

Fracture and impulse based finite-discrete element modeling of fragmentation

A. Paluszny · X. H. Tang · R. W. Zimmerman

Received: 28 August 2012 / Accepted: 17 April 2013 / Published online: 1 May 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract A numerical method for fragmentation is presented that combines the finite element method with the impulse-based discrete element method (impulse-based FDEM). In contrast to existing methods, fragments are not represented as a conglomeration of spheres; instead, their shapes are represented using solid modeling techniques, and are the result of multiple fracture growth. Fracture growth within each three-dimensional fragment is controlled by stress intensity factors computed using the finite element method and the reduced virtual integration technique. Non-convex fragment interaction and movement is modeled using impulse dynamics, rather than a penalty-based method. Collisions leading to fracture are handled individually by propagating pre-existing internal flaws and cracks. The method utilizes decoupled geometry and mesh representation, and local failure and propagation criteria. Fractures that reach volume boundaries lead to further fragmentation. The approach is demonstrated by the fragmentation of a sphere, which exhibits a velocity-dependent fragment size distribution. The distribution is characterized by a two-parameter Weibull distribution, an emergent property of the simulation. Results are in good agreement with experimental data.

Keywords Fracture · Fragmentation · Finite element · Arbitrary mesh · Impulse method · Weibull distribution

1 Introduction

Understanding the relationship between fracture and fragmentation is one of the open questions in computational fracture mechanics. Feedback loops between the two processes strongly influence the behavior of materials and structures, with applications to geomaterials, explosives, mining engineering, and biomechanics.

Fragmentation simulation involves capturing the interaction between fragments, and fracture growth within individual fragments. The first process requires modeling motion, collision detection, force transfer due to impact and compression, and energy loss during collision. The second process includes defining meso-scale material properties and rock heterogeneities, crack nucleation, and simultaneous propagation of multiple cracks. Challenges of a combined approach include accurate geometric representation of fragments and cracks, accurate mechanical computations of particle motion and energy transfer, efficient processing of large volumes of data, and incorporating dynamically defined boundary conditions. Existing numerical methods generally treat this problem using discontinuous mechanics principles.

The discrete element method (DEM) [1–3] has been used to simulate fragmentation [4–6], and to model fracture propagation [7, 8]. DEM is well suited to model multi-body dynamics. However, its disadvantages include difficulty in capturing complex geometries, such as high aspect ratio fractures, using spherical particles; requiring micro-to-macro calibration of material properties; and, relying on the costly estimation of penetration depth and volume for penalty computations. Discontinuous deformation analysis [9] is an extension of the DEM that is focused on calculating the interaction of discrete blocks along discontinuities for problems of fractured and jointed rock masses. Its advantages include the fact that the equations of motion are always satisfied, stabil-

A. Paluszny (✉) · X. H. Tang · R. W. Zimmerman
Department of Earth Science and Engineering, Imperial College,
London, UK
e-mail: apaluszn@imperial.ac.uk

ity is ensured without requiring additional damping parameters, and penetration and traction are avoided at each step. Methods that combine the finite element method and DEM have been applied to deform individual fragments [10]. Early work focused on hybrid finite-discrete methods (FDEM) to model fragmentation due to rock blasting using spherical elements [11–13]. Recent work includes the combined modeling of wave propagation and plastic deformation of spherical conglomerates due to collision [14,15]. Fracturing within fragments has been modeled using a local smeared crack approach [16] and a local single-crack approach [17]. FEM is usually implemented for the hardening part of the stress-strain curve, and is used before the material reaches its tensile strength. In FDEM, fragmentation does not usually rely on the use of failure criteria to drive propagation. Instead, fracture surface creation is often a function of bond breakage between DEM particles [18], or is based on the partition of individual fragments using simple breakage lines or planes [19]. FDEM models with discrete fracture representation have been proposed, with fracture growth as a result of the evaluation of a strain-softening, anisotropic damage, rotating crack failure criterion [20]. The previous approaches often lead to patterns with excessive branching and mesh dependence, leading to mesh adaptive methods [21]. FDEM-based crack growth has shown to be mesh-size sensitive [22], as it directly depends on the accuracy of the local stress/strain field, and does not rely on an energy-based growth approach. A further disadvantage of DEM-based methods is that they require the micro-properties of the individual spheres to be calibrated against the mechanical macro-properties of materials [23].

As opposed to DEM, the impulse method [24] is an efficient alternative that handles collisions between fragments geometrically, based on polyhedral tracing of the bodies, whereby object trajectories are used to estimate time-of-impact [25], and collision response is a function of impulse resolution [26]. The method has been shown to be energy conservative [27], and relies on meso-scale material properties for the calculation of the impulse-momentum form of Newton's second law to compute collision response within a multi-body system. In contrast to penalty-based methods, objects are not assumed to penetrate during the simulation, and penetration distances and overlapping volumes need not be computed. The impulse method has been applied to simulate sensorless manipulation of objects [28], to simulate haptic interfaces in robotics [29], to model motion of deformable joints in the context of bioengineering [30,31], among others.

The simulation of crack propagation using the finite element method has been shown to be well suited to reproduce fracture propagation in 3D [32–34]. Fractures, in the context of fragmentation, benefit from a discrete representation, as they inherently define fragment boundaries as the domain subdivides. Early versions of the application of FEM

to crack propagation relied on tagging and deleting “broken” mesh elements [35,36]. Fractures can be defined as entities within the mesh to avoid re-meshing (e.g. smeared crack and anisotropic damage model). In general, methods that rely on fixed meshes require them to be sufficiently refined to capture stress singularities that may ensue during the simulation. Other methods aim to reduce complexity by representing only the boundaries of the bodies, but are less suited to capture heterogeneity in the matrix (e.g., the boundary element method). Mesh-free methods [37,38] bypass meshing completely and define domains as sets of points, which introduces difficulties such as domain interface blurring, and costly computational operations. The Extended Finite Element (XFEM) avoids re-meshing by defining enrichment functions to represent discrete fractures at a sub-grid level. XFEM has been applied to grow [39] and intersect [40] fractures in 2D and 3D [41], including the modeling of intra-fracture friction [42]. XFEM has difficulties representing intra-element fracture intersection, or multiple fractures in one element, and requires special approaches to handle fracture tips within elements. The cracking particle method is a numerical method specific to the simulation of crack propagation, in which crack surfaces are represented by a set of point-based enriched functions [43], with no explicit definition of fracture surfaces. Fragmentation relying on this method usually requires re-interpretation of fracture geometry during growth. Finite element-based fracture propagation has been limited in the past by the technology of controlling geometry and mesh generation, but is now facilitated by NURBS-based geometric representation and mature mesh generation technology cf. [44–47]. In the present work, a discrete fracture growth method is used, which represents fracture geometry explicitly using solid modeling techniques, and for which the mesh is only an instrument to enable finite element-based computation of stress intensity factors.

An approach to simulate fragmentation that combines finite element method based fracturing and impulse dynamics and collision will now be presented (cf. Fig. 1). The main novelty of this combined method is that fracturing is not handled as a loss of cohesion between elements; instead, fractures have an explicit geometric representation, and their growth is driven by stress concentrations at the tips. Fracturing is modeled using a continuous approach, while fragmentation is controlled by the discontinuous impulse method. Advantages of this approach include discrete fracture and fragment polyhedral representation, finite element-based stress, strain and stress intensity factors computation, and element-wise definition of standard mechanics material properties such as elastic constants and material toughness. It exploits the impulse-based method for collision detection and resolution of dynamic behavior. Collisions are predicted based on the geometry of the bodies and their physical behavior, as opposed to a posteriori correction based on mesh and object

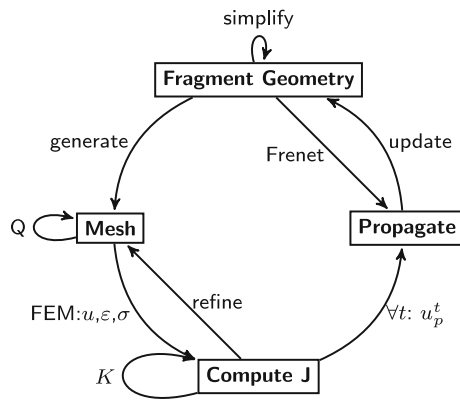


Fig. 1 Fracture-driven fragmentation. Fracture geometry evolves as a function of propagation vectors, u_p^t , which is the result of evaluating propagation and failure laws based on stress intensity factors, K , approximated at its tips, t . Mesh quality Q is a function of element shape and size relationship to tip proximity. The finite element method is applied to solve for d , displacements, σ , stresses, and, ε , strains are derived by approximating that the medium is linear elastic and isotropic. Frenet frames are computed at the tip locations to determine the local coordinate system of the fracture

penetration. The method is demonstrated with the fragmentation of a sphere impacted at a range of velocities. The resulting cumulative volume distributions are characterized by a two-parameter Weibull equation [48]. The method is able to reproduce Weibull fragment size distributions measured experimentally; these distributions are therefore an emergent property of the model.

