Algorithmic and modelling aspects of discrete element method with applications to shot preening and peen forming processes.

Han, Kuanjin

Civil & Computational Engineering Centre
School of Engineering
University of Wales, Swansea

Algorithmic and Modelling Aspects of Discrete Element Method With Applications to Shot Peening and Peen Forming Processes

Kuanjin Han

June 2004

Dissertation Submitted To The University Of Wales In Candidature For The Degree Of Doctor Of Philosophy

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Declaration

This is to certify that this dissertation has not been previously accepted in part or in substance in candidature for any degree and is not concurrently submitted in candidature for any degree at any other university.

29.11.2004
Date

Statement I

This is to certify that except where specific reference to other investigation is made, the work presented in this dissertation is the result of the investigations of the candidate.

29.11.2004
Date

Statement II

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

29.11.2004
Date

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Prof. D. Perić and Prof. D.R.J. Owen, for their support, guidance and encouragement during this research.

I am much indebted to many researchers at Rockfield Software Ltd., Swansea. Special thanks are to Dr. J. Yu and Dr. A.J.L. Crook, who introduced me to the discrete element method and offered lots of trouble shooting at the early stage of this work.

I appreciate very much my husband's support and patience.

ABSTRACT

Several major algorithmic and modelling aspects of the discrete element method are presented, with particular effort made to establish a general framework for the combined finite/discrete element simulation of shot peening and peen forming processes.

On the basis of a general review of the contact detection algorithms, the alternating digital tree (ADT) and the augmented spatial digital tree (ASDT) are presented as examples of tree based search algorithms; whilst the dynamic cell based search algorithm is presented as an example of cell/grid based search algorithms. Extensive numerical tests are provided to compare the performance of the approaches discussed and to perform a parametric study of the dynamic cell algorithm.

Contact resolution for circular and spherical discrete objects is comprehensively discussed. Following the establishment of geometric relations between contact pairs, four interaction laws for the normal contact are reviewed, together with a general analysis of the equivalence between them.

A modified classic Coulomb friction model is proposed for 3D contact problems, and a new time stepping scheme is developed to ensure both short and long term stability of the contact models. In addition, two different schemes that incorporate energy dissipation into the contact model are reviewed.

A new approach is proposed for contact resolution of 2D superquadrics.

An advancing front technique based algorithm is described to generate a random packing of disks, ellipses and polygons; whereas a geometric based compression algorithm is developed to generate a random packing of spheres of various sizes within a geometric domain.

A general algorithmic framework is established for the combined finite/discrete element analysis of shot peening processes, particularly for the evaluation and comparison of several interaction laws governing the contact between shot and workpiece. The proposed model is also verified with the experiments conducted elsewhere.

A two stage combined finite/discrete element and explicit/implicit solution procedure for the simulation of peen forming process is developed, and verified with experiments.

Contents

1 **Introduction** **1**

1.1 Overview of Numerical Methods for Discrete Systems 1

1.2 Main Building Blocks of the Discrete Element Procedure 3

1.2.1 Representation of discrete objects 3

1.2.2 Contact detection 4

1.2.3 Contact resolution 4

1.2.4 Time stepping 5

1.3 Objectives of the Research 7

1.4 Layout of the Thesis 7

2 **Contact Detection** **11**

2.1 Introduction 11

2.2 Body representation 11

2.3 Tree Based Search Algorithms: ADT and ASDT 13

2.3.1 Spatial binary tree structure for a set of rectangles 13

2.3.2 Alternating digital tree algorithm 13

2.3.3 Augmented spatial digital tree algorithm 17

2.4 Cell/Grid Based Search Algorithms 19

2.4.1 Static cell based algorithm (S-Cell) 19

2.4.2 No Binary Search (NBS) 20

2.4.3 Dynamic cell based search algorithm (D-Cell) 20

2.5 Numerical Tests 23

2.6 Concluding Remarks 28

3 Contact Resolution for Circular and Spherical Discrete Objects 32
3.1 Introduction 32
3.2 Contact Geometry and Forces 33
3.2.1 Disk/segment and disk/disk contact 33
3.2.2 Sphere/facet and sphere/sphere contact 38
3.3 Normal Interaction Laws 42
3.3.1 The linear/Hooke model 42
3.3.2 The Hertz models 42
3.3.3 The Winkler models 44
3.3.4 The power law model 45
3.3.5 Equivalence between models 46
3.4 Modelling of Tangential Friction 47
3.4.1 Modified classic Coulomb friction model for spheres 48
3.4.2 Computation of tangential force at discrete time steps 50
3.5 Modelling of Energy Dissipation 52
3.5.1 Hysteretic model 52
3.5.2 Viscous damping model 54
3.6 Rheological Representation of Normal Contact 55
3.7 Critical time step 56
3.7.1 Critical time step for nonlinear systems 57
3.7.2 Time step scheme for power law based contact models 58
3.8 Concluding Remarks 59

4 Contact Resolution for Non-circular Discrete Objects 62
4.1 Introduction 62
4.2 Polygonal Approximation of a Superquadric 63
4.2.1 Uniformly spaced sampling 63
4.2.2 Adaptive sampling 64
4.3 Intersections of Two Polygons 67
4.3.1 A linear algorithm 67

4.3.2 Modification . . . 69

4.4 Contact Forces and Directions . . . 70

4.4.1 Contact geometry . . . 70

4.4.2 Definition of normal contact forces . . . 71

4.4.3 Normal and tangential directions . . . 72

4.4.4 Reference contact point . . . 73

4.4.5 Normal force magnitude and choice of energy function . . . 74

4.4.6 Extensions . . . 74

4.5 Illustrations . . . 76

4.6 Concluding Remarks . . . 76

5 Packing of Disks, Ellipses and Polygons with the Advancing Front Technique 80

5.1 Introduction . . . 80

5.2 Packing of Disks . . . 81

5.2.1 Advancing front approach: closed form . . . 83

5.2.2 Advancing front approach: open form . . . 89

5.2.3 Packing under gravity . . . 91

5.2.4 Illustrations . . . 92

5.3 Packing of Convex Polygons . . . 92

5.4 Packing of Ellipses . . . 98

5.5 Concluding Remarks . . . 100

6 Sphere Packing with a Geometric Based Compression Algorithm 103

6.1 Introduction . . . 103

6.2 Algorithmic Description . . . 104

6.2.1 Compression of a given packing . . . 104

6.2.2 Refilling . . . 109

6.2.3 Remarks on the algorithm . . . 110

6.3 Illustrations . . . 111

6.4 Concluding Remarks . . . 113

7 Combined Finite/Discrete Element Simulations of Shot Peening Processes 115

7.1 The Shot Peening Process . . . 115

7.1.1 Peening media . . . 117

7.1.2 The Almen system for intensity control . . . 118

7.1.3 Coverage . . . 119

7.1.4 Saturation . . . 119

7.2 Review of Modelling of the Shot Peening Process . . . 120

7.3 Combined FEM/DEM: A General Description . . . 121

7.4 2D Analysis . . . 122

7.4.1 Comparisons of linear and Hertz models with different values of α . . 125

7.4.2 Comparisons of two partial contact factor evaluation schemes . . . 129

7.5 3D Analysis . . . 129

7.5.1 Comparison with axisymmetric model . . . 130

7.5.2 Choices of parameter values α for normal contact models . . . 132

7.5.3 Effect of contact friction . . . 133

7.5.4 Effect of element size . . . 133

7.5.5 Effect of single and multiple shot impact . . . 134

7.6 Verification against Experimental Measurements . . . 135

7.7 Concluding Remarks . . . 138

8 Combined Finite/Discrete Element and Explicit/Implicit Simulations of Peen Forming Processes 141

8.1 The Peen Forming Process . . . 141

8.2 Review of Modelling of the Peen Forming Process . . . 142

8.3 Combined FE/DE and Explicit/Implicit Scheme: a General Description . . . 143

8.4 Residual Stress Profile: Combined FE/DE Explicit Approach . . . 145

8.5 Global Deformation and Stress Distribution: Implicit Static Solution . . . 146

8.5.1 Residual profile application . . . 147

8.5.2 Incremental implicit analysis . . . 147

8.5.3 Generalised implementation . . . 150

8.6 Numerical Example 150

8.6.1 The first stage analysis: residıal profile 150

8.6.2 The second stage analysis: deformation and stress re-distribution . . . 152

8.7 Concluding Remarks 155

9 Conclusions and Further Research 157

9.1 Conclusions 157

9.2 Recommendations for Further Reseaıch 159

9.2.1 Modelling of friction 159

9.2.2 Contact resolution for polyhedrons and 3D superquadrics 160

9.2.3 Packing of irregular shaped dscrete objects 160

9.2.4 Alternative solution strategies 160

9.2.5 Parallelisation 160

Chapter 1

Introduction

1.1 Overview of Numerical Methods for Discrete Systems

Over the last decades numerical modelling, supplemented by a tremendous increase in computer power, has emerged as an integrated part of engineering design. There is no doubt that the finite element method (FEM) is the most powerful and versatile numerical tool. Based initially on continuum mechanics and further extended to general partial differential systems, the FEM can be employed in a wide range of engineering applications, including linear and nonlinear, static and dynamic, thermal and mechanical problems.

Many practical problems, however, may exhibit a strong discrete/discontinuous nature. Typical examples are granular and particulate materials in process engineering and geomechanics. The system often consists of an excessively large number of individual discrete objects and the overall behaviour of such a system is determined by the motion of the objects involving interaction primarily through adhesive/frictional contact. Another type of engineering applications is characterised by a transformation from a continuum to a discontinuum state. The phenomenon can be found in masonry or concrete structural failure, particle comminution and grinding in ball mills, rock blasting in open and underground mining, fracture of ceramic or glass-like materials under high velocity impact, high speed machining operations, etc. Besides the discrete/discontinuous nature, these applications also have such features as being *highly dynamic*, *rapidly changing domain configurations*, and *multi-physics phenomena*.

The numerical simulation of such discontinuous behaviours, especially at a large scale, has proved to be very challenging and become a subject of increasing research interest.

Although the conventional finite element approach has a powerful capability of modelling the nonlinear response of continua, and significant progress has also been made in the modelling of contact of deformable bodies, the continuum nature of its algorithmic framework imposes restrictions when dealing with fracturing, granular and discontinuous media. Firstly, it is not efficient with the finite element representation of a large number of discrete objects whose deformation is of secondary importance. Secondly, the contact detection procedures available in the FEM are not adequate to handle contact with constantly changing and evolving configurations in an unpredictable manner.

Consequently an alternative numerical approach within a discontinuous framework has been exploited. The discrete element method (DEM), among other discontinuous methodologies such as discontinuous deformation analysis (DDA) and the manifold method, has become a promising numerical tool capable of simulating problems of a discrete or discontinuous nature.

The classic DEM was originated in geotechnical and granular flow applications in the 70's by Cundall and Strack [1]. In its framework a discrete system is considered as an assembly of individual discrete objects which are treated as *rigid* and represented by *discrete elements* as simple geometric entities. The dynamic response of discrete elements depends on the interaction forces which can be short-ranged, such as mechanical contact, and/or medium-ranged, such as attraction forces in liquid bridges, and obey Newton's second law. By tracking the motion of individual discrete objects and handling the interaction of the objects, the dynamic behaviour of a discrete system can essentially be simulated.

Successful applications of the DEM to many practical problems and to increasingly even more diversified areas are evident from the work reported in the proceedings of a series of DEM conferences held over the past decade (see e.g. [2] [3] [4] [5]). Recent developments are accompanied with an increased mathematical rigour and algorithmic robustness in the implementations, which will help the methodology to become more readily recognised and more frequently applied in engineering practice.

Nevertheless, the applicability of the classic DEM is limited when the deformability of certain objects in the system needs to be considered. A particular problem which motivated the current research is the simulation of shot peening and peen forming processes, where both a continuous region (workpiece) and a large number of discrete objects (shot) are simultaneously present. The deformation of the workpiece is a result of a coupled dynamic interaction between the two types of media. It is obvious that a combined finite/discrete element method offers a more powerful solution capability if the shot are modelled by discrete elements and the workpiece is discretised by finite elements. Additional examples that can considerably benefit from this combined solution strategy include fracture damage modelling such as rock blasting, mining applications, and projectile impact.

1.2 Main Building Blocks of the Discrete Element Procedure

In discrete element modelling, an individual object is modelled as a geometrically simplified entity that interacts with other objects through boundary contact. At each time step, objects in contact are identified with a contact detection algorithm; then their contact forces are calculated based on appropriate interaction laws. The motion of each discrete object is governed by Newton's second law. A set of governing equations is built up and integrated with respect to time, to update each object's position, velocity and acceleration. The main building blocks of the discrete element procedure are described as follows.

1.2.1 Representation of discrete objects

In the classic DEM, discrete objects are treated as *rigid* and represented either by regular geometric shapes, such as disks/spheres and superquadrics, or by irregular geometric shapes, such as polygons/polyhedrons or clustering/clumping of regular shapes to form compound shapes. In the combined FE/DE methodology, the concept of discrete elements extends to individual continuous regions which can be further discretised by finite elements if the deformability needs to be considered.

Disk/sphere is the first used and simplest element in the DEM. One of the attractive features of the element stems from its geometric simplicity, smooth and continuous boundary. Contact resolution for this type of discrete element is therefore trivial, and computationally efficient. While this element has attractive geometric properties, it also has limitations in terms of modelling capabilities. Idealising materials such as grains and concrete aggregates as perfect disks (or spheres) is not always realistic and may not produce correct dynamic behaviour [6]. One of the reasons is that this type of element cannot provide resistance to rolling motion. This has lead to the introduction of more sophisticated elements to represent the discrete system more realistically.

Contrary to disk/sphere elements where only the radius can be modified, polygonal elements (polygons/polyhedrons) offer increased flexibility in terms of shape variation. Since the boundary of this type of element is no longer smooth, some complex situations, for instance corner/corner contact, often arise in the contact resolution. Meanwhile, more information is required to define the geometry of a polygonal element.

Higher order discrete elements can be used, such as superquadrics and hyperquadrics as proposed in [7], which may represent many simple geometric entities (for instance, disk, sphere, ellipse and ellipsoid) within the framework. However, this mathematical elegancy may be offset by the expensive computation involved in the contact resolution.

Besides the element shapes, the size distribution of discrete elements may also play an important part in the correct simulation of many practical problems. In addition, the generation of an initial configuration that represents realistic situations becomes a modelling issue with practical importance.

1.2.2 Contact detection

A global search algorithm is periodically performed to build a list of all possible contacts. Since a significant percentage of the total computational effort of a discrete element solution may be associated with this task, a robust and efficient contact detection algorithm is essentially required.

A naive approach would be to perform an exhaustive check for contacts of each discrete object with all other discrete objects. Therefore the order of its algorithmic complexity is $O(N^2)$ with N the total number of discrete objects. When N exceeds a few hundred discrete objects, the operation becomes far too expensive to perform.

Some search algorithms used in general computing technology and computer graphics have been adopted for this purpose. Algorithms such as bucket sorting, heap sorting, quick sorting, binary tree and quadrant tree data structure all originated from general computing algorithms. However, applications of these algorithms in finite element or discrete element (FE/DE) codes need some modifications to meet the requirements of particular FE/DE body representations and the kinematic resolution.

In recent years efficient contact detection algorithms have been developed, for instance, the space cell decomposition; a combination with the binary tree search; the double-ended spatial sorting (DESS); no binary search (NBS); a space filling curve technique; the alternating digital tree (ADT); the augmented spatial digital tree (ASDT) methods; and CGrid (D-Cell). They generally fall into two categories; tree based search and cell/grid based search, which will be reviewed in Chapter 2.

1.2.3 Contact resolution

The identified pairs with potential contact are then kinematically resolved based on their actual shapes. The contact forces are evaluated according to certain constitutive relationship or appropriate physically based interaction laws. In this work, contact interaction laws will be adopted within the framework of the penalty method for all contacting pairs. The penalty method only approximately satisfies the contact displacement constraints for finite values of the penalty coefficients, allowing a small amount of overlap to occur. The maximum overlap occurring during the impact depends on the penalty coefficient selected and the velocities of the objects prior to the impact. Increasing the penalty coefficients will decrease the maximum overlap but an excessively large value may have undesirable effects on the stability of the simulation procedure.

In general, the interaction laws describe the relationship between the overlap and the corresponding repulsive force for a contact pair. For rigid discrete elements, the interaction laws may be developed on the basis of the physical phenomena involved. The Hertz normal contact model that governs elastic contact of two spheres (assumed rigid in the DEM) in the normal direction is such an example. For 'wet' particles the interaction law may include the effects of a liquid bridge. In other cases, adhesion of the contact objects may need to be considered.

For contact pairs associated with finite elements in the combined FE/DE method, a linear spring-like contact model is normally applied, mainly due to its simplicity and also because more complex models usually do not yield any significantly different results. This is due, as will be exploited later, to the fact that the actual deformation of the contact pairs is effectively taken into account by the underlying finite elements.

1.2.4 Time stepping

The configuration of the entire discrete system is evolved by employing an explicit time integration scheme. With this scheme, no global stiffness matrix needs to be formed and inverted, which makes the operations at each time step far less computationally intensive. However, any explicit time integration scheme is only conditionally stable. If the time increment is too large, the errors are magnified, resulting an unstable solution. In order to maintain a stable solution, the time step has to be smaller than a critical value, which is related to the maximum eigenvalue of the governing equations.

Among the explicit schemes, the central difference method is very popular for dynamics and contact/impact since it is a single step scheme and offers a second order accuracy. The procedure is outlined in the following.

Consider the dynamic equilibrium equation

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} = \boldsymbol{F}^{ext} - \boldsymbol{F}^{int} \tag{1.1}$$

where $\boldsymbol{M}$ and $\boldsymbol{C}$ are respectively the mass and damping matrices of the system; and $\boldsymbol{u}, \dot{\boldsymbol{u}}$ and $\ddot{\boldsymbol{u}}$ are respectively the displacement, velocity and acceleration vectors; and $\boldsymbol{F}^{int}$ and $\boldsymbol{F}^{ext}$ are respectively the internal and external forces.

A combination of previous and current midpoint velocities, $\dot{\boldsymbol{u}}_{n-\frac{1}{2}}$ and $\dot{\boldsymbol{u}}_{n+\frac{1}{2}}$, as illustrated in Fig. 1.1, forms the second order difference equation that defines the acceleration at time

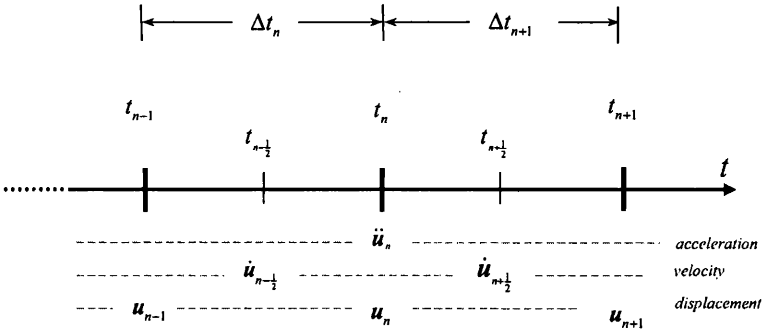


Figure 1.1: Central difference integration scheme

t_n as

$$\ddot{\boldsymbol{u}}_n = \ddot{\boldsymbol{u}}(t_n) = \frac{\dot{\boldsymbol{u}}_{n+\frac{1}{2}} - \dot{\boldsymbol{u}}_{n-\frac{1}{2}}}{\Delta t_{n+\frac{1}{2}}} \tag{1.2}$$

Through the direct integration of the acceleration term with respect to the time increment, the velocity at time t_n can be approximated as

$$\dot{\boldsymbol{u}}_n = \dot{\boldsymbol{u}}(t_n) = \frac{\boldsymbol{u}_{n+\frac{1}{2}} - \boldsymbol{u}_{n-\frac{1}{2}}}{\Delta t_{n+\frac{1}{2}}} = \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_{n-1}}{2\Delta t_{n+\frac{1}{2}}} = \frac{\Delta t_{n+1}\dot{\boldsymbol{u}}_{n+\frac{1}{2}} + \Delta t_n \dot{\boldsymbol{u}}_{n-\frac{1}{2}}}{2\Delta t_{n+\frac{1}{2}}} \tag{1.3}$$

where the midpoint velocities are defined as

$$\dot{\boldsymbol{u}}_{n-\frac{1}{2}} = \dot{\boldsymbol{u}}(t_{n-\frac{1}{2}}) = \frac{\boldsymbol{u}_n - \boldsymbol{u}_{n-1}}{\Delta t_n} \qquad \dot{\boldsymbol{u}}_{n+\frac{1}{2}} = \dot{\boldsymbol{u}}(t_{n+\frac{1}{2}}) = \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t_{n+1}} \tag{1.4}$$

and the time increments are calculated as

$$\Delta t_n = t_n - t_{n-1} \qquad \Delta t_{n+1} = t_{n+1} - t_n \qquad \Delta t_{n+\frac{1}{2}} = \frac{1}{2}(\Delta t_n + \Delta t_{n+1}) \tag{1.5}$$

By substituting Eqs. (1.2) and (1.3) into (1.1) the velocity at time $t_{n+\frac{1}{2}}$ is determined in terms of known (previous) velocity, displacement and force quantities,

$$\dot{\boldsymbol{u}}_{n+\frac{1}{2}} = \dot{\boldsymbol{u}}(t_{n+\frac{1}{2}}) = \left[2\boldsymbol{M} + \boldsymbol{C}\Delta t_{n+1}\right]^{-1}\left[(2\boldsymbol{M} - \boldsymbol{C}\Delta t_n)\dot{\boldsymbol{u}}_{n-\frac{1}{2}} + 2\Delta t_{n+\frac{1}{2}}(\boldsymbol{F}^{ext} - \boldsymbol{F}^{int})\right] \tag{1.6}$$

Thus the displacement at time t_{n+1} can be calculated as

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}(t_{n+1}) = \boldsymbol{u}_n + \dot{\boldsymbol{u}}_{n+\frac{1}{2}}\Delta t_{n+1} \tag{1.7}$$

and the force vector is updated by

$$\boldsymbol{F}^{ext}_{n+1} = \boldsymbol{F}^{ext}(t_{n+1}) \qquad \boldsymbol{F}^{int}_{n+1} = \boldsymbol{F}^{int}(\boldsymbol{u}_{n+1}) \tag{1.8}$$

Since displacements and velocities are computed at different time instances in the central difference algorithm, a starting scheme is requirel to provide an initial value for velocity at $t = \frac{1}{2}\Delta t_1$. The following procedure is used to conpute $\dot{\boldsymbol{u}}_{\frac{1}{2}}$ in this work:

$$\dot{\boldsymbol{u}}_{\frac{1}{2}} = \boldsymbol{u}_0 + \boldsymbol{M}^{-1}(\boldsymbol{F}^{ext} - \boldsymbol{F}^{int} - \boldsymbol{C}\dot{\boldsymbol{u}}_0)\frac{\Delta t_1}{2} \tag{1.9}$$

where $\boldsymbol{u}_0$ and $\dot{\boldsymbol{u}}_0$ are the initial displacements anl velocities respectively.

For a linear system the critical time step can be cvaluated as [8]

$$\Delta t_{cr} = \frac{2}{\omega_{nax}} \tag{1.10}$$

where ω_{max} is the maximum eigenvalue of the system. However, the above result may not be valid since an impact system is generally nonlinear, as will be demonstrated in Chapter 3.

1.3 Objectives of the Research

This research is derived primarily from the need tc develop numerical solution procedures for the simulation of industrial shot peening and peen forming operations. To successfully pursue the goal and possibly extend the methodology to more broad applications, it is necessary to enhance and advance some algorithmic aspects of the general discrete element approach.

The main objectives of the research are therefore

- to locate a fast contact detection algorithm br large scale discrete element simulations;
- to formulate contact resolution for regular and irregular discrete objects, with special attention given to the investigations of the commonly used interaction laws in the DEM community;
- to develop an efficient approach for the preparation of an initial random discrete object configuration;
- to establish a general algorithmic framework respectively for the combined finite/discrete element analysis of shot peening and peen forming processes.

1.4 Layout of the Thesis

Chapter 1 provides a general overview of the discrete element method, and the main building blocks of the discrete element procedure.

Chapter 2 reviews the contact detection algorithms for applications involving large numbers of discrete objects, among which the alternating digital tree (ADT) and the augmented spatial digital tree (ASDT) are presented as examples of tree based search algorithms, whilst the dynamic cell based search algorithm, or *'D-cell'* for short, is presented as an example of cell/grid based search algorithms. Extensive numerical tests are provided for the performance comparison of the algorithms, and for the parametric study of the 'D-cell' itself.

Chapter 3 is devoted to a comprehensive discussion on contact resolution for circular and spherical discrete objects. With the establishment of geometric relations between contact pairs, four interaction laws commonly used in the DEM community for the normal contact are reviewed, together with a general analysis of the equivalence among them. A modified classic Coulomb friction model is proposed for 3D contact problems. In addition, a new time stepping scheme is also developed. Finally, two different schemes that incorporate energy dissipations into the contact model are reviewed.

Chapter 4 introduces a new approach to perform contact resolution for 2D superquadrics, which is accomplished by firstly representing any superquadric with a convex polygon through adaptive sampling; then clipping two polygons to identify the overlap zone; and employing an efficient linear algorithm to search for intersections and overlap area of the clipped polygons; finally utilising a well-established polygon/polygon contact framework to determine the contact forces and directions.

Chapter 5 describes an advancing front based algorithm with two different implementation versions, closed form and open form, for the generation of a random packing of disks, polygons and ellipses of different sizes within a given geometric domain.

Chapter 6 proposes a geometric based compression algorithm to generate a random packing of spheres of various sizes within a given geometric domain, in which the compression strategies, namely global compression, random compression and shaking, are detailed. The efficiency of the algorithm is demonstrated via numerical examples.

Chapter 7 establishes a general algorithmic framework for the combined finite/discrete element analysis of shot peening processes, where a general description of the combined finite/discrete element method is given first, followed by 2D plane strain analysis and 3D modelling of the problem. Special attention is given to the evaluation and comparison of the interaction laws governing the contact of shot and workpiece.

Chapter 8 presents a two stage combined finite/discrete element and explicit/implicit solution procedure for the simulation of peen forming processes, which is accomplished by identifying the residual profiles through shot peening a small region of a workpiece with explicit dynamic analysis, followed by employing the implicit static solution to obtain the deformation and stress distribution of the entire workpiece.

Chapter 9 presents the conclusions and recommendations for further research.

Bibliography

[1] P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29:47–65, 1979.

[2] Graham G. W. Mustoe. *1st US conference on discrete element methods,* Colorado School of Mines, Colorado, USA - October 1989.

[3] John R. Williams. *2nd US conference on discrete element methods,* M.I.T. Cambridge, USA - March 1993.

[4] Nenad Bićanić. *4th international conference on analysis of discontinuous deformation,* University of Glasgow, Glasgow, UK - June, 2001.

[5] Benjamin K. Cook and Richard P. Jensen. *3rd international conference on discrete element methods,* Santa Fe, New Mexico, USA - September 2002.

[6] V. Poulopoulos. *A C++ implementation of the discrete element method.* MSc Thesis, University of Wales Swansea, 2002.

[7] J. R. Williams and R. O'Connor. A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries. *Eng. Comp.*, 12:185-201, 1995.

[8] Ted Belytschko, Wing Kam Liu and Brian Moran. *Nonlinear finite elements for continua and structures.* John Wiley & Sons Ltd, 2000.

Chapter 2

Contact Detection

2.1 Introduction

In the discrete element simulation of problems with a large number of discrete objects, as much as 60-70% of the computational time could be spent in detecting and tracking the contact between discrete objects. Due to a large diversity of object shapes used in the DEM, many efficient contact processing algorithms often adopt a two-phase solution strategy. The first phase, termed *contact detection* or *global search*, identifies the number of objects which are considered as potential contactors of a given object. In this phase all the objects are approximated by simple geometric entities, such as rectangles or spheres. The second phase, termed *contact resolution* or *local search*, resolves the details of the contact pairs based on their geometric shapes. The following discussions are restricted to the contact detection, while the contact resolution will be presented in the next two chapters.

As mentioned in Chapter 1, efficient contact detection algorithms have been developed in recent years (see e.g. [1], [2], [3], [4], [5], [6], [7], [8]). Contact detection algorithms for applications involving large numbers of discrete objects generally fall into two categories; tree based search and cell/grid based search. The alternating digital tree (ADT) [6] and the augmented spatial digital tree (ASDT) [7] will be used here as examples of tree based search algorithms, whilst the CGrid [8], termed the dynamic cell based search algorithm in this work, is to be presented as an example of cell/grid based search algorithms. A review of tree based search algorithms can be found in the paper by Feng and Owen [7].

2.2 Body representation

In order to accommodate different shapes of discrete objects, any arbitrarily shaped object is represented with an axis aligned bounding box which is possibly further extended by a buffer zone. The contact detection problem can be stated as: determine all the members in a set of N rectangles in a $n-$dimensional Euclidean space $\boldsymbol{E}^n$ that overlap with a given (target) rectangle.

Consider a rectangle in $\boldsymbol{E}^n$, whose sides are parallel to the axes $x^1, x^2, \cdots, x^n$. Suppose that a^i, b^i $(i = 1, ..., n)$ are respectively the minimum and maximum coordinates of the rectangle in the x^i direction. Then the corresponding length of the side in this direction, h^i, is defined by $h^i = b^i - a^i$. In addition, points $\boldsymbol{a} = \{a^1, a^2, \cdots, a^n\}$ and $\boldsymbol{b} = \{b^1, b^2, \cdots, b^n\}$ are the two characteristic vertices of the rectangle.

Rectangles can be represented in different ways [9], for instance, by their characteristic parts or points. The point-based method is adopted here, which reduces each rectangle to a point $\boldsymbol{p} = \{p^1, p^2, \cdots\}$ but normally in a higher $(2n)$ dimensional space. As an example, a segment in 1D space is represented as a point in 2D space (Fig. 2.1). This point is termed the *representative point* of the rectangle, and can be defined by more than one set of parameters, such as

1. two characteristic points $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e. $\boldsymbol{p} = \{\boldsymbol{a}, \boldsymbol{b}\}$;
2. one of the characteristic points and side lengths $\boldsymbol{h}$, i.e. $\boldsymbol{p} = \{\boldsymbol{a}, \boldsymbol{h}\}$; or $\boldsymbol{p} = \{\boldsymbol{b}, \boldsymbol{h}\}$;
3. the centroid of the rectangle and n half-lengths $\boldsymbol{h}/2$, i.e. $\boldsymbol{p} = \{(\boldsymbol{a} + \boldsymbol{b})/2, \boldsymbol{h}/2\}$;

In the first case, the representative point is described by two sets of location parameters; while in the second and third cases, it is defined in terms of one set of location parameters and one set of dimension parameters.

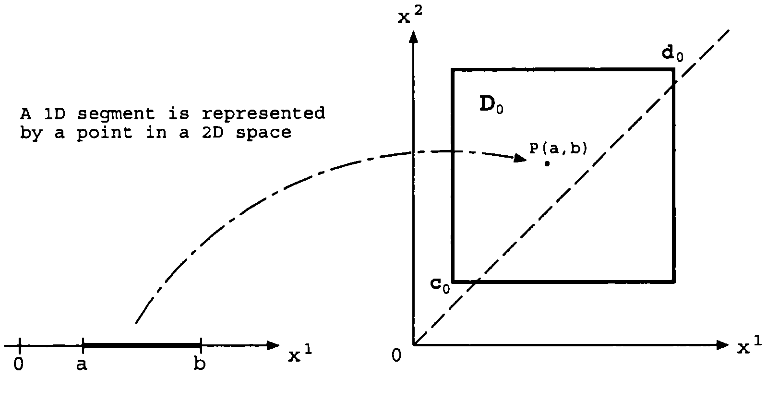


Figure 2.1: Point representation scheme and initial embedding region in space $\boldsymbol{E}^{2n}$

2.3 Tree Based Search Algorithms: ADT and ASDT

2.3.1 Spatial binary tree structure for a set of rectangles

The data representation of rectangles considered in both ADT and ASDT is a spatial binary tree structure which is based on the principle of recursive bisection of the embedding space.

A binary tree is a non-sequential type of data structure. Each node in the tree can have two children: left child and right child, and, apart from the root node, also has a parent. A node without any child is termed a *leaf.* Each node also has a *key* field that decides which of its children will be accessed during the tree traversal. With the root being level 0, each node is assigned a hierarchy level that is defined as one more than the level of its parent. The level of a node indicates the path length from the root to this node.

A spatial binary tree is a binary tree in which each node is associated with a spatial subregion in addition to the original data stored. The primary operations associated with the binary tree include *insertion, deletion*, and the *region search.* The costs of these operations are largely influenced by the shape of the tree. Generally, the operations can be undertaken in $O(N \ln N)$ time for well balanced trees, while poor performance is to be expected from highly degenerated trees [12]. The quality of a tree may be indicated by the *total length/level* of the tree, which is defined as the sum of all the node levels in the tree.

2.3.2 Alternating digital tree algorithm

The alternating digital tree (ADT) algorithm adopts the first point representation scheme, as described earlier, to represent rectangles, and a spatial binary tree to represent the collection of rectangles.

Tree generation

Consider a set of N points, $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N\}$, in a $2n$−dimensional space $\boldsymbol{E}^{2n}$, which represent N rectangles in $\boldsymbol{E}^n$. Assume that $\boldsymbol{D}_0$ is the minimum rectangular domain that contains all the representative points in $\boldsymbol{E}^{2n}$. $\boldsymbol{D}_0$ can also be represented by its two corner points, $\boldsymbol{c}_0$ and $\boldsymbol{d}_0$, as $(\boldsymbol{c}_0, \boldsymbol{d}_0)$.

The generation of a spatial binary tree begins by selecting, for instance, the first point $\boldsymbol{p}_1$ in the given list as the root node which represents the whole region $\boldsymbol{D}_0$. The region is now bisected across the x^1 direction and $x_0^1 = (c_0^1 + d_0^1)/2$, is taken as the key for the root node. The second point $\boldsymbol{p}_2$ is inserted to the left child of the root if the condition $a_2^1 < x_0^1$ holds, and the left half region of $\boldsymbol{D}_0$, $c_0^1 \leq x^1 < x_0^1$, is also assigned to the node. Otherwise, this

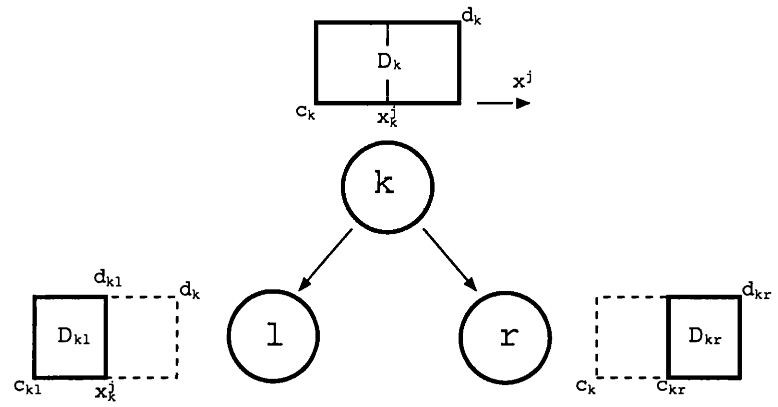


Figure 2.2: Associated subregion bisection

point is inserted to the right child, and the right half region $x_0^1 \leq x^1 \leq d_0^1$ is assigned to the node. At each of these nodes, the represented subregion is further bisected across the x^2 direction and $x_1^2 = (c_1^2 + d_1^2)/2$ is chosen as the key for both nodes.

Generally, if a node m at the hierarchy level k represents a region $\boldsymbol{D}_k = (\boldsymbol{c}_k, \boldsymbol{d}_k)$, the half subregions associated with its left and right children, $\boldsymbol{D}_{kl} = (\boldsymbol{c}_{kl}, \boldsymbol{d}_{kl})$ and $\boldsymbol{D}_{kr} = (\boldsymbol{c}_{kr}, \boldsymbol{d}_{kr})$, result from the bisection of $\boldsymbol{D}_k$ by a plane normal to the jth coordinate axis, where j is chosen cyclically from the $2n$ directions in $\boldsymbol{E}^{2n}$ as

$$j = 1 + \mathrm{mod}(k, 2n) \tag{2.1}$$

Consequently, the key for the node m is set to be $x_k^j = (c_k^j + d_k^j)/2$ and $\boldsymbol{c}_{kl}, \boldsymbol{d}_{kl}$ and $\boldsymbol{c}_{kr}, \boldsymbol{d}_{kr}$ are respectively defined as (see Fig. 2.2)

$$\begin{array}{llll} c_{kl}^i = c_k^i, d_{kl}^i = d_k^i & \text{for } i \neq j & \qquad c_{kl}^j = c_k^j, d_{kl}^j = x_k^j \\ c_{kr}^i = c_k^i, d_{kr}^i = d_k^i & \text{for } i \neq j & \qquad c_{kr}^j = x_k^j, d_{kr}^j = d_k^j \end{array} \tag{2.2}$$

Similarly, the insertion of a point $\boldsymbol{p}_i$ in the list to the tree can be undertaken in a recursive manner. Starting from the root node, the tree is traversed downward until an appropriate unfilled node is found. At a node of level k, its left child is chosen to be visited next if the condition $p_i^j < x_k^j$ is satisfied (where x_k^j is the key of the node and j as determined by (2.1)); otherwise its right child should be visited. The tree is completed when all the points in the list are inserted in the tree.

