ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2024
M. Skouras and H. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 8

Generalized eXtended Finite Element Method for Deformable Cutting via Boolean Operations

Q. M. Ton-That[1] and P. G. Kry[2] and S. Andrews[1]

[1]École de Technologie Supérieure, Canada
[2]McGill University, Canada

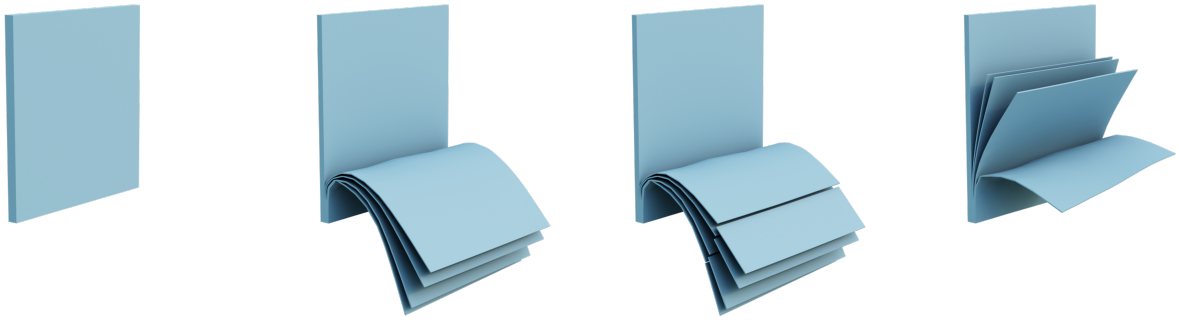

Figure 1: *Multiple thin cuts can be made in a slab to create thin deformable sheets, where the mass and stiffness decreases with each cut. An orthogonal cut across these sheets demonstrates the ability of our generalized method to handle intersecting cuts.*

Abstract

Traditional mesh-based methods for cutting deformable bodies rely on modifying the simulation mesh by deleting, duplicating, deforming or subdividing its elements. Unfortunately, such topological changes eventually lead to instability, reduced accuracy, or computational efficiency challenges. Hence, state of the art algorithms favor the extended finite element method (XFEM), which decouples the cut geometry from the simulation mesh, allowing for stable and accurate cuts at an additional computational cost that is local to the cut region. However, in the 3-dimensional setting, current XFEM frameworks are limited by the cutting configurations that they support. In particular, intersecting cuts are either prohibited or require sophisticated special treatment. Our work presents a general XFEM formulation that is applicable to the 1-, 2-, and 3-dimensional setting without sacrificing the desirable properties of the method. In particular, we propose a generalized enrichment which supports multiple intersecting cuts of various degrees of non-linearity by leveraging recent advances in robust mesh-Boolean technology. This novel strategy additionally enables analytic discontinuous integration schemes required to compute mass, force and elastic energy. We highlight the simplicity, expressivity and accuracy of our XFEM implementation across various scenarios in which intersecting cutting patterns are featured.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

Cutting and fracturing of deformable solids is important for many computer graphics applications, such as virtual surgery, visual effects for fracture and tearing in film and games, and engineering simulations. However, there are longstanding challenges related to the robustness and computational performance of simulations involving cutting. These mainly stem from the topological

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

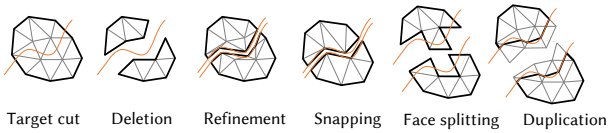

Figure 2: *Traditional mesh-based cutting methods modify the simulation mesh to approximate the cut's geometry.*

and geometrical changes of the underlying model that cutting introduces.

Mesh-based approaches for elastic simulation are favored in many interactive computer graphics applications due to their efficiency, in particular the finite element method (FEM). Yet cutting leads to numerous challenges for FEM. On the one hand, remeshing operations allow accurate reconstruction of the cutting geometry, however the simulation may become unstable due to the resulting ill-shaped elements. Element deletion, splitting, and duplication techniques can be used to avoid remeshing, but physical fidelity is sacrificed due to imprecise simulation quantities ensuing from these operations.

The extended finite element method (XFEM) was developed specifically to alleviate the shortcomings of FEM when dealing with cutting and fracturing. The approach enriches the model with additional degrees of freedom that specifically handle discontinuities introduced by cutting. State-of-the-art XFEM algorithms demonstrate an impressive ability to generate conforming cuts without remeshing or other geometric modifications. However, a key challenge here is computing integrals associated with simulation forces, masses and stiffness over discontinuous domains introduced by cuts. Recent work by Koschier et al. [KBT17] addresses this challenge by proposing a specialized discontinuous quadrature rule. However, their approach does not support the case of intersecting cuts, which is a requirement for many real-life applications.

Our work proposes a novel cutting model based on mesh Boolean operations, unlocking a new set of expressive XFEM enrichments that support multiple intersecting cut geometries. Furthermore, the method greatly simplifies the resulting discontinuous integrals by leveraging the divergence theorem for efficient, analytic and accurate integration of mass and stiffness, leading to high-fidelity simulations. Figure 1 shows a preview of our results.

2. Related Work

We begin with a brief survey of simulation methods for cutting elastic solids, and highlight their strengths and weaknesses. We refer readers wishing to dive deeper into the relevant work on cutting simulation to the enlightening surveys on the topic [BSM*02, WWD15, WM18]. Early cutting approaches in the field of computer graphics relied on directly modifying the simulation mesh to account for the resulting discontinuities. Figure 2 shows visual examples of the reviewed mesh modification strategies for cutting.

Element deletion. One of the most straightforward approaches for updating a mesh-based model due to cutting are element deletion methods [DCA99, CDA00]. Elements intersected by the cutting path are simply removed, resulting in increased computational efficiency as the simulation progresses, at the cost of inaccurate physical and visual outcomes.

Element refinement. On the other hand, element refinement methods subdivide the simulation mesh in cut regions, conforming to the discontinuity's geometry, thus yielding accurate computation of mass, force and elastic energy. Bielser et al. [BMG99] proposed a tetrahedron subdivision scheme based on cut edges supporting piece-wise linear cuts in elements, which was later improved by Bielser and Gross [BG00] and Mor and Kanade [MK00] to decrease the rate of creation of new elements. A state machine algorithm was designed by Bielser et al. [BGTG03] to account for general interactive cutting tool interactions with the simulated models. Nienhuys and Frank van der Stappen [NFvdS01] alternatively proposed to project vertices in the cut region to the discontinuity. More recently, Paulus et al. [PUC*15] targeted element reducing refinement schemes by considering element adjacencies in the subdivision process. Unfortunately, despite these efforts, the quality of the mesh eventually decreases, leading to slow and unstable simulations. These severe disadvantages led to the development of face splitting approaches [NvdS00, MG04, SHGS06], which restrict the cut path to lie on element facets, in an attempt to better balance the trade off between accuracy, stability and computational cost. While face splitting preserves mass and volume properties correctly, it remains a poor approximation of the desired discontinuities.

Element duplication. A popular choice for performing mesh-based cutting are element duplication techniques, due to their simplicity and reliability. However, this often comes at the cost of reduced physical accuracy. Molino et al. [MBF04] first proposed the virtual node algorithm (VNA) which creates copies of original elements, resulting in non-manifold topologies which capture the cut geometry at the simulation mesh's resolution. Although the resulting mass and stiffness properties increase artificially, the visual representation is embedded in the simulation mesh, conforming to the cut's geometry. Improved VNA variants have since been proposed that enable higher resolution visually conforming cuts [SDF07, WJST15]. Perhaps the closest approach to ours in this class of method is that of Sifakis et al. [SDF07], since they also support arbitrary cutting configurations. However, our work differs in two distinct ways. First, floating point precision issues are mitigated through our use of a robust technique for computing mesh arrangements [ZGZJ16]. Additionally, since our method employs an exact spatial integration technique, rather than element duplication, the physical properties computed for cut regions using our approach are more precise.

