Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.081240

ARTICLE

Multibody Dynamics Using Quasi Energy and Momentum Conservative Algorithm Implemented with a Novel Four-Node Co-Rotational Quadrilateral Shell Element

Zhongxue Li[1,*], Yu Peng[1], Jin Xu[2], Loc Vu-Quoc[3] and Bassam A. Izzuddin[4]

[1]Department of Civil Engineering, Zhejiang University, Hangzhou, China
[2]Simulation & Design Development, Altair Engineering Inc., Shanghai, China
[3]Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[4]Department of Civil and Environmental Engineering, Imperial College London, London, UK

*Corresponding Author: Zhongxue Li. Email: lizx19993@zju.edu.cn

Received: 26 February 2026; Accepted: 30 June 2026; Published: 27 July 2026

ABSTRACT: This paper proposes a computational theory for multibody dynamics based on a novel co-rotational formulation of a four-node quadrilateral shell element, designed to address nonlinear dynamic problems in flexible multibody systems undergoing arbitrarily large displacements and rotations. To circumvent the numerical inefficiency caused by the asymmetric tangent stiffness matrix in conventional co-rotational approaches, an incrementally additive vectorial rotational variable is introduced, which ensures the symmetry of the element's tangent stiffness matrix in both global and local coordinate systems. The adoption of this vectorial rotational variable substantially improves computational efficiency and numerical stability. Hamilton's principle is employed to derive the system's dynamic equilibrium differential equation. For time integration, a generalized midpoint scheme coupled with a quasi energy–momentum conservative algorithm is used. This scheme ensures nearly exact conservation of the system's total energy, linear momentum, and angular momentum in long-term simulations. The accuracy and stability of the proposed methodology are verified through four benchmark cases: an L-shaped plate, a ruler-shaped plate, a hemispherical shell with an 18° top opening, and a non-smooth shell-three intersecting plates. The results indicate that after the external loads are removed, the system's total energy, total linear momentum, and total angular momentum are conserved with near-exact precision. The numerical results show excellent agreement with reference solutions from prior publications, and the computational process remains robustly stable.

KEYWORDS: Co-rotational approach; four-node quadrilateral shell element; energy-momentum conservative algorithm; multibody dynamics

1 Introduction

The co-rotational framework, first introduced in [1–3], is widely recognized as an efficient strategy for geometrically nonlinear analysis, particularly for problems involving large displacements and rotations [4–9]. A significant advancement in this field was the Element-Independent Co-rotational (EICR) method [10,11], which facilitates the formulation of new co-rotational elements while preserving the vector transformation matrix. This advancement broadens the applicability of the approach to various element types. Crisfield and Moita [12] incorporated the concept of enhanced strain into the traditional co-rotational quadrilateral element, applying it to the analysis of two-dimensional solid beams, and subsequently extended the approach to three-dimensional solid elements [13–15]. Masud et al. [16,17] introduced an eight-node

hexahedral element based on the EICR method, with the objective of modeling the bending-dominated geometrically nonlinear behavior of multilayer shells. Utilizing the natural mode method, Argyris et al. [18] developed a triangular shell element, in which the geometric stiffness matrix is derived by introducing disturbances into the equilibrium equations. Kan et al. [19] employed a streamlined two-step methodology: it first simplifies internal force calculations by omitting projector matrix terms, and then enforces element equilibrium via a global-frame correction, thereby eliminating complex local-global rotation coupling while providing a unified, simplified framework for diverse element types.

Despite its demonstrated element generality and high accuracy [20–24], the practical application of the co-rotational formulation is hindered by prohibitive computational costs, particularly in nonlinear transient analysis. This limitation primarily stems from the asymmetric tangent stiffness matrix inherent to most co-rotational formulations. Since conventional rotational degrees of freedom (DOF) are non-additive and non-commutative, the tangent stiffness matrix cannot be derived directly from the second-order partial derivatives of the strain energy functional with respect to nodal variables. This asymmetry not only increases the complexity of numerical computation but also consumes more computer resources, significantly reducing solution efficiency and thus limiting its engineering application to large-scale or complex dynamic problems.

To address the aforementioned issues, Li and Izzuddin [25,26] introduced vectorial rotational variables as nodal rotational DOFs, developing a series of novel co-rotational beam elements [27] and curved shell elements [28–32]. In computing the second-order partial derivatives of the strain energy functional, the differentiation order with respect to any nodal variables is commutative. This commutativity ensures that the derived element's tangent stiffness matrix exhibits symmetry under both local and global coordinate systems, thereby improving computational performance and reducing storage needs. Furthermore, during the incremental solution procedure, the increments of the vectorial rotational variables can be updated by simple addition, eliminating the need to construct rotation matrices for vector updates and leading to more efficient computation. With the introduction of vectorial rotational variables, the potential of the co-rotational method for high-fidelity simulations and advanced engineering applications is further unlocked, promising broader adoption in the future.

Traditional time integration algorithms, particularly the Newmark family of methods [33–35], while unconditionally stable for linear problems, often exhibit energy drift and numerical instability in nonlinear regimes [36,37]. This is primarily due to significant geometric nonlinearities and strong coupling between rigid body motion and elastic deformation in such systems. These two difficulties challenge the linear assumptions within time steps and lead to instability and/or inaccuracies in long-duration simulations [38,39].

To address convergence issues in nonlinear dynamic analysis, Belytschko and Schoeberle [40] proposed using energy conservation as a stability criterion for time integration algorithms. Hughes et al. [41] introduced a constrained energy method, incorporating Lagrange multipliers into the trapezoidal integration scheme. Building on this work, Kuhl and Ramm [42] added momentum conservation as an additional constraint. However, this approach of enforcing energy conservation via Lagrange multipliers still suffers from convergence difficulties during computation. In contrast, the Energy–Momentum Conservation (EMC) algorithm proposed by Simo [36,43–45] exhibits unconditional stability for both linear and nonlinear dynamic problems. This method was later extended to beams [46–48], shells [49–52], and multibody systems [53–55], demonstrating remarkable stability even in long-duration simulations.

Nonlinear time integration methods that preserve energy conservation continue to evolve. Magisano et al. [49] improved upon Simo and Tarnow's approach by redefining the midpoint internal force vector using (1) averaged endpoint stresses and (2) a time-averaged strain-displacement tangent operator. By

retaining the Hamiltonian structure present in linear elastodynamic equations, Sánchez et al. [56] developed a symplectic Hamiltonian finite element method capable of conserving both energy and momentum. Schiebl and Romero [57] addressed the discretization of energy and momentum conserving time integration algorithms in atomic particle dynamics, with a specific focus on three key aspects: handling periodic boundary conditions, approximating three-body interactions, and extending the framework to functional potentials. Zhang et al. [58–60] enforced energy conservation as a constraint and enhanced the accuracy of the formulation by selecting three or more interpolation points for computing internal forces. To address the fundamental challenge of defining a reference frame in the deformed configuration for 3D Euler–Bernoulli beam theory, Santana et al. [61] proposed two innovative approaches based on Gram–Schmidt orthogonalization and displacement derivatives. These approaches were subsequently extended by Chhang et al. [62] to dynamic problems, yielding a stable time integration algorithm which strictly preserves both energy and momentum.

The solution scheme combining the co-rotational framework with the energy conserving algorithm can stably solve dynamic problems involving large displacements and rotations. Crisfield and Shi [20], by modifying the equations of motion and introducing correction coefficients, developed an approximately energy-conserving co-rotational algorithm for beam and rod structures, which significantly improves computational efficiency. Zhong and Crisfield [63] extended this algorithm to shell elements. Based on the co-rotational formulation, Almeida and Awruch [64] proposed an implicit time integration algorithm, which approximately conserves system energy and was successfully applied to analyze the nonlinear dynamic behavior of laminated composite shells undergoing large rotations. Yang and Xia [65,66] innovatively adopted a predictor-corrector iterative strategy combined with the generalized energy-momentum method, achieving efficient and stable solutions for co-rotational shell elements.

This paper proposes a new computational theory for multibody dynamics by employing a novel co-rotational formulation of the four-node quadrilateral shell element [29,31] and introducing a quasi-energy–momentum conserving algorithm [67]. In the proposed co-rotational formulation, the rotational variables at each node are represented by vector components. Because the sequence of differentiation is commutative when computing the second-order partial derivatives of the energy functional with respect to the nodal variables, the tangent stiffness matrix of the element exhibits symmetry. Assumed membrane and shear strains [29,68–70] are employed to mitigate the computational inaccuracies and inefficiencies arising from membrane and shear locking. Different from the exact energy–momentum conserving (EMC) algorithm [44], which constructs the strain energy functional using both strain and stress quantities, and replaces the mid-step strain values with the arithmetic mean of the endpoint strains while evaluates the stresses at the midpoint, our quasi-energy–momentum conserving algorithm directly evaluates the strain energy functional using the strain vectors. In our approach, all strain vectors, as well as their first and second partial derivatives with respect to the nodal variables, are computed via the arithmetic mean of the corresponding endpoint values. Furthermore, after discretizing the dynamic equilibrium differential equations, the term containing the first-order time derivatives of the rotational variables is incorporated into the equivalent load vector. This yields a symmetric mass matrix, consequently improving computational speed and reducing storage requirements. The proposed methodology ensures the nearly exact conservation of total energy, linear momentum, and angular momentum, while demonstrating excellent numerical stability throughout the computation.

The structure of this paper is organized as follows: Section 2 presents the co-rotational framework of the four-node quadrilateral shell element, where the element formulation is derived under local coordinate system. Section 3 establishes the dynamic equilibrium differential equations and employs a generalized mid-point interpolation method for incremental iterative solution. Section 4 validates the accuracy and stability

of the developed multibody dynamics approach through four classic shell-structure examples. Section 5 provides the conclusions.

2 Co-Rotational Formulation

In Fig. 1, $\mathbf{d}_4$ and $\mathbf{t}_4$ denote the translational displacement vectors of Node 4 in the global (O-X-Y-Z) and local (o-x-y-z) coordinate system, respectively. $\mathbf{p}_{10}$ and $\mathbf{p}_1$ represent the normal vectors of the mid-surface at Node 1 under the initial and deformed configurations in the global coordinate system, while $\mathbf{r}_{10}$ and $\mathbf{r}_1$ are the corresponding mid-surface normal vectors under the initial and deformed configuration within the local coordinate system. U, V, and W represent the displacement components along the X, Y, and Z directions of the global coordinate system, respectively. The element's transition from the initial to the current configuration consists of two stages. First, it rotates as a rigid body to an intermediate configuration without deformation. Then, it proceeds from this intermediate configuration to the current configuration through pure deformation, during which rigid-body translation also occurs.

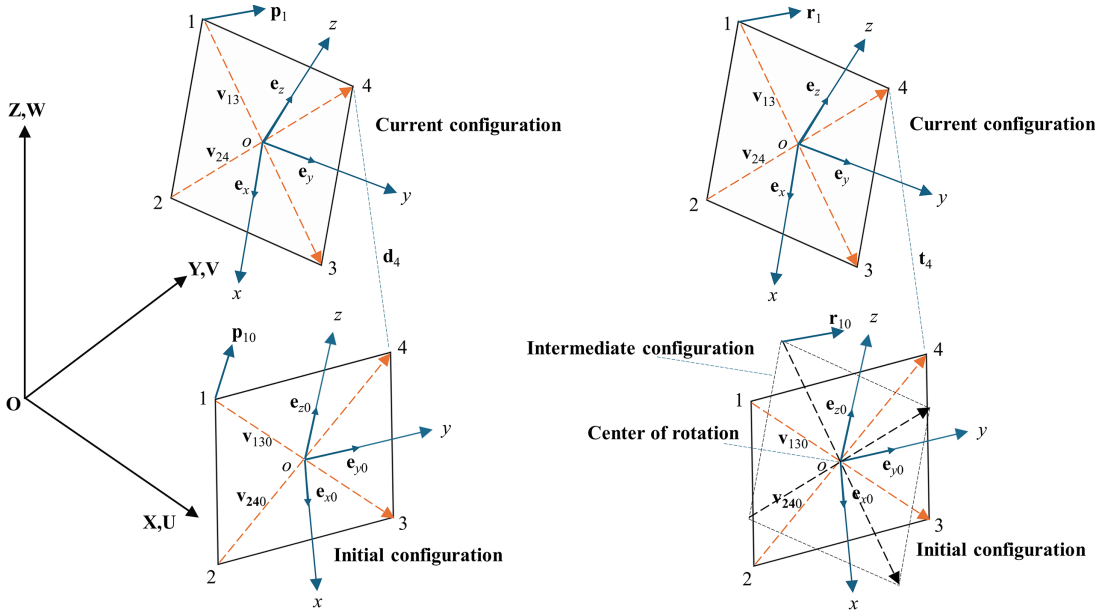

Figure 1: Co-rotational framework of the element.

For the element in its initial configuration, the local coordinate system's directional vectors are defined by the following equations:

$$\begin{cases} \mathbf{e}_{x0} = \dfrac{\mathbf{e}_{130} - \mathbf{e}_{240}}{|\mathbf{e}_{130} - \mathbf{e}_{240}|} \\ \mathbf{e}_{y0} = \dfrac{\mathbf{e}_{130} + \mathbf{e}_{240}}{|\mathbf{e}_{130} + \mathbf{e}_{240}|} \\ \mathbf{e}_{z0} = \mathbf{e}_{x0} \times \mathbf{e}_{y0} \end{cases} \tag{1a,b,c}$$

where the vectors $\mathbf{e}_{130}$ and $\mathbf{e}_{240}$, oriented from Node 1 toward Node 3 and from Node 2 toward Node 4, respectively, are the normalized direction vectors along the quadrilateral's diagonals. They are calculated from the vectors $\mathbf{v}_{130}$ and $\mathbf{v}_{240}$ as

$$\begin{cases} \mathbf{e}_{130} = \dfrac{\mathbf{v}_{130}}{|\mathbf{v}_{130}|} = \dfrac{\mathbf{X}_{30}-\mathbf{X}_{10}}{|\mathbf{X}_{30}-\mathbf{X}_{10}|} \\ \mathbf{e}_{240} = \dfrac{\mathbf{v}_{240}}{|\mathbf{v}_{240}|} = \dfrac{\mathbf{X}_{40}-\mathbf{X}_{20}}{|\mathbf{X}_{40}-\mathbf{X}_{20}|} \end{cases} \tag{2a,b}$$

where $\mathbf{X}_{i0}$ $(i = 1, 2, 3, 4)$ denotes the global coordinates of Node i in the initial configuration.

Similarly, for the deformed configuration, these directional vectors of the local coordinate system can be obtained from the normalized vectors $\mathbf{e}_{13}$ and $\mathbf{e}_{24}$ using:

$$\begin{cases} \mathbf{e}_x = \dfrac{\mathbf{e}_{13} - \mathbf{e}_{24}}{|\mathbf{e}_{13} - \mathbf{e}_{24}|} \\ \mathbf{e}_y = \dfrac{\mathbf{e}_{13} + \mathbf{e}_{24}}{|\mathbf{e}_{13} + \mathbf{e}_{24}|} \\ \mathbf{e}_z = \mathbf{e}_x \times \mathbf{e}_y \end{cases} \tag{3a,b,c}$$

where $\mathbf{e}_{13}$ and $\mathbf{e}_{24}$ are calculated as follows:

$$\begin{cases} \mathbf{e}_{13} = \dfrac{\mathbf{v}_{13}}{|\mathbf{v}_{13}|} = \dfrac{(\mathbf{X}_{30} + \mathbf{d}_3) - (\mathbf{X}_{10} + \mathbf{d}_1)}{|\,(\mathbf{X}_{30} + \mathbf{d}_3) - (\mathbf{X}_{10} + \mathbf{d}_1)\,|} \\ \mathbf{e}_{24} = \dfrac{\mathbf{v}_{24}}{|\mathbf{v}_{24}|} = \dfrac{(\mathbf{X}_{40} + \mathbf{d}_4) - (\mathbf{X}_{20} + \mathbf{d}_2)}{|\,(\mathbf{X}_{40} + \mathbf{d}_4) - (\mathbf{X}_{20} + \mathbf{d}_2)\,|} \end{cases} \tag{4a,b}$$

In the global coordinate system, this four-node quadrilateral shell element has five or six DOFs per node—three for translation and two or three for rotation. The element's nodal variable vector is:

$$\mathbf{u}_G^T = \left\langle \begin{array}{cccccccc} \mathbf{d}_1^T & \mathbf{n}_{g1}^T & \mathbf{d}_2^T & \mathbf{n}_{g2}^T & \mathbf{d}_3^T & \mathbf{n}_{g3}^T & \mathbf{d}_4^T & \mathbf{n}_{g4}^T \end{array} \right\rangle \tag{5}$$

where $\mathbf{d}_i^T = \left\langle U_i \quad V_i \quad W_i \right\rangle$ represents the three translational DOFs of Node i. For nodes of a smooth shell, or nodes away from the intersection line of a non-smooth shell, $\mathbf{n}_{gi}^T = \left\langle p_{i,n} \quad p_{i,m} \right\rangle$ represents the two rotational DOFs of Node i, $p_{i,n}$ and $p_{i,m}$ denote the two components of the mid-surface normal vector $\mathbf{p}_i$ at Node i with smaller absolute values. For nodes located on the intersection lines of a non-smooth shell, $\mathbf{n}_{gi}^T = \left\langle e_{iy,n} \quad e_{iy,m} \quad e_{iz,n} \right\rangle$, $e_{iy,n}$, $e_{iy,m}$ and $e_{iz,n}$ denote the three components with smaller absolute values of the nodal orientation vectors $\mathbf{e}_{iy}$ and $\mathbf{e}_{iz}$ at Node i [32].

