

# Cutting and Fracturing Models without Remeshing

Chao Song, Hongxin Zhang, Yuan Wu, and Hujun Bao

State Key Lab of CADCG, Zhejiang University,  
Hangzhou 310027, P.R. China

{songchao, zhx, wuyuan, bao}@cad.zju.edu.cn

**Abstract.** A finite element simulation framework for cutting and fracturing model without remeshing is presented. The main idea of proposed method is adding a discontinuous function for the standard approximation to account for the crack. A feasible technique is adopted for dealing with multiple cracks and intersecting cracks. Several involved problems including extended freedoms of finite element nodes as well as mass matrix calculation are discussed. The presented approach is easy to simulate object deformation while changing topology. Moreover, previous methods developed in standard finite element framework, such as the stiffness warping method, can be extended and utilized.

**Keywords:** Physically based animation, finite element method, fracturing model, without remeshing.

## 1 Introduction

In industry design and digital entertainment, the simulation for cutting, fracturing models and their deformation has been widely applied. Hence relevant research has been an import area in computer graphics, virtual reality and computer aided design.

One of the key issue on this topic is how to deal with the changing of shape topology during simulating cutting and fracturing in which crack initialization and crack growth are included. Various methods based on dynamics and statics were proposed in recent research. There are mainly three kinds of simulation techniques on space discretizing, which are mass-spring system, finite element method (FEM) and meshless method. Meanwhile, the research and applications of FEM and meshless methods are received increasing concern because of their high controllability and stability.

The standard FEM simulates fracturing and cutting problems by remeshing models around a growing crack. However, remeshing is computationally expensive and lots of physical parameters for new element nodes have to be calculated. On the other hand, it is increasingly difficult to guarantee the simulation stability. In this paper, we leverage an extended finite element method that adjusts the element approximation function to account for element discontinuous based on standard FEM framework. In our proposed approach, the remeshing procedure is not a necessary step. Moreover, many previous computation techniques, such as techniques of accessory calculation, can be easily utilized.

Our work has three main contributions:

- We propose an approach of attaching additional degree of freedoms (DOFs) on element nodes and make it pretty easily to be implemented.

- We improve the approach for tackling multiple cracks and intersecting cracks to meet the demand in computer graphics and virtual reality.
- We adopt the stiffness warping technique to enhance our simulation framework for compensating the nonlinear factors of deformation.

Moreover, in order to optimize the calculation and stability, several relevant implementation issues, such as reasonable choices of the mass matrix, are discussed in this paper.

## 2 Related Work

In the literature of computer animation, Terzopoulos and Fleischer [1,2] used a distance threshold between two nodes to judge fracture during simulating viscoelastic and plastic deformation. They demonstrated this technique with sheets paper and cloth that could be torn apart. Later, the mass-spring system was applied to simulate the fracturing models, including work of Norton *et al.* [3], simulating the mud crack pattern by Hirota *et al.* [4]. The most important virtue of the mass-spring system in fracture simulation is relied on the simple data structure which leads to simple implementations. But two of its disadvantages are calculation instability and limited reality of resultant animations.

In 1999, O'Brien *et al.* did excellent work in simulating fracture models using standard FEM framework. They adopted separation tensor and remeshing finite element mesh of model to successfully simulate brittle fracture [5], ductile fracture [6] and surface cracks pattern [7]. Müller [8] did similar work to implement real-time simulation of brittle fracture.

Molino and Bao as well as their colleagues [9,10] originally presented a virtual node algorithm to deal with the troubles of remeshing in standard FEM. The algorithm duplicates cutting elements that meet specific conditions and the remeshing procedure is not necessary. The virtual node algorithm developed the simulation for topology changed by FEM. The drawback of the algorithm is in the complexity of geometry data structure and the strictly limitation in that the smallest possible unit would be individual nodes. Wicke *et al.* [11] presented a finite element on convex polyhedra to simulating cutting models. Their method also does not need additional remeshing, but expensive calculations are required.

Pauly *et al.* [12] applied the meshless method to fracture simulation in computer graphics. Although it has high cost in calculating approximation functions, meshless method has great advantages in dealing with point sample models and large strain deformation.

