INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **56**:1999–2013 (DOI: 10.1002/nme.652)

An algorithm to simulate surface creation without element deletion or remeshing†

J. M. Gerken[1,*,‡], J. G. Bennett[1] and F. W. Smith[2]

[1]*Los Alamos National Laboratory, ESA-EA, MS P946, Los Alamos, NM 87545, U.S.A.*
[2]*Department of Mechanical Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A.*

SUMMARY

One of the major obstacles to modelling discrete fracture and fragmentation of materials within the finite element (FE) framework is the inability to easily model the resulting geometry changes. This paper presents an algorithm for three-dimensional element cohesion and separation implemented in the explicit FE code DYNA3D. The algorithm handles mesh definition and solution difficulties necessary for general element separation with minimal additional effort required of the user.

The algorithm modifies the standard mesh definition so that element separation does not require remeshing. The original continuity of the structure is maintained by averaging the nodal accelerations for each coincident node set then applying this average acceleration to each of the nodes in the set. When specified failure criteria for an interface are met, averaging is no longer performed, thereby allowing the coincident nodes to separate and create new surface. The new surface can be automatically incorporated in the contact definition so that the contact of these new surfaces is modelled using the same contact considerations of the original FE model.

Following a discussion of historical and current FE fracture/fragmentation models, the general framework of the algorithm is presented. Also, results are presented which show that the algorithm simulates surface creation in a variety of simulations. Published in 2003 by John Wiley & Sons, Ltd.

KEY WORDS: finite element; fracture simulation; element separation

1. INTRODUCTION

Simulating fracture has long been a goal of the numerical modelling community. The major reason for this protracted effort is that fracture is a significant occurrence in terms of structural behaviour. Fracture not only affects the deformation and load carrying capacity of the structure,

*Correspondence to: J. M. Gerken, Los Alamos National Laboratory, ESA-EA, MS P946, Los Alamos, NM 87545, U.S.A.

‡E-mail: gerken@lanl.gov

Contract/grant sponsor: Work performed at Los Alamos National Laboratory operated by the University of California for the Department of Energy; contract/grant number: W-7405-ENG-36

Received 24 August 2001
Revised 28 February 2002
Accepted 10 June 2002

Published in 2003 by John Wiley & Sons, Ltd.

it also changes the geometry. These items imply two areas that must be addressed to provide a numerical model of fracture: (i) the onset and continuation of fracture, and (ii) the geometry change/surface creation that accompanies fracture. Both of these areas have independently provided researchers ample subject matter and the coupling of the two areas is still an elusive target.

Numerical evaluation of fracture onset criteria is a somewhat mature area in which, with a reasonable amount of effort, a numerical modeller can evaluate fracture criteria to determine when a structure will fracture [1]. While many of these methods may be cumbersome to apply in a general sense (i.e. for any number of cracks occurring at arbitrary locations in a numerical model), their application has proven useful in many scenarios. Modelling geometry changes that are not a direct result of structural deformation has likewise been implemented in and applied to a wide range of numerical models with similar success. Difficulties that arise in modelling geometry changes in general, and specifically fracture, include the evaluation of failure onset and growth criteria and following geometry evolution that results from crack propagation. For example, the accurate evaluation of shear band formation resulting in material separation using a finite-element (FE) model requires exquisite knowledge of the material behaviour in the plastic regime and an FE model that can both capture and follow this behaviour.

Use of the FE method to simulate surface creation can be grouped into two categories. The first is intra-element techniques in which surface creation is modelled within an FE by modification to the elemental stiffness matrix. Typical intra-element techniques include modified interpolation functions to match discontinuities between crack faces [2] and erosion of stress bearing capacity based on continuum damage mechanics [3] or other criteria. The primary advantage of the intra-element method is that surface creation is incorporated into the element formulation and the use of these techniques is similar in complexity to the use of other FEs. The disadvantage to using these methods is the mesh distortion that can arise because of large relative motions between the surface faces and/or formation of many new surfaces that result from several cracks propagating through a structure. In addition, erosion of stress bearing capacity of elements often results in crack tip blunting that is not typical of crack propagation.

The second category is exo-element techniques in which the surface creation is explicitly modelled by separation of adjacent FEs. Common exo-element techniques include continuous element interfaces [4, 5], zero or small volume interface elements [6, 7], and interface cohesive models [8]. Another, less common, method of exo-element separation is a Lagrange multiplier approach which is commonly used to connect discrete domains of elements and has found use in parallelization of numerical schemes for elliptic partial differential equations [9]. The advantage of the exo-element technique is that the explicit handling of surface creation is similar to methods of discretization used for standard simulations in which the geometry does not change. One common disadvantage is that the mesh must usually be defined so that elements can separate from one another, either during original discretization or during the solution process. If this is addressed in the original discretization, a method to maintain the continuity of unseparated interface must also be incorporated into the FE solution.

This paper presents a method to address the non-standard mesh definition required by exo-element techniques. Specifically, this paper presents a method to modify a standard FE mesh to allow for element separation and to enforce displacement continuity across element interfaces until a failure criterion at the interface has been met. The method has been

incorporated into the explicit FE code DYNA3D [10] and is transparent to the modeller. The method allows for simple evaluation of failure criteria and provides a trusted framework for the verification and validation for numerical models of fracture and fragmentation.