Region Search

A necessary step in the region search is to define the search region (rectangle) in $\boldsymbol{E}^{2n}$. Assume that $\overline{\boldsymbol{x}}_{min}$ and $\overline{\boldsymbol{x}}_{max}$ are two corner points of the initial target rectangle in $\boldsymbol{E}^n$ and therefore can also be represented by a point, $\overline{\boldsymbol{p}} = (\overline{\boldsymbol{x}}_{min}, \overline{\boldsymbol{x}}_{max})$ in $\boldsymbol{E}^{2n}$. In order for a rectangle in $\boldsymbol{E}^n$, represented by a point $\boldsymbol{p} = (\boldsymbol{a}, \boldsymbol{b})$ in $\boldsymbol{E}^{2n}$, to overlap with the target rectangle, the following conditions must be satisfied:

$$\boldsymbol{a} \leq \overline{\boldsymbol{x}}_{max}, \quad \boldsymbol{b} \geq \overline{\boldsymbol{x}}_{min} \tag{2.3}$$

or

$$a^i \leq \overline{x}^i_{man}, \quad b^i \geq \overline{x}^i_{min} \quad (i = 1, \cdots, n) \tag{2.4}$$

As the rectangles concerned are within the region $\boldsymbol{D}_0 = (\boldsymbol{c}_0, \boldsymbol{d}_0)$ condition (2.3) can be rewritten as

$$\boldsymbol{c}_{0,min} \leq \boldsymbol{a} \leq \overline{\boldsymbol{x}}_{max}, \quad \overline{\boldsymbol{x}}_{min} \leq \boldsymbol{b} \leq \boldsymbol{d}_{0,max} \tag{2.5}$$

where $\boldsymbol{c}_{0,min}$ and $\boldsymbol{d}_{0,max}$ are respectively the first and last n components of $\boldsymbol{c}_0$ and $\boldsymbol{d}_0$. Geometrically, the above condition defines a rectangular search region $\boldsymbol{S}$ in $\boldsymbol{E}^{2n}$, whose lower and upper corners are respectively $(\boldsymbol{c}_{0,min}, \overline{\boldsymbol{x}}_{min})$ and $(\overline{\boldsymbol{x}}_{max}, \boldsymbol{d}_{0,max})$. Any points lying in this region means that their represented rectangles overlap with the target rectangle. The aim of the region search step is to identify all the rectangles whose representative points lie within the rectangular region $\boldsymbol{S}$, based on the binary tree structure created in the previous step. Fig. 2.3 depicts the search region for a target segment.

The search procedure can be accomplished by a *preorder* tree traversal, starting from the root node. At any traversed node k, if no overlap is detected between its associated region $\boldsymbol{D}_k$ and the search region $\boldsymbol{S}$, the complete set of points stored in the subtree rooted at k can be excluded from further search. Only when overlap occurs between these two regions, the point store in the node and its two subtrees need to be checked for possible overlap. The procedure is graphically illustrated in Fig. 2.4.

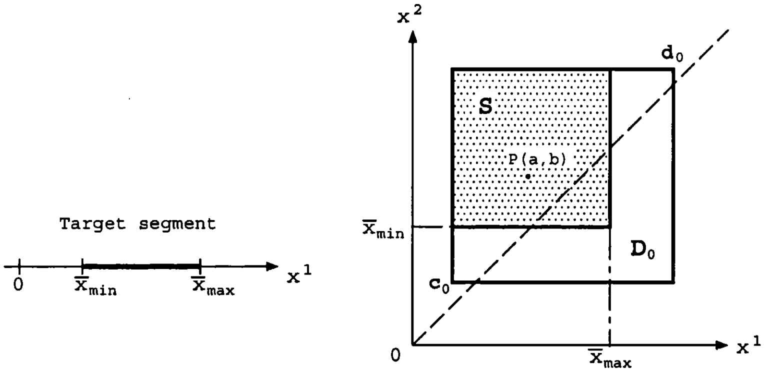


Figure 2.3: The search region (shaded) in $\boldsymbol{E}^2$ for a target segment

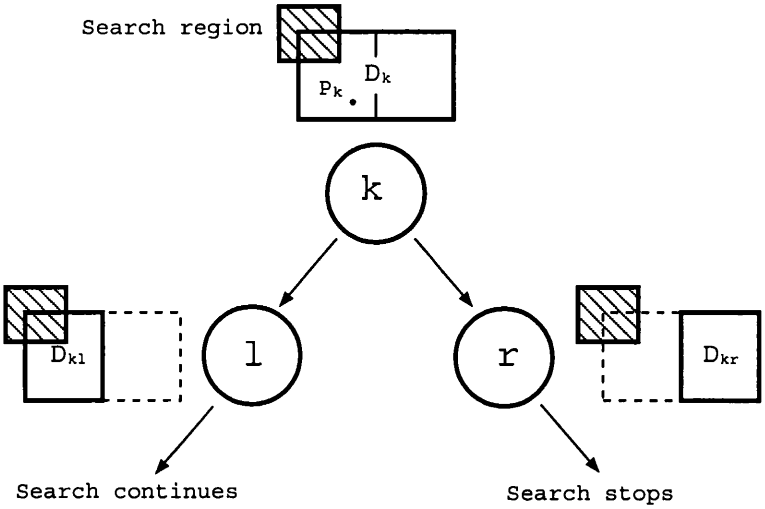


Figure 2.4: Basic search step

Possible performance enhancements

The performance of the ADT algorithm described above can be further enhanced.

Tightening of the associated subregions. Although each subregion associated with a node in the tree contains all the points stored in the node and all its descendants, it may not be the bounding region of these points, except for the initial region $\boldsymbol{D}_0$. For a leaf node the associated subregion contains only the point stored in the leaf. This suggests that the associated subregions in the tree nodes may be slightly larger than necessary, resulting in some unnecessary steps in the search procedure. The problem can be easily solved by tightening the subregions down to the actual bounding domains of the points stored in the nodes and their two subtrees. The tightening can be achieved by employing a *postorder* tree traversal. For a leaf node, the associated region is simply reduced to the point stored. For a non-leaf node k, the associated bounding region $\overline{\boldsymbol{D}}_k$ can be symbolically determined by

$$\overline{\boldsymbol{D}}_k = \overline{\boldsymbol{D}}_{kl} \cup \overline{\boldsymbol{D}}_{kr} \cup \boldsymbol{p}_k \tag{2.6}$$

where $\overline{\boldsymbol{D}}_{kl}$ and $\overline{\boldsymbol{D}}_{kr}$ are respectively the bounding subregions associated with its left and right children; and $\boldsymbol{p}_k$ is the point store in the node k. See Fig. 2.5 for an illustration.

Alternative point representation scheme. In most applications, the rectangles are small compared with the whole region that contains all the rectangles. Therefore $\boldsymbol{c}_i$ and $\boldsymbol{d}_i$ are close in value, which, together with the relation $\boldsymbol{d}_i = \boldsymbol{c}_i + \boldsymbol{h}_i$, indicates that the representative points are clustered near and above the diagonal and thus not well distributed [9]. In fact, all

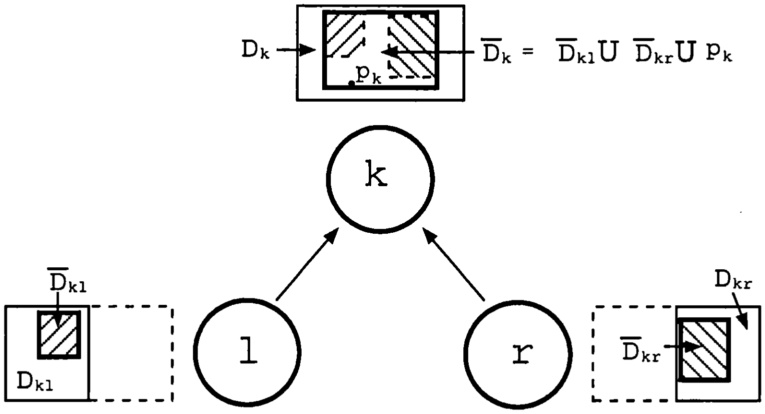


Figure 2.5: Tightening of associated subregions by postorder tree traversal

the points lie in a strip region bounded by the minimum and maximum sizes of the rectangles. The built binary tree with ADT algorithm may be significantly degenerated if no tree balancing is performed. This observation motivates the proposal of the ASDT algorithm [7].

2.3.3 Augmented spatial digital tree algorithm

Point representation scheme

In the augmented spatial digital tree (ASDT) algorithm, a rectangle is represented only by its lower corner point, i.e. $\boldsymbol{p} = \boldsymbol{a}$, and the coordinates of upper corner point, $\boldsymbol{b}$, are regarded as the augmented information to the representative point, which is now present in the same space, $\boldsymbol{E}^n$, where the original rectangle resides.

In general, the same binary tree generation procedure as in the ADT can be applied. However, since the dimension of the embedding space is now reduced from $2n$ in the ADT to n, all the subregions associated with tree nodes are n-dimensional, which are obtained by consecutively bisecting the bounding region of all points, $\widetilde{\boldsymbol{D}}_0$, in n directions in a cyclic order. In addition, as the bounding region is much smaller than that in the ADT, while the total number of points remains the same, the space is generally filled more uniformly. Consequently, the resulting binary tree may be much better balanced compared to that in the ADT.

Additional bounding subregions

The key to the ASDT algorithm is that any node k is additionally assigned a subregion, $\boldsymbol{E}_k$, which is the bounding rectangular domain of all the rectangles represented by the points stored in the subtree rooted at node k. These bounding domains can be determined by a

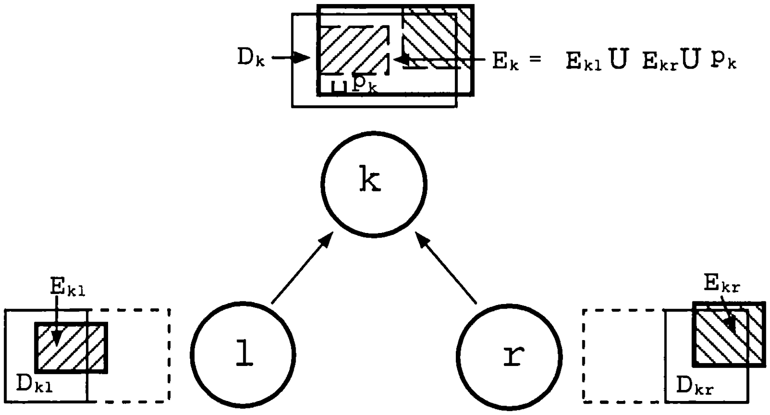


Figure 2.6: Determination of additional bounding region $\boldsymbol{E}_k$ by a postorder tree walk

postorder tree traversal, utilising the upper corner points of the rectangles. At a leaf node, the region is set to the rectangle represented by the point stored, i.e $\boldsymbol{E}_k = \boldsymbol{p}_k$. Then at each internal node, $\boldsymbol{E}_k$ can be determined, as shown in Fig. 2.6, by

$$\boldsymbol{E}_k = \boldsymbol{E}_{kl} \cup \boldsymbol{E}_{kr} \cup \boldsymbol{p}_k \tag{2.7}$$

where $\boldsymbol{E}_{kl}$ and $\boldsymbol{E}_{kr}$ are the regions associated with its left and right children and $\boldsymbol{p}_k$ is the point stored in the node. The procedure takes N steps and ends at the root node. The computed bounding region at the root node, $\boldsymbol{E}_0$, is the bounding rectangle of all the rectangles concerned.

These additional subregions, $\boldsymbol{E}_k$, can be represented by their two corner vertices and could overwrite where the regions $\widetilde{\boldsymbol{D}}_k$ are stored, thereby no extra space may be required. More importantly, these regions are used in the subsequent search phase to serve the same role as the regions $\boldsymbol{D}_k$ play in the ADT. $\boldsymbol{E}_k$ is generally very similar to $\overline{\boldsymbol{D}}_k$ but in a much lower space. As $\boldsymbol{E}_k$ is no longer obtained by bisecting its parent region in one coordinate direction, a maximum n number of coordination comparisons are required at each node in the search step. Note that the same number of coordinate values as in the ADT is needed to be stored at each node.

In summary, the ASDT algorithm adopts a different point representation scheme that uses only the lower corner vertex to represent a rectangle, with the upper corner vertex serving as the augmented information. As a result, the ASDT algorithm can work on the original n-dimensional space and in general a well balanced tree can be expected. This, together with the introduction of an additional bounding subregion, reduces significantly the number of node visits in the region search. The numerical comparisons conducted in [7] indicate that the ASDT is more than four times faster than the ADT.

It is noted that the memory requirement of both ADT and ASDT are proportional to N which is optimal, while the total operations T_{op} may be expressed as

$$T_{op} = O(N \ln N)$$

2.4 Cell/Grid Based Search Algorithms

The main procedures behind cell based search algorithms are: (1) to divide the domain that the discrete objects occupy into regular cells/grids; (2) to map each object to one of the grid cells; and (3) for each object in a cell, to check for possible contacts with objects in the same cell and in the immediate neighbouring cells. The difference among various algorithms lies how the mapping is represented and the check is performed.

2.4.1 Static cell based algorithm (S-Cell)

The static cell based algorithm (S-Cell) is one of the first efficient search algorithms employed in the DEM simulation.

Suppose a system consists of N circular disks occupying a bounding space. Firstly, the space is virtually subdivided into n_x cells in the x direction and n_y cells in the y direction. An array of $n_x \times n_y$ is required to store information for the cells. Then each disk is mapped onto a grid cell according to its central coordinates. Within each cell, a list data structure is normally used to store the disks. Note that some cells may have no disks assigned.

For each cell, a contact check among the disks in the cell and against disks in the neighbouring cells is performed. Utilising the contact symmetry between two disks can reduce the number of the immediate neighbouring cells from 8 to 4. Upon looping over all the cells, the contacting pairs are readily identified.

The above search procedure is simple to implement, and can be extended to rectangles. The memory requirement is approximately proportional to $(n_x n_y + N)$, while the CPU cost is difficult to estimate precisely for general cases. It is clear that both the memory and CPU costs are strongly dependent on the size and spatial distributions of the objects. In the case of similarly sized and densely packed objects in a (nearly) rectangular region, the procedure has a nearly linear complexity and thus can be very effective. However, the performance can be severely deteriorated if the object size is largely varied, and/or the objects are scattered in an irregular domain. As a consequence, the static cell algorithm would be an ideal contact detection algorithm for particle compaction problems with similar size, but more likely result in a computational bottleneck for high speed impact problems [13].

2.4.2 No Binary Search (NBS)

A so-called *no binary search* (NBS) contact detection algorithm [4] is essentially designed to overcome some of the drawbacks of the S-cell algorithm. The key algorithmic aspects of the NBS include: (1) A special list type data structure is designed to avoid the use of an array for all the cells, so the memory requirement is almost proportional to N regardless of the spatial distribution of the objects. (2) A dynamic cell search avoids looping over all the cells. Therefore the CPU cost, if the objects are of similar size, is linear,

$$T_{op} = O(N)$$

Consequently, the NBS can exhibit a superior performance, particularly for large scale problems, to the general tree based search algorithms and the static cell algorithm. It is, however, limited by its applicability to systems comprising objects of similar size.

2.4.3 Dynamic cell based search algorithm (D-Cell)

The dynamic cell based search algorithm 'CGrid' [8], or the *'D-cell'* for short, is primarily designed to overcome the limitation of NBS and therefore can be applied to systems comprising discrete objects of various sizes. For the convenience of illustrating the concepts of D-cell, a 2D case is considered although extension to 3D problems is straight forward. Note that the D-cell algorithm is presented here in an easier understandable way than its original form and is implemented differently.

Mapping of discrete objects to cells

Assume a system consists of N discrete objects, represented by N bounding boxes of different sizes occupying a bounding space $[x_{min}, y_{min}] \times [x_{max}, y_{max}]$. In the preliminary step, the space is virtually subdivided into n_x cells in the x direction and n_y cells in the y direction:

$$n_x = \mathrm{Int}\Big[\frac{x_{max} - x_{min}}{x_m}\Big], \quad n_y = \mathrm{Int}\Big[\frac{y_{max} - y_{min}}{y_m}\Big]$$

where x_m and y_m are user specified cell sizes in x and y directions respectively. Then each discrete object is mapped onto an integer grid cell (i_x, i_y) according to its lower bounding box coordinates (x_l, y_l):

$$i_x = \mathrm{Int}\Big[\frac{x_l - x_{min}}{x_m}\Big], \quad i_y = \mathrm{Int}\Big[\frac{y_l - y_{min}}{y_m}\Big]$$

Thus each object is assigned to one and only one cell. An example is given in Fig. 2.7, where for instance, disk 1 is assigned to the cell (3,4), and disk 2 is assigned to the cell (3,2).

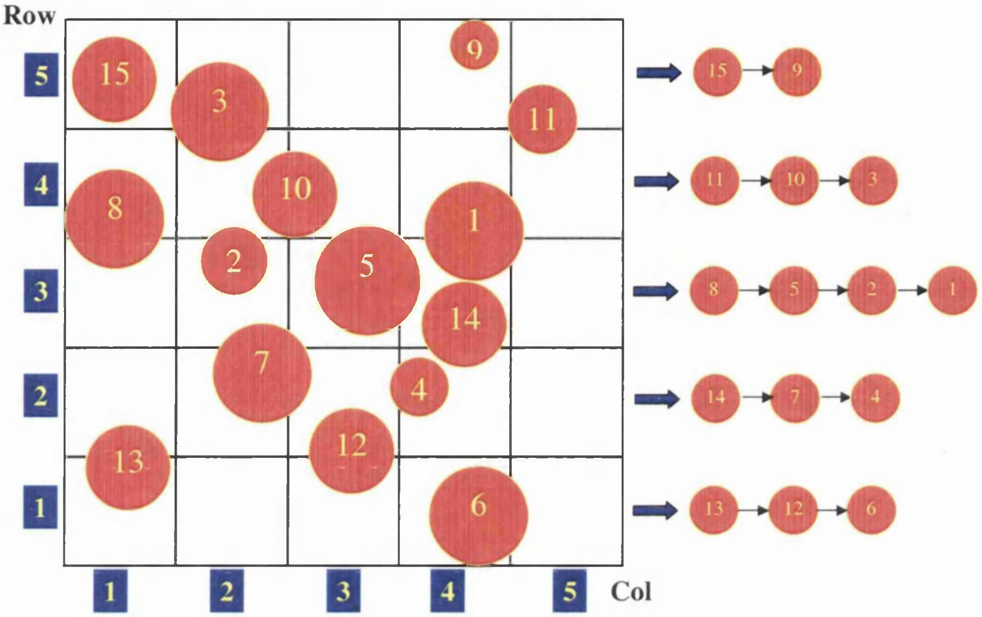


Figure 2.7: Mapping of discrete objects onto cells and the linked list

An object is said to be mapped to a particular row if it is mapped to any cell of that row. Equivalently, an object is said to be mapped to a particular column if it is mapped to any cell of that column.

Representation of the mapping

The obvious way to represent the mapping is the use of a 2D array of size $n_x \times n_y$, as done in the S-cell. This is extremely expensive in terms of RAM requirements, especially for loose packing where the total number of discrete objects may be much smaller than the total number of cells. Similar to the NBS, linked lists are used here to avoid empty cells so as to reduce memory requirements significantly. All the objects are mapped first to the rows of cells and a singly connected list $Y_{i_y}(i_y = 1, ..., n_y)$ is established to represent objects in the i_y-th row. Objects are mapped by looping over all objects in ascending numerical order. In Fig. 2.7, list Y_3 (for Row 3) is formed by placing disk 1 onto the list, and then pushing it by disk 2, which is pushed by disk 5, lastly by disk 8.

The rows are processed sequentially starting from the first non-empty row, and the cells of the row are then handled from the left to the right. Suppose that a non-empty row (e.g. Row 3) is the row being processed with the corresponding linked list $Y_{i_y}(Y_3)$. As will be explained later, $Y_{i_y}(Y_3)$ may have been updated at the previous step, with objects whose upper-bound y-coordinates y_u are greater than the lower-bound y-coordinate of the current row. The modified linked list Y_3 is:

$$Y_3 = \{1, 2, 5, 8, 7, 14\}$$

where disks 7 and 14 are migrated from previous row (Row 2) since their upper-bound y-coordinates are greater than the lower y-coordinate of Row 3. Then the objects in list $Y_{i_y}(Y_3)$ are mapped onto cells of the row by building the linked list $X_{i_x}(i_x = 1, ..., n_x)$ according to their integerised coordinates i_x. A particular (X_{i_x}, Y_{i_y}) list contains all objects with integerised coordinates i_x and i_y. For example, (X_4, Y_3) contains disks 1 and 14, while (X_5, Y_3) is empty. In addition, an ordered list is maintained to provide a sequential list of non-empty cells in the row.

Starting from the first cell in the ordered list, the cells in the row are processed sequentially following the order in the list; therefore empty cells can be avoided. For the first non-empty cell i_1, the corresponding list X_{i_1} may comprise objects originally in the cell and those migrated from the previous row. Overlap checks are performed among the original objects and against the migrated ones. No checks are needed, however, for the updated objects since they have been checked in previous rows. Before proceeding to the next cell, all the objects in the current cell are examined so that those with larger upper-bound x-coordinates than the lower-bound x-coordinate of the next cell in the ordered list are migrated and appended to the object list of the next cell. Therefore for objects in Row 3 (Fig. 2.7), disk 5 will be migrated to cell 4 and disk 1 will be migrated to cell 5. Consequently, the objects in the new cell to be processed may have four different types of membership:

- Type 00: objects originally in the cell
- Type 01: objects originally in the row and migrated from the previous cell
- Type 10: objects originally in the previous rows and migrated directly to the cell;
- Type 11: objects originally in the previous rows, first migrated to the previous cells and then migrated to the cell from the previous cell.

Overlap checks

With each object being one of the above four types, overlap checks in the cell are performed based on the following three cases, as illustrated in Fig. 2.8:

1) Among objects of type 00;
2) Type 00 objects against objects of other types;

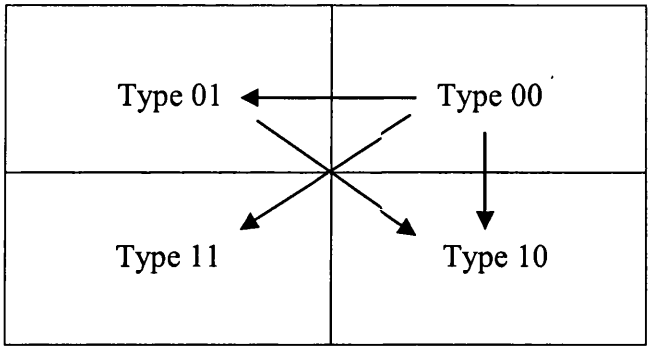


Figure 2.8: Overlap checks among different types of objects

3) Type 01 objects against type 10 objects.

The row is completely processed when the last non-empty cell is reached. Before going to the next row, all the objects in the row, including the original and migrated ones, are checked so that any object with a greater upper-bound y-coordinate than the lower-bound y-coordinate of the next row is migrated to the new row.

The D-cell contact detection algorithm is summarised in Box 2.1.

It should be noted that mapping discrete objects to the columns first and processing columns sequentially is an equivalent choice to processing rows sequentially as described above.

Provided the number of cell columns/rows is significantly less than the number of discrete objects, it can be proved that the memory requirement for the D-cell algorithm is $O(N)$. Also for a fixed cell size the computational time T_{op} may be expressed as

$$T_{op} = O(N + \epsilon)$$

where ϵ represents the costs associated with the maintenance of various lists used in the algorithm.

Although the cell size can be user-defined and different from the discrete object sizes, the performance of the algorithm is strongly dependent on the cell dimension selected. The optimal cell size is not easy to determine analytically since it may depend on several factors. Numerical experiments are therefore necessary to derive an empirical formulation or practical guidance for the selection of a (nearly) optimal cell size.

2.5 Numerical Tests

A number of numerical tests are provided: (1) to compare the efficiency of the D-cell algorithm with the ASDT and NBS algorithms; and (2) to perform parametric study of the D-cell

Box 2.1: The D-cell Contact Detection Algorithm

```
• Specify cell sizes in $x$ and $y$ directions, $x_m, y_m$;
• Evaluate minimum and maximum coordinates of space $[x_{min}, y_{min}] \times [x_{max}, y_{max}]$, and divide into cells:
  $n_x = \mathrm{Int}\left[\frac{x_{max}-x_{min}}{x_m}\right], \quad n_y = \mathrm{Int}\left[\frac{y_{max}-y_{min}}{y_m}\right]$
• Build $Y_{iy}(i = 1, ..., n_y)$ list;
• Contact detection:
    – Sort out non-empty rows $nn_y$;
    – Loop over non-empty rows $i = 0, ..., nn_y$
      {
        * if $(i > 0)$, migrate boxes from previous row(s) $(0, ..., i-1)$ with upper-bound $y$-coordinates $y_u$ greater than the lower-bound $y$ coordinate of the current row;
        * Build $X_{ix}(i = 1, ..., n_x)$ list;
        * Sort out non-empty cells $nn_x$;
        * Loop over non-empty cells $j = 0, ..., nn_x$
          {
            · if $(j > 0)$, migrate boxes from previous cell(s) $(0, ..., j-1)$ with upper-bound $x$-coordinates $x_u$ greater than the lower-bound $x$ coordinate of the current cell;
            · Overlap checks among different types of objects (as illustrated in Fig. 2.8);
          }
      }
```

algorithm itself. The problem domains are chosen to be rectangle (2D) or cuboid (3D), with randomly filled disks or spheres. The particle scale ranges from one million to five million, having a packing density ρ=0.1, 0.2, 0.3, 0.6, 0.8 respectively. The particle radius R is randomly generated by one of the following prescribed distributional functions:

Table 2.1: CPU(s) time: 1m disks with radius URD, $\rho = 0.6$

Algorithm	$\lambda = 1$		$\lambda = 10$		$\lambda = 100$	
ASDT	50.878	2503%	51.684	2561%	51.244	2584%
NBS	2.794	37.4%	3.235	60.7%	3.395	71.2%
D-cell	2.033	–	2.012	–	1.983	–

Table 2.2: CPU(s) time: 1m spheres with radius URD, $\lambda = 1$

Algorithm	$\rho = 0.1$		$\rho = 0.3$		$\rho = 0.6$	
ASDT	172.32	5605%	189.66	4868%	196.77	4164%
D-cell	3.074	–	3.895	–	4.725	–

- Mono-size, or identical radius R;
- Uniformly random distribution (URD) between $[R_{min}, R_{max}]$ (size ratio $\lambda = R_{max}/R_{min}$);
- Normal (Gauss) random distribution (NRD) between $[R_{min}, R_{max}]$ with average $R = (R_{min} + R_{max})/2$ and deviate $\sigma = (R - R_{min})/3 = (R_{max} - R)/3$;
- Log-normal random distribution (LNRD).

All the results are obtained on an Athlon 1.4 GHz PC with 512 RAM.

The first batch of tests consists of 1 million disks with a packing density of 0.6. Assuming an uniformly random distribution (URD) of disk radii, three different size ranges, controlled by λ, are considered. Note that $\lambda = 1$ corresponds to a mono-sized case. The CPU costs for the ASDT, NBS and D-cell are recorded in Table 2.1. It is evident that both NBS and D-cell are much more efficient than ASDT, with about 20 speed-up for all the cases, while D-cell is about 40~70% faster than NBS. In particular, for the mono-sized disks, D-cell achieves a 40% reduction in CPU times than NBS, indicating that a better implementation is achieved.

Comparison of 3D D-cell with ASDT is conducted with a problem comprising 1 million mono-sized spheres. NBS is not considered since 3D NBS is very complex to be implemented. Table 2.2 indicates that D-cell is over 40 times faster than ASDT.

The following numerical experiments are provided to examine such parameters as particle scale, cell size, packing density and particle size distributions on the performance of the D-cell algorithm itself. The results, as listed in Tables 2.3~2.9 and illustrated in Figs. 2.9~2.12, suggest that

Table 2.3: CPU vs ρ & λ: 1m disks, URD

Density ρ	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
0.1	1.642	1.602	1.612
0.2	1.742	–	–
0.3	1.832	1.792	1.782
0.6	2.033	–	–

Table 2.4: CPU vs ρ & λ: 2m disks, URD

Density ρ	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
0.1	1.777	1.753	1.743
0.2	1.898	–	–
0.3	1.988	1.948	1.933
0.6	2.173	–	–

- The D-cell algorithm is not sensitive to the range of object sizes; In fact less CPU costs are required for a larger λ due to a less number of contact pairs present.
- A denser packing leads to a higher CPU time since there are more contact pairs;
- The D-cell algorithm is not sensitive to particle size distributions;
- Slightly less CPU time is required for rectangle 2:1 domain, which suggests that the first cut should be applied along the direction with a larger dimension, if the objects are uniformly distributed in the domain;
- Figs. 2.10∼2.12 indicate that there exists a relatively large optimal cell size range, which should be three times (for 2D problems) or five times (for 3D problems) the average object size.

Table 2.5: CPU vs ρ & λ: 5m disks, URD

Density ρ	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
0.1	1.959	1.933	1.929
0.2	2.072	–	–
0.3	2.173	2.131	2.113
0.6	2.383	–	–

Table 2.6: CPU vs ρ & λ: 1m spheres, URD

Density ρ	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
0.1	3.074	2.874	2.844
0.2	3.545	–	–
0.3	3.896	3.525	3.455
0.6	4.727	–	–

Table 2.7: CPU vs ρ & λ: 2m spheres, URD

Density ρ	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
0.1	3.335	3.129	3.084
0.2	3.881	–	–
0.3	4.261	4.001	3.936
0.6	5.248	–	–

Table 2.8: CPU vs size distribution & λ: 1m disks, ρ=0.3

Radius distribution	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
URD	1.833	1.793	1.783
NRD	1.833	1.853	1.833
LNRD	1.833	1.842	1.853

Table 2.9: CPU(s) vs domain shape & λ: 1m disks, $\rho = 0.3$

Domain shape	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
Square	1.833	1.793	1.783
Rectangle 2:1	1.833	1.773	1.753
Rectangle 1:2	1.853	1.803	1.793

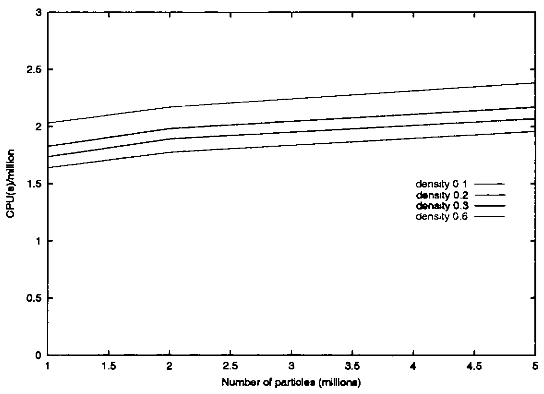


Figure 2.9: Effect of particle scale on CPU: mono size

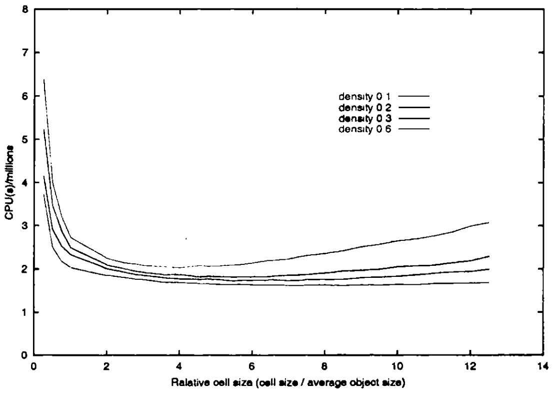


Figure 2.10: Optimal cell size *vs* packing density: URD

2.6 Concluding Remarks

The tree based search and the cell/grid based search are the most general contact detection algorithms for applications involving large numbers of discrete objects. In this chapter, the alternating digital tree (ADT) and the augmented spatial digital tree (ASDT) are reviewed and presented as examples of tree based search algorithms, whilst the dynamic cell based search algorithm, or *'D-cell'* for short, is presented as an example of cell/grid based search algorithm. Numerical tests show that the D-cell is much more efficient than the tree based search algorithms for large scale problems. The parametric study of the algorithm itself indicates that the performance of the D-cell is strongly dependent on the cell dimensions chosen. Taking three times for 2D problems or five times for 3D problems the average object size often yields a good performance.

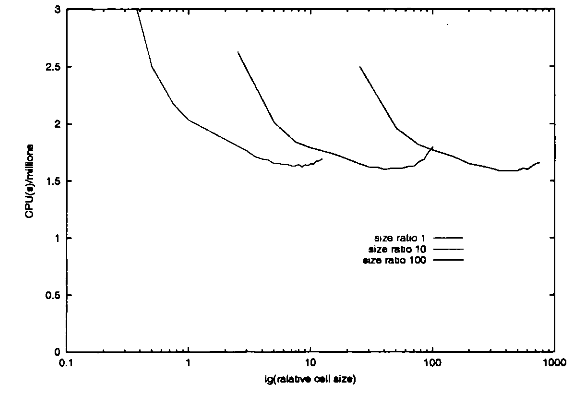


Figure 2.11: Optimal cell size *vs* particle size ratio λ: URD, $\rho = 0.1$

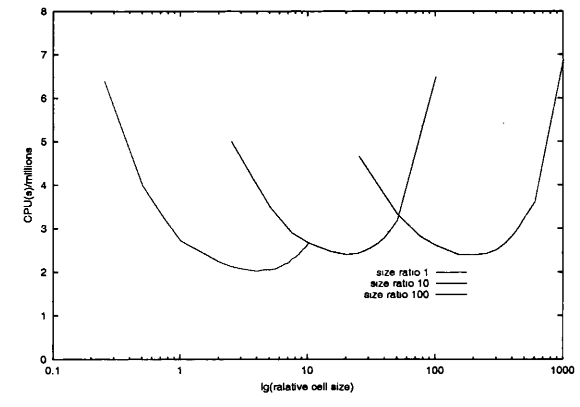


Figure 2.12: Optimal cell size *vs* particle size ratio λ: URD, $\rho = 0.6$

Bibliography

[1] T. Belytschko and M. O. Neal. Contact-impact by the pinball algorithm with penalty and lagrange methods. *Int. J. Numer. Meth. Engng.*, 31:547-572, 1991.

[2] A. Munjiza, D. R. J. Owen and N. Bicanic. A combined finite-discrete element method in transient dynamics of fracturing solids. *Eng. Comp.*, 12:145-174, 1995.

[3] E. Perkins and J. R. Williams. A fast contact detection algorithm insensitive to object sizes. *Eng. Comp.*, 12:185-201, 1995.

[4] A. Munjiza and K. R. F. Andrews. NBS contact detection algorithm for bodies of similar size. *Int. J. Numer. Meth. Engng.*, 43:131-149, 1998.

[5] R. Diekmann, J. Hungershover, M. Lux, L. Taenzer and J. M. Wierum. Efficient contact search for finite element analysis. *Proceedings of ECCOMAS 2000*, Barcelona, 2000.

[6] J. Bonet and J. Peraire. An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *Int. J. Num. Meth. Engng.*, 31:1–17, 1991.

[7] Y. T. Feng and D. R. J. Owen. An sugumented spatial digital tree algorithm for contact detection in computational mechanics. *Int. J. Numer. Meth. Engng.*, 55: 159-176, 2002.

[8] E. Perkins and J. R. Williams. CGrid: Neighbor searching for many body simulation. *4th Int. Conf. on Analysis of Discontinuous Deformation,* Glasgow, UK - June 6-8, 2001.

[9] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison Wesley, Reading, MA, 1990.

[10] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Comm. ACM*, 18(9):509–517, 1975.

[11] H. Samet. *Spatial Data Structures: Quadtree, Octrees and Other Hierarchical Methods.* Addison Wesley, Reading, MA, 1989.

[12] T. H. Cormen, C. E. Leiserson, and E. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, MA, 1989.

[13] M. G. Cottrell. *The development of Rational Computational strategies for the numerical modelling of high velocity impact.* Ph.D. Thesis, University of Wales Swansea, 2002.

Chapter 3

Contact Resolution for Circular and Spherical Discrete Objects

3.1 Introduction

Following the contact detection procedure, the identified pairs in potential contact are subjected to contact resolution.

