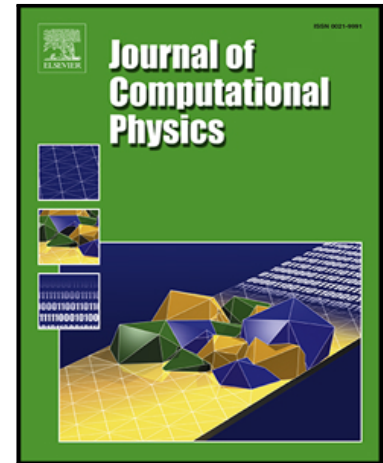


A Material Point Method for Reissner-Mindlin Shells

Tianrun Gao, David Hyde

PII: S0021-9991(26)00666-2
DOI: <https://doi.org/10.1016/j.jcp.2026.115317>
Reference: YJCPH 115317



To appear in: *Journal of Computational Physics*

Received date: 21 December 2025
Revised date: 14 May 2026
Accepted date: 17 August 2026

Please cite this article as: Tianrun Gao, David Hyde, A Material Point Method for Reissner-Mindlin Shells, *Journal of Computational Physics* (2026), doi: <https://doi.org/10.1016/j.jcp.2026.115317>

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: <https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article>. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

A Material Point Method for Reissner-Mindlin Shells

Tianrun Gao^a, David Hyde^{a,*}

^aDepartment of Computer Science, Vanderbilt University, 1400 18th Avenue South, Nashville, TN, USA 37212-2846

Abstract

This study proposes a novel material point method (MPM) for thin Reissner–Mindlin (RM) shells. The weak form of the RM shell is derived on 2D curved surfaces using Hamilton’s principle, and then the formulations in the 2D surface space are transformed to 3D Cartesian space in which the MPM background grid is defined. The proposed MPM shell method describes shell dynamics using a unified equation that is compatible with either the updated or total Lagrangian framework. The discretization of the weak form is performed with a general interpolation material point (GIMP) approach. Newmark- β time integration is adopted, which allows a much larger time step than the widely used update-stress-last (USL) scheme. For numerical stability, stress points are employed, and the gradients of shell field values are calculated on the curvilinear surface using the kernel function method. At stress points, stress tensors are updated incrementally, considering large deformations and rotations. Furthermore, an anti-hourglass term is proposed to further improve the numerical stability of computing pseudo-normals of the shell. Self- or multi-body contact of the shell can be treated easily, inheriting MPM’s natural handling of collisions and topological changes. A set of benchmark cases demonstrates the robustness and accuracy of the proposed method. Notably, the MPM shell method can produce highly accurate predictions with an MPM grid cell size much larger than the shell thickness, which eliminates the need for a very fine grid to resolve stress in the thickness direction, as in traditional MPM. To the authors’ knowledge, this is the first MPM work that uses thin shell theory and avoids relying on a meshed discretization of the shell.

Keywords: Material point method, Thin shell, Hybrid particle-grid method, Reissner-Mindlin shell, Kernel function

1. Introduction

Thin shell structures [1] appear and are modeled in many industries, such as ocean [2], aerospace [3], civil [4], or even biomedical engineering [5], due to their light weight and compactness properties. Under extreme conditions, thin structures are often the first to undergo large deformations, which can pose safety threats such as the buckling of rocket fairings, cooling towers, or vehicle roofs. Therefore, accurately and efficiently simulating the deformation of thin shells remains an important topic in computational physics research [6, 7, 8, 9].

The conceptually simplest simulation approach for thin shell structures is to treat the shell as any other volumetric three-dimensional structure, whereby conventional techniques like the finite element method (FEM) can be directly applied. However, for a method like FEM to produce accurate results, multiple layers of elements should be deployed in the thickness direction. Thus, in order to preserve element quality, meshing of thin shells typically requires an extensive number of elements, resulting in needless computational cost. Additionally, shear or volume locking problems may occur in this approach [10], reducing predictive accuracy.

Instead of the volumetric approach, a promising alternative is to apply thin shell theory to the structure. Thin shell theory generally treats three-dimensional shells as codimension-one (i.e., two-dimensional) surfaces. Various plate and shell theories have been developed (see [11, 12, 13] and the references therein), such as Timoshenko’s plate theory, Cosserat plate theory, and third-order shear deformation plate theory

*Corresponding author.

Email address: david.hyde.1@vanderbilt.edu (David Hyde)

[14]. However, two of the most popular shell models are Reissner-Mindlin (RM) and Kirchhoff-Love (KL) shells. For instance, these models are implemented in popular modeling packages such as ABAQUS [15], NGSolve [16, 17], and COMSOL [18]. Describing the bending stiffness of the surface using the so-called fiber direction [19, 20, 21] yields the RM shell model [22, 23, 24], in which the fiber direction is not necessarily perpendicular to the shell surface. The RM shell has been reported to suffer from transverse shear locking if the shape function and element integration are not properly handled [25]. For this problem, in recent decades there have been numerous studies focusing on the iso-geometric analysis (IGA) approach to deploy high-order smooth shape functions to simulate the RM shell more accurately [26, 27]. The KL shell model [28, 29] assumes that the fiber direction is always perpendicular to the shell surface, and this shell model eliminates the rotational degree of freedom and therefore the locking effect. However, the KL shell requires C^1 continuity for the shape functions, and the formulations (expressed using convected coordinates) are quite complicated for calculating the normal direction and its gradient. In contrast, the RM shell only requires C^0 continuity, and the governing equations are noticeably succinct compared to the KL shell.

In addition to specially designed elements for the shell based on FEM, meshless methods have also been used to simulate shell deformation. Smoothed particle hydrodynamics (SPH) is the earliest meshless method, proposed in [30], and recently it has been used to model shell structures [31, 32]. Peridynamics, another meshless method that is suitable for crack modeling, has also recently been applied for shell simulation [33, 34]. The element-free Galerkin (EFG) method is proposed in [35], which deploys disordered points in the field and then constructs shape functions using the least-squares method. A similar approach is called the reproducing kernel particle method (RKPM) [36], where the shape function is established using moving least squares combined with the window function. Recently, these two methods have been seen in the simulation of shell structures [37, 38]. The material point method (MPM) [39, 40] is another meshless method based on the Galerkin approach, which distinguishes itself in that MPM does not require constructing shape functions using particles as in EFG or RKPM. In MPM, a set of Lagrangian particles is deployed in the field, carrying all the field values, and a static background grid is used to solve the governing equations. Owing to this feature, MPM has been shown to robustly simulate numerous solid and fluid dynamics such as large deformation and contact [39], impact [41], penetration [42], fracture [43], and surface tension [44, 45], among others. The conventional linear MPM has the so-called cell-crossing instability [46], and to alleviate this issue, the generalized interpolation material point method (GIMP) [47] and the convected particle domain interpolation (CPDI) [48] are proposed. Using B-spline shape functions [49] also shows improved performance.

Due to MPM's advantages in handling large deformations, several studies have applied MPM to simulate thin structures in computational mechanics and computer graphics. In early explorations, thin structures were not modeled as "true" thin shells: prior MPM studies often regard the shell as a volumetric 3D body, such as the cantilever beam of [50] or the cylindrical thin wall of [51]. While these simulations produce accurate results, the consequence of this modeling choice is that the required grid cell size is extremely fine, e.g., $1/8$ of the shell's thickness [50], in order to resolve the stress in the thickness direction. A similar study is presented in [52], where the CPDI method is used to simulate beam vibration using polygonal cells in the beam body; however, the grid cell size is again required to be fine in the thickness direction to achieve accurate results. Compared to volumetric approaches, a shell modeling method that reduces a volumetric thin body into a surface—and avoids the requirement of fine grid resolution in the thickness direction—is naturally appealing. In this vein, membrane structures are simulated using the traditional MPM method in [53], where bending stiffness is not considered, and a similar study can be found in [54]. An MPM formulation for shell structures is also reported in [55]; however, severe numerical instability appears under large deformations and was not solved since then. Considering the difficulty in using MPM for shells, a combination of FEM and MPM is therefore explored to couple thin shells with other materials [56]. Compared to the computational mechanics community, MPM-based shell methods seem to receive more attention in the computer graphics community, in which cloth simulation is usually explored [57, 58, 59, 60], where folded and twisted behaviors are displayed; however, these methods are all conventionally mesh-based. In [61], subdivision FEM is deployed in the shell domain, and this method is coupled with MPM to handle contact, yielding successful results for simulating severely twisted shells. This method was used in [62] to simulate cloth deformation. However, this method relies on mesh-based FEM to calculate deformation tensors, which loses the meshless feature of MPM in the particle domain.

We highlight two recent studies based on MPM for thin structure simulation. In [63], the rotational degree of freedom is added to the beam using the CPDI method. Although accurate results are obtained, this method

still requires a volumetric approach for element deployment, which can impose a strong restriction on the grid cell size required for accuracy (in their examples, the authors choose a grid cell size equal to half the distance between discrete sample points on the simulated beams; they do not show results when their grid is relatively very coarse). A so-called solid shell MPM based on CPDI is proposed in [64], where the shell particles are characterized with hexahedral cells. In this work, the deformation strains of the hexahedral cells must be treated with care to eliminate the locking phenomenon. Moreover, this method still requires that the grid cell size be almost the same as the shell thickness. Neither of [63, 64] uses a thin shell theory such as KL or RM. Overall, an MPM for “genuine” thin shells is still lacking from the literature—a fact pointed out by de Vaucorbeil et al. in [46], where the authors “[...] observe a lack of MPM formulations for structural elements such as beams and shells”. Our work resolves this gap in the literature, yielding a genuine MPM for thin shells with appealing numerical properties.

In this study, we propose an MPM shell method based on the RM shell theory. This method only requires one layer of particles in the shell and totally eliminates the need of deploying a grid cell size finer than the shell thickness as in traditional MPM. Our method is distinct from CPDI, which requires a mesh in the shell body, and in this study, GIMP is used instead to make the MPM shell method possess the “meshless” feature as much as possible. The RM shell under large deformation is derived using Hamilton’s principle on a 2D curved surface, and a transformation is made to describe the MPM shell in the 3D Cartesian space, which is critical to make MPM possible for the shell theory. The kernel function method is used to calculate the gradients of field values on the curvilinear surface of the shell. Stress points and an enhanced stability term further improve the numerical stability of the MPM shell method. Additionally, self- or multi-body contact between shells can be treated easily in the present method.

The remainder of this paper is organized as follows. In Section 2, the kinematics and dynamics of the shell are described based on Hamilton’s principle. Section 3 elaborates on the MPM discretization method based on the weak form, including the transformation from 2D to 3D, P2G and G2P steps, and time integration. Sections 4 and 5 describe the stress estimation on the shell particles, enhanced numerical stability, contact treatment, and further implementation details. A set of cases is simulated using the MPM shell method in Section 6 to showcase the advantages of the proposed method.

2. Shell Governing Equations

We adopt the RM shell theory for our MPM shell method. Although RM shells have been explored using FEM and SPH, they remain unexplored in MPM, as clearly pointed out in [46]. In order to use RM shells in the context of MPM, in what follows, we derive the governing equations of the RM shell in the weak form using Hamilton’s principle. Although the introduction of continuous governing equations here is similar to the setup of [8], which considered thin shells using SPH, we emphasize that the discretization in the subsequent sections using MPM is fundamentally distinct.

2.1. Shell kinematics

When the shell structure is not deformed, a material point $\mathbf{X}_0$ in the shell body can be described completely using coordinates on the mid-surface $\boldsymbol{\varphi}_0$ and the material fiber direction $\mathbf{n}_0$ (in the reference configuration Ω_X), written as

$$\mathbf{X}_0(\xi_1, \xi_2, \xi_3) = \boldsymbol{\varphi}_0(\xi_1, \xi_2) + \xi_3 \mathbf{n}_0(\xi_1, \xi_2). \quad (1)$$

Using a curvilinear coordinate system, the mid-surface $\boldsymbol{\varphi}_0$ can be described using only two spatial coordinates; the local curvilinear coordinates (ξ_1, ξ_2) describe the position of $\mathbf{X}_0$ on the curved surface. ξ_3 denotes the position in the material fiber direction, which will be called the pseudo-normal in this study. Then, when the shell deforms, the corresponding deformed point (in the current configuration Ω_x) on the shell body can be expressed as

$$\mathbf{x}(\xi_1, \xi_2, \xi_3) = \boldsymbol{\varphi}(\xi_1, \xi_2) + \xi_3 \mathbf{n}(\xi_1, \xi_2), \quad (2)$$

where $\boldsymbol{\varphi}$ and $\mathbf{n}$ are the deformed mid-surface and pseudo-normal corresponding to $\boldsymbol{\varphi}_0$ and $\mathbf{n}_0$, respectively. Assuming no thickness deformation, $\|\mathbf{n}_0\| = \|\mathbf{n}\| = 1$. Furthermore, we note that $\mathbf{n}$ is not always perpendicular to the mid-surface in the RM shell model.

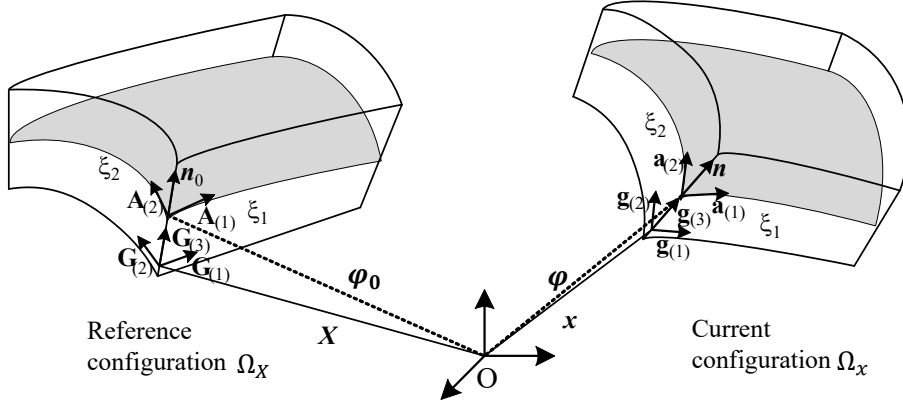


Fig. 1. Schematic of an RM shell in reference and current configurations.

In order to obtain the local deformation gradient $\mathbf{F}(\xi_1, \xi_2, \xi_3)$, the local convected vectors in the reference and current configurations should be calculated, expressed as

$$\begin{aligned}\mathbf{G}_{(\alpha)} &= \frac{\partial \mathbf{X}_0}{\partial \xi_\alpha} = \frac{\partial \varphi_0}{\partial \xi_\alpha} + \xi_3 \frac{\partial \mathbf{n}_0}{\partial \xi_\alpha} = \mathbf{A}_{(\alpha)} + \xi_3 \frac{\partial \mathbf{n}_0}{\partial \xi_\alpha}, \quad (\alpha = 1, 2) \\ \mathbf{G}_{(3)} &= \frac{\partial \mathbf{X}_0}{\partial \xi_3} = \mathbf{n}_0,\end{aligned}\quad (3)$$

and

$$\begin{aligned}\mathbf{g}_{(\alpha)} &= \frac{\partial \mathbf{x}}{\partial \xi_\alpha} = \frac{\partial \varphi}{\partial \xi_\alpha} + \xi_3 \frac{\partial \mathbf{n}}{\partial \xi_\alpha} = \mathbf{a}_{(\alpha)} + \xi_3 \frac{\partial \mathbf{n}}{\partial \xi_\alpha}, \quad (\alpha = 1, 2) \\ \mathbf{g}_{(3)} &= \frac{\partial \mathbf{x}}{\partial \xi_3} = \mathbf{n},\end{aligned}\quad (4)$$

respectively. As shown in Figure 1, $\mathbf{G}_{(\alpha)}$ are convected vectors at (ξ_1, ξ_2, ξ_3) and $\mathbf{A}_{(\alpha)}$ are those on the mid-surface, i.e., $\mathbf{A}_{(\alpha)}(\xi_1, \xi_2) = \mathbf{G}_{(\alpha)}(\xi_1, \xi_2, 0)$. For simplicity, $\mathbf{A}_{(\alpha)}$ are deployed orthogonal to each other in the reference configuration. Note that this work focuses on shells with initially smooth geometry, which means that there are no numerical discontinuities such as corners or stiffened ribs involved in the shell. For shells with stiffened ribs, the multi-patch approach [65] may be adopted in a future study. Given Equations 3 and 4, the deformation gradient $\mathbf{F}(\xi_1, \xi_2, \xi_3)$ at the height ξ_3 along the thickness direction can be derived as

$$\mathbf{F}(\xi_1, \xi_2, \xi_3) = [\mathbf{g}_{(\alpha)}] [\mathbf{G}_{(\alpha)}]^{-1}. \quad (5)$$

2.2. Shell dynamics

The governing equations of the shell are described in the weak form using Hamilton's principle, expressed as

$$\int_0^T (\delta K - \delta U - \delta W) dt = 0, \quad (6)$$

where δK , δU and δW are variational kinetic energy, potential energy, and external work, respectively, with δ the variational operator, and T is the current simulation time. We can estimate δK at the local shell surface $\mathcal{S}_0$ in the reference configuration as

$$\begin{aligned}\delta K &= \iint_{\mathcal{S}_0} \int_{-\zeta/2}^{\zeta/2} \rho (\dot{\varphi} + \xi_3 \dot{\mathbf{n}}) \cdot \delta (\dot{\varphi} + \xi_3 \dot{\mathbf{n}}) d\xi_3 d\xi_1 d\xi_2, \\ &= \iint_{\mathcal{S}_0} (I_0 \dot{\varphi} \cdot \delta \dot{\varphi} + I_1 \dot{\mathbf{n}} \cdot \delta \dot{\varphi} + I_1 \dot{\varphi} \cdot \delta \dot{\mathbf{n}} + I_2 \dot{\mathbf{n}} \cdot \delta \dot{\mathbf{n}}) d\xi_1 d\xi_2,\end{aligned}\quad (7)$$

where ζ is the shell thickness, and I_0 and I_2 are written as

$$I_0 = \int_{-\zeta/2}^{\zeta/2} \rho d\xi_3, \quad I_1 = \int_{-\zeta/2}^{\zeta/2} \rho \xi_3 d\xi_3 = 0, \quad I_2 = \int_{-\zeta/2}^{\zeta/2} \rho \xi_3^2 d\xi_3, \quad (8)$$

with ρ the shell density. In Equation (7), $I_1 = 0$ is based on the assumption that the material is uniform in the thickness direction, and this assumption holds throughout this work. Herein, $\dot{\bullet}$ and $\ddot{\bullet}$ denote the first- and second-order derivative of $\bullet$ with respect to time. Integrating δK in the time interval $[0, T]$, we obtain

$$\int_0^T \delta K dt = - \int_0^T \iint_{\mathcal{S}_0} (I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n}) d\xi_1 d\xi_2 dt. \quad (9)$$

Likewise, the variational work exerted on the shell surface can be calculated with

$$\delta W = - \iint_{\mathcal{S}_0} (\mathbf{f} \cdot \delta \boldsymbol{\varphi} + \mathbf{m} \cdot \delta \boldsymbol{\theta}) d\xi_1 d\xi_2, \quad (10)$$

where $\mathbf{f}$ and $\mathbf{m}$ are the force and moment per area exerted on the mid-surface in the reference configuration, respectively. Considering $\delta \mathbf{n} = \delta \boldsymbol{\theta} \times \mathbf{n}$ and neglecting the drilling rotation around pseudo normal, we have $\mathbf{m} \cdot \delta \boldsymbol{\theta} = (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n}$, where ‘ $\times$ ’ denotes the cross product operator.

For δU , the variational potential energy, we express this quantity in the reference configuration as

$$\begin{aligned} \delta U &= \int_{\Omega_X} \mathbf{P} : \delta \mathbf{F} dV = \int_{\Omega} \text{tr} (\mathbf{P} \delta \mathbf{F}^T) dV \\ &= \int_{\Omega_0} \text{tr} \left(\mathbf{P} \delta \left\{ [\mathbf{g}_{(\alpha)}] [\mathbf{G}_{(\alpha)}]^{-1} \right\}^T \right) dV = \int_{\Omega} \text{tr} \left(\mathbf{P} [\mathbf{G}_{(\alpha)}]^{-T} \delta \mathbf{g}_{(\alpha)}^T \right) dV \\ &= \int_{\Omega_0} \text{tr} \left[\mathbf{P} [\mathbf{G}_{(\alpha)}]^{-T} \delta \left\{ \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right) + \xi_3 \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right) + (0, 0, \mathbf{n}) \right\}^T \right] dV \\ &= \iint_{\mathcal{S}_0} \int_{-\zeta/2}^{\zeta/2} \text{tr} \left[\mathbf{P} [\mathbf{G}_{(\alpha)}]^{-T} \delta \left\{ \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right) + \xi_3 \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right) + (0, 0, \mathbf{n}) \right\}^T \right] d\xi_3 d\xi_1 d\xi_2 \\ &= \iint_{\mathcal{S}_0} \text{tr} \left[\mathbf{N} \delta \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right)^T + \mathbf{M} \delta \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right)^T + \mathbf{N} \delta (0, 0, \mathbf{n})^T \right] d\xi_1 d\xi_2, \end{aligned} \quad (11)$$

where

$$\mathbf{N} = \int_{-\zeta/2}^{\zeta/2} \mathbf{P} [\mathbf{G}_{(\alpha)}]^{-T} d\xi_3, \quad \mathbf{M} = \int_{-\zeta/2}^{\zeta/2} \mathbf{P} [\mathbf{G}_{(\alpha)}]^{-T} \xi_3 d\xi_3. \quad (12)$$

Here, $\boldsymbol{\sigma}$ and $\mathbf{P}$ are the Cauchy stress and first Piola-Kirchhoff (PK) stress, respectively, with the relation written as $\mathbf{P} = J \boldsymbol{\sigma} \mathbf{F}^{-T}$, with $J = |\mathbf{F}|$. In implementation, the two integrals in Equation (12) can be calculated using the Gaussian quadrature approach, and three points turn out to be sufficient for an accurate integration.