In the local coordinate system, the four-node quadrilateral shell element has five DOFs per node, resulting in a total of 20 nodal variables:

$$\mathbf{u}_L^T = \left\langle \begin{array}{cccccccccc} \mathbf{t}_1^T & \boldsymbol{\theta}_1^T & \mathbf{t}_2^T & \boldsymbol{\theta}_2^T & \mathbf{t}_3^T & \boldsymbol{\theta}_3^T & \mathbf{t}_4^T & \boldsymbol{\theta}_4^T \end{array} \right\rangle \tag{6}$$

where $\mathbf{t}_i^T = \left\langle u_i \quad v_i \quad w_i \right\rangle$ represents the three translational DOFs of Node i, and $\boldsymbol{\theta}_i^T = \left\langle r_{ix} \quad r_{iy} \right\rangle$ represents the two rotational DOFs of Node i, r_{ix} and r_{iy}denote the components along the x and y directions of the mid-surface normal vector $\mathbf{r}_i$ at Node i, respectively.

In the initial configuration, the nodal coordinates $\mathbf{x}_{i0}$ in the local coordinate system can be calculated from the global-coordinate ones as:

$$\mathbf{x}_{i0} = \begin{Bmatrix} x_{i0} \\ y_{i0} \\ z_{i0} \end{Bmatrix} = \mathbf{R}_0 \mathbf{v}_{i0} \tag{7}$$

where $\mathbf{R}_0^T = \begin{bmatrix} \mathbf{e}_{x0} & \mathbf{e}_{y0} & \mathbf{e}_{z0} \end{bmatrix}$ is the directional matrix composed of the normalized vectors along the three axes of the local coordinate system in the initial configuration; $\mathbf{v}_{i0} = \mathbf{X}_{i0} - \mathbf{X}_{10}$.

The nodal translational DOFs in the local coordinate system are obtained from the global-coordinate ones as:

$$\mathbf{t}_i = \mathbf{x}_i - \mathbf{x}_{i0} = \mathbf{R}\left(\mathbf{d}_i + \mathbf{v}_{i0}\right) - \mathbf{R}_0\mathbf{v}_{i0} \tag{8}$$

where $\mathbf{R}^T = \begin{bmatrix} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z \end{bmatrix}$ is the directional matrix of the local coordinate system in the current configuration, and $\mathbf{x}_i$ represents the nodal coordinates in the local coordinate system at the current configuration.

For nodes of a smooth shell, or nodes of a non-smooth shell located away from the intersection, the relationship between the local and global components of the shell's mid-surface normal vector is expressed as:

$$\mathbf{r}_{i0} = \mathbf{R}_0\mathbf{p}_{i0} \tag{9}$$

$$\mathbf{r}_i = \mathbf{R}\mathbf{p}_i \tag{10}$$

where $\mathbf{r}_{i0}$ and $\mathbf{r}_i$ denote the mid-surface normal vectors at Node i in the local coordinate system for the initial and current configurations, respectively; $\mathbf{p}_{i0}$ and $\mathbf{p}_i$ represent their counterparts in the global coordinate system.

For nodes on the intersection line of non-smooth shells, three global rotational DOFs are employed per node. The relationship between the local and global components of the shell's mid-surface normal vector is then given by:

$$\mathbf{r}_i = \mathbf{R}\mathbf{R}_i^T\mathbf{R}_{i0}\mathbf{p}_{i0} \tag{11}$$

where, $\mathbf{R}_{i0}^T = \begin{bmatrix} \mathbf{e}_{ix0} & \mathbf{e}_{iy0} & \mathbf{e}_{iz0} \end{bmatrix}$ and $\mathbf{R}_i^T = \begin{bmatrix} \mathbf{e}_{ix} & \mathbf{e}_{iy} & \mathbf{e}_{iz} \end{bmatrix}$ represent the orientation matrices of Node i under the initial and current configurations, respectively.

At a node on the intersection line of a non-smooth shell, the triad of orientational vectors $\mathbf{e}_{ix0}$, $\mathbf{e}_{iy0}$ and $\mathbf{e}_{iz0}$ in the initial configuration coincides with the direction vectors of the three axes of the global coordinate system. The initial orientation matrix $\mathbf{R}_{i0}$ formed by these three vectors, is then given by:

$$\mathbf{R}_{i0} = \begin{bmatrix} \mathbf{e}_{ix0}^T \\ \mathbf{e}_{iy0}^T \\ \mathbf{e}_{iz0}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

In the global coordinate system, the initial value of the nodal mid-surface normal vector is calculated from the cross product of the tangent vectors along the natural coordinate ξ-axis and η-axis at the node:

$$\overline{\mathbf{p}}_{i0} = \frac{\partial\left[\sum_{j=1}^{4} N_j\left(\xi,\eta\right)\mathbf{X}_{j0}\right]}{\partial\xi} \times \frac{\partial\left[\sum_{j=1}^{4} N_j\left(\xi,\eta\right)\mathbf{X}_{j0}\right]}{\partial\eta}\Big|_{(\xi_i,\eta_i)} \tag{13}$$

where "×" is the symbol of cross product, (ξ_i, η_i) are the natural coordinates corresponding to Node i, and $N_i\left(\xi,\eta\right)$ is the shape function for Node i of the four-node quadrilateral shell element.

For a node shared by adjacent elements in a smooth shell, or by elements in the same patch of a non-smooth shell, the mid-surface normal is the average of the element normals at that node:

$$\mathbf{p}_{i0} = \frac{\sum_e \left(\overline{\mathbf{p}}_{i0}/|\overline{\mathbf{p}}_{i0}|\right)}{\left|\sum_e \left(\overline{\mathbf{p}}_{i0}/|\overline{\mathbf{p}}_{i0}|\right)\right|} \tag{14}$$

where e represents all elements containing node i.

In the local coordinate system, the quadrilateral element employs the Green-Lagrange strain measure suitable for shallow shells, which is split into the membrane strain $\boldsymbol{\varepsilon}_m$, the bending strain $z_1\boldsymbol{\chi}$, and the out-of-plane shear strain $\boldsymbol{\gamma}$. The element's potential energy functional is given by:

$$\Pi = \frac{1}{2}\int_V \left[\left(\boldsymbol{\varepsilon}_m + z_1\boldsymbol{\chi}\right)^T \mathbf{D}_1\left(\boldsymbol{\varepsilon}_m + z_1\boldsymbol{\chi}\right) + \boldsymbol{\gamma}^T\mathbf{D}_2\boldsymbol{\gamma}\right] dV - W_{ext} \tag{15}$$

where V is the element's volume; $W_{ext} = \mathbf{f}_{ext}^T\mathbf{u}_L$ denotes the work done by external forces; $\mathbf{f}_{ext}$ represents the external force vector of the element; $\mathbf{u}_L$ denotes the nodal variables of the element in the local coordinate system; $\mathbf{D}_1$ and $\mathbf{D}_2$ are the elasticity matrices.

Taking the stationarity of the potential energy, the element's internal force vector in the local coordinate system is obtained:

$$\mathbf{f}_{\text{int}} = \mathbf{f}_{ext} = \int_V \left[\left(\mathbf{B}_m + z_1\mathbf{B}_b\right)^T \mathbf{D}_1\left(\boldsymbol{\varepsilon}_m + z_1\boldsymbol{\chi}\right) + \mathbf{B}_\gamma^T D_2\boldsymbol{\gamma}\right] dV \tag{16}$$

The element's tangent stiffness matrix is obtained by computing the partial derivative of the internal force vector with respect to the nodal variables $\mathbf{u}_L$ in the local coordinate system:

$$\mathbf{k}_T = \int_V \left[\mathbf{B}_m^T\,\mathbf{D}_1\mathbf{B}_m + \left(z_1\right)^2\mathbf{B}_b^T\mathbf{D}_1\mathbf{B}_b + \frac{\partial\mathbf{B}_m^T}{\partial\mathbf{u}_L^T}\mathbf{D}_1\boldsymbol{\varepsilon}_m + \mathbf{B}_\gamma^T\mathbf{D}_2\mathbf{B}_\gamma\right] dV \tag{17}$$

where $\mathbf{B}_m$, $z_1\mathbf{B}_b$ and $\mathbf{B}_\gamma$ are respectively the first-order partial derivatives of the membrane strain $\boldsymbol{\varepsilon}_m$, bending strain $z_1\boldsymbol{\chi}$, and shear strain $\boldsymbol{\gamma}$ with respect to the nodal variables $\mathbf{u}_L$ in the local coordinate system (see Eqs. (A.1a–2b, A.4a–c, A.7a,b) on Pages 202–203 of the reference [29]); $z_1 = \frac{\zeta h}{2}$, and ζ is the natural coordinate along the shell thickness; $\frac{\partial\mathbf{B}_m^T}{\partial\mathbf{u}_L^T}$ denotes the second-order partial derivative of the membrane strain with respect to the nodal variables $\mathbf{u}_L$ (see Eq. (A.8) on Page 203 of the reference [29]).

It can be seen from Eqs. (15)–(17) that the element tangent stiffness matrix $\mathbf{k}_T$ is essentially obtained by calculating the second partial derivatives of the element strain energy functional (a scalar) with respect to the element nodal vector $\mathbf{u}_L$ in the local coordinate system. Since the order of differentiation with respect to the nodal variables is commutative when computing these second partial derivatives, the resulting element tangent stiffness matrix $\mathbf{k}_T$ is necessarily symmetric.

3 Computational Theory of Multibody Dynamics for Four-Node Quadrilateral Shell Element

For a conservative system, the Hamiltonian H, which represents the total mechanical energy, is defined as the summation of the kinetic energy K and the total potential energy V, and the total potential energy V comprises the internal strain energy V_{int} and the potential energy V_{ext} arising from conservative external loads:

$$H = K + V_{\text{int}} - V_{ext} = constant \tag{18}$$

Differentiating the Hamiltonian H with respect to time gives:

$$\frac{dH}{dt} = \frac{dK}{dt} + \frac{dV_{\text{int}}}{dt} - \frac{dV_{ext}}{dt} = 0 \tag{19}$$

In the global coordinate system, the coordinates of any point within the element can be calculated by:

$$\mathbf{X}(\xi, \eta, \zeta, t) = \mathbf{X}_0(\xi, \eta, 0) + \mathbf{d}(\xi, \eta, t) + \frac{h}{2}\zeta \mathbf{p}(\xi, \eta, t) \tag{20}$$

where, $\mathbf{X}_0(\xi, \eta, 0)$ represents the coordinates of the projection point of this point onto the shell's mid-surface in the initial configuration. The variable $\mathbf{d}(\xi, \eta, t)$ denotes the translational displacement of this projection point, while $\mathbf{p}(\xi, \eta, t)$ represents the normal vector to the mid-surface at the same location. The symbol h indicates the thickness of the shell.

The velocity is computed by taking the time derivative of the coordinates:

$$\dot{\mathbf{X}}(\xi, \eta, \zeta, t) = \dot{\mathbf{d}}(\xi, \eta, t) + \frac{1}{2}\zeta h \dot{\mathbf{p}}(\xi, \eta, t) = \sum_{i=1}^{4} N_i \left(\dot{\mathbf{d}}_i + \frac{1}{2}\zeta h \dot{\mathbf{p}}_i \right) \tag{21}$$

where $\dot{\mathbf{d}}_i$ and $\dot{\mathbf{p}}_i$ represent the first-order derivatives of $\mathbf{d}_i$ and $\mathbf{p}_i$ with respect to time, respectively.

Integrating the nodal kinetic energy over the element volume yields the kinetic energy functional of the four-node quadrilateral shell element:

$$K^e = \frac{1}{2}\rho \int_V \left[\sum_{i=1}^{4} N_i \left(\dot{\mathbf{d}}_i + \frac{\zeta h \dot{\mathbf{p}}_i}{2} \right) \right]^T \left[\sum_{j=1}^{4} N_j \left(\dot{\mathbf{d}}_j + \frac{\zeta h \dot{\mathbf{p}}_j}{2} \right) \right] dV \tag{22}$$

The time derivative of the element kinetic energy functional is calculated as:

$$\frac{dK^e}{dt} = \rho \int_V \left[\left(\sum_{i=1}^{4} N_i \dot{\mathbf{d}}_i \right)^T \left(\sum_{j=1}^{4} N_j \ddot{\mathbf{d}}_j \right) + \frac{\zeta^2 h^2}{4} \left(\sum_{i=1}^{4} N_i \dot{\mathbf{p}}_i \right)^T \left(\sum_{j=1}^{4} N_j \ddot{\mathbf{p}}_j \right) \right] dV \tag{23}$$

For Node i away from intersections of shells, the first derivative of the normal vector $\dot{\mathbf{p}}_i$ with respect to time is:

$$\dot{\mathbf{p}}_i = \frac{\partial \mathbf{p}_i}{\partial \mathbf{n}_{gi}^T} \dot{\mathbf{n}}_{gi} = \begin{bmatrix} \frac{\partial p_{i,X}}{\partial p_{i,n}} & \frac{\partial p_{i,X}}{\partial p_{i,m}} \\ \frac{\partial p_{i,Y}}{\partial p_{i,n}} & \frac{\partial p_{i,Y}}{\partial p_{i,m}} \\ \frac{\partial p_{i,Z}}{\partial p_{i,n}} & \frac{\partial p_{i,Z}}{\partial p_{i,m}} \end{bmatrix} \begin{Bmatrix} \dot{p}_{i,n} \\ \dot{p}_{i,m} \end{Bmatrix} \tag{24}$$

where $m \neq n \neq l$, and $m, n, l \in \{X, Y, Z\}$.

The second derivative of the normal vector $\ddot{\mathbf{p}}_i$ with respect to time is:

$$\ddot{\mathbf{p}}_i = \frac{\partial \mathbf{p}_i}{\partial \mathbf{n}_{gi}^T} \ddot{\mathbf{n}}_{gi} + \frac{\partial^2 \mathbf{p}_i}{\partial \mathbf{n}_{gi}{}^{T2}} \dot{\mathbf{n}}_{gi} \otimes \dot{\mathbf{n}}_{gi} \tag{25}$$

where:

$$\ddot{\mathbf{n}}_{gi} = \begin{Bmatrix} \ddot{p}_{i,n} \\ \ddot{p}_{i,m} \end{Bmatrix} \tag{26}$$

Define the matrix $\mathbf{A}_{1i}$ and the vector $\mathbf{A}_{2i}$:

$$\mathbf{A}_{1i} = \frac{\partial \mathbf{p}_i}{\partial \mathbf{n}_{gi}^T} = \begin{bmatrix} \frac{\partial p_{i,X}}{\partial p_{i,n}} & \frac{\partial p_{i,X}}{\partial p_{i,m}} \\ \frac{\partial p_{i,Y}}{\partial p_{i,n}} & \frac{\partial p_{i,Y}}{\partial p_{i,m}} \\ \frac{\partial p_{i,Z}}{\partial p_{i,n}} & \frac{\partial p_{i,Z}}{\partial p_{i,m}} \end{bmatrix} \tag{27}$$

$$\mathbf{A}_{2i} = \frac{\partial^2 \mathbf{p}_i}{\partial \mathbf{n}_{gi}{}^{T2}} \dot{\mathbf{n}}_{gi} \otimes \dot{\mathbf{n}}_{gi} = \begin{Bmatrix} \frac{\partial^2 p_{i,X}}{\partial p_{i,n}{}^2} \dot{p}_{i,n}^2 + 2\frac{\partial^2 p_{i,X}}{\partial p_{i,n} \partial p_{i,m}} \dot{p}_{i,n} \dot{p}_{i,m} + \frac{\partial^2 p_{i,X}}{\partial p_{i,m}{}^2} \dot{p}_{i,m}^2 \\ \frac{\partial^2 p_{i,Y}}{\partial p_{i,n}{}^2} \dot{p}_{i,n}^2 + 2\frac{\partial^2 p_{i,Y}}{\partial p_{i,n} \partial p_{i,m}} \dot{p}_{i,n} \dot{p}_{i,m} + \frac{\partial^2 p_{i,Y}}{\partial p_{i,m}{}^2} \dot{p}_{i,m}^2 \\ \frac{\partial^2 p_{i,Z}}{\partial p_{i,n}{}^2} \dot{p}_{i,n}^2 + 2\frac{\partial^2 p_{i,Z}}{\partial p_{i,n} \partial p_{i,m}} \dot{p}_{i,n} \dot{p}_{i,m} + \frac{\partial^2 p_{i,Z}}{\partial p_{i,m}{}^2} \dot{p}_{i,m}^2 \end{Bmatrix} \tag{28}$$

where:

$$\begin{cases} \frac{\partial^2 p_{i,n}}{\partial p_{i,n}{}^2} = \frac{\partial^2 p_{i,n}}{\partial p_{i,m}{}^2} = \frac{\partial^2 p_{i,n}}{\partial p_{i,n} \partial p_{i,m}} = 0 \\ \frac{\partial^2 p_{i,m}}{\partial p_{i,n}{}^2} = \frac{\partial^2 p_{i,m}}{\partial p_{i,m}{}^2} = \frac{\partial^2 p_{i,m}}{\partial p_{i,n} \partial p_{i,m}} = 0 \\ \frac{\partial^2 p_{i,l}}{\partial p_{i,n}{}^2} = -\frac{1}{p_{i,l}} - \frac{p^2{}_{i,n}}{p^3{}_{i,l}} \\ \frac{\partial^2 p_{i,l}}{\partial p_{i,m}{}^2} = -\frac{1}{p_{i,l}} - \frac{p^2{}_{i,m}}{p^3{}_{i,l}} \\ \frac{\partial^2 p_{i,l}}{\partial p_{i,n} \partial p_{i,m}} = -\frac{p_{i,n} p_{i,m}}{p^3{}_{i,l}} \end{cases} \tag{29a–e}$$