2 Fragmentation

The main objective of this work is to simulate fracture, \mathcal{F} , and interaction, \mathcal{G} , of a set of fragments $\{H\}$. The interaction of the fragments is a function of

$$\mathcal{G}(\{H\}, \{\alpha(H)\}, g, \rho, \mu, e) = \{c\} \quad (1)$$

where g is gravity, ρ is the rock density, μ is friction, e describes non-fracture energy loss during collision, $\{\mathcal{F}(H)\}$ is the fracturing of individual fragments, and $\{c\}$ is a set of ensuing collisions, each defined by a set of impulses at a set of locations

$$\{c_i\} = \{(\mathbf{p}_i, l_i)\} \quad (2)$$

where \mathbf{p}_i is the impulse vector and l_i is the impulse location, and

$$\mathbf{p}_i = m_i \Delta \mathbf{v}_i \quad (3)$$

where m_i is the object mass and \mathbf{v}_i is its velocity, defined at discrete locations by contributions from the linear and angular velocities of the colliding bodies. At each contact location, the contact force is defined as $\mathbf{f}_i = d\mathbf{p}_i/dt$. Fragmentation is driven by fracture growth:

$$\{H_i^n\} = \mathcal{F}(H_i, \mathbf{f}_i, E, \nu, K_{IC}) \quad (4)$$

where $\{H_i^n\}$ are new fragments created during growth, E is Young's modulus, ν is Poisson's ratio, and K_{IC} is the material toughness. Fracture growth simulation involves the propagation of multiple fractures within a single 3D body of arbitrary shape (see Sect. 6).

Each fragment is populated by a set of initial flaws, f^o , representing inclusions, defects, and initial fractures [49], part of the material characterization [50]:

$$\forall H \exists \{f_i^o\}_h \longrightarrow \text{center}(f_i^o) = \text{random}() \quad (5)$$

where $\{f_i^o\}_h$ is a set of flaws, in fragment h , of varying radius r_i . Position candidates are computed using a Mersenne Twister random number calculator. If any point in the flaw is closer than a minimal distance to a previously inserted flaw, the insertion fails. This minimal distance is referred to as the *spacing*. It follows that

$$\forall f_i \forall f_{i'} |\overline{f_i f_{i'}}| > d_{min} \quad (6)$$

where d_{min} is the minimum separating distance, and

$$d_{min} > \max(r_i, r_{i'}) + s_{max} \quad (7)$$

where s_{max} is the maximum spacing between two fractures. The flaw density is decreased by assigning a larger s_{max} , its upper limit is defined by the minimum edge length of the mesh. Choosing a larger s_{max} will create swarms of evenly spaced flaws, while a smaller value yields flaw clustering. Radii are assumed to have a Gaussian size distribution.

As the simulation progresses, collisions are classified by magnitude, and are individually handled as fracture-inducing events. Thus, the simulation of fracture and fragment interaction is computationally separated, and are only coupled by the continuous transfer of force and geometric information. The fragmentation method is summarized by Algorithm 1.

Algorithm 1 FRAGMENTATION(n)

Require: Set of interacting bodies $\{H\} = \{h_0, h_1, \dots\}$ where $i = 0..n$, n = total bodies, and their respective contact forces $\{\mathbf{f}_i^h\}$.

Ensure: Bodies collide due to gravity and compressive forces.

```

1: while  $\mathcal{G}(\{H\}, \{\alpha(H)\}, g, \rho, \mu, e) = \{c\}$  do
2:   while  $\{H\} \neq \emptyset$  do
3:     for all  $c_i$  in  $\{c\}$  do
4:        $c_i = \{\mathbf{f}_i = \{\mathbf{f}_0, \dots, \mathbf{f}_c\} \cup p_i = \{p_0 \dots p_c\}\}$ ;
5:       for all  $H_i$  do
6:          $v_i^h = \text{Volume}(H_i)$ 
7:         if  $v_i^h > f_{\text{threshold}}$  then
8:            $\{H_i^n\} = \mathcal{F}(H_i, \mathbf{f}_i, E, \nu, K_{IC})$ ;
9:         end if
10:        Add new fragments,  $H = H \cup \{H_i^n\}$ 
11:      end for
12:    end for
13:  end while
14: end while
  
```

3 Geometric representation

All domains are represented using solid modeling techniques that take into account the evolutionary nature of their geometries [51]. Each fracture, f_i , evolves from a set of growth curves, and is stored as a parametric surface, f_i^p . The fragment boundary, m , is also initially represented by a set of surfaces. It follows that

$$\forall f \mid (\exists i \mid f_i^p \in f_i) \vee (f_i^p \in m) \quad (8)$$

$$\forall f \mid f_i^p \xrightarrow{\text{meshing}} \{t_i\}_{i \rightarrow \text{triangles}} \quad (9)$$

$$= \text{closed non-convex polyhedron} \quad (10)$$

after meshing, each surface is triangulated. Tips are defined implicitly by the underlying surface parametric space. It follows that

$$\forall i \mid (\exists f_i \mid t_i \in f_i) \vee (t_i \in m) \quad (11)$$

$$\forall i \mid (\exists f_i \mid b_i \in f_i) \vee (b_i \in m) \quad (12)$$

where t_i is a triangle representing part of the tip surface, and b_i is a tip segment.

At each growth step, a new tip front curve is formed from the movement of the original tip. The fracture surface is deformed to accommodate this constraint using NURBS constraint-based deformation [52]. Figure 2 shows (a) the initial stage of the simulation, (b) an advanced state where fracture propagation has occurred, and (c) the final fragmentation.

When a crack tip reaches a free surface it stops, and the fracture body merges with the foreign object. There are two distinguishable cases: the tip stops on a block boundary, or it stops on a fracture wall. In the first case, the fracture shape becomes part of the block description. In the second case, fracture shapes merge into one object. Thus, tips can exist within the body of a fracture, or within a block. In other words, tips can make a negative or a positive volume change. It becomes necessary to treat these in a flexible fashion. In order to allow this flexibility, progression fronts are defined as cutting surfaces that subdivide the domain as the tip advances.

4 Impulse dynamics

The impulse-based method is a strategy to model the dynamics of multiple bodies with frequent collisions and contact modes with frequent changes [24,26]. It has been shown to converge towards the exact solution of the dynamics problem as the step size decreases [53]. Unlike DEM-based methods, which compute explicit constraint forces at contact points, the impulse method captures collision impulses between bodies by solving the linear complementarity problem [54,55]. Its main advantage is that it does not require full geometric intersection checks. Additionally, in contrast to penalty and

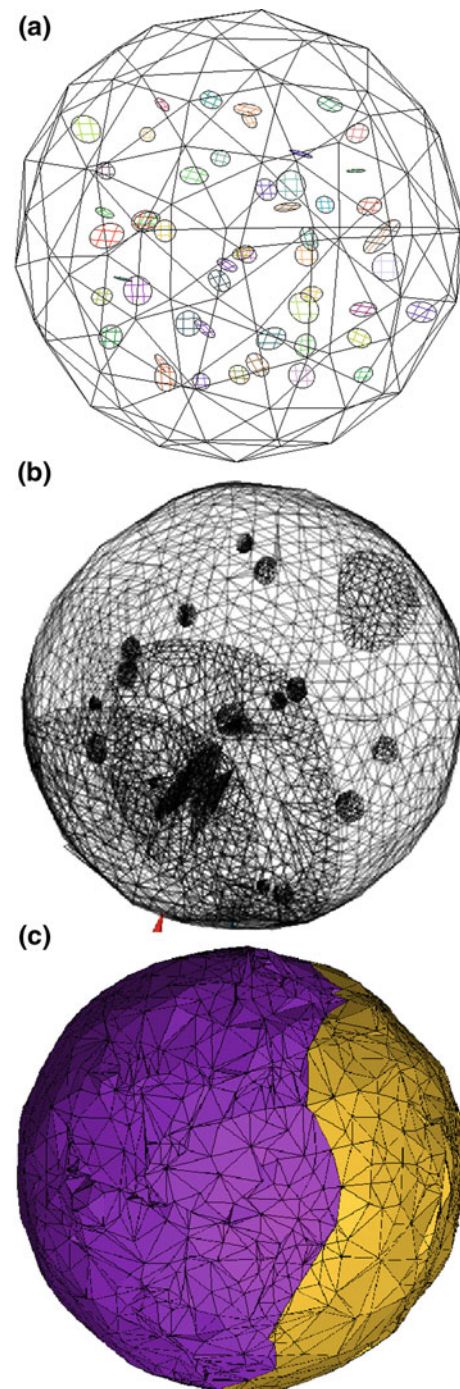


Fig. 2 Fragmentation driven by fracture growth. **a** A set of initially planar, elliptic flaws grow as a function of stress intensity factor measurement at the tip locality. **b** Fractures advance at a relative speed controlled by the propagation factor exponent. **c** Once fractures reach the solid's boundary, the fragment domain is automatically sub-divided. In **b** the red cone at the bottom-left represents the point of impact, in **c** the object is rotated to the left by 45°

Lagrange-based methods, it does not rely on arbitrary penalty parameters that define repulsion as a function of penetration, but instead predicts collisions as a function of impact between proximal bodies.

