DOI: 10.1111/cgf.15273

COMPUTER GRAPHICS *forum*
Volume 44 (2025), number 1, e15273

A Hybrid Lagrangian–Eulerian Formulation of Thin-Shell Fracture

L. Fan,[1] F. M. Chitalu[1] and T. Komura[1,2]

[1]Centre for Transformative Garment Production, Pak Shek Kok, Hong Kong
{glovelfan, floyd.m.chitalu}@gmail.com, taku@cs.hku.hk
[2]The University of Hong Kong, Pokfulam, Hong Kong

Abstract

The hybrid Lagrangian/Eulerian formulation of continuum shells is highly effective for producing challenging simulations of thin materials like cloth with bending resistance and frictional contact. However, existing formulations are restricted to materials that do not undergo tearing nor fracture due to the difficulties associated with incorporating strong discontinuities of field quantities like velocity via basis enrichment while maintaining C^1 continuity or H^2 regularity. We propose an extension of this formulation to simulate dynamic tearing and fracturing of thin shells using Kirchhoff–Love continuum theory. Damage, which manifests as cracks or tears, is propagated by tracking the evolution of a time-dependent phase-field in the co-dimensional manifold, where a moving least-squares (MLS) approximation then captures the strong discontinuities of interpolated field quantities near the crack. Our approach is capable of simulating challenging scenarios of this tearing and fracture, all-the-while harnessing the existing benefits of the hybrid Lagrangian/Eulerian formulation to expand the domain of possible effects. The method is also amenable to user-guided control, which serves to influence the propagation of cracks or tears such that they follow prescribed paths during simulation.

Keywords: fracture, thin shell, material point method, physical simulation

CCS Concepts: • Computing methodologies → Physical simulation

1. Introduction

Simulating volumetric thin-shells like cloth, rubber and sheets of metal is at the core of many applications in computer graphics, including complex visual effects in film. Unfortunately, most thin-shell formulations assume a topologically fixed continuum, which makes them challenging to simulate with fracture effects. In particular, the recent hybrid Lagrangian/Eulerian formulation [JGT17, GHF*18] can simulate complex effects including bending resistance and (automatic) contact handling but is confined to a continuum setting without tearing nor fracture effects in elastic materials. To overcome this issue, alternative strategies have been developed for inducing this *discontinuity* of field quantities using a variety of mesh-based methods. For instance, some techniques replace the continuous assumption with a discrete shell model at the cost of simplifying measures of bending (*i.e.* flexural) energy over mesh edges and with separate treatment of contact [GHDS03]. Other methods propose using thin plates [BDW13], or 2D finite elements [PNDJO14] with stress-based explicit remeshing and refinement for handling crack propagation.

In this paper, we resolve continuum shell fractures and tears by modelling the evolution of a time-dependent damage field with continuum damage mechanics (CDM). A related approach was considered previously by Fan *et al.* [FCK22] for the case of brittle fracture in 3D volumetric solids, but their formulation is based on a rigid-body simplification and thus does not easily generalize to soft materials represented as thin shells. Conversely, some attempts have focused on soft volumetric solids to derive dynamic material damage evolution. In particular, Wolper *et al.* [WFL*19] simulate dynamic fracture involving large elastoplastic deformation, while their follow-up work [WCL*20] adopts a geometric approach [WDG*19] to crack-modelling for simulating fracture in anisotropic materials. Our method is tailored toward continuum shells based on the Lagrangian/Eulerian formulation of Guo *et al.* [GHF*18], but goes further and establishes a new model for simulating detailed and consistent crack-paths for a variety of materials characterized by a high ratio of width to thickness.

Our key contribution is a novel formulation of fracture in thin shells that can be used to simulate effects in soft and near-brittle

materials that are represented as continuum shells. The *fracture* is simulated by tracking the evolution of phase fields (PFs) as a function of stress, where complex crack merging and bifurcation behaviours are also handled automatically. An algorithm for ensuring consistent crack-paths is also presented, where newly traced paths over time (extracted from temporal PFs) are an exact extension of the previous time-steps. We then couple this explicit crack geometry with the moving least-squares (MLS) method for interpolating discontinuous field quantities. We also propose a novel plane-stress formulation that satisfies the Kirchhoff–Love assumption using arbitrary 3D hyperelastic/elastoplastic models. The effectiveness of our method is demonstrated on numerous examples depicting a wide range of complex thin-shell fractures.

A summary of our contributions is as follows:

- A novel plane-stress formulation satisfying the Kirchhoff–Love assumption with fracture;
- An algorithm to extract consistent non-manifold crack paths from damage PFs;
- An enrichment technique for interpolation functions, using MLS method to model discontinuous field quantities in a domain represented by NURBS surfaces;

The rest of the paper is organized as follows. After reviewing related works in § 2, we describe our Kirchhoff–Love shell formulation in § 3. Next, we outline our unified approach for simulating shell dynamics and crack propagation under the MPM framework (see § 4). We then describe how we update damage using a novel PF method that enables material healing to consistently extract the crack (see § 5). This is followed by how we stably compute the stress that satisfies the Kirchhoff–Love zero transverse (normal) stress condition on cracked surfaces. We also show how we separate fragments in material space using meshfree MLS NURBS (see § 7.2). We present the experimental results in § 8 and conclude the paper in § 9.

2. Related Work

In this section, we review related work about thin-shell fracture simulation in both computer graphics and engineering.

Thin shell formulation: Simulation of thin shells in high resolution can be done either by using traditional triangle mesh-based methods [BW98, NSO12, NPO13, HVTG08, PNDJO14] or Lagrangian/Eulerian discretization [JGT17, GHF*18, LZX22]. For applying mesh-based methods, several hurdles exist, such as the need to introduce refinement procedures for avoid locking behaviour [NPO13] and the costly collision detection process due to the increased number of degrees of freedom [TWS05, LZ14, HVTG08]. In contrast, Lagrangian/Eulerian discretization of thin shells (*i.e.* [JGT17, GHF*18, LZX22]) can handle high-resolution contact and collision without the extra burden. Collisions and contacts are implicitly handled through a background grid. For example, Guo *et al.* [GHF*18] propose a method to combine the material point method (MPM) with Kirchhoff–Love shell, which can resolve massive collisions (including self-collisions) automatically. The objective of our research is to build on top of such an approach and also enable simulating phenomena such as tearing and cracks.

Fracture simulation and PF method: Physically based fracture simulation dates back several decades [OH99, TF88, NTB*91, OBH02]. Early works initiate cracks via a simple stretching limit which cannot describe all physical cracks. More recent methods apply the principal stress threshold criterion, *i.e.* Rankine condition [KLB14] for brittle material. Once a crack is initiated, the crack direction is explicitly computed and then propagated using maximum tangential stress criterion [ES63], strain energy density criterion [Sih74], displacement-correlation [CMSK20] or the J-integral model [LNTHZ20]. These explicit methods, however, require complex geometry operations for cutting the domain. Furthermore, they have difficulties in handling branching cracks which are quite common in real-world fractures.

The PF method [FM98, BFM00, AMS*14] is an alternative that overcomes the drawbacks of explicit methods. Each discrete element has a phase c which ranges from 0 to 1 indicating the severity of damage. Crack initiation and propagation can thus be treated implicitly by tracking the evolution of a PF. The PF method has been successfully applied to simulate thick shell [KADLK21], brittle thin-shell [KADL*16] and ductile thin-shell [PADLK21].

One limitation of PF method is excessively wide crack paths [FCK22] that are caused by a lack of strong discontinuity. Particles still couple in the damaged region even though they should separate physically. Therefore, cracks usually span several elements/particles in the width direction. These elements are over-stretched because they lose part or all of their ability to resist tensile loading. This artefact is especially prominent in ductile thin-shells due to its co-dimensional nature. The crack width can be reduced by using a higher-resolution mesh or adaptive local refinement [PZM*20] at, of course, increased computational cost. Another way to mitigate this artefact is by element deletion [PADLK21] but this method loses mass as large gaps and is not suitable for graphics applications. The monotonicity of the phase value c will also contribute to the excessively wide crack path [XAM21]. A continuum crack (wide damaged region) cannot disappear after its formation. In other words, the material cannot be healed once damaged as c increases monotonically.

We also adopt a PF approach for simulating tearing/crack propagation but we avoid the excessive expansion of the damage region by introducing explicit crack paths that separate the crack region as done in the work of Fan *et al.* [FCK22]. Moreover, Fan *et al.* [FCK22] compute the crack path as the skeleton of the damaged region: the temporal consistency of the crack path is ignored and thus the crack propagation process cannot be simulated. Thus, we develop a novel scheme that constructs a consistent approximation of the crack surface given a time-varying PF.