Therefore, $\dot{\mathbf{p}}_i$ and $\ddot{\mathbf{p}}_i$, can be expressed as:

$$\begin{cases} \dot{\mathbf{p}}_i = \mathbf{A}_{1i} \dot{\mathbf{n}}_{gi} \\ \ddot{\mathbf{p}}_i = \mathbf{A}_{1i} \ddot{\mathbf{n}}_{gi} + \mathbf{A}_{2i} \end{cases} \tag{30a,b}$$

For Node *i* located on an intersection line of non-smooth shells,

$$\dot{\mathbf{p}}_i = \left(\frac{\partial \mathbf{R}_i^T \mathbf{R}_{i0} \mathbf{p}_{i0}}{\partial \mathbf{n}_{gi}^T} \right) \dot{\mathbf{n}}_{gi} = \left(\frac{\partial \begin{bmatrix} \mathbf{e}_{ix} & \mathbf{e}_{iy} & \mathbf{e}_{iz} \end{bmatrix}}{\partial \mathbf{n}_{gi}^T} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{i0} \right) \begin{Bmatrix} \dot{e}_{iy,n} \\ \dot{e}_{iy,m} \\ \dot{e}_{iz,n} \end{Bmatrix} \tag{31}$$

$$\ddot{\mathbf{p}}_i = \frac{\partial \mathbf{p}_i}{\partial \mathbf{n}_{gi}^T} \ddot{\mathbf{n}}_{gi} + \frac{\partial^2 \mathbf{p}_i}{\partial \mathbf{n}_{gi}^{T2}} \dot{\mathbf{n}}_{gi} \otimes \dot{\mathbf{n}}_{gi} \tag{32}$$

Write Eqs. (31) and (32) in the form of Eq. (30a,b); then we have:

$$\mathbf{A}_{1i} = \frac{\partial \mathbf{p}_i}{\partial \mathbf{n}_{gi}^T} = \frac{\partial \begin{bmatrix} \mathbf{e}_{ix} & \mathbf{e}_{iy} & \mathbf{e}_{iz} \end{bmatrix}}{\partial \mathbf{n}_{gi}^T} \mathbf{p}_{i0} \tag{33}$$

$$\mathbf{A}_{2i} = \frac{\partial^2 \mathbf{p}_i}{\partial \mathbf{n}_{gi}{}^{T^2}} \dot{\mathbf{n}}_{gi} \otimes \dot{\mathbf{n}}_{gi} = \left(\begin{bmatrix} \frac{\partial^2 \mathbf{e}_{ix}}{\partial \mathbf{n}_{gi}^T \partial \mathbf{n}_{gi}^T} & \frac{\partial^2 \mathbf{e}_{iy}}{\partial \mathbf{n}_{gi}^T \partial \mathbf{n}_{gi}^T} & \frac{\partial^2 \mathbf{e}_{iz}}{\partial \mathbf{n}_{gi}^T \partial \mathbf{n}_{gi}^T} \end{bmatrix} \mathbf{p}_{i0} \right) \dot{\mathbf{n}}_{gi} \otimes \dot{\mathbf{n}}_{gi} \tag{34}$$

where:

$$\begin{cases} \dfrac{\partial \mathbf{e}_{iy}}{\partial \mathbf{n}_{gi}^T} = \begin{bmatrix} \dfrac{\partial \mathbf{e}_{iy}}{\partial e_{iy,n}} & \dfrac{\partial \mathbf{e}_{iy}}{\partial e_{iy,m}} & \dfrac{\partial \mathbf{e}_{iy}}{\partial e_{iz,n}} \end{bmatrix} \\ \dfrac{\partial \mathbf{e}_{iz}}{\partial \mathbf{n}_{gi}^T} = \begin{bmatrix} \dfrac{\partial \mathbf{e}_{iz}}{\partial e_{iy,n}} & \dfrac{\partial \mathbf{e}_{iz}}{\partial e_{iy,m}} & \dfrac{\partial \mathbf{e}_{iz}}{\partial e_{iz,n}} \end{bmatrix} \\ \dfrac{\partial \mathbf{e}_{ix}}{\partial \mathbf{n}_{gi}^T} = \mathbf{e}_{iy} \times \dfrac{\partial \mathbf{e}_{iz}}{\partial \mathbf{n}_{gi}^T} + \dfrac{\partial \mathbf{e}_{iy}}{\partial \mathbf{n}_{gi}^T} \times \mathbf{e}_{iz} \end{cases} \tag{35a,b,c}$$

$$\frac{\partial^2 \mathbf{e}_{ix}}{\partial \mathbf{n}_{gi}^T \partial \mathbf{n}_{gi}^T} = \mathbf{e}_{iy} \times \frac{\partial^2 \mathbf{e}_{iz}}{\partial \mathbf{n}_{gi}^T \partial \mathbf{n}_{gi}^T} + \frac{\partial^2 \mathbf{e}_{iy}}{\partial \mathbf{n}_{gi}^T \partial \mathbf{n}_{gi}^T} \times \mathbf{e}_{iz} + \frac{\partial \mathbf{e}_{iy}}{\partial \mathbf{n}_{gi}^T} \times \frac{\partial \mathbf{e}_{iz}}{\partial \mathbf{n}_{gi}^T} + \frac{\partial \mathbf{e}_{iy}}{\partial \mathbf{n}_{gi}^T} \times \frac{\partial \mathbf{e}_{iz}}{\partial \mathbf{n}_{gi}^T} \tag{36}$$

The expressions for the first-order partial derivatives in Eqs. (35) and (36) are given by Eqs. (B20a–k) and (B21a–c) on Pages 596–597 of the reference [32], while those for the second-order partial derivatives are given by Eq. (B47a–r) on Page 600 of the reference [32].

Substitute the time derivatives of the shell's mid-surface normal vector at Node *i* into Eq. (23) and define the term $\mathbf{f}_m$ containing $\mathbf{A}_{2j}$:

$$\mathbf{f}_m = \frac{\rho h^3}{12} \begin{Bmatrix} \mathbf{0} \\ \int_{-1}^{1} \int_{-1}^{1} N_1 \sum_{j=1}^{4} N_j \mathbf{A}_{11}^T \mathbf{A}_{2j} J_a d\xi d\eta \\ \vdots \\ \mathbf{0} \\ \int_{-1}^{1} \int_{-1}^{1} N_4 \sum_{j=1}^{4} N_j \mathbf{A}_{14}^T \mathbf{A}_{2j} J_a d\xi d\eta \end{Bmatrix} \tag{37}$$

where J_a represents the Jacobian determinant,

$$J_a = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{vmatrix} \tag{38}$$

After rearrangement, the following expression is obtained:

$$\frac{dK^e}{dt} = \begin{Bmatrix} \dot{\mathbf{d}}_1 \\ \dot{\mathbf{n}}_{g1} \\ \vdots \\ \dot{\mathbf{d}}_4 \\ \dot{\mathbf{n}}_{g4} \end{Bmatrix}^T \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \dots & \mathbf{C}_{14} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{11} & \dots & \mathbf{0} & \mathbf{Q}_{14} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_{41} & \mathbf{0} & \dots & \mathbf{C}_{44} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{41} & \dots & \mathbf{0} & \mathbf{Q}_{44} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{d}}_1 \\ \ddot{\mathbf{n}}_{g1} \\ \vdots \\ \ddot{\mathbf{d}}_4 \\ \ddot{\mathbf{n}}_{g4} \end{Bmatrix} + \begin{Bmatrix} \dot{\mathbf{d}}_1 \\ \dot{\mathbf{n}}_{g1} \\ \vdots \\ \dot{\mathbf{d}}_4 \\ \dot{\mathbf{n}}_{g4} \end{Bmatrix}^T \mathbf{f}_m \tag{39}$$

where:

$$\begin{cases} \mathbf{C}_{ij} = \rho h \int_{-1}^{1}\int_{-1}^{1} N_i N_j J_a d\xi d\eta \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{Q}_{ij} = \dfrac{\rho h^3}{12} \int_{-1}^{1}\int_{-1}^{1} N_i N_j \mathbf{A}_{1i}^T \mathbf{A}_{1j} J_a d\xi d\eta \end{cases} \tag{40a,b}$$

The definition of the element mass matrix $\mathbf{m}$ is as follows:

$$\mathbf{m} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \cdots & \mathbf{C}_{14} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{11} & \cdots & \mathbf{0} & \mathbf{Q}_{14} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_{41} & \mathbf{0} & \cdots & \mathbf{C}_{44} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{41} & \cdots & \mathbf{0} & \mathbf{Q}_{44} \end{bmatrix} \tag{41}$$

Thus, the first-order time derivative of the element kinetic energy is given by:

$$\frac{dK^e}{dt} = \dot{\mathbf{u}}_G^T \mathbf{m} \ddot{\mathbf{u}}_G + \dot{\mathbf{u}}_G^T \mathbf{f}_m \tag{42}$$

The mass matrices $\mathbf{m}$ of all four-node quadrilateral elements in the system are assembled into the global mass matrix $\mathbf{M}$, the nodal variable vectors $\mathbf{u}_G$ of all elements are assembled into the global nodal variable vector $\mathbf{U}$, and $\mathbf{f}_m$ are assembled into $\mathbf{F}_m$. The first-order time derivative of system's total kinetic energy is then obtained:

$$\frac{dK}{dt} = \sum_n \frac{dK^e}{dt} = \dot{\mathbf{U}}^T \mathbf{M} \ddot{\mathbf{U}} + \dot{\mathbf{U}}^T \mathbf{F}_m \tag{43}$$

The time derivative of the element strain energy functional is calculated as:

$$\frac{dV_{\text{int}}^e}{dt} = \dot{\mathbf{u}}_G^T \left(\frac{\partial \mathbf{u}_L}{\partial \mathbf{u}_G^T} \right)^T \mathbf{f}_{\text{int}} == \dot{\mathbf{u}}_G^T \mathbf{T}^T \mathbf{f}_{\text{int}} = \dot{\mathbf{u}}_G^T \mathbf{f}_G \tag{44}$$

where $\mathbf{T}$ denotes the transformation matrix from local to global coordinate system (see Eq. (39) on Page 580, and Eqs. (B1)–(B21c) on Pages 595–597 of the reference [32]), $\mathbf{f}_G$ represents the element's internal force vector in the global coordinate system, with

$$\begin{cases} \mathbf{T} = \dfrac{\partial \mathbf{u}_L}{\partial \mathbf{u}_G^T} \\ \mathbf{f}_G = \mathbf{T}^T \mathbf{f}_{\text{int}} \end{cases} \tag{45a,b}$$

The total linear momentum $\mathbf{L}^e$ and the total angular momentum $\mathbf{J}^e$ (the reference point is at the origin of global coordinate system) of a shell element can be calculated as following,

$$\mathbf{L}^e = \int_{\text{V}} \rho \dot{\mathbf{X}} d\text{V} = \int_{\text{V}} \rho \left(\dot{\mathbf{d}} + \frac{1}{2} h \zeta \dot{\mathbf{p}} \right) d\text{V} = \int_{\text{A}} \rho \dot{\mathbf{d}} h d\text{A} = \int_{\text{A}} \rho h \sum_{i=1}^{4} N_i \dot{\mathbf{d}}_i d\text{A} \tag{46}$$

$$\begin{aligned}\mathbf{J}^e &= \int_{\mathrm{V}} \mathbf{X} \times \rho \dot{\mathbf{X}} d\mathrm{V} = \int_{\mathrm{A}} \overline{\mathbf{X}} \times \rho h \dot{\mathbf{d}} d\mathrm{A} + \frac{1}{12} \rho h^3 \int_{\mathrm{A}} \mathbf{p} \times \dot{\mathbf{p}} d\mathrm{A} \\ &= \rho h \int_{\mathrm{A}} \sum_{i=1}^{4} \left[N_i \left(\mathbf{X}_{i0} + \mathbf{d}_i \right) \times \dot{\mathbf{d}}_i \right] d\mathrm{A} + \frac{1}{12} \rho h^3 \int_{\mathrm{A}} \sum_{i=1}^{4} \left(N_i \mathbf{p}_i \times \dot{\mathbf{p}}_i \right) d\mathrm{A}\end{aligned} \tag{47}$$

where $\overline{\mathbf{X}}$ is the global coordinate of a point of shell mid-surface at Time t, with

$$\overline{\mathbf{X}} = \overline{\mathbf{X}}_0 \left(\xi, \eta, 0 \right) + \mathbf{d} \left(\xi, \eta, t \right) \tag{48}$$

The total linear momentum and total angular momentum of the system are respectively expressed as:

$$\mathbf{L} = \sum_e \mathbf{L}^e = \sum_e \rho h \int_A \sum_{i=1}^{4} N_i \dot{\mathbf{d}}_i dA \tag{49}$$

$$\mathbf{J} = \sum_e \mathbf{J}^e = \sum_e \rho h \int_A \left[\sum_{i=1}^{4} N_i \left(\mathbf{X}_{i0} + \mathbf{d}_i \right) \times \dot{\mathbf{d}}_i + \sum_{i=1}^{4} \frac{h^2}{12} N_i \left(\mathbf{p}_i \times \dot{\mathbf{p}}_i \right) \right] dA \tag{50}$$

The time derivative of the system's total strain energy is obtained by assembling the time derivatives of the individual elemental strain energies:

$$\frac{dV_{\mathrm{int}}}{dt} = \sum_n \frac{dV_{\mathrm{int}}^e}{dt} = \sum_n \left(\dot{\mathbf{u}}_G^T \mathbf{f}_G \right) = \dot{\mathbf{U}}^T \mathbf{F}_{\mathrm{int}} \tag{51}$$

where n represents all elements in the system; $\mathbf{F}_{\mathrm{int}}$ is the system's internal force vector in the global coordinate system, assembled from $\mathbf{f}_G$,

$$\mathbf{F}_{\mathrm{int}} = \sum_n \mathbf{f}_G = \sum_n \left(\mathbf{T}^T \mathbf{f}_{\mathrm{int}} \right) \tag{52}$$

The system's external work accumulated from t_n to t_{n+1} is:

$$V_{ext} = \int_{t_n}^{t_{n+1}} \dot{\mathbf{U}}^T \mathbf{F}_{ext} dt \tag{53}$$

The first-order time derivative of the system's external work is:

$$\frac{dV_{ext}}{dt} = \dot{\mathbf{U}}^T \mathbf{F}_{ext} \tag{54}$$

Substituting the time derivatives of the system's kinetic energy functional $\frac{dK}{dt}$, strain energy fuctional $\frac{dV_{\mathrm{int}}}{dt}$, and external work $\frac{dV_{ext}}{dt}$ into Eq. (19), and eliminating the $\dot{\mathbf{U}}^T$term yields the system's dynamic equilibrium differential equation:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{F}_m + \mathbf{F}_{\mathrm{int}} - \mathbf{F}_{ext} = 0 \tag{55}$$

The term $\mathbf{F}_m$ containing the first-order time derivatives of the rotational variables is then incorporated into the equivalent load vector.

For $[t_n, t_{n+1}]$ with time step length $\Delta t = t_{n+1} - t_n$, based on the generalized energy-momentum method [71,72], appropriate generalized midpoints $t_{n+1-\alpha_f}$ and $t_{n+1-\alpha_m}$ are selected:

$$\begin{cases} t_{n+1-\alpha_f} = \left(1-\alpha_f\right) t_{n+1} + \alpha_f t_n \\ t_{n+1-\alpha_m} = \left(1-\alpha_m\right) t_{n+1} + \alpha_m t_n \end{cases} \tag{56a,b}$$

where α_f and α_m are integration parameters, controlled by the spectral radius ρ_∞ (see Eq. (67a,b) in the reference [67]).