The impulse method for multi-body collision is subdivided into three steps [56]:

1. collision detection, by means of estimating the time of collision, t_c , between critical collision pairs, which is computed using the conservative advancement method [26] using the intersection of the geometric Minkowski sum of the moving bodies [57],
2. dynamic evolution, which comprises the integration of the equations of motion for each body, described by Newton's law and the conservation of angular momentum,
3. the collision response, which applies a collision impulse to each critical pair that collide, modeled using Guendelman's energy conservative approach [58,59].

The collision detection finds the timestep t_i , which occurs before object contact (see Fig. 3a). A pair of impulses is applied to avoid penetration. Penetration depth, area, and volume are not computed. Collision time is assumed to be infinitesimal, and so the positions of the bodies are treated as constant during a collision. Velocities and positions are updated at each time-step using a semi-implicit Euler time-stepping scheme.

The following equations describe the computation of the impulse between two bodies during a collision (see Fig. 3b). The change of linear velocity and the impulse are related by Eq. 3. The change of angular velocity is expressed as

$$\Delta\omega(t) = \mathbf{I}_a^{-1} \mathbf{r}_a \times \mathbf{p} \quad (13)$$

where \mathbf{r}_a is the vector from the center of mass to the contact point, and \mathbf{I}_a^{-1} is the inertia tensor, computed numerically for

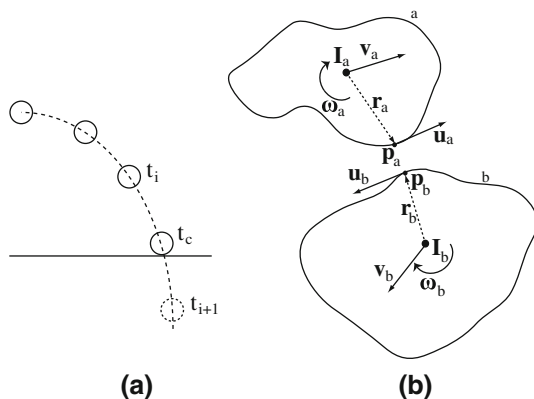


Fig. 3 Computation of collision response using impulse-based method. In **a** each dot represents the location of a body at timestep t_i . If penetration occurs at time step t_{i+1} , a pairs of impulses is applied to avoid this penetration. The time for collision t_c is estimated in order to avoid geometric overlap during collision. In **b** two bodies collide (figure after Mirtich 1996): \mathbf{r}_a and \mathbf{r}_b are vectors from the center of mass to the contact point, \mathbf{p}_a and \mathbf{p}_b are the applied impulses, \mathbf{u}_a and \mathbf{u}_b are the relative velocities at the contact point, \mathbf{v}_a and \mathbf{v}_b are linear velocities, ω_a and ω_b are angular velocities, and \mathbf{I}_a and \mathbf{I}_b are the respective inertia tensors

each fragment, as in [60]. The relative velocity between the colliding bodies is defined as

$$\Delta\mathbf{u}(t) = \left[\left(\frac{1}{m_a} + \frac{1}{m_b} \right) \mathbf{1} - (\mathbf{r}_a \mathbf{I}_a^{-1} \mathbf{r}_a + \mathbf{r}_b \mathbf{I}_b^{-1} \mathbf{r}_b) \right] \mathbf{p} = \mathbf{K} \mathbf{p} \quad (14)$$

where $\mathbf{1}$ is the identity matrix, and \mathbf{K} is the constant, nonsingular, symmetric, and positive definite collision matrix [26]. The inversion of \mathbf{K} plays a central role in the computation of the impulses during the collision, as $\mathbf{p} = \mathbf{K}^{-1} \Delta\mathbf{u}$.

Hahn [24] introduced friction into the approach by neglecting the tangential velocity after collision

$$\mathbf{u}_f = -e u_n^b \mathbf{n} \quad (15)$$

where \mathbf{u}_f is the relative velocity after collision, \mathbf{n} is the normal vector at the contact point, e is the restitution coefficient, and u_n^b is the normal component of the relative contact velocity before collision.

Friction is implemented by applying Newton's law, considering the static friction condition

$$|\mathbf{p}_t| < \mu |\mathbf{p}_n| \quad (16)$$

where \mathbf{p}_t and \mathbf{p}_n are the impulse components tangent and normal to the collision, respectively, and μ is the coefficient of friction. The relative velocity in the normal direction before and after collision should satisfy $u_n^f = -e u_n^b$ [58], in order for the system to be energy conservative [59]. When the static friction condition is not satisfied, the magnitude of normal component of impulse \mathbf{p} is defined as

$$|\mathbf{p}_n| = -\frac{u_n^b (e + 1)}{\mathbf{n}^T \mathbf{K} (\mathbf{n} - \mu \mathbf{t})} \quad (17)$$

where \mathbf{t} is the tangential vector at contact point. Finally, the impulse is defined as

$$\mathbf{p} = |\mathbf{p}_n| \mathbf{n} - \mu |\mathbf{p}_n| \mathbf{t}. \quad (18)$$

Energy loss during a collision due to fracture propagation, wave propagation, heat, sound, and other dissipative energy mechanisms, is currently represented by a constant restitution parameter. Restitution is taken here to be 0.8, corresponding to a loss of 36 % of the energy at each collision. Accurate and efficient estimation of collision energy loss is a key unresolved challenge, part of our work in progress.

The computed impulse is used to resolve the dynamics of the system as a function of time and body geometry. The algorithm deals with convex bodies such as tetrahedra and hexahedra directly, and relies on convex body decomposition of the original domain to handle non-convex shapes. This method is viable for the simulation of gravity-driven dynamics, for continuous and discontinuous contact simulation. Using this approach, collisions are the only mechanism through which forces are transmitted between bodies.

5 Deformation

Fragments are handled as independent FEM domains. Deformation is modeled quasistatically assuming an homogeneous, isotropic, linear elastic medium [61]:

$$\boldsymbol{\sigma} = \mathbf{D}_e(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (19)$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the strain and stress vectors, $\boldsymbol{\sigma}_0$ and $\boldsymbol{\varepsilon}_0$ are the initial strain and stress, and \mathbf{D}_e is a linear elastic stiffness matrix. For a static system

$$\boldsymbol{\partial}\boldsymbol{\sigma} + \mathbf{F} = 0 \quad (20)$$

where $\boldsymbol{\partial}$ is the divergence operator and \mathbf{F} represents the external body forces, i.e. gravity, dilatation, and acceleration. The ensuing system is solved using FEM, for which displacements are the solution variable defined at the nodes, and material properties are defined at the element Gaussian integration points at which stresses and strains are computed. The FEM inversion of the matrix is performed using the Fraunhofer SAMG Solver [62], which solves for vectorial fields using the multi-grid method.

The fragment domain is discretized using a three-dimensional mesh composed of a set of isoparametric quadratic bars, triangles and tetrahedra. Elements around the crack tip are quarter-point tetrahedra, which better capture the singularities at the fracture tips, and improve accuracy in stress intensity factor calculations [63,64]. Figure 4 illustrates the stress contours computed during the collision of a sphere with the ground, the non-symmetric nature of the field is attributed to the effect of growing internal fractures.

The insides of the fractures are not meshed. These exist as negative volumes surrounded by matrix elements. Meshes are generated based on the geometry, and their optimization is based on geometry, whereby

$$t_r = O(d_t) \quad (21)$$

where t_r is the triangle resolution, O is a quadratic function, and d_t is the distance to the closest fracture NURBS surface tip.