In the context of discrete element or combined finite/discrete element (FE/DE) modelling, the contact between discrete objects or between discrete object and the finite element is simulated along their boundaries by appropriate *penalty based interaction laws*, which essentially govern the relation of the distance of the relative approach or overlap, and the generated repulsive force between the contacting pairs. Discrete objects are considered to be rigid, but a certain small overlap between discrete objects or between discrete object and the finite element is allowed, i.e. a *soft disk model* [1] is adopted. This '*soft*' model may be justified by the fact that both discrete object and the finite element are physically deformable and thus the allowed overlap may partially offset the error introduced by the assumed rigidity of the discrete objects.

Furthermore, it is assumed that the overlap is sufficiently small for small deformation theory to be applicable when describing the deformation of the contacting region of discrete objects. If δ denotes the maximum contact overlap (or penetration), this *small contact deformation assumption* requires that

$$\delta \ll 2R \tag{3.1}$$

where R is the radius of the disk or sphere, or a characteristic length of the irregular shaped object.

This chapter deals with contact resolution for circular and spherical discrete objects (disks and spheres), as well as for disk/segment and sphere/facet contact, which is generally required in the combined FE/DE modelling. For instance, in shot peening/peen forming simulations, the (deformable) boundary is approximated by a set of 2-noded segments (in 2D) or 3-noded triangular facets (in 3D) as a result of the finite element discretisation of the target material. Issues regarding non-circular discrete objects (polygons and superquadrics) contact will be discussed in the following chapter.

3.2 Contact Geometry and Forces

3.2.1 Disk/segment and disk/disk contact

Referring to Fig. 3.1(a), which illustrates the full contact between a circular disk and a segment (of the finite element), the following geometrical relations can be easily established: Suppose that d denotes the distance between the centre, O, of the disk and the segment $\overline{AB}$, and I, J represent the two intersection points of the segment with the disk circumference. A local system is set up with C as its origin, and the coordinates of I, J are assumed to be $-b$ and b; then an overlap function that gives the shortest distance between a point on the segment to the circumference of the disk can be expressed as

$$g(x) = \sqrt{R^2 - x^2} - d \qquad x \in [-b, b] \tag{3.2}$$

The maximum penetration or overlap between disk and segment, denoted by δ, is reached at $x = 0$:

$$\delta = g(0) = R - d \tag{3.3}$$

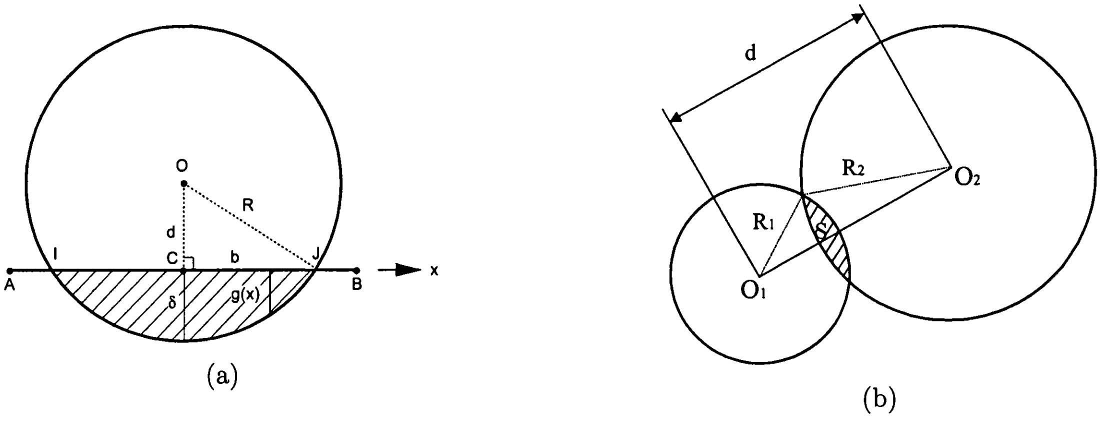


Figure 3.1: Geometrical relation – (a) disk/segment contact; (b) disk/disk contact

Clearly, the contact between disk and segment occurs only if $\delta > 0$, i.e. $d < R$. The length of $\overline{IJ}$ is defined as the full contact width, denoted by w. The centre point, C, of $\overline{IJ}$ is assumed to be the contact point, and the length of either $\overline{IC}$ or $\overline{CJ}$ is the half contact width b:

$$b = \frac{w}{2} = \sqrt{R^2 - d^2} = \sqrt{\delta(2R - \delta)} \tag{3.4}$$

The area of the overlap (contact) region (the shaded area in Fig. 3.1(a)), denoted by S, can be calculated as

$$S = 2R^2 \mathrm{arctg}\frac{\delta}{b} - b(R - \delta) \tag{3.5}$$

Under the small contact deformation assumption (3.1), b and S can be respectively approximated by

$$b = \sqrt{2R\delta} \tag{3.6}$$

$$S = \frac{4}{3}b\delta \tag{3.7}$$

In the case of disk/disk contact (Fig. 3.1(b)), the contact occurs only when the distance d between their centres is less than the sum of their radii, i.e.

$$\delta = R_1 + R_2 - d > 0 \tag{3.8}$$

When $\delta > 0$, this penetration will generate a contact traction $\boldsymbol{p}$ along the contact interface, which can be assumed to be a function of the contact position x

$$\boldsymbol{p} = \begin{cases} \boldsymbol{p}(x) & x \in [-b, b] \\ 0 & \text{otherwise} \end{cases}$$

Physically, this traction results from the deformation of bodies under contact. If the contact is assumed to be elastic, the surface traction may, at least theoretically, be determined by solving the equilibrium and compatibility equations at the interface between the two contacting bodies. However, it appears that except for a few special cases such as two elastic spheres in normal contact, no closed form expression for the surface traction can be established. Fortunately, for the discrete element approach it may not be necessary to give an accurate surface traction distribution, but the corresponding resultant force may play a more important part in the simulation.

The resultant force transmitted from one surface to another through an area of contact is obtained in principle by integrating the contact traction along the contact width as follows

$$\boldsymbol{F} = \int_{-b}^{b} \boldsymbol{p}(x)dx$$

This force is assumed to act at the contact point C, and it is further distributed to its two ends for the segment according to the distance ratio between $\overline{AC}$ and $\overline{CB}$.

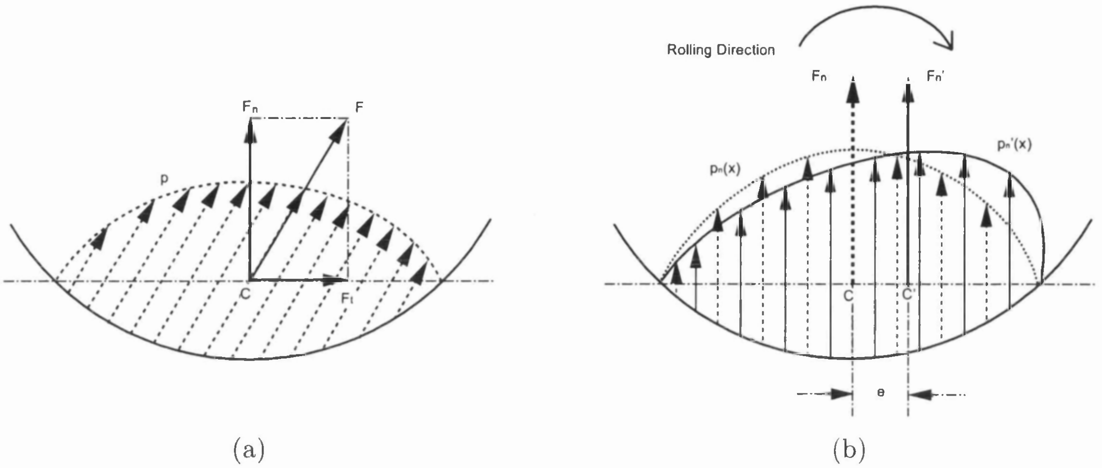


Figure 3.2: Contact forces: (a) general case; (b) normal contact trace distribution/force with rolling.

The resultant force $\boldsymbol{F}$ can be resolved into a normal component $\boldsymbol{F}_n$ acting along the common normal $\boldsymbol{n} = \vec{CO}$, and a tangential component $\boldsymbol{F}_t$ in the tangential plane sustained by sliding friction of the two contact surfaces, as shown in Fig. 3.2(a).

The role of contact interaction models is to essentially establish a certain analytical relation between contact forces $\boldsymbol{F}$ and the (maximum) contact overlap δ. Different assumptions regarding the contact force will result in different contact models.

In general, the tangential contact force, or the frictional force in this case, may interact with the normal force in some complex way, see e.g. [2] for more detailed discussion. In this work, an assumption is made that the normal contact force is independent of the frictional force, i.e. the normal and tangential contacts are uncoupled in the sense that the presence of friction in the contact will not affect the magnitude of the normal force.

The previous discussion regarding the normal force assumes a symmetric distribution of the normal contact traction $p_n(x)$ with respect to the origin of the local system, i.e. the contact point C. When the contact surfaces are in some relative angular motion, the normal contact traction will somehow redistribute so that the resulting normal force has an eccentricity with regard to its stationary position which will then generate a moment resistant to this relative rolling, as depicted in Fig. 3.2(b).

For disk/segment contact, the above discussion assumes a *full contact* (Fig. 3.1(a)), or more specifically, that the two intersection points I, J of the segment with the disk lie within the segment. In many situations, however, a so-called *partial contact* may occur, in which a disk may touch more than one segment of the polygonal boundary, or equally one or two ends of

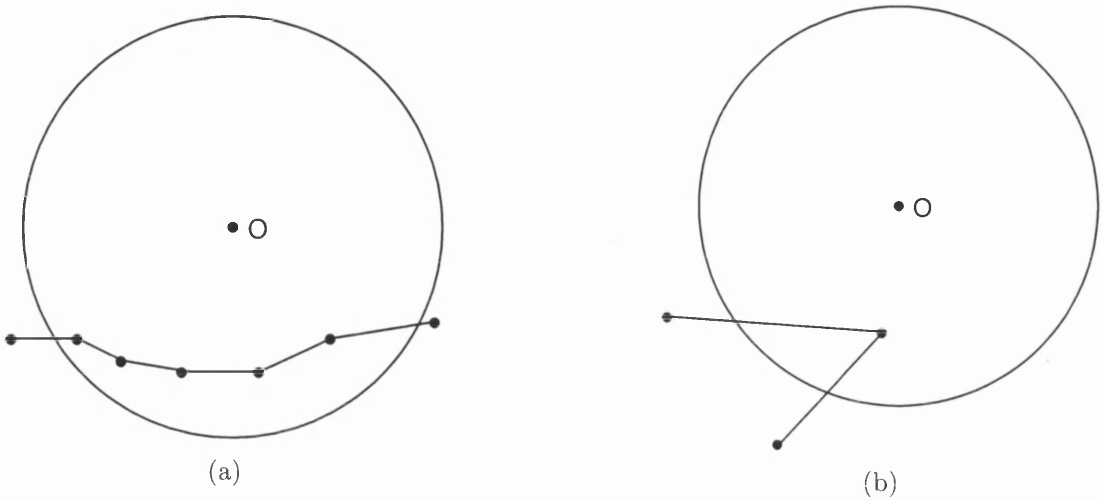


Figure 3.3: Partial contact: (a) disk contacts with multi-segments; (b) disk hits the corner of the boundary

the segment lie within the disk. Fig. 3.3 illustrates two different partial contact cases usually encountered in DEM simulations.

For partial contact, the corresponding resultant forces will be different from the full contact in terms of both magnitude and direction, which can be evaluated as follows: Firstly, the segment is assumed to be sufficiently long (by extending one or two of its ends, if necessary) so that a full contact situation is virtually established. Accordingly, a full contact force F_n can be computed based on a proper contact model. Then the partial contact force F_n^* is obtained by scaling F_n with a factor γ, i.e.

$$F_n^* = \gamma F_n \tag{3.9}$$

The scaling factor γ may be determined with different methods. Two approaches are given below:

- Overlap area ratio [3]: the scaling factor γ is determined by

$$\gamma = \frac{S_p}{S_f} \tag{3.10}$$

  where S_p and S_f are respectively the overlap area of partial and full contacts, as illustrated in Fig. 3.4(a).

- Contact width ratio: the scaling factor γ is computed by

$$\gamma = \frac{w_p}{w_f} \tag{3.11}$$

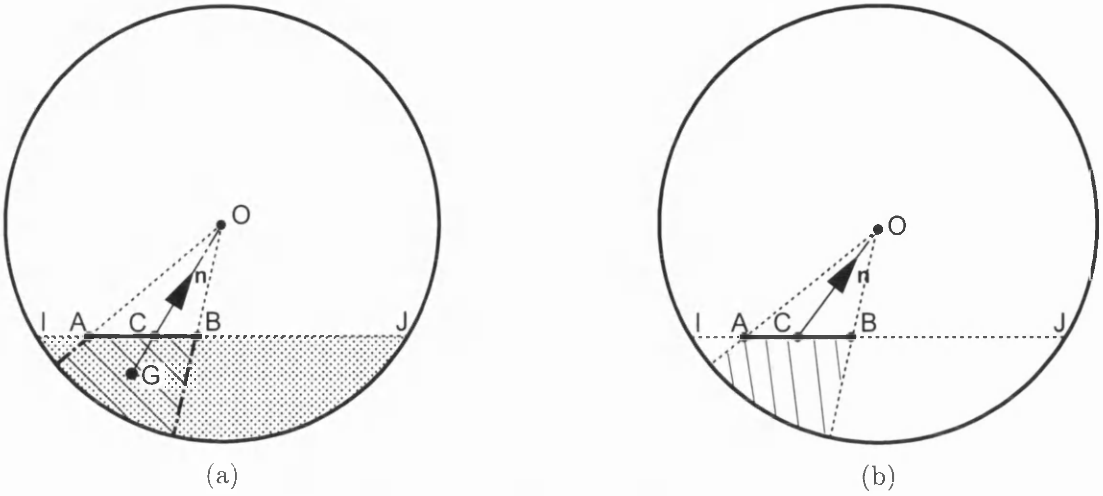


Figure 3.4: Definition of scaling factor γ and force direction $\boldsymbol{n}$: (a) overlap area ratio; (b) contact width ratio.

where $w_p = |\overline{AB}|$ and $w_f = |\overline{IJ}|$ are respectively the contact width of partial and full contacts as shown in Fig. 3.4(b).

The first approach has been used in [3] and the second one is first proposed here. Computing the scaling factor with the above two methods may result in different directions and contact points of the resultant forces. For the first approach, the direction of the resultant normal contact force $\boldsymbol{n}$ is defined by $\overline{GO}$, where G is the position of the centroid of the contact overlap area, and the contact point is determined by the intersection point of $\overline{GO}$ with the segment, as illustrated in Fig. 3.4(a). The numerical operations involved in the computation of S_p, S_f, and the positions of G and the contact point are significant, which may considerably degrade the overall performance of the DEM procedure. For this reason, the second approach is considered, in which the contact point C is assumed to be the midpoint of the contact width $\overline{AB}$ and the direction of the resultant force $\boldsymbol{n}$ is taken to be the direction of $\overline{CO}$. The required number of numerical operations is very small compared to the first approach.

Theoretically, it seems that the first approach may give more accurate solution, but taking into consideration the interactive model as a whole, no noticeable difference in terms of solution accuracy has been observed in the applications, as will be numerically demonstrated in Chapter 7.

As the disk may have contact with more than one segment, the resultant normal force (Fig. 3.5) acting on the disk is the sum of the normal contact forces between the disk and each segment:

$$\boldsymbol{F}_d = \sum_i F_i \boldsymbol{n}_i \tag{3.12}$$

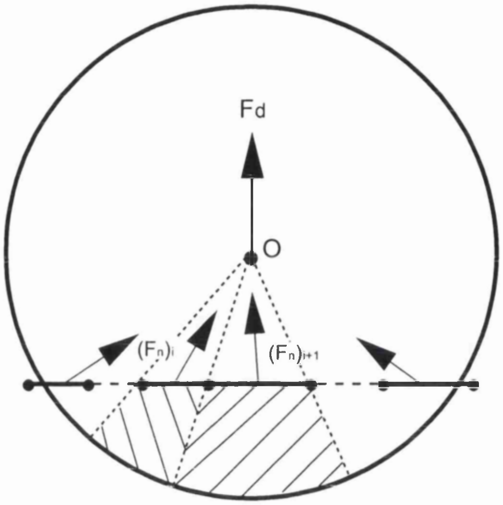


Figure 3.5: Computation of resultant normal force on disk

where $\boldsymbol{n}_i$ is the direction of normal contact force between the disk and the i-th segment.

3.2.2 Sphere/facet and sphere/sphere contact

Assume that the facet cuts the sphere into two parts with the upper spherical cap being the contact penetration, as illustrated in Fig. 3.6. Let δ denote the maximum overlap (or penetration) between the sphere and the facet. Then the overlap volume V_f can be evaluated as

$$V_f = \frac{\pi}{2} b^2 \delta \tag{3.13}$$

where $b = \sqrt{R^2 - (R-\delta)^2} = \sqrt{\delta(2R-\delta)}$ is the contact radius. With the small contact deformation assumption that $\delta \ll 2R$, V_f can be approximated by

$$V_f = \pi R \delta^2 \tag{3.14}$$

To evaluate the normal contact force, it is necessary to distinguish partial contact from full contact. In the latter case, the contact overlap occurs within the interior area of the facet, while in the former case, the sphere may touch more than one facet. One case of partial contact is illustrated in Fig. 3.7.

In the case of partial contact, the normal force is scaled by a factor γ which takes the partial contact situation into account. Two options are provided for the determination of γ:

$$\gamma = \frac{S_p}{S_f} \tag{3.15}$$

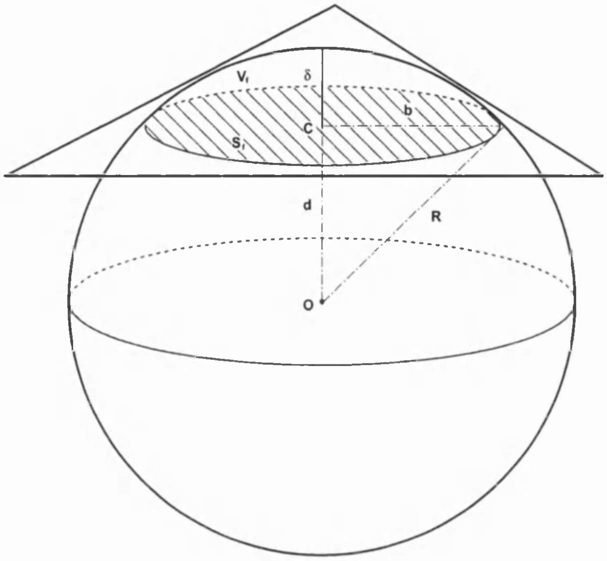


Figure 3.6: Sphere/facet contact: full contact

or

$$\gamma = \frac{V_p}{V_f} \tag{3.16}$$

where S_f and V_f are respectively the contact area and overlap volume in full contact; S_p and V_p respectively the contact area and volume in partial contact.

For the full contact (Fig. 3.6), the contact plane is a circle and the upper spherical cap represents the overlap volume. Therefore V_f is defined by (3.14) and S_f can be evaluated as

$$S_f = \pi b^2 \tag{3.17}$$

For partial contact (Fig. 3.7), however, evaluations of contact area S_p and overlap volume V_p are not so straightforward.

An elegant way to compute S_p, which is the overlap area between the facet and the contact circle, is to directly utilise the disk/segment partial contact algorithm as described in the previous subsection. Consider two different partial contact cases of a 3-noded facet overlapping with a circle as, respectively, illustrated in Figs. 3.8(a) and (b), where the centre of the circle, C, lies inside the facet in the first instance, and outside in the second case. Denote by 1, 2 and 3 the three nodes of the facet in an anticlockwise sense and also define the directions of the edges in a similar manner. Further assume the overlap areas of the three edges of the facet with the circle to be $S_i, i = 1, 2, 3$ with G_i as their geometric centres. Since the overlap of each edge of the facet with the circle can be viewed as a segment/disk partial contact

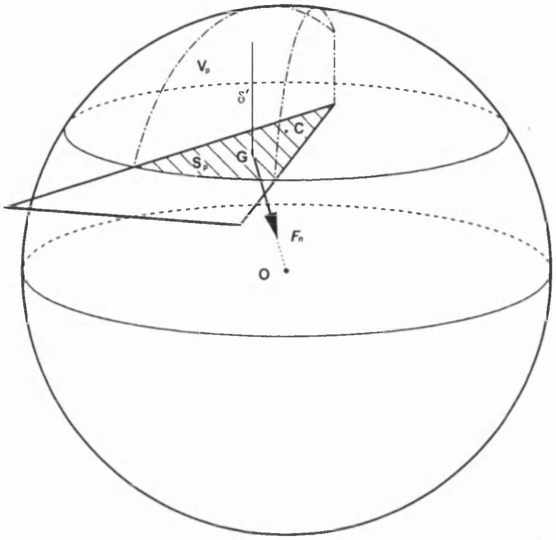
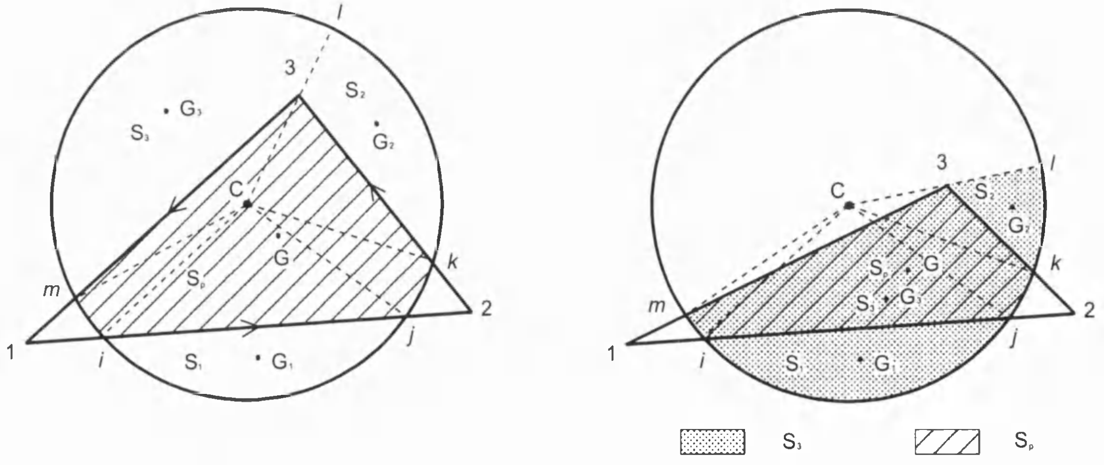


Figure 3.7: Sphere/facet contact: partial contact

problem, the three overlap areas S_i can therefore be evaluated following the same procedure as the disk/segment partial contact algorithm.

In the first instance as shown in Fig. 3.8(a), S_p can be computed by

$$S_p = \pi b^2 - \sum_{i=1}^{3} S_i \tag{3.18}$$



(a)

(b)

Figure 3.8: Area overlap of facet with contact circle: (a) case 1; (b) case 2

In the second instance shown in Fig. 3.8(b), S_3 is the area of the region enclosed by the arc $\widehat{mijkl}$ and two lines $\overline{l3}$ and $\overline{3m}$. Therefore S_p is determined by

$$S_p = S_3 - S_1 - S_2 \tag{3.19}$$

To derive a formula for a general situation, let S_i be a signed value and defined as *positive* if the centre of the circle lies on the *left side* of the edge, and *negative* otherwise. According to the definition, S_3 is negative in this particular case. Consequently S_p can be expressed as

$$S_p = \begin{cases} \pi b^2 - \sum_{i=1}^{3} S_i & \text{if case 1} \\ -\sum_{i=1}^{3} S_i & \text{if case 2} \end{cases} \tag{3.20}$$

Suppose that G is the geometric centre of the contact area S_p. If G, G_i and C also symbolically represent their coordinates, G can be determined as

$$G = \begin{cases} \Big[(\pi b^2)C - \sum_{i=1}^{3} S_i G_i\Big]/S_p & \text{if case 1} \\ -\Big[\sum_{i=1}^{3} S_i G_i\Big]/S_p & \text{if case 2} \end{cases} \tag{3.21}$$

Note that the computational procedure outlined above can also be applied to the situation where the facet is a polygon with any number of edges.

The overlap volume V_p is a slice of the spherical cap. Although it is possible to derive an algorithm to evaluate it exactly, it is often sufficient in terms of accuracy to compute V_p approximately as described below.

Suppose that δ' is the shortest distance between G and the surface of the sphere as illustrated in Fig. 3.7, then V_p is approximately computed by

$$V_p = S_p \delta' \tag{3.22}$$

This formulation is sufficiently accurate when the overlap area S_p is relatively smaller than the contact circle, which is normally the case where the size of the facet is much smaller than the sphere in the conbined FEM/DEM simulation.

Finally, $\boldsymbol{F}_n$ is assumed to act along the direction connecting the sphere's centre O and G (which is coincident with C in the full contact case), and is further distributed to the 3 nodes of the facet according to the area coordinates of G.

For sphere/sphere contact (Fig. 3.11), the contact overlap volume V_f is the sum of the two overlap spherical caps:

$$V_f = V_1 + V_2 = \frac{\pi}{6}\delta_1(3b^2+\delta_1^2) + \frac{\pi}{6}\delta_2(3b^2+\delta_2^2) = \frac{\pi}{6}3b^2(\delta_1+\delta_2) + \frac{\pi}{6}(\delta_1^3 b + \delta_2^3 b) \doteq \frac{\pi}{2}b^2\delta \tag{3.23}$$

where $\delta = \delta_1 + \delta_2$ is the overlap gap and b the contact half width.

3.3 Normal Interaction Laws

In this section, four interaction models, namely linear (Hooke), Hertz, Winkler and power law models for the normal contact, will be presented together with a general analysis of the equivalence between them. As mentioned earlier, the interaction laws basically define the relationship between the normal contact force and the (maximum) penetration δ:

$$F_n = F(\delta) \tag{3.24}$$

Generally speaking, these four models can be broadly divided into two groups. The first group directly assumes the relation between the resultant force and the maximum overlap; while the second group gives the distribution of the normal contact traction and then integrates it to obtain the $F_n - \delta$ relation. In this sense, the linear and power law models belong to the first group, whereas the Hertz and Winkler models fall into the second group. The description of the linear, Hertz and power law models may be found elsewhere, e.g. [4], but the Winkler models are first introduced in this work.

3.3.1 The linear/Hooke model

The linear or Hooke model is possibly the simplest normal contact model that assumes the following linear relationship between F_n and δ

$$F_n = k_n \delta \tag{3.25}$$

where k_n is a model parameter to be chosen. Note that F_n is independent of the contact width and radius of the disk/sphere in this model.

3.3.2 The Hertz models

The classic model for 3D contact

It is well known that the classic Hertzian theory [5] was the first model that successfully established the normal contact traction distribution when two elastic spheres without surface adhesion are in contact under a normal pressing force. This theory is also generally adopted for modelling the normal contact of two rigid spheres in the *soft disk model* employed here.

In the Hertz model, the normal contact traction is represented by the ordinates of a hemisphere,

$$p_n(x) = q_0 \sqrt{a^2 - r^2} \tag{3.26}$$

where a is the radius of the hemisphere and r is the distance to the contact centre. Furthermore, the total applied normal force can be expressed in terms of the relative approach of two spheres δ by

$$F_n = k_n \delta^{\frac{3}{2}} \tag{3.27}$$

with the stiffness coefficient

$$k_n = \frac{4E^* \sqrt{R^*}}{3} \tag{3.28}$$

where R^* is the relative radius defined by

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2}$$

and E^* is the relative Young's modulus related to the elastic properties E_1, E_2, and μ_1, μ_2 of the two spheres, given by the following expression

$$\frac{1}{E^*} = \frac{1-\mu_1^2}{E_1} + \frac{1-\mu_2^2}{E_2}$$

It should be emphasised that in the present *soft disk model* philosophy, E_1, E_2, μ_1 and μ_2 may not be necessarily considered as the material properties of the spheres, and should rather be considered as the model parameters.

In the case of the contact between a sphere (of radius R) and a half-plane (of radius ∞),

$$k_n = \frac{4E'\sqrt{R}}{3} \qquad (E' = \frac{E}{1-\mu^2}) \tag{3.29}$$

Generalisation to 2D contact

It is not straightforward to extend the above classic Hertz model to include the contact of two disks. The work of Poritsky [6] is adopted here, through an extension of the classic Hertzian theory. In this generalised Hertz model, the normal force F_n is defined with respect to the normal overlap of two disks in contact by:

$$F_n = k_n \frac{b^2}{R^*}$$

with the stiffness coefficient

$$k_n = \frac{\pi}{4} \frac{1 + (k_4/k_3)^2}{k_1 + k_2}$$

where

$$k_1 = \frac{1-\mu_1}{G_1}, \quad k_2 = \frac{1-\mu_2}{G_2}, \quad k_3 = \frac{2}{\pi}(k_1 + k_2), \qquad k_4 = \frac{1-2\mu_1}{G_1} - \frac{1-2\mu_2}{G_2}$$

and G_1 and G_2 are the shear moduli of the disks. The contact force between a disk and segment can be easily obtained from the above expression as a special case with one of the disks' radii being infinity:

$$F_n = k_n \frac{b^2}{R} \tag{3.30}$$

Furthermore, the assumed rigidity of the disk also reduces the expression of k_n to

$$k_n = \beta E \tag{3.31}$$

where

$$\beta = \frac{\pi}{4}[1 + \frac{\pi^2(1-2\mu)^2}{4(1-\mu)^2}]\frac{E}{2(1-\mu^2)}$$

Under the small contact deformation assumption that $\delta \ll 2R$, expression (3.30) can be simplified as

$$F_n = 2k_n\delta \tag{3.32}$$

3.3.3 The Winkler models

The disk/segment model

The Winkler model described below is for disk/segement contact, although extention to disk/disk contact is straightforward. Similar to the classic Hertzian theory, the Winkler model establishes a contact traction function. It essentially assumes that the segment and the disk consist of many sub-segments and each sub-segment is in contact with the closest sub-segment on the disk, as illustrated in Fig. 3.9. Furthermore, in accordance with Winkler's elastic foundation theory, each pair of overlapping line segments on both segment and disk may then be considered independently. More specifically, the normal contact traction may be approximately expressed by

$$p_n(x) = k_n g(x) \qquad x \in [-b, b] \tag{3.33}$$

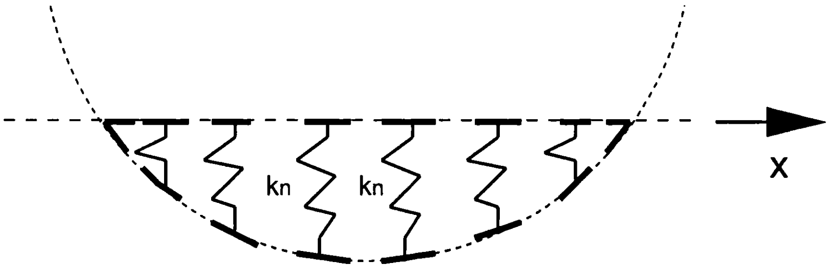


Figure 3.9: The Winkler contact model

where the stiffness coefficient k_n is a model parameter and $g(x)$ is the overlap function defined by (3.2). Then the resultant normal force can be obtained by

$$F_n = \int_{-b}^{b} p_n(x)dx = k_n S \tag{3.34}$$

where S is the overlap area. In addition, the maximum and average traction are respectively

$$p_{max} = p_n(0) = k_n \delta \tag{3.35}$$

and

$$\overline{p}_n = \frac{\int_{-b}^{b} p_n(x)dx}{w} = k_n \frac{S}{w} \tag{3.36}$$

For the present small contact deformation assumption,

$$\overline{p}_n = \frac{2}{3} k_n \delta = \frac{2}{3} p_{n max} \tag{3.37}$$

or

$$p_{n max} = 1.5 \overline{p}_n \tag{3.38}$$

i.e. the maximum normal contact traction is 1.5 times the average normal contact traction. Furthermore,

$$F_n \approx \frac{4}{3} k_n \delta b \approx \bar{k}_n \delta^{\frac{3}{2}} \tag{3.39}$$

with

$$\bar{k}_n = \frac{4\sqrt{2R}}{3} k_n$$

Generalisation to sphere/facet contact

Following the same procedure as described above, the expression for sphere/facet contact can be derived:

$$F_n = k_n \, V_f \tag{3.40}$$

where k_n is a model parameter to be chosen. The approximation of V_f by (3.14) leads to

$$F_n = \pi k_n \, R \, \delta^2 \tag{3.41}$$

3.3.4 The power law model

The power law model [7] gives the relation of the total contact force and the overlap as

$$F_n = k_n \delta^m \tag{3.42}$$

where the stiffness coefficient k_n and m are two model parameters.

3.3.5 Equivalence between models

For the above four normal contact models, except that the linear model is obviously a special case of the power law model (with $m = 1$), there seems no obvious links among the power law model (including the linear model), Hertz models and the Winkler models. Under the small contact deformation assumption, however, some equivalence among them may exist. Table 3.1 summaries the models with the classic Hertzian model in both general and approximate forms.

From the table, the following relations can be established under the small deformation assumption:

- The generalised Hertz model is in fact equivalcnt to the linear model, and the contact width and the size of the disk/sphere will not affect the normal contact force. This equivalence also suggests that the stiffness coefficient k_n in the linear model may be taken as
$$k_n = 2\beta E \tag{3.43}$$
- It is interesting to note that the proposed 2D Winkler model is almost equivalent to the classic Hertz model, at least in form if the spring stiffness coefficient k_n is numerically chosen as
$$k_n = \frac{E'}{\sqrt{2}} \tag{3.44}$$
- All the models can be considered as special cases of the power law model with different values of m and k_n; the corresponding value of m for each model is listed in the fourth column of the table.

The above observations are significant in the following aspects:

Table 3.1: *Normal Contact Models in General and Approximate Forms*

Model	General	Approximation	Power of δ	Coefficient
Name	Form	($\delta \ll 2R$)	(m)	
Linear/Hooke	$k_n^L \delta$	$k_n \delta$	1	k_n
Classic Hertz (3D)	$k_n^{CH} \delta^{\frac{3}{2}}$	$k_n \delta^{\frac{3}{2}}$	$\frac{3}{2}$	$\frac{4E\sqrt{R}}{3(1-\mu^2)}$
Generalised Hertz (2D)	$\frac{k_n^{GH} b^2}{R}$	$2k_n \delta$	1	βE
2D Winkler	$k_n^{W_2} S$	$\bar{k}_n \delta^{\frac{3}{2}}$	$\frac{3}{2}$	$\frac{4\sqrt{2R}}{3} k_n$
3D Winkler	$k_n^{W_3} V_f$	$\bar{k}_n \delta^2$	2	$\pi R k_n$
Power Law	$k_n^{PL} \delta^m$	$k_n \delta^m$	m	k_n

- All the models considered can be represented by the power law model;
- Within the power law model, our interest can be restricted to the following two particular cases only: (1) the linear (Hooke) model ($m = 1$); and (2) the Hertz model ($m = 3/2$);
- As mentioned earlier, the contact models considered are penalty based. Therefore the stiffness coefficient k_n associated with the above two models needs to be selected properly. It may not be necessary for k_n to take the values as defined in (3.43) and (3.44), but should be modified by a factor α. Consequently the two contact models can be expressed in a single form as

  $$F_n = \alpha E R^{m-1} \delta^m \qquad (\text{with } m = 1, \frac{3}{2}) \tag{3.45}$$

  The introduction of the factor α in the models has the advantage that α is a non-dimensional constant, which should be much easier to choose than the original k_n.

What needs to be further investigated is clearly the comparison of the performance of the two models, the linear and Hertz, each with a chosen non-dimensional constant α. Such an investigation will be conducted by means of numerical experiments in Chapter 7.