Enforcing discontinuity in NURBS: As we use NURBS as our representation for the thin shells, we review methods that can enforce discontinuity in NURBS. Knot insertion is an effective way to introduce strong discontinuity inside NURBS, but is only valid for cases where the shape boundary is represented by a 1D curve [ATL22]. For NURBS surfaces, knot insertion cannot be done locally; the entire patch needs to be split at the same uv parameters. Although T-splines [CLdB17] and LR-splines [PZM*20] are variations of NURBS that allow for local knot insertion, they tend to produce staircase artefacts as the fragments are cut off along the

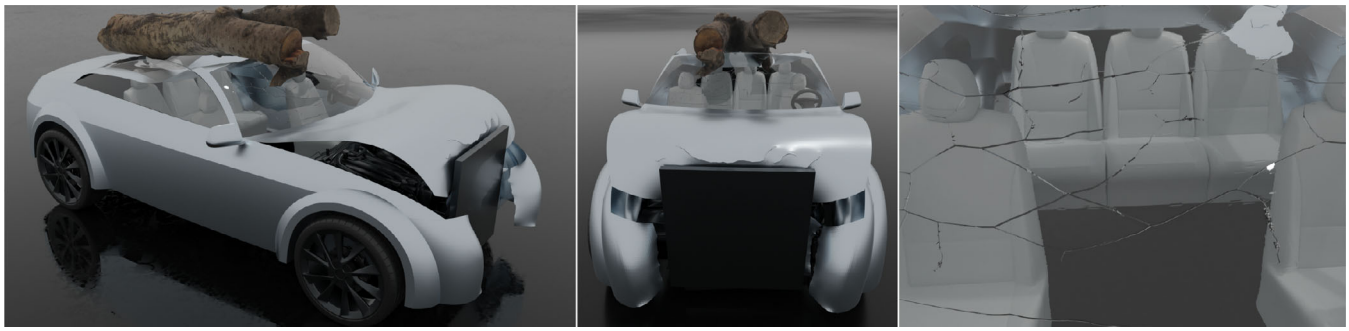


Figure 1: *Our approach can simulate fracturing and tearing of complex thin-shells with a diverse composition of material. Here we simulate damage effects on the body of a sports car that is composed of 12 NURBS patches. A heavy impactor initially hits the bumper, breaking it into three pieces. The bonnet is bent under compression, leaving permanent deformation and fracture. A tree then falls onto the car, perforating the front windscreen to generate cracks that emulate broken laminated glass.*

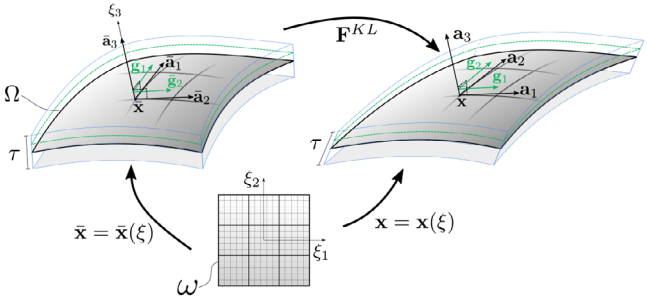


Figure 2: *Mapping between parameter domain ξ, reference/undeformed surface (left) and deformed surface (right) of a Kirchhoff–Love shell.*

knots. The most popular method, extended isogeometric analysis (XIGA) [LNTHZ20], adds an enrichment term to the NURBS basis function which can achieve exact decoupling. However, finding an enrichment function that can manage branching cracks is non-trivial. Our MLS formulation replaces the NURBS basis function in the cracked region with MLS approximation to achieve exact cutting under arbitrary branching cracks.

3. Shell Kinematics

In this section, we describe the Kirchhoff–Love formulation used for simulating thin shells. We begin by establishing the notation that we adopt throughout. Scalars will be denoted with unbolded lowercase *s*, vectors as boldened lowercase $\mathbf{v}$ and matrices as boldened uppercase $\mathbf{M}$. The overbar $\bar{\cdot}$ indicates a variable which is defined with-respect-to the reference/undeformed configuration. We also follow the general convention that Greek letter indices, *e.g.* α are used for 2D parameterization, thus taking values 1 and 2, and Latin letter indices, *e.g.* i are used for 3D parameterization, thus taking values 1, 2 and 3.

Consider a thin shell with a mid-surface Ω and thickness τ as shown in Figure 2. The kinematic motion of this shell is given by the following:

$$\begin{cases} \bar{\mathbf{r}}(\xi) = \bar{\mathbf{x}}(\xi_1, \xi_2) + \xi_3 \bar{\mathbf{a}}_3(\xi_1, \xi_2), \\ \mathbf{r}(\xi) = \mathbf{x}(\xi_1, \xi_2) + \xi_3 \mathbf{a}_3(\xi_1, \xi_2) \end{cases} \tag{1}$$

where $\bar{\mathbf{r}} \in IR^3$ is the *undeformed* position of a material point with $\xi \in IR^3$, and $\bar{\mathbf{x}} \in IR^3$ is its position in the mid-surface, *i.e.* with $\xi_3 \equiv 0$. The corresponding vector $\bar{\mathbf{a}}_3 = \frac{\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2}{|\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2|} \in IR^3$ is the normal to the mid-surface at (ξ_1, ξ_2), where $\bar{\mathbf{a}}_i = \frac{\partial \bar{\mathbf{x}}}{\partial \xi_i}$ is the tangent direction. The respective counterpart $\mathbf{r}$ (with $\mathbf{a}_3$ and $\mathbf{x}$) is the image under stretch and shear loading of the continuum shell.

3.1. Shell motion

Continuum shell motion $\mathbf{r}(\xi)$ is composed of two components. These are the Kirchhoff–Love $\mathbf{r}^{KL}$ component and shearing/compression component $\mathbf{r}^{S}$, respectively, that make up the shell motion as follows:

$$\mathbf{r}(\xi) = \mathbf{r}^{S}(\mathbf{r}^{KL}(\xi)). \tag{2}$$

The Kirchhoff–Love assumption states that fibre-lines which are normal to the shell mid-surface shall remain orthogonal to this mid-surface as the shell deforms.

$$\mathbf{r}^{KL}(\xi) = \mathbf{x}(\xi_1, \xi_2) + \xi_3 \mathbf{a}_3^{KL}(\xi_1, \xi_2), \tag{3}$$

as the Kirchhoff–Love component of kinematic motion where $\mathbf{a}_3^{KL} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|}$ is a unit vector representing the normal at (ξ_1, ξ_2) with $-\tau/2 \leq \xi_3 \leq \tau/2$. The normal stress is neglected under thin-shell assumption, which is also known as *the zero transverse (normal) stress condition*. The shearing/compression component will induce a change of length and direction of fibre lines.

Given the volumetric mapping of the continuum shell from a reference to the deformed configuration $\phi(\bar{\mathbf{x}}) = \mathbf{r}(\bar{\mathbf{r}}^{-1}(\bar{\mathbf{x}})) : \bar{\Omega}^{\tau} \to \Omega^{\tau}$ its Jacobian

$$\mathbf{F} = \frac{\partial \phi}{\partial \bar{\mathbf{x}}} = \sum_{i}^{3} \mathbf{g}_i \otimes \bar{\mathbf{g}}^i, \tag{4}$$

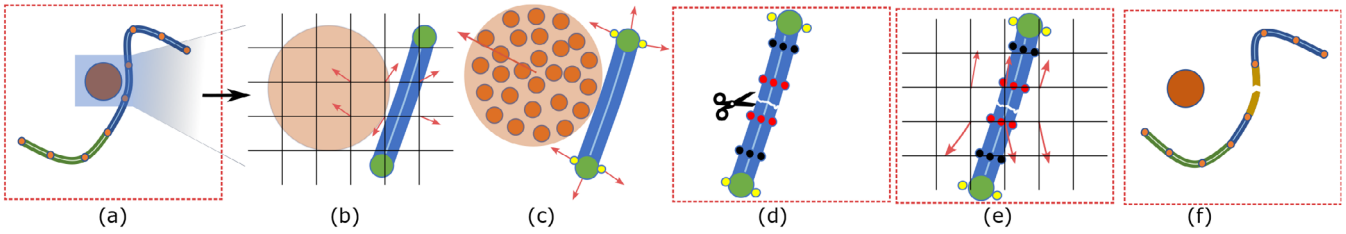


Figure 3: *System overview. (a) The thin shell is represented by NURBS surface. (b) P2G transfer. (c) Particle locations and deformation gradients are updated. (d) Plasticity update, damage update and crack extraction. A cut is made according to the phase field (see § 5.2). (e) Fragments separate due to contact force. (f) The geometry near the cut is represented by the MLS formulation (see § 7.2).*

is the deformation gradient representing a linear approximation of change-of-shape around $\bar{\mathbf{x}}$, where $\otimes$ is the tensor/outer product acting on the covariant $\mathbf{g}_i$ and contravariant basis vector $\bar{\mathbf{g}}^i$, respectively. This definition of $\mathbf{F}$ permits the multiplicative decomposition as (*cf.*Equation 2)

$$\mathbf{F} = \mathbf{F}^S\mathbf{F}^{KL}, \quad \mathbf{F}^S = \sum_i^3 \mathbf{g}_i \otimes \mathbf{g}^{KL,i}, \quad \mathbf{F}^{KL} = \sum_i^3 \mathbf{g}_i^{KL} \otimes \bar{\mathbf{g}}^i \tag{5}$$

where $\mathbf{g}_i^{KL} = \frac{\partial \mathbf{r}^{KL}}{\partial \xi_i}$ and $\mathbf{g}^{KL,i}$ are the covariant and contravariant basis vectors, respectively, that satisfy the aforementioned zero transverse stress condition (see also Clyde *et al.* [CTT17]).