Fracture and cutting have been studied extensively in the mechanics literature. There are a lot of related work on the extended finite element method which has an initial form for small strain and statics application [13,14]. The method has been developing in mechanics and even the virtual node algorithm can also be included. In this paper, we adopt the foundational theory and develop it for application of movie industry and virtual reality.

The paper is organized as follows. Our method is mainly presented in Section 3. Section 4 discusses the solutions of multiple cracks and intersection cracks. In Section 5,

a brief consideration on simulation control is given. Section 6 describes our implementation and several examples are provided. Paper summary and conclusions are presented in the last section.

### 3 Method Description

In general, a model in our computation framework is represented as a domain  $\Omega \subset R^3$ . For each point  $p \in \Omega$ , the movement is represented

$$p : \Omega \times R \rightarrow R^3 : (\mathbf{X}, 0) \mapsto \mathbf{x}(\mathbf{X}, t),$$

where  $\mathbf{X}(p)$  is the material coordinates,  $\mathbf{x}(\mathbf{X}, t)(p)$  is the location coordinates of point  $p$  at time  $t$ . Let  $\mathbf{u}(p) = \mathbf{x}(p) - \mathbf{X}(p)$  be the displacement. If there are  $n$  cracks in the model, we represent them as  $\Gamma_{ci}, i \in [1, n]$ . The simulation goal is getting all points coordinates after the model movement or deformation. In addition, as for cutting simulation, the cutting face can be taken as a crack. It will not be distinguished in this paper.

In our simulation, a model is discreted into tetrahedron finite elements without regarding of the cracks. But our method can be utilized in other types of elements without any difficulty. The dynamics equation of simulation is a PDE based on time

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}(\mathbf{x} - \mathbf{X}) = \mathbf{f}_{\text{ext}}, \quad (1)$$

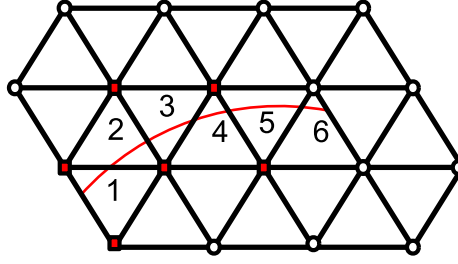
where  $\mathbf{M}$ ,  $\mathbf{C}$ ,  $\mathbf{K}$  and  $\mathbf{f}_{\text{ext}}$  are mass matrix, damp matrix, stiffness matrix and extern force vector of nodes, respectively. In the above equation, symbol  $\mathbf{x}$  and  $\mathbf{X}$  are the column vectors which are composed by space coordinates and material coordinates of all nodes, respectively. All these coefficients are calculated as in standard FEM, that is,  $\mathbf{M} = \sum \mathbf{m}^e$ ,  $\mathbf{C} = \sum \mathbf{c}^e$ ,  $\mathbf{K} = \sum \mathbf{k}^e$  and  $\mathbf{f}_{\text{ext}} = \sum \mathbf{f}_{\text{ext}}^e$ . The matrices with superscript  $e$  stand for the element distribution matrices.

#### 3.1 Adding Extended Freedoms

In order to make the description clearer, we call the element with a crack as *crack element* (e.g., the element 1 to 6 in Figure 1). In this section, we will explain our simulation method by limiting one crack in one crack element without loss of generality. We will discuss more on the solutions for multiple cracks in one crack element in the next section.

Our crack propagation is per-element based one, i.e., a crack advances on a complete element within a time-step. Similar to add virtual nodes in the virtual node algorithm, we simply add extended freedoms for related nodes in our method instead. To achieve this goal, let  $K_{nT}$  denote the node set related to the crack  $n$ . Regarding for the crack  $n$ , we attach additional freedoms  $\mathbf{a}_i = \{a_{ix}, a_{iy}, a_{iz}\}^T$  on node  $i \in K_{nT}$  (Figure 1). The key problem is how to construct  $K_{nT}$  corresponding to crack  $n$ . Hence a two-step procedure is adopted in our framework:

- (1) After mapping the crack to the initial configuration of a model, for each edge of the model, determine whether there exists intersection point between it and crack  $n$ . If it is true, add the intersection point into  $S_n$ . And then



**Fig. 1.** Add extended freedoms for nodes. The red curve represents a crack. Nodes represented by red rectangle need to add extended freedoms. Nodes represented by circle without filling are not necessary to add extended freedoms.

