overline{\mathbf{u}_e}]\frac{\partial \mathbf{N}_\mathbf{u}}{\partial \mathbf{X}}$
    for $i = 1$ to nNode do
        Form $\mathbf{N}^a_\mathbf{u}$(nDim, NDOFEL), $\mathbf{B}^a_\mathbf{u}$(nTens, nDim), and $\mathbf{G}^a_\mathbf{u}$(nDim$^2$, nDim) matrices
        $\mathbf{N}_\mathbf{u}$(1 : nDim, nDim ∗ (i − 1) + 1 : nDim ∗ i) = $\mathbf{N}^a_\mathbf{u}$(1 : nDim, 1 : nDim)
        $\mathbf{B}_\mathbf{u}$(1 : nTens, nDim ∗ (i − 1) + 1 : nDim ∗ i) = $\mathbf{B}^a_\mathbf{u}$(1 : nTens, 1 : nDim)
        $\mathbf{G}_\mathbf{u}$(1 : nDim$^2$, nDim ∗ (i − 1) + 1 : nDim ∗ i) = $\mathbf{G}^a_\mathbf{u}$(1 : nDim$^2$, 1 : nDim)
    end
                                                    // end of nodal point loop
    Get $\mathbf{S}, \mathbf{e}, \boldsymbol{\sigma}, \mathbf{D}_\mathbb{C}$ ← SUBROUTINE umatHyperelastic(PROPS, $\mathbf{F}$)        // UMAT
     returns the constitutive response
    Form $\Sigma_\mathbf{F}$(nDim ∗ (i − 1) + 1 : nDim ∗ i, nDim ∗ (j − 1) + 1 : nDim ∗ j) = $\mathbf{F}$
     // double nested for loop over nNode with i=j
    Form $\Sigma_\mathbf{S}$(nDim ∗ (i − 1) + 1 : nDim ∗ i, nDim ∗ (j − 1) + 1 : nDim ∗ j) = $S_{ij}\mathbb{1}$
     // double nested for loop over nDim
    Calculate: $\mathbf{k}^e_\mathbf{uu} = \mathbf{k}^e_\mathbf{uu} + w_\text{int}(k)\det(J_\xi)\left(\mathbf{G}^\top_\mathbf{u}\Sigma_\mathbf{S}\mathbf{G}_\mathbf{u} + (\mathbf{B}_\mathbf{u}\Sigma_\mathbf{F})^\top\mathbf{D}_\mathbb{C}(\mathbf{B}_\mathbf{u}\Sigma_\mathbf{F})\right)$
    Calculate: $\mathbf{R}^e_\mathbf{u} = \mathbf{R}^e_\mathbf{u} - w_\text{int}(k)\det(J_\xi)\left((\mathbf{B}_\mathbf{u}\Sigma_\mathbf{F})^\top\mathbf{S} - \rho_\text{R}\mathbf{N}^\top_\mathbf{u}\mathbf{b}\right)$
end
                                                    // end of integration point loop
Assign AMATRX = $\mathbf{k}^e_\mathbf{uu}$ and RHS = $\mathbf{R}^e_\mathbf{u}$
```

Procedure 3: Cauchy stress-based updated Lagrangian user element subroutine (UEL) implementation of hyperelasticity in Abaqus/ Standard

```
Input  : PROPS, COORDS, JELEM, JTYPE, NNODE, NDOFEL, TIME, DTIME, U,
         DU, V, A, PREDEF, JDLTYP, NDLOAD, MDLOAD, DDLMAG, ALMAG
Output: AMATRX, RHS, PNEWDT, ENERGY, SVARS

Get nInt ← PROPS and nDim, nTens ← JTYPE
Initialize: $\{\mathbf{N}^a_\mathbf{u}, \mathbf{B}^a_\mathbf{u}, \mathbf{G}^a_\mathbf{u}, \mathbf{N}_\mathbf{u}, \mathbf{B}_\mathbf{u}, \mathbf{G}_\mathbf{u}, \Sigma_{\boldsymbol{\sigma}}, \mathbf{k}^e_\mathbf{uu}, \mathbf{R}^e_\mathbf{u}\} = 0$
Get nodal displacement vectors of the element: $\mathbf{u}_e, \Delta\mathbf{u}_e$
Calculate updated coordinate: $\mathbf{x}_e = \mathbf{x}_e + \mathbf{u}_e$
Reshape nodal coordinate and displacement vector in matrix form: $\overline{\mathbf{x}_e}, [\overline{\mathbf{u}_e}], \overline{\Delta\mathbf{u}_e}$
Get $w_\text{int}, \boldsymbol{\xi}_\text{int}$ ← SUBROUTINE gaussQuadrtr(nDim, nNode)
for $k = 1$ to nInt do