6 Fracture propagation

We assume that each fragment has a set of pre-existing flaws, which grow into fractures in response to the collision stresses:

$$H \supset \{f_i^o\} \cup \{f_i\} \quad (22)$$

Each fracture evolves from a set of growth curves, and is stored as a set of parametric surfaces. At each growth step, a new tip front curve forms from the movement of the original tip. The region between these two curves, represented by a ruled surface, corresponds to the new crack surface area; this is called the progression front. These curves describe the

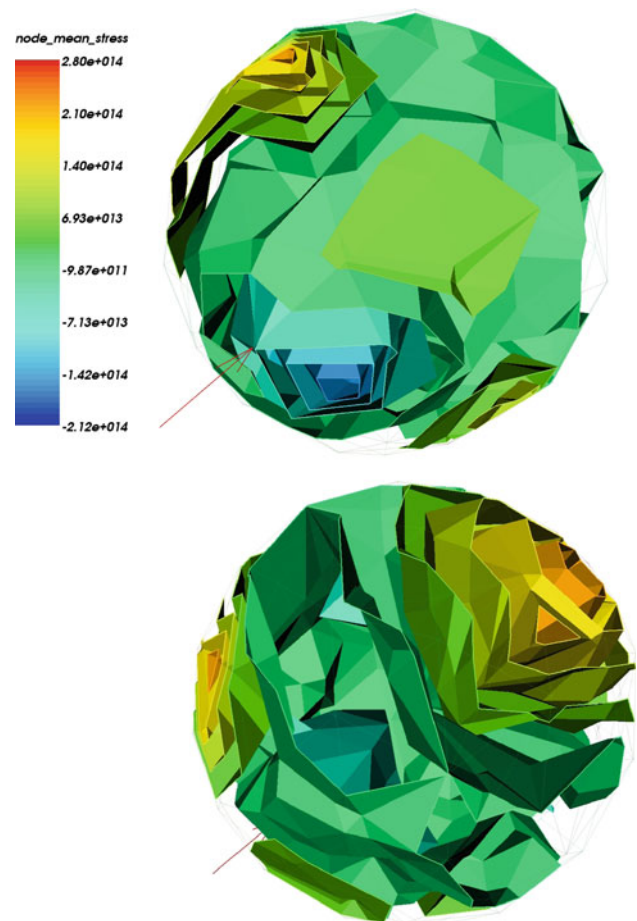


Fig. 4 Mean stress state contours of sphere at impact. The red arrow indicates the location of impact. The images are two rotations of the same 3D contour distribution. Notice the compression area ensuing around the point of impact, and the tensile fields emerging around growing fractures

growth of the fracture body, a polyhedron that grows driven by the addition of the progression front to its boundaries. Figure 5a shows an example of progression of the ensuing fracture mesh during growth, as a function of the local stress field depicted in Fig. 5b.

6.1 Stress intensity factor computation

Stress intensity factors (SIF) are approximated at the center of each fracture tip segment. Most methods, such as the stiffness derivative [65] and the displacement correlation technique, rely on a symmetric brick-like mesh around the tip, in which case the tip region must be discretized by symmetric prism and hexahedral elements arranged in an annular manner. In the present approach, mesh optimization is based on geometry, and is thus arbitrarily generated to optimize the number and quality of the meshes. Therefore, a specific mesh structure around the tip cannot be guaranteed. More flexible

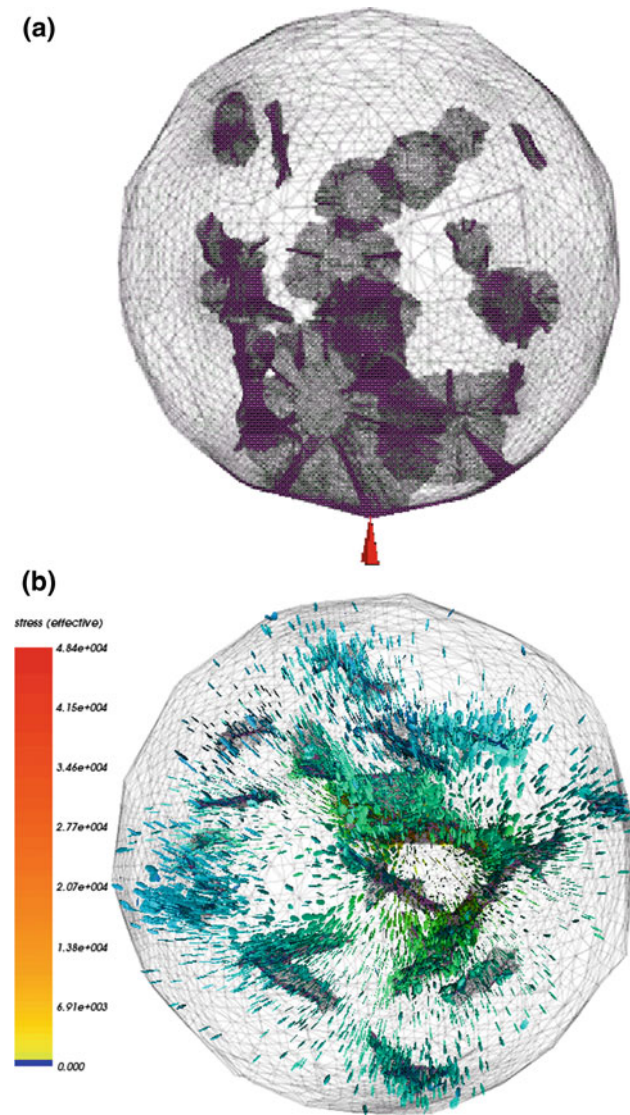


Fig. 5 Fracture growth. **a** Side view, and **b** top view of the same fragmentation. The figure illustrates multiple, simultaneous, discrete fracture growth. The stress field is represented by colored tensors, plotted in logarithmic scale

integration techniques are implemented to obtain accurate SIF values: specifically, the reduced virtual integration technique [49], based on the J-integration over a virtual cylinder extracted from an arbitrarily-defined mesh [66]. This allows SIFs to be computed accurately, independently of the mesh layout.

The stress intensity factor is a measure of the energy concentrated around the crack tip. For linear elastic, brittle materials, $J = G$, where G is the strain energy release rate, and the J Integral, introduced by [67] and [68], characterizes the amplitude of the singular field around a sharp fracture tip. The J Integral, defined as a volume integral in local space, is the energy released by a unit fracture extension in the direction of the fracture plane, defined as

$$J_v = \frac{1}{A_c} \int_V \left(\sigma_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial q_k}{\partial x_i} dv \quad (23)$$

where W denotes the strain energy density, A_c is the increased crack surface area, δ is the Kronecker delta, and q is an arbitrary weighting vector function representing the virtual crack extension. It follows that $J = G = K^2 E_{eff}$, and [69]

$$E_{eff} = E \left[\frac{1}{1 + \nu^2} + \left(\frac{\nu}{\nu + 1} \right) \frac{\varepsilon_z}{\varepsilon_x + \varepsilon_y} \right] \quad (24)$$

where E is Young's modulus, ν is Poisson's ratio, and ε_x , ε_y , and ε_z are the local principal strains.

6.2 Propagation angle

The propagation angle is determined by a 3D angle criterion for multi-axial loading [70], which is a modified maximum circumferential stress method [71] that takes into account K_I , K_{II} , and K_{III} . The criterion considers growth in terms of two deflection angles φ_o and ψ_o which are perpendicular and tangent to the crack tip, respectively, as a function of maximum principal stress σ_o and the stress components σ_φ , σ_z , $\tau_{\varphi z}$ in the reference cylindrical coordinates (φ, z) defined around the tip. It follows that the deflection angle φ_o is computed as

$$\frac{\partial \sigma_o}{\partial \varphi} \bigg|_{\varphi=\varphi_o} = 0 \quad \text{and} \quad \frac{\partial^2 \sigma_o}{\partial \varphi^2} \bigg|_{\varphi=\varphi_o} < 0 \quad (25)$$

the deflection angle ψ_o is defined by the orientation of σ_o as

$$\psi_o = \frac{1}{2} \arctan \left(\frac{2\tau_{\varphi z}(\varphi_o)}{\sigma_\varphi(\varphi_o) - \sigma_z(\varphi_o)} \right) \quad (26)$$

and the component of the local stress intensity factor in the direction of propagation, K_v is defined as

$$K_v = \frac{1}{2} \cos(\varphi_o/2) \left\{ K_{cs} + \sqrt{K_{cs}^2 + 4K_{III}^2} \right\} \quad (27)$$

where $K_{cs} = K_I \cos(\varphi_o/2)^2 - 3/2 K_{II} \sin(\varphi_o)$. φ_o is computed numerically *a priori*, as a function of the normalized modal K_i^n where

$$K_i^n = \frac{|K_i|}{K_I + |K_{II}| + |K_{III}|} \quad (28)$$

for $i = \{I, II, III\}$, and assuming that $K_I > 0$. It follows that

- for pure mode I - $\varphi_o = 0^\circ \wedge \psi_o = 0^\circ$
- for pure mode II - $\max(\varphi_o) = 70.5^\circ \wedge \psi_o = 0^\circ$
- for pure mode III - $\varphi_o = 0^\circ \wedge \max(\psi_o) = 45^\circ$

In areas where the discretization error is high, e.g. in the case of proximal objects ready to intersect, the propagation angle is assumed to be zero.