Voxel representations. The reliability of duplication methods, coupled with the convenience of voxel representations [FG99] led to further advancements in mesh-based cutting by demonstrating improved efficiency, simplicity and robustness [DGW11, JBB*10, SSSH11, WDW11]. Ultimately, accuracy remains the inevitable major drawback of element duplication. Acceleration methods also exist to complement mesh modification. Ganovelli et al. [GCMS00] combined subdivision schemes with multi-resolution mesh representations, while other work has proposed incremental matrix

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

factorization updates when using approximate fast time integration schemes [LLKC21, YCP16, YPC20].

eXtended Finite Element Method. The pioneering work of Moës et al. [MDB99] in the engineering literature proposed XFEM, which eliminates the pain points accompanying previous mesh modification approaches. The fundamental principle behind XFEM is to capture the discontinuity in the function space where our FEM discretization lives using enrichment functions, rather than in the geometry of the underlying mesh itself, allowing for remeshing-free simulation of discontinuous phenomena. This separation between simulation mesh and cut representations allowed the work that followed to enhance the capabilities of XFEM by focusing on these so-called enrichments functions. Daux et al. [DMD*00] proposed hierarchical junction enrichments to accommodate intersecting cuts by treating them as branched cracks, whereas Kaufmann et al. [KMB*09] achieved sub-element continuous cutting resolution by modeling enrichments as solutions to the harmonic equation subject to boundary conditions imposed by the cut's path. Mousavi et al. [MGS11a, MGS11b] later extended this work to intersecting branched cracks and higher order basis functions. The aforementioned methodologies are particularly well-suited to crack propagation simulations in 2D. Other work has demonstrated the suitability of XFEM in large deformation scenarios [JK09]. Although these qualities are quite impressive, failure to integrate mass and stiffness accurately in XFEM directly results in jarring simulation instability [KBT17, JK09]. Later work proposed discontinuous quadrature rules to address this issue [MKOW12, MKO13, KBT17]. More recent work demonstrates the ongoing relevance of XFEM [HSK*19, CMSK20] in scenarios requiring high fidelity discontinuous behavior. Several works in the mechanical engineering literature [RHS*11, HH04, SAB06, DSMB09] claim a similar feature set to ours, yet lack demonstrations on 3D scenarios. In contrast, within the graphics literature, mesh-based cutting algorithms either support accurate cutting behavior, or intersecting cuts, but not both. Koschier et al. [KBT17] show advantages over these existing techniques, yet we extend their method to intersecting cuts as well as faster and simpler spatial integration without loss of functionality.

Mesh-free methods. The numerous meshless methods developed in the past decades demonstrate the continuing desire for remeshing-free cutting methodologies. However, they are considered computationally costly compared to mesh-based approaches, requiring much denser discretizations in order to accurately capture simulation details. As our work does not address challenges in meshless simulation, we only provide a cursory overview of the many meshless simulation methods supporting discontinuities that have been developed in both the engineering and computer graphics literature [BLG94, BGL94, NRBD08, OFTB96, DG96, JL12, SOG06, SOG08, SOG09, MKN*04, HFG*18, MO20].

Remeshing-free. Although XFEM aims to avoid modifying the simulation mesh, the visual mesh inevitably requires remeshing for visual fidelity. Similarly, remeshing a background integration mesh is common [RHS*11, JCD02, BCXZ03] to capture physically accurate behavior, especially when the same regions can be cut more than once. When simplifying assumptions on the simulation can be made, fixed quadrature may be sufficient [SAB06, DSMB09]. In this sense, Koschier et al. [KBT17] develop a "true" remeshing-free XFEM approach. Reliance on mesh processing can be decreased even further via implicit representations of the discontinuity, but these do not support intersecting cuts [ZKBT17, MMDH*23, HSK*19]. The benefits of avoiding remeshing the simulation mesh are twofold. The quality of the original uncut mesh's shape functions is preserved during cutting [She02], and the complexity of the simulation scales with the number of discontinuities, rather than with their geometric discretization's resolution. Likewise, meshless methods, although termed "meshless", may also use a background mesh for accurate spatial integration [NRBD08, BKO*96], which does not affect the displacement field's discretization.

3. Background

In this section, we provide a brief overview of the fundamentals of dynamic simulation of elastic solids using the finite element and extended finite element methods, thus laying the groundwork for our proposed methodology.

3.1. Elastodynamics

Simulations of mechanical systems obeying the principle of least action satisfy the Euler-Lagrange equation

$$\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}L - \nabla_{\mathbf{q}}L = 0\,, \tag{1}$$

where $L(\mathbf{q},\dot{\mathbf{q}},t)$, known as the *Lagrangian*, is a scalar function describing the dynamic behavior of our system at any time t, and $\mathbf{q}(t)$ are generalized coordinates describing its state.

We focus on the particular case of non-linear elastodynamic systems in d embedding dimensions. Here, the central quantity of interest is the displacement field $\mathbf{u}(\mathbf{X},t) : \Omega \to \mathbb{R}^d$ that parameterizes the deformed spatial configuration $\mathbf{x}(\mathbf{X},t) \in \mathbb{R}^d$ of $\mathbf{X}$. The deformed configuration can be computed at any time t as

$$\mathbf{x}(\mathbf{X},t) = \mathbf{X} + \mathbf{u}(\mathbf{X},t). \tag{2}$$

In this setting, the Lagrangian measures the balance between total kinetic and potential energies in a continuous, but bounded, material domain Ω, whose behavior is characterized by a mass density ρ, a strain energy density Ψ, and body forces $\mathbf{b}$, such that

$$L(\mathbf{u},\dot{\mathbf{u}},t) = \int_\Omega \frac{1}{2}\rho\dot{\mathbf{u}}^T\dot{\mathbf{u}} - \Psi(\mathbf{u}) + \mathbf{u}^T\mathbf{b}\,\partial\Omega\,. \tag{3}$$

3.2. Finite Element Method

The FEM expresses $\mathbf{u}(\mathbf{X},t)$ on a meshed domain Ω, with nodes V and elements E, in the form of a linear combination of time-varying coefficients $\mathbf{u}_i(t) \in \mathbb{R}^d$ and spatially varying basis functions $\phi_i(\mathbf{X}) \in \mathbb{R}$:

$$\mathbf{u}(\mathbf{X},t) = \sum_{i\in V} \mathbf{u}_i(t)\phi_i(\mathbf{X})\,. \tag{4}$$

Observe from Eq. 4 that each coefficient and basis function is associated with a single node $i \in V$.

While the coefficients $\mathbf{u}_i(t)$ are obtained during the simulation by time integration, the basis functions $\phi_i(\mathbf{X})$ are constructed prior to

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

the simulation using local geometric interpolation functions $N_i^e(\mathbf{X})$, commonly referred to as *shape functions*, associated with node i and supported in the domain Ω^e of element e. We can easily construct these shape functions in each element e independently by solving the interpolation problem

$$P(\mathbf{X}_i)^T \mathbf{a}_j^e = \delta_{ij} \; \forall \, i, j \in V^e, \tag{5}$$

where $P(\cdot)$ is a polynomial basis, $\mathbf{a}_j^e$ its coefficients and δ_{ij} is the Kronecker delta. The shape function is then computed by the dot product $N_i^e(\mathbf{X}) = P(\mathbf{X})^T \mathbf{a}_i^e$ for $\mathbf{X} \in \Omega^e$. A popular choice of polynomial basis in graphics is the 3D linear monomial $P(\mathbf{X}) = [1 \; X \; Y \; Z]^T$.

The meshing of Ω is assumed to satisfy $\bigcup_{e \in E} \Omega^e = \Omega$ and $\Omega^e \cap \Omega^{\tilde{e}} = \emptyset$ for $e \neq \tilde{e}$. In other words, the union of elements covers the domain, and elements are non-overlapping except at their interfacing boundaries. This allows basis functions to be defined concisely by the union of shape functions over the mesh one-ring neighborhood

$$\phi_i(\mathbf{X}) = \sum_{e \in E^i} N_i^e(\mathbf{X}),$$

where E^i denotes the elements incident on node i. Consequently, Eq. 4 is continuous everywhere in Ω. By construction, the basis functions satisfy the Kronecker delta property $\phi_i(\mathbf{X}_j) = \delta_{ij}$, such that $\mathbf{u}(\mathbf{X}_i, t) = \mathbf{u}_i(t)$ is interpolating at nodes, facilitating the imposition of boundary and initial conditions.

3.3. Equations of Motion

We can inject Eq. 3 into Eq. 1 and let $\mathbf{q}_i(t) = \mathbf{u}_i(t)$, to recover Newton's equations of motion

$$\mathbf{M}\ddot{\mathbf{u}} = -\nabla U(\mathbf{u}) + \mathbf{f}, \tag{6}$$

where the mass matrix $\mathbf{M} \in \mathbb{R}^{|V|d \times |V|d}$ has $d \times d$ blocked entries $\mathbf{M}_{ij}$, the force vector $\mathbf{f} \in \mathbb{R}^{|V|d}$ has $d \times 1$ blocked entries $\mathbf{f}_i$, U is the total elastic potential energy, and $\mathbf{u}$ is a vector collecting the coefficients $u_i(t)$. We compute their values by integration over the material domain:

$$\mathbf{M}_{ij} = \int_\Omega \rho \phi_i(\mathbf{X}) \phi_j(\mathbf{X}) d\Omega \otimes \mathbf{I}_{\mathbf{d} \times \mathbf{d}}, \tag{7}$$

$$\mathbf{f}_i = \int_\Omega \mathbf{b} \phi_i(\mathbf{X}) d\Omega \otimes \mathbf{1}_d, \tag{8}$$

$$U(\mathbf{u}) = \int_\Omega \Psi(\mathbf{u}) d\Omega. \tag{9}$$

Although there are various choices of stable time integration schemes for solving Eq. 6, our work assumes a first-order backward Euler (BDF1) discretization with time step h, which yields the following optimization problem:

$$\mathbf{u}^{t+1} \leftarrow \arg\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \tilde{\mathbf{u}}\|_{\mathbf{M}}^2 + h^2 U(\mathbf{u}). \tag{10}$$

Here, $\|\cdot\|_{\mathbf{M}}^2$ denotes the mass-weighted squared 2-norm, and $\tilde{\mathbf{u}} = \mathbf{u}^t + h\dot{\mathbf{u}}^t + h^2 \mathbf{M}^{-1} \mathbf{f}$.

