

# Semi-Analytical Approach to Element-Level Integration for the Solid Nonlinear Finite Element

Eli Hanukah

Mechanical Engineering Department, Braude College of Engineering, P.O. Box 78,  
Karmiel, 2161002, Israel

## Abstract

The Nonlinear Finite Element Method is a widely used numerical technique to solve engineering problems associated with nonlinear solid continua. Often, real-life engineering problems require immense computational resources. Thus, raising the computational efficiency of the method is a highly desirable goal. A nonlinear global system solver requires constant reevaluation of the element-level internal forces together with the nonlinear stiffness matrix. For this purpose, commercial software employs the standard numerical integration based on the quadrature method (e.g., Gauss points). Consequently, computational complexity increases linearly with the number of integration points. Therefore, the derivation of alternative, highly efficient integration approaches is essential, and this is the main goal of our study.

Herein, we propose new element-level integration formulae of the internal forces and of the nonlinear tangent stiffness matrix. Our formulae admit the “full” order integration requirement. Moreover, we demonstrate analytically that the computational cost of the proposed schemes is roughly equivalent to One-Point quadrature, irrespective of the element type (e.g., tetrahedral, hexahedral, wedge, etc.,) and irrespective of the element order (e.g., 8-node brick, 20-node brick, etc.,). Code implementation of our formula follows a rather familiar, standard, manner. However, prior to code implementation, it requires sets of coefficients/special weights to be pre-computed or adopted from the literature. Undoubtedly, the proposed integrators require significantly more coefficients than the standard numerical integration.

Recently, the Semi-Analytical approach has been employed to produce highly efficient case-specific integrators for the mass matrices. Here, we generalize and expand this approach to all element-level integrals. To this end, the integrands are decomposed into two multiplicative parts. The first part includes kinetic and kinematic functions which admit “full” integration criteria for one sampling point (e.g., the centroid). While the second part consists of mesh-independent and displacement-independent polynomial functions. We integrate the above polynomials analytically, to derive coefficient sets. Those coefficients are incorporated in the resulting scheme’s subroutine. In other words, we take advantage of the mathematical structure of the integrand to produce a highly efficient yet case-specific integration formula (e.g., our internal forces rule can’t be used for the mass of stiffness matrices and vice versa). Importantly, code implementation doesn’t involve either meta-programming or a computer algebra system for explicit (closed-form) code generation.

**Key words:** Finite Element Technology; Semi-Analytical Integration; Full Integration; Nonlinear Tangent Stiffness Matrix; Closed-Form; Element-Level Integration;

## 1 Introduction

Nonlinear Finite Element Method (NFEM) is the most widely used numerical tool to solve engineering problems associated with nonlinear continua. Often, practical problems require extensive computational resources. Therefore, raising the computational efficiency of the overall procedure is highly desirable.

Fundamentally, NFEM requires a nonlinear solver (e.g., Newton/modified Newton-Raphson, etc.) to achieve a global system solution. Such a solver is required to update the internal and external forces together with the nonlinear stiffness matrix every computational step i.e., the solution procedure goes back and forth between element-level computations and global system iterations. All element-level evaluations, such as the mass matrix, internal forces, and the nonlinear tangent stiffness matrix, require integration in the element domain. Most commonly, those are performed using the Standard (ST) numerical integration, based on a specific quadrature scheme. Thus, computational complexity grows linearly with the number of integration points, which increases dramatically the element formation cost. The goal of our study is to radically improve element-level integration efficiency for the solid NFEM. We achieve this goal by proposing new integration formulae, based on the Semi-Analytical approach. The proposed method is sufficiently general i.e., it addresses all isoparametric solid elements of general order, including both, geometrical and material non-linearity.

It is known that increasing the efficiency of element-level integration yields significant CPU time savings [1]. Noticeable savings can be achieved by "Closed-Form" code generation using Computer Algebra Systems CAS [28, 1]. For the linear elasticity and regular shape elements (straight-sided), a combination of analytical integration, by means of CAS, together with closed-form code generation, yields dramatic improvements in CPU time (e.g., [29, 30]). In addition, special structural formulations based on Cosserat Point Element (CPE) theory, technically do not require integration (e.g., [22, 21]). Yet, to the best of our knowledge, for a fully nonlinear FE, there is no alternative **systematic** approach to dramatically over-perform the standard numerical integration. Namely, there is no alternative method, which can be consistently applied to all solid elements having a general constitutive behavior i.e., whether 1D, 2D, or 3D elements, arbitrary shape & order (e.g., quadrilateral, triangular, bricks, tetrahedral, wedges, etc.). This is the gap we attempt to close in the present contribution.

Recently, the Semi-Analytical (SA) approach has been employed to derive highly efficient integration formulas for the consistent/lumped mass matrices [15, 17, 18]. We generalize this method and specialize it for the nonlinear tangent stiffness matrix and the internal forces. The resulting integration formulae are case-specific, i.e., a unique formula for the mass matrix, an additional unique formula for the internal forces, and the last one for the nonlinear tangent stiffness matrix. Yet, the resulting schemes are sufficiently general, in the sense that the same formula addresses all isoparametric solid elements of an arbitrary order (whether a wedge, tetrahedral, hexahedral, etc.,).

SA integration presented here, admits the "full" order integration requirement. Specifically, it is exact for a regular shape and a constant strain element, while approximation for all the other cases. Importantly, computation-wise, the SA formulae are roughly equivalent to the Standard (ST) One-Point quadrature, regardless of the

element order! Thus, the higher the element order, the more pronounced the efficiency advantages of our schemes.

Throughout the study, we explicitly consider the isoparametric element and a Total Lagrangian formulation. Both of those choices are widely used and accepted. Therefore, they serve as a good framework to demonstrate the main ideas of the contribution. However, the SA method is not restricted by those choices. Incorporation of the Updated Lagrangian formulation follows the same guidelines.

Subroutine implementation of the proposed integration rules, **do not** require special programming strategies such as meta-programming or the use of CAS for closed-form code generation. The resulting formulae take a familiar form, their code implementation follows conventional tactics.

Organization of the paper is as follows: section (2) recalls the essential well-established element-level kinematic and kinetic relations and formulae, for one to be able to follow through later derivations. Then, subsection (2.1) recalls the Standard (ST) numerical integration approach. Next, subsection (2.2) recalls and discusses the "full" order vs. the "reduced" order integration concepts. Section (3) begins with the general concept of the Semi-Analytical approach; it presents the guidelines that we follow to derive the proposed formulae. In addition, computational equivalence between the full integration SA scheme and the ST One-Point quadrature is briefly discussed. Subsections (3.1),(3.2) and (3.3) contain derivation of the SA integration schemes for the mass matrix, internal forces and the nonlinear tangent stiffness matrix respectively. Finally, section (4) discusses, highlights, and summarises the important points and takeaways of the study.