6.3 Failure

Three main failure criteria have been used to determine growth [72]: the maximum circumferential stress criterion [71] for which the implementation is straightforward, accurate, and dependent on mesh refinement; the minimum strain energy density criterion [73], which is less accurate and also mesh-dependent; and the maximum strain energy release rate criterion, which requires complex implementation of stress intensity factor measurements at the fracture tips, but the accuracy of which is mesh-independent. An energy-based approach is used in the present work, in order to remain discretization-independent. This requires the computation of SIF around the tip.

A failure criterion, such as the sub-critical failure criterion e.g. [74], determines whether or not local fracture propagation will occur:

$$K_{IC} \geq K_v \geq K_I^* \quad (29)$$

where K_{IC} is the critical stress intensity factor, or toughness, and K_I^* is the sub-critical stress intensity factor, usually defined as some fraction of K_{IC} .

6.4 Propagation extent

The propagation criterion describes the magnitude of propagation. Fractures are assumed to grow quasistatically, as a result of applying the average force occurring during impact in an iterative manner until growth ceases. Dynamic effects of growth are incorporated by adjusting the propagation law to the velocity at which objects collide.

An empirical Paris-Walker approach [75,76] is used, which relies on a power law depending on the SIF and SIF ratio, which has been shown to increase exponentially as a function of impact velocity [77], as well as on material-specific, experimentally-measured properties:

$$\frac{da}{dN} = C [(1-R)^p K_{max}]^{\alpha^*} \quad (30)$$

where α^* and C are the Paris constants, K_{max} is the maximum stress intensity factor local to the fracture tip, and

$$R = K_{min}/K_{max} \quad (31)$$

where K_{min} is the minimum stress intensity factor local to the fracture tip, and R is the stress ratio. Figure 6 shows the relationship between the exponent and the fracture advance.

The parameter p is modeled as a function of damage, which has been shown to be proportional to the cube of the impact velocity [78]. In this work, it is assumed that $\alpha^* = 10$, which is in the range of measured growth exponents for quasi-brittle materials e.g.

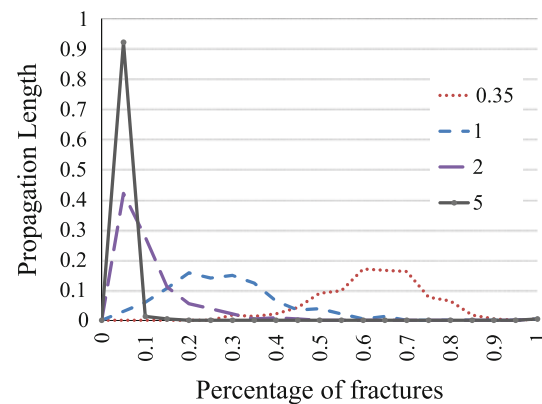


Fig. 6 Fracture growth as a function of the propagation law exponent. For higher propagation exponents, only few cracks grow at a higher speed, whereas for lower exponents the distribution of growth tends to Gaussian

[79]. Growth is computed using a simplified form of Eq. 30

$$a_{adv} = a_{max} (G/G_{max})^{\alpha = \alpha^* (1-D^*)^2} \quad (32)$$

where G is the strain energy release rate, a function of the stress intensity factor, G_{max} is the fracture's maximum G , a_{adv} is the advance distance at a single point at the crack tip, a_{max} is the maximum crack advance, and α is the modified growth exponent. The exponent p is defined by

$$D^* = f(D) \quad (33)$$

where $D \in [0, 1]$ is a variable representing damage of the fragment due to impact in terms of micro-crack nucleation during collision. It is assumed that [78]

$$D \propto v_c^3 \quad (34)$$

where v_c is the collision velocity, $D = 0$ for $v_{min} = 110$ m/s and $D = 1$ for $v_{max} = 220$ m/s after [80]. Specifically, it is proposed that

$$D^* = \left(2 \frac{(v_c - v_{min})}{v_{max}} \right)^3 \quad (35)$$

where α varies exponentially as depicted in Fig. 7. It follows that

$$v_c = \frac{v_{max}}{2} \sqrt[3]{1 - \sqrt{\alpha/\alpha^*}} + v_{min}. \quad (36)$$

This approach models high and low velocity scenarios that exhibit fragmentation properties that are consistent with experimental data. Propagation and failure criteria are both material and in-situ condition dependent, and should be adjusted as part of the experimental setup of the deformation experiments. Details of the implementation and validation of the three-dimensional fracture propagation,

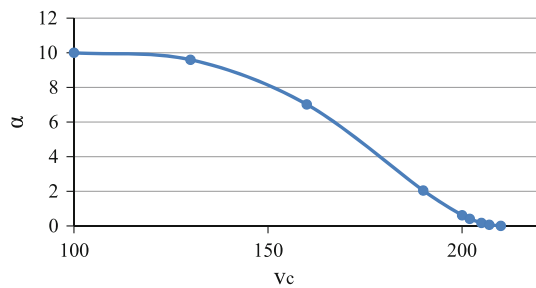


Fig. 7 Propagation exponent variation with velocity. The velocity ranges between 100 and 220 m/s, which yields exponentially varying α values

using a standard Paris law propagation law, are discussed in [49].

6.5 Geometric crack propagation

The propagation criterion estimates the extension of the crack and its direction. Given this information, the new locations of the tip nodes are computed, a curve is interpolated through them, and the crack front is extended by lofting a surface between the old and new tip nodes. Fractures propagate until they reach a fixed boundary (such as a rock wall), or make contact with another fracture. Intersections between the growing fractures are handled using 3D Boolean operations. Furthermore, the same geometric operations are used to merge intersecting fracture shapes.

The body of each fracture is represented by a closed polyhedron using BREP [50]. It encloses a region corresponding to the inside of the fracture. A fracture may evolve into a cluster of multiple cracks by coalescing with neighboring cracks. The simulator tracks crack tips individually during growth. Thus, any edge on the crack surface might become, at some stage of growth, a new tip curve. After each iteration, the length advance and propagation angle jointly define the geometric extension of the fracture. This results in a fracture aperture that is an emergent property, given by the separation of the fracture walls due to deformation, rather than being defined arbitrarily. After each fracture propagation step, the fracture BREP is updated using the deformation of the FEM mesh.

Boolean operations handle fracture coalescence. If fracture propagation extends beyond a neighboring crack wall, the algorithm corrects the tip shape to merge with the proximal crack as it intersects it. This behavior mimics how an opening mode crack, upon intersection with another open crack, ceases to grow and connects both paths, thereby forming a single body. Geometric housekeeping of the dataset is key to the following automatic re-meshing procedure. Frac-

ture intersection eventually triggers the formation of sub-domains.

7 From fracture to fragment

Once fractures reach the boundaries of the body and fragment the domain, the resulting bodies are handled as independent continuous domains. This strategy relies on two processes: automatic recognition and creation of volumetric domains using the flood-fill algorithm, and automatic identification of modeling capacity of the domain. The first is a process that will create and identify separate volumes as fragments subdivide. The second is a method that classifies fragments as a function of their volume using a modeling threshold, $v_{\text{threshold}}$, and a fracturing threshold, $f_{\text{threshold}}$. The first defines a volume so small that new fragments below this size are discarded from the simulation, thereby avoiding the insertion of microscopic elements into the simulation. Alternatively, these smaller-sized fragments can be re-approximated as spheres, and reinserted to the simulation as spherical micro-particles.

In contrast, the $m_{\text{threshold}}$ defines the smallest fragment that will continue to be fractured. In order to improve simulation efficiency, small fragments can be fractured using single initial flaws, which grow to fragment the domain in a single step. These two thresholds allow the three scales at which the simulation operates to be linked, as fines are represented by low cost spheres, small fragments are handled using simplified fracture growth (modeled as the growth of a single flaw originating at the point of contact with larger force), and larger fragments are modeled in great detail. This flexibility is the key to conducting simulations that can balance variable time and resolution constraints. In these simulations, $v_{\text{threshold}} = 1 \times 10^{-4} \text{ m}^3$ and $f_{\text{threshold}} = 1 \times 10^{-2} \text{ m}^3$.