2. FE FRACTURE MODELS

The three most common approaches to exo-element modelling of fracture are the continuous interface, interface element, and the cohesive models. Each provides an adequate method for surface creation with little distinction between the models. The differences between these models is somewhat tenuous in that they each incorporate an interface behaviour, evaluate failure, then modify the interface behaviour to simulate surface creation. The similarity between the interface element and cohesive models is exemplified in the work of Foulk *et al.* [11] in which cohesive models have been recast in an interface stiffness matrix form and Bolzon [12] in which the author presents a derivation of the FE equations with separate consideration of interfacial conditions at a crack face. Recasting the continuous interface method as a cohesive model in which the traction–displacement relationship is infinite at zero displacement and zero everywhere else provides the final link between the three models. Although the distinction between these three techniques is not great, the early development of each followed parallel paths with few connections and therefore, provides a convenient means of discussion of exo-element fracture techniques.

2.1. Continuous interface

The continuous interface approach is the most used method of the exo-element techniques. The reason for its popularity is the relative simplicity of its implementation. With this method, the interface between two adjacent elements remains continuous with no relative motion at the interface until fracture occurs. This model, therefore, does not affect the behaviour of the structure before fracture occurs. This model requires a computational method to selectively constrain elements together and a method for evaluation of failure criteria.

In the simplest version of this model, a crack is assumed to propagate along a symmetry plane in a two-dimensional model. As load is applied and the failure criteria are satisfied, the boundary condition constraining nodes to the symmetry plane are sequentially released and the simulated crack propagates. The appeal of this model is twofold; first the selective enforcement of boundary conditions is a simple matter given the current state of development of FE codes, and secondly, the crack path is known *a priori* so the evaluation of fracture criteria can be applied at known crack tip locations. The utility and simplicity of this method is exemplified by its availability in a wide variety of FE codes (e.g. Reference [13]).

A more complex application of the continuous interface model is constraining adjacent elements together so that cracks can propagate between two elements within the mesh. The added complexity of this method is due to the necessity of selectively constraining adjacent nodes to one another. If the crack path is known or estimated *a priori*, the added complexity can be limited to the interfaces along the path, and the fracture criteria evaluation can be done as in the symmetry plane method. This method has also been used with some success. See, for example, the work of Kanninen *et al.* [14] in which the authors use sequential node

release along a single line of double-noded elements to model crack growth in a weld induced residual stress field.

If the crack path is unknown prior to the simulation, the crack must be allowed to propagate along many different paths, with the FE solution providing the evolution of the crack path. With this method, as a crack approaches an element, it may propagate along any edge of the element. The difficulties of this method include selectively applying constraints to coincident node groups that may range in number from two to eight or more and also evaluation of fracture criteria at arbitrary locations in the mesh. While the increase in the number of nodes at a single location adds some computational complexity in terms of the amount of data that must be stored, evaluation of standard fracture criteria for complex geometry, with an arbitrary number of cracks propagating along unknown paths is, by far, the more difficult task. It has proven to be a major obstacle to developing validated models of fracture for general application. One development of this method in two dimensions has been done by Gerken [15] and applied to a simulation of fracture of high explosives [16]. The method was implemented into the implicit FE code ABAQUS/Standard and the authors used available routines to selectively constrain nodes together and evaluated fracture based on the average stress at the interface and elastic plastic fracture mechanics applied to an assumed small crack located at the interface.

2.2. Interface element

The interface element method, as its name suggests, uses an element located at the interface between two structural FEs and are incorporated into the global stiffness matrix during assembly and solution. This interface element either transfers load between the structural elements to simulate continuous material or does not transfer load to simulate free surface between the two elements. These interface elements typically have the dimension spanning the adjacent elements much smaller than other dimensions on the interface. The idea behind this type of model is that all the necessary behaviour for fracture can be handled in this interface element and the behaviour of the structure is modelled with the structural elements. In comparison with the continuous interface models, the interface element method does not increase the degrees of freedom, but will increase the size and possibly affect the numerical stability of the global stiffness matrix. These additions are balanced by the elimination of the constraint enforcement that is necessary for the continuous interface method.

These methods have gained particular popularity in the modelling of discontinuities in geologic and structural composite materials and their development has a timeframe similar to that of the development of the computational implementation of the FE method. One early development of a model for jointed rock is that of Goodman *et al.* [6] which extends the nodal linkage model of Ngo and Scordellis [17] to a two-dimensional interface element. This interface element is deformable and will transfer both compressive and shear load. If the load at the interface is tension, neither shear nor normal load is transferred across the interface. Perhaps the most seminal work in this area is that of Desai *et al.* [7] in which the authors further develop and investigate a finite thickness interface element to model soil–structure interactions and rock joints which was first presented by Zienkiewicz *et al.* [18].

There is a distinct connection between the interface element and the intra-element techniques with the main difference lying in the location and method of crack propagation. For the intra-element methods a crack may propagate through the bulk material and is simulated by

modifications to the stiffness matrix of a structural element. In the interface element method, a crack is only allowed to propagate along the edges of a structural element and the behaviour of the interface elements is intended to simulate the behaviour of a crack.