Based on the Newmark-β method, the nodal velocities $\dot{\mathbf{U}}\left(t_{n+1-\alpha_f}\right)$ and accelerations $\ddot{\mathbf{U}}\left(t_{n+1-\alpha_m}\right)$ at the generalized midpoints are given by:

$$\begin{cases} \dot{\mathbf{U}}\left(t_{n+1-\alpha_f}\right) = \frac{(1-\alpha_f)\gamma}{\beta\Delta t}\left[\mathbf{U}\left(t_{n+1}\right) - \mathbf{U}\left(t_n\right)\right] - \frac{\gamma-\beta-\alpha_f\gamma}{\beta}\dot{\mathbf{U}}\left(t_n\right) - \frac{(\gamma-2\beta)(1-\alpha_f)}{2\beta}\ddot{\mathbf{U}}\left(t_n\right)\Delta t \\ \ddot{\mathbf{U}}\left(t_{n+1-\alpha_m}\right) = \frac{\left(1-\alpha_m\right)}{\beta\Delta t^2}\left[\mathbf{U}\left(t_{n+1}\right) - \mathbf{U}\left(t_n\right)\right] - \frac{\left(1-\alpha_m\right)}{\beta\Delta t}\dot{\mathbf{U}}\left(t_n\right) - \frac{\left(1-2\beta-\alpha_m\right)}{2\beta}\ddot{\mathbf{U}}\left(t_n\right) \end{cases} \tag{57a,b}$$

where γ and β are integration parameters, which are also controlled by the spectral radius ρ_∞ (see Eq. (70a,b) in the reference [67]).

At the generalized midpoints $t_{n+1-\alpha_f}$ and $t_{n+1-\alpha_m}$, the system's dynamic equilibrium differential Eq. (55) can be expressed as:

$$\mathbf{M}\left(t_{n+1-\alpha_f}\right)\ddot{\mathbf{U}}\left(t_{n+1-\alpha_m}\right) + \mathbf{F}_m\left(t_{n+1-\alpha_f}\right) + \mathbf{F}_{\text{int}}\left(t_{n+1-\alpha_f}\right) - \mathbf{F}_{ext}\left(t_{n+1-\alpha_f}\right) = 0 \tag{58}$$

Considering Eq. (57), the first term of Eq. (58) can be rewritten as:

$$\begin{aligned} \mathbf{M}\left(t_{n+1-\alpha_f}\right)\ddot{\mathbf{U}}\left(t_{n+1-\alpha_m}\right) = \mathbf{M}\left(t_{n+1-\alpha_f}\right)\Bigg[&\frac{\left(1-\alpha_m\right)}{\beta\Delta t^2}\left(\mathbf{U}\left(t_{n+1}\right) - \mathbf{U}\left(t_n\right)\right) \\ &- \frac{\left(1-\alpha_m\right)}{\beta\Delta t}\dot{\mathbf{U}}\left(t_n\right) - \frac{\left(1-2\beta-\alpha_m\right)}{2\beta}\ddot{\mathbf{U}}\left(t_n\right)\Bigg] \end{aligned} \tag{59}$$

All variables at time t_n are known. All known terms in Eq. (59) are consolidated into $\mathbf{F}_a\left(t_{n+1-\alpha_f}\right)$:

$$\mathbf{F}_a\left(t_{n+1-\alpha_f}\right) = \mathbf{M}\left(t_{n+1-\alpha_f}\right)\left[\frac{1-\alpha_m}{\beta\Delta t^2}\mathbf{U}\left(t_n\right) + \frac{1-\alpha_m}{\beta\Delta t}\dot{\mathbf{U}}\left(t_n\right) + \frac{1-2\beta-\alpha_m}{2\beta}\ddot{\mathbf{U}}\left(t_n\right)\right] \tag{60}$$

then, Eq. (58) can be rewritten as:

$$\mathbf{M}\left(t_{n+1-\alpha_f}\right)\frac{1-\alpha_m}{\beta\Delta t^2}\mathbf{U}\left(t_{n+1}\right) = \mathbf{F}_{ext}\left(t_{n+1-\alpha_f}\right) + \mathbf{F}_a\left(t_{n+1-\alpha_f}\right) - \mathbf{F}_m\left(t_{n+1-\alpha_f}\right) - \mathbf{F}_{\text{int}}\left(t_{n+1-\alpha_f}\right) \tag{61}$$

Eq. (61) can be solved using an incremental iterative method. The incremental equation is written as:

$$\begin{aligned} &\left[\mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)\frac{1-\alpha_m}{\beta\Delta t^2} + \left(1-\alpha_f\right)\mathbf{K}_G^{k-1}\left(t_{n+1-\alpha_f}\right)\right]\Delta\mathbf{U}^k\left(t_{n+1}\right) \\ &= \mathbf{F}_{ext}^k\left(t_{n+1-\alpha_f}\right) + \mathbf{F}_a^{k-1}\left(t_{n+1-\alpha_f}\right) - \mathbf{F}_m^{k-1}\left(t_{n+1-\alpha_f}\right) - \mathbf{F}_{\text{int}}^{k-1}\left(t_{n+1-\alpha_f}\right) - \mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)\frac{1-\alpha_m}{\beta\Delta t^2}\mathbf{U}^{k-1}\left(t_{n+1}\right) \end{aligned} \tag{62}$$

where, the iteration step is indicated by the superscripts $k-1$ and k, respectively; $\Delta\mathbf{U}^{k}$ is the infinitesimal displacement increment; $\mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)$, $\mathbf{K}_G^{k-1}\left(t_{n+1-\alpha_f}\right)$, $\mathbf{F}_a^{k-1}\left(t_{n+1-\alpha_f}\right)$, $\mathbf{F}_m^{k-1}\left(t_{n+1-\alpha_f}\right)$ and $\mathbf{F}_{\text{int}}^{k-1}\left(t_{n+1-\alpha_f}\right)$ are evaluated using the nodal variables at the end of the previous iteration step $k-1$; $\mathbf{M}^{k}\left(t_{n+1-\alpha_f}\right)$ and $\mathbf{F}_m^{k}\left(t_{n+1-\alpha_f}\right)$ are provisionally substituted with $\mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)$ and $\mathbf{F}_m^{k-1}\left(t_{n+1-\alpha_f}\right)$, their increments are neglected within the current iteration. The purpose of this treatment is to obtain both an equivalent mass matrix and an equivalent elemental tangent stiffness matrix that are symmetric. This symmetry helps reduce computer memory requirements and improves the solution efficiency of the time-discretized dynamic differential equations. This treatment does not compromise the accuracy of energy and momentum conservation. Once the nodal variables at the next iteration are obtained, these terms are recomputed using the updated variables. Thus, the computation of these terms is essentially delayed by one iteration. When the iteration converges for the current time step, the resulting error is negligible. Moreover, since all subsequent calculations are performed using the updated nodal variables, this error does not accumulate.

In Eq. (62), $\mathbf{K}_G^{k-1}\left(t_{n+1-\alpha_f}\right)$ is the system's tangent stiffness matrix, it is assembled from the element's tangent stiffness matrix $\mathbf{k}_G^{k-1}\left(t_{n+1-\alpha_f}\right)$ at the $(k-1)$th iteration step of time $t_{n+1-\alpha_f}$, thus $\mathbf{K}_G\left(t_{n+1-\alpha_f}\right)$ is obtained as following,

$$\begin{aligned}\mathbf{K}_G\left(t_{n+1-\alpha_f}\right) &= \sum_n \mathbf{k}_G\left(t_{n+1-\alpha_f}\right) = \sum_n \frac{\partial \mathbf{f}_G}{\partial \mathbf{u}_G^T}\Big|_{t_{n+1-\alpha_f}} \\ &= \sum_n \left[\mathbf{T}^T\left(t_{n+1-\alpha_f}\right)\mathbf{k}_T\left(t_{n+1-\alpha_f}\right)\mathbf{T}\left(t_{n+1-\alpha_f}\right) + \frac{\partial \mathbf{T}}{\partial \mathbf{u}_G^T}\Big|_{t_{n+1-\alpha_f}}\mathbf{f}_{\text{int}}\left(t_{n+1-\alpha_f}\right)\right]\end{aligned} \tag{63}$$

where, the calculation of $\frac{\partial \mathbf{T}}{\partial \mathbf{u}_G^T}$ can be found in the reference [32] (See Eq. (41) on Page 580, and Eqs. (B22)–(B47r) on Pages 597–600).

As can be seen from Eqs. (15) and (16), (45a,b), and (63), the elemental tangent stiffness matrix $\mathbf{k}_G$ in the global coordinate system is fundamentally obtained by taking the second partial derivatives of the element strain energy functional—a scalar quantity—with respect to the element nodal displacement vector $\mathbf{u}_G$. Because the order of differentiation with respect to the nodal variables is commutative when computing these second partial derivatives, the resulting elemental tangent stiffness matrix $\mathbf{k}_G$ is necessarily symmetric. Consequently, the global tangent stiffness matrix $\mathbf{K}_G\left(t_{n+1-\alpha_f}\right)$ of the system in Eq. (63) is also symmetric.

In Eq. (62), the terms $\mathbf{F}_{ext}^{k}\left(t_{n+1-\alpha_f}\right)$, $\mathbf{F}_a^{k-1}\left(t_{n+1-\alpha_f}\right)$, $\mathbf{F}_m^{k-1}\left(t_{n+1-\alpha_f}\right)$, $\mathbf{F}_{\text{int}}^{k-1}\left(t_{n+1-\alpha_f}\right)$ and $\mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)\frac{1-\alpha_m}{\beta\Delta t^2}\mathbf{U}^{k-1}\left(t_{n+1}\right)$ are incorporated into the equivalent load vector. The equivalent stiffness matrix $\overline{\mathbf{K}}^{k}\left(t_{n+1-\alpha_f}\right)$ and the equivalent load vector $\overline{\mathbf{P}}^{k}\left(t_{n+1-\alpha_f}\right)$ are then defined as:

$$\begin{cases}\overline{\mathbf{K}}^{k}\left(t_{n+1-\alpha_f}\right) = \mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)\dfrac{1-\alpha_m}{\beta\Delta t^2} + \left(1-\alpha_f\right)\mathbf{K}_G^{k-1}\left(t_{n+1-\alpha_f}\right) \\ \overline{\mathbf{P}}^{k}\left(t_{n+1-\alpha_f}\right) = \mathbf{F}_{ext}^{k}\left(t_{n+1-\alpha_f}\right) + \mathbf{F}_a^{k-1}\left(t_{n+1-\alpha_f}\right) - \mathbf{F}_m^{k-1}\left(t_{n+1-\alpha_f}\right) \\ \qquad -\mathbf{F}_{\text{int}}^{k-1}\left(t_{n+1-\alpha_f}\right) - \mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)\dfrac{1-\alpha_m}{\beta\Delta t^2}\mathbf{U}^{k-1}\left(t_{n+1}\right)\end{cases} \tag{64a,b}$$

since both $\mathbf{M}^{k-1}\left(t_{n+1-\alpha_f}\right)$ and $\mathbf{K}_G^{k-1}\left(t_{n+1-\alpha_f}\right)$ are symmetric, the equivalent stiffness matrix $\overline{\mathbf{K}}^{k}\left(t_{n+1-\alpha_f}\right)$ is also symmetric.

Substituting Eq. (64) into Eq. (62) yields:

$$\overline{\mathbf{K}}^{k}\left(t_{n+1-\alpha_f}\right)\Delta\mathbf{U}^{k}\left(t_{n+1}\right) = \overline{\mathbf{P}}^{k}\left(t_{n+1-\alpha_f}\right) \tag{65}$$

solving this equation obtains the incremental displacement $\Delta\mathbf{U}^k\left(t_{n+1}\right)$ at the kth iteration step. The nodal variables are updated as follows:

$$\mathbf{U}^k\left(t_{n+1}\right)=\mathbf{U}^{k-1}\left(t_{n+1}\right)+\Delta\mathbf{U}^k\left(t_{n+1}\right) \tag{66}$$

After each iteration step, update the equivalent stiffness and load terms using the newly obtained nodal variables.

An energy-based convergence criterion is employed. The ratio between the work done by the residual load at the kth iteration step and the work done at the first iteration step is calculated as below

$$cc=\frac{E^k}{E^1}=\frac{\left[\overline{\mathbf{P}}^k\left(t_{n+1-\alpha_f}\right)\right]^T\Delta\mathbf{U}^k\left(t_{n+1}\right)}{\left[\overline{\mathbf{P}}^1\left(t_{n+1-\alpha_f}\right)\right]^T\Delta\mathbf{U}^1\left(t_{n+1}\right)} \tag{67}$$

Convergence is attained once this ratio meets $cc\le 1\times10^{-6}$ and the iteration number satisfies $k\ge 3$. The iterative process for the time step $\left[t_n,t_{n+1}\right]$ terminates, then a new iteration procedure starts.

To overcome membrane and shear locking problems, assumed membrane strain, assumed shear strain, and their first- and second-order partial derivatives with respect to the local nodal variables $\mathbf{u}_L$ are employed to replace the corresponding conforming strains and their first- and second-order partial derivatives in the computation of $\mathbf{f}_{\text{int}}\left(t_{n+1-\alpha_f}\right)$ and $\mathbf{k}_T\left(t_{n+1-\alpha_f}\right)$:

$$\begin{aligned}\mathbf{f}_{\text{int}}\left(t_{n+1-\alpha_f}\right)=\int_V\Big[&\tilde{\mathbf{B}}_m^T\left(t_{n+1-\alpha_f}\right)\mathbf{D}_1\tilde{\boldsymbol{\varepsilon}}_m\left(t_{n+1-\alpha_f}\right)+\frac{\zeta^2h^2}{4}\mathbf{B}_b^T\left(t_{n+1-\alpha_f}\right)\mathbf{D}_1\boldsymbol{\chi}\left(t_{n+1-\alpha_f}\right)\\&+\tilde{\mathbf{B}}_\gamma^T\left(t_{n+1-\alpha_f}\right)\mathbf{D}_2\tilde{\boldsymbol{\gamma}}\left(t_{n+1-\alpha_f}\right)\Big]dV\end{aligned} \tag{68}$$

$$\begin{aligned}\mathbf{k}_T\left(t_{n+1-\alpha_f}\right)=\int_V\Bigg[&\tilde{\mathbf{B}}_m^T\left(t_{n+1-\alpha_f}\right)\mathbf{D}_1\tilde{\mathbf{B}}_m\left(t_{n+1-\alpha_f}\right)+\frac{\zeta^2h^2}{4}\mathbf{B}_b^T\left(t_{n+1-\alpha_f}\right)\mathbf{D}_1\mathbf{B}_b\left(t_{n+1-\alpha_f}\right)\\&+\frac{\partial\tilde{\mathbf{B}}_m^T}{\partial\mathbf{u}_L^T}\Big|_{t_{n+1-\alpha_f}}\mathbf{D}_1\tilde{\boldsymbol{\varepsilon}}_m\left(t_{n+1-\alpha_f}\right)+\tilde{\mathbf{B}}_\gamma^T\left(t_{n+1-\alpha_f}\right)\mathbf{D}_2\tilde{\mathbf{B}}_\gamma\left(t_{n+1-\alpha_f}\right)\Bigg]dV\end{aligned} \tag{69}$$

In Eqs. (68) and (69), the values of $\tilde{\boldsymbol{\varepsilon}}_m$, $\boldsymbol{\chi}$, $\tilde{\boldsymbol{\gamma}}$ and their first- and second-order partial derivatives with respect to the local nodal variables $\mathbf{u}_L$ at the generalized midpoints $t_{n+1-\alpha_f}$ are interpolated by using their counterparts at two endpoints of the current time step $\left[t_n,t_{n+1}\right]$ as following:

$$\begin{cases}\tilde{\boldsymbol{\varepsilon}}_m\left(t_{n+1-\alpha_f}\right)=\left(1-\alpha_f\right)\tilde{\boldsymbol{\varepsilon}}_m\left(t_{n+1}\right)+\alpha_f\tilde{\boldsymbol{\varepsilon}}_m\left(t_n\right)\\ \boldsymbol{\chi}\left(t_{n+1-\alpha_f}\right)=\left(1-\alpha_f\right)\boldsymbol{\chi}\left(t_{n+1}\right)+\alpha_f\boldsymbol{\chi}\left(t_n\right)\\ \tilde{\boldsymbol{\gamma}}\left(t_{n+1-\alpha_f}\right)=\left(1-\alpha_f\right)\tilde{\boldsymbol{\gamma}}\left(t_{n+1}\right)+\alpha_f\tilde{\boldsymbol{\gamma}}\left(t_n\right)\\ \tilde{\mathbf{B}}_m\left(t_{n+1-\alpha_f}\right)=\left(1-\alpha_f\right)\tilde{\mathbf{B}}_m\left(t_{n+1}\right)+\alpha_f\tilde{\mathbf{B}}_m\left(t_n\right)\\ \mathbf{B}_b\left(t_{n+1-\alpha_f}\right)=\left(1-\alpha_f\right)\mathbf{B}_b\left(t_{n+1}\right)+\alpha_f\mathbf{B}_b\left(t_n\right)\\ \tilde{\mathbf{B}}_\gamma\left(t_{n+1-\alpha_f}\right)=\left(1-\alpha_f\right)\tilde{\mathbf{B}}_\gamma\left(t_{n+1}\right)+\alpha_f\tilde{\mathbf{B}}_\gamma\left(t_n\right)\\ \frac{\partial\tilde{\mathbf{B}}_m^T}{\partial\mathbf{u}_L^T}\Big|_{t_{n+1-\alpha_f}}=\left(1-\alpha_f\right)\frac{\partial\tilde{\mathbf{B}}_m^T}{\partial\mathbf{u}_L^T}\Big|_{t_{n+1}}+\alpha_f\frac{\partial\tilde{\mathbf{B}}_m^T}{\partial\mathbf{u}_L^T}\Big|_{t_n}\end{cases} \tag{70a–g}$$

where, $\tilde{\boldsymbol{\varepsilon}}_m$, $\tilde{\boldsymbol{\gamma}}$ and their first and second-order partial derivatives with respect to the local nodal variables $\mathbf{u}_L$ at two endpoints of the current time step $\left[t_n,t_{n+1}\right]$ are calculated as in the reference [29] (see Eqs. (55)–(68) and the text from Line 8 on page 191 to Line 20 on page 192).