8 Application: fragmentation of a sphere

The methodology is now demonstrated by numerically fragmenting a brittle sphere. The sphere radius is 100 mm, and collides with a flat surface at 90 degrees, with a velocity at impact of 100, 173, 190, or 202 m/s. The sphere's Young's modulus is 20 GPa, and its Poisson's ratio is 0.25. A low value of K_{IC} of $1.5 \times 10^{-6} \text{ Pa m}^{1/2}$ is assigned, to encourage fragmentation. The volume-based cumulative fragment size distribution is computed, and the shape of the fragments and their final geometric distribution are studied. The sphere initially contains fifty planar disk flaws, randomly located, with a normal distribution of radii having a mean of 5 mm and a standard deviation of 0.5 mm (Fig. 2a).

Figure 4 illustrates the mean stress contours of the sphere at impact. A compressive zone is observed around the impact

location, with tensile stress concentrations at flaws located away from the impact location. By adjusting the propagation exponent, α , the relative speed at which fractures grow at impact is controlled. For low velocities, $\alpha > 10$ (100 m/s), fractures with a higher SIF at their tips will grow considerably faster than will their neighbors. Figure 2b shows growth for $\alpha = 10$: the fracture at the impact location grows significantly faster than the others. The sphere is fragmented into two pieces in this case (Fig. 2c). For $\alpha = 5$ (173 m/s) and $\alpha = 2$ (190 m/s), thirty-five and seventy-five fragments are obtained, respectively, as shown in Fig. 8a, b. Finally, for $\alpha = 0.35$ (202 m/s), modeling a high velocity impact, 775 fragments are obtained (Fig. 8c). These experiments are referred to as case s.l.1 ($\alpha = 10$), s.h.2 ($\alpha = 5$), s.h.3 ($\alpha = 2$), and s.vh.4 ($\alpha = 0.35$). In all cases and images, the point of impact is at the bottom of the sphere.

These results indicate that cumulative fragment mass distributions are characterized by a two-parameter cumulative Weibull distribution, as has been observed in laboratory experiments [48]. The two-parameter cumulative Weibull

distribution is commonly used to characterize brittle behavior [81]. It can be expressed as

$$y = 1 - \exp \left[- \left(\frac{x}{x_c} \right)^m \right] \quad (37)$$

where x is the fragment volume, y is the corresponding volume fraction, m is the shape factor, and x_c is the scaling factor.

Fragment mass distributions resulting from the previous experiment are measured and compared to experimental data. Fragment sizes decrease as the velocity increases. In the graphs, cases s.h.2 and s.h.3 correspond to a lower velocity and a higher propagation exponent, and case s.vh.4 corresponds to a higher velocity and a lower propagation exponent. For the fragment mass distribution, $F(m)$, in Fig. 9, a power law distribution with an exponential cut-off [80] is observed. This corresponds to a power-law distribution of the small fragment size, and an exponential regime for the larger fragments.

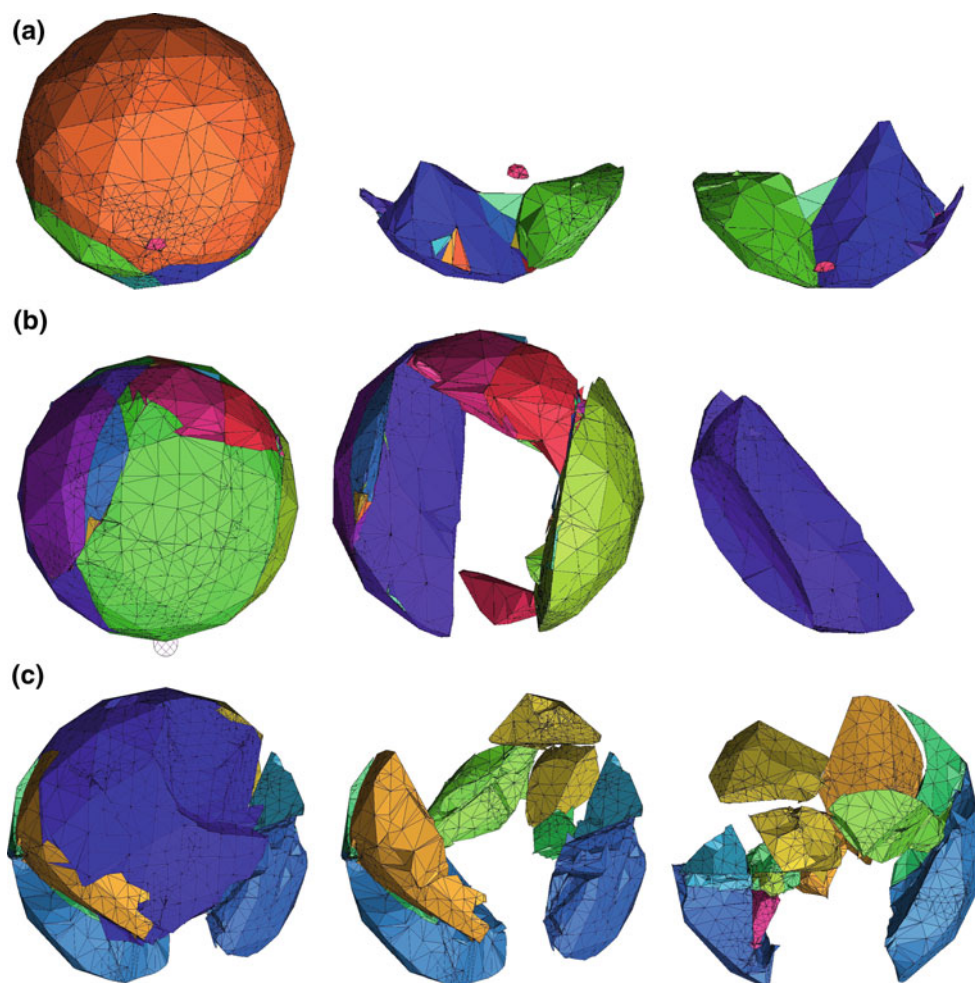


Fig. 8 Fragment shapes for **a** case s.h.2—low, **b** case s.h.3—mid, and **c** case s.vh.4—high velocity impact

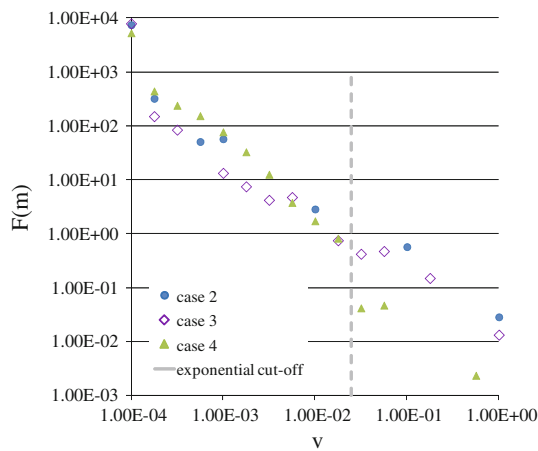


Fig. 9 Fragment mass distribution, where v is the volume of the fragment, and $F(m)$ is the frequency of fragments of a specific volume. Notice that the simulation generates fragments below the volume threshold, $f_{\text{threshold}} = 1 \times 10^{-4} \text{ m}^3$, these are not further fragmented, they are accumulated as the simulation advances but do not interact with the remaining fragments

In terms of the two-parameter Weibull distribution, [48] observed that x_c reduces linearly as a function of increased velocity, whereas m reduces exponentially with velocity. In the present experiments, both x_c and m reduce as the velocity increases

$$(x_c : 0.47 \rightarrow 0.164, m : 3.08 \rightarrow 1.5). \quad (38)$$

In Fig. 10, the distribution presented by [80] is presented, in normalized form, for comparison. The distribution resulting from the low velocity case fits well with the fragment size distributions presented therein, for the chosen function $\alpha = f(v)$. For the higher velocity case, a reduced shape and scaling factor for the fitted cumulative Weibull distribution is obtained. In order to maintain numerical accuracy, object sizes are capped using a defined volume threshold of $f_{\text{threshold}} = 1 \times 10^{-4} \text{ m}^3$, which limits fragments from further breaking apart, and affects the accuracy of the distribution of smaller fragment sizes.

