

Simulating Fracture in Anisotropic Materials Containing Impurities

Avirup Mandal
avirupmandal@ee.iitb.ac.in
IIT Bombay
India

Parag Chaudhuri
paragc@cse.iitb.ac.in
IIT Bombay
India

Subhasis Chaudhuri
sc@ee.iitb.ac.in
IIT Bombay
India



Figure 1: A piece of steak breaks on pulling, according to anisotropic muscle fibers in the meat, to reveal fine-scale geometry around the fracture (top row). A pizza slice is separated from the whole, along the cuts made by an artist, and leaves stretchy bits of cheese hanging (bottom row). This is an example of artist-controlled fracture design using our method.

ABSTRACT

Fracture simulation of real-world materials is an exceptionally challenging problem due to complex material properties like anisotropic elasticity and the presence of material impurities. We present a graph-based finite element method to simulate dynamic fracture in anisotropic materials. We further enhance this model by developing a novel probabilistic damage mechanics for modelling materials with impurities using a random graph-based formulation. We demonstrate how this formulation can be used by artists for directing and controlling fracture. We simulate and render fractures for a diverse set of materials to demonstrate the potency and robustness of our methods.

CCS CONCEPTS

- Computing methodologies → Physical simulation; • Theory of computation → Random network models.

KEYWORDS

fracture, anisotropy, material impurities, graph-based FEM, random graphs

ACM Reference Format:

Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2022. Simulating Fracture in Anisotropic Materials Containing Impurities. In *ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '22), November 3–5, 2022, Guanajuato, Mexico*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3561975.3562956>

1 INTRODUCTION

We often gaze obliviously at the wide array of material fractures occurring around us in the real world. However, if we take a closer look at them, we will notice that not all the materials fracture the same way. Many real-world materials have anisotropic fibers along certain directions; this makes them stiffer in these preferred directions and softer in others. This imbalance of stiffness influences crack propagation during fracture. In this work, we present a method to simulate dynamic fracture in anisotropic materials using a graph-based finite element method.

Moreover, the state or condition of the material affects the fracture as well. While some materials remain fresh, others get corrupted over time due to environmental and human interventions, which in turn impose uncertainties on the fracture criteria. Purely deterministic approaches with pre-defined fracture thresholds cannot model such phenomena. We propose a novel probabilistic damage mechanics model to model changes in material properties due to the presence of impurities.

Fracture is a prohibitively complex phenomenon to model. Including anisotropic elasticity and a probabilistic approach of fracture further increases this complexity manyfold. We enhance a



Figure 2: A meat-filled loaf is torn apart. The outer bread is an isotropic material while the meat inside contains anisotropic fibers.

graph-based FEM [Mandal et al. 2021a] [Khodabakhshi et al. 2016] method for simulating fracture by adding anisotropy to it.

We employ random graph algorithms to implement a novel probabilistic damage mechanics that utilize the graph structure of the simulation object mesh in graph-based FEM to model fracture in the presence of material impurities. Even when simulation strategies are available for fracture, getting a specific fracture pattern, as desired by an artist, to appear during simulation can require hours of manual parameter tuning. We present a method for control of fracture that allows an artist to direct the fracture by designing impurity maps on the objects. These maps govern the distribution of impurities in the material of the object. In conjunction with our method for simulating fracture in anisotropic materials with impurities, the impurity maps can guide the cracks formed during simulation in a manner such that the fracture pattern closely resembles the impurity map.

The specific contributions of this work are:

- A remeshing-free method to simulate fracture in anisotropic materials using graph-based FEM.
- A novel, random graph-based, probabilistic damage mechanics to simulate fracture in the presence of material impurities.
- We prove using a Markov random process abstraction that even after the introduction of impurities, no sharp discontinuities or spurious divergences arise in the value of various node parameters of the FEM simulation mesh.
- A method for artist-controlled design of fracture, in the presence of anisotropy and material impurities.

The rest of the paper is organized as follows. We start by presenting a discussion of related work in Section 2. In Section 3 we describe how we incorporate anisotropy into graph-based FEM. Section 4 and 5 present our model for probabilistic damage mechanics, detailing its formulation as a random graph. Section 6 describes our framework for artist controlled fracture design. Next, we present the results of fracture simulations created using our method for various kinds of materials in Section 7. Finally, we conclude our paper by discussing some future directions for our work.

2 RELATED WORK

We start with an overview of anisotropic elastic energies for deformable solids. Next, we discuss the existing methods for mesh-based and meshless fracture simulation. Subsequently, we delve

into the graph-based finite element method for fracture simulation. Finally, we end this section by presenting a short review of random graphs.

Finite Element Method is one of the most popular approaches to model object deformation. Müller and Gross [2004] proposed a co-rotational model with Cauchy linear strain for deformable volumetric mesh simulation. Their model can handle artifacts produced due to large deformation and rotation. Large plastic flow is rendered in works by Bargteil et al. [2007], Irving et al. [2004] and Stomakhin et al. [2012]. Smith et al. [2018] modelled extreme deformation of flesh using Neo-Hookean elasticity model. Using a set of lower order invariants the authors later extended their work to include a larger class of isotropic [Smith et al. 2019] elasticity models. Xu et al. [2015] proposed a novel algorithm to model anisotropy using principle stretches of hyper-elastic strain energy density. Kim et al. [2019] developed a new anisotropic elastic density which is inversion-free and contains no singularity.

Fracture simulation in computer graphics was first introduced by Terzopoulos and Fleischer [1988] on visco-elastic fracture simulation. Early fracture algorithms used to model brittle fracture using mass-spring dynamics [Hirota et al. 2000] [Aoki et al. 2004]. Early work on FEM-based fracture goes back to the seminal work by O'Brien and Hodges [1999]. In their work, the authors present a nodal stress-based analysis for brittle fracture that was later extended to ductile fracture [O'Brien et al. 2002]. Müller and Gross [2004] Bao et al. [2007] also developed algorithms for simulating brittle as well as ductile fracture. These FEM-based fracture techniques remesh the original mesh periodically to incorporate the crack opening. Due to multiple remeshing, these methods suffer from multiple drawbacks like strain accumulation near crack tips, degenerate element generation and forming ill-conditioned basis matrices which subsequently lead to severe stability issues. Several techniques are proposed to alleviate these problems. Some of these methods include local mesh refinement at run-time to repair degenerate tetrahedra [Wicke et al. 2010], remeshing depending on gradient flow for improved fracture resolution [Chen et al. 2014], adaptive subdivision schemes for tetrahedral [Koschier et al. 2015] and triangular meshes [Pfaff et al. 2014] along with stress relaxation around fractured areas. Pfaff et al. [2014] also presents a method to propagate the fracture along the guided paths. However, their method requires extensive remeshing which is time-consuming.