4. Discretization Using MPM

In this section, we summarize discretization of the continuous equations in § 3 for simulation with MPM, providing a high level perspective for our specific contributions described in later sections. An overview of our method is also shown in Figure 3.

Inline with Guo *et al.* [GHF*18], particles are classified into four groups as follows: (I) traditional particles for modelling non-thin-shell objects in the scene; (II) control particles for modelling the thin-shell objects; (III) shearing/compression particles for modelling non-KL deformation of the thin-shells and (IV) quadrature particles for evaluating physical properties of the thin shell. At any given timestep n, the particles of Types I–III will have position $\mathbf{x}_p^n \in IR^3$, velocity $\mathbf{v}_p^n \in IR^3$, initial mass $m_p \in IR$ and volume $V_p^0 \in IR$. Particle types are further characterized by the additional physical properties they carry. Type-I store a deformation gradient $\mathbf{F}_p^n \in IR^{3\times3}$ as in standard MPM, while Type-II particles are the NURBS control points, which define the geometry of the thin shells (see § 7.2). Type-III store an elastic shearing basis vector $\mathbf{a}_{p3}^n \in IR^3$. And Type-IV store the Kirchhoff–Love component of the deformation gradient $\mathbf{F}_q^{KL,n} \in IR^{3\times3}$, its elastic component $\mathbf{F}_q^{KL,E,n} \in IR^{3\times3}$ and plastic component $\mathbf{F}_q^{KL,P,n} \in IR^{3\times3}$. A damage variable $c \in IR$ is also stored at these Type-IV particles, with another variable c^{his} as the largest value of c over the history of crack evolution (see § 5.1 for detail). We use notations $\Gamma^{(I)}$, $\Gamma^{(II)}$, $\Gamma^{(III)}$ and $\Gamma^{(IV)}$ to denote sets of the four types of particles, respectively.

1. P2G transfer: We first transfer particle mass and momentum (from Types I to III) to grid nodes via $m_\mathbf{i}^n = \sum_p m_p w_{\mathbf{i}p}^n$ and $(m\mathbf{v})_\mathbf{i}^n = \sum_p m_p w_{\mathbf{i}p}^n \mathbf{v}_p^n$, where $w_{\mathbf{i}p} = \mathcal{N}(\mathbf{x}_p^n - \mathbf{x}_i)$ is the weight function of distance between an MPM particle $\mathbf{x}_p^n$ and a grid node with index $\mathbf{i}$ and position $\mathbf{x}_\mathbf{i}$.

2. Momentum update: The net internal elastic force at node $\mathbf{i}$

$$\mathbf{f}_\mathbf{i}^{\text{int},n} = \mathbf{f}_\mathbf{i}^{(\text{I}),n} + \mathbf{f}_\mathbf{i}^{(\text{II}),n} + \mathbf{f}_\mathbf{i}^{(\text{III}),n}$$

is then computed from the gradients of the elastic potentials from respective particles. The term $\mathbf{f}_\mathbf{i}^{(\text{I}),n}$ is the canonical MPM force term of Type-I particles, which is evaluated from the elastic potential ψ (we use the Neo-Hookean model).

The generalized force from Kirchhoff–Love quadrature points (via Type-II particles) is given by

$$\mathbf{f}_\mathbf{i}^{(\text{II}),n} = -\sum_{p\in\Gamma^{(\text{II})}} w_{\mathbf{i}p}^n \sum_{q\in\Gamma^{(\text{IV})}} V_q^0 \tilde{\mathbf{P}}(\mathbf{P}, c) : \frac{\partial \mathbf{F}_q^{KL,Etr}}{\partial \mathbf{x}_p^{KL}}. \tag{6}$$

We have $\mathbf{P} = \frac{\partial \psi}{\partial \mathbf{F}^{KL}}(\mathbf{F}_q^{KL,Etr})$ as the *trial* first Piola–Kirchhoff stress (PK1) that is determined from the trial elastic deformation gradient $\mathbf{F}_q^{KL,Etr}$. This trial PK1 stress does not satisfy the zero transverse normal stress condition of the Kirchhoff–Love model. We, therefore, need its in-plane projection, which we obtain by calculating the corresponding trial Cauchy stress $\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F}^{KL})}\mathbf{P}(\mathbf{F}^{KL})^T$ and its in-plane components (see § 6). It is from this trial in-plane PK1 that the weakened PK1 stress $\tilde{\mathbf{P}}$ is evaluated with damage c (see § 5.1). The remaining term $\mathbf{f}_\mathbf{i}^{\text{III},n}$ is the generalized shearing/compression force. We refer readers to Guo *et al.* [GHF*18] for details.

The updated velocity on node $\mathbf{i}$ is

$$\bar{\mathbf{v}}_\mathbf{i}^{n+1} = (m\mathbf{v})_\mathbf{i}^n / m_\mathbf{i}^n + \Delta t(\mathbf{f}_\mathbf{i}^{\text{int},n} + \mathbf{f}_\mathbf{i}^{\text{ext},n})/m_\mathbf{i}^n,$$

where Δt is the timestep size, and $\mathbf{f}^{\text{ext},n}$ is the external loading due to, *e.g.* gravity. This velocity will be updated by $\mathbf{v}_\mathbf{i}^{n+1} = \bar{\mathbf{v}}_\mathbf{i}^{n+1}$ if no crack or contact is present near this node; otherwise, $\mathbf{v}_\mathbf{i}^{n+1}$ needs to be corrected with the contact force $\mathbf{f}_{\mathbf{i},\text{con}}$ by $\mathbf{v}_\mathbf{i}^{n+1} = \bar{\mathbf{v}}_\mathbf{i}^{n+1} + \Delta t \mathbf{f}_{\mathbf{i},\text{con}}/m_\mathbf{i}^n$. The contact force can be computed accurately using the multi-velocity field method, which we summarize in our technical supplement document.

3. G2P transfer: The velocity of particles of Types I–III are updated from the grid by

$$\mathbf{v}_p^{n+1} = \sum_\mathbf{i} w_{\mathbf{i}p}^n \mathbf{v}_\mathbf{i}^{n+1}.$$

4. Advection: Particle positions are then updated using

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}, \quad p \in \Gamma^{(\mathrm{I})} \cup \Gamma^{(\mathrm{II})}, \tag{7}$$

and

$$\mathbf{x}_p^{n+1} = \sum_{s \in \Gamma^{(\mathrm{II})}} \mathcal{W}_s^{\mathrm{NURBS}} \mathbf{x}_s^{n+1}, \quad p \in \Gamma^{(\mathrm{III})}, \tag{8}$$

where $\mathcal{W}_s^{\mathrm{NURBS}}$ is the NURBS interpolation function from a Type-II particle p to a Type-III particle s. For the vicinity of cracks, we instead use MLS to account for discontinuities (see § 7.2).

5. Plasticity update: Return mapping is used to calculate the corrected plastic strain satisfying the plasticity yield condition: We use the return mapping method of Guo *et al.* [GHF*18] for Type-III particles to model shearing/compression resistance and adopt the associative flow rule of Ambati *et al.* [AKDL18] on Type-IV particles to model bending resistance with fracture.

6. Damage update and crack extraction: The damage state c and history c^{his} of each quadrature point are updated (*cf.* § 5.1), where c^{his} is then used to extract crack geometry at a given timestep which we describe in § 5.2.

5. Co-Dimensional Fracture

This section presents a detailed account of our approach for simulating fractures on thin shells. First, the damage in the domain is calculated using the PF method [CC06]. Then, a crack is explicitly computed to separate the fragments, where we extend the method of Fan *et al.* [FCK22] to ensure consistent cracks and allow seamless cutting of thin shells without any undesirable artefacts.

5.1. Calculating stress and damage

Here we describe how we compute the damage c and how we use it to compute the weakened PK1 stress $\tilde{\mathbf{P}}$ for Equation (6). Our damage calculation is based on the local PF method [CC06], but we enable material healing to avoid over-stretching artefacts.