## 2 Background

In this section we recall the basics of the well-established Total Lagrangian (TL) formulation. In particular, we end up with expressions for the consistent mass matrix, internal forces and the nonlinear tangent stiffness matrix. Those require numerical integration. Subsection (2.1) emphasizes the standard quadrature-based numerical integration. Then, subsection (2.2) recalls the "full" and "reduced" order integration approaches. Even though we confine ourselves to isoparametric TL formulation throughout the section and the study, the proposed SA approach is not limited by those constraints. We keep this section extremely brief since all of the below are well-established relations that can be found in at least several great books e.g. [42].

Within the framework of the isoparametric concept, the displacement field  $\mathbf{u}$  is approximated in the element domain by  $\mathbf{u} \approx \mathbf{u}_e = \sum_{I=1}^n N_I(\xi)\mathbf{u}_I(t)$  where  $n$  denotes the number of element nodes,  $\xi = (\xi, \eta, \zeta)$  stand for the local coordinates,  $N_I(\xi)$ , ( $I = 1, \dots, n$ ) are the shape functions,  $\mathbf{u}_I$  represent the nodal displacements, while  $\Omega_e$  refer to the initial element configuration. Position vectors in the initial and current configurations are denoted by  $\mathbf{X}$  and  $\mathbf{x}$ . They are interpolated in the element domain using the same shape functions

$$\mathbf{X}_e = \sum_{I=1}^n N_I(\xi) \mathbf{X}_I \quad , \quad \mathbf{x}_e = \sum_{I=1}^n N_I(\xi) \mathbf{x}_I(t) \quad (1)$$

Where  $\mathbf{X}_I$  and  $\mathbf{x}_I$  stand for respective nodal positions. Shape functions  $N_I$  usually take a rather simple form, most frequently a polynomial interpolation. For example, for a standard twenty-node hexahedral element, the first shape function is given by  $N_1(\xi) = -\frac{1}{8}((1-\xi)(1-\eta)(1-\zeta)(2+\xi+\eta+\zeta))$ . For later convenience, we emphasize that the shape functions are mesh & displacement-independent  $N_I \neq N_I(\mathbf{X}_J)$ ,  $N_I \neq N_I(\mathbf{u}_J(t))$ . One can easily analytically differentiate those functions with respect to local coordinates. Also, those functions and their derivatives can easily be analytically integrated in the parent element domain (parametric space) to result in a number. Further down the line, we'll use those numbers as coefficients.

It is important to distinguish between differentiation of  $N_I(\xi)$  with respect to local and global coordinates

$$\nabla_{\xi} N_I(\xi) = \begin{Bmatrix} N_{I;\xi}(\xi) \\ N_{I;\eta}(\xi) \\ N_{I;\zeta}(\xi) \end{Bmatrix} = \begin{Bmatrix} N_{I;1} \\ N_{I;2} \\ N_{I;3} \end{Bmatrix}, \quad (I = 1, \dots, n) \quad (2)$$

where semicolon is used to denote partial differentiation with respect to local coordinates e.g.  $N_{I;3} = \frac{\partial N_I}{\partial \zeta}$ . Similarly, gradients with respect to initial global coordinates  $X_1, X_2, X_3$  or current global coordinates  $x_1, x_2, x_3$  are given by

$$\nabla_{\mathbf{X}} N_I = \begin{Bmatrix} N_{I,1} \\ N_{I,2} \\ N_{I,3} \end{Bmatrix}, \quad \nabla_{\mathbf{x}} N_I = \begin{Bmatrix} N_{I,1} \\ N_{I,2} \\ N_{I,3} \end{Bmatrix}, \quad (I = 1, \dots, n) \quad (3)$$

where comma represents partial differentiation with respect to global coordinates. The transformation between the gradients of different configurations is given by

$$\nabla_{\mathbf{X}} N_I = \mathbf{J}_e^{-T} \nabla_{\xi} N_I, \quad \nabla_{\mathbf{x}} N_I = \mathbf{j}_e^{-T} \nabla_{\xi} N_I, \quad (I = 1, \dots, n) \quad (4)$$

where with the help of (1), entries of JACOBI matrices  $\mathbf{J}_e = \text{Grad}_{\xi} \mathbf{X}_e = \frac{\partial \mathbf{X}_e}{\partial \xi}$  and  $\mathbf{j}_e = \text{Grad}_{\xi} \mathbf{x}_e = \frac{\partial \mathbf{x}_e}{\partial \xi}$  take the next form

$$[\mathbf{J}_e] = \begin{bmatrix} X_{1;\xi} & X_{1;\eta} & X_{1;\zeta} \\ X_{2;\xi} & X_{2;\eta} & X_{2;\zeta} \\ X_{3;\xi} & X_{3;\eta} & X_{3;\zeta} \end{bmatrix}, \quad X_{m;k} = \sum_{I=1}^n N_{I;k} X_{m,I}, \quad (m, k = 1, 2, 3) \quad (5)$$

$$[\mathbf{j}_e] = \begin{bmatrix} x_{1;\xi} & x_{1;\eta} & x_{1;\zeta} \\ x_{2;\xi} & x_{2;\eta} & x_{2;\zeta} \\ x_{3;\xi} & x_{3;\eta} & x_{3;\zeta} \end{bmatrix}, \quad x_{m;k} = \sum_{I=1}^n N_{I;k} x_{m,I}, \quad (m, k = 1, 2, 3) \quad (6)$$

Relations (4) require inverse transform of the above  $3 \times 3$  matrices (5),(6). Inverse of an arbitrary real square  $3 \times 3$  matrix  $A$  is given by  $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$ , where determinant and adjoint matrix follow simple rule

$$\begin{aligned} \det(A) &= a_{1,1}a_{2,2}a_{3,3} - a_{1,1}a_{2,3}a_{3,2} - a_{1,2}a_{2,1}a_{3,3} + \\ &\quad a_{1,2}a_{2,3}a_{3,1} + a_{1,3}a_{2,1}a_{3,2} - a_{1,3}a_{2,2}a_{3,1} \\ \text{Adj}(A) &= \begin{bmatrix} a_{2,2}a_{3,3} - a_{2,3}a_{3,2} & -a_{1,2}a_{3,3} + a_{1,3}a_{3,2} & a_{1,2}a_{2,3} - a_{1,3}a_{2,2} \\ -a_{2,1}a_{3,3} + a_{2,3}a_{3,1} & a_{1,1}a_{3,3} - a_{1,3}a_{3,1} & -a_{1,1}a_{2,3} + a_{1,3}a_{2,1} \\ a_{2,1}a_{3,2} - a_{2,2}a_{3,1} & -a_{1,1}a_{3,2} + a_{1,2}a_{3,1} & a_{1,1}a_{2,2} - a_{1,2}a_{2,1} \end{bmatrix} \end{aligned} \quad (7)$$

thus we can write

$$[\mathbf{J}_e^{-1}] = \frac{1}{J_e} \text{Adj}(\mathbf{J}_e) \quad , \quad [\mathbf{J}_e^{-T}] = \frac{1}{J_e} \text{Adj}^T(\mathbf{J}_e) \quad , \quad [\mathbf{j}_e^{-T}] = \frac{1}{j_e} \text{Adj}^T(\mathbf{j}_e) \quad (8)$$

$$J_e = \det(\mathbf{J}_e) \quad , \quad j_e = \det(\mathbf{j}_e) \quad (9)$$

In terms of (5),(6),(8),(9) deformation gradient  $\mathbf{F}_e$  is given by

$$\mathbf{F}_e = \mathbf{j}_e \mathbf{J}_e^{-1} \quad , \quad J_e^F = \det(\mathbf{F}_e) = \frac{j_e}{J_e} \quad (10)$$