2.3. Cohesive zone

The cohesive zone or cohesive interface method has become a popular model in recent years and has shown much promise in the simulation of fracture processes using a generalized local approach. The basis of these models stems from early work in fracture mechanics in which it was postulated that a zone of stress bridging advances ahead of a crack tip [19, 20]. These ideas are incorporated into the cohesive model by defining a traction–displacement ($\sigma - \delta$) relationship in which the traction applied to the adjacent elements is a function of the relative displacement at the interface. These methods have been implemented into standard FE analysis by including the surface traction as a mixed boundary condition or conversely, by converting the $\sigma - \delta$ relationship to an element stiffness matrix and directly incorporating the interface as an element.

The roots of these models can be traced back to the use of spring (i.e. force–displacement) elements to constrain adjacent nodes together. The most influential work on recent efforts is the work of Needleman [21] and Xu and Needleman [8] in which the authors have presented a phenomenological $\sigma - \delta$ relationship taking into account both shear and normal deformations at the interface. Subsequent investigation of models of the type presented in References [8, 21] have shown that the details of fracture can be accurately reproduced if the mesh adequately resolves the material cohesive zone [22]. Analytical and experimental investigation of interfacial $\sigma - \delta$ relationships for particulate composites has recently been undertaken by Liu *et al.* [23, 24]. This work provides a physical basis for use of $\sigma - \delta$ relationships to model fracture.

2.4. Discussion

The use of these techniques to simulate fracture in two-dimensional models appears to have reached a critical mass as indicated by the exponential increase in their application to a wide variety of problems, with the implementation usually resulting from user-written programs. Although this increase has been rapid in recent years, there still exists a vacuum in commercially available codes for methods of simulating fracture. There has been a limited amount of effort to apply a fracture modelling technique using the available capabilities of codes written for general application [15, 16, 25, 26] but direct incorporation of these methods as a standard modelling technique is still lacking.

Application of these techniques to three-dimensional simulation is far lagging its two-dimensional counterpart. Although the application of the fracture techniques to three-dimensional problems is conceptually straightforward, the lag appears to stem from the difficulties inherent in the logistics of the discretization of the structure and the additional computational expense necessary for these models. While commercially available pre-processing tools are available and will do an adequate job of discretizing most structures for FE simulation, the construction of a mesh, for example, in which each element is defined by unique node numbers is not a simple matter. In addition, selective application of inter-element constraints is not a well developed or widely available technique in commercial FE codes. Remeshing or mesh modification, which is an alternative to non-standard mesh definitions, is not easily extended to three dimensions and also is not widely available.

As recently as 1999, Ortiz and Pandolfi [27] state that 'The three-dimensional tracking of dynamic cracks in solids undergoing large-scale plasticity has received scant attention in the FE literature.' The work for which this statement served as an introduction was the development of a three-dimensional model for the cohesive technique and has been the platform of many recent applications in three-dimensional simulations [28–30]. In addition, other element separation methods have been developed and implemented in research and government developed FE codes (e.g. DYNA3D and LUSAS) primarily in the form of deformable interface methods [31–33]. Separation techniques for a node pair have been available for some time in many widely available FE codes but their general application to three dimensions is less than straight forward.

Another obstacle to modelling failure is the development of accurate, generally applicable, interface failure criteria. While there are many analytical and phenomenological techniques for evaluating failure, the predictive ability of these models in a numerical approximation can be inadequate. Currently, the modelling of a geometrically complex, non-linear structure, with an arbitrary number of failure locations, is a task that has only been accomplished in which the numerical model is a close approximation to the failure process itself. Because the numerical models that are commonly used to simulate structures do not closely approximate failure processes, it is likely that failure criteria for application in these scenarios will be phenomenological and therefore will need rigorous investigation to determine their envelope of utility.

3. ALGORITHM

A method that greatly simplifies the use of exo-element fracture modelling has been programmed into a research version of the FE code DYNA3D [10]. DYNA3D is an explicit FE code for analysing the transient dynamic response of three-dimensional solids and structures. The method is designed to be transparent to the modeller so that minimal effort is required to simulate material failure. It also provides a testbed to evaluate interface failure models in an explicit dynamics FE code.

This algorithm provides two principle modifications to the standard DYNA3D simulation. One is the preprocessing modification to the mesh and the second is the modification to the FE solution procedure. Shown in Figure 1 is a flow chart showing the changes in the steps for DYNA3D input processing for the fracture algorithm and also implies the necessary input from the user. In a standard DYNA3D analysis, data for all the materials are read, then data for all the nodes, elements, loads, etc. are read, and then the code proceeds with the FE solution. Implied in the flow chart, the user must indicate in the input file which groups of elements will be allowed to fracture by including a flag on the material comment card for each fracturing material. The user must also provide the data for the fracture criteria for each of these materials. All other materials are treated in the standard manner. This method allows the analyst to select the region that will be allowed to fracture and therefore, limit computational expense. After data for all the materials are read, the nodes and elements are read as usual.

