COLLISION RESPONSE ANALYSIS AND FRACTURE SIMULATION OF DEFORMABLE OBJECTS FOR COMPUTER GRAPHICS

A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in the Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan

By

Jie Shen

Fall 2000

0-612-63965-7

Permission To Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
Saskatoon. Saskatchewan. Canada S7N 5A9

Abstract

Computer Animation is a sub-field of computer graphics with an emphasis on the time-dependent description of events of interest. It has been used in many disciplines such as entertainment, scientific visualization, industrial design, multimedia, etc. Modeling of deformable objects in a dynamic interaction and/or fracture process has been an active research topic in the past decade. The main objective of this thesis is to provide a new effective approach to address dynamic interaction and fracture simulation.

With respect to dynamic interaction between deformable objects, this thesis proposes a new semi-explicit local collision response analysis (CRA) algorithm which is better than most previous approaches in three aspects: computational efficiency, accuracy and generality. The computational cost of the semi-explicit local CRA algorithm is guaranteed to be O(n) for each time step, which is especially desirable for the collision response analysis of complex systems. With the use of the Lagrange multiplier method, the semi-explicit local CRA algorithm avoids shortcomings associated with the penalty method and provides an accurate description of detailed local deformation during a collision process. The generic geometric constraint and the Gauss-Seidel iteration for enforcing a loading constraint such as the Coulomb friction law make the semi-explicit local CRA algorithm general enough to handle arbitrary oblique collisions. The experimental results indicate that the semi-explicit local CRA approach is capable of capturing all the key features during a collision of deformable objects and matches closely with the theoretical solution of a classic collision problem in solid mechanics.

For fracture simulation, a new element-split method is proposed, which has a sounder mechanical basis than previous approaches in computer graphics and is more flexible so as to accommodate different material fracture criteria such that different failure patterns are obtained accordingly. Quantitative simulation results show that the element-split approach is consistent with the theoretical Mohr's circle analysis and the slip-line theory in plasticity, while qualitative results indicate its visual effectiveness.

Acknowledgments

Many people have contributed to this thesis work and provided support during my years at the University of Saskatchewan. First of all, I would like to express my sincerest gratitude to my supervisor, Dr. Yee-Hong Yang, who sparked my interest in computer graphics and image processing, and provided encouragement at every key time. I would also like to thank Dr. Derek Eager and Dr. Tony Kusalik for serving on my reading committee and for their valuable comments which greatly improved this manuscript. Next, I would like to thank Dr. Glen Watson for being my cognate examiner and for providing a different point of view of the thesis work.

I would also like to thank the former members of the Computer Vision, Graphics and Image Processing group. In particular, Russ Muzzolini, Nan Weng, Tanya Grigorishin and Bernie Martin Andrew Genswein gave me all sorts of technical help in the lab.

Finally, I would like to thank my wife, Dr. Lijia Zhu, who has provided unlimited love, encouragement, and understanding in the process of completing this thesis. I dedicate this dissertation to her.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 An Overview of Computer Animation

Computer animation is a sub-field of computer graphics with a special consideration of the dimension of time such that a vast amount of information can be presented in a sequential manner. The basic idea of such presentation is to display a sequence of still frames of images at a certain fast speed to create an illusion of motion. The typical display rate is 24 frames per second (fps) for film or at least 30 fps on a raster display device.

The visual effect of animation relies on the characteristics of human visual system. The human eye consists of millions of receptors which function as sensors to sample light from the environment. Because of a certain time lag in the reaction time of these sensors and some mechanical limitations of the human visual system such as blinking and tracking, the entire human visual system could not sense changes from the environment at a rate higher than a cutoff frequency, f_c. If a sequence of still image frames are displayed at a rate greater than

f_c, then the receptors are not able to respond quickly enough to convey the content change of each frame to the brain. As a result, an illusion of continuous transition through image frames is produced. This property of the visual system is commonly known as visual retention.

Computer animation has been used in many fields such as television, video, film, gaming, scientific visualization, etc. since the 1960's. Early applications such as MSGEN [12,13] are characterized as *computer-assisted animation systems* because they primarily deal with in-between calculation of 2-D key frames involved in conventional animation. More recent systems such as Maya from Alias/Wavefront, Flame from Softimage, and 3-D Studio Max from Kinetics are categorized as full 3-D *computer-generated animation systems* in which a sequence of images are generated by computers.

Computer-generated animation techniques can be categorized into kinematic and dynamic approaches. The basic idea of the first approach is to specify the motion of objects with a choice of forward or inverse schemes. In the forward kinematic scheme, the values of parameters corresponding to the degrees of freedom in the system are specified to produce the desired movement. It is a time-consuming and trial-and-error process. In contrast, the inverse kinematic scheme is the reverse process.

The kinematic approach is not constrained by basic principles in physics, especially in mechanics. For instance, inertia and acceleration of objects are not considered in determining their motions. As a result, the resulting animation, especially with deformable objects, does not obey the basic physical laws, and therefore is usually less realistic. Such problems lead to the development of the dynamic approach which is based upon the principles of physics, especially mechanics. The dynamic approach is also called physics-based animation. It offers unsurpassed realism over the kinematic approach because the former actively reacts to applied forces (e.g. surface loads), to constraints (e.g. prescribed trajectories), to ambient media (e.g. viscous fluids), or to impenetrable obstacles (e.g. supporting rigid objects). This thesis work is an extension to existing physics-based approaches for the modeling of deformable objects.

A typical development cycle of animation includes five stages: (1) creation of a layout; (2) generation of objects; (3) rendering of objects; (4) incorporation of motion and (5) post-production. This thesis work is related only to stage (4) without addressing issues in other steps.

1.2 Motivation of this Study

Visualization of the dynamic interaction and fracture among deformable objects can be applied in both the entertainment and engineering industries. In entertainment industries, realistic computer animation of deformable objects is an appealing feature of all computer animation software packages. In engineering, the simulation of such dynamic interaction would give engineers a better understanding of the interaction process. In particular, the dynamic load during such a process is usually much larger than the static counterpart, so a dynamic failure estimation should be more accurate than the more typical usage of static stress analysis.

To develop a visualization system for deformable objects, three major parts are required in a visualization pipeline, as shown in Figure 1.1. The main focus of this thesis is on the second part of the pipeline. Within this part, two areas, collision response analysis and fracture simulation, are particularly addressed.

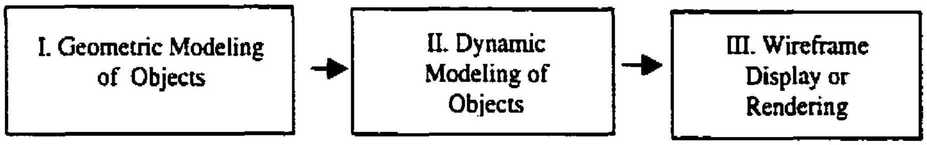

I. Geometric Modeling of Objects
II. Dynamic Modeling of Objects
III. Wireframe Display or Rendering

Figure 1.1: A visualization pipeline.

Collision response analysis is the kernel component in the dynamic modeling of deformable objects. A considerable amount of research work has been devoted to the collision response analysis of deformable objects in the past 15 years. Many of the existing approaches focus on a computationally efficient solution which is essential to applications in computer animation. However, these approaches are accompanied by some shortcomings or limitations in their accuracy and/or generality (the details will be described in Section 2.1). This thesis focuses on a novel, accurate, general-purpose, and yet efficient solution to the response analysis of arbitrary collisions between deformable objects.

Dynamic fracture simulation is an active research area in computer animation. A series of studies have been conducted so far with details described in Section 2.2. However, only tensile failure criterion is used in those studies. In this thesis work, shear failure criteria are also incorporated into a new element-split scheme such that more control flexibility can be achieved in adopting different failure criteria to generate different failure patterns.

1.3 Outline of this Thesis Work

This thesis work consists of two major parts addressing two main issues: dynamic interaction and fracture simulation of deformable objects. In the simulation of dynamic interaction, t space-time relationship among objects is classified into the following stages.

(1) *Non-collision stage*: Each object moves in space without interference from other objects.

(2) *Beginning of collision stage*: The moment at which an object just touches another one.

(3) *Collision stage*: Period during which two colliding objects interact with each other with a non-negative normal stress built up in the contact area. Two objects might stick to each other or slide relative to each other.

(4) *End of collision stage*: The moment at which an object is just about to leave the other one, i.e., the normal stress at the contact area turns to zero or negative.

An illustration of the four stages is shown in Figure 1.2. The overall strategy in handling the dynamic interaction between deformable objects is the combination of the constraint method and the global-rotation-local-deformation. The constraint method is used to enforce the correctness of the simulation of the collision process. The Lagrangian dynamics is used to describe the global rotation of each object with the merits of simplicity and time efficiency, while the finite element method (FEM) is employed to simulate the local contact between colliding objects with the advantages of flexibility and ease in handling an arbitrary contact surface and domain.

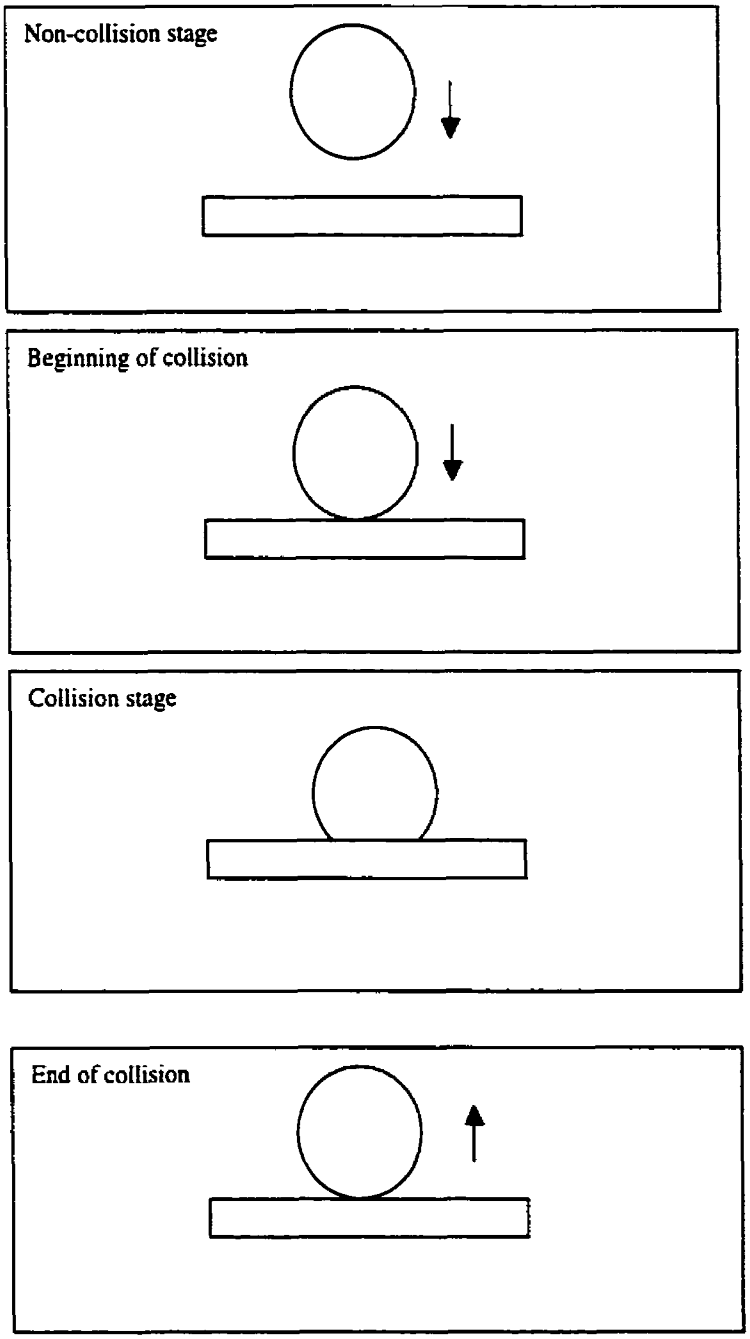


Figure 1.2: Four stages of the space-time relationship between objects.

In this thesis, a new semi-explicit local collision response analysis algorithm is proposed to achieve three goals: computational efficiency, accuracy and generality. Its computational cost at each time step is guaranteed to be O(n) in all cases leading to a solution which is faster than most previous approaches if the same number of domain nodes are used. Since the Lagrange multiplier method is used, the semi-explicit local collision response analysis algorithm provides better accuracy than the penalty method used in many previous approaches. The geometric and loading constraints are handled in a general way such that the algorithm can be used to describe any arbitrary complex collision between deformable objects.

A typical relationship between collision response analysis and collision detection is shown in Figure 1.3 in which the result of collision detection determines the use of the non-collision analysis module or the collision analysis module.

Fracture simulation is tackled by a new element-split scheme which is based upon the stress information within each element. It has the flexibility of incorporating a variety of material failure criteria including both tensile and shear types, while in previous approaches in computer graphics, only tensile failure criterion could be handled.

The evaluation strategy used in this thesis work is to compare the numerical results of the proposed algorithms with existing analytical solutions in solid mechanics. No real-world experiment is conducted and no comparison between the numerical results and the real-world experimental data is carried out due to the following considerations:

(1) The main focus of this thesis work is on the numerical efficiency and accuracy of the proposed algorithms rather than on the demonstration of the consistency between the numerical results and real-world test data.

(2) Even though the realistic representation of dynamic interaction is the final goal of any physics-based approach, numerical accuracy is one of the most important necessary conditions for such a representation. In this thesis, a series of evaluations are performed to verify the numerical accuracy of the proposed algorithms.

(3) As long as the proposed approach is algorithmically or numerically correct, a right choice of material parameters would produce a natural representation of dynamic interaction and an arbitrary choice of material parameters might generate super-natural or exaggerating deformation of objects. It should be noted that physics laws can be preserved in both cases. How to choose correct material parameters is the responsibility of application animators and is not studied in this thesis.

The effectiveness of the semi-explicit local collision response analysis algorithm proposed in this thesis is quantitatively evaluated by comparing the results of the numerical calculation with the analytical solutions of classic collision problems in solid mechanics. In this way, the numerical accuracy in predicting the collision length and in maintaining the momentum conservation, energy conservation and impulse-momentum equality can be determined such that there is no need for verifying the algorithm with the results in real experimental tests.

The element-split fracture simulation scheme is evaluated with the common shear failure pattern in triaxial tests in civil engineering. The orientation of the failure plane in numerical examples is verified by the analytical prediction which is obtained by using the Mohr's circle method. The shear failure pattern of a broken plate is compared with Nadai's experimental result which is related to the slip-line theory of plasticity [51].

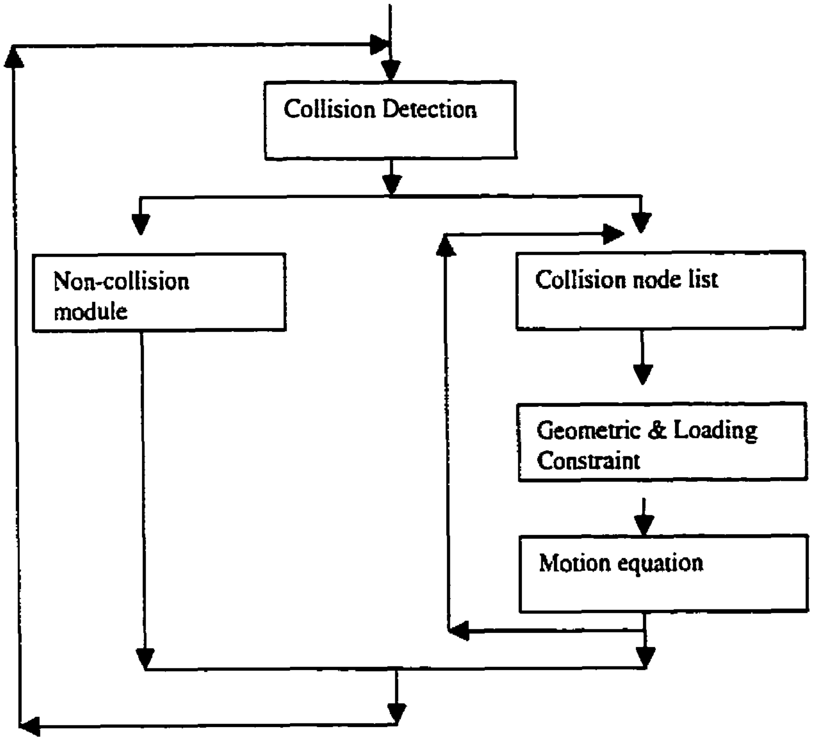


Figure 1.3: Outline of a main flowchart for analysing the dynamic interaction between objects.

1.4 Contributions

The major contributions of this thesis work include:

- This is the first time that the forward Lagrange multiplier method is applied in computer graphics. In addition, the combination of the semi-explicit time integration, Lagrangian dynamics, a fast iterative method for friction and a generic geometric constraint matrix, leads to an accurate, fast and general-purpose solution to collision response analysis of deformable objects. The undesired shortcomings associated with the penalty coefficient of the penalty method are overcome.

- The local finite element method, a new variation to the conventional finite element method is proposed. All the calculations are conducted at the element level (i.e., local finite element level) such that no expensive computation in the factorization, inversion and assembly of global matrices is needed, leading to an O(n) semi-explicit local collision response analysis (CRA) algorithm. Semi-explicit time integration with a fast iterative method for contact friction makes the semi-explicit local CRA algorithm simple and fast. A generic geometric contact constraint matrix allows the semi-explicit local CRA algorithm to be capable of handling arbitrary collision between deformable objects.
- The numerical experiments indicate that the results using the semi-explicit local CRA algorithm match with the analytical solution of a classic collision problem in solid mechanics and the algorithm is able to accurately estimate the time duration of collision leading to:
    - ♦ the correct prediction of stresses inside the object during the collision period which is crucial for the analysis of separation or fracture of different parts of the object.
    - ♦ the correct prediction of deformation inside the object caused by a given impact, especially the propagation of elastic waves inside the object due to the collision.
    - ♦ less trial-and-error for animators in determining the appropriate incremental time interval for the collision period.
- The element-split scheme for fracture simulation is advanced to handle the cases of arbitrary orientation of a splitting plane within each element with the flexibility to incorporate different material failure criteria into this scheme because it is based on

element stress rather than nodal force as in previous approaches. Different failure patterns can be obtained by using different material failure criteria. This gives the element-split approach more control potential than existing ones in computer graphics.

1.5 Organization

The organization of this thesis is as follows. Chapter 2 introduces an overview of previous studies in the area of modeling deformable objects as well as the objectives of this thesis work. Chapter 3 presents a brief introduction to one of the most powerful numerical analysis tools, the finite element method, and a comparison with other methods. Chapter 4 describes some special treatments to the collision detection of arbitrarily-shaped objects covered by triangular surface patches on the basis of existing approaches. Chapter 5 proposes a new semi-explicit local collision response algorithm for modeling dynamic interaction of deformable objects. The results of numerical experiments are also given. Chapter 6 presents a new element-split scheme for fracture simulation, failure criteria of material, and experimental demonstrations. Chapter 7 summarizes the findings of this thesis work and points out possible future work. In appendix A through G, the detailed algorithms, nomenclature and glossary are given.

Chapter 2

Previous Work, Problems, and Objectives

In this chapter, previous research studies in two main issues in physics-based approaches: collision response analysis and fracture simulation, are reviewed and the existing problems are identified. The objectives of this thesis study are also stated.

2.1 Collision Response Analysis

Collision response analysis is one of the most crucial parts in computer animation of deformable objects. In this section, the existing approaches in collision response analysis are compared with the new approach developed in this thesis in terms of three important factors: computational efficiency, accuracy and generality. Such comparison differentiates the new approach with existing ones.

In the past 15 years, researchers have proposed many solutions to this task. However, there is still plenty room for better approaches in terms of computational efficiency, accuracy and generality. The main purpose of this work is to propose a new semi-explicit local collision response analysis algorithm, which is accurate in terms of description of collision response, general-purpose in terms of no extra assumptions on geometric constraint formula, nodal connection at contact surface, deformation zones, etc., and yet fast in terms of guaranteed O(n) time cost.

2.1.1 Computational Efficiency

In this thesis, a new semi-explicit local collision response analysis algorithm is proposed. Its computational cost is guaranteed to be O(n) time in all cases, where n is the total number of domain nodes. The details of the algorithms will be explained in Section 5.4. In the conventional finite element approach [48,69-71], since a narrow non-zero diagonal band in the motion equations is not guaranteed to be formed when a system becomes complex, the profile solver likely takes more than O(n) to solve the system motion equations. Even if a sparse matrix solver is used, extra procedures are needed to take care of zero entries within the diagonal band [59]. The total cost still likely ends up higher than O(n). Similarly, in the deformable models developed by Baraff and Desbrun [4,26] the computational cost can not be guaranteed as O(n), especially when the total number of domain nodes becomes large.

Even though the boundary element approach [39] claims that its time cost without including the pre-computation can be reduced to O(ms), where m is the total number of surface nodes and s is the maximum of the number of boundary condition changes and the number of boundary value changes, it should be kept in mind that this conclusion is based upon an assumption that the boundary values and conditions change only at very few surface nodes. This assumption may be acceptable in static or quasi-static cases but absolutely not in dynamic cases. In dynamic cases such as a collision process, the inertia forces of all surface nodes (a part of boundary value) usually change simultaneously. This forces the approach to take at least O(m^2) time. It is difficult to say that O(m^2) must be higher than O(n) consumed by the semi-explicit local collision response analysis approach where n is the total number of domain nodes. However, consideration of the limitations of the boundary element approach as explained in Section 2.1.3 eliminates it as a candidate for a general-purpose solution. So far the approach reported in [39] handles only the simplest static analysis like the volumetric deformable model in [24], which is not the topic focused in this research work.

It should be stated that the global deformation approaches [29.56,77] are normally faster than the semi-explicit local collision response analysis approach because in the former only very few domain nodes are considered. However, what is sacrificed in the global deformation approaches is the realistic representation of collision response which is one focus of this research work as explained in Section 2.1.2.

2.1.2 Accuracy

Penalty vs Lagrange Multiplier Method

An essential component of collision response analysis is to impose constraints at the interface between two or more colliding objects. Geometric constraints are imposed on the nodal displacement variables, while force constraints are imposed on the nodal force variables. The constraints can be a user-specified value or a certain condition between solution variables.

The Lagrange multiplier method and the penalty method are two widely used procedures to impose the constraints [11].

Let's consider the following variational formulation of a discrete system model for steady-state analysis:

$$\Pi = \frac{1}{2}\mathbf{U}^T\mathbf{K}\mathbf{U} - \mathbf{U}^T\mathbf{R}\,, \tag{2.1}$$

with the condition

$$\frac{\partial \Pi}{\partial \mathbf{U}_i} = 0 \qquad \text{for all } i(=1,\ldots,n)\,. \tag{2.2}$$

where **U**, **K** and **R** are the displacement vector, stiffness matrix (order $n \times n$) and load vector, respectively. Π is the total potential energy of the system which is the sum of the strain energy and the potential energy of the applied loads. Equation (2.2) represents the principle of minimum total potential energy [60].

If m linearly independent discrete constraints $\mathbf{BU} = \mathbf{V}$ are imposed onto the solution using the Lagrange multiplier method we have [8]

$$\Pi^{*}(\mathbf{U},\lambda) = \frac{1}{2}\mathbf{U}^T\mathbf{K}\mathbf{U} - \mathbf{U}^T\mathbf{R} + \lambda^T(\mathbf{BU} - \mathbf{V})\,, \tag{2.3}$$

where λ is a vector of m additional variables, the Lagrange multipliers. **B** is a matrix of dimension $m \times n$. By invoking $\delta\Pi^{*} = 0$ and considering that $\delta\mathbf{U}$ and $\delta\boldsymbol{\lambda}$ are arbitrary, m extra equations are introduced into the original system of equations as follows

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^{T} \\ \mathbf{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{V} \end{bmatrix}. \qquad (2.4)$$

In the penalty method, an additional constant, the penalty coefficient (α), of relatively large magnitude is added to the system as follows

$$\Pi^{*}(\mathbf{U}) = \frac{1}{2}\mathbf{U}^{T}\mathbf{K}\mathbf{U} - \mathbf{U}^{T}\mathbf{R} + \frac{\alpha}{2}(\mathbf{B}\mathbf{U} - \mathbf{V})^{T}(\mathbf{B}\mathbf{U} - \mathbf{V}). \qquad (2.5)$$

By invoking $\delta\Pi^{*} = 0$ and considering that $\delta\mathbf{U}$ is arbitrary, the following equation results

$$(\mathbf{K} + \alpha\mathbf{B}^{T}\mathbf{B})\mathbf{U} = \mathbf{R} + \alpha\mathbf{B}^{T}\mathbf{V}. \qquad (2.6)$$

Comparison between equations (2.4) and (2.6) indicates that the advantage of the penalty method is that it is easy to implement because no extra equation is introduced. However, some drawbacks of the penalty method for collision response analysis include the following:

(1) The penalty term leads to an inaccurate solution [75].

(2) It increases the stiffness of the equation of motion resulting in an undesirable requirement for a smaller time step in integration [75].

(3) The penalty coefficient may need to be adjusted manually on a trial-and-error basis and is problem dependent.

(4) Some penalty methods use the penetration depth or the like as input to determine the repulsive force. Since the penetration depth is dependent on the size of time step, different size of time steps may result in different collision responses. In other words, they are sensitive to the size of time step.

On the contrary, the Lagrange multiplier method can be used to avoid most of the above shortcomings with the price of extra equations to be solved. Since the number of contact nodes in a system is normally much smaller than the total number of nodes in the system, such extra computation or memory cost is tolerable. Hence, in this thesis, the Lagrange multiplier method is adopted.

In the past 15 years, some researchers use the penalty method for the sake of simplicity, while others adopt it simply because the collision response analysis is not the major issue in their studies. Terzopoulos et al. [71] add a potential energy [76], $c_1 e^{\frac{f(\mathbf{r})}{\varepsilon}}$, around each object, where f is an inside/outside function of an object and $\mathbf{r}$ is the position vector of a particle. c_1 and ε are constants which determine the shape of the potential, and are chosen to prevent the penetration between objects. The resulting collision force is determined by the gradient of the potential. One potential problem with this approach is that it is computationally expensive to construct the inside/outside function for complex arbitrarily-shaped objects. As well, the lack of relationship between these two constants and material parameters of objects causes difficulty in using material parameters directly as an input for the collision response analysis. Terzopoulos and Witkin [72] propose a hybrid formulation of rigid dynamics and nonrigid linear elasticity. However, their motion equation still does not contain any collision constraint components as independent variables to be solved, resulting in an inaccurate solution in the collision response analysis. Some other similar approaches of using the penalty method include [4,29], even though their main focus may not be on the accurate collision response analysis.

Several researchers have conducted research related to the Lagrange multiplier method in computer graphics. Baraff [3] uses the Lagrange multiplier method to handle the interconnection of different parts of articulated figures. Metaxas and Terzopoulos [48,49] use the Lagrange multiplier method to describe point-to-point interconnections [7] between deformable parts in a self-assembly modeling process. The use of point-to-point constraints poses a strict requirement for the alignment of surface nodes of deformable parts to be assembled. Even though the equation of motion with Lagrange multiplier is used in collision response analysis, the Lagrange multipliers are used only for computing the interconnecting forces between deformable parts, while the collision forces between the assembled deformable complex and the environment are determined simply by the reaction constraints [57]. The Lagrange multipliers for point-to-point interconnections are essentially internal contact forces between deformable sub-parts which are not separable after the self-assembly process. In the sense of collision response, their approach uses the reaction constraint. Even though it is simple and requires no extra differential equations, the reaction constraint approach has the following shortcomings:

(1) Only one reaction constraint is allowed at each node at any time. This limits the scope of the approach.

(2) It is difficult to accurately determine the frictional force between two colliding deformable objects which are separable.

Platt et al. [57] give an excellent summary on different types of constraints. They use augmented Lagrangian constraints for constructing deformable objects which are incompressible and moldable. Witkin and Welch [77] use the Lagrange multiplier method to formulate constrained dynamics of non-rigid structures, similar to that of [58]. Their formulation focuses on the case when the constraint is known in advance, e.g. in motion control or in path control. In other words, their approach is not designed for handling cases where the constraint is not known before a simulation, i.e., the cases of arbitrary collision. Baraff and Witkin [5] combine Witkin's constraint dynamics [77] and Baraff's analytic contact force model for rigid bodies [1,2] to form a scheme for the dynamic simulation of non-penetrating flexible bodies. They propose a two-phase model of collision response of flexible bodies, which requires the determination of impulse at collision points in phase 1 and the enforcement of the acceleration constraint of the gap function between two colliding objects in phase 2. The possible limitation of this two-phase model is that the acceleration inequality constraint usually requires complex quadratic programming [1,2].

Global vs Local Deformation

Global deformation means that the deformation of an object is approximated by a linear combination of a set of basis shape or mode vectors. Its original idea comes from the concept of super element and mode superposition in engineering analysis [8,34] and similar ideas have been used in computer graphics since 1989 [56]. The advantage of the global deformation approach is the reduction of computational cost, while the disadvantage is the loss of local deformation details.

For the first time in computer graphics, Pentland and Williams [56] achieve the goal of separate representation of dynamic behavior and geometric form by using the global deformation technique [6]. A system is built using polynomial deformation mappings to couple a vibration-mode ("modal") representation of object dynamics together with volumetric models of object geometry. The collision response is treated from a sort of artistic point of view by using different vibration modes. The calculation of the true contact surface in a non-zero-length collision stage for deformable objects is not performed. Consequently, deformation of objects looks interesting, but not quite physically realistic. For instance, the deformation of the ball in Figure 2 in reference [56] should be asymmetrical when it collides with the beam, because the part of the ball which is in contact with the beam should deform the most.

Witkin and Welch [77], Baraff and Witkin [5] use the concept of global deformation to describe flexible objects. It is a compromise between the extremes of the nodal and rigid formulations. The changes in the shape of objects are approximated by the global deformation which is simply a parametric "space warp" of all the discretized nodes of the objects. Since fewer nodes are used in calculating the global deformation as compared to the nodal approach, this scheme is fast. In addition, the stiffness problem due to local interactions is eliminated because the shape parameters are global in their effect.

Faloutsos et al. [29] extend Witkin and Welch's approach to accommodate a hierarchy of deformation of objects in a nonlinear fashion with respect to the state parameters. The approach has a similar shortcoming as Pentland and Williams' approach.

Using the local deformation approach means that a reasonable number of domain nodes are used to partition each object domain into a finite number of sub-domains such that the local detailed deformation can be sufficiently described. For some cases like dynamic fracture simulation or realistic representation of asymmetric deformation due to oblique complex collision, the local deformation approach probably is one of most suitable candidates. Since generality is one of the three goals to be achieved in this work, the local deformation approach is adopted in this thesis. To address the requirement of computational efficiency, one of the most efficient solutions for the local deformation approach is described in Section 5.4.

2.1.3 Generality

Boundary Element vs Finite Element Method

The collision response analysis could be performed by using the finite difference method, the boundary element method [9] or the finite element method [8]. Since the finite difference method usually requires special modifications to define irregular boundaries and complex boundary conditions, it is not used to solve motion equations in this research work. Even though the computational mesh for the BEM is simpler than that for the finite element method, the boundary element method has the following disadvantages:

(1) The difficulties associated with singular integrals in the boundary element method are significant and are often highly underestimated [9].

(2) It is difficult to use the boundary element method to handle heterogeneous domains [9].

(3) It is difficult, if not impossible, to use the boundary element method to conduct an effective dynamic fracture simulation as shown in Figure 2.1.

Since the main focus of this research work is to propose an accurate, fast and general-purpose solution to the collision response analysis, the boundary element method is eliminated because of its limitations. One major disadvantage of the conventional finite element method is its higher computational cost, compared to the boundary element method. This shortcoming is overcome by introducing a new variation of the conventional finite element method in this thesis such that no factorization, inversion and assembly of global matrices are required, and in the meantime all advantages of the finite element method compared to the boundary element method, are retained.

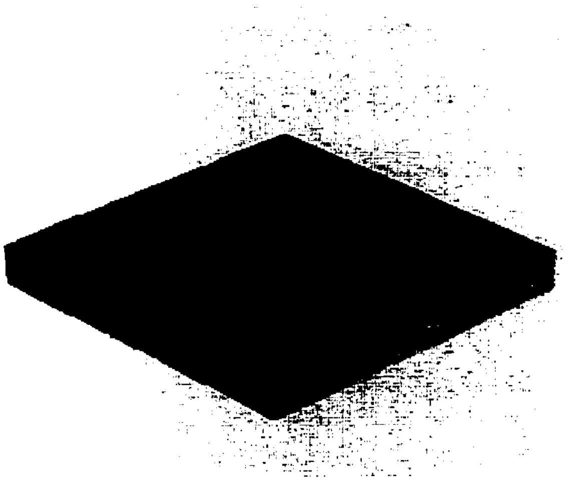


Figure 2.1: Dynamic fracture simulation of a plate caused by a falling cube with the semi-explicit local collision response analysis approach used in this thesis.

Partitioned Local Deformation vs Arbitrary Local Deformation

Partitioned local deformation means that the deformation field of a deformable object is partitioned or simplified into several different types of deformation zones each of which is handled differently, while the arbitrary local deformation does not impose such simplification.