4 Numerical Examples

To evaluate the computational accuracy and stability of the proposed theory, multi-body dynamics analyses are conducted on four benchmark problems: an L-shaped plate, a ruler-shaped plate, a hemispherical shell with an 18° opening at the top, and a non-smooth shell formed by three intersecting plates. The numerical results are subsequently compared with reference data from [66,72–77]. All parameters are reported in SI units.

4.1 L-Shaped Plate

The geometric description, mesh discretization and load configuration for the L-shaped plate are presented in Fig. 2. The plate has thickness $h = 0.1$ m, density $\rho = 1\ \mathrm{kg/m^3}$, Young's modulus $E = 1 \times 10^6$ Pa, and Poisson's ratio $v = 0.3$. The plate is discretized respectively using three different mesh densities: 30 elements ($N_e = 2^1$), 120 elements ($N_e = 2^2$), and 480 elements ($N_e = 2^3$), where N_e represents the number of elements along the short edge. The simulation uses a time step $\Delta t = 1 \times 10^{-4}$ s and the applied nodal loads are given by:

$$P_1 = \begin{Bmatrix} -3 \\ 0 \\ 1 \end{Bmatrix} f(t) \quad P_2 = \begin{Bmatrix} 0 \\ 3 \\ 0 \end{Bmatrix} f(t) \quad P_3 = \begin{Bmatrix} 0 \\ 0 \\ -1 \end{Bmatrix} f(t)$$

$$P_4 = \begin{Bmatrix} 3 \\ 0 \\ 0 \end{Bmatrix} f(t) \quad P_5 = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} f(t) \quad P_6 = \begin{Bmatrix} -3 \\ 0 \\ 0 \end{Bmatrix} f(t)$$

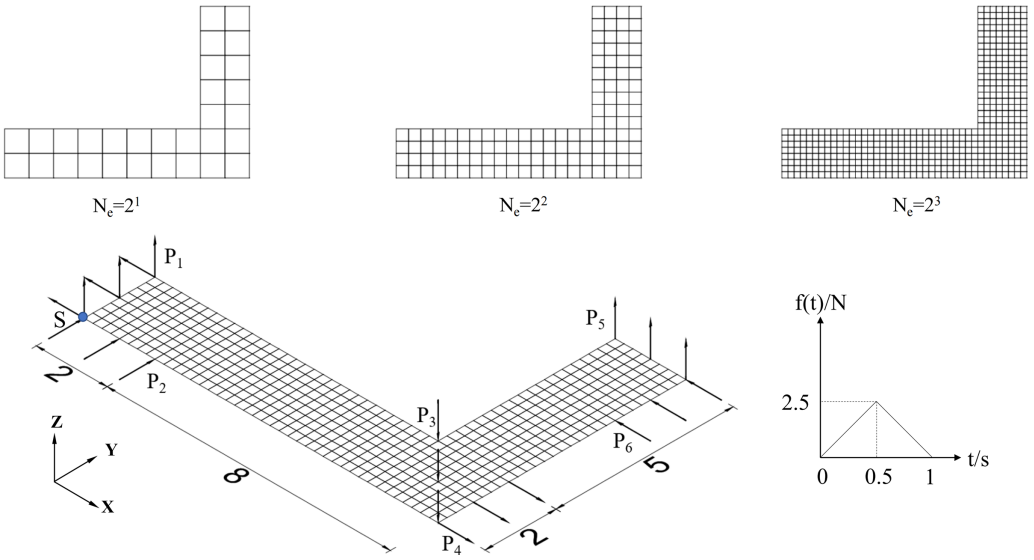

Figure 2: Mesh discretization and load configuration of the L-shaped plate.

The L-shaped plate is unconstrained along all boundaries and exhibits low stress levels, indicating that strength-related issues are negligible. Fig. 3a,b presents the time histories of the system's total energy and linear momentum, respectively, for the 480-element model. After the external load is removed at t = 1 s, the results demonstrate excellent conservation properties. The total energy curve is compared with that reported by Lavrenčič and Brank [73], who employed a quadrilateral shell element based on a mixed variational

formulation along with an energy–momentum conservation algorithm. The close agreement between the two sets of results validates the accuracy of the present numerical method.

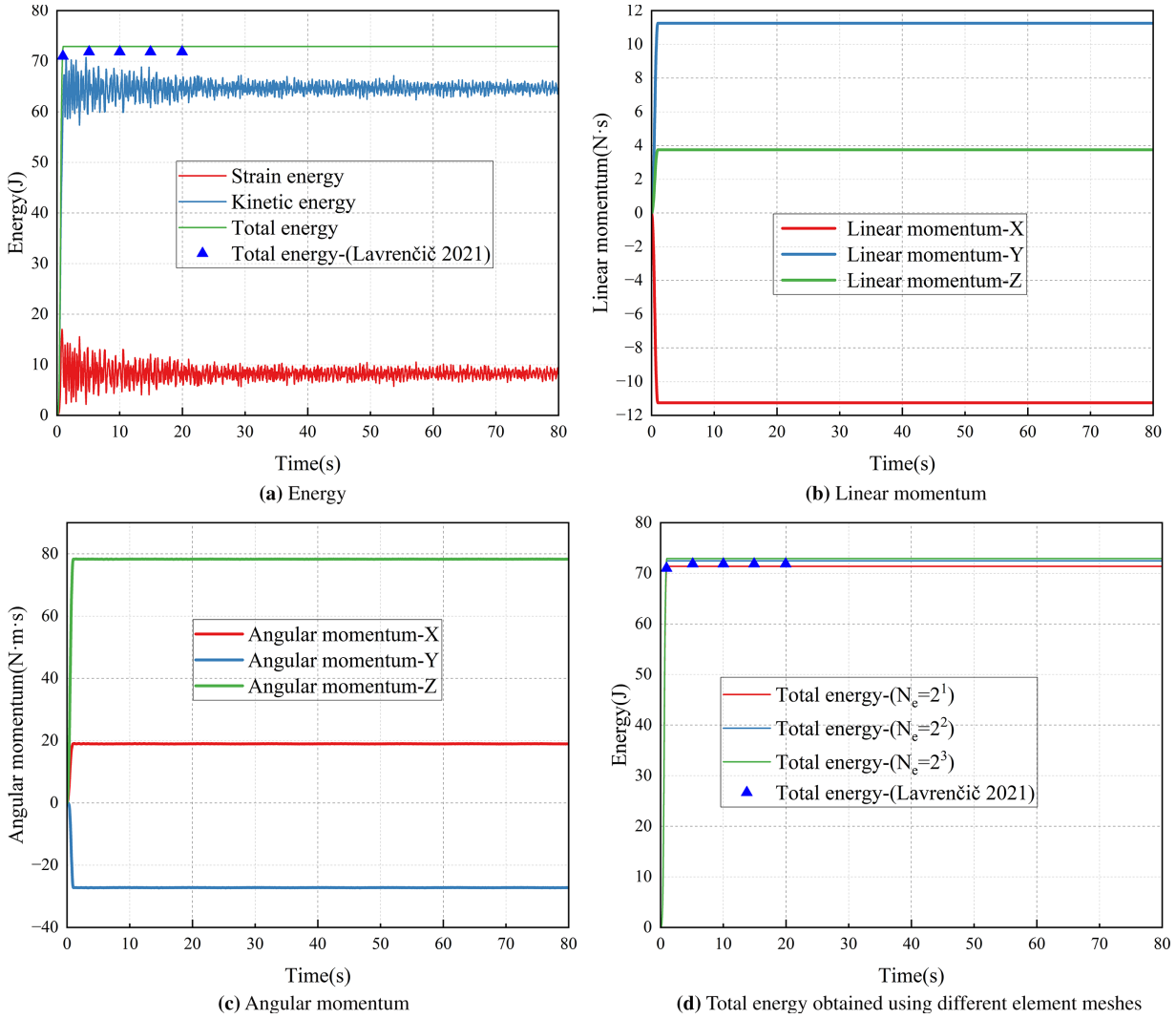

Figure 3: Time-history curves for the L-shaped plate.

Fig. 3c presents the time history curves of the system's angular momentum, which remain constant after the removal of the external load, thereby demonstrating the conservation properties of the proposed algorithm with respect to momentum. Fig. 3d compares the time history curves of the system's total energy obtained respectively with three different element mesh densities (N_e = 2^1, 2^2, 2^3), demonstrating good convergence of the proposed algorithm.

Fig. 4a–c presents the time history curves of displacements at point S on the L-shaped plate along the *X*, *Y*, and *Z* axes, respectively, obtained using three different meshes. It is shown that even with a coarse mesh, very accurate results can be achieved.

Table 1 reports the values of the system's total energy after load removal for five different element mesh densities (N_e = 2^1, 2^2, 2^3, 2^4, 2^5).

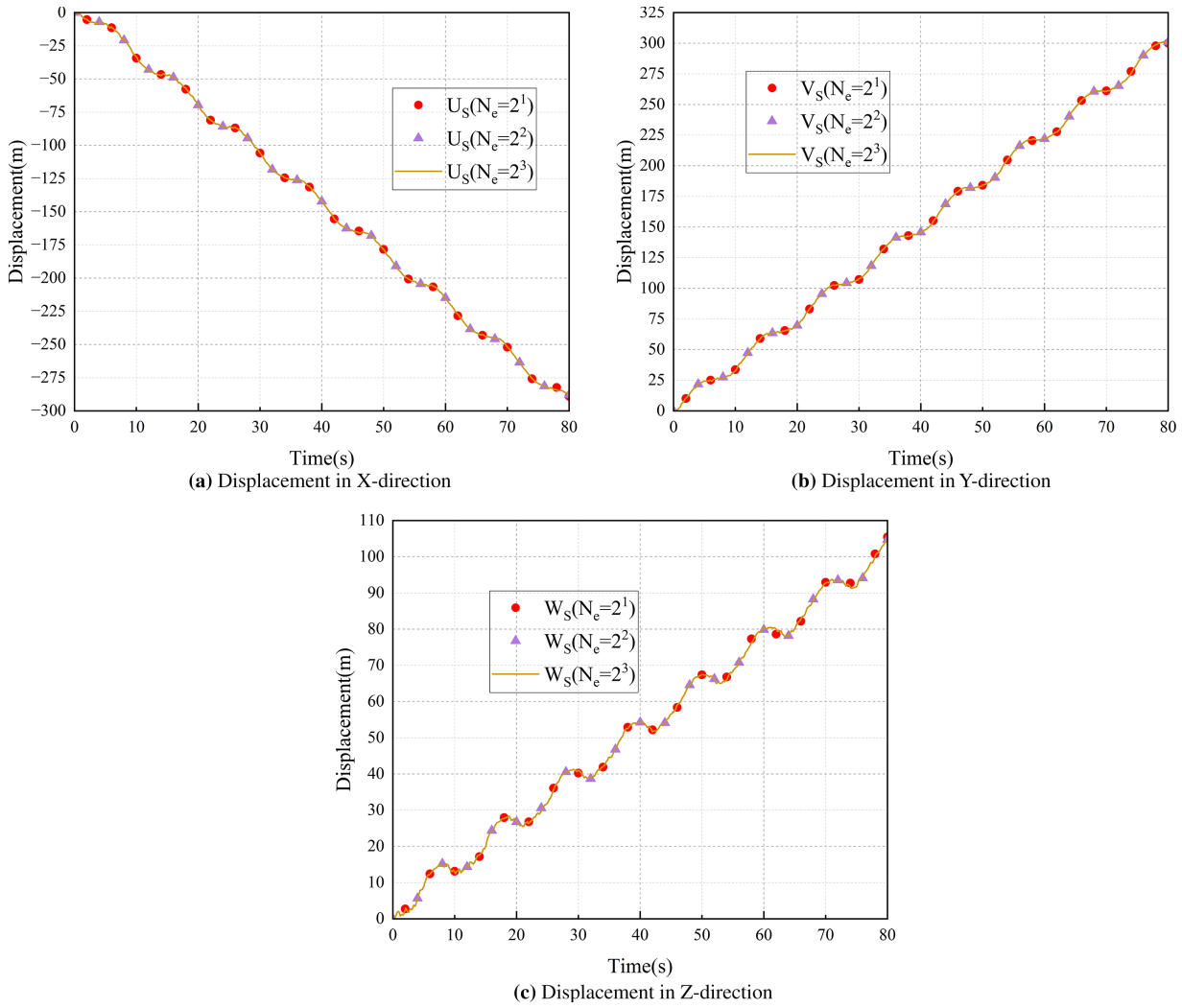

Figure 4: Time-history curves of displacements at Point S for different element meshes.

Table 1: System's total energy of the L-shaped plate for five different element mesh densities.

Number of Elements	Total System Energy after Load Removal (J)	Relative Error: $\lvert E_N - E_{ref}\rvert / E_{ref}$
30	71.373	2.586%
120	72.455	1.110%
480	72.908	0.491%
1920	73.135	0.182%
7680	73.268	0.000%

Defining the relative error as $|E_N - E_{ref}|/E_{ref}$, where E_N is the value obtained respectively from five different element meshes, E_{ref} is the value obtained from the finest mesh of 7680 elements ($N_e = 2^5$). The convergence behavior is also evaluated on a log-log scale (Fig. 5). Linear regression yields a slope of −1.27,

with a coefficient of determination R^2 = 0.998, confirming a convergence rate of approximately 1.27 in the energy norm.

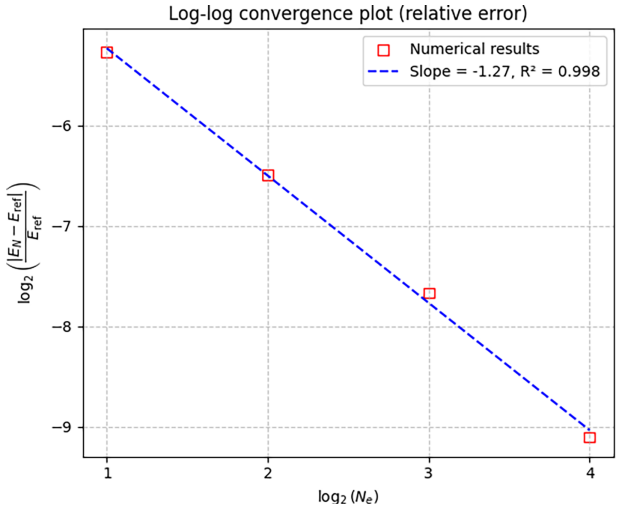

Figure 5: Convergence analysis in log-log scale.

After time $t = 1\,s$, the external impulsive load is removed, and the plate subsequently enters a phase of free motion. During this phase, kinetic energy and strain energy are continuously exchanged within the system. Fig. 6 illustrates the spatial tumbling trajectory of the L-shaped plate over a 20-s interval, with consecutive frames shown at 1-s increments. The plate undergoes large-displacement tumbling motion, accompanied by significant bending and twisting deformations.

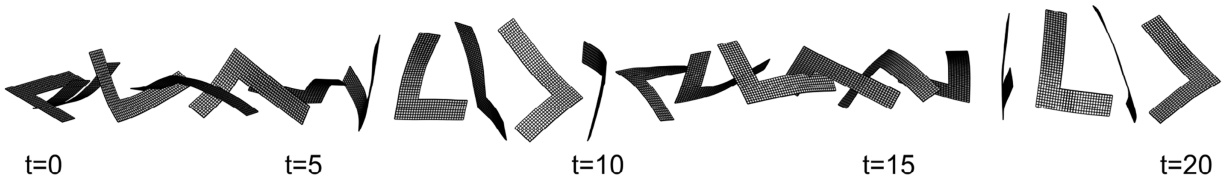

Figure 6: L-shaped plate's motion trajectory.