Once the data for the entire mesh have been read, the meshes for the groups of elements that will be allowed to fracture are modified. Each of the elements in the group is redefined so that they have unique node numbers. This allows for each element to completely separate

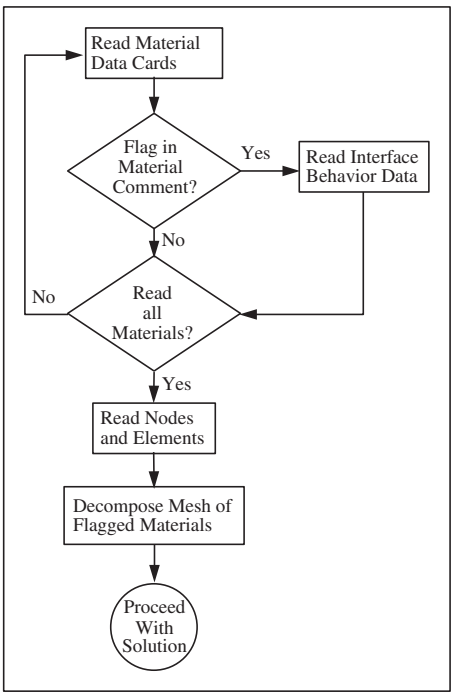

Figure 1. Flow chart showing input algorithm for fracture simulation.

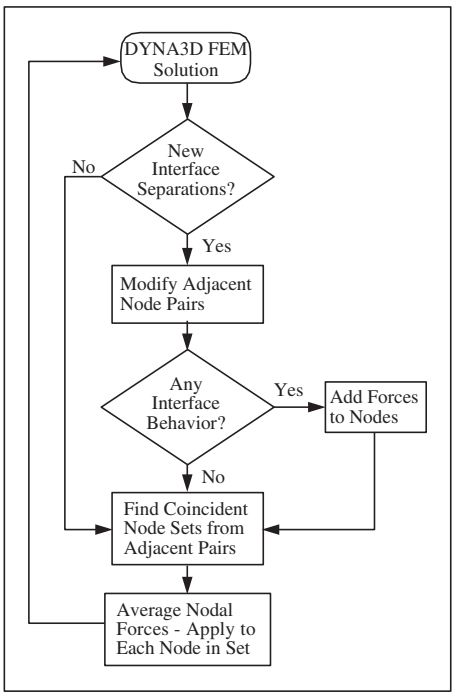

Figure 2. Flow chart showing solution algorithm for fracture simulation.

from the rest of the mesh based on the element definition at the beginning of the solution and obviates the need to redefine the mesh during the solution procedure. While this type of definition increases the number of nodes to 8 times the number of elements in the group for the entire simulation, it works well with the solution and post-processing procedures of DYNA3D. After the element connectivity has been redefined and the necessary nodes added to the node list, displacement boundary conditions that were placed on the original nodes are applied to the added coincident nodes. The additional memory requirements include the additional space for the added nodes and, with N being the total number of interfaces in the element group, a $2 \times N$ integer array to store the two element numbers that make up the interface and an $8 \times N$ integer array to store the eight nodes that make up the interface.

Shown in the flow chart in Figure 2 is the solution algorithm for the fracture element groups. After an FE solution has been obtained, each unfractured interface in each fracture group is evaluated to determine if it should separate. For a developer, the failure criteria can be quite general with only a few things required by the general program. Information that is either available or derivable to evaluate the failure criteria include:

(i) nodal variables—acceleration, velocity, displacement, position, force, mass, etc.,
(ii) element variables—ε, $\dot{\varepsilon}$, σ, material parameters, etc.,
(iii) time—increment, total,
(iv) interface parameters and history variables.

If the failure criteria determine that the interface has failed, the coincident nodes on this interface are no longer defined as adjacent to each other and are allowed to separate. If the failure model includes interface behaviour, such as a $\sigma - \delta$ relationship, this is incorporated by including these forces in the nodal force vector of the FE solution. Once the interface failure has been evaluated, the list of adjacent interface nodes is used to group all the nodes that are coincident. Then for each of these coincident node sets, the average acceleration for the set is determined using Newton's second law as follows:

$$\mathbf{F} = \sum_i \mathbf{f}_i \tag{1}$$

$$M = \sum_i m_i \tag{2}$$

$$\bar{\mathbf{a}} = \mathbf{F}/M \tag{3}$$

where i is the number of coincident nodes in the set, $\mathbf{F}$ is the total force vector acting on the set, $\mathbf{f}_i$ is the force vector on each node in the set, M is the total mass of the set, m_i is the mass of each node and $\bar{\mathbf{a}}$ is the average acceleration of the set. The force vector, $\mathbf{f}_i$, for each node in the set is then replaced with the average force acting on the entire set as follows:

$$\mathbf{f}_i = m_i \bar{\mathbf{a}} \tag{4}$$

This method ensures that all of the nodes that are coincident maintain the same acceleration until separation occurs.