Cani-Gascuel and Desbrun [15] propose a unique approach to handle collision response between deformable objects by a combination of global and local deformation. An implicit isopotential surface is statically used to coat each base structure, an internal physically based model. The implicit layer performs collision detection and generates the local deformations due to contact, while the base structure controls the global scale behavior. The implicit surface is generated by a set of skeletons s_i (i =1, ..., n) with associated field functions f_i as follows:

$$\begin{aligned} &\{P \in \Re^3 \mid f(P) = c_1\}, \\ &f(P) = \sum_{i=1}^{n} f_i(P), \end{aligned} \tag{2.7}$$

where c_1 denotes an isovalue. f is called the "field function" which is the summation of f_i, the implicit contribution of the i-th skeleton. Skeletons can be any geometric primitives, points, curves, parametric surfaces, simple volumes, etc., which admit a well defined distance function. Contact between objects is dealt with in two steps: (1) A negative field g modeling compression is added in the interpenetration region, as shown in Figure 2.2, in order to generate a contact surface with the other object; (2) A positive field p modeling the transverse propagation of deformations is added in the propagation region. These two treatments lead to the vanishing of interpenetration region in the left subfigure of Figure 2.2, as illustrated in the right subfigure of Figure 2.2.

This approach is able to generate a relatively accurate contact surface without increasing computation cost. However, one possible problem is that the interpenetration region in Figure 2.2 is not necessarily the area with the greatest deformation during a collision. For instance, consider the situation where the right sub-part of object 1 is much stiffer than its left sub-part. In such a case, probably no deformation is generated by the model because the right sub-part is stiff and the left sub-part is not in the interpenetration and propagation regions. Another problem is that no experimental evaluation has been conducted with respect to the length of the collision period which is the most important factor to determine the deformation with a specific stiffness for the colliding objects. Furthermore, how to divide into these three regions remains quite arbitrary. The approach proposed in this thesis focuses on the general treatment of arbitrary local deformation.

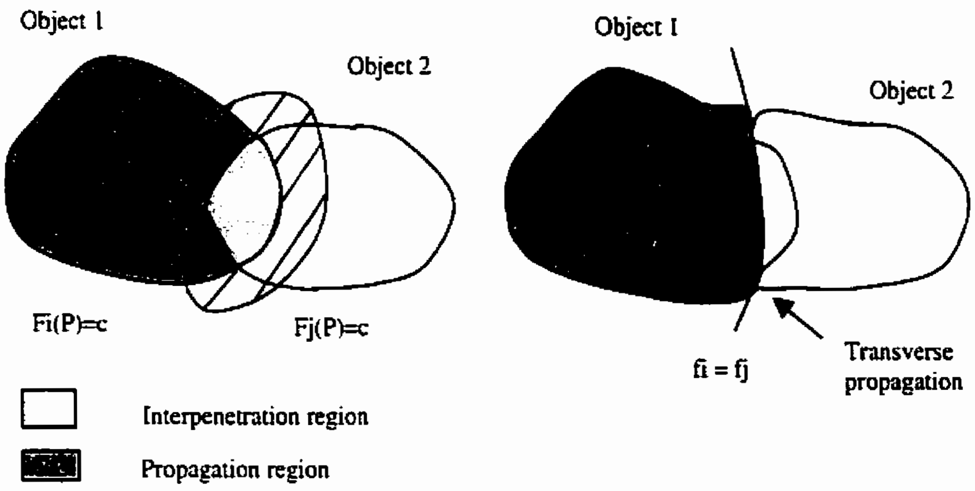


Figure 2.2: Modeling contact consists of different deformation fields in the interpenetration region and in the propagation regions [15].

2.1.4 Comments on Constraint Methods

Common to all constraint methods is that the interaction between objects is considered as a kind of constraint to the entire system. To impose the constraint, options include: (1) to insert a special type of interface element at the collision point; (2) to mathematically calculate the interface force at the contact area and consider such interface force as external loading to the colliding objects. The first option represents some old approaches which are associated with the penalty method, while the second option corresponds to the Lagrange multiplier approaches in Section 2.1.2. For the sake of completeness of the literature review, the variations of the first option are introduced below.

♦ **Spring**

Inserting a spring dynamically (i.e., during a simulation) at the collision point is the most intuitive way to handle collisions [50]. The precondition for using a spring is that the simulation system should allow for applying the spring forces as external forces to the colliding objects. Whenever a collision occurs, a spring is temporarily inserted at the collision point. The stiffness of the spring should be large enough to withstand the impact between the two objects. After the collision, the spring is eliminated from the system. This constraint is easy to understand and implement.

One major problem with this approach is that it is conceptually supposed to pass only the force along the axis of the spring without considering the forces perpendicular to it. When sliding exists at the interface between the two objects, the skewed spring hardly functions properly, as shown in Figure 2.3. Another problem is the connection between the spring and the surface node on the surface of each colliding object. Most often, the surface nodes of the two colliding objects are not contiguous to each other, as shown in Figure 2.4(a). That is, the surface nodes of the two objects are not lined up horizontally and the number of surface nodes of both objects is not the same. As a result, it is difficult to use rectangular elements to connect the surface nodes of the two objects directly. One possible solution is that a transition layer is developed which consists of the projections of the nodes of surfaces of two colliding objects *A* and *B*, as shown in Figure 2.4(b) [67].

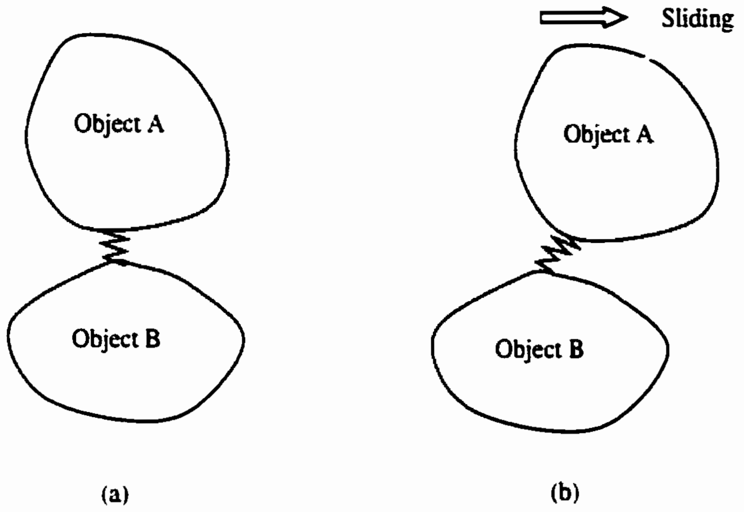


Figure 2.3: A spring connecting two objects. (a) before sliding; (b) after sliding.

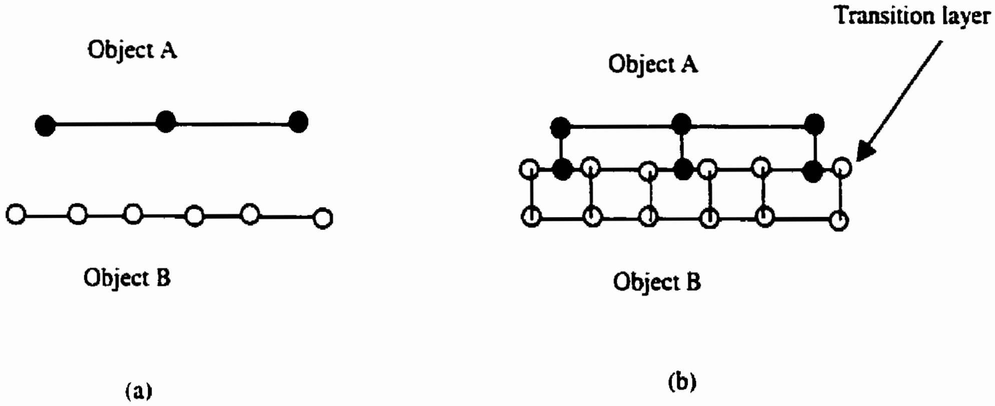


Figure 2.4: Surface non-contiguous nodes and a transition layer proposed by Simo et al. [67].

One drawback associated with this technique is the extra layer of nodes which is difficult to locate on curved surfaces. Another shortcoming is the high computational cost when the stiffness of the spring has to be given a large magnitude [1], because a large value of stiffness requires a small time step for accurate numerical integration.

- **Pinball**

Pinballs are statically used (i.e., pre-allocated before a simulation) to form an interface layer for each object in the system, as shown in Figure 2.5 [10]. Each pinball is embedded in one finite element at the surface of the object. Then, the collision detection is simplified to check only the penetration between surface pinballs of different objects. The penetration depth is calculated by using the coordinates of the center of each pinball. On the basis of that depth, the reaction force can be calculated in a similar way as in calculating the spring force.

One problem with the pinball approach is that the time step before collision can not be large. Otherwise, the two colliding pinballs could entirely pass over each other in a single time step, resulting in no collision between these two pinballs. Another problem is that handling the friction between pinballs is still not well solved. Since the surface of the object is simulated by pinballs, the resulting surface is not even such that when one pinball roll over another row of pinballs, the normal at the contact point varies or oscillates with time. But this should not exist.

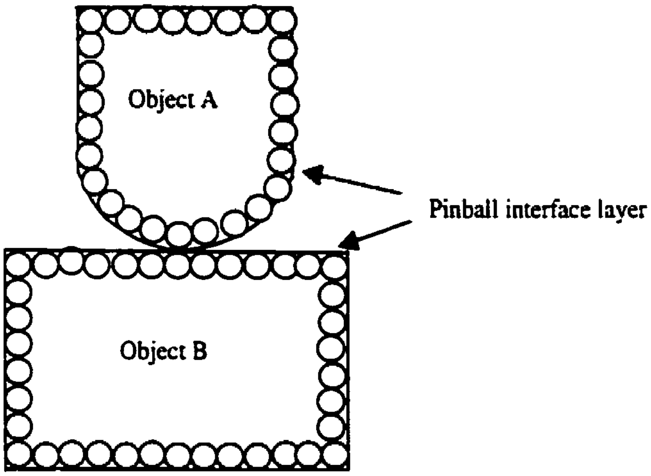


Figure 2.5: A pinball interface layer shown in two dimensions.

- **Point-to-point and point-to-surface elements**

The interface or contact surface formed by adjacent objects can be generally classified as point-to-point, point-to-surface and surface-to-surface [63]. In two dimensions, the second and third cases degenerate into point-to-edge and edge-to-edge, as shown in Figures 2.6 and 2.7. A framework for contact interface between objects is proposed using two types of elements, point-to-point (p-to-p) and point-to-surface (p-to-s). The p-to-p element is a two-node rod-like element. Its length is usually set to be relatively smaller than the size of the object with which it is associated. Like the pinball approach [10], the p-to-p element is statically allocated to the surface of each object prior to numerical analysis or simulation, resulting in efficient collision detection (Figure 2.6(a)). The p-to-s element is used as a bridge connecting a p-to-p element with another object to which the p-to-p element is penetrating, as shown in Figure 2.6(b). It is assumed that the surface of the penetrated object can be approximately discretized into triangular divisions. Under this assumption, the p-to-s element is defined as a tetrahedron with four nodes, three of which are connected to the penetrated object and the remaining one is linked to a p-to-p element, which penetrates the object. Like the spring, the p-to-s is dynamically allocated and deallocated at the interface between objects during the analysis.

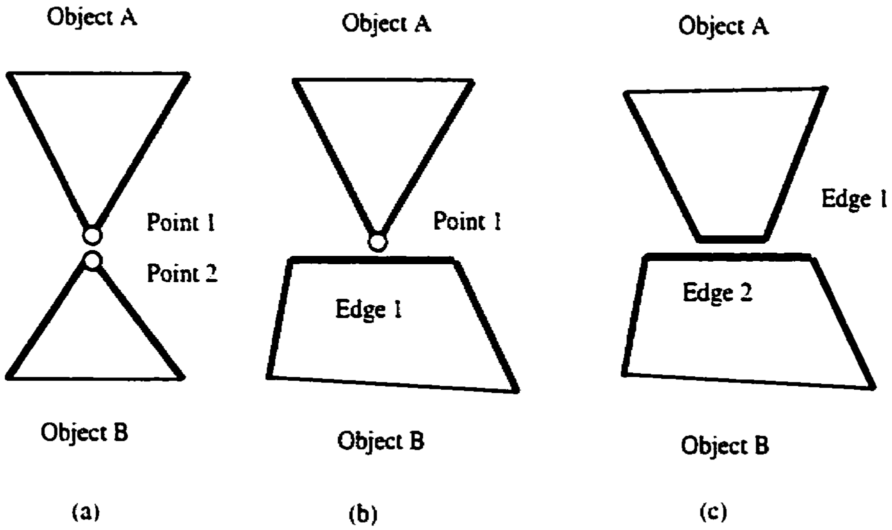


Figure 2.6: Three types of contact interfaces between objects in two dimensions. (a) point-to-point: (b) point-to-surface: (c) surface-to-surface.

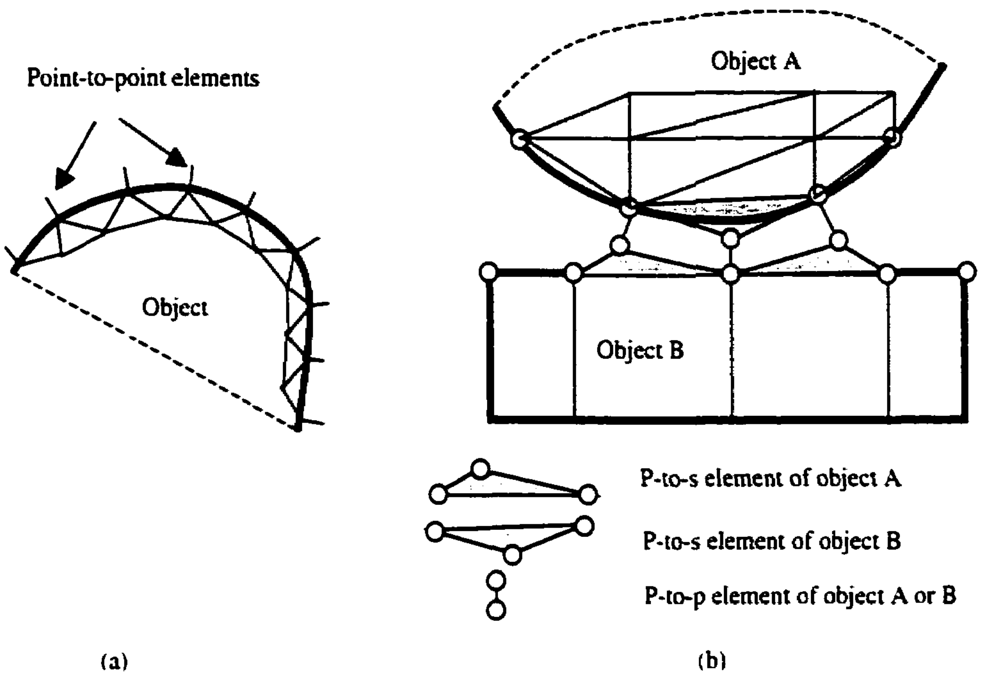

Point-to-point elements
Object
Object A
Object B
P-to-s element of object A
P-to-s element of object B
P-to-p element of object A or B
(a)
(b)

Figure 2.7: A framework for interface contact using (a) point-to-point element and (b) point-to-surface element.

The combined usage of p-to-p and p-to-s elements can avoid the shortcoming associated with the spring approach (Fig. 2.3) and the requirement for small time step with the pinball approach. However. this approach can not elegantly handle cases where there is sliding at the interface. Regeneration of p-to-p and p-to-s elements at the interface after each time step could cope with the friction cases. but the computation cost would be too high to be of practical use.

- **Problems with interface components**

Overall, the problem with this type of interface component is that it does not deal with friction very well. In addition, many investigators indicated the problem of high stiffness of differential equations, if the penalty method is used [1,57,75]. The high stiffness forces users to use a small time step which adds a big computation cost.

Witkin [75] indicates that the penalty method related to spring is not a good communication way or linkage between the penetration depth and the reaction forces. He advocates the direct calculation of reaction force.

2.2 Fracture Simulation

Fracture is a common phenomenon in the natural world. Dynamic fracture, i.e. the fracture in a dynamic collision process, is difficult to describe in a realistic way. If we further consider the dynamic fracture of deformable objects, the task becomes more challenging because of the coupling of the large dynamic deformation and fracture, which requires a good collision detection scheme, a good collision response analyzer and a good fracture simulator.

In computer graphics, several studies have been conducted in the past with respect to fracture (static or dynamic). Terzopoulos and Fleischer [69,70] compare the distance between two adjacent nodes with respect to a given threshold. If the distance exceeds the threshold, the occurrence of the failure is assumed. Norton et al. [53] use a similar strategy to study the dynamic fracture of 3D solid objects. Mazarak et al. [47] apply this type of failure criteria in modeling the fracture of rigid objects in an explosion process. Smith et al.

[68] use this type of failure criteria in a mass-node system. In essence, the approaches used by all of the above include a brute force tension failure criterion which has the following drawbacks: (1) It can only handle the tensional failure but not shear failure; (2) It is unable to gracefully handle the failure in a general orientation rather than in the direction connecting two adjacent nodes.

O'Brien and Hodgins [54] advance a step from previous studies by using a node-split scheme in which the forces at a given node are compared with the threshold to determine if the node needs to be split in two. One major shortcoming of this approach is that it limits the potential possibility of using other more complex failure criteria which are based upon the element stresses. Only the basic tensile failure criterion is used.

Some other related studies include the static fracture generation by Hirota et al. [37] and in-plane fracture by Neff and Fiume [52]. Static crack patterns are created by using a mass and spring system. Only tensile failure criterion is eligible to be used in such a type of system. The propagation algorithm proposed by Neff and Fiume is developed for in-plane 2-1/2 D cases and involves some heuristic treatments which are beyond the principle of a physics-based approach.

This thesis also addresses the issue of realistic computer animation of dynamic fracture of deformable objects with the following features:

(1) The element-split scheme which is originally proposed by Shen and Yang [65] for hexahedron elements will be extended to tetrahedron elements to handle fractures in arbitrary orientations.

(2) Beside the simplest tension failure criteria, shear type criteria such as Tresca's criterion (no friction) and Coulomb-Mohr's criterion (with friction) are investigated to identify their influence on the fracture pattern and their potential application in computer animation.

Chapter 3

A Numerical Tool: Finite Element Method

A brief introduction to the history of the finite element method is given in this chapter and the reasons to choose it as a numerical tool in this study are also provided.

3.1 Introduction

The finite element method (FEM) is a variational procedure in which the approximating functions such as algebraic polynomials are used to approximate the solution variables in simple subdomains (called finite elements) into which a given domain is divided. Modern computer architecture and computing techniques promote the development and application of such method.

The basic idea of the FEM originated from matrix analysis of airplane structures in aerospace in the 1950s [46]. According to the structure matrix analysis method, an entire structure can be approximated by an assembly of a finite number of mechanical elements in finite sizes such that the function of each element is analogous to the role of a brick in a building.

The concept of the structure matrix method was extended to the solution of plane stress problems in elastic mechanics in 1960 and the terminology of "finite element method" was adopted [20]. Any continuous medium has to be discretized in the following way in order to be analyzed by the FEM:

- The continuum is divided into a finite number of blocks (or elements), each of which is linked to its adjacent elements only at certain specified points (or nodes).
- Within each element, the displacement is approximated by a simple function such as a linear or higher-order polynomial. The relationship between nodal force and nodal displacement is determined by the variational principle. Here, the nodal force and nodal displacement represent the force and displacement at each node of an element.
- Assembling the nodal force-displacement relation of all elements leads to a set of linear algebraic equations in which nodal displacements are unknowns. Solving such a set of equations provides the displacement information at a finite number of nodes within the continuum, i.e., an approximate solution to the problem.

Nowadays, the FEM has already obtained a dominant position in solid mechanics and is in a position of competing with the finite difference method in fluid dynamics.

3.2 Comparison between Finite Element Method and Other Methods

In general, at least four approaches exist in describing the deformation of objects:

(1) Classic analytical methods [19]

In the classic analytical methods, a continuum is assumed to contain an infinite number of micro-blocks with infinitesimal size. By letting the size of the micro-block tend to zero, a set of differential equations describing the mechanical behavior of the continuum are obtained. Solving such a set of equations leads to an analytical solution of unknowns at any point within the continuum. However, when non-linear and non-uniform material properties or irregular geometry are encountered, an analytical solution is difficult, if not impossible, to find.

(2) Particle system and node-mass system[31,61,62,66]

Particle systems were developed to address the cases in which the continuum mechanics and the finite element method are not suitable. One example is modeling a system which contains a significant number of complex elements and/or events in a randomized manner. The basic idea of particle system is to model these elements and/or events directly rather than blending them into some kind of continuum.

In a particle system, the material is represented as a large collection of microscopic particles which interact with each other by obeying simple physical laws at the microscopic level. The disadvantage of particle system is that normally hundreds of thousands of particles are involved in a system such that its computation cost is huge. Since the focus of this study is on structured deformable objects, the particle system does not offer an advantage over the following three approaches.

Node-mass system is a simple way to model the structured deformable objects [47,53,76]. However, one potential shortcoming is that when it is used in fracture simulation, arbitrary fracture orientation is not possible except the direction in which two adjacent nodes are connected.

(3) Finite difference method [30]

The basic idea of the finite difference method is to replace continuous derivatives in differential equations with the ratio of changes in the variables over small, but finite increments. For example, a first-order derivative $\partial u / \partial x$ is substituted by $\delta u / \delta x$. With such type of substitutions, a differential equation can be transformed to a finite difference equation.

Taylor series or interpolation polynomials can be used to approximate various derivatives. Forward, backward, and central-difference approximations to the first derivative are defined by the following three equations, respectively:

$$\frac{\partial u}{\partial x} = \frac{\delta u}{\delta x} = \frac{u_{i+1,j} - u_{i,j}}{\delta x}, \tag{3.1}$$

$$\frac{\partial u}{\partial x} = \frac{\delta u}{\delta x} = \frac{u_{i,j} - u_{i-1,j}}{\delta x} \quad and \tag{3.2}$$

$$\frac{\partial u}{\partial x} \approx \frac{\delta u}{\delta x} = \frac{u_{i+1,j} - u_{i-1,j}}{2\delta x}, \tag{3.3}$$

where u refers to a function of x and y; i and j are the indexes of a node in a finite difference mesh, as illustrated in Figure 3.1.

One advantage of the finite difference method is its simplicity of concept and implementation. For instance, the combination of iterative solution technique and the finite difference formulation can achieve a fast solution of thousands or more nodes in 2-d cases. One big disadvantage of the finite difference method is that special procedures are required to handle irregular boundaries which introduce uneven meshes. With the finite element method, such special procedures are not needed.

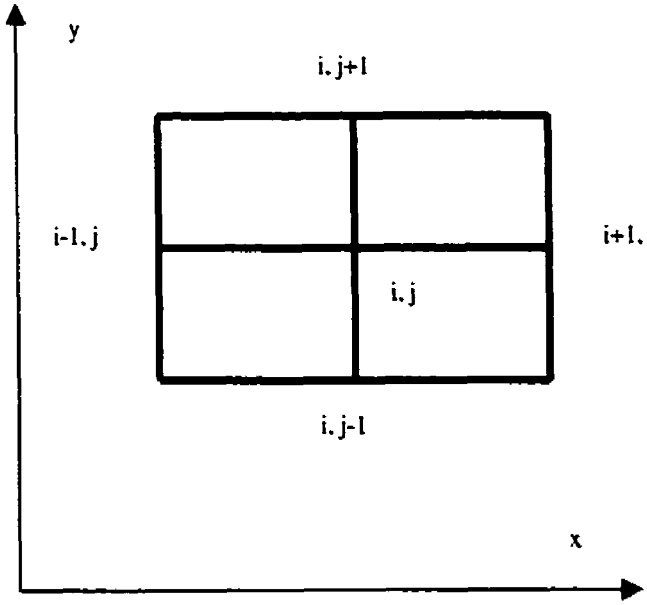


Figure 3.1: A finite difference mesh.

(4) Boundary element method [9]

The boundary element method (BEM) transforms the differential operator defined in the domain to integral operators defined on the boundary. In the BEM, only the boundary is discretized for the Laplace equation. Even though the computational mesh for the BEM is simpler than that for the finite element method (FEM), the BEM has the following disadvantages:

- The difficulties associated with singular integrals in the BEM are significant and are often highly underestimated.
- It is difficult to use the BEM to handle heterogeneous domains.
- It is difficult, if not impossible, to use the BEM to conduct an effective dynamic fracture simulation.

Besides, with the BEM long narrow objects also has to be broken into assemblies of boundary elements because of numerical instabilities.

Since one main focus of this research work is to propose an accurate, fast and general-purpose solution to the collision response analysis, the BEM is eliminated because of its limitations.

(5) Finite element method [46]

In the FEM, each element can be assigned different material properties to simulate the material non-uniformity; iterative and incremental methods may be used to solve non-linear problems; arbitrary mesh generation eliminates the difficulty in modeling irregular geometric shapes. The main advantages of the FEM over the first four methods include:

- It is applicable to solving all continuum problems such as stress analysis of non-uniform materials, anisotropic materials, non-linear stress-strain relations, and complex boundary conditions, heat transfer, fluid dynamics and electro-magnetism, to name a few.
- With the availability of different types of elements, it is expected that FEM provides a higher accuracy of solution than the finite difference method.
- With a piecewise polynomial approximation of weak forms of boundary- or initial-value problems over a domain partition of the solution to the system, the FEM is easily used in handling irregular domain and in selecting coordinate functions which are independent of the geometry of the domain.

With the consideration of the above facts, the FEM is chosen as a numerical tool to facilitate the modeling of deformable objects. One major disadvantage of the conventional FEM is its higher computational cost, compared to the BEM. This shortcoming is overcome by introducing a new variation of the FEM in this thesis work such that no factorization and assembly of global matrices are required, and in the meantime all advantages over the BEM are kept intact.

The FEM can be implemented in three different ways: displacement-based, equilibrium-based, and hybrid approach. With the displacement-based method, the displacements within elements are considered as unknowns, while in the equilibrium-based approach stresses are assumed to be unknown. In the hybrid method, both stresses and displacement are viewed as unknowns. In this thesis work, the displacement-based approach is used, which is actually the most widely-used approach due to its simplicity and good numerical properties.

Chapter 4

Collision Detection

A brief summary of existing collision detection methods is given in this chapter. Since collision detection is not the main focus in this thesis work, existing methods are adopted for the new semi-explicit local collision response analysis algorithm described in the next chapter.

4.1 Existing Methods

Collision detection is to detect where and when two objects contact each other. It has been extensively investigated in the past [21]. The simplest approach is to use the bounding volume and spatial decomposition techniques. When two objects are far away, the bounding volume method works very well [14,16,21,38]. The spatial decomposition technique is used to deal with the problem when the objects are close to each other. The computational cost of this approach is high, because of the recursive subdivisions involved.

Baraff [2] uses the principle of geometric coherence to devise algorithms on the basis of local features. This improves the performance of collision detection in a dynamic environment. Cohen et al. [21] further generalize the idea of coherence leading to the I-COLLIDE system which can simulate hundreds of objects undergoing rigid motion.

Herzen et al. [35] develop a general algorithm for dealing with curved objects with time dependent parametric surfaces. Since subdivision is used, the computation cost is high. A similar method is presented by Duff [28].

Pentland and William [55] use implicit functions to represent shape and the 'inside-outside' functions for collision detection. However, it is found not to be robust by Duff [28].

Lin and Manocha [44] develop algorithms for curved objects with spline surfaces and algebraic surfaces in rigid motion. Baraff and Witkin [5] use polygonal approximation of the objects and the resulting polygons are checked for collision.

Some other algorithms have been developed to address the collision and self-collision detection in the modeling of cloth and hair [17,42,43,78]. In this thesis, the simple detection method proposed by Moore and Wilhelms [50] is extended to detect the collision between deformable objects with triangular surface patches.

To facilitate the collision detection of arbitrary dynamic interaction among deformable objects with discretized surface triangular patches, the following special treatments are used in this work. They are not complete in terms of collision detection and should be used in conjunction with existing schemes such as the space-time bounding volume method [14,16,38] and the geometry coherence method [2] to achieve computation efficiency.

4.2 Collision Detection of Arbitrarily-Shaped Object

During dynamic interaction among objects, it is possible that no collision happens for a period of time. Hence, it is more efficient to use a large time step, D_t, during such a period. A smaller time step, D_{t_c}, should be used when collision occurs.

The collision between two arbitrarily-shaped objects can be decomposed into many atomic collision processes in each of which one node of the penetrating object penetrates through a triangular surface patch of the penetrated object. The assembly of this type of atomic collision processes allows us to describe complex situations in which in some contact regions the first object penetrates the second object while in other contact regions the second penetrates the first.

Let us consider such an atomic collision process of a node of one object penetrating a triangular patch of another object. The starting and ending position of the penetrating node in a time step is designated by *e* and *f*, respectively, while the starting and ending position of the penetrated triangular patch are represented by three nodes a_0-b_0-c_0 and *a-b-c*, respectively, as shown in Figure 4.1. If the length of each time step in the transition stage from non-collision to collision is adapted to be small, the path from *e* to *f* can be approximated by a straight line. The normal vector of the penetrated triangular patch pointing to the outside of the penetrated object is denoted by $\mathbf{n}_{abc}$. The intersection between vector **ef** and triangular patch a_0-b_0-c_0 is represented by point d_0, while the intersection with triangular patch *a-b-c* is denoted by point *d*.

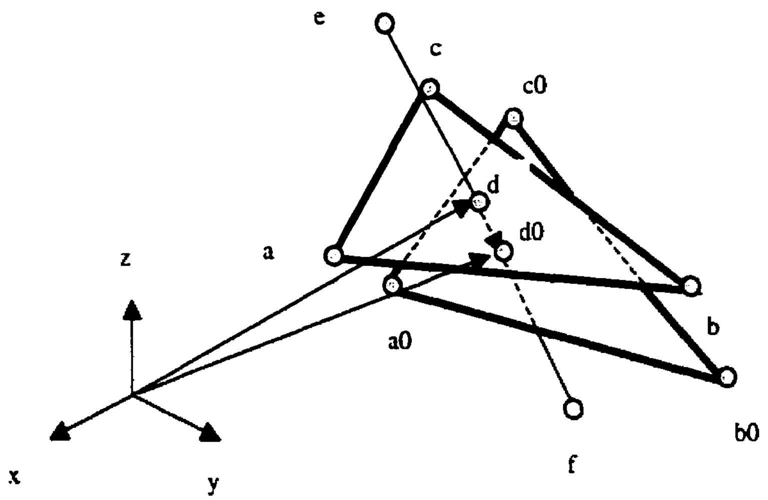


Figure 4.1: An atomic collision process of a node (e & f) on a penetrating object and a triangular patch (a-b-c & a_0-b_0-c_0) of a penetrated object.

The calculation of position of points *d* and d_0 is performed by solving the intersection point between the triangular patch (*a-b-c* or a_0-b_0-c_0) and the line determined by points *e* and *f*. For instance, the vector corresponding to the intersection point *d* formed by triangular patch *a-b-c* and line *e-f* can be expressed by

$$\begin{aligned} \mathbf{d} &= \mathbf{a} + t_1(\mathbf{c} - \mathbf{a}) + t_2(\mathbf{b} - \mathbf{a}), \\ \mathbf{d} &= \mathbf{e} + t_3(\mathbf{f} - \mathbf{e}), \end{aligned} \tag{4.1}$$

where t_1 and t_2 are the barycentric coordinates of triangle patch *a-b-c*. t_3 is the barycentric coordinate of line *e-f*.

In the Cartesian coordinate system, equation (4.1) is rewritten as

$$\begin{bmatrix} x_c - x_a & x_b - x_a & x_e - x_f \\ y_c - y_a & y_b - y_a & y_e - y_f \\ z_c - z_a & z_b - z_a & z_e - z_f \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} x_e - x_a \\ y_e - y_a \\ z_e - z_a \end{bmatrix}, \tag{4.2}$$

where x_i, y_i and z_i are the Cartesian coordinates of node i in directions X, Y and Z, respectively. A similar equation can be written for the intersection point d_0 formed by the triangular patch a_0-b_0-c_0 and line *e-f*. Equation (4.2) has a unique solution if the following determinant is not equal to zero

$$\Delta = \begin{vmatrix} x_c - x_a & x_b - x_a & x_e - x_f \\ y_c - y_a & y_b - y_a & y_e - y_f \\ z_c - z_a & z_b - z_a & z_e - z_f \end{vmatrix}.$$

The different range values of t_1, t_2, and t_3 correspond to different spatial relationships between the penetrating node and the penetrated triangular patch, as illustrated in Table 4.1.

The arbitrary spatial relationship between vector **ef** and $\mathbf{n}_{abc}$ leads to the following three categorized situations for collision detection:

(a) $\mathbf{ef} \bullet \mathbf{n}_{abc} < 0$

In this category of situations, the movement of the penetrating node is in the opposite direction of the surface outward normal $\mathbf{n}_{abc}$. Depending on the location of points *d* and d_0 relative to points *e* and *f*, the following sub-cases exist:

(a.1) d is between e and f

Figure 4.2 (a) shows that in this sub-case a collision certainly happens.

Table 4.1 Parametric range values and the corresponding spatial relationship

Condition	Meaning
$0 \le t_1 \le 1, 0 \le t_2 \le 1, t_1 + t_2 \le 1$	The intersection point d is within the triangular patch *a-b-c*. Otherwise, d is outside the patch.
$0 \le t_3 \le 1$	The intersection point d is between points e and f.
$t_3 < 0$	d is beyond e.
$t_3 > 1$	d is beyond f.