3.4. eXtended Finite Element Method

The extended finite element method, as its name implies, extends FEM by adapting the basis $\{\phi_i\}$ of $\mathbf{u}(\mathbf{X}, t)$ with enrichments $\{\psi_i^c\}$ where discontinuities Γ^c exist, which we will often refer to as *cuts*. In the elastodynamic setting, Γ^c is represented by spatially continuous time-varying paths traced by a cutting tool, such as a knife or a scalpel, and the enriched basis allows a meshed deformable body Ω to separate across Γ^c.

The fundamental building block used to achieve such a feat is the sign function $\mathrm{sgn}^c(\mathbf{X}) : \mathbb{R}^d \to \{-1, 1\}$ associated with cut Γ^c, which encodes the geometry of Γ^c implicitly, thus generalizing to d dimensions. Traditionally, $\mathrm{sgn}^c(\mathbf{X})$ need only form a binary partitioning of space whose interface coincides with Γ^c. Such a function is easily obtained by normalizing the common signed distance function $\mathrm{sdf}^c(\mathbf{X})$ to the cut Γ^c.

Basis function enrichments are subsequently constructed from sgn^c as d-dimensional signed Heaviside step functions, also known as *shifted sign enrichments* (SSE),

$$\psi_i^c(\mathbf{X}) = \frac{1}{2}(\mathrm{sgn}^c(\mathbf{X}) - \mathrm{sgn}^c(\mathbf{X}_i)). \tag{11}$$

One can intuitively deduce that $\psi_i^c(\mathbf{X}) = \pm 1$ whenever $\mathbf{X}$ lies on the side of the cut across from node i, while it vanishes when sharing the same sidedness with node i. An important property of the SSE is that it preserves the Kronecker delta property, thus keeping $\mathbf{u}(\mathbf{X}, t)$ interpolating.

Enriched basis functions $\phi_i(\mathbf{X})\psi_i^c(\mathbf{X})$ are thus understood as original basis functions ϕ_i whose domains Ω^i are restricted to the opposite side of Γ^c from node i. If Ω^i is not intersected by Γ^c, then $\psi_i^c(\mathbf{X}) = 0$ everywhere. Consequently, it is unnecessary for Γ^c to enrich such a node i, allowing for purely local modifications to the displacement field , which can therefore be conveniently written as

$$\mathbf{u}(\mathbf{X}, t) = \sum_{i \in V} \mathbf{u}_i(t) \phi_i(\mathbf{X}) + \sum_{c=1}^{|\Gamma|} \sum_{i \in \mathcal{V}^c} \mathbf{u}_i^c(t) \phi_i(\mathbf{X}) \psi_i^c(\mathbf{X}), \tag{12}$$

where $\Gamma = \{\Gamma^c\}$ is the set of all discontinuities at time t, $\mathcal{V}^c$ is the set of nodes enriched by Γ^c, and $\mathbf{u}_i^c(t)$ are node i's coefficients associated with Γ^c.

We discover from Eq. 12 two of XFEM's major advantages over traditional mesh-based cutting algorithms:

1. Topological changes to mesh Ω are never required, i.e., XFEM is remeshing free, because the original basis functions $\phi_i(\mathbf{X})$ are always reused. Thus, mesh quality is preserved during the whole simulation.
2. Cut geometries Γ^c are fully decoupled from the mesh Ω, being entirely captured by $\mathrm{sgn}^c(\mathbf{X})$ which appears in the enrichments $\psi_i^c(\mathbf{X})$, allowing Γ^c to represent a large class of complex discontinuities accurately.

3.5. Discontinuous Integration

Naturally, a redefinition of the displacement field $\mathbf{u}(\mathbf{X}, t)$ also redefines the mass, force and elastic energy from Eqs. 7, 8 and 9. In the enriched basis, we obtain mass block matrices $\mathbf{M}_{ij}^{cl} \in \mathbb{R}^{d \times d}$

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

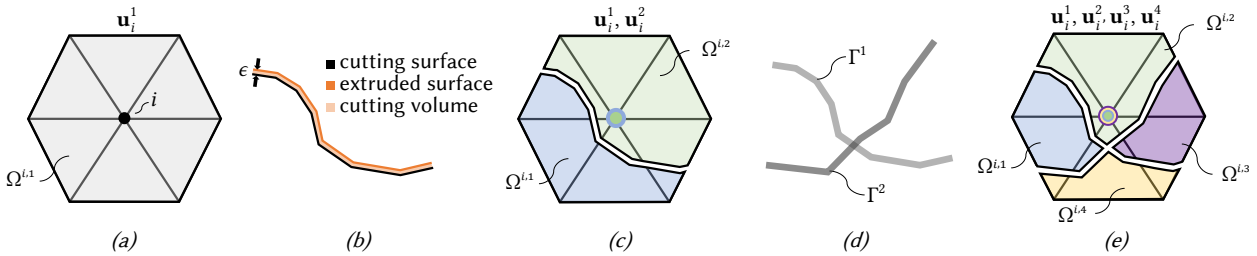

Figure 3: *A 2D visualization showing the 1-ring neighborhood Ω^i of a node i controlled by degree of freedom (DOF) $\mathbf{u}_i^1$. (a) The original 1-ring neighborhood for node i. (b) A cutting volume is generated by extruding the polygonal cutting surface by thickness ϵ. (c) A mesh Boolean operation adjusts $\Omega^{i,1}$ and creates a new subdomain $\Omega^{i,2}$ with its associated DOF $\mathbf{u}_i^2$. (d) A subsequent intersecting cut Γ^2 results in (e) 2 new subdomains $\Omega^{i,3}$ and $\Omega^{i,4}$ and DOFs $\mathbf{u}_i^3$ and $\mathbf{u}_i^4$.*

and force block vectors $\mathbf{f}_i^c \in \mathbb{R}^d$ for nodes i, j enriched by cuts c, l respectively, written as

$$\mathbf{M}_{ij}^{cl} = \int_\Omega \rho \phi_i \psi_i^c \phi_j \psi_j^l d\Omega \otimes \mathbf{I}_{d\times d}, \tag{13}$$

$$\mathbf{f}_i^c = \int_\Omega \mathbf{b} \phi_i \psi_i^c d\Omega \otimes \mathbf{1}_d. \tag{14}$$

The mass matrix and force vector may then be constructed by flattening Eq. 13 and Eq. 14 in the cut indices c and l's dimensions. While the total elastic potential $U(\mathbf{u})$ now also takes into account enriched basis functions $\phi_i(\mathbf{X})\psi_i^c(\mathbf{X})$ and their coefficients $\mathbf{u}_i^c(t)$, its specific definition depends on the selected material model. In all three cases, specialized integration techniques must be adopted to account for the discontinuous integrands.

4. Methodology

We now depart from previous work by proposing a *generalized* XFEM framework that allows mesh-based simulations to accurately simulate cutting scenarios. Our approach aims to eliminate limitations imposed by many of the aforementioned approaches, specifically their inability to handle arbitrary intersecting cuts while also accurately and robustly simulating physical behavior.

4.1. Cut Modeling

Our method assumes the discontinuities Γ^c represent time-varying trajectories of a cutting tool's sharp edge passing through the simulated domain Ω. Approximating the tool's sharp edge by a curve in 3-dimensional space, Γ^c becomes a swept surface, an inherently 2D entity. Koschier et al. [KBT17] highlight the advantages of triangle mesh representations of Γ^c in XFEM. Expanding on this, we introduce a simple modification – solidification of the swept surface – which lays the groundwork for our generalized XFEM. The swept surfaces Γ^c are lifted to 3D by injecting a thickness $\epsilon > 0$ along their normal direction, making Γ^c a thin, closed solid with boundary $\partial\Gamma^j$.

We represent the solid Γ^c by a triangle meshing of its boundary $\partial\Gamma^c$, enabling the use of recent state-of-the-art robust mesh Boolean operations [ZGZJ16] as intuitive, yet powerful geometry processing tools for surface mesh manipulation. Such tools will prove fundamental to unlocking a new set of expressive enrichment functions and integration schemes, which we present in Sections 4.2 and 4.3.

4.2. Generalized Enrichments

The need for a new set of enrichment functions is motivated by examining existing ones. In the 1D setting, discontinuities are points on the real line $\mathbb{R}$, which can only partition space into two regions. Consequently, it is sufficient to capture each discontinuity by a single sign function $\text{sgn}^c(\mathbf{X})$, and each node need only yield a single enrichment per cut Γ^c.