Virtual Node Algorithm (VNA) [Molino et al. 2004] simulates fracture without any remeshing by duplicating the damaged elements depending on the yield threshold. Extra degrees of freedom are then added to these elements to incorporate partial or full crack openings. Later, VNA-based methods are improved to facilitate cutting at lower than mesh resolution [Sifakis et al. 2007]. Like VNA, eXtended Finite Element Method (XFEM) also adds extra degrees of freedom to the nodes of a damaged tetrahedron using appropriate enrichment functions. Koschier et al. [2017] proposed an XFEM-based algorithm to simulate pre-defined cuts in a 3D mesh. Chitalu et al. [2020] extended the pre-defined cuts to dynamic brittle fracture generation using a novel displacement-correlated algorithm on top of XFEM. However, XFEM has several limitations like ambiguities of crack-tip enrichment in 3D, instabilities induced by large volume ratios in Heaviside enrichments and difficulties in handling branch enrichments in 3D.

Another less explored avenue to simulate fracture is the Boundary Element Method which simulates cracks on the outer surface mesh as opposed to the volume elements. BEM was introduced to graphics by James and Pai [1999] for simulating elastostatic deformable objects. Hahn and Wojtan [2015] and Zhu et al. [2015] used BEM with stress intensity factors (SIFs) along crack fronts to successfully simulate brittle fracture. To alleviate high computational cost for solving singular integrals Hahn and Wojtan [2016] extended the original work to propose an algorithm for fast approximation of the singular integrals. Currently, to the best of our knowledge, there exists no work on BEM ductile fracture in graphics. Hahn and Wojtan [2015] varies the toughness throughout the material (in a single direction) with a cosine interpolation function that allows to generate granular fracture patterns. However, their method is deterministic and does not capture the randomness of fracture. In our method, we can vary the toughness of the material as per any random fracture patterns drawn by the artist.

Peridynamics directly solves the integral equations of continuum mechanics. Thus unlike FEM or Material Point Method (MPM), peridynamics enjoys the advantage of not imposing necessary boundary conditions for simulating fracture. Levine et al. [2015] approximate the behaviour of peridynamics using mass-spring systems to simulate brittle fracture. Chen et al. [2018] presented a peridynamics-based fracture algorithm for simulating brittle as well as ductile fracture.

The potential of Material Point Method to handle extreme topological change was introduced to the graphics community by Stomakhin et al. [2013]. Compatible Particle-in-Cell (CPIC) algorithm proposed by Hu et al. [2018a], allows the simulation of sharp discontinuities inside a material. Using CPIC the authors are able to generate dynamic cuts on an object mesh. Wolper et al. [2019] introduced MPM-based Continuum Damage Mechanics (CDM) with a variational energy-based formulation for crack evolution and fracture generation. CD-MPM was able to render dynamic fracture simulation for isotropic materials. Further work [Wolper et al. 2020] explored non-local CDM with anisotropic damage mechanics to simulate anisotropic fracture for ductile materials. In a recent

work [Fan et al. 2022] MPM has been used to simulate brittle fracture. Major limitations of MPM include difficulty in enforcing essential boundary conditions, the disappearance of intricate crack patterns due to excessive smoothing, and high computational cost.

Reddy and Srinivasa [2015] proposed a finite element analysis to show that for any hyper-elastic material, the magnitude of the nodal forces can be decomposed completely along the edges composing the element. Further, the directions of force coincide with the unit vectors corresponding to the edges. Building upon this idea, Khodabakhshi et al. [2016] introduces a novel fracture simulation method called Graph-based Finite Element Analysis (Gra-FEA). The dependency of Gra-FEA on the underlying mesh resolution is studied in another work [Khodabakhshi et al. 2019]. However, these Gra-FEA methods only focus on 2D materials consisting of triangular elements with linear strain energy density. In recent works, Mandal et al. [2021a] Mandal et al. [2021b] extend the idea to a dynamic visual simulation of both ductile and brittle fracture in 3D with linear as well as non-linear strain energy density.

During the last two decades, the random graph models have been used extensively for the studies of evolving communication networks. Starting from classic Erdős-Rényi random graph [Gilbert 1959] which randomly joins edges between two nodes, various random graph algorithms [Barabási and Albert 1999] [Davis et al. 2021] are proposed as models in a variety of problems like the dynamic topology of the world wide web to the transmission flow of the COVID-19 pandemic. Works by [Pittel 2010] and [Janson and Warnke 2021] give a rigorous mathematical analysis of the properties of the evolution of dynamic multi-graph networks and preferential attachment with them. We use a random graph-based formulation to represent probabilistic damage mechanics in a material with impurities.

3 ANISOTROPY IN GRAPH-BASED FEM

We briefly explain the graph-based FEM for fracture simulation. Then we delve into the details of how we incorporate anisotropic elasticity in it.

3.1 Graph-based FEM for Fracture

In graph-based FEM, given an object represented as a volumetric mesh with tetrahedral elements [Mandal et al. 2021a], the graph structure induced by the edges of the finite elements between the vertices is exploited for FEM computations. It can be proved [Reddy and Srinivasa 2015] that for any tetrahedral element, Δ_e , with hyper-elastic strain energy density, Ψ^e , the nodal internal elastic forces can be resolved along the edges of Δ_e .

$$\mathbf{f}_{e_i}^{int} = -V_e \frac{\partial \Psi^e}{\partial \mathbf{r}_i^e} = -2V_e \sum_{\substack{j=1 \\ j \neq i}}^{n_e} \frac{\partial \Psi^e}{\partial (\mathbf{d}_{ij}^e)^2} d_{ij}^e \hat{\mathbf{d}}_{ij}^e \quad (1)$$

where $\mathbf{f}_{e_i}^{int}$ denotes internal elastic force at the i^{th} node, the parameter $d_{ij}^e = ||\mathbf{r}_i^e - \mathbf{r}_j^e||$ is the distance between the i^{th} and j^{th} node of Δ_e and $\hat{\mathbf{d}}_{ij}^e$ is unit vector along d_{ij}^e . Moreover, V_e and n_e represent volume and the number of nodes of Δ_e respectively.

In order to simulate fracture in graph-based FEM, the rectangular Cartesian components of the Piola-Kirchhoff stress tensor are projected along the direction of the edges to calculate the normal stress.



Figure 3: A bar made of pure gold is stretched (Top row). A bar made of gold-copper alloy is stretched (Bottom row). The impure gold i.e. gold-copper alloy is much more ductile and deforms more before fracturing.