(a.2) d is beyond e

There are three variations for this sub-case. If d_0 is between e and f, a collision definitely occurs, as illustrated in Figure 4.2(b). When d_0 is beyond f, a collision will happen. If d_0 is beyond e, the collision status depends upon whether or not the penetrating node is already in the collision node list. That is if the penetrating node has already penetrated the triangular patch in the last time step, then the collision is still kept in the current time step. Otherwise, no collision is assumed in the current time step.

(a.3) d is beyond f

There are also three variations and no collision occurs in all these variations. Figure 4.2(c) shows the case where d_0 is also beyond f.

(b) $\mathbf{ef} \bullet \mathbf{n}_{abc} = 0$

This is a critical situation which is at the transition point between collision and non-collision, as shown in Figure 4.2(d). The status of collision at the previous step determines that at the current step.

(c) $\mathbf{ef} \bullet \mathbf{n}_{abc} > 0$

In this category of situations, **ef** is in the same direction as the surface outward normal of the penetrated triangular patch. Three sub-cases follow.

(c.1) *d* is between *e* and *f*

No matter what relative spatial relationship is between d_0 and **ef**, no collision happens, as illustrated in Figure 4.2(e), because $\mathbf{n}_{abc}$ always represents the outward normal of the object.

(c.2) *d* is beyond *e*

In this sub-case, no collision occurs, as illustrated in Figure 4.2(f).

(c.3) *d* is beyond *f*

No matter what spatial relationship between d_0 and **ef** is, the collision status inherits the collision status in the last time step between the penetrating node and the penetrated triangular patch, as shown in Figure 4.2(g).

If rotation is coupled with translation, the time marching step should not be chosen to be large because the curved path of the penetrating node or the penetrated triangular patches causes the errors in determining the collision point by using the intersection between **ef** and the penetrated triangular patch.

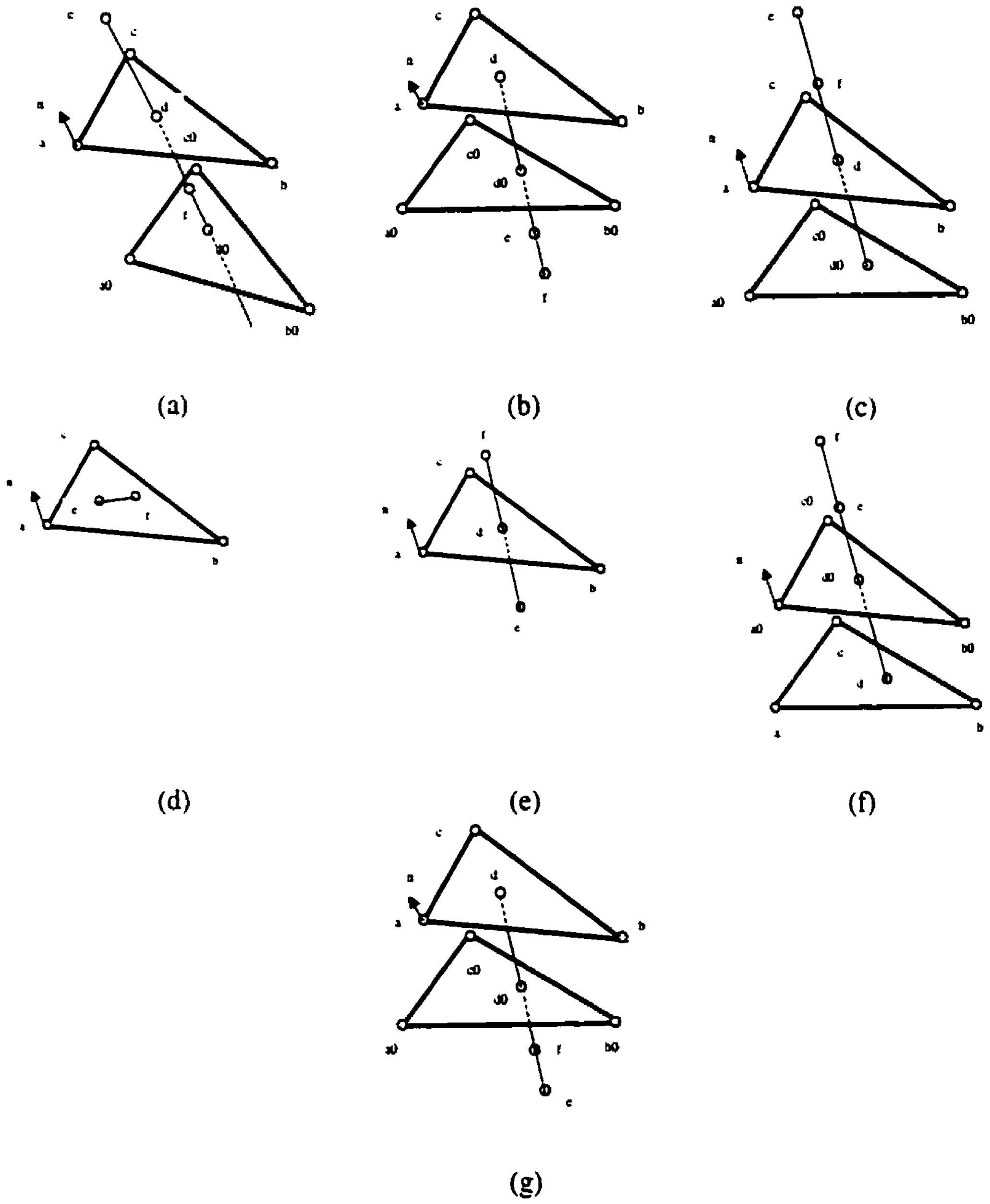


Figure 4.2: Different cases for collision detection of deformable objects.

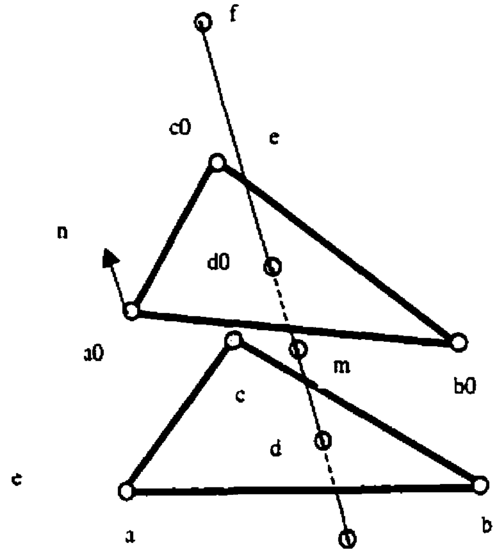


Figure 4.3: Pattern 2 for calculating the beginning of the collision.

Two special cases need to be handled in a different way. In case 1, *e* coincides with *f* and the penetrated triangular patch does not move either. Under such circumstance, the collision mode (i.e., status flag being in collision or non-collision stage) of the object remains unchanged, compared to that at previous step. In case 2, *e* still coincides with *f* but the penetrated triangular patch is moving during the current time step. The signed distance between the stationary point (*e* or *f*) and the moving triangular patch is used in such case to determine the collision status. The sign of the distance is determined by the sign of the dot product of vector *ef* and the normal of the triangular patch.

4.3 Determining the Beginning Instant of Collision

The length of the collision period between two objects depends on their stiffness. Usually, the entire length is quite short relative to the time of free travel of objects in space. Transition from the non-collision stage to the collision stage should be considered carefully if the developed algorithm is to be able to be applied to a wide spectrum of material property.

One basic strategy in the semi-explicit local collision response analysis approach is to use two different stepping time intervals for the non-collision and collision stages. Since the time interval for the non-collision stage, D_t, is much greater than that for the collision stage, D_{t_c}, the time interval between the end of the last time step of free travel and the exact beginning instant of a collision, D_{t_f}, is a fraction of D_t.

A brute-force approach of looking for D_{t_f} is the bisection method which is simple and robust, but is not efficient computationally [50]. In this study, a one-step analytical method is used to estimate D_{t_f}, which provides enough information to the collision response analyzer due to the nature of the prediction-correction procedure in the semi-explicit local collision response analysis algorithm proposed in the next section.

Depending on whether or not the penetrated triangular patch is movable, there are two basic patterns for the determination of the beginning of a collision. In pattern 1, the penetrated patch is not movable, as shown in Figure 4.2(e). The beginning of collision is represented by point *d* as follows:

$$\mathbf{d} = \mathbf{e} + \dot{\mathbf{u}}|_e\, t + \frac{1}{2}\ddot{\mathbf{u}}|_e\, t^2, \tag{4.3}$$

where $\dot{\mathbf{u}}|_e$ and $\ddot{\mathbf{u}}|_e$ refer to the velocity and acceleration of the penetrating node at position **e**, respectively. When rotation is coupled with translation, equation (4.3) should be modified to

$$\mathbf{d} = \mathbf{e} + \dot{\mathbf{u}}|_e\, t + \frac{1}{2}\ddot{\mathbf{u}}|_e\, t^2 + (\mathbf{r}|_e - \mathbf{r}'|_e), \tag{4.4}$$

where $\mathbf{r}|_e$ is the vector from the rotation center **o** to **e**, where **o** is the center of mass of the object associated with **e**. $\mathbf{r}'|_e$ is obtained by rotating $\mathbf{r}|_e$ by $\Delta\theta = \dot{\theta}|_e\, t + 0.5\ddot{\theta}|_e\, t^2$. $\dot{\theta}|_e$ and $\ddot{\theta}|_e$ refer to the angular velocity and acceleration of the penetrating node at position **e**, respectively.

For pattern 2 where the penetrated triangular patch is movable as shown in Figure 4.3, the beginning of a collision is determined by solving for t from the following equations:

$$\begin{aligned} \mathbf{m} &= \mathbf{e} + \dot{\mathbf{u}}_0|_e\, t + \frac{1}{2}\ddot{\mathbf{u}}|_e\, t^2, \\ \mathbf{m} &= \mathbf{d}_0 + \dot{\mathbf{u}}_0|_{d_0}\, t + \frac{1}{2}\ddot{\mathbf{u}}|_{d_0}\, t^2, \end{aligned} \tag{4.5}$$

where **m** refers to the location of the collision. **d** and $\mathbf{d}_0$ are the intersection points between **ef** and triangular patches a-b-c and a_0-b_0-c_0, respectively. Again, if rotation is involved, only an approximate estimation can be obtained by

$$\begin{aligned}\mathbf{m} &= \mathbf{e} + \dot{\mathbf{u}}_0 \,|_e\, t + \frac{1}{2}\ddot{\mathbf{u}}\,|_e\, t^2 + (\mathbf{r}|_e - \mathbf{r}'|_e),\\ \mathbf{m} &= \mathbf{d}_0 + \dot{\mathbf{u}}_0 \,|_{d_0}\, t + \frac{1}{2}\ddot{\mathbf{u}}\,|_{d_0}\, t^2 + (\mathbf{r}|_{d_0} - \mathbf{r}'|_{d_0}),\end{aligned} \tag{4.6}$$

The approximate solution provided by equations (4.4) and (4.6) may not be accurate enough for other collision response analyzers without a prediction-correction process. To improve on the accuracy, the approach introduced in this section should be used in conjunction with the bisection method.

Similar to the collision detection process described in Section 4.2, different spatial relationships between $\mathbf{ef}$ and $\mathbf{n}_{abc}$ correspond to different cases as shown in Table 4.2. In the table, *none* means the situation where there is no need to calculate the beginning of a collision, and *arbitrary* refers to no limitation on the variation of a parameter. t_3 and t_{3_0} correspond to the location of $\mathbf{d}$ and $\mathbf{d}_0$ on vector $\mathbf{ef}$, respectively.

4.4 Static Contact

Static contact can be viewed as a special case of dynamic collision. The unique characteristic of static contact is that objects remain in contact in a fixed relative position for an unlimited period of time until some new external forces disturb this static equilibrium. In this thesis, quasi-static contact is also categorized as static contact, in which both contacting objects may be moving in space but are stationary with respect to each other. It is important to distinguish between static contact and dynamic collision, because the time step used in the corresponding collision response analysis may be different.

Table 4.2 Calculation of the beginning of collision

$t_3 = t_{3\_0}$	if collision happens			pattern 1
	otherwise			none
$t_3 \neq t_{3\_0}$	$\mathbf{ef} \bullet \mathbf{n} < 0$	d is between e & f	d_0 is arbitrary	pattern 2
		d is beyond e	d_0 is beyond f	pattern 2
			d_0 is beyond e	none
			d_0 is between e & f	pattern 2
		d is beyond f	d_0 is arbitrary	none
	$\mathbf{ef} \bullet \mathbf{n} = 0$	d is arbitrary	d_0 is arbitrary	none
	$\mathbf{ef} \bullet \mathbf{n} < 0$	d is between e & f	d_0 is arbitrary	none
		d is beyond e	d_0 is beyond f	pattern 2
			d_0 is beyond e	none
			d_0 is between e & f	pattern 2
		d is beyond f	d_0 is arbitrary	none

The criterion to identify a static contact in an atomic collision process (Figure 4.1) is the so-called three-time-step principle in which the kinematic variables of nodes at contact areas in three contiguous time steps are examined to determine if a static contact occurs. Basically the static contact is determined by checking the following two conditions:

(1) fixed position between the penetrating node and the penetrated triangular patch

Let P denote the projection of the penetration distance of the penetrating node in the normal direction of the penetrated triangular patch at the end of each time step of the prediction-correction collision response analysis. Its backward finite difference in the time domain is written as

$$ {}^{n+1}P = {}^{n}P + h\ {}^{n}\dot{P}, \tag{4.7} $$

$$ {}^{n+1}\dot{P} = {}^{n}\dot{P} + h\ {}^{n}\ddot{P}, \tag{4.8} $$

where h is the time interval. The left superscript of P refers to the time step and the dot on top of P denotes the derivative w.r.t. time. If ${}^{n}P = {}^{n+1}P = {}^{n+2}P = 0$, then ${}^{n}\dot{P} = {}^{n+1}\dot{P} = 0$ and ${}^{n}\ddot{P} = 0$ via equations (4.7) and (4.8), respectively. Like P, the linear interpolation parameters $t_1(t)$ and $t_2(t)$ in equation (4.1) also have similar relationships as in equations (4.7) and (4.8). ${}^{n}\ddot{P}$, ${}^{n}\ddot{t}_1(t)$ and ${}^{n}\ddot{t}_2(t)$ signify the inertia property of the pair of penetrating node and penetrated triangular patch in contact. If ${}^{n}\ddot{P} = {}^{n}\ddot{t}_1(t) = {}^{n}\ddot{t}_2(t) = 0$ and no additional new external forces exert on the pair of penetrating node and penetrated triangular patch or there is no change in existing external forces, then

${}^{n+i}\ddot{P} = {}^{n+i}\ddot{t}_1(t) = {}^{n+i}\ddot{t}_2(t) = 0 \quad (i \geq 1)$. This further implies that ${}^{n+i+1}\dot{P} = {}^{n+i+1}\dot{t}_1(t) = {}^{n+i+1}\dot{t}_2(t) = 0$ and ${}^{n+i+2}P = {}^{n+i+2}t_1(t) = {}^{n+i+2}t_2(t) = 0 \quad (i \geq 1)$, i.e., a temporarily fixed position between the penetrating node and the penetrated triangular patch is formed.

(2) fixed distance between two objects

In the course of a dynamic collision between objects, the relative position between the penetrating node and the penetrated triangular patch may be fixed, but the distance between the centers of mass of two colliding objects is continuously changing. In order to distinguish between static contact and dynamic collision, let D denote the distance function between the centers of mass of objects associated with the penetrating node and the penetrated triangle patch. Its finite difference approximation in the time domain is of the similar form as in equations (4.7) and (4.8). ${}^{n}\ddot{D}$ signifies the inertial property of the pair of objects in contact. If ${}^{n}\ddot{D} = 0$ and no new external forces exert on the pair of objects or there is no change in existing external forces, then ${}^{n+i}\ddot{D} = 0 \quad (i \geq 1)$. This in turn means that ${}^{n+i+1}\dot{D} = {}^{n+i+2}D = 0 \quad (i \geq 1)$, i.e., the distance between the two objects is temporarily fixed.

The combined check of the above two conditions should be enough to detect a static contact in most cases.

4.5 Self-Collision Detection

Since the approach in this thesis is based upon the decomposition of the entire collision process into many atomic collision processes each of which is very primitive (Figure 4.1) and therefore poses no limitation to the problem to be solved, self-collision detection can be readily conducted if we admit the case where the penetrating node and the penetrated triangular patch could be from the same object. However, the penetrating node is not allowed to be one of three nodes defining the penetrated triangular patch. Inclusion of the self-collision detection obviously increases the total time cost for the collision detection.

Chapter 5

Collision Response

Collision response analysis is to describe the dynamic behavior of objects after a collision occurs. In this chapter, a new semi-explicit local collision response analysis algorithm is proposed. It contains two key components: the forward Lagrange multiplier method and the local finite element method. The advantage of the semi-explicit local collision response analysis algorithm is to provide a combination of realistic representation and fast solution (guaranteed O(n) time) for arbitrary collision response.

As to the notation, left superscripts refer to the geometric configuration of objects and left subscripts to the reference geometric configuration. A comma in right subscripts is used to represent a partial differentiation with respect to a certain coordinate, e.g., $u_{i,j} = \partial u_i / \partial x_j$. A dot on top of a variable means the differentiation w.r.t. time, i.e., $\dot{x} = dx/dt$.

5.1 Object Modeling

In order to obtain a continuous solution of the mechanics problem concerned in this thesis, variational methods of approximation can be used. They include Rayleigh and Ritz, Galerkin, Petrov-Galerkin (Weighted-residuals), Kantorovitch, Trefftz, and the finite element method [60]. All these methods except the last one are traditional variational approaches which have three main shortcomings: 1) difficulty in handling irregular domain; 2) difficulty in selecting coordinate functions; 3) dependence of coordinate functions on the geometry of the domain. These limitations can be overcome by the finite element method which is a piecewise polynomial approximation of weak forms of boundary- or initial-value problems over a domain partition of the solution to the system. Locally, the finite element method represents a function as a polynomial in much the same spirit as the classical Lagrange and Hermite interpolation methods.

5.1.1 Spatial and Time Discretization

In the 3-D Euclidean space, the geometry of the domain is partitioned into a finite number of subdomains each of which is called a finite element. Independent polynomials are defined and their linear combination forms an approximation to the solution within each element. In the domain of time, solution variables are discretized by using finite difference.

5.1.2 Description of Deformation and Stress

Let ${}^n x_i$ and ${}^0 x_i$ be the spatial coordinate system of the deformed configuration at time step *n* and the material coordinate system of the undeformed configuration, respectively. Following the notation in classical continuum mechanics [32], the Lagrangian-Green strain tensor is used to describe finite strains,

$$ {}^n_0E_{ij} = \frac{1}{2}\left[\frac{\partial\, {}^n_0u_i}{\partial\, {}^0x_j} + \frac{\partial\, {}^n_0u_j}{\partial\, {}^0x_i} + \sum_k \frac{\partial\, {}^n_0u_k}{\partial\, {}^0x_i}\frac{\partial\, {}^n_0u_k}{\partial\, {}^0x_j}\right], \tag{5.1} $$

where ${}^n_0u_i = {}^nx_i - {}^0x_i$ refers to the component of a displacement vector in direction *i*. To be compatible with the Lagrangian-Green strain tensor, the second Piola-Kirchhoff stress tensor, S_{ji} , is adopted to describe the stress state of deformed objects,

$$ {}^n_0S_{ji} = \det\mathbf{W}\ \ \mathbf{W}^{-1}\ {}^n\sigma_{\beta\alpha}\left(\mathbf{W}^{-1}\right)^T, \tag{5.2} $$

where ${}^n\sigma_{\beta\alpha}$ is the Eulerian-Cauchy stress tensor which is measured in the deformed configuration at time step *n*. **W** is the material deformation gradient defined as

$$ \mathbf{W} = \begin{bmatrix} \frac{\partial\left({}^nx_1\right)}{\partial\left({}^0x_1\right)} & \frac{\partial\left({}^nx_1\right)}{\partial\left({}^0x_2\right)} & \frac{\partial\left({}^nx_1\right)}{\partial\left({}^0x_3\right)} \\ \frac{\partial\left({}^nx_2\right)}{\partial\left({}^0x_1\right)} & \frac{\partial\left({}^nx_2\right)}{\partial\left({}^0x_2\right)} & \frac{\partial\left({}^nx_2\right)}{\partial\left({}^0x_3\right)} \\ \frac{\partial\left({}^nx_3\right)}{\partial\left({}^0x_1\right)} & \frac{\partial\left({}^nx_3\right)}{\partial\left({}^0x_2\right)} & \frac{\partial\left({}^nx_3\right)}{\partial\left({}^0x_3\right)} \end{bmatrix}, $$

where the left superscripts n and 0 refer to the deformed configuration at time step n and the original configuration at time step 0, respectively. det $\mathbf{W}$ in equation (5.2) is the determinant of the deformation gradient $\mathbf{W}$ and basically denotes the change in volume due to the deformation.

5.1.3 Material Abstraction

For the sake of simplicity, only isotropic, elastic materials are considered in this thesis. The relationship between stress and strain can be expressed as

$$\begin{bmatrix} s_{11} \\ s_{22} \\ s_{33} \\ s_{23} \\ s_{31} \\ s_{12} \end{bmatrix} = g_3 \begin{bmatrix} 1 & g_1 & g_1 & 0 & 0 & 0 \\ g_1 & 1 & g_1 & 0 & 0 & 0 \\ g_1 & g_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & g_2 \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{22} \\ e_{33} \\ 2e_{23} \\ 2e_{13} \\ 2e_{12} \end{bmatrix}, \tag{5.3a}$$

where s_{ij} and e_{ij} are stress and strain components, respectively. g_1, g_2 and g_3 are dependent upon material parameters as follows:

$$g_1 = \frac{\mu}{1-\mu}, \quad g_2 = \frac{1-2\mu}{2(1-\mu)}, \quad g_3 = \frac{E(1-\mu)}{(1-\mu)(1-2\mu)}, \tag{5.3b}$$

where E is Young's modulus and μ is Poisson's ratio. They are two elastic material parameters.

5.1.4 Partition of the Change in Geometric Configuration of Objects

The change in geometric configuration of objects can be decomposed into rotation and translation both of which can be further divided into local and global components. Global translation and rotation are called rigid-body movement, while local translation and rotation contribute to local deformation. Even though it is possible to include the local rotational degrees of freedom in an analysis [23,79], they are not used in this thesis because of their high computation cost. For elastic continuum material, we can use different magnitude of translatory displacements of nodes to approximate any moderate amount of local rotation. In the case of huge local rotation, the object is partitioned into two sub-objects. Thus, in this thesis the change in geometric configuration is partitioned into global rotation, global translation and local translation.

5.1.5 Treatment of Finite Rotation

It is more difficult to describe the rotation of objects as compared with translation because vector addition is not suitable in this case. Two typical ways to handle global rotation are as follows.

Gimbal angle approach: The basic concept of the gimbal angle approach is that the rotation of an object is interpreted as a prefixed order of successive sub-rotations corresponding to three components of the rotation $(\theta_x, \theta_y, \theta_z)$ which refer to the rotations about axes x, y and

z, respectively. Another choice to specify the orientation is the Euler angles [29]. More accurately, there are at least ten versions of Euler angles that have been used.

Even though this approach guarantees the uniqueness of the computed orientation of the object, it does not necessarily mean that the solution is correct because different order of successive sub-rotations gives different orientations. In general, an assumption of small rotation increment is required. The singularity problem, gimbal lock, will occur if one specific rotation component is equal to 90°, because a denominator in the calculation of incremental gimbal angles will become zero.

Rotation vector approach: The underlying idea of the rotation vector approach is to describe an arbitrary rotation of an object by using a specific rotation about a corresponding axis. One common practice is to use the incremental rotation matrix [33],

$$ {}^{n}\Delta\mathbf{R} = \begin{bmatrix} 1 & -{}^{n}\Delta\theta_z & {}^{n}\Delta\theta_y \\ {}^{n}\Delta\theta_z & 1 & -{}^{n}\Delta\theta_x \\ -{}^{n}\Delta\theta_y & {}^{n}\Delta\theta_x & 1 \end{bmatrix}, \tag{5.4} $$

where $({}^{n}\Delta\theta_x, {}^{n}\Delta\theta_y, {}^{n}\Delta\theta_z)$ are the increment of rotation angles w.r.t. x, y, and z axes, respectively. The precondition for the correctness of this matrix is infinitesimal rotation. However, the requirement for infinitesimal rotation will increase the total number of incremental steps tremendously. If the rotation in one time step is finite rather than infinitesimal, it is easy to prove that ${}^{n}\Delta\mathbf{R}$ no longer preserves the salient properties of an orthonormal transformation. One remedy is to reorthonormalize the rotated basis in an affine space to avoid the loss of orthogonality and unity.

In this thesis work, the Hamilton's quaternion is used as a representation of finite rotation such that no re-orthonormalization and no restriction on infinitesimal rotation and gimbal lock are required. During each time step, the sub-rotations $({}^n\Delta\theta_x, {}^n\Delta\theta_y, {}^n\Delta\theta_z)$ around three Cartesian coordinate axes can be transformed to a single rotation ${}^n\Delta\theta$ according to Theorem 1 which is proved in Appendix A. If three sub-rotations do not happen simultaneously in one time step, then the time interval needs to be divided into finer ones such that the sub-rotations occur simultaneously in each new time interval.

Theorem 1 If $({}^n\dot{\theta}_x, {}^n\dot{\theta}_y, {}^n\dot{\theta}_z)$ refers to the average angular velocities of the sub-rotations about the global Cartesian coordinate axes x, y, and z in time step n and these sub-rotations take place simultaneously in this time interval $(= h)$ with the rotation angles varying continuously from 0 to ${}^n\Delta\theta_j$ $(j = x, y, z)$, then the combination of these sub-rotations is equivalent to a single rotation about the axis $\mathbf{p} = ({}^n\dot{\theta}_x, {}^n\dot{\theta}_y, {}^n\dot{\theta}_z)$ with the rotation angle

$${}^n\Delta\theta^{(i)} = \sqrt{({}^n\Delta\theta_x)^2 + ({}^n\Delta\theta_y)^2 + ({}^n\Delta\theta_z)^2} = h\sqrt{({}^n\dot{\theta}_x)^2 + ({}^n\dot{\theta}_y)^2 + ({}^n\dot{\theta}_z)^2}\ .$$

The Hamilton's quaternion compactly represents an arbitrary rotation about an axis passing through the origin. By means of quaternion $(\cos\frac{{}^n\Delta\theta}{2}, \sin\frac{{}^n\Delta\theta}{2}\ {}^n\mathbf{n})$, the rotation matrix of an arbitrary rotation in time step n can be derived as

$$
{}^{n}\Delta\mathbf{R}=\begin{bmatrix}
\cos {}^{n}\Delta\theta+\left({}^{n}n_x\right)^2(1-\cos {}^{n}\Delta\theta) & -{}^{n}n_z \sin {}^{n}\Delta\theta+{}^{n}n_x\ {}^{n}n_y(1-\cos {}^{n}\Delta\theta) & {}^{n}n_y \sin {}^{n}\Delta\theta+{}^{n}n_x\ {}^{n}n_z(1-\cos {}^{n}\Delta\theta) \\
{}^{n}n_z \sin {}^{n}\Delta\theta+{}^{n}n_x\ {}^{n}n_y(1-\cos {}^{n}\Delta\theta) & \cos {}^{n}\Delta\theta+\left({}^{n}n_y\right)^2(1-\cos {}^{n}\Delta\theta) & -{}^{n}n_x \sin {}^{n}\Delta\theta+{}^{n}n_y\ {}^{n}n_z(1-\cos {}^{n}\Delta\theta) \\
-{}^{n}n_y \sin {}^{n}\Delta\theta+{}^{n}n_x\ {}^{n}n_z(1-\cos {}^{n}\Delta\theta) & {}^{n}n_x \sin {}^{n}\Delta\theta+{}^{n}n_y\ {}^{n}n_z(1-\cos {}^{n}\Delta\theta) & \cos {}^{n}\Delta\theta+\left({}^{n}n_z\right)^2(1-\cos {}^{n}\Delta\theta)
\end{bmatrix}, \quad (5.5)
$$

where ${}^{n}\mathbf{n}=({}^{n}n_x, {}^{n}n_y, {}^{n}n_z)$ refers to a unit vector of the rotation axis and ${}^{n}\Delta\theta$ is the incremental rotation angle about ${}^{n}\mathbf{n}$ in time step *n*. ${}^{n}\Delta\mathbf{R}$ is the incremental rotation matrix.

Since a rotation matrix corresponds to a specific type of orthogonal transformation in an affine space, it is easy to derive ${}^{n}\mathbf{R}={}^{n}\Delta\mathbf{R}\ {}^{n-1}\mathbf{R}$ (Appendix B), where ${}^{n}\mathbf{R}$ and ${}^{n-1}\mathbf{R}$ are rotation matrices in time steps *n* and *n*-1, respectively. In an incremental analysis, the entries in ${}^{n}\Delta\mathbf{R}$ can be calculated according to Theorem 1 and then ${}^{n}\mathbf{R}$ is computed by ${}^{n}\mathbf{R}={}^{n}\Delta\mathbf{R}\ {}^{n-1}\mathbf{R}$. The initial rotation matrix, ${}^{0}\mathbf{R}$, equals the unity matrix.

5.2 Governing Equations of System

As a compromise between efficiency and functionality, the Lagrangian dynamics is used to describe global rotation because of its simplicity and the finite element method is used to describe translation because of its flexibility. On the basis of D'Alembert's principle and the principle of virtual displacement, the semi-discrete system governing equations that describe the equilibrium of a system is approximated by

$$\mathbf{M}\ {}^{n}\ddot{\mathbf{u}}_t + \mathbf{C}\ {}^{n}\dot{\mathbf{u}}_t + {}^{n-1}\mathbf{K}\ {}^{n}\mathbf{u}_t = {}^{n}\mathbf{R}\,, \tag{5.6}$$

$$\frac{d}{dt}\left({}^{n}\mathbf{M}_I\ {}^{n}\dot{\theta}\right) = {}^{n}\mathbf{Q}\ \ , \tag{5.7}$$

where the global mass matrix **M** and damping matrix **C** are assumed to be constant. ${}^{n}\mathbf{M}_I$ is the inertia tensor. ${}^{n-1}\mathbf{K}$ refers to the global stiffness matrix at step n-1. ${}^{n}\mathbf{R}$ is the external load vector and ${}^{n}\mathbf{Q}$ the torque vectors caused by external forces. ${}^{n}\mathbf{u}_t$, ${}^{n}\dot{\mathbf{u}}_t$, ${}^{n}\ddot{\mathbf{u}}_t$ are translatory displacement, velocity, acceleration, respectively, and determined by central difference as follows

$${}^{n}\dot{\mathbf{u}}_t = \frac{1}{2h}\left\{{}^{n+1}\mathbf{u}_{t_1} - {}^{n-1}\mathbf{u}_t\ \right\}, \tag{5.8}$$

$${}^{n}\ddot{\mathbf{u}}_t = \frac{1}{h^2}\left\{{}^{n+1}\mathbf{u}_t\ - 2\ {}^{n}\mathbf{u}_t + {}^{n-1}\mathbf{u}_{t_1}\right\}, \tag{5.9}$$

where h is the interval in each time step.

5.3 Geometric and Loading Constraints due to Contact

There are two types of constraints which must be considered during a collision. The first one is the geometric constraint which imposes the requirement of geometric coherence to the displacement of two colliding objects, such as the prevention of inter-penetration between objects and the allowance of sliding between objects if the tangential force exceeds the frictional capacity at the interface. The global geometric constraint at time step $n+1$ can be expressed by

$$ {}^{n+1}\mathbf{G}\left\{{}^{n+1}_{n}\Delta\mathbf{u} + {}^{n}\mathbf{X}\right\} = {}^{n+1}\mathbf{G}\;{}^{n+1}\mathbf{X} = 0, \tag{5.10} $$

where ${}^{n}\mathbf{X}$ and ${}^{n+1}\mathbf{X}$ are coordinate vectors at time step n and $n+1$, respectively. ${}^{n+1}_{n}\Delta\mathbf{u} = {}^{n+1}\mathbf{u} - {}^{n}\mathbf{u}$. ${}^{n+1}\mathbf{u}$ is the total displacement vector at time step $n+1$ with two components, ${}^{n+1}\mathbf{u} = {}^{n+1}\mathbf{u}_t + {}^{n+1}\mathbf{u}_r$, where ${}^{n+1}\mathbf{u}_t$ and ${}^{n+1}\mathbf{u}_r$ are the linear displacement vector contributed by translation and rotation, respectively.

For the atomic process in Figure 5.1, the interpolation of coordinates within each triangular patch is linear. The parameters t_1, t_2, and t_3 are calculated by equation (4.2). Let the sub-matrix ${}^{n+1}\mathbf{G}^{(j)}$ of ${}^{n+1}\mathbf{G}$ represent part of the geometric constraint matrix corresponding to the contribution of the penetrating node j whose starting and ending position is e and f, respectively, in time step $n+1$. The local geometric constraint with respect to penetrating node j can be expressed in the following generic way (here, 'generic' means that the constraint matrix is independent of the geometric format of the surface of the object):

$$ {}^{n+1}\mathbf{G}^{(j)}\ {}^{n+1}\mathbf{X}^{(j)} = 0 \ , \tag{5.11a} $$

where

$$ {}^{n+1}\mathbf{G}^{(j)} = \begin{bmatrix} 1-t_1-t_2 & 0 & 0 & t_2 & 0 & 0 & t_1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1-t_1-t_2 & 0 & 0 & t_2 & 0 & 0 & t_1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1-t_1-t_2 & 0 & 0 & t_2 & 0 & 0 & t_1 & 0 & 0 & -1 \end{bmatrix}, $$

$$ {}^{n+1}\mathbf{X}^{(j)} = \begin{bmatrix} x_a & y_a & z_a & x_b & y_b & z_b & x_c & y_c & z_c & x_j & y_j & z_j \end{bmatrix}^T . \tag{5.11b} $$

The global geometric constraint ${}^{n+1}\mathbf{G}$ is formed by assembling the local geometric constraint ${}^{n+1}\mathbf{G}^{(j)}$ over all penetrating nodes similar to assembling the element stiffness matrices into the global stiffness matrix.