Raising the embedding space to $\mathbb{R}^2$ transforms discontinuities from points to curves, revealing two important limitations of the previous approach. First, a single cut Γ^c may not pass through the same nodal support Ω^i twice, since it would require the model to allow multiple nodal enrichments associated with the same cut. Second, assuming the previous scenario can be avoided, if $k \geq 2$ cuts pass through and intersect within nodal neighborhood Ω^i, then Ω^i will separate into $S > k$ independent subsets of Ω^i. In other words, the resulting k enriched degrees of freedom will be insufficient to capture the motion of S distinct pieces of material.

Daux et al. [DMD*00] and later Mousavi et al. [MGS11a] recognized limitations related to intersecting cuts and proposed *branched* enrichments located at intersections between the discontinuities. Discontinuities Γ^c in $\mathbb{R}^2$ intersect at points, such that a branching structure emerges, unambiguously separating Γ^c into a hierarchical structure. Temporal coherence of the cuts then allows so-called main *cracks*, i.e., discontinuities Γ^c, to separate and branch out into secondary cracks. In $\mathbb{R}^3$, however, this branching structure is ambiguous. Discontinuities Γ^c evolve from curves to surfaces, and the intersections between them, from points to curves. Unfortunately, intersecting surfaces do not unequivocally separate into distinct geometries. For these reasons, we propose a *generalized* enrichment that is effective for 1D, 2D, and 3D embeddings.

We derive our enrichments from the realization that delineating the sidedness of space with respect to a discontinuity does not

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

align with our objective, which is to simulate material separation. Instead, enrichments should aim to identify the distinct pieces of material resulting from said discontinuities, allowing the displacement field $\mathbf{u}(\mathbf{X},t)$ to treat each piece independently. We thus discard the traditional sign-based enrichments in favor of a material indicator function. Because the spatial behavior of $\mathbf{u}(\mathbf{X},t)$ is entirely defined by its basis functions $\phi_i(\mathbf{X},t)$, it is sufficient to restrict our attention to the role of indicator functions associated with nodal domain Ω^i.

Separating Ω^i into its distinct pieces, or *nodal subdomains* $\Omega^{i,s}$, due to cuts Γ is trivially achieved in the continuous setting by the Boolean difference operator $\{\Omega^{i,s}\} = \Omega^i \setminus \bigcup_{c=1}^{|\Gamma|} \Gamma^c$. In the discrete setting, although Boolean operations are straightforwardly applied to implicit representations, it is generally non-trivial to identify the resulting connected components. Fortunately, mesh Booleans benefit from the best of both worlds. Given an explicit polygon mesh representation of the boundaries $\partial\Omega^i$ and $\partial\Gamma^c$ of the solids Ω^i and Γ^c respectively, mesh Booleans yield the desired subdomains $\Omega^{i,s}$ in the form of their meshed boundaries $\partial\Omega^{i,s}$. The key additional benefit here is that identification of the subdomains $\mathcal{S}^i = \{\Omega^{i,s}\}$, i.e., the connected components, is trivial via breadth/depth first search over the graph of adjacent triangles of $\mathcal{S}^i$.

Building upon this, we make use of an unsigned distance measure

$$\text{dist}(\mathbf{X}, \partial\Omega^{i,s}) = \min_{\mathbf{Y}\in\partial\Omega^{i,s}} \|\mathbf{X}-\mathbf{Y}\|_2^2 \tag{15}$$

between a point $\mathbf{X} \in \mathbb{R}^d$ and the meshed nodal subdomain boundary $\partial\Omega^{i,s}$. We use this distance to define our generalized XFEM enrichment function $\psi_i^s(\mathbf{X})$ as the aforementioned material indicator function

$$\psi_i^s(\mathbf{X}) = \begin{cases} 1 & \text{dist}(\mathbf{X},\Omega^{i,s}) < \text{dist}(\mathbf{X},\Omega^{i,s'}) \;\forall\, s' \neq s \\ 0 & \text{otherwise} \end{cases}, \tag{16}$$

where the subscripts s and s' identify the nodal subdomains of node i. Intuitively, $\psi_i^s(\mathbf{X})$ identifies its assigned independent piece of material by a positive sign inside of it, while it vanishes outside. Hence, enriched basis functions $\phi_i(\mathbf{X})\psi_i^s(\mathbf{X})$ are restricted to $\Omega^{i,s}$, which we control using enriched degrees of freedom $\mathbf{u}_i^s(t)$. Figure 3 shows an illustration of this. Due to our cuts' thickness parameter ϵ, the unsigned distance between two separate nodal subdomains $\Omega^{i,s}$ and $\Omega^{i,s'}$ is guaranteed to be greater than or equal to ϵ, i.e.,

$$\mathbf{X} \in \Omega^{i,s} \rightarrow \text{dist}(\mathbf{X},\Omega^{i,s'}) - \text{dist}(\mathbf{X},\Omega^{i,s}) \geq \epsilon \;\forall\, s' \neq s\,, \tag{17}$$

freeing $\psi_i^s(\mathbf{X})$ of ambiguity at material points $\mathbf{X} \in \Omega \setminus \bigcup_{c=1}^{|\Gamma|} \Gamma^c$. Our displacement field is then easily described as

$$\mathbf{u}(\mathbf{X},t) = \sum_{i\in V} \sum_{s\in\mathcal{S}(i)} \mathbf{u}_i^s(t)\phi_i(\mathbf{X})\psi_i^s(\mathbf{X})\,, \tag{18}$$

where $\mathcal{S}(i)$ is the index set of subdomains $\Omega^{i,s}$ of node i.

Kaufmann et al. [KMB*09] briefly discussed similar generalized enrichments, although they ultimately propose harmonic enrichments, which only support intersecting cuts by adhering to the branched crack assumption [MGS11a]. Our approach is also reminiscent of virtual node algorithms (VNA), which rely on assigning new degrees of freedom to "scoops" of material severed from nodal 1-rings. In contrast to VNA, our approach places no restriction on the configuration of the cuts, offers adaptivity out of the box thanks to the composability of Boolean operations, and integrates physical quantities accurately for simulation.

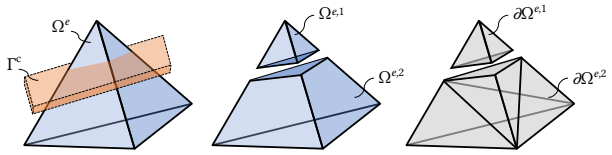

Figure 4: *Two element sub-domains $\Omega^{e,1}$ and $\Omega^{e,2}$ are created by cut Γ^c. Mesh Booleans provide us with boundary tessellations $\partial\Omega^{e,1}$ and $\partial\Omega^{e,2}$.*

4.3. Spatial Integration

The benefits of mesh Booleans also carries over to the problem of accurate spatial integration. We briefly remind the reader that separability of the integrals allows *element*-wise integration, such that

$$\int_\Omega F'(\mathbf{X})d\Omega = \sum_{e\in E} \int_{\Omega^e} F'(\mathbf{X})d\Omega^e\,, \tag{19}$$

and Eqs. 9, 13, and 14 are computed by substituting $F'(\mathbf{X})$ with the appropriate integrand. Thus, as usual in the FEM literature, we focus on element-wise quantities. From this perspective, we can apply the Boolean difference operator once more to yield *element subdomains* $\{\Omega^{e,s}\} = \Omega^e \setminus \bigcup_{c=1}^{|\Gamma^c|} \Gamma^c$ in which the enriched basis is piecewise continuous. We denote the restriction of Ω^e to the cuts Γ^c as $\mathcal{E}^- = \Omega^e \cap \bigcup_{c=1}^{|\Gamma^c|} \Gamma^c$, allowing us to further split Eq. 19 into element subdomain integrals:

$$\int_{\Omega^e} F'(\mathbf{X})d\Omega^e = \sum_{s=1}^{|\mathcal{S}^e|} \int_{\Omega^{e,s}} F'(\mathbf{X})d\Omega^{e,s} + \int_{\mathcal{E}^-} F'(\mathbf{X})d\mathcal{E}^-\,. \tag{20}$$

Examining Eq. 20 reveals that the second term on the right-hand side vanishes as the cut solids' thickness ϵ vanishes due to Γ^c having zero volume, such that $\mathcal{E}^- = \emptyset$.

Consequently, the discontinuous element integral is fully captured by the sum of subdomain integrals in the limit. While this statement is only useful to motivate the validity of the approximation in the continuous setting, limitations in the discrete setting are inevitable. To the best of our knowledge, any scalar number representation has limited precision in practice, due to finite computational resources. Our implementation uses a 64-bit floating point representation, thus disallowing thickness values that approach machine epsilon. Fortunately, real-world thickness values for knives and scalpels range in the tenths of millimeters, a range of values that is easily represented by the floating point precision of our simulations. However, relative thickness of the cut geometry with respect to element size also needs to be considered. Geometrically small domains with high resolution are necessarily discretized by proportionally small elements. This forces a need to reduce the cut solids' thickness to maintain integral approximation error under an acceptable threshold.