Combining Equations (6), (9), (10) and (11), we have

$$\begin{aligned} \int_0^T (\delta K - \delta U - \delta W) dt &= - \int_0^T \iint_{\mathcal{S}_0} \left\{ I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n} - \mathbf{f} \cdot \delta \boldsymbol{\varphi} - (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n} \right. \\ &\quad \left. + \text{tr} \left[\mathbf{N} \delta \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right)^T + \mathbf{M} \delta \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right)^T + \mathbf{N} \delta (0, 0, \mathbf{n})^T \right] \right\} d\xi_1 d\xi_2 dt = 0. \end{aligned} \quad (13)$$

Since T is arbitrary, then by the fundamental theorem of calculus, we obtain

$$\begin{aligned} \iint_{\mathcal{S}_0} \left\{ I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n} - \mathbf{f} \cdot \delta \boldsymbol{\varphi} - (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n} \right. \\ \left. + \text{tr} \left[\mathbf{N} \delta \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right)^T + \mathbf{M} \delta \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right)^T + \mathbf{N} \delta (0, 0, \mathbf{n})^T \right] \right\} d\xi_1 d\xi_2 = 0. \end{aligned} \quad (14)$$

This equation is a general weak form for the shell derived using only mid-surface and pseudo-normal motions. Note that this formulation is different from the conventional ones given in [66] and [67], where the rotation angles are directly used in the governing equations. In contrast, the present formulation separates the rotation angle from the pseudo-normal in the weak form, which renders the governing equation more concise.

3. MPM Discretization

3.1. Brief overview of the method

In the present MPM for RM shells, the overall particle-to-grid (P2G) and grid-to-particle (G2P) steps are retained as a merit of MPM. The difference is that MPM background integration is implemented by a transformation from 2D curved surface to 3D Cartesian space, as derived in Section 3.2, and the P2G and G2P steps are derived based on the transformed equation in Equation (18). In addition, stress points (Section 4.1) are deployed to increase the robustness of this method, and the deformation gradients, strain, and stress are directly calculated using the kernel function approach introduced in Section 4.2, instead of relying on grids as in conventional MPM.

3.2. From 2D curved surface to 3D Cartesian space

At first glance, Equation (14) above seems impossible to implement based on MPM, because this equation is described on the 2D curved surface, which is incompatible with MPM described in 3D Cartesian space. We propose a novel transformation to map this equation described in 2D curvilinear surface space into an equation in 3D Cartesian space.

Keeping in mind that all field values do not change along the thickness direction, we can integrate Equation (14) in the thickness direction again:

$$\int_{-\zeta/2}^{\zeta/2} \iint_{\mathcal{S}_0} \left\{ I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n} - \mathbf{f} \cdot \delta \boldsymbol{\varphi} - (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n} + tr \left[\mathbf{N} \delta \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right)^T + \mathbf{M} \delta \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right)^T + \mathbf{N} \delta (0, 0, \mathbf{n})^T \right] \right\} d\xi_1 d\xi_2 d\xi_3 = 0. \quad (15)$$

Given that the field values are invariant along the thickness direction, the convected vectors in the ξ_α system can be written in the Cartesian reference or current configuration via

$$\begin{aligned} \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right)^T &= \mathbf{L} [\mathbf{g}_{(\alpha)}^0]^T \nabla_{\mathbf{x}} \boldsymbol{\varphi}, & \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right)^T &= \mathbf{L} [\mathbf{g}_{(\alpha)}^0]^T \nabla_{\mathbf{x}} \mathbf{n}, \\ \text{or} \quad \left(\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2}, 0 \right)^T &= \mathbf{L} [\mathbf{G}_{(\alpha)}^0]^T \nabla_{\mathbf{X}} \boldsymbol{\varphi}, & \left(\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2}, 0 \right)^T &= \mathbf{L} [\mathbf{G}_{(\alpha)}^0]^T \nabla_{\mathbf{X}} \mathbf{n}, \end{aligned} \quad (16)$$

with

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (17)$$

Given the transformation above, then Equation (15) is written as

$$\begin{aligned} & \int_{\Omega_x} \frac{1}{J} \left[I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n} - \mathbf{f} \cdot \delta \boldsymbol{\varphi} - (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n} \right. \\ & \quad \left. + tr \left(\mathbf{NL} [\mathbf{g}_{(\alpha)}^0]^T \nabla_{\mathbf{x}} \delta \boldsymbol{\varphi} + \mathbf{ML} [\mathbf{g}_{(\alpha)}^0]^T \nabla_{\mathbf{x}} \delta \mathbf{n} + \mathbf{N}_{(3)} \cdot \delta \mathbf{n} \right) \right] dV_x \\ &= \int_{\Omega_X} \left[I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n} - \mathbf{f} \cdot \delta \boldsymbol{\varphi} - (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n} \right. \\ & \quad \left. + tr \left(\mathbf{NL} [\mathbf{G}_{(\alpha)}^0]^T \nabla_{\mathbf{X}} \delta \boldsymbol{\varphi} + \mathbf{ML} [\mathbf{G}_{(\alpha)}^0]^T \nabla_{\mathbf{X}} \delta \mathbf{n} + \mathbf{N}_{(3)} \cdot \delta \mathbf{n} \right) \right] dV_X = 0, \end{aligned} \quad (18)$$

where V_x and V_X are the infinitesimal volume in the current and reference configurations, respectively.

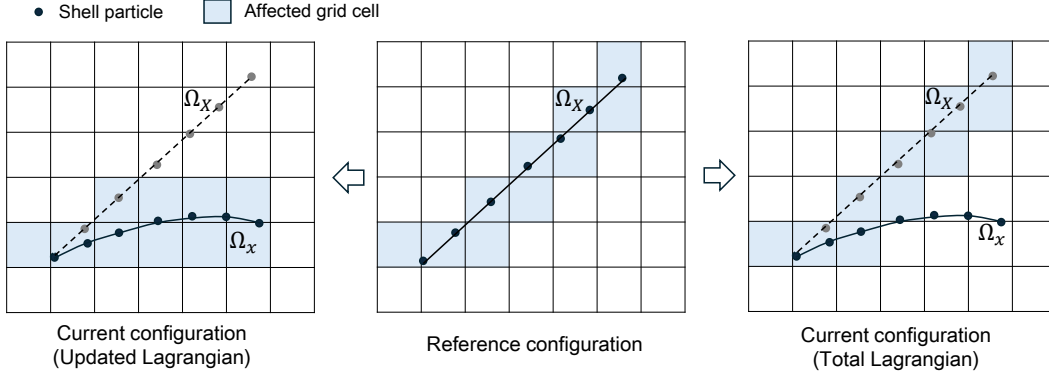


Fig. 2. Reference (Ω_X) and current (Ω_x) configurations of a shell and their corresponding affected grid cells. Either updated or total Lagrangian frameworks can be used to describe the governing equations, unified in Equation (18). The diagrams here assume linear interpolation is used.

Equation (18) above transforms the 2D curved surface into 3D Cartesian space, which renders MPM feasible for the simulation of RM shells in 3D space. In fact, Equation (18) describes the shell dynamics in 4D space due to the additional integration in Equation (15), but this does not affect the kinematics of the mid-surface and pseudo-normal.

Another merit of this formula is that both updated and total Lagrangian descriptions can be described in the same framework: this choice can be decided by selecting the transformation matrix $[\mathbf{g}_{(\alpha)}^0]$ or $[\mathbf{G}_{(\alpha)}^0]$, which are short for $[\mathbf{g}_{(\alpha)}(\xi_1, \xi_2, 0)]$ or $[\mathbf{G}_{(\alpha)}(\xi_1, \xi_2, 0)]$, respectively. As shown in Figure 2, when a shell body deforms to a configuration Ω_x , choosing updated and total Lagrangian descriptions will lead to different affected grid cells (shown in light blue). In this study, the total Lagrangian framework will be focused on, because only the affected grid cells around the shell particles in the reference configuration are used; therefore, the associated “active” grid nodes do not change during a simulation, which avoids recomputations needed in the updated Lagrangian framework. In addition, the total Lagrangian framework is normally more stable and robust since it avoids potential numerical issues arising from the changing derivatives in the updated configuration. Nonetheless, a comparison between the two frameworks is given in Section 6.2.1.

The RKPM approach has recently been extended to shell simulations [68, 69]. Similarly, RKPM also uses a background mesh for numerical integration, which is usually in a total Lagrangian framework. Although an updated Lagrangian framework is also feasible in principle, mesh distortion is not absolutely eradicated. In contrast, the present MPM for RM shells uses a Cartesian background mesh, allowing both total and updated Lagrangian frameworks, and therefore there is theoretically no potential mesh distortion issue.

3.3. P2G

In MPM, information from (generally, moving) particles is transferred to a static background grid via a particle-to-grid (P2G) step, and updates for momentum and other terms are then solved for on the grid. Updated quantities on the grid are then interpolated back to particles in a grid-to-particle (G2P) step, and finally, particle-based quantities such as position are updated [70]. This hybrid particle-grid strategy renders MPM flexible for solving the problems with large deformations, especially since compared to a technique like FEM there is no risk of degraded element quality since only the particles move.

In the initial configuration, assuming there are N_p shell particles, Equation (18) can be rewritten using shell particles as

$$\sum_p^{N_p} \int_{\Omega_X^p} \left[I_0 \ddot{\boldsymbol{\varphi}} \cdot \delta \boldsymbol{\varphi} + I_2 \ddot{\mathbf{n}} \cdot \delta \mathbf{n} - \mathbf{f} \cdot \delta \boldsymbol{\varphi} - (\mathbf{m} \times \mathbf{n}) \cdot \delta \mathbf{n} + \text{tr} \left(\mathbf{NL}[\mathbf{G}_{(\alpha)}^0]^T \nabla \delta \boldsymbol{\varphi} + \mathbf{ML}[\mathbf{G}_{(\alpha)}^0]^T \nabla \delta \mathbf{n} + \mathbf{N}_{(3)} \cdot \delta \mathbf{n} \right) \right] H_p(\mathbf{X}) dV_X. \quad (19)$$

In the original MPM, $H_p(\mathbf{X}) = H(\mathbf{X} - \boldsymbol{\varphi}_p)$, with $\boldsymbol{\varphi}_p$ the position of particle p , denotes the Dirac function with its particle domain width being zero; however, this method presents numerical instability. In the GIMP

method, $H(\mathbf{X} - \boldsymbol{\varphi}_p)$ denotes the particle characteristic function, which has a non-zero particle domain width. For more details, the readers are referred to [47]. In this study, the width of shell particles Δx is selected as the (uniform) grid cell width Δx_g , i.e., $\Delta x = \Delta x_g$, which shows high robustness and accuracy.

If the field value of a shell particle is approximated using the grid, the mid-surface and pseudo-normal are written as

$$\begin{aligned}\delta\boldsymbol{\varphi}(\mathbf{X}) &= \delta\boldsymbol{\varphi}_I S_I(\mathbf{X}), & \delta\mathbf{n}(\mathbf{X}) &= \delta\mathbf{n}_I S_I(\mathbf{X}), \\ \ddot{\boldsymbol{\varphi}}(\mathbf{X}) &= \ddot{\boldsymbol{\varphi}}_I S_I(\mathbf{X}), & \ddot{\mathbf{n}}(\mathbf{X}) &= \ddot{\mathbf{n}}_I S_I(\mathbf{X}),\end{aligned}\quad (20)$$

where $S_I(\mathbf{X})$ is the shape function corresponding to node I . In this study, the widely used linear quadrilateral and brick elements are employed for 2D and 3D cases, respectively, in the grid domain in this study. This shape function is widely used in linear MPM, but a complete linear MPM is not stable due to continuity C^0 , which is why the GIMP with continuity C^1 is adopted for shape functions. Note that high-order shape functions such as B-splines may provide better stability. Based on the GIMP method and considering Equations (19) and (20), Equation (19) can be rewritten as

$$\begin{aligned}& \sum_p^{N_p} \int_{\Omega_X^p} [I_{0p} \ddot{\boldsymbol{\varphi}}_J S_J \cdot \delta\boldsymbol{\varphi}_I S_I + I_{2p} \ddot{\mathbf{n}}_J S_J \cdot \delta\mathbf{n}_I S_I - \mathbf{f}_p \cdot \delta\boldsymbol{\varphi}_I S_I - (\mathbf{m} \times \mathbf{n}) \cdot \delta\mathbf{n}_I S_I] H_p \\& + tr(\mathbb{N}_p \nabla S_I \delta\boldsymbol{\varphi}_I + \mathbb{M}_p \nabla S_I \delta\mathbf{n}_I + \mathbf{N}_{(3)p} \cdot S_I \delta\mathbf{n}_I) H_p dV_X \\& = \sum_p^{N_p} \int_{\Omega_X^p} \delta\boldsymbol{\varphi}_I^T (I_{0p} \ddot{\boldsymbol{\varphi}}_J S_J S_I + \mathbb{N}_p \nabla S_I - \mathbf{f}_p S_I) H_p \\& + \delta\mathbf{n}_I^T [I_{2p} \ddot{\mathbf{n}}_J S_J S_I + \mathbb{M}_p \nabla S_I + \mathbf{N}_{(3)p} S_I - (\mathbf{m} \times \mathbf{n})_p S_I] H_p dV_X = 0,\end{aligned}\quad (21)$$

with

$$\mathbb{N}_p = \mathbf{NL}[\mathbf{G}_{(\alpha)}^0]^T, \quad \mathbb{M}_p = \mathbf{ML}[\mathbf{G}_{(\alpha)}^0]^T. \quad (22)$$

Using the variational principle, the equation above can be reorganized as

$$\begin{aligned}& \sum_p^{N_p} I_{0p} \bar{S}_{JI} \ddot{\boldsymbol{\varphi}}_J = \sum_p^{N_p} (-\mathbb{N}_p \nabla \bar{S}_{Ip} + \mathbf{f}_p \bar{S}_{Ip}), \\& \sum_p^{N_p} I_{2p} \bar{S}_{JI} \ddot{\mathbf{n}}_J = \sum_p^{N_p} [-\mathbb{M}_p \nabla \bar{S}_{Ip} - \mathbf{N}_{(3)p} \bar{S}_{Ip} + (\mathbf{m} \times \mathbf{n})_p \bar{S}_{Ip}],\end{aligned}\quad (23)$$

with

$$\bar{S}_{JI} = \int_{\Omega_X^p} S_J S_I H_p dV, \quad \bar{S}_{Ip} = \int_{\Omega_X^p} S_I H_p dV, \quad \nabla \bar{S}_{Ip} = \int_{\Omega_X^p} \nabla S_I H_p dV. \quad (24)$$

In practice, the new shape functions are rewritten as

$$\bar{S}_{Ip} = \frac{\int_{\Omega_X^p} S_I H_p dV}{\int_{\Omega_X^p} H_p dV}, \quad \nabla \bar{S}_{Ip} = \frac{\int_{\Omega_X^p} \nabla S_I H_p dV}{\int_{\Omega_X^p} H_p dV}, \quad (25)$$

to ensure the unity property of the shape functions for a particle p , i.e.,

$$\sum_I \bar{S}_{Ip} = 1, \quad \sum_I \nabla \bar{S}_{Ip} = 0. \quad (26)$$

Equations (23)–(26) show how the GIMP shape functions are derived based on the original linear MPM shape functions. On the left-hand side of Equation (23), the coupling effect between different nodes leads to a consistent mass matrix, which requires solving a linear system to derive $\ddot{\boldsymbol{\varphi}}_J$ and $\ddot{\mathbf{n}}_J$, which is usually time-consuming. To circumvent this, the row mass lumping technique is used instead to solve $\ddot{\boldsymbol{\varphi}}$ and $\ddot{\mathbf{n}}$

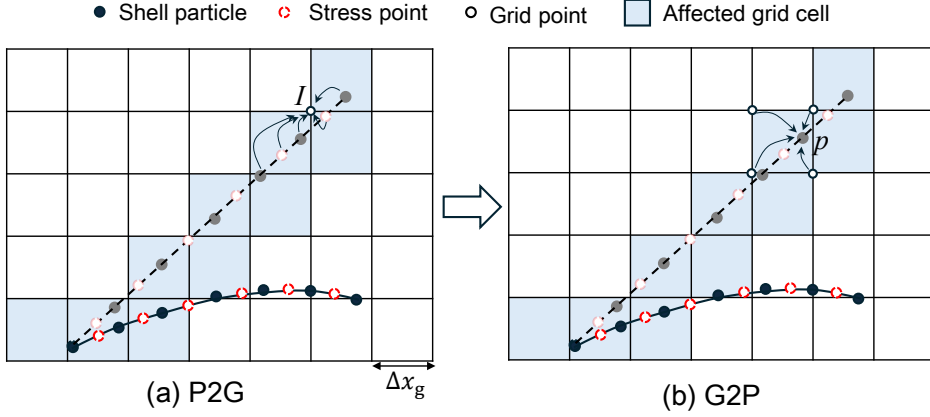


Fig. 3. Discretization with MPM includes (a) P2G and (b) G2P steps. Our implementation uses a total Lagrangian framework with stress points (red; introduced in Section 4.1). In the P2G step, quantities stored on the shell particles and stress points are mapped to the grid nodes, and in the G2P step, quantities on the grid nodes are mapped back only to the shell particles.

explicitly, expressed as

$$\begin{aligned} \ddot{\varphi}_I \sum_p^{N_p} I_{0p} \bar{S}_{Ip} &= \sum_p^{N_p} (-\mathbb{N}_p \nabla \bar{S}_{Ip} + \mathbf{f}_p \bar{S}_{Ip}), \\ \ddot{\mathbf{n}}_I \sum_p^{N_p} I_{2p} \bar{S}_{Ip} &= \sum_p^{N_p} [-\mathbb{M}_p \nabla \bar{S}_{Ip} - \mathbf{N}_{(3)p} \bar{S}_{Ip} + (\mathbf{m} \times \mathbf{n})_p \bar{S}_{Ip}], \end{aligned} \quad (27)$$

with which the acceleration $\ddot{\varphi}_I$ and $\ddot{\mathbf{n}}_I$ at the nodes can be obtained efficiently. In Equation (27), one can observe that the neighboring particles of a grid node should be looped over to get equivalent mass or force at a node, which will be called a P2G step, as shown in Figure 3(a). We note that in Figure 3(a), the particles in the domain include shell particles and stress points (which will be introduced in Section 4.1).

In the P2G step, the mid-surface velocity and pseudo-normal *rotational* velocity should also be mapped from the shell particles to the grid, expressed as

$$\dot{\varphi}_I = \sum_p^{N_p} \dot{\varphi}_p \bar{S}_{Ip} / \sum_p^{N_p} \bar{S}_{Ip}, \quad \dot{\boldsymbol{\theta}}_I = \sum_p^{N_p} \dot{\boldsymbol{\theta}}_p \bar{S}_{Ip} / \sum_p^{N_p} \bar{S}_{Ip}. \quad (28)$$

Here, $\dot{\boldsymbol{\theta}}_p$ and $\ddot{\boldsymbol{\theta}}_p$ are the rotational velocity and acceleration, which will be introduced in Equation (30). These two equations will be used in the G2P step. In the total Lagrangian framework, the particles that a grid cell contains can be searched beforehand and stored for later use, and the neighboring particles of a node will not change thereafter. Thus, the P2G step is quite efficient.

To clarify how the variables are represented and transferred between particles and grid nodes, we provide a brief summary of the P2G step here. The key task of the P2G step is to know the force terms and mass terms on the right-hand and left-hand sides of Equation (27), for which our goal is to calculate $\mathbb{N}_p$, $\mathbf{f}_p$, $\mathbb{M}_p$, $\mathbf{N}_{(3)p}$, $(\mathbf{m} \times \mathbf{n})_p$ for forces, and I_{0p} , I_{2p} for mass, respectively. To calculate these values, one can refer to Equations (40), (47), (12), (22), (10) and (8) in sequence (refer to Algorithm 1). Given those values, the summation in Equation (27) forms a P2G mapping process. Then $\ddot{\varphi}_I$ and $\ddot{\mathbf{n}}_I$ are obtained followed by the G2P step.

