Procedia Computer Science

Volume 80, 2016, Pages 245–256

ICCS 2016. The International Conference on Computational Science

Finite Element Model for Brittle Fracture and Fragmentation

Wei Li[1], Tristan J. Delaney[1], Xiangmin Jiao[1], Roman Samulyak[12]*, and Cao Lu[1]

[1] Dept. of Applied Mathematics & Statistics, Stony Brook University, Stony Brook, NY 11794, USA
{wei.li,tristan.delaney,xiangmin.jiao,roman.samulyak,cao.lu}@stonybrook.edu
[2] Computational Science Initiative, Brookhaven National Laboratory, Upton, NY 11973

Abstract

A new computational model for brittle fracture and fragmentation has been developed based on finite element analysis of non-linear elasticity equations. The proposed model propagates the cracks by splitting the mesh nodes alongside the most over-strained edges based on the principal direction of strain tensor. To prevent elements from overlapping and folding under large deformations, robust geometrical constraints using the method of Lagrange multipliers have been incorporated. The model has been applied to 2D simulations of the formation and propagation of cracks in brittle materials, and the fracture and fragmentation of stretched and compressed materials.

Keywords: brittle fracture, fragmentation, collision detection, finite elements method, nonlinear elasticity

1 Introduction

Computational study of brittle fracture networks are of great interest to engineers who work with material that breaks before it can undergo plastic deformation. This can occur when the material is in a "pre-stressed" configuration, where small deformations and strains can lead the material to surpass its internal critical stress. Of great importance to engineers is the prediction of crack nucleation and overall dependence of the resulting crack pattern on material properties.

Computational methods are of growing importance in modeling brittle fracture in materials. Due to the complex nature of fracture mechanics, several phenomenological methods have been proposed. Such methods include generalized and extended finite element methods (X-FEM) [9, 11], cohesive element (CE) models [3], and spring models [15].

The extended FEM (X-FEM) [11] and the generalized FEM (GFEM) [12], which are closely related and both belong to the partition of unity methods (PUM) [9], enrich the traditional FEM function space with families of discontinuous shape functions, which can model the displacements of either the crack tip or opposite sides of the crack plane. The main advantage lies

*Corresponding author.

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.317

in the fact that the interface can be studied within each individual element without having to constantly remesh near the crack tip. The types of enrichment functions can model a variety of engineering problems the crack tip including branching [7], material interfaces, and soft discontinuities. More recently, a phantom node method, which is a variant of the extended FEM was developed in [13].

Cohesive zone models (CZM) were first introduced by Barenblatt [3] and have been incorporated into commercial FEM codes. The CZM is intriguing because it explicitly avoids the creation of stress singularities by modeling inter-element traction-displacement relationships. Elements become separated when their tractions exceed a critical threshold, and the location of the "cohesive elements" (CE) can generate complex fracture networks. Cohesive elements have been used in traditional finite element codes by Xu and Needleman [18] and Ortiz and Camacho [6]. Additionally, cohesive zone models have been incorporated into other finite element frameworks such as X/G-FEM [11], meshless methods [2], and isogeometric analysis [14]. While CZMs may provide complex fracture structures, it was noted in [8] that CZMs are dependent on aspects of the mesh.

Spring models were introduced in [5, 10] and have the advantage of very simple implementation of solid mechanics and fracture mechanics. For example, Meakin [10] modeled fracture using a two-dimensional network of springs with a critical tension parameter. Over-strained springs were removed to simulate the propagation of fracture. Beale added random defects and perturbations to a spring model to investigate their effects on the propagation of the crack surface [5]. However, both of these models implemented fracture mechanisms by removing springs; thereby losing mass conservation and "obliterating" material under compression.

Recently, Wei et al. [15] investigated the use of mass conservative spring models, and were able to reproduce complex fracture networks in two and three dimensions. The method used nonlinear optimization of the global energy functional and split vertices adjacent to springs that were strained past a pre-defined threshold. The advantage of this method was that it conserved mass and produced rich crack networks throughout the material. The work demonstrated complex fracture patterns, which qualitatively change due to variations in material properties.

The spring network model of [15] is difficult to integrate into pre-existing finite element software. The intent of this paper is to apply the fracture mechanisms of [15] to existing finite element codes. In this work, we describe the fracture mechanism incorporated into a finite element solid mechanics code as well as collision detection algorithms to prevent inter-element penetration.

The remainder of the paper is divided as follows. In Section 2, the principles of continuum mechanics and their implementation within the finite element method are briefly discussed. In Section 3, the fracture mechanism is described, and particular emphasis is placed on the detection of intersecting elements and their resolution through the introduction of Lagrange multipliers. In Section 4, we present some verification of our FEM code and simulation results with our fracture model. Finally, we present concluding remarks in Section 5.