The nonlinear strain-displacement  $6 \times 3$  matrix  $\mathbf{B}_I$ , ( $I = 1, \dots, n$ ) is given by

$$\mathbf{B}_I = \begin{bmatrix} F_{1,1}N_{I,1} & F_{2,1}N_{I,1} & F_{3,1}N_{I,1} \\ F_{1,2}N_{I,2} & F_{2,2}N_{I,2} & F_{3,2}N_{I,2} \\ F_{1,3}N_{I,3} & F_{2,3}N_{I,3} & F_{3,3}N_{I,3} \\ F_{1,1}N_{I,2} + F_{1,2}N_{I,1} & F_{2,1}N_{I,2} + F_{2,2}N_{I,1} & F_{3,1}N_{I,2} + F_{3,2}N_{I,1} \\ F_{1,2}N_{I,3} + F_{1,3}N_{I,2} & F_{2,2}N_{I,3} + F_{2,3}N_{I,2} & F_{3,2}N_{I,3} + F_{3,3}N_{I,2} \\ F_{1,1}N_{I,3} + F_{1,3}N_{I,1} & F_{2,1}N_{I,3} + F_{2,3}N_{I,1} & F_{3,1}N_{I,3} + F_{3,3}N_{I,1} \end{bmatrix} \quad (11)$$

Importantly, in the above, shape functions are differentiated with respect to global  $X_1, X_2, X_3$  coordinates. Next,  $G_{IJ}$ , ( $I, J = 1, \dots, n$ ) matrix is recalled

$$G_{IJ} = (\nabla_{\mathbf{X}} N_I)^T \mathbf{S}_e (\nabla_{\mathbf{X}} N_J) \quad (12)$$

$$G_{IJ} = [N_{I,1}, N_{I,2}, N_{I,3}]^T \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{Bmatrix} N_{J,1} \\ N_{J,2} \\ N_{J,3} \end{Bmatrix}, \quad (13)$$

where  $\mathbf{S}_e$  is the Second Piola-Kirchhoff stress tensor. Following (12) it becomes apparent that differentiation in the above (13) is done with respect to global coordinates  $X_1, X_2, X_3$ .

Finally, we recall the expressions, for efficient integration of which, the present work exists i.e. - the consistent mass matrix, internal forces, and the nonlinear tangent stiffness matrix. External forces are omitted since their SA treatment follows identical guidelines proposed below and is quite immediate.

The consistent mass matrix is given by

$$M_{IJ} = \int_{\Omega_e} \rho_0 N_I N_J d\Omega_e = \int_{\Omega_e} \rho_0 N_I(\boldsymbol{\xi}) N_J(\boldsymbol{\xi}) J_e(\boldsymbol{\xi}; \mathbf{X}_K) d\Omega_e \quad (14)$$

$$(I, J, K = 1, \dots, n)$$

where  $\rho_0$  denotes the initial density. Since the mass matrix does not depend on the displacement, it needs to be evaluated only once. In that sense, even though its efficient evaluation is desirable, it is hardly the "bottleneck" of the overall procedure. On the contrary, a nonlinear solver (such as Newton/Newton-Raphson) requires the internal forces and the nonlinear stiffness matrix to be evaluated every step of the global system solution. This, in turn, makes their integration efficiency improvements extremely sought after!

The internal forces  $\mathbf{R}_I$  are given by

$$\mathbf{R}_I = \int_{\Omega_e} \mathbf{B}_I^T \mathbf{S}_e d\Omega_e \quad , \quad (I = 1, \dots, n) \quad (15)$$

whereas here,  $\mathbf{S}_e$  is the Voigt notation of the Second Piola-Kirchhoff stress tensor  $\mathbf{S}$  that is  $\mathbf{S}_e = \{S_{11}, S_{22}, S_{33}, S_{12}, S_{23}, S_{13}\}^T$ .

The nonlinear tangent stiffness matrix  $\mathbf{K}_{T\,IJ}$  is given by

$$\mathbf{K}_{T\,IJ} = \int_{\Omega_e} [G_{IJ}\mathbf{I} + \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J] d\Omega_e , \quad (I, J = 1, \dots, n) \quad (16)$$

where for 3D problem,  $\mathbf{I}$  is a  $3 \times 3$  identity matrix, strain-displacement matrix  $\mathbf{B}_I$  is given by (11),  $G_{IJ}$  is detailed in (13),  $\mathbf{D}$  stands for incremental constitutive tensor (which is merely a  $6 \times 6$  Voigt notation of incremental constitutive tensor  $\mathbb{C} = \frac{\partial \mathbf{S}}{\partial \mathbf{C}}$ , with  $\mathbf{C}$  being the right Cauchy-Green deformation tensor  $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ ). The above (16) contains two additive terms. The first, associated with  $G_{IJ}$  is sometimes referred to as the "initial stress" or the "geometric" stiffness matrix. While the other can be referenced as the "material" stiffness matrix. Both of them contribute to the non-linearity of  $\mathbf{K}_{T\,IJ}$  which makes analytical integration impossible.

## 2.1 Standard (ST) Numerical Integration

The integrand of the internal forces (15) and of the tangent stiffness matrix (16) is a nonlinear function, which generally speaking, can not be integrated analytically in the element domain  $\Omega_e$ . Thus, an approximate integration approach must be adopted. Most commonly, commercial FE packages utilize the Gauss quadrature or a similar integration rule based on the use of integration points (e.g. [4, 37]).