3.4. G2P and particle quantity updates

After the accelerations $\ddot{\varphi}_I$ and $\ddot{\mathbf{n}}_I$ at the nodes are obtained, we can perform a G2P step that interpolates nodal acceleration values back to shell particles. We emphasize that all the external forces on the particles, including gravity, pressure, force, etc., have already been handled in the P2G step, i.e., on the right-hand side of Equation (27).

Considering the unity property of Equation (26), $\ddot{\varphi}_i$ and $\ddot{\mathbf{n}}_i$ at the particles can be calculated using the GIMP shape functions as

$$\ddot{\varphi}_i = \bar{S}_{Ii} \ddot{\varphi}_I, \quad \ddot{\mathbf{n}}_i = \bar{S}_{Ii} \ddot{\mathbf{n}}_I. \quad (29)$$

It is worth noting that the G2P step is only implemented on the shell particles i , not on the stress points (the values of stress points will be interpolated from shell particles as introduced in Section 4.1), as shown in Figure 3(b). The rotational acceleration $\ddot{\boldsymbol{\theta}}_i$ can be derived using the relation

$$\ddot{\boldsymbol{\theta}}_i = \mathbf{n}_i \times \ddot{\mathbf{n}}_i - (\dot{\boldsymbol{\theta}}_i \cdot \mathbf{n}_i) \mathbf{n}_i \times \dot{\boldsymbol{\theta}}_i + (\mathbf{n}_i \cdot \ddot{\boldsymbol{\theta}}_i) \mathbf{n}_i, \quad (30)$$

and when the rotation with respect to the pseudo-normal is neglected, the equation above reduces to $\ddot{\boldsymbol{\theta}}_i = \mathbf{n}_i \times \ddot{\mathbf{n}}_i$.

Given $\ddot{\varphi}_i$ and $\ddot{\boldsymbol{\theta}}_i$, field values can be updated on the particles. In this paper, we use the explicit Newmark- β method [71], which allows a much larger integration time step and is much more stable than the traditional first-order time integration used in MPM. In the explicit Newmark- β method, there are two sub-time steps: prediction and correction steps. In the prediction step, the field values are updated as

$$\begin{aligned} \tilde{\varphi}_i^{n+1} &= \varphi_i^n + \Delta t \dot{\varphi}_i^n + \frac{\Delta t^2}{2} (1 - 2\beta) \ddot{\varphi}_i^n, \\ \dot{\tilde{\varphi}}_i^{n+1} &= \epsilon \dot{\varphi}_i^n + (1 - \epsilon) \dot{\tilde{\varphi}}_i^n + \Delta t (1 - \gamma) \ddot{\varphi}_i^n, \\ \tilde{\boldsymbol{\theta}}_i^{n+1} &= \boldsymbol{\theta}_i^n + \Delta t \dot{\boldsymbol{\theta}}_i^n + \frac{\Delta t^2}{2} (1 - 2\beta) \ddot{\boldsymbol{\theta}}_i^n, \\ \dot{\tilde{\boldsymbol{\theta}}}_i^{n+1} &= \epsilon \dot{\boldsymbol{\theta}}_i^n + (1 - \epsilon) \dot{\tilde{\boldsymbol{\theta}}}_i^n + \Delta t (1 - \gamma) \ddot{\boldsymbol{\theta}}_i^n, \end{aligned} \quad (31)$$

where the parameters are chosen as $\beta = 0.25$ and $\gamma = 0.5$. The time step Δt is chosen as

$$\Delta t = 0.8 \min(\Delta x_g, \zeta) / c, \quad (32)$$

with c the speed of sound in the shell material and ζ the shell thickness. It is worth noting that $\dot{\varphi}_i^n$ and $\dot{\boldsymbol{\theta}}_i^n$ should be derived from the MPM background grid, i.e.,

$$\dot{\varphi}_i^n = \bar{S}_{Ii} \dot{\varphi}_I^n, \quad \dot{\boldsymbol{\theta}}_i^n = \bar{S}_{Ii} \dot{\boldsymbol{\theta}}_I^n, \quad (33)$$

where $\dot{\varphi}_I^n$ and $\dot{\boldsymbol{\theta}}_I^n$ have been derived in the P2G step using Equation (28). In Equation (31), ϵ is a factor that blends between the particle-in-cell (PIC) [72] and fluid-implicit-particle (FLIP) [73, 74] methods; this is inspired by e.g. the α used in [75]. Since PIC brings in excessive numerical dissipation, $\epsilon = 1$ is chosen for all dynamic cases. In quasi-static cases, ϵ can be smaller than 1 to make the simulation enter equilibrium faster, as introduced in Section 5.4. Other advanced schemes such as the affine particle-in-cell (APIC) method [76] can be employed as a substitute, which is worth consideration in future work.

When the prediction step is completed, the subsequent correction step is computed as

$$\begin{aligned} \varphi_i^{n+1} &= \tilde{\varphi}_i^{n+1} + \beta \Delta t^2 \ddot{\varphi}_i^{n+1}, \\ \dot{\varphi}_i^{n+1} &= \epsilon \dot{\tilde{\varphi}}_i^{n+1} + (1 - \epsilon) \dot{\varphi}_i^n + \gamma \Delta t \ddot{\varphi}_i^{n+1}, \\ \boldsymbol{\theta}_i^{n+1} &= \tilde{\boldsymbol{\theta}}_i^{n+1} + \beta \Delta t^2 \ddot{\boldsymbol{\theta}}_i^{n+1}, \\ \dot{\boldsymbol{\theta}}_i^{n+1} &= \epsilon \dot{\tilde{\boldsymbol{\theta}}}_i^{n+1} + (1 - \epsilon) \dot{\boldsymbol{\theta}}_i^n + \gamma \Delta t \ddot{\boldsymbol{\theta}}_i^{n+1}, \end{aligned} \quad (34)$$

where all the accelerations $\ddot{\varphi}_i^{n+1}$ and $\ddot{\boldsymbol{\theta}}_i^{n+1}$ are calculated based on the prediction step. We note that n and m are time step and sub-time step, respectively. It means that, for instance, if Equation (31) is sub-time step m in time step n , Equation (34) will be $m + 1$ in n , and then in time step $n + 1$, Equation (31) and (34) are $m + 2$ and $m + 3$, respectively. In this time integration, the implementation of the two sub-time steps is the same apart from the calculation of Equation (31) or (34), as shown in Algorithm 1. This is why the stress tensors are updated based on the sub-time step m , as introduced in Section 4.3.

Finally, after the rotational increment $\Delta\theta = (\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$ is obtained, the pseudo-normal $\mathbf{n}^{m+1}$ at sub-time step $m + 1$ can be updated by a Rodrigues rotational matrix $\Delta\mathbf{R}$, i.e., $\mathbf{n}^{m+1} = \Delta\mathbf{R}\mathbf{n}^m$, with

$$\Delta\mathbf{R} = \mathbf{I} + \frac{\sin \|\Delta\theta\|}{\|\Delta\theta\|} \mathbf{\Theta} + \frac{(1 - \cos \|\Delta\theta\|)}{\|\Delta\theta\|^2} \mathbf{\Theta}^2, \quad (35)$$

where the skew-symmetric matrix $\mathbf{\Theta}$ is

$$\mathbf{\Theta} = \begin{bmatrix} 0 & -\Delta\theta_3 & \Delta\theta_2 \\ \Delta\theta_3 & 0 & -\Delta\theta_1 \\ -\Delta\theta_2 & \Delta\theta_1 & 0 \end{bmatrix}. \quad (36)$$

4. Particle Stress Update

This section elaborates on how to obtain the stress tensors required in Equation (27). Readers are referred to Algorithm 1 for more details regarding the calculation orders of deformation gradient, strain, and stress.

4.1. Stress points

The stress oscillation phenomenon usually occurs when stress is estimated at the moving particles, and the oscillation may make the simulation unstable, leading to the so-called hourglass instability. The stress points method is widely used to alleviate this phenomenon in meshless methods, such as SPH [77], peridynamics [78], and RKPM [38], and it is discussed comprehensively in [79]. The stress points method deploys another set of particles, called stress points, usually in a staggered approach in the computational domain to estimate stress¹. For lattice-distributed shell particles, the typical staggered approach to distribute the stress points is shown in Figure 4. In the P2G step, the stress points and shell particles are both used, i.e., the subscript ‘ p ’ in Equation (27) includes both kinds of particles.

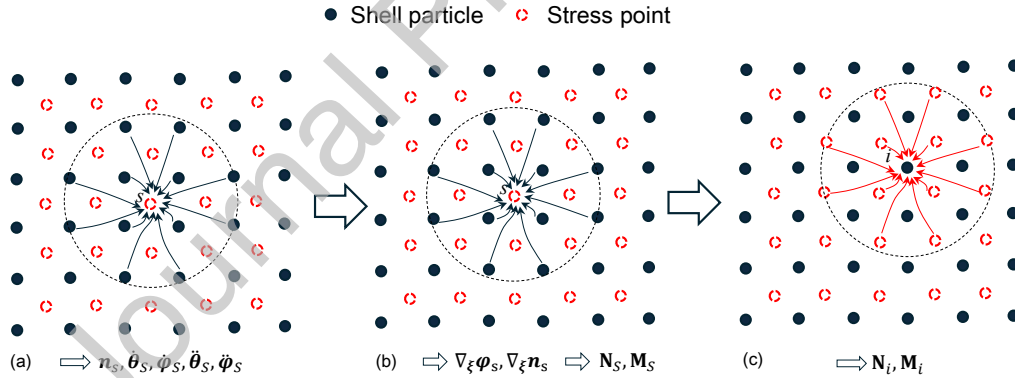


Fig. 4. Shell particles are used to update kinematic field values of the shell, and stress points are used to calculate stress tensors of the shell. (a) The kinematic field values at stress points are interpolated from the neighboring shell particles, and then (b) the stress points calculate the stress tensors. (c) The stress tensors of shell particles are interpolated from their neighboring stress points.

4.2. Gradient calculations at stress points

In the original MPM, the strain tensor is estimated using the background grid to calculate the deformation gradient, and then the stress tensors are calculated accordingly. However, this method may not work well for thin structures. For instance: (1) The gradients of the mid-surface and pseudo-normal should be calculated accurately in the shell plane to handle the plane stress state. It is difficult to calculate the deformation in a

¹Note that it is not necessary to position the stress points in a strictly staggered way: as long as the stress points do not entirely overlap with the shell particles, this method is known to work well.

desired direction using original MPM; (2) Even if the deformation can be estimated, the widely used linear shape functions may lead to excessive error and locking phenomena.

To solve these two barriers, we adopt the kernel function method to estimate the gradients of the mid-surface and pseudo-normal accurately. The main idea is to use neighboring shell particles for an accurate estimation. For a shell particle i or stress point s , its neighboring shell particle neighbors within the radius $1.4\Delta x$ are constructed, forming a set $\mathcal{N}_i$ or $\mathcal{N}_s$, where Δx is the initial particle spacing, and the surface area of the shell particle i is $S_i = \Delta x^2$. In this work, we let $\Delta x = \Delta x_g$. Herein, a C^2 Wendland kernel function $W(r, h)$ [80] is used, written as

$$W(r, h) = \begin{cases} (1 - q/2)^4(1 + 2q), & q \leq 2, \\ 0, & q > 2, \end{cases} \quad (37)$$

where $q = r/h$, with h the smoothing length of the kernel function.

Before calculating the gradients, the field values of the stress points, i.e., $\mathbf{n}_s$, $\dot{\boldsymbol{\varphi}}_s$, $\ddot{\boldsymbol{\varphi}}_s$, $\dot{\boldsymbol{\theta}}_s$ and $\ddot{\boldsymbol{\theta}}_s$ should be obtained using the neighboring shell particles that have been updated in either Equation (31) or 34. Using the kernel function given above, the corresponding field value ' $\langle \bullet \rangle_s$ ' can be interpolated using

$$\langle \bullet \rangle_s = \left(\sum_{\mathcal{N}_s} \langle \bullet \rangle_i W_{si} S_i \right) / \sum_{\mathcal{N}_s} W_{si} S_i, \quad (38)$$

where

$$W_{si} = W(\|\boldsymbol{\varphi}_{0s} - \boldsymbol{\varphi}_{0i}\|, h), \quad (39)$$

where $\boldsymbol{\varphi}_0$ is defined in Equation (1). With Equation (38) and Equation (31) or 34, all the field values can be updated for the stress points, as shown in Figure 4(a).

After the field values are obtained, the field gradient can be calculated with the kernel function. Referring to the corrected matrix method [81], the gradients of mid-surface and pseudo-normal $\frac{\partial \boldsymbol{\varphi}}{\partial \xi_\alpha}$ and $\frac{\partial \mathbf{n}}{\partial \xi_\alpha}$ can be calculated using

$$\begin{aligned} \left[\frac{\partial \boldsymbol{\varphi}}{\partial \xi_1}, \frac{\partial \boldsymbol{\varphi}}{\partial \xi_2} \right]_s &= \left[\sum_{\mathcal{N}_s} (\boldsymbol{\varphi}_i - \boldsymbol{\varphi}_s) \otimes \nabla_\xi W_{sj} S_i \right] \mathbf{L}' \tilde{\mathbf{L}}_s^{-1}, \\ \left[\frac{\partial \mathbf{n}}{\partial \xi_1}, \frac{\partial \mathbf{n}}{\partial \xi_2} \right]_s &= \left[\sum_{\mathcal{N}_s} (\mathbf{n}_i - \mathbf{n}_s) \otimes \nabla_\xi W_{si} S_i \right] \mathbf{L}' \tilde{\mathbf{L}}_s^{-1}, \end{aligned} \quad (40)$$

where the correction matrix $\tilde{\mathbf{L}}_s$ is

$$\tilde{\mathbf{L}}_s = \mathbf{L}'^T \left[\sum_{\mathcal{N}_s} \mathbf{Q}_{0s}^T (\boldsymbol{\varphi}_{0i} - \boldsymbol{\varphi}_{0s}) \otimes \nabla_\xi W_{si} S_i \right] \mathbf{L}', \quad (41)$$

with the gradient of the kernel function calculated as

$$\nabla_\xi W_{si} = \nabla_\xi W(\|\boldsymbol{\varphi}_{0si}\|, h) = \frac{\mathbf{Q}_{0s}^T (\boldsymbol{\varphi}_{0s} - \boldsymbol{\varphi}_{0i})}{h \|\boldsymbol{\varphi}_{0s} - \boldsymbol{\varphi}_{0i}\|} \frac{\partial W}{\partial q} \Big|_{\|\boldsymbol{\varphi}_{0s} - \boldsymbol{\varphi}_{0i}\|/h}. \quad (42)$$

Herein, $\mathbf{L}'$ is

$$\mathbf{L}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T, \quad (43)$$

which is used to extract the convected vectors on the shell surface. The process above is depicted in Figure 4(b). The calculation of gradients in Equation (40) is very robust, because $\tilde{\mathbf{L}}_s$ is calculated in the reference configuration, and it is not possible to be singular as long as the particle distribution is not extremely distorted in the beginning of the simulation.

Equations 37–43 provide a general method for calculating the gradient of field values in the initial curvilinear system, which overcomes the difficulty of strain estimation for shells using standard MPM. Based on this method, the gradients $\frac{\partial \langle \bullet \rangle}{\partial \xi_\alpha}$ of any field values ($\bullet$) can be calculated effortlessly. After $\frac{\partial \boldsymbol{\varphi}}{\partial \xi_\alpha}$ and $\frac{\partial \mathbf{n}}{\partial \xi_\alpha}$ are obtained, the deformation gradient $\mathbf{F}$ can be derived using Equation (5), followed by the stress resultants $\mathbf{N}$ and $\mathbf{M}$ in Equation (12) with a Gaussian quadrature method.

4.3. Incremental stress update at stress points

To calculate $\mathbf{P}$ (shown in Equation (12)), the main task is to update $\boldsymbol{\sigma}$. For the shell structure, the plane stress state is assumed since the shell is usually relatively thin compared to the structure scale. Normally, the shell usually undergoes large rotations, which requires that the stress state be estimated considering different current configurations through the corresponding orthogonal rotation matrix $\mathbf{Q}$, which correlates the current coordinates to the global coordinates. Based on $\mathbf{Q}$, the tensor state in the global coordinates $\mathbb{S}$ can be transformed into that in the local coordinates $\hat{\mathbb{S}}$, written as $\hat{\mathbb{S}} = \mathbf{Q}^T \mathbb{S} \mathbf{Q}$. To obtain $\mathbf{Q}$, the normal is calculated using $\mathbf{t} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|}$ in the current configuration. Then we have $\mathbf{Q} = [\mathbf{e}_1^l \ \mathbf{e}_2^l \ \mathbf{t}]$, with $\mathbf{e}_1^l = \mathbf{a}_1 / \|\mathbf{a}_1\|$ and $\mathbf{e}_2^l = \mathbf{t} \times \mathbf{e}_1^l$.

To update the stress in an incremental way, the incremental strain tensor $\Delta \mathbf{d}_m$ given a sub-time increment is calculated as

$$\Delta \mathbf{d}_m = \frac{1}{2} (\Delta \mathbf{F}_m \mathbf{F}_m^{-1} + \mathbf{F}_m^{-T} \Delta \mathbf{F}_m^T), \quad (44)$$

where

$$\Delta \mathbf{F} = \left[\frac{\partial(\Delta \boldsymbol{\varphi} + \xi_3 \Delta \mathbf{n})}{\partial \xi_1}, \frac{\partial(\Delta \boldsymbol{\varphi} + \xi_3 \Delta \mathbf{n})}{\partial \xi_2}, \Delta \mathbf{n} \right] [\mathbf{G}_{(\alpha)}(\xi_1, \xi_2, \xi_3)]^{-1} = [\Delta \mathbf{g}_{(\alpha)}] [\mathbf{G}_{(\alpha)}]^{-1}. \quad (45)$$

Then Equation (44) becomes

$$\Delta \mathbf{d}_m = \frac{1}{2} \left([\Delta \mathbf{g}_{(\alpha)}]_m [\mathbf{g}_{(\alpha)}]_m^{-1} + [\mathbf{g}_{(\alpha)}]_m^{-T} [\Delta \mathbf{g}_{(\alpha)}]_m^T \right). \quad (46)$$

Here, we update field values in the sub-time step m , i.e., prediction and correction steps, due to the Newmark- β method in Section 3.4. Considering the large rotation motion of the shell and objective stress rate, the Kirchhoff stress tensor $\boldsymbol{\tau} = J \boldsymbol{\sigma}$ can be updated using

$$\boldsymbol{\tau}_m = \mathbb{Q}_m \boldsymbol{\tau}_{m-1} \mathbb{Q}_m^T + \mathbb{Q}_m (\mathbf{C}_e \Delta \hat{\mathbf{d}}_m) \mathbb{Q}_m^T, \quad (47)$$

where $\mathbb{Q}_m$ is the relative strain tensor, written as $\mathbb{Q}_m = \mathbf{F}_m \mathbf{F}_{m-1}^{-1} = [\mathbf{g}_{(\alpha)}]_m [\mathbf{g}_{(\alpha)}]_{m-1}^{-1}$. The incremental strain tensor should be estimated in local coordinates as $\Delta \hat{\mathbf{d}} = (\Delta \hat{d}_{11}, \Delta \hat{d}_{22}, \Delta \hat{d}_{12}, \Delta \hat{d}_{23}, \Delta \hat{d}_{31})^T$. For linear elasticity, the constitutive matrix $\mathbf{C}_e$ is expressed as

$$\mathbf{C}_e = \begin{bmatrix} \mathbf{C}_p & 0 \\ 0 & \mathbf{C}_s \end{bmatrix}, \quad \mathbf{C}_p = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix}, \quad \mathbf{C}_s = \kappa \frac{E}{1+\nu} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (48)$$

where $\mathbf{C}_p$ and $\mathbf{C}_s$ are the stiffness matrices for the plane stress state and the shear state in the transverse direction, with E and ν the Young's modulus and Poisson's ratio, respectively. For the RM shell, the shear correction factor should be chosen as $\kappa = 5/6$ [82]. The stress component $\hat{\tau}_{33} = 0$ is enforced considering the plane stress state. Elastoplastic deformation is also considered in this study, and it can be incorporated effortlessly based on the incremental stress update mentioned above. The readers are referred to [8] for more details regarding elastoplastic deformation. After the stress tensor is updated, the stress resultants $\mathbf{N}_s$ and $\mathbf{M}_s$ can be obtained, as shown in Figure 4(b).