2 Finite Element Based Brittle Fracture Model

Finite element analysis is an important tool in the study of solid mechanics and fracture mechanics in particular. Linear elastic fracture mechanics (LEFM) has been well-studied using methods such as X-FEM and CZM. However, the range of applications of LEFM is limited by the assumptions of linear elasticity, which is valid for small displacements and small strains. In many applications involving brittle fracture, large displacements and rotations may occur while the strain exhibited within a material may still be in the elastic regime. Therefore, it is

important to take into account these geometric nonlinearities in the continuum formulation. In particular, we use a quasi-static updated Lagrangian description of the physical system following that in [4].

For any solid body of some material, denoted at time τ as ${}^{\tau}V$, the material is in energetic equilibrium when the principle of virtual work is satisfied, namely that the body is in a minimum energy configuration. The principle of virtual work for a body at some future time $t+\Delta t$ can be written as [4]

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\tau_{ij}\delta_{t+\Delta t}e_{ij}\,{}^{t+\Delta t}dv = {}^{t+\Delta t}\mathscr{R}, \tag{1}$$

where ${}^{t+\Delta t}\tau_{ij}$ is the Cauchy stress tensor of the material, $\delta_{t+\Delta t}e_{ij}$ is the linear variation of linear strain tensor at time $t+\Delta t$, and ${}^{t+\Delta t}\mathscr{R}$ is the total sum external virtual work due to surface tractions and body forces.

Since the configuration of the body at time $t+\Delta t$ is unknown, (1) cannot be solved directly and instead must be rewritten in term of some previously known configuration. In the *updated Lagrangian* formulation, (1) may be rewritten in terms of the material body in its current configuration at time t,

$$\int_{{}^{t}V} {}_{t}C_{ijrs}\,{}_{t}\varepsilon_{rs}\delta\,{}_{t}\varepsilon_{ij}\,{}^{t}dv + \int_{{}^{t}V} {}^{t}\tau_{ij}\delta_{t}\eta_{ij}\,{}^{t}dv = {}^{t+\Delta t}\mathscr{R} - \int_{{}^{t}V} {}^{t}\tau_{ij}\delta_{t}e_{ij}\,{}^{t}dv, \tag{2}$$

which is a nonlinear equation in the incremental nodal displacements u_i.

The nonlinear equation (2) can be solved using an iterative method. We choose to use the modified Newton-Raphson scheme of [4] to iteratively determine the solution of (2). In particular, the stiffness matrix associated with (2) is computed only once and is held constant throughout the Newton-Raphson iteratin. Lagrange multipliers are used to accommodate several types of (linear) Dirichlet boundary conditions on the vertex displacement degrees of freedom, u_i. We solve (2) iteratively by solving for several incremental displacements of the form

$$\begin{bmatrix} {}^{t}\boldsymbol{K} & \boldsymbol{G}^{T} \\ \boldsymbol{G} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}^{(k)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} {}^{t+\Delta t}\mathscr{R} - {}^{t+\Delta t}_{t+\Delta t}\boldsymbol{F}^{(k-1)} \\ \boldsymbol{b}^{(k-1)} \end{bmatrix}, \tag{3}$$

where ${}^{t}\boldsymbol{K}$ is the tangent stiffness matrix of the body in configuration ${}^{t}V$, ${}^{t+\Delta t}_{t+\Delta t}\boldsymbol{F}^{(k-1)}$ is the internal virtual work of the deformed body in configuration ${}^{t+\Delta t}V^{(k-1)}$. Within each step of the Newton-Raphson iteration, the nodal displacements are updated as

$${}^{t+\Delta t}u_i^{(k+1)} = {}^{t+\Delta t}u_i^{(k)} + \Delta u_i^{(k+1)}, \tag{4}$$

$${}^{t+\Delta t}u_i^{(0)} = {}^{t}u_i. \tag{5}$$

The vector of Dirichlet conditions, $\boldsymbol{b}^{(k-1)}$, is updated within each iteration to ensure that each incremental displacements enforce the correct boundary conditions at each increment. The update is computed via

$$\boldsymbol{b}^{(k-1)} = \boldsymbol{b} - \boldsymbol{G}\,{}^{t+\Delta t}\boldsymbol{u}^{(k-1)}. \tag{6}$$

As the nodal displacements ${}^{t+\Delta t}u_i^{(k-1)}$ are updated, the internal virtual work of the current updated configuration, ${}^{t+\Delta t}_{t+\Delta t}\boldsymbol{F}^{(k-1)}$, is updated and subtracted from the external virtual work.