The second type of constraint is the loading constraint. It is assumed that the contact force at each penetrating node is transferred to the three nodes of the penetrated triangular patch through a linear interpolation which is the same as the interpolation of coordinates in equation (5.11b). The contact forces at both the penetrating node and the nodes of the penetrated triangular patches are considered as external loading to the system. Such extra loading contributed by the penetrating node j is calculated by $({}^{n+1}\mathbf{G}^{(j)})^T\ {}^{n}\mathbf{\Lambda}^{(j)}$, where ${}^{n}\mathbf{\Lambda}^{(j)} = \begin{bmatrix} {}^{n}\lambda_x^{(j)} & {}^{n}\lambda_y^{(j)} & {}^{n}\lambda_z^{(j)} \end{bmatrix}^T$ contains the components of contact force at node j in directions x, y, and z. It should be noted that only the contact forces at the penetrating nodes are independent unknowns, while those at the nodes of the penetrated triangular patches are dependent through the Newtonian action-reaction law and the linear interpolation. Such type of linear interpolation of the penetrating force among three nodes of the penetrated triangular patch guarantees the equilibrium of translatory forces but not rotational forces in the system. In order to conserve angular momentum for the contact, algorithmic moment arms (numerically-corrected moment arms) should be used. In computer animation, this numerical correction may not be needed depending upon the desired accuracy imposed by the users. The assembly of the extra loading caused by all penetrating nodes leads to the global contact force vector $({}^{n-1}\mathbf{G})^T\ {}^{n}\mathbf{\Lambda}$ which will be used in a set of modified system governing equations.

Another aspect of the loading constraint is that the contact forces should obey the basic friction law. In this thesis, the basic Coulomb law is adopted and the static friction coefficient is assumed to be the same as the dynamic one. Let us consider the penetrating node j with a contact force vector (FC_x, FC_y, FC_z), i.e.,

$$FC_x = \lambda_x^{(j)}, \quad FC_y = \lambda_y^{(j)}, \quad FC_z = \lambda_z^{(j)}. \tag{5.12a}$$

If the normal vector of the penetrated triangular patch is expressed by (X_1, Y_1, Z_1), the normal force $\mathbf{FC}_n$ and the norm of the tangential force $|\mathbf{FC}_t|$ in Figure 5.1 are expressed as

$$\mathbf{FC}_n = (X_1 FC_x + Y_1 FC_y + Z_1 FC_z)\frac{\mathbf{n}_{abc}}{|\mathbf{n}_{abc}|} \quad \text{and} \tag{5.12b}$$

$$|\mathbf{FC}_t| = \sqrt{FC_x^2 + FC_y^2 + FC_z^2 - |\mathbf{FC}_n|^2}\,, \tag{5.12c}$$

respectively. If $\mathbf{FC}_n$ is along the direction $-\mathbf{n}_{abc}$, then a tension mode is reached. In such case, both the normal and tangential contact forces at the penetrating node are set to be zero. Let ϕ be the friction angle between two objects. The Coulomb friction law, $|\mathbf{FC}_t| = |\mathbf{FC}_n| \cdot \tan\phi$, is enforced by using a Gauss-Seidel iterative algorithm (Appendix C) in which the tangential contact force is updated at each step.

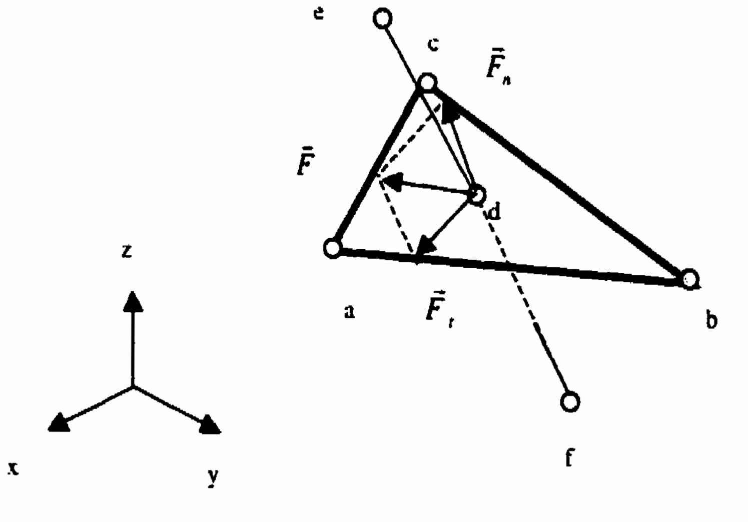


Figure 5.1: Normal and tangential components of the contact force.

5.4 A New Semi-Explicit Local Collision Response Analysis Algorithm

Before the new algorithm is presented, the conventional Lagrange multiplier method and its relationship with time integration are introduced first.

Conventional Lagrange Multiplier Method

For arbitrary collision between deformable objects, Lagrange multipliers may be introduced into the equation of motion as follows:

$$\mathbf{M}\ {}^{n+1}\ddot{\mathbf{u}}+\mathbf{K}\ {}^{n+1}\mathbf{u}+{}^{n+1}\mathbf{G}^{T}\ {}^{n+1}\lambda={}^{n+1}\mathbf{R}, \tag{5.13a}$$

$${}^{n+1}\mathbf{G}\{\ {}^{n+1}\mathbf{u}+\mathbf{X}\}=\mathbf{0}. \tag{5.13b}$$

where **M**, **K** and **R** are the mass vector, stiffness matrix and external load vector, respectively. Damping matrix **C** is omitted here simply for the sake of concise description and will be added later. Left superscript, n+1, refers to time step n+1. λ is the Lagrange multiplier vector which represents the surface contact forces. **G** is a geometric constraint matrix to prevent any penetration between colliding objects and its order likely varies during a dynamic contact process. ${}^{n+1}\mathbf{u}$, ${}^{n+1}\dot{\mathbf{u}}$, ${}^{n+1}\ddot{\mathbf{u}}$ and **X** are the nodal displacement, velocity, acceleration vectors at time step n+1 and nodal coordinate vector at step 0, respectively.

Mathematically, equation (5.13) represents a set of linear differential equations of second order and, in principle, we should be able to solve it using standard procedures for differential equations [22]. However, the procedures for the solution of general differential equations could become computationally expensive if the matrices are large, unless some special methods are used by taking advantage of the characteristics of the matrices **K**, **C** and **M** [8]. Let's take a look at one of the very few effective methods for the solutions of dynamic analysis, the Beta-*m* direct integration method proposed by Katona and Zienkiewicz [41]. Here the term 'direct' means that prior to the numerical integration, no transformation of the equations into a different form is conducted. '*m*' implies the highest derivative to be retained. For the sake of efficiency, the second order Beta-2 method is used as follows

$$ {}^{n+1}\mathbf{u} = \mathbf{q}_0 + b_0\,\Delta({}^{n+1}\ddot{\mathbf{u}}), \tag{5.14a}$$

$$ {}^{n+1}\dot{\mathbf{u}} = \mathbf{q}_1 + b_1\Delta({}^{n+1}\ddot{\mathbf{u}}), \tag{5.14b}$$

$$ {}^{n+1}\ddot{\mathbf{u}} = \mathbf{q}_2 + b_2\Delta({}^{n+1}\ddot{\mathbf{u}}), \tag{5.14c}$$

where Δ is a finite difference operator and

$$ \Delta({}^{n+1}\ddot{\mathbf{u}}) = {}^{n+1}\ddot{\mathbf{u}} - {}^{n}\ddot{\mathbf{u}}, \tag{5.14d}$$

$$ \mathbf{q}_0 = {}^{n}\mathbf{u} + h\ {}^{n}\dot{\mathbf{u}} + \frac{1}{2}h^2\ {}^{n}\ddot{\mathbf{u}}, \tag{5.14e}$$

$$ \mathbf{q}_1 = {}^{n}\dot{\mathbf{u}} + h\ {}^{n}\ddot{\mathbf{u}}, \tag{5.14f}$$

$$ \mathbf{q}_2 = {}^{n}\ddot{\mathbf{u}}, \tag{5.14g}$$

and

$$b_0 = \frac{1}{2} h^2 \beta_0, \tag{5.14h}$$

$$b_1 = h\beta_1, \tag{5.14i}$$

$$b_2 = 1, \tag{5.14j}$$

$$h = {}^{n+1}t - {}^{n}t, \tag{5.14k}$$

where ${}^{n+1}t$ and ${}^{n}t$ are time instant at step n+1 and n, respectively.

Two well-known Beta-2 methods are the constant-average-acceleration method, also known as the trapezoidal rule ($\beta_0 = \beta_1 = 0.5$) and a single step version of the central difference method ($\beta_0 = 0,\ \beta_1 = 0.5$). If $\beta_0 = 0$, the time integration is explicit. Otherwise, it is referred to as implicit, i.e., ${}^{n+1}\mathbf{u}$ is dependent upon ${}^{n+1}\ddot{\mathbf{u}}$ as specified in equation (5.14a).

Substitution of equation (5.14) into equation (5.13), which corresponds to the conventional Lagrange multiplier method, leads to

$$\begin{bmatrix} [b_2\mathbf{M} + b_0\mathbf{K}] & {}^{n+1}\mathbf{G}^T \\ b_0\ {}^{n+1}\mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta({}^{n+1}\ddot{\mathbf{u}}) \\ {}^{n+1}\lambda \end{bmatrix} = \begin{bmatrix} {}^{n+1}\mathbf{R} - \{\mathbf{M}q_2 + \mathbf{K}q_0\} \\ -{}^{n+1}\mathbf{G}\{g_0 + \mathbf{X}\} \end{bmatrix}, \tag{5.15}$$

where the rows of geometric constraint matrix ${}^{n+1}\mathbf{G}$ are linearly independent. If $b_0 = 0$, then the above system of equations is singular. According to equation (5.14h), $b_0 = 0$ corresponds to $\beta_0 = 0$, i.e., the explicit time integration. Therefore, explicit time integration will cause the singularity problem of motion equation (5.15). One intuitive explanation is that in the explicit time integration, the time-dependent variables are approximated by only known variables at the previous time step leading to inaccurate or singular solutions. To avoid this singularity problem, the forward Lagrange multiplier method is introduced below.

Forward Lagrange Multiplier Method

The basic concept of the forward Lagrange multiplier method is that displacement constraints at time step ${}^{n+1}t$ are correlated with Lagrange multipliers at time step ${}^{n}t$. It was originally proposed in numerical analysis in a simple 2-D format [18]. In this thesis, to the best knowledge of the author, it is applied for the first time to computer graphics and extended to 3-D format as follows:

$$\mathbf{M}\ {}^{n}\ddot{\mathbf{u}}_{t} + \mathbf{C}\ {}^{n}\dot{\mathbf{u}}_{t} + {}^{n}\mathbf{K}\ {}^{n}\mathbf{u}_{t} + \left({}^{n+1}\mathbf{G}\right)^{T}\ {}^{n}\Lambda\ = {}^{n}\mathbf{F}\ , \tag{5.16}$$

$${}^{n+1}\mathbf{G}\left\{{}^{n+1}\mathbf{u} + {}^{0}\mathbf{X}\right\} = 0 \quad . \tag{5.17}$$

where ${}^{n}\Lambda$ is the Lagrange multiplier vector which basically consists of the contact forces at the interface between objects. $\left({}^{n+1}\mathbf{G}\right)^{T}$ is the transposed constraint matrix which is one time step ahead of the other terms in equation (5.16). The right subscript t denotes translation. Compared with equation (5.13), a special characteristics of equation (5.16) is that the geometric constraint matrix is one step ahead of the Lagrange multiplier vector. Substitution of equation (5.14) with $\beta_0 = 0$ and $\beta_1 = 0.5$ into equation (5.16) leads to

$$
{}^{n+1}\mathbf{u}_t = \left(\frac{\mathbf{M}}{h^2}+\frac{\mathbf{C}}{2h}\right)^{-1}
\left({}^{n}\mathbf{F}+\frac{\mathbf{M}}{h^2}(2\,{}^{n}\mathbf{u}_t - {}^{n-1}\mathbf{u}_t)+\frac{\mathbf{C}}{2h}\,{}^{n-1}\mathbf{u}_t - {}^{n}\mathbf{K}\,{}^{n}\mathbf{u}_t - \left({}^{n+1}\mathbf{G}\right)^T\,{}^{n}\boldsymbol{\Lambda}\right) \tag{5.18}
$$

The right-hand side of equation (5.18) can be divided into two components:

$$
{}^{n+1}\mathbf{u}_t^{*} = \left(\frac{\mathbf{M}}{h^2}+\frac{\mathbf{C}}{2h}\right)^{-1}\left({}^{n}\mathbf{F}+\frac{\mathbf{M}}{h^2}(2\,{}^{n}\mathbf{u}_t - {}^{n-1}\mathbf{u}_t)+\frac{\mathbf{C}}{2h}\,{}^{n-1}\mathbf{u}_t - {}^{n}\mathbf{K}\,{}^{n}\mathbf{u}_t\right), \tag{5.19a}
$$

$$
{}^{n+1}\mathbf{u}_t^{\lambda} = \left(\frac{\mathbf{M}}{h^2}+\frac{\mathbf{C}}{2h}\right)^{-1}\left(-\left({}^{n+1}\mathbf{G}\right)^T\,{}^{n}\boldsymbol{\Lambda}\right), \tag{5.19b}
$$

where ${}^{n+1}\mathbf{u}_t^{*}$ refers to the translatory displacement vector without considering contact forces in step n+1 in the prediction stage. ${}^{n+1}\mathbf{u}_t^{\lambda}$ is the translatory displacement vectors due to contact forces in the correction stage. The displacement vector after the prediction-correction process becomes

$$
{}^{n+1}\mathbf{u}_t = {}^{n+1}\mathbf{u}_t^{*} + {}^{n+1}\mathbf{u}_t^{\lambda}. \tag{5.20}
$$

The combination of equations (5.17), (5.19b), (5.20) and ${}^{n+1}\mathbf{u} = {}^{n+1}\mathbf{u}_t + {}^{n+1}\mathbf{u}_r$ gives

$$
{}^{n}\boldsymbol{\Lambda} = \left[{}^{n+1}\mathbf{G}\left(\frac{\mathbf{M}}{h^2}+\frac{\mathbf{C}}{2h}\right)^{-1}\left({}^{n+1}\mathbf{G}\right)^T\right]^{-1}{}^{n+1}\mathbf{G}\left[{}^{n+1}\mathbf{u}_r + {}^{n+1}\mathbf{u}_t + {}^{0}\mathbf{X}\right], \tag{5.21}
$$

where ${}^{n+1}\mathbf{u}_r$ is the linear displacement vector contributed by rotation and can be further divided into two components,

$$ {}^{n+1}\mathbf{u}_r = {}^{n+1}\mathbf{u}_r^{\bullet} + {}^{n+1}\mathbf{u}_r^{\lambda} = {}^{n}\mathbf{u} + \Delta\left({}^{n+1}\mathbf{u}_r^{\bullet}\right) + \Delta\left({}^{n+1}\mathbf{u}_r^{\lambda}\right), \tag{5.22} $$

where $\Delta\left({}^{n+1}\mathbf{u}_r^{\bullet}\right)$ is determined by the incremental rotation matrix $\Delta\left({}^{n+1}\mathbf{R}_r^{\bullet}\right)$ which corresponds to the rotation angle $h\,{}^{n}\dot{\theta}$ and $\Delta\left({}^{n+1}\mathbf{u}_r^{\lambda}\right)$ is dependent on the incremental $\Delta\left({}^{n+1}\mathbf{R}_r^{\lambda}\right)$ caused by the rotation angle $\frac{h}{2}\left({}^{n+1}\dot{\theta} - {}^{n}\dot{\theta}\right)$. h is the time interval of step $n+1$.

Local Finite Element Method

In this section, a new variation of the conventional finite element method, the local finite element method, is described. Here, 'local' means that no factorization, inversion or assembly of any global matrix is required, i.e., all calculations are performed at the local element level. Actually, there is even no need for storing any global matrix in this approach.

In the local finite element method, the first step is to let matrices **M** and **C** be in lumped forms. In the next step, 4-node tetrahedron constant-strain elements are used in the system. Since the stress and strain within each element are constant, the values of stress and strain of each element need to be calculated once for each time step. In addition, the calculation of stress and strain in such an element is trivial.

For the translatory displacement, the following inner loop proceeds as follows:

- In the prediction stage, the displacement at current time step n+1 is calculated by using equation (5.18). The calculation reflects the initial estimate without considering the contribution from contact surface forces. Since matrices $\mathbf{M}$ and $\mathbf{C}$ are in lumped forms and more importantly the calculation of the term ${}^{n}\mathbf{K}\,{}^{n}\mathbf{u}$ requires only the information of last time step n which is known at the current time step n+1, the calculation of equation (5.18) can proceed element by element without using global matrices. In addition, because the primitive 4-node tetrahedron elements are used, element-wise calculation of each sub-part of ${}^{n}\mathbf{K}\,{}^{n}\mathbf{u}$ becomes trivial.
- In the correction stage, the contribution of contact surface forces to the displacement at current time step n+1 is calculated by using equation (5.19). The term ${}^{n}\mathbf{\Lambda}$ refers to the contact surface forces at last time step n, which is known at the current time step. The term ${}^{n+1}\mathbf{G}$ is the generic geometric constraint matrix at the current time step n+1. Before the beginning of the inner loop, ${}^{n+1}\mathbf{G}$ is estimated by using the coordinates at the last time step. After one or more passes of the inner loop, it is updated by using the coordinates at the current time step. In either way, the calculation of equation (5.19) proceeds element by element by replacing ${}^{n+1}\mathbf{G}$ with its sub-matrix ${}^{n+1}\mathbf{G}^{(j)}$ which is associated with each element at contact surface and defined in equation (5.11b).
- In the loading constraint stage, the Coulomb friction law is enforced. The contact surface forces at last time step n are updated by using equation (5.21). The terms ${}^{n+1}\mathbf{G}$, ${}^{n+1}\mathbf{u}_t$ and ${}^{n+1}\mathbf{u}_r$ are calculated by using the most up-to-date information at the current time step. Similarly, the calculation of equation (5.21) is performed on an element basis by

replacing ${}^{n+1}\mathbf{G}$ with its sub-matrix ${}^{n+1}\mathbf{G}^{(j)}$ which is associated with each element at contact surface.

An outer loop, a Gauss-Seidel iteration, is used to correctly impose the loading constraint at the contact surface, i.e., the Coulomb friction law which is explained in Section 3.3. Numerical experiments indicate that the contact surface forces subject to the Coulomb friction law converge in about 3 passes of the outer loop. In this work, the number of iteration for the outer loop is set to 3. Therefore, the overall computational cost of this two-layer loop is guaranteed to be $O(n)$, where n is the total number of domain nodes.

As to the global rotation, in favor of computational efficiency the Lagrangian dynamics equation for angular state variables in the collision stage is approximately transformed to

$$ {}^{n}\mathbf{M}_{I}\ {}^{n}\ddot{\theta} + {}^{n}\dot{\theta} \times \left({}^{n}\mathbf{M}_{I}\ {}^{n}\dot{\theta} \right) + {}^{n}\mathbf{Q}^{\lambda} = {}^{n}\mathbf{Q} \ , \tag{5.23} $$

where ${}^{n}\mathbf{Q}^{\lambda}$ and ${}^{n}\mathbf{Q}$ are the torque vectors caused by contact and external forces, respectively. ${}^{n}\mathbf{M}_{I}$ is the inertia tensor. From this relationship, the increment of angular velocity is approximated by

$$ {}^{n+1}\Delta\dot{\theta} = h\left({}^{n}\mathbf{M}_{I} \right)^{-1} \left\{ {}^{n}\mathbf{Q} - {}^{n}\mathbf{Q}^{\lambda} - {}^{n}\dot{\theta} \times \left({}^{n}\mathbf{M}_{I}\ {}^{n}\dot{\theta} \right) \right\}. \tag{5.24} $$

From the result of the left-hand-side of the above equation, ${}^{n+1}\Delta\mathbf{R}_{r}^{*}$ and ${}^{n+1}\Delta\mathbf{R}_{r}^{\lambda}$ can be calculated and used for the calculation of equation (5.22). The computational cost for the object rotation is bound by $O(b)$, where b is the total number of objects in the systems and is usually much smaller than the total number of domain nodes. Therefore, it is negligible. The computational cost of equation (5.22) is O(n). Thus, at each time step the total computational cost for calculating both translation and rotation is still O(n). The algorithmic outline of the above scheme is given in Appendix D.

It should be noted that in the outer loop described above, ${}^{n+1}\mathbf{u}_t^{\lambda}$ in equation (5.19b) and ${}^{n}\boldsymbol{\Lambda}$ in equation (5.21) are calculated by using the updated geometric configuration (${}^{n+1}\mathbf{G}$, ${}^{n+1}\mathbf{u}_r$, ${}^{n+1}\mathbf{u}_t$) at time step n+1 in the manner of Gauss-Seidel iteration, even though only three passes are normally required for the purpose of computer animation. Therefore, the integration used here is not merely an explicit time integration method except that the pure explicit time integration is used in the first Gauss-Seidel iteration to obtain the first estimation. More accurately, the author calls it as a semi-explicit time integration method which avoids the stability problem of time steps associated with the pure explicit time integration. In addition, the use of Lagrange multipliers does not require decreasing the stable time step, in contrast to the addition of the penalty in the penalty method [10]. Overall, the approach described in this section is called semi-explicit local collision response analysis method.

5.5 Adaptive Time Integration

By considering the difference in the time length in free-travel and collision, a two-time-interval scheme is used here. The basic idea is to use a big time interval for free-travel in space and to use a small time interval for collision. In this way, the total computation cost would not increase dramatically and in the meantime all key behaviors of object interaction and movement are captured.

The outline of the scheme is illustrated in Appendix E. Before an incremental step-by-step integration analysis, the displacement vector at time step *n* is estimated as

$${}^{n}\mathbf{u}_t = {}^{n+1}\mathbf{u}_t - h\ {}^{n+1}\dot{\mathbf{u}}_t + 0.5h^2\ {}^{n+1}\ddot{\mathbf{u}}_t, \qquad (5.25)$$

$${}^{n}\theta = {}^{n+1}\theta - h\ {}^{n+1}\dot{\theta} + 0.5h^2\ {}^{n+1}\ddot{\theta}, \qquad (5.26)$$

where $\mathbf{u}_t$ and *6* refer to translatory displacement and rotational angle vectors, respectively. *h* is the time interval between step *n* and *n*+1.

The mass at each node is assumed to be constant during a collision process. Considering the time efficiency of the algorithm, lumped mass is adopted such that only diagonal entries of the mass matrix are non-zero and the calculation of the inverse mass matrix becomes trivial.

A linked list, *CollisionList*, is used to contain the information of all penetrating nodes and the corresponding nodes of the penetrated triangular patches as well as the information of the intersection points. A boolean flag, *CollisionMode*, is used to store the result of collision detection. In each time step, the status of *CollisionList* and *CollisionMode* is checked and the resulting space-time stage, which is defined in Section 1.3, is inferred according to Table 5.1. For the collision among multiple objects, this method can be easily extended.

Special treatment is needed for the transition stage from the non-collision to collision. Since the time interval of the previous steps is D_t which is usually much larger than the time interval in a collision process, D_{t_c}, we need to calculate a corrected time interval for the transition stage, D_{t_f}, which is usually a fraction of D_t . The basic procedure to determine D_{t_f} is to obtain the minimum of the real roots of t in either equations (4.5) or (4.6) with respect to all penetrating nodes.

Table 5.1 *Space-time stage* determined by *CollisionList* and *CollisionMode*

CollisionList	*CollisionMode*	*Space-time stage*
Empty	No	Non-collision
Empty at the beginning	Yes	Beginning of collision
Empty at the end	Yes	End of collision
Not empty	arbitrary	collision

5.6 Numerical Experiments

The semi-explicit local collision response analysis approach is implemented using MS Visual C++ 5.0 under Windows NT 4.0. The animation results in animated gif format can be found under feanim package of projects at http://www.cs.usask.ca/research/research_groups/vision/. The approach is verified both qualitatively and quantitatively below.

5.6.1 Qualitative Evaluation

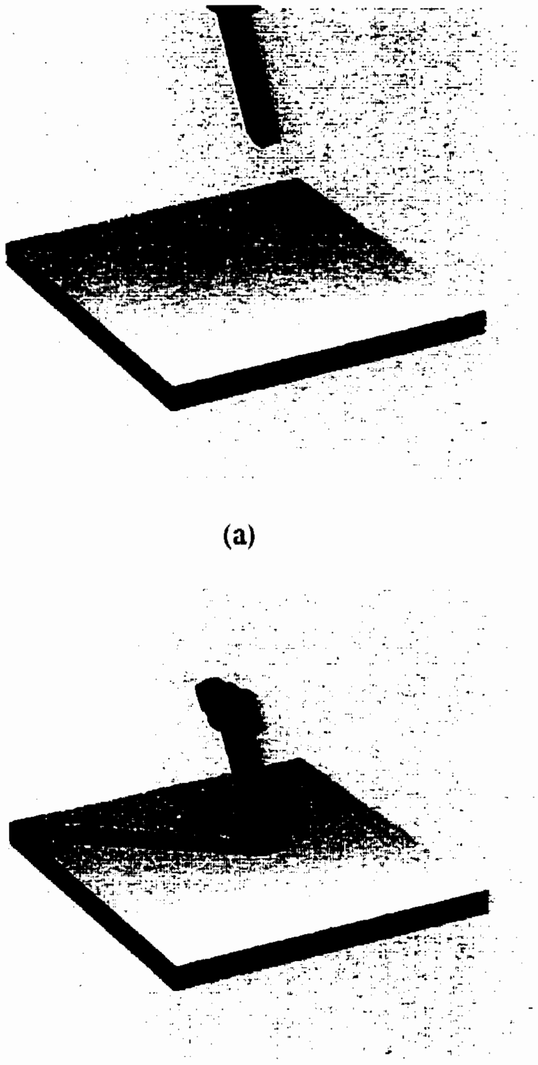
The qualitative evaluation is carried out by conducting a number of simulation tests. If the objects in these tests move in the expected manner after collision, then the simulation is accepted.

Figure 5.2 shows an example of an oblique collision between an elastic hammer and the hard ground. The hammer is designed in such a way that the length of its head is different in two directions. The asymmetrical dancing (deformation) of the hammer is caused by the oblique collision and asymmetrical parts of the hammer head. Figure 5.2(f) shows how the ground limits and supports the deformation of the hammer when its head and tail touch the ground.

The second example (Figure 5.3) demonstrates the impact process of a hammer dropping onto a flexible table. Figure 5.3 (a) is the initial state, while (b) and (c) are different views of the moment of impact. Both the hammer and the table deform in a free style as expected.

The third example shows the oblique impact of a flying hammer with a lamp which sits on a table as illustrated in Figure 5.4. Two objects (lamp and table) are initially in static contact mode and the hammer is moving in space due to gravity and the initial speed. The collision between the hammer and the lamp is illustrated by three frames in (a), (b) and (c). Both rotation and sliding contact are involved in this example.

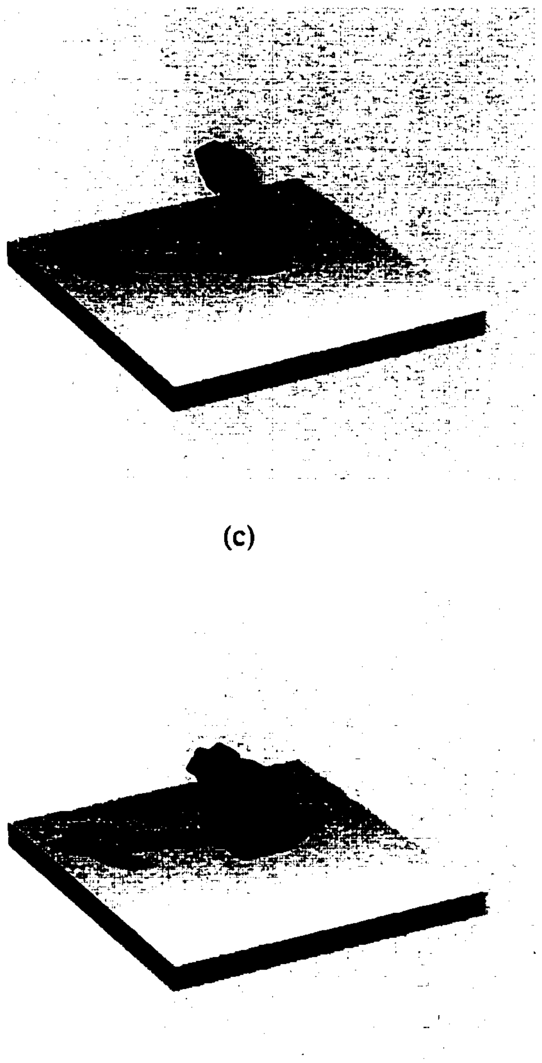
The fourth example is an elastic cup colliding with a rigid wall obliquely as illustrated in Figure 5.5 which demonstrates the asymmetrical local deformation of the cup. This delicate detail is difficult, if not impossible, to be produced by using the global deformation approach.