4.4. Stress resultants at shell particles

In Equation (27), the subscript 'p' includes all the shell particles and stress points. Though the stress results are obtained at the stress points, $\mathbf{N}_i$ and $\mathbf{M}_i$ are not known at shell particles. Here, stress results of shell particles are obtained through the kernel function with interpolation from the neighboring stress points. The neighboring stress points of a shell particle i , forming a set $\mathcal{N}'_i$, are searched within the radius of $1.4\Delta x$. Then for a stress resultant $\langle \bullet \rangle_i$, it can be interpolated as

$$\langle \bullet \rangle_i = \left(\sum_{\mathcal{N}'_i} \langle \bullet \rangle_s W_{is} S_s \right) / \sum_{\mathcal{N}'_i} W_{is} S_s, \quad (49)$$

where we refer to Equation (38) for notation. This process is depicted in Figure 4(c).

With all the stress tensors calculated at the sub-time step m , the acceleration can be derived, and then the field values are updated to the sub-time step $m+1$. For details, the readers are referred to Algorithm 1.

5. Other Issues

5.1. Enhanced numerical stability

The pseudo-normal in the RM shell model is usually subject to hourglass instability. In [8], three anti-hourglass terms were proposed to eliminate this instability; we adopt their ideas and adapt them to our MPM shell setting. The key idea of this method is to calculate the hourglass amplitude to resist the hourglass mode. The hourglass amplitude of the pseudo-normal between shell particles i and j can be calculated using

$$\Delta \mathbf{n}_{ij} = -\frac{1}{4} [\mathbf{n}_{i,\xi} + \mathbf{n}_{j,\xi}] (\mathbf{Q}_{0i} + \mathbf{Q}_{0j}) (\boldsymbol{\varphi}_{0j} - \boldsymbol{\varphi}_{0i}) + \mathbf{n}_j - \mathbf{n}_i, \quad (50)$$

where $\mathbf{n}_{i,\xi}$ and $\mathbf{n}_{j,\xi}$ can be calculated using the method described in Section 4.2. Based on Equation (50), the anti-hourglass term is expressed as

$$\ddot{\mathbf{n}}_i^{hg} = \frac{\chi E}{\rho \zeta^2} \sum_{\mathcal{N}_i} \|\Delta \mathbf{n}_{ij}\| \Delta \mathbf{n}_{ij} W_{ij}(\|\boldsymbol{\varphi}_{0ij}\|, h) S_j, \quad (51)$$

where $\ddot{\mathbf{n}}_i^{hg}$ can effectively suppress the hourglass mode in the pseudo-normal. This term is then added to the acceleration $\ddot{\mathbf{n}}_i$ in Equation (29), which is obtained during a G2P step. One can quickly observe that $\ddot{\mathbf{n}}_i^{hg}$ is anti-symmetric, which can ensure momentum conservation. Here, χ is a coefficient that is selected as $\chi = 0.1$ in the present work. Numerical experiments show that varying this coefficient does not significantly affect simulation accuracy, as discussed in Section 6.2.1, but the presence of this term considerably enhances the numerical stability (see also Section 6.2.2).

5.2. Contact treatment

MPM can naturally handle contact without specific treatments, and this feature is utilized in the present MPM shell method. Unlike in prior sections, where we focused on the total Lagrangian framework, the updated Lagrangian framework is leveraged for the simulation of contact behavior. In this context, first, the mid-surface increment at the grid node I can be calculated according to Equation (28) in the P2G step as

$$\Delta \boldsymbol{\varphi}_I = \sum_p^{N_p} \Delta \boldsymbol{\varphi}_p \bar{S}_{Ip} / \sum_p^{N_p} \bar{S}_{Ip}. \quad (52)$$

Then, the gradient of the mid-surface increment at particle p can be calculated in the G2P step as

$$\nabla (\Delta \boldsymbol{\varphi})_p = \nabla \bar{S}_{Ip} (\Delta \boldsymbol{\varphi}_I). \quad (53)$$

Given the gradient of the mid-surface increment, the shell deformation can be derived. In this study, we only consider normal contact behavior, and we can therefore extract the strain increment in the shell normal direction, denoted with $\langle \Delta \varepsilon \rangle$, expressed as

$$\langle \Delta \varepsilon \rangle = \frac{1}{2} \mathbf{e}^T \mathbf{Q}^T \left[\nabla (\Delta \boldsymbol{\varphi})_p + \nabla (\Delta \boldsymbol{\varphi})_p^T \right] \mathbf{Q} \mathbf{e}, \quad (54)$$

with $\mathbf{e} = (0, 0, 1)^T$. Then the strain in the shell normal direction can be updated as

$$\langle \varepsilon \rangle_{p,m+1} = \langle \varepsilon \rangle_{p,m} + \langle \Delta \varepsilon \rangle_{p,m}. \quad (55)$$

Only using Equation (54) usually leads to error accumulation. To solve this issue, $\langle \varepsilon \rangle$ is reinitialized every 100 time steps. In the reinitialization, the shell particle neighbors $\mathcal{N}_p^C$ within the radius of $2.2\Delta x$ are constructed for shell particle p , and for shell particle j in $\mathcal{N}_p^C$, if $\|\boldsymbol{\varphi}_{0p} - \boldsymbol{\varphi}_{0j}\| > 4\Delta x$, then shell particle p is under potential contact, with the contact mark $M^C = 1$, otherwise $M^C = 0$. Then we have

$$\langle \varepsilon \rangle = \begin{cases} \langle \varepsilon \rangle, & M^C = 1, \\ 0, & M^C = 0. \end{cases} \quad (56)$$

Algorithm 1 Updating field values at shell particles in a sub-time step, using the MPM shell method.

- 1: Interpolate $\dot{\varphi}_s, \dot{\theta}_s, \varphi_s, \theta_s, \Delta\varphi_s, \mathbf{n}_s$ and $\Delta\mathbf{n}_s$ for stress points from shell particles [Eq. (38)].
 - 2: Calculate $\left[\frac{\partial\varphi}{\partial\xi_\alpha}\right]_p, \left[\frac{\partial\mathbf{n}}{\partial\xi_\alpha}\right]_p, \left[\frac{\partial\Delta\varphi}{\partial\xi_\alpha}\right]_p, \left[\frac{\partial\Delta\mathbf{n}}{\partial\xi_\alpha}\right]_p$ at shell particles and stress points [Eq. (40)];
 - 3: **for** Gauss points g **do**
 - 4: Calculate $\mathbf{F}_s^g, \Delta\mathbf{d}_s^g, \boldsymbol{\tau}_s^g$ [Eqs. (5, 46, 47)] at stress points;
 - 5: **end for**
 - 6: Calculate $\mathbf{N}_s, \mathbf{M}_s, \mathbf{N}_s, \mathbf{M}_s$ at stress points;
 - 7: Interpolate $\mathbf{N}_i, \mathbf{M}_i, \mathbf{N}_i, \mathbf{M}_i$ for shell particles from stress points;
 - 8: P2G: Loop over the neighbors of a grid node I , including the shell particles and stress points, to calculate the force terms, $\mathbf{N}_p, \mathbf{f}_p, \mathbf{M}_p, \mathbf{N}_{(3)p}, (\mathbf{m} \times \mathbf{n})_p$ [Eqs. (40, 47, 12, 22, 10)], and mass terms, I_{0p}, I_{2p} [Eq. (8)], on the right-hand and left-hand sides, respectively [Eq. (27)]. Also, the velocity at node I is obtained [Eq. (28)].
 - 9: Accelerations on the node $\ddot{\varphi}_I$ and $\ddot{\mathbf{n}}_I$ are derived (For Dirichlet boundary conditions, set $\ddot{\varphi}_I$ and $\ddot{\mathbf{n}}_I$ as prescribed, Section 5.3);
 - 10: G2P: Loop over the grid node neighbors of a shell particle i to get $\ddot{\varphi}_i, \ddot{\mathbf{n}}_i, \dot{\varphi}_i$ and $\dot{\theta}_i$ [Eqs. (29) and (30)] (For Dirichlet boundary conditions, set $\ddot{\varphi}_i, \ddot{\mathbf{n}}_i, \dot{\varphi}_i$ and $\dot{\theta}_i$ as prescribed, Section 5.3);
 - 11: Add the stabilization term $\ddot{\mathbf{n}}_i^{hg}$ [Eq. (51)] into $\ddot{\mathbf{n}}_i$;
 - 12: Use Newmark- β scheme [Eq. (31) or (34)] to update the field values, $\dot{\varphi}_i, \dot{\theta}_i, \varphi_i, \theta_i$ and $\Delta\varphi_i$, followed by $\mathbf{n}_i$ and $\Delta\mathbf{n}_i$ [Eq. (35)].
-

After the normal strain is calculated, the normal stress can be estimated as

$$\langle\sigma\rangle = \eta E \langle\varepsilon\rangle, \quad (57)$$

where η is a coefficient that is analogous to the scale factor used in the normal contact stiffness treatment [83]. The coefficient η is selected as 0.05 in this study, and the coefficient effect on simulation results is discussed in Section 6.3. Finally, the contact stress tensor $\mathbf{N}^C$ is calculated as

$$\mathbf{N}^C = \zeta \mathbf{Q} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \langle\sigma\rangle \end{bmatrix} \mathbf{Q}^T. \quad (58)$$

$\mathbf{N}^C$ is added to $\mathbf{N}$ in Equation (27) if one needs to simulate the contact behavior. It is worth noting that this algorithm is suitable for either self- or multi-body contact.

5.3. Boundary conditions in MPM

In the present framework, the treatment of boundary conditions is straightforward. For Dirichlet boundary conditions, boundary shell particles i are assigned the associated prescribed field values. Considering that the shell particles may not be conformal to the background Cartesian nodes, the shell particles are enforced with prescribed field values, e.g., velocity, rotational velocity, etc., after the G2P process. For the background MPM grid, grid nodes within a radius $\mu\Delta x_g$ of a corresponding boundary shell particle can be detected and determined as boundary nodes, and then the corresponding calculated accelerations in Equation (27) are set as prescribed values. Here, μ is selected as 0.5, and this value shows no significant effect on results, as discussed in Section 6.2.1. The method is practical, though not perfect; a mathematically strict enforcement of non-conforming boundary conditions is difficult, and may be considered in future work. For Neumann boundary conditions, all the external forces or moments are exerted on the shell particles and transformed into $\mathbf{f}_p$ or $\mathbf{m}_p$ as shown in Equation (10). Then the remainder of the simulation loop proceeds as usual.

5.4. Quasi-static simulation

In order to simulate the case of a system in equilibrium, one needs to make the simulation achieve a steady state quickly. Under these circumstances, numerical dissipation is not important. Therefore, the

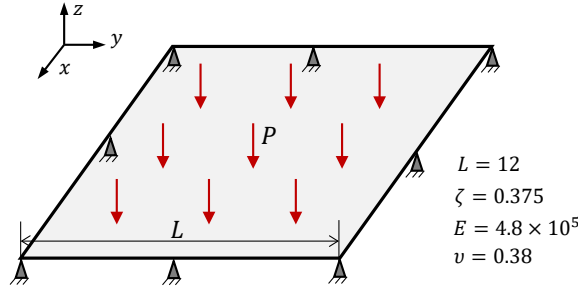


Fig. 5. Schematic of the plate under external pressure load.

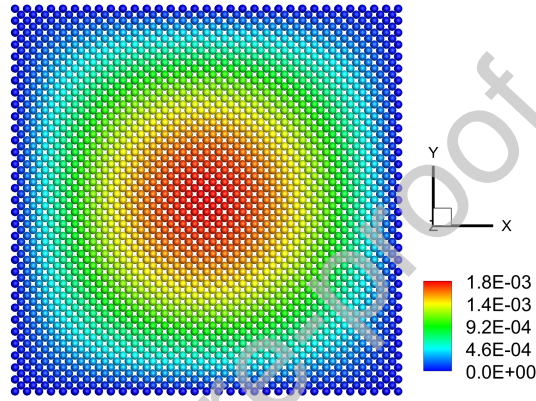


Fig. 6. Plate under pressure load: plate displacement in the z direction for $L/\zeta = 32$.

factor ϵ (which blends between PIC and FLIP) can be chosen lower than 1. In practice, ϵ is chosen in a range of 0.5–0.95 in this study. The choice of this value does not affect accuracy; it only affects the rate at which equilibrium is reached.

In order to further clarify the steps of the MPM shell method proposed above, pseudocode of the entire algorithm is given in Algorithm 1, where the update of field values in one sub-time step is presented for simplicity.

6. Numerical Validations

6.1. Plate under pressure load

In our first numerical test case, a square plate with its four edges simply supported is leveraged to verify the accuracy of the proposed MPM for thin shells. Figure 5 shows the plate under an external pressure load $P = 0.05$, where the dimensions and the material parameters are also given. Quasi-static simulation (Section 5.4) is used to achieve the equilibrium state of the plate, and the deflection at the plate center is measured for comparison with a reference solution. Since there is no analytical solution for the plate under large deformation, we compare to a reference simulated using the ABAQUS software package (S4R element with element size $\Delta x = 1/80$). Figure 6 displays the plate displacement in the z direction for $L/\zeta = 32$. In order to demonstrate the convergence rate of the present MPM shell solver, various resolutions, i.e., $\Delta x_g = L/4$, $L/8$, $L/12$, $L/16$, $L/20$ and $L/32$, are used for the simulation. The error versus the grid resolution, Δx_g , is shown in Figure 7(a), where a first-order convergence rate is observed for $\zeta = 0.375$, i.e., $L/\zeta = 32$. The first-order convergence is mainly due to the adopted first-order gradient estimation of the field values using the kernel function method.

To further estimate the performance of the MPM shell solver, the shell is set as very thin— $L/\zeta = 320$ and $L/\zeta = 1600$. In these scenarios, employing traditional MPM is infeasible, because traditional MPM would

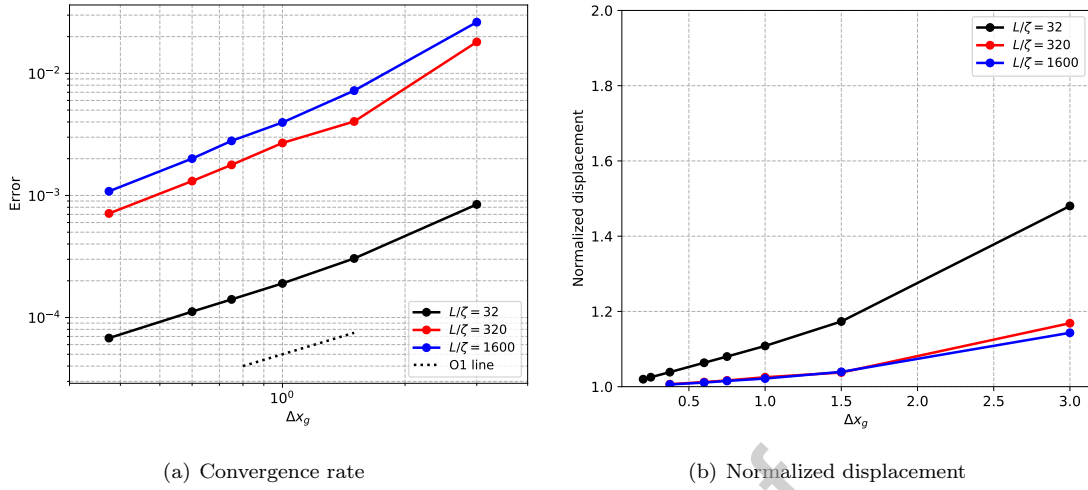


Fig. 7. Plate under pressure load: (a) convergence rate and (b) normalized displacement versus grid cell size for different L/ζ .

require multiple grid cells in the thickness direction overlapping the shell in order to resolve the transverse stress, which will incur extreme computational expense. In addition, traditional MPM also encounters the so-called “shear locking” phenomenon, and this means an overestimation of the transverse shear stress, which makes the result unable to converge to the ground truth even when increasing the mesh resolution. Figure 7(a) shows that the convergence rate of our method still holds even when the plate is extremely thin for $L/\zeta = 320$ and $L/\zeta = 1600$. The normalized displacements for various thicknesses are shown in Figure 7(b). One can observe that when increasing the resolution, the predicted results gradually approach the reference. Thus, no locking effect is seen in this MPM shell method. This is because the gradient estimation is through a kernel function method, which circumvents the low-order shape function for gradient estimation used in the traditional MPM. In fact, the present gradient estimation is akin to the reproducing kernels of RKPM. In a similar way, the employed gradient estimation with a correction matrix can also ensure first-order reproducing ability, which is important in RM shells to eliminate shear locking. Since it can calculate gradients of mid-surface displacement, the shear locking issue does not occur. Noticeably, regarding $L/\zeta = 1600$ with the resolution $\Delta x_g = 1$, i.e., $\Delta x_g/\zeta = 50$, the present MPM shell method can achieve a relative error less than 0.02. Employing $\Delta x_g/\zeta = 50$ in traditional MPM to pursue a highly accurate result would be almost impossible, because it is not practical for a shape function to be high-order enough to resolve the strain in the thickness direction under such an extremely large grid cell size-to-thickness ratio, $\Delta x_g/\zeta$.

The proposed MPM shell method can allow large $\Delta x_g/\zeta$ because the strain in the thickness direction is estimated on the shell surface using the kernel function. In addition, the adopted kernel function eliminates the possibility of the shear locking effect. Therefore, the grid cell size is not necessary to be smaller than the thickness. This feature presents a clear advantage of the present MPM shell method over traditional MPM approaches for thin structures.

6.2. Beam and flag

6.2.1. Beam vibration

Two-dimensional beam vibration is a typical case to test the performance of a shell solver. An advantage of the material point method, and in turn, our shell MPM, is that there is no difficulty in translating our three-dimensional formulation to 2D. For 2D, the curvilinear system (ξ_1, ξ_2, ξ_3) becomes (ξ_1, ξ_2) , and the governing equations follow the same way in Equation (15) by removing one dimension. As shown in Figure 8(a), the beam is set with an initial inclination angle of 18° , and is subject to a horizontal gravity $f_x = 10$. We choose $L = 1$ for simplicity. The material parameters are given in the schematic. This case is dynamic, and the quasi-static simulation is abandoned.

First, the beam thickness is selected as $\zeta = 0.02$, and two boundary conditions, i.e., pinned ($\ddot{\varphi}=0$) and clamped ($\ddot{\varphi}=0, \ddot{\theta}=0$), are enforced at point A. For the two boundary conditions, Figure 9 displays

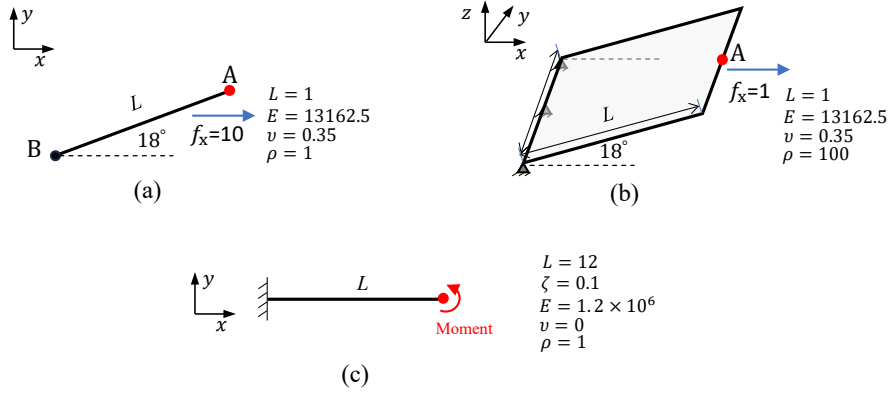


Fig. 8. Schematic of the (a) beam and (b) plate under horizontal gravity, and (c) beam under pure bending.

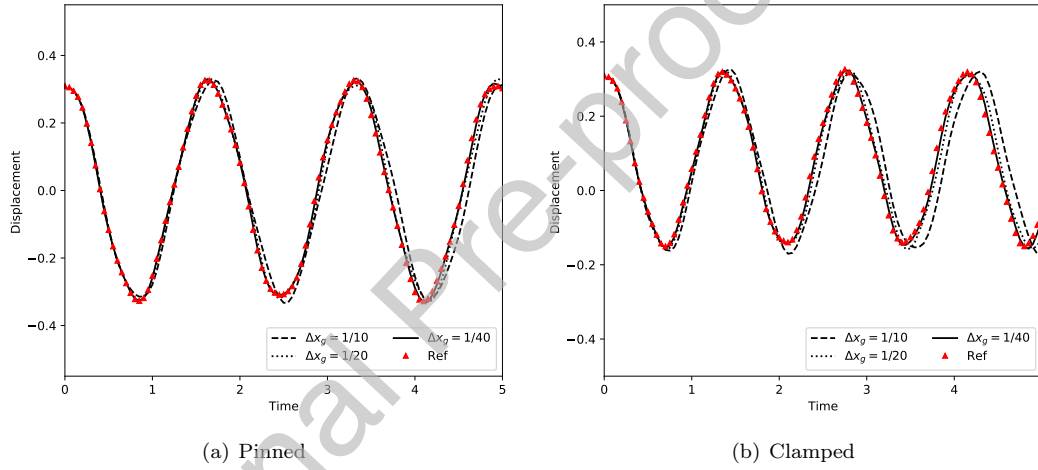


Fig. 9. Beam vibration: the beam deflection subject to a horizontal gravity simulated using various grid resolutions, where two kinds of boundary conditions, (a) pinned and (b) clamped, are deployed.

the beam deflection history at the tip simulated using various resolutions, where the reference results are from ABAQUS software package (B21 element with element size $\Delta x = 1/100$). One can clearly observe a convergence trend as the grid resolution increases. Figure 10 depicts the beam deformation profiles of the two boundary conditions using converged resolution $\Delta x_g = L/40$. It clearly shows that the von Mises stress field is quite smooth without oscillations, which indicates that the present method is quite stable.