Without damage, the weakened PK1 stress $\tilde{\mathbf{P}} \equiv \mathbf{P}$ that is stored on Type-IV quadrature particles will be equivalent to the trial in-plane PK1 in Equation (6), *i.e.* the (local) material is healthy with $c = 0$ where $0 \leq c \leq 1$. A material particle in fully damaged state is then represented with $c = 1$. To measure in-plane stress, we require the Cauchy stress $\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F}^{KL})}\mathbf{P}(\mathbf{F}^{KL})^T$ which is a descriptor of force density in world space. Upon calculating this descriptor, we remove normal components to leave only the in-plane components as described in § 6, which is necessary to satisfy the Kirchhoff–Love zero transverse normal stress condition.

The effective (in-plane) stress is then determined by $\overline{\sigma} = \sigma_1$ via singular value decomposition $\boldsymbol{\sigma} = \mathbf{U}\Sigma\mathbf{V}^\intercal$ with $\mathrm{diag}(\Sigma) = (\sigma_1, \sigma_2, \sigma_3)$ such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$ [FCK22]. We estimate the damage value $\overline{c}$ using (*cf.* Eq. 7 in Fan *et al.* [FCK22])

$$\overline{c} = \begin{cases} (1+\mathcal{H})\left(1 - \frac{\sigma^f}{\overline{\sigma}}\right), & \sigma^f \leq \overline{\sigma} \leq \left(1+\frac{1}{\mathcal{H}}\right)\sigma_f \\ 1, & \left(1+\frac{1}{\mathcal{H}}\right)\sigma_f \leq \overline{\sigma} \end{cases}, \tag{9}$$

where σ_f is a user parameter specifying the (local) damage threshold, $\mathcal{H} = \overline{\mathcal{H}} l^{\mathrm{ch}}/(1 - \overline{\mathcal{H}} l^{\mathrm{ch}})$ is the brittleness factor, $\overline{\mathcal{H}} = {\sigma_f}^2/2EG_f$ is a material constant, E and G_f are Young's modulus and Mode-I fracture energy, respectively and $l^{\mathrm{ch}} = \sqrt{3h^2}$ is the characteristic length based on MPM grid spacing h.

(a) *Overstretched* **(b)** *Ours*

Figure 4: *Extreme over-stretching (a) and our proposed method (b). Material near the crack is severely stretched when cutting a thin shell by the original phase-field method.*

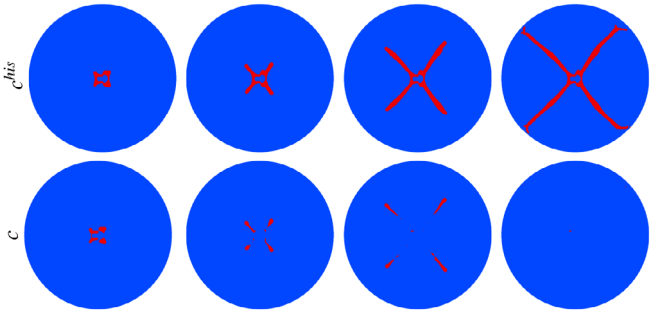


Figure 5: *The evolution of phase c (bottom row) and its largest value in history c^{his} (top row). c^{his}'s area gradually expands, indicating the propagation of cracks, which is used to extract a consistent crack path. c vanishes when the elastic force is unloaded where fragments are separated. This effectively avoids over-stretching artefacts.*

Damage is updated from Equation (9) as $c^{n+1} = \overline{c}$, offering a simple update-rule when compared with conventional fully elastodynamic PF methods that use $c^{n+1} = \max(\overline{c}, c^n)$ to prevent healing of damaged material (see, *e.g.* Refs. [WFL*19, CC06, HH17]). We have found that the conventional approach suffers from over-stretching artefacts in damaged regions as shown in Figure 4. We avoid such artefacts via our explicit crack representation that provides exact (rather than smeared) locations of strong discontinuity. Moreover, we also store and update the largest value of damage over time $c^{\mathrm{his}} = \max(c^{\mathrm{his}}, c^{n+1})$, which is then used to extract crack geometry as described in § 5.2. The weakened (Cauchy) stress is finally evaluated via rankine strain-softening (see Refs. [FCK22, CC06] for detail) to arrive at

$$\tilde{\mathbf{P}}(\mathbf{P}, c) = \frac{\partial \psi}{\partial \mathbf{F}^{KL}}(\mathbf{F}_q^{KL,Etr}, c) = \det(\mathbf{F}^{KL})\boldsymbol{\sigma}'^{-\intercal},$$

as the weakened PK1 stress that we use in Equation (6) where $\boldsymbol{\sigma}'$ is the weakened Cauchy stress tensor that satisfies the zero transverse (normal) stress condition.

An example of the evolution of phase is shown in Figure 5. Our use of c^{his} for extracting the crack region simplifies the problem of propagating smeared crack paths in a fully elastodynamic MPM

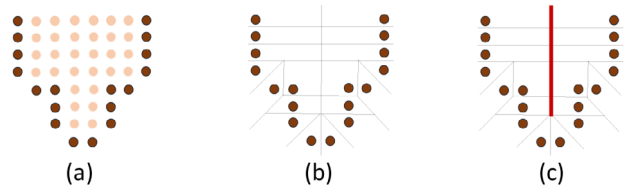


Figure 6: *The crack extraction procedure by Fan* et al. *[FCK22]. (a) Extracting the boundary particles of the damaged region, (b) computing the Voronoi diagram and (c) extracting the crack that corresponds to the medial axis within the damaged area.*

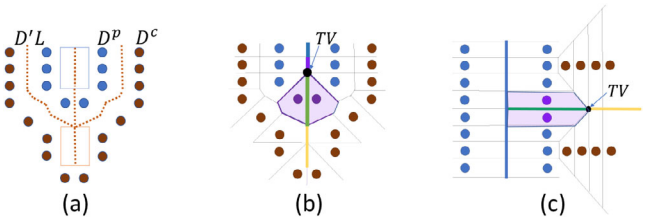


Figure 7: *(a) Extending the crack in a consistent manner. (b) Connecting the previous crack and the extended crack line. (c) A branching/merging crack connected to the previous crack (see text for the details).*

setting. Moreover, our approach, which is based on material healing, is a simplified model of temporal singular crack-tip stress, where computed damage merely serves to prescribe the local region within which to extend the crack at the current point in time. Persistence of strong discontinuities in the material is then embodied by the explicit crack surface itself rather than storing permanent damage on the particles, which has the benefit of alleviating over-stretching artefacts that are common with smeared crack representations due to the permanent weakening of internal forces via stress.

5.2. Consistent crack extraction

In this section, we describe a consistent crack extraction algorithm based on Voronoi tessellations, extending the method in Fan *et al.* [FCK22] by ensuring that extended crack paths of previous time-steps remain unchanged over time. An overview of the original method in 2D is shown in Figure 6, which summarizes crack extraction from the damaged particles with $c^{\text{his}} = 1$ as follows: (a) Damaged particles composing the domain boundary are extracted and then (b) used to construct a Voronoi diagram as the Voronoi 'seeds' before (c) extracting the medial axis produced by seeds that lie on opposing sides of the domain boundary as the crack path. This results in an inconsistent path when the damaged area expands: we thus revise the method such that the crack path gradually extends in a consistent manner (see below, Figure 7). The set notations used in this section can be found in Table 1.

Computing a consistent crack path: We use the superscript notation $(\cdot)^p$ and $(\cdot)^c$ to denote the previous and the current extraction of the crack from damaged particles, respectively. To perform extraction, we first identify fully damaged particle boundaries as $\mathcal{D}$. Due to the evolution of the PF, $\mathcal{D}^p$ (blue points in Figure 7a) is not a precise subset of $\mathcal{D}^c$ (brown points in Figure 7a) as the smeared crack will broaden while simultaneously propagating forward. As a result, the extracted crack lines $\mathcal{DL}^p$ are not a subset of $\mathcal{DL}^c$. To keep the geometry of the previous extraction, we add $\mathcal{LS}$ to $\mathcal{D}^c$ as the Voronoi tessellation seeds, and term it as $\mathcal{D}'$. After computing its medial axis $\mathcal{D}'\mathcal{L}$ (dashed line in Figure 7a), we only keep (1) new lines whose defining seeds are from $\mathcal{D}^c$ (in orange box in Figure 7a); and (2) the subset of historical lines $\mathcal{L}$ (in blue box in Figure 7a). The two lines (1) and (2) produced by the above process are usually disconnected and needs to be connected for the crack to correctly propagate.