3.4 Modelling of Tangential Friction

Friction is one of the fundamental issues in DEM modelling. The classic Coulomb friction law is often employed in engineering applications for its simplicity. The discontinuous nature of the friction force in this model, however, imposes some numerical difficulty when the relative sliding velocity reverses its direction and/or during the transition from *sliding* (*sticking*) to sticking (sliding). The difficulty is usually overcome by artificially introducing a '*transition zone*' which smears the discontinuity in the numerical computation. With this model, the tangential force is obtained with respect to the total relative tangential displacement between the contact pairs, following the standard perfectly elasto-plastic theory of friction as shown in Fig. 3.10. Initially in the loading case, the tangential force increases linearly with a slope of k_t with the total relative tangential displacement until it reaches its maximum value $(F_t)_{max} = \mu F_n$, where μ is the frictional coefficient. Thereafter further increase of the relative displacement will not change the magnitude of the force, but only causes sliding of one body on another. Upon decrease of the relative displacement, i.e. in unloading case, the force will decrease with the same slope as in the initial loading case. If this unloading is large enough, the force reaches its maximum value in another direction. When reloading happens, it will

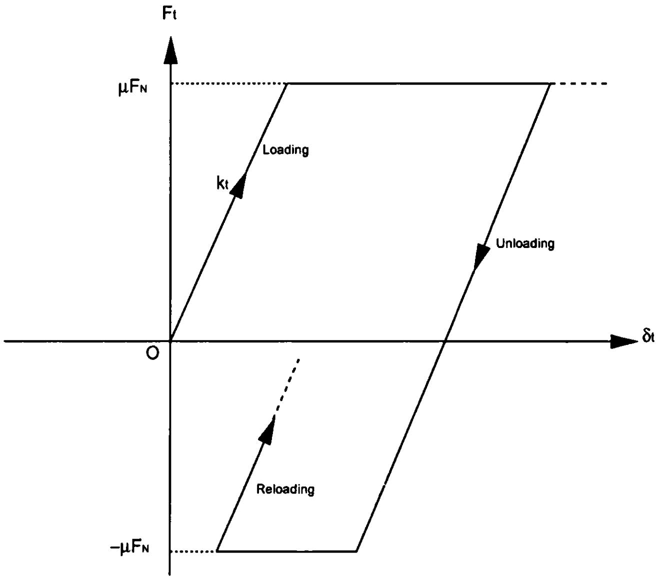


Figure 3.10: Tangential contact model

follow the same rule as that of unloading. These initial loading, unloading and reloading situations are illustrated in Fig. 3.10.

The above friction model can be readily applied to 2D contact problems. It is however difficult to be employed in general 3D contact problems where the relative motion at the contact surface can be very complex. A modified classic Coulomb friction model is therefore proposed for 3D contact, in which the analogue of the classic friction law with perfectly elasto-plastic theory is applied.

3.4.1 Modified classic Coulomb friction model for spheres

To discuss the friction model for transverse movement, consider two spheres in contact, as shown in Fig. 3.11. Let R_1, R_2, $\boldsymbol{V}_1, \boldsymbol{V}_2$, ω_1, ω_2 denote respectively the radii, the velocities of the centres O_1 and O_2, and the angular velocities of two spheres. Assume that sphere 1 and sphere 2 are respectively target and contactor, with C as their contact point, and that the tangential displacement and contact forces on the contactor are considered. Then a local coordinate system is defined with the contact point C as the origin, the direction from O_1 toward O_2 as the normal contact direction denoted by a normalised vector $\boldsymbol{n}$, and the contact plane as the tangential plane.

Following the standard arguments of the elasto-plastic theory of friction, the total relative tangential displacement of the two contact spheres, $\boldsymbol{u}_t$, may be decomposed into an *elastic*

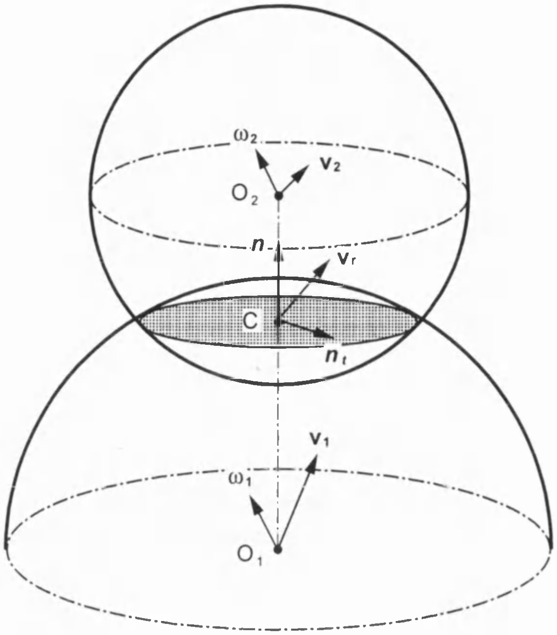


Figure 3.11: Two spheres in contact

or *adherence* component and a *plastic* or *slip* component (see e.g. [7, 8, 9]), i.e.

$$\boldsymbol{u}_t = \boldsymbol{u}_t^e + \boldsymbol{u}_t^p \tag{3.46}$$

Within the elastic region the tangential force $\boldsymbol{F}_t$ is defined as

$$\boldsymbol{F}_t = k_t \boldsymbol{u}_t^e \tag{3.47}$$

where k_t is the elastic tangential stiffness or penalty coefficient.

The occurrence of slip is determined by the *yield function* or *slip condition* Φ, generally a function of contact forces, frictional coefficient and a set of internal variables which take into account the history dependence of the friction phenomenon. For simple Coulomb friction, Φ takes the form

$$\Phi = \|\boldsymbol{F}_t\| - \mu F_n \tag{3.48}$$

where μ is the frictional coefficient, and F_n the magnitude of the normal contact force $\boldsymbol{F}_n$.

Assuming that the normal force is constant, the *flow* or *slip* rule that governs the evolution of the tangential slip displacement component is defined as

$$\dot{\boldsymbol{u}}_t^p = -\dot{\lambda}\frac{\partial \Psi}{\partial \boldsymbol{F}_t} \tag{3.49}$$

where Ψ is the slip potential, which determines the frictional direction; and $\dot{\lambda}$ is consistent with the slipping/sticking condition or the so-called Kuhn-Tucker conditions:

$$\begin{cases} \Phi \leq 0 \\ \dot{\lambda} \geq 0 \\ \dot{\lambda}\Phi = 0 \end{cases} \tag{3.50}$$

For *isotropic* frictional contact, the slip potential Ψ is defined by

$$\Psi = \|\boldsymbol{F}_t\| \tag{3.51}$$

that leads to

$$\frac{\partial \Psi}{\partial \boldsymbol{F}_t} = \frac{\boldsymbol{F}_t}{\|\boldsymbol{F}_t\|} \tag{3.52}$$

Therefore,

$$\dot{\boldsymbol{u}}_t^p = -\dot{\lambda}\frac{\boldsymbol{F}_t}{\|\boldsymbol{F}_t\|} \tag{3.53}$$

3.4.2 Computation of tangential force at discrete time steps

Noting the analogy with elasto-plastic theory, the computation of the tangential force at discrete time instants $\{t_n\}$ can be performed in the following main steps:

Incremental relative sliding distance

At the current time instant t_{n+1}, the relative velocity vector at the contact point C is computed by:

$$\boldsymbol{V}_r = \boldsymbol{V}_{c_1} - \boldsymbol{V}_{c_2} \tag{3.54}$$

where $\boldsymbol{V}_{c_1}$ and $\boldsymbol{V}_{c_2}$ are the velocity vectors at the contact point respectively for contact spheres 1 and 2, and evaluated respectively by

$$\boldsymbol{V}_{c_1} = \boldsymbol{V}_1 + \boldsymbol{\omega}_1 \times \boldsymbol{r}_{c1} \qquad \boldsymbol{r}_{c1} = \overline{CO_1} \tag{3.55}$$

and

$$\boldsymbol{V}_{c_2} = \boldsymbol{V}_2 + \boldsymbol{\omega}_2 \times \boldsymbol{r}_{c2} \qquad \boldsymbol{r}_{c2} = \overline{CO_2} \tag{3.56}$$

$\boldsymbol{V}_r$ can be decomposed into two components, $\boldsymbol{V}_{r_n}$ in the normal direction and $\boldsymbol{V}_{r_t}$ in the tangential direction of the contact plane:

$$\boldsymbol{V}_{r_n} = (\boldsymbol{V}_r \cdot \boldsymbol{n})\,\boldsymbol{n}$$

and

$$\boldsymbol{V}_{r_t} = \boldsymbol{V}_r - \boldsymbol{V}_{r_n}$$

The current tangential direction is thus defined by

$$\boldsymbol{n}_t = \frac{\boldsymbol{V}_{r_t}}{\|\boldsymbol{V}_{r_t}\|} \tag{3.57}$$

The incremental relative tangential displacement $\Delta \boldsymbol{u}_t$ is then computed by

$$\Delta \boldsymbol{u}_t = \boldsymbol{V}_{r_t} \Delta t = \|\boldsymbol{V}_{r_t}\| \, \boldsymbol{n}_t \, \Delta t \tag{3.58}$$

Note that this displacement is contributed to by the transverse movement and the relative rotation of the two spheres.

Co-rotation of the tangential force at the previous time step

Suppose the tangential force at the previous time instant t_n is $\boldsymbol{F}_t^n$. Since the contact plane and tangential direction at t_n may not coincide with the current values due to the relative rotation between the two spheres, $\boldsymbol{F}_t^n$ has to be properly transformed to the current contact plane before the force update can be carried out. The transformation is achieved by the following length-preserving co-rotation

$$\boldsymbol{F}_{t_R}^n = \boldsymbol{T}_r \, \boldsymbol{F}_t^n$$

where $\boldsymbol{T}_r$ is a specially selected orthogonal transformation matrix such that the following two conditions are satisfied:

$$\|\boldsymbol{F}_{t_R}^n\| = \|\boldsymbol{F}_t^n\|$$

and

$$\boldsymbol{F}_{t_R}^n \cdot \boldsymbol{n} = 0$$

The co-rotated $\boldsymbol{F}_t^n$ in the current contact plane, i.e. $\boldsymbol{F}_{t_R}^n$, is then decomposed into two components, one in the $\boldsymbol{n}_t$ direction and another in $\boldsymbol{n}_t'$ which are orthogonal to $\boldsymbol{n}_t$ in the contact plane:

$$\boldsymbol{F}_{t_R}^n = \beta_1 \, \boldsymbol{n}_t + \beta_2 \, \boldsymbol{n}_t'$$

where

$$\beta_1 = \boldsymbol{F}_{t_R}^n \cdot \boldsymbol{n}_t$$

$$\beta_1 = \boldsymbol{F}_{t_R}^n \cdot \boldsymbol{n}_t$$

$$\beta_2^2 = \|\boldsymbol{F}_{t_R}^n\|^2 - \beta_1^2$$

tem

The transformation is accomplished by further rotating $\boldsymbol{F}_{t_R}^n$ to the local coordinate system

$$\theta_r = \omega_r \Delta t \tag{3.59}$$

where $\omega_r = (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \cdot \boldsymbol{n}$ defines the relative normal rotational velocity.

Update of the current tangential force

After the tangential force $\boldsymbol{F}_t^n$ has been properly rotated into the current tangential direction $\boldsymbol{n}_t$, a trial tangential force can be evaluated as

$$\boldsymbol{F}_t^{trial} = \boldsymbol{F}_{t_R}^n + k_t \Delta \boldsymbol{u}_t \tag{3.60}$$

after which the slip criterion is checked:

$$\Phi(\boldsymbol{F}_t^{trial}) = \|\boldsymbol{F}_t^{trial}\| - \mu \|\boldsymbol{F}_n\| \leq 0 \tag{3.61}$$

Finally, the tangential force $\boldsymbol{F}_t$ at the current time step t_{n+1} can be determined as

$$\boldsymbol{F}_t = \begin{cases} \|\boldsymbol{F}_t^{trial}\| \, \boldsymbol{n}_t & \text{if } \Phi(\boldsymbol{F}_t^{trial}) \leq 0 \\ \mu \|\boldsymbol{F}_n\| \, \boldsymbol{n}_t & \text{if } \Phi(\boldsymbol{F}_t^{trial}) > 0 \end{cases} \tag{3.62}$$

Remark 1. The derivation of the above procedure assumes that only a small amount of relative rotation occurs in neighbouring time steps so that the two contact planes are not significantly different. This assumption is justified in the explicit analysis where the time step is very small.

Remark 2. The procedure only requires that the tangential force of the previous time step is saved and therefore minimises the storage requirement.

Remark 3. The evaluation of the tangential force in the *sphere-facet contact* case can be achieved by setting the radius of the target sphere to be infinite and the angular velocity to be zero.

Remark 4. The frictional force for the rotations is neglected.

3.5 Modelling of Energy Dissipation

Energy dissipation is a critical characteristics of contact problems, therefore the contact models presented above should also represent the feature of energy losses in the simulation. Two approaches are given below.

3.5.1 Hysteretic model

The basic idea of the hysteretic energy dissipation model is to introduce a different stiffness coefficient k_n to the interaction laws for loading and unloading cases [10]. When it is applied to the general power law model, the normal force is changed to

$$F_n = \begin{cases} k_n \delta^m & \text{for loading} \\ k_n' (\delta - \delta_0)^m & \text{for unloading} \end{cases} \tag{3.63}$$

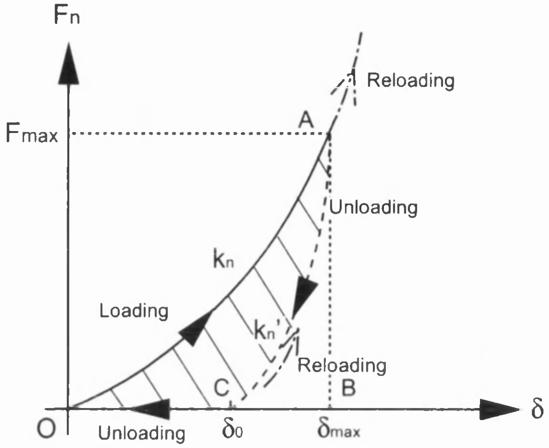


Figure 3.12: The hysteretic energy dissipation model

At the initial loading stage, the normal force follows the original power law curve with the stiffness coefficient k_n. Unloading follows a steeper slope curve with a larger stiffness coefficient k_n' until the normal force becomes zero. It is noted that a finite overlap δ_0 remains. Upon further unloading the force remains zero, or upon reloading, it increases with slope k_n' until the original loading curve is reached. This loading, unloading and reloading features of the model are illustrated in Fig. 3.12. The (shaded) area enclosed by a full loading and unloading path represents the amount of energy lost during impact. Suppose that F_{max} and δ_{max} are respectively the maximum force and overlap reached, the remaining overlap δ_0 is given by

$$\delta_0 = \delta_{max}\Big[1 - \Big(\frac{k_n}{k_n'}\Big)^{\frac{1}{m}}\Big] \tag{3.64}$$

The energy loss may be measured by a factor ρ termed the *effective coefficient of restitution* defined by

$$\rho = \frac{v_n'}{v_n} \tag{3.65}$$

where v_n and v_n' are respectively the impacting and recoiling velocities of the disk. This coefficient can be defined equivalently in terms of the original normal kinetic energy E_n and the rebounding kinetic energy E_n' by

$$\rho = \sqrt{\frac{E_n'}{E_n}} \tag{3.66}$$

In Fig. 3.12, E_n and E_n' are respectively equal to the areas enclosed by (curved) lines $O \rightarrow A \rightarrow B \rightarrow O$ and $A \rightarrow C \rightarrow B \rightarrow A$. Under the definition of (3.65) or (3.66), it can be shown that if k'_n is a fixed value (independent of the load history), then a constant coefficient of restitution given by

$$\rho = (\frac{k_n}{k'_n})^{\frac{1}{2m}} \tag{3.67}$$

results from this hysteretic model. Therefore for the linear Hookian model ρ equals

$$\rho = \sqrt{\frac{k_n}{k'_n}} \tag{3.68}$$

while for the Hertz model, ρ is given by

$$\rho = \sqrt[3]{\frac{k_n}{k'_n}} \tag{3.69}$$

In practice, the user can choose a proper ratio of k_n and k'_n to achieve the desired amount of energy loss during impact. The numerical procedure involved in the implementation of this energy loss model is complex as loading and unloading situations need to be identified and the histories of F_{max}, δ_{max} and δ_0 should be stored due to the model's history dependent property.

3.5.2 Viscous damping model

An alternative to the above model is to utilise a viscous damping term to dissipate energy. If this approach is applied to the power law contact model, the normal force is modified to

$$F'_n = k_n\delta^m + C_n\dot{\delta} \tag{3.70}$$

where $\dot{\delta}$ is the relative normal velocity at the contact point and C_n is the damping coefficient. Such a model also produces a constant coefficient of restitution, independent on the impact velocity. For the linear contact model, ρ is computed as

$$\rho = e^{-\frac{\pi\xi_n}{\sqrt{1-\xi_n^2}}} \tag{3.71}$$

where ξ_n is the critical damping coefficient (see Section 3.6 for its definition). As usually implemented, however, the force is purely repulsive and the viscous term is set to be zero if there is no overlap. Wen the initial contact (i.e. overlap) is established, a discontinuity in the force may occur since velocity is maximum at $\delta = 0$. This discontinuity could be eliminated by employing some '*weighted*' techniques. See [4] and [7] for more details.

3.6 Rheological Representation of Normal Contact

The final normal contact model that combines the original power law model with the viscous damping energy loss model for the normal contact of disk and segment may be physically represented by a rheological model that comprises a (non-)linear spring and a parallel dashpot as shown in Fig. 3.13, where the disk is described by a lumped mass and the segment is represented by two lumped masses.

The instantaneous stiffness K_n of the rheological system is defined by

$$K_n = \frac{dF_n}{d\,\delta} = mk_n\delta^{m-1} \tag{3.72}$$

For partial contact, the corresponding stiffness becomes

$$K_n' = \frac{d(\gamma F_n)}{d\,\delta} = K_n(\gamma + \frac{\delta}{m}\frac{d\gamma}{d\,\delta}) \tag{3.73}$$

As the second term in the right hand side of the above equation is smaller compared to the first term, K_n' can be approximated by

$$K_n' = \gamma K_n \tag{3.74}$$

Consequently, the instantaneous frequency is defined by

$$\omega_n = \sqrt{\frac{K_n}{\overline{M}}} \tag{3.75}$$

where $\overline{M}$ is the equivalent mass of the system defined by

$$\overline{M} = \frac{M(M_A + M_B)}{M + M_A + M_B}$$

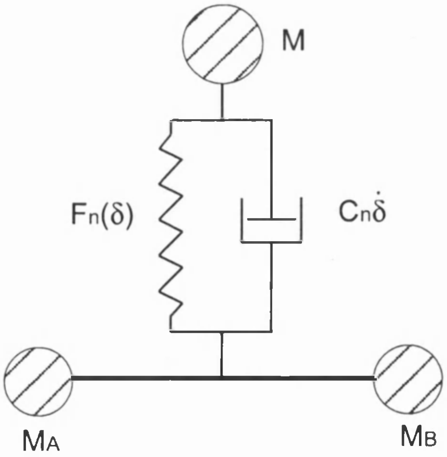


Figure 3.13: The rheological representation of normal contact model

Then the viscous damping coefficient C_n can be expressed by

$$C_n = 2\omega_n \xi_n \overline{M} \tag{3.76}$$

3.7 Critical time step

In the explicit solution procedure, the marching time step must not be greater than the *critical time step* Δt_{cr} to ensure the numerical stability of the scheme. Conventionally Δt_{cr} is defined by the highest frequency of the whole system, ω_{max}, as

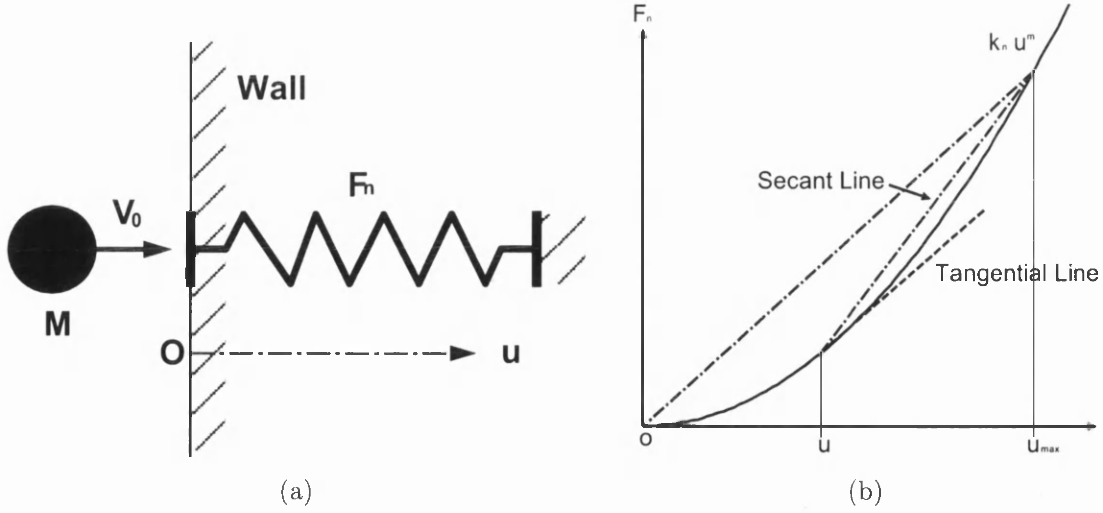
$$\Delta t_{cr} = \frac{2}{\omega_{max}} \tag{3.77}$$

It is essential to point out that the derivation of the above formula for the evaluation of the critical time step is under the condition that the dynamic system is *linear*. In the nonlinear case, the critical time step evaluated by (3.77) may not be applicable.

This issue has been highlighted by Yu [11], who has demonstrated that the critical time step derived from the linear contact interaction model in the conventional way cannot guarantee a stable solution. Following Yu's work, a time step scheme that can ensure the stability of the system for the normal contact models proposed in Section 3 is established as follows.

Fig. 3.14(a) illustrate a sphere/rigid wall impact system and its equivalent mass-spring representation. The contact dynamic system can be expressed as

$$M\ddot{u} + F_n(u) = 0 \tag{3.78}$$



(a)

(b)

Figure 3.14: Rigid sphere/wall impact: (a) An equivalent mass-spring system; (b) Two linearised systems

where M is the mass of the sphere; and F_n the contact force defined by the contact laws as

$$F_n(\delta) = \begin{cases} k_n \delta^m & \text{if } \delta > 0 \\ 0 & \text{if } \delta \leq 0 \end{cases}$$

Also assume the initial impact velocity is v_0, and the initial position of the sphere is $u(0) = 0$. Note that F_n is a *nonlinear* function even for the linear contact model ($m = 1$). The implication of this nonlinearity is that the conventionally determined critical time step will become questionable in the current case. Furthermore, the system involving contact is a special nonlinear dynamic system in the sense that as soon as the contact couple separates, no further contact occurs.

Consequently, the critical time step should be determined such that the central difference algorithm can be either energy preserving or energy dissipating, i.e., the total energy after contact has been lost should be equal to, or less than, the energy prior to establishing contact. In other words, it requires that the computed exit velocity of the sphere should be equal to or less than the initial impact velocity.

3.7.1 Critical time step for nonlinear systems

For general nonlinear dynamic systems, a universally and practically applicable stability theory regarding the selection of time steps is still under development. In fact, the determination of the critical time step is problem-dependent and often difficult to implement. In addition, a *globally* applicable critical time step may be questionable as an adaptive time step scheme is generally more preferable than a constant time step scheme in nonlinear situations. Therefore, the evaluation of a *local limit* at each time step is often performed.

Conventionally, this local critical time step is determined by first linearizing the nonlinear system [3] at each time step and then taking the critical time step of the corresponding linearized (tangential) system as the current value. The linearized system takes the form of

$$M(\ddot{\Delta u}) + K_L(u)\,\Delta u = 0 \tag{3.79}$$

with

$$K_L(u) = \frac{d\{F_n(u)\}}{du} = m\,k_n\,u^{m-1}$$

Apparently, at the initial contact position $u = 0$, K_L will be zero for all models with $m > 1$, which gives rise to an infinite time step at the first time increment.

To avoid this problem, another linear system is proposed to be the reference system with the corresponding K_L defined by

$$K_L(u) = \frac{F_n(u_{max}) - F_n(u)}{u_{max} - u} = k_n \frac{u_{max}^m - u^m}{u_{max} - u} \tag{3.80}$$

Box 3.1: Central Difference Scheme With Adaptive Time Steps

- Acceleration at time step n:

$$a_n = -F_n(u_n)/M$$

- Velocity at time step $n + 1/2$:

$$v_{n+\frac{1}{2}} = v_{n-\frac{1}{2}} + a_n(\Delta t_{n-1} + \Delta t_n)/2$$

- Displacement at time step $n + 1$:

$$u_{n+1} = u_n + v_{n+\frac{1}{2}} \Delta t_n$$

where u_{max} is the maximum contact penetration determined by energy conservation as

$$u_{max} = \sqrt[m+1]{\frac{(m+1)M\,v_0^2}{2\,k_n}} \tag{3.81}$$

This linear system can be interpreted as a *secant* linear system compared with the previous *tangential* system, as illustrated in Fig. 3.14(b). The conventionally determined critical time step for this secant linear system is

$$\Delta t_{cr}(u) = 2\sqrt{\frac{M}{K_L(u)}} \tag{3.82}$$

Except that for the linear contact model, $\Delta t_{cr}(u)$ will not be constant.

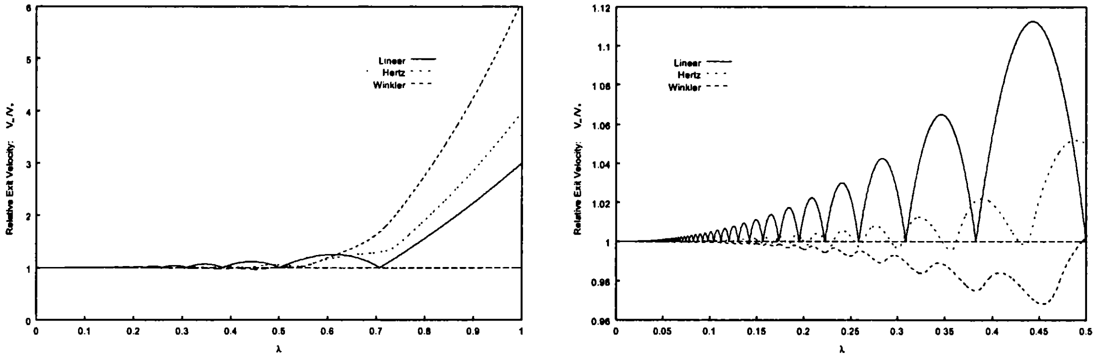
3.7.2 Time step scheme for power law based contact models

If Δt is chosen by scaling $\Delta t_{cr}(u)$ with a factor λ,

$$\Delta t = \lambda\,\Delta t_{cr}(u) \tag{3.83}$$

then the range of λ that leads to a stable solution for different power law based contact models will be numerically investigated below.

Box 3.1 lists the central difference algorithm with an adaptive time step. The time step at the n-th interval is taken as $\Delta t_n = \lambda\,\Delta t_{cr}(u_n)$ and the exit velocity of the sphere, v_{ex}, is recorded. Fig. 3.15 plots the relative exit velocity, v_{ex}/v_0, with various values of λ for the linear, Hertz



(a) (b)

Figure 3.15: Computed relative exit velocity of sphere vs. λ: (a) $\lambda \in (0,1]$; (b) $\lambda \in (0,0.5]$

and Winkler models. It is important to point out that these curves are independent of M, k_n and v_0, and are only affected by the value of m.

In Fig. 3.15(a), diverging solutions are observed for all models with $\lambda = 1$, i.e. $\Delta t = \Delta t_{cr}(u)$. In fact, except for $\lambda = 1/\sqrt{2}$ for the linear case, the solutions also diverge for $\lambda > 0.5$. A more interesting relationship for $\lambda \in (0, 0.5]$ is illustrated in Fig. 3.15(b).

For the linear model, the exact energy conservation is achieved at an infinite number of distinct values, termed *exact points*, and a relative exit velocity larger than 1 is obtained at the remainder. The maximum exit velocity in the interval of any two consecutive exact points rapidly reduces for smaller values of λ and these exact points also become much closer.

For the Winkler model, a slightly dissipated solution is always attained for $\lambda \in (0, 0.48)$, and gradually becomes more accurate when λ decreases. The behaviour of the Hertz model lies between those of the linear and Winkler models.

The above results are obtained under the conditions of $u(0) = 0$ and without damping. If $u(0) = u_0 \neq 0$ and some percentage of damping is applied, the response of the linear model may improve. Having taken into account the major relevant factors, we recommend a suitable range for λ in terms of stability and accuracy as

$$\lambda = 0.1 \sim 0.3 \tag{3.84}$$

3.8 Concluding Remarks

Algorithmic and modelling aspects are presented for contact resolution of disk/segment, disk/disk, sphere/facet and sphere/sphere contact. Following the establishment of geometric relations between contact pairs, four interaction laws, namely linear (Hooke), Hertz, Winkler and power laws, for the normal contact have been introduced. It is found that under the small contact deformation assumption, all the contact models considered can be unified under a general power law form, in which the linear and Hertz interaction laws are the two particular models of practical interest. As the disk or sphere is assumed to be rigid, all the interaction laws proposed are in fact penalty based which imposes the impenetrability condition in a weak form. The choice of penalty value for each model is therefore governed by the accuracy and stability conditions.

A modified classic Coulomb friction model is proposed for 3D contact problems, in which the analogue of the classic friction law with perfectly elasto-plastic theory is applied.

A new time stepping scheme is also developed, which can ensure both short and long term stability of the contact models. In addition, two different schemes that incorporate energy dissipation into the contact model have been reviewed.

Bibliography

[1] P. A. Cundall and R. D. Hart. Numerical modelling of discontinua. *Eng. Comp.*, 9:101–113, 1992.

[2] K. L. Johnson. *Contact Mechanics.* Cambridge University Press, Cambridge, UK, 1985.

[3] N. Petrinić. *Aspects of Discrete Element Modelling Involving Facet-to-Facet Contact Detection and Interaction.* Ph.D. Thesis, University of Wales Swansea, 1996.

[4] O. R. Walton. Numerical simulation of inelastic, frictional particle-particle interactions. In M.C. Roco, editor, *Particulate Two-Phase Flow*, pages 884–911. Butterworths, Boston, USA, 1992.

[5] S.P. Timoshenko and J.N. Goodier. *Theory of Elasticity.* 3rd ed. McGraw-Hill, NY, 1970.

[6] H. Poritsky. Stresses and deflections of cylindrical bodies in contact with application to contact of gears and locomotive wheels. *J. Applied Mechanics*, 17:191–201, 1950.

[7] J.T. Oden and J.A.C. Martins. Models and computational methods for dynamic friction phenomena. *Computer Methods in Applied Mechanics and Eng.*, 52:527–634, 1985.

[8] D. Perić and D.R.J. Owen. Computational model for 3D contact problems with friction based on the penalty method. *Int. J. Numer. Meth. Engng.*, 35:1289–1309, 1992.

[9] D. Perić and D.R.J. Owen. Finite element applications to the nonlinear mechanics of solids. *Rep. Prog. Phys.*, 61:1495–1574, 1998.

[10] O. R. Walton and R. L. Braun. Viscosity and temperature calculations for assemblies of inelastic frictional disks. *J. Rheology*, 30:949–980, 1986.

[11] J. Yu. *A contact interaction framework for numerical simulation of multi-body problems and aspects of damage and fracture for brittle materials.* Ph.D. Thesis, University of Wales Swansea, 1999.

Chapter 4

Contact Resolution for Non-circular Discrete Objects

4.1 Introduction

Disks and spheres are the most commonly used discrete objects in discrete element simulations, mainly due to their geometric simplicity and computational efficiency. However, they completely lack angularity, or in other words, they cannot provide resistance to rolling motion. It has been increasingly recognised that the angularity of discrete objects plays an important role in simulating the dynamic behaviour of many practical problems. Consequently some non-circular/spherical discrete objects have been introduced, such as polygons/polyhedrons (non-smooth boundary) and superquadrics (smooth boundary).

The difficulty associated with polygon/polygon contact is the proper handling of corner/corner contact in which gap functions and normal directions are not well defined. In the commonly used node/segment contact models the direction of normal force may exhibit a discontinuity when the contact pair is evolving. Recently a simple non-ambiguous corner/corner contact algorithm has been developed by Feng and Owen [1] within an energy based theoretical framework, in which the normal and tangential directions, magnitude and reference contact position of the contact forces are uniquely defined. This algorithm makes it possible to tackle more complex contact situations and particularly promotes the present development of a robust superquadric contact model.

Conventionally a superquadric may be preferred over a polygon since it can be defined with less information. In addition, for superquadric/superquadric contact, corner/corner contact situations do not exist, as is the case in polygon/polygon contact. Nevertheless, superquadrics, though mathematically elegant, are not widely used as substantial computational cost is involved in the contact resolution. For instance, an overlap check between two superquadrics needs to find the intersections of two nonlinear functions, which is a very expensive operation and may sometimes fail in finding a solution. For this reason, Williams and Pentland [2] proposed a more efficient discretised version but with a degraded accuracy.

Following their investigation, and taking advantage of Feng and Owen's model, a new algorithm is proposed in this work to perform contact resolution for 2D superquadrics, which features

- an adaptive sampling procedure to approximate a superquadric with a convex polygon;
- clipping two polygons by excluding those edges lying outside the overlap zone;
- an effective algorithm employed to search for intersections and overlap area of the clipped polygons;
- a robust polygon/polygon contact model adopted to determine the contact forces and directions.

In the remainder of the chapter, the algorithmic details will be presented, followed by numerical examples which will demonstrate the performance of the algorithm proposed.

4.2 Polygonal Approximation of a Superquadric

A 2D superquadric is defined by the following implicit function:

$$f(x,y) = \left(\frac{x}{a}\right)^m + \left(\frac{y}{b}\right)^m - 1 = 0 \tag{4.1}$$

where parameters a, b, and m (integer) determine the shape of the superquadric. If $m = 2$ an ellipse or disk (with $a = b$) is recovered; By varying a and b, the represented object can be stretched in the x and y directions. Fig. 4.1 shows that a superquadric becomes more 'blocky' as m increases from 2 to infinity.

When superquadrics are used as discrete elements, a computational bottleneck is associated with the searching for intersections of two nonlinear functions, as mentioned earlier. A common practice is to approximate a superquadric with a convex polygon.

4.2.1 Uniformly spaced sampling

In [2], the closed region of a 2D superquadric is separated into two sub-regions, each of which is described by a single valued function

$$y = \pm b \sqrt[m]{1 - \left(\frac{x}{a}\right)^m} \tag{4.2}$$

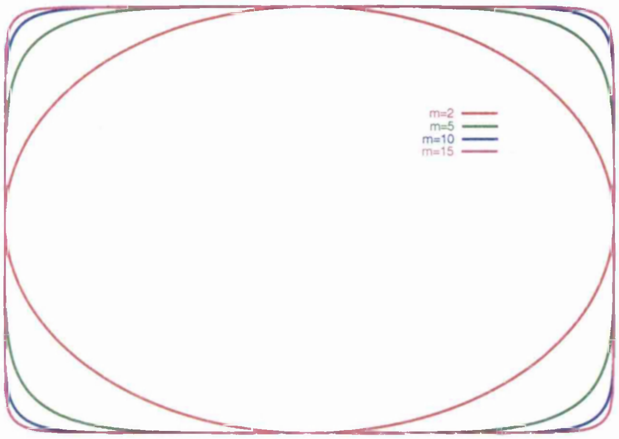


Figure 4.1: Superquadrics with a=2, b=1, m=2,5,10,15 respectively

Each half-region is discretised with a uniformly spaced sampling, Δx, along the x axis. A simple linear interpolation between boundary points is considered. Then a search algorithm is applied to the boundary points. Once a search point is determined to lie between two of the discretised x coordinates, all other boundary points can be eliminated from the search. This approach is simple and efficient, particularly for the subsequent contact detection, but the approximation accuracy, especially around large curvature areas, may be significantly degraded as a result of the uniformly spaced sampling, as illustrated in Fig. 4.4(a).

4.2.2 Adaptive sampling

An adaptive sampling procedure is employed in this work in order to achieve a better polygonal representation of a superquadric. In principle, a superquadric could be approximated by either an 'inscribed' or 'circumscribed' polygon, as illustrated in Fig. 4.2, but the former is more rational and simpler to generate. Due to symmetry, only one quarter of the superquadric needs to be considered, which is equivalent to approximating the superquadric curve with piecewise linear segments. The most commonly used method is by means of *curve subdivision*. As a prerequisite of the procedure the superquadric is expressed in a parametric form

$$P(t) = (x(t), y(t)) : \begin{cases} x(t) = a\cos^{\frac{m}{2}}(t) \\ y(t) = b\sin^{\frac{m}{2}}(t) \end{cases} \tag{4.3}$$

Note that the above expression is simple but may not necessarily be the best choice. To facilitate the description of the procedure, the concept of *curve flatness* is first introduced. Suppose that $P(t)(t \in [t_0, t_1])$ is part of the curve, with $P_0 = P(t_0)$ and $P_1 = P(t_1)$ as the

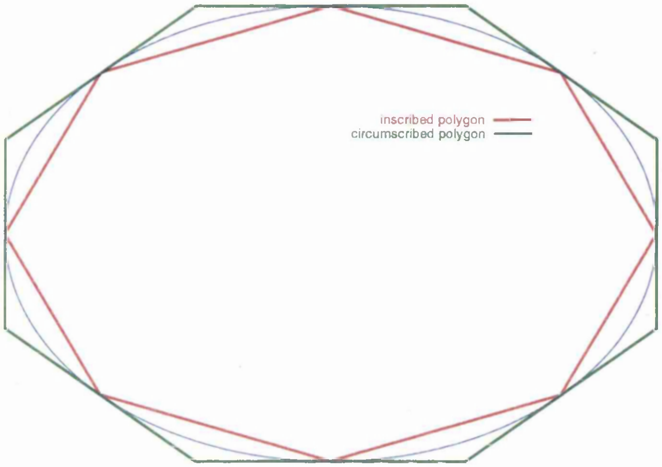


Figure 4.2: Inscribed and circumscribed polygons

two end points. The flatness of the curve is defined as

$$f = \delta_{max}/l \tag{4.4}$$

where δ_{max} is the maximum distance between the curve and the line segment $\overline{P_0P_1}$, and l is the length of $\overline{P_0P_1}$, as illustrated in Fig. 4.3. However, δ_{max} is relatively difficult to evaluate, and is thus replaced by the distance δ of the middle point (in the parametric space) P_2 to the segment. The curve is considered to be 'flat' if the following condition is satisfied

$$\delta/l < \tau \tag{4.5}$$

where τ is a given tolerance.