    Get $\mathbf{N}, \frac{\partial \mathbf{N_u}}{\partial \boldsymbol{\xi}}$ ← SUBROUTINE interpFunc(nDim, nNode)
    Calculate: $\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = \overline{\mathbf{x}_e}\frac{\partial \mathbf{N_u}}{\partial \boldsymbol{\xi}}$      // map to the reference configuration to calculate F
    Calculate: $\frac{\partial \mathbf{N_u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{N_u}}{\partial \boldsymbol{\xi}}\left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}}\right)^{-1}$ and $J_\xi$
    Calculate: $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \overline{\mathbf{x}_e}\frac{\partial \mathbf{N_u}}{\partial \boldsymbol{\xi}}$      // map to the current configuration
    Calculate: $\frac{\partial \mathbf{N_u}}{\partial \mathbf{x}} = \frac{\partial \mathbf{N_u}}{\partial \boldsymbol{\xi}}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right)^{-1}$ and $j_\xi$
    Calculate deformation gradient: $\mathbf{F} = \mathbb{1} + [\overline{\mathbf{u}_e}]\frac{\partial \mathbf{N_u}}{\partial \mathbf{X}}$
    for $i = 1$ to nNode do
        Form $\mathbf{N}^a_\mathbf{u}$(nDim, NDOFEL), $\mathbf{B}^a_\mathbf{u}$(nTens, nDim), and $\mathbf{G}^a_\mathbf{u}$(nDim$^2$, nDim) matrices
        $\mathbf{N}_\mathbf{u}$(1 : nDim, nDim ∗ (i − 1) + 1 : nDim ∗ i) = $\mathbf{N}^a_\mathbf{u}$(1 : nDim, 1 : nDim)
        $\mathbf{B}_\mathbf{u}$(1 : nTens, nDim ∗ (i − 1) + 1 : nDim ∗ i) = $\mathbf{B}^a_\mathbf{u}$(1 : nTens, 1 : nDim)
        $\mathbf{G}_\mathbf{u}$(1 : nDim$^2$, nDim ∗ (i − 1) + 1 : nDim ∗ i) = $\mathbf{G}^a_\mathbf{u}$(1 : nDim$^2$, 1 : nDim)
    end
                                                            // end of nodal point loop
    Get $\mathbf{e}, \boldsymbol{\sigma}, \mathbf{D}$ ← SUBROUTINE umatHyperelastic(PROPS, $\mathbf{F}$)      // UMAT returns
     the constitutive response
    Form $\Sigma_{\boldsymbol{\sigma}}$(nDim ∗ (i − 1) + 1 : nDim ∗ i, nDim ∗ (j − 1) + 1 : nDim ∗ j) = $\sigma_{ij}\mathbb{1}$
     // double nested for loop over nDim
    Calculate: $\mathbf{k}^e_\mathbf{uu} = \mathbf{k}^e_\mathbf{uu} + w_\text{int}(k)\det(j_\xi)\left(\mathbf{G}^\top_\mathbf{u}\Sigma_\mathbf{S}\mathbf{G}_\mathbf{u} + \mathbf{B}^\top_\mathbf{u}\mathbf{D}\mathbf{B}_\mathbf{u}\right)$
    Calculate: $\mathbf{R}^e_\mathbf{u} = \mathbf{R}^e_\mathbf{u} - w_\text{int}(k)\det(j_\xi)\left(\mathbf{B}^\top_\mathbf{u}\boldsymbol{\sigma} - \rho\mathbf{N}^\top_\mathbf{u}\mathbf{b}\right)$
end
                                                        // end of integration point loop
Assign AMATRX = $\mathbf{k}^e_\mathbf{uu}$ and RHS = $\mathbf{R}^e_\mathbf{u}$
```

Similar to the previous case, the constitutive behavior of the material can be changed by including a different subroutine corresponding to finite strain plasticity or finite strain viscoelasticity, etc. instead of using `umatHyperelastic`. Similar to the previous case, traction and body force can be applied using built-in Abaqus elements.