(i) Total and updated Lagrangian frameworks

The total and updated Lagrangian frameworks are both leveraged for simulation. Under resolution $\Delta x_g = L/20$, the numbers of shell particles and stress points are 21 and 20, respectively, and the number of grid nodes is 40,401. The time step is 0.00011, and the simulation wall times for the two frameworks are shown in Table 1. Figure 11 illustrates the simulated results using the two different frameworks unified within Equation (18). In Figure 11(a), the deformed beam is clearly seen with velocity contour and vectors, and the affected grid nodes are in the initial position due to the employed total Lagrangian framework. In contrast, the affected nodes are always surrounding the deformed beam in the updated Lagrangian framework, as shown in Figure 11(b). Figure 12(a) displays the simulated beam deflection in the two different frameworks with $\Delta x_g = L/20$, and it shows that the two curves almost overlap. This comparison demonstrates that choosing different frameworks will not lead to a significant difference in prediction accuracy. However, the

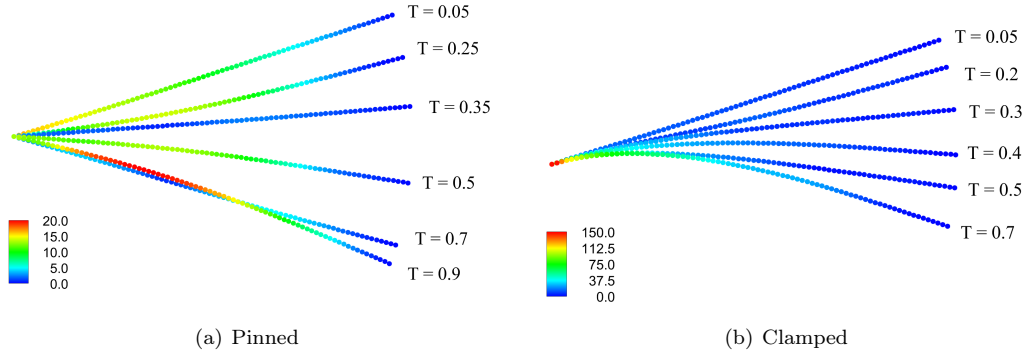


Fig. 10. Beam vibration: vibration snapshots for the (a) pinned and (b) clamped boundary conditions, where the contour indicates von Mises stress.

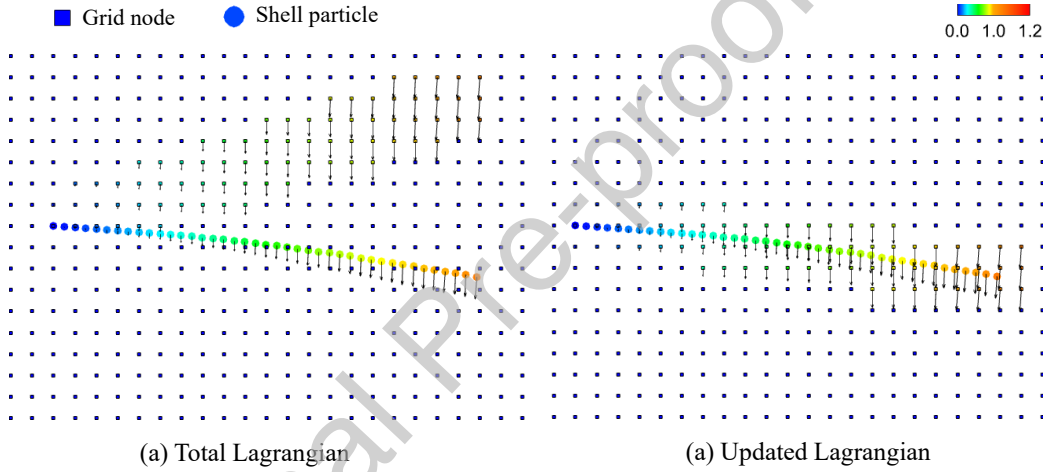


Fig. 11. Beam vibration: grid nodes and shell particles at $t = 0.5$ simulated using (a) total and (b) updated Lagrangian frameworks, where the contour and vectors indicate velocity.

major problem with the updated Lagrangian framework is that it requires a larger computational cost than the total Lagrangian version, as shown in Table 1. This is because the total Lagrangian framework only needs to construct the neighbors of cells and particles once.

(ii) Coefficient χ

In Section 5.1, an enhanced term is proposed to improve numerical stability. This term can effectively eliminate the hourglass modes in the pseudo-normal motion. Here we perform a brief ablation on χ and observe the influence on the simulation results. The clamped beam is chosen for simulation, and χ is set as $\chi = 0.001, 0.01$ or 0.1 . The beam deflections simulated using various coefficient values are plotted in Figure 12(b), and it is observed that all the curves almost overlap. This demonstrates that the coefficient χ does not affect the simulation accuracy (though, as shown in Section 6.2.2, it does improve numerical stability). In light of this, the coefficient is chosen as $\chi = 0.1$ throughout our paper.

In addition, for the enforcement of Dirichlet boundary conditions, background MPM grid nodes within a radius $\mu\Delta x_g$ of a corresponding boundary shell particle are detected and determined as boundary nodes, on which the corresponding accelerations are set as prescribed values in Equation (27). Figure 12(c) displays the results simulated using different values of coefficient μ . It shows that the differences are small, where a slight difference is that, apart from $\mu = 0$, all other values collapse to one curve and provide better simulation results. Thus, we choose $\mu = 0.5$ in this study.

Table 1: Wall time spent on a simulation period of 10 using total and updated Lagrangian frameworks, performed on two AMD EPYC 75F3 32-Core Processors using TBB.

	Total Lagrangian	Updated Lagrangian
Wall time (s)	45	635

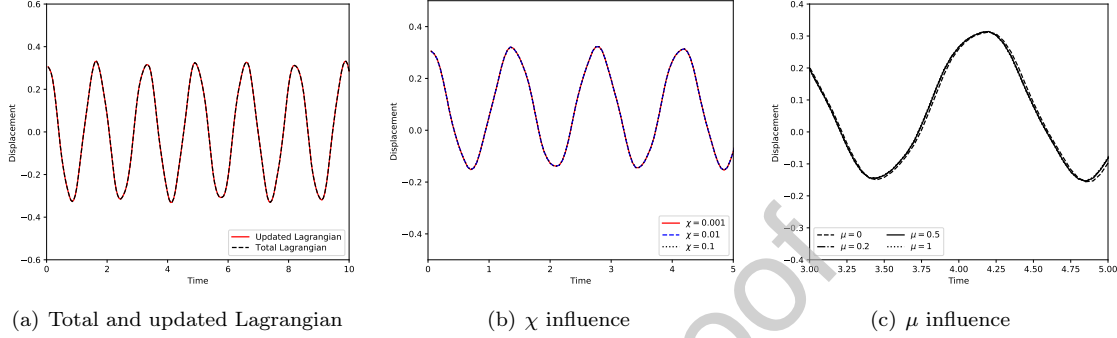


Fig. 12. Beam vibration: (a) pinned beam deflection in the total and updated Lagrangian frameworks. (b) clamped beam deflection using various values of coefficient χ . (c) clamped beam deflection using various values of coefficient μ .

(iii) Kinetic energy conservation

Kinetic energy conservation is an important issue in dynamic simulation. Although formal analysis of our MPM shell method's energy behavior is beyond the scope of the present work, we offer numerical experiments to demonstrate our method's performance. A long-term period of vibration is simulated for the two boundary conditions, and the displacement and kinetic energy are recorded in the simulation, as shown in Figure 13. In Figure 13(a), the displacement history and vibration amplitude clearly show that the amplitude keeps the original value. Similar behavior is also observed in the kinetic energy history, with the average kinetic energy remaining nearly constant throughout the long-term simulation, as shown in Figure 13(b). Our method performs similarly well for the clamped boundary condition, shown in Figure 13(c) and (d). This example demonstrates that kinetic energy is numerically well conserved in our proposed MPM shell method. Although the widely-used USL time integration shows conservation in kinetic energy [46], our study selects a Newmark- β scheme, and the results show that this scheme can also ensure the kinetic energy in MPM while allowing a much larger time step.

(iv) Very thin beam

A central advantage of the proposed shell MPM over traditional MPM is that our method can simulate thin shells with a very large grid cell size-to-shell thickness ratio $\Delta x_g/\zeta$. In order to demonstrate this, we further reduce the thickness of the beam to $\zeta = 0.005$ and $\zeta = 0.001$ for the pinned boundary condition. Figure 14(a) plots the deflection with thickness $\zeta = 0.005$ and grid size $\Delta x_g = 0.025$, and it shows that a large $\Delta x_g/\zeta = 5$ can even produce good agreement with the reference. Furthermore, an even thinner beam with $\zeta = 0.001$ is simulated using grid size $\Delta x_g = 0.0125$, and good agreement is still observed in Figure 14(b) under a very large $\Delta x_g/\zeta = 12.5$. We emphasize that, under such a large grid cell size-to-shell thickness ratio, it is hard to give an accurate prediction using the traditional MPM approach.

6.2.2. Flag flapping

The 2D beam described above is a produced from a reduction of our full 3D MPM shell algorithm. We now consider a 3D counterpart of the 2D beam: a 3D flag, shown in Figure 8(b). The left end of the flag is pinned, i.e., $\dot{\varphi}=0$, and the flag is subject to an external horizontal gravity. This case is derived from the 2D beam case, adding a spanwise width of L directed into the page. The dimensions and material parameters are shown in the figure. For simplicity, $L = 1$ is chosen, and the flag thickness is chosen as $\zeta = 0.02$. First, a set of grid resolutions is selected, i.e., $\Delta x_g = 1/10, 1/20$ and $1/40$. Then the flag deflection (in z direction)

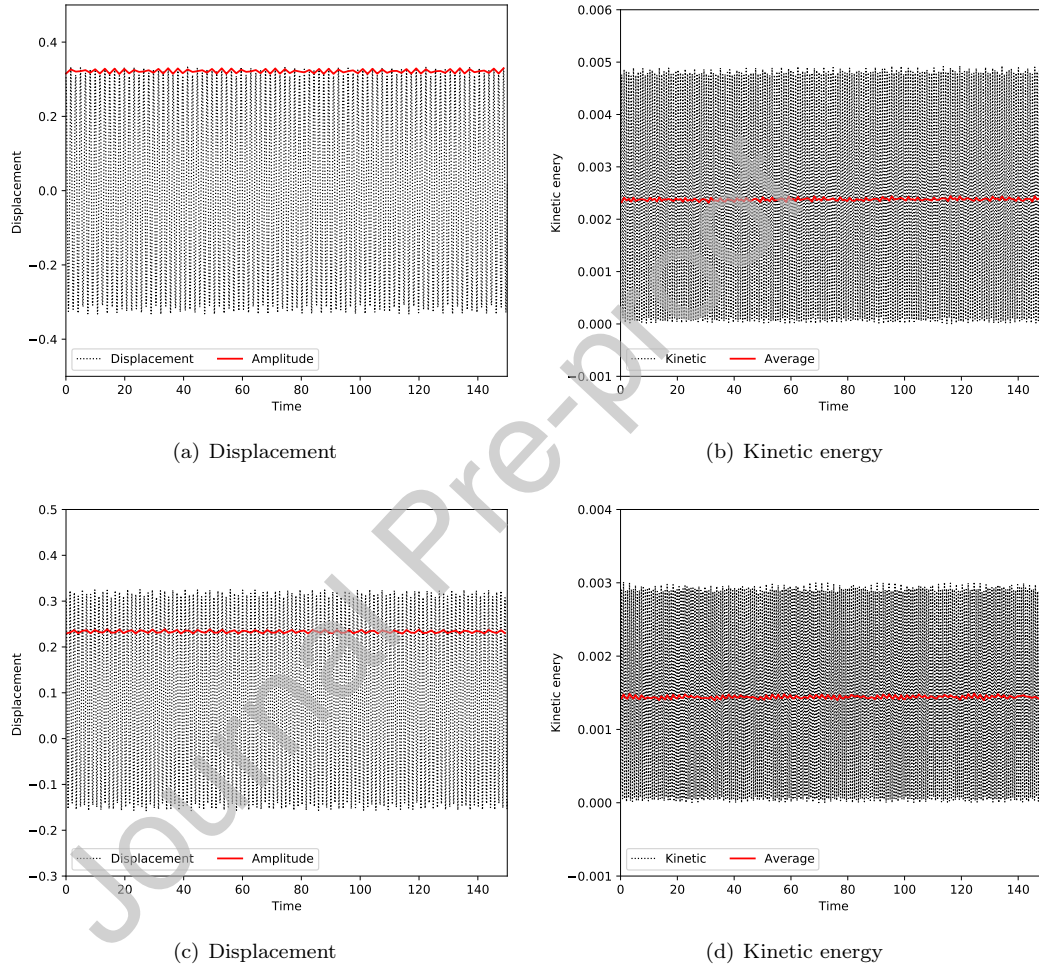


Fig. 13. Beam vibration: vibration displacement and kinetic energy in the long-term simulation, where (a) and (b) are from the pinned boundary, and (c) and (d) are from the clamped boundary.

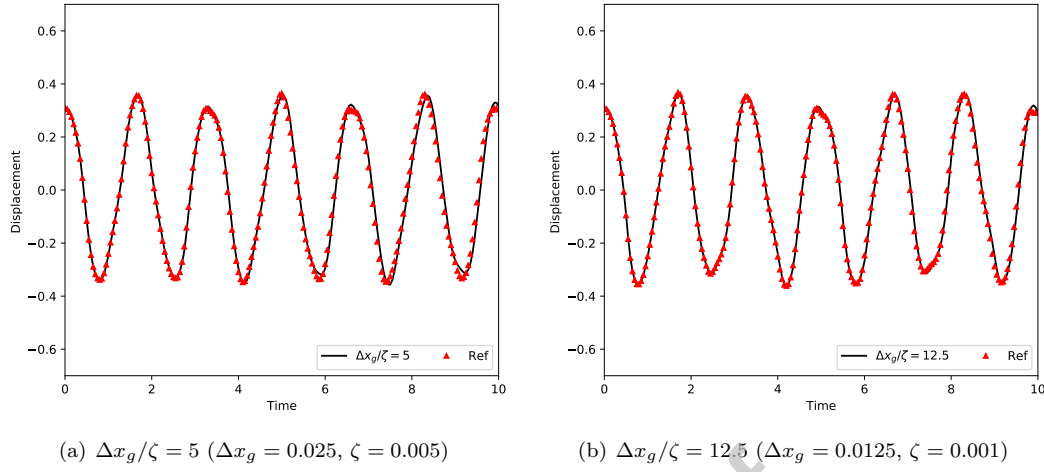


Fig. 14. Beam vibration: vibration displacement simulated using different ratios between the grid cell size and shell thickness, i.e., (a) $\Delta x_g / \zeta = 5$ and (b) $\Delta x_g / \zeta = 12.5$.

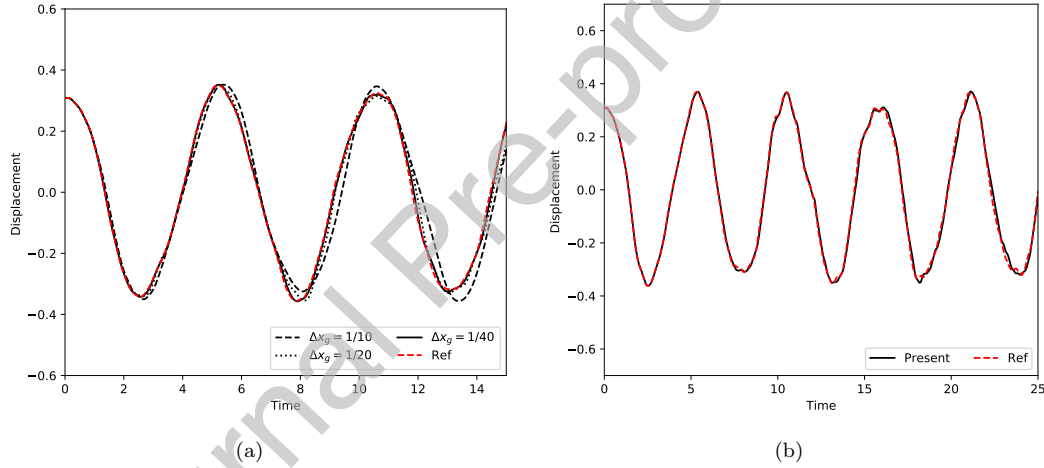


Fig. 15. Flag flapping: (a) the flag deflections in z direction (probe point A) with thickness $\zeta = 0.02$, simulated using various grid resolutions. (b) the flag deflections (probe point A) with thickness $\zeta = 0.005$, where the grid cell size is $\Delta x_g = 0.02$, i.e., $\Delta x_g / \zeta = 4$.

at the probe point A is recorded versus time, as shown in Figure 15(a). It shows that at the resolution $\Delta x_g = 1/40$, the simulation almost achieves convergence, where the reference is obtained from ABAQUS using the S4R element with the element size $\Delta x = 1/80$.

In order to further test the performance of the proposed method for a very thin shell, the flag thickness is set as $\zeta = 0.005$. Fig. 15(b) plots the predicted deflection history using the MPM shell versus the reference, where a quite good agreement with the reference is observed. The deformation flag profiles are shown in Figure 16. It shows that when the thickness is large, the flag deformation is more similar to that of a hard plate. When the flag thickness is small, the deformation will be more chaotic, similar to a real-world flag. This case further demonstrates the capability of the present MPM shell method for the simulation of very thin shells. In order to present the effect of the stabilization coefficient χ , we use $\chi = 0$ and $\chi = 0.1$ for simulation, and the simulated flag profiles at $T = 9.12$ are shown in Figure 17. It is readily apparent that $\chi = 0$ leads to hourglass modes. In contrast, $\chi = 0.1$ totally removes this instability. Similarly, the effect of deploying the stress points is also presented in Figures 17(c) and 17(d). Instability starts to occur at a very early stage, $T = 2.4$, when no stress points are used; in contrast, with stress points, the simulated velocity field is quite smooth.

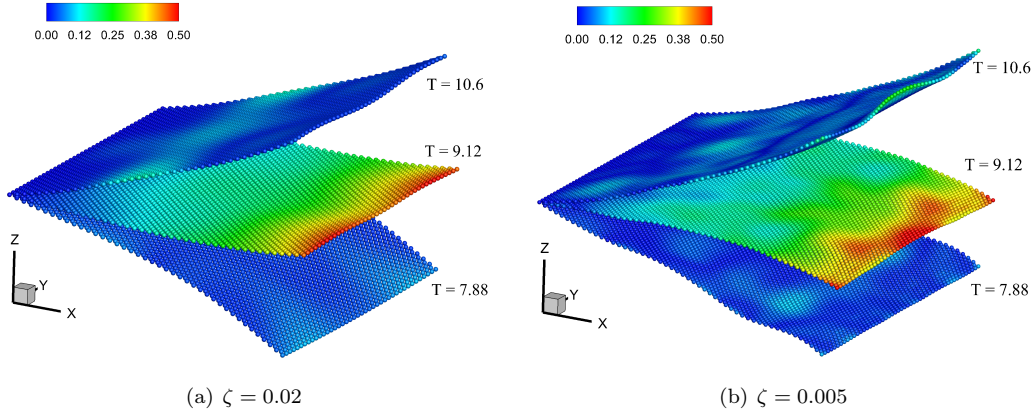


Fig. 16. Flag flapping: simulated flag profiles at different instants, with thickness (a) $\zeta = 0.02$ and (b) $\zeta = 0.005$, where the contour indicates velocity.