Table 2 compares the CPU times for solving the discretized structural dynamic equations using two approaches for the assembled equivalent tangent stiffness matrix. The first approach exploits symmetry and sparsity via one-dimensional storage and LDLT decomposition; the second uses full-matrix storage and Gaussian elimination, ignoring these properties (as required for asymmetric matrices). The former requires substantially less CPU time than the latter, demonstrating that our proposed multibody dynamics formulation—which yields symmetric equivalent mass and global tangent stiffness matrices—is computationally far more efficient than conventional co-rotational formulations with asymmetric element-level tangent stiffness matrices.

Table 2: Comparison of computational efficiency: the L-shaped plate.

Comparison Aspect	One-Dimensional Storage + LDLT Decomposition			Full-Matrix Storage + Gaussian Elimination		
	2 × 15	4 × 30	8 × 60	2 × 15	4 × 30	8 × 60
CPU time per 100 steps	0.774 s	4.143 s	13.460 s	2.050 s	62.047 s	10,436.720 s
CPU Time 800,000 steps	6191.219 s (1.720 h)	33,142.310 s (9.206 h)	107,678.010 s (29.911 h)	16,398.470 s (4.555 h)	496,376 s* (137.882 h*)	83,493,760 s* (23,192.711 h*)

Note: Values marked with (*) are estimated from the CPU time required for 100 time steps. Full-matrix computation of the equivalent tangent stiffness matrix for 800,000 time steps would result in prohibitively long CPU times; thus, only estimated values are reported.

4.2 Ruler-Shaped Plate

The geometric description, mesh discretization and load configuration of the ruler-shaped plate are presented in Fig. 7. The plate has thickness $h = 0.002$ m, density $\rho = 7800\,\text{kg/m}^3$, Young's modulus $E = 2.06 \times 10^{11}$ Pa, and Poisson's ratio $v = 0$. The time step for simulation is $\Delta t = 5 \times 10^{-6}$ s. A set of linearly time-varying uniformly distributed line loads act on the edges and interior of the strip plate.

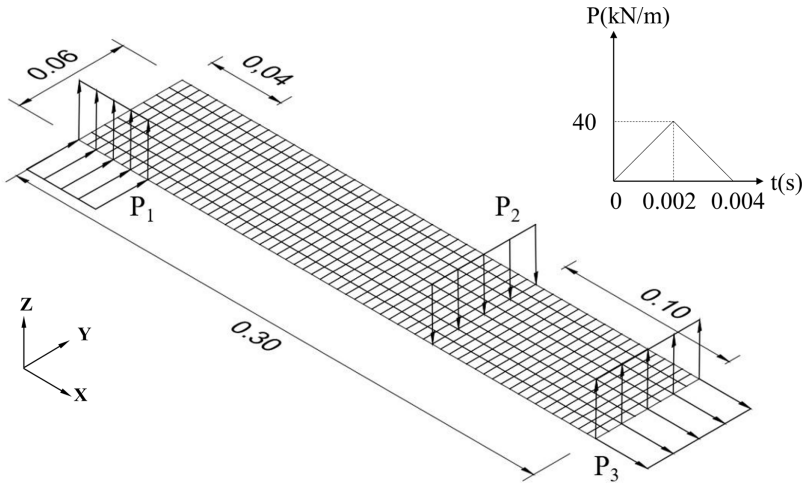

Figure 7: Mesh discretization and load configuration of the ruler-shaped plate.

A medium mesh density featuring an 8 × 60 element configuration (480 elements and 549 nodes) is employed in the simulation. The time-history curves of the system's strain energy, kinetic energy, total energy, linear momentum, and angular momentum are presented in Fig. 8a–c, respectively. As illustrated, these quantities remain nearly exactly conserved throughout the simulation and show excellent agreement with the reference solutions reported by Kuhl and Ramm [72] and Yang and Xia [66].

Table 3 presents the system's total energy after load removal, computed using different mesh densities. From the coarse mesh to the medium and fine meshes, the maximum variation in the system's total energy is only 0.03 J, corresponding to a relative error of approximately 0.012%. As the mesh density increases, the system's total energy changes marginally and approaches a stable value, indicating that the numerical solution has essentially converged.

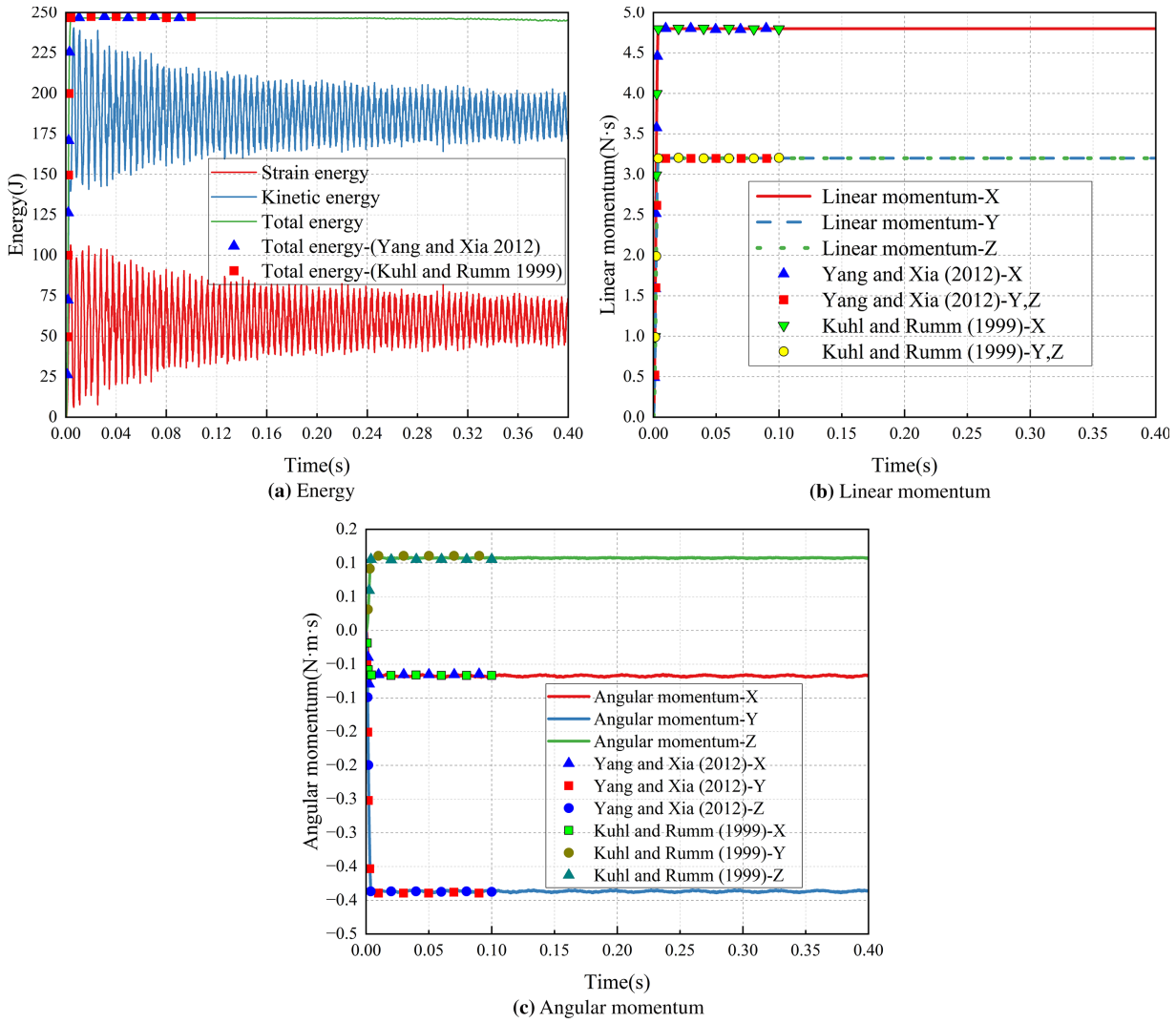

Figure 8: Time-history curves for the ruler-shaped plate.

Table 3: The system's total energy of the ruler-shaped plate with different mesh densities.

Element Meshes	Total Energy after Load Removal (J)	Relative Error as $\lvert E_N - E_{ref} \rvert / E_{ref}$
4 × 30	246.57	0.012%
8 × 60	246.58	0.008%
16 × 120	246.60	0.000%

Fig. 9 presents the motion trajectory of the ruler-shaped plate within the time duration [0, 0.4 s], with consecutive frames interval of 0.01 s.

Table 4 compares the CPU times for solving the discretized structural dynamic equations of the ruler-shaped plate using two approaches for the assembled equivalent tangent stiffness matrix. The first approach exploits symmetry and sparsity via one-dimensional storage and LDLT decomposition; the second uses full-matrix storage and Gaussian elimination, ignoring these properties (as required for asymmetric matrices).

Figure 9: Ruler-shaped plate's motion trajectory.

Table 4: Comparison of computational efficiency: the ruler-shaped plate.

Comparison Aspect	One-Dimensional Storage + LDLT Decomposition		Full-Matrix Storage + Gaussian Elimination	
	4 × 30	8 × 60	4 × 30	8 × 60
CPU time per 100 steps	3.381 s	15.145 s	134.828 s	10,621.090 s
CPU Time 80,000 steps	2704.891 s (0.751 h)	12,116.310 s (3.366 h)	107,862 s* (29.962 h*)	8,496,872 s* (2360.242 h*)

Note: Values marked with (*) are estimated from the CPU time required for 100 time steps. Full-matrix computation of the equivalent tangent stiffness matrix for 80,000 time steps would result in prohibitively long CPU times; thus, only estimated values are reported.

4.3 Hemispherical Shell with an 18° Opening at the Top

Fig. 10 illustrates the geometry, mesh discretization, and loading configuration of a hemispherical shell featuring an 18° opening at the apex. The shell is characterized by radius R_{rad} = 10 m, thickness h = 0.04 m, density $\rho = 1 \times 10^{-3}$ kg/m^3, Young's modulus $E = 6.825 \times 10^7$ Pa, and Poisson's ratio ν = 0.3. The simulation time step is set to $\Delta t = 1.25 \times 10^{-4}$ s. Under symmetric loading condition, the model exhibits pronounced large-displacement response. This benchmark problem has also been investigated in the literature [74,75]. In the present study, the shell is discretized using 1024 elements with 1088 nodes.

Fig. 11 presents the time-history curves of the system's strain energy, kinetic energy, and total energy over the time interval [0, 2.5 s], along with a detailed view of the energy response within [0, 0.1 s]. As shown, following the removal of the external load at t = 0.03 s, the structure enters a state of free vibration. During this phase, kinetic energy and strain energy are continuously exchanged, while the total energy remains constant, thereby satisfying the conservation condition of the system.

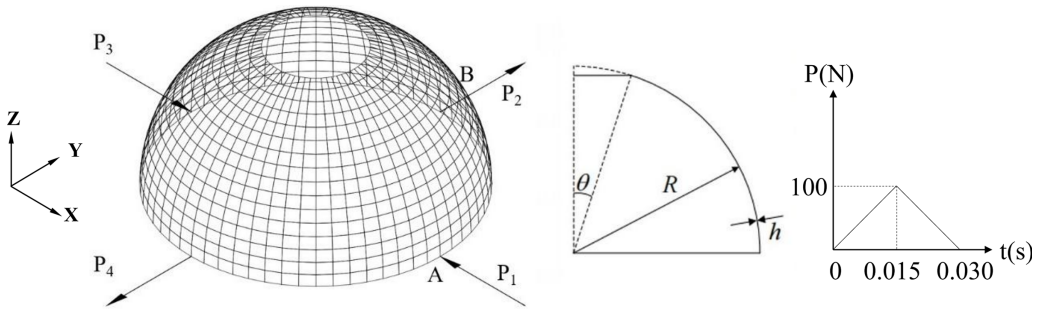

Figure 10: Mesh discretization and load configuration of the hemispherical shell with an 18° opening at the top.

Fig. 12 illustrates the time-history curves of the displacements at Point A (X-direction) and Point B (Y-direction) over the interval [0, 2.5 s]. The maximum and minimum displacement magnitudes at both nodes are identical, and both displacement responses exhibit periodic variations. Furthermore, the simulation remains numerically stable throughout the entire time domain.

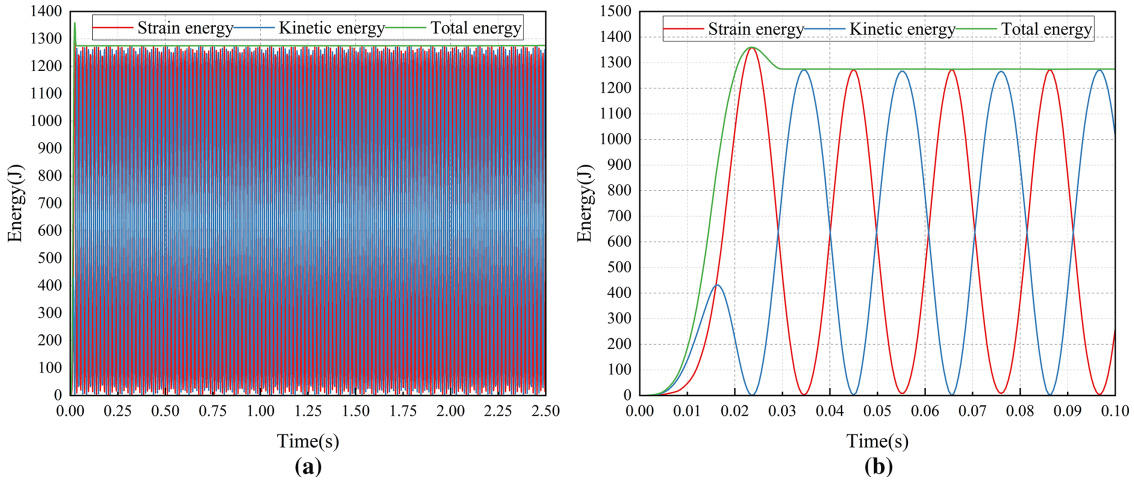

Figure 11: Time-history curves of energy for the hemispherical shell with an opening: (**a**) [0, 2.5 s]; (**b**) [0, 0.1 s].

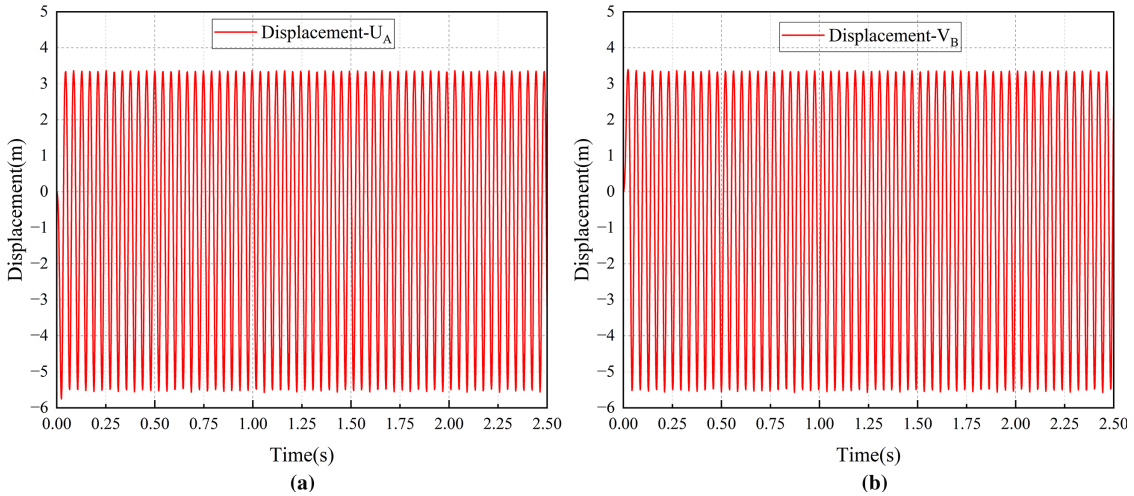

Figure 12: Time-history curves of the displacements at Point A and B of the hemispherical shell with an opening: (**a**) in X-direction at point A; (**b**) in Y-direction at point B.

Table 5 presents the system's total energy after load removal with different mesh densities. As the discretization is refined, the total system energy asymptotically approaches a stable value.

Table 5: The system's total energy of the hemispherical shell with different mesh densities.

Number of Elements	Total Energy after Load Removal (J)	Relative Error as $\lvert E_N - E_{ref} \rvert / E_{ref}$
1024	1274.9	0.786%
4096	1283.5	0.117%
16,384	1285.0	0.000%

Fig. 13 depicts the vibration sequence of the hemispherical shell within [0, 0.11 s], with snapshots taken at 0.01 s intervals.