When normal stress along an edge exceeds a critical threshold, it is labelled as fractured and normal stress along that edge is set to zero. Finally, the original internal energy density of the fractured element, Ψ_{ori}^e , is then reformulated as

$$\Psi_{frac}^e = \left[\frac{|\sigma_{12}^{e'}| + |\sigma_{13}^{e'}| + |\sigma_{14}^{e'}| + |\sigma_{23}^{e'}| + |\sigma_{24}^{e'}| + |\sigma_{34}^{e'}|}{|\sigma_{12}^e| + |\sigma_{13}^e| + |\sigma_{14}^e| + |\sigma_{23}^e| + |\sigma_{24}^e| + |\sigma_{34}^e|} \right] \Psi_{ori}^e \quad (2)$$

where $\{\sigma_{12}^e \dots \sigma_{34}^e\}$ and $\{\sigma_{12}^{e'} \dots \sigma_{34}^{e'}\}$ are the components of normal stress tensor along the direction of the edges before and after fracture respectively. This reformulation of internal hyper-elastic strain energy density due to fracture helps the fractured pieces move independently without any explicit remeshing. For more details regarding the fracture simulation with graph-based FEM, interested readers may take a look at the work by Khodabakhshi et al. [2016] Mandal et al. [2021a]. We examined two approaches for the reformulation of strain energy. The first approach is uniformly weakening as Equation 2. The other approach is projecting out stresses along the fractured edges and just weakening the fractured edge. However, we noticed that for a high-resolution mesh the results are very similar. We added this comparison study in the supplement.

3.2 Anisotropic Elasticity

Isotropic materials compress or stretch equally in all directions irrespective of the direction of applied force. But many real-world materials have bundles of fibers in some preferred directions which makes them stiffer in those directions and softer in others. When stretched/compressed, these materials will resist more in the stiffer directions compared to softer directions, before damage sets in. Accordingly, this makes the fracture of anisotropic materials significantly different from the isotropic ones.

In order to model anisotropic materials, we add an anisotropic hyper-elastic strain energy density along with the regular isotropic strain energy density.

$$\Psi_{tot}^e = \Psi_{aniso}^e + \Psi_{iso}^e \quad (3)$$

where Ψ_{tot}^e , Ψ_{iso}^e and Ψ_{aniso}^e denote total, isotropic and anisotropic strain energy density respectively.

To model anisotropic strain energy density, we define two anisotropic invariants [Weiss et al. 1996]

$$\begin{aligned} IV_C &= \mathbf{a}^T \mathbf{C} \mathbf{b} \\ V_C &= \mathbf{a}^T \mathbf{C}^T \mathbf{C} \mathbf{b} \end{aligned} \quad (4)$$

where \mathbf{a}, \mathbf{b} are constant anisotropic fiber directions and they may or may not be equal to each other. However, as argued by Kim et al. [2019] Kim and Eberle [2020], there is a major disadvantage of using

these invariants to formulate anisotropic strain energy. As these invariants are based on the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F}$, it squares away the inversion information of an element stored in the deformation gradient \mathbf{F} . Inspired by lower-order, sign-preserving isotropic invariants proposed by Smith et al. [2019], Kim et al. [2019] propose a sign-preserving anisotropic invariant

$$I_4 = \mathbf{a}^T \mathbf{S} \mathbf{a} \quad (5)$$

where the stretch matrix \mathbf{S} is obtained from the polar decomposition of deformation gradient $\mathbf{F} = \mathbf{R}\mathbf{S}$. We use this inversion-safe, anisotropic strain energy density as follows.

$$\Psi_{AA}^e = \frac{\mu}{2} \left(\sqrt{I_5} - \Pi(I_4) \right)^2 \quad (6)$$

where $I_5 = \mathbf{a}^T \mathbf{C} \mathbf{a}$ and Π is the signum function. For a detailed discussion on how this energy density satisfies the condition of inversion-safety and stability, please refer [Kim et al. 2019]. As an example of fracture simulation after incorporating anisotropy in graph-based FEM, see the top row of Figure 1. Here we illustrate the fracture of a piece of steak which tears along the anisotropic fiber direction. Also, notice the thread-like stretchy fibers that get revealed when the meat is torn apart.

3.3 Theoretical Considerations

Graph-based FEM crucially depends on the fact that all involved strain energies can be represented as a function of edge lengths of the simulation mesh. This has to be true for anisotropic strain energy density Ψ_{AA}^e , for the above-described formulation to work. We prove that this is indeed so, in the enclosed supplemental document.

The theorems, proved in the supplemental material, together give a geometric interpretation of any hyper-elastic strain energy density in terms of the edge length for an undamaged mesh.

We also present a proof sketch in supplemental material to show that the deformation of the fractured/damaged mesh can then be interpreted as another smooth map from one Riemannian manifold to another. We believe this interpretation provides a convincing argument in support of the representative capacity of graph-based FEM to model fracture/damage.

4 PROBABILISTIC DAMAGE MECHANICS

So far, we have been simulating materials that are (theoretically) 100% pure. But in the real world, it is impossible to get a material without some impurity. Moreover, impurities are distributed over the material randomly. Incorporating impurities in a mathematically consistent way poses a huge challenge. Producing the desired effect of these impurities on the fracture simulation is even

more challenging as the random distribution of impurities imposes probabilistic threshold criteria on fracture.

In graph-based FEM, cracks due to fracture are generated depending on the stress value along the direction of an edge contained in an underlying graph structure. We find this graph structure is well suited to use a random graph [Pittel 2010] [Janson and Warnke 2021] strategy to incorporate probabilistic threshold criteria.

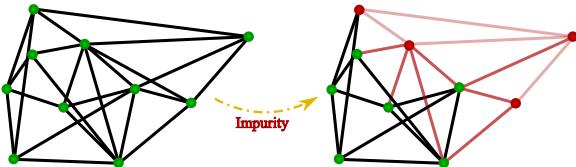


Figure 4: Impurities are added to a few nodes (shown in red) of the graph. Edges in light red, between red nodes with impurity, have the lowest threshold for fracture. Edges in darker red have only one node with impurity and thus have a higher threshold for fracture. Black edges between nodes with no impurity have the highest threshold for fracture.

Let us define a random graph growth process $\{G(n_v, \mu)\}$ on a vertex set $[n_v]$ with μ successive steps. Let $G(n_v, 0)$ be an empty graph. Recursively, given a graph $G(n_v, \mu)$ of vertex degree $\mathbf{d} = \{d_1, d_2, \dots, d_n\}$, a new graph $G(n_v, \mu + 1)$ is obtained by inserting a set of new edges between any two still disjoint random nodes i and j with $i \neq j$. The probability that an edge e will join two currently disjoint vertices, $i \neq j$, is proportional to $(d_i + \alpha)(d_j + \alpha)$. Here d_i, d_j are the degrees of i, j in $G(n_v, \mu)$ and $\alpha > 0$ is a user-defined constant. The degree of any node refers to the number of other nodes it is connected to. The sequence $\{G(n_v, \mu)\}$ is a Markov process.