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

A mesh Boolean difference operation produces an explicit representation of the boundary $\partial\Omega^{e,s}$ of the subdomain $\Omega^{e,s}$ (see Figure 4) in the form of triangle meshes. As such, integrals over closed subdomains can be computed trivially by using the divergence theorem:

$$\int_{\Omega^{e,s}} F'(\mathbf{X})\, d\Omega^{e,s} = \int_{\partial\Omega^{e,s}} F(x)\cdot \mathbf{n}\, dS. \tag{21}$$

The right-hand side of Eq. 21 can be computed by integrating $F(\mathbf{X})$ on each triangle of $\partial\Omega^{e,s}$, given that our integrands of interest are continuous everywhere in $\Omega^{e,s} \supseteq \partial\Omega^{e,s}$. Fortunately, in the case of linear shape functions, $F'(\mathbf{X})$ is an order two polynomial within $\Omega^{e,s}$ for mass matrix entries of Eq. 13, decreasing to order one in Eq. 14 for force vector entries, resulting in antiderivatives $F(\mathbf{X})$ of orders three and two, respectively, in Eq. 21. Integration of elastic energy is even easier, as hyper-elastic energies Ψ are piecewise constant per region $\Omega^{e,s}$, leading to $\int_{\Omega^{e,s}} \Psi(\mathbf{X}) d\Omega^{e,s} = \Psi(\mathbf{X}_g)|\Omega^{e,s}|$, where Ψ need only be evaluated at a single arbitrary point $\mathbf{X}_g \in \Omega^{e,s}$ and $|\Omega^{e,s}|$ is the volume of the element subdomain.

Integrating such polynomials accurately on triangular domains is a straightforward exercise by analytic integration using symbolic calculation, or precomputed polynomial quadrature rules, simplifying implementation in 3D. Our XFEM framework thus accurately approximates Eq. 20 by ignoring its vanishing rightmost term and strictly computing Eq. 21, yielding physically precise mass, force, and elastic energy using only the boundary description $\partial\Omega^{e,s}$ corresponding to element subdomain $\Omega^{e,s}$. We also point out that this is much simpler than other spatial integration approaches that are typically used in this setting, since they generally rely on either mesh refinement [KMB*09, MGS11a, MGS11b, RHS*11] or hierarchical polynomial quadrature construction [MKOW12, MKO13, KBT17]. Such a simplification is revealed by leveraging mesh Boolean technology to obtain explicit polygons on which exact computation of polynomial contour integrals required by the divergence theorem is trivial and cheap.

4.4. Algorithm

We provide pseudocode for a single time step of our cutting simulation in Algorithm 1. The remainder of this section is dedicated to providing additional practical implementation considerations about our method.

Mesh data structures. The implementation of our cutting algorithm uses three independent geometric data structures:

- *Tetrahedral mesh:* The mesh used for elastic simulation. Since we use an XFEM framework, it is never modified.
- *Per node triangle meshes:* Each node i of the input tetrahedral mesh maintains a list of triangle meshes describing the boundaries $\{\partial\Omega^{i,s}\}$ of its (sub)domains $\{\Omega^{i,s}\}$, i.e., the connected components resulting from repeated cutting of its original 1-ring, which are modified by mesh Boolean operations. Initially, a node stores only the 1-ring boundary triangulation of the tetrahedral mesh.
- *Per element triangle meshes:* Each element e of the input tetrahedral mesh similarly maintains a triangle mesh list, starting with its initial boundary and developing a list of connected component meshes as it is cut. Such triangle meshes $\{\partial\Omega^{e,s}\}$ represent element sub-domains $\{\Omega^{e,s}\}$ with continuous integrands, and they are also modified by mesh Boolean operations.

Algorithm 1 Per time-step algorithm of our cutting method.

1: **for** node $i \in V$ **do**
2: **for** subdomain s of node i **do**
3: $\Omega^{i,s}_{\text{old}}, \mathbf{u}^{i,s}_{\text{old}} \leftarrow \Omega^{i,s}, \mathbf{u}^{i,s}$
4: $\mathcal{B}^i \leftarrow$ connected components of $\Omega^{i,s}_{old} \setminus \Gamma$
5: **if** $|\mathcal{B}^i| > 1$ **then** ▷ Nodal subdomain cut
6: $\{\Omega^{i,s}\} \leftarrow \{\Omega^{i,s}\} \setminus \Omega^{i,s}_{\text{old}} \cup \mathcal{B}^i$ ▷ Update subdomains
7: $\mathbf{u}^{i,s} \leftarrow \mathbf{u}^{i,s}_{\text{old}}\ \forall\, \Omega^{i,s} \in \mathcal{B}^i$ ▷ Duplicate DOFs
8: **end if**
9: **end for**
10: **end for**
11: **for** element $e \in E$ **do**
12: **for** subdomain s of element e **do**
13: $\Omega^{e,s}_{\text{old}} \leftarrow \Omega^{e,s}$
14: $\mathcal{B}^e \leftarrow$ connected components of $\Omega^{e,s}_{\text{old}} \setminus \Gamma$
15: **if** $|\mathcal{B}^e| > 1$ **then** ▷ Element subdomain cut
16: $\{\Omega^{e,s}\} \leftarrow \{\Omega^{e,s}\} \setminus \Omega^{e,s}_{\text{old}} \cup \mathcal{B}^e$ ▷ Update subdomains
17: **end if**
18: **end for**
19: **end for**
20: **for** element $e \in E$ **do**
21: **if** e was enriched or e was cut **then**
22: recompute mass matrix $\mathbf{M}^e$ and forces $\mathbf{f}^e$ by Eq. 21
23: **end if**
24: **end for**
25: assemble global mass $\mathbf{M} \leftarrow \sum_{e\in E} \mathbf{M}^e$ and forces $\mathbf{f} \leftarrow \sum_{e\in E} \mathbf{f}^e$
26: $\mathbf{u}^{t+1} \leftarrow$ solve Eq. 10

Collision culling. Geometric overlap queries are used to avoid unnecessary mesh Boolean computations on nodal and element subdomains. The loops at lines 1-2 and 11-12 of Algorithm 1 are accelerated by spatial data structures for efficiently querying nodal and element subdomain intersections with cuts Γ. Specifically, a primary culling phase determines the set of elements intersected by a cut using a precomputed axis-aligned bounding box (AABB) hierarchy over tetrahedra, from which we derive the set of intersected nodal 1-rings. This first culling phase is carried out by overlap tests between triangles, tetrahedra and AABBs. A secondary culling phase may then be carried out using triangle-triangle overlap tests between the cutting mesh and subdomain meshes to determine which of the aforementioned elements and nodes are intersected by the cut. In our implementation, we find the first culling phase to be sufficient.

Connected components. We can efficiently determine the values of the predicates on lines 5 and 15 by computing the intersection curves between triangle meshes of the subdomains and the volumetric cut. If the intersection curves contain loops, then there must be multiple connected components resulting from the Boolean difference operation. The connected components are then computed by a traversal over the graph of adjacent triangles resulting from the mesh Boolean difference operation. The updates on lines 7 and 16 are then straightforward, requiring

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Table 1: *Cut thickness parameter prescribed for each mesh model.*

Mesh	ϵ (m)	Average edge length (m)
Accordion	1e−4	0.17
Cube	1e−4	1.14
Octopus	1e−4	2.39
Tetrahedron	1e−3	1.77
Thin	1e−4	0.14

only simple insertion and deletion operations of the per node and per element triangles meshes. Compared to element duplication methods [MBF04, WJST15, SDF07], our approach avoids the need to recompute material connectivity between connected components of adjacent mesh elements by instead independently modifying each node's original 1-ring.

Spatial integration. Line 22 of Algorithm 1 leverages the triangle mesh representation of our element subdomains to recompute the element integrals as described in Section 4.3. Only elements whose nodes have gained new enrichments or whose domain has been split will have different mass and force dimensions and/or values. We thus limit integral computations to those particular elements.

Degree of freedom duplication. A requirement for enriched degrees of freedom is that they preserve the same displacement field before and after a cut. In other words, introducing a discontinuity in the model should not instantaneously change its momentum or displacement. The traditional XFEM formulation is purely extrinsic, i.e., enrichment functions remain fixed once they have been created, such that their associated degrees of freedom are always initialized to zero. In our case, previous enrichment functions can be modified due to cuts intersecting. However, thanks to our simple indicator-based generalized enrichments, preserving the displacement field during cuts is achieved by simply duplicating the degrees of freedom associated with a cut subdomain to its resulting child subdomains (see line 7 of Algorithm 1). Mathematically, it is easy to see that the values of the nodal terms $\mathbf{u}_i^s(t)\phi_i(\mathbf{X})\psi_i^s(\mathbf{X})$ of Equation 18 are preserved by replacing them by a sum over newly enriched basis functions of child subdomains, all multiplied by the parent's previous coefficients $\mathbf{u}_i^s(t)$.