(a)

(b)

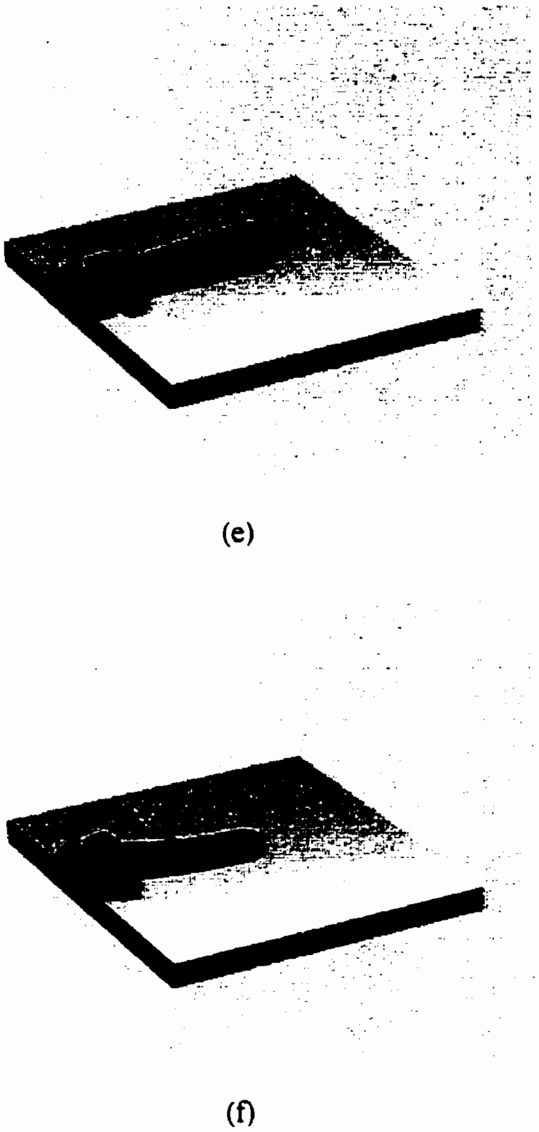
Figure 5.2: Images illustrating an oblique collision between a flexible hammer and a rigid plate. (nodes: 64, elements: 70, CRA time cost: 0.48 sec/frame)



(c)

(d)

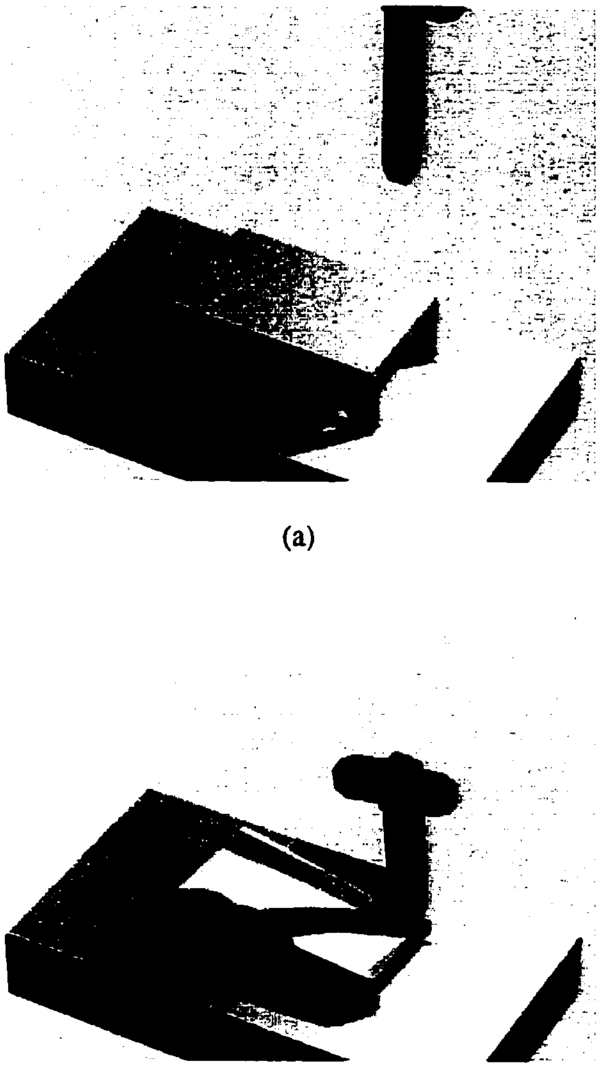
Figure 5.2: Images illustrating an oblique collision between a flexible hammer and a rigid plate. (nodes: 64, elements: 70, time cost of collision response analysis: 0.48 sec/frame)



(e)

(f)

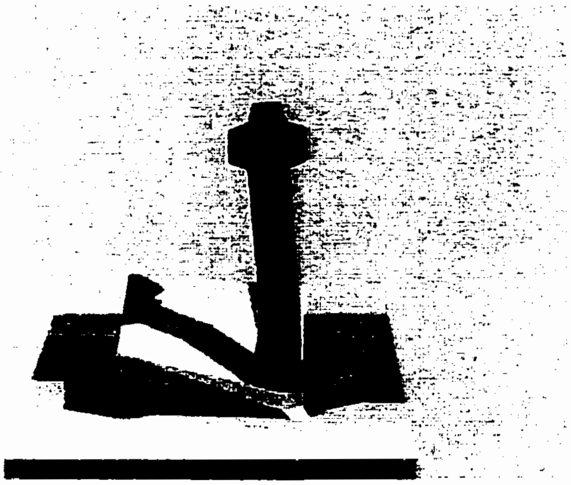
Figure 5.2: Images illustrating an oblique collision between a flexible hammer and a rigid plate. (nodes: 64, elements: 70, CRA time cost: 0.48 sec/frame)



(a)

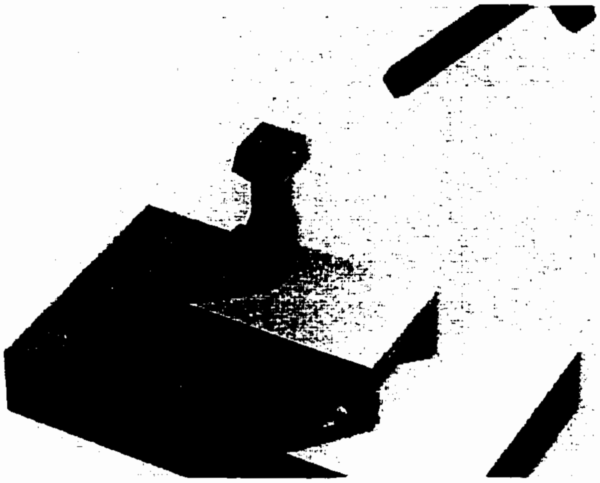
(b)

Figure 5.3: Images illustrating the collision between a flexible hammer and a flexible table. (nodes: 156, elements: 180, time cost of collision response analysis: 2.2 sec/frame)



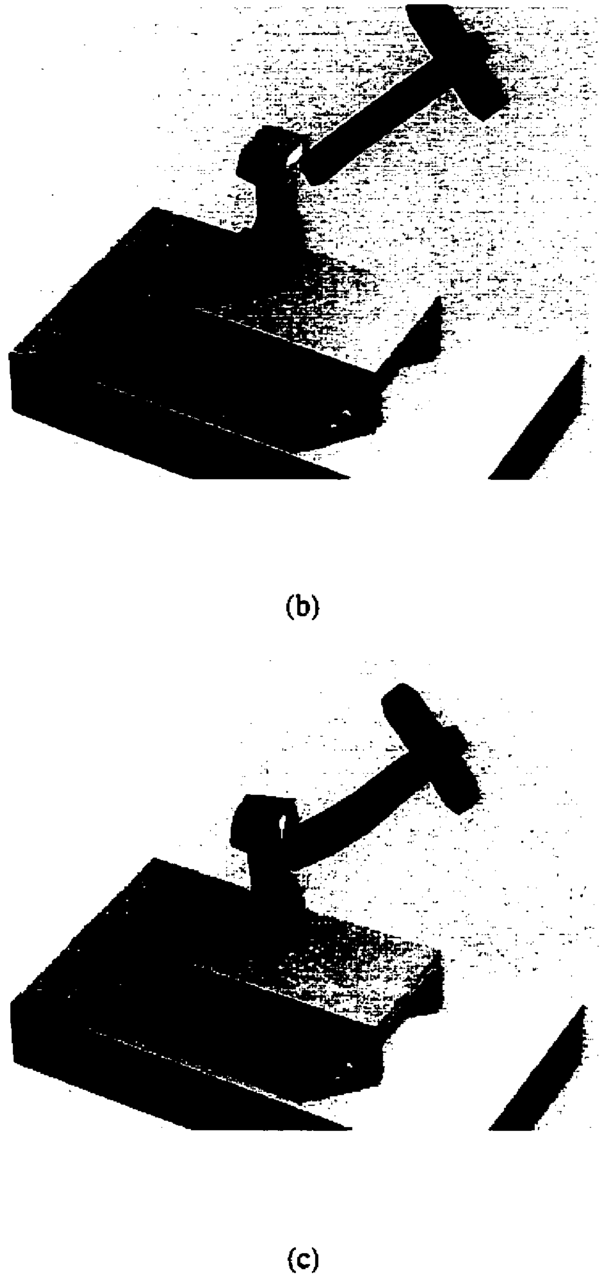
(c)

Figure 5.3: Images illustrating the collision between a flexible hammer and a flexible table.



(a)

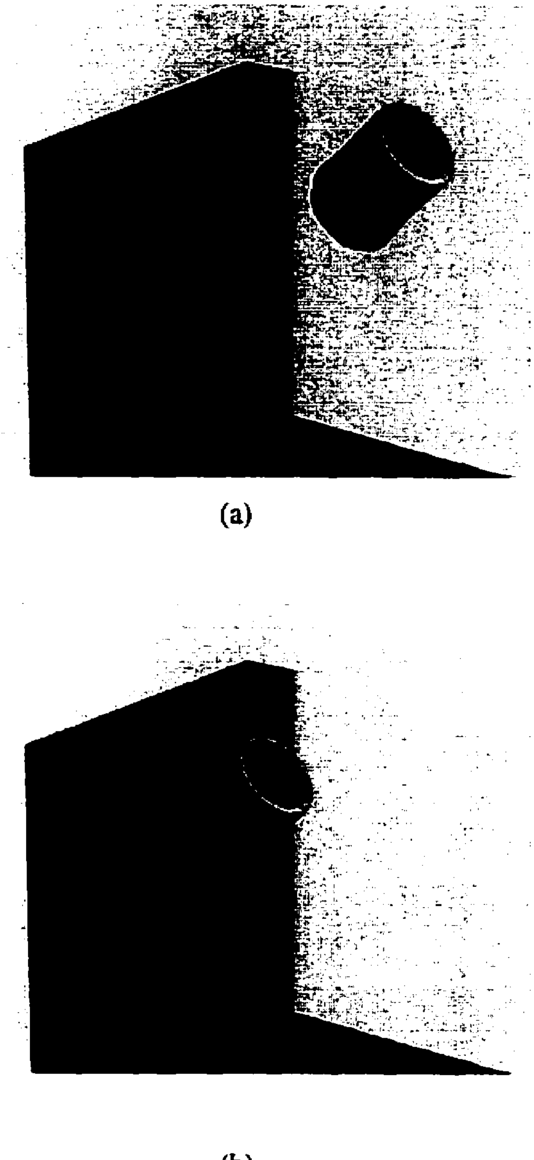
Figure 5.4: Images illustrating the collision between a flexible hammer and a flexible lamp on a table. (nodes: 156, elements: 180, time cost of collision response analysis: 3.4 sec/frame)



(b)

(c)

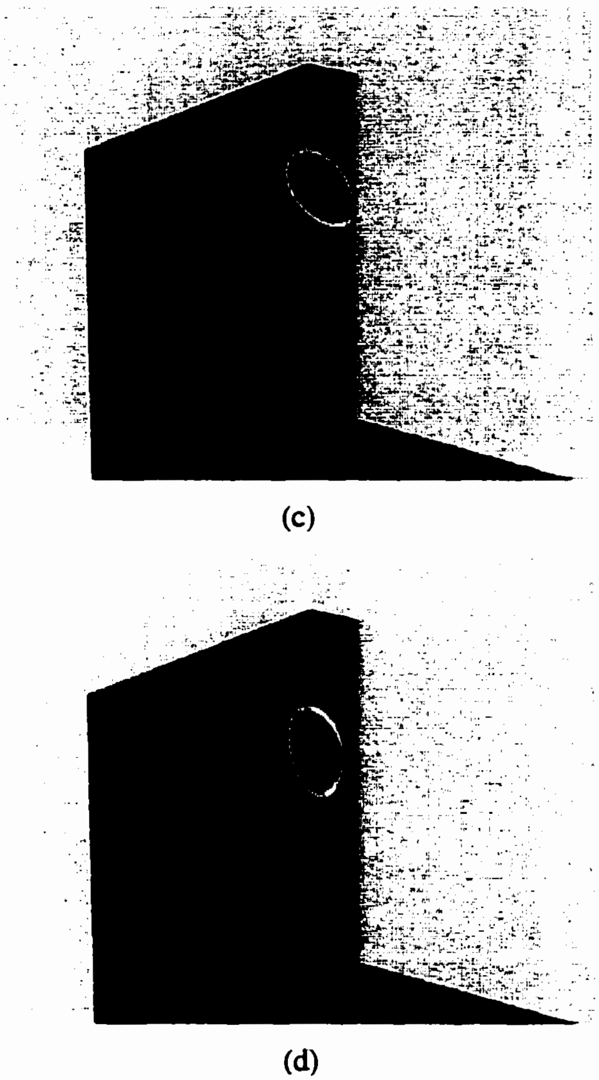
Figure 5.4: Images illustrating the collision between a flexible hammer and a flexible lamp on a table. (nodes: 156, elements: 180, time cost of collision response analysis: 3.4 sec/frame)



(a)

(b)

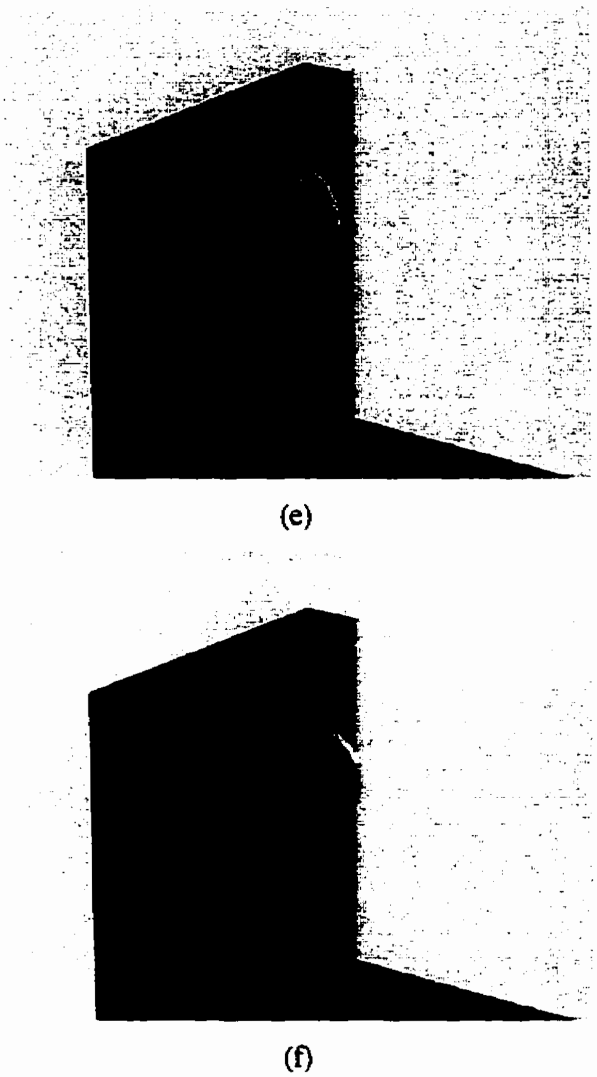
Figure 5.5: Images illustrating oblique collision of an elastic cup with a rigid wall. (nodes: 341, elements: 923, time cost of collision response analysis: 8.6 sec/frame)



(c)

(d)

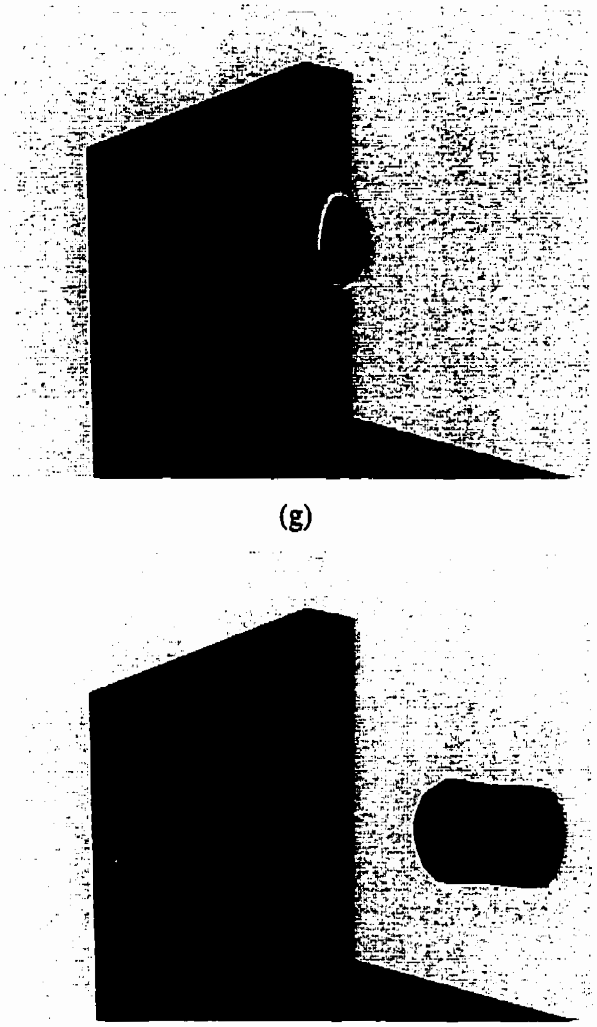
Figure 5.5: Images illustrating oblique collision of an elastic cup with a rigid wall. (nodes: 341. elements: 923, time cost collision response analysis: 8.6 sec/frame)



(e)

(f)

Figure 5.5: Images illustrating oblique collision of an elastic cup with a rigid wall. (nodes: 341. elements: 923. time cost of collision response analysis: 8.6 sec/frame)



(g)

(h)

Figure 5.5: Images illustrating oblique collision of an elastic cup with a rigid wall. (nodes: 341, elements: 923, time cost of collision response analysis: 8.6 sec/frame)

5.6.2 Quantitative Evaluation

The experiments to extensively test the algorithm described in this chapter are presented in this section. The numerical accuracy is obtained by comparing the simulation result with the theoretical solution. For dynamic interactions between objects, the aspects which are evaluated include a) the time length of the collision period; b) momentum conservation; c) energy conservation; and d) the impulse-momentum relationship.

It is difficult to find a theoretical solution for the collision between complex-shaped objects. One possibility is to choose some simple cases where theoretical solutions exist in solid mechanics. Let us consider a classic collision example in solid mechanics shown in Figure 5.6 in which each of the two prismatic rods is divided into ten equal length elements.

5.6.2.1 Length of Collision Period

The time length of a collision process is one of the most important factors which needs to be evaluated, because it directly affects how large the contact force could be if the contact impulse is given. Consequently, it influences the accuracy in predicting whether or not a falling object will be broken by comparing the stress caused by the contact force with the tensile strength of the material. If the time length of collision could not be predicted accurately, an algorithm could still give a correct prediction about the movement of the colliding object after the collision by using the principle of momentum conservation. But the deformation of the colliding objects can not be predicted in a physically correct way and one potential problem is in predicting the separation of two objects sticking together.

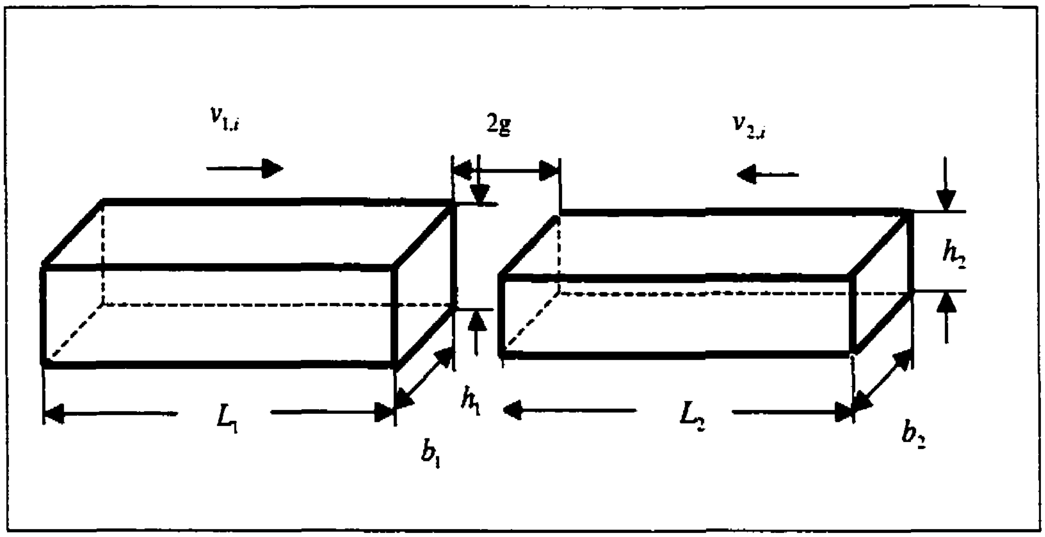


Figure 5.6: Impact problem between two prismatic rods. ($L_1 = L_2 = 0.254$ m, $b_1 = b_2 = 0.0254$ m, $h_1 = 0.0254$ m, $h_2 = 0.0224$ m, $g = 2.54\times10^{-4}$ m, $v_{1,i} = 5.1359$ m/s, $v_{2,i} = -5.1359$ m/s, $\rho = 7844$ kg/m^3, $E = 206.82\times10^9$ N/m^2. This is a typical setting of material parameters which can be found in Reference [95])

Figure 5.7 shows the numerical results of simulation of the problem in Figure 5.6 using the algorithm introduced in this chapter. The theoretical solution for the length of the collision period is [18]

$$t_{collision} = 2L\sqrt{\frac{\rho}{E}}, \tag{5.27}$$

where L is the length of the rod. ρ and E are the density and the elastic modulus, respectively. The theoretical time length of collision, according to solid mechanics, is

9.9×10^{-5} sec, while the numerical time length is 10.125×10^{-5} sec which covers the range of non-zero impact force in Figure 5.7. The relative error is 2.27%.

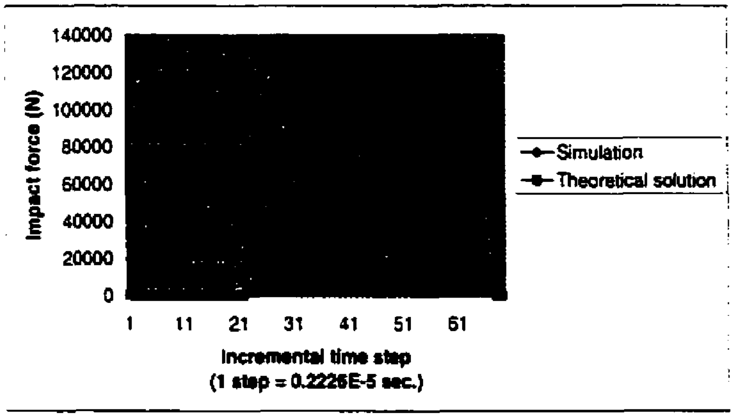


Figure 5.7: Impact-force relationship for the problem in Figure 5.6.

5.6.2.2 Momentum Conservation

The conservation of momentum means that the total momentum of the colliding objects is kept unchanged after a collision. For the case shown in Figure 5.6, we have

$$m_1\dot{\mathbf{u}}_{1,i} + m_2\dot{\mathbf{u}}_{2,i} = m_1\dot{\mathbf{u}}_{1,f} + m_2\dot{\mathbf{u}}_{2,f}, \tag{5.28}$$

where m_1 and m_2 are the mass of rods 1 and 2, respectively. $\dot{\mathbf{u}}_{1,i}$ and $\dot{\mathbf{u}}_{2,i}$ are the velocity of rods 1 and 2 before the collision, $\dot{\mathbf{u}}_{1,f}$ and $\dot{\mathbf{u}}_{2,f}$ are the velocities after the collision. The numerical result is expressed by

$$\begin{aligned} m_1\dot{\mathbf{u}}_{1,i} + m_2\dot{\mathbf{u}}_{2,i} &= 1.2854\times 5.1359 + 0.9818\times(-5.1359) = 1.5595, \\ m_1\dot{\mathbf{u}}_{1,f} + m_2\dot{\mathbf{u}}_{2,f} &= 1.2854\times(-3.6870) + 0.9818\times 6.3936 = 1.5378. \end{aligned} \tag{5.29}$$

The relative error for the momentum conservation is 1.39%. One intuitive check is to consider two colliding objects with different mass and opposite directions of velocity. If the magnitude of velocity of the objects before the collision is the same, then the lighter object will obtain a higher speed than the heavier object after the collision. The numerical result reflects this tendency by the fact that $|v_{2,f}|$ (= 6.3936) is larger than $|v_{1,f}|$ (= 3.6870).

5.6.2.3 Energy Conservation

Energy exists in many forms such as mechanical, thermal, chemical, and nuclear energy. In this thesis, only mechanical energy is considered which includes kinematic and potential energy as well as dissipative frictional energy. In elastic collisions both momentum and mechanical energy are conserved, while in inelastic collisions only momentum is conserved. For the case shown in Figure 5.6, the energy conservation means

$$\frac{1}{2}m_1\dot{\mathbf{u}}_{1,i}^2 + \frac{1}{2}m_2\dot{\mathbf{u}}_{2,i}^2 = \frac{1}{2}m_1\dot{\mathbf{u}}_{1,f}^2 + \frac{1}{2}m_2\dot{\mathbf{u}}_{2,f}^2. \tag{5.30}$$

The numerical result gives

$$\begin{aligned}\frac{1}{2}m_1\dot{\mathbf{u}}_{1,i}^2+\frac{1}{2}m_2\dot{\mathbf{u}}_{2,i}^2&=0.5\times\left[1.2854\times5.1359^2+0.9818\times5.1359^2\right]=29.9010,\\ \frac{1}{2}m_1\dot{\mathbf{u}}_{1,f}^2+\frac{1}{2}m_2\dot{\mathbf{u}}_{2,f}^2&=0.5\times\left[1.2854\times3.67^2+0.9818\times6.3936^2\right]=28.8029.\end{aligned}\tag{5.31}$$

Equation (5.31) indicates that the relative error for energy conservation is 3.67%.

5.6.2.4 Impulse-Momentum Equality

The impulse-momentum relationship states that the impulse in a collision process equals the change in the momentum of the colliding objects. For the case shown in Figure 5.6,

$$\int_0^{t_{collision}} F\,dt=-m_1(\dot{\mathbf{u}}_{1,f}-\dot{\mathbf{u}}_{1,i})=m_2(\dot{\mathbf{u}}_{2,f}-\dot{\mathbf{u}}_{2,i})\,. \tag{5.32}$$

The integration of the calculated impact force (diamond-symbol line) in Figure 5.7 gives $\int_0^{t_{collision}} F\,dt=11.075\ N\cdot s$, while the change in the momentum of the two rods is

$$\begin{aligned}m_1(\dot{\mathbf{u}}_{1,f}-\dot{\mathbf{u}}_{1,i})&=1.2854\times(-3.6870-5.1359)=-11.3409,\\ m_2(\dot{\mathbf{u}}_{2,f}-\dot{\mathbf{u}}_{2,i})&=0.9818\times(6.3936+5.1359)=11.3192.\end{aligned}\tag{5.33}$$

Therefore, the relative errors for rods 1 and 2 are 2.35% and 2.16%, respectively.

5.6.2.5 Convergence of Gauss-Seidel Iteration for Contact Forces

In each time step, an iteration is needed by most algorithms to correct the calculation of the contact forces with respect to enforcing the geometric and loading constraints. Figure 5.8 gives an example of convergence when the friction capacity is not reached. In such a case, only the geometric constraint needs to be considered. The figure shows that the calculated contact forces converge quickly and at most three iterations are required to achieve a reasonable accuracy (error < 0.01%).

If the friction capacity is reached during a time step, then the tangential contact force needs to be adjusted according to the chosen friction law, i.e., the loading constraint also needs to be considered. Figure 5.9 illustrates the convergence of the algorithm in the time step where the loading constraint is required to take into account. Similarly, three iterations are sufficient for the convergence with an error of less than 0.01 %.

5.7 Comparisons with Other Methods

The semi-explicit local collision response analysis approach and other existing methods are compared as follows.

Complexity analysis indicates that the proposed semi-explicit local collision response analysis algorithm is of O(n) time complexity, where n is the total number of domain nodes. This makes the algorithm faster than most existing ones [4,26,48,69,70,71] if the same number of domain nodes are used for comparison. The O(n) nature makes this algorithm preferable in the simulation of complex systems when the total number of domain nodes is very large.

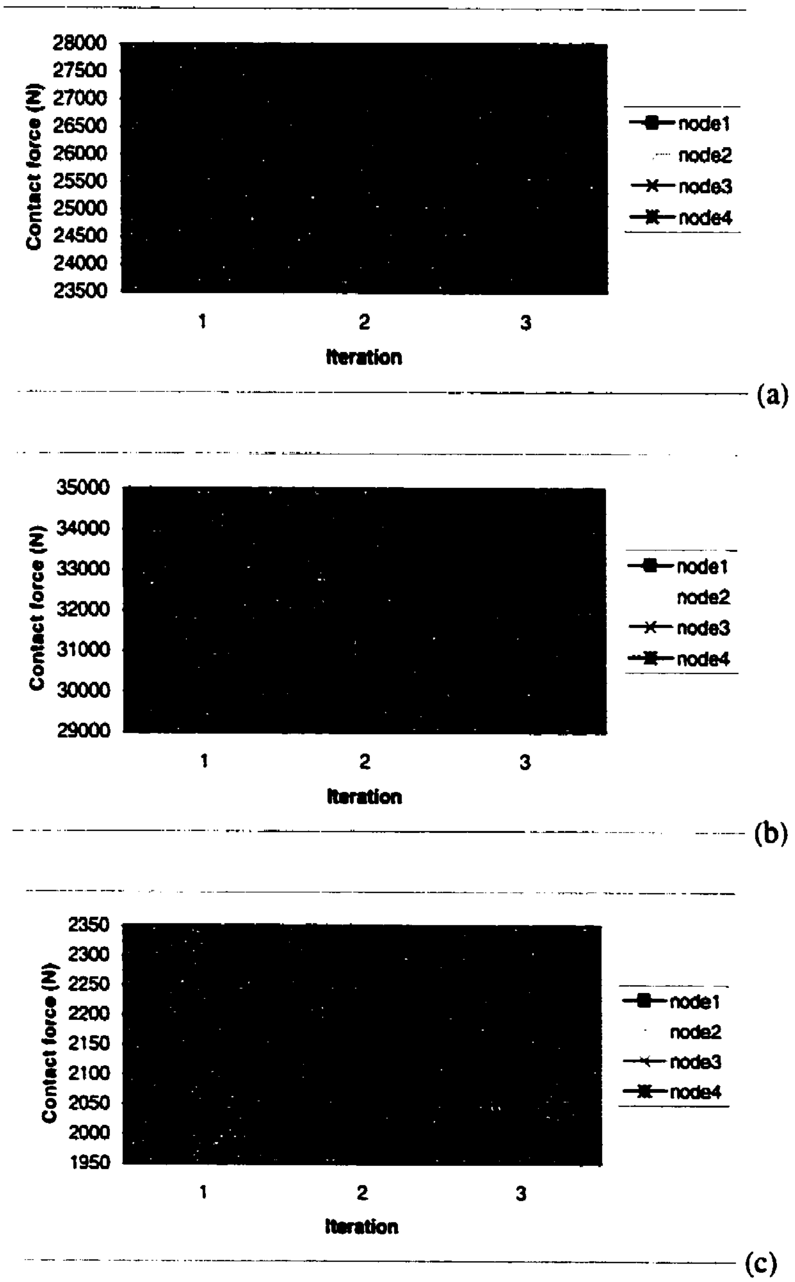


Figure 5.8: The convergence of contact forces when frictional forces do not exceed the friction capacity. (a) beginning of the collision period; (b) middle of the collision period; (c) end of the collision period.

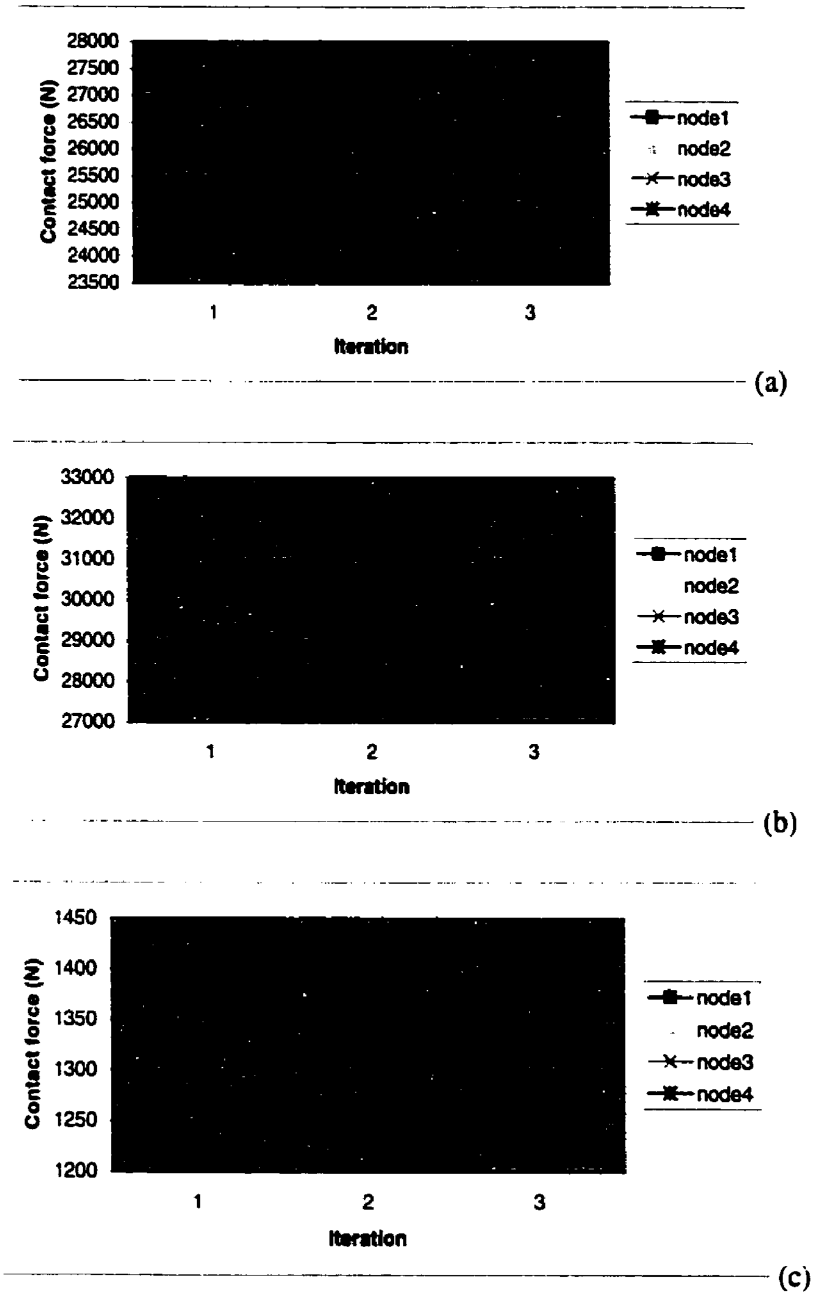


Figure 5.9: The convergence of contact forces when frictional forces exceed the friction capacity. (a) beginning of the collision period; (b) middle of the collision period; (c) end of the collision period.

One major difference between the semi-explicit local collision response analysis approach and existing methods is that the semi-explicit local collision response analysis approach is able to give an accurate prediction about the time length of the collision process. Without the correct calculation of this information, it is very difficult, if not impossible, to correctly estimate the magnitude of impact force for a given collision. There is no experimental evaluation of the calculated time length of collision in the methods proposed by Cani-Gascuel and Desbrun [15], Baraff and Witkin [5].

Another benefit of the semi-explicit local collision response analysis approach is that no major assumption is made except in using the well-known d'Alambert principle for the motion equation of the finite element method and Lagrangian Dynamics. In the method proposed by Cani-Gascuel and Desbrun [15], three different regions (interpenetration, propagation and the remaining) need to be set up for each colliding object. No clear rule is introduced as to how to partition an object into these different regions. Even though some appealing visual animations are generated with this assumption, the generality of the approach is in doubt. For instance, the approach is not well-suited to the case when the colliding objects have non-uniform stiffness, especially when the portion in the interpenetration region happens to be much stiffer than the remaining region, which leads to no deformation in such a case. In addition, since the different deformation fields are applied in the interpenetration and propagation regions, the ability of the approach to predict the correct wave movement inside the colliding objects during a collision is in doubt.

