

High-Resolution Brittle Fracture Simulation with Boundary Elements

David Hahn*
IST Austria

Chris Wojtan†
IST Austria

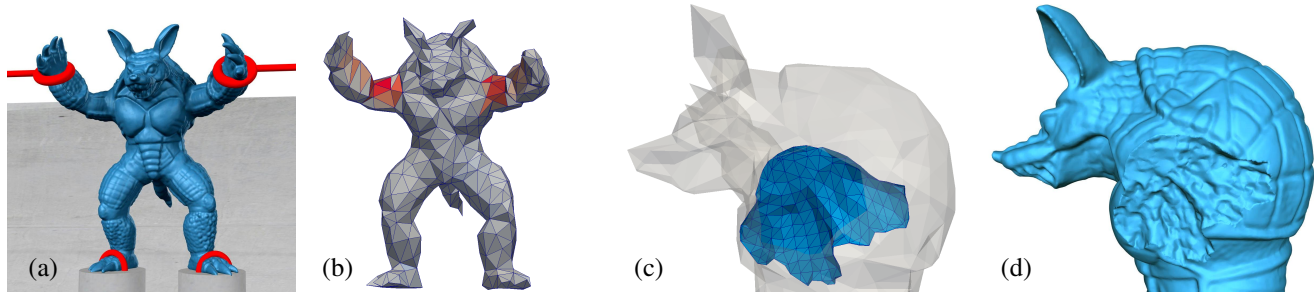


Figure 1: *Tearing the armadillo: (a) input geometry and illustration of boundary conditions; (b) BEM mesh with maximal principal stress shown in red; (c) BEM mesh of a generated fracture; (d) high-resolution fracture surface.*

Abstract

We present a method for simulating brittle fracture under the assumptions of quasi-static linear elastic fracture mechanics (LEFM). Using the boundary element method (BEM) and Lagrangian crack-fronts, we produce highly detailed fracture surfaces. The computational cost of the BEM is alleviated by using a low-resolution mesh and interpolating the resulting stress intensity factors when propagating the high-resolution crack-front.

Our system produces physics-based fracture surfaces with high spatial and temporal resolution, taking spatial variation of material toughness and/or strength into account. It also allows for crack initiation to be handled separately from crack propagation, which is not only more reasonable from a physics perspective, but can also be used to control the simulation.

Separating the resolution of the crack-front from the resolution of the computational mesh increases the efficiency and therefore the amount of visual detail on the resulting fracture surfaces. The BEM also allows us to re-use previously computed blocks of the system matrix.

CR Categories: G.1.9 [Mathematics of Computing]: Numerical Analysis—Integral Equations; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: boundary elements, brittle fracture, crack propagation

*e-mail: david.hahn@ist.ac.at

†e-mail: wojtan@ist.ac.at

1 Introduction

Computer graphics researchers have developed diverse methods for fracturing virtual objects. Purely geometric approaches use either pre-fractured models or pre-defined fracture patterns, while fracture simulations (including mass-spring systems, finite element methods, and mesh-less methods) respect the underlying physics of the fracture process. Unfortunately, these simulation methods become inefficient if all the detail of the fracture surfaces is present in the computational model, effectively limiting the visual detail that can be captured. Adaptive re-meshing (or re-sampling in mesh-less methods) is typically used to mitigate this limitation. Some methods also use heuristics to add more visual detail to fracture surfaces as a post-process, but they do not influence the simulation in any way.

The standard approach in fracture simulation for computer graphics is to cut or re-mesh one element at a time as the crack propagates through the material. This requires small time-steps and is analogous to an *Eulerian* reference frame as the crack advances through space. Our approach departs from this traditional viewpoint by adopting a *Lagrangian* reference frame for crack propagation, as illustrated in Fig. 2. This new point of view allows us to utilize techniques from front tracking, improves the achievable resolution of fracture surfaces, and still treats the underlying physics with acceptable accuracy. We combine a boundary element method (BEM) on a coarse computational mesh with a high-resolution crack propagation scheme, resulting in a fast and efficient framework for brittle fracture. Our results show that this method is capable of producing detailed, realistic fracture surfaces while avoiding complicated mesh manipulations.

Furthermore, a common artifact in computer graphics simulations is *artificial shattering*, which results from the use of stress-based fracture criteria. We introduce a new method based on linear elastic fracture mechanics (LEFM) that avoids this problem altogether by treating crack initiation and propagation separately. Our method utilizes both *strength* and *toughness* in the physically correct way, where crack *initiation* depends on stress, but crack *propagation* is governed by stress intensity factors.

The main contributions of our method are:

- an efficient symmetric Galerkin BEM implementation for quasi-static brittle fracture simulation, where each entry in the system matrix is computed only once;

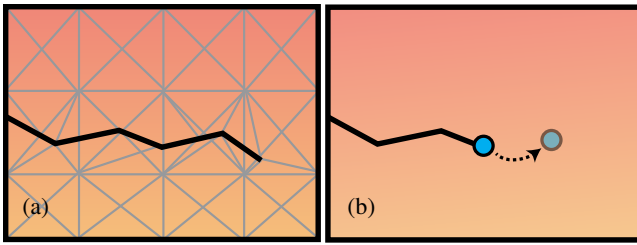


Figure 2: Standard methods propagate fractures by re-meshing (a), while our approach treats the crack-front as a Lagrangian flow problem (b).

- fracture criteria treating crack initiation and propagation physically correctly in the context of LEFM;
- an interpolation scheme that allows for much more visual detail of fractures than present in the computational mesh; and
- a new, fast, and simple treatment of spatially varying toughness during crack propagation yielding realistic fracture patterns.

As far as we are aware, this is also the first time a BEM is used for fracture simulation in computer graphics.

2 Related work

In this section we summarize previous work on handling fractures in solid objects, both in computer graphics as well as mechanical engineering.

Purely geometric methods such as [Su et al. 2009; Müller et al. 2013] are popular in computer graphics, since they are typically much faster than any simulation. Fractures are either pre-defined during modeling or generated by applying a pre-defined fracture pattern and then updating the topology of the object. Some methods also combine physics-based deformation models with geometric fracturing [Iben and O’Brien 2006; Glondu et al. 2012; Schwartzman and Otaduy 2014]. Furthermore, additional visual detail can be generated for coarse simulation results by modifying the fracture surfaces on a smaller scale [Chen et al. 2014]. For the remainder of this discussion we focus on continuum mechanical models of fracture.

Physics-based fracture simulations typically consist of three steps: first deformations and/or internal forces (stresses) are computed, then some fracture criterion is evaluated, and finally the geometric model is updated to account for newly formed cracks.

2.1 Deformation

There is a wide range of methods available for the deformation step, such as mass-spring networks [Norton et al. 1991; Hirota et al. 1998; Hirota et al. 2000; Smith et al. 2001; Levine et al. 2014], finite element methods (FEM) [O’Brien and Hodgins 1999; Müller et al. 2001; O’Brien et al. 2002; Bao et al. 2007; Wicke et al. 2010; Zheng and James 2010; Koschier et al. 2014] and extended FEM (X-FEM) [Moës et al. 2002; Gravouil et al. 2002; Abdelaziz and Hamouine 2008; Kaufmann et al. 2009; Mousavi et al. 2011], mesh-less or particle-based methods [Pauly et al. 2005] such as the material point method (MPM) [Stomakhin et al. 2013; Stomakhin et al. 2014], and boundary element methods (BEM) [Aliabadi 1997; James and Pai 1999; Wilde and Aliabadi 1999; Frangi et al. 2002; Sutradhar et al. 2008; Kielhorn 2009; Messner and Schanz 2010; Sauter and Schwab 2011; Keeler and Bridson 2014].

While mass-spring systems discretize deformable objects as a collection of point-masses with forces between them (usually confined to a small neighborhood of each point), all other methods listed above are based on continuum mechanics. The main differences between FEM, X-FEM, MPM, and BEM can be characterized by the distribution of degrees of freedom and the resulting approximation spaces. Degrees of freedom in FEM and X-FEM are stored on (the vertices of) a volumetric mesh, spanning a piecewise polynomial function space (FEM), which can be enriched by additional, specialized basis functions (X-FEM). The MPM usually stores mass and velocity on particles, while forces are computed on a regular grid, and data is interpolated between the grid and particles.

The degrees of freedom in a BEM, however, are stored on a surface mesh and the boundary integral form of the governing equations is used instead of (volumetric) partial differential equations. While this limits the possibility to treat spatially varying elasticity parameters, it reduces the required number of degrees of freedom. Furthermore, only the boundary data is approximated, but the governing equations within the domain are satisfied exactly using a fundamental solution (also called Green’s function). In contrast, a FEM approximates the solution in a (typically) piecewise polynomial space.

2.2 Fracture criteria

The second step of the fracture simulation is evaluating the fracture criterion, i.e. deciding whether a material fails at a certain location under a given load. Terzopoulos and Fleischer [1988] use a maximal strain threshold, while O’Brien and colleagues [1999; 2002] use a stress-based separation tensor at the nodes of a tetrahedral FEM mesh. This method has been combined with dynamic mesh improvement [Wicke et al. 2010], and also modified by Pfaff et al. [2014] to fracture thin-plate triangle meshes. Koschier et al. [2014] use a similar maximal tensile stress criterion to initiate cracks, and then propagate them until they reach the surface of the object. They acknowledge, however, that cracks should terminate based on the energy dissipated in the process, as done by Glondu et al. [2013].