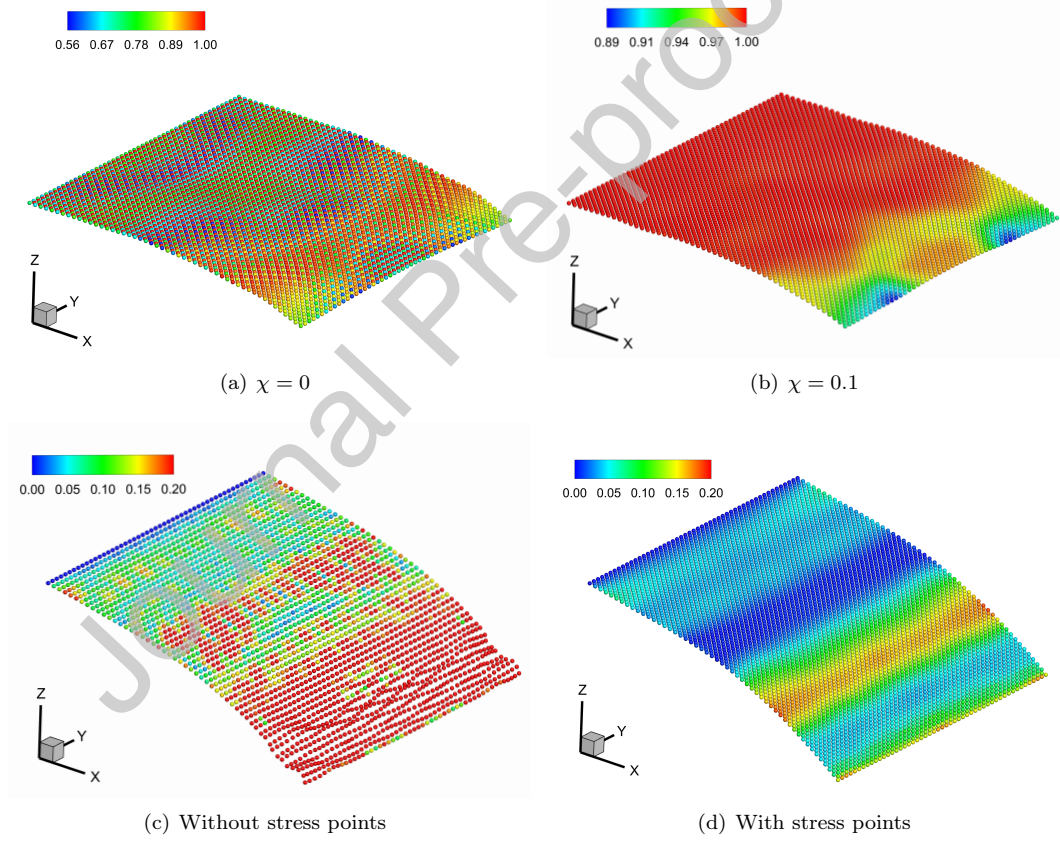


Fig. 17. Flag flapping: simulated flag profiles using different stabilization coefficients (a) $\chi = 0$ and (b) $\chi = 0.1$ at $T = 9.12$, where the contour indicates the third component of pseudo normal, and (c) without and (d) with stress points at $T = 2.4$, where the contour indicates velocity.

6.2.3. Pure bending

The pure bending beam is displayed in this section to check the performance of the method under an external moment. Figure 8(c) illustrates the case setup, where the beam is clamped at the left end, and an external moment $M \times 50\pi/3$ is exerted at the tip. All the parameters are shown in this schematic.

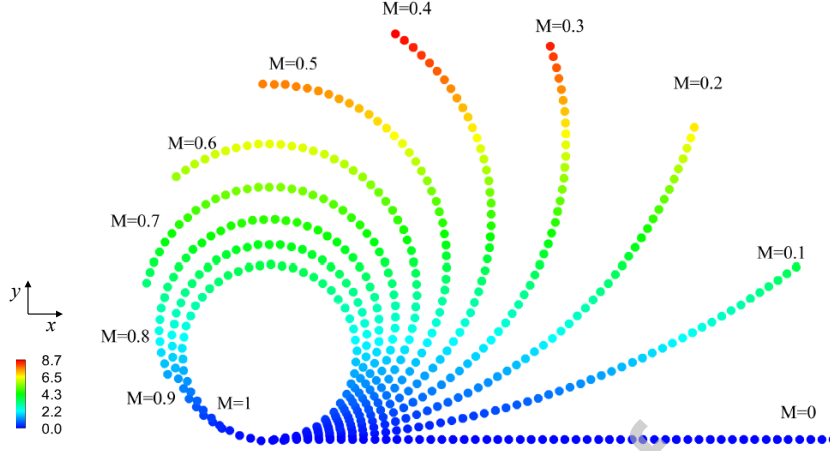


Fig. 18. Pure bending: deformed beam profiles under various bending moments, where the contour indicates the displacement in the y direction.

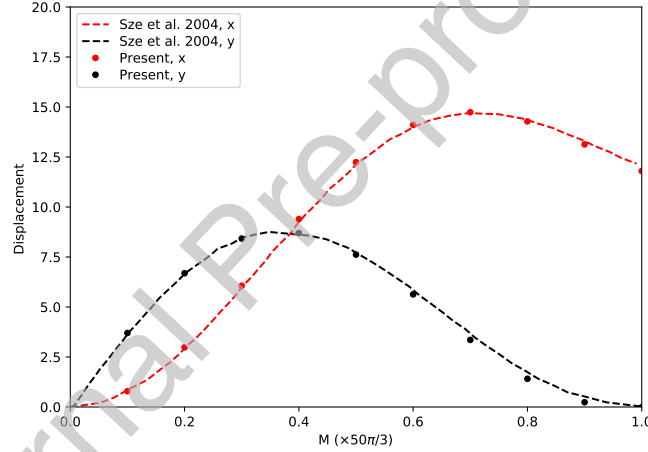


Fig. 19. Pure bending: the beam tip deflections in x and y directions under various bending moments. Comparison is shown to the reference from [84].

Quasi-static simulation is used in this case to quickly achieve the equilibrium state.

The grid resolution is set as $\Delta x_g = L/25 = 4.8\zeta$, which is about five times the thickness of the shell. The deformed beam profiles under various bending moments are illustrated in Figure 18. The figure shows that the deformed shape is similar to a circle, due to the pure moment state along the beam. The tip deflections under various bending moments are plotted in Figure 19, and they are compared to the reference given in [84]; tight agreement is observed. Note that pure bending is extremely challenging if traditional MPM is used for simulation, because pure bending highly relies on an accurate calculation of the stress along the thickness direction. In contrast, the present MPM shell method can provide very accurate predictions under a large grid cell size-to-shell thickness ratio.

6.3. Collision of rings

In order to demonstrate the integration of contact treatment with our method, the collision of two deformable rings is considered. The setup and dimensions of this case are illustrated in Figure 20(a), where two rings move towards each other with the same velocity magnitude. As introduced in Section 5.2, the updated Lagrangian framework with contact treatment is used for simulation.

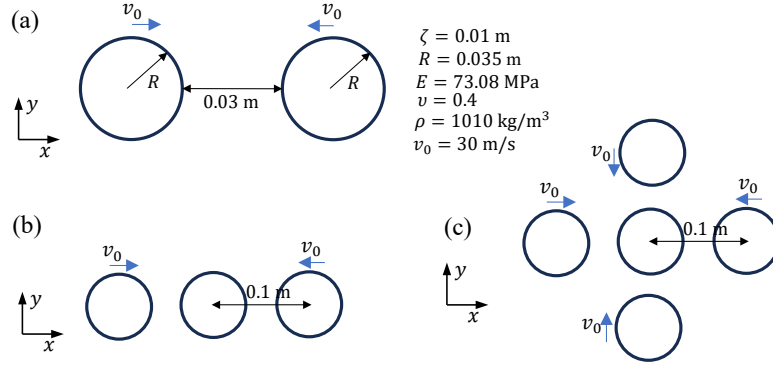


Fig. 20. Schematic of the ring collision example.

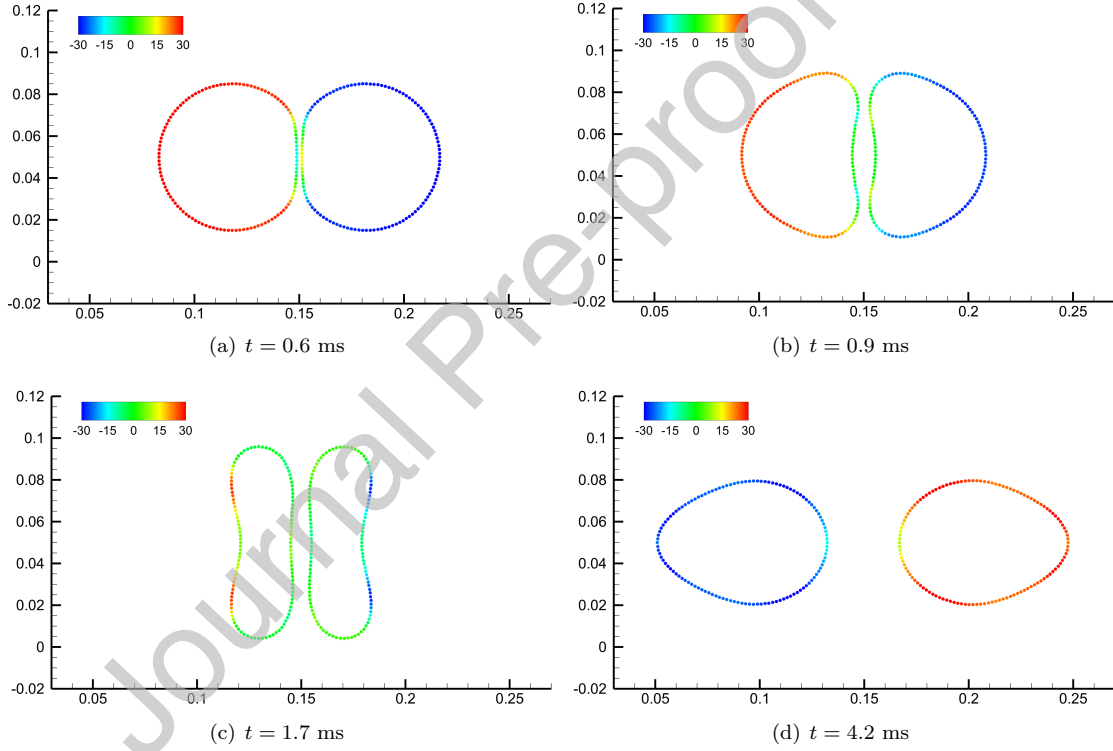
Fig. 21. Collision of rings: deformed rings at different instants simulated with $\Delta x_g = L_c/60$, where contour indicates horizontal velocity.

Figure 21 displays the deformed rings at various instants simulated with $\Delta x_g = L_c/60$, where L_c is the ring perimeter. The contact behavior is clearly shown, including both collision and separation. It is worth noting that the ring thickness is large compared to the ring radius in this case, and therefore the RM shell theory may not accurately resemble the result simulated using a volumetric model. Hence, the present simulation is slightly different from similar experiments in works such as [85]. The history of kinetic, strain and total energy is plotted versus time in Figure 22(a). It is observed that the total energy does not dissipate much after the collision using two different grid cell sizes. In contrast, the simulations in [85, 86] undergo a large energy loss after the collision process. In order to test the effect of η on simulation results, various values of η , from 0.05 to 1, are employed for simulation, and the kinetic energy over time is shown in Figure 22(b). It displays that the difference between different η values is very small. This demonstrates that the

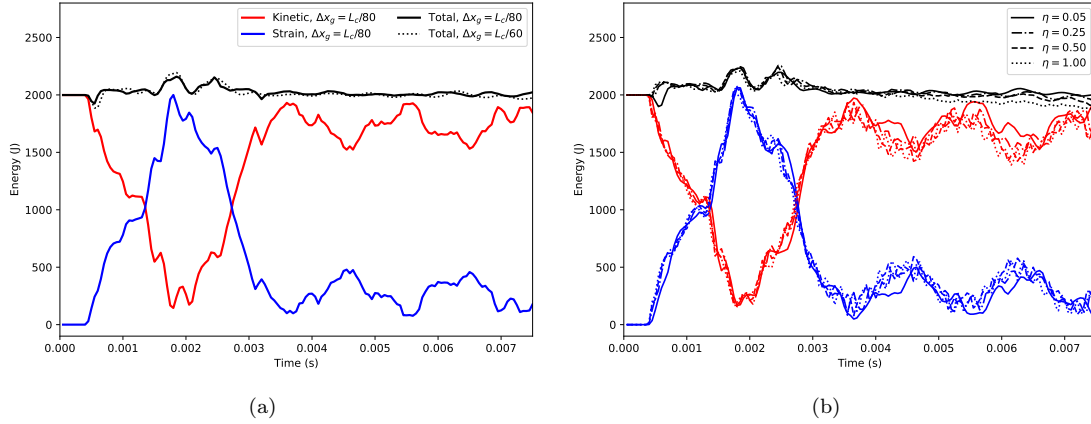


Fig. 22. Collision of rings: (a) history of kinetic, strain and total energy simulated using resolutions $\Delta x_g = L_c/60$ and $L_c/80$; (b) the effect of coefficient η (used in Equation (57) for contact treatment) on Kinetic energy, simulated using $\Delta x_g = L_c/60$, where red, blue and black colors represent kinetic, strain and total energy, respectively.

variation of η does not significantly affect the simulation results, and $\eta = 0.05$ is enough for normal contact treatment. In fact, this value is close to the scale factor 0.1 for treating the normal contact force in the method proposed by Benson et al. [83]. This demonstrates that the MPM shell method can handle contact easily without obvious energy dissipation.

To further demonstrate the capability and robustness of the proposed method in handling both self-contact and multi-body contact, simulations of three-ring and five-ring systems are conducted. The setup is illustrated in Figure 20(b), where one more cylinder is added with the distance being 0.1 m, and Young's modulus is set as $E = 40$ MPa to present self-contact behaviors. The results are displayed in Figure 23. Notably, at time $t = 3$ ms in Figure 23, both self-contact and multi-body contact occur simultaneously. Additionally, four cylinders are deployed around a centered cylinder with a distance of 0.1 m as shown in Figure 20(c). More complex contact interactions are observed at time $t = 2$ ms in Figure 24. The collision and separation processes are captured naturally in both cases. These results demonstrate that the present method can handle complex cases with a simple contact algorithm, which is naturally compatible with the MPM approach. In contrast, existing approaches often require more complicated algorithms for contact treatments, as reported in [64].

6.4. Pinched cylinder

The pinched cylinder [84] is a classic case to test the accuracy of a shell solver for large deformation. In this case, a pair of concentrated forces F are exerted on two opposite midpoints of the cylinder, as shown in Figure 25, where the material parameters are also given. Three probe points, A, B and C, are located on the cylinder wall to measure the deformation of the structure. Note that the quasi-static simulation approach is used in this case to quickly achieve the equilibrium state.

First, a resolution of $\Delta x_g = L_c/120 = 0.259$ m is deployed for the simulation, where $L_c = 2\pi R$ is the cylinder circumference. Under resolution $\Delta x_g = 0.259$ m, the numbers of shell particles and stress points are 8,640 and 8,480, respectively. Figure 26 shows the deformed cylinder under different loads. One can see that the von Mises stress field is quite smooth in the cylinder, and even under a large force of $F = 40$ kN, the simulation is still very stable. The displacements at the three probe points are recorded in Figure 27, and the results are compared to the reference given in [84], which is usually taken as the most reliable result [33, 87]. A higher resolution $\Delta x_g = L_c/160$ is also implemented for comparison, and it is observed that the two resolutions yield almost the same results. In addition, the two predictions exhibit tight agreement with the reference solution. Moreover, it shows that the prediction of the coarse resolution $\Delta x_g = L_c/120 = 0.259$ m achieves quite good accuracy. A solid-shell MPM (SSMPM) is proposed [64] in a recent study, and its simulation results using $\Delta x_g = 0.08$ m are shown in Figure 27 as comparison, where one can clearly observe that the present results are closer to the reference. It is worth noting that the present method uses the resolution $\Delta x_g = 0.259$ m, not only much larger than the shell thickness $\zeta = 0.094$ m, but also almost four

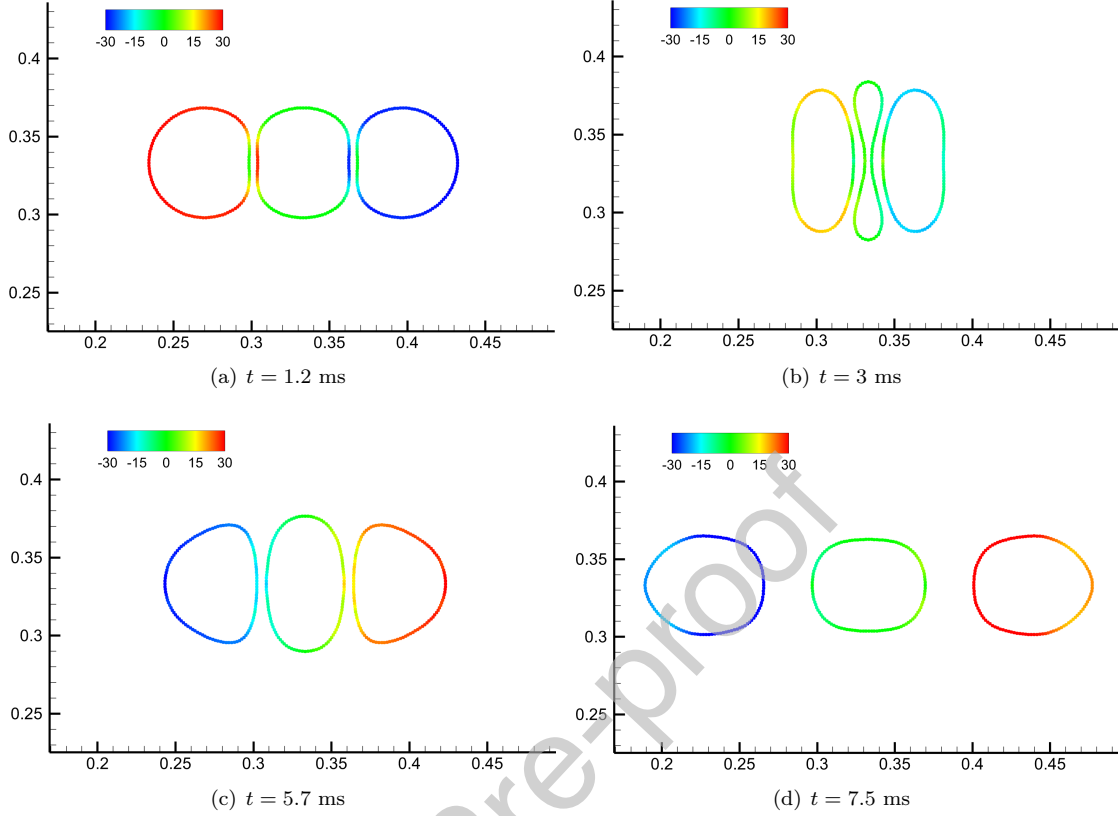


Fig. 23. Collision of rings: three deformed rings at different instants simulated with $\Delta x_g = L_c/60$, where contour indicates horizontal velocity.

times that used in SSMPM, and with a much coarser grid resolution we can yield a much more reliable prediction. This comparison demonstrates that the present method shows higher accuracy than SSMPM. In addition, the results simulated using pure MPM with $\Delta x_g = 0.094$ m reported in [64] are also exhibited in Figure 27, where the deviation from the reference is extremely large, and [64] reports that even deploying four layers of particles in the thickness direction cannot yield a reliable result. Furthermore, the results of the SPH shell proposed in [31] are also plotted in Figure 27 for comparison, of which the resolution is $\Delta x = L_c/160$. Obviously, a large deviation is observed against the reference. In contrast, our method achieves a convergent result even using a lower resolution $\Delta x_g = L_c/120$. This case once again demonstrates the advantage of our method for thin structures.

6.5. Pinched hemisphere

A pinched hemisphere example is considered in this section to test the elastoplastic performance of our MPM shell method. The setup of this case is shown in Figure 28, where the hemisphere is subject to the outward and inward force $2F$ exerting at four quarter points on the edge. The material parameters are displayed in the figure, where Y and ε_p are the yield stress and equivalent plastic strain, respectively. The quasi-static simulation approach is again used to quickly achieve equilibrium.