At each increment, (3) is solved using an incomplete LU-preconditioned GMRES solver, which is available in MATLAB and PETSc. The iteration stops once the vector of incremental

nodal displacements is within a predefined tolerance

$$\frac{\left\|\Delta \boldsymbol{u}^{(k)}\right\|_2}{\left\|\Delta \boldsymbol{u}^{(0)}\right\|_2} < \text{tol.} \tag{7}$$

3 Crack Formation and Propagation Algorithm

The cracks form and propagate when some part of the brittle material exceeds its critical strain. In the network based model, the overstrained bonds are split [15]. In the FEM based approach, we split overstrained nodes and generate new edges to introduce cracks. In order to determine the overstrained nodes, we calculate the nodal strain tensor by averaging the value of strains from the neighboring elements. The principal strain is the largest eigenvalue of the strain tensor, and its eigenvector is the maximum strain direction. We collect a set of overstrained nodes by comparing their principal strains with the fracture critical strain threshold. For each overstrained node, we split edges that are most orthogonal to the maximum strain direction. Depending on the location of the node, we propose some rules:

1. If a node u_1 is an interior node, we split two edges with minimum angles to the orthogonal direction of the maximum strain. u_1 is the splitting end of the two edges.

   (a) If the two edges belong to the same element, we will discard one with a larger angle and pick a different edge.

   (b) If the other end of a chosen edge, vertex u_2, is on the boundary or crack surface, u_2 will also be a splitting end. This is to prevent the case where two chunks are connected by a single vertex.

2. If a node v_1 is a boundary node, we select one edge with the minimum angle to the orthogonal direction of the maximum strain.

   (a) If the chosen edge is an interior edge, v_1 will be the splitting end. Otherwise, no edge will be split.

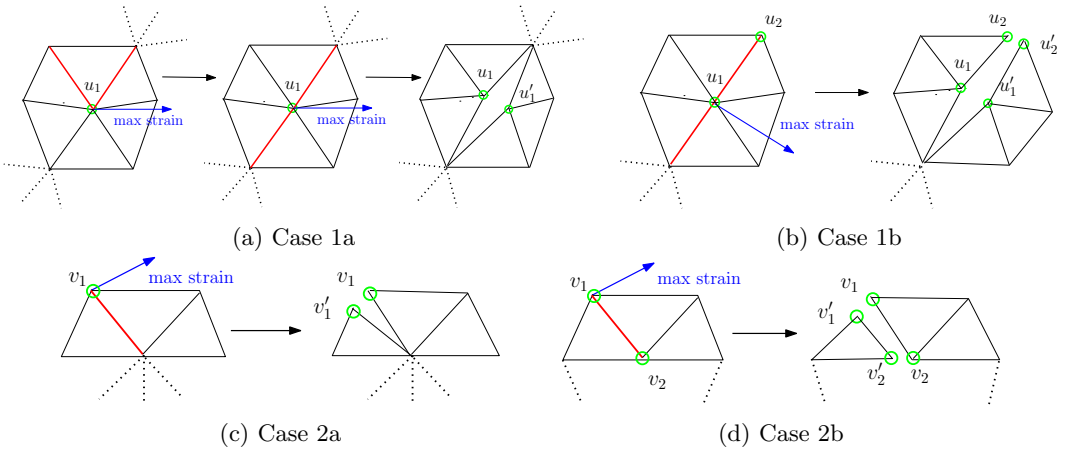
   (b) Similar to case 1 if the other end of the chosen edge, vertex v_2, is on the boundary or crack surface, v_2 will also be a splitting end.

After each node is processed, we update the element connectivity list and other related data structures, as is shown in Figure 1.

3.1 Collision Detection

When cracks form and the mesh breaks into fragments, the mesh may fold and the fragments may overlap, creating unphysical states. Introduction of proper constraints resolves this issue. It provides a physics-based interaction model between fragments and the cracked material and leads to the propagation of cracks.

Before elimination, the overlapping and folding should be detected first. A standard way is to do pairwise triangles collision detection, in which triangles have at least one edge on crack surface. In order to reduce the number of pairwise comparison, the computational field is



(a) Case 1a (b) Case 1b

(c) Case 2a (d) Case 2b

Figure 1: Illustrations of the mesh splitting process.

divided into coarse grids, and elements are assigned to each grid based on where the coordinates of their center fall into.

The detection process sweeps from upper left corner grid block towards lower right corner grid block, and only elements in right, lower or right lower diagonal adjacent grid would be detected pairwise for each grid block.

For each pair of triangles, first of all, every vertex of each element is examined if it is inside of the other one. This step checks 6 vertices using relative coordinates within each triangle, and would preclude succeeding examinations in most overlapping situations. However, there are cases that triangles are just intersecting each other by edges, thus, we need to check edge-to-edge intersections pair-wisely, and this line segments intersection checking happens between 9 pairs.

In cases of stiff material, both the deformation and the overlapped parts are very small. The round-off error makes it difficult to assert whether these elements are overlapped. Presetting a "threshold" value helps to resolve this problem. By adding this controlling parameter to the algorithm, we adjust the precision of the overlap detection. This is crucial for the simulation because adding too few constraints would impair the correct physical system we would like to obtain, while adding too many constraints would result in an over-determined state of the system, eventually leading to a failure.