One important distinction to make is the difference between the material parameters *strength* and *toughness*. Strength describes how much load is required to start a fracture, whereas toughness determines whether an existing crack will (continue to) propagate [Freund 1998; Rabczuk 2013]. For example, a strong but fragile specimen will withstand a large amount of load as long as it contains no cracks, but once it starts to fracture, cracks will quickly cut completely through it. On the other hand, in a weak but tough specimen small cracks will appear at relatively low loads, but only few of them will propagate (where the load is sufficiently high). Our examples in §8 demonstrate the desired behavior; see also Fig. 10b. A similar concept has also been used in [Smith et al. 2001].

Consequently, working only with maximal stress criteria for brittle fractures can lead to artificial shattering, as many nodes in a high-stress region may fulfill the criterion at the same time. Pfaff et al. [2014] propose a local update procedure to mitigate this issue, while Koschier et al. [2014] first find such regions and then pick only the point of maximal stress within each region to initiate a fracture.

We believe that the proper way to deal with brittle fractures is to treat crack initiation and propagation separately. Even nominally brittle materials exhibit a very small plastic zone around crack-fronts, which is not captured by the LEFM model [Gross and Seelig 2011]. Consequently, stress is not the proper measure for crack propagation, as it is singular at the crack-front. We use stress intensity factors instead, which describe the magnitude of the singular

stress field in the vicinity of crack-fronts [Freund 1998].

2.3 Geometry update

The final step of the fracture simulation is to update the (discretized) object to include the newly added cracks. In mass-spring or particle-based systems this is usually done very efficiently by removing interactions between mass-points/particles. Fractures are included in X-FEM by modifying the approximation space with discontinuous enrichment functions [Moës et al. 2002; Gravouil et al. 2002; Abdelaziz and Hamouine 2008; Mousavi et al. 2011], while this is also reasonably efficient, it does complicate the integration of the elements’ stiffness matrices. For FEM and BEM adding fracture surfaces requires modification of the mesh. This can be problematic for FEM, since cutting elements (as done in [O’Brien and Hodgins 1999; O’Brien et al. 2002; Wicke et al. 2010; Pfaff et al. 2014]) may cause slivers, and demand further mesh-improvement. A more stable approach is to duplicate elements rather than cut them [Molino et al. 2004], but this may cause undesirable changes to the mass-distribution around fracture surfaces; Wojtan and Turk [2008] propose a sub-element accurate mass computation and Nesme et al. [2009] describe how to handle sub-element accurate stiffness.

Since a BEM deals only with a surface mesh, adding fracture geometry is much more straightforward. A “DualBEM” approach requires both sides of a crack to be represented in the mesh [Portela et al. 1992; Wilde and Aliabadi 1999]. This means that fractures need to be connected to existing surfaces where they meet. However, when reducing the problem to computing only crack-opening displacements (instead of absolute displacements of both sides of the crack), LEFM problems can be solved using a single triangle-sheet per crack [Frangi et al. 2002].

Our method builds on the BEM formulation presented by Frangi et al. [2002], who have shown that it accurately computes stress intensity factors (SIF) in fractured objects. While it is known that SIF are underestimated near boundaries (which would not happen in a DualBEM [Wilde and Aliabadi 1999]), it has the advantage that adding new fracture surfaces can be done simply by inserting triangles into the mesh; no changes need to be made to previously existing elements. This allows us to re-use the system matrix from the previous time-step and add new blocks representing the most recently added triangles. We present more details of this method in §4. While Frangi et al. [2002] only consider static fracture configurations, our algorithm also computes dynamic crack propagation, integrates spatially-varying toughness, and generates high-resolution fracture surfaces.

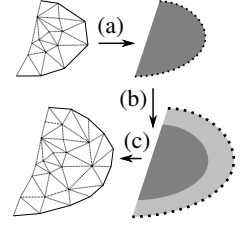
3 Overview

In this section we outline our fracture simulation method, before providing details in the following sections. Starting from a given (possibly highly detailed) surface mesh, we first convert it to an implicit surface for the reasons discussed in §7 and construct a coarse triangle mesh for the BEM. In some of our examples we start directly with a coarse mesh.

We then apply given boundary conditions and compute surface displacements and stresses on the coarse mesh (§4). Based on the surface stresses, we start new cracks if the conditions given in §5.1 are met. Any new crack will be added to both the coarse mesh and the high-resolution implicit surface.

Whenever new fracture surface elements are added to the coarse mesh, we update the BEM and obtain stress intensities at the crack-front (§4.1). Our crack propaga-

tion algorithm then proceeds as illustrated in the inset figure: (a) interpolate stress intensities from the BEM mesh to crack-front markers (§6.1); (b) propagate the markers (§5.2); (c) update the BEM mesh and repeat. We also consider spatially varying toughness during propagation, as described in §6.2. Crack propagation stops if the crack-front intersects another surface, or the stress intensity becomes too low. The simulation terminates if either no new fracture surfaces have been generated, or after a user-specified number of time-steps.



Finally, we find disconnected geometry components using the implicit surface and convert them to high-resolution meshes for rendering. In order to produce a world-space visualization, we also interpolate displacements from the BEM solution to the high-resolution meshes (§7). Results produced with our method are shown in §8.

4 Boundary Element Method

In the following paragraphs we briefly summarize the derivation of the symmetric Galerkin boundary element method (SGBEM) for linear elasticity problems. We follow the notation of Kielhorn [2009], to which we refer the interested reader for further details.

We start with the equations of linear elastostatics in the absence of body forces: the elastostatic equilibrium of a solid object is then $\nabla \cdot \boldsymbol{\sigma} = 0$. Stress $\boldsymbol{\sigma}$ is related to strain $\boldsymbol{\varepsilon}$ by an isotropic constitutive model $\boldsymbol{\sigma} := 2\mu\boldsymbol{\varepsilon} + \lambda \text{tr}(\boldsymbol{\varepsilon})\mathbf{I}$ with constant Lamé parameters (μ, λ) and linearized strain $\boldsymbol{\varepsilon} := (\nabla\mathbf{u} + \nabla\mathbf{u}^\top)/2$. Here $\mathbf{u}(\mathbf{x})$ is the displacement vector field, mapping a material-space coordinate \mathbf{x} to a world-space coordinate $\mathbf{x} + \mathbf{u}(\mathbf{x})$.

This problem can be written as a boundary integral equation (1), where $\Gamma = \partial\Omega$ denotes the boundary of the computational domain.

$$\mathbf{u}_\Gamma(\mathbf{x}) = \text{Tr}_\mathbf{x} \int_\Gamma \mathbf{q}_\Gamma(\mathbf{y})\mathbf{U}(\mathbf{y}, \mathbf{x})d\mathbf{s}_\mathbf{y} - \text{Tr}_\mathbf{x} \int_\Gamma \mathbf{u}_\Gamma(\mathbf{y})(\mathcal{T}_\mathbf{y}\mathbf{U})^\top(\mathbf{y}, \mathbf{x})d\mathbf{s}_\mathbf{y} \quad (1)$$

Here \mathbf{u}_Γ are boundary displacements and \mathbf{q}_Γ are boundary tractions (i.e. forces per surface area). The boundary trace operator Tr “restricts” a function from Ω to Γ (not to be confused with the trace of a matrix, tr), while the traction operator \mathcal{T} acts as a generalized normal derivative. Finally, \mathbf{U} is the fundamental solution such that:

$$\mathbf{u}(\mathbf{y}) = \int_\Omega (\mathcal{L}_\mathbf{x}\mathbf{U}(\mathbf{y}, \mathbf{x}))\mathbf{u}(\mathbf{x})d\mathbf{x} \quad \forall \mathbf{y} \in \Omega$$

$$\mathcal{L} := ((\lambda + \mu)\nabla\nabla\cdot) + \mu\nabla^2.$$

Intuitively, the fundamental solution (or Green’s function) $\mathbf{U}(\mathbf{y}, \mathbf{x})$ gives the displacement observed at point \mathbf{y} due to a concentrated point load at \mathbf{x} for an infinite body. Applying the boundary traction operator to Eq. (1) yields the boundary integral equation for tractions:

$$\mathbf{q}_\Gamma(\mathbf{x}) = \mathcal{T}_\mathbf{x} \int_\Gamma \mathbf{q}_\Gamma(\mathbf{y})\mathbf{U}(\mathbf{y}, \mathbf{x})d\mathbf{s}_\mathbf{y} - \mathcal{T}_\mathbf{x} \int_\Gamma \mathbf{u}_\Gamma(\mathbf{y})(\mathcal{T}_\mathbf{y}\mathbf{U})^\top(\mathbf{y}, \mathbf{x})d\mathbf{s}_\mathbf{y}. \quad (2)$$

Depending on the type of the given boundary conditions, the SGBEM applies Eq. (1) on surfaces where displacements are

known, and Eq. (2) where tractions are known [Frangi et al. 2002; Sutradhar et al. 2008; Kielhorn 2009]. We use piecewise linear and piecewise constant approximations for displacements and tractions respectively. The system-matrix for this linear boundary integral problem is assembled by regularization and quadrature of the 4 integral operators in Eq. (1) and (2), see also Eq. (12). Separating known and unknown boundary data yields the following linear block-matrix system, which is the starting point for our implementation; see Eq. (5.16) in [Kielhorn 2009]:

$$\begin{bmatrix} \mathbf{V} & -\mathbf{K} \\ \mathbf{K}^\top & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_D \\ \mathbf{f}_N \end{bmatrix}. \quad (3)$$

Here \mathbf{u} and \mathbf{q} are unknown boundary displacements and tractions respectively, while the right hand side is assembled using the known Dirichlet and Neumann boundary data. The first and second row in Eq. (3) represent the discretized versions of Eq. (1) and (2) respectively. The equations for each block are given in the Appendix. We note that the first block, \mathbf{V} , is symmetric positive definite, and the system can be solved using Schur complements; see Eq. (5.18) in [Kielhorn 2009] for details. The whole system-matrix, however, is dense. Please also note that the required regularization and quadrature techniques have been omitted here for brevity and can be found in great detail in the book by Sauter and Schwab [2011]; these are implemented in the HyENA library, available under LGPL [HyENA 2013], which we use in our implementation.