5. Results

In this section, we describe our results on various example cutting simulations that help distinguish our method from prior state of the art work. Specifically, we show that our approach allows multiple intersecting cuts in challenging scenarios, yet maintains stability and accuracy throughout. Our experiments were performed on an Intel i9-13900K CPU with 64 GB of memory using the Eigen [GJ*10] for linear algebra and libigl [JP*18] mesh Boolean implementation described in the work by Zhou et al. [ZGZJ16]. For simulation, we use the stable neo-Hookean model [SGK18], a mass density $\rho = 10^3$, Young's modulus 10^6, Poisson's ratio $\nu = 0.4$, time step $\Delta t = 0.033$ s and the solver uses 1 Newton iteration by default. Specific choices of cut thickness ϵ compared to model size are given in Table 1. Animations of the experiments in this section can also be found in the supplementary video.

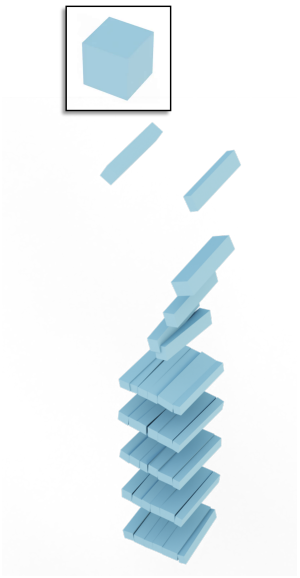

Figure 5: *A coarse 5-tetrahedron cube is cut into 36 sticks via 10 intersecting cuts.*

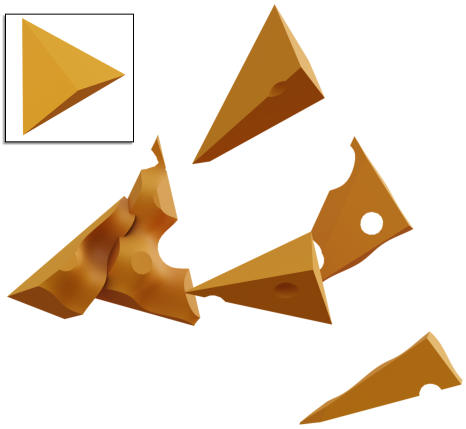

Figure 6: *A single tetrahedron is cut multiple times with varying curved and flat geometries, resulting in a Swiss cheese pattern.*

5.1. Examples

Thinly cut slab. Figure 1 showcases the accuracy of our spatial integration by cutting a slab into thin deformable sheets. In our experiment, the first thin sheet is cut into even thinner sheets, which reveals the expected separation between them, due to the correct mass and stiffness decreasing periodically at each slice, similar to the experiments presented in Koschier et al. [KBT17]. However, we go one step further and throw a final intersecting cut perpendicular to all previous sheets, which is a limitation of the previous work. This final cut once again reduces the mass and stiffness of the cut sheets, allowing them to rise closer to their rest configuration.

Diced cube. The experiment shown in Figure 5 aims to challenge our method's support for intersecting cuts. Specifically, we cut

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Figure 7: *An elastic hanging cube is cut horizontally multiple times, resulting in an accordion structure. The last cut rips apart the top layers.*

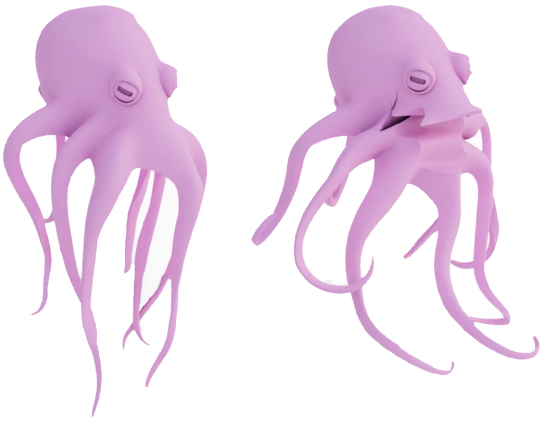

Figure 8: *An octopus happily smiles thanks to a sinusoidal cut carving out its mouth, and a second intersecting cut allowing its jaw to drop.*

a horizontally stretched coarse cube model composed of only 5 tetrahedra into 36 sticks by slicing it progressively with a grid of 5 cuts in the vertical direction that intersect with 5 cuts in the horizontal direction.

Swiss cheese tetrahedron. Figure 6 shows that our method also supports higher-order non-linear cutting geometries independent of the mesh resolution. In this example, a single tetrahedron is cut by intersecting wavy and flat cuts in different directions, which generate multiple enrichments per node at the same time step due to their complex intersection configurations. The result is a swiss cheese pattern being carved out of a single element.

Elastic accordion. Figure 7 shows an example where accurate stiffness properties are particularly highlighted. We partially cut a hanging cube a large number of times at different levels, reducing the total stiffness in the large object due to the appearing holes disconnecting material at the cut, thus allowing the cube to stretch down closer to the ground. Finally, we disconnect one part of this accordion by a vertical intersecting cut on one side, resulting in the cube accelerating significantly towards the ground.

Smiling octopus. The last example validates our algorithm on more complex shapes, such as the octopus in Figure 8, by first carving its mouth, and intersecting this cut with a second cut, producing a jaw drop which resembles a laughing behavior. This particular example uses a Young's modulus of 10^7.

5.2. Comparisons

We summarize our timing results compared to our implementation of Koschier et al. [KBT17]'s robust XFEM algorithm, which we refer to as RXFEM in Table 2. Our comparison focuses on statistics related to time steps in which cutting occurs, but also collects total simulation time. In addition, we encourage fairness by only measuring time in frames where no intersecting cut occurs, which their method does not support. Because the swiss cheese tetrahedron example only has a single non-intersecting cut during the simulation, we do not deem its comparison with RXFEM to be relevant.

The results show that the computational overhead due to mesh Boolean operations used by our approach is typically less than 1% on larger models such as the accordion and octopus, but is significantly larger on smaller models, where the locality of cut processing cannot be leveraged. In terms of aggregate computational cost, our approach is comparable to RXFEM, as highlighted by average time step duration and total simulation time considering both cut and uncut frames. However, we must note the fact that our approach exhibits non-negligible peaks in frames where cutting occurs, as reported by the maximum time step duration data. It is important to recognize that this local, yet high overhead is mostly tied to the use of exact adaptive floating point arithmetic in the mesh Boolean implementation we use [ZGZJ16]. More recent algorithms relying on alternative representations without sacrificing exactness could be leveraged to accelerate our method [CLSA20, CPAL22, TNWK22]. Fortunately, our method does not leak the cost of cut processing into subsequent frames in which cutting does not occur. Specifically, RXFEM

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Table 2: *Timing results comparing our method and RXFEM. Columns refer to average times by default and their abbreviations used in column headings are EI for recomputing an element's integrals, MB for mesh Booleans applied per element and node subdomain, $\partial\Psi$ for elastic energy derivatives, CG for the Newton-CG solve, and Δt avg for time step execution time. Special cases include # DOFs, i.e., total number of degrees of freedom, Δt max for maximum time step execution time, Total for total simulation time, where C refers to frames involving cutting, and C+U refers to both cut and uncut frames. Timings are in seconds unless otherwise indicated.*

	# DOFs	EI (ms)	MB (ms)	$\partial\Psi$ (ms)	CG	Δt max	Δt avg	Total (C)	Total (C+U)
Sticks (RXFEM)	144	2.48		5.6	0.001	0.032	0.011	0.097	1.664
Sticks (Ours)	144	0.36	26.9	0.0	0.001	0.278	0.009	0.898	1.307
Thin (RXFEM)	7.2k	1.98		84.8	0.306	1.789	0.6	61.5	270.0
Thin (Ours)	7.2k	0.289	9.59	3.875	0.293	3.355	0.627	123.3	282.4
Accordion (RXFEM)	22.8k	0.755		153.2	4.909	9.14	5.764	1036	3112
Accordion (Ours)	22.8k	0.061	6.437	8.796	4.811	9.203	5.785	1088	3124
Octopus (RXFEM)	20.8k	0.683		120.96	2.26	12.241	2.092	122.2	1255
Octopus (Ours)	20.8k	0.048	9.332	7.18	2.382	17.3	2.05	137.7	1230

incurs overhead due to their 24-point quadrature rule per element subdomain which they use to compute elastic energy and its derivatives in all time steps. Average time step timings show that our approach is generally slightly faster in frames where cutting does not occur, since our element subdomain elasticity computations require a single evaluation, seen in Table 2, where an order of magnitude speedup can be observed compared to RXFEM. Nevertheless, it is obvious that the linear solve required by time integration remains the main bottleneck.