Transformation of the volume element admits  $dv = \det(\mathbf{F})dV$ . Hence, integration in the element domain admits  $d\Omega_e = J_e d_{\square}$  where  $d_{\square}$  is a parameter space. Consider an arbitrary tensor/vector valued function  $\mathbf{A}(\boldsymbol{\xi}; \mathbf{X}_I; \mathbf{u}_J; c_0, \dots, c_k)$ ,  $(I, J = 1, \dots, n)$ , where  $c_0, \dots, c_k$  is a set of material properties. Then, an integral of  $\mathbf{A}$  in the element domain is given by

$$\bar{\mathbf{A}} = \int_{\Omega_e} \mathbf{A} d\Omega_e = \int_{\Omega_{\square}} \mathbf{A}(\boldsymbol{\xi}; \mathbf{X}_I; \mathbf{u}_J; c_0, \dots, c_k) J_e(\boldsymbol{\xi}; \mathbf{X}_K) d\Box = \int_{\Omega_{\square}} \mathbf{A} J_e d\xi d\eta d\zeta \quad (I, J, K = 1, \dots, n)$$

In particular, for the hexahedral element it becomes  $\int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{A} J_e d\xi d\eta d\zeta$ . Following the standard numerical integration, the approximation of the integral takes the form

$$\bar{\mathbf{A}} \stackrel{ST}{\approx} \sum_{q=1}^{n_{ip}} W_q \mathbf{A}(\boldsymbol{\xi}_q; \mathbf{X}_I; \mathbf{u}_J; c_0, \dots, c_k) J_e(\boldsymbol{\xi}_q; \mathbf{X}_K) , \quad (I, J, K = 1, \dots, n) \quad (17)$$

where  $n_{ip}$  is the number of integration points,  $W_q$ ,  $(q = 1, \dots, n_{ip})$  denotes weights associated with each point -  $\boldsymbol{\xi}_q = (\xi_q, \eta_q, \zeta_q)$ . It follows immediately from the above (17), that the higher the  $n_{ip}$  the more additive terms are included in the summation. Therefore - computational complexity grows linearly with  $n_{ip}$ .

Each quadrature has its order (rank), i.e. scheme of order  $n$  will integrate polynomial functions of order  $n$  exactly. Typically, the higher the quadrature order the higher  $n_{ip}$ , the higher the computational cost of element formation. As a general

consideration, it is important to lower the element formation cost by lowering  $n_{ip}$  as much as possible. The most computationally inexpensive integration rule, using which the global system converges (performs sufficiently well), is desirable.

## 2.2 "Full" Order Integration

Following the previous subsection (2.1), an important question arises - What order of quadrature should one use to evaluate (14),(15),(16)? The higher the scheme's order the higher the  $n_{ip}$  and therefore the element formation cost. The lower the order and  $n_{ip}$ , the less accurate and rank-deficient matrices become. Thus, two important integration strategies have been discovered/formulated and widely adopted — "Full" order and "Reduced" order integration. **i)** Full order integration requires quadrature order to match the integrand order provided **regular shape** and **constant strain** element. **ii)** Reduced order integration happens when quadrature order is lower than full integration. While full integration could be associated with its own problems such as over-stiffness, the reduced integration results in even worse outcomes such as rank deficient element stiffness matrix, which leads to instability, spurious singular mode, zero-energy, or hourglass mode.

As a general guideline, the reduced integration is not recommended for a higher order elements ((44)pp.62). Broadly speaking, full integration is the default choice, a convergence necessity. There is no need to go higher than full order due to computational costs and still present over-stiffness problems. Also, reduced order integration, if used for linear elements, usually employs a special stabilization technique. Herein, we'll propose a full integration formula at a significantly lower (than reduced) computational cost.

Now it is important to recall and detail the practical mathematical meaning of "regular shape" & "constant strain" element. **i)** Regular shape element has several equivalent statements such as - constant metric element or constant Jacobian element i.e.  $J_e(\xi, \mathbf{X}_I) = const = J_e(\mathbf{X}_I)$ , ( $I = 1, \dots, n$ ), where  $\mathbf{X}_I$  is the initial nodal positions (mesh). Therefore, given a regular shape element, one can sample the Jacobian at the centroid and result in an exact value  $J_e|_{reg. shape} = J_e|_{centroid} = J_e^0$ . Visually, for an 8-node brick element, this is true as long as an element is parallelepiped (however skewed). For a 20-node hexahedral element parallelepiped shape, all middle nodes must be in the middle of the edges. In fact, for all higher-order elements, the edges are straight lines while middle nodes are in the middle of the edges. For a wedge element triangular bases are equal and parallel to each other etc. In addition, it immediately follows that  $\mathbf{J}_e$  given by (5) is also coordinate independent  $\mathbf{J}_e|_{reg. shape} = \mathbf{J}_e^0$ , together with its inverse  $\mathbf{J}_e^{-1}|_{reg. shape} = \frac{1}{J_e^0} \text{Adj}(\mathbf{J}_e^0) = \mathbf{J}_e^{0^{-1}}$ , (see (8)).

**ii)** Constant strain implies that the strain tensor is coordinate independent, i.e. the deformation gradient (10) is also constant  $\mathbf{F}_e = \mathbf{F}_e|_{centroid} = \mathbf{F}_e^0$ . With the help of (10) and (6) it must hold that  $\mathbf{j}_e = \mathbf{j}_e^0$ ,  $\mathbf{F}_e^0 = \mathbf{j}_e^0 \mathbf{J}_e^{0^{-1}}$ . In addition, for a constant strain element, the stress tensor must remain constant so that  $\mathbf{S}_e|_{\mathbf{F}_e=const} = \mathbf{S}_e|_{centroid} = \mathbf{S}_e^0$ .

In summary, an integration formula that admits full order integration, is exact pro-

vided the below conditions and is an approximation for all the other cases

$$\begin{aligned} J_e|_{\text{regular shape}} = \text{const} &= J_e^0 \quad , \quad \mathbf{F}_e|_{\text{constant strain}} = \mathbf{const} = \mathbf{F}_e|_{\text{centroid}} = \mathbf{F}_e^0 \\ \mathbf{J}_e|_{J_e=\text{const}} &= \mathbf{J}_e^0 \quad , \quad \mathbf{J}_e^{-1}|_{J_e=\text{const}} = \mathbf{J}_e^{0-1} = \frac{1}{J_e^0} \text{Adj}(\mathbf{J}_e^0) \quad , \quad \mathbf{j}_e|_{\mathbf{F}_e=\text{const}} = \mathbf{j}_e^0 \quad (18) \\ \mathbf{S}_e|_{\mathbf{F}_e=\text{const}} &= \mathbf{S}_e|_{\text{centroid}} = \mathbf{S}_e^0 \end{aligned}$$

here and throughout the text, the upper index "0" denotes evaluation at the centroid.

### 3 Semi-Analytical (SA) Integration

In this section, we detail the general concept behind our SA approach to element-level full order integration. Then, we discuss the computational cost of the resulting formula. Next, in subsections (3.1),(3.2),(3.3) we specialize it to the element mass matrix, the internal forces and the nonlinear tangent stiffness matrix respectively.