4.1 SGBEM for quasi-static fracture dynamics

Assuming an unfractured object, proper boundary conditions, and a triangle mesh such that we can assemble and solve Eq. (3), we describe how to add a crack to this system in the following paragraphs. Extending an existing crack follows the same principle. Since the crack relieves internal stress, the crack faces are traction-free in the quasi-static equilibrium.

In material-space, the crack has two coincident faces, Γ_c^+ and Γ_c^- , with opposite surface normals and unknown displacements \mathbf{u}^+ and \mathbf{u}^- . The crack-front is the common bounding curve of these faces. However, treating these coincident surfaces when formulating a BEM is challenging due to the singularities of the fundamental solution. Furthermore, stress is singular at the crack-front.

There are two methods available to resolve this issue. The DualBEM [Portela et al. 1992; Wilde and Aliabadi 1999] uses Eq. (1) on one of the fracture's faces, and Eq. (2) on the other. This method, however, requires careful treatment of integrations over coincident elements, as well as special "crack-tip elements" to handle the singular stress field.

On the other hand, the SGBEM formulation of Frangi et al. [2002] reduces the system by solving for the crack-opening displacement $\Delta \mathbf{u} := \mathbf{u}^+ - \mathbf{u}^-$ instead of both \mathbf{u}^+ and \mathbf{u}^- . Consequently, the fracture can be represented by a single sheet of triangles. On this surface, Eq. (2) is applied since we have zero traction boundary conditions. The reduced system is

$$\begin{bmatrix} \mathbf{V} & -\mathbf{K} & -\mathbf{K}_c \\ \mathbf{K}^\top & \mathbf{D} & \mathbf{D}_c \\ \mathbf{K}_c^\top & \mathbf{D}_c^\top & \mathbf{D}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{u} \\ \Delta \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_D \\ \mathbf{f}_N \\ 0 \end{bmatrix}. \quad (4)$$

To resolve the issue of the stress-singularity at the crack-front, we simply require that the opening displacement is zero at the crack-front and use a piecewise constant approximation of tractions. This means that there is no unknown data stored directly at the crack-front.

Whenever we add new fracture elements, the blocks \mathbf{K}_c , \mathbf{D}_c and \mathbf{D}_{cc} , as well as the unknown vector $\Delta \mathbf{u}$, grow accordingly, but

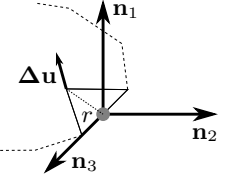
none of the previously computed parts of these blocks need to be changed.

A disadvantage of this formulation is that fractures are never topologically connected to the surface of the object, hence displacements—and consequently stress intensity factors (SIF)—will be under-estimated for crack-fronts in the vicinity of the object's surface. Please refer to [Frangi et al. 2002] for quantitative comparisons. Intuitively, this slightly reduces crack speeds near the boundary. We believe, however, that this problem is outweighed by the simplicity of the formulation, since it does not require modifying any existing elements when adding new fractures. This choice is also very efficient, because the number of unknowns on crack surfaces is reduced by half compared to the DualBEM. We have implemented this method using the HyENA [2013] library, with minor modifications in order to assemble the new matrix blocks \mathbf{K}_c , \mathbf{D}_c and \mathbf{D}_{cc} .

Solving Eq. (4) immediately provides crack-opening displacements (COD), $\Delta \mathbf{u}$, which lends itself to fast evaluation of stress intensity factors (SIF) via the displacement correlation technique, cf. Eq. (25) in [Ingraffea and Wawrzynek 2003]:

$$\begin{aligned} K_I &= \mu \sqrt{2\pi} \frac{\Delta u_I}{\sqrt{r}(2-2\nu)}, \\ K_{II} &= \mu \sqrt{2\pi} \frac{\Delta u_{II}}{\sqrt{r}(2-2\nu)}, \\ K_{III} &= \mu \sqrt{\pi} \frac{\Delta u_{III}}{\sqrt{2r}}. \end{aligned} \quad (5)$$

For every crack-front edge in the BEM mesh we choose the interior node of the triangle containing this edge as the correlation point. The COD, $\Delta \mathbf{u}$, is evaluated at the correlation point and projected onto the local orthonormal coordinate system ($\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$) of the crack-front, yielding Δu_I , Δu_{II} , and Δu_{III} . The stress-intensities (K_I, K_{II}, K_{III}) are evaluated according to Eq. (5), where ν is Poisson's ratio and r is the distance from the crack-front to the correlation point, as illustrated in the inset figure. Finally, we average the resulting SIF from the edges to the crack-front nodes.



We note that there exist more accurate methods for extracting SIF from simulation results, such as path-independent integrals [Ingraffea and Wawrzynek 2003], but displacement correlation is the fastest and sufficiently accurate for our purposes.

5 Fracture dynamics

Having described how we compute quasi-static stress intensities at crack-fronts in the previous section, we now present our method for simulating dynamically growing fractures. First, we discuss crack initiation based on surface stresses, and then we introduce 3D crack-front motion. Later on, in §6, we discuss how we increase the fracture resolution and our treatment of spatially varying toughness.

5.1 Crack initiation

It is possible to initiate cracks anywhere inside the object, based on any criterion due to interior stress, surface stress, or at any location specified by the user. Unless otherwise stated we choose to use surface stresses for efficiency, since interior evaluations are somewhat costly in a BEM.

We start a new crack if the following conditions hold:

- (a) an element’s principal stress exceeds the local material strength,
- (b) this element has not initiated a crack before,
- (c) this element is farther than the average BEM edge-length from any other crack-front, and
- (d) the (user-specified) limit for the maximum number of cracks in the simulation has not been reached yet.

Condition (a) is our main criterion for crack initiation, and we use the maximal principal stress to account for tensile fracture, as well as the minimal principal stress for compressive fracture. The element where stress exceeds the local strength the most will be fractured first.

Since we do not cut any elements when a crack meets another surface, we keep a list of fractured elements to evaluate condition (b), so any element can initiate at most one crack. Otherwise we risk creating overlapping elements which introduces untreated singularities in the BEM integrals.

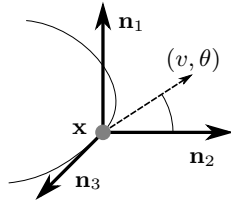
We use conditions (c) and (d) to increase the efficiency of our method. The intuition for (c) is that starting a crack in the vicinity of another crack-front will very likely become redundant as soon as that crack-front propagates. We find that condition (d) is also an easy way to allow the user some control over the simulation.

In order to evaluate condition (a), we follow the principal stress computation used for co-rotational FEM [Irving et al. 2004]. This can be done for every triangle in the BEM mesh by conceptually extending it to a tetrahedron in the normal direction, following §3 in [Sumner and Popović 2004]. We then compute a SVD of the per-element deformation gradient $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and evaluate the principal stresses based on the strain measure $(\mathbf{\Sigma} - \mathbf{I})$. Evaluating condition (a) is now straightforward, using the maximal and minimal principal stresses and comparing to the tensile and compressive strength respectively. Strength is evaluated at the centroid of the triangle in question.

When starting a new crack (i.e. when all four conditions are met), we need two unit-vectors to describe its orientation: the fracture surface normal \mathbf{n}_1 , and the “forward” direction \mathbf{n}_2 along the fracture surface (see inset figure below). The former is given by the orientation of the maximal/minimal principal stress, which is found in the corresponding column of \mathbf{V} and we choose the inward normal of the triangle for the latter (i.e. the crack is set to propagate away from the surface and into the object).

5.2 Crack propagation

When new fracture elements are added to the BEM mesh, we update the blocks \mathbf{K}_c , \mathbf{D}_c , and \mathbf{D}_{cc} of the system matrix in Eq. (4) (also see Appendix) and compute stress intensities for all nodes on the crack-front using Eq. (5). We then use our crack propagation criterion, Eq. (7), to determine whether the crack propagates at a point \mathbf{x} , where stress intensities (K_I, K_{II}, K_{III}) are given in the local coordinate system $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3)$, as illustrated in the inset figure. If the crack propagates, we determine the propagation speed $v(\mathbf{x})$ and (out-of-plane) direction $\theta(\mathbf{x})$ according to Eq. (8) and (10) respectively. We discuss our treatment of spatially varying toughness in §6.2.



The three loading modes are defined following [Freund 1998] and illustrated in Fig. 3; I: “opening” along the surface normal \mathbf{n}_1 , II:

“sliding” along the crack-front normal \mathbf{n}_2 , and III: “tearing” along the crack-front tangent \mathbf{n}_3 .

Intuitively we expect that a crack propagates if the strain energy release rate G exceeds the energy required to create the new fracture surfaces, described by the material’s surface energy γ : $G \geq 2\gamma$. (The factor of 2 accounts for the fact that the crack creates two surfaces; see [Freund 1998] for details.) Furthermore, a homogeneous material under pure mode-I tension should exhibit a planar fracture (orthogonal to \mathbf{n}_1 , i.e. $\theta = 0$), while mode-II loading should cause the crack to (smoothly) turn away from that plane. Finally, mode-III loading both contributes to the forward propagation of the crack, but also causes the crack-front to twist around \mathbf{n}_2 ; see illustration in Fig. 3, as well as results in Fig. 4. We found that we do not need to handle this twisting explicitly, as it will happen automatically due to varying results for K_{II} along the crack-front if mode-III loading is present (see Fig. 4). This means that it is sufficient to compute only $v(\mathbf{x})$ and $\theta(\mathbf{x})$ during crack propagation. We terminate crack propagation when the crack-front meets a surface (either another fracture or the object’s boundary), which is implemented using intersection tests on an implicit surface representation, as described in §7.