A multigraph process is defined as a sequence $\{MG(n_v, \mu)\}$ of multigraphs where self-loops and multiple edges on a single vertex are allowed. Given a current multigraph, $MG(n_v, \mu)$, with degree \mathbf{d} on the vertex set n_v , a new edge joins two vertices, i and j , with probability proportional to $2(d_i + \alpha)(d_j + \alpha)$ and forms a loop $i \rightarrow i$ with probability proportional to $(d_i + \alpha)(d_j + \alpha + 1)$. Note that in multigraph while $i \neq j$, they may necessarily not be disjoint. This random process forms a Markov chain whose state space is the set of all multigraphs on the vertex set n_v .

In our context, we start from an intermediate multigraph, $MG(n_v, \mu)$, with degree \mathbf{d} on a vertex set of n_v nodes. Let us assume that the structure of the intermediate multigraph which contains n_v nodes and n_e edges, coincides with the initial undamaged graph generated by our FEM simulation mesh model. Then graph-based FEM for fracture simulation can be conceived as removing edges from a graph, contrary to the addition of edges as explained above. More specifically, the removed edges in $MG(n_v, \mu)$ denote those fractured edges of the FEM mesh on which the internal force and normal stress are zero. However, it still remains a Markov chain whose state space is the set of multigraphs. Given any multigraph, $MG(n_v, \mu)$ with degree \mathbf{d} , we define the probability to remove an edge between

vertices i and j as

$$p_{ij} = \begin{cases} 1 - h \left[2 \left(d_i + \frac{1}{|\alpha_i|} \right) \left(d_j + \frac{1}{|\alpha_j|} \right) \right] & \text{Material degradation} \\ 1 - h \left[2 (d_i + |\alpha_i|) (d_j + |\alpha_j|) \right] & \text{Material upgradation} \end{cases} \quad (7)$$

where α_i and α_j are random but constant initial values associated with node i and j . Function h maps the random numbers from sample space to probability space. Note that we have not included the probability of self-loop in Equation 7 as it does not occur in a consistent FEM mesh. To incorporate the impurities of any material, we assign Additive White Gaussian Noise (AWGN) to the nodes of the object mesh at the start of the simulation.

$$\alpha_s \in \mathcal{N}(0, \sigma^2), \quad \alpha_s \neq 0 \quad \forall s \in n_v \quad (8)$$

Higher values of α_k (for any node k) denote a higher amount of impurities and vice versa. We consider two different kinds of effects on material after adding impurities,

- **Material degradation:** This is the case when material strength decreases after adding impurities e.g., adding mud to asphalt/tar.
- **Material upgradation:** This is the case when material strength increases after adding impurities e.g., adding copper to pure gold.

Function h can be any strictly increasing function with range $[0, 1]$. One such function is

$$h(x) = \frac{e^x - 1}{e^x + 1} \quad \forall x \in (0, \infty) \quad (9)$$

The choice of function depends on the user and is not unique. In case of material degradation, as evident from Equation 7, if the amount of impurities increases, higher values of α_k makes the function value goes down. This, in turn, makes the fracture probability of the corresponding edge go up. A similar line of argument can be applied in the case of material upgradation but in the other way round. To summarize, in our method edges of the volumetric mesh have explicit yield strengths that vary based on the probabilistic approach. As shown in Figure 4, when impurities are added to the nodes (shown in red), the strength of the edge between them goes down/up. The change in the strength of the edge depends on the impurity content of the corresponding nodes. Moreover, if the degree/connectivity of a node decreases, i.e., the neighbourhood of the node gets damaged, then the probability of that node getting fractured increases. This closely resembles the real-world phenomena where it is much easier to break a partially damaged material compared to an intact one.

5 MARKOV RANDOM FIELD BASED ANALYSIS OF IMPURITY ADDITION

Even in the presence of impurities, a material can be assumed to be locally homogeneous. In other words, we can say that the distribution of any material property of an object is a spatial Markov Random Field (MRF), i.e., material property for any point in the object, depends on the properties of other points only in its local neighbourhood. It is important to examine, after adding AWGN for impurity, whether the posterior distribution preserves this locality

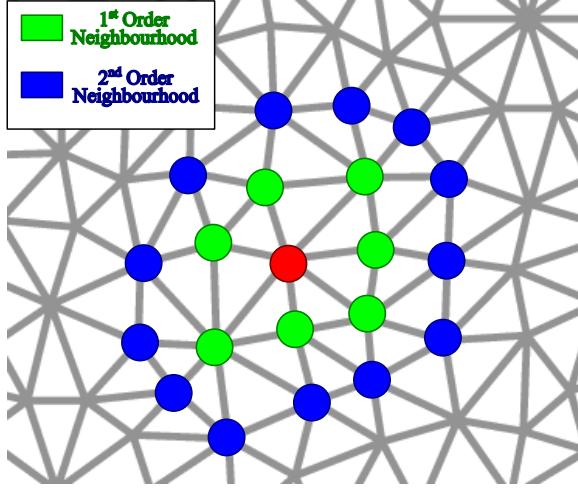


Figure 5: Neighbourhood structure of a node. Green and blue points refer to the 1st-order and 2nd-order neighbourhood to the original red coloured node respectively.

property. Without this locality property, the object may show undesired kinks or spurious divergence due to uneven acceleration and velocity caused by the addition of impurities.

Let us first consider the neighbourhood structure of a node in an object. As shown in Figure 5, for any given node i , its 1st-order neighbourhood consists of the nodes which are directly connected to it. The 2nd-order neighbourhood denotes the nodes that are connected to the 1st-order nodes. Similarly, we can define higher-order neighbourhoods. We assume that any material property associated with a node will only depend on its 1st-order neighbourhood.

Now, let us assume that $X = \{x_s\}_{s \in n_v}$ denotes a *Markov process* where x_s represents any node variable e.g., position, velocity, acceleration etc., normalized to $[-1, +1]$. A neighbourhood relation can be defined as $\eta = \{\eta_s\}_{s \in n_v}$, where $\eta_s \subseteq n_v$ is the set of neighbours of s . Now for a MRF, the following property has to hold.

$$\begin{aligned} P(X_s = x_s | X_r = x_r, r \in n_v, r \neq s) \\ = P(X_s = x_s | X_r = x_r, r \in \eta_s) \end{aligned} \quad (10)$$

where η_s denotes n^{th} -order neighbourhood of node s . Equation 10 indicates that the value of any node variable depends only on the nodes which reside inside its local neighbourhood, η_s .

Using Hammersley-Clifford theorem, we can say that a distribution, P defines a MRF on n_v with neighbourhood relation η iff it is a Gibbs distribution with respect to the same graph, (n_v, η) . Mathematically it can be represented as

$$\begin{aligned} P(X_s = x_s | X_r = x_r, r \in n_v, r \neq s) &= \frac{\exp \{-\sum_{c \in A} V_c(x)\}}{\sum_{s \in n_v} \exp \{-\sum_{c \in A} V_c(x)\}} \\ &= P(X_s = x_s | X_r = x_r, r \in \eta_s) \end{aligned} \quad (11)$$

and $U(x)$ is the energy function

$$U(x) = \sum_{c \in A} V_c(x) \quad (12)$$

where the set A consists of cliques (collection of nodes such that every two nodes are neighbours) in (n_v, η) that contains x_s . $V_c(x)$ is called the clique potential that only depends on x_s .