- (a) if the intersection point is a vertex, add the vertex into  $K_{nT}$ ,
- (b) if the intersection is not a vertex, add two vertices of the edge into  $K_{nT}$ ;
- (2) Find the crack fringe curve  $L_{nf}$ . For each point  $q \in S_n$ , if  $q \in L_{nf}$  and meet two conditions:  $q$  is not in the model surface and is not in the other crack surface. Then,
  - (a) if  $q$  is a vertex (element node), remove the vertex from  $K_{nT}$ ;
  - (b) if  $q$  is not a vertex, remove two vertices of the edge including  $q$  from  $K_{nT}$ .

Compared with previous methods of adding extended finite element in mechanics literature, our above method is simpler in three dimension simulations.

### 3.2 Crack Element Approximation

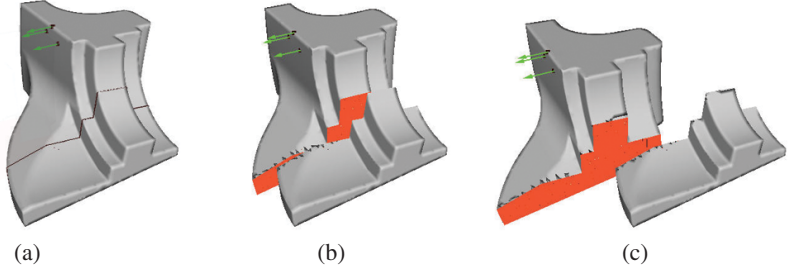
After adding extended degree of freedoms, a crack element has at least one node with extended freedom. In our method, we assume that the displacement field can be decomposed into a continuous part and a discontinuous part, i.e.,  $\mathbf{u} = \mathbf{u}^{cont} + \mathbf{u}^{disc}$  in the crack element. Hence an approximating function of displacement field presented by [15] can be adopted:

$$\mathbf{u}(\mathbf{X}) = \sum_{I=1}^4 N_I(\mathbf{X}) \mathbf{u}_I + \sum_{I \in K_T} N_I(\mathbf{X}) [H(f(\mathbf{X})) - H(f(\mathbf{X}_I))] \mathbf{a}_I, \quad (2)$$

where  $f(\mathbf{X}) = 0$  is an implicit surface representation of the crack,  $N_I(x)$  is tetrahedron shape function. In Equation 2, the first term  $\mathbf{u}^{cont} = \sum_{I=1}^4 N_I(\mathbf{X}) \mathbf{u}_I$  is a continuous part and has the same shape as standard FEM approximation. And the second term  $\mathbf{u}^{disc} = \sum_{I \in K_T} N_I(\mathbf{X}) [H(f(\mathbf{X})) - H(f(\mathbf{X}_I))] \mathbf{a}_I$  is a discontinuous part that indicates the difference between the two sides of crack due to the *Heaviside function*

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (3)$$

As for a normal element, the standard FEM approximation is performed actually. In fact,  $\mathbf{u}^{disc}$  is a zero vector when the Heaviside function is constant in the whole element. It is worth noting that there always exists movement independence between the two



**Fig. 2.** Movement independence. (a) initial cutting model. (b) and (c) removing one part from the model.

sides of a crack. This is critical and is a primary criterion for judging the feasibility of a simulation method of cutting and fracturing. An example in Figure 2 shows the movement independence of our simulation.

### 3.3 Crack Element Calculating

In a crack element, the initial coordinates, initial velocity and initial accelerate of extended freedom is  $(\mathbf{0}, \mathbf{0}, \mathbf{0})$ . The calculation of the corresponding mass of extended freedom and the element mass matrix is vital and can affect the movement independence.

In standard FEM, there are mainly two choices for setting mass matrix. One is the *average mass matrix* which is setting matrix as a diagonal matrix, whose nonzero value is the model mass divided by the total number of the nodes. The other one is the *compatibility mass matrix* which is calculated by element approximation and density. It is straightforward to extend the average mass matrix by setting the mass value of corresponding the all extended freedoms be zero. Unfortunately, our experiences show it may make the coefficient matrix of solve system be singular when multiple cracks are occurred in a model and there are more than one crack in the same element.