Table 1: *A summary of the set notation.*

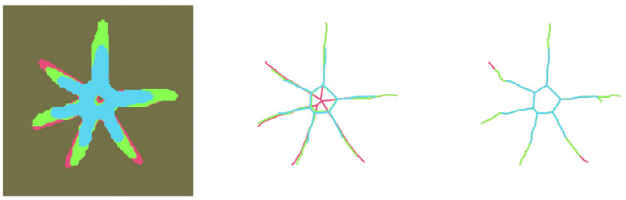
Set	Description
$\mathcal{D}$	Fully damaged particle boundary
$\mathcal{DL}$	Extracted lines using $\mathcal{D}$
$\mathcal{L}$	Historical lines
$\mathcal{LS}$	Voronoi seeds that define historical lines
$\mathcal{NS}$	Voronoi seeds that define new lines
$\mathcal{TV}$	Terminal vertices of crack lines
$\mathcal{TS}$	Voronoi seeds that define terminal lines
$\mathcal{N}$	Neighbours of seeds

Extend crack tips: In most cases, the crack tends to propagate along the crack tip, where new lines are added to the previous extracted lines. We extract the terminal vertices $\mathcal{TV}$ (black dot in Figure 7b) that is connected to the terminal line of $\mathcal{L}$ (purple line in Figure 7b). Among the cells that are connected to $\mathcal{TV}$, we seek those that are connected to the newly added terminal line (yellow line in Figure 7b). This cell's boundary is the bridge connecting previous crack tips to newly extracted lines (green line in Figure 7b). This boundary is added into $\mathcal{L}$ and the seeds that produce it are added into $\mathcal{LS}$ for the next extraction.

Extend a branch to the stem: A branching crack may be produced at the stem of previous crack lines as a result of a crack-merge or bifurcation. After extending crack tips (yellow line in Figure 7c), we identify new terminal vertices $\mathcal{TV}$ (black dot in Figure 7c). Among the cells connected to $\mathcal{TV}$, we check if any of them are connected with a previous line (blue line in Figure 7c). If one is discovered, we traverse along the cell boundary that connects the previous line and $\mathcal{TV}$ and add these edges to $\mathcal{L}$. The corresponding seed pairs that form the edges (purple dots in Figure 7c) are also found and added to $\mathcal{LS}$.

After extending the crack tips and connecting branches, the extracted crack path is guaranteed to be consistent with the previous extraction without gaps. The crack path can then be used to introduce strong discontinuity to the simulation.

Comparison to Fan *et al.* [FCK22]: In Figure 8, we show an example where we compare the proposed method with Fan *et al.* [FCK22]. Simply computing the medial axis using the boundary of the damaged region in each stage (see Figure 8a) will result in inconsistent crack paths (see Figure 8b). Our method can produce a crack that propagates overtime in a consistent manner as shown in Figure 8c.



(a) Phase field. (b) [FCK22]. (c) Ours.

Figure 8: *Comparison of our crack extraction method with Fan* et al. *[FCK22]: (a) A phase-field evolution in three stages; cyan, green and pink. (b) Extracted inconsistent crack paths using the method of Fan* et al. *[FCK22], where the medial axis is computed independently with the damages at each stage. (c) Extracted consistent crack path using our method. Note that part of the cracks in later stages extends those in the earlier stages.*

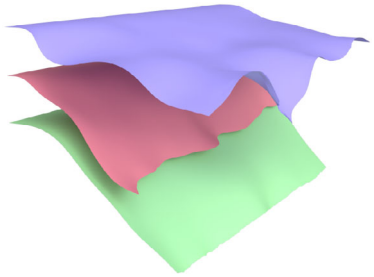


Figure 9: *Deformation of the shell after being torn: we compare our in-plane stress model with Jiang* et al. *[JGT17] and Guo* et al. *[GHF*18]. Jiang* et al. *[JGT17] (red) suffer from volume loss while Guo* et al. *[GHF*18] (purple) experience material separation near the cracked region (the sharp boundary). Our in-plane model (green) is robust and stable even in regions near the crack. The original shell's area is* $1.0\,\mathrm{m}^2$. *After cracking, the shell's surface shrinks to* $0.668\,\mathrm{m}^2$ *in Jiang* et al. *[JGT17] while our method resolves to* $0.994\,\mathrm{m}^2$.

6. In-Plane Stress

In this section, we describe our approach to calculating the stress satisfying the 'zero transverse stress condition'. Satisfying this condition $\boldsymbol{\sigma}\mathbf{g}_3^{KL} = \vec{0}$ means that terms of the stress related to $\mathbf{g}_3^{KL}$ are discarded to keep only those corresponding to the mid-surface tangent plane. Jiang *et al.* [JGT17] satisfy this condition using a 2D deformation gradient to then compute a projected 3D stress, while Guo *et al.* [GHF*18] use the components of the deformation gradient related to the lamina strain to define a plane strain. These methods generally work well but suffer from severe volume-loss or undesirable separation of particles when a crack is introduced as shown in Figure 9. We propose a novel plane-stress formulation satisfying the zero transverse stress condition without artefacts upon fracture.

The trial Cauchy stress from § 4 can be written with covariant basis vectors defined in the fibre and lamina direction as follows:

$$\boldsymbol{\sigma} = \sum_{i=1}^{3}\sum_{j=1}^{3} \sigma^{ij}\,\mathbf{g}_i^{KL} \otimes \mathbf{g}_j^{KL}, \tag{10}$$

using the deformed basis vectors $\mathbf{g}_i^{KL}$ to identify the respective lamina and fibre components of this stress. To solve for the unknown contravariant components σ^{ij}, we construct the following system:

$$\mathbf{G}^{KL}\boldsymbol{\varsigma} = \mathrm{vec}(\boldsymbol{\sigma}). \tag{11}$$

where $\mathbf{G}^{KL} \in IR^{9\times 9}$ is the covariant basis matrix. This matrix is defined such that each row is given by the tensor-product $\mathrm{vec}(\mathbf{g}_i^{KL} \otimes \mathbf{g}_j^{KL})^T \in IR^{1\times 9}$ of two basis vectors in vectorized form with $1 \leq i, j \leq 3$. The term $\boldsymbol{\varsigma} \in IR^{9\times 1}$ is the vector of unknown contravariant components σ^{ij} to give, *e.g.*

$$\mathrm{vec}\left(\mathbf{g}_1^{KL} \otimes \mathbf{g}_1^{KL}\right)^T \cdot \sigma^{11} = \sigma_{11} \tag{12}$$

as the equation of the first row. The term σ_{ij} is simply an entry of the 3×3 (trial) Cauchy stress matrix/tensor that is at the ith row and jth column. Thus, solving Equation (11) for $\boldsymbol{\varsigma}$ gives all components σ^{ij} from which we compute the Cauchy stress satisfying the zero transverse stress condition by

$$\boldsymbol{\sigma}_\perp = \sum_{\alpha=1}^{2}\sum_{\beta=1}^{2} \sigma^{\alpha\beta}\,\mathbf{g}_\alpha^{KL} \otimes \mathbf{g}_\beta^{KL} \in IR^{3\times 3}. \tag{13}$$

Equation (13) is what we use as the Kirchhoff–Love bending stress. In contrast with Jiang *et al.* [JGT17], our approach operates directly on the 3D stress tensor, which is a simpler and more robust method: Jiang *et al.* [JGT17] first construct a 2D (planer) deformation gradient $\mathbf{F} \in IR^{2\times 2}$ of the mid-surface followed by a calculation the corresponding 2D stress that is then extrapolated to 3D. The use of a 2D $\mathbf{F}$ has the potential for underestimating the volume ratio $J = \det(\mathbf{F})$ of the true deformation gradient $\mathbf{F} \in IR^{3\times 3}$ because the elastic energy only penalizes area rather than the true volume change of the shell. Furthermore, we found the negative weights introduced by MLS approximation near the crack region aggravate the volume loss artefact.

7. NURBS-Based Lagrangian Representation

In this section, we describe our NURBS-based shell representation, and how we introduce crack as discontinuity in the continuum shell geometry using MLS.

7.1. NURBS

We represent the shell kinematics as a second-order NURBS surface satisfying C^1 continuity on the shell surface, which is required by numerical quadrature. The NURBS surface representation is widely used in the industry for object design; using this representation allows us to directly apply simulation to models designed for prototyping. Also, its polynomial nature allows us to not only design smooth, continuous surfaces with less computation, but also to well approximate the geometry of the cut/cracked region by a MLS representation, which is also a polynomial representation.

Complex geometries usually contain multiple patches due to the tensor-product nature of NURBS. To simulate such geometries, we adopt the virtual uncommon knot insertion method [CGA*17] to seamlessly stitch different NURBS patches with little extra effort, which we summarize in our technical supplement document.