Restricting our attention now to intersecting cut scenarios, one should also consider the nature of the simulated example as a predictor of overhead. For instance, our cube sticks, swiss cheese tetrahedron and thinly cut slab examples are subject to multiple cuts re-entering the same elements multiple times, which inevitably leads to an increase in mesh Boolean operation complexity. That being said, it is important to note that our work does not focus on computational efficiently.

6. Conclusion

In this paper, we present an innovative approach for cutting, based on the extended finite element method, that addresses shortcomings inherent in conventional techniques. Our method employs mesh Boolean operations to identify distinct components of cut element and nodal domains that are subsequently enriched using a generalized enrichment function. Our approach is demonstrated to be effective for simulating complex intersecting cutting configurations that were previously not possible.

6.1. Limitations and Future Work

The utilization of mesh Boolean operations in our prototype incurs some performance drawbacks. However, the off-the-shelf mesh Boolean implementation used by our prototype is not tailored for our specific application, and we believe that optimizing these routines would yield considerable performance enhancements. Furthermore, our current work does not handle cut surfaces introduced in world space, nor self collision detection and handling. In those scenarios, resolving the discontinuity's geometry robustly would consider the relative motion between the cut path and the elastic model's path in time using continuous collision detection.

Similar to previous works, our XFEM implementation is not robust to degenerate cuts, such as those aligning with mesh element facets (triangles, edges and vertices for tetrahedral meshes). In such occurrences, the consistency of one-ring subdomains with neighboring nodes may be compromised, and manifests as artificial "sticking" in the simulation. Unfortunately, the solidification of the cut path by a thickness parameter increases the risk of such degeneracies. In contrast, the cut thickness is robust to rounding issues when converting the mesh Boolean output of Zhou et al. [ZGZJ16] to floating point representation due to ϵ remaining far from the machine epsilon. We found that employing a strategy to gently perturb the cutting surface upon detecting such cases helps alleviate the issue, but is not foolproof.

Our approach is not limited to linear tetrahedral meshes. Extending it to higher order meshes should only require increasing the order of the divergence integral quadrature, and explicitly constructing the boundary of the supports of additional FEM nodes. Such nodes may lie on the interior of elements, in which case their support is the element itself, or on element facets, in which case their support is the union of their incident elements.

As future work, we intend to explore the use of non-linear local iterative solvers to enhance the performance of our simulation pipeline. These approaches show promise in efficiently managing structural changes prevalent in cutting and fracture. Additionally, investigating partial cutting of tetrahedral models is an intriguing avenue for research, although extending existing techniques [KMB*09] to 3D simulations is non-trivial.

Acknowledgements

This research was funded by the FRQNT Doctoral scholarship 349238. We are grateful to the Fonds de recherche du Québec for their resources. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) via the Discovery grant program and Alliance grant ALLRP-570702-21 with Symgery.

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

References

[BCXZ03] BELYTSCHKO T., CHEN H., XU J., ZI G.: Dynamic crack propagation based on loss of hyperbolicity and a new discontinuous enrichment. *International Journal for Numerical Methods in Engineering 58*, 12 (2003), 1873–1905. doi:https://doi.org/10.1002/nme.941. 3

[BG00] BIELSER D., GROSS M.: Interactive simulation of surgical cuts. In *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications* (2000), pp. 116–442. doi:10.1109/PCCGA.2000.883933. 2

[BGL94] BELYTSCHKO T., GU L., LU Y.: Fracture and crack growth by element free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering 2*, 3A (1994), 519. 3

[BGTG03] BIELSER D., GLARDON P., TESCHNER M., GROSS M.: A state machine for real-time cutting of tetrahedral meshes. In *11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings.* (2003), pp. 377–386. doi:10.1109/PCCGA.2003.1238279. 2

[BKO*96] BELYTSCHKO T., KRONGAUZ Y., ORGAN D., FLEMING M., KRYSL P.: Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering 139*, 1 (1996), 3–47. doi:https://doi.org/10.1016/S0045-7825(96)01078-X. 3

[BLG94] BELYTSCHKO T., LU Y. Y., GU L.: Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering 37*, 2 (1994), 229–256. doi:https://doi.org/10.1002/nme.1620370205. 3

[BMG99] BIELSER D., MAIWALD V. A., GROSS M. H.: Interactive cuts through 3-dimensional soft tissue. *Computer Graphics Forum 18*, 3 (1999), 31–38. doi:https://doi.org/10.1111/1467-8659.00325. 2

[BSM*02] BRUYNS C. D., SENGER S., MENON A., MONTGOMERY K., WILDERMUTH S., BOYLE R.: A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh cutting using virtual tools. *The Journal of Visualization and Computer Animation 13*, 1 (2002), 21–42. doi:https://doi.org/10.1002/vis.275. 2

[CDA00] COTIN S., DELINGETTE H., AYACHE N.: A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *The visual computer 16*, 8 (2000), 437–452. 2

[CLSA20] CHERCHI G., LIVESU M., SCATENI R., ATTENE M.: Fast and robust mesh arrangements using floating-point arithmetic. *ACM Trans. Graph. 39*, 6 (nov 2020). doi:10.1145/3414685.3417818. 9

[CMSK20] CHITALU F. M., MIAO Q., SUBR K., KOMURA T.: Displacement-correlated XFEM for simulating brittle fracture. *Computer Graphics Forum 39*, 2 (2020), 569–583. doi:https://doi.org/10.1111/cgf.13953. 3

[CPAL22] CHERCHI G., PELLACINI F., ATTENE M., LIVESU M.: Interactive and robust mesh booleans. *ACM Trans. Graph. 41*, 6 (nov 2022). doi:10.1145/3550454.3555460. 9

[DCA99] DELINGETTE H., COTIN S., AYACHE N.: A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *Proceedings of the Computer Animation* (USA, 1999), CA '99, IEEE Computer Society, p. 70. 2

[DG96] DESBRUN M., GASCUEL M.-P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation'96: Proceedings of the Eurographics Workshop in Poitiers, France, August 31–September 1, 1996* (1996), Springer, pp. 61–76. 3

[DGW11] DICK C., GEORGII J., WESTERMANN R.: A hexahedral multigrid approach for simulating cuts in deformable objects. *IEEE Transactions on Visualization and Computer Graphics 17*, 11 (2011), 1663–1675. doi:10.1109/TVCG.2010.268. 2

[DMD*00] DAUX C., MOËS N., DOLBOW J., SUKUMAR N., BELYTSCHKO T.: Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering 48*, 12 (2000), 1741–1760. 3, 5

[DSMB09] DUAN Q., SONG J.-H., MENOUILLARD T., BELYTSCHKO T.: Element-local level set method for three-dimensional dynamic crack growth. *International Journal for Numerical Methods in Engineering 80*, 12 (2009), 1520–1543. doi:https://doi.org/10.1002/nme.2665. 3

[FG99] FRISKEN-GIBSON S.: Using linked volumes to model object collisions, deformation, cutting, carving, and joining. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (1999), 333–348. doi:10.1109/2945.817350. 2

[GCMS00] GANOVELLI F., CIGNONI P., MONTANI C., SCOPIGNO R.: A multiresolution model for soft objects supporting interactive cuts and lacerations. *Computer Graphics Forum 19*, 3 (2000), 271–281. doi:https://doi.org/10.1111/1467-8659.00419. 2

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3, 2010. URL: http://eigen.tuxfamily.org. 8

[HFG*18] HU Y., FANG Y., GE Z., QU Z., ZHU Y., PRADHANA A., JIANG C.: A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph. 37*, 4 (jul 2018). doi:10.1145/3197517.3201293. 3

[HH04] HANSBO A., HANSBO P.: A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering 193*, 33 (2004), 3523–3540. doi:https://doi.org/10.1016/j.cma.2003.12.041. 3

[HSK*19] HAFNER C., SCHUMACHER C., KNOOP E., AUZINGER T., BICKEL B., BÄCHER M.: X-CAD: optimizing CAD models with extended finite elements. *ACM Trans. Graph. 38*, 6 (nov 2019). doi:10.1145/3355089.3356576. 3

[JBB*10] JEŘÁBKOVÁ L., BOUSQUET G., BARBIER S., FAURE F., ALLARD J.: Volumetric modeling and interactive cutting of deformable bodies. *Progress in biophysics and molecular biology 103*, 2-3 (2010), 217–224. 2

[JCD02] JI H., CHOPP D., DOLBOW J. E.: A hybrid extended finite element/level set method for modeling phase transformations. *International Journal for Numerical Methods in Engineering 54*, 8 (2002), 1209–1233. doi:https://doi.org/10.1002/nme.468. 3

[JK09] JEŘÁBKOVÁ L., KUHLEN T.: Stable cutting of deformable objects in virtual environments using XFEM. *IEEE Computer Graphics and Applications 29*, 2 (2009), 61–71. doi:10.1109/MCG.2009.32. 3

[JL12] JUNG H., LEE D. Y.: Real-time cutting simulation of meshless deformable object using dynamic bounding volume hierarchy. *Computer animation and virtual worlds 23*, 5 (2012), 489–501. 3