The sampling starts by choosing the two end points of the curve P_0 and P_1 as the two vertices

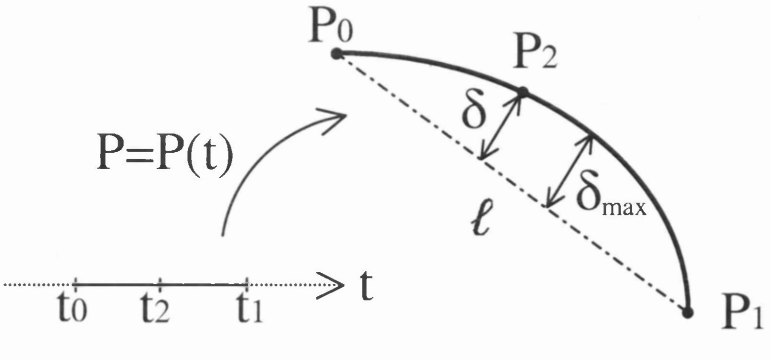


Figure 4.3: Curve flatness test

Box 4.1: Adaptive Sampling

```
• Given parameter $t_0$ and $t_1$, tolerance $\tau$;
• Compute $P_0 = P(t_0)$,   $P_1 = P(t_1)$;
• Output $P_0$;
• Call function: adaptive sampling($t_0, t_1, P_0, P_1$);

Function: adaptive sampling($t_0, t_1, P_0, P_1$)
• Compute $t_2 = (t_0 + t_1)/2$,   $P_2 = P(t_2)$;
• If(flat($P_0, P_1, P_2$)) (refer to Eq.(4.5)), output $P_2$;
  else {
      Call function: adaptive sampling($t_0, t_2, P_0, P_2$);
      Call function: adaptive sampling($t_2, t_1, P_2, P_1$).
  }
```

of the initial approximation polygon. The 'middle' point P_2 is evaluated as

$$P_2 = P(t_2); \qquad t_2 = (t_0 + t_1)/2 \tag{4.6}$$

The flatness check (4.5) is then applied to determine if the curve needs to be subdivided. The procedure terminates if the flatness condition is satisfied; otherwise P_2 is output as the third vertex, with the initial curve $P(t)$ being subdivided into two parts: $t \in [t_0, t_2])$ and $t \in [t_2, t_1])$. The sampling procedure is then recursively applied to each part of the curve until they become 'flat'. All the points/vertices output during the recursive subdivision define an 'inscribed' polygon of the superquadric.

One of the nice features of the adaptive sampling approach (summarised in Box 4.1) is that all the vertices are output in the right (anticlockwise) order for defining the polygon, therefore no vertex re-arrangement is required.

The adaptively sampled polygon, as shown in Fig. 4.4(b), is clearly a better approximation than the uniformly spaced sampled polygon (Fig. 4.4(a)), though the same number of vertices are defined.

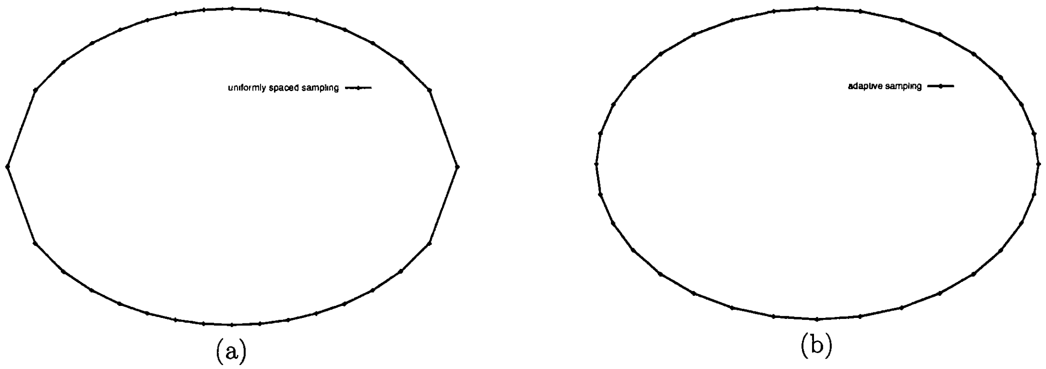


Figure 4.4: 32-sided polygonal approximation of a superquadric: (a) uniformly spaced sampling; (b) adaptive sampling.

4.3 Intersections of Two Polygons

Once the superquadrics are represented with polygons, a contact detection algorithm is applied to identify the pairs with potential contact. Subsequently the contact resolution is performed, by searching for intersections of the contacting polygons first, followed by employing the polygon/polygon contact model.

Assume that two polygons, P and Q, to be considered have n and m edges/vertices respectively. A naive algorithm for resolving the intersections of any two polygons can be readily formulated by sequentially checking the intersection of each edge of one polygon with each edge of the other polygon, and its order of complexity is thus nm.

4.3.1 A linear algorithm

A linear algorithm is proposed by O'Rourke *et al* [3], where two special pointers, distinguishing one edge on each polygon, are maintained. The pointers are advanced around the polygons such that their edges 'chase' one another, searching for the intersection points. All the intersection points can be found within two cycles around the polygons, and thus the algorithm achieves a linear complexity of $2(n+m)$.

Starting from two arbitrarily chosen edges, each from either polygon, e.g. $\boldsymbol{p} \in P$ and $\boldsymbol{q} \in Q$, the linear algorithm comprises a set of 'advancing rules' embedded in a repeat loop. At each step, two edges are checked for intersection. If an intersection is found, the incremental overlap area is also computed. Then the edge(s) of the polygon(s) is(are) further advanced following the advancing rules. When the loop is complete, all the intersection points, if they exist, and the overlap area are readily obtained. The difficult issue for implementation is

Table 4.1: Advancing Rules

	$s < 0$	$s = 0$	$0 < s < 1$	$s = 1$	$s > 1$
$t < 0$	c	q	p	p	p
$t = 0$	p	c	c	p	p
$0 < t < 1$	q	c	c	p	p
$t = 1$	q	q	q	p,q	p
$t > 1$	q	q	q	q	c

how to handle degenerated cases where two edges intersect at one or two of their vertices, or are co-linear. Meanwhile, numerical roundoff errors need to be carefully handled in order to achieve a robust implementation.

This work adopts the linear algorithm in a modified form. The equations of edges $\boldsymbol{p}$ and $\boldsymbol{q}$ can be expressed in their parametric form as

$$p(s) = p_s + (p_e - p_s)s \tag{4.7}$$

$$q(t) = q_s + (q_e - q_s)t \tag{4.8}$$

where s and t are parameters; and p_s, q_s and p_e, q_e are the start and end points of the edges respectively. By setting $p(s) = q(t)$, a pair (s, t) can be found which corresponds to the intersection of the two lines where $\boldsymbol{p}$ and $\boldsymbol{q}$ lie. The edges intersect only if $0 \leq t \leq 1$ and $0 \leq s \leq 1$. Table 4.1 presents the advancing rules with respect to different values of s and t, in which 'p' stands for 'advancing next edge of polygon P', and 'q' stands for 'advancing next edge of polygon Q', while 'c' refers to 'checking the relative position of the

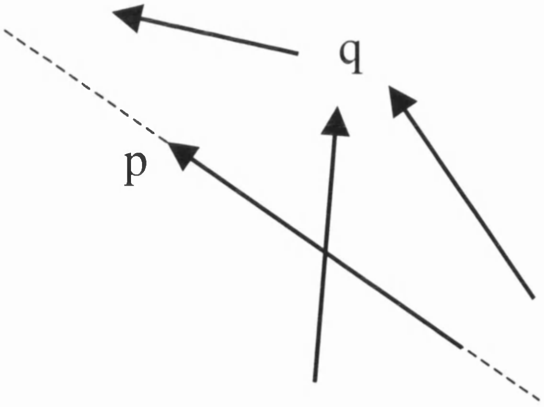


Figure 4.5: Advancing q: q_e is on the right side of p

Box 4.2: Linear Algorithm for Convex Polygon Intersections

```
• Given two polygons $P$ of $n$ edges/vertices and $Q$ of $m$ edges/vertices;
• Choose one edge $\boldsymbol{p} \in P$ and another edge $\boldsymbol{q} \in Q$ arbitrarily;
• Set $k = 1$;
• Do {
    – Solve Eqs.(4.7) and (4.8) for parameters $s$ and $t$;
    – If $0 \leq s \leq$ and $0 \leq t \leq 1$, an intersection point $C$ is found. If the
      first intersection point is found for the second time, the procedure
      should be terminated; else output $C$;
    – Advance $p$ or $q$ according to the rules listed in Table 4.1;
    – $k \leftarrow k + 1$;
  }while $k < 2(n + m)$
```

two edges: advancing q if q_e is on the right side of p; advancing p otherwise' (see Fig. 4.5 for an illustration). Box 4.2 outlines the algorithm based on the new advancing rules for non-degenerated cases.

Remark 1: The linear algorithm is clearly superior to the naive one for large n and m, but it is not the case for $n, m \leq 4$.

Remark 2: All the intersection points may be found in less than two cycles. In fact, the loop can be terminated if the first intersection point is found for the second time. If no overlap exits, however, two cycles around the polygons still need to be performed, indicating that the efficiency of the algorithm could possibly be improved.

4.3.2 Modification

Though the linear algorithm provides an efficient procedure to search for intersections of two polygons, the required computational cost still dominates the total cost of the contact resolution. In fact, the overlap of two superquadrics only occurs in a very small zone in discrete element simulations. Hence if the possible overlap zone can be identified, the searching for intersection may be performed only for the edges lying in the zone. This is achieved through the following two steps:

1. The possible overlap zone (rectangle) is determined by the overlap area of the bounding boxes of two polygons under consideration;
2. By clipping the polygons against this zone and discarding edges lying outside of the zone, the linear algorithm described in the previous subsection is employed to search for intersections of the two polygons.

Polygon clipping against a rectangular window is a fundamental operation in computer graphics and a number of effective algorithms have been proposed. The Sutherland-Hodgman polygon-clipping algorithm is employed and slightly modified in the sense that only those edges with two ends lying outside of the clipping zone are excluded. Due to the small clipping (overlap) zone, the two polygons are likely to be clipped into two polylines with very few edges.

Note that with the above modification two polygons without overlap may be detected prior to the intersection procedure, if the bounding boxes of polygons do not overlap; or if one of the polygons lies completely outside of the clipping zone.

As a result, the performance of the algorithm can be significantly improved. Tests show that the modified algorithm achieves a 4-times speedup in general in comparison to the original approach.

4.4 Contact Forces and Directions

Polygon/polygon contact is a typical corner/corner contact problem, or its special case, such as corner/edge contact, or its extensions for more complex contact situations. The corresponding contact forces and directions are computed by utilising the corner/corner contact model developed by Feng and Owen [1]. The algorithmic aspect is outlined below.

4.4.1 Contact geometry

Suppose that two polygons P and Q are in contact as shown in Fig. 4.6(a), where p and q are two penetrating vertices, and two sets of edges connecting p and q intersect at points g and h with coordinates $\boldsymbol{x}_g = (x_g, y_g)$ and $\boldsymbol{x}_h = (x_h, y_h)$ respectively. Each polygon has both translational and rotational motions. The position of polygon P can be fully described by the coordinates of vertex p, $\boldsymbol{x}_p = \{x_p, y_p\}$, and the rotational angle θ with respect to its original position.

The overlap area, S, of the two polygons is determined by four points p, h, q and g. As points g and h are determined by the position of polygon P while polygon Q is assumed fully fixed, S is therefore a function of $\boldsymbol{x}_p$ and θ, i.e. $S = S(\boldsymbol{x}_p, \theta)$.

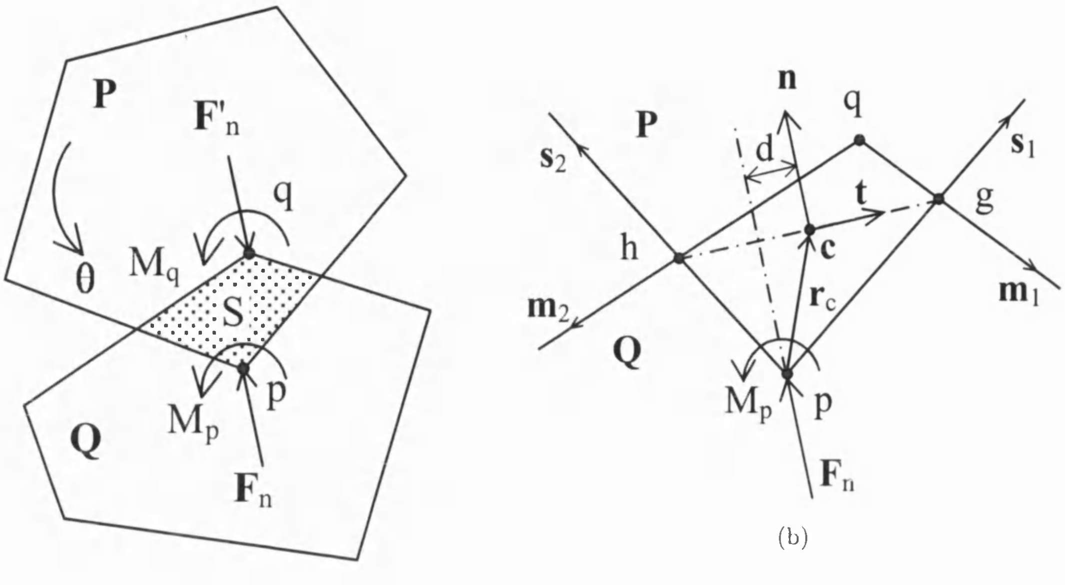


Figure 4.6: (a) Corner-corner contact of two polygons; (b) Normal contact model

4.4.2 Definition of normal contact forces

An important assumption made is that a contact energy potential W is associated with the contact overlap area

$$W = W(S) \tag{4.9}$$

where $W(S)$ is assumed to be a monotone increasing function of the overlap area S. Then the normal contact force $\boldsymbol{F}_n$, exerted on vertex p of polygon P, is defined as

$$\boldsymbol{F}_n = -\frac{\partial W(S)}{\partial \boldsymbol{x}_p} = -\nabla_{x_p} W(S) \tag{4.10}$$

Mathematically, it means that the normal force should be applied along a direction that can decrease the contact energy $W(S)$ with the maximum rate, i.e. the negative gradient of $W(S)$. Eq. (4.10) further leads to

$$\boldsymbol{F}_n = -\frac{dW(S)}{dS} \cdot \frac{\partial S(\boldsymbol{x}_p, \theta)}{\partial \boldsymbol{x}_p} = -W'(S) \cdot \nabla_{x_p} S = \|\boldsymbol{F}_n\| \, \boldsymbol{n} \tag{4.11}$$

where

$$\boldsymbol{n} = -\frac{\nabla_{x_p} S}{\|\nabla_{x_p} S\|} \tag{4.12}$$

$$\|\boldsymbol{F}_n\| = W'(S)\|\nabla_{x_p} S\| \tag{4.13}$$

Eqs. (4.11) and (4.12) both reveal that $\boldsymbol{n}$ is the normal direction and geometrically the direction by which moving the body along reduces the overlap area S most effectively, while Eq. (4.13) provides a scheme to determine the magnitude of $\boldsymbol{F}_n$.

If the object has translational motion only, the normal force can be applied at any point. As rotational motion is also allowed, an additional moment must be simultaneously present. Since vertex p is chosen as the reference point, the force is assumed to be applied at this point, but an associated moment, M_p, should be applied as well. Following the same argument as for the definition of the normal force, the moment M_p can be defined as follows

$$M_p = -\frac{\partial W(S)}{\partial \theta} = -\frac{dW(S)}{dS} \cdot \frac{\partial S(\boldsymbol{x}_p, \theta)}{\partial \theta} = -W'(S)S_{,\theta} \tag{4.14}$$

The pair $\{\boldsymbol{F}_n, M_p\}$ constitutes a force-moment system acting at point p. According to elementary statics theory, this force system can be equivalently replaced by a single force $\boldsymbol{F}_n$ but acting at a different position. This unique and preferable position is termed the *reference contact point* and the corresponding offset distance from p, as illustrated in Fig. 4.6(b), is determined by

$$d = M_p / \|\boldsymbol{F}_n\| \tag{4.15}$$

Note that the normal force $\boldsymbol{F}'_n$ acting on polygon Q can be applied at point q and its magnitude is equal to $-\boldsymbol{F}_n$. However the associated moment M_q at the point is not equal to M_p but should be a value such that the resulting force-moment system $(\boldsymbol{F}'_n, M_q)$ balances $(\boldsymbol{F}_n, M_p)$. If $\boldsymbol{F}'_n$ is applied at the contact point (on the second polygon) as introduced above, no accompanying moment is additionally needed.

4.4.3 Normal and tangential directions

The two intersection points g and h determine a direction $\overrightarrow{hg}$=$\Delta\boldsymbol{x}_{gh} = \boldsymbol{x}_g - \boldsymbol{x}_h$, whose unit vector, $\boldsymbol{t}$, is

$$\boldsymbol{t} = \frac{1}{b_w}\{\Delta x_{gh}, \Delta y_{gh}\} \tag{4.16}$$

in which $\Delta x_{gh} = x_g - x_h$, $\Delta y_{gh} = y_g - y_h$; and b_w is the distance between the two intersection points:

$$b_w = \|\Delta\boldsymbol{x}_{gh}\| = \sqrt{\Delta x_{gh}^2 + \Delta y_{gh}^2} \tag{4.17}$$

By assuming that vertex q is the origin of the coordinate system, the overlap area S can be expressed as

$$S = \frac{1}{2}(x_p\,\Delta y_{gh} - y_p\,\Delta x_{gh}) \tag{4.18}$$

Then it can be shown that

$$\frac{\partial S}{\partial x_p} = \Delta y_{gh} \tag{4.19}$$

and

$$\frac{\partial S}{\partial y_p} = -\Delta x_{gh} \tag{4.20}$$

Thus,

$$\nabla_{x_p} S = \{\Delta y_{gh}, -\Delta x_{gh}\} \tag{4.21}$$

Consequently,

$$\|\nabla_{x_p} S\| = \sqrt{\Delta y_{gh}^2 + \Delta x_{gh}^2} = b_w \tag{4.22}$$

and

$$\boldsymbol{n} = -\frac{\nabla_{x_p} S}{\|\nabla_{x_p} S\|} = \frac{1}{b_w}\{-\Delta y_{gh}, \Delta x_{gh}\} \tag{4.23}$$

Clearly $\boldsymbol{n} \cdot \boldsymbol{t} = 0$, i.e. the normal contact direction is the direction that is perpendicular to the line linking the two intersection points. In other words, $\boldsymbol{t}$ defines the tangential direction.

For the important role played in the current model, the line connecting two intersection points g and h, $\overline{gh}$, is termed the *contact line*, and its length b_w is referred to as the *contact width.*

4.4.4 Reference contact point

To locate the position of the reference contact point, the expression for computing the associated moment M_p at vertex p needs to be established. It is shown that

$$\frac{\partial S}{\partial \theta} = -(x_c - x_p)\Delta x_{gh} - (y_c - y_p)\Delta y_{gh} \tag{4.24}$$

in which

$$x_c = (x_g + x_h)/2, \qquad y_c = (y_g + y_h)/2 \tag{4.25}$$

Obviously $\boldsymbol{x}_c = (x_c, y_c)$ are the coordinates of the middle point, c, of line $\overline{gh}$. Denoting vector $\overrightarrow{pc}$ by $\boldsymbol{r}_c = \boldsymbol{x}_c - \boldsymbol{x}_p$, Eq. (4.24) can be rewritten as

$$\frac{\partial S}{\partial \theta} = -b_w \boldsymbol{r}_c \times \boldsymbol{n} \tag{4.26}$$

Thus

$$M_p = -W'(S)\frac{\partial S}{\partial \theta} = \boldsymbol{r}_c \times \boldsymbol{F}_n \tag{4.27}$$

i.e. M_p is exactly the moment produced by $\boldsymbol{F}_n$ about point p when $\boldsymbol{F}_n$ is applied at point c. Therefore it is concluded that point c, i.e. the middle point of the contact line $\overline{gh}$, is the *reference contact point.*

Table 4.2: Several forms of $W(S)$ and $\|\mathbf{F_n}\|$

	$W(S)$	$W'(S)$	$\|\boldsymbol{F}_n\|$
Linear	$k_n S$	k_n	$k_n b_w$
Hertz-type	$\frac{2}{3} k_n S^{3/2}$	$k_n S^{1/2}$	$k_n S^{1/2} b_w$
Power	$k_n S^m / m$	$k_n S^{m-1}$	$k_n S^{m-1} b_w$

4.4.5 Normal force magnitude and choice of energy function

With the result of (4.22) applied to (4.13), the magnitude of the normal force $\boldsymbol{F}_n$ is further expressed as

$$\|\boldsymbol{F}_n\| = b_w W'(S) \tag{4.28}$$

Thus, different selections of energy function, W(S), or more precisely its derivatives, $W'(S)$, will lead to different magnitudes of the normal force, but the force direction will remain the same. Several possible options for $W(S)$ and $W'(S)$ are suggested and listed in Table 4.2, where the parameter k_n is the penalty coefficient.

4.4.6 Extensions

Although developed for corner/corner contact, the above model can also be extended for more general contact cases. Some are illustrated in Fig. 4.7.

In Fig. 4.7(a), the two penetrating edges of polygon Q are co-linear, so the corner/corner contact model readily reduces to the special case of corner/edge contact, with the normal direction being perpendicular to the edge intersected.

For the contact where more than one corner of a polygon is penetrating into another polygon, as shown in Fig. 4.7(b), the overlap area S can be obtained as

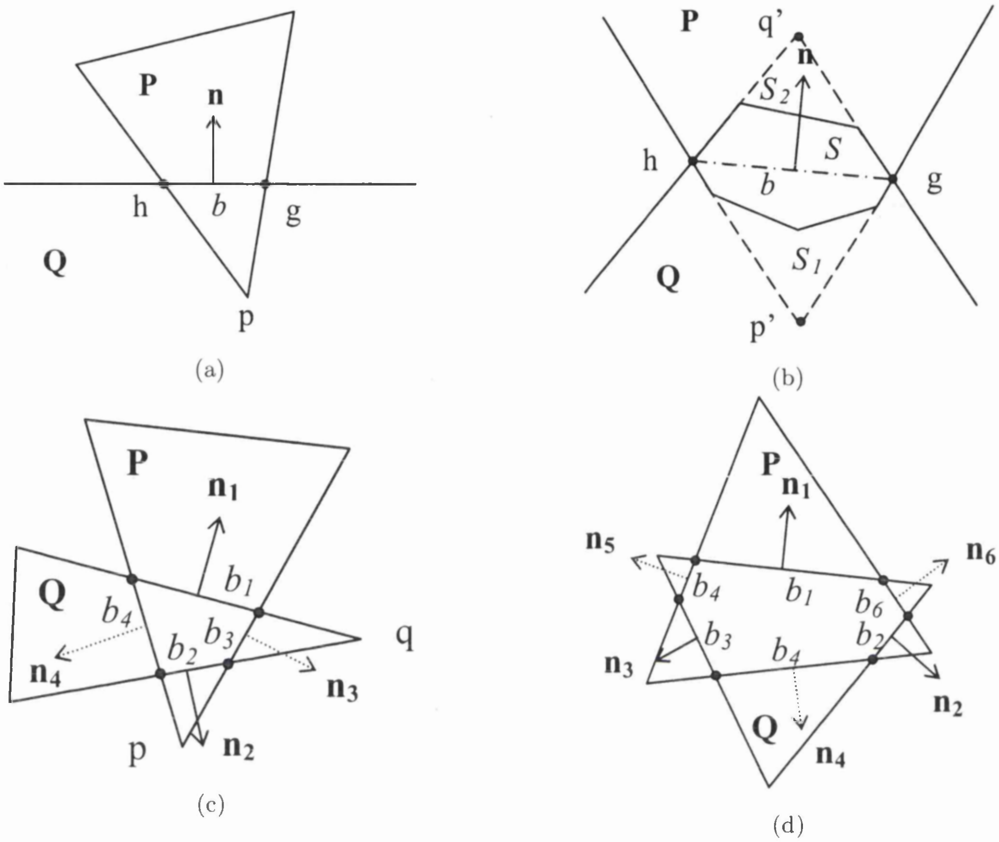
$$S = \bar{S} - S_1 - S_2 \tag{4.29}$$

where $\bar{S}$ is the whole area enclosed by lines $p' - g - q' - h - p'$. As S_1 and S_2 are independent of $\boldsymbol{x}_{p'}$ and θ,

$$\frac{\partial S}{\partial \boldsymbol{x}_{p'}} = \frac{\partial \bar{S}}{\partial \boldsymbol{x}_{p'}}, \quad \frac{\partial S}{\partial \theta} = \frac{\partial \bar{S}}{\partial \theta} \tag{4.30}$$

the corner/corner model can be extended to cover this contact situation.

If one polygon penetrates through another polygon as illustrated in Figs. 4.7(c) and (d), it can be decomposed into several corner/corner contacts, and then the corner/corner model is



(a)

(b)

(c)

(d)

Figure 4.7: Other contact situations

applied individually. For instance, the normal contact force acting on polygon P in Fig. 4.7(c) consists of two forces applied at two different contact points, and then the total force is

$$\boldsymbol{F}_n = -W'(S)(b_1\boldsymbol{n}_1 + b_2\boldsymbol{n}_2) \tag{4.31}$$

The total force acting on polygon Q can be computed as

$$\boldsymbol{F}_n^{'} = -W'(S)(b_3\boldsymbol{n}_3 + b_4\boldsymbol{n}_4) \tag{4.32}$$

It is noted that the two forces $\boldsymbol{F}_n$ and $\boldsymbol{F}_n^{'}$ are in fact equal in magnitude, opposite in direction and co-linear, thereby no net moment is produced by the couple. The contact case in Fig. 4.7(d) can be similarly handled.

4.5 Illustrations

A 2D hopper filling problem is provided to demonstrate the performance of the proposed algorithm. Particles are superquadrics with $a = 4.5mm, b = 3.5mm$ and $m = 2, 5, 15$ respectively; as well as 3-sided and 4-sided polygons. They are initially randomly packed at the top of the hopper and drop down under the action of gravity. The normal contact force is modelled by the linear law with a viscous damping (damping coefficient 0.8) applied; while the tangential force is given by Coulomb's law with a frictional coefficient of 0.5. Fig. 4.8~4.12 illustrate the four filling phases for the selected superquadrics and polygons respectively. The CPU cost for each simulation is about 500 - 800 seconds, depending on the particle type used. Generally speaking, when a superquadric becomes more 'blocky' as m gets larger, fewer vertices/edges are used to represent the superquadric with a polygon, resulting in less CPU cost in simulation.

4.6 Concluding Remarks

An efficient and robust algorithm is proposed in this work for contact resolution of 2D superquadrics, which is accomplished by firstly representing any superquadric with a convex polygon by means of adaptive sampling; then clipping two polygons to identify the overlap zone; and employing an efficient linear algorithm to search for intersections and overlap area of the clipped polygons; finally utilising an established polygon/polygon contact framework to determine the contact forces and directions. The accuracy of the algorithm is controlled by the number of sampling points used to represent the original superquadric with a polygon.

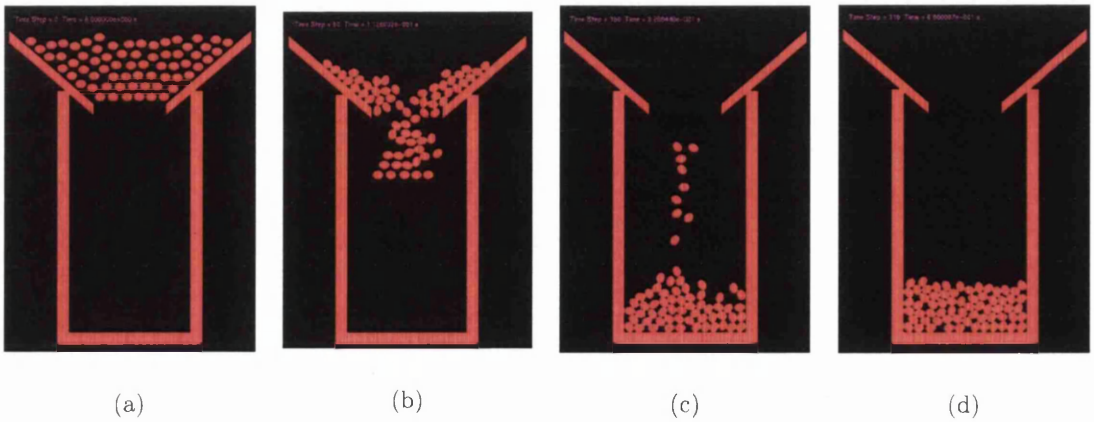
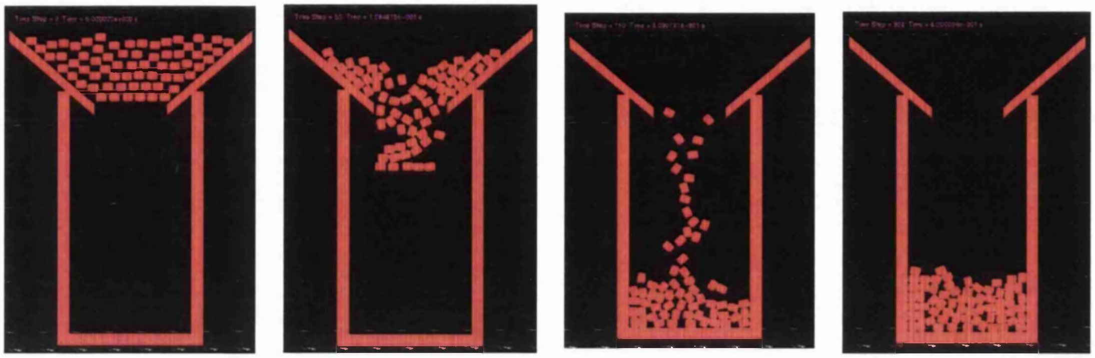
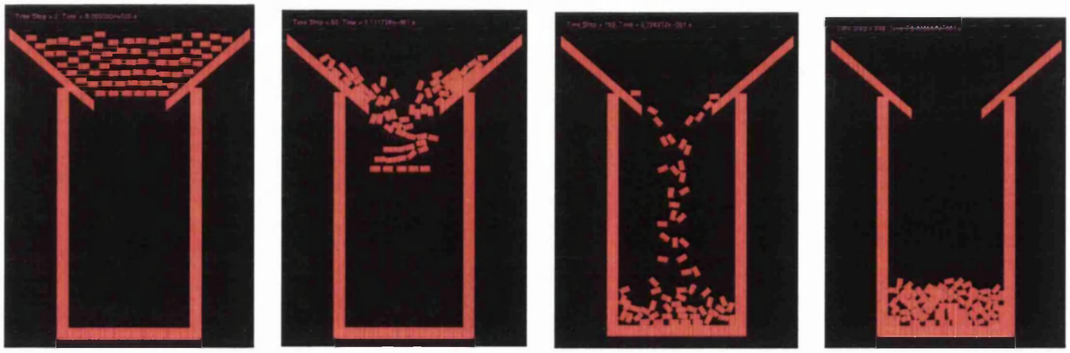

(a) (b) (c) (d)

Figure 4.8: Hopper filling with superquadrics of $m = 2$



(a) (b) (c) (d)

Figure 4.9: Hopper filling with superquadrics of $m = 5$



(a) (b) (c) (d)

Figure 4.10: Hopper filling with superquadrics of $m = 15$

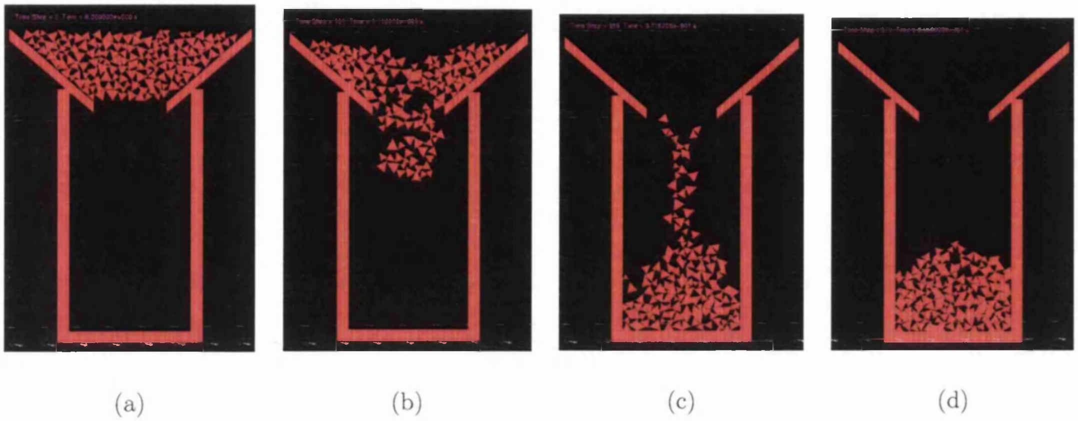

(a) (b) (c) (d)

Figure 4.11: Hopper filling with 3-sided polygons

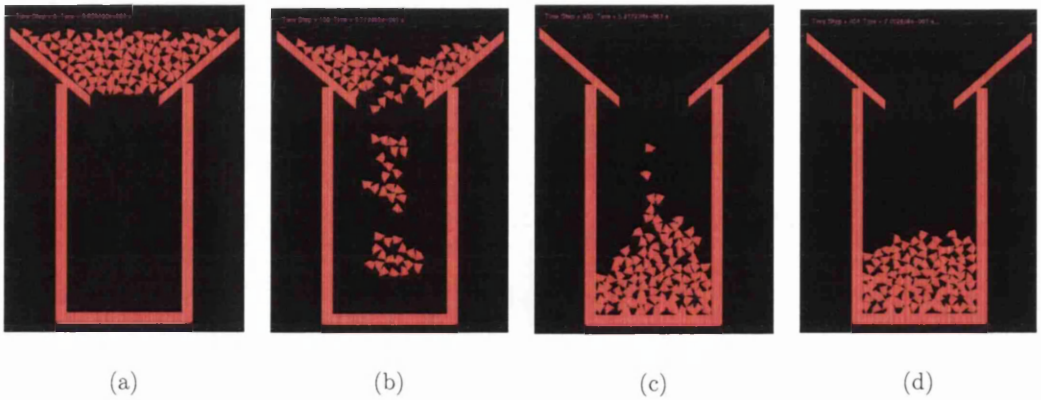

(a) (b) (c) (d)

Figure 4.12: Hopper filling with 4-sided polygons

Bibliography

[1] Y. T. Feng and D. R. J. Owen. A 2D polygon/polygon contact model: algorithmic aspects. *Int. J. Numer. Meth. Engng.*, 21: 265-277, 2004.

[2] J. R. Williams and A. Pentland. Superquadrics and model dynamics for discrete elements in interactive design. *Eng. Comp.*, 9:115-128, 1992.

[3] Joseph O'Rourke, Chi-Bin Chien, Thomas Olson and David Naddor. A new linear algorithm for intersecting convex polygons. *Computer Graphics and Image Processing*, 19:384-391, 1982.

Chapter 5

Packing of Disks, Ellipses and Polygons with the Advancing Front Technique

5.1 Introduction

In discrete element simulations, the pre-processing often involves the preparation of an initial random discrete object configuration that represents realistic situations. It is however not trivial to effectively generate such a random packing at a large scale, particularly when discrete objects of various sizes and geometric domains of different shapes are present.