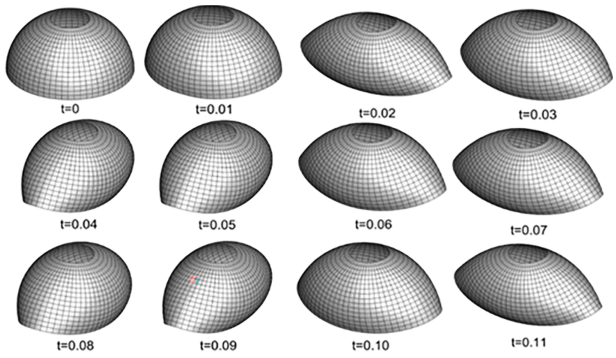

Figure 13: Vibration sequence of the hemispherical shell with an opening.

Table 6 compares the CPU times for solving the discretized structural dynamic equations of the hemispherical shell with an opening using two approaches for the assembled equivalent tangent stiffness matrix. The first approach exploits symmetry and sparsity via one-dimensional storage and LDLT decomposition; the second uses full-matrix storage and Gaussian elimination, ignoring these properties (as required for asymmetric matrices).

Table 6: Comparison of computational efficiency: the hemispherical shell with an opening.

Comparison Aspect	One-Dimensional Storage + LDLT Decomposition	Full-Matrix Storage + Gaussian Elimination
	1024	**1024**
CPU time per 10 steps	5.145 s	9034.172 s
CPU Time 20,000 steps	10,289.200 s (2.858 h)	18,068,344 s* (5018.984 h*)

Note: Values marked with (*) are estimated from the CPU time required for 10 time steps. Full-matrix computation of the equivalent tangent stiffness matrix for 20,000 time steps would result in prohibitively long CPU times; thus, only estimated values are reported.

4.4 Three Intersecting Plates

The configuration of the three intersecting plates is shown in Fig. 14. The plate has thickness $h = 0.02\,\mathrm{m}$, density $\rho = 1\,\mathrm{kg/m^3}$, Young's modulus $E = 2 \times 10^7$ Pa, and Poisson's ratio $\nu = 0.25$. The three intersecting plates is discretized into 432 elements (element meshes $28 \times 6 + 20 \times 6 + 18 \times 8$) with 507 nodes. The simulation uses a time step of $\Delta t = 5 \times 10^{-5}$ s, and the uniformly distributed loads (see Fig. 14) are given by:

$$\mathrm{P}_1 = \begin{Bmatrix} 2 \\ 0 \\ 2 \end{Bmatrix} q(\mathrm{t}) \quad \mathrm{P}_2 = \begin{Bmatrix} -2 \\ -4 \\ -2 \end{Bmatrix} q(\mathrm{t}) \quad \mathrm{P}_3 = \begin{Bmatrix} 10 \\ 6 \\ -1 \end{Bmatrix} q(\mathrm{t}) \quad \mathrm{P}_4 = \begin{Bmatrix} -1 \\ 1 \\ 1 \end{Bmatrix} q(\mathrm{t})$$

where

$$q(t) = \begin{cases} 0.5t & 0 \le t < 0,5 \\ 0.5 - 0.5t & 0.5 \le t < 1.0 \\ 0 & t \ge 1.0 \end{cases}$$

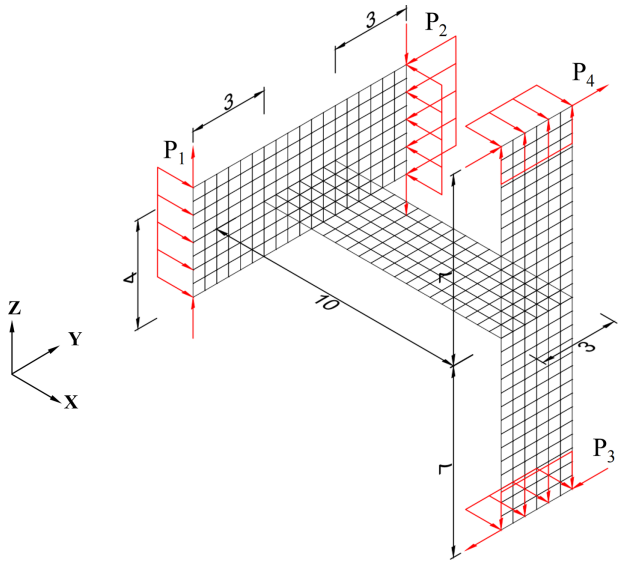

Figure 14: Mesh discretization of three intersecting plates.

Fig. 15 presents the time-history curves of the system energy for the three intersecting plates. It is observed that the total energy of the system remains conservative after the removal of the external load. Zhang et al. [76] and Chróścielewski and Witkowski [77] also analyzed this model using a geometrically exact shell element formulation, implemented with an energy–momentum conserving algorithm. The total energy obtained in the present study is compared with that reported by Chróścielewski and Witkowski [77], thereby validating the accuracy of the current results.

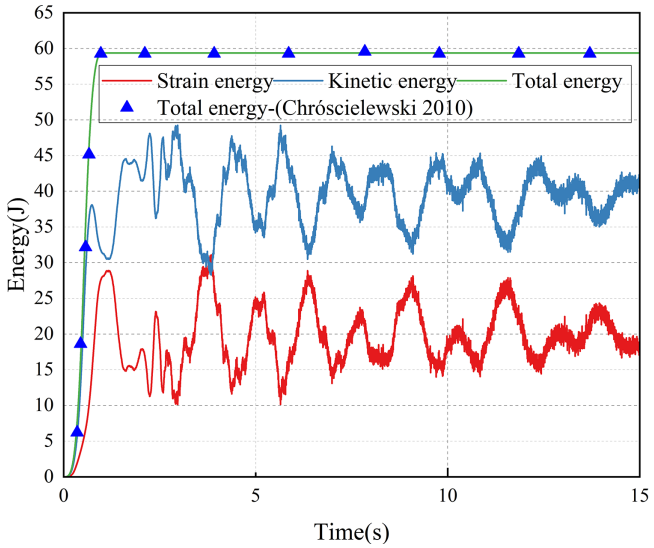

Figure 15: Time-history curves of the system energy for the three intersecting plates.

The time-history curves of the linear and angular momenta for the three intersecting plates are presented in Fig. 16a,b, respectively. The total linear momentum exhibits near-exact conservation following load removal. The total angular momentum, however, while approximately conserved, exhibits small oscillations throughout its time history.

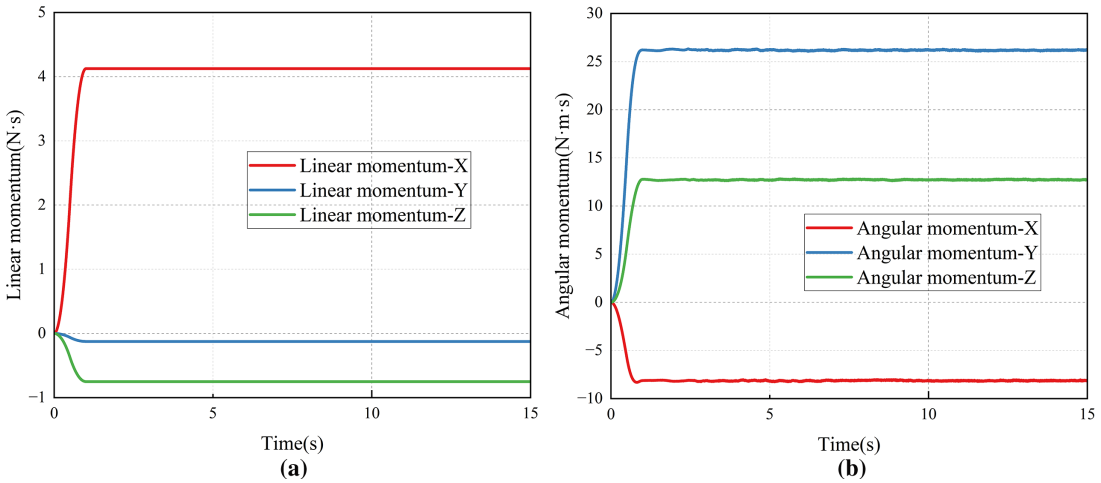

Figure 16: Time-history curves for the three intersecting plates: (**a**) Linear momentum; (**b**) Angular momentum.

Table 7 presents the total energy of the three intersecting plates upon load removal for different mesh densities.

Table 7: The system's total energy of the three intersecting plates with different mesh densities.

Number of Elements	Total Energy after Load Removal (J)	Relative Error as $\lvert E_N - E_{ref} \rvert / E_{ref}$
432	59.375	0.210%
1728	59.455	0.076%
6912	59.500	0.000%

Fig. 17 shows the motion trajectories of the three intersecting plates over a period of 14 s, with a time interval of 1 s between adjacent frames. The three intersecting plates undergo large displacements and large rotations in space.

Table 8 compares the CPU times for solving the discretized structural dynamic equations of the three intersecting plates using respectively two approaches for the assembled equivalent tangent stiffness matrix. The first approach exploits symmetry and sparsity via one-dimensional storage and LDLT decomposition; the second uses full-matrix storage and Gaussian elimination, ignoring these properties (as required for asymmetric matrices).

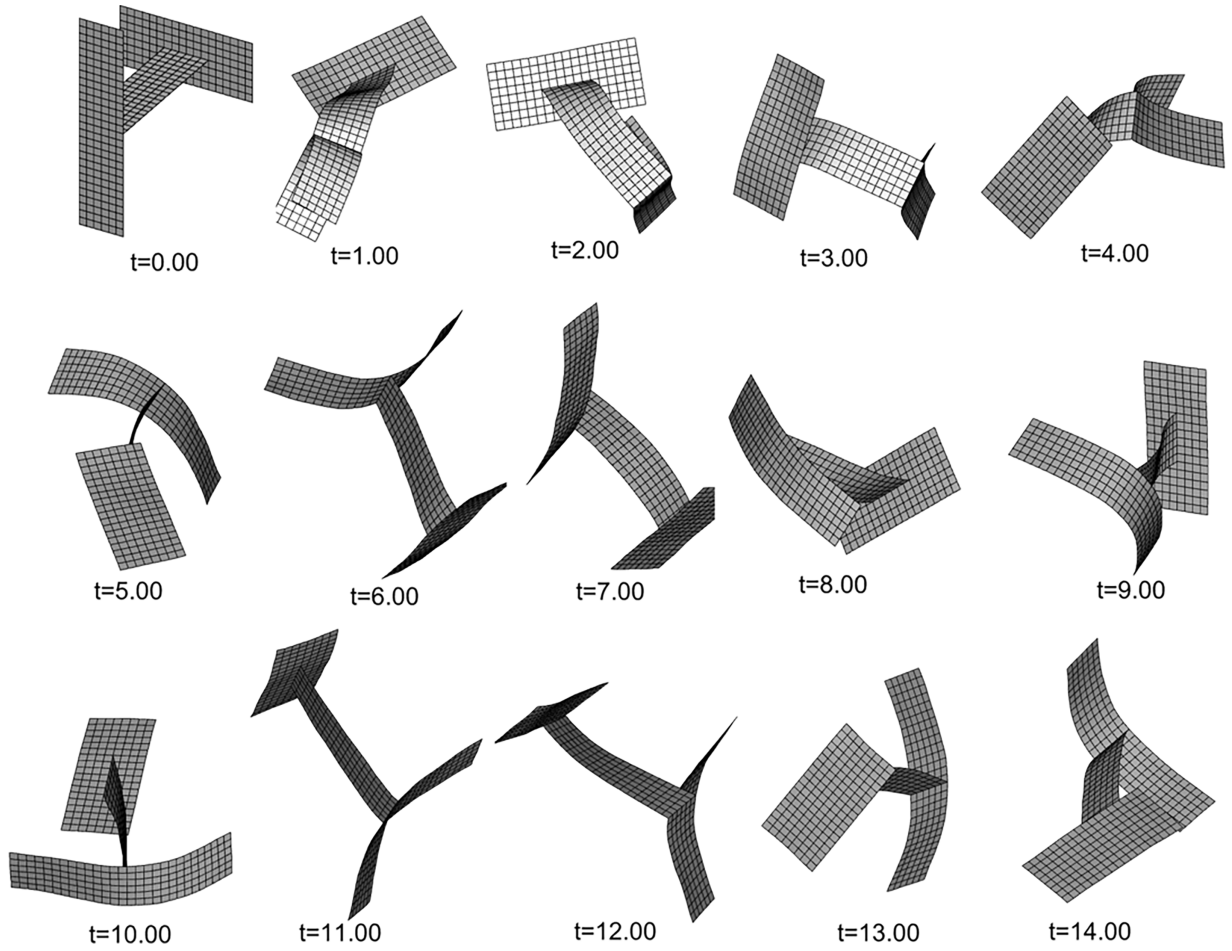

Figure 17: Motion trajectories of the three intersecting plates.

Table 8: Comparison of computational efficiency: the three intersecting plates.

Comparison Aspect	One-Dimensional Storage + LDLT Decomposition	Full-Matrix Storage + Gaussian Elimination
	432	**432**
CPU time per 10 steps	13.351 s	668.344 s
CPU Time 300,000 steps	40,053.520 s (11.126 h)	2,005,032 s* (556.953 h*)

Note: Values marked with (*) are estimated from the CPU time required for 10 time steps. Full-matrix computation of the equivalent tangent stiffness matrix for 300,000 time steps would result in prohibitively long CPU times; thus, only estimated values are reported.

5 Conclusions

This paper has proposed a novel computational framework for multibody dynamics capable of analyzing the nonlinear dynamic behavior of flexible multibody systems undergoing large displacements and large rotations. The core contribution lies in integrating a co-rotational four-node quadrilateral shell element—formulated using incrementally additive vectorial rotational variables—with a quasi-energy–momentum conserving algorithm. This integration effectively resolves the issue of asymmetric tangent stiffness matrices inherent in conventional co-rotational formulations. Specifically, during the discretization of the dynamic equilibrium differential equations, the term in the inertial forces containing the first-order time derivatives of the rotational variables is incorporated into the equivalent load vector. This treatment yields a symmetric mass matrix and a symmetric equivalent tangent stiffness matrix, thereby enhancing computational performance and reducing storage requirements.

The accuracy and robustness of the proposed method have been validated through numerical simulations of four benchmark problems: an L-shaped plate, a ruler-shaped plate, a hemispherical shell with an 18° opening at the top, and a non-smooth shell-three intersecting plates. The numerical results show excellent agreement with data reported in the literature, demonstrating the method's superior performance in computational accuracy, efficiency, and numerical stability.

In this work, the term "multibody systems" is not intended in its literal sense to refer exclusively to systems composed of multiple distinct bodies. Instead, it is adopted in a broader context to denote systems that are free of boundary constraints and capable of undergoing large translational motions as well as arbitrarily large rotations under dynamic loading. Such systems may comprise either a single body or multiple bodies. The present study specifically addresses smooth and non-smooth shell structures considered as single-body systems. In this regard, numerous related studies on multibody dynamics cited in the references below also primarily employ single-body systems as numerical examples. The extension of the proposed formulation to multibody dynamics problems involving non-conservative loads, or to multibody systems consisting of multiple bodies interconnected by joints, is left as a direction for future research.

Acknowledgement: None.

Funding Statement: This research was funded by National Natural Science Foundation of China (No. 11672266) and and the Fundamental Research Funds for the Central Universities, China (No. 343593).