First, a grid resolution of $\Delta x_g = L_c/20$ is applied, with $L_c = \pi R/2$, which means $\Delta x_g = 0.157$ mm, about three times the thickness of the hemispherical shell. Under resolution $\Delta x_g = 0.157$ mm, the numbers of shell particles and stress points are 1,057 and 1,020, respectively. Figure 29 displays the deformation profiles of the hemisphere under various loads, and the contour indicates equivalent plastic strain. It shows that the strain field is quite smooth and without an apparent hourglass instability, thanks to the adopted stress points and stabilization term in Equation (51). Figure 30 presents a quantitative comparison between the present results and references from [88] and [64], in which the displacements at probe points A and B

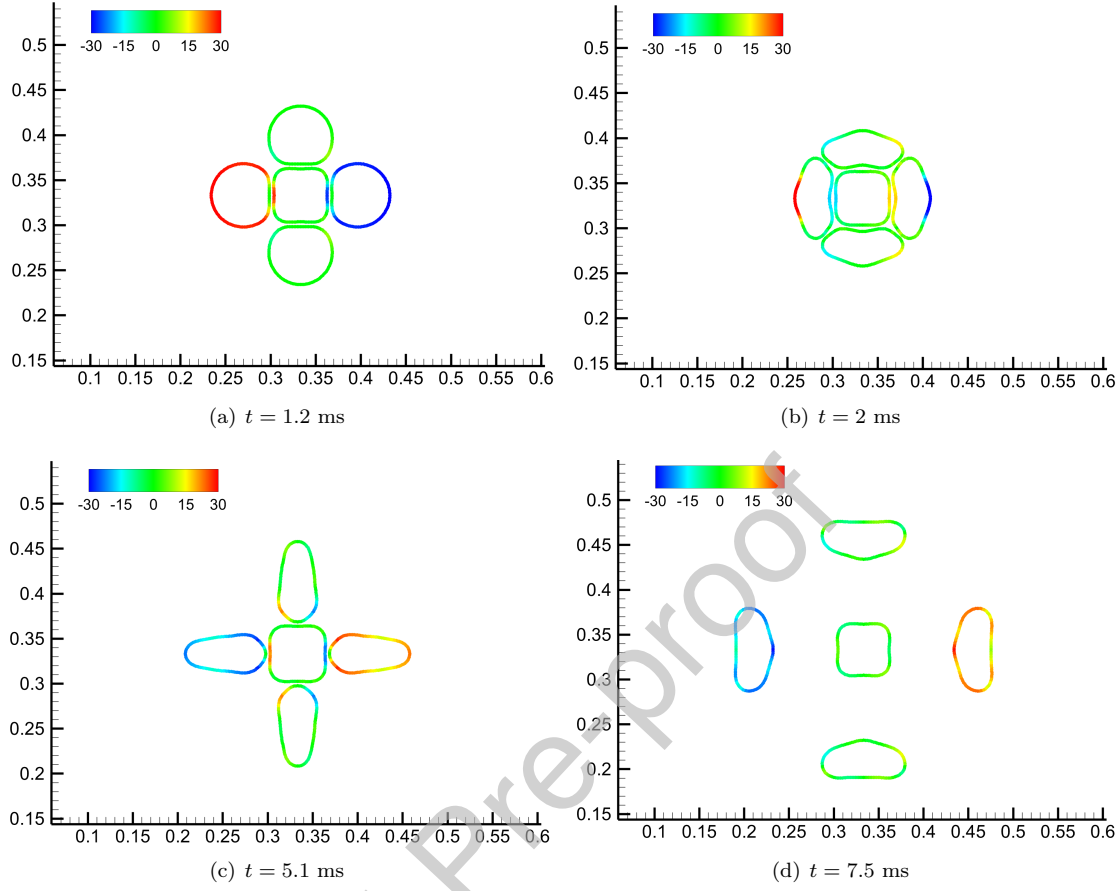


Fig. 24. Collision of rings: five deformed rings at different instants simulated with $\Delta x_g = L_c/60$, where contour indicates horizontal velocity.

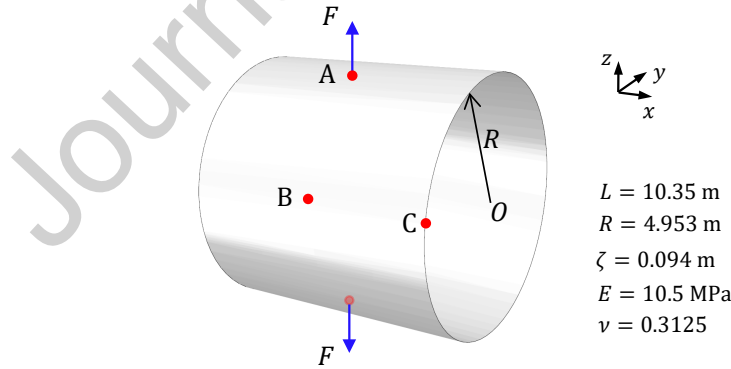


Fig. 25. Schematic of the pinched cylinder.

are plotted. It is shown that the present results agree quite well with both references. Results using grid resolution $\Delta x_g = L_c/40$ are also given in Figure 30, and we observe that the difference between higher and lower resolutions is quite small, which indicates that the lower grid cell size $\Delta x_g = L_c/20$ can achieve a reliable simulation. In [64], the deployed grid cell size of 0.4 mm can get reliable results using an improved MPM approach, where no larger resolution is tested; however, traditional MPM sees extremely large error

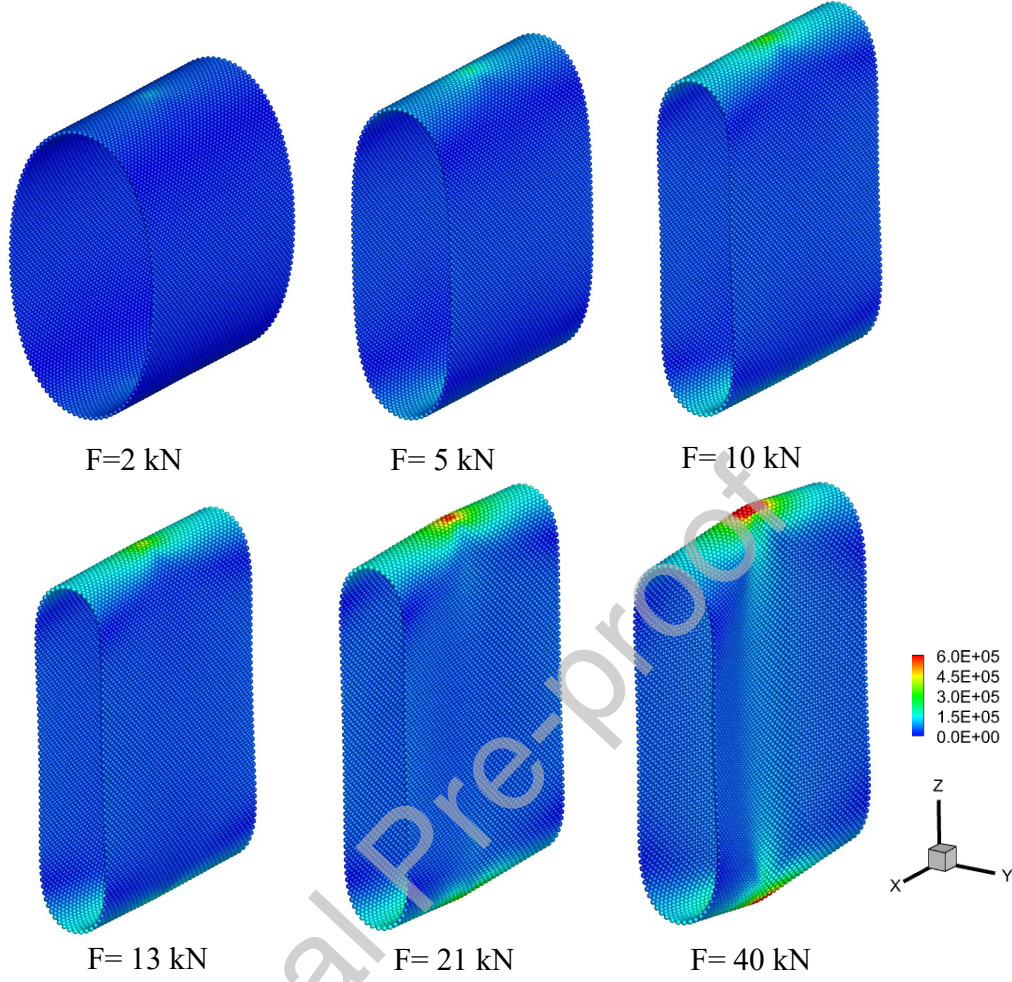


Fig. 26. Pinched cylinder: simulated deformation profiles under different loads with $\Delta x_g = L_c/120$, where the contour indicates averaged von Mises stress in the thickness direction.

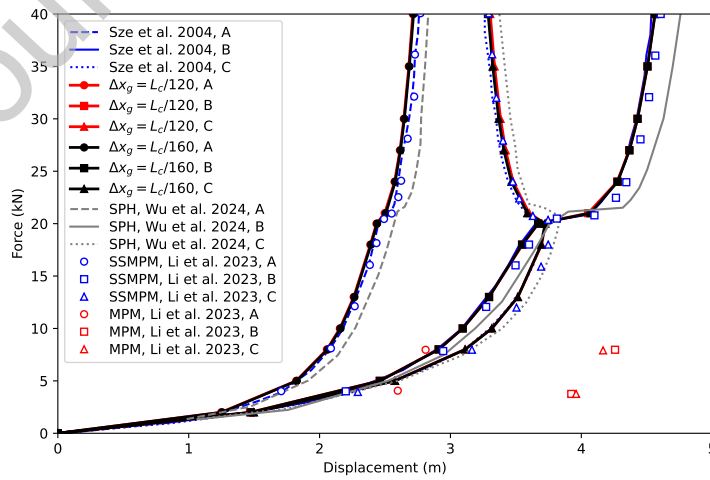


Fig. 27. Pinched cylinder: displacements at three probe points simulated using the grid resolution $\Delta x_g = L_c/120$ and $\Delta x_g = L_c/160$ under different external loads, where the widely used reference is from [84]. In addition, the results of SPH shell [31], solid-shell MPM (SSMPM) and pure MPM [64] are used as comparisons.

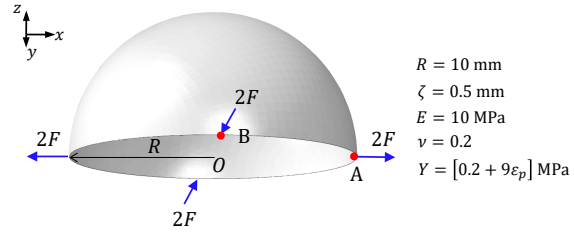


Fig. 28. Schematic of the pinched hemisphere.

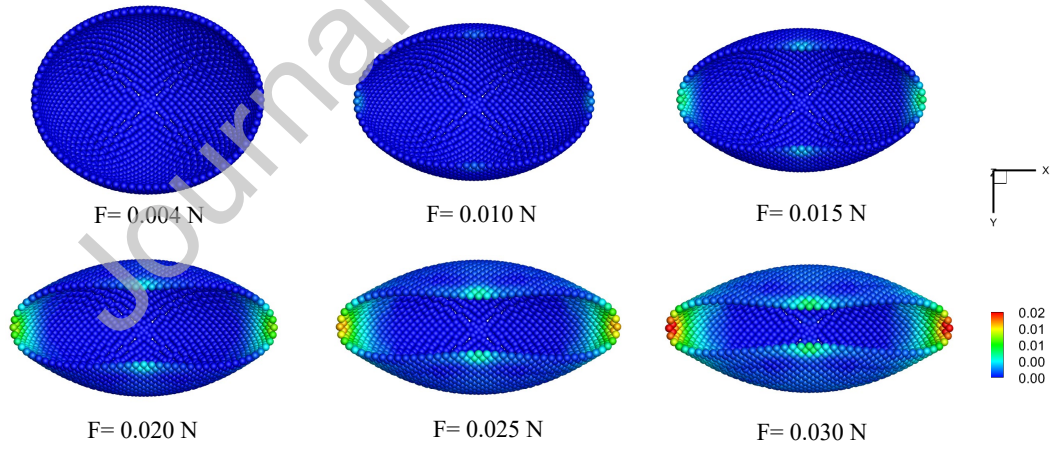


Fig. 29. Pinched hemisphere: simulated deformation profiles under different loads, where the contour indicates averaged equivalent plastic strain in the thickness direction.

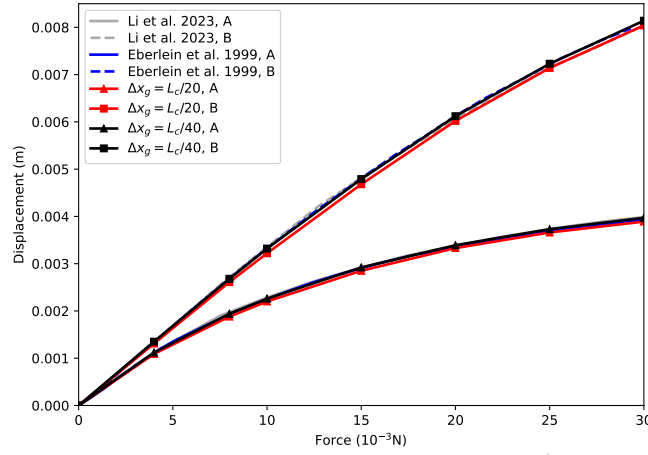


Fig. 30. Pinched hemisphere: displacements at three probe points simulated using the grid resolution $\Delta x_g = L_c/20$ and $\Delta x_g = L_c/40$ under different external loads, compared with the references in [88] and [64].

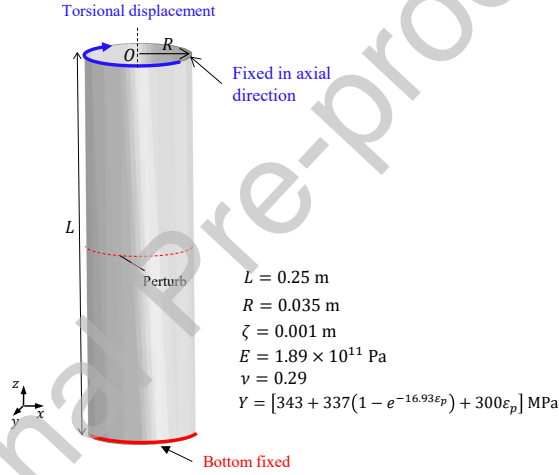


Fig. 31. Schematic of the twisted cylinder.

even for small deformation. It is worth noting that the proposed shell MPM, with Δx_g three times the shell thickness, can provide reliable prediction results, and this property is difficult to achieve using traditional MPM.

6.6. Twisted hollow cylinder

In this section, a twisted hollow cylinder is simulated using the proposed shell MPM. This case was first studied in [29] using the IGA method and was later simulated in [34] using the peridynamics method. In this case, a torsional displacement, fixed in the axial direction, is exerted on the cylinder top with its bottom fixed, as shown in Figure 31. Due to a perturbation in the cylinder body, the cylinder will enter a buckling deformation mode as the torsional displacement increases. The parameters of the cylinder materials are shown in the schematic. The quasi-static simulation mode is used in this case.

The grid resolution is chosen as $\Delta x_g = 2\pi R/86 = 2.55$ mm, similar to that used in [34]. Under resolution $\Delta x_g = 2.55$ mm, the numbers of shell particles and stress points are 8,514 and 8,428, respectively. The deformation profiles of the cylinder under different torsional angles are shown in Figure 32, where the torsional buckling phenomenon is clearly captured using the proposed MPM shell method. To quantitatively estimate the performance of the simulation, the deformation defined using the displacement norm average

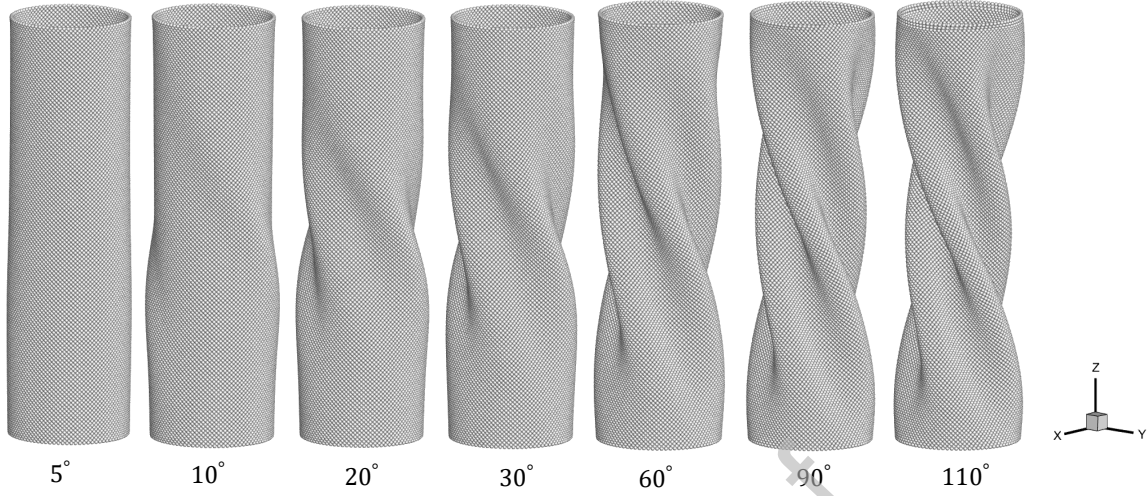


Fig. 32. Twisted hollow cylinder: deformed profiles of the cylinder twisted at different torsional angles.

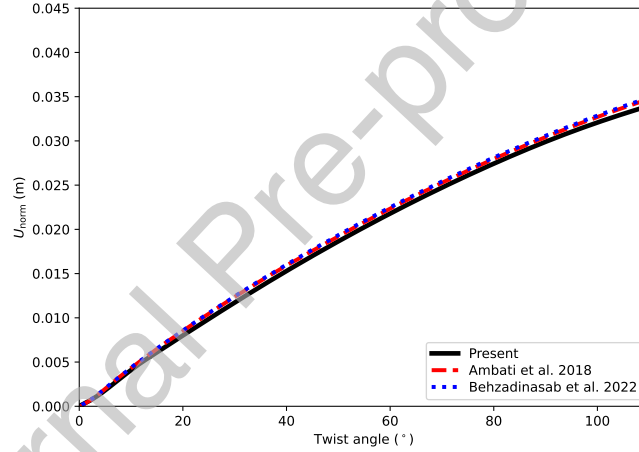


Fig. 33. Twisted hollow cylinder: the norm of displacement, as defined in Equation (59), versus twist angle, and comparison to the references in [29] and [34].

of all particles, U_{norm} , is recorded at different torsional angles, with U_{norm} expressed as

$$U_{\text{norm}} = \sqrt{\sum_{N_p} \|\varphi_p - \varphi_{0p}\|^2 / N_p}, \quad (59)$$

where N_p is the shell particle number. Figure 33 displays the comparison of present results with those in [29] and [34]. It is shown that the curve predicted using the present MPM shell method agrees quite well with that simulated using isogeometric FEM. We note that the selected grid resolution Δx_g is 2.5 times the shell thickness.

7. Concluding Remarks

This study developed a novel MPM for RM shells that avoids meshing the shell geometry and enables a grid cell size much coarser than the shell thickness. The weak form of the MPM shell was derived on the 2D curved shell surface using Hamilton's principle, and a transformation from the 2D curved surface to 3D

Cartesian space enabled solving the weak form in the context of MPM. The proposed MPM shell method describes the dynamics using a set of equations compatible with either the updated or total Lagrangian framework. The P2G and G2P steps are implemented based on the GIMP approach, and Newmark- β time integration is adopted, which allows a much larger time step than the widely used USL scheme. Stress points are also adopted, to improve numerical stability, and the gradients of shell field values are calculated on the curvilinear surface using the kernel function method. The incremental stress update is implemented on stress points, considering large deformations and rotations. In addition, an anti-hourglass term further improves numerical stability of the pseudo-normal of the shell. Additionally, self- or multi-body contact of the shell can be simulated in the present method.

The proposed MPM shell method is validated using a set of benchmark cases. Across all test cases, we observed smooth field values without numerical instability, and predictions that agree quite well with reference solutions from the literature. In various examples, we demonstrate that our MPM shell method can yield a reliable prediction using a grid cell size much larger than the shell thickness, which eliminates the requirement of resolving the thin shells using the computational grid as in traditional MPM or other techniques. Furthermore, the present method displays a good performance in handling shell contact behaviors. Overall, we find the proposed MPM shell method to be robust, accurate, and highly efficient compared to traditional MPM. Given the first-order convergence rate observed in the proposed method, a higher-order MPM could be pursued in the future, especially given recent works such as [89]. Incorporating fracture mechanics [90] could also be considered as future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that support the findings of this study are available on request from the corresponding author.