3.2 Contact Constraints

For frictionless contact problems, the contact condition, known as a Kuhn-Tucker condition, can be written as

$$g_N \leq 0,\ p_N \leq 0 \ \text{ on } \Gamma_c,\ g_N p_N = 0, \tag{8}$$

where g_N is the normal component of the gap function, and p_N is the normal component of contact stress.

We adopt the method of Lagrangian multipliers to resolve the contact problem. The contact contribution and its variation can be written as:

$$\Pi_c = \int_{\Gamma_c} \lambda_N g_N \, dA, \ \ C_c = \int_{\Gamma_c} (\lambda_N \delta g_N + \delta\lambda_N g_N) \, dA, \ , \tag{9}$$

where λ_N denotes the Lagrangian multiplier which can be identified as the contact pressure p_N, and δg_N is the variation of the normal contact gap function [16]. The Lagrangian multipliers add constraints contribution to the weak formulation of the solids in contact, and prevent further penetration between overlapping elements and push already overlapped triangles back to separated status.

Equation (9) can be generally discretized by introducing discretization for the gap function in each contact area Γ_c^i, that

$$g_N^i = \boldsymbol{C_i}^T \boldsymbol{u}, \qquad \boldsymbol{C} = [\boldsymbol{C}_1|\boldsymbol{C}_2|...|\boldsymbol{C}_{n_s}], \qquad \boldsymbol{\Lambda} = (\lambda_1, \lambda_2, ..., \lambda_{n_s}), \tag{10}$$

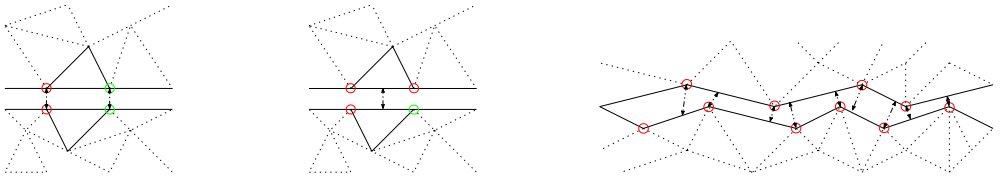
which $\boldsymbol{C}$ depends on the choice of discretization, thus

$$\int_{\Gamma_c} \lambda_N g_N \, dA \;\rightarrow\; \boldsymbol{\Lambda}^T \boldsymbol{C}^T \boldsymbol{u}, \quad \int_{\Gamma_c} \lambda_N \delta g_N \, dA \;\rightarrow\; \delta \boldsymbol{u}^T \boldsymbol{C} \boldsymbol{\Lambda}, \quad \int_{\Gamma_c} \delta \lambda_N g_N \, dA \;\rightarrow\; \delta \boldsymbol{\Lambda}^T \boldsymbol{C} \boldsymbol{u}, \tag{11}$$

The constraints can be arranged into the KKT matrix as

$$\begin{bmatrix} \left({}_t^{t+\Delta t}\boldsymbol{K}_L + {}_t^{t+\Delta t}\boldsymbol{K}_{NL}\right) & \boldsymbol{G}^T & \boldsymbol{C}^T \\ \boldsymbol{G} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda}_D \\ \boldsymbol{\lambda}_c \end{bmatrix} = \begin{bmatrix} \mathscr{R} - {}_t^t\boldsymbol{F} \\ \boldsymbol{b} \\ \boldsymbol{g_c} \end{bmatrix}, \tag{12}$$

There are several different approaches to discretize the gap function (10) along contact surfaces. For example, one could use node-to-node contact, node-to-segment contact, or segment-to-segment contact [17], as is shown in Figure 2.



(a) Node-to-node. (b) Segment-to-segment. (c) Node-to-segment.

Figure 2: Different contact constraint discretizations.

In the situation for brittle material, the crack surfaces are complicated, thus searching for all contact constraints on surfaces becomes a difficult task. In order to enhance time efficiency, we utilize features of *half-edge* to detect pairwise collision between elements, and collect contact constraint between each pair. Due to possible variety of collision scenarios, we adopt node-to-segment contact relation exclusively to adapt to different situations.

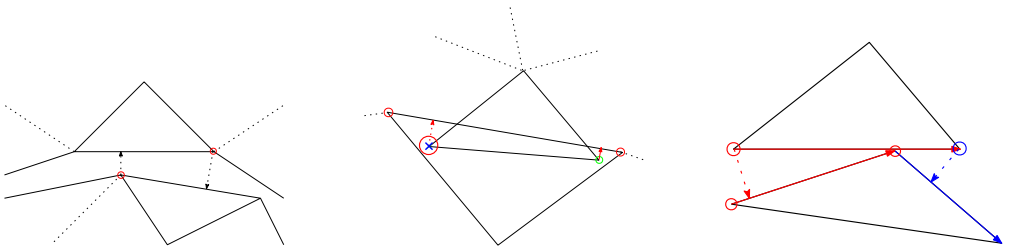
To obtain the smallest yet effective set of node-to-segment constraints, we propose the following rules for choosing master segment and slave node, as is shown in Figure 3:

1. Each pair of overlapped elements could have no more than 2 node-to-segment contact constraints; each element could have no more than 1 contact constraints in which it acts as master segment or slave node.