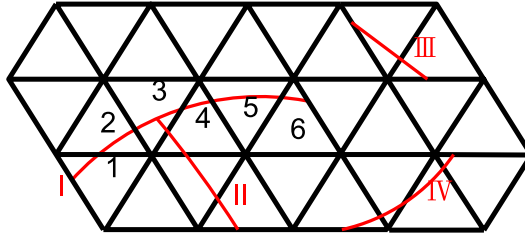
We also tried to make the corresponding mass of extended freedom the same value as the mass of corresponding node as [10]. But it is evident to increase total mass of the model and simulation error. More seriously, the movement independence is failure in our simulation framework.

To sum up the above arguments, compatibility mass matrix have to be adopted. As for a crack element, the element mass matrix should be calculated as follows

$$\begin{cases} \mathbf{m}_{ij}^{uu} = \int_{\Omega^e} N_i N_j d\omega^e \\ \mathbf{m}_{ij}^{ua} = \int_{\Omega^e} N_i N_j(\mathbf{X}) [H(f(\mathbf{X})) - H(f(\mathbf{X}_j))] d\omega^e \\ \mathbf{m}_{ij}^{aa} = \int_{\Omega^e} N_i(\mathbf{X}) [H(f(\mathbf{X})) - H(f(\mathbf{X}_i))] N_j d\omega^e \end{cases} \quad (4)$$

where  $\mathbf{m}_{ij}^{\xi\eta}(\xi, \eta = u, a)$  denotes the corresponding mass between the  $\xi$ -th freedom of node  $i$  and the  $\eta$ -th freedom of node  $j$ , and  $\Omega^e$  is the space domain of the element.

Strictly following Equation (4) can guarantee the movement independence. But it makes difficulty on applying Equation (4) in interactive application. That is, we do not know a model will fracture or be cut during deformation and choose average mass



**Fig. 3.** Multiple cracks in a model. Red curves indicate cracks.

matrix at beginning of simulation. In order to avoid this problem, we calculate the mass matrix as follows. At beginning of simulation, we adopt the average mass matrix if there is no crack in the model. When the model has cracks, evaluate the mass matrix by calculating  $\mathbf{m}_{ij}^{ua}$  and  $\mathbf{m}_{ij}^{au}$  only if  $i = j$ , and keep the others  $\mathbf{m}_{ij}^{\xi\eta}(\zeta, \eta = u, a)$  be zero. Our experience shows that our approach is feasible and decreases the cost of calculation.

Beside the evaluation of mass matrices, the calculation on crack elements is almost the same as in standard finite element method by applying the approximation function in Equation (2). The above treatment will be only invoked if the element includes cracks. Thus a small amount of additional calculation are required.

## 4 Multiple Cracks and Intersecting Cracks

Commonly there are multiple cracks and intersecting cracks in a model. We therefore improve the methods presented by mechanics literature to meet the demand of computer graphics applications.

### 4.1 Independent Cracks

In section 3, we have already known that an element is not affected by a crack if Hevi-side function is constant in it. So if there are multiple cracks in a model and any two cracks of them is far, not in the same element and not intersecting. We call these cracks be independent (independent cracks, as if III and IV in Figure 3). We can deal with these cracks separately. Certainly, corresponding crack elements crack set is different for every crack. The tackling can be expressed as

$$\mathbf{u}(\mathbf{X}) = \sum_{I=1}^4 N_I(\mathbf{X}) \mathbf{u}_I + \sum_{n=1}^{n_c} \sum_{I \in K_{nT}} N_I(\mathbf{X}) [H(f_n(\mathbf{X})) - H(f_n(\mathbf{X}_I))] \mathbf{a}_I^n \quad (5)$$

where  $n$  is the crack flag in the element, implicit function  $f_n(\mathbf{X}) = 0$  represents the surface of crack  $n$ ,  $\mathbf{a}_I^n$  is the corresponding extended freedom of crack  $n$  at node  $I$ . And term  $K_{nT}$  is the node set in which every node need to add extended freedom of corresponding crack  $n$ .