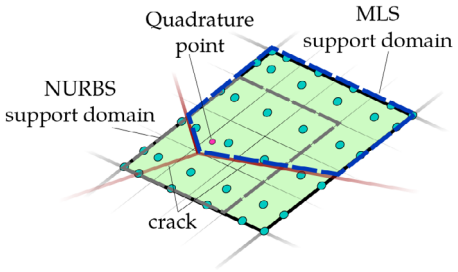


Figure 10: *Approximating the field quantities of a quadrature point near the crack: The red lines are extracted crack paths, and the turquoise blue points depict the control points (Type-II particles). A single quadrature point is shown here in pink, which is affected by the crack. Without the crack, this point would use a rectangular support radius (grey). The purple dashed line represents the (typically larger) custom support radius of the quadrature point that is used with the crack for MLS interpolation.*

7.2. Resolving discontinuities in NURBS

In this subsection, we show our MLS interpolation of field quantities, which we use when a cut is made within the thin shell. The method will provide a scheme to interpolate the quantities of the Type-II and Type-IV particles near the crack(s), which will be done in parameter space ω.

The NURBS shape functions are designed to maintain smooth continuity of the interpolated field quantities across patches, which can limit our ability to accurately represent discontinuities due to cracks. We can overcome this limitation by utilizing the MLS approximation of the field quantities that are stored at particles located near the crack, as demonstrated in Figure 10. A Type-II and Type-IV particles are together considered separated by the crack if their connecting line segment intersects this crack in parameter space. Information is then transferred between a Type-II and Type-IV only if they reside on the same side of the crack. Particles far from the crack shall use standard NURBS shape functions. Pauly *et al.* [PKA*05] propose to include a control point near the crack tip even if it is on the other side to avoid weight discontinuity in the domain. However, we found it to have negligible influence on the PF evolution but at the cost of more complex geometry operations and computation time. In practice, we choose to use the simple binary criterion.

Our MLS approximation will estimate field variables (like deformed covariant basis vectors $\mathbf{g}_i^{KL}$) at a quadrature particle $q(\xi) \in \Gamma^{(\mathrm{IV})}$ from the control points within its support domain by

$$u^h(\xi) = \sum_{i=1}^{|C^{(q)}|} \mathcal{W}_i^{\mathrm{MLS}}(\xi) u_i, \tag{14}$$

where u^h is the approximation of the unknown field quantity at $\xi \in IR^2$ using known values u_i at the control point $C_i^{(q)}$ that is in the support domain centred at $q(\xi)$. We also have $\mathcal{W}_i^{\mathrm{MLS}}$ as the shape/weight function associated with the i'th control point, which is given by

$$\mathcal{W}_i^{\mathrm{MLS}}(\xi) = \mathbf{p}^\intercal(\xi)(\mathbf{A}^{-1}(\xi)\mathbf{B}(\xi))_i, \tag{15}$$

where $\mathbf{p}(\xi) = [1, \xi_1, \xi_2, \xi_1^2, \xi_1\xi_2, \xi_2^2]^\intercal \in IR^{6\times 1}$ is the polynomial basis vector; and $\mathbf{A} \in IR^{6\times 6}$ is given by

$$\mathbf{A}(\xi) = \sum_{k=1}^{|C^{(q)}|} W(\xi - \xi^{(k)})\mathbf{p}(\xi^{(k)})\mathbf{p}^\intercal(\xi^{(k)}), \tag{16}$$

where $\xi^{(k)} \in IR^2$ are the parameter coordinates of the kth control point. The matrix $\mathbf{B} \in IR^{m\times|C^{(q)}|}$ is defined as follows:

$$\mathbf{B} = \begin{bmatrix} W(\xi - \xi^{(1)})\mathbf{p}(\xi^{(1)}) & \dots & W(\xi - \xi^{(|C(q)|)})\mathbf{p}(\xi^{(|C(q)|)}) \end{bmatrix}. \tag{17}$$

We also have $W(\xi - \xi^{(k)}) = w(\xi_1 - \xi_1^{(k)})w(\xi_2 - \xi_2^{(k)})$ as the MLS weight function. As we adopt a rectangular support domain, this weighting is computed as a multiplication along each direction of the parameter space with

$$w(r) = \begin{cases} 2/3 - 4r^2 + 4r^3 & r \le 0.5 \\ 4/3 - 4r + 4r^3 - 4/3r^3 & 0.5 \le r \le 1.0 \\ 0 & r > 1.0 \end{cases}, \tag{18}$$

which is a piece-wise cubic polynomial where $r = \|\xi - \xi^{(k)}\|_2/d^{(k)}$, and $d^{(k)}$ is the support domain size of the kth control point. First- and second-order derivatives of the MLS shape function in Equation (14) are also required to compute the deformation gradient of quadrature points, which can be calculated using the chain rule.

We determine $d^{(k)}$ dynamically to avoid insufficient control-points near the crack to avoid MLS rank deficiency: Starting from $d^{(k)} = 1.5l$, we count the number of neighbouring control-points (on the same side of the explicit crack described in § 5.2) to then expand the list farther away from the crack if the current control-point has less than nine neighbours (Nine neighbours because we are using a quadratic polynomial basis vector. Otherwise 16 for cubic [LNTHZ20].). Throughout our experiments, we found this to require a support domains size of no more than approximately $d^{(k)} = 4.5l$, which may also be set as a fixed constant.

8. Results

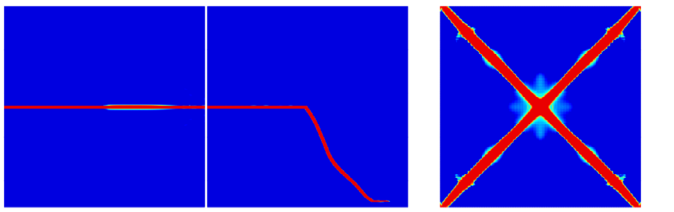
In this section, we showcase various examples to demonstrate the versatility of our method for simulating tearing and fracture of thin materials represented as continuum shells. We list all timing information and simulation parameters in Table 2. All simulations were run on an Intel Core i9-7920X processor (2.90 GHz, 24 cores) with 62.6GB RAM.

8.1. Mode tests

We verify our method by comparing it with engineering benchmarks. Figure 11 shows our propagation results under (a) Mode-I and (b) Mode-II loading. Our simulator produces the expected behaviour with the given loading conditions. We also evaluate our method's ability to resolve fracture within a simply supported plate under uniform pressure. The results are identical to Kiendl *et al.* [KADL*16] (refer to Figure 12 in their work).

Table 2: *Performance summary. Columns: material density ρ (kg/m^3); Young's modulus E (J/m^2); Poisson's ratio ν; reference stress σ^f (J/m^2); MPM grid size h (m); MPM time-step size Δt (s); total particle number in $\Gamma^{(I)}$, $\Gamma^{(II)}$ and $\Gamma^{(III)}$ ($N_{(I),(II),(III)}$); number of Gaussian quadrature points $\in \Gamma^{(IV)}$ ($N_{(IV)}$); total number of time-steps ($N_{\Delta t}$); average computation time per time-step in seconds ($s/\Delta t$). The total time of Figure 17 includes 0.5 h of rigid body simulation time.*

Scene	ρ	E	ν	σ_f	h	Δt	$N_{(I),(II),(III)}$	$N_{(IV)}$	$N_{\Delta t}$	$s/\Delta t$
Figure 1	—	—	—	—	1.0E-2	5.0E-6	170K	630K	12,000	2.73
Body	500	3.2E8	0.3	6.5E6	—	—	—	—	—	
Window	300	2.8E9	0.3	8.2E6	—	—	—	—	—	
Figure 4	500	3.2E5	0.3	4.5E4	1.5E-2	2.0E-4	23K	160K	4000	1.29
Figure 9	100	3.2E3	0.3	1.5E2	1.0E-2	5.0E-5	20K	160K	3500	1.08
Figure 12	1.0E3	3.2E5	0.33	8.0E4	1.0E-2	2.0E-4	26k	160K	4500	1.57
Figure 13	1000	4.2E5	0.3	—	1.0E-2	2.0E-4	19K	160K	8000	1.23
Figure 15	20	3.2E3	0.3	5.0E2	9.0E-3	2.0E-4	103K	502K	6000	2.80
Figure 16	20	3.2E3	0.3	5.0E2	9.0E-3	2.0E-4	183K	502K	3400	4.10
Figure 17	4.0E3	3.2E10	0.18	2.2E7	1.0E-2	1.0E-6	60K	1.25M	4300	4.02
Figure 14	100	2.8E5	0.3	5.5E4/8.0E3	1.0E-2	1.0E-4	74K	500K	4000	3.00/3.02



(a) Mode-I. (b) Mode-II. (c) Simply supported plate.