In the Ising model [Ising 1925], which is widely used in this context, the energy function $U(x)$ is given by

$$U(x) = -\beta \sum_{\langle s, t \rangle} x_s x_t, \quad \beta > 0 \quad (13)$$

where $s, t \in n_v$ denotes all the neighbouring pairs. Using the Ising model, we can define a prior distribution of X , i.e., any node variable without impurity as

$$f(x) = \left(\frac{1}{\sum_{s \in n_v} \exp \{U(x)\}} \right) \exp \left\{ -\beta \sum_{\langle s, t \rangle} x_s x_t \right\} \quad (14)$$

The degradation process $Y = \{y_s\}_{s \in n_v}$ is then defined as

$$y_s = x_s + \alpha_s \quad (15)$$

where α_s is the normalized randomness in the node variable introduced due to impurity. So the degradation model is a conditional probability distribution given as

$$f_{Y|X}(y|x) = \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{|\eta_s|}{2}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{s \in S} (y_s - x_s)^2 \right\} \quad (16)$$

Now we can define the distribution of node variables after the impurity is added as

$$\begin{aligned} f(y) &= \int_x f(y|x) f(x) dx \\ f(y) &= \int_{x \in \eta_s} \left(\frac{1}{\sum_{s \in n_v} \exp \{U(x)\}} \right) \\ &\quad \exp \left\{ -\beta \sum_{\langle s, t \rangle} x_s x_t - \frac{1}{2\sigma^2} \sum_{s \in S} (y_s - x_s)^2 \right\} dx \end{aligned} \quad (17)$$

Now from Equation 17 we can see that node variable distribution depends only on the nodes t that reside inside the n^{th} -order neighbourhood, η_s of the node s . Thus we can say that Y is also a Markov random process. This proves that the addition of impurities does not affect the locality property of any node variable. We validate our random graph model for simulation of materials with impurities with various examples in Section 7.

6 ARTIST CONTROLLED FRACTURE DESIGN

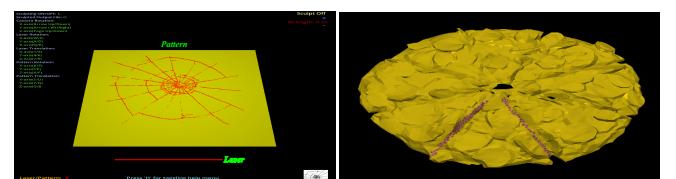


Figure 6: Our interactive user interface allows the artist to design impurity maps on objects (left). An artist interacts with and marks shallow cuts on a pizza mesh model using an impurity map (right).

We present a method to allow artists to design any specific fracture pattern that they want by leveraging our random graph-based formulation for impurity addition. An interactive interface, shown on the left in Figure 6, allows an artist to control a virtual sculpting tool to interactively create an *impurity map* on the object. A surface map can also be propagated inwards into the object, akin to a volumetric texture. The impurity map is called so because it is used to control the distribution of impurities in the region that are marked by the artist.

While using the interface, the artist can draw a free-hand impurity map using a laser sculpting tool, by placing it on the object mesh in any position. When it intersects an edge of the mesh, we weaken the fracture threshold of that edge by adding impurity to the two nodes which contain the edge. The artist can also control the potency α_k of the impurity being added.

Furthermore, our framework can embed any pattern from an image to an object mesh. Given the image of a pattern, our framework generates a design with small line segments that closely replicate the pattern. The user can also translate and rotate this design. The edges where the segments making up the pattern intersect an object mesh, the pattern gets embedded into the mesh as an impurity map and weakens the intersected edges as discussed before. The user can also control the weakening parameter value, i.e., lower the number, weaker the edge strength and vice versa. The zero value of the weaken parameter corresponds to a sharp cut. During simulation when the fracture propagates, the cracks are guided by the impurity map and thus follow the pattern created by the artist.

In the right image of Figure 6, we show the impurity map being added to a pizza model. Here like a real-life pizza maker who makes shallow cuts using the laser tool on the pizza to make slices but does not completely separate them, we weaken the pizza along these shallow cuts. The final simulation is shown in the bottom row of Figure 1. When the pizza slice is pulled apart, it closely follows these shallow cut lines but at the same time generates stretchy bits of cheese.

7 RESULTS

We showcase a wide variety of fracture simulation demos to demonstrate the robustness of our method and the diversity it can handle.¹ We use an implicit backward Euler integrator with a conjugate gradient solver [Sin et al. 2013] to solve the system dynamics. All the simulations are performed on a machine with Intel Core i7-9750H CPU at 2.60 GHz. Our multi-threaded implementation runs purely on the CPU using 12 threads and is not accelerated by a GPU. The simulation results are raytraced in Houdini for visualization. Simulation parameters for all our experiments are reported in the supplemental document. The 3D models used to generate the results are obtained from turbosquid.com, cgtrader.com, free3d.com and sketchfab.com. Volumetric simulation meshes are generated using open-source software TetWild [Hu et al. 2018b].

7.1 Anisotropic Fracture

For anisotropic material, we use anisotropic energy density from Equation 6 along with invertible Neo-Hookean energy density [Xu

et al. 2015]. The plasticity model we use is described in the supplemental material.

In the top row of Figure 1 we tear a piece of steak to reveal the intricate fracture pattern originating along the anisotropic fiber direction. The anisotropic stretchy muscle fibers inside the meat can be seen in the figure as fine thread-like structures.

Similarly, in Figure 7 (Top row), a wooden log is broken apart by pulling it from one side. The log has fibers along the direction of x -axis. In the figure, these grains produce photo-realistic fracture effects. In Figure 2, a piece of meat-filled loaf of bread is torn apart. Here we use a single mesh model with different material properties at different parts of the mesh. The coating is made of normal bread modelled as an isotropic material, while the meat inside contains anisotropic fibers in x -direction. It can be seen that the two materials fracture differently. The meat resists more before fracture due to its fiber strength.



Figure 7: A piece of dry wooden branch is broken by pulling it from one side (Top row). A damp branch that has water (figuratively) as impurity is broken in the same way as before (Bottom row). The water weakens the integrity of the material.

7.2 Impurity Induced Fracture

In Figure 7, we compare the fracture of a dry vs a rotten and damp wooden log. The addition of the impurity weakens the material here, just like a rotten wooden log has lower material strength after getting damp by absorbing water and thus produces more disjoint segments. Note that there is no explicit wetting simulation here and water is (figuratively) being considered to be like a material weakening impurity, added to produce the effect of dampness in the material.