In the method proposed by Baraff and Witkin [5] and Witkin and Welch [77] a concept of the global deformation that applies a parametric "space warp" to all points in the body is used. This avoids the increase in the stiffness of system motion equations due to local interactions, but also constrains its ability to generate a realistic local contact surface during a collision.

The approaches developed by Pentland and Williams [56] and Faloutsos et al. [29] actually do not address the problem of collision response. A sort of artistic or heuristic representation is used to describe the behavior of deformable objects after a collision. The semi-explicit local collision response analysis approach focuses on the other end of the spectrum, namely, realistic physical representation of the collision response.

The boundary element method approach used by James and Pai [39] is a real time solution to static elastic problems. However, it does not possess the capability of handling the dynamic collision response analysis and fracture simulation. Besides, the singular integrals in the BEM impose numerical difficulties. This thesis work focuses on a general-purpose solution which is also computationally efficient by using a new variation of the conventional finite element method.

It is well known that the addition of penalty in the penalty method will always decrease the stable time step while the addition of Lagrange multipliers in the Lagrange multiplier method does not [10]. The semi-explicit local collision response analysis algorithm avoids the stability problem caused by the penalty. In addition, the semi-explicit scheme used in this thesis is different from the pure explicit time integration, because a Gauss-Seidel iteration is used to calculate the displacement unknowns at time step n+1 by using the updated geometric configuration at time step n+1 instead of time step n. Therefore, the semi-explicit local collision response analysis algorithm avoids the stability problem associated with the pure explicit time integration.

The differences between the proposed approach with some engineering commercial packages include:

- Stiffness matrix is always considered as constant to reduce computational cost. This is not acceptable in engineering analysis.

- Simplest 4-node tetra elements are used in favor of computational efficiency. This is likely not accurate enough in engineering analysis.

- Collision detection scheme used in this thesis is more general than that used in engineering analysis.

- The treatment of finite rotation is also more general than that used in engineering analysis.

Chapter 6

Fracture Simulation

Fracture can be caused by the propagation of cracks in the continuum. In this chapter, a new fracture simulation approach is introduced.

6.1 Concept of Stresses and Principal Stresses

Fig. 6.1 shows a three-dimensional state of stresses on an infinitesimal parallelepiped element of a larger continuous body, without counting the variation of stress with position inside this element. These stresses can be described in terms of either a stress tensor $\sigma_{ij}(i, j = x, y, z)$ or a matrix form shown by the following equation [64]:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix}. \qquad (6.1)$$

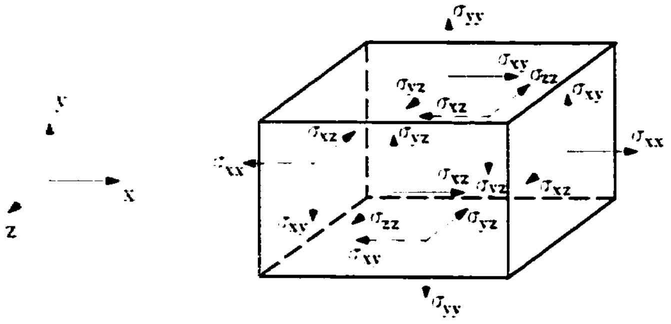


Figure 6.1: A three-dimensional state of stresses.

The stress σ in an original x-y-z coordinate system can be transformed into σ' in another x'-y'-z' system by

$$\boldsymbol{\sigma}' = \mathbf{T}\boldsymbol{\sigma}\mathbf{T}^T, \qquad (6.2)$$

$$\mathbf{T} = \begin{bmatrix} \cos(x,x') & \cos(y,x') & \cos(z,x') \\ \cos(x,y') & \cos(y,y') & \cos(z,y') \\ \cos(x,z') & \cos(y,z') & \cos(z,z') \end{bmatrix}, \qquad (6.3)$$

where **T** is a rotation matrix consisting of direction cosines between two coordinate systems, x-y-z and x'-y'-z'. The angle between two arbitrary coordinates is represented by parentheses enclosing the corresponding coordinate labels separated by a comma.

According to the theory of matrices, a specific set of rotations of coordinates exists such that $\boldsymbol{\sigma}'$ becomes:

$$\boldsymbol{\sigma}' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \tag{6.4}$$

where $\sigma_1, \sigma_2, \sigma_3$ = principal stresses which are the solutions of the following equations:

$$\sigma^3 - I_1\sigma^2 + I_2\sigma - I_3 = 0, \tag{6.5}$$

$$I_1 = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}, \tag{6.6}$$

$$I_2 = \sigma_{xx}\sigma_{yy} + \sigma_{xx}\sigma_{zz} + \sigma_{yy}\sigma_{zz} - \sigma_{xy}^{\;2} - \sigma_{yz}^{\;2} - \sigma_{xz}^{\;2}, \tag{6.7}$$

$$I_3 = \begin{vmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{vmatrix}, \tag{6.8}$$

where I_1, I_2 and I_3 are called the first, second and third invariants of the stress, respectively.

The principal stresses and their directions can be determined by solving the following eigenvalue problem:

$$(\boldsymbol{\sigma} - \omega \mathbf{I})\mathbf{u} = 0, \qquad (6.9)$$

where ω and $\mathbf{u}$ are the eigenvalues and eigenvectors, respectively. **I** is the unity matrix.

6.2 Failure Criteria of Material

A failure criterion $f(\sigma_{ij})$of a certain type of material is a function of stress, strain, and other parameters such that when $f(\sigma_{ij}) < k_0$ material failure does not occur and when $f(\sigma_{ij}) = k_0$, the material is in a failure state where its plastic deformation becomes unlimited. Here, k_0 is a failure constant depending on material properties. The failure function $f(\sigma_{ij})$ corresponds to a failure surface in stress space which is the bound or limit to the failure surface.

There are two main types of failure criteria for various types of materials: tensile and shear.

6.2.1 Tensile Failure Criterion

Tensile failure criterion means that tensile stresses are compared to certain threshold to determine if a material fails. One of the most common tensile failure criteria is

$$f(\sigma_{ij}) = \sigma_1 \geq F_t, \qquad (6.10)$$

where σ_1 is the major principal stress inside an object and F_t is the tensile strength of the material. In previous studies on fracture simulation in computer animation, a kind of equivalent nodal tensile strength or string tensile strength was used to simulate the fracture due to tension inside an object. However, tensile failure is only one of the simplest form of failure in reality. In this thesis, a new element-split scheme is proposed to incorporate not only tensile but also shear failure criterion with an aim of being capable in simulating different failure patterns.

6.2.2 Shear-Type Failure Criteria

Shear-type failure criteria mean that shear stresses are checked against certain threshold to determine if a material fails. The existing shear-type yield and failure criteria can be classified into two categories: non-frictional and frictional models. according to whether or not the models take into account the frictional components in the shear strengths.

Non-frictional models

♦ Tresca's yield criterion [36]

This criterion states that plastic strain occurs when the maximum shear stress reaches a certain value *k* (shear strength), as shown in the following equation:

$$f(\sigma_{ij}) = \frac{\mathrm{Max}(|\sigma_1 - \sigma_2|, |\sigma_2 - \sigma_3|, |\sigma_3 - \sigma_1|)}{2}, \tag{6.11}$$

where $f\left(\sigma_{ij}\right)$ is a function of shear stresses, i.e., difference between principal stresses. When its value is greater than or equal to shear strength k, the material fails. Equation (6.11) represents a prism with a hexagonal cross-section, centred on the hydrostatic axis ($\sigma_1 = \sigma_2 = \sigma_3$) in the principal stress space, as shown in Fig. 6.2(a). The terminology 'hydrostatic' originally comes from hydrology. If one object is put under water at a certain depth, the object is subject to a uniform hydrostatic pressure in three Cartesian coordinates. Later, people use this terminology in solid mechanics to represent a special type of stress state in which the principal stresses of the object are equal (i.e., $\sigma_1 = \sigma_2 = \sigma_3$). Graphically, if we construct a principal stress space in which the three orthogonal coordinates are respectively σ_1, σ_2 and σ_3, each point on the hydrostatic axis refers to a certain stress state in which all principal stresses of the object are equal with the magnitude being the distance between the origin and the point on the hydrostatic axis.

♦ von Mises's yield criterion [36]

It states that plastic strain occurs when the maximum strain energy due to shearing equals a critical value k which depends on material properties. This criterion takes into account the contribution of the intermediate principal stress and is more easily handled mathematically as compared with Tresca's criterion. The detailed algebraic form is

$$\begin{aligned} f\,(\sigma_{ij}) &= J_2 = \frac{1}{2} s_{ij} s_{ij} \\ &= \frac{1}{6}[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{xx} - \sigma_{zz})^2] + \sigma_{xy}{}^2 + \sigma_{yz}{}^2 + \sigma_{zx}{}^2. \end{aligned} \tag{6.12}$$

This equation corresponds to a cylindrical surface centred on the hydrostatic axis in principal stress space as illustrated in Fig. 6.2(b).

Frictional models

- Mohr-Coulomb failure criterion

It is based on the following Mohr-Coulomb law [25]:

$$f(\sigma_{ij}) = \tau = c + \sigma \tan\phi . \tag{6.13}$$

where τ is the shear strength and σ is normal stress at the interested point inside the object. c is material cohesion strength and ϕ is material friction angle. Here, cohesion means the material strength without the contribution of internal friction. Equation (6.13) can be transformed into the following form in the three-dimensional stress space by considering $\tau = \frac{\sigma_1 - \sigma_3}{2}$ and $\sigma = \frac{\sigma_1 + \sigma_3}{2}$:

$$\begin{aligned} & f(\sigma_{ij}) = \sigma_1 - \sigma_3 - [\sin\phi(\sigma_1 + \sigma_3) + 2c\cos\phi] \\ & k_0 = 0 \end{aligned} \tag{6.14}$$

where k_0 is the failure threshold (=0) which means that when $f(\sigma_{ij})$ is greater than or equal to 0, the material fails. Equation (6.14) is equivalent to an irregular hexagonal pyramid surface centered on the hydrostatic axis in principal stress space shown in Fig. 6.2(c).

♦ Drucker-Prager's yield criterion

Drucker and Prager [27] use a conical surface to round off the hexagonal pyramid surface for the mathematical convenience as shown in Fig. 6.2(d), and propose using the following modified form of the Mohr-Coulomb law that takes into account all the principal stresses:

$$\begin{aligned} f(\sigma_{ij}) &= \alpha I_1 + J_2^{1/2} \\ &= \alpha\sigma_{ii} + \left(\frac{1}{2} s_{ij} s_{ij}\right)^{1/2}. \end{aligned} \tag{6.15}$$

where

$$I_1 = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}. \tag{6.16}$$

Frictional models are more general than the non-frictional ones because they include the frictional component. In this thesis, Tresca's criterion and Mohr-Coulomb criterion are adopted to show the difference between shear and tensile failure criteria. Due to the flexibility of the element-split scheme, which will be described in the next section, any material failure criterion written in element stress format, can be easily incorporated into the fracture simulation. This is a crucial advantage of the element-split approach in this thesis over previous work in computer graphics.

Figure 6.2 is a graphical representation of different failure criteria in principal stress space. The surfaces in this figure are essentially equivalent to the mathematical representation in Equations (6.11) through (6.16).

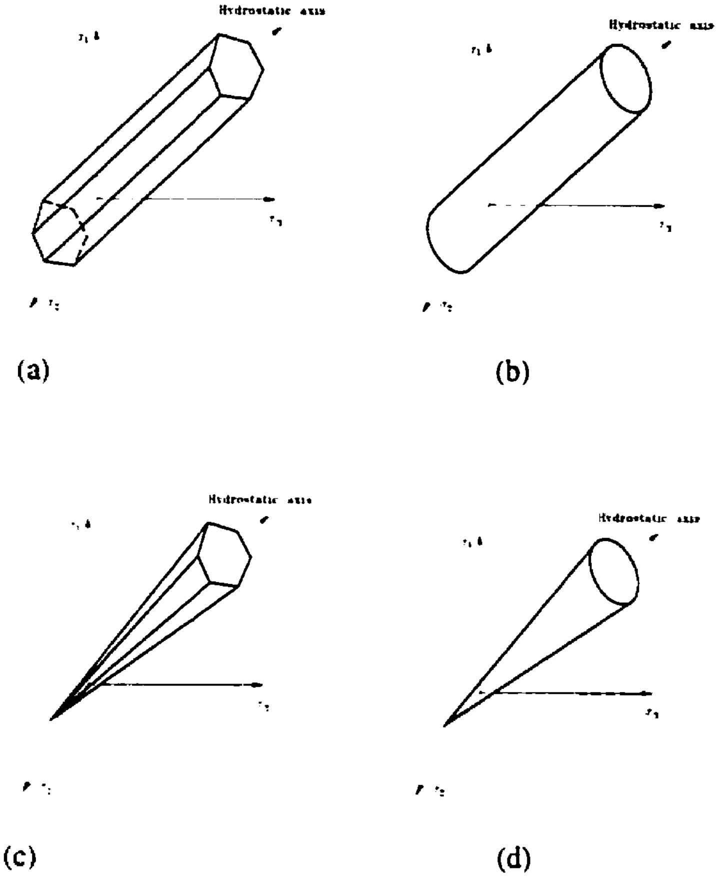


Figure 6.2: Graphical representation of shear-type yield and failure. (a) Tresca's yield surface; (b) von Mises's yield surface; (c) Mohr-Coulomb failure surface; (d) Drucker-Prager's yield surface.

6.3 Element-Split Scheme

Even though the node-split scheme of O'Brien and Hodgins [54] is an advance from previous string-split schemes [53,68,70] in which the distance between two adjacent nodes is compared with a threshold to determine when a failure occurs between these two nodes, it has a serious limitation on the failure criterion of material, in particular, only tensile failure criterion is allowed in their approach.

By considering that most existing material failure criteria are expressed in the format of element stress, a new element-split scheme is proposed on the basis of the author's original approach [65] to overcome the shortcomings of previous schemes.

6.3.1 Geometric Representation of Objects

Theoretically, no geometric information is needed for the finite element simulation of fracture except the information of volume mesh which divides the objects of interest into a finite number of elements such as tetrahedra or hexahedra. The volume mesh can be produced by using commercial automatic mesher [73] or existing meshing algorithm such as the advanced-front method [45].

In this thesis work, a four-node tetrahedron element is adopted as the basic unit for the structure of objects because of its simplicity in element stress and strain calculation. Actually, the stress and strain are constant within each tetrahedron element such that their calculation is needed only once for each element. Some parts of the objects may be easily represented by hexahedron elements. In such a case, these hexahedron elements are internally converted to tetrahedron elements in a way as illustrated in Figure 6.3. It should be noted that the subdivision is not unique.

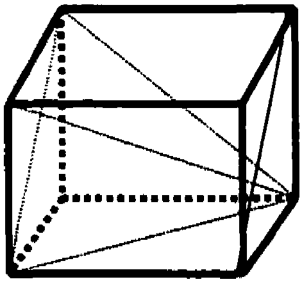


Figure 6.3: Subdivision of a hexahedron into six tetrahedra.

6.3.2 Normal of Element Cutting Plane

Since the four-node tetrahedron element is used in this thesis work for the sake of efficiency, the stress and strain are constant within each element. Therefore, it is reasonable to assume that the normal of a potential split surface within each element is constant, i.e., a single element cutting plane is used to approximate the real split surface.

In the case of tensile failure criterion, the normal of the element cutting plane is in the direction of major principal stress within the element, while in the case of shear failure criterion, the normal of the element cutting plane is between the directions of the major and minor principal stresses. To better understand the shear failure case, let's consider the equilibrium of a micro rectangular block which is subject to a vertical and horizontal force F_1 and F_3, respectively, as shown in Figure 6.4.

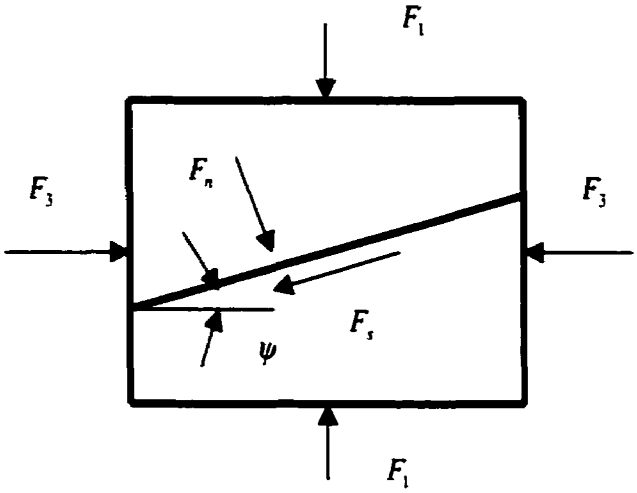


Figure 6.4: Equilibrium of a micro rectangular block within an object.

The interface force F_n and F_s in Figure 6.4 represent the forces exerting on the lower part of the block. The equilibrium of this rectangular block is given by:

$$F_n = F_1 \cos\psi + F_3 \sin\psi \,. \tag{6.17}$$

$$F_s = F_1 \sin\psi - F_3 \cos\psi \,, \tag{6.18}$$

where F_n and F_s are the normal and shear forces on an inclined plane within the block. On the basis of equation (6.17), normal stress on the inclined plane can be expressed as

$$\sigma_n = \frac{F_n}{A} = \frac{F_1 \cos\psi}{A_1 / \cos\psi} + \frac{F_3 \sin\psi}{A_3 / \sin\psi} = \sigma_1 \cos^2\psi + \sigma_3 \sin^2\psi , \tag{6.19}$$

where A_1 and A_3 are the surface area on which F_1 and F_3 are applied, respectively. A is the area of the inclined plane. σ_1 and σ_3 refer to the major and minor principal stresses, respectively. By considering the following double angle identities

$$\cos 2\psi = \frac{1 + \cos 2\psi}{2} , \tag{6.20}$$

$$\sin 2\psi = \frac{\sin 2\psi}{2} , \tag{6.21}$$

equation (6.19) can be rewritten as

$$\sigma_n = \frac{\sigma_1 + \sigma_3}{2} + \frac{\sigma_1 - \sigma_3}{2} \cos 2\psi , \tag{6.22}$$

where $\frac{\sigma_1 + \sigma_3}{2}$ and $\frac{\sigma_1 - \sigma_3}{2}$ are the mean and deviatoric stresses, respectively. ψ is the angle between the major principal stress and the normal to the inclined plane.

The shear stress acting on the inclined plane is expressed as

$$\sigma_s = \frac{F_s}{A} = \frac{F_1 \sin\psi}{A_1 / \cos\psi} - \frac{F_3 \cos\psi}{A_3 / \sin\psi} = (\sigma_1 - \sigma_3) \sin\psi \cos\psi . \tag{6.23}$$

Using $\sin\psi\cos\psi = \sin 2\psi / 2$, we have

$$\sigma_s = \frac{(\sigma_1 - \sigma_3)\sin 2\psi}{2}. \qquad (6.24)$$

From equation (6.24), it can be concluded that shear stress, σ_s, reaches its maximum where $\psi = 45^\circ$, i.e., the normal of the inclined plane is 45° from the direction of major principal stress. This defines the orientation of the element cutting plane in the case of shear failure criterion. Currently, only isotropic material is considered. But the scheme in this thesis can be easily extended to anisotropic material.

6.3.3 Element Split Scheme

After the normal of the element cutting plane is determined, ideally a cutting plane equation can be constructed by letting the plane pass through the centroid of the element. However, this method causes the problem in dividing the two split parts of the original element into two sets of tetrahedron elements, because the orientation of the cutting plane and geometric configuration of the element could be arbitrary leading to a variety of subcases to be considered.

To simplify the problem, the following scheme is used to approximate the real cutting plane within each element:

(1) Construct a cutting plane which passes through the centroid of the element with its normal being the direction of major principal stress in the case of tensile failure criterion or the direction between the major and minor principal stresses in the case of shear failure criterion.

(2) Calculate the distance of the four nodes of the tetrahedron element to the cutting plane constructed in step (1) and identify node *m* which has the minimum value in terms of the distance to the cutting plane.

(3) Translate the cutting plane such that it passes through node *m* identified in step (2). The resulting plane is an approximate cutting plane which leads to a limited number of split patterns of any arbitrary tetrahedron element and cutting plane, as enumerated in Figure 6.5.

(4) Calculate the intersection between the cutting plane and the other three element edges which do not have node *m* as their edge node.

(5) Create new nodes at the edge intersection and split the original element into two sets. One set is just a tetrahedron, while the second set consists of two tetrahedra, as shown in Figure 6.5(b) and (c).

Beside the general case shown in Figure 6.5, there are two special cases in which the approximate cutting plane passes through two or three nodes of the element, leading to two or one split tetrahedron element, respectively.

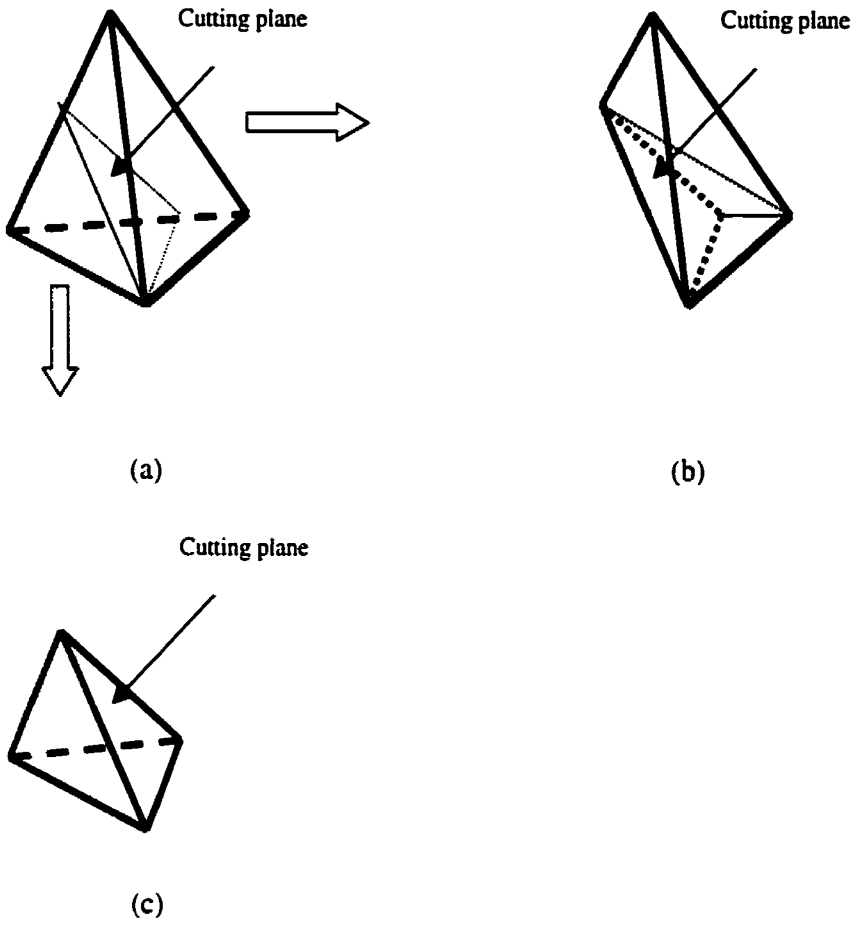


Figure 6.5: Split of an arbitrary tetrahedron element by an approximate cutting plane. (a) original element which is split into two parts represented by (b) and (c); (b) upper right part of the original element divided by the cutting plane; (c) lower left part of the original element divided by the cutting plane.

6.3.4 Fracture Propagation

Fracture propagation means a process by which fracture cracks traverse within an object simultaneously or sequentially. One basic precondition for fracture propagation is that two adjacent elements have reached their material failure criterion. Usually, the normals of the cutting planes of two adjacent elements are not in the same direction. This makes the geometric handling of fracture propagation a difficult task.

In order to simplify the fracture propagation process, the following averaging element-cutting-plane-normal scheme is proposed.

(1) Loop over each element and calculate stress information within each element. Check the material failure criterion to determine the status of element failure.

(2) For each failed element, construct an approximate cutting plane as explained in Section 6.3.3 and associate the plane with the corresponding node *m* as defined in Section 6.3.3.

(3) Loop over every node and establish node-element relationship and node-fracture-element relationship which specify which element or failed element is associated with the current node.

(4) For each node which is associated with two or more failed elements, calculate the average of normals of the cutting planes of these failed elements. Create a new cutting plane which passes through the current node with the averaged normal.

(5) Use the new cutting plane to cut any element which possesses the current node and has a valid intersection with the new cutting plane calculated in step (4). A valid intersection means that the cutting plane has an intersection with at least one element edge (not counting the intersection with the end nodes of the edge), as shown in Figure 6.5(a).

(6) Use the new cutting plane to arrange any element which possesses the current node and does not have a valid intersection with the new cutting plane. Here, the word "arrange" means a possible replacement of the nodal connection of the element by replacing the current node with a newly-generated mirror node which is at a short distance from the

current node, as illustrated in Figure 6.6. The detailed procedures of such type of arrangement are as follows:

- On the new approximate cutting plane constructed in step (2), construct a vector in the surface normal direction at the location of the current node. The length of the vector is very small and a new mirror node is created at the other end point of this vector.
- Use the new cutting plane as a boundary, loop over all elements which possess the current node and do not have a valid intersection with the new cutting plane. If the centroid of an element is at the same side as the new mirror node w.r.t. the cutting plane, then the current node in the nodal connection of the element is replaced by the new mirror node, as shown by element *E2* in Figure 6.6(b). Otherwise, the nodal connection of the element is kept intact, as illustrated by element *E1* in Figure 6.6(b).

6.4 Numerical Experiments

The element-split approach is implemented using MS Visual C++ 5.0 under Windows NT 4.0. The effectiveness of the approach is verified both qualitatively and quantitatively below.

6.4.1 Quantitative Evaluation

The evaluation of fracture propagation is generally difficult because very few analytical solutions are available to describe an entire fracture propagation process. A partial evaluation is to use the formula in material mechanics for the estimation of the place where the fracture of an object is likely to happen first. Such evaluation can be accomplished only in the cases where the analytical solution of stresses inside the object is available, because various types of failure criteria can be determined on the basis of stress information.

Test Case 1: Fracture of a bent beam

Figure 6.7 shows fracture propagation in a bent beam. The beam is fixed at its left end and an incremental vertical load is applied downwards at its right end. Since there is a U-shaped cut at the middle of the beam, high stress is likely to occur in the area around the cut. Furthermore, the stress at the left corner of the cut should be higher than that at the right corner, because the larger bending moment exists at the left corner. The geometric configuration of the beam is shown in Figure 6.8. On the basis of equation (57) in [74], the normal stress in the cross section of the beam is calculated by

$$\sigma_x = \frac{M\, y}{I_z}, \tag{6.25}$$

where σ_x is the normal stress in the cross section. M is the bending moment and I_z is the area moment of inertia with respect to the neutral axis Z where the bending stress is zero. For the rectangular beam, I_z, is given by

$$I_z = \frac{bh^3}{12}, \tag{6.26}$$

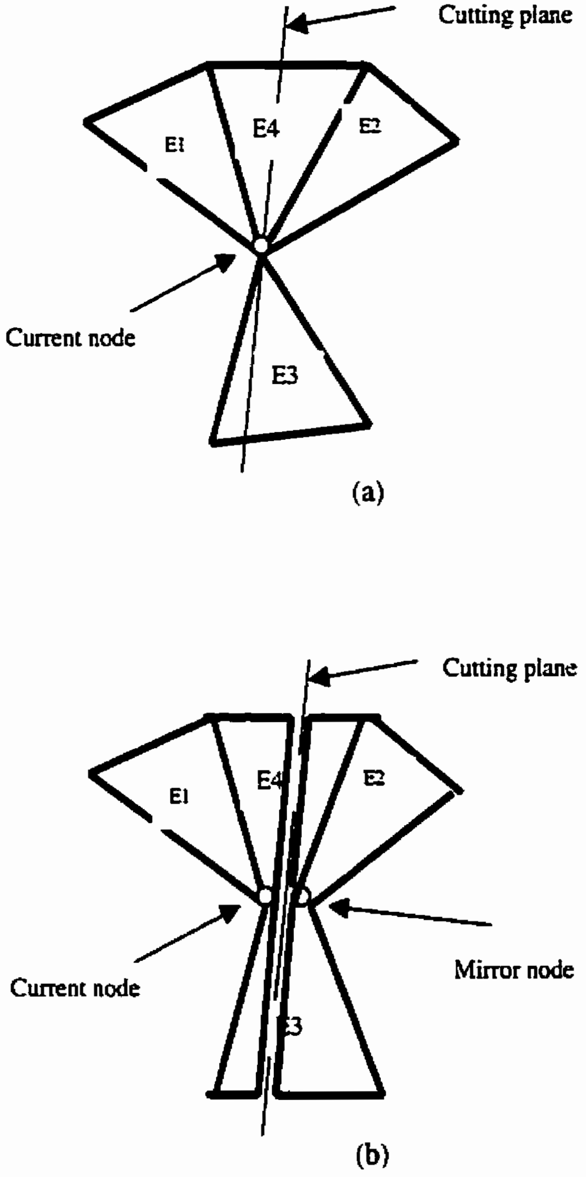


Figure 6.6: Arrangement of elements to either current node or its mirror node. (a) element connection before split and arrangement; (b) element connection after split and arrangement.

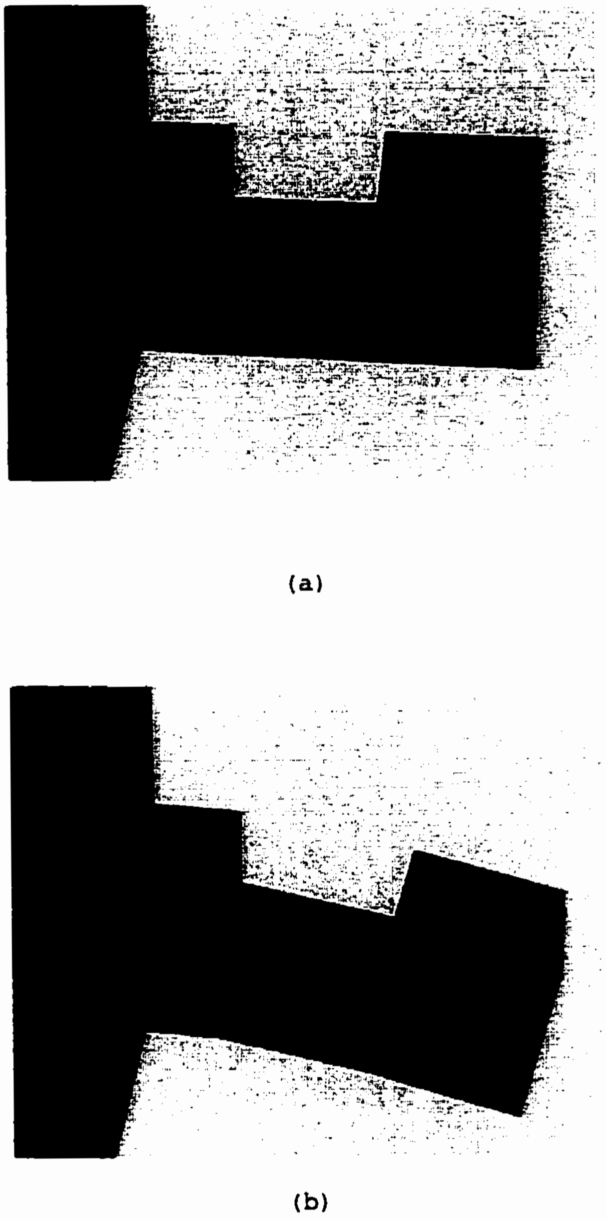


Figure 6.7: Frames of images illustrating the fracture of a beam bent from its left-hand side with tensile failure criterion. (a) initial state; (b) occurrence of cracks. (nodes: 24, elements: 25, time cost: 0.42 sec/frame)

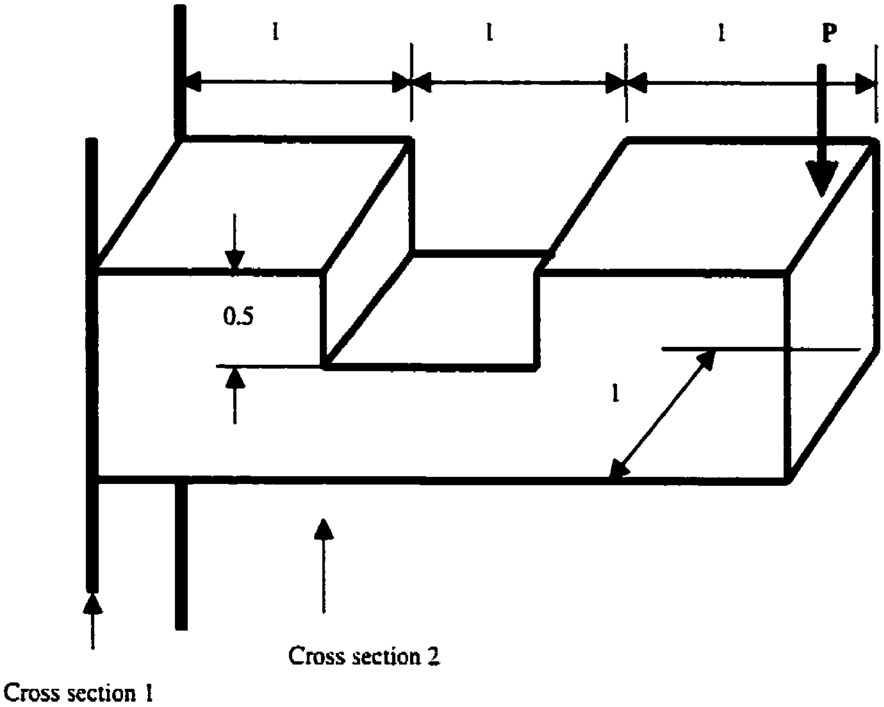


Figure 6.8: Geometric size of the test beam (unit = 1 meter).

where b and h are the width and height of the beam, respectively. The maximum normal stress in cross section 1 in Figure 6.8 is

$$(\sigma_x)_{\max} = \frac{M\ y_{\max}}{I_z} = \frac{(3P)(0.5)}{\frac{1\times 1^3}{12}} = 18P, \tag{6.27}$$

where P is a vertical external load exerting at the right end of the beam, as shown in Figure 6.8. As to cross section 2, the maximum normal stress is

$$(\sigma_x)_{\max} = \frac{M\ y_{\max}}{I_z} = \frac{(2P)(0.25)}{\frac{1\times 0.5^3}{12}} = 48P. \qquad (6.28)$$

Since the maximum normal stress in cross section 2 is larger than that in cross section 1, fracture should happen first in cross section 2 if the material property is considered uniform inside the beam. This is supported by Figure 6.7 (b). Therefore, it is reasonable for cracks to be generated first around the left corner.