Figure 11: *Reproducing benchmark test results in Kiendl* et al. *[KADL*16]. We restrict the crack from propagating further downward in (b) by applying a Neumann boundary condition at the bottom of the place.*

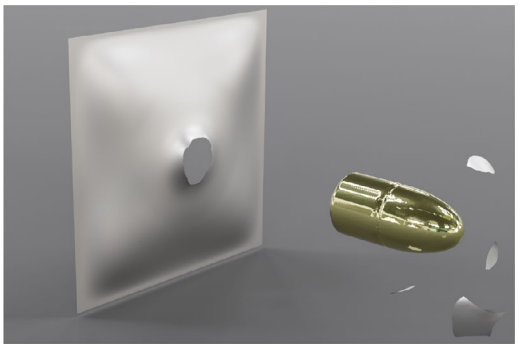


Figure 12: *A high-speed bullet hits a piece of thin metal sheet and leaves a permanent deformation.*

Figure 13: *A sheet is pre-cut with an Archimedean spiral curve. Our MLS formulation enables exact cutting with complex crack geometry.*

8.2. Elasto-plasticity

In Figure 12, we show the result of a simulation, involving plastic deformation for reproducing permanent-denting behaviour that is common in ductile fracture. For this test, we run a simulation where a high-speed bullet pierces a thin metal sheet, leaving a clean cut as fragment debris fly in the direction of travel. We are naturally able to incorporate the effect of plasticity in the shell.

8.3. Prescribed cut paths

Figure 13 demonstrates the versatility and controllable aspect of our method by prescribing a single crack path or multiple paths. This pre-cutting exemplifies our ability to incorporate user-control, which is desirable for artistic design and manipulation of simulation results.

(a) (b)

Figure 14: *Controlling crack behaviour when two rigid bunnies come into contact with a piece of cloth that is fixed along its four edges. (a) Homogeneous material; (b) grey region material is weakened by reducing the reference stress.*

8.4. Consistent crack paths with branching

Figure 14(a) shows that our method can robustly extract a consistent non-manifold crack path in a time-varying PF. This is a significant improvement over Fan *et al.* [FCK22], and is a novel result which overcomes a key problem of sensitivity (to input data) when computing medial axes that we use to define crack faces. As we fundamentally rely on PFs, our system is able to produce damage effects with bifurcation and merging which we use to extract and apply explicit crack paths on thin shells for modelling strong discontinuities.

8.5. Influencing fracture with material parameters

Our method likewise facilitates simulating objects with diverse properties since each particle can carry its own material parameters for characterizing behaviour. Weakened or reinforced regions can be simply represented as materials with different reference stress thresholds. We can also propagate cracks with complex material coupling scenarios as in Figure 14(b), which shows a crack initiating in a region where the material is softer and propagating to the rest of the domain. Note how the crack is not necessarily limited to following only the weakened region. Instead, secondary cracks are also allowed to appear in the unweakened region after crack initiation.

8.6. Complex surface geometries

To model complex geometries like cars and clothes, several NURBS patches are needed due to a single NURBS patch's tensor-product property. Across the patch boundaries, a C1 continuity needs to be imposed to ensure the smoothness of the geometry.

In Figure 15, the garment is torn by force applied on the bottom. Complex tears with bifurcations are produced leading to three fragments in the end. Figure 16 shows a related example in which an adorned mannequin is inflated at rest pose, resulting in the tearing of its outfit with several tears. We also show a more complex scenario in Figure 1 where a car composed of 12 NURBS patches with diverse materials is crashed by a frontal impactor and then subsequently by a tree.

8.7. Breaking glass

Our local PF formulation can likewise be coupled to a rigid body simulator. In Figure 17, we replicate the thin glass breakage test of Fan *et al.* [FCK22], where the glass is hit by a spherical marble.

Figure 15: *A garment is fixed on its shoulder strap and is pulled from its bottom. Branching cracks are generated and pass between patch borders.*

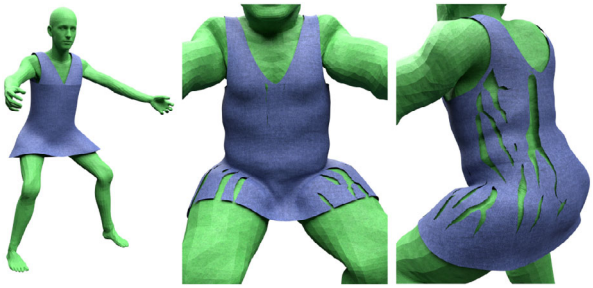


Figure 16: *Inflatable mannequin: The body of a mannequin dressed in blue is inflated, resulting in the tearing of its outfit (dress). Cracks pass through the boundary and tear the dress into fragments.*

The crack initiates from the contact region to form multiple circular patterns outwards.

8.8. Timing

The proposed method is efficient, introducing minimal overhead to the hybrid Langrangian/Eulerian formulation of thin shells [GHF*18] for fracture simulation. Figure 18 shows time spent in each task for simulating garment-tear. The most time-consuming part is the fully elasto-dynamic MPM simulation which takes about

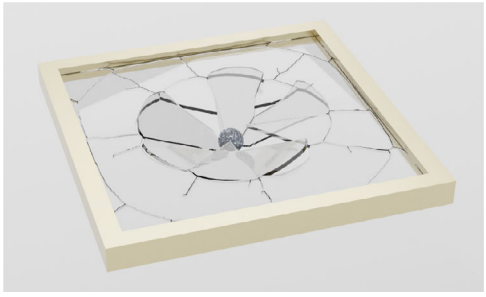


Figure 17: *A ball hits a thin glass to generate realistic angular crack patterns. Due to the restriction of tiny timestep size, we use a rigid body solver to simulate the dynamics of fragments as Fan* et al. *[FCK22].*

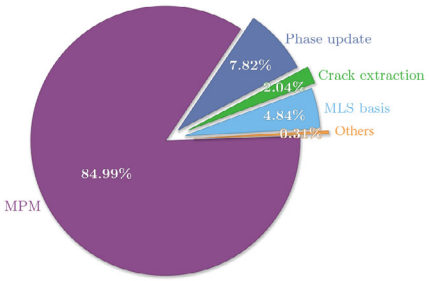


Figure 18: *Example breakdown of total simulation time using the result shown in Figure 15.*

85% of the total simulation time. This task alone is equivalent to the shell simulation in Guo *et al.* [GHF*18]. The remaining four parts are related to fracture simulation which takes the rest 15% time. In effect, our fracture simulations can be completed in a few hours or minutes as shown in Table 2.

9. Conclusion and Discussion

We have presented an extension to the hybrid Lagrangian/Eulerian formulation of thin shells to include (controllable) fracture. Thin shells are modelled with the Kirchhof–Love model of elasticity (with a novel in-plane stress evaluation), which is discretized using a NURBS surface-based representation. A MLS interpolation scheme is then proposed for capturing strong discontinuities in velocity fields with the aid of explicit crack paths. These crack paths are updated according to a stress-dependent evolution of PFs that resolve crack initiation and propagation automatically with branching and merging effects. Unlike previous methods, our approach provides a consistent approximation of the crack, which also prevents over-stretching near the crack boundary due to permanent material healing. The presented method is shown to be capable of simulating challenging scenarios of fracturing thin shells to open new ways toward user-directed control of this fracture process.

Our method obtains visually plausible results by making a few simplifying assumptions about the evolution of damage. Our elastic constitutive law is dependent on the *history* of deformation but only in the case of sustained external force-loading since we evaluate damage c from the strain at the current timestep. In contrast, conventional fully elasto-dynamic MPM methods that use maximum phase values fall into the dilemma of expanding crack regions forward while simultaneously broadening in perpendicular normal directions, which we overcome.

We adopted the MLS method for its flexibility in constructing basis functions which allows us to simulate cracks with arbitrary branchings. While we use a NURBS-based representation, our method is amenable to either NURBS- or subdivision-based surfaces as their basis functions have the sought property of being twice differentiable, which is required to compute covariant basis vectors $\mathbf{g}_3^{KL}$ for evaluating the deformation gradient. Subdivision surfaces are common in graphics, but NURBS are also ideal and crucial for designs in manufacturing of cloth and other models including mechanical tools, vehicles as well as biological tissue.

Our method gives artists maximum control over the fracture propagation pattern. Users can choose either full control by prescribing the crack path at the beginning of simulation or partial control by weakening/reinforcing the material properties. The former enforces crack to propagate along the prescribed path strictly, while the latter animates the natural breakage pattern in composite materials where the crack propagation favours the weakened region but not necessarily the same path.

Our method gives arbitrary complex cracks thanks to the evolving PF. The extracted crack path is consistent temporally. Compared with Pfaff *et al.* [PNDJO14], our method features natural branching and merging fracture pattern without the need of remeshing and complex geometry operations.

Limitations and future work: As we rely on explicit integration, simulating stiff materials will inherently require extremely small timesteps to impact total simulation time. This issue can of-course be remedied by using implicit integration with larger timesteps, but the difficulty of capturing crack propagation with large time-steps inherently limits the maximum time step size. Formulating an implicit integrator for continuum shell fracture problems, therefore, remains an open problem with additional requirements on devising good pre-conditioners for a reliable iterative solver.