Using our impurity-inducing model, we can also strengthen the material. We demonstrate by simulating the addition of copper to pure gold. In Figure 3, we stretch a thin bar of pure gold (in the top row) and gold-copper alloy (in the bottom row). The gold-cooper alloy demonstrates much more ductility and deforms more before undergoing fracture compared to the thin bar of pure gold.

7.3 Effect of Impurity Potency

In Figure 8, we show the effect of impurity potency on fracture. We take a hollow cylindrical tube and add an impurity map in the middle of it. The impurity map is displayed on the left-most image of Figure 8. The impurity map shows the location of the impurities (in magenta) added to the object. The yellow regions

¹Figure 13 contains an image of raw meat that some viewers may find disturbing.

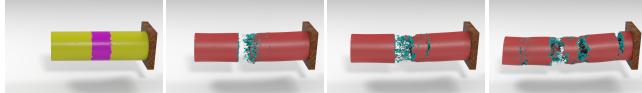


Figure 8: Illustration of impurity map (left). The effect of the impurity is decreased from left to right (next three).

have no impurity. The tube is then hinged at one end and pulled from the other. In the next three images of Figure 8, the α_k value is decreased from left to right. Higher values of α_k mean more potent impurities, i.e., they cause more weakening of the material in this case. As evident from Figure 8, decreasing the value of α_k lessens the effect of impurities on the overall strength of the object and causes fractures in multiple locations in the tube as opposed to only in the region where impurities have been added.

7.4 Artist Controlled Fracture



Figure 9: A porcelain column breaks on impact according to the embedded impurity map (inset images) with a single sharp fracture line in the first two images and a more complex fracture pattern in the last two images.

Figure 10 shows an impurity map created from an image of a pattern (pattern is depicted in Figure 6 (left)) applied to a slab mesh and demonstrates a simulation of controlled ductile fracture. When the slab is pulled from both ends, it comes apart along the embedded pattern lines. In the bottom row of Figure 1, we render a pizza slice being pulled out from a pizza, which has been pre-cut by an artist as described earlier. This is another demonstration of artist-controlled ductile fracture. In Figure 9 a porcelain column is embedded with



Figure 10: The slab model, embedded with the impurity map created from an image pattern (left), is torn apart by pulling it from both ends (middle, right).

impurity maps corresponding to a single sharp fracture line and a more complex fracture pattern. When the column is hit by a metal ball, the simulation produces sharp shards or fragments that cleanly break along the fracture edges defined by the pattern. This illustrates the ability of our method to produce artist-controlled brittle fractures as well.

7.5 Effect of the Random Graph

The random graph-based implementation is crucial to simulating materials with impurities and for artist control of fracture. In absence of the random graph, disintegration due to fracture is more chaotic and does not follow the impurity distribution closely.

In Figure 11, we see an artist-designed impurity map in the first image for a brittle object. When simulated with the random graph formulation, the object breaks only at the intended location, i.e., the arm. However, when the same impurity map is used without the random graph formulation, the simulation causes more disintegration than intended. More results on this are shown in the supplementary video and document.

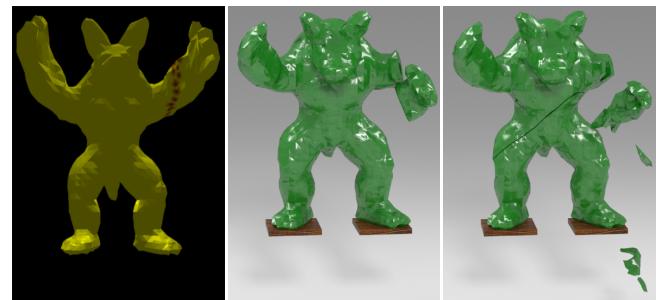


Figure 11: The artist-created impurity map is shown in the left image. Simulated fracture pattern with (middle) and without (right) random graph is shown next.

7.6 Effect of Different Noise Distributions

In Figure 12 we show that altering just the noise distribution model for impurity addition from Gaussian, to uniform within our random graph formulation does not produce any large variation in the generated fracture pattern.



Figure 12: Changing the noise model used to add the impurity within the random graph does not produce much difference in the fracture pattern. Simulated fracture pattern produced when the impurity is generated using Gaussian random variable (middle) and uniform random variable (right).

7.7 Comparison Study

7.7.1 Comparison With Real Materials. We compare our fracture simulation framework with real-world materials. In Figure 13 (top) we show a piece of raw meat being torn apart by a lion. We render a similar simulation of tearing of a piece of steak using our framework beside it. As visible in the figure, our framework can reproduce the real-world fracture phenomena, e.g., intricate fracture of the stretchy fibers in the anisotropy direction (green circle), a partially separated and dangling piece of flesh (blue circle). Similarly, in the bottom row of Figure 13, we present a real-world and a simulated

fractured wooden branch. As depicted in the figure, our framework can reproduce various intricate fracture patterns, e.g., extruded portion of broken branch generated due to extra anisotropic stiffness (green circle) and thread-like stiff wooden fibers (red circle).



Figure 13: A piece of raw meat torn apart by a lion is compared with the tearing of meat simulated using our method (top). A broken piece of a real wooden branch is compared with the breaking of a wooden branch simulated using our method (bottom.) ©Top left image: *Dreamstime* royalty-free images.

7.7.2 Comparison With Other Existing Methods. We implemented the XFEM method presented by Koschier et al. [2017] Mandal et al. [2022] and simulated the porcelain column fracture example illustrated in Figure 14. While the visual simulation of XFEM remains similar to our method, our random graph-based FEM offers a significant numerical advantage over XFEM. Initially the column object consists of 6.7k DOFs (nodes) and 40.3k tetrahedra. In order to simulate the split, we need to enrich appropriate nodes for XFEM. After enrichment, the number of DOFs (original nodes and enriched nodes together) for XFEM simulation increases to 6.9k for a single split (Figure 9 left). In contrast, the number of DOFs remains the same as beginning in random graph-based FEM. We noticed simulation of XFEM takes around 2.17 sec/frame while our graph-based FEM requires only 0.31 sec/frame. The XFEM and graph-based FEM simulations are run on the same system, with similar thread-sharing implementations. The underlying collision detection and visualization strategies also remain the same.

We also present a numerical comparison between our method and XFEM-based fracture simulation by Chitalu et al. [2020]. Using their open source code-base [Chitalu 2020], the XFEM-based method requires 2175 sec to render 16 number of cuts for the same porcelain column. On the other hand, our graph-based FEM model requires only 0.43 sec to render the same number of cuts on the same model. Thus, our method is faster and more efficient than existing methods in the literature.