2. The master segment must be a boundary edge. The slave node must have the smallest distance towards the opposite element among all feasible nodes, and that distance must not exceeds the preset detection precision threshold, where the distance is negative in case of penetration.

3. If there are more than one feasible contact constraint pair, we compare the angle between the master segment and two outgoing edges of slave node, and choose the pair of node-segment with the smallest angle, and the angle must not exceed the preset threshold.

4. If all above rules does not satisfy, there will be no constraints subscribed for the given pair of overlapped elements.



(a) 2 constraints per pair. (b) Choose the smallest gap. (c) Choose the smallest angle.

Figure 3: Rules on choosing contact.

The Lagrangian Multipliers are obtained in a predict-and-correct manner, which use predicted position without contact constraints for both node x_b and segment (x_a, x_c) to calculate the surface norm $\boldsymbol{n}$ and the gap function g_N, which are calculated in the following steps:

1. Project slave node x_b on the master edge (x_a, x_c) as x_b' and calculate ξ

$$x_b' = \xi x_a + (1-\xi)x_c, \qquad (x_b' - x_b) \cdot (x_a - x_c) = 0. \tag{13}$$

2. Calculate the normal gap distance as a linear relation for displacement u_a, u_b, u_c

$$g_N = (\xi(x_a + u_a) + (1-\xi)(x_c + u_c) - (x_b + u_b)) \cdot \boldsymbol{n}d. \tag{14}$$

3. Assemble the coefficient of $\boldsymbol{u}$ into $\boldsymbol{C}$ and constant into $\boldsymbol{g_c}$ in (12).

As is shown in Figure 4, this node-to-segment contact will reduce to a node-to-node contact if ξ is 0 or 1 in some cases.

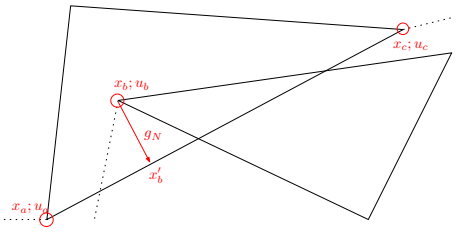


Figure 4: Contact constraint calculation.

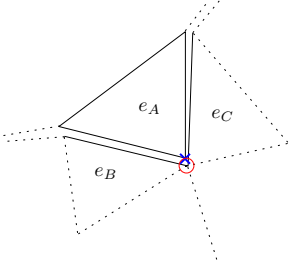


Figure 5: Redundant constraints.

Even if we apply these rules to how contact constraints are added, there could still be numerous duplicate or redundant contact constraints. For example, a duplicate constraint

occurs when adding degenerated node-to-segment contact, which is essentially a node-to-node contact, that the slave node is exactly the same vertex sharing by adjacent elements. Redundant constraints occur when the mesh breaks up totally at one vertex, when 3 or more elements are in node-to-node contact with one another, and vertices constraints between v_a and v_c may be derived from constraints between v_a and v_b, and constraints between v_b and v_c, as shown in Figure 5

Redundant constraints would result in a singular or badly conditioned KKT system. To resolve the problem, we firstly collect all contact constraints and apply QR factorization to extract the largest linear independent rows from the contact constraints matrix. After that, we add the deducted contact constraint matrix $\boldsymbol{C}$ to the KKT system.

4 Numerical Results

In this section, we present several numerical results with the proposed finite-element fracture model. We start with a verification of the FEM code for linear elasticity, and then present results for the fracture model.

4.1 Verification of the Finite Element Code

We verify our finite element code using a method of manufactured solution applied to an L-shaped body, as shown in Figure6(a). The displacement field is prescribed in polar coordinates as

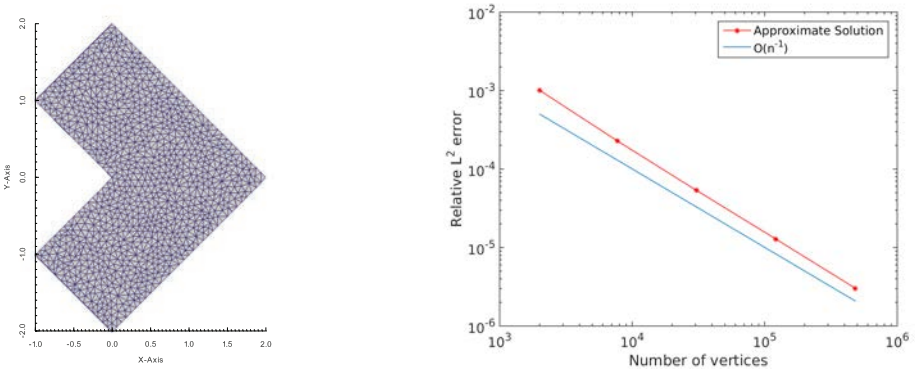
$$\begin{aligned} u_r(r,\theta) &= \frac{1}{2\mu} r^{\alpha} \left[-(\alpha+1)\cos((\alpha+1)\theta) + (C_2 - (\alpha+1))\, C_1 \cos((\alpha-1)\theta) \right], \\ u_\theta(r,\theta) &= \frac{1}{2\mu} r^{\alpha} \left[(\alpha+1)\sin((\alpha+1)\theta) + (C_2 + \alpha - 1)\, C_1 \sin((\alpha-1)\theta) \right]. \end{aligned} \tag{15}$$