## 4.2 Multiple Cracks and Intersecting Cracks

Independent cracks can grow to intersect in the same element during the simulation. Regarding for this problem, Equation (5) may be failed. Budyn *et al.* [16] proposed a method of intersecting cracks for engineering application. Our experiments proved applying their method for computer graphics application will be not able to guarantee the movement independence. Hence we have to extend and improve their methods.

For the reason of clarity but without loss of generality, we only explain the occurrence when there are two cracks in one element. Obviously, the two cracks can be intersecting or not. But we treat both cases unitedly.

In our simulation, it is necessary to assume that the two cracks cut the element of in order. As for intersecting crack, the later crack face  $c$  is stopped by the former crack face  $C$  (see Figure 4). But when adding corresponding extended freedom, the later crack should be assumed to whole cutting the element. Firstly, we must judge the two crack mutual location according to their implicitly representing. And then we apply the following equation [16]:

$$\mathbf{u}(\mathbf{X}) = \sum_{I=1}^4 N_I(\mathbf{X})\mathbf{u}_I + \sum_{I \in K_{CT}} N_I \varphi_{cI}(\mathbf{X})\mathbf{a}_I^c + \sum_{I \in K_{CT}} N_I(\mathbf{X}) [H(f_C(\mathbf{X})) - H(f_C(\mathbf{X}_I))]\mathbf{a}_I^C, \quad (6)$$

where

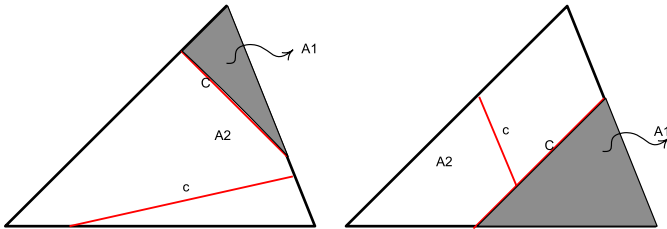
$$\varphi_{cI}(\mathbf{X}) = \begin{cases} H(f_C(\mathbf{X})) - H(f_C(\mathbf{X}_I)), & \mathbf{X} \in A1, \\ H(f_c(\mathbf{X})) - H(f_c(\mathbf{X}_I)), & \mathbf{X} \in A2. \end{cases}$$

Here the extended freedoms  $\mathbf{a}_I^c$ ,  $\mathbf{a}_I^C$  are matching crack  $c$  and  $C$ .

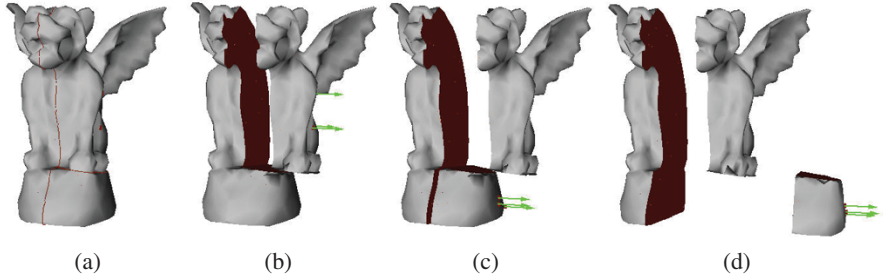
It is worth noting that we apply Equation (6) for the elements not only including intersecting cracks as in literature [16]. In order to simulation for computer graphics and virtual reality, we improve the approach as following.

As for the case that two cracks are not intersecting in one element, Equation (6) is used directly. When two cracks intersect and coincide, they will fusion to one crack and stop growth. The element in which the fusion points of the two cracks may be tackled by taking it into intersecting crack element.

In addition, if other elements adjacent with the multiple cracks element meet: (i) be cut by crack  $C$  and (ii) have no less than one node with extended freedoms corresponding crack  $c$ . It also must be tackled with Equation (6) even though it is not cut by more



**Fig. 4.** Two cracks in one model. The filled region is A1, and the remaining part is A2.



**Fig. 5.** The movement of intersecting cracking. (a) initial cutting model. (b) moving the right upper part. (c) moving the right lower part. (d) the finally movement.

than one crack. Element 1, 5 and 6 in Figure 3 must be tackled with equation (6) and the above method. Figure 6 shows the movement independent in a model with intersecting and cutting crack.