We also perform a comparison study between fracture simulation for isotropic materials performed using the algorithm presented in [Mandal et al. 2021b] [Mandal et al. 2021a] versus fracture simulation for anisotropic materials and in materials with impurities, simulated with our method. For this experiment, we use a rectangular slab that is hinged at one end and pulled from another end. The middle image of Figure 15 shows the anisotropic fibre directions,



Figure 14: A porcelain column breaks on impact according to the embedded impurity map (inset image) with a single sharp fracture line. Fracture is simulated using our method (middle) and XFEM (right). Simulation of XFEM takes around 2.17 sec/frame while our method requires 0.31 sec/frame.

drawn in black stripes. The impurity map is depicted on the right-most image of the figure. The impurity map shows the location of the impurities (in magenta) added to the object. The yellow regions have no impurity. In Figure 16 we show the fracture results for isotropic (1st & 2nd column) and anisotropic (3rd & 4th column) material. Moreover, on the right side of each image, we show its corresponding strain profile. The strain values are depicted via a colour map with red (largest), green (medium) and blue (smallest) colours. Fracture of the same slab with impurity-laden material is seen in the last two columns of the same image.

The slab made of isotropic material breaks with a nearly clean horizontal fracture near the free edge being pulled. On the other hand, the slab made of anisotropic material breaks along fracture lines that are guided by the anisotropic fiber direction. Here we can clearly see patches that break along the slanted fracture lines. Due to the presence of anisotropic fibers, the material is stronger along that direction. In the last two columns of Figure 16, we show the fracture of the same slab, made of an impure material. The impurities are heavily concentrated towards the middle of the slab in the shape of a band. The slab contains a few more scattered pockets of impurity over the whole body. It can be seen that the slab primarily gets damaged along the impurity band and produces more disjoint fragments compared to the other two cases that have no impurities. The computational cost of the simulation with anisotropic materials or with materials having impurities is almost the same as simulating only isotropic materials.

8 CONCLUSION

Our work presents a method to simulate fracture in anisotropic materials simulated using graph-based FEM. We introduce a novel probabilistic damage mechanics formulation to incorporate the uncertainties during fracture, caused by the presence of material impurities. We leverage this formulation to create a framework that allows the artist-controlled design of fracture. We demonstrate the appeal and accuracy of all of the above methods for ductile and brittle fracture of various objects for anisotropic materials and materials with or without impurities. Longer-term damage phenomena like failures caused by fatigue are an interesting avenue for future work.



Figure 15: Illustration of the isotropic slab (left) in rest position, the direction of anisotropy (middle) and impurity map (right).

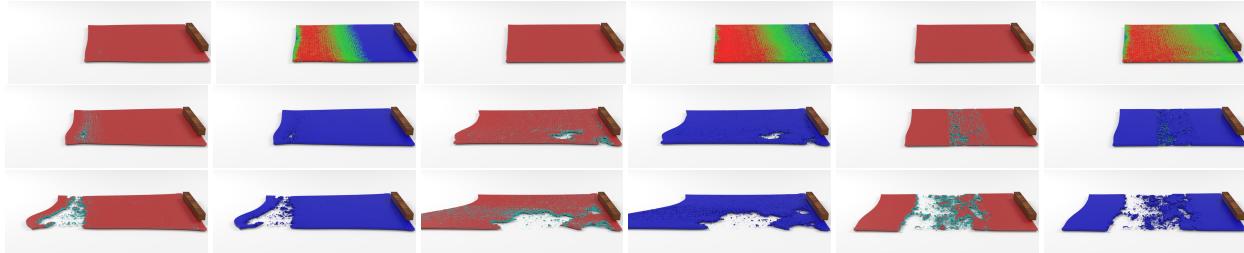


Figure 16: Comparison of fracture and corresponding strain profile for slab made of isotropic material (1st & 2nd column) [Mandal et al. 2021b], anisotropic material (3rd & 4th column) and material with impurities (5th & 6th column).