Consider an integral

$$\bar{I} = \int_{\Omega_e} I d\Omega_e = \int_{\Omega_{\square}} I(\boldsymbol{\xi}; \mathbf{X}_I; \mathbf{u}_J; c_0, \dots, c_k) J_e(\boldsymbol{\xi}; \mathbf{X}_K) d\Omega_{\square}, \quad (I, J, K = 1, \dots, n) \quad (19)$$

An integrand  $I$  can be either a tensor/vector/scalar-valued function, which depends on the local coordinates  $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ , initial nodal positions (mesh)  $\mathbf{X}_I$ , nodal displacements  $\mathbf{u}_J(t)$ , a set of material properties  $c_0, \dots, c_k$ . Most commonly, integrand  $I$  is a nonlinear function with respect to local coordinates  $\boldsymbol{\xi}$  as well as to other arguments. Our goal is to derive an approximate formula for the above integral (19), such that it will be exact for (18), i.e., admit full integration (see subsec. (2.2)), while approximate for all the other cases. To this end, we

i) we decompose the integrand of the above (19) to two multiplicative parts

$$\begin{aligned} I &= I_1(\boldsymbol{\xi}; \mathbf{X}_I; \mathbf{u}_J; c_0, \dots, c_k) I_2(\boldsymbol{\xi}) \\ \bar{I} &= \int_{\Omega_{\square}} J_e(\boldsymbol{\xi}; \mathbf{X}_K) I_1(\boldsymbol{\xi}; \mathbf{X}_I; \mathbf{u}_J; c_0, \dots, c_k) I_2(\boldsymbol{\xi}) d\Omega_{\square}, \quad (I, J, K = 1, \dots, n) \quad (20) \end{aligned}$$

Given regular shape & constant strain element (18), the first part  $J_e I_1$  can be exactly evaluated using ONE sampling point  $(J_e I_1)|_{\text{centroid}} = J_e^0 I_1^0$ . The second part  $I_2(\boldsymbol{\xi})$  consists of mesh-independent and displacement-independent polynomials with respect to local coordinates  $\boldsymbol{\xi}$ . Importantly, even though the integrand  $I$  is a nonlinear function with respect to local coordinates  $\boldsymbol{\xi}$ , provided (18), the above decomposition (20) is always possible, as is revealed by careful examination of the mathematical structure of the element-level computations.

ii) Thus, SA approximation of (20) takes the form

$$\bar{I} \stackrel{SA}{\approx} \bar{I}|_{\substack{J_e=\text{const} \\ \mathbf{F}_e=\text{const}}} = J_e^0 I_1^0 \int_{\Omega_{\square}} I_2(\boldsymbol{\xi}) d\Omega_{\square}$$

Thus, the integration rule becomes

$$\bar{I}^{SA} = J_e^0 I_1^0 I_2^* \quad , \quad I_2^* = \int_{\Omega_{\square}} I_2(\boldsymbol{\xi}) d\Omega_{\square} \quad (21)$$

Importantly,  $I_2^*$  is merely a number/coefficient/weight. Prior to code implementation of the above formula (21),  $I_2^*$  have to be adopted from the literature (if exists) or to be pre-computed. One needs to pre-compute it only once since it is both mesh-independent and deformation-independent. Practically,  $I_2$  is either a shape function/shape function multiplication (e.g.  $N_I(\xi)$ ,  $N_I N_J$ ), or its derivative/derivatives multiplication (e.g.  $N_{I;2} = \frac{\partial N_I}{\partial \eta}$  or  $N_{I;2} N_{J;3}$ ). In some sense,  $I_2^*$  resembles weights  $W_q$ , ( $q = 1, \dots, n_{ip}$ ), from the standard approach (17). However, weights  $W_q$  are associated with a specific integration point in a particular quadrature; weights are not related to the integral being evaluated. To the contrary,  $I_2^*$  makes sense only for a case-specific integrator (21), since  $I_2(\xi)$  is a specific polynomial function which appears in the integrand decomposition (20).

Why do we loosely state that **roughly** speaking, the SA approach is computationally equivalent to the One-Point quadrature? Originally, we are evaluating (19), which can be tackled by numerical integration (17) in a straightforward manner

$$\bar{I} = \int_{\Omega_{\square}} J_e I d\square \stackrel{ST}{\approx} W_1 J_e^1 I^1 + W_2 J_e^2 I^2 + \dots + W_{n_{ip}} J_e^{n_{ip}} I^{n_{ip}} \quad (22)$$

where  $J_e^q = J_e|_{\xi_q}$ ,  $I^q = I|_{\xi_q}$ , ( $q = 1, \dots, n_{ip}$ ). Alternatively, one can apply the ST scheme to the decomposed form (20) of the same integral (19) i.e.

$$\bar{I} = \int_{\Omega_{\square}} J_e I_1 I_2 d\square \stackrel{ST}{\approx} W_1 J_e^1 I_1^1 I_2^1 + W_2 J_e^2 I_1^2 I_2^2 + \dots + W_{n_{ip}} J_e^{n_{ip}} I_1^{n_{ip}} I_2^{n_{ip}} \quad (23)$$

where  $I_1^q = I_1|_{\xi_q}$ ,  $I_2^q = I_2|_{\xi_q}$ , ( $q = 1, \dots, n_{ip}$ ). Clearly, both (22) and (23) produce the same result, however, strictly speaking, computationally they are not **exactly** the same. Our full integration SA formula (21), i.e.,  $\bar{I}^{SA} = J_e^0 I_1^0 I_2^*$  is equivalent to one additive term in the above (23). In summary, our formula (21) is equivalent to One-Point of (23), while each additive term in (23) is **roughly** equivalent to each term in the original (22).

What order quadrature should one use to evaluate (19) using the full integration ST approach? Following (20) and (18), integral (19) is given by

$$\bar{I} \Big|_{\substack{J_e=const \\ F_e=const}} = \int_{\Omega_{\square}} J_e^0 I_1^0 I_2(\xi) d\square$$

Apparently, the exact evaluation of the above, requires quadrature order to match the order of  $I_2(\xi)$ . For  $I_2$  of the 2<sup>nd</sup> order vs. the 4<sup>th</sup> order, the number of integration points  $n_{ip}$  can vary quite significantly, especially in 3D. While for the SA approach, the order of  $I_2$  is irrelevant to the computational complexity of (21), since  $I_2^*$  is still one number. Thus, the higher the order of  $I_2$  the higher the computational efficiency advantage offered by the proposed method.

Regarding code implementation. Often, Semi-Analytical work in the field is associated with meta-programming or extremely lengthy (yet efficient) codes generated by CAS (closed-form approach). None of this is necessary for the proposed method. To illustrate, following the present case, the integration formula (21) is similar to the one-point of the ST quadrature use (22). The very same is true regarding subroutine implementation.