Also incorporated in the algorithm is a modification to the slidesurfaces with adaptive new definitions (SAND) contact algorithm in which the fully separated interfaces are added to the contact surface definition. The SAND contact algorithm is an automatic contact algorithm that uses external element faces as the contact surfaces and was designed to allow for the addition of element faces to the contact surface as elements fail and expose new element faces. The algorithm determines all the external element faces at the start of the simulation and then will add element faces to the contact surface as elements fail. For use with the fracture algorithm, all the faces of elements in a fracture group are included in the contact surface as external faces. This is because all the elements have unique nodes and, therefore, *appear* to be completely external. This automatically determined contact surface is modified to exclude each element face that is part of an interface. As the interface fails the two surfaces that define the interface are added to the contact surface. This is useful, for example, in fragmentation simulations in which many pieces may evolve from a single structure and subsequently interact. While this technique can be useful, it should be noted that the memory requirement and computational expense of including all the element surfaces for a fracture group in the contact algorithm preclude its use in all but the most necessary situations.

4. EXAMPLES

This section demonstrates the use of the algorithm described above and makes comparison with previous work. These include comparisons of simulation details in which similar problems are analysed with and without fracture, and comparisons of the results of simulation with another

Table I. Properties of alumina.

Young's modulus = 366 GPa
Density = 3.96 g/cc
Poisson's ratio = 0.22
Tensile strength = 310 MPa

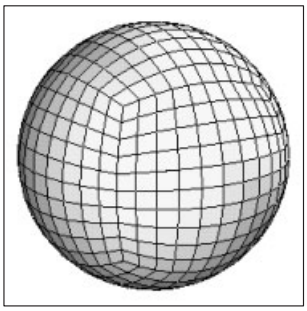

Figure 3. Mesh of ceramic ball containing 2800 elements.

implementation of a fracture algorithm. Although no study of failure criteria has been made, it is interesting to note how simple models of failure perform and help to point out deficiencies in current methods.

4.1. Fracture/no fracture comparison

To investigate the performance of the algorithm relative to a standard simulation, a sample problem has been developed. It is a simulation of 10 ms of a 0.01 m diameter ceramic ball impacting a rigid plate. The ball has an initial velocity of 75 m/s normal to the plate and 50 m/s tangent to the plate. In the co-ordinates of the simulation, the normal velocity is in the negative y-direction and the tangent velocity is in the positive z-direction. The static and dynamic coefficients of friction between all surfaces are 0.5 and 0.1, respectively. The rigid plate has been modelled as a $0.02\text{m} \times 0.02\text{m} \times 0.01\text{m}$ rigid material with 25 brick elements and has a contact stiffness similar to the stiffness of common steel. The ball has been modelled as a linear elastic material with the nominal properties of Alumina (Al_2O_3) given in Table I. The failure criterion is a tensile stress failure criterion—i.e. if the tensile stress across an interface is greater than the tensile strength of Alumina, the interface fails. Because fragmentation is expected to occur, the modified SAND contact algorithm has been used.

For comparison six different meshes of the ceramic ball were constructed. The number of elements in the six different meshes are 256, 425, 648, 2800, 8775, and 20 000. Figure 3 shows the mesh of the ceramic ball containing 2800 elements. The meshes for the other simulations are similar. Shown in the fractured meshes of Figure 4 is the 2800 element ceramic ball after impact with the rigid surface. It can be seen that the mesh has broken into several large pieces of ceramic with many small particles occurring near the impact and rear section of the ball.

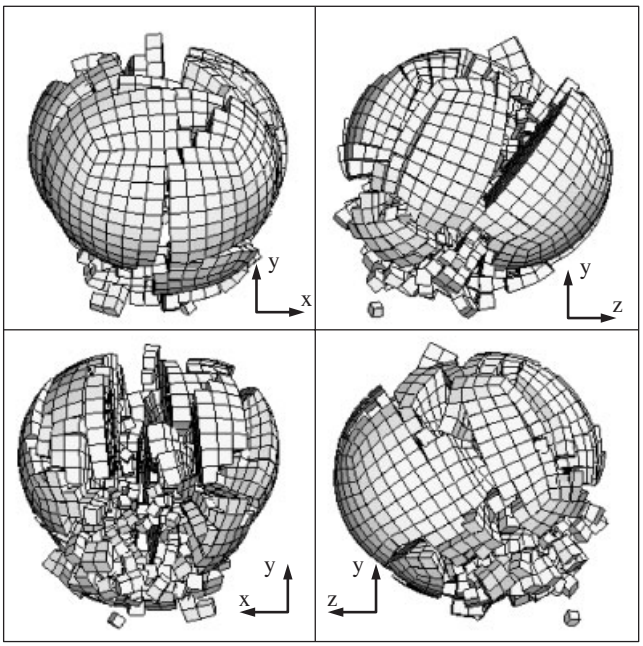

Figure 4. Fractured 2800 element ceramic ball 5.0 ms after impact.

This result is typical of all the simulations, which have typically four major features:

(i) a fracture along the centre of the ball, in the plane of the velocity vector,
(ii) a major fracture along the centre of the ball, in a plane approximately perpendicular to the plane of the velocity vector,
(iii) a major fracture plane at about the mid-section of the ball, but parallel with the rigid surface,
(iv) disperse fragmentation near the bottom rear of the ball.