Disk/sphere packing has been a long-standing research topic for centuries due to its academic and industrial importance. Since the 17th century, mathematicians have worked on a topic termed optimal geometric packing [1]. This classic packing problem concerns the arrangement without overlapping of equal size disks inside different geometrical domains to achieve the highest packing density. With the advent of computers in the late 1950's, computer simulation has since been extensively employed to generate different packing, used specially for the modelling of various materials. Bernal [2] was among the first to use computer simulated random packing of monosize spheres to model the structures and properties of liquids and amorphous metal alloys[3, 4, 5]. A recent application to the simulation of cementitious materials can be found in [6].

Disks/spheres may be sufficient to represent realistic situations in many applications, but more diverse shaped objects, such as ellipses and polygons, are sometimes required. Very limited work has been reported with regards to the development of efficient packing algorithms for ellipses and polygons.

In this chapter, random packing of such 2D discrete objects as disks, ellipses and polygons is discussed whilst sphere packing will be addressed in the next chapter.

5.2 Packing of Disks

In the case of equal size circular disks and rectangular containers, the initial packing may be implemented by two so called *lattice* packing: *square* and *hexagonal* packing, as shown in Fig. 5.1. Ignoring the gap between the disks and the boundary, the second arrangement (Fig. 5.1(b)) achieves an optimal packing, i.e. having the greatest density/volume fraction.

In many practical situations, however, different size disks have to be used. The existing packing algorithms (see e.g. [4, 6, 9, 10, 11]) are mainly of a *non-constructive* nature and therefore may involve substantial computer costs for large scale situations.

One approach sequentially generates disks by determining their central locations randomly and the disks which overlap with the existing ones are rejected. However, with the increase of volume fractions, the number of rejections also increases significantly, making the whole procedure very time-consuming. Meanwhile, the predefined disk size distribution may not be achieved due to the rejection of some disks.

Another investigation adopts a two-stage procedure. All disks are initially assumed to have a maximum radius and thus can have a lattice packing as illustrated in Fig. 5.1(a) or (b). Since the actual radii are smaller, there are usually relatively large gaps present between the disks. Later, the disks are compressed to reduce any gaps by pushing the boundaries toward the disks in one or several directions, as illustrated in Fig. 5.2.

A hopper is commonly used to mix different sizes of disks together (Fig. 5.3). The disks are initially packed regularly in layers at the top of the hopper. Under the action of gravity, they

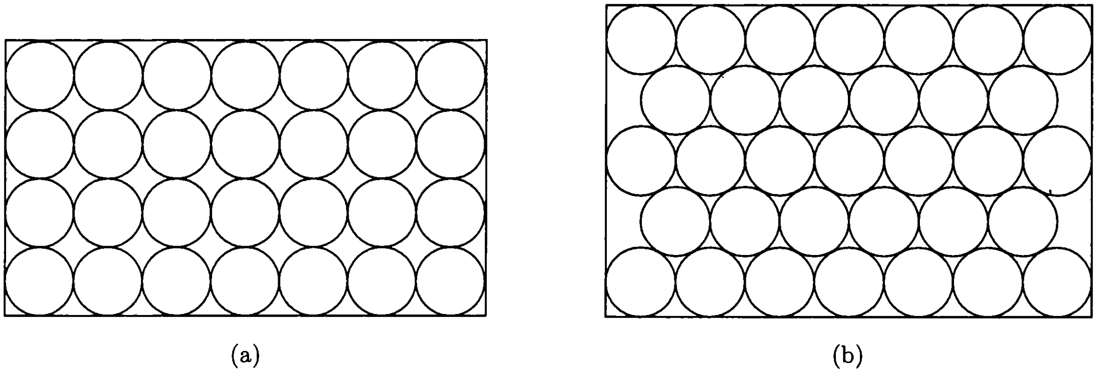


Figure 5.1: Two lattice packings: (a) square arrangement; (b) hexagonal arrangement

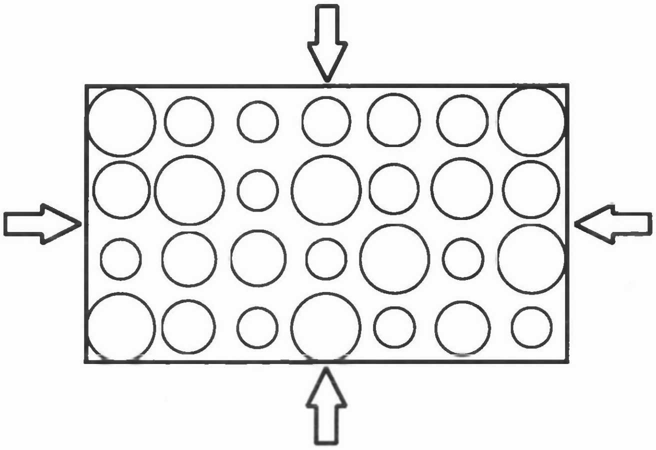


Figure 5.2: Different size disks first regularly packed and then compressed

fall through the chute and settle down with a random mixture of different size disks in the container beneath.

Both the compression of the boundary and the movement of disks under gravity force in the above two approaches are normally simulated by the same numerical procedure employed in the DEM analysis. However, the CPU time required to undertake this pre-processing can be *substantial*. For example, it takes 1905 CPU seconds to complete the whole procedure, shown in Fig. 5.3, from the initial regular pack (Fig. 5.3(a)) to the final mixture (Fig. 5.3(c)) on an Origin 2000 with one 195MHz R10000 processor, although only 63 disks are involved.

In this section, a novel numerical procedure is described to constructively generate a realistic random dense packing for large number of different sizes of disks within a geometric domain, motivated by the idea of the *advancing front technique* employed in mesh generation

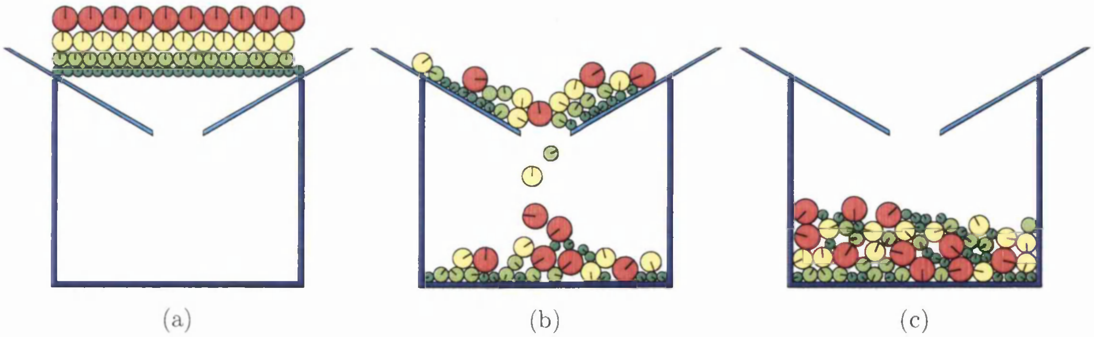


Figure 5.3: A hopper used to mix different size disks together

procedures [12]. Depending on whether the domain boundary is included in the initial front and how the front is formed, two different versions of the algorithm will be developed. The effectiveness of the algorithm in terms of the CPU cost will be numerically demonstrated.

5.2.1 Advancing front approach: closed form

Consider the problem of randomly filling a geometric domain with different size disks. Assume that the sizes can be randomly determined based on a given distributional function, and the domain is defined by a rectangle of $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$. The closed form advancing front algorithm first creates three disks which form an initial front. New disks are then generated by continuously advancing the front until the whole domain is filled.

The first three disks and initial front generation

The first three disks, denoted by D_1, D_2 and D_3, can be generated to have the densest packing (i.e. they are in touch with each other) and placed in the centre of the domain, as shown in Fig 5.4(a). A triangle $D_1D_2D_3$ is formed by joining their centres together. The third disk should be generated so that D_1, D_2 and D_3 form an anti-clockwise system. The corresponding three sides of the formed triangle are also assumed to have directions, with $D_1 \rightarrow D_2$, $D_2 \rightarrow D_3$ and $D_3 \rightarrow D_1$ as positive. The oriented polygon (the triangle) formed by these directed segments is termed the *initial front*, which can be represented, in terms of disks, as $D_1 \rightarrow D_2 \rightarrow D_3 \rightarrow D_1$. The region surrounded by the front is considered to have

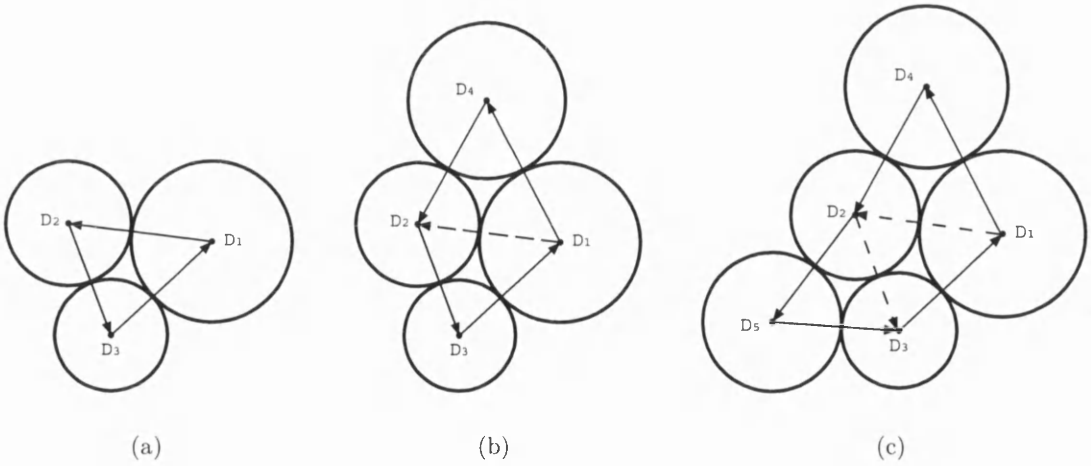


Figure 5.4: (a) Setup of the first three disks and initial front; (b) Generation of Disk 4 and update of the front; (c) Generation of Disk 5 and update of the front

been occupied by the disks. The defined directions of the frontal segments ensure that any new disk should be placed on the right hand side when travelling along the positive directions of the segments.

With the initial front established, new disks can be generated to fill the space by incrementally advancing the front until it completely covers the original domain. Details of the generation of the new disk and the update of the front are described below.

New disk generation and front update

Starting from the initial front, one segment, e.g. $D_1 \rightarrow D_2$, is chosen as the first *current active segment.* After having generated Disk 4 (D_4), which is in contact with both disks 1 and 2 and lies on the right hand side of segment $D_1 \rightarrow D_2$ as shown in Fig 5.4(b), segment $D_1 \rightarrow D_2$ is deleted from the initial front and two new segments, $D_1 \rightarrow D_4$ and $D_4 \rightarrow D_2$, are inserted into the front. The new front thus becomes $D_1 \rightarrow D_4 \rightarrow D_2 \rightarrow D_3 \rightarrow D_1$. Then segment $D_2 \rightarrow D_3$ is selected as the current active segment and Disk 5 (D_5) is generated in a similar manner, as shown in Fig. 5.4(c). The new front is obtained by deleting the chosen active segment $D_2 \rightarrow D_3$ from the previous front and inserting two new segments $D_2 \rightarrow D_5$ and $D_5 \rightarrow D_3$ into it.

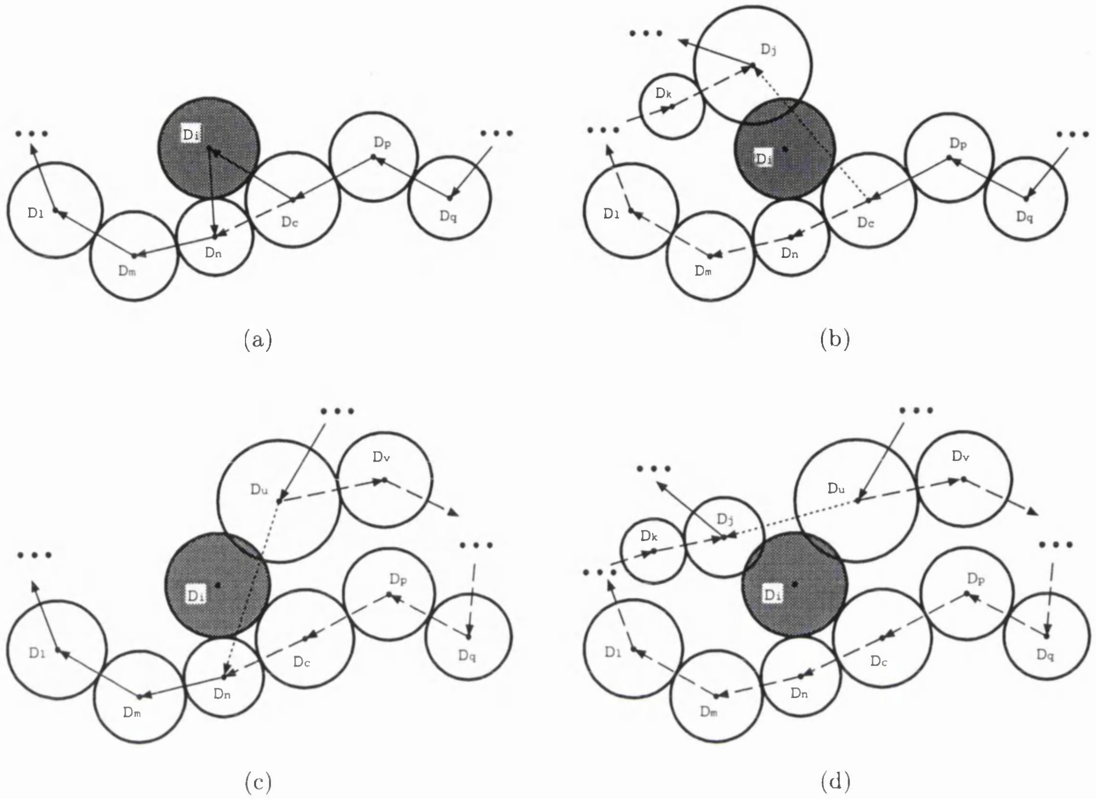
The above procedure can be performed until the whole domain is filled. In general cases, however, new disks generated may overlap with the existing disks on the front. A simple algorithm is designed to overcome the problem.

Suppose the selected active segment from the current front, ... $\rightarrow D_q \rightarrow D_p \rightarrow D_c \rightarrow D_n \rightarrow D_m \rightarrow D_l \rightarrow$ ..., is $D_c \rightarrow D_n$, as illustrated in Fig. 5.5. The segments/disks, ... $\rightarrow D_q \rightarrow D_p$, which are listed before $D_c \rightarrow D_n$, are termed the *preceding* segments/disks of $D_c \rightarrow D_n$ and accordingly the segments/disks $D_m \rightarrow D_l \rightarrow$ ... are termed the *subsequent* segments/disks. The algorithm includes two steps:

1. Determine the central coordinates of the new disk i that is in contact with both disks c and n. This is a simple geometric problem and no further explanation is necessary.

2. Check if the new disk i overlaps with any disk on the front, which may give rise to four different cases:

   (a) No overlapping occurs (Fig. 5.5(a)). In this case accept disk i, and mark it if its centre lies outside of the domain. Then delete the segment $D_c \rightarrow D_n$ from the front and insert two new segments, $D_c \rightarrow D_i$ and $D_i \rightarrow D_n$. The front is advanced as ... $\rightarrow D_q \rightarrow D_p \rightarrow D_c \rightarrow D_i \rightarrow D_n \rightarrow D_m \rightarrow D_l \rightarrow$ ....



(a) (b) (c) (d)

Figure 5.5: Four possible cases when generating a new disk

(b) Overlapping occurs with (at least) one disk on the subsequent front and no overlapping is detected for disks on the preceding front list (Fig. 5.5(b)). Suppose that disk j is the first encountered disk overlapping with disk i. In this case, delete the disks from n up to k in the front, join c and j to create an *intermediate active segment* $D_c \rightarrow D_j$. Then assume n to be j, go back to Steps 1 and 2 to re-determine the position of disk i (with the same radius) and recheck if the new position is acceptable.

(c) Overlapping occurs with (at least) one disk on the preceding front and no overlapping is detected for disks on the subsequent front list (Fig. 5.5(c)). Suppose that disk u is the first encountered disk overlapping with the disk i. In this case, delete the disks from v up to c from the front, join u and n to create an intermediate active segment $D_u \rightarrow D_n$. Then assume c to be j, and repeat Steps 1 and 2.

(d) Overlapping occurs with disks both on the preceding and subsequent front lists (Fig. 5.5(d)). Suppose that disks j and u are respectively the first encountered disks overlapping with disk i on the preceding and subsequent front lists. In this case, delete the disks from v up to k from the front, join u and j to create an intermediate active segment $D_u \rightarrow D_j$. Then assume c to be u and n to be j, and repeat Steps 1 and 2.

The following features can be derived from the above algorithm: A local optimal packing is achieved when adding a new disk since it is in contact with the two disks associated with the current (intermediate) active segment; As intermediate active segments are only temporarily presented and removed after new disks are successfully generated, the front is always closed, leaving no gap between disks on the front. The region inside the front represents the disk-filled domain and any new disk is placed outside the region.

Note that in Cases (b), (c) and (d), the deleted partial list of the front plus the intermediate segment encloses an unfilled region, and thereby forms a *'void'* or *'hole'* in the domain. Although no attempt is made in the above algorithm to fill the hole, it is possible to modify the algorithm so that in these cases the new disk D_i is attempted first to be placed into the hole. If not successful, then repeat the procedure outlined above.

Segment deactivation and procedure termination

Starting from a 3-active-segment front, the algorithm proposed above advances the front by choosing one active segment at a time to place a new disk that achieves a local optimal packing. If both centres of the two disks on the same segment are outside the domain to be filled, this segment becomes *deactivated* and will no longer be selected to generate a new disk later on, although there is a very small possibility around the domain corners that new valid disks can still be generated based on this segment.

When all the segments on the front become deactivated, the domain is considered completely filled and the procedure is terminated. Finally, all the generated disks should be validated to exclude those that have parts lying outside the rectangular domain (Fig. 5.6).

Further issues

Besides the features already mentioned, a number of issues related to the algorithm proposed above deserve further remarks.

- *Segment visit sequence.* The selection of the location and orientation of the first three disks can be very flexible. Similarly, the selection of the current active segment from the front can be chosen in several ways. Theoretically, the segments on the front can be visited in any order but a logical way of choosing an active segment will benefit, at

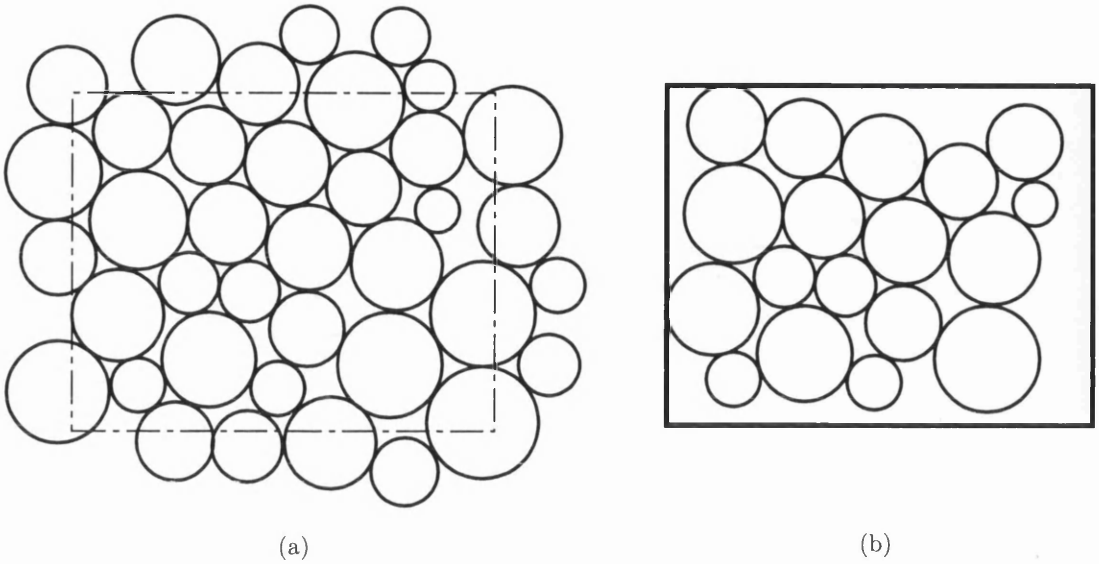
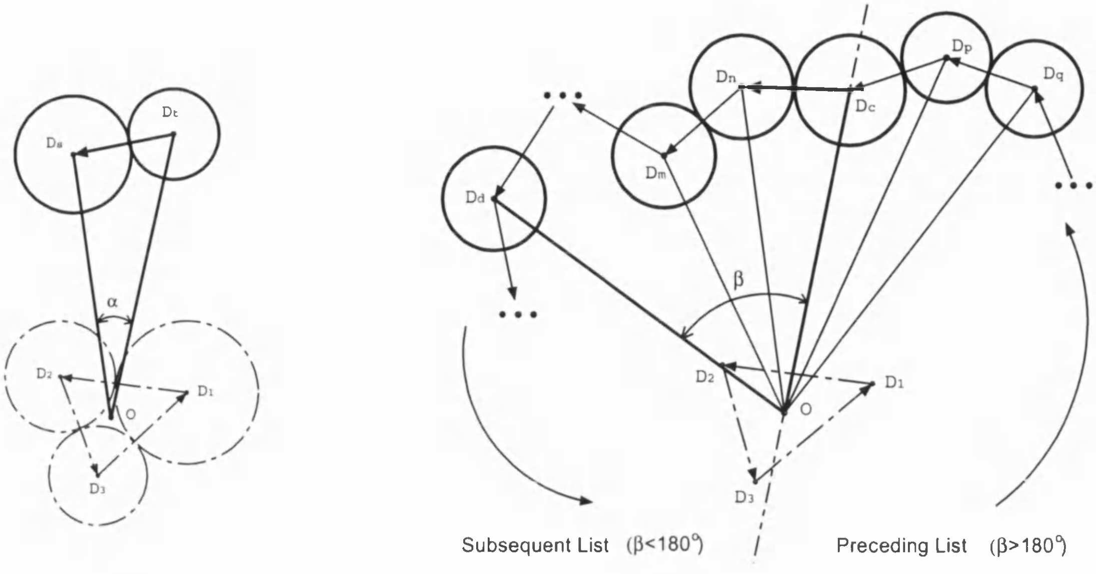


Figure 5.6: Final stage of the disk generation: (a) Disks generated; (b) Disks inside the domain.

least, the programming. In addition, different visit order of the frontal segments will result in different front shapes that will further affect the performance of the algorithm. Ideally, the front should be advanced evenly in all spatial directions, which would keep the front size as small as possible and at the same time reduce the chance that the cases shown in Fig. 5.5(b)∼(d) can occur, thereby decreasing the computational cost.

- *Definition of preceding/subsequent segment.* When a new disk is initially found to overlap with some disks on the front, it is essential to identify if these disks are on the *preceding* or *subsequent* segment lists of the current active segment so that a correct intermediate active segment can be temporarily formed to attain a new position of the disk, as demonstrated in Fig. 5.5(b)∼(d). Since the front is closed, it cannot determine, in a strict sense, whether one disk/segment is in the preceding or subsequent position relative to another segment. Therefore a sensible relative position clarification rule should be developed. A particular identification system is adopted in the present work as illustrated in Fig. 5.7.

  The idea of this identification system is to associate each segment on the current front with an angle formed by the segment with a point O, as shown in Fig. 5.7(a). The point is selected to be the geometric centre of the initial triangle (front) $\triangle D_1D_2D_3$. The associated angle for a segment with two disks D_s and D_t, is defined to be the angle $\alpha = \angle D_tOD_s$, which is assumed positive if points D_t, D_s and O are positioned in an anti-clockwise order, but negative otherwise.



(a)

(b)

Figure 5.7: Relative position identification system: (a) Definition of a segment angle α; (b) Definition of an accumulated angle β associated with disk D_d with respect to the current active segment $D_c \rightarrow D_n$

To identify the relative position of a disk D_d on the front against the current active segment $D_c \rightarrow D_n$, an angle β is associated with the disk and calculated by accumulating all the angles of the frontal segments situated between disks D_c and D_d in the current front list (Fig. 5.7(b)). If $\beta \leq 180^o$, this disk can be considered as a subsequent disk of the current active segment, but as a preceding disk otherwise.

For implementation, two separate checks, the forward and backward check, are introduced. The former starts to check the disks forward from D_m in the list and ends when an overlap is encountered or the accumulated angle of one disk is greater than 180^o, while the latter starts to check the disks backward from D_p and ends when the same condition is met.

Note that this relative position determination scheme plays an important part in the earlier stage of the disk generation when the front size is small, but becomes less important when more disks are generated as an overlapping is most likely to occur for disks local to the current active segment.

- *Computational cost.* The main cost associated with the generation of a new disk arises from the overlap check with all the disks in the current front. The front size, i.e. the number of disks/segments on the front, is growing as the increase of the number of disks. The size of the current front may be approximated by $O(\sqrt{n})$, where n is the number of disks generated. Therefore when adopting a simple linear search scheme that checks each disk on the front for possible overlap, the complexity of the algorithm for generating N disks will be in order of $O(N\sqrt{N})$. The cost can be reduced by employing an advanced search algorithm, for instance, the dynamic cell based search algorithm described in chapter 2, a complexity of $O(N)$ can be achieved.

- *Further tightening.* As the current algorithm can achieve only a local optimal packing, the disks may be further packed by means of, for instance, boundary compression and gravity compression.
- *Disadvantage.* The disadvantage of this algorithm is that relatively large gaps may be present around the boundary, as illustrated in Fig. 5.6(b). Although it is possible to remove some of these by applying additional approaches, it would be of practical importance if this problem can be dealt with directly within the algorithmic framework. This consideration has motivated the development of the second version of the algorithm, as presented below.

5.2.2 Advancing front approach: open form

The main idea of this development is to start generating the disks from the (bottom) boundary and the remaining disks are generated in a layer by layer manner. The fundamental difference from the previous version at the algorithmic level is that the front involved will be of an *open form.*

Consider again a rectangular domain. Denote the left, bottom, right and top boundaries as B_l, B_b, B_r and B_t respectively. In this version of the algorithm, the disks will be generated within the domain except for the upper boundary. The top boundary B_t will not be taken into account until the final stage, therefore can be viewed as the *'lid'* of the rectangular box. The first layer of disks will be placed against the bottom wall B_b, starting from the left corner. The walls are treated as disks with infinite radius. The initial front can be established as an open list: $B_l \rightarrow B_b \rightarrow B_r$, and $B_l \rightarrow B_b$ is selected as the first active segment.

After the first disk (denoted as D_1) is generated, the front becomes $B_l \rightarrow D_1 \rightarrow B_b \rightarrow B_r$. Then the active segment is chosen to be $D_1 \rightarrow B_2$ which leads to the generation of Disk 2 (or D_2). The front after D_2 is updated as $B_l \rightarrow D_1 \rightarrow D_2 \rightarrow B_b \rightarrow B_r$. The first layer of disks is completed when one disk, for instance, Disk D_n, touches the right wall. The corresponding front will be $B_l \rightarrow D_1 \rightarrow D_2 \rightarrow ... \rightarrow D_n \rightarrow B_r$. Note that the bottom wall B_b has been removed from the front. See Fig. 5.8(a) for an illustration.

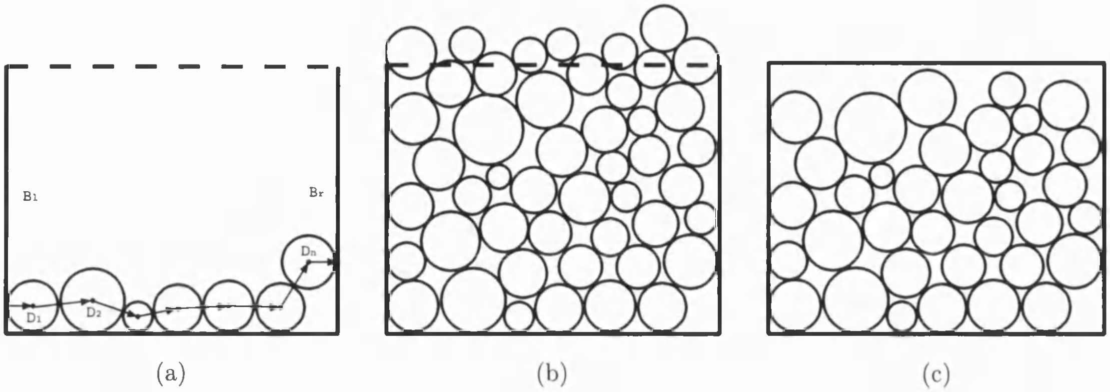


Figure 5.8: Advancing Front - Open form: (a) First layer of disks; (b) Disks generated; (c) Disks inside the domain.

The second layer can be built upon the first layer following a similar procedure as outlined in the previous closed form algorithm. In particular, when a new disk is generated, it should be checked for possible overlap with all the existing disks on the front. The same four cases could arise and the same actions should be taken accordingly.

The subsequent layers can be placed in a similar fashion. The domain is completely filled when all the disks on the front lie above the top boundary, as shown in Fig. 5.8(b). The final valid disks are obtained by excluding the disks lying above the top wall as illustrated in Fig. 5.8(c).

This open form algorithm inherits most of the features associated with the closed form version. Most of the issues discussed earlier are also applicable to the current situation. However, due to the fact that the front is now in an open form, the relative positions of preceding and subsequent disks/segments are clearly defined. No special identification system is therefore necessary. Most importantly, as the front size can be bounded by a constant value for a fairly well shaped front, the open form algorithm is inherently of a linear complexity of $O(N)$.

As the domain boundaries participate in the disk generation procedure, the open form algorithm is in general more complex in terms of implementation. In addition, the final disk distribution is affected to a certain degree by how the disks next to the boundaries are placed in the algorithm. In the current implementation, the first and last disks of each layer are always in touch with the left and right walls respectively. A different packing style will result if at each layer the starting active segment can be chosen from the middle of the front, or even selected randomly.

5.2.3 Packing under gravity

In both closed form and open form procedures, the local and global stability of generated disks under any external force is not taken into account. Therefore some disks may not be in equilibrium, for instance, under the action of gravity, and their positions may be easily altered. This situation can be handled within the current algorithmic framework by applying extra rules/conditions when determining the position of a new disk.

Suppose that gravity is considered and acts in the direction $\boldsymbol{g}$. The above packing algorithms should be modified so that a new disk is inserted where its geometric and mechanical positions are both acceptable. If D_1 and D_2 are the two disks at the current active front and a new disk D_n is geometrically accepted, the mechanical stability of the new disk under the action of gravity can be easily established by simply checking the intersection of the gravitational force and the front segment. The new disk is stable if the intersection occurs within the segment, and unstable otherwise. This argument is justified by the fact that when the intersection point lies within the segment the contact forces between the new disk and the other two disks are of a compression nature, and therefore the new disk will be stable at the position. These situations are respectively illustrated in Figs. 5.9(a) ∼ (c), where the new disk is both geometrically and mechanically acceptable in (a), while it is only geometrically acceptable but not mechanically stable in (b) and (c) and therefore will be rejected. In the latter cases, a new active front should be chosen to continue the packing procedure until a case (a) situation is achieved for the disk. The difference is that in case (b), the subsequent front should be chosen as the next active segment while in case (c) the preceding front should be selected instead.

Note that the above outlined algorithm can be successfully implemented in the open form approach due to the presence of the domain boundary which acts as a container, provided that the disks are packed from the bottom. However, it is not the case for the closed form

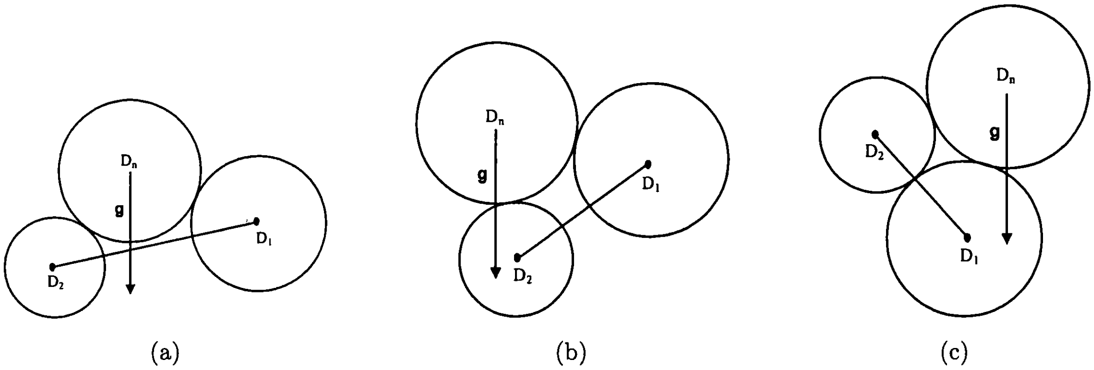


Figure 5.9: The new disk D_n is: (a) stable; (b) and (c) unstable under the action of gravity.

approach. Therefore the packing algorithm considering gravity is only applicable to the open form version.

5.2.4 Illustrations

Examples are provided to illustrate the performance of the proposed two versions of the advancing front algorithm. Disks to be generated are assumed to have radii that are evenly distributed in a range of $[R_{min}, R_{max}]$ and determined by utilising the evenly distributed random function provided in the various computer languages.

Fig. 5.10 demonstrates the final arrangement of disks generated in a unit square by the closed form advancing front algorithm for two different disk radius ranges: $[0.1, 0.1]$ and $[0.01, 0.1]$. The left column figures show all the disks generated while the right column figures show only the disks within the domain. Clearly, for equal size disks (Fig. 5.10(a)), the current algorithm reproduces the optimal hexagonal lattice packing. For a different radius distribution (Fig. 5.10(c)), a certain randomness of disk distribution is obviously present. It is also evident that some rather large gaps around the boundary are left after culling takes place (Figs. 5.10(b) and (d)).

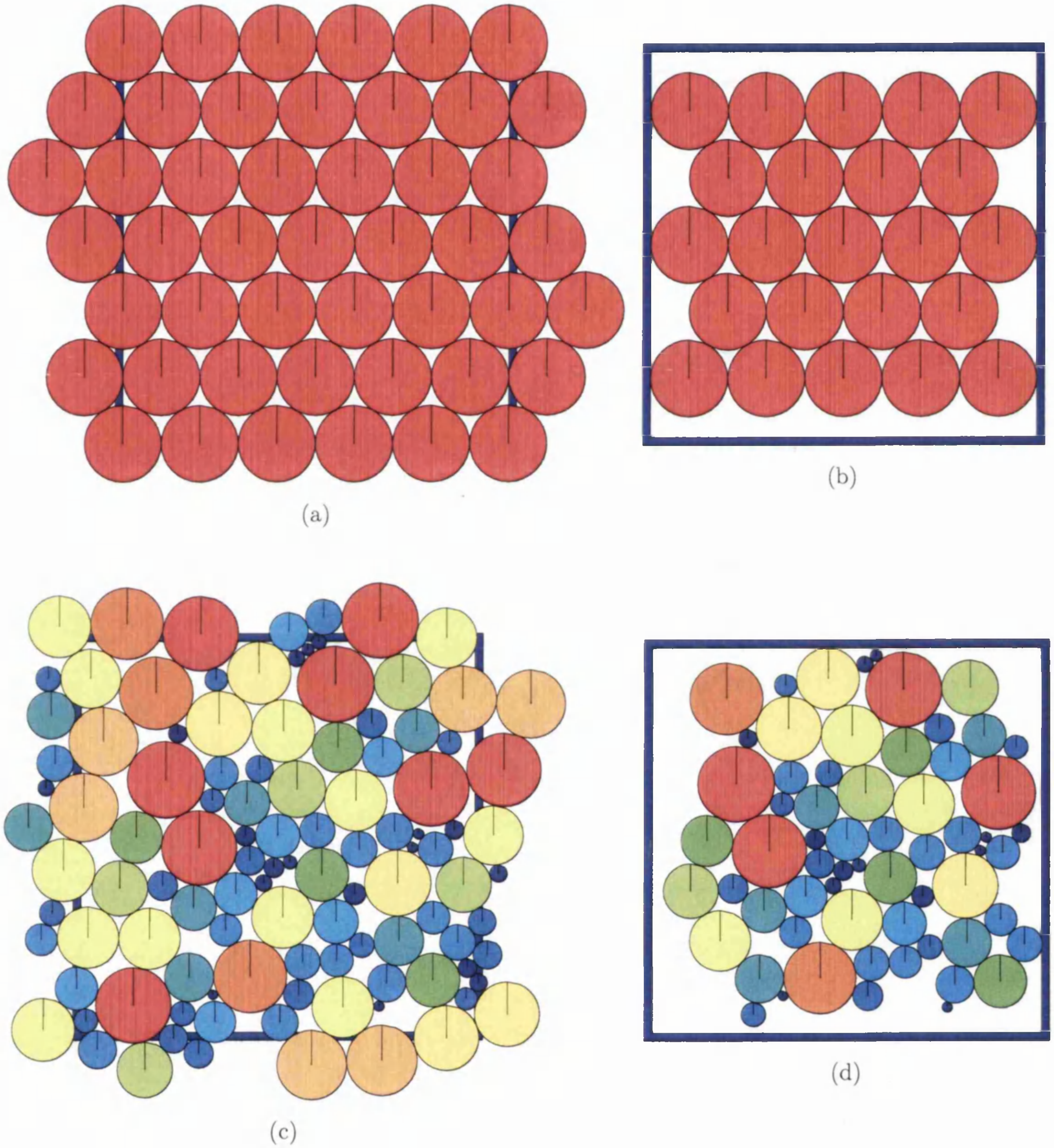
The final arrangement of disks generated by the open form advancing front algorithm for the same radius distribution ranges and the same domain are shown in Fig. 5.11. Note that because the boundary walls participate in the generation of disks at all stages, the optimal hexagon lattice packing is not produced for equal size disks (Fig. 5.11(a)), although it is still possible if additional rules are applied in the algorithm for this special case. Notably, the gaps around the boundary, except the top, have been reduced compared to the previous cases.