Our fracture criterion can then be written as follows: the strain energy release rate is given by Anderson [2005] as:

$$G = \frac{1 - \nu^2}{E} (K_I^2 + K_{II}^2) + \frac{1 + \nu}{E} K_{III}^2, \quad (6)$$

where E is Young’s modulus and ν is Poisson’s ratio. Using the material’s toughness $K_c^2 := 2\gamma E / (1 - \nu^2)$ and introducing the *effective stress intensity* $K_{\text{eff}}^2 := K_I^2 + K_{II}^2 + K_{III}^2 / (1 - \nu)$, the crack propagates if

$$K_{\text{eff}} \geq K_c. \quad (7)$$

To find the propagation speed, we use a linear approximation for the ratio of static and dynamic SIF (see Eq. (7.4.5) in [Freund 1998]):

$$v \approx c_R (1 - K_c^2 / K_{\text{eff}}^2), \quad (8)$$

where c_R is the Rayleigh wave speed, which is approximately $c_R \approx 0.9\sqrt{\mu/\rho}$, where ρ is the material’s density [Gross and Seelig 2011].

Finally we compute θ similar to Patricio and Mattheij [2007], who describe the propagation direction for 2D cases. Extending their work to 3D, we group the K_I and K_{III} terms in K_{eff} together, which drive the crack forward, while K_{II} causes the crack to turn. Using $K_{I,III}^2 := K_I^2 + K_{III}^2 / (1 - \nu)$ instead of K_I in Eq. (32) and (34) in [Patricio and Mattheij 2007], we obtain the stress intensity acting across the propagation direction θ :

$$K_\theta = K_{I,III} \cos^3(\theta/2) - 3K_{II} \sin(\theta/2)\cos^2(\theta/2), \quad (9)$$

which has the maximum

$$\theta^* = 2 \arctan \left[\left(K_{I,III} - (K_{I,III}^2 + 8K_{II}^2)^{1/2} \right) / (4K_{II}) \right] \quad (10)$$

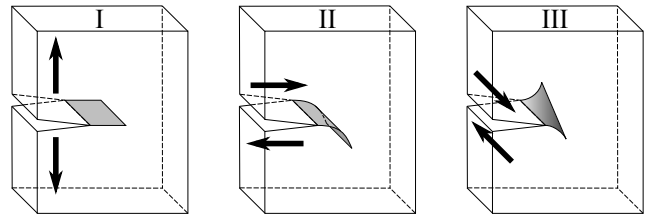


Figure 3: Loading modes (arrows) and expected crack propagation behavior (shaded).

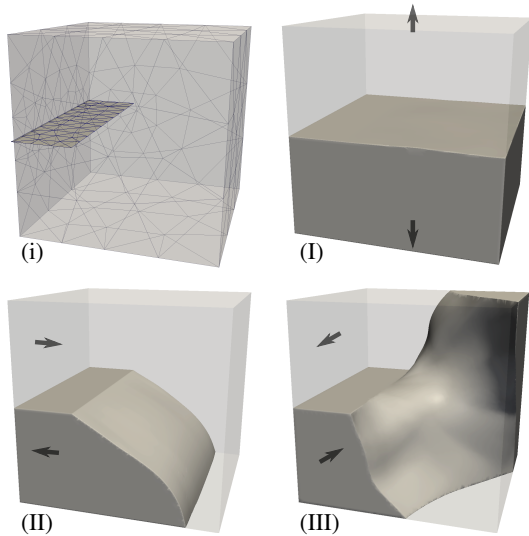


Figure 4: Initial BEM mesh (i) and results (I–III) for a cube containing a planar edge-crack under mode I, II, and III loading.

if $K_{II} \neq 0$, otherwise $\theta^* = 0$. In a homogeneous material, the crack propagates in the direction θ^* ; in §6.2 we extend this to account for spatial variations of the material’s toughness. Please note that the propagation speed in Eq. (8) is limited by the Rayleigh wave speed, which makes it straightforward to choose the time-step such that a maximal distance cannot be exceeded.

6 High-resolution fracture simulation

We could now evaluate the fracture criterion described in §5 at every crack-front node in the BEM mesh, advance the crack-front over a time-step, and then update the BEM by inserting new elements and growing the system matrix as described in §4.1. However, doing this would result in fracture surfaces that are of the same resolution as the BEM mesh. Handling a highly detailed fracture this way would make the BEM prohibitively expensive. In this section, we describe how we simulate high-resolution crack propagation using a coarse elasticity solver.

6.1 Homogeneous material

Our goal is to separate the resolution of crack propagation from the resolution of the underlying elasticity simulation. To this end, we conceptually subdivide each crack-front edge by placing n equally-spaced markers on it. Consequently, our fracture surface will have roughly n -times the resolution of the BEM mesh. We then interpolate both the stress intensities and the local coordinate system linearly over the edge, evaluate the fracture criterion at each of the markers, and propagate them *independently*. To achieve high temporal resolution, we use smaller time-steps than the BEM updates during propagation (called “sub-steps”).

The key point is that the crack propagation criterion is based on stress intensity factors, which describe the stress field in the immediate vicinity of a crack-front (with respect to its local coordinate system). The rapid change of stress around the moving crack-front is therefore captured by applying the SIF at the updated positions of the crack-front markers in each sub-step. The stress far away from the crack-front changes much more slowly (both in time and space), which is accounted for by the BEM update at the end of each full time-step.

So far, we have increased the resolution of the crack-front motion, but the generated fracture surface would still show kinks as θ^* changes from one full BEM time-step to the next (when using a forward Euler scheme). This issue could be solved by using a higher-order time integration, but as recomputing θ^* requires a BEM update, it would be expensive to do so.

Intuitively the goal is to have a *smooth* transition from the previous propagation direction to the new one after each BEM update. To this end, we store the previous direction at each crack-front marker and linearly blend to the new direction over time as we take forward Euler sub-steps. The number of sub-steps is chosen equal to n , i.e. the number of markers per edge. We typically choose n in the range of 10–50 for our examples; see Table 1. This means that we can take about 10–50 sub-steps before needing to update the BEM solution.

After the sub-steps are integrated, we first apply edge subdivision and collapse operations to (almost) uniformly sample the high-resolution crack-front. We then compute a smoothed version of this curve by averaging the positions of every n markers. Finally, we add new triangles to the BEM mesh connecting the previous and the new low-resolution crack-fronts.

6.2 Spatially varying toughness

So far we have considered crack propagation in a linear elastic homogeneous material. While the BEM inherently limits the ability to treat materials with spatially varying elasticity properties, we find that we can produce a number of interesting effects in terms of fracture surfaces by toughness variations only. Note that both strength and toughness are material parameters that do not affect the BEM, whereas variations in the elasticity parameters would require adjusting the fundamental solution.

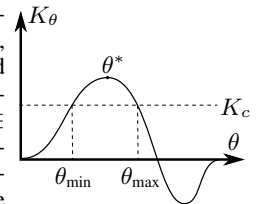
The strategy of propagating a fracture strictly according to stress intensity (along θ^* , Eq. (10)) appropriately yields maximal elastic energy release in homogeneous materials. Unfortunately, this is not true for finite time-steps in materials with spatially varying toughness, as more energy can be released by proceeding through weaker regions (instead of entering tougher regions and stopping earlier). We introduce new crack propagation rules below to compensate for this observation.

To model the effects of toughness variation on the crack’s propagation direction, we use the heuristic that a crack should follow the negative toughness gradient, i.e. move towards a less tough region nearby, whenever possible. Note that, just as in §5.2, each marker on the crack-front may only move in the plane locally orthogonal to the crack-front.

The effective stress intensity (K_θ) depends on the propagation direction according to Eq. (9), and we choose the optimum (θ^*), Eq. (10), if the toughness is (locally) homogeneous.

However, if there is a non-zero toughness gradient at a crack-front sample \mathbf{x} , we instead compute an interval of valid directions using 1D Brent-Dekker root-finding, such that $K_\theta \geq K_c(\mathbf{x}) \forall \theta \in (\theta_{\min}, \theta_{\max})$, as shown in the inset figure. We then project the toughness gradient onto the plane orthogonal to the crack-front and convert to polar coordinates to find the in-plane angle of the gradient ($\tilde{\theta}_\nabla$). We choose the valid direction $\theta_\nabla \in (\theta_{\min}, \theta_{\max})$ closest to the gradient angle $\tilde{\theta}_\nabla$.

To avoid numerical artifacts when the toughness gradient is either close to zero or almost parallel to the crack-front, the current propagation direction is computed as a weighted average



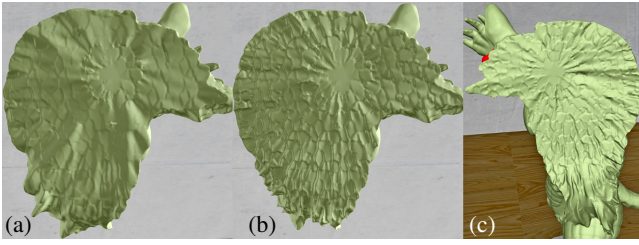


Figure 5: Comparison of the default (a), §6.2, and alternative (b, c), §6.3, crack propagation method using a material model with large elongated (rolled) grains.

$\theta_w = w\theta^* + (1-w)\theta_\nabla$, where the weight is based on the magnitude $\|\nabla K_{c,p}\|$ of the projected toughness gradient:

$$w = 1 / \left(1 + \frac{\|\nabla K_{c,p}\| d}{K_c} \right), \quad (11)$$

and d is a characteristic length, which we choose to be the average BEM edge-length, making the weight dimensionless.