The comparison of the computational details of the simulation without and with fracture and for a varying number of elements are shown in Tables II and III, respectively. In the table, the columns indicate the total number of elements, total number of nodes (Total number of elements $\times 8$ in the fracturing simulation), memory required, and time per element step ($TotalTime/(\#\ steps \times \#\ elem)$). The time per element step is the time it takes to process one element one time step and is a measure of the computational cost of a particular method of simulation.

The comparison shows that for all the simulations the fracture model is approximately 3–4 times more expensive, in terms of time per element step, than the non-fracturing model. In terms of memory, the fracture algorithm requires 3.6–4.8 times more memory than the non-fracturing algorithm. Because the fracturing algorithm increases the memory requirements in direct proportion to the size of the problem, it is expected that this memory penalty would increase as the size of the problem increases. For use in a model of a structure in which the general location of failure might be known *a priori*, the group of elements that are

Table II. Ten millisecond simulation with no fracture.

	Elements	Nodes	Memory (Mb)	Time (μs)
1	256	321	0.920	17.1
2	425	520	1.370	16.5
3	648	779	1.959	15.5
4	2800	3137	7.386	14.2
5	8775	9504	22.143	13.9
6	20000	21271	49.518	14.8

Table III. Ten millisecond simulation with fracture.

Case	Elements	Nodes	Memory (Mb)	Time (μs)
1	256	2048	3.327	60.7
2	425	3400	5.381	54.4
3	648	5184	8.076	53.4
4	2800	22400	33.847	52.5
5	8775	70200	105.04	56.0
6	20000	160000	238.66	61.6

allowed to fracture can be limited and therefore limit the computational expense of the fracture algorithm.

4.2. Comparison with cohesive zone model

As a comparison of the implementation of the algorithm presented here with that of another method, a simulation similar to that of Pandolfi *et al.* in Reference [30] has been performed. Presented by Pandolfi *et al.* in Reference [30] are the results and simulation of the impact of a notched three-point-bend steel beam experiment illustrated in Figure 5. The simulation used a cohesive interface model of fracture and showed remarkably good comparisons with the experimental results. The authors presented excellent agreement with experimentally observed crack initiation time and velocity for one of the simulations presented.

To compare results of the algorithm presented here, the mesh in Figure 6 was constructed. The colours indicate element material groups, with the purple being the impactor and the light blue and tan are the specimen. The mesh size along the centreline of the specimen is approximately $1.4\,\text{mm} \times 1.6\,\text{mm}$, which corresponds to the coarser mesh size used by Pandolfi *et al.* The specimen was modelled as a bi-linear elastic–plastic steel with a tangent modulus in the plastic region of 50 GPa. This is in contrast to the J2 plasticity with power-law hardening, rate dependency and linear thermal softening material model used by Pandolfi *et al.* The impactor has been given a constant velocity of $10\,\text{m/s}$ with contact conditions between the specimen and the impactor. In the figure, the tan group has been modelled with the separation algorithm. The failure criterion is a normal stress failure, as was used in the previous section. The failure normal stress is 2 GPa, which corresponds to the maximum cohesive stress used by Pandolfi *et al.*

The results of this simulation showed, as expected, that the crack propagates straight ahead from the pre-notch. The most notable difference in the two simulations is the crack initiation

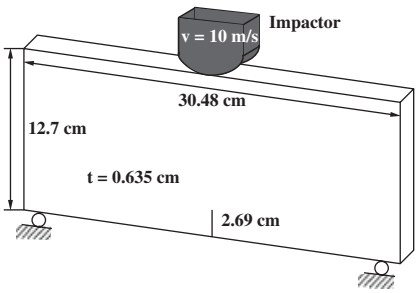

Figure 5. Experiment of Pandolfi *et al.* in Reference [30].

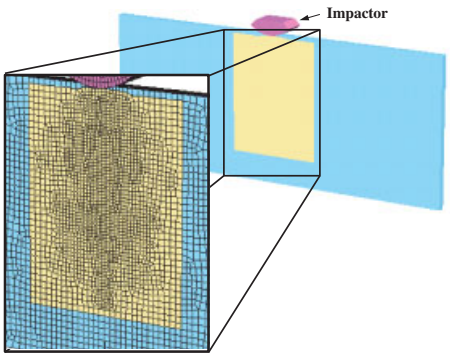

Figure 6. Mesh containing 16 384 total brick elements.

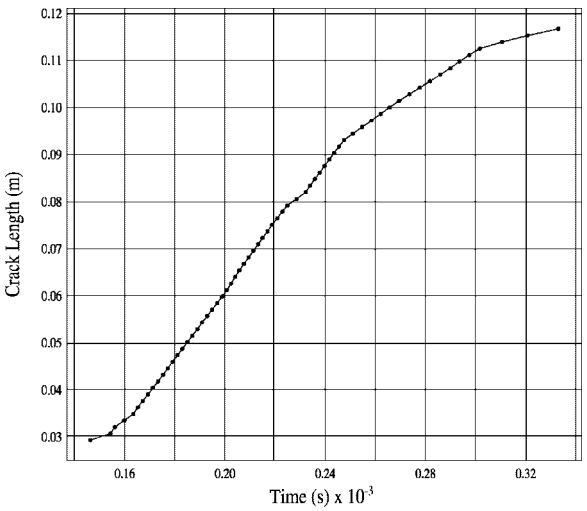

Figure 7. Length vs time for the simulated crack growth.

time. The initiation time for this simulation is at 0.134 ms whereas, the cohesive zone model predicts initiation at approximately 0.11 ms for the cohesive zone model, which is approximately the same as that seen in the experiment. The subsequent crack growth predicted by this model is shown as length vs time in Figure 7 and as crack velocity vs time in Figure 8. In comparison with the cohesive zone model, these results show some similarities. Notably, the crack velocity in both simulations shows a slow start followed by a more rapid growth rate. Also, a rough estimate of the average crack speeds for both models shows that after this slow start, the crack speed is about 700 m/s for about 0.1 ms.