Why the proposed formula (21) to evaluate (19) is so efficient with respect to (23)? For instance, for  $I_2(\xi)$  of a 6<sup>th</sup> order in 3D, one might need dozens of integration points (depending on the domain), while the proposed (21) is still equivalent to only one. Quadrature (cubature), such as Gauss points, is well established, the most effective, and the most general integration scheme, reaching far beyond mechanics and NFEM. They are over-performed due to the highly specialized nature of the proposed formulas. Our approach exists in the element-level NFEM domain. It relies strongly on the existence of the decomposition (20), which in turn has meaning due to the existence of the full integration conditions (18), and the coefficients are pre-computed using analytical integration, which is possible once again due to unique nature of the integrand. It is important to emphasize, that SA formulas are case-specific integrators. We combine an essential convergence criteria (18) together with a special mathematical structure of the (element-level) integrals to analytically decompose the integrand (20). Even though, the resulting formula (21) somehow resembles the standard approach (22),(23), one can not apply this formula to an integral with a different structure. For instance, our formula for the nonlinear tangent stiffness matrix (30) can not be used for the mass matrix or the internal forces and vice versa. In contrast, the standard approach quadrature can tackle a general problem. In addition, while the standard approach is versatile in terms of rank i.e., one can apply a variety of distinct quadrature to evaluate a problem (19) with different levels of accuracy, our SA formula admits the full integration criteria, no less & no more. Luckily, it fits the demand of a specific problem being tackled in the study.

### 3.1 Consistent Mass Matrix

Semi-Analytical integration for the consistent and lumped (diagonal) mass matrices has been thoroughly discussed in [15, 17, 18]. However, herein we omit the higher-order models and include only the basic One-Point full integration model, which keeps this contribution complete and coherent. The decomposition (20) of (14) comes quite naturally

$$(N_I(\xi) N_J(\xi)) (\rho_0 J_e(\xi; \mathbf{X}_K)) , \quad (I, J, K = 1, \dots, n)$$

where the first part  $\rho_0 J_e$  is coordinate and mesh-dependent; to admit the full integration (18), it is sufficient to evaluate it at the centroid  $\rho_0^0 J_e^0$ . The second part  $N_I N_J$  is a mesh-independent & displacement-independent polynomial function, the analytical integration of which results in a set of numerical coefficients/weights

$$M_{IJ}^{SA} = M_{IJ}^0 \rho_0^0 J_e^0 , \quad M_{IJ}^0 = \int_{\Omega_\square} N_I N_J d\square , \quad (I, J = 1, \dots, n) \quad (24)$$

where a set of coefficients  $M_{IJ}^0$  is either adopted from the literature or pre-computed once prior to subroutine implementation. Each element type results in a different set. For example, shape functions  $N_I$  of a 10-node tetrahedral element, are quadratic with respect to the local coordinates (e.g.,  $N_7 = 4\eta(1 - \xi - \eta - \zeta)$ ). Coefficients  $M_{IJ}^0$  are pre-calculated following the rule  $M_{78}^0 = \int_0^{+1} \int_0^{1-\zeta} \int_0^{1-\zeta-\eta} N_7 N_8 d\xi d\eta d\zeta = \frac{2}{315}$ , see [11].

### 3.2 Internal Forces

Internal forces are given by (15). With the help of (11) & (4), strain-displacement matrix  $\mathbf{B}_I$ , ( $I = 1, \dots, n$ ) is decomposed such that

$$\mathbf{B}_I = \sum_{r=1}^3 \hat{\mathbf{B}}_r N_{I;r} \quad , \quad (I = 1, \dots, n)$$

$$\hat{\mathbf{B}}_r = \frac{1}{J_e} \begin{bmatrix} F_{1,1}a_{r,1} & F_{2,1}a_{r,1} & F_{3,1}a_{r,1} \\ F_{1,2}a_{r,2} & F_{2,2}a_{r,2} & F_{3,2}a_{r,2} \\ F_{1,3}a_{r,3} & F_{2,3}a_{r,3} & F_{3,3}a_{r,3} \\ F_{1,1}a_{r,2} + F_{1,2}a_{r,1} & F_{2,1}a_{r,2} + F_{2,2}a_{r,1} & F_{3,1}a_{r,2} + F_{3,2}a_{r,1} \\ F_{1,2}a_{r,3} + F_{1,3}a_{r,2} & F_{2,2}a_{r,3} + F_{2,3}a_{r,2} & F_{3,2}a_{r,3} + F_{3,3}a_{r,2} \\ F_{1,1}a_{r,3} + F_{1,3}a_{r,1} & F_{2,1}a_{r,3} + F_{2,3}a_{r,1} & F_{3,1}a_{r,3} + F_{3,3}a_{r,1} \end{bmatrix} \quad (25)$$

where  $a_{rn}$ , ( $r, n = 1, 2, 3$ ) are entries of the adjoint matrix  $a_{rn} = [Adj(\mathbf{J}_e)]_{rn}$ , see (7). Using the above (25), the integrand of (15) takes the form  $\sum_{r=1}^3 J_e \mathbf{B}_r^T \mathbf{S}_e N_{I;r}(\xi)$ , which admits (20) given the (18). Then, the SA approximation to (15) takes the next form

$$\mathbf{R}_I^{SA} = \sum_{r=1}^3 J_e^0 \mathbf{B}_r^{0T} \mathbf{S}_e^0 N_{I;r}^* = J_e^0 (\mathbf{B}_1^{0T} N_{I1}^* + \mathbf{B}_2^{0T} N_{I2}^* + \mathbf{B}_3^{0T} N_{I3}^*) \mathbf{S}_e^0$$

$$N_{I;r}^* = \int_{\Omega_\square} N_{I;r}(\xi) d\square \quad , \quad (I = 1, \dots, n, r = 1, 2, 3)$$

where  $N_{I;r}^*$  is a set of coefficients/weights having real numerical values. Those coefficients have to be one-time pre-computed prior to code implementation e.g. for a 20-node brick element  $N_{16;3} = \frac{\partial N_{16}}{\partial \zeta} = \frac{(\xi-1)(\eta^2-1)}{4}$ ,  $N_{16;3}^* = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} N_{16;3} d\xi d\eta d\zeta = \frac{4}{3}$ .

### 3.3 Nonlinear Tangent Stiffness Matrix

The nonlinear tangent stiffness matrix is given by (16). The integrand consists of two additive terms, the first often referred to as the "geometric" or "initial stress" stiffness matrix, while the other as the "material" stiffness matrix. Following the SA approach described in Sec. (3), we decompose both additive terms.  $G_{IJ}$  is given by (12), with the help of (4) & (13),  $G_{IJ}$ , ( $I, J = 1, \dots, n$ ) is rewritten as

$$G_{IJ} = \left[ \frac{1}{J_e^2} [Adj(\mathbf{J}_e)] [\mathbf{S}_e] [Adj^T(\mathbf{J}_e)] \right] \bullet \left[ (\nabla_\xi N_I) (\nabla_\xi N_J)^T \right] \quad (27)$$

where bold dot between two matrices of the same rank ( $3 \times 3$ ) denotes scalar product (double contraction) i.e.  $C = [A] \bullet [B]$ ,  $C = \sum_{m,n=1}^3 [A]_{mn} [B]_{mn}$ . Hence, SA approximation takes the next form

$$\int_{\Omega_e} G_{IJ} d\Omega_e \stackrel{SA}{\approx} G_{IJ}^{SA} = \frac{1}{J_e^0} [Adj(\mathbf{J}_e^0)] [\mathbf{S}_e^0] [Adj^T(\mathbf{J}_e^0)] \bullet [\mathbf{K}_{IJ}^0]$$

$$[\mathbf{K}_{IJ}^0]_{mn} = \int_{\Omega_\square} N_{I;m} N_{J;n} d\square \quad , \quad (I, J = 1, \dots, n, m, n = 1, 2, 3) \quad (28)$$

Once again we accentuate, coefficient matrices  $\mathbf{K}_{IJ}^0 = \int_{\Omega_{\square}} (\nabla_{\xi} N_I) (\nabla_{\xi} N_J)^T d\square$  are either one time pre-computed prior to in code implementation or adopted from the literature (if exist). Coefficient matrices are element-specific due to the element-specific nature of the shape functions and integration domain.