As for more than two cracks in an element, we deal with them according to Equation (6) similar with two cracks. In fact, the method is not increasing calculation other than judge and save the mutual location and easy to be implemented. Moreover, the restriction is not the same strictly as virtual nodes algorithm in that every fracture part must include no less than one node. In our simulation experiments, there is no trouble caused by the problem.

## 5 Simulation Control

In general, most of the previous research in mechanics is as for engineering application dealing with *small deformation*. We therefore want to enhance our simulation method for *large deformation*. In the method described in section 3, Cauchy strain is adopted to obtain stiffness matrix. The linear calculation will lead to non-realistic results. Therefore the stiffness warping technique for compensating the nonrealistic are applied in our simulation framework. As for fracture control, we mainly use the results from the previous research.

### 5.1 Stiffness Warping in Crack Element

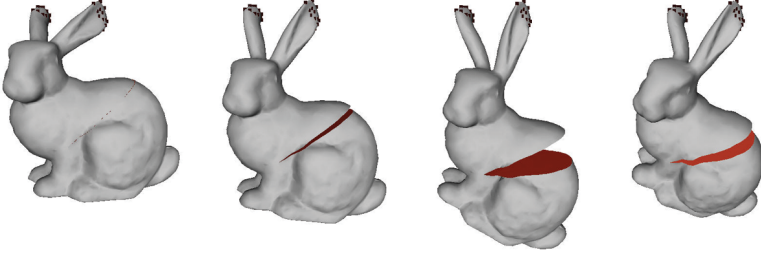
By applying the stiffness warping technique proposed in [17] to our simulation framework, Equation (2) is rewritten as

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \sum_i \sum_j \mathbf{R}_{ij}^e \mathbf{k}_{ij}^e (\mathbf{R}_{ij}^{e-1} \mathbf{x}_j - \mathbf{X}_j) = \mathbf{f}_{ext}, \quad (7)$$

where  $j$  is node index,  $i$  is index of the element which includes node  $j$ ,  $\mathbf{R}_{ij}^e$  is a corresponding sub-matrix of the rotation of element  $i$  matching node  $j$ ,  $\mathbf{k}_{ij}^e$  is a corresponding sub-matrix of stiffness matrix of element  $i$ . Please note that the rotation matrix is between current location and initial location.

As for a crack element, it is necessary to calculate rotation of every part taken into by cracks separately. This can be obtained by performing polar decomposition for the





**Fig. 6.** Movement sequence of *Bunny* with cutting crack

conversion matrix of location. Obviously, in the crack element, the coefficient of matrix calculation in Equation (7) should be performed according to every part because of their rotation may be different. In the example showed in Figure 6, the *bunny* with a cutting crack moves under gravity while the two ears of the model are constrained. Our algorithm generally provides visual-pleasant during the simulation of model deformation.

## 5.2 Fracture Control

As for the model being cut, initial cracks are provided by users while the simulation procedure is performed automatically. When simulating fracturing model, the Rankine condition of maximal principal stress is used to control element fracture as [8,9]. The stress of an element is smoothed by volume weighted averaging with adjacent elements before the principal stress is obtained by doing eigenvalue decomposition to stress matrix. If the maximal principle stress is greater than a tensile threshold (positive), or the minimal principle stress is less than a compressive threshold (negative), the element is cut by a crack controlled along the principle direction.

When the distance between one crack fringe and the other crack face is less than a threshold, the former directly propagates to intersect the latter. In addition, if the distance and the orientation angle between two crack fringes are less than threshold values, the two cracks fusion into one crack. After that, the tackle method is performed that has been discussed in Section 3 and 4.