While exact similarities between the two simulations are limited, the model presented here has reproduced the general nature of the cohesive zone model. The differences might be explained by the differences in the particulars of the models. The cohesive zone model dissipates energy in both the material model and the cohesive zone model while the model presented

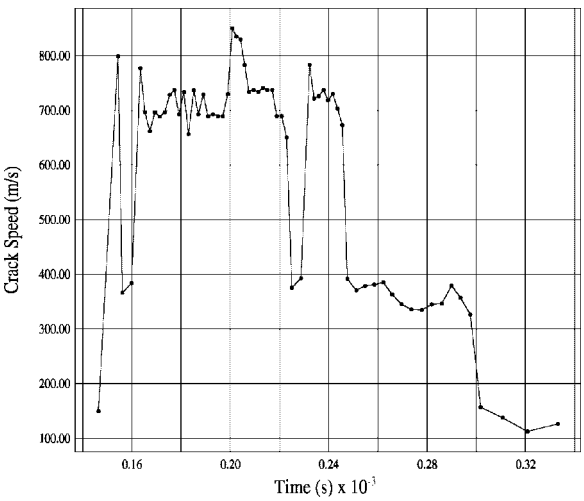

Figure 8. Crack speed vs time for the simulated crack growth.

here only dissipates energy in the material model. It is a plausible that a higher fidelity failure criterion, such as the cohesive model of Pandolfi *et al*., would more accurately reproduce the experimental results.

5. SUMMARY AND CONCLUSIONS

In the past, many methods have been employed to model fracture within the FE framework. In previous developments, intra-element approaches such as continuum damage as well as more complex methods such interface separation have been used with some success. While many examples of their use are available, their widespread use in simulation is limited. This seems to stem from the complexities involved in the implementation and use of such algorithms. The algorithm presented in this paper is designed to simplify the use of interface separation fracture modelling while providing a testbed for much needed generally applicable failure criteria.

The algorithm, implemented in the explicit FE code DYNA3D, handles the mesh definition and solution difficulties with minimal additional effort required of the user. The method uses a standard input file for an FE simulation of a structure modelled with standard 8 node brick elements. The user must indicate the element group(s) that will be allowed to have surface(s) initiate and propagate and also must input interface failure parameters for each element group. The algorithm then modifies the standard mesh definition so that unique node numbers define each element. The original continuity of the structure is maintained by averaging the nodal accelerations for each coincident node set then applying this average acceleration to each of the nodes in the set. When the failure criteria for an interface are met, averaging is no longer performed, thereby allowing the coincident nodes to separate. In addition to surface creation, modifications to the SAND contact algorithm will include the newly created surfaces in the contact surface definitions. This allows the contact of these new surfaces to be modelled using the same contact considerations of the original FE model.

While an investigation of the failure criteria has not been studied here, a simple, normal stress failure criterion has demonstrated that the algorithm will reproduce some of the features of fracture, while also showing some deficiencies. It is expected that faithful use of a higher fidelity failure model would more accurately reproduce the details of fracture. Also, in a comparison with a non-fracturing standard simulation, the algorithm is shown to be 3–4 times more computationally expensive. This is a reasonable penalty for the ability to simulate surface creation in a simulation.

ACKNOWLEDGEMENTS

The work presented in this paper is an account of work performed at Los Alamos National Laboratory operated by the University of California for the Department of Energy under contract W-7405-ENG-36.