Fragment shape distribution has been shown to depend on scale, material, velocity of impact, and angle of impact [48, 82, 83]. Here, the focus is on reproducing the fragment size variation as a function of the velocity of impact, while scale, material and angle of impact are kept constant. The idea is to reproduce, not only fragment size distributions, but also final fragment shape characteristics as a function of velocity. For case s.h.2, most fragmentation occurs around the impact location. Conical fragmentation areas around the zone of impact are reproduced, along with wedge and cap shaped fragments (see Fig. 11). One large fragment dominates the distribution, and fractures tend to grow radially out of the center of the model. For case s.h.3, a wider distribution of fragment sizes occurs, and the formation of wedges and

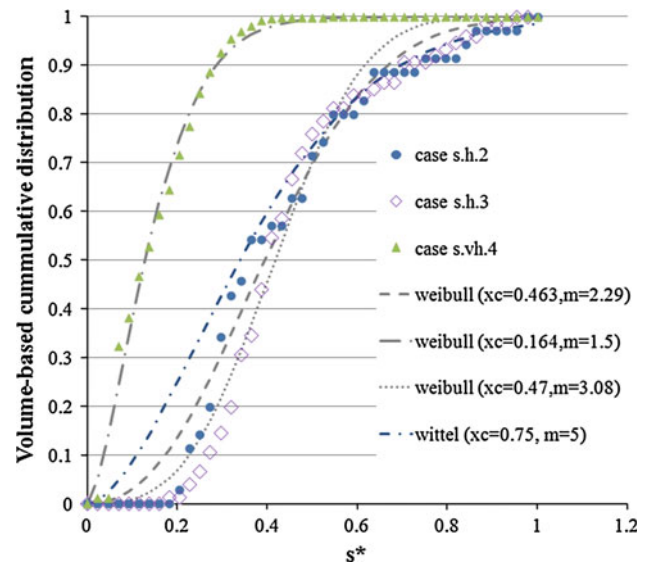


Fig. 10 Cumulative fragment mass distribution, where s^* is the normalized fragment volume, and the y axis plots the volume-based cumulative distribution of the fragments

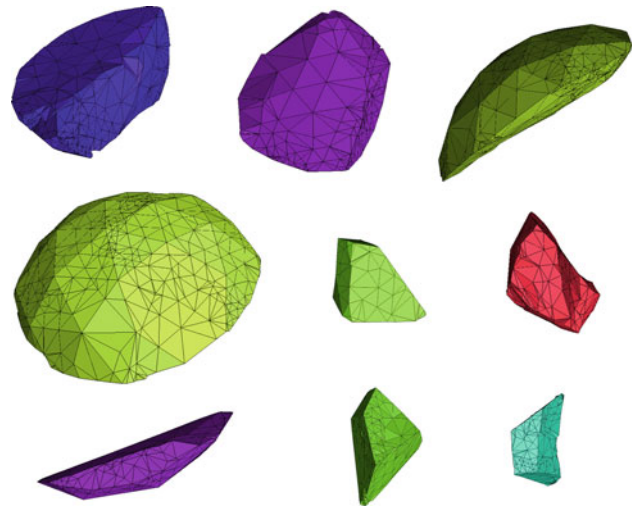


Fig. 11 Collection of fragment shapes for s.h.2, s.h.3, and s.vh.4

caps is observed. Case s.vh.4, which models higher velocity impact, exhibits a wider distribution of fragment sizes. Significant crushing around the impact location occurs in this case, and prominent wedge fragments are created radially. Wedges are formed due to the growth of meridian plane fractures. These have been observed to form as an effect of the compressive conical region on the sphere's stress state [82].

9 Conclusions

Simulation capabilities to model coupled fracture and fragmentation processes using a combined finite element, impulse-based approach, have been presented. The method

can be implemented as an extension of a 3D fracture propagation code, provided that the representation of the cracks is explicit and geometry-based. Geometric collision is based on polyhedral tracing, divided into broad and narrow phase contact detection for increased efficiency. Thus, objects are not allowed to penetrate during the collision. Instead, object trajectories are used to estimate time-of-impact, and contact between bodies is modeled by collisions at contact locations. This results in a smaller geometric computation burden, as penetration distances and penetration points, as would be required by the penalty method, need not be computed, and material properties are allowed to be defined at the macro-scale. The advantages of the proposed method include

- fragmentation that is controlled by fracturing rather than driven by bond-breakage,
- explicit geometric representation of fractures and fragments via geometric modeling of surfaces and solids, and
- direct definition of macroscopic material properties.

Fragments can assume polyhedral or spherical shapes depending on their size, and fracture propagation is modeled using an energy-based growth approach. The Paris law propagation exponent is defined as a function of impact velocity. This methodology has been demonstrated on the problem of fragmenting a sphere, with the resulting fragment size distribution quantified. The observed fragment volume distribution can be fitted using a two-parameter Weibull distribution, and the average fragment sizes decreases as a function of the impact velocity.

Acknowledgments The authors thank Rio Tinto for supporting this work, through the Rio Tinto Center for Advanced Mineral Recovery at Imperial College London.

References

1. Cundall P (1971) A computer model for simulating progressive large scale movements in blocky rock systems. In: Proceedings of the symposium of the international society for rock mechanics, vol 1. Nancy, France
2. Cundall PA, Strack ODL (1979) A discrete numerical model for granular assemblies. *Geotechnique* 29:47–65
3. Mustoe G (1992) A generalized formulation of the discrete element method. *Eng Comput* 9:181–190
4. Kun F, Herrmann HJ (1996) A study of fragmentation processes using a discrete element method. *Comput Method Appl Mech Eng* 138:3–18
5. Wang YN (2009) Three-dimensional rock-fall analysis with impact fragmentation and fly-rock modelling. Dissertation, University of Texas at Austin
6. Wang Y, Tonon F (2010) Discrete element modelling of rock fragmentation upon impact in rock fall analysis. *Rock Mech Rock Eng* 44:23–35
7. Wittel F, Schulte-Fischedick J, Kun F (2003) Discrete element simulation of transverse cracking during the pyrolysis of carbon fibre reinforced plastics to carbon/carbon composites. *Comput Mater Sci* 28:1–15
8. Tillemans HJ, Herrmann HJ (1995) Simulating deformations of granular solids under shear. *Phys A* 217:261–288
9. Shi G, Goodman R (1984) Discontinuous deformation analysis. The 25th US symposium on rock mechanics :84–0269
10. Ghaboussi J (1988) Fully deformable discrete element analysis using a finite element approach. *Comput Geotech* 5:175–195
11. Preece D, Taylor L (1989) Complete computer simulation of crater blasting including fragmentation and rock motion. In: Proceedings of the fifth annual symposium on explosives and blasting research, society of explosives engineers. New Orleans, USA
12. Preece D, Taylor L (1990) Spherical element bulking mechanisms for modeling blasting induced rock motion. In: Proceedings of the third international symposium on rock fragmentation by blasting. Brisbane, Australia
13. Taylor L, Preece D (1992) Simulation of blasting induced rock motion using spherical element models. *Eng Comput* 9:243–252
14. Munjiza A (2004) The combined finite-discrete element method. Wiley, Chichester
15. Munjiza A, Knight EE, Rougier E (2011) Computational mechanics of discontinua. Wiley, Chichester
16. Munjiza A, Andrews K, White J (1999) Combined single and smeared crack model in combined finite-discrete element analysis. *Int J Numer Methods Eng* 44:41–57
17. Williams J, Connor R (1999) Discrete element simulation and the contact problem. *Arch Comput Methods Eng* 6:279–304
18. Wittel F (2010) Single particle fragmentation in ultrasound assisted impact comminution. *Granul Matter* 12:447–455
19. Bagherzadeh K, Mirghasemi A, Mohammadi S (2010) Numerical simulation of particle breakage of angular particles using combined DEM and FEM. *Powder Technol* 205:15–29
20. Pine R, Owen D, Coggan J, Rance J (2007) A new discrete fracture modelling approach for rock masses. *Geotechnique* 57: 757–766
21. Owen D, Feng Y, de Souza Neto E, Cottrell M, Wang F, Pires FA, Yu J (2004) The modelling of multi-fracturing solids and particulate media. *Int J for Numer Methods Eng* 60:317–339
22. Munjiza A, John N (2002) Mesh size sensitivity of the combined FEM/DEM fracture and fragmentation algorithms. *Eng Fract Mech* 69:281–295
23. Yang B, Jiao Y, Lei S (2006) A study on the effects of microparameters on macroproperties for specimens created by bonded particles. *Eng Comput* 23:607–631
24. Hahn JK (1988) Realistic animation of rigid bodies. *Comput Graph* 22:299–308
25. Lin M, Canny J (1991) A fast algorithm for incremental distance calculation. In: Proceeding of the IEEE international conference on robotics and automation, IEEE. Sacramento, CA, USA
26. Mirtich B, Canny J (1994) Impulse-based dynamic simulation. In: Goldberg K, Halperin P, Latombe JC, Wilson R (eds) Workshop on algorithmic foundations of robotics. A.K. Peters, Boston
27. Tang X, Paluszny A, Zimmerman RW (2012) Impulse-based simulation of particle flow during subsurface block cave mining. XXIII ICTAM. Beijing, China, 19–24
28. Reznik D, Brown S, Canny J (1997) Dynamic simulation as a design tool for a microactuator array. In: IEEE international conference on robotics and automation, vol 2. p 1675–1680
29. Hollerbach J, Thompson W, Shirley P (1999) The convergence of robotics, vision, and computer graphics for user interaction. *Int J Rob Res* 18:1088–1100
30. Sueda S, Kaufman A, Pai DK (2008) Musculotendon simulation for hand animation. *ACM Trans Graph* 27(3):1–83
31. Weinstein R, Guendelman E, Fedkiw R (2008) Impulse-based control of joints and muscles. *IEEE Trans Vis Comput Graph* 14(1):37–46