6.3 Alternative method for crack propagation

Note that the method described in §6.2 will prioritize smooth direction changes over producing high-frequency details, making it robust under both very high or almost zero toughness gradients. We use this procedure for all of our results, except for Fig. 5(b, c) and the shattering cube in our video. These two examples follow a more aggressive alternative strategy described in this section, which can yield more visual detail under certain conditions, but is also more prone to creating noticeable artifacts on the fracture surface.

The alternative method proceeds as follows: first, we compute a smoothly changing direction $\tilde{\theta}^*$ ignoring all toughness variations for each sub-step (as in §6.1). We then linearly interpolate the valid interval $(\hat{\theta}_{\min}, \hat{\theta}_{\max})$ between the previous and the current full-step. Finally, we add the direction according to the toughness gradient to find the actual propagation angle $\hat{\theta}_w = w\tilde{\theta}^* + (1-w)\hat{\theta}_\nabla$, where $\hat{\theta}_\nabla$ is the closest direction to the projected toughness gradient $\tilde{\theta}_\nabla$ contained in the interpolated valid interval $(\hat{\theta}_{\min}, \hat{\theta}_{\max})$. A comparison of these two methods is shown in Fig. 5.

7 Implementation details

We use OpenVDB [2014] implicit surfaces to store the high-resolution fractures, as well as the object’s surface, instead of the low-resolution triangle mesh used in the BEM. Fractures are represented by cutting away a thin strip of material, whose thickness is slightly larger than the voxel size, around each crack. This allows us to store the high resolution surfaces, avoid complicated meshing operations, and perform intersection tests quickly in order to determine whether a crack-front has reached another surface.

At the start of a simulation, we perform a pre-computation, starting with loading either a given BEM mesh, or a high-resolution triangle mesh and building the BEM mesh with quadric-error-based simplification [Garland and Heckbert 1997], implemented using VCGlib [2014]. In cases where the high-resolution surface protrudes too far out of the coarse BEM mesh, we slightly offset the latter in the normal direction. Furthermore, we initialize the implicit surface and assemble the BEM system matrix as in Eq. (3). Once this pre-computation is done, each time-step consists of the following steps: compute displacements (§4) and surface stresses,

crack initiation (§5.1), crack propagation using sub-steps (§5.2), BEM mesh update, and assembly of additional blocks in the BEM matrix (§4.1), Eq. (4). Updates to the implicit surface occur after every sub-step for the examples marked with (s) in Table 1 or at the end of each full-step otherwise.

Once the simulation is complete (either a maximal number of time-steps has been done, or no new elements have been added to the BEM), we use a breadth-first-search on the implicit surface to find separate fragments. We then extract a high-resolution mesh of each fragment for rendering.

Visualizing displacements. Since the implicit surface is built in undeformed material-space, and displacements are only computed on the coarse mesh, we need to interpolate displacements from the BEM mesh onto the high-resolution mesh in order to produce a world-space visualization. We first build a coarse OpenVDB grid storing indices of all triangles in the coarse mesh intersecting each voxel. We then use this grid to accelerate closest-triangle queries on the BEM mesh. Note that the displacement field defined by the object’s boundary displacement (\mathbf{u}_Γ) and the fundamental solution is continuous in space. The crack opening displacement ($\Delta\mathbf{u}$) adds the discontinuity on fracture surfaces. Both of these contributions are solved for using Eq. (4) and in combination determine the displacements of both faces of a crack (\mathbf{u}^+ , \mathbf{u}^-) in the visual mesh.

We first perform an interior evaluation of the continuous part (\mathbf{u}_{cont}) at all fracture surface nodes in the BEM mesh, which can be interpreted as the displacement of a crack’s medial axis. For each vertex in the visual mesh, we then find the closest triangle in the coarse mesh and perform linear interpolation of the continuous displacement field within that triangle.

If the vertex is in the vicinity of a crack, we first decide to which face of the crack (Γ_c^+ or Γ_c^-) it corresponds, based on the high-resolution surface normals. We then interpolate the COD and compute either $\mathbf{u}^+ = \mathbf{u}_{\text{cont}} + \Delta\mathbf{u}/2$ or $\mathbf{u}^- = \mathbf{u}_{\text{cont}} - \Delta\mathbf{u}/2$. Finally, we move the vertex to its displaced position.

Animation output. For efficiency reasons, both in terms of runtime and disk space, we write the complete state of the simulation only at full time-steps, whereas we write only updates to the implicit surface at sub-steps. In a post-processing pass we then combine the incremental updates since the last full time-step to build a high-resolution mesh for each of the sub-steps. Displacements are interpolated piecewise linearly over time between the previous and following full time-step. Note that brittle crack propagation is very fast, consequently our time-steps are usually of the order of milliseconds to microseconds, depending on the stiffness of the material; see Eq. (8).

8 Results

In this section we present results produced with our method. We first show some well-known test cases for homogeneous materials to demonstrate that our method produces fracture surfaces in line with the theoretical predictions of LEFM. We then show examples of various material toughness fields and their influence on the fracture surfaces. Finally, we present animations showing both the high temporal resolution of our fractures, as well as rigid-body animations of the resulting fragments, also shown in our accompanying video. These rigid-body dynamics are currently done as a post-process and are not yet integrated into our simulation framework; this is an interesting direction for future work. Our images were rendered using Autodesk Maya[®] or ParaView [2014].

Verification test cases. First, we consider a cube containing a planar edge-crack under mode I, II, and III loading respectively.

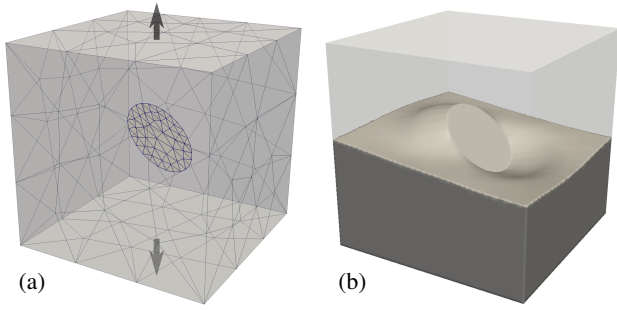


Figure 6: A cube containing a 45° inclined penny-shaped crack is broken under uniform tension: initial BEM mesh (a) and one fragment of the final result (b). The crack smoothly approaches the plane orthogonal to the direction of tension, starting from its prescribed initial configuration.

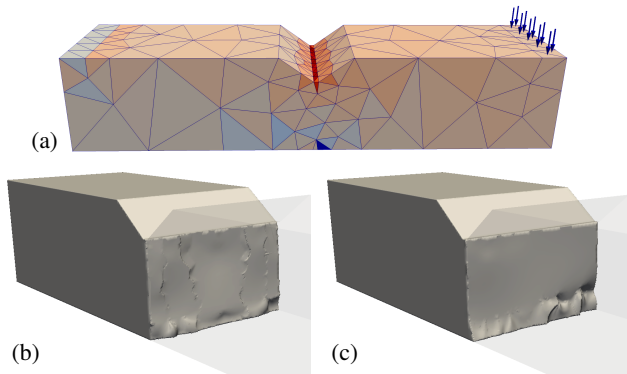


Figure 7: Initial BEM mesh (a) colored by maximal principal stress and results for the 3-point-bending test of a notched bar. Our crack initiation method produces 5 fractures (b), but we can restrict it to a single one (c).

The expected fracture behavior (as described in §5.2 and illustrated in Fig. 3) is captured by the simulation as shown in Fig. 4.

Next, we consider a cube under uniform tension along the vertical axis, containing a penny-shaped crack inclined at 45° to the direction of tension. As shown in Fig. 6, the resulting fracture surface is smooth and approaches the plane orthogonal to the direction of tension. Our method’s output for this standard test case is qualitatively very close to the result of Gravouil et al. [2002].

Finally, we reproduce the well-known 3-point-bending test of a notched bar. In this case there are no fractures in the initial geometry. Instead, the notch creates a stress concentration, which causes cracks to naturally initiate nearby. This test case shows that our crack initiation method agrees with LEFM theory and experiments.

Please note the small ridges in Fig. 7b, which are formed by individual cracks propagating parallel to each other and eventually intersecting. Similar patterns are often found in nature and are sometimes referred to as “rivers” in the literature (some are also visible in the photograph of the fracture in Fig. 11b). Fig. 7c also demonstrates our ability to control the simulation by limiting crack initiation to a single fracture event.

The 3-point-bending test is also one of the few cases where the limitations of the BEM formulation, discussed in §4.1, become visually apparent as small ripples close to the bottom surface of the bar. In that region, COD, and hence SIF are influenced by the nearby sur-

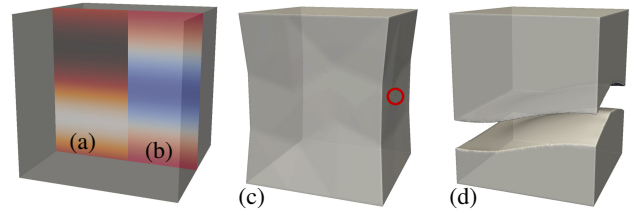


Figure 8: Toughness (a) and strength (b) fields used to control the fracturing of a cube under mode-I tension (c) and the final result (d). Minimal toughness is shown in white, minimal strength in blue. The red circle in (c) highlights the location of crack initiation.

face, causing deviations in the crack-front’s path.

Varying strength and toughness. We now demonstrate how the simulation can be controlled using spatially varying strength and/or toughness fields. First, we consider a cube under uniform vertical tension similar to Fig. 6, but without any initial cracks. We then use the strength field shown in Fig. 8b to bias crack initiation towards the middle of the cube.