Decomposition of the "material" part, uses representation (25). Accordingly, the integrand becomes

$$\mathbf{B}_I^T \mathbf{D} \mathbf{B}_J = \sum_{r,t=1}^3 \mathbf{B}_r^T \mathbf{D} \mathbf{B}_t (N_{I;r} N_{J;t}) , \quad (I, J = 1, \dots, n) \quad (29)$$

thus, full-order SA approximation takes the form

$$\int_{\Omega_e} \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J d\Omega_e \stackrel{SA}{\approx} \sum_{r,t=1}^3 J_e^0 \mathbf{B}_r^{0\ T} \mathbf{D}^0 \mathbf{B}_t^0 [\mathbf{K}_{IJ}^0]_{rt}$$

Finally, the overall SA approximation of fully integrated (16) is given by

$$\mathbf{K}_{T\,IJ}^{SA} = G_{IJ}^{SA} \mathbf{I} + \sum_{r,t=1}^3 J_e^0 \mathbf{B}_r^{0\ T} \mathbf{D}^0 \mathbf{B}_t^0 [\mathbf{K}_{IJ}^0]_{rt} , \quad (I, J = 1, \dots, n) \quad (30)$$

where  $G_{IJ}^{SA}$  is given by (28).

## 4 Discussion and conclusions

Currently, commercial FEA software packages, mostly use the standard numerical integration for element mass matrix, internal forces, and the nonlinear tangent stiffness matrix. For example, ANSYS adopts a 14-point quadrature to evaluate the mass and stiffness matrices of the 20-node hexahedral element, while ABAQUS adopts a 27-point quadrature for full and 8-point scheme for reduced integration. Clearly, the more integration points the higher the element formation cost. In this study, we specialize, for the first time, in the recently suggested Semi-Analytical approach to derive a new, full-order, integration formulae for element mass & stiffness matrices plus the internal forces.

Our formulas admit "full" order integration i.e. they are exact for a regular shape & constant strain element (18), regardless of an element shape or an order.

Code wise, the suggested schemes (24),(26),(30), are implemented in a conventional manner. No meta-programming is required; no explicit code generation (closed-form) by means of the Computer Algebra Systems (CAS) is necessary.

The present contribution is neither restricted by a specific element or an element order nor by a particular constitutive behavior. All solid elements can be implemented, whether 2D or 3D, whether wedge, triangular, bricks, quadrilateral, tetrahedral, etc., whether linearly elastic, hyper-elastic, or other constitutive behavior.

In terms of efficiency, Our formulae are roughly equivalent to One-Point quadrature, regardless of an element order! Subsection (2.1) elaborates on this loose statement.

Throughout the study, we adopt the nonlinear Total Lagrangian formulation. However, specialization to the Updated Lagrangian formulation follows exactly the same

guidelines developed here. Also, external forces are omitted from the study, yet, given the guidelines, their inclusion follows a straightforward manner. Finally, we've explicitly considered the isoparametric elements, due to their widest commercial acceptance. Once again, the SA approach can be specialized for other element concepts if necessary.

The higher the element order, that is, the higher the shape functions order, the more pronounced the efficiency advantage of the proposed method, see section (3).

The consistent mass matrix (14) is deformation independent. Consequently, it is sufficient to calculate it only once (24). Therefore, even though its efficient evaluation is desirable, it is hardly the "bottleneck" of the global nonlinear system solution. In contrast, the nonlinear stiffness matrix is to be evaluated every step of the global system solution, which makes its computational efficiency improvements extremely in demand.

SA formulas (24),(26),(30) contain coefficient sets. Numerical values of those coefficients can be obtained via literature (e.g.[17],[18]) or by one-time pre-computation. Clearly, those coefficients are element-specific, due to the element-specific nature of the shape functions and the integration domain. To illustrate, coefficients for a 10-node tetrahedral element differ from coefficients for a 15-node wedge element.

SA integration rules (24),(26),(30) are case-specific integrators — mass matrix formula (24) cannot be applied to the stiffness matrix (16) or to the internal forces (15) and vice versa.

## References

- [1] M. S. Alnæs and K.-A. Mardal. On the efficiency of symbolic computations combined with code generation for finite element methods. *ACM Transactions on Mathematical Software (TOMS)*, 37(1):1–26, 2010.
- [2] M. Cerrolaza and J. Osorio. Relations among stiffness coefficients of hexahedral 8-noded finite elements: A simple and efficient way to reduce the integration time. *Finite elements in analysis and design*, 55:1–6, 2012.
- [3] R. Cools. An encyclopaedia of cubature formulas. *Journal of Complexity*, 19(3):445–453, 2003. ISSN 0885-064X. doi: [https://doi.org/10.1016/S0885-064X\(03\)00011-6](https://doi.org/10.1016/S0885-064X(03)00011-6). URL <https://www.sciencedirect.com/science/article/pii/S0885064X03000116>. Oberwolfach Special Issue.
- [4] R. Cools. An encyclopaedia of cubature formulas. *Journal of complexity*, 19(3):445–453, 2003.
- [5] C. A. Felippa. A compendium of fem integration formulas for symbolic work. *Engineering computations*, 21(8):867–890, 2004.
- [6] J. A. González, R. Kolman, S. Cho, C. A. Felippa, and K. Park. Inverse mass matrix via the method of localized lagrange multipliers. *International Journal for Numerical Methods in Engineering*, 113(2):277–295, 2018.
- [7] A. Gravouil, T. Elguedj, and H. Maigre. An explicit dynamics extended finite element method. part 2: Element-by-element stable-explicit/explicit dynamic scheme. *Computer Methods in Applied Mechanics and Engineering*, 198(30-32):2318–2328, 2009.