where $\alpha \approx 0.544483737$ is the solution to the equation $\alpha \sin(2\omega) + \sin(2\omega\alpha) = 0$ with $\omega = 3\pi/4$, $C_1 = -(\cos((\alpha+1)\omega)) / (\cos((\alpha-1)\omega))$, and $C_2 = (2(\lambda+2\mu)) / (\lambda+\mu)$ [1]. This test case contains a strain/stress singularity at the reentrant corner located at $(0,0)$, and hence is useful for testing the initiation of fractures.

Figure 6(b) shows the linear convergence of relative L^2 displacement error for several levels of mesh refinement. The linear convergence is indicative of the strain singularity, which appears in the reentrant corner at $(0,0)$, as seen in Figure 7.

4.2 2D Simulation of Fracture

To test the finite element fracture code, we simulate a square plate made of a brittle material (ceramic or glass), which has small defects on the top and bottom sides (see Figure 8). Exerting increasing deformation along the x direction on the left and right sides, we observe the formation and growth of cracks around the defects. In this simulation, the deformation displacement dx is varied based on the number of cracks at each step. If the number of cracks is larger than 2, we reduce the displacement dx. Otherwise, if there is no crack occurring in a series of steps, we increase the displacement dx. The physical parameters for this simulation are as follows: Young's Modulus is 5 GPa, the Poisson ratio is 0.18, and the critical breakdown strain is 1.3817×10^{-4}. We use a mesh whose minimum triangle area is 0.0005 and minimum triangle angle is $\pi/6$. The boundary conditions are gliding boundary conditions on the left and right along the y direction, which is a fixed dx for the x component while the y component is free, and



(a) (b)

Figure 6: (a) The initial material configuration for the L-shaped structure. This particular discretization contains 1003 vertices and 1878 elements. (b) Convergence plot of the L-shape displacements compared to number of vertices.

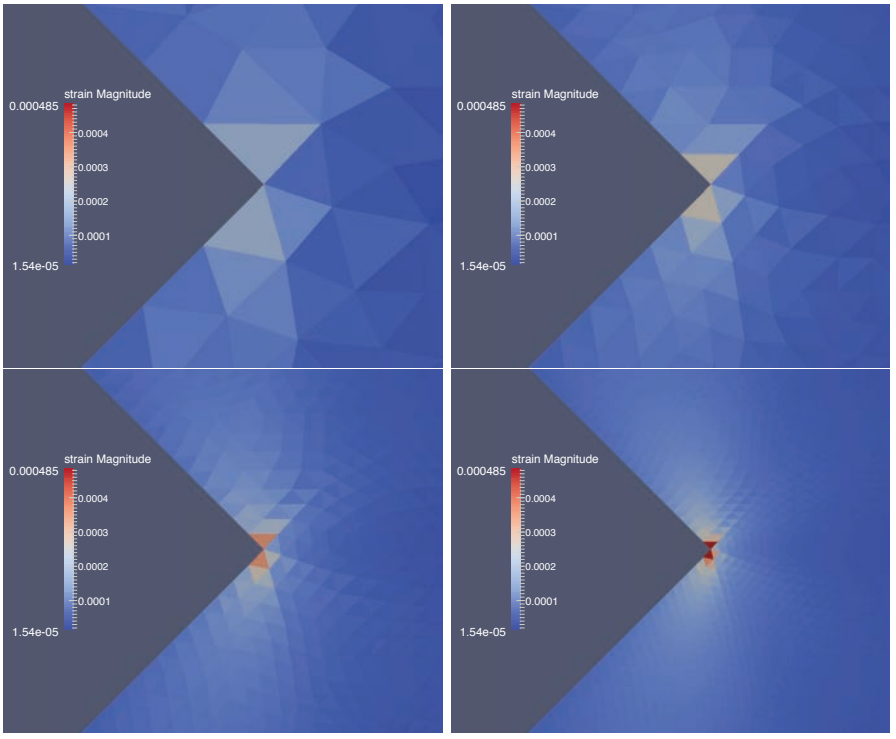


Figure 7: Close up view of reentrant corner and the Frobenius norm of the linear strain tensor for various levels of refinement.