In a homogeneous material, the crack would propagate under mode-I loading and form a planar surface, similar to the one in Fig. 4b. However, we use a toughness field, Fig. 8a, to bias crack propagation to a lower, parallel plane. As shown in Fig. 8c and 8d the fracture starts in the low-strength region and smoothly approaches the plane of minimal toughness. We limit crack initiation to a single fracture in order to make the propagation behavior clearly visible.

Having shown that our method produces the expected results in standard test cases and is controllable in the intended way, we will proceed with results obtained using toughness fields with more artistic spatial variation.

Toughness models. While real-world toughness fields may be very complicated and a lot of time could be spent on accurately describing them, the goal of this section is to use simple models in order to achieve realistic looking fracture surfaces.

Our simplest toughness model aims to describe a material composed of many parallel layers. We choose a maximum and a minimum toughness value and blend between the two using a weight of the form $(\cos(\mathbf{k} \cdot \mathbf{x} + c) + 1)/2$ at any point \mathbf{x} , where the constant vector \mathbf{k} defines both orientation and frequency of the layers. Even though this is an oversimplification, it produces interesting results, as shown in Fig. 9: the overall crack orientation as well as the number of fragments are affected. Furthermore, fine-scale stair-step patterns are visible on the fracture surfaces, consistent with the cleavage patterns of layered materials in nature.

Another interesting subject are materials with a granular structure. Such grains appear on a micron scale in many metals, but they also appear on larger scales in rock compounds or concrete, for example. At this point we do not make any assumptions on the scale at which we run our simulations, as both cases may be of interest and their respective fracture patterns exhibit many similarities.

We use the freely available tool DREAM3D [2014] to generate a realistic grain structure. We first generate a surface mesh of the grain boundaries and then assign a toughness value proportional to a narrow-band distance function to the grain boundaries. That is, toughness is minimal at grain boundaries and maximal deep inside each grain.

Results obtained with granular toughness fields are shown in Fig. 1, 10, 11, and 12. In these examples, perturbations due to varying toughness propagate with the crack-front, leaving clearly visible

directional patterns called “chevrons” in their wake. Consequently the propagation direction is discernible in the final fracture surface. Such chevrons and river lines are a signature of brittle fractures in nature [Becker and Lampman 2002]. In addition, surface roughness increases with distance from the initiation site, as expected. To our knowledge this is the first method in computer graphics to produce these physically plausible details.

Furthermore, in Fig. 10b we use material parameters of concrete, which has a relatively high toughness (compared to its strength). Consequently, we see numerous cracks appearing in the simulation, but only a few of these propagate and some stop before reaching another surface. In this example, we chose a slightly transparent visualization to show this behavior.

We compare our result to a real world image in Fig. 11 for a granular material with the elastic properties of concrete. Even though scale and boundary conditions are not matched to the real world example, similar fracture patterns appear. Finally, Fig. 12 demonstrates that our method readily handles non-trivial topology.

9 Discussion

In this paper, we have presented a simulation of brittle fracture dynamics using a boundary element method. We compute stresses on surfaces, but we use stress intensities at crack-fronts (where the stress field is singular). Crack initiation is based on surface stresses and can be controlled by modifying the material’s strength, whereas crack propagation is based on stress intensity and controlled by modifying toughness. This separation allows us to avoid artificial shattering often seen in purely stress-based fracture simulations; it also enables us to produce highly detailed fractures due to spatially varying toughness.

We sample the crack-front at a higher resolution than the BEM mesh when simulating crack propagation. The path of the propagating crack-front defines the fracture surface. This allows us to increase the resolution of the fracture surface compared to the BEM mesh.

Consequently our method is capable of producing highly detailed results in a reasonably short time on commodity laptop hardware (see Table 1). While one could quickly add visual detail to coarse fracture surfaces in a post-processing step, such a process lacks the realism that our results exhibit, as many brittle fracture patterns

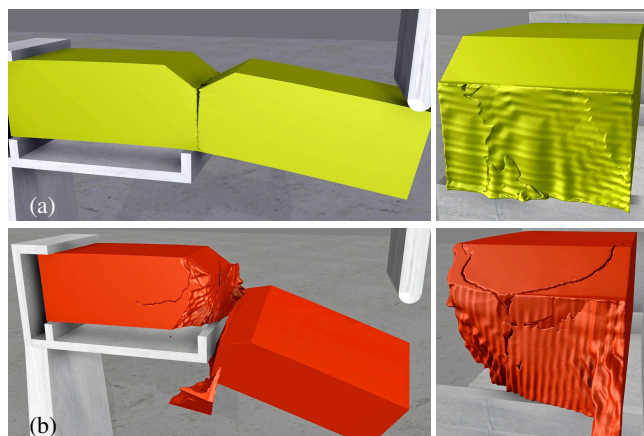


Figure 9: The 3-point bending test, using the same parameters as Fig. 7, except for a layered toughness field oriented horizontally (a) or vertically (b). This variation affects the number of fragments.

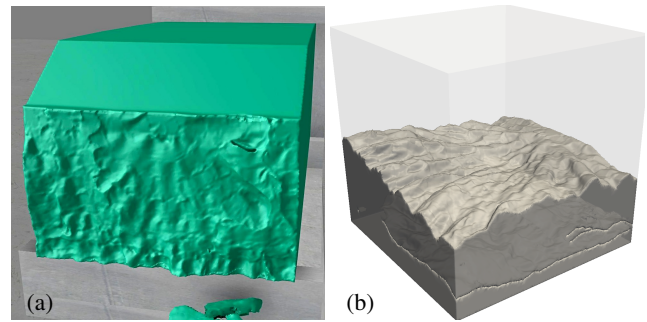


Figure 10: Results generated with granular toughness fields. Please note the chevrons indicating the propagation direction.

found in nature arise from the direction of crack propagation; also the topology of the fragments would not change, as opposed to our results (see Fig. 9 and our video).

Since our method is based on quasi-static linear elasticity, it is inherently inadequate for large displacements and ductile fracture. However, it is well suited for handling brittle fracture in reasonably stiff materials. In such materials, fractures typically propagate very fast—on the order of 1km/s in PMMA (“acrylic glass”) and concrete—and hence a high temporal resolution is required to trace the crack-front. Our sub-stepping scheme avoids the need to update the BEM solution for every step; instead we take a fixed number of sub-steps before the update (at the end of the full time-step). During sub-stepping we assume that the stress intensity factors (SIF) remain constant. These SIF are specified in the crack-front’s local coordinate frame, which automatically handles the fact that the stress field in the vicinity of the crack-front shifts as the front propagates. Contrary to our method, Pfaff et al. [2014] require local updates to model this effect.

While we choose small time-steps to achieve good visual detail, our method is fairly stable at larger time-steps. However, using extremely large time-steps may cause intersecting elements in the BEM mesh, which means that there are quasi-singular integrals during matrix assembly. Furthermore, large time-steps (and therefore large fracture surface elements) may cause “incomplete cracks”, if the high-resolution geometry is not sufficiently well represented in the BEM. Consequently, small gaps would remain in the implicit surface, connecting two components that should be separate fragments. In our current implementation we remove these artifacts by eroding the level-set by at most one voxel and then segmenting the eroded level-set; we undo the erosion once fragments are separated.

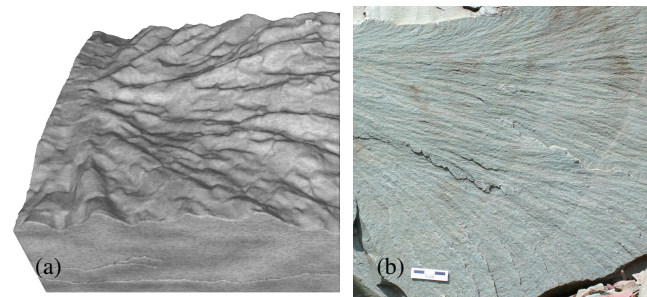


Figure 11: Simulation result (a) vs. photograph of a fracture (b) for a granular material. Photograph (b) by Michael C. Rygel via Wikimedia Commons (cropped).

Table 1: Performance overview of our method. Columns: ($\#tris_i$) initial and final number of triangles in the BEM mesh; ($\#voxels$) dimensions of the implicit surface’s bounding box in voxel units; ($\#ts$) number of time-steps; ($\#ss$) number of sub-steps; ($\#ci$) max. number of crack initiations, (u) unlimited, (m) only one manually added crack; (t_s) CPU time for initial implicit surface construction; (t_a) CPU time for initial BEM matrix assembly; (t_{ts}) average computation time per time-step; (t_t) total computation time of the simulation (including I/O and initial BEM mesh generation); (τ) total simulated time. All timings are given in seconds and measured on an Intel[®] Core[™]i5 2.6GHz dual-core HT CPU. Examples marked with (s) generate sub-step output for slow motion animations.