32. Schöllmann M, Fulland M, Richard H (2003) Development of a new software for adaptive crack growth simulations in 3D structures. *Eng Fract Mech* 70:249–268
33. Bremberg D, Dhondt G (2008) Automatic crack-insertion for arbitrary crack growth. *Eng Fract Mech* 75:404–416
34. Gurses E, Miehe C (2009) A computational framework of three-dimensional configurational-force-driven brittle crack propagation. *Comput Methods Appl Mech Eng* 198:1413–1428
35. Beissel S, Johnson G, Popelar C (1998) An element-failure algorithm for dynamic crack propagation in general directions. *Eng Fract Mech* 61:407–425
36. Tang C, Yang W, Fu Y, Xu X (1998) A new approach to numerical method of modelling geological processes and rock engineering problems continuum to discontinuum and linearity to nonlinearity. *Eng Geol* 49:207–214
37. Bordas S, Rabczuk T, Zi G (2008) Three-dimensional crack initiation, propagation, branching and junction in non-linear materials by an extended mesh-free method without asymptotic enrichment. *Eng Fract Mech* 75:943–960
38. Belytschko T, Tabbara M (1996) Dynamic fracture using element-free Galerkin methods. *Int J Numer Methods Eng* 39:923–938
39. Moes N, Dolbow J, Belytschko T (1999) A finite element method for crack growth without remeshing. *Int J Numer Methods Eng* 46:131–150
40. Daux C, Moes N, Dolbow J, Sukumar N, Belytschko T (2000) Arbitrary branched and intersecting cracks with the extended finite element method. *Int J Numer Methods Eng* 48:1741–1760
41. Sukumar N, Moes N, Moran B, Belytschko T (2000) Extended finite element method for three-dimensional crack modelling. *Int J Numer Meth Eng* 48:1549–1570
42. Dolbow J, Moes N, Belytschko T (2001) An extended finite element method for modelling crack growth with frictional contact. *Comput Meth Appl Mech Eng* 190:6825–6846
43. Rabczuk T, Belytschko T (2004) Cracking particles: a simplified mesh-free method for arbitrary evolving cracks. *Int J Numer Method Eng* 61:2316–2343
44. Owen SJ (1998) A survey of unstructured mesh generation technology. In: 7th international meshing roundtable, Sandia National Laboratories, 239–267
45. Klingner BM, Shewchuk JR (2007) Aggressive tetrahedral mesh improvement. *Proceedings of the 16th international meshing roundtable*. Seattle, Washington, 3–23
46. Labelle F, Shewchuk JR (2007) Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans Graph* 26:57.1–57.10
47. Si H (2010) Constrained Delaunay tetrahedral mesh generation and refinement. *Finite Elem Anal Des* 46:33–46
48. Cheong Y, Reynolds G, Salman A, Hounsflow M (2004) Modelling fragment size distribution using two-parameter Weibull equation. *Int J Miner Process* 74:S227–S237
49. Paluszny A, Zimmerman RW (2011) Numerical simulation of multiple 3D fracture propagation using arbitrary meshes. *Comput Method Appl Mech Eng* 200:953–966
50. Paluszny A, Matthai SK (2009) Numerical modeling of discrete multi-crack growth applied to pattern formation in geological brittle media. *Int J Solids Struct* 46:3383–3397
51. Martha LF, Wawrzynek PA, Ingraffea AR (1993) Arbitrary crack representation using solid modeling. *Eng Comput* 9:63–82
52. Greca RL, Daniel M, Bac A (2006) Controlled local deformation of NURBS surfaces to satisfy several punctual constraints. In: *Curves and surfaces*. Avignon, France
53. Schmitt AA, Bender JS, Prautzsch H (2005) Internal report 2005 17 on the convergence and correctness of impulse-based dynamic simulation I. Technical report, Universitaet Karlsruhe
54. Baraff D (1994) Fast contact force computation for nonpenetrating rigid bodies. *Comput Graph* 28:23–34
55. Chang B, Colgate JE (1997) Real-time impulse-based simulation of rigid body systems for haptic display. In: *Proceedings of the 1997 ASME international mechanical engineering congress and exhibition*, 1–8
56. Mirtich B, Canny J (1995) Impulse-based simulation of rigid bodies. In: *I3D '95 Proceedings of the 1995 symposium on Interactive 3D graphics*
57. van den Bergen G (2004) Ray casting against general Convex objects with application to continuous collision detection. Technical report
58. Guendelman E, Bridson R, Fedkiw R (2003) Nonconvex rigid bodies with stacking. *ACM Trans Graph* 22:871–878
59. Tang X, Paluszny A, Zimmerman R (2012) Impulse-based simulation of particle flow during subsurface block-cave mining. In: *XXIII ICTAM*. Beijing, China
60. Tonon F (2004) Explicit exact formulas for the 3-D tetrahedron inertia tensor in terms of its vertex coordinates. *J Math Stat* 1:8–10
61. Jaeger JC, Cook NGW, Zimmerman RW (2007) *Fundamentals of rock mechanics*. Blackwell Publishing, Oxford
62. Stüben K (2001) A review of algebraic multigrid. *J Comput Appl Math* 128(1–2):281–309
63. Banks-Sills L (1991) Application of the finite element method to linear elastic fracture mechanics. *Appl Mech Rev* 44:447–461
64. Banks-Sills L, Sherman D (1986) Comparison of methods for calculating stress intensity factors with quarter-point elements. *Int J Fract* 32:127–140
65. Parks DM (1974) A stiffness derivative finite element technique for determination of crack tip stress intensity factors. *Int J Fract* 10:487–502
66. Cervenka J, Saouma V (1997) Numerical evaluation of 3-D SIF for arbitrary finite element meshes. *Eng Fract Mech* 57: 541–563
67. Cherepanov G (1967) The propagation of cracks in a continuous media. *J Appl Math Mech* 31:503–512
68. Rice JR (1968) A path independent integral and the approximate analysis of strain concentration by notches and cracks. *J Appl Mech* 35:379–386
69. Nikishkov GP, Atluri SN (1987) Calculation of fracture mechanics parameters for an arbitrary three-dimensional crack, by the equivalent domain integral method. *Int J Numer Methods Eng* 24:1801–1821
70. Schöllmann M, Richard HA, Kullmer G, Fulland M (2002) A new criterion for the prediction of crack development in multiaxially loaded structures. *Int J Fract* 117:129–141
71. Erdogan F, Sih G (1963) On the crack extension in plates under plane loading and transverse shear. *J Basic Eng* 85:519–527
72. Bouchard P, Bay F, Chastel Y (2003) Numerical modelling of crack propagation: automatic remeshing and comparison of different criteria. *Comput Methods Appl Mech Eng* 192:3887–3908
73. Sih G, Macdonald B (1974) Fracture mechanics applied to engineering problems-strain energy density fracture criterion. *Eng Fract Mech* 6:361–386
74. Segall P (1984) Rate-dependent extensional deformation resulting from crack growth in rock. *J Geophys J Impact Eng* 89:4185–4195
75. Charles R (1958) Static fatigue of glass. *J Appl Phys* 29:1549–1560
76. Walker S, Bloen D (1957) Studies of flexural strength of concrete, Part 3, effects of variations in testing procedures. In: *ASTM proceedings*, 1122–1139
77. Harrigan J, Reid S, Peng C (1999) Inertia effects in impact energy absorbing materials and structures. *Int J Impact Eng* 22:955–979
78. Ahrens TJ, Rubin AM (1993) Impact-induced tensional failure in rock. *J Geophys J Impact Eng* 98:1185–1203
79. Ciavarella M, Paggi M, Carpinteri A (2008) One, no one, and one hundred thousand crack propagation laws: a generalized Barenblatt and Botvina dimensional analysis approach to fatigue crack growth. *J Mech Phys Solids* 56:3416–3432

80. Wittel F, Carmona H, Kun F, Herrmann H (2008) Mechanisms in impact fragmentation. *Int J Fract* 154:105–117
81. Shih T (1980) An evaluation of the probabilistic approach to brittle design. *Eng Fract Mech* 13:257–271
82. Gorham D, Salman A (2005) The failure of spherical particles under impact. *Wear* 258:580–587
83. Lu C, Danzer R, Fischer FD (2002) Fracture statistics of brittle materials: Weibull or normal distribution. *Phys Rev E* 65:067102