8.2 Brief description of the user element subroutine and input files in GitHub repository

The user element subroutine, `uel_nlmech_pk2.for`, available in the GitHub repository implemented PK-II stress-based total Lagrangian element as an example. The structure of the subroutine shares a large similarity with the isotropic linear elastic subroutine described in the previous section. Similar to the previous case, a total of 8 types of elements are available; 4 of them are 3D continuum solid elements and 4 of them are 2D plane-strain elements with reduced and full-integration schemes. The same technique of overlaying built-in Abaqus elements on the user element has been used for post-processing the output results.

In this implementation, two different options were made available to evaluate the hyperelastic constitutive response; (a) Neo-Hookean material and (b) Arruda-Boyce material. Henceforth, the properties to be specified in the input file are different than the linear elastic case. The following table lists the properties that need to be specified in the input file:

Name of the property	Property location	UEL variable
Shear modulus, μ	`props(1)`	`G`
Bulk modulus, κ	`props(2)`	`kappa`
Locking stretch λ_{L}	`props(3)`	`lam_L`
No of integration points	`jprops(1)`	`nInt`
Type of material	`jprops(2)`	`matFlag`
No of post-processing variables	`jprops(3)`	`nPostVars`

Table 3: List of properties for the hyperelastic element.

Users should note the following scenarios when using the code:

1. `matFlag=1` corresponds to Neo-Hookean material and in this case, the user needs to specify `lam_L=0.0` whereas `matFlag=2` corresponds to Arruda-Boyce material which has a finite locking stretch value that user needs to specify.
2. Since no special formulation has been implemented to alleviate volumetric locking, for nearly incompressible hyperelastic materials, it is advised to use higher order elements with $\kappa \leq 100\mu$, otherwise in specific circumstances, it could result in severe volumetric locking and consequently wrong results.

Bibliographic references

Bergström, B. (1999). *Large strain time-dependent behavior of elastomeric materials* [Doctoral dissertation, Massachusetts Institute of Technology].

Bower, A. F. (2009, October 5). *Applied Mechanics of Solids* (0th ed.). CRC Press. https://doi.org/10.1201/9781439802489

Chester, S. A., Di Leo, C. V., & Anand, L. (2015). A finite element implementation of a coupled diffusion-deformation theory for elastomeric gels. *International Journal of Solids and Structures*, *52*, 1–18. https://doi.org/10.1016/j.ijsolstr.2014.08.015

de Borst, R., Crisfield, M. A., Remmers, J. J. C., & Verhoosel, C. V. (2012). *Non-linear Finite Element Analysis of Solids and Structures* (2nd ed.). John Wiley & Sons, Inc.

Reddy, J. N. (2015). *An Introduction to Nonlinear Finite Element Analysis*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780198525295.001.0001

Zienkiewicz, O. C., Taylor, R. L., & Fox, D. (2014). *The Finite Element Method for Solid and Structural Mechanics* (7th ed). Elsevier/Butterworth-Heinemann.
OCLC: ocn858011693.

Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2013). *The Finite Element Method: Its Basis and Fundamentals* (Seventh edition). Elsevier, Butterworth-Heinemann.
OCLC: ocn852808496.