REFERENCES

1. Anderson TL. *Fracture Mechanics Fundamentals and Applications* (2nd edn). CRC Press: New York, 1995; 565–594.
2. Gao HJ, Klein P. Numerical simulation of crack growth in an isotropic solid with randomized internal cohesive bonds. *Journal of the Mechanics and Physics of Solids* 1998; **46**(2):187–218.
3. Lemaitre J. *A Course on Damage Mechanics* (2nd edn). Springer: New York, 1996.
4. Liaw BM, Kobayashi AS, Emery AF. Double noding technique for mixed mode crack propagation studies. *International Journal for Numerical Methods in Engineering* 1984; **20**(5):967–977.
5. Ingraffea AR. Nodal grafting for crack propagation studies. *International Journal for Numerical Methods in Engineering* 1977; **11**(7):1185–1197.
6. Goodman RE, Taylor RL, Brekke TL. A model for the mechanics of jointed rock. *Journal of the Soil Mechanics and Foundations Division* (*Proceedings of the American Society of Civil Engineers*) 1968; **98**:637–659.
7. Desai CS, Zaman MM, Lightner JG, Siriwardane HJ. Thin-layer element for interfaces and joints. *International Journal for Numerical and Analytical Methods in Geomechanics* 1998; **8**(1):19–43.
8. Xu XP, Needleman A. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids* 1994; **42**(9):1397–1434.
9. Farhat C, Roux FX. Method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**(6):1205–1227.
10. Englemann BE, Whirley RG. *DYNA3D*: *A Nonlinear*, *Explicit*, *Three-Dimensional Finite Element Code for Solid and Structural Mechanics.* UCRL-MA-107254 Rev. 1, Lawrence Livermore National Laboratory, Livermore, CA, 1993.
11. Foulk JW, Allen DH, Helms KLE. Formulation of a three-dimensional cohesive zone model for application to a finite element algorithm. *Computer Methods in Applied Mechanics and Engineering* 2000; **183**(1–2):51–66.
12. Bolzon G. Hybrid finite element approach to quasi-brittle fracture. *Computers and Structures* 1996; **60**(5): 733–741.
13. Hibbitt, Karelsson and Sorenson Inc. *ABAQUS/Standard Version 6.1.* Pawtucket, RI, 2001.
14. Kanninen MF, Brust FW, Ahmad J, Abou-Sayed IS. The numerical simulation of crack growth in weld-induced residual stress fields. In *Residual Stress and Stress Relaxation*, Kula E, Weiss V (eds). Plenum Press: New York, 1982; 227–248.
15. Gerken JM. An implicit finite element method for discrete dynamic fracture. *Master's Thesis*, Colorado State University, Fort Collins, CO, 1996.
16. Gerken JM, Bennett JG, Smith FW. Numerical simulation of the mechanically coupled cook-off experiment. *Journal of Materials and Technology*: *Transactions of the American Society of Mechanical Engineers* 2002; **124**(2):266–273.
17. Ngo D, Scordellis AC. Finite element analysis of reinforced concrete beams. *Journal of the American Concrete Institute* 1967; **64**(3):152–163.
18. Zienkiewicz OC, Best B, Dullage C, Stagg KG. Analysis of nonlinear problems in rock mechanics with particular reference to jointed rock systems. In *Proceedings of the Second Congress of the International Society for Rock Mechanics*, Belgrade, Yugoslavia, 1970; 8–14.
19. Barenblatt GI. Mathematical theory of equilibrium cracks. *Advances in Applied Mechanics* 1962; **7**:56–129.
20. Dugdale DS. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids* 1960; **8**:100–104.

21. Needleman A. A continuum model for void nucleation by inclusion debonding. *Journal of Applied Mechanics* 1987; **54**(3):525–531.
22. Camacho GT, Ortiz M. Computational modelling of impact damage in brittle materials. *International Journal of Solids and Structures* 1996; **33**(20–22):2899–2938.
23. Liu C, Stout MG, Asay BW. Stress bridging in a heterogeneous material. *Engineering Fracture Mechanics* 2000; **67**(1):1–20.
24. Liu C. HE fracture tests—determination of the stress-bridging law. LA-UR-01-4445 2001, Los Alamos National Laboratory, Los Alamos, NM.
25. Hoff R, Rubin CA, Hahn GT. A new finite-element technique for modelling stable crack growth. *Engineering Fracture Mechanics* 1986; **23**(1):105–118.
26. Kirkpatrick SW, Giovanola JH, Simons JW. DYNA3D analysis of dynamic fracture of weldments. In *Proceedings of the International Seminar on Quasi-Impulsive Analysis*, Wakiyama K, Tachibana E, Imai K, Kitano T (eds). Osaka, Japan, 1996; A6.1-15.
27. Ortiz M, Pandolfi A. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International Journal for Numerical Methods in Engineering* 1999; **44**(9):1267–1282.
28. Pandolfi A, Krysl P, Ortiz M. Finite element simulation of ring expansion and fragmentation: the capturing of length and time scales through cohesive models of fracture. *International Journal of Fracture* 1999; **95**(1–4):279–297.
29. Ruiz G, Ortiz M, Pandolfi A. Three-dimensional finite-element simulation of the dynamic Brazilian tests on concrete cylinders. *International Journal for Numerical Methods in Engineering* 2000; **48**(7):963–994.
30. Pandolfi A, Guduru PR, Ortiz M, Rosakis AJ. Three dimensional cohesive-element analysis and experiments of dynamic fracture in C300 steel. *International Journal of Solids and Structures* 2000; **37**(27):3733–3760.
31. Reedy ED, Mello FJ, Guess TR. Modelling the initiation and growth of delaminations in composite structures. SAND95-3070, Sandia National Laboratories, Albuquerque, NM, 1996.
32. Alfano G, Crisfield MA. Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues. *International Journal for Numerical Methods in Engineering* 2001; **50**(7):1701–1736.
33. Smith FW, Bennett JG, Gerken JM. Software tools for dynamic cracking. LA-UR-00-5493, Los Alamos National Laboratory, Los Alamos, NM, 2000. http://www.lanl.gov/orgs/esa/ea_gerken/SoftwareTools/Software Tools.pdf [26 Feb. 2002].