- [8] E. Hanukah. Development of a higher order closed-form model for isotropic hyperelastic cylindrical body, including small vibration problem. *arXiv preprint arXiv:1312.0083*, 2013.
- [9] E. Hanukah. Higher order closed-form model for isotropic hyperelastic spherical shell (3d solid). *arXiv preprint arXiv:1401.0204*, 2013.
- [10] E. Hanukah. A new closed-form model for isotropic elastic sphere including new solutions for the free vibrations problem. *arXiv preprint arXiv:1311.0741*, 2013.
- [11] E. Hanukah. Consistent mass matrix of ten nodes tetrahedral element based on analytical integration. *arXiv preprint arXiv:1411.1341*, 2014.
- [12] E. Hanukah. Exact integration scheme for six-node wedge element mass matrix. *arXiv preprint arXiv:1412.6588*, 2014.
- [13] E. Hanukah. Mass matrix integration scheme for fifteen-node wedge element. *arXiv preprint arXiv:1501.00175*, 2014.
- [14] E. Hanukah. Semi-analytical mass matrix for 8-node brick element. *arXiv preprint arXiv:1410.3195*, 2014.
- [15] E. Hanukah. On semi-analytical integration specified for mass matrix of finite elements. *arXiv preprint arXiv:1506.02168*, 2015.
- [16] E. Hanukah and S. Givli. Free vibration of an isotropic elastic skewed parallelepiped—a closed form study. *European Journal of Mechanics-A/Solids*, 53: 131–142, 2015.
- [17] E. Hanukah and S. Givli. A new approach to reduce the number of integration points in mass-matrix computations. *Finite Elements in Analysis and Design*, 116: 21–31, 2016.
- [18] E. Hanukah and S. Givli. Improving mass matrix and inverse mass matrix computations of hexahedral elements. *Finite Elements in Analysis and Design*, 144: 1–14, 2018.
- [19] E. Hanukah and B. Goldshtain. A structural theory for a 3d isotropic linear-elastic finite body. *arXiv preprint arXiv:1207.6767*, 2012.
- [20] H. E. Hinnant. A fast method of numerical quadrature for p-version finite element matrices. *International Journal for Numerical Methods in Engineering*, 37 (21):3723–3750, 1994.
- [21] M. Jabareen. Keynote: The cosserat point element (cpe)—a new approach for finite element formulation. In *The 7th International Conference on Computational Methods (ICCM2016)*, 2016.
- [22] M. Jabareen, E. Hanukah, and M. Rubin. A ten node tetrahedral cosserat point element (cpe) for nonlinear isotropic elastic materials. *Computational Mechanics*, 52:257–285, 2013.
- [23] S. Johnson, S. Sivakumar, and D. Nagarajan. A 13-point sampling point scheme for 20-node brick elements. *International Journal of Computational Materials Science and Engineering*, 10(04):2150019, 2021.
- [24] S. Johnson, S. Sivakumar, and D. Nagarajan. A nine-point sampling point scheme for twenty node brick element. In *Journal of Physics: Conference Series*, volume 1913, page 012143. IOP Publishing, 2021.
- [25] A. Korncoff and S. J. Fenves. Symbolic generation of finite element stiffness matrices. *Computers & structures*, 10(1-2):119–124, 1979.

- [26] K. L. Lawrence, R. V. Nambiar, and B. Bergmann. Closed form stiffness matrices and error estimators for plane hierachic triangular elements. *International journal for numerical methods in engineering*, 31(5):879–894, 1991.
- [27] I. Lozada, J. Osorio, D. Griffiths, and M. Cerrolaza. Semi-analytical integration of the 8-node plane element stiffness matrix using symbolic computation. *Numerical Methods for Partial Differential Equations: An International Journal*, 22(2):296–316, 2006.
- [28] I. Lozada, D. Griffiths, and M. Cerrolaza. Semi-analytical integration of the elastic stiffness matrix of an axisymmetric eight-noded finite element. *Numerical Methods for Partial Differential Equations*, 26(6):1624–1635, 2010.
- [29] S. E. McCaslin, P. S. Shiakolas, B. H. Dennis, and K. L. Lawrence. A new approach to obtaining closed-form solutions for higher order tetrahedral finite elements using modern computer algebra systems. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 54792, pages 225–231, 2011.
- [30] S. E. McCaslin, P. S. Shiakolas, B. H. Dennis, and K. L. Lawrence. Closed-form stiffness matrices for higher order tetrahedral finite elements. *Advances in Engineering Software*, 44(1):75–79, 2012.
- [31] M. A. Moetakef, K. L. Lawrence, S. P. Joshi, and P. S. Shiakolas. Closed-form expressions for higher order electroelastic tetrahedral elements. *AIAA Journal*, 33(1):136–142, 1995.
- [32] J. C. Osorio, M. Cerrolaza, and M. Perez. Optimising the stiffness matrix integration of n-noded 3d finite elements. *International Journal of Computational Science and Engineering*, 16(2):173–180, 2018.
- [33] M. Pavlović. Symbolic computation in structural engineering. *Computers & structures*, 81(22-23):2121–2136, 2003.
- [34] P. Shiakolas, R. Nambiar, K. Lawrence, and W. Rogers. Closed-form stiffness matrices for the linear strain and quadratic strain tetrahedron finite elements. *Computers & structures*, 45(2):237–242, 1992.
- [35] S. Shun-Cheng and D. Zhu-Ping. Iterative method using consistent mass matrix in axisymmetrical finite element analysis of hypervelocity impact. *International journal of impact engineering*, 21(10):817–825, 1998.
- [36] H. Songyang, W. Dongdong, W. Zhenyu, and L. Zhiwei. Precise mid-node lumped mass matrices for 3d 20-node hexahedral and 10-node tetrahedral finite elements. *Chinese Journal of Theoretical and Applied Mechanics*, pages 1–13, 2023.
- [37] A. H. Stroud. *Approximate calculation of multiple integrals*. Prentice Hall, 1971.
- [38] A. Tkachuk and M. Bischoff. Direct and sparse construction of consistent inverse mass matrices: general variational formulation and application to selective mass scaling. *International Journal for Numerical Methods in Engineering*, 101(6):435–469, 2015.
- [39] L. Videla, M. Cerrolaza, and N. Aparicio. Explicit integration of the stiffness matrix of a four-noded-plane-elasticity finite element. *Communications in numerical methods in engineering*, 12(11):731–743, 1996.
- [40] L. Videla, T. Baloa, D. Griffiths, and M. Cerrolaza. Exact integration of the stiffness matrix of an 8-node plane elastic finite element by symbolic computation. *Numerical Methods for Partial Differential Equations: An International Journal*,

- 24(1):249–261, 2008.
- [41] B. Wang, D. Babu, R. Nambiar, and K. Lawrence. Shape design sensitivity analysis using closed-form stiffness matrix for hierachic triangular elements. *Computers & structures*, 43(1):69–75, 1992.
  - [42] P. Wriggers. *Nonlinear finite element methods*. Springer Science & Business Media, 2008.
  - [43] S. R. Wu and W. Qiu. Nonlinear transient dynamic analysis by explicit finite element with iterative consistent mass matrix. *Communications in numerical methods in engineering*, 25(3):201–217, 2009.
  - [44] O. Zienkiewicz, R. Taylor, and D. Fox. *The finite element method for solid and structural mechanics* elsevier, 2005.