Author Contributions: Zhongxue Li: Conceptualization (lead); Methodology (lead); Software (lead); Writing—original draft (supporting); Writing—review & editing (equal). Yu Peng: Investigation (supporting)—numerical examples; Writing—original draft (supporting). Jin Xu: Investigation (supporting)—numerical examples; Writing—original draft (supporting). Loc Vu-Quoc: Methodology (supporting); Validation (supporting); Writing—review & editing (equal). Bassam A. Izzuddin: Methodology (supporting); Validation (supporting); Writing—review & editing (equal). All authors: Engaged in critical discussions regarding the theoretical development and contributed to the final manuscript. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wempner G. Finite elements, finite rotations and small strains of flexible shells. Int J Solids Struct. 1969;5(2):117–53. doi:10.1016/0020-7683(69)90025-0.
2. Belytschko T, Hsieh BJ. Non-linear transient finite element analysis with convected co-ordinates. Int J Numer Methods Eng. 1973;7(3):255–71. doi:10.1002/nme.1620070304.
3. Belytschko T, Hsieh BJ. Nonlinear transient analysis of shells and solids of revolution by convected elements. AIAA J. 1974;12(8):1031–5. doi:10.2514/3.49406.
4. Wang G, Qi ZH, Xu JS. A high-precision co-rotational formulation of 3D beam elements for dynamic analysis of flexible multibody systems. Comput Methods Appl Mech Eng. 2020;360(6):112701. doi:10.1016/j.cma.2019.112701.
5. Rong YF, Sun F, Sun Q, Liang K. Geometrically nonlinear analysis utilizing co-rotational framework for solid element based on modified Hellinger-Reissner principle. Comput Mech. 2023;71(1):127–42. doi:10.1007/s00466-022-02229-z.
6. Chen L, Liu SW, Bai R, Chan SL. Co-rotational formulations for geometrically nonlinear analysis of beam-columns including warping and Wagner effects. Int J Str Stab Dyn. 2023;23(5):2350052. doi:10.1142/s0219455423500529.
7. Han QH, Wu C, Liu MJ, Wu H. Corotational isogeometric shear deformable geometrically exact spatial form beam element for general large deformation analysis of flexible thin-walled beam structures. Thin Walled Struct. 2024;198(7):111684. doi:10.1016/j.tws.2024.111684.
8. Grange S, Bertrand D. Co-rotational 3D beam element using quaternion algebra to account for large rotations: formulation theory and static applications. Int J Solids Struct. 2024;293(13):112746. doi:10.1016/j.ijsolstr.2024.112746.
9. Kan ZY, Deng J, Li YT, Song XG. A comprehensive study and insight in co-rotational approach based geometrically nonlinear analysis with planar solid elements. Int J Numer Methods Eng. 2025;126(19):e70106. doi:10.1002/nme.70106.
10. Rankin CC, Brogan FA. An element independent corotational procedure for the treatment of large rotations. J Press Vessel Technol. 1986;108(2):165–74. doi:10.1115/1.3264765.
11. Rankin CC, Nour-Omid B. The use of projectors to improve finite element performance. Comput Struct. 1988;30(1–2):257–67. doi:10.1016/0045-7949(88)90231-3.
12. Crisfield MA, Moita GF. A co-rotational formulation for 2-D continua including incompatible modes. Int J Numer Methods Eng. 1996;39(15):2619–33. doi:10.1002/(sici)1097-0207(19960815)39:15<2619::aid-nme969>3.0.co;2-n.
13. Crisfield MA, Moita GF. A unified co-rotational framework for solids, shells and beams. Int J Solids Struct. 1996;33(20–22):2969–92. doi:10.1016/0020-7683(95)00252-9.
14. Moita GF, Crisfield MA. A finite element formulation for 3-D continua using the co-rotational technique. Int J Numer Methods Eng. 1996;39(22):3775–92. doi:10.1002/(sici)1097-0207(19961130)39:22<3775::aid-nme23>3.0.co;2-w.
15. Crisfield MA, Galvanetto U, Jelenić G. Dynamics of 3-D co-rotational beams. Comput Mech. 1997;20(6):507–19. doi:10.1007/s004660050271.
16. Masud A, Tham CL. Three-dimensional corotational framework for elasto-plastic analysis of multilayered composite shells. AIAA J. 2000;38(12):2320–7. doi:10.2514/2.901.
17. Masud A, Tham CL, Liu WK. A stabilized 3-D co-rotational formulation for geometrically nonlinear analysis of multi-layered composite shells. Comput Mech. 2000;26(1):1–12. doi:10.1007/s004660000144.
18. Argyris JH, Papadrakakis M, Karapitta L. Elasto-plastic analysis of shells with the triangular element TRIC. Comput Methods Appl Mech Eng. 2002;191(33):3613–36. doi:10.1016/S0045-7825(02)00308-0.
19. Kan ZY, Dong KJ, Chen BS, Peng HJ, Song XG. The direct force correction based framework for general co-rotational analysis. Comput Methods Appl Mech Eng. 2021;385:114018. doi:10.1016/j.cma.2021.114018.
20. Crisfield MA, Shi J. An energy conserving co-rotational procedure for non-linear dynamics with finite elements. Nonlinear Dyn. 1996;9(1):37–52. doi:10.1007/BF01833292.
21. Wu TK, Liu ZY, Ma ZQ. Nonlinear static and dynamic analysis of corotational shell formulated on the special Euclidean group SE(3). Nonlinear Dyn. 2024;112(17):14773–803. doi:10.1007/s11071-024-09633-5.

22. Wang BY, Liu ZY, Ma ZQ, Xu SH. A 3D corotational curved beam element formulated on SE(3) group for geometrically nonlinear static and dynamic analysis. Comput Mech. 2025;76(6):1595–620. doi:10.1007/s00466-025-02664-8.
23. Lesiv H, Izzuddin BA. Consistency and misconceptions in co-rotational 3D continuum finite elements: a zero-macrospin approach. Int J Solids Struct. 2023;281(3):112445. doi:10.1016/j.ijsolstr.2023.112445.
24. Grange S, Bertrand D. Co-rotational 3D beam element using quaternion algebra to account for large rotations: dynamic equilibrium and applications. Int J Solids Struct. 2024;302(13):112975. doi:10.1016/j.ijsolstr.2024.112975.
25. Li ZX, Izzuddin B. Application of vectorial rotational variables in large displacement analysis of structures. In: Fourth International Conference on Advances in Steel Structures. Amsterdam, The Netherlands: Elsevier; 2005. p. 1501–6. doi:10.1016/b978-008044637-0/50224-9.
26. Izzuddin BA. An enhanced co-rotational approach for large displacement analysis of plates. Int J Numer Methods Eng. 2005;64(10):1350–74. doi:10.1002/nme.1415.
27. Li ZX. A co-rotational formulation for 3D beam element using vectorial rotational variables. Comput Mech. 2007;39(3):309–22. doi:10.1007/s00466-006-0029-x.
28. Li ZX, Izzuddin BA, Vu-Quoc L. A 9-node co-rotational quadrilateral shell element. Comput Mech. 2008;42(6):873–84. doi:10.1007/s00466-008-0289-8.
29. Li ZX, Zhuo X, Vu-Quoc L, Izzuddin BA, Wei HY. A four-node corotational quadrilateral elastoplastic shell element using vectorial rotational variables. Int J Numer Methods Eng. 2013;95(3):181–211. doi:10.1002/nme.4471.
30. Li ZX, Liu YF, Izzuddin BA, Vu-Quoc L. A stabilized co-rotational curved quadrilateral composite shell element. Int J Numer Methods Eng. 2011;86(8):975–99. doi:10.1002/nme.3084.
31. Li ZX, Zheng T, Vu-Quoc L, Izzuddin BA. A 4-node co-rotational quadrilateral composite shell element. Int J Str Stab Dyn. 2016;16(9):1550053. doi:10.1142/s0219455415500534.
32. Li ZX, Li TZ, Vu-Quoc L, Izzuddin BA, Zhuo X, Fang Q. A nine-node corotational curved quadrilateral shell element for smooth, folded, and multishell structures. Int J Numer Methods Eng. 2018;116(8):570–600. doi:10.1002/nme.5936.
33. Newmark NM. A method of computation for structural dynamics. Urbana, IL, USA: University of Illinois, Engineering Experiment Station; 1959. Report No.: 43.
34. Hilber HM, Hughes TJR, Taylor RL. Improved numerical dissipation for time integration algorithms in structural dynamics. Earthq Eng Struct Dyn. 1977;5(3):283–92. doi:10.1002/eqe.4290050306.
35. Chung J, Hulbert GM. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. J Appl Mech. 1993;60(2):371–5. doi:10.1115/1.2900803.
36. Simo JC, Tarnow N, Wong KK. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. Comput Methods Appl Mech Eng. 1992;100(1):63–116. doi:10.1016/0045-7825(92)90115-Z.
37. Romero I, Armero F. Numerical integration of the stiff dynamics of geometrically exact shells: an energy-dissipative momentum-conserving scheme. Int J Numer Methods Eng. 2002;54(7):1043–86. doi:10.1002/nme.463.
38. Zhang R, Zhong H. A quadrature element formulation of an energy-momentum conserving algorithm for dynamic analysis of geometrically exact beams. Comput Struct. 2016;165(12):96–106. doi:10.1016/j.compstruc.2015.12.007.
39. Kolsti KF, Kunz DL. A time-marching collocation method based on quintic Hermite polynomials and adjustable acceleration and jerk constraints. Int J Numer Methods Eng. 2014;99(8):547–65. doi:10.1002/nme.4681.
40. Belytschko T, Schoeberle DF. On the unconditional stability of an implicit algorithm for nonlinear structural dynamics. J Appl Mech. 1975;42(4):865–9. doi:10.1115/1.3423721.
41. Hughes TJR, Caughey TK, Liu WK. Finite-element methods for nonlinear elastodynamics which conserve energy. J Appl Mech. 1978;45(2):366–70. doi:10.1115/1.3424303.
42. Kuhl D, Ramm E. Constraint Energy Momentum Algorithm and its application to non-linear dynamics of shells. Comput Methods Appl Mech Eng. 1996;136(3–4):293–315. doi:10.1016/0045-7825(95)00963-9.
43. Simo JC, Wong KK. Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum. Int J Numer Methods Eng. 1991;31(1):19–52. doi:10.1002/nme.1620310103.

44. Simo JC, Tarnow N. A new energy and momentum conserving algorithm for the non-linear dynamics of shells. Int J Numer Methods Eng. 1994;37(15):2527–49. doi:10.1002/nme.1620371503.
45. Simo JC, Tarnow N, Doblare M. Non-linear dynamics of three-dimensional rods: exact energy and momentum conserving algorithms. Int J Numer Meth Eng. 1995;38(9):1431–73. doi:10.1002/nme.1620380903.
46. Jelenić G, Crisfield MA. Dynamic analysis of 3D beams with joints in presence of large rotations. Comput Methods Appl Mech Eng. 2001;190(32–33):4195–230. doi:10.1016/S0045-7825(00)00344-3.
47. Li JZ, Liu YK, Li H, Cui NG, Yu KP. A unified family of high-order energy-conserving time integrators for nonlinear dynamical problems. Int J Numer Methods Eng. 2026;127(2):e70253. doi:10.1002/nme.70253.
48. Chen J, Huang ZH, Yi RH, Tian Q. A parallel variational integrator for simulating dynamics of large-scale geometrically exact beam systems on SE(3). Int J Numer Methods Eng. 2026;127(2):e70249. doi:10.1002/nme.70249.
49. Magisano D, Leonetti L, Garcea G. Unconditional stability in large deformation dynamic analysis of elastic structures with arbitrary nonlinear strain measure and multi-body coupling. Comput Methods Appl Mech Eng. 2022;393:114776. doi:10.1016/j.cma.2022.114776.
50. Abuteir BW, Harkati E, Boutagouga D, Mamouri S, Djeghaba K. Thermo-mechanical nonlinear transient dynamic and Dynamic-Buckling analysis of functionally graded material shell structures using an implicit conservative/decaying time integration scheme. Mech Adv Mater Struct. 2022;29(27):5773–92. doi:10.1080/15376494.2021.1964115.
51. Leonetti L, Kiendl J. A mixed integration point (MIP) formulation for hyperelastic Kirchhoff-Love shells for nonlinear static and dynamic analysis. Comput Methods Appl Mech Eng. 2023;416(7):116325. doi:10.1016/j.cma.2023.116325.
52. Chau AK, Brun M, Ventura P, Zahrouni H, Potier-Ferry M. Explicit dynamics and buckling simulations with 7-p shell elements and enhanced assumed strain. Finite Elem Anal Des. 2025;247(2):104346. doi:10.1016/j.finel.2025.104346.
53. Bauchau OA. A self-stabilized algorithm for enforcing constraints in multibody systems. Int J Solids Struct. 2003;40(13–14):3253–71. doi:10.1016/S0020-7683(03)00159-8.
54. Gebhardt CG, Romero I, Rolfes R. A new conservative/dissipative time integration scheme for nonlinear mechanical systems. Comput Mech. 2020;65(2):405–27. doi:10.1007/s00466-019-01775-3.
55. Gu SZ, Chen J, Tian Q. An adaptive time-step energy-preserving variational integrator for flexible multibody system dynamics. Appl Math Model. 2025;138(2):115759. doi:10.1016/j.apm.2024.115759.
56. Sánchez MA, Cockburn B, Nguyen NC, Peraire J. Symplectic Hamiltonian finite element methods for linear elastodynamics. Comput Methods Appl Mech Eng. 2021;381(5):113843. doi:10.1016/j.cma.2021.113843.
57. Schiebl M, Romero I. Energy-momentum conserving integration schemes for molecular dynamics. Comput Mech. 2021;67(3):915–35. doi:10.1007/s00466-020-01971-6.
58. Zhang HM, Xing YF. A three-parameter single-step time integration method for structural dynamic analysis. Acta Mech Sin. 2019;35(1):112–28. doi:10.1007/s10409-018-0775-y.
59. Zhang HM, Xing YF. A framework of time integration methods for nonsmooth systems with unilateral constraints. Appl Math Comput. 2019;363(2):124590. doi:10.1016/j.amc.2019.124590.
60. Zhang HM, Xing YF, Ji Y. An energy-conserving and decaying time integration method for general nonlinear dynamics. Int J Numer Methods Eng. 2020;121(5):925–44. doi:10.1002/nme.6251.
61. Santana MVB, Sansour C, Hjiaj M, Somja H. An equilibrium-based formulation with nonlinear configuration dependent interpolation for geometrically exact 3D beams. Int J Numer Methods Eng. 2022;123(2):444–64. doi:10.1002/nme.6862.
62. Chhang S, Sansour C, Keo P, Hjiaj M, Battini JM, Santana MVB. An energy-conserving time integration scheme for nonlinear dynamics analysis of geometrically exact 3D Euler-Bernoulli beams. Int J Numer Methods Eng. 2025;126(1):e7611. doi:10.1002/nme.7611.
63. Zhong HG, Crisfield MA. An energy-conserving co-rotational procedure for the dynamics of shell structures. Eng Comput. 1998;15(5):552–76. doi:10.1108/02644409810225715.
64. Almeida FS, Awruch AM. Corotational nonlinear dynamic analysis of laminated composite shells. Finite Elem Anal Des. 2011;47(10):1131–45. doi:10.1016/j.finel.2011.05.001.

65. Yang JS, Xia PQ. Energy conserving and decaying algorithms for corotational finite element nonlinear dynamic responses of thin shells. Sci China Technol Sci. 2012;55(12):3311–21. doi:10.1007/s11431-012-5002-7.
66. Yang JS, Xia PQ. Corotational nonlinear dynamic analysis of thin-shell structures with finite rotations. AIAA J. 2015;53(3):663–77. doi:10.2514/1.j053147.
67. Li ZX, Lin XD, Vu-Quoc L, Izzuddin BA, Wei HY, Xu J, et al. A quasi energy and momentum conservative algorithm implemented with a co-rotational quadrilateral shell element formulation using vectorial rotational variables. Int J Numer Methods Eng. 2025;126(17):e70128. doi:10.1002/nme.70128.
68. MacNeal RH. A simple quadrilateral shell element. Comput Struct. 1978;8(2):175–83. doi:10.1016/0045-7949(78)90020-2.
69. MacNeal RH. Derivation of element stiffness matrices by assumed strain distributions. Nucl Eng Des. 1982;70(1):3–12. doi:10.1016/0029-5493(82)90262-X.
70. Roh HY, Cho MH. The application of geometrically exact shell elements to B-spline surfaces. Comput Methods Appl Mech Eng. 2004;193(23–26):2261–99. doi:10.1016/j.cma.2004.01.019.
71. Kuhl D, Crisfield MA. Energy-conserving and decaying Algorithms in non-linear structural dynamics. Int J Numer Methods Eng. 1999;45(5):569–99. doi:10.1002/(sici)1097-0207(19990620)45:5<569::aid-nme595>3.0.co;2-a.
72. Kuhl D, Ramm E. Generalized energy-momentum method for non-linear adaptive shell dynamics. Comput Methods Appl Mech Eng. 1999;178(3–4):343–66. doi:10.1016/S0045-7825(99)00024-9.
73. Lavrenčič M, Brank B. Energy-decaying and momentum-conserving schemes for transient simulations with mixed finite elements. Comput Methods Appl Mech Eng. 2021;375(15):113625. doi:10.1016/j.cma.2020.113625.
74. Zhang J, Liu DH, Liu YH. Degenerated shell element with composite implicit time integration scheme for geometric nonlinear analysis. Int J Numer Methods Eng. 2016;105(7):483–513. doi:10.1002/nme.4975.
75. Key SW, Hoff CC. An improved constant membrane and bending stress shell element for explicit transient dynamics. Comput Methods Appl Mech Eng. 1995;124(1–2):33–47. doi:10.1016/0045-7825(95)00785-Y.
76. Zhang R, Stanciulescu I, Yao XH, Zhong HZ. An energy-momentum conserving scheme for geometrically exact shells with drilling DOFs. Comput Mech. 2021;67(1):341–64. doi:10.1007/s00466-020-01936-9.
77. Chróścielewski J, Witkowski W. Discrepancies of energy values in dynamics of three intersecting plates. Numer Methods Biomed Eng. 2010;26(9):1188–202. doi:10.1002/cnm.1208.