Example	$\#tris_i$	$\#tris_f$	$\#voxels$ (x y z)			$\#ts$	$\#ss$	$\#ci$	t_s	t_a	t_{ts}	t_t	τ
1 Armadillo lay. (video) (s)	1000	1128	429	510	391	5	16	u	6.02	8.91	4.57	40.29	2.8E-5
2 Armadillo grains (Fig. 1)	1000	1845	641	763	583	12	15	6	10.07	8.52	5.79	92.35	4.1E-5
3 Armadillo split (video)	1000	1160	641	763	583	9	50	m	10.07	7.76	3.00	49.04	1.0E-4
4 Bunny ears (video)	900	1084	785	778	609	7	25	u	8.80	6.09	2.84	38.53	1.8E-5
5 Bunny fragile (video)	1000	5380	318	314	247	27	14	12	3.35	7.69	31.94	874.43	2.1E-4
6 Bunny grains (video)	1000	2680	318	314	247	17	20	6	2.18	7.00	11.61	207.35	1.9E-4
7 Bunny smash (video) (s)	1000	2228	318	314	247	18	20	6	2.25	6.96	16.73	311.21	2.0E-4
8 Column (video)	1000	4613	183	617	181	28	12	17	2.25	19.98	20.15	590.62	3.2E-3
9 Cube 45° crack (Fig. 6) (s)	244	852	505	505	505	10	30	m	5.10	1.00	9.03	96.42	6.7E-4
10 Cube concrete (Fig. 10b)	120	1890	339	339	339	24	33	u	2.73	0.68	7.23	177.00	1.3E-3
10 Cube shatter (video)	120	2421	233	233	233	12	41	40	1.22	0.63	22.46	271.32	1.1E-3
11 Cube controlled (Fig. 8)	120	391	205	205	205	12	20	1	0.73	0.84	0.75	10.60	1.4E-3
12 Cube mode I (Fig. 4I)	312	424	339	339	339	6	50	m	3.05	1.64	1.64	14.52	1.0E-3
12 Cube mode II (Fig. 4II)	312	430	339	339	339	6	50	m	2.92	1.39	1.44	12.96	1.0E-3
12 Cube mode III (Fig. 4III)	312	459	339	339	339	7	50	m	2.99	1.52	1.58	15.56	1.2E-3
13 Horses statue (Fig. 12)	1000	3134	419	349	149	39	20	u	2.34	8.94	17.48	695.00	1.1E-3
14 Notched bar (Fig. 7b)	416	943	1005	205	255	11	20	u	3.38	1.94	2.97	37.98	5.0E-5
14 N. bar, 1 crack (Fig. 7c)	416	764	1005	205	255	18	20	1	3.39	1.82	1.35	29.51	8.1E-5
15 N. bar lay. horiz. (Fig. 9a)	416	691	1005	205	255	8	30	3	3.49	1.92	2.32	23.94	5.4E-5
15 N. bar lay. vert. (Fig. 9b)	416	1212	1005	205	255	12	30	3	3.28	1.79	3.93	52.25	8.1E-5
16 N. bar grains (Fig. 10a)	416	1043	1005	205	255	13	20	5	3.58	1.87	3.63	52.69	5.9E-5

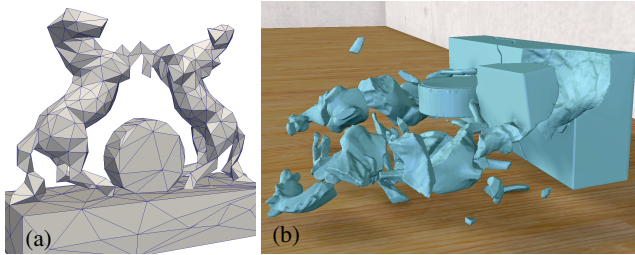


Figure 12: Initial BEM mesh (a) and resulting fragments (b) of a statue dropped on the floor.

Finally, we would like to point out some interesting directions for future work. The scaling of the BEM can be improved by using fast multi-pole methods or adaptive cross approximation. One could also consider using a time-domain BEM [Messner and Schanz 2010]. While we emphasize that we do not need to change existing elements during our fracture simulation, it could be beneficial to simplify the mesh of a fracture surface once the crack-front has moved on, keeping the number of degrees of freedom as low as possible.

While it is difficult to directly compare to volumetric finite element methods, our experiments showed that a single FEM based elastostatic solution for the edge-cracked cube example in Fig. 4i takes about 0.2s with an adaptive mesh resolution similar to our BEM mesh. (These times were obtained with the open source code at <http://elmerfem.org>.) Our solution (Fig. 4I) takes 300 sub-steps in 14.5s (on the same machine; see Table 1), which amortizes the BEM cost to 0.05s per sub-step. While this comparison is by no means exhaustive, it indicates that our approach is a promising alternative to volumetric FEM.

To summarize, we have presented a new technique for simulating brittle fracture based on boundary elements and Lagrangian crack-front propagation. We have also provided the first method in computer graphics for computing fractures in the presence of high-resolution material toughness variations. Because it focuses the computational effort directly on the high-resolution crack-front geometry, our method is capable of efficiently generating extremely detailed and physically plausible fracture patterns.

Acknowledgments

We cordially thank everyone who contributed to this paper, especially Prof. M. Schanz and his group for providing and supporting the HyENA library, the OpenVDB online community, as well as all proof-readers and anonymous reviewers.

This work was funded by IST Austria and ERC Grant 638176.

References

- ABDELAZIZ, Y., AND HAMOUINE, A. 2008. A survey of the extended finite element. *Computers & Structures* 86, 1141–1151.
- ALIABADI, M. 1997. Boundary element formulations in fracture mechanics. *Applied Mechanics Reviews* 50, 2, 83–96.
- ANDERSON, T. L. 2005. *Fracture mechanics: fundamentals and applications*. CRC press.
- BAO, Z., HONG, J.-M., TERAN, J., AND FEDKIW, R. 2007. Fracturing rigid materials. *TVCG* 13, 370–378.
- BECKER, W., AND LAMPMAN, S. 2002. Fracture appearance and mechanisms of deformation and fracture. *Materials Park, OH: ASM International 2002*, 559–586.

- CHEN, Z., YAO, M., FENG, R., AND WANG, H. 2014. Physics-inspired adaptive fracture refinement. *ACM Trans. Graph.* 33, 113:1–113:7.
- DREAM3D, 2014. A Digital Representation Environment for the Analysis of Microstructure in 3D. v. 4.2.4. <http://dream3d.bluequartz.net>.
- FRANGI, A., NOVATI, G., SPRINGHETTI, R., AND ROVIZZI, M. 2002. 3D fracture analysis by the symmetric Galerkin BEM. *Computational Mechanics* 28, 3–4, 220–232.
- FREUND, L. B. 1998. *Dynamic Fracture Mechanics*. Cambridge Monographs on Mechanics. Cambridge University Press.
- GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH 97, Annual Conference Series*, 209–216.
- GLONDU, L., MUGUERCA, L., MARCHAL, M., BOSCH, C., RUSHMEIER, H., DUMONT, G., AND DRETTAKIS, G. 2012. Example-based fractured appearance. *Computer Graphics Forum* 31, 4, 1547–1556.
- GLONDU, L., MARCHAL, M., AND DUMONT, G. 2013. Real-time simulation of brittle fracture using modal analysis. *TVCG* 19, 201–209.
- GRAVOUIL, A., MOËS, N., AND BELYTSCHKO, T. 2002. Non-planar 3D crack growth by the extended finite element and level sets – part II: level set update. *INT J NUMER METH ENG* 53, 11, 2569–2586.
- GROSS, D., AND SEELIG, T. 2011. *Fracture Mechanics*, 2nd ed. Springer.
- HIROTA, K., TANOUE, Y., AND KANEKO, T. 1998. Generation of crack patterns with a physical model. *The Visual Computer* 14, 126–137.
- HIROTA, K., TANOUE, Y., AND KANEKO, T. 2000. Simulation of three-dimensional cracks. *The Visual Computer* 16, 371–378.
- HYENA, 2013. Hyperbolic and Elliptic Numerical Analysis. Graz University of Technology. <http://www.mech.tugraz.at/HyENA> v. 2013-Nov-04.
- IBEN, H. N., AND O'BRIEN, J. F. 2006. Generating surface crack patterns. In *ACM SIGGRAPH/Eurographics SCA 2006*, 177–185.
- INGRAFFEA, A. R., AND WAWRZYNEK, P. A. 2003. *Finite Element Methods for Linear Elastic Fracture Mechanics, Chapter 3.1 in Comprehensive Structural Integrity*. Elsevier Science Ltd.
- IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. In *ACM SIGGRAPH/Eurographics SCA 2004*, 131–140.
- JAMES, D. L., AND PAI, D. K. 1999. Artdefo: Accurate real time deformable objects. In *SIGGRAPH 99, Annual Conference Series*, 65–72.
- KAUFMANN, P., MARTIN, S., BOTSCH, M., GRINSPUN, E., AND GROSS, M. 2009. Enrichment textures for detailed cutting of shells. *ACM Trans. Graph.* 28, 50:1–50:10.
- KEELER, T., AND BRIDSON, R. 2014. Ocean waves animation using boundary integral equations and explicit mesh tracking. In *ACM SIGGRAPH/Eurographics SCA 2014*, 11–19.
- KIELHORN, L. 2009. *A time-domain symmetric Galerkin BEM for viscoelastodynamics*. Verl. der Techn. Univ. Graz.
- KOSCHIER, D., LIPPONER, S., AND BENDER, J. 2014. Adaptive tetrahedral meshes for brittle fracture simulation. In *ACM SIGGRAPH/Eurographics SCA 2014*.
- LEVINE, J. A., BARGTEIL, A. W., CORSI, C., TESSENDORF, J., AND GEIST, R. 2014. A peridynamic perspective on spring-mass fracture. In *ACM SIGGRAPH/Eurographics SCA 2014*.
- MESSNER, M., AND SCHANZ, M. 2010. An accelerated symmetric time-domain boundary element formulation for elasticity. *Engineering Analysis with Boundary Elements* 34, 11, 944–955.
- MOËS, N., GRAVOUIL, A., AND BELYTSCHKO, T. 2002. Non-planar 3D crack growth by the extended finite element and level sets – part I: mechanical model. *INT J NUMER METH ENG* 53, 11, 2549–2568.
- MOLINO, N., BAO, Z., AND FEDKIW, R. 2004. A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph.* 23, 385–392.
- MOUSAVI, S. E., GRINSPUN, E., AND SUKUMAR, N. 2011. Higher-order extended finite elements with harmonic enrichment functions for complex crack problems. *INT J NUMER METH ENG* 86, 4–5, 560–574.
- MÜLLER, M., MCMILLAN, L., DORSEY, J., AND JAGNOW, R. 2001. Real-time simulation of deformation and fracture of stiff materials. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 113–124.
- MÜLLER, M., CHENTANEZ, N., AND KIM, T.-Y. 2013. Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Trans. Graph.* 32, 115:1–115:10.
- NESME, M., KRY, P. G., JEŘÁBKOVÁ, L., AND FAURE, F. 2009. Preserving topology and elasticity for embedded deformable models. *ACM Trans. Graph.* 28, 52:1–52:9.
- NORTON, A., TURK, G., BACON, B., GERTH, J., AND SWEENEY, P. 1991. Animation of fracture by physical modeling. *The Visual Computer* 7, 210–219.
- O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *SIGGRAPH 99, Annual Conference Series*, 137–146.
- O'BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. *ACM Trans. Graph.* 21, 291–294.
- OPENVDB, 2014. DreamWorks Animation. v. 2.2.0. <http://www.openvdb.org>.
- PARAVIEW, 2014. Kitware Inc. v. 4.1.0. <http://www.paraview.org>.
- PATRICIO, M., AND MATTHEIJ, R. 2007. Crack propagation analysis. *CASA-report 0723*.
- PAULY, M., KEISER, R., ADAMS, B., DUTRÉ, P., GROSS, M., AND GUIBAS, L. J. 2005. Meshless animation of fracturing solids. *ACM Trans. Graph.* 24, 957–964.
- PFÄFF, T., NARAIN, R., DE JOYA, J. M., AND O'BRIEN, J. F. 2014. Adaptive tearing and cracking of thin sheets. *ACM Trans. Graph.* 33, 110:1–110:9.
- PORTELA, A., ALIABADI, M. H., AND ROOKE, D. P. 1992. The dual boundary element method: effective implementation for crack problems. *INT J NUMER METH ENG* 33, 6, 1269–1287.