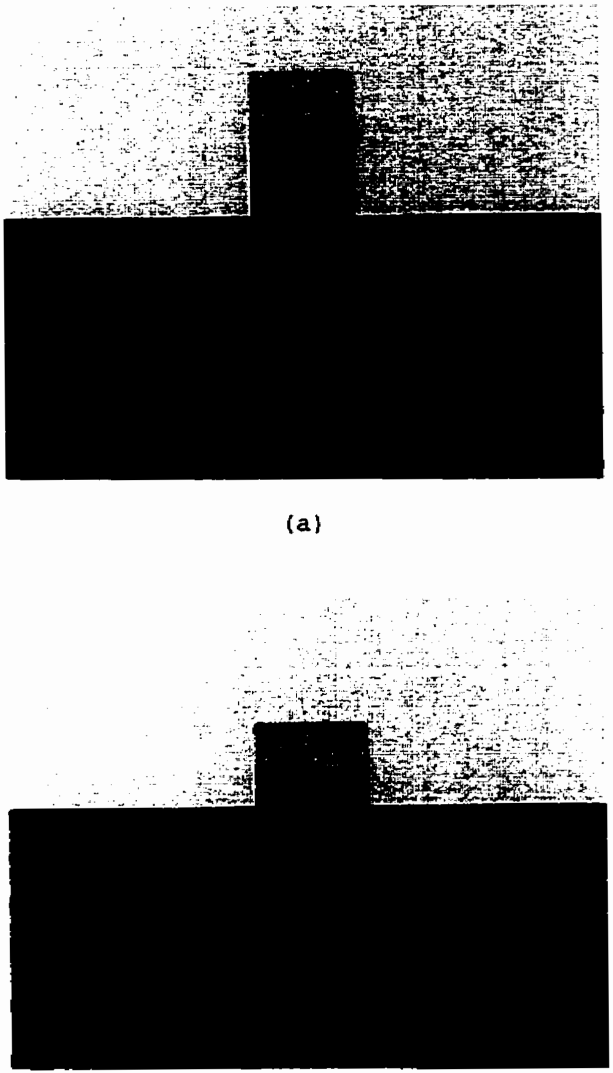
Test Case 2: Fracture of a compressed rectangular pile

A rectangular pile is put on a flat surface and is compressed from its top surface, as shown in Figures 6.9 in which the results by using tensile and Mohr-Coulomb shear failure criteria are presented. The tensile and shear failure patterns are quite different. With the tensile failure criterion, the normal of crack surface has a small angle with the vertical axis of the pile, as shown in Figure 6.9(b), while in the case of the shear failure criterion, the normal of crack surface has an approximate 45 degrees with the vertical axis of the pile, as illustrated in Figure 6.9(c). It is interesting to note that the shear failure pattern provided by the numerical simulation is consistent with the commonly-encountered failure patterns of test specimens in unconfined triaxial tests in civil engineering [40].

According to Karafiath [40], a failure state exists at a point in a material if at any plane through that point the shear stress equals the shear strength. A graphical construction, Mohr's circle representation, can be used to show the orientation of the failure plane inside a test specimen, as in Figure 6.10. The Mohr's circle is an intuitive representation of stress as well as material failure conditions, especially in 2D cases. Let us consider stresses first. The stresses at a point vary in different orientations just like the stress matrix in equation (6.1) is mathematically rotated by using equation (6.2). We can use a circle to fully represent the stresses at the point in different orientation. This circle is called the Mohr's circle with the horizontal axis as normal stress and the vertical axis as shear stress. Material failure criteria, such as the Mohr-Coulomb criterion, are usually obtained by a series of tests in which the normal stress is applied at different magnitudes and accordingly different shear failure stresses are measured. On the basis of these test data, a material failure line in 2D cases and a failure surface in 3D cases can be constructed by least-squares fitting or other numerical methods. The Mohr-Coulomb criterion can be expressed as a straight line in the coordinate system of the Mohr's circle as shown in Figure 6.10. The angle α which the failure plane encloses with the plane of the major principal stress in a triaxial test specimen is

$$\alpha = 45^\circ + \frac{1}{2}\phi \,. \tag{6.29}$$

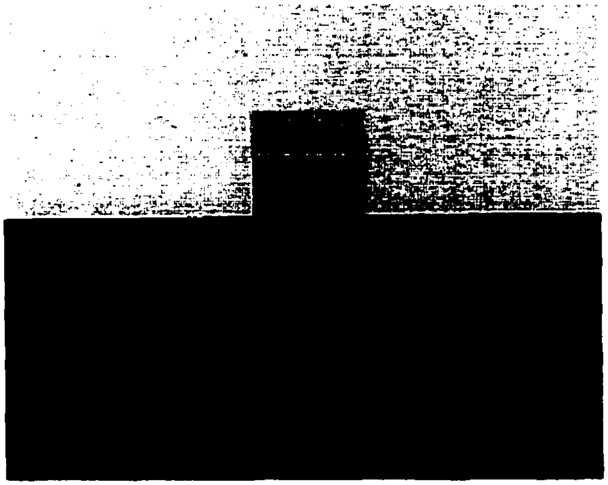
where ϕ is the material friction angle as in Equation (6.13). In the numerical analysis related to Figure 6.9(c), ϕ equals 10° leading to $\alpha = 50^\circ$ which is close to the orientation of the failure plane in Figure 6.9(c). In Figure 6.9(c), the failure plane is represented by the dark line segments and is located from the upper left-hand side to the lower right-hand side. If you replace these line segments by a straight line, the angle between this straight line and horizontal axis is close to the theoretical value 50°.



(a)

(b)

Figure 6.9: Images illustrating the fracture of a rectangular pile compressed from its top surface. (a) initial state; (b) tensile failure; (c) shear failure. (nodes: 36, elements: 60, time cost: 2.3 sec/frame)



(c)

Figure 6.9: Images illustrating the fracture of a rectangular pile compressed from its top surface. (a) initial state; (b) tensile failure; (c) shear failure. (nodes: 36, elements: 60, time cost: 2.3 sec/frame)

6.4.2 Qualitative Evaluation

Test Case 3: Smashing a cube against a wall

A cube (gold color) is thrown obliquely onto a vertical wall at a certain height, as shown in Figure 6.11. The wall is still and rigid, while the cube is fracturable and deformable. After the cube hits the wall. it is smashed into several pieces which then fall toward the ground. Tensile failure criterion is used in this example. Even though the cube itself is a very simple geometric object, an oblique collision does show the capability of the approach in handling the fracture in an arbitrary dynamic collision and the combination of fracture and deformation.

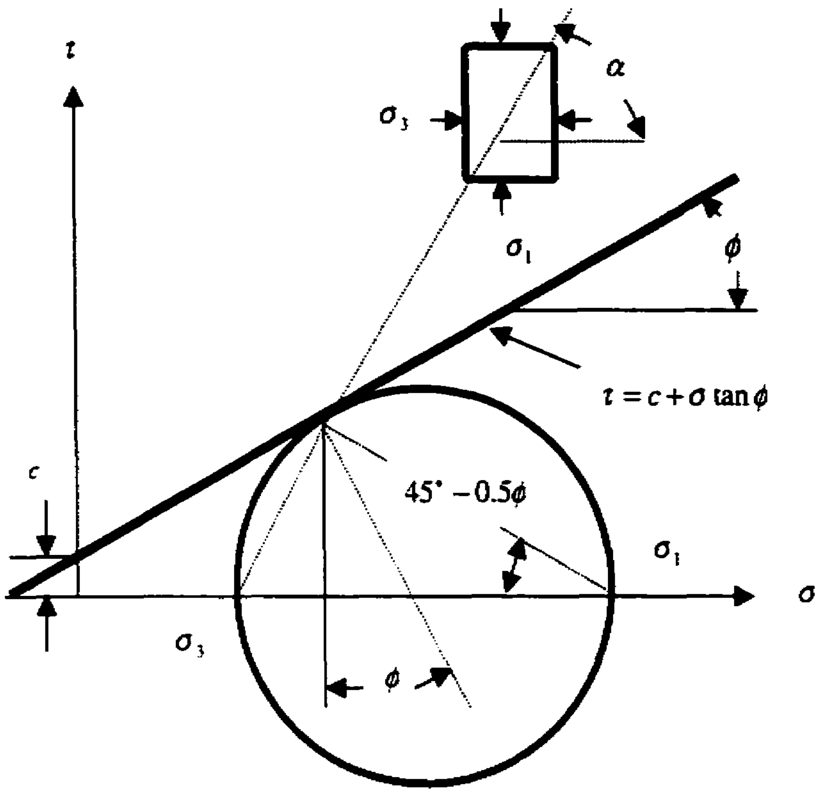


Figure 6.10: Mohr's circle representation of failure conditions in triaxially loaded sample [40].

Test Case 4: Fracture of a base caused by a falling cube

A cube is originally located at a certain height above a fracturable plate. After the falling cube hits the plate, dynamic fracture occurs inside the plate. All nodes on the bottom face of the plate are fixed and those nodes on the top face of the plate are free to move. No extra loading condition is introduced except gravity acting on the mass of the falling cube. Three types of failure criteria (tensile, Mohr-Coulomb and Tresca) are used to show the different failure patterns under the same dynamic loading conditions, as illustrated in Figures 6.12, 6.13 and 6.14. From them, it can be seen that the tensile and shear failure criteria produce different failure patterns which will be useful for end users to simulate the fracture of different materials. In the case of tensile failure criterion, the failure pattern resembles stack-like parallel cracks, while in the case of shear failure, the failure pattern is similar to slip-line pattern in plasticity, as shown in Figure 6.15 [51]. The tangents of slip lines are in the directions along which the principal shearing stress acts. The principal shearing stress is defined by one half of the difference between the major and minor principal stresses.

One distinction of the element-split approach from previous ones is that with the element-split approach, there is a flexibility to choose different failure criteria such as tensile and shear or any valid formula in the element stress format such that end-users have more control power to generate different failure patterns.

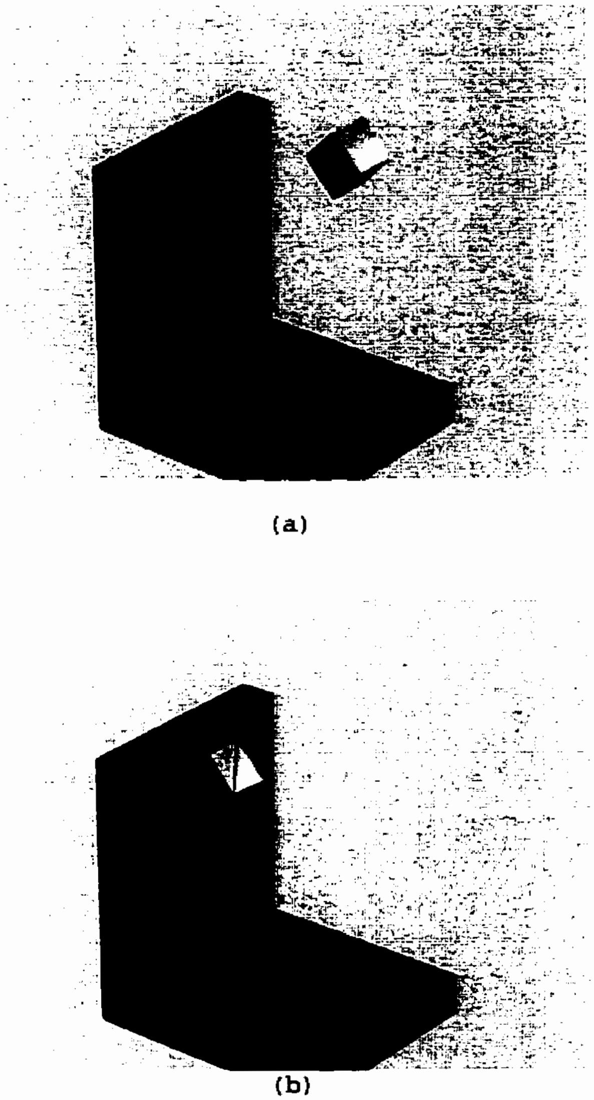
The current version of element-split approach fails in the following cases:

- Anisotropic material

  If the material has a preferred failure orientation, the current approach will fail.

- Plastic failure

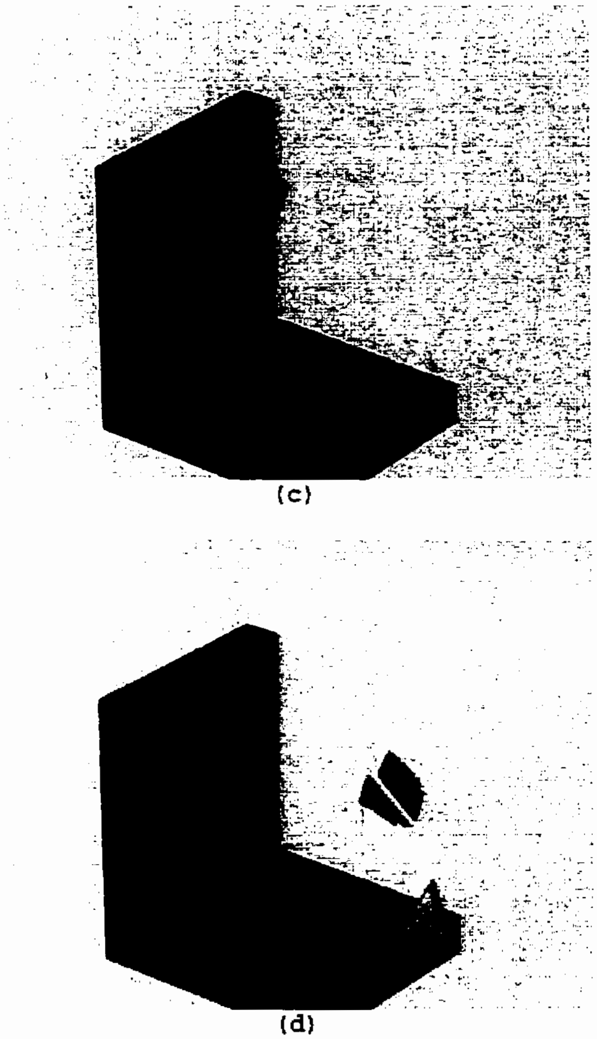
  The current approach can not handle with the plastic fracture where the plastic energy is consumed at the fracture tip.



(a)

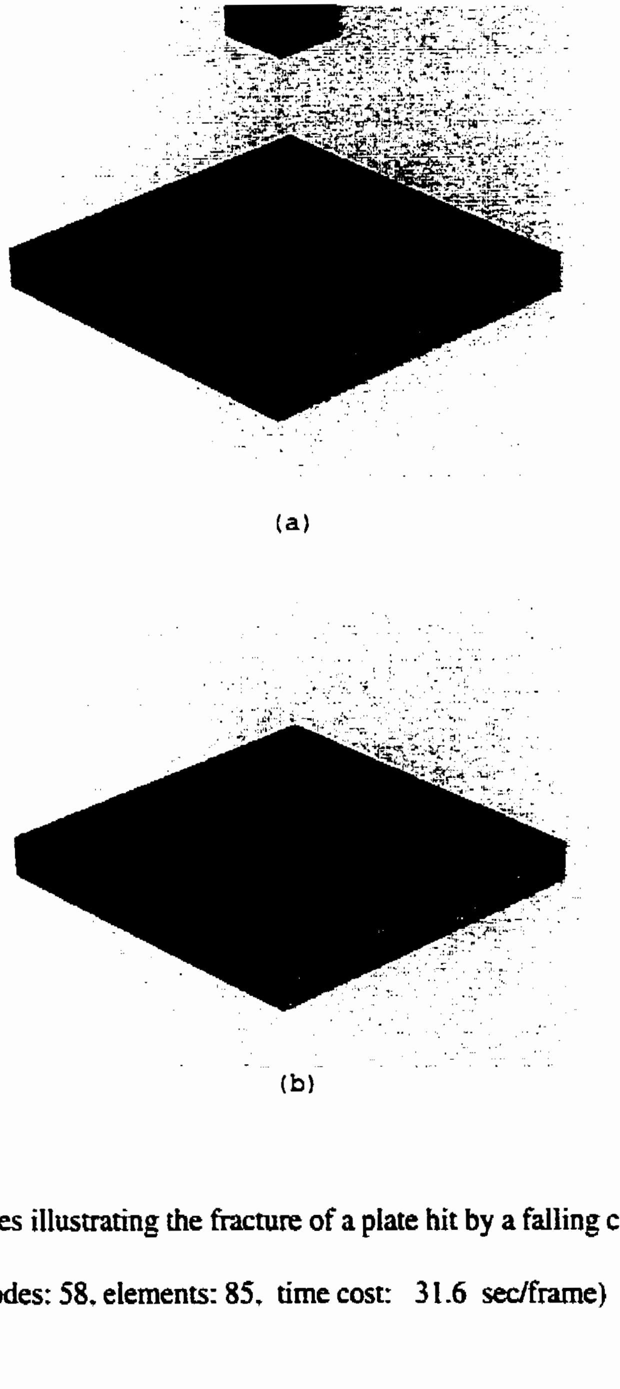
(b)

Figure 6.11: Images illustrating an oblique crashing of a cube onto a still wall. (nodes: 24, elements: 15, time cost: 1.7 sec/frame)


(c)

(d)

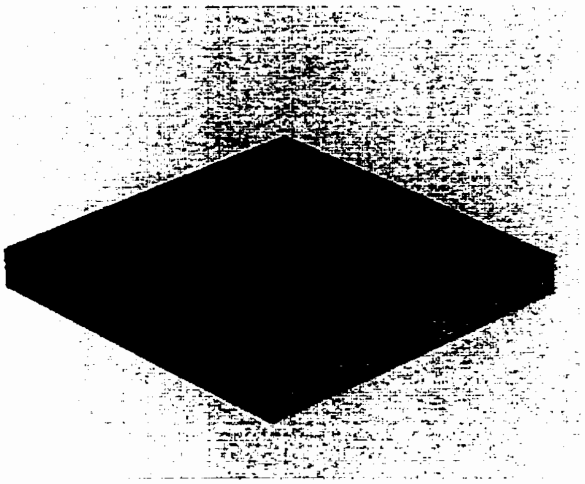
Figure 6.11: Images illustrating an oblique crashing of a cube onto a still wall. (nodes: 24, elements: 15, time cost: 1.7 sec/frame)



(a)

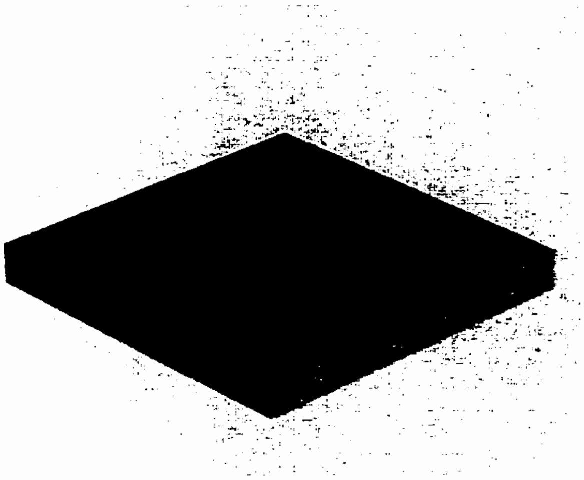
(b)

Figure 6.12: Images illustrating the fracture of a plate hit by a falling cube with tensile failure criterion. (nodes: 58, elements: 85, time cost: 31.6 sec/frame)



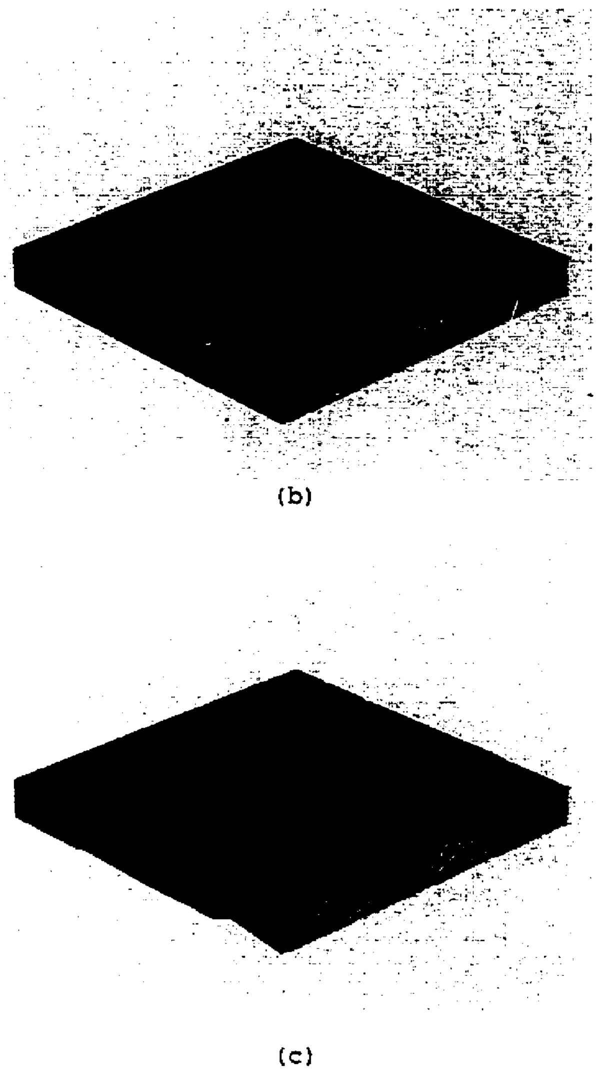
(c)

Figure 6.12: Images illustrating the fracture of a plate hit by a falling cube with tensile failure criterion.



(a)

Figure 6.13: Images illustrating the fracture of a plate hit by a falling cube with Mohr-Colomb's shear failure criterion. (nodes: 58, elements: 85, time cost: 129 sec/frame)



(b)

(c)

Figure 6.13: Images illustrating the fracture of a plate hit by a falling cube with Mohr-Colomb's shear failure criterion. (nodes: 58, elements: 85. time cost: 129 sec/frame)

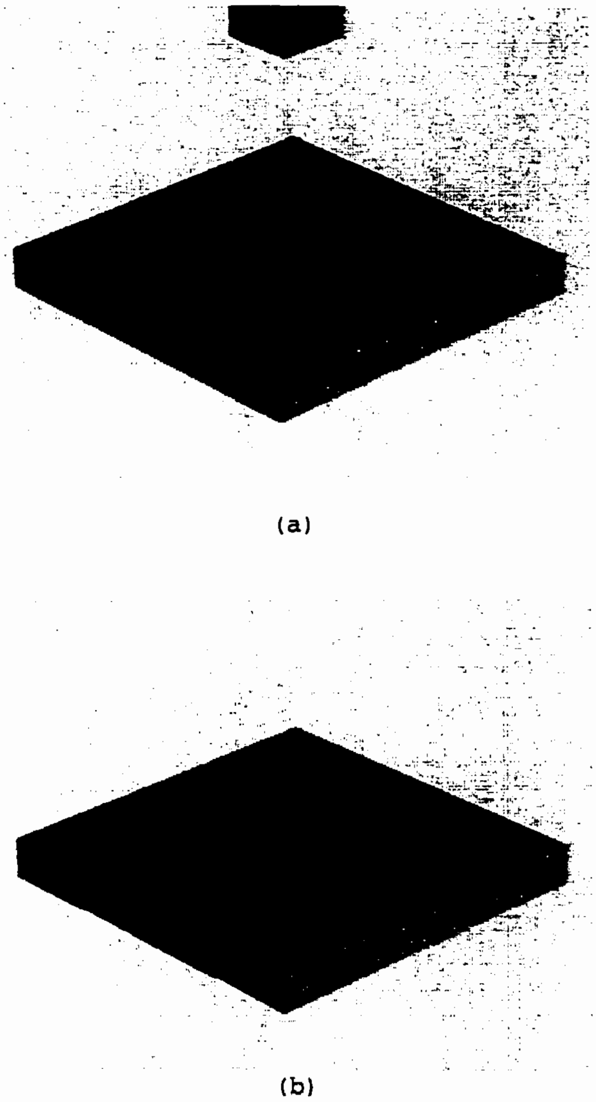
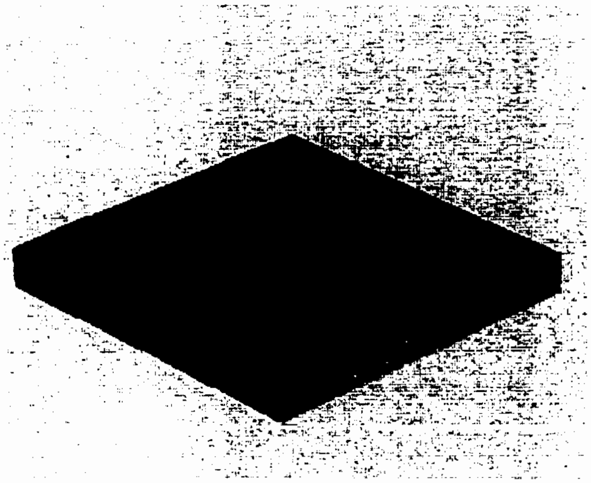


Figure 6.14: Images illustrating the fracture of a plate hit by a falling cube with Tresca's shear failure criterion. (nodes: 58, elements: 85, time cost: 131.3 sec/frame)



(c)

Figure 6.14: Images illustrating the fracture of a plate hit by a falling cube with Tresca's shear failure criterion. (nodes: 58, elements: 85, time cost: 131.3 sec/frame)

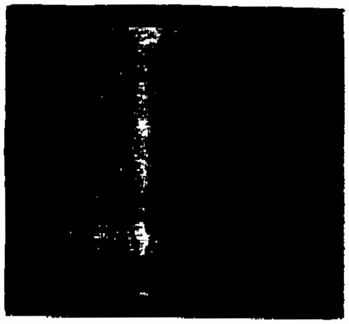


Figure 6.15: Experimental failure pattern on a polished marble cylinder which illustrates the slip line pattern [51].

Chapter 7

Conclusions

In this thesis, the author presents an accurate, general and fast approach to handle arbitrary collision between deformable objects. The following conclusions can be drawn from analyzing the experimental simulations introduced in Sections 5.6 and 6.4:

- The prediction accuracy of collision length is acceptable with 2.27% of error for the test case given in Section 5.6.2.1.
- The relative error for the momentum conservation is 1.39% for the test case in Section 5.6.2.2 and should be acceptable for solution (i.e., less than 5% for analysis solutions).
- The relative errors for energy conservation and impulse-momentum equality are 3.67% and 2.25%, respectively for the test case in Section 5.6.2.
- The calculated contact forces converge quickly and at most three iterations are required to achieve an acceptable accuracy (fluctuation = 0.01% which is less than 1% for convergence check).

- The simulated fracture of the beam in test case 1 in Section 6.4 is supported by an approximate analysis on the basis of material mechanics.
- The failure pattern of the compressed rectangular pile is consistent with the failure pattern of unconfined triaxial test specimens as well as the slip-line theory of plasticity. The failure patterns predicted by different failure criteria are different.

The major contributions to collision response analysis include the following:

(1) The forward Lagrange multiplier method is used for the first time in computer graphics to solve the collision response problem between arbitrarily-shaped deformable objects or between objects and arbitrarily-shaped environment in cases of static contact (or quasi-static contact) and dynamic collision. This approach overcomes most of the shortcomings associated with the penalty methods and the singularity problem of the conventional Lagrange multiplier method with the semi-explicit direct time integration (Sections 1 and 3.4).

(2) A new local finite element method is introduced to reduce computational cost. All calculations are performed at the local element level such that no global matrix factorization, inversion and assembly are needed, leading to a fast solution (guaranteed O(n) time complexity). The contact constraint in the collision response analysis is imposed in a generic way such that there is no assumption made w.r.t. the connectivity of nodes associated with different deformable objects, the formulation of constraint, the geometric format of surface, and the deformation zones or the distribution of deformation in each object.

(3) The quantitative numerical experiment indicates that the semi-explicit local collision response analysis approach matches very closely with the analytical solution of a classic collision example in solid mechanics, which numerically proves its effectiveness as a physics-based approach. The qualitative experiments show that the semi-explicit local collision response analysis approach can describe the local detailed deformation in any oblique complex collision without the need for "tweaking" the penalty coefficient or any other human intervention.

Overall, the semi-explicit local collision response analysis solution to collision response analysis is accurate, general-purpose and yet fast, which represents a new compromise between efficiency and functionality. It overcomes the shortcoming of the conventional finite element method for being computationally expensive [48,69-71] and the limitations associated with the boundary element method as explained in Section 2.1.3 [39]. It also provides more realistic contact deformation than the global deformation approaches [29,56,77], avoids the undesired properties associated with the penalty method [71,76], and avoids additional assumptions on deformation zones made in the global-local approach [15].

A new element-split scheme is proposed to simulate the fracture propagation. Since it is developed in the format of element stress, it is in principle compatible with most material failure criteria. As a result, the element-split approach is able to handle both tensile and shear failure which usually generate different failure patterns, while previous approaches in computer animation field are capable of describing only the tensile failure.

7.1 Future work

The animation system developed in this thesis has several limitations or shortcomings. The deformable object in this thesis is limited to structured material such as solids. One possible future extension to the current system is to handle the interaction between fluid and solid objects.

The fracture simulation scheme may be incorporated in an explosion simulation system which requires realistic representation of dynamic fracture in dynamic interaction between fluids and solids.

One limitation to the current fracture simulation scheme is that the material is assumed to be isotropic. Incorporation of anisotropic material properties will enhance the system to handle the fracture simulation of layered objects. The proposed element-split scheme provides a very good base for such type of enhancement.

In this thesis work, the simulation results are compared with the analytical solution of classic problems in solid mechanics. Another better way is to compare them with what really happens in the "real world" situation.

References

1. Baraff D. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics (Proc. SIGGRAPH)*. 1989;23(4):19-28.

2. Baraff D. Curved surfaces and coherence for non-penetrating rigid body simulation. *Computer Graphics (Proc. SIGGRAPH)*. 1990;24(4):19-28.

3. Baraff D. Linear-time dynamics using Lagrange Multipliers. *Computer Graphics (Proc. SIGGRAPH)*. 1996;30:137-146.

4. Baraff D, Witkin A. Large steps in cloth simulation. *Computer Graphics (Proc. SIGGRAPH)*. 1998;32:46-54.

5. Baraff D, Witkin A. Dynamic simulation of non-penetrating flexible bodies. *Computer Graphics (Proc. SIGGRAPH)*. 1992;26(2):303-308.

6. Barr A.H. Global and local deformations of solid primitives. *Computer Graphics (Proc. SIGGRAPH)*. 1984;18(3):21-30.

7. Barzel R., Barr A. A modeling system based on dynamic constraints. *Computer Graphics*. 1988;22(4):179-188.

8. Bathe K. *Finite Element Procedures*. New Jersey: Prentice-Hall Inc.; 1996.

9. Beer G., Watson J. *Introduction to Finite and Boundary Element Methods for Engineers*. New York: Wiley; 1992.

10. Belytschko T., Neal M.O. Contact-impact by the pinball algorithm with penalty and Lagrangian methods. *International Journal for Numerical Methods in Engineering*. 1991;31:547-572.

11. Bertsekas D. *Constrained Optimization and Lagrangian Multiplier Method.* New York: Academic Press; 1982.

12. Burtnyk N., Wein M. Computer-generated key-frame animation. *Journal of Society for Motion Picture and Television Engineers.* 1971;80:149-153.

13. Burtnyk N., Wein M. Towards a computer animating production tool. Brunel, England:174-185. Proceedings of Eurocomp Congress.

14. Cameron S. Collision detection by four-dimensional intersection testing. *IEEE Transactions of Robotics and Automation.* 1990;6(3):291-302.

15. Cani-Gascuel M., Desbrun M. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics.* 1997;3(1):39-50.

16. Canny J. Collision detection for moving polyhedra. *IEEE Transactions of Pattern Analysis and Machine Intelligence.* 1986;8(2):200-209.

17. Carigan M., Yang Y., Thalmann N.M., Thalmann D. Dressing animated synthetic actors with complex deformable clothes. *Computer Graphics (Proc. SIGGRAPH).* 1992;26(2):99-104.

18. Carpenter N.J., Taylor R.L., Katona M.G. Lagrange constraints for transient finite element surface contact. *International Journal for Numerical Methods in Engineering.* 1991;32:103-128.