REFERENCES

- K. Aoki, N. H. Dong, T. Kaneko, and S. Kuriyama. 2004. Physically based simulation of cracks on drying 3D solids. In *Proceedings Computer Graphics International, 2004*. IEEE, 357–364.
- Zhaosheng Bao, Jeong-Mo Hong, Joseph Teran, and Ronald Fedkiw. 2007. Fracturing Rigid Materials. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (March 2007), 370–378.
- Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512.
- Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins, and Greg Turk. 2007. A Finite Element Method for Animating Large Viscoplastic Flow. *ACM Trans. Graph.* 26, 3 (July 2007), 16–es.
- Wei Chen, Fei Zhu, Jing Zhao, Sheng Li, and Guoping Wang. 2018. Peridynamics-Based Fracture Animation for Elastoplastic Solids. *Computer Graphics Forum* 37, 1 (2018), 112–124.
- Zhili Chen, Miaojun Yao, Renguo Feng, and Huamin Wang. 2014. Physics-Inspired Adaptive Fracture Refinement. *ACM Trans. Graph.* 33, 4, Article 113 (July 2014), 7 pages.
- Floyd M. Chitalu. 2020. MCUT. Retrieved August 04, 2022 from <https://github.com/cutdigital/mcut>
- Floyd M. Chitalu, Qinghai Miao, Kartic Subr, and Taku Komura. 2020. Displacement-Correlated XFEM for Simulating Brittle Fracture. *Computer Graphics Forum* 39, 2 (2020), 569–583.
- Jessica T. Davis, Matteo Chinazzi, Nicola Perra, Kunpeng Mu, Ana Pastore y Piontti, Marco Ajelli, Natalie E. Dean, Corrado Gioannini, Maria Litvinova, Stefano Merler, Luca Rossi, Kaiyuan Sun, Xinyue Xiong, Ira M. Longini, M. Elizabeth Halloran, Cécile Viboud, and Alessandro Vespignani. 2021. Cryptic transmission of SARS-CoV-2 and the first COVID-19 wave. *Nature* 600, 7887 (2021), 127–132.
- Linxi Fan, Floyd M. Chitalu, and Taku Komura. 2022. Simulating Brittle Fracture with Material Points. *ACM Trans. Graph.* 41, 5, Article 177 (may 2022), 20 pages.
- E. N. Gilbert. 1959. Random Graphs. *The Annals of Mathematical Statistics* 30, 4 (1959), 1141 – 1144.
- David Hahn and Chris Wojtan. 2015. High-Resolution Brittle Fracture Simulation with Boundary Elements. *ACM Trans. Graph.* 34, 4, Article 151 (July 2015), 12 pages.
- David Hahn and Chris Wojtan. 2016. Fast Approximations for Boundary Element Based Brittle Fracture Simulation. *ACM Trans. Graph.* 35, 4, Article 104 (July 2016), 11 pages.
- Koichi Hirota, Yasuyuki Tanoue, and Toyohisa Kaneko. 2000. Simulation of three-dimensional cracks. *The Visual Computer* 16 (2000), 371 – 378.
- Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018a. A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-Way Rigid Body Coupling. *ACM Trans. Graph.* 37, 4, Article 150 (July 2018), 14 pages.
- Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018b. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages.
- G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Grenoble, France) (SCA '04). Eurographics Association, Goslar Germany, Germany, 131–140.
- Ernst Ising. 1925. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik* 31, 1 (1925), 253–258.
- Doug L. James and Dinesh K. Pai. 1999. ArtDefo: Accurate Real Time Deformable Objects. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–72.
- Svante Jansson and Lutz Warnke. 2021. Preferential attachment without vertex growth: Emergence of the giant component. *The Annals of Applied Probability* 31, 4 (2021), 1523 – 1547.
- Parisa Khodabakhshi, J. N. Reddy, and Arun Srinivasa. 2016. GraFEA: a graph-based finite element approach for the study of damage and fracture in brittle materials. *Mecanica* 51 (2016), 3129 – 3147.
- Parisa Khodabakhshi, J. N. Reddy, and Arun Srinivasa. 2019. A nonlocal fracture criterion and its effect on the mesh dependency of GraFEA. *Acta Mechanica* 230 (2019), 3593–3612.
- Theodore Kim, Fernando De Goes, and Hayley Iben. 2019. Anisotropic Elasticity for Inversion-Safety and Element Rehabilitation. *ACM Trans. Graph.* 38, 4, Article 69 (July 2019), 15 pages.
- Theodore Kim and David Eberle. 2020. Dynamic Deformables: Implementation and Production Practicalities. In *ACM SIGGRAPH 2020 Courses* (Virtual Event, USA) (SIGGRAPH '20). Association for Computing Machinery, New York, NY, USA, Article 23, 182 pages.
- Dan Koschier, Jan Bender, and Nils Thuerey. 2017. Robust EXTended Finite Elements for Complex Cutting of Deformables. *ACM Trans. Graph.* 36, 4, Article 55 (July 2017), 13 pages.
- Dan Koschier, Sebastian Lipponer, and Jan Bender. 2015. Adaptive Tetrahedral Meshes for Brittle Fracture Simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) (SCA '14). Eurographics Association, Goslar, DEU, 57–66.
- J. A. Levine, A. W. Bargteil, C. Corsi, J. Tessendorf, and R. Geist. 2015. A Peridynamic Perspective on Spring-Mass Fracture. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) (SCA '14). Eurographics Association, Goslar, DEU, 47–55.
- Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2021a. Remeshing-Free Graph-Based Finite Element Method for Ductile and Brittle Fracture. *CoRR* abs/2103.14870 (2021). arXiv:2103.14870 <https://arxiv.org/abs/2103.14870>
- Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2021b. Scalable Visual Simulation of Ductile and Brittle Fracture. In *ACM SIGGRAPH 2021 Posters* (Virtual Event, USA) (SIGGRAPH '21). Association for Computing Machinery, New York, NY, USA, Article 41, 2 pages.
- Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2022. Interactive Physics-Based Virtual Sculpting with Haptic Feedback. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 1, Article 9 (may 2022), 20 pages.

- Neil Molino, Zhaosheng Bao, and Ron Fedkiw. 2004. A Virtual Node Algorithm for Changing Mesh Topology during Simulation. In *ACM SIGGRAPH 2004 Papers* (Los Angeles, California) (*SIGGRAPH '04*). Association for Computing Machinery, New York, NY, USA, 385–392.
- Matthias Müller and Markus Gross. 2004. Interactive Virtual Materials. In *Proceedings of Graphics Interface* (London, Ontario, Canada). Canadian Human-Computer Communications Society, Waterloo, CAN, 239–246.
- James F. O'Brien, Adam W. Bargteil, and Jessica K. Hodgins. 2002. Graphical Modeling and Animation of Ductile Fracture. *ACM Trans. Graph.* 21, 3 (July 2002), 291–294.
- James F. O'Brien and Jessica K. Hodgins. 1999. Graphical Modeling and Animation of Brittle Fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH '99*). ACM Press/Addison-Wesley Publishing Co., USA, 137–146.
- Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, and James F. O'Brien. 2014. Adaptive Tearing and Cracking of Thin Sheets. *ACM Trans. Graph.* 33, 4, Article 110 (July 2014), 9 pages.
- Boris Pittel. 2010. On a random graph evolving by degrees. *Advances in Mathematics* 223, 2 (2010), 619–671.
- J.N. Reddy and Arun Srinivasa. 2015. On the force–displacement characteristics of finite elements for elasticity and related problems. *Finite Elements in Analysis and Design* 104 (2015), 35 – 40.
- Eftychios Sifakis, Kevin G. Der, and Ronald Fedkiw. 2007. Arbitrary Cutting of Deformable Tetrahedralized Objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) (*SCA '07*). Eurographics Association, Goslar, DEU, 73–80.
- F. S. Sin, D. Schroeder, and J. Barbić. 2013. Vega: Non-Linear FEM Deformable Object Simulator. *Computer Graphics Forum* 32, 1 (2013), 36–48.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (March 2018), 15 pages.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (Feb. 2019), 15 pages.
- Alexey Stomakhin, Russell Howes, Craig Schroeder, and Joseph M. Teran. 2012. En-ergetically Consistent Invertible Elasticity. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics Conference on Computer Animation* (Lausanne, Switzerland) (*EUROSCA'12*). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 25–32.
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. Graph.* 32, 4, Article 102 (July 2013), 10 pages.
- Demetri Terzopoulos and Kurt Fleischer. 1988. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 269–278.
- Jeffrey A. Weiss, Bradley N. Maker, and Sanjay Govindjee. 1996. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering* 135, 1 (1996), 107–128.
- Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O'Brien. 2010. Dynamic Local Remeshing for Elasto-plastic Simulation. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) (*SIGGRAPH '10*). Association for Computing Machinery, New York, NY, USA, Article 49, 11 pages.
- Joshuah Wolper, Yunuo Chen, Minchen Li, Yu Fang, Ziyin Qu, Jiecong Lu, Meggie Cheng, and Chenfanfu Jiang. 2020. AnisoMPM: Animating Anisotropic Damage Mechanics. *ACM Trans. Graph.* 39, 4, Article 37 (July 2020), 16 pages.
- Joshuah Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. 2019. CD-MPM: Continuum Damage Material Point Methods for Dynamic Fracture Animation. *ACM Trans. Graph.* 38, 4, Article 119 (July 2019), 15 pages.
- Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbić. 2015. Nonlinear Material Design Using Principal Stretches. *ACM Trans. Graph.* 34, 4, Article 75 (July 2015), 11 pages.
- Yufeng Zhu, Robert Bridson, and Chen Greif. 2015. Simulating Rigid Body Fracture with Surface Meshes. *ACM Trans. Graph.* 34, 4, Article 150 (July 2015), 11 pages.