the free boundary condition everywhere else. At each simulation step, we modify the gliding Dirichlet boundary condition for increased displacements on the left and right sides of the mesh. We then solve the updated-Lagrangian formulation for the nonlinear elasticity problem using

the quasi- Newton method. A crack processing routine is applied to the mesh to open cracks around overstrained nodes, based on the principal direction of the stress tensor of the node, while the stress is an averaged value from the vertex's surrounding elements. To ensure the system's finite element discretized virtual work plus Lagrangian multiplier matrix to be non-singular or not badly scaled, we add extra boundary conditions for isolated elements. In our current case, because of lacking contact constraint of elements touching each other, we only placed two fixed points for each isolated piece of glass. In the end, we check if the crack number exceeds the maximum number of cracks prescribed. If so, we reduce the displacements dx at one step and re-run the procedure using the old data; otherwise we keep the deformation to be unchanged and run step 2 and 3 for a release of energy through crack formation until there are no more cracks occurring. Figure 8 shows some sample results of the fracture of a brittle plate under tension.

To test the robustness of the model, we simulate a brittle disk that has a hole in the center. The hole is in contact with an expanding softer material core, which is not displayed; see Figures 9. While exerting increasing expansion on the inner core, which is passed by contact condition towards the outer disk, cracks begin to form around the hole. The applied deformation of the inner core is such that the inner part of the disk completely disintegrates into grains, forming the so-called comminuted zone, while the outer part of the disk remains intact.

Under increased deformation of the inner core, the disk undergoes complete disintegration

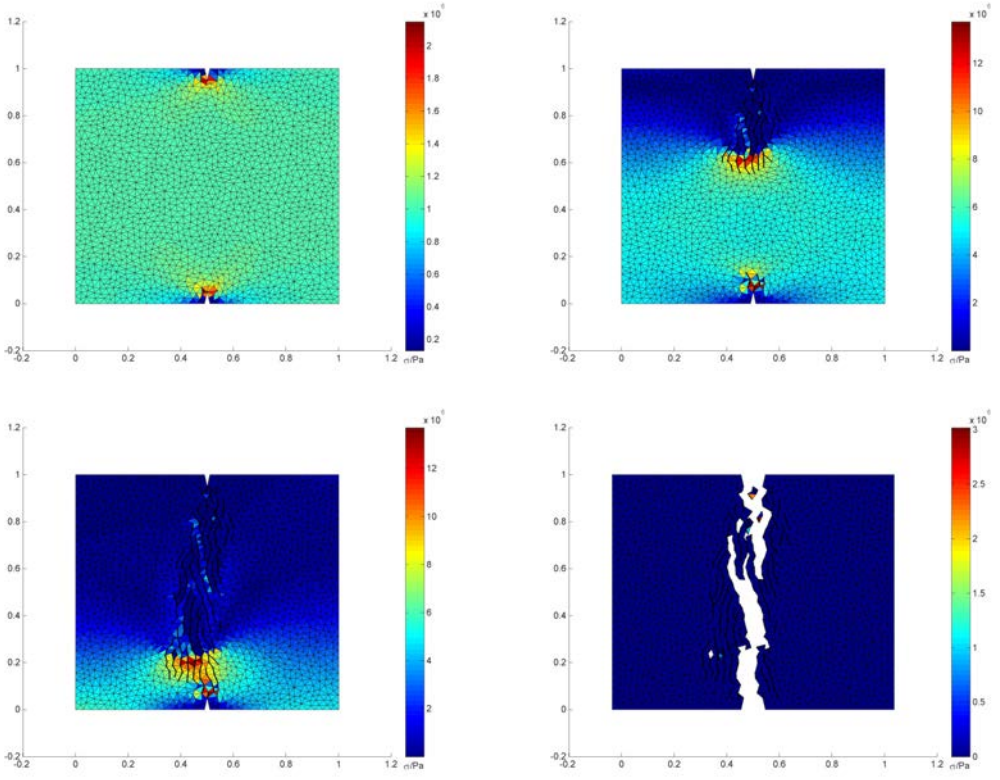


Figure 8: Fracture of brittle plate under tension.

Figure 9: Fracture of brittle disk under core expansion.

into disjoint fragment (see Figures 10). This simulation, performed using a coarse initial mesh, gives results that qualitatively agree with previously published data. The robustness of the algorithm for the selection of constraints still need some improvements for for highly refined meshes.