19. Chandrasekharaiah D. *Continuum Mechanics.* Boston: Academic Press; 1994.

20. Clough R.W. The finite element method in plane stress analysis. Pittsburgh; 1960:345-378. Proceedings of 2nd ASCE Conference on Electronic Computation.

21. Cohen J., Lin M., Manocha D., Ponamgi M. I-collide: An interactive and exact collision detection system for large-scale environment; 1995:189-196. Proceedings of ACM Interactive 3D Graphics Conference.

22. Collatz L. *The Numerical Treatment of Differential Equations*. New York: Springer-Verlag; 1966.

23. Cook R.D. A plane hybrid element with rotational d.o.f. and adjustable stiffness. *International Journal for Numerical Methods in Engineerng*. 1987;24:1499-1508.

24. Cotin S., Delingette H., Ayache N. Real time volumetric deformable models for surgery simulation. In: *Volume Lectures Notes in Computer Science, Volume 11*; 1996. Visualization in Biomedical Computing.

25. Desai C., Christian J. *Numerical Methods in Geotechnical Engineering*. McGraw Hill Book Company; 1977.

26. Desbrun M., Schroder P., Barr A. Interactive animation of structured deformable objects.; 1999:1-8. Graphics Interface'99.

27. Drucker D., Prager W. Soil mechanics and plastic analysis or limit design. *Q. Appl. Math.* 1952;10(2):157-165.

28. Duff T. Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry. *Computer Graphics (Proc. SIGGRAPH)*. 1992;26(2):131-139.

29. Faloutsos P., Panne M.V.D., Terzopoulos D. Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics*. 1997;3(3):201-213.

30. Forsythe G. *Finite-Difference Methods for Partial Differential Equations*. New York: Wiley; 1960.

31. Fournier A., Reeves W. A simple model of ocean waves. *Computer Graphics (Proc. SIGGRAPH)*. 1986;27:131-320.

32. Fung Y. C. *Foundations of Solid Mechanics*. Prentice-Hall Inc.; 1965.

33. Goldstein H. *Classical Mechanics*. Reading, MA: Addison-Wesley; 1950.

34. Guyan R. Reduction of stiffness and mass matrices. *AIAA Journal*. 1965;3(2):380.

35. Herzen B.V., Barr A.H., Zatz H.R. Geometric collisions for time-dependent parametric surfaces. *Computer Graphics (Proc. SIGGRAPH)*. 1990;24(4):39-48.

36. Hill R. *The Mathematical Theory of Plasticity*. Clarendon Press; 1950.

37. Hirota K., Tanoue Y., Kaneko T. Generation of crack patterns with a physical model. *The Visual Computer*. 1998;14:126-137.

38. Hubbard P. Collision detection for interactive graphics applications. *IEEE Transactions of Visualization and Computer Graphics*. 1995;1(3):218-230.

39. James D., Pai D. ArtDefo, Accurate real time deformable objects. *Computer Graphics (Proc. SIGGRAPH)*. 1999.

40. Karafiath L. *Soil Mechanics for Off-Road Vehicle Engineering*. Trans Tech Publications; 1978.

41. Katona M.. Zienkiewicz O. A unified set of single step algorithms, Part 3: The Beta-m method, a generation of the Newmark scheme. *International Journal of Numerical Methods in engineering*. 1985;21:1345-1359.

42. Kurihara T.. Anjyo K.. Thalmann D. Hair animation with collision detection. In: *Models and Techniques in Computer Animation*; 1993:128-138.

43. Lafleur D., Thalmann M.N., Thalmann D. Cloth animation with self-collision detection. In: *Models and Techniques in Computer Animation*; 1992:179-185.

44. Lin M.C., Manocha D. Interference detection between curved objects for computer animation. In: *Models and Techniques in Computer Animation*; 1993:43-57.

45. Lohner R. Progress in grid generation via the advancing front technique. *Engineering with Computers*. 1996;12:186-210.

46. MacNeal R. *Finite Elements: Their Design and Performance*. Marcel Dekker, Inc.; 1994.

47. Mazarak O., Martins C., Amanatides J. Animating exploding objects. June; 1999. Graphics Interface '99.

48. Metaxas D., Terzopoulos D. Dynamic deformation of solid primitives with constraints. *Computer Graphics (Proc. SIGGRAPH)*. 1992;26(2):309-312.

49. Metaxas D., Terzopoulos D. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1993;15(6):580-591.

50. Moore M., Wilhelms J. Models and Techniques in Computer Animation. *Computer Graphics (Proc. SIGGRAPH)*. 1988;22(4):289-298.

51. Nadai A. *Plasticity*. New York: McGRAW-Hill Book Company; 1931.

52. Neff M., Fiume E. A visual model for blast waves and fracture. June; 1999. Graphics Interface '99.

53. Norton A., Turk G., Bacon B., Gerth J., Sweeney P. Animation of fracture by physical modeling. *The Visual Computer*. 1991;7:210-217.

54. O'Brein J.F., Hodgins J.K. Graphical modeling and animation of brittle fracture. *Computer Graphics (Proc. SIGGRAPH).* 1999;33:137-146.

55. Pentland A.P. Linear shape from shading. *International Journal of Computer Vision.* 19904:153-162.

56. Pentland A.P., Williams J. Good vibrations: Modal dynamics for graphics and animation. *Computer Graphics (Proc. SIGGRAPH).* 1989;23(3):185-192.

57. Platt J.C., Barr A.H. Constraint methods for flexible models. *Computer Graphics (Proc. SIGGRAPH).* 1988;22(4):279-288.

58. Platt J. Constraint methods for neural networks and computer graphics. California, USA: Caltech; 1989.

59. Press W., Teukolsky S., Vettering W., Flannery B. *Numerical Recipes in C.* Cambridge University Press; 1997.

60. Reddy J.N. *Energy and Variational Methods in Applied Mechanics.* John Wiley & Sons; 1984.

61. Reeves W. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions of Graphics.* 1983;2(2):91-108.

62. Reeves W., Blau R. Approximate and probabilistic algorithms for shading and rendering sturctured particle system. *Computer Graphics (Proc. SIGGRAPH).* 1985;19(3):313-322.

63. Shen J. Deformable object modelling for computer graphics using the time-dependent finite element method. M.Sc. thesis. Saskatoon, Canada: Department of Computer Science, University of Saskatchewan; 1997.

64. Shen J., Kushwaha R. *Soil-Tool Interaction, a Finite Element Perspective.* Marcel Dekker Inc.; 1998.

65. Shen J, Yang Y.H. Deformable Object Modeling using the Time-Dependent Finite Element Method. *Graphical Model and Image Processing*. 1998;60:461-487.

66. Sim K. Particle animation and rendering using data parallel computation. *Computer Graphics (Proc. SIGGRAPH)*. 1986;24(4):405-413.

67. Simo J.C., Wriggers P., Taylor R.L. A perturbed Lagrangian formulation for the finite element solution of contact problems. *Comp. Methods Appl. Mech. Eng.* 1985;50:163-180.

68. Smith J., Witkin A., Baraff D. Fast and controllable simulation of the shattering of brittle objects. Proceedings of Graphics Interface's 2000.

69. Terzopoulos D., Fleischer K. Deformable models. *The Visual Computer*. 1988;4:306-331.

70. Terzopoulos D., Fleischer K. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. *Computer Graphics (Proc. SIGGRAPH)*. 1988;22:269-278.

71. Terzopoulos D., Platt J., Barr A., Fleischer K. Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH)*. 1987;21(4):205-214.

72. Terzopoulos D., Witkin A. Physically based models with rigid and deformable components. Alberta, Canada; 1988:146-154. Proceedings of Graphics Interface.

73. Thompson J. A reflection on grid generation in the 90s: trends needs and influences. Mississippi State University. April; 1996:1029-1110. 5th International Conference on Numerical Grid Generation in Computational Field Simulations.

74. Timoshenko S. *Strength of Materials*. D. Van Nostrand Company, Inc.; 1955.

75. Witkin A. *An Introduction to Physically Based Modeling: Constrained Dynamics*. Lecture notes, Robotics Institute, Carnegie Mellon University; 1997.

76. Witkin A., Fleischer K., Barr A. Energy constraints on parameterized models. *Computer Graphics (Proc. SIGGRAPH)*. 1987;21(4):225-232.

77. Witkin A., Welch W. Fast animation and control of non-rigid structures. *Computer Graphics (Proc. SIGGRAPH)*. 1990;24(4):243-252.

78. Yang Y., Thalmann M.N. An improved algorithm for collision detection in cloth animation with human body; 1993:237-251. Proceedings of Pacific Graphics'93.

79. Yunus S.M., Saigal S., Cook R.D. On improved hybrid finite elements with rotational degrees of freedom. *International Journal for Numerical Methods in Engineering*. 1989;28:785-800.

Appendix A

Proof of Theorem 1

Theorem 1 If $(\dot{\theta}_x^{(i)}, \dot{\theta}_y^{(i)}, \dot{\theta}_z^{(i)})$ respectively refer to the average angular velocities of the sub-rotations about the global Cartesian coordinate axes x, y, and z in time step *i* and these rotations take place simultaneously in this time interval with the rotation angles varying continuously from 0 to $\Delta\theta_j^{(i)}$ $(j = x, y, z)$, then the combination of these rotations is equivalent to a single rotation about the axis vector $\mathbf{p} = (\dot{\theta}_x^{(i)}, \dot{\theta}_y^{(i)}, \dot{\theta}_z^{(i)})$ with the rotation angle $\Delta\theta^{(i)} = \sqrt{\left(\Delta\theta_x^{(i)}\right)^2 + \left(\Delta\theta_y^{(i)}\right)^2 + \left(\Delta\theta_z^{(i)}\right)^2}$.

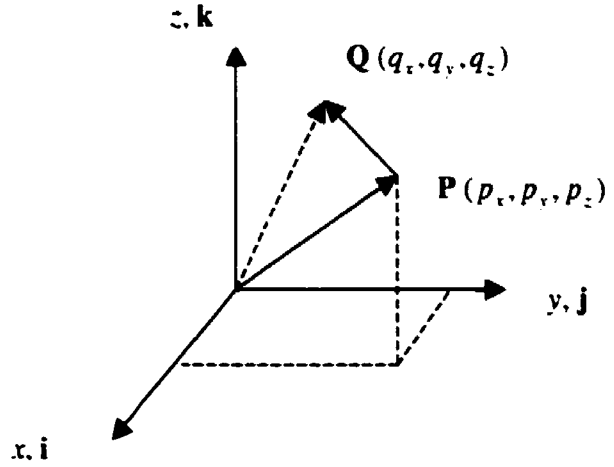


Figure A.1 A rotation axis **P** and an arbitrary point **Q**

Let $(\dot\theta_x^{(i)}, \dot\theta_y^{(i)}, \dot\theta_z^{(i)})$ be average angular velocity about axes x, y, and z, respectively, in time step i. Then we can approximate the continuous transformation in step i by constant speeds of sub-rotations $(\dot\theta_x^{(i)}, \dot\theta_y^{(i)}, \dot\theta_z^{(i)})$. If the three sub-rotations are equivalent to the single rotation about **P** axis at any moment in time step i, then the theorem is proved.

At an arbitrary time instant t, let $\mathbf{P} = \dot\theta_x^{(i)}\mathbf{i} + \dot\theta_y^{(i)}\mathbf{j}, + \dot\theta_z^{(i)}\mathbf{k}$ be the rotation axis for the single rotation and $\mathbf{Q} = q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}$ be an arbitrary point in the global Cartesian coordinate system. The translatory velocity caused by three sub-rotations is

$$\begin{aligned}\mathbf{V}_Q &= \dot\theta_x\mathbf{i}\times(q_y\mathbf{j}+q_z\mathbf{k})+\dot\theta_y\mathbf{j}\times(q_x\mathbf{i}+q_z\mathbf{k})+\dot\theta_z\mathbf{k}\times(q_x\mathbf{i}+q_y\mathbf{j})\\ &=(\dot\theta_y q_z-\dot\theta_z q_y)\mathbf{i}+(\dot\theta_z q_x-\dot\theta_x q_z)\mathbf{j}+(\dot\theta_x q_y-\dot\theta_y q_x)\mathbf{k}\end{aligned} \tag{A.1}$$

The translatory velocity at Point **Q** caused by the single rotation about axis **P** is given by

$$\begin{aligned}\mathbf{V}_Q &= \mathbf{P}\times\mathbf{PQ}\\ &=(\dot\theta_x\mathbf{i}+\dot\theta_y\mathbf{j}+\dot\theta_z\mathbf{k})\times\left[(q_x-\dot\theta_x)\mathbf{i}+(q_y-\dot\theta_y)\mathbf{j}+(q_z-\dot\theta_z)\mathbf{k}\right]\\ &=(\dot\theta_y q_z-\dot\theta_z q_y)\mathbf{i}+(\dot\theta_z q_x-\dot\theta_x q_z)\mathbf{j}+(\dot\theta_x q_y-\dot\theta_y q_x)\mathbf{k}\end{aligned} \tag{A.2}$$

Comparison between equations ( ) and ( ) indicates that two types of rotation are equivalent because they produce the same translatory velocity at any point of the object.

Since we approximate the angular velocities of $(\dot{\theta}_x^{(i)}, \dot{\theta}_y^{(i)}, \dot{\theta}_z^{(i)})$ to be constant in each time step, the single rotation axis **P** is then fixed and the incremental rotation angle about **P** is

$$\begin{aligned}\Delta\theta^{(i)} &= h\sqrt{(\dot{\theta}_x^{(i)})^2 + (\dot{\theta}_y^{(i)})^2 + (\dot{\theta}_z^{(i)})^2} \\ &= \sqrt{(\Delta\theta_x^{(i)})^2 + (\Delta\theta_y^{(i)})^2 + (\Delta\theta_z^{(i)})^2}\end{aligned} \tag{A.3}$$

where *h* is the length of time step *i*. Therefore, the theorem is proved.

Appendix B

Incremental Rotation Matrix

Rotation of an object is basically an orthogonal transformation. To derive the incremental rotation matrix $\mathbf{IR}_t$, let us assume that $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is a base of the linear space which corresponds to the Cartesian coordinate system. $(\mathbf{f}_1', \mathbf{f}_2', \mathbf{f}_3')$ and $(\mathbf{f}_1'', \mathbf{f}_2'', \mathbf{f}_3'')$ are respectively a set of vectors at two time steps, n-1 and n, as shown in Figure B.1. These two sets of vectors can be expressed by the multiplication of a rotation matrix and the base $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ as follows

$$[\mathbf{f}_1' \quad \mathbf{f}_2' \quad \mathbf{f}_3'] = \begin{bmatrix} r_{xx}' & r_{yx}' & r_{zx}' \\ r_{xy}' & r_{yy}' & r_{zy}' \\ r_{xz}' & r_{yz}' & r_{zz}' \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] = \mathbf{R}_T' [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] \qquad \text{(B.1)}$$

$$[\mathbf{f}_1'' \quad \mathbf{f}_2'' \quad \mathbf{f}_3''] = \begin{bmatrix} r_{xx}'' & r_{yx}'' & r_{zx}'' \\ r_{xy}'' & r_{yy}'' & r_{zy}'' \\ r_{xz}'' & r_{yz}'' & r_{zz}'' \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] = \mathbf{R}_T'' [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] \qquad \text{(B.2)}$$

The combination of above two equations leads to

$$[\mathbf{f}_1'' \quad \mathbf{f}_2'' \quad \mathbf{f}_3''] = \mathbf{R}_T'' (\mathbf{R}_T')^{-1} [\mathbf{f}_1' \quad \mathbf{f}_2' \quad \mathbf{f}_3'] \qquad \text{(B.3)}$$

Therefore, the iterative form of the incremental rotation matrix can be expressed by

$${}^{n}\mathbf{IR}_t = {}^{n}\mathbf{R}_t \left({}^{n-1}\mathbf{R}_t\right)^{-1} \qquad \text{(B.4)}$$

where ${}^{n}\mathbf{IR}_{t}$ refers to the incremental rotation matrix from time step *n*-1 to *n*.

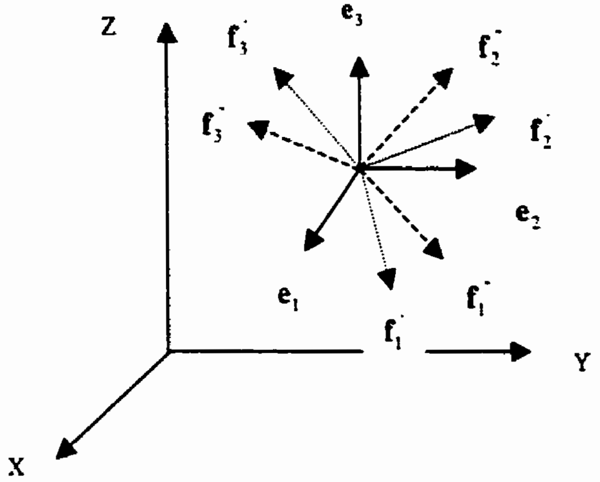


Figure B.1: A set of vectors at two time steps, *n*-1 and *n*.

Appendix C

Enforcement of the Coulomb Law

```
$F_n = (X_1 F_x + Y_1 F_y + Z_1 F_z)$
if $F_n < 0$      // tension mode
        $\vec{F}^{'} = 0$
        remove the penetrating node from *CollisionList*
else    // compression mode
        if $|\vec{F}_t| = 0$
          $\vec{F}^{'} = \vec{F}_n$
        else
          if $|\vec{F}_t| > |\vec{F}_n| \cdot \tan\delta$      // exceed the friction capacity
              $|\vec{F}_t^{'}| = |\vec{F}_n| \cdot \tan\delta$
          else                    // stick mode
              $|\vec{F}_t^{'}| = |F_t|$
          endif
          let $\vec{F}_t = \vec{F} - \vec{F}_n = (F_x - X_1 F_x,\ \ F_y - Y_1 F_y,\ \ F_z - Z_1 F_z) = (X_2,\ \ Y_2,\ \ Z_2)$
          $\vec{F}_t^{'} = \dfrac{(X_2,\ \ Y_2,\ \ Z_2) \cdot |F_t^{'}|}{\sqrt{X_2^2 + Y_2^2 + Z_2^2}} = (X_3,\ \ Y_3,\ \ Z_3)$
          $\vec{F}^{'} = \vec{F}_t^{'} + \vec{F}_n = (X_3 + X_1 F_x,\ \ Y_3 + Y_1 F_y,\ \ Z_3 + Z_1 F_z)$
        endif
endif
```

Appendix D

An Incremental Scheme for Collision Response with Gauss-Seidel Iteration

Step 1 calculate the displacement component due to non-collision factors

$$^{n+1}\mathbf{u}_t^{*} = \left(\frac{\mathbf{M}}{h^2}+\frac{\mathbf{C}}{2h}\right)^{-1}\left({}^{n}\mathbf{F}+\frac{\mathbf{M}}{h^2}(2\ {}^{n}\mathbf{u}_t - {}^{n-1}\mathbf{u}_t)+\frac{\mathbf{C}}{2h}\ {}^{n-1}\mathbf{u}_t - {}^{n}\mathbf{K}\ {}^{n}\mathbf{u}_t \right)$$

$$\Delta\left({}^{n+1}\mathbf{u}_r^{*}\right) \leftarrow {}^{n+1}\mathbf{IR}_t^{*} \leftarrow \Delta\left({}^{n+1}\mathbf{R}_t^{*}\right) = h\ {}^{n}\dot{\theta}^{*}\ {}^{n}\mathbf{R}_t$$

Step 2 Gauss-Seidel iteration calculates the contact forces and the displacement component due to collision factor

Step 2.1 Initialization

$$\Delta\left({}^{n+1}\mathbf{u}_t^{\lambda}\right)=0,\ \ {}^{n}\lambda=0.\ \ \Delta\left({}^{n+1}\mathbf{u}_r^{\lambda}\right)=0$$

calculate parameters $t_1^{(j)}$. $t_2^{(j)}$, and $t_3^{(j)}$ for all penetrating node j (=1,...., N_c)

Step 2.2 Gauss-Seidel iteration for i =1,2,.... N_{gs}

2.2.1 update the coordinates of contact nodes

$$\begin{bmatrix} x_a & y_a & z_a & x_b & y_b & z_b & x_c & y_c & z_c & x_j & y_j & z_j \end{bmatrix}^T =$$
$$\left\{\Delta\left({}^{n+1}\mathbf{u}_t^{\lambda}\right)+\Delta\left({}^{n+1}\mathbf{u}_t^{*}\right)+\Delta\left({}^{n+1}\mathbf{u}_r^{\lambda}\right)+\Delta\left({}^{n+1}\mathbf{u}_r^{*}\right)+\mathbf{X}_n\right\}$$

2.2.2 calculate the penetration components of node j

$$\begin{bmatrix} p_x^{(j)} \\ p_y^{(j)} \\ p_z^{(j)} \end{bmatrix} = \begin{bmatrix} (1-t_1^{(j)}-t_2^{(j)})x_a + t_2^{(j)}x_b + t_1^{(j)}x_c - x_j \\ (1-t_1^{(j)}-t_2^{(j)})y_a + t_2^{(j)}y_b + t_1^{(j)}y_c - y_j \\ (1-t_1^{(j)}-t_2^{(j)})z_a + t_2^{(j)}z_b + t_1^{(j)}z_c - z_j \end{bmatrix}$$

2.2.3 calculate the increment of contact forces

$$d = (1 - t_1^{(j)} - t_2^{(j)})^2 \left(\frac{h^2}{m_a} + \frac{2h}{c_a} \right) + (t_2^{(j)})^2 \left(\frac{h^2}{m_b} + \frac{2h}{c_b} \right) + (t_1^{(j)})^2 \left(\frac{h^2}{m_c} + \frac{2h}{c_c} \right)$$

$$\begin{bmatrix} \Delta\lambda_x^{(j)} \\ \Delta\lambda_y^{(j)} \\ \Delta\lambda_z^{(j)} \end{bmatrix} = \frac{1}{d} \begin{bmatrix} p_x^{(j)} \\ p_y^{(j)} \\ p_z^{(j)} \end{bmatrix}$$

2.2.4 enforce the Coulomb law using Appendix I

insert Appendix I here

2.2.5 update contact forces ${}^{n}\lambda$

$$\begin{bmatrix} \lambda_x^{(j)} \\ \lambda_y^{(j)} \\ \lambda_z^{(j)} \end{bmatrix} = \begin{bmatrix} \lambda_x^{(j)} + \Delta\lambda_x^{(j)} \\ \lambda_y^{(j)} + \Delta\lambda_y^{(j)} \\ \lambda_z^{(j)} + \Delta\lambda_z^{(j)} \end{bmatrix}$$

2.2.6 update the rotation displacement component due to collision factor

$$\Delta\left({}^{n+1}\mathbf{u}_r^{i} \right) \leftarrow {}^{n}\dot{\theta},\ {}^{n+1}\dot{\theta} \ \leftarrow \mathbf{Q}_n^{i} \leftarrow {}^{n}\lambda$$

2.2.7 update the translation displacement component due to collision factor

$$\Delta\left({}^{n+1}u_t^{\lambda}\right) \leftarrow \Delta\left({}^{n+1}u_t^{\lambda}\right) - \left\{\begin{matrix}
\Delta\lambda_x^{(j)}(1-t_1^{(j)}-t_2^{(j)})\left(\dfrac{h^2}{m_a}+\dfrac{2h}{c_a}\right) \\
\Delta\lambda_y^{(j)}(1-t_1^{(j)}-t_2^{(j)})\left(\dfrac{h^2}{m_a}+\dfrac{2h}{c_a}\right) \\
\Delta\lambda_z^{(j)}(1-t_1^{(j)}-t_2^{(j)})\left(\dfrac{h^2}{m_a}+\dfrac{2h}{c_a}\right) \\
\Delta\lambda_x^{(j)}\ t_2^{(j)}\left(\dfrac{h^2}{m_b}+\dfrac{2h}{c_b}\right) \\
\Delta\lambda_y^{(j)}\ t_2^{(j)}\left(\dfrac{h^2}{m_b}+\dfrac{2h}{c_b}\right) \\
\Delta\lambda_z^{(j)}\ t_2^{(j)}\left(\dfrac{h^2}{m_b}+\dfrac{2h}{c_b}\right) \\
\Delta\lambda_x^{(j)}\ t_1^{(j)}\left(\dfrac{h^2}{m_c}+\dfrac{2h}{c_c}\right) \\
\Delta\lambda_y^{(j)}\ t_1^{(j)}\left(\dfrac{h^2}{m_c}+\dfrac{2h}{c_c}\right) \\
\Delta\lambda_z^{(j)}\ t_1^{(j)}\left(\dfrac{h^2}{m_c}+\dfrac{2h}{c_c}\right) \\
\Delta\lambda_x^{(j)}\left(\dfrac{h^2}{m_j}+\dfrac{2h}{c_j}\right) \\
\Delta\lambda_y^{(j)}\left(\dfrac{h^2}{m_j}+\dfrac{2h}{c_j}\right) \\
\Delta\lambda_z^{(j)}\left(\dfrac{h^2}{m_j}+\dfrac{2h}{c_j}\right)
\end{matrix}\right\}$$

2.2.8 convergence test for the Gauss-Seidel iteration

if $\dfrac{\sqrt{\sum_{j=1}^{N_c}\left[(\Delta\lambda_x^{(j)})^2+(\Delta\lambda_y^{(j)})^2+(\Delta\lambda_z^{(j)})^2\right]}}{\sqrt{\sum_{j=1}^{N_c}\left[(\lambda_x^{(j)})^2+(\lambda_y^{(j)})^2+(\lambda_z^{(j)})^2\right]}} < tol$, then break;

otherwise, return to Step 2.2.

Step 3 Calculate displacements in the current time step

$$\Delta\left({}^{n+1}\mathbf{u}_t\right)=\Delta\left({}^{n+1}\mathbf{u}_t^{*}\right)+\Delta\left({}^{n+1}\mathbf{u}_t^{\lambda}\right)$$

$$\Delta\left({}^{n+1}\mathbf{u}_r\right)=\Delta\left({}^{n+1}\mathbf{u}_r^{*}\right)+\Delta\left({}^{n+1}\mathbf{u}_r^{\lambda}\right)$$

Step 4 Calculate velocities and accelerations

$${}^{n}\dot{\mathbf{u}}_t=\frac{1}{2h}\left\{{}^{n+1}\mathbf{u}_t-{}^{n-1}\mathbf{u}_t\right\}$$

$${}^{n}\ddot{\mathbf{u}}_t=\frac{1}{h^2}\left\{{}^{n+1}\mathbf{u}_t-2\ {}^{n}\mathbf{u}_t+{}^{n-1}\mathbf{u}_t\right\}$$

$$\Delta\left({}^{n+1}\dot{\theta}\right)=h\ {}^{n}\mathbf{M}_l^{-1}\left\{{}^{n}\mathbf{Q}-{}^{n}\mathbf{Q}^{\lambda}\right\}$$

$${}^{n+1}\dot{\theta}={}^{n}\dot{\theta}+\Delta\left({}^{n+1}\dot{\theta}\right)$$

Step 5 Increment the time step $n \leftarrow n+1$, return to step 1

Appendix E

An Two-time-interval Scheme for Dynamic Interaction between Deformable Objects

Step 1 Initialization

$$ {}^{0}\mathbf{u}_t = {}^{1}\mathbf{u}_t - D_t \; {}^{1}\dot{\mathbf{u}}_t + 0.5 D_t^2 \; {}^{1}\ddot{\mathbf{u}}_t $$

$$ {}^{0}\theta = {}^{1}\theta - D_t \; {}^{1}\dot{\theta} + 0.5 D_t^2 \; {}^{1}\ddot{\theta} $$

Step 2 Incremental step-by-step integration **for** i (=1,..., N_{step})

2.1 $\; {}^{n+1}\mathbf{X} = {}^{n}\mathbf{X} + \Delta\left({}^{n+1}\mathbf{u}_t\right) + \Delta\left({}^{n+1}\mathbf{u}_r\right)$

2.2 calculate the reference force vector **TF** (= ${}^{n}\mathbf{K}\;{}^{n}\mathbf{u}$) equivalent to element stress

2.3 calculate the displacement component due to non-collision factors

$$ {}^{n+1}\mathbf{u}_t^{*} = \left(\frac{\mathbf{M}}{h^2} + \frac{\mathbf{C}}{2h}\right)^{-1} \left({}^{n}\mathbf{F} + \frac{\mathbf{M}}{h^2}(2\;{}^{n}\mathbf{u}_t - {}^{n-1}\mathbf{u}_t) + \frac{\mathbf{C}}{2h}\;{}^{n-1}\mathbf{u}_t - {}^{n}\mathbf{K}\;{}^{n}\mathbf{u}_t \right) $$

$$ {}^{n+1}\theta^{*} = h^2 \;{}^{n}\mathbf{M}_I^{-1}\;{}^{n}\mathbf{Q} + 2\;{}^{n}\theta - {}^{n-1}\theta $$

$$ {}^{n+1}\mathbf{u}_r^{*} \leftarrow {}^{n+1}\theta^{*},\; {}^{n}\theta $$

($h = D_t$ if *CollisionMode* = FALSE; $h = D_{t\_c}$, otherwise.)

2.4 (*CollisionMode*, *CollisionList*) ← result of collision detection

2.5 **if** (CollisionMode is FALSE & CollisionList is empty) // non-collision stage

use the Lagrange Dynamics to describe the global movement of objects

2.6 **else if** (CollisionMode is TRUE & CollisionList is empty) // beginning of collision

$$ {}^{n+1}\mathbf{u}_t^{*} = \left(\frac{\mathbf{M}}{h^2} + \frac{\mathbf{C}}{2h}\right)^{-1} \left({}^{n}\mathbf{F} + \frac{\mathbf{M}}{h^2}(2\;{}^{n}\mathbf{u}_t - {}^{n-1}\mathbf{u}_t) + \frac{\mathbf{C}}{2h}\;{}^{n-1}\mathbf{u}_t - {}^{n}\mathbf{K}\;{}^{n}\mathbf{u}_t \right) $$

$$ {}^{n+1}\theta^{*} = h^2 \;{}^{n}\mathbf{M}_I^{-1}\;{}^{n}\mathbf{Q} + 2\;{}^{n}\theta - {}^{n-1}\theta $$

```
      ${}^{n+1}\mathbf{u}_r^* \leftarrow {}^{n+1}\theta^*, {}^{n}\theta$
         ( $h = D_{t\_f}$  which corresponds to the exact beginning of the collision)

2.7  **else**   // collision stage
     insert Appendix D (except Step 1) here
2.8  return to Step 2
```

Appendix F

Nomenclature

α	penalty coefficient.
ϕ	internal friction angle of a material.
Λ	Lagrange multiplier vector.
λ	Lagrange multiplier.
δ	rotation angle.
ρ	density.
μ	Poison's ratio.
σ	stress.
$\sigma_1, \sigma_2, \sigma_3$	principal stresses.
τ	shear stress or strength.
Π	total potential energy of the system.
$\mathbf{C}$	damping matrix.
c	cohesion of a material.
E	elastic modulus.
$\mathbf{FC}_n$	normal force.
$\mathbf{FC}_t$	tangential force.
$\mathbf{G}$	geometric constraint matrix.
h	time interval.

$I_1, I_2 and I_3$	first, second and third invariants of the stress, respectively.
$\mathbf{IR}$	incremental rotation matrix (Appendix B).
$\mathbf{K}$	stiffness matrix.
k_0	failure threshold.
$\mathbf{M}$	mass matrix.
$\mathbf{M}_I$	inertia tensor.
P	projection of the penetration distance of a penetrating node.
$\mathbf{Q}$	external torque vector.
$\mathbf{Q}^i$	vector of torques caused by contact forces
$\mathbf{R}$	external load vector.
t	time.
$\mathbf{T}$	rotational matrix.
$\mathbf{U}$	displacement matrix.
$\mathbf{u}$	displacement vector.
$\dot{\mathbf{u}}$	velocity vector.
$\ddot{\mathbf{u}}$	acceleration vector.
$\mathbf{u}_r$	rotational displacement vector.
$\mathbf{u}_t$	translatory displacement vector.
$\mathbf{W}$	material deformation gradient.
$\mathbf{X}$	coordinate vector.

Appendix G

Glossary

Collision detection the act of detecting where and when a collision occurs.

Collision response the behavior of objects after a collision happens.

Computer animation a technique to add the dimension of time into computer graphics.

Continuum mechanics mechanics of continuum media by which an object may be divided infinitely and the problem is defined using the mathematical fiction of an infinitesimal.

Deformable object an object which is allowed to deform.

Failure stress the stress at which an object starts to fail.

Elasticity the property by which an object returns to its initial geometric configuration after removal of external loads.

Finite difference method a technique to replace continuous derivatives in equations governing a system with the ratio of changes in the variables over small, but finite increments.

Finite element method a variational procedure in which the approximating functions such as algebraic polynomials are used to approximate the solution variables in simple subdomains (called finite elements) into which a given domain is divided.

Failure criterion a binary standard used to identify whether or not a material fails.

Fracture the breaking of an object into two or more parts.

Normal stress the stress perpendicular to the cross section of an object.

Particle system a system consisting of many particles each of which is controlled by a specific birth and death mechanism.

Plasticity the property by which an object remains at its deformed shape after removal of external loads.

Principal stress the normal stress on the plane where shear stresses do not exist.

Rigid object an object which is not allowed to deform.

Shear stress the stress tangential to or inside the cross section of an object.

Strain the deformation per unit length.

Stress the intensity of force, i.e., the force per unit area.