We successively update damage from in-plane stress with bending resistance but do not strictly account for shearing and/or compression during these updates since we consider stress only at Type-IV particles. Thus, severe shearing or compressive deformation of the continuum shell will only have indirect influence on the weakening of (bending) stress as shearing forces also penalize in-plane deformations. In the future, we will explore incorporating damage from shearing and compression components. One possible way is to have two PFs, which are stored in both Type-III and Type-IV particles, respectively. Since they are located in the same coordinate system, a consistent crack can be extracted using the union of these two PFs.

Acknowledgements

This work is partly supported by the Innovation and Technology Commission of the HKSAR Government under the ITSP-Platform grant (Ref: ITS/335/23FP) and the InnoHK initiative (TransGP

project). The research work was in part conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

Conflicts of Interest

None of the authors have a conflict of interest to disclose.

References

[AKDL18] Ambati M., Kiendl J., De Lorenzis L.: Isogeometric Kirchhoff–Love shell formulation for elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering 340* (2018), 320–339.

[AMS*14] Amiri F., Millán D., Shen Y., Rabczuk T., Arroyo M.: Phase-field modeling of fracture in linear thin shells. *Theoretical and Applied Fracture Mechanics 69* (2014), 102–109.

[ATL22] Andrade H., Trevelyan J., Leonel E.: A NURBS-discontinuous and enriched isogeometric boundary element formulation for two-dimensional fatigue crack growth. *Engineering Analysis with Boundary Elements 134* (2022), 259–281.

[BDW13] Busaryev O., Dey T. K., Wang H.: Adaptive fracture simulation of multi-layered thin plates. *ACM Transactions on Graphics 32*, 4 (July 2013), 1–6.

[BFM00] Bourdin B., Francfort G. A., Marigo J.-J.: Numerical experiments in revisited brittle fracture. *Journal of the Mechanics and Physics of Solids 48*, 4 (2000), 797–826.

[BW98] Baraff D., Witkin A.: Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (1998), pp. 43–54.

[CC06] Cervera M., Chiumenti M.: Mesh objective tensile cracking via a local continuum damage model and a crack tracking technique. *Computer Methods in Applied Mechanics and Engineering 196* (2006), 304–320.

[CGA*17] Coox L., Greco F., Atak O., Vandepitte D., Desmet W.: A robust patch coupling method for NURBS-based isogeometric analysis of non-conforming multipatch surfaces. *Computer Methods in Applied Mechanics and Engineering 316* (2017), 235–260.

[CLdB17] Chen L., Lingen E. J., de Borst R.: Adaptive hierarchical refinement of NURBS in cohesive fracture analysis. *International Journal for Numerical Methods in Engineering 112*, 13 (2017), 2151–2173.

[CMSK20] Chitalu F. M., Miao Q., Subr K., Komura T.: Displacement-correlated XFEM for simulating brittle fracture. *Computer Graphics Forum 39* (2020), 569–583.

[CTT17] Clyde D., Teran J., Tamstorf R.: Simulation of nonlinear Kirchhoff-Love thin shells using subdivision finite elements. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA* (2017), vol. *17*.

[ES63] Erdogan F., Sih G. C.: On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engineering 85*, 4 (December 1963), 519–525. https://doi.org/10.1115/1.3656897.

[FCK22] Fan L., Chitalu F. M., Komura T.: Simulating brittle fracture with material points. *ACM Transactions on Graphics 41*, 5 (May 2022), 1–20.

[FM98] Francfort G. A., Marigo J.-J.: Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids 46*, 8 (1998), 1319–1342.

[GHDS03] Grinspun E., Hirani A. N., Desbrun M., Schröder P.: Discrete shells. In *Symposium on Computer Animation* (2003), D. Breen and M. Lin (Eds.), The Eurographics Association. https://doi.org/10.2312/SCA03/062-067.

[GHF*18] Guo Q., Han X., Fu C., Gast T., Tamstorf R., Teran J.: A material point method for thin shells with frictional contact. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–15.

[HH17] Homel M. A., Herbold E. B.: Field-gradient partitioning for fracture and frictional contact in the material point method. *International Journal for Numerical Methods in Engineering 109*, 7 (2017), 1013–1044.

[HVTG08] Harmon D., Vouga E., Tamstorf R., Grinspun E.: Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 Papers* (2008), pp. 1–4.

[JGT17] Jiang C., Gast T., Teran J.: Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–14.

[KADL*16] Kiendl J., Ambati M., De Lorenzis L., Gomez H., Reali A.: Phase-field description of brittle fracture in plates and shells. *Computer Methods in Applied Mechanics and Engineering 312* (2016), 374–394.

[KADLK21] Kikis G., Ambati M., De Lorenzis L., Klinkel S.: Phase-field model of brittle fracture in Reissner–Mindlin plates and shells. *Computer Methods in Applied Mechanics and Engineering 373* (2021), 113490.

[KLB14] Koschier D., Lipponer S., Bender J.: Adaptive tetrahedral meshes for brittle fracture simulation. In *Symposium on Computer Animation* (2014), pp. 57–66.

[LNTHZ20] Li W., Nguyen-Thanh N., Huang J., Zhou K.: Adaptive analysis of crack propagation in thin-shell structures via an isogeometric-meshfree moving least-squares approach. *Computer Methods in Applied Mechanics and Engineering 358* (2020), 112613.

[LZ14] Lu J., Zheng C.: Dynamic cloth simulation by isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering 268* (2014), 475–493.

[LZX22] LV A., ZHU Y., XIAN C.: Efficient cloth simulation based on the material point method. *Computer Animation and Virtual Worlds 33*, 3-4 (2022), e2073.

[NPO13] NARAIN R., PFAFF T., O'BRIEN J. F.: Folding and crumpling adaptive sheets. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 1–8.

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 1–10.

[NTB*91] NORTON A., TURK G., BACON B., GERTH J., SWEENEY P.: Animation of fracture by physical modeling. *The Visual Computer 7*, 4 (1991), 210–219.

[OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. In *Proceedings of ACM SIGGRAPH 2002* (Aug. 2002), ACM Press, pp. 291–294. http://graphics.berkeley.edu/papers/Obrien-GMA-2002-08/.

[OH99] O'BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH 1999* (Aug. 1999), ACM Press/Addison-Wesley Publishing Co., pp. 137–146. http://graphics.berkeley.edu/papers/Obrien-GMA-1999-08/, http://doi.acm.org/10.1145/311535.311550.

[PADLK21] PROSERPIO D., AMBATI M., DE LORENZIS L., KIENDL J.: Phase-field simulation of ductile fracture in shell structures. *Computer Methods in Applied Mechanics and Engineering 385* (2021), 114019.

[PKA*05] PAULY M., KEISER R., ADAMS B., DUTRÉ P., GROSS M., GUIBAS L. J.: Meshless animation of fracturing solids. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 957–964.

[PNDJO14] PFAFF T., NARAIN R., DE JOYA J. M., O'BRIEN J. F.: Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 1–9.

[PZM*20] PAUL K., ZIMMERMANN C., MANDADAPU K. K., HUGHES T. J., LANDIS C. M., SAUER R. A.: An adaptive space-time phase field formulation for dynamic fracture of brittle shells based on LR NURBS. *Computational Mechanics 65*, 4 (2020), 1039–1062.

[Sih74] SIH G. C.: Strain-energy-density factor applied to mixed mode crack problems. *International Journal of Fracture 10*, 3 (1974), 305–321.

[TF88] TERZOPOULOS D., FLEISCHER K.: Modeling inelastic deformation: Viscolelasticity, plasticity, fracture. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques* (1988), pp. 269–278.

[TWS05] THOMASZEWSKI B., WACKER M., STRAßER W.: A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Vienna, Austria (2006), pp. 107–116.

[WCL*20] WOLPER J., CHEN Y., LI M., FANG Y., QU Z., LU J., CHENG M., JIANG C.: AnisoMPM: Animating anisotropic damage mechanics. *ACM Transactions on Graphics 39*, 4 (Aug. 2020), 37:1–37:16.

[WDG*19] WANG S., DING M., GAST T. F., ZHU L., GAGNIERE S., JIANG C., TERAN J. M.: Simulation and visualization of ductile fracture with the material point method. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (July 2019), 1–20.

[WFL*19] WOLPER J., FANG Y., LI M., LU J., GAO M., JIANG C.: CD-MPM: Continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–15.

[XAM21] XUE T., ADRIAENSSENS S., MAO S.: Mapped phase field method for brittle fracture. *Computer Methods in Applied Mechanics and Engineering 385* (2021), 114046.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information

Supporting Information