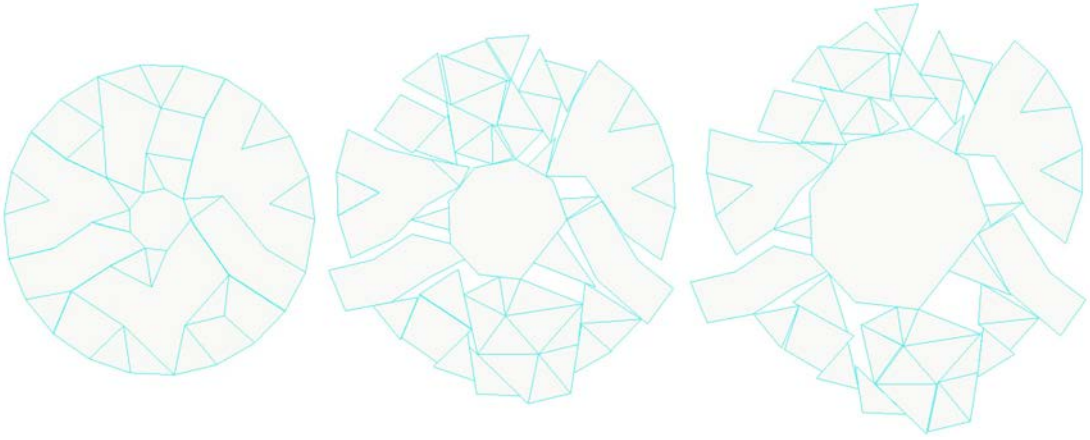


Figure 10: Complete disintegration of brittle disk under core expansion.

5 Conclusions

A numerical method for brittle fracture based on a nonlinear finite element approach has been developed, implemented, and tested. The method incorporates ideas from spring network models into the finite element framework. Geometric considerations, such as collision detection and the boundary conditions required to resolve them, are also discussed in depth. The fracture mechanism behaves independently of the underlying finite element function spaces used. The model has been applied to 2D simulations of the formation and propagation of cracks in brittle materials, and the fracture and fragmentation of stretched and compressed materials. Future investigations will concentrate on extending the algorithm to three dimensional solids and two-dimensional surfaces in three-dimensional space.

Acknowledgements The authors would like to thank Prof. James Glimm and Dr. Michael Greenfield for insightful discussions. Research was partially supported by US Army Research Office under contracts #W911NF0910306 and GS04T09DBC0017.

References

[1] J. Alberty, C. Cartensen, A. Funken, and R. Klose. Matlab implementation of the finite element method in elasticity. *Computing*, 69:239–263, 2002.

[2] E. Barbieri and M. Meo. A meshless cohesive segments method for crack initialization and propagation in composites. *Advanced Composite Materials*, 18(1):45–63, 2011.

[3] G. I. Barenblatt. The mathematical theory of equilibrium cracks in brittle fracture. *Advances in Applied Mechanics*, 7(1):55–129, 1962.

[4] K.-J. Bathe, E. Ramm, and E. Wilson. Finite element formulations for large deformation dynamic analysis. *Int. J. Numer. Meth. Engrg.*, 9:353–386, 1975.

[5] P. D. Beale and D. J. Srolovitz. Elastic fracture in random materials. *Physical Review B*, 37(10):5500, 1988.

[6] G. T. Comacho and O. M. Computational modeling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33(20):2899–2938, 1996.

[7] C. Daux, N. Moës, J. Dolbow, N. Sukumar, and T. Belytschko. Abritrary branched and intersecting cracks with the extended finite element method. *Int. J. Numer. Meth. Engrg.*, 48:1741–1760, 2000.

[8] R. De Borst. Numerical aspects of cohesive-zone models. *Engineering Fracture Mechanics*, 70(14):1743–1757, 2003.

[9] T.-P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *Int. J. Numer. Meth. Engrg.*, 84:253–304, 2010.

[10] P. Meakin, G. Li, L. Sander, E. Louis, and F. Guinea. A simple two-dimensional model for crack propagation. *Journal of Physics A: Mathematics and General*, 22(9):1393, 1989.

[11] N. Moës and T. Belytschko. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics*, 69:813–833, 2002.

[12] J. Pereira, C. A. Duarte, and X. Jiao. Three-dimensional crack growth with hp-generalized finite element and face-offsetting methods. *Computational Mechanics*, 46(3):431–453, 2010.

[13] J.-H. Song, T. Menouillard, and A. Tabarraei. Explicit dynamic finite element method for failure with smooth fracture energy dissipations. *Mathematical Problems in Engineering*, 2013, 2013.

[14] C. V. Verhoosel, M. A. Scott, R. de Borst, and T. J. Hughes. An isogeometric approach to cohesive zone modelling. *Int. J. Numer. Meth. Engrg.*, 87(1-5):336–350, 2011.

[15] H. Wei and R. Samulyak. Mass-conservative network model for brittle fracture. *J. Coupl. Syst. Multiscale Dynamics*, 2:79 – 90, 2014.

[16] P. Wriggers. Finite element algorithms for contact problems. *Archives of Computational Methods in Engineering*, 2,4:1–49, 1995.

[17] P. Wriggers. *Computational Contact Mechanics*, chapter 6-9, pages 109–182. Springer-Verlag, Berlin, 2006.

[18] X.-P. Xu and A. Needleman. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42(9):1397–1434, 1994.