[JP*18] JACOBSON A., PANOZZO D., ET AL.: libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 8

[KBT17] KOSCHIER D., BENDER J., THUEREY N.: Robust extended finite elements for complex cutting of deformables. *ACM Trans. Graph. 36*, 4 (jul 2017). doi:10.1145/3072959.3073666. 2, 3, 5, 7, 8, 9

[KMB*09] KAUFMANN P., MARTIN S., BOTSCH M., GRINSPUN E., GROSS M.: Enrichment textures for detailed cutting of shells. *ACM Trans. Graph. 28*, 3 (jul 2009). doi:10.1145/1531326.1531356. 3, 6, 7, 10

[LLKC21] LI J., LIU T., KAVAN L., CHEN B.: Interactive cutting and tearing in projective dynamics with progressive cholesky updates. *ACM Trans. Graph. 40*, 6 (dec 2021). doi:10.1145/3478513.3480505. 3

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

[MBF04] MOLINO N., BAO Z., FEDKIW R.: A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph. 23*, 3 (aug 2004), 385–392. doi:10.1145/1015706.1015734. 2, 8

[MDB99] MOËS N., DOLBOW J., BELYTSCHKO T.: A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering 46*, 1 (1999), 131–150. 3

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proceedings of Graphics Interface 2004* (2004), GI 2004, pp. 239–246. 2

[MGS11a] MOUSAVI S. E., GRINSPUN E., SUKUMAR N.: Harmonic enrichment functions: A unified treatment of multiple, intersecting and branched cracks in the extended finite element method. *International Journal for Numerical Methods in Engineering 85*, 10 (2011), 1306–1322. doi:https://doi.org/10.1002/nme.3020. 3, 5, 6, 7

[MGS11b] MOUSAVI S. E., GRINSPUN E., SUKUMAR N.: Higher-order extended finite elements with harmonic enrichment functions for complex crack problems. *International Journal for Numerical Methods in Engineering 86*, 4-5 (2011), 560–574. doi:https://doi.org/10.1002/nme.3098. 3, 7

[MK00] MOR A. B., KANADE T.: Modifying soft tissue models: Progressive cutting with minimal new element creation. In *Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention* (Berlin, Heidelberg, 2000), MICCAI '00, Springer-Verlag, p. 598–607. 2

[MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 141–151. 3

[MKO13] MÜLLER B., KUMMER F., OBERLACK M.: Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering 96*, 8 (2013), 512–528. doi:https://doi.org/10.1002/nme.4569. 3, 7

[MKOW12] MÜLLER B., KUMMER F., OBERLACK M., WANG Y.: Simple multidimensional integration of discontinuous functions with application to level set methods. *International Journal for Numerical Methods in Engineering 92*, 7 (2012), 637–651. doi:https://doi.org/10.1002/nme.4353. 3, 7

[MMDH*23] MONTES MAESTRE J. S., DU Y., HINCHET R., COROS S., THOMASZEWSKI B.: Differentiable stripe patterns for inverse design of structured surfaces. *ACM Trans. Graph. 42*, 4 (jul 2023). doi:10.1145/3592114. 3

[MO20] MAGNOUX V., OZELL B.: Real-time visual and physical cutting of a meshless model deformed on a background grid. *Computer Animation and Virtual Worlds 31*, 6 (2020), e1929. 3

[NFvdS01] NIENHUYS H.-W., FRANK VAN DER STAPPEN A.: A surgery simulation supporting cuts and finite element deformation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001* (2001), pp. 145–152. 2

[NRBD08] NGUYEN V. P., RABCZUK T., BORDAS S., DUFLOT M.: Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation 79*, 3 (2008), 763–813. doi:https://doi.org/10.1016/j.matcom.2008.01.003. 3

[NvdS00] NIENHUYS H.-W., VAN DER STAPPEN A. F.: Combining finite element deformation with cutting for surgery simulations. In *Eurographics 2000 - Short Presentations* (2000), Eurographics Association. doi:10.2312/egs.20001017. 2

[OFTB96] ORGAN D., FLEMING M., TERRY T., BELYTSCHKO T.: Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational mechanics 18* (1996), 225–235. 3

[PUC*15] PAULUS C. J., UNTEREINER L., COURTECUISSE H., COTIN S., CAZIER D.: Virtual cutting of deformable objects based on efficient topological operations. *Vis. Comput. 31*, 6–8 (jun 2015), 831–841. doi:10.1007/s00371-015-1123-x. 2

[RHS*11] RICHARDSON C. L., HEGEMANN J., SIFAKIS E., HELLRUNG J., TERAN J. M.: An XFEM method for modeling geometrically elaborate crack propagation in brittle materials. *International Journal for Numerical Methods in Engineering 88*, 10 (2011), 1042–1065. doi:https://doi.org/10.1002/nme.3211. 3, 7

[SAB06] SONG J.-H., AREIAS P. M. A., BELYTSCHKO T.: A method for dynamic crack and shear band propagation with phantom nodes. *International Journal for Numerical Methods in Engineering 67*, 6 (2006), 868–893. doi:https://doi.org/10.1002/nme.1652. 3

[SDF07] SIFAKIS E., DER K. G., FEDKIW R.: Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, p. 73–80. 2, 8

[SGK18] SMITH B., GOES F. D., KIM T.: Stable neo-Hookean flesh simulation. *ACM Trans. Graph. 37*, 2 (mar 2018). doi:10.1145/3180491. 8

[She02] SHEWCHUK J.: What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures. *Proceedings of the 11th International Meshing Roundtable 73* (09 2002). 3

[SHGS06] STEINEMANN D., HARDERS M., GROSS M., SZEKELY G.: Hybrid cutting of deformable solids. In *Proceedings of the IEEE Conference on Virtual Reality* (USA, 2006), VR '06, IEEE Computer Society, p. 35–42. doi:10.1109/VR.2006.74. 2

[SOG06] STEINEMANN D., OTADUY M. A., GROSS M.: Fast arbitrary splitting of deforming objects. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), pp. 63–72. 3

[SOG08] STEINEMANN D., OTADUY M. A., GROSS M.: Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/eurographics symposium on computer animation* (2008), pp. 87–94. 3

[SOG09] STEINEMANN D., OTADUY M. A., GROSS M.: Splitting meshless deforming objects with explicit surface tracking. *Graphical Models 71*, 6 (2009), 209–220. 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2006). doi:https://doi.org/10.1016/j.gmod.2008.12.004. 3

[SSSH11] SEILER M., STEINEMANN D., SPILLMANN J., HARDERS M.: Robust interactive cutting based on an adaptive octree simulation mesh. *The Visual Computer 27* (2011), 519–529. 2

[TNWK22] TRETTNER P., NEHRING-WIRXEL J., KOBBELT L.: EMBER: exact mesh Booleans via efficient & robust local arrangements. *ACM Trans. Graph. 41*, 4 (2022). doi:10.1145/3528223.3530181. 9

[WDW11] WU J., DICK C., WESTERMANN R.: Interactive High-Resolution Boundary Surfaces for Deformable Bodies with Changing Topology. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2011)* (2011). doi:10.2312/PE/vriphys/vriphys11/029-038. 2

[WJST15] WANG Y., JIANG C., SCHROEDER C., TERAN J.: An adaptive virtual node algorithm with robust mesh cutting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2015), SCA '14, p. 77–85. 2, 8

[WM18] WANG M., MA Y.: A review of virtual cutting methods and technology in deformable objects. *The International Journal of Medical Robotics and Computer Assisted Surgery 14*, 5 (2018), e1923. doi:https://doi.org/10.1002/rcs.1923. 2

[WWD15] WU J., WESTERMANN R., DICK C.: A survey of physically based simulation of cuts in deformable bodies. *Computer Graphics Forum 34*, 6 (2015), 161–187. doi:https://doi.org/10.1111/cgf.12528. 2

[YCP16] YEUNG Y.-H., CROUCH J., POTHEN A.: Interactively cutting and constraining vertices in meshes using augmented matrices. *ACM Trans. Graph. 35*, 2 (feb 2016). doi:10.1145/2856317. 3

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

[YPC20] YEUNG Y.-H., POTHEN A., CROUCH J.: AMPS: Real-time mesh cutting with augmented matrices for surgical simulations. *Numerical Linear Algebra with Applications 27*, 6 (2020), e2323. doi:https://doi.org/10.1002/nla.2323. 3

[ZGZJ16] ZHOU Q., GRINSPUN E., ZORIN D., JACOBSON A.: Mesh arrangements for solid geometry. *ACM Trans. Graph. 35*, 4 (jul 2016). doi:10.1145/2897824.2925901. 2, 5, 8, 9, 10

[ZKBT17] ZEHNDER J., KNOOP E., BÄCHER M., THOMASZEWSKI B.: Metasilicone: design and fabrication of composite silicone with desired mechanical properties. *ACM Trans. Graph. 36*, 6 (nov 2017). doi:10.1145/3130800.3130881. 3

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.