## 6 Implementation and Examples

According to the description above, we implemented a prototype of model simulation in a compatible PC with Intel CPU. Similar to [18], we leverage a variant of implicit integration method for dynamics simulation in this paper. The main steps are:

1. Update velocities of all nodes according to  $\mathbf{v}^{i+1} = \mathbf{v}^i + \Delta \mathbf{v}^i$ .
2. Update displacements of all nodes according to  $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{v}^i \cdot \Delta t$ .
3. Processing collision and determining the boundary condition.
4. Solve difference of velocities  $\Delta \mathbf{v}^{i+1}$  by

$$(\mathbf{M} + \Delta t \mathbf{C} + \Delta t^2 \mathbf{K}) \Delta \mathbf{v}^{i+1} = \Delta t (\mathbf{f}_{ext} - \mathbf{C} \mathbf{v}^i - \sum_i \sum_j \mathbf{R}_{ij}^e \mathbf{k}_{ij}^e (\mathbf{R}_{ij}^{e-1} \mathbf{x}_j - \mathbf{X}_j) - \Delta t \mathbf{v}^i).$$

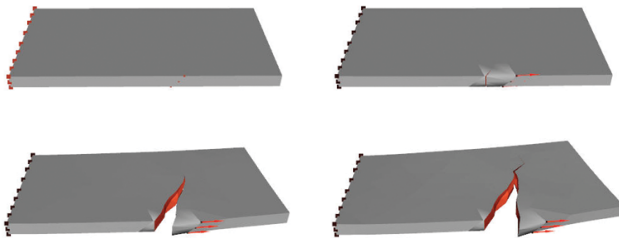
In our simulation framework, the total number of elements is invariant while the number of freedoms of a model may be altered. Real displacement of every node are the displacement of initial freedoms according to Equation (2). The displacement of extended freedoms can be used for calculating point positions in cracks. It is worth mentioning that the dimension of the final linear system will rise up when the number of crack elements is rising. However, the only amended the approximation function of crack element, common element calculation is according with standard FEM. It follows that the calculating cost will not increase a lot. As our simulation framework can be tailored to emulate the virtual node algorithm (VNA), there is no significant advantage in computation performance compared with VNA. But regarding for the implementation of data structure and geometry processing operators, our method is simpler and easier to integrate with existing techniques of standard FEM.

In this paper, adjusting the approximation of crack element is based on the *theory of partition of unity* whose feasibility has been thoroughly discussed in [19,20]. In addition, we utilize a linear FEM framework with stiffness warping to simulate large deformations for the applications of computer graphics. These treatments ensure the stability of our simulation method. Compared with standard FEM, our method do not need any remeshing. Therefore the instability factor due to the occurrence of long and thin tetrahedra while cutting and fracturing models in standard FEM can be avoided.

Several computation examples (e.g., Figure 2, 5, 6, 8 and 9) are provided in this paper to demonstrate the performance of our proposed method. Figure 7 is a plane constrained on the left side while pulling on the right side. As expected, when the pulling force surpass a pre-set threshold, the plane is slit open into to two parts.

Figure 8 is an example of cutting model. We cut a plane into several pieces firstly, and then drop it on top of a sphere. During the period of falling and colliding with other objects, the movement of all of the parts is independent. In this example, the initial DOFs is 4671 and the DOFs is 7026 after cutting. The simulation refresh rate in this example can reach interactive speed, and it takes less than 1 second for every frame in average.

An example of a *flyman* model objecting to bump on the face is showed in Figure 9. The initial flyman, which is a complete model without any cracks, breaks into several pieces during the movement. The initial model has 450 nodes and 1320 elements. At the end of simulation, the model has total 2139 DOFs and every frame costs 700ms in average.



**Fig. 7.** The fracturing *plane* subject to tension. Red arrows represent the direction of pulling force.

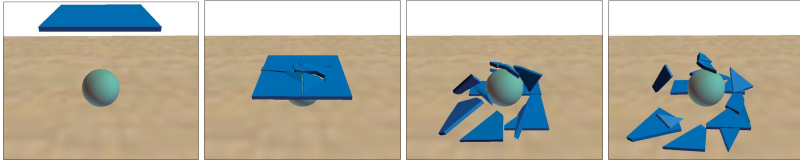


Fig. 8. A *plane* being cut into several parts and falling down

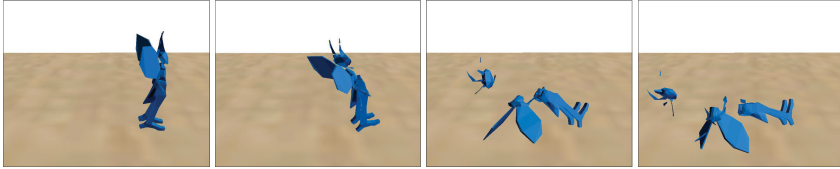


Fig. 9. A *flyman* subjected to bump on face

## 7 Conclusion and Future Work

In this paper, a simulation framework for cutting and fracturing is proposed. A novel approximation function for crack element is employed on the basis of standard FEM. Several key problems, such as tackling for the mass matrix, multiple cracks and intersecting cracks and compensating deformation, are dissolved. Compared with standard FEM, our proposed method does not need any remeshing, and can decrease computation cost as well as increase simulation stability.