References

- [1] S. Timoshenko, S. Woinowsky-Krieger, Theory of plates and shells, McGraw-hill, 1959.
- [2] Y. Xu, Z. Wei, Q. Pei, X. Li, Y. Li, [Fem-bem analysis with subdivision surface for acoustic analysis of thin-shell structures under sea floor and sea level reflection conditions](#), Ocean Engineering 342 (2025) 122821. doi:<https://doi.org/10.1016/j.oceaneng.2025.122821>. URL <https://www.sciencedirect.com/science/article/pii/S0029801825025041>
- [3] C. G. Dávila, P. P. Camanho, A. Turon, Effective simulation of delamination in aeronautical structures using shells and cohesive elements, Journal of Aircraft 45 (2) (2008) 663–672.
- [4] S. Ahmad, B. M. Irons, O. Zienkiewicz, Analysis of thick and thin shell structures by curved finite elements, International journal for numerical methods in engineering 2 (3) (1970) 419–451.
- [5] J. Tang, M. Lu, Y. Xie, W. Yin, A novel efficient FEM thin shell model for bio-impedance analysis, Biosensors 10 (6) (2020) 69.
- [6] D. Wu, C. Zhang, X. Hu, [An SPH formulation for general plate and shell structures with finite deformation and large rotation](#), Journal of Computational Physics 510 (2024) 113113. doi:<https://doi.org/10.1016/j.jcp.2024.113113>. URL <https://www.sciencedirect.com/science/article/pii/S0021999124003620>
- [7] J. Boustani, M. F. Barad, C. C. Kiris, C. Brehm, [An immersed boundary fluid–structure interaction method for thin, highly compliant shell structures](#), Journal of Computational Physics 438 (2021) 110369. doi:<https://doi.org/10.1016/j.jcp.2021.110369>. URL <https://www.sciencedirect.com/science/article/pii/S0021999121002643>
- [8] T. Gao, H. Qiu, L. Fu, A semi-meshless Lagrangian finite-volume framework based on Voronoi diagram for general elastoplastic Reissner-Mindlin shell, Journal of Computational Physics 501 (2024) 112802.
- [9] A. Goza, T. Colonius, [A strongly-coupled immersed-boundary formulation for thin elastic structures](#), Journal of Computational Physics 336 (2017) 401–411. doi:<https://doi.org/10.1016/j.jcp.2017.02.027>. URL <https://www.sciencedirect.com/science/article/pii/S0021999117301201>
- [10] W. K. Liu, Y. Guo, S. Tang, T. Belytschko, A multiple-quadrature eight-node hexahedral finite element for large deformation elastoplastic analysis, Computer Methods in Applied Mechanics and Engineering 154 (1-2) (1998) 69–132.
- [11] M. L. Sanchez, P. M. Pimenta, A. Ibrahimbegovic, A simple geometrically exact finite element for thin shells—part 1: statics, Computational Mechanics 72 (6) (2023) 1119–1139.
- [12] J. N. Reddy, Theory and analysis of elastic plates and shells, CRC press, 2006.
- [13] E. Ventsel, T. Krauthammer, E. Carrera, Thin plates and shells: Theory, analysis, and applications, Applied Mechanics Reviews 55 (4) (2002) B72.
- [14] J. N. Reddy, A simple higher-order theory for laminated composite plates, Journal of applied mechanics 51 (4) (1984) 745–752.
- [15] ABAQUS Documentation, [ABAQUS 6.5 documentation: Chapter 3, section 6](#), accessed: 2025-12-20 (2009). URL <https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.5/books/stm/default.htm?startat=ch03s06ath84.html>
- [16] J. Schöberl, C++ 11 implementation of finite elements in NGSolve, Institute for analysis and scientific computing, Vienna University of Technology 30.
- [17] NGSolve Team, [8.1. Reissner-Mindlin and Kirchhoff-Love plates](#), accessed: 2025-12-20 (2024). URL https://doc.ngsolve.org/ngs24/SaS/plates_derivation.html
- [18] COMSOL, [The shell and plate interfaces](#), accessed: 2025-12-20 (2022). URL https://doc.comsol.com/6.2/doc/com.comsol.help.sme/sme_ug_shell_plate.08.15.html
- [19] T. J. Hughes, W. K. Liu, Nonlinear finite element analysis of shells: Part I. Three-dimensional shells, Computer Methods in Applied Mechanics and Engineering 26 (3) (1981) 331–362.
- [20] T. J. Hughes, W. K. Liu, Nonlinear finite element analysis of shells-part II. two-dimensional shells, Computer Methods in Applied Mechanics and Engineering 27 (2) (1981) 167–181.
- [21] W. K. Liu, E. Law, D. Lam, T. Belytschko, Resultant-stress degenerated-shell element, Computer Methods in Applied Mechanics and Engineering 55 (3) (1986) 259–300.
- [22] E. Reissner, On bending of elastic plates, Quarterly of applied mathematics 5 (1) (1947) 55–68.
- [23] E. Reissner, The Effect of Transverse Shear Deformation on the Bending of Elastic Plates, Journal of Applied Mechanics 12 (2) (1945) A69–A77.
- [24] R. Mindlin, Influence of Rotatory Inertia and Shear on Flexural Motions of Isotropic, Elastic Plates, Journal of Applied Mechanics 18 (1) (1951) 31–38.
- [25] T. Belytschko, H. Stolarski, W. K. Liu, N. Carpenter, J. S. Ong, Stress projection for membrane and shear locking in shell finite elements, Computer Methods in Applied Mechanics and Engineering 51 (1-3) (1985) 221–258.
- [26] D. Benson, Y. Bazilevs, M.-C. Hsu, T. Hughes, Isogeometric shell analysis: the Reissner–Mindlin shell, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 276–289.
- [27] W. Dornisch, S. Klinkel, B. Simeon, Isogeometric Reissner–Mindlin shell analysis with exactly calculated director vectors, Computer Methods in Applied Mechanics and Engineering 253 (2013) 491–504.
- [28] J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wüchner, Isogeometric shell analysis with kirchhoff–love elements, Computer methods in applied mechanics and engineering 198 (49-52) (2009) 3902–3914.
- [29] M. Ambati, J. Kiendl, L. De Lorenzis, Isogeometric Kirchhoff–Love shell formulation for elasto-plasticity, Computer Methods in Applied Mechanics and Engineering 340 (2018) 320–339.
- [30] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Monthly Notices of the Royal Astronomical Society 181 (3) (1977) 375–389.
- [31] D. Wu, C. Zhang, X. Hu, An SPH formulation for general plate and shell structures with finite deformation and large rotation, Journal of Computational Physics 510 (2024) 113113.
- [32] F.-R. Ming, A.-M. Zhang, X.-Y. Cao, A robust shell element in meshfree SPH method, Acta Mechanica Sinica 29 (2) (2013) 241–255.
- [33] Q. Zhang, S. Li, A.-M. Zhang, Y. Peng, On nonlocal geometrically exact shell theory and modeling fracture in shell structures, Computer Methods in Applied Mechanics and Engineering 386 (2021) 114074.
- [34] M. Behzadinasab, M. Alaydin, N. Trask, Y. Bazilevs, A general-purpose, inelastic, rotation-free Kirchhoff–Love shell formulation for peridynamics, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114422.
- [35] T. Belytschko, Y. Y. Lu, L. Gu, Element-free Galerkin methods, International Journal for Numerical Methods in Engineering 37 (2) (1994) 229–256.

- [36] W. K. Liu, S. Jun, Y. F. Zhang, Reproducing kernel particle methods, *International Journal for Numerical Methods in Fluids* 20 (8-9) (1995) 1081–1106.
- [37] H. Noguchi, T. Kawashima, T. Miyamura, Element free analyses of shell and spatial structures, *International Journal for Numerical Methods in Engineering* 47 (6) (2000) 1215–1240.
- [38] Y. Peng, A. Zhang, F. Ming, A thick shell model based on reproducing kernel particle method and its application in geometrically nonlinear analysis, *Computational Mechanics* 62 (3) (2018) 309–321.
- [39] D. Sulsky, Z. Chen, H. L. Schreyer, A particle method for history-dependent materials, *Computer Methods in Applied Mechanics and Engineering* 118 (1-2) (1994) 179–196.
- [40] D. Sulsky, S.-J. Zhou, H. L. Schreyer, Application of a particle-in-cell method to solid mechanics, *Computer physics communications* 87 (1-2) (1995) 236–252.
- [41] S. Ma, X. Zhang, X. Qiu, Comparison study of MPM and SPH in modeling hypervelocity impact problems, *International Journal of Impact Engineering* 36 (2) (2009) 272–282.
- [42] P. Huang, X. Zhang, S. Ma, X. Huang, Contact algorithms for the material point method in impact and penetration simulation, *International Journal for Numerical Methods in Engineering* 85 (4) (2011) 498–517.
- [43] Y. Liang, B. Chandra, K. Soga, Shear band evolution and post-failure simulation by the extended material point method (XMPM) with localization detection and frictional self-contact, *Computer Methods in Applied Mechanics and Engineering* 390 (2022) 114530.
- [44] D. A. B. Hyde, S. W. Gagniere, A. Marquez-Razon, J. Teran, An implicit updated Lagrangian formulation for liquids with large surface energy, *ACM Transactions on Graphics (TOG)* 39 (4). doi:10.1145/3414685.3417845.
- [45] J. Chen, V. Kala, A. Marquez-Razon, E. Gueidon, D. A. B. Hyde, J. Teran, A momentum-conserving implicit material point method for surface tension with contact angles and spatial gradients, *ACM Transactions on Graphics (TOG)* 40 (4) (2021) 1–16. doi:10.1145/3450626.3459874.
- [46] A. de Vaucorbeil, V. P. Nguyen, S. Sinaie, J. Y. Wu, Material point method after 25 years: Theory, implementation, and applications, *Advances in Applied Mechanics* 53 (2020) 185–398.
- [47] S. G. Bardenhagen, E. M. Kober, et al., The generalized interpolation material point method, *Computer Modeling in Engineering and Sciences* 5 (6) (2004) 477–496.
- [48] A. Sadeghirad, R. M. Brannon, J. Burghardt, A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations, *International Journal for numerical methods in Engineering* 86 (12) (2011) 1435–1456.
- [49] M. Steffen, P. C. Wallstedt, J. Guilkey, R. Kirby, M. Berzins, Examination and analysis of implementation choices within the material point method (MPM), *Computer Modeling in Engineering & Sciences* 31 (2) (2008) 107.
- [50] Z.-P. Chen, X. Zhang, X.-M. Qiu, Y. Liu, A frictional contact algorithm for implicit material point method, *Computer Methods in Applied Mechanics and Engineering* 321 (2017) 124–144.
- [51] R. Ni, J. Li, X. Zhang, X. Zhou, X. Cui, An immersed boundary-material point method for shock-structure interaction and dynamic fracture, *Journal of Computational Physics* 470 (2022) 111558.
- [52] V. P. Nguyen, C. T. Nguyen, T. Rabczuk, S. Natarajan, On a family of convected particle domain interpolations in the material point method, *Finite Elements in Analysis and Design* 126 (2017) 50–64.
- [53] A. R. York, D. Sulsky, H. L. Schreyer, The material point method for simulation of thin membranes, *International Journal for Numerical Methods in Engineering* 44 (10) (1999) 1429–1456.
- [54] A. Gilmanov, S. Acharya, A hybrid immersed boundary and material point method for simulating 3D fluid–structure interaction problems, *International journal for numerical methods in fluids* 56 (12) (2008) 2151–2177.
- [55] B. Banerjee, Shell formulation for the material point method.
- [56] Y. Lian, X. Zhang, X. Zhou, Z. Ma, A FEMP method and its application in modeling dynamic response of reinforced concrete subjected to impact loading, *Computer Methods in Applied Mechanics and Engineering* 200 (17-20) (2011) 1659–1670.
- [57] J. Wen, J. Barbič, Kirchhoff-love shells with arbitrary hyperelastic materials, *ACM Transactions on Graphics (TOG)* 42 (6) (2023) 1–15.
- [58] Y. Chen, T. Xie, C. Yuksel, D. Kaufman, Y. Yang, C. Jiang, M. Li, Multi-layer thick shells, in: *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–9.
- [59] M. Li, D. M. Kaufman, C. Jiang, Codimensional incremental potential contact, *ACM Transactions on Graphics* 40 (4) (2021) 1–24.
- [60] M. Tang, T. Wang, Z. Liu, R. Tong, D. Manocha, I-cloth: Incremental collision handling for gpu-based interactive cloth simulation, *ACM Transactions on Graphics (TOG)* 37 (6) (2018) 1–10.
- [61] Q. Guo, X. Han, C. Fu, T. Gast, R. Tamstorf, J. Teran, A material point method for thin shells with frictional contact, *ACM Transactions on Graphics (TOG)* 37 (4) (2018) 1–15.
- [62] X. Han, T. F. Gast, Q. Guo, S. Wang, C. Jiang, J. Teran, A hybrid material point method for frictional contact with diverse materials, *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2 (2) (2019) 1–24.
- [63] J. Kang, M. A. Homel, E. B. Herbold, Beam elements with frictional contact in the material point method, *International Journal for Numerical Methods in Engineering* 123 (4) (2022) 1013–1035.
- [64] J. Li, R. Ni, Z. Zeng, X. Zhang, An efficient solid shell material point method for large deformation of thin structures, *International Journal for Numerical Methods in Engineering* 125 (1) (2024) e7359.
- [65] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches, *Computer Methods in Applied Mechanics and Engineering* 199 (37-40) (2010) 2403–2416.
- [66] T. Belytschko, J. I. Lin, T. Chen-Shyh, Explicit algorithms for the nonlinear dynamics of shells, *Computer Methods in Applied Mechanics and Engineering* 42 (2) (1984) 225–251.
- [67] J. C. Simo, D. D. Fox, On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parametrization, *Computer Methods in Applied Mechanics and Engineering* 72 (3) (1989) 267–304.
- [68] Y.-X. Peng, A.-M. Zhang, F.-R. Ming, Numerical simulation of structural damage subjected to the near-field underwater explosion based on SPH and RKPM, *Ocean Engineering* 222 (2021) 108576.
- [69] J. Wang, Y. Bazilevs, A general-purpose meshfree Kirchhoff–Love shell formulation, *Engineering with Computers* 41 (3) (2025) 1379–1410.
- [70] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, A. Selle, The material point method for simulating continuum materials, in: *ACM SIGGRAPH 2016 Courses*, 2016, pp. 1–52.
- [71] T. J. Hughes, W. Liu, Implicit-explicit finite elements in transient analysis: Implementation and numerical examples, *Journal of Applied Mechanics, Transactions ASME* 45 (2) (1978) 375–378.
- [72] F. H. Harlow, The particle-in-cell computing method for fluid dynamics, *Methods Computational Physics* 3 (1964) 319–343.

- [73] J. U. Brackbill, H. M. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *Journal of Computational physics* 65 (2) (1986) 314–343.
- [74] J. U. Brackbill, D. B. Kothe, H. M. Ruppel, FLIP: a low-dissipation, particle-in-cell method for fluid flow, *Computer Physics Communications* 48 (1) (1988) 25–38.
- [75] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, A. Selle, A material point method for snow simulation, *ACM Transactions on Graphics (TOG)* 32 (4) (2013) 1–10.
- [76] C. Jiang, C. Schroeder, A. Selle, J. Teran, A. Stomakhin, The affine particle-in-cell method, *ACM Transactions on Graphics (TOG)* 34 (4) (2015) 1–10.
- [77] C. Dyka, P. Randles, R. Ingel, Stress points for tension instability in SPH, *International Journal for Numerical Methods in Engineering* 40 (13) (1997) 2325–2341.
- [78] Q. Zhang, S. Li, A.-M. Zhang, Y. Peng, J. Yan, A peridynamic Reissner-Mindlin shell theory, *International Journal for Numerical Methods in Engineering* 122 (1) (2021) 122–147.
- [79] T. Belytschko, Y. Guo, W. Kam Liu, S. Ping Xiao, A unified stability analysis of meshless particle methods, *International Journal for Numerical Methods in Engineering* 48 (9) (2000) 1359–1400.
- [80] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in Computational Mathematics* 4 (1) (1995) 389–396.
- [81] P. Randles, L. D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, *Computer Methods in Applied Mechanics and Engineering* 139 (1-4) (1996) 375–408.
- [82] J. Reddy, *Theory and Analysis of Elastic Plates and Shells*, CRC Press, 2006.
- [83] D. J. Benson, J. O. Hallquist, A single surface contact algorithm for the post-buckling analysis of shell structures, *Computer methods in applied mechanics and engineering* 78 (2) (1990) 141–163.
- [84] K. Sze, X. Liu, S. Lo, Popular benchmark problems for geometric nonlinear analysis of shells, *Finite Elements in Analysis and Design* 40 (11) (2004) 1551–1569.
- [85] Q. H. Bui, V. P. Nguyen, A. de Vaucorbeil, A New Contact Algorithm for the Total-Lagrangian Material Point Method, *International Journal for Numerical Methods in Engineering* 126 (17) (2025) e70105.
- [86] X.-W. Zhou, Y.-F. Jin, K.-Y. He, Z.-Y. Yin, An improved explicit MPM formulation and its coupling scheme with FEM, *Computer Methods in Applied Mechanics and Engineering* 436 (2025) 117734.
- [87] A. Norouzzadeh, R. Ansari, A Comprehensive Comparative Study on the Nonlinear Finite Element and Isogeometric Analyses of Shell-Type Structures, *International Journal for Numerical Methods in Engineering* 126 (22) (2025) e70180.
- [88] R. Eberlein, P. Wriggers, Finite element concepts for finite elastoplastic strains and isotropic stress response in shells: theoretical and computational analysis, *Computer Methods in Applied Mechanics and Engineering* 171 (3-4) (1999) 243–279.
- [89] S. Bai, M. F. Liu, C. Schroeder, [Second order accurate discretization of the material point method on periodic domains](https://doi.org/10.1016/j.jcp.2025.114589), *Journal of Computational Physics* 548 (2026) 114589. doi:<https://doi.org/10.1016/j.jcp.2025.114589>. URL <https://www.sciencedirect.com/science/article/pii/S002199912500871X>
- [90] J. Wolper, Y. Fang, M. Li, J. Lu, M. Gao, C. Jiang, CD-MPM: continuum damage material point methods for dynamic fracture animation, *ACM Transactions on Graphics (TOG)* 38 (4) (2019) 1–15.

Appendix A. Rotational plate

A rotational plate is simulated using the present method. The schematic and setup of this case are illustrated in Figure A.1, where the square plate rotates with respect to the x axis. The snapshots at different times of the rotational plate simulated with resolution $\Delta x_g = 1/20$ m are shown in Figure A.2. One can clearly observe the von Mises stress and velocity fields. The coordinate y over time at one of the corners is plotted in Figure A.3(a). In addition, the momentum and angular momentum of the system are shown in Figure A.3(b) and (c). In this case, these two momentum variables are expected to maintain their initial values, since there is no external momentum or angular momentum input to the system. The present simulation results show that the conservation property of momentum or angular momentum is well achieved.

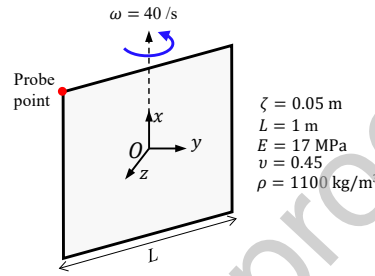


Fig. A.1. Schematic of the rotational plate.

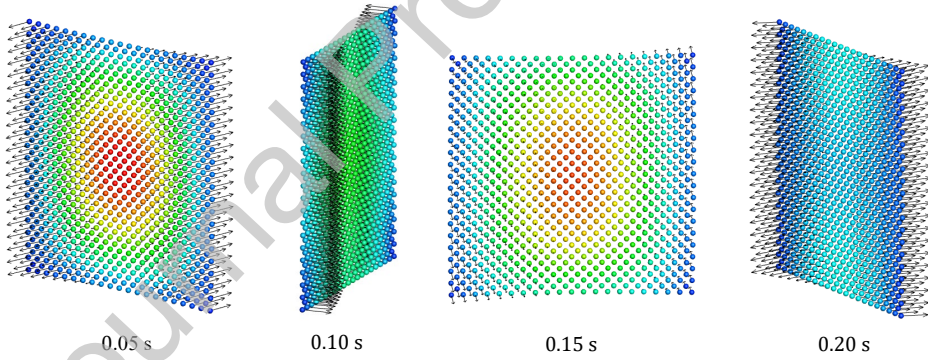


Fig. A.2. Rotational plate: the rotating plate at different times, where contour and vectors indicate von Mises stress and velocity, respectively.

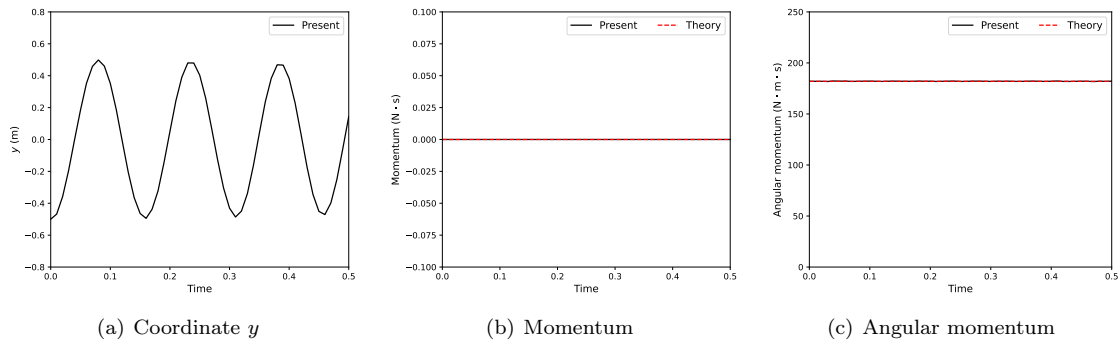


Fig. A.3. Rotational plate: (a) Coordinate y at the probe point, (b) momentum and (c) angular momentum of the system.