Figs. 5.12 and 5.13 illustrate the ability of the current development to handle different shapes of polygonal domains, either convex or concave case. Note that gravity is not taken into consideration in packing shown in Fig. 5.12, so some of the disks are in unstable positions. Packing with gravity are depicted in Figs. 5.13(a)$\sim$(d), in which the first disk is placed at the bottom left corner of the packing domain.

Further tests have also demonstrated the effectiveness of the algorithm developed. For the case where the number of front segments to be checked when generating a disk is limited to a maximum value of 5 for both forward and backward checks, it takes only 3.77 seconds for the open form version to generate 1,000,000 disks on a PC with a one 1GHz processor.

5.3 Packing of Convex Polygons

The above advancing front approach can also be employed to generate a random packing of convex polygons without essential changes. The modifications that needs to be implemented



(a)

(b)

(c)

(d)

Figure 5.10: Closed form advancing front - generated packings in a unit square for two different disk radius ranges: (a,b) [0.1, 0.1]; (c,d) [0.01, 0.1] before (left column) and after (right column) culling

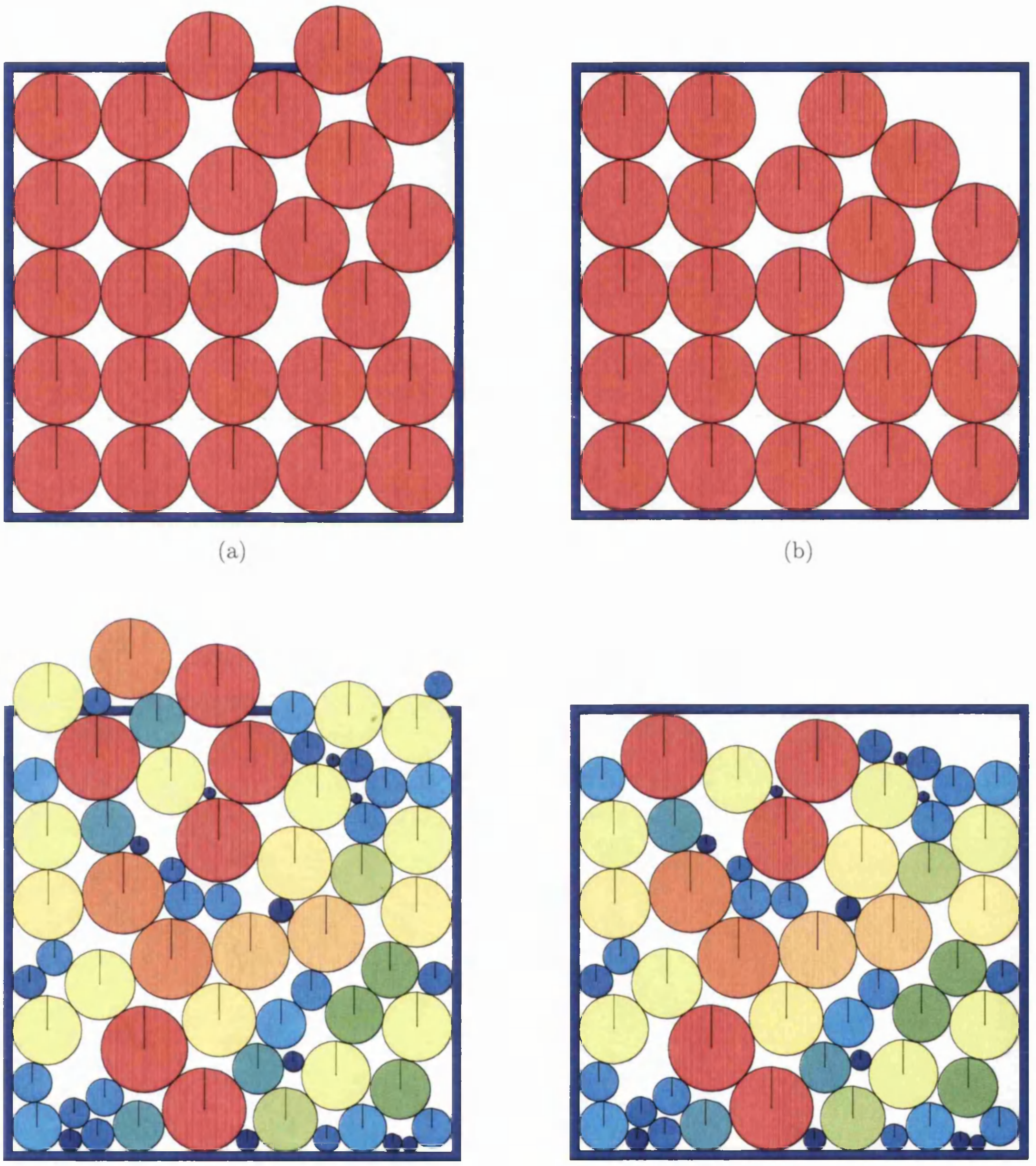


Figure 5.11: Open form advancing front - generated packings in a unit square for two different disk radius ranges: (a,b) [0.1, 0.1]; (c,d) [0.01, 0.1] before (left column) and after (right column) culling

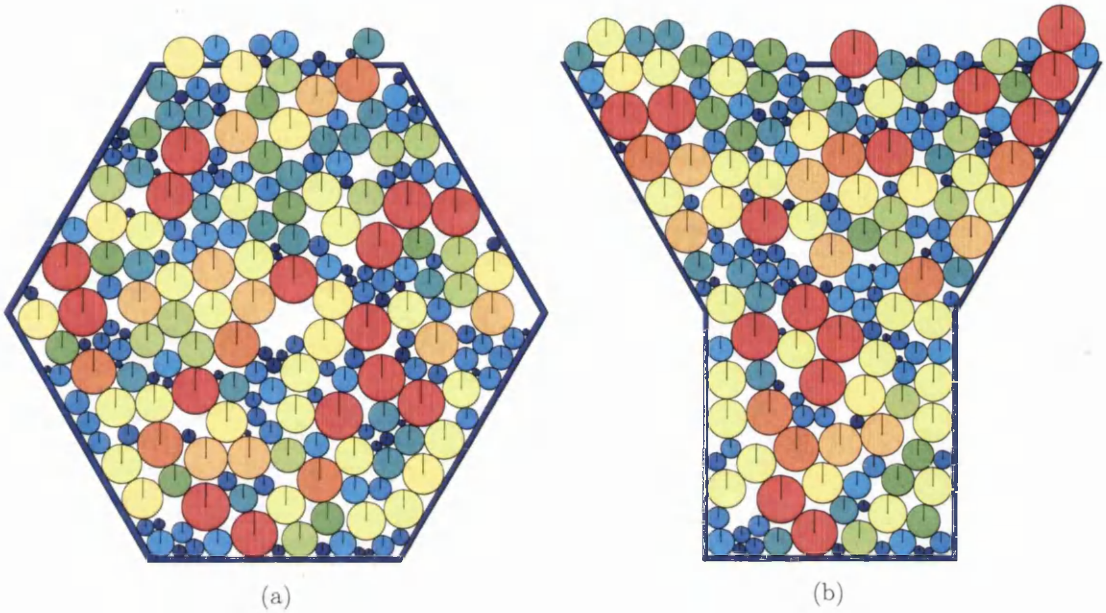


Figure 5.12: Open form advancing front - packings in different shapes of domains without the consideration of gravity.

include the overlap check of two polygons and the procedure to determine the position of a new polygon where it touches the other two polygons associated with the active front.

Suppose that the new polygon W has k vertices/edges and that its shape, size and orientation can be generated randomly or with any given patterns. When packing the polygon, only translational motion is allowed and no rotational re-arrangement is permitted.

Given two polygons, the overlap check can be performed by employing the modified linear algorithm described in Chapter 4.

If P and Q are the two polygons associated with the current active front, the position of a new polygon W can be determined by constructing the locus of a reference point in polygon W when it slides, by translation motion only, along the boundary of polygon P or Q. The locus, closely associated with the classical 'Minkowski sum' of the two polygons, is a convex polygon itself with the number of vertices equal to the sum of the vertices of polygons P (or Q) and W, and can be explicitly constructed in a linear time [14].

Fig. 5.14(a) shows the locus of the centroid of a triangle sliding along the boundary of a pentagon. When the loci of polygon W with polygons P and Q are all constructed, the two intersection points of the loci can be obtained, which correspond to the two possible positions at which polygon W can be inserted in touch with polygons P and Q. The correct location

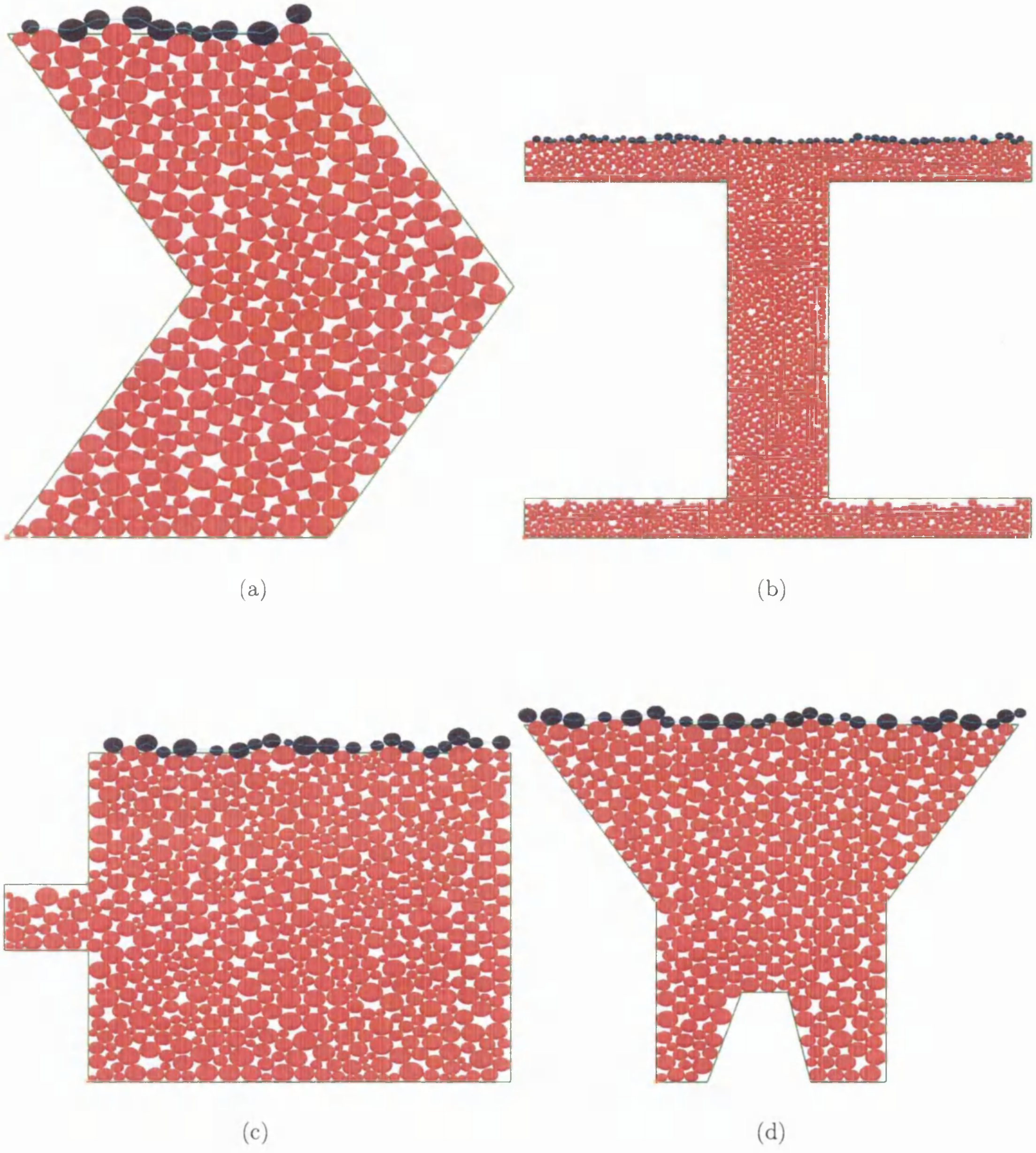


Figure 5.13: Open form advancing front - packings in different shapes of domains with the consideration of gravity.

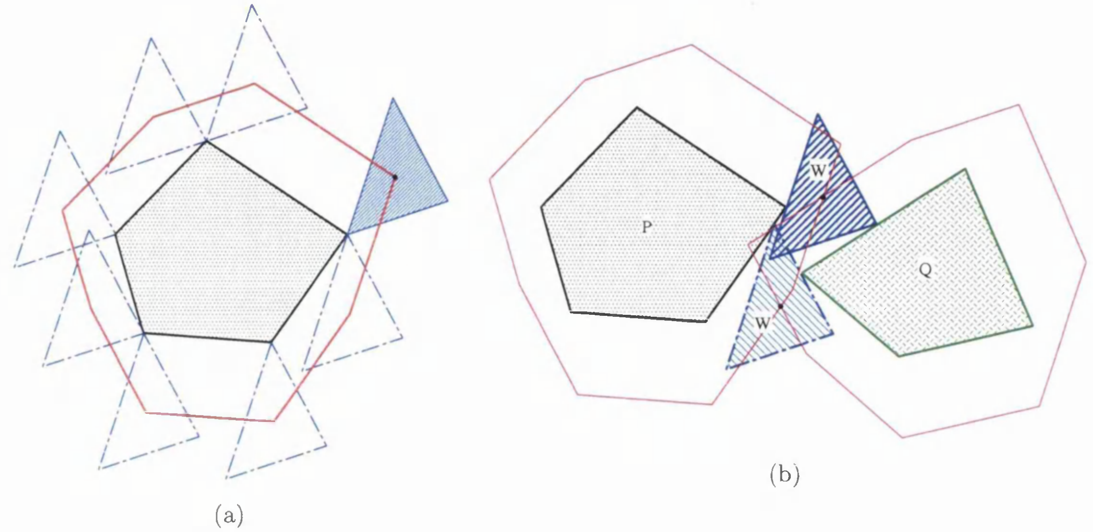


Figure 5.14: (a) Locus of the centroid of a triangle sliding along a pentagon; (b) Two possible packing positions where polygon W just touches polygons P and Q

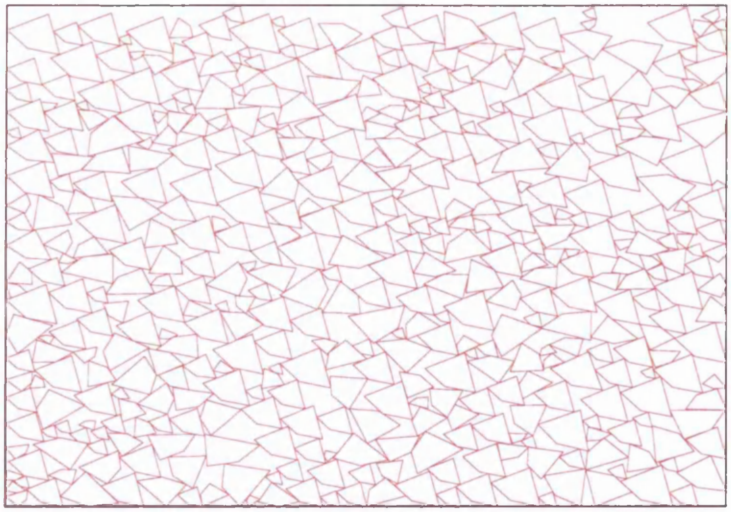


Figure 5.15: Random packing of polygons with open form advancing front

can be selected according to the default front advancing direction. The procedure outlined is illustrates in Fig. 5.14(b).

With the above modifications, the performance of the resulting polygonal packing algorithm is almost as efficient as that for disks. An example is illustrated in Fig. 5.15.

5.4 Packing of Ellipses

Similar to polygon packing, two special operations include the overlap check of two ellipses and the determination of the position of a new ellipse. However, the implementation is more complex than that of polygon packing due to the nonlinear nature.

Consider two ellipses P and Q with their major and minor axes respectively as (a, b) and (a_1, b_1). Without loss of generality, it is assumed that the centre of ellipse P is coincided with the origin of the global coordinate system and the major axis is aligned with the x direction, while the centre of ellipse Q is located at (x_c, y_c) and its major axis is rotated by an angle ϕ to the x direction. The general equations for ellipses P and Q are respectively

$$f(x,y) = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$$

$$g(x,y) = A(x-x_c)^2 + B(y-y_c)^2 + 2C(x-x_c)(y-y_c) - 1 = 0$$

where

$$A = \frac{\cos^2\phi}{a_1^2} + \frac{\sin^2\phi}{b_1^2}, \quad B = \frac{\sin^2\phi}{a_1^2} + \frac{\cos^2\phi}{b_1^2}, \quad C = \sin\phi\cos\phi\Big(\frac{1}{a_1^2} - \frac{1}{b_1^2}\Big)$$

The coordinates of an arbitrary point p_i on ellipse P can be alternatively represented in parametric form as

$$x_i = a\cos\theta, \qquad y_i = b\sin\theta, \qquad \theta \in [0, 2\pi]$$

The *'distance'* of point p_i to ellipse Q may be given by

$$d = g(x_i, y_i) = \bar{g}(\theta)$$

By finding an $\bar{\theta}$ which gives a minimum distance $d_{min} = \bar{g}(\bar{\theta})$, the overlap condition for ellipses P and Q is

$$d_{min} < 0$$

It can be shown that $\bar{\theta}$ satisfies

$$a(Ax_c + Cy_c)\sin\theta - b(By_c + Cx_c)\cos\theta + \frac{1}{2}(b^2B - a^2A)\sin 2\theta + abC\cos 2\theta = 0$$

and therefore can be solved analytically or iteratively. Note that the above equation is equivalent to that in [15] but derived in a different manner.

The centre position of a new ellipse to be inserted is determined by first constructing the locus of the centre sliding in a pure translational motion along the boundary of each ellipse on the active front, and then finding the intersection points of the two loci. Since the loci are not elliptical curves, the central position cannot be obtained analytically, but an iterative solution procedure must be employed.

Consider sliding ellipse Q along the boundary of ellipse P. The centre coordinates of ellipse Q can be expressed in terms of parameter $\theta \in [0, 2\pi]$ as

$$x_c = a\cos\theta + u\lambda, \qquad y_c = b\sin\theta + v\lambda,$$

in which

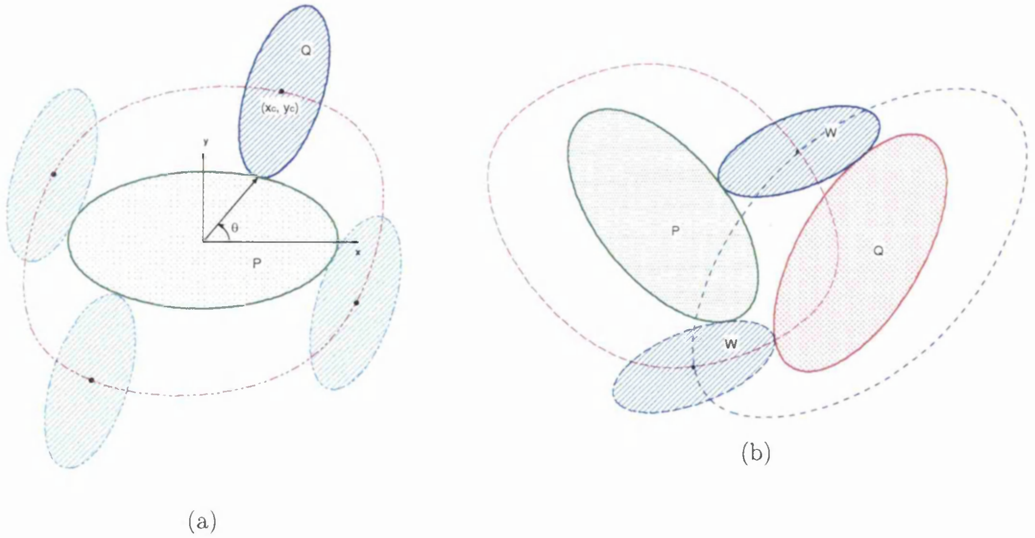
$$u = B\cos\theta - C\sin\theta, v = A\sin\theta - C\cos\theta,\ \lambda = 1/\sqrt{Au^2 + Bv^2 + 2Cuv}$$

Fig. 5.16(a) illustrates the locus of ellipse Q and its non-elliptical shape is clearly observed.

The Newton-Raphson iterative algorithm is employed to obtain the intersections of the two loci, as shown in Fig. 5.16(b), of a new ellipse W sliding along the boundaries of ellipses P and Q. These two intersection points give the centre position of ellipse W to be inserted.

In the implementation, ellipses and all the loci are hierarchically represented as polygons with different levels of approximation, which not only speeds up the overlap check of two ellipses, but provides a better initial guess for Newton-Raphson iterative procedure.

Fig. 5.17 illustrates a random packing of ellipses with the closed form advancing front approach.



(b)

(a)

Figure 5.16: (a) Locus of the center of ellipse Q sliding along ellipse P; (b) Two possible packing positions where ellipse W just touches ellipses P and Q

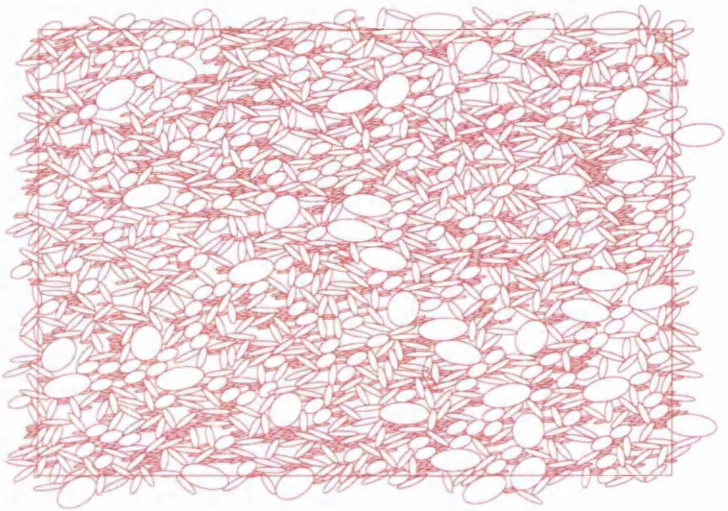


Figure 5.17: Random packing of ellipses with closed form advancing front

5.5 Concluding Remarks

An advancing front based algorithm is described for the generation of a random packing of disks, ellipses and polygons of various sizes within a geometric domain. Depending on whether the domain boundary is included in the initial front and how the front is formed, two different versions of the algorithm, termed the closed and open form respectively, are presented. The open form approach has an inherent linear complexity, while the closed form can achieve the same complexity under a relatively weak condition. The generated packing is not a globally optimal arrangement but achieves a locally highest density from the algorithmic point of view. The mechanical stability of the packing can be improved by considering the action of an external force, such as gravity, during the packing procedure. The performance of the algorithm has been illustrated via numerical examples.

Bibliography

[1] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer-Verlag, Berlin, Germany, 1988.

[2] J. D. Bernal. A geometrical approach to the structure of liquids. *Nature*, 183:141-147, 1959.

[3] G. D. Scott. Packing of spheres. *Nature*, 188:908-909, 1960.

[4] J. L. Finney. Fine structure in randomly packed, dense clusters of hard spheres. *Mater. Sci. Engng*, 23:199-205, 1976.

[5] D. C. Koskenmaki. A model for the structure of amorphous metals. *Mater. Sci. Engng*, 23:207-210, 1976.

[6] M. Stroeven. *Discrete Numerical Modelling of Composite Materials.* Ph.D. Thesis, Delft University of Technology, 1999.

[7] Y. T. Feng, K. Han and D. R. J. Owen. Filling domains with disks: an advancing front approach. *Int. J. Numer. Meth. Engng.*, 56: 699-703, 2003.

[8] Y. T. Feng, K. Han and D. R. J. Owen. An advancing front packing of polygons, ellipses and spheres. Proceedings of 3rd Int. Conf. on Discrete Element Methods, B. K. Cook & R. P.Jensen (ed.): pp93-98, 2002.

[9] M.J. Vold. The sediment volume in dilute suspension of spherical particles. *J. Phys. Chem.*, 64:1616–1671, 1960.

[10] W. M. Visscher and M. Bolsterli. Random packing of equal and unequal spheres in two and three dimensions. *Nature*, 239:504-507, 1972.

[11] L. Oger, A. Gervois, J.-P. Troadec and N. Rivier. Voronoi tesselation of packings of shperes. *Phil. Mag. B*, 74:177-197, 1996.

[12] J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comp. Phys.*, 72:449–466, 1987.

[13] F. P. Preparata and M. I. Shamos. *Computational Geometry, An Introduction.* Springer-Verlag, Berlin, Germany, 1985.

[14] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publisher, Boston, USA, 1991.

[15] J. M. Ting. A robust algorithm for ellipse-based discrete element modelling of granular materials. *Computers and Geotechnics*, 13(3):175-186, 1992.

Chapter 6

Sphere Packing with a Geometric Based Compression Algorithm

6.1 Introduction

In the previous chapter, a novel algorithm based on the idea of advancing front techniques for finite element mesh generations is introduced for the random packing of 2D disks, polygons and ellipses. As only the fronts, each comprising two disks/polygons/ellipses, need to be maintained, the approach has proved to be very efficient with a linear complexity.

In principle this approach can also be extended to sphere packing. However, both the algorithm and implementation are much more complex. At the algorithmic level, the fronts consist of triangular facets, each obtained by joining the centers of three spheres. As the facet can't be fully covered by the associated spheres, some spheres may penetrate through the facets without necessarily overlapping the existing spheres. These difficulties significantly increase the algorithmic complexity. At the implementation level, a far larger number of different circumstances need to be dealt with properly when different shaped geometric domains are considered.

It should be noted that in many applications, the packing configurations are produced under various physical forces, especially the action of gravity. Consequently, a compressive packing direction is introduced to represent the realistic situation as closely as possible.

A new approach, termed the *compression algorithm*, is proposed in this work to generate a random packing of spheres of various sizes within a given geometric domain, which is motivated by the idea of compression complemented by an efficient physical process to increase packing density - shaking. This algorithm is characterised by its simplicity, high efficiency and applicability not only to the random packing of disks and spheres, but to other shaped particles.

6.2 Algorithmic Description

Consider the problem of randomly filling a domain with spheres of different sizes. The sphere radius R is randomly generated by a prescribed distributional function, and the domain Ω can be any arbitrary polyhedron or simple shape, such as a cube or cylinder. The compression direction $\bar{v}_g$ is specified.

Starting with an initial packing configuration, the algorithm proposed is a two-step procedure: (1) compress the initial packing; and (2) refill the space remaining, and if successful, compress the spheres by using the techniques in (1). Repeat the procedure until the domain is fully filled.

There are different methods to create an initial packing within a geometric domain. The spheres can be randomly distributed, as shown in Fig. 6.3(a), or regularly positioned. All the spheres are checked against each other so that no overlap exists.

It is highlighted that no real force is involved in the packing procedure, therefore it is a geometric based approach.

6.2.1 Compression of a given packing

The basic idea of compressing a given packing can be stated as: if a sphere's immediate neighbours are known, it can be moved along the given compression direction to a new position in touch with the first neighbouring sphere(s) by assuming that the neighbours are temporarily static. In one iteration after all spheres have been re-positioned, a more tightened packing should be achieved. Such iterations are repeated until no further compression is possible. Apparently, determining the dynamically changing neighbours of a sphere, which is accomplished by a search algorithm, is the key to the success of the algorithm.

There exist two slightly different options in searching for the neighbourhood information of a sphere. In the first option, the neighbour list of a sphere is built, then followed immediately by repositioning before proceeding to the next sphere. In the second option, the neighbour lists of all the spheres are obtained altogether at the beginning of the iteration, and the spheres are then compressed sequentially. To ensure that the neighbour information obtained in the beginning remains valid for the whole iteration, a bounding box is assigned to each sphere, and the movement of any sphere must be confined within the bounding box. Note that such a requirement is also necessary for the first option to essentially define the 'close' neighbours of a sphere and to compute the maximum allowable moving distance. Consequently, the determination of the neighbour information of spheres is equivalent to the contact detection among spheres represented by axis aligned bounding boxes.

Since contact detection comprises a major proportion of the total CPU time involved in the complete packing procedure, an efficient search algorithm is expected. The dynamic cell based search algorithm, or 'D-cell', described in Chapter 2 is naturally the choice. As this approach is suitable to handle the contact among all the objects than the contact of one object with others at a time, the second option is therefore chosen.

Compression

Prior to any iteration, a bounding box with buffer zone is assigned to each sphere. Issues relevant to the bounding boxes will be discussed later.

Once the neighbour list is built, the next step is to evaluate the maximum moving distance of each sphere along the compression direction. Assume in Fig. 6.1 that $\bar{v}_g = (t_x, t_y)^T$ is the given compression direction with its normal $\bar{n}_g = (n_x, n_y)^T$; C is the sphere to be compressed, and $T_i(i = 1, ..., n_t$, where n_t is the number of neighbours of sphere C) is one of C's neighbour spheres. For the convenience of illustration, a 2D problem is taken as an example and the

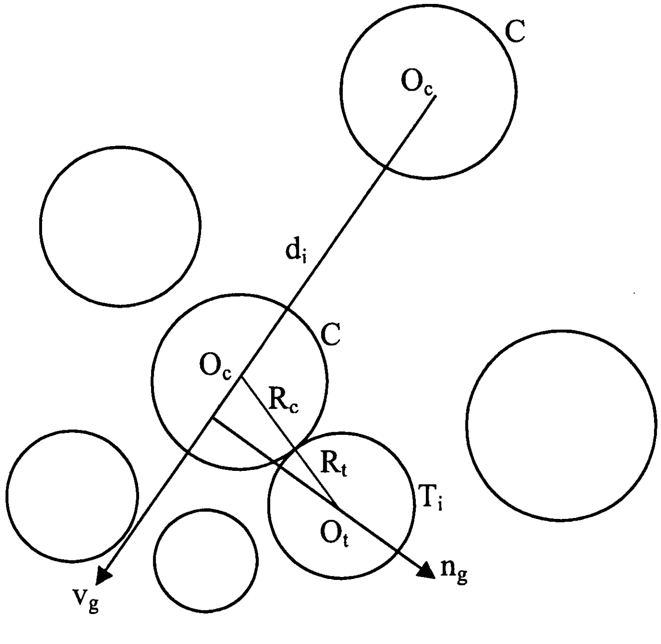


Figure 6.1: Maximum moving distance d_c for sphere C

maximum moving distance can be calculated as:

$$d_i = t - \sqrt{(R_c + R_t)^2 - s^2} \tag{6.1}$$

with

$$t = n_y x_{ct} - n_x y_{ct}, \quad s = t_y x_{ct} - t_x y_{ct}$$

$$x_{ct} = x_c - x_t, \quad y_{ct} = y_c - y_t$$

where R_c and R_t are respectively the radius of sphere C and T_i; and (x_c, y_c), (x_t, y_t) their centre coordinates. By looping over all the neighbouring spheres, the moving distance d_c of sphere C should be the smallest value of d_i, i.e.

$$d_c = \min(d_i), \quad \text{i=1,...,}n_t \tag{6.2}$$

Nevertheless, as mentioned earlier, the movement of the sphere must be confined within the bounding box, or in other words, d_c must not be greater than the size of buffer zone b_u:

$$d_c = b_u \quad if \quad d_c > b_u \tag{6.3}$$

Furthermore, the computed distance should be checked with the boundary of the packing domain. If overlap occurs, the distance should be recomputed based on the shape of the boundary so that the sphere will just touch the boundary. Clearly any shape of boundary can be accommodated in the above procedure. The new position of the sphere, X_c, is then updated as

$$X_c \leftarrow X_c + d_c \bar{v}_g \tag{6.4}$$

Within one iteration, each sphere will be moved a distance between $[0, b_u]$ along the compression direction. The procedure is repeated until no further movement is possible, or a certain convergence criterion is met. Here a 'potential energy' based convergency check is employed, in which the 'height' h_e and the 'potential energy' p_e of any sphere are respectively defined as

$$h_e = \bar{v}_g \cdot (X_c - X_0) \tag{6.5}$$

$$p_e = h_e R^3 \tag{6.6}$$

where X_c are the centre coordinates of the sphere, and X_0 the coordinates of a reference point. The total 'potential energy' P_e of the system involving all the spheres can thus be calculated as

$$P_e = \sum p_e \tag{6.7}$$

The procedure is terminated if the difference of potential energy between two consecutive iterations is less than a given tolerance τ:

$$|P_e^c / P_e^p - 1| < \tau \tag{6.8}$$

where P_e^c and P_e^p are respectively the total 'potential energy' of the spheres at the current and previous iteration steps.

The above algorithm is defined as *global compression*, and is outlined in Box 6.1.

Shaking

If spheres are compressed in the given compression direction only, a highly chained configuration oriented to the direction is rapidly developed. As a result, only a limited compression can be achieved. Such a phenomenon is due to the inability of the above compression procedure to move the spheres elsewhere other than in the compression direction only, and thus is equivalent to a situation where particles are highly sticky. A simple scheme to overcome this difficulty is to introduce for each sphere a randomly generated local compression direction $\bar{v}_l$, which is different from the global direction $\bar{v}_g$, having a component normal to $\bar{v}_g$ as defined by

$$\bar{v}_l = \alpha\bar{v}_g + \beta\bar{v}_n \qquad \text{with } \bar{v}_g \cdot \bar{v}_n = 0, \qquad \|\bar{v}_n\| = 1 \tag{6.9}$$

where the normal direction $\bar{v}_n$ is randomly generated, and $\alpha \in [0, 1]$ and $\beta \in [0, \bar{\beta}]$ are also two uniformly distributed random variables in the intervals. The value of $\bar{\beta}$ defines the angle of the cone where any local compression direction lies (Fig. 6.2), i.e. the maximum proportion of the normal component in the local compression direction. With the randomly generated local compression direction, the global compression algorithm is similarly employed to undertake

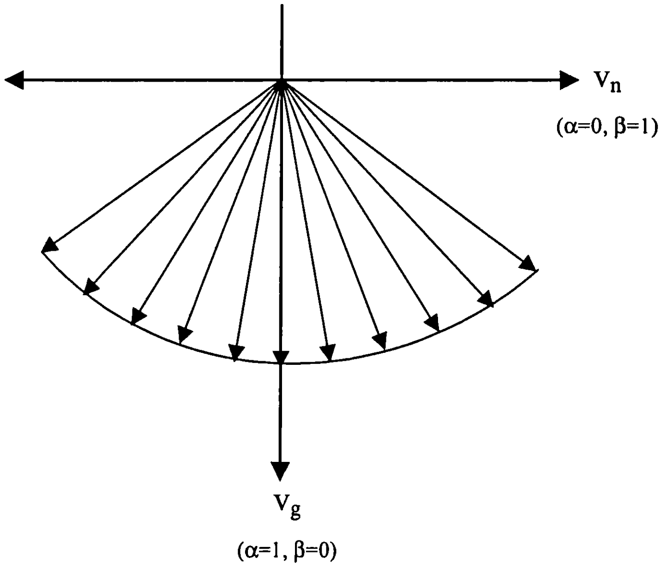


Figure 6.2: Random compression directions

Box 6.1: Global Compression Algorithm

```
• Given an initial packing, compression direction $\bar{v}_g$, and tolerance $\tau$, calculate the initial total potential energy $P_e^c$;
• DO {
    – Set $P_e^p = P_e^c$;
    – Assign a bounding box with buffer zone for each sphere;
    – Perform contact detection to obtain neighbour list for each sphere;
    – Loop over spheres to compute allowable moving distance $d_i(i = 1, ..., N)$
      {
        * Loop over neighbour list (j=1,...,$n_t$)
          {
            · Evaluate the maximum moving distance $d_j$ along the compression direction:
          }
        * $d_i = min(d_j)$
          $d_i = b_u \quad if \quad d_i > b_u$
        * Check with domain boundary;
        * Update position of the sphere:
          $X_i \leftarrow X_i + d_i\bar{v}_g$
      }
    – Calculate current potential energy $P_e^c$;
} while($|P_e^c/P_e^p - 1| > \tau$)
```

compression by simply replacing the global compression direction $\bar{v}_g$ with the local direction $\bar{v}_l$. This is termed *random compression.*

With the strategy stated above, the packing density is significantly increased. It is observed, however, that local arches are often formed which prevent further increase of packing density. A well known physical process, *shaking*, is conceptually incorporated into the packing procedure to break the arches. To do so, it is only necessary to perform several additional

Box 6.2: Compression Algorithm

```
• Given an initial packing, compression direction $\bar{v}_g$, number of shaking
  $N_s$ and tolerance $\tau$, calculate the initial total potential energy $P_e^c$;
• Perform global compression (refer to the algorithm in Box 6.1);
• Calculate current potential energy $P_e^c$;
• DO {
    – Set $P_e^p = P_e^c$;
    – Shaking for $N_s$ times;
    – Random compression;
    – Calculate current potential energy $P_e^c$;
  } while($|P_e^c/P_e^p - 1| > \tau$)
```

iterations with compression along randomly generated local directions $\bar{v}_h$, which are perpendicular to the global compression direction. The remainder of the compression procedure is exactly the same as that in Box 6.1. As this compression does not change the total 'potential energy' of the system, the convergence check is not applicable, but the number of iterations is specified instead. The procedure is referred to as *shaking.*

Consequently, the compression algorithm is the combination of the three different compression procedures: global compression, random compression and shaking, which is summarised in Box 6.2.