However, there are still several limitations in our simulating. The first is that cutting or fracturing models in our method is element-based. When crack only cut an element half, our method in this paper cannot deal with it currently. The others, such as control fracture more efficiently and collision between difference fracturing parts, should be accomplished. We will explore more on those topics in near future.

**Acknowledgements.** This project is supported in partial by the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (No. 705027), 973 Program of China (No.2002CB312102), and NSFC (No.60505001).

## References

1. Terzopoulos, D., Fleischer, K.: Deformable models. *The Visual Computer* 4, 306–331 (1988)
2. Terzopoulos, D., Fleischer, K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In: *Computer Graphics (SIGGRAPH 1988 Proceedings)*, pp. 269–278 (1988)
3. Norton, A., Turk, G., Bacon, B., Gerth, J., Sweeney, P.: Animation of fracture by physical modeling. *The Visual Computer* 7, 210–217 (1991)
4. Hirota, K., Tanoue, Y., Kaneko, T.: Generation of crack patterns with a physical model. *The Visual Computer* 14, 126–137 (1998)

5. O'Brien, J.F., Hodgins, J.K.: Graphical modeling and animation of brittle fracture. In: ACM SIGGRAPH 1999. Computer Graphics Proceedings, Annual Conference Series, pp. 137–146 (1999)
6. O'Brien, J.F., Bargteil, A., Hodgins, J.: Graphical modeling of ductile fracture. *ACM Trans. Graph.* 21, 291–294 (2002)
7. Iben, H.N., O'Brien, J.F.: Generating surface crack patterns. In: Symposium on Computer Animation, pp. 177–185 (2006)
8. Müller, M., McMillan, L., Dorsey, J., Jagnow, R.: Real-time simulation of deformation and fracture of stiff materials. In: Proc. Eurographics Workshop. Eurographics Association. *Compute. Anim. And Simu.* 2001, pp. 99–111 (2001)
9. Molino, N., Bao, Z., Fedkiw, R.: A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph.* 23(3), 385–392 (2004)
10. Bao, Z., Hong, J.M., Teran, J., Fedkiw, R.: Fracturing rigid materials. *IEEE Trans. Vis. Comput. Graph.* 13(2), 370–378 (2007)
11. Wicke, M., Botsch, M., Gross, M.: A finite element method on convex polyhedra. In: Computer Graphics Forum (Proc. Eurographics 2007), vol. 26, pp. 355–364 (2007)
12. Pauly, M., Keiser, R., Adams, B., Dutré, P., Gross, M.H., Guibas, L.J.: Meshless animation of fracturing solids. *ACM Trans. Graph.* 24(3), 957–964 (2005)
13. Belytschko, T., Black, T.: Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 45(5), 601–620 (1999)
14. Moes, N., Dolbow, J., Belytschko, T.: A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 46(1), 131–150 (1999)
15. Zi, G., Belytschko, T.: New crack-tip elements for xfem and applications to cohesive cracks. *International Journal for Numerical Methods in Engineering* 57, 2221–2240 (2003)
16. Budyn, E., Zi, G., Moes, N., Belytschko, T.: A method for multiple crack growth in brittle materials without remeshing. *International Journal for Numerical Methods in Engineering* 62, 1741–1770 (2004)
17. Müller, M., Gross, M.H.: Interactive virtual materials. In: Graphics Interface (2004), pp. 239–246 (2004)
18. Baraff, D., Witki, A.P.: Large steps in cloth simulation. In: ACM SIGGRAPH (1998), pp. 43–54 (1998)
19. Duarte, C.A., Oden, J.T.: An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 139, 237–262 (1996)
20. Melenk, J.M., Bubska, I.: The partition of the unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* 139, 289–314 (1996)