- RABCZUK, T. 2013. Computational methods for fracture in brittle and quasi-brittle solids: State-of-the-art review and future perspectives. *ISRN Applied Mathematics 2013*, Article ID 849231.
- SAUTER, S. A., AND SCHWAB, C. 2011. *Boundary Element Methods*. Springer Berlin Heidelberg.
- SCHVARTZMAN, S. C., AND OTADUY, M. A. 2014. Fracture animation based on high-dimensional voronoi diagrams. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 15–22.
- SMITH, J., WITKIN, A., AND BARAFF, D. 2001. Fast and controllable simulation of the shattering of brittle objects. *Computer Graphics Forum 20*, 81–91.
- STOMAKHIN, A., SCHROEDER, C., CHAI, L., TERAN, J., AND SELLE, A. 2013. A material point method for snow simulation. *ACM Trans. Graph.* 32, 102:1–102:10.
- STOMAKHIN, A., SCHROEDER, C., JIANG, C., CHAI, L., TERAN, J., AND SELLE, A. 2014. Augmented mpm for phase-change and varied materials. *ACM Trans. Graph.* 33, 138:1–138:11.
- SU, J., SCHROEDER, C., AND FEDKIW, R. 2009. Energy stability and fracture for frame rate rigid body simulations. In *ACM SIGGRAPH/Eurographics SCA 2009*, 155–164.
- SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph.* 23, 399–405.
- SUTRADHAR, A., PAULINO, G. H., AND GRAY, L. J. 2008. *Symmetric Galerkin boundary element method*. Springer.
- TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. *SIGGRAPH Comput. Graph.* 22, 269–278.
- VCGLIB, 2014. The VCG Library, Visual Computing Lab, CNR-ISTI. <http://vcg.isti.cnr.it/~cignoni/newvcglib/html>.
- WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph.* 29, 49:1–49:11.
- WILDE, A. J., AND ALIABADI, M. H. 1999. A 3-d dual BEM formulation for the analysis of crack growth. *Computational Mechanics* 23, 3, 250–257.
- WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph.* 27, 47:1–47:8.
- ZHENG, C., AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. *ACM Trans. Graph.* 29, 69:1–69:13.

Appendix

The entries of the system matrix blocks in Eq. (3) are defined as follows, cf. Eq. (5.17) in [Kielhorn 2009]:

$$\begin{aligned} \mathbf{V}_{i,k} &= \int_{\Gamma} \varphi_i(\mathbf{x}) \int_{\Gamma} U(\mathbf{y} - \mathbf{x}) \varphi_k(\mathbf{y}) ds_{\mathbf{y}} ds_{\mathbf{x}} \\ \mathbf{K}_{i,l} &= \int_{\Gamma} \varphi_i(\mathbf{x}) \int_{\Gamma} (\mathcal{T}_{\mathbf{y}} U)^{\top}(\mathbf{y} - \mathbf{x}) \psi_l(\mathbf{y}) ds_{\mathbf{y}} ds_{\mathbf{x}} \\ \mathbf{D}_{j,k} &= \int_{\Gamma} \psi_j(\mathbf{x}) \mathcal{T}_{\mathbf{x}} \int_{\Gamma} (\mathcal{T}_{\mathbf{y}} U)^{\top}(\mathbf{y} - \mathbf{x}) \psi_l(\mathbf{y}) ds_{\mathbf{y}} ds_{\mathbf{x}}. \end{aligned} \quad (12)$$

Table 2: Material parameters used in our examples: Young’s modulus (E), Poisson’s ratio (ν), density (ρ), tensile strength (S_c), and toughness (K_c). Refer to Table 1 for the examples’ numbering. Strength is omitted for examples where crack initiation has been disabled. Please note that the visual appearance of the results depends not only on the magnitude of toughness variation, but also on its spatial frequency. (*) Strength varies by +/-82% in the controlled cube example (#11).

#	E	ν	ρ	UTS	K_c
1	3.1E+9	0.327	1200	7.6E+7	1.3E+6 +/-23%
2	3.1E+9	0.327	1200	7.6E+7	1.0E+6 +90%
3	3.1E+9	0.327	1200	7.6E+7	1.0E+6 +10%
4	2.5E+10	0.2	2300	3.0E+6	8.0E+5 +/-63%
5	3.1E+9	0.327	1200	7.6E+7	1.0E+4
6	3.1E+9	0.327	1200	7.6E+7	5.0E+5 +80%
7	3.1E+9	0.327	1200	7.6E+7	1.0E+6
8	3.1E+9	0.327	1200	7.6E+7	1.0E+4 +20%
9	3.1E+9	0.3	1200	-	1.0E+4
10	2.5E+10	0.2	2300	3.0E+6	5.0E+5 +10%
11	3.1E+9	0.327	1200	4.2E+7*	1.3E+6 +/-60%
12	3.1E+9	0.327	1200	-	1.0E+6
13	3.1E+9	0.327	1200	7.6E+7	1.0E+5 +/-10%
14	3.1E+9	0.327	1200	2.8E+8	1.0E+6
15	3.1E+9	0.327	1200	7.6E+7	2.6E+4 +/-96%
16	3.1E+9	0.327	1200	7.6E+7	1.0E+6 +/-10%

We use piecewise-linear shape functions, ψ_j , for boundary displacements and piecewise-constant shape functions, φ_i , for boundary tractions. Note that these shape-functions have compact support (i.e. each is non-zero only in a small area), hence the integrals need to be evaluated only over a small patch of the surface, analogous to a Galerkin FEM discretization.

The additional blocks in Eq. (4) are defined in the same way, except that the inner integrals in \mathbf{K}_c and \mathbf{D}_c are evaluated on fractures rather than regular surfaces. Similarly both integrals in \mathbf{D}_{cc} are evaluated on fracture surfaces. All integrals are evaluated numerically based on Gaussian quadrature, as implemented in the HyENA library. The quadrature order o_q depends on the distance $\|\mathbf{y} - \mathbf{x}\|$: $o_q = 2$ for all points that are further than five times the size of the element containing \mathbf{x} , $o_q = 3$ if the distance is between five and one element sizes, and $o_q = 4$ for all points that are closer than the element size.

The right hand side of Eq. (3) is assembled using the known per-node displacements \mathbf{g}_D and known per-element tractions \mathbf{g}_N as follows:

$$\begin{aligned} \mathbf{f}_{D_i} &= \int_{\Gamma} \varphi_i(\mathbf{x}) \left[\frac{1}{2} \psi_l(\mathbf{x}) \mathbf{g}_{Dl} + \int_{\Gamma} (\mathcal{T}_{\mathbf{y}} \mathbf{U})^{\top}(\mathbf{y} - \mathbf{x}) \psi_l(\mathbf{y}) \mathbf{g}_{Dl} ds_{\mathbf{y}} \right. \\ &\quad \left. - \int_{\Gamma} \mathbf{U}(\mathbf{y} - \mathbf{x}) \varphi_k(\mathbf{y}) \mathbf{g}_{Nk} ds_{\mathbf{y}} \right] ds_{\mathbf{x}} \end{aligned} \quad (13)$$

and

$$\begin{aligned} \mathbf{f}_{N_j} &= \int_{\Gamma} \psi_j(\mathbf{x}) \left[\frac{1}{2} \varphi_k(\mathbf{x}) \mathbf{g}_{Nk} + \int_{\Gamma} (\mathcal{T}_{\mathbf{y}} \mathbf{U})(\mathbf{y} - \mathbf{x}) \varphi_k(\mathbf{y}) \mathbf{g}_{Nk} ds_{\mathbf{y}} \right. \\ &\quad \left. - \mathcal{T}_{\mathbf{x}} \int_{\Gamma} (\mathcal{T}_{\mathbf{y}} \mathbf{U})^{\top}(\mathbf{y} - \mathbf{x}) \psi_l(\mathbf{y}) \mathbf{g}_{Dl} ds_{\mathbf{y}} \right] ds_{\mathbf{x}}. \end{aligned} \quad (14)$$