6.2.2 Refilling

After an initial packing has been compressed, further steps are required, if necessary, to fill the void created from compression until the whole domain is full. There are different methods to achieve this goal. In this work, a strategy called *sphere insertion* is adopted. First of all, the domain lid L is defined, which is usually one of the domain boundary surfaces opposite to the compression direction. The idea is to generate a new sphere on the lid, and then drop it along the compression direction to touch the nearest existing sphere(s). Such an insertion procedure can also be accomplished by applying the above compression algorithm to this single sphere. The generation and dropping procedure is repeated till no space is left to allow

Box 6.3: Creating New Batch of Spheres Algorithm

```
• Given the maximum number of rejections $M_r$; Set number of rejections $n_r = 0$
• DO {
    – Generate a new sphere on the domain lid $L$. If it overlaps with any existing sphere(s), reject it and set $n_r = n_r + 1$;
    – Drop the sphere to touch the nearest sphere(s) along the compression direction.
  } while($n_r < M_r$)
```

Box 6.4: Filling Domain Algorithm

```
• DO {
    – Compression loop (refer to Compression Algorithm in Box 6.2);
  } while(Creating New Batch of Spheres: algorithm in Box 6.3)
```

the insertion of new spheres. This process creates a new batch of spheres, which can be compressed by using the compression algorithm in Box 6.2.

The complete filling domain algorithm is summarised in Boxes 6.3 and 6.4.

6.2.3 Remarks on the algorithm

The buffer zone used to extend the bounding box of each sphere plays an important role in the algorithm. The size of buffer zone has two competing effects: (1) it directly controls the length of the neighbour list in contact detection. The larger the buffer zone is, the longer the neighbour list will be, or the more expensive the contact detection; (2) it controls the maximum moving distances of spheres, particularly at the early stage of compression. The larger the buffer zone is, the larger the moving distance will be, i.e. fewer iterations are needed. Therefore, the choice of the buffer zone size should balance the contact detection cost and the total number of iterations.

The bounding boxes are usually extended with the same buffer zone size along each coordinate direction. However, since the spheres are moved in certain directions only at different compression stages, it is prudent to extend the bounding boxes along these directions. For instance, a buffer zone is added only in the global compression direction in the global compression phase; and in the directions perpendicular to the global compression direction during shaking. Further improvement can be achieved through adaptive control of the buffer zone size for each sphere in each packing phase. As the moving distances of spheres become less and less with the increase of packing density, the buffer zone size can also be reduced accordingly to reduce the cost associated with the contact detection.

At the later stage of compression, a large proportion of spheres may hardly be moved. This 'mobility' of spheres can be monitored easily. Clearly, any sphere that was not moved at the previous iteration can only be moved if at least one of its neighbours has been repositioned. By applying this strategy, local movement checks can be avoided for those immobile spheres.

It is also found beneficial to move spheres according to their heights in the global compression direction, i.e. to process spheres starting from the bottom and then upward. This can be explained by the fact that for a sphere to be processed, it is more likely to be moved with a larger distance if the lower level spheres in the neighbour list are moved first.

6.3 Illustrations

The validation of the proposed compression algorithm is tested via numerical examples.

Figs. 6.3(a)$\sim$(h) illustrate different phases of filling a cylindrial domain with spheres. The global compression direction is downwards from top to bottom of the cylinder, and the sphere radius is chosen to be uniformly distributed within $[0.01, 0.02]mm$. The initially packed 3071 spheres, as shown in Fig. 6.3(a), are randomly positioned in the cylinder, which is compressed by implementing the algorithm in Box 6.2, leading to a tightened packing (Fig. 6.3(b)). To fill the void space created following the compression, a new batch of spheres (7261) are generated, with green colour in Fig. 6.3(c), and then compressed (Fig. 6.3(d)). The procedure is repeated till no space is left to allow the insertion of more spheres. The final configuration is presented in Fig. 6.3(h), with a total number of 26787 spheres and 52.89% packing density. It takes 181 seconds on a 1.4GHz PC to complete the packing.

Figs. 6.4(a) and (b) also show the final configurations of filling two polyhedral domains with a total number of 51036 and 10280 spheres respectively.

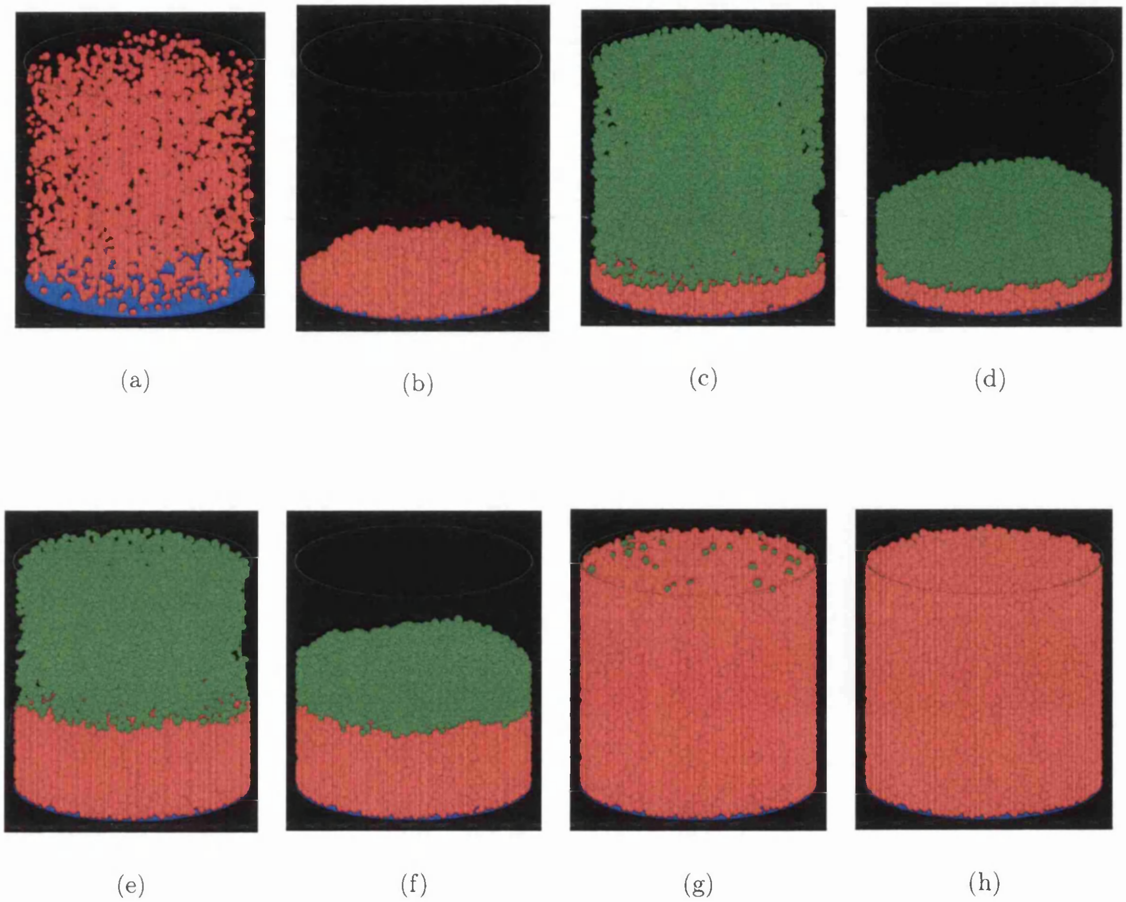
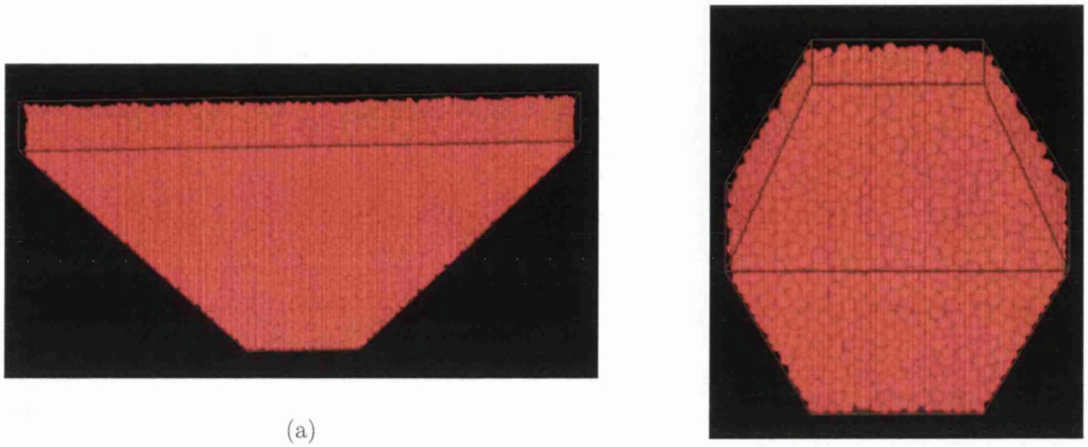


Figure 6.3: Different phases of filling a cylindrical domain with spheres



(a)

(b)

Figure 6.4: Final configurations of filling polyhedral domains. Sphere radii uniformly distributed within: (a) [0.02, 0.05]mm; (b) [0.05, 0.01]mm

6.4 Concluding Remarks

A geometric based approach, *compression algorithm*, has been presented in this work for the random filling of geometric domains with spheres of various sizes. It is motivated by the idea of compression complemented by an efficient physical process to increase packing density - shaking. By specifying a global compression direction, the compression procedure is the combination of three different strategies: global compression, random compression and shaking. As contact detection comprises a major proportion of the computational time for a complete packing, the dynamic cell based search algorithm, or *D-cell*, has been employed to achieve an efficient packing performance. The compression algorithm features both simplicity and high efficiency, which has been demonstrated through test examples. In addition, the proposed algorithm can be applied not only to the random packing of disks and spheres, but to other shaped particles.

Bibliography

[1] J. D. Bernal. A geometrical approach to the structure of liquids. *Nature*, 183:141-147, 1959.

[2] G. D. Scott. Packing of spheres. *Nature*, 188:908-909, 1960.

[3] W. M. Visscher and M. Bolsterli. Random packing of equal and unequal spheres in two and three dimensions. *Nature*, 239:504-507, 1972.

[4] J. L. Finney. Fine structure in randomly packed, dense clusters of hard spheres. *Mater. Sci. Engng*, 23:199-205, 1976.

[5] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer-Verlag, Berlin, Germany, 1988.

[6] Y. T. Feng, K. Han and D. R. J. Owen. An Advancing Front Packing of Polygons, Ellipses and Spheres. *3rd Int. Conf. on Discrete Element Methods,* Santa Fe, New Mexico, USA - Sept 23-25, 2002.

Chapter 7

Combined Finite/Discrete Element Simulations of Shot Peening Processes

7.1 The Shot Peening Process

Shot peening is a cold working process in which the surface of a mechanical part is bombarded with small media called shot, as illustrated in Fig. 7.1. The impact of each shot produces an indentation or dimple on the workpiece surface due to the plastic flow of material (Fig. 7.2(a)). This indentation has a larger surface area than the original surface and, cumulatively, the indentations attempt to create a surface expansion. The elastically deformed sub-surface layers act to resist this surface expansion, inducing a compressive stress at the surface, which is balanced by a tensile stress of lower magnitude in the core of the material (Fig. 7.2(b)).

Benefits obtained by shot peening are the result of the effect of the compressive stress and the cold working induced. It is well known that cracks will not initiate or propagate in a compressively stressed zone. Since nearly all fatigue and stress corrosion failures originate at the surface of a part, compressive stresses introduced by shot peening provide considerable increase in part life. The maximum compressive residual stress produced at or under the surface of a part by shot peening is at least as great as half the yield strength of the material being peened [1]. Many materials will also increase in surface hardness due to the cold working effect of shot peening.

Shot peening techniques were initially developed for use within the aeronautics industry principally for improving the fatigue life of parts. It is finding increasing applicability in other industrial sectors. In the automotive industry the technique is employed to improve

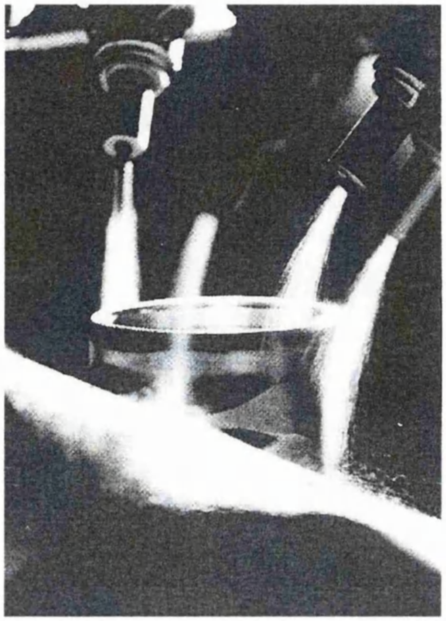


Figure 7.1: (reproduced from [1]): Shot peening

the fatigue life of parts such as connecting rods, crankshaft, gears and springs, whilst in shipbuilding it is used to repair cavitation damage in propellers and in the post-treatment of welds in ship hulls. Within the general engineering sector parts such as storage tanks, gas and steam turbines, piping in chemical, petroleum and nuclear plant, etc. are frequently shot peened.

Shot peening is also used in the shape forming of sheet components to produce a desired

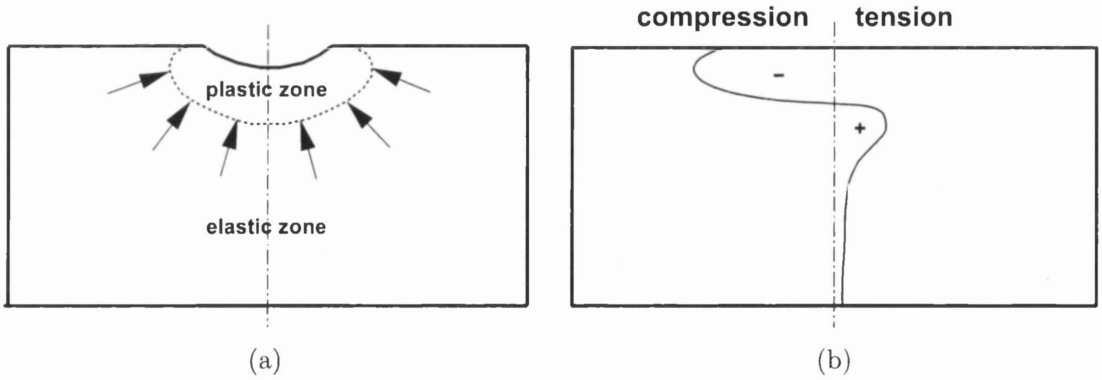


Figure 7.2: Shot Peening: (a) Indentation; (b) Typical residual stress profile

curvature and shape or to correct the shape of a component, which will be described in the following chapter.

Some key parameters of the shot peening process are outlined below. See [1] and [2] for more details.

7.1.1 Peening media

Media typically used for shot peening are small spheres of cast steel, conditioned cut wire, ceramic and glass materials. Cast steel shot is by far the most popular and has the advantages of low initial cost and reasonably good durability. Sizes range from 0.18mm to 2.36mm and two hardness levels are used: regular shot at 42-52 HRC and hard shot at 55-65 HRC.

Cut wire shot is produced by chopping steel wire into cylinders where the length equals the diameter. The initial cost of cut wire shot is very expensive when compared to cast steel shot but the overall cost falls rapidly in use because it is much more durable. Cut wire shot has two other interesting advantages. (1) It rarely breaks into sharp half spheres and (2) all the conditioned spheres are very close to the normal size and, therefore, have essentially the same mass. Cut wire shot is available in both carbon and stainless steel (for use where iron contamination is undesirable).

Glass beads are also frequently used where iron deposits on the part surface would be detrimental. In addition, they can be used to provide a desired surface finish or effect. Glass beads are available in sizes down to 0.05mm and are therefore useful for peening into sharp radii or threads and when very low intensities are required on delicate parts.

Ceramic beads are made from zirconium oxide and are very hard and totally inert. They are reasonably durable but expensive. They find application in situations where no foreign metal can be tolerated, such as in chemical processing equipment and in rework of aging aircraft structures.

Peening media must be uniform in size and essentially spherical in shape with no sharp edges or broken particles.

It has been found that the hardness of the shot will influence the magnitude and depth of compressive stress, therefore the peening media should always be at least as hard as or harder than the parts being peened, unless surface finish is a critical factor.

Shot size has a difference effect on the depth of the compressive layer on an aluminum alloys than it does on other materials. An aluminum part, peened to a given Almen intensity, will have a deeper layer of compressive stress when peened with larger shot than when peened with smaller shot at the same intensity. In steel and titanium alloys, for instance, the depth of compression remains fairly constant with intensity, regardless of shot size.

7.1.2 The Almen system for intensity control

The energy of shot stream is a function of the media size, material, hardness, velocity and impingement angle. In order to specify, measure and calibrate peening energy, J.O. Almen of General Motors Research Laboratories developed a method utilising SAE1070 spring steel specimens which he called Almen strips. In his method, an unpeened Almen strip is fastened to a steel block and exposed to a stream of peening shot for a given period of time. Upon removal from the block, the residual compressive stress and surface plastic deformation produced by the peening impacts will have caused the Almen strip to curve, convex on the peened surface. The height of this curvature when measured in a standard Almen gauge is called arc height. Fig. 7.3 illustrates the concept of the Almen system.

There are three standard Almen strips currently in use: "A" strip 0.051 inch thick, "C" strip 0.094 inch thick and "N" strip 0.031 inch thick. The approximate relationship between the A, N and C strips is: 3N=A=0.3C. The usable range of curvature on the Almen strips is 0.004 to 0.024 inch. Intensity designations should include both the arc height and the type of Almen strip used: e.g. 10A intensity means 0.010 inch (0.25cm) arc height on the A strip.

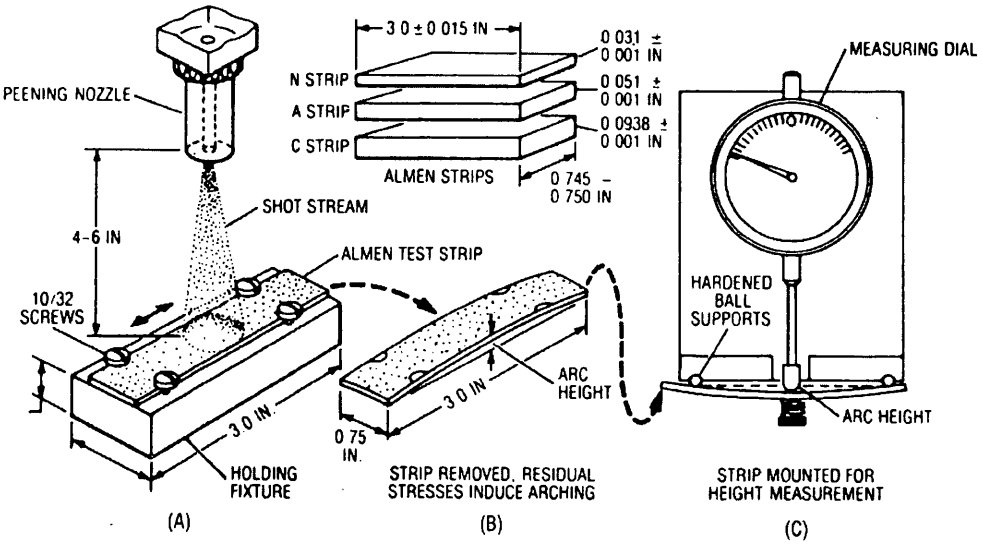


Figure 7.3: (reproduced from [1]): The Almen Strip system

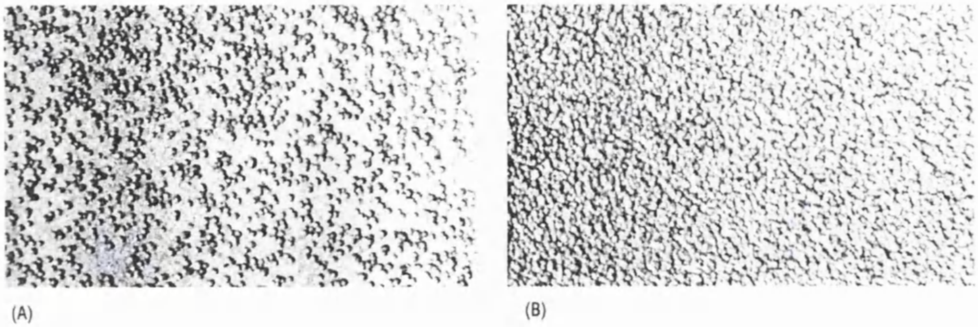


Figure 7.4: (reproduced from [1]): Shot peening coverage: (a) partial; (b) full

7.1.3 Coverage

Coverage is defined as the extent (in percent) of uniform and complete dimpling or obliteration of the original surface of the part. Fig. 7.4 shows photographs of partial and full coverage on aluminum strips. 100% coverage is reached when the original surface of the material is obliterated entirely by overlapping peening dimples. Coverage (in percent) above 100% is simply multiples of the exposure time necessary to achieve 100% coverage. For example, 150% coverage would be achieved by exposing the part 1.5 times the time required to achieve 100% coverage.

It is essential when shot peening for fatigue, stress corrosion cracking and most other applications that all of the critical areas be fully peened (100% coverage).

7.1.4 Saturation

An Almen arc height is not properly termed intensity unless saturation is achieved. In order to measure Almen strip saturation, an intensity curve must be developed. When a series of Almen strips are peened with a fixed machine setting for different exposure times, a saturation curve similar to the curve shown in Fig. 7.5 can be developed. Saturation is defined as the earliest point on the curve where doubling the exposure time produces no more than a ten percent increase in arc height.

It should be understood that saturation of the Almen strip and coverage of the part are not the same, nor will they necessarily occur at the same exposure time on the parts being peened. Saturation is the calibration of the energy of the shot stream. Coverage refers to the population of peening dimples on a surface as verified by coverage inspection techniques.

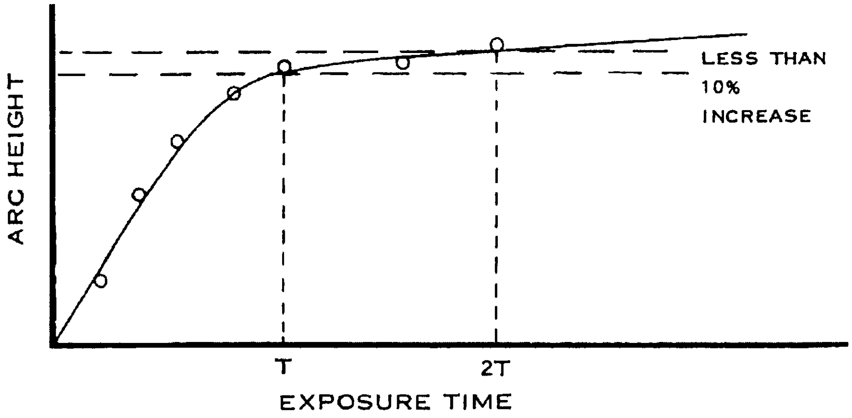


Figure 7.5: (reproduced from [1]): Saturation curve

7.2 Review of Modelling of the Shot Peening Process

To date, shot peening technology is little understood and commonly referred to as a '*black art*' rather than a science. Although a certain amount of research has been carried out experimentally or empirically, little was done theoretically and/or numerically. This situation is not surprising, however, due to the complex manner in which target materials respond to the multi-impact of shot.

The earlier analytic works, including those of Al-Hassani [3], Al-Obaid [4] and Hills [5] *et al.*, among others, attempt to establish theoretical expressions for the relation between the residual stress distribution and the operation parameters. The complex nature of the shot peening process, however, makes this approach unsuitable for dealing with practical industrial situations in which, for instance, different combinations of operation parameters, complex geometry, various material properties and 3D situations, are commonly encountered.

Over the past decade, the finite element method (FEM) has been employed in the simulation of shot peening operations. Some recent applications are reported in [6]. This research focuses on single shot representation by 3D FEM, from which an equivalent initial surface pressure force distribution is predicted and correlation with experimental tests made for simple component geometries. This approach is, however, an *indirect* and *inverse* simulation in nature. Its limitations makes it difficult to apply to more realistic shot peening processes.

Besides the work of Al-Obaid [7, 8], the direct simulation of single shot impacting on surface by employing FEM is also undertaken in [9]. An elasto-plastic or elasto-viscoplastic material description is employed within the nonlinear FE formulation for a transient dynamic response of the system. The main feature of the scheme is that shot is represented by a *deformable*

FE discretisation.

A numerical scheme employed for the simulations of practical industrial shot peening should be able to cope with medium velocity shot impact (up to 150m/s), inertial effects, dynamic loading and complicated frictional contact interaction; amongst other phenomena involved. First of all, it can effectively model the effect of a large number of shot. It is impractical to use a deformable FE representation of the shot and the use of DEM provides a clear advantage. The target material is modelled by FEM. With the combined FE/DE strategy, considerable computational benefits can be gained, particularly in large scale simulations.

Since shot peening is a highly dynamic event, *explicit solution techniques* can well capture the dynamic characteristics and provide detailed time history of the stresses and the evolution of plastic strain fields during the impact, as well as the residual stress profile in the target material. The explicit approach also takes advantage of a much simplified handling of the frictional contact between the shot and the impact surface offered by appropriate interaction laws.

The first combined FE/DE approach for the direct simulation of a shot peening process has been conducted by Petrinić [11]. This preliminary study has shown very promising results regarding the applicability of the approach to the simulation of practical shot peening processes. Nevertheless, many issues in terms of theory and implementation remain open.

The first issue related to the successful development of a combined FEM/DEM for the simulation of shot peening processes lies primarily in the appropriate modelling of contact between shot and workpiece. In the present work, the applicability of a selected set of commonly used interaction laws presented in Chapter 3 will be extensively examined in the context of the coupled FEM/DEM with a special attention given to the proper selection of parameter values involved.

7.3 Combined FEM/DEM: A General Description

In the combined FE/DE approach, the FEM is applied to determine the behaviour of the workpiece whose material properties are described by a particular constitutive law; elasto-plastic or elasto-viscoplastic constitutive laws are most commonly used for that purpose. The shot, on the other hand, is modelled by the DE technique as a circular disk (for 2D analysis) or sphere (for 3D analysis) with a radius R.

Generally speaking, both workpiece and shot are elastically and plastically deformable. However, given the fact that the plastic deformation of the shot is of secondary importance and that the surface hardness of the shot is generally greater than that of the workpiece, the shot can be considered to be rigid but a certain small overlap between shot and workpiece is allowed, i.e. a *soft disk model* described in Chapter 3 is adopted. The impact of shot on the workpiece is simulated along their boundaries by appropriate penalty based interaction laws.

By applying FE discretisation to the workpiece and based on an appropriate variational principle, the following general nonlinear dynamic system of equations for the workpiece, with proper imposed boundary and initial conditions, can be established

$$\boldsymbol{M}_w\ddot{\boldsymbol{u}}_w + \boldsymbol{C}_w\dot{\boldsymbol{u}}_w + \boldsymbol{F}^{in}_w = \boldsymbol{F}^{ex}_w + \boldsymbol{F}^{im}_w \tag{7.1}$$

where $\boldsymbol{M}_w$ and $\boldsymbol{C}_w$ are respectively the mass and damping matrices of the workpiece; $\boldsymbol{u}_w$, $\dot{\boldsymbol{u}}_w$ and $\ddot{\boldsymbol{u}}_w$ are respectively the displacement, velocity and acceleration vectors; and $\boldsymbol{F}^{in}_w$ and $\boldsymbol{F}^{ex}_w$ are respectively the internal and external forces. It is convenient to express the forces imparted by the shot on the workpiece in a separate force vector $\boldsymbol{F}^{im}_w$.

Based on Newton's second law, the governing equations for the shot are given by

$$\boldsymbol{M}_s\ddot{\boldsymbol{u}}_s = \boldsymbol{F}^{im}_s \tag{7.2}$$

where $\boldsymbol{M}_s$ the diagonal mass matrix; $\boldsymbol{u}_s$ and $\ddot{\boldsymbol{u}}_s$ are respectively the displacement and acceleration vectors of the shot; and $\boldsymbol{F}^{im}_s$ is the impact force from the workpiece.

Let $\boldsymbol{u}_b$ denote displacements of the boundary nodes of the workpiece which may at some stage be in contact with the shot, and $\boldsymbol{u}_i$ denote displacements of the inner nodes which will not make contact with the shot, and assume further more that no external forces apart from the impact force is exerted on the workpiece. Then Eq. (7.1) can be rewritten in matrix form as

$$\begin{bmatrix} \boldsymbol{M}_i & \boldsymbol{M}_{ib} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_b \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}_i \\ \ddot{\boldsymbol{u}}_b \end{Bmatrix} + \begin{bmatrix} \boldsymbol{C}_i & \boldsymbol{C}_{ib} \\ \boldsymbol{C}_{bi} & \boldsymbol{C}_b \end{bmatrix} \begin{Bmatrix} \dot{\boldsymbol{u}}_i \\ \dot{\boldsymbol{u}}_b \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{F}^{in}_i \\ \boldsymbol{F}^{in}_b \end{Bmatrix} = \begin{Bmatrix} 0 \\ \boldsymbol{F}^{im}_b \end{Bmatrix} \tag{7.3}$$

The impact forces $\boldsymbol{F}^{im}_s$ and $\boldsymbol{F}^{im}_b$ between shot and workpiece in equations (7.2) and (7.3) are governed by contact interaction laws. Generally speaking, they both are nonlinear functions of, at least, $\boldsymbol{u}_s$ and $\boldsymbol{u}_b$, which consequently couple (7.2) and (7.3) together.

The coupled nonlinear system of equations (7.2) and (7.3) are solved explicitly by the central difference algorithm described in Chapter 1. To take advantage of the central difference scheme, the mass matrix of the FE part, $\boldsymbol{M}_w$, is assumed to be diagonal so as to avoid the solution of a linear system of equations at each time step.

Also note that in the context of the above explicit solution procedure, the *critical time step* Δt_{cr} is bounded by the stability condition defined in Chapter 3.

7.4 2D Analysis

In this section, attention is given to 2D plane strain analysis. It should be pointed out that 2D plane strain representation does not provide suitable strain and stress constraints for numerical simulations of realistic shot peening processes. Instead, it is considered to provide a convenient platform for parametric studies. In the case of a single shot, 2D axisymmetric conditions could be the appropriate analysis setting, while in general, 3D simulation is necessary.

Consider a single shot system as shown in Fig. 7.6(a), where the workpiece occupies a space domain Ω with Γ as its boundary, and the shot, represented by a circular disk, travels with a speed of $\boldsymbol{v}$ toward the workpiece.

As a result of the FE discretisation using linear (triangular or quadrilateral) elements for the domain, the boundary Γ is approximated by a set of facets, which in the 2D case can be represented by the line segments, as shown in Fig. 7.6(b). Consequently the contact between the shot and workpiece can be modelled by the contact of a disk with a segment.

It should be emphasised that although each segment itself is one of the edges of an element, and thus deformable, it is assumed to be rigid in the present DEM contact model. The actual deformable feature of the segment is, however, taken into consideration by the deformation of the whole FE mesh when the coupled system of equations from both finite elements and discrete elements are solved. This is illustrated in Fig. 7.7. This feature constitutes the major difference that the role of contact interaction laws play in the pure DEM and the combined FEM/DEM simulations. It leads to the view that all the interaction models to be presented are penalty based method in the present combined FE/DE approach.

DE modelling of disk/segment contact has been established in Chapter 3. In the remainder

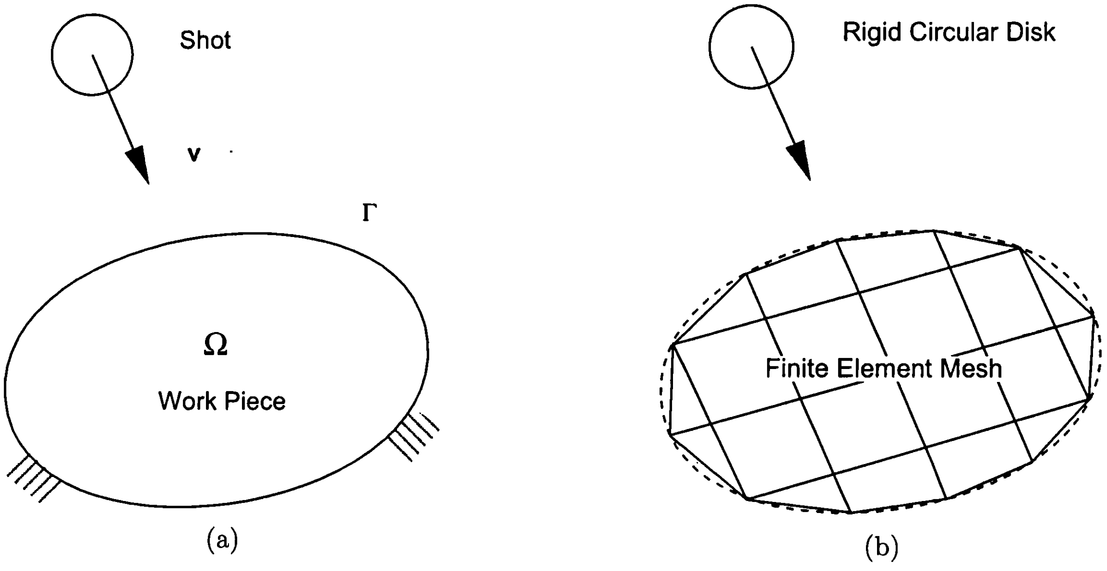


Figure 7.6: Single shot system – (a) original; (b) FE/DE discretisation

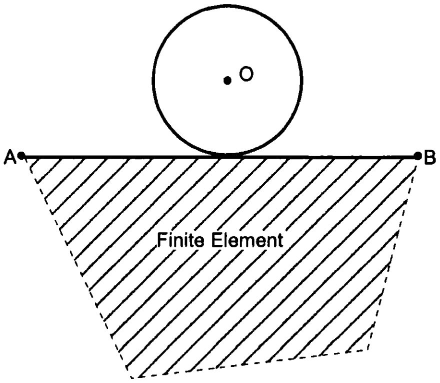


Figure 7.7: Disk and segment contact

of the section, a set of numerical experiments will be conducted to provide an assessment of the performances of the linear and Hertz normal contact interaction models proposed in Chapter 3 with different values of the model parameter (i.e penalty) α within the combined FE/DE strategy applied to a single shot impact system. The results are also compared with those using solely FE simulations, in which both workpiece and shot are discretised to finite elements, and the standard penalty algorithm with node-to-node contact is employed to simulate the contact between the workpiece and shot. Similar to the interaction laws considered, the penalty value associated with the algorithm is also assumed to be $k_n = \alpha E$, where E is the Young's modulus of the workpiece and α is a non-dimensional parameter.

The single shot system to be considered is the 2D version of the system investigated by Edberg *et al.* [9] where both numerical simulation and experimental verification have been undertaken. However, there is no direct comparison between the present computation and Edberg's. In addition, the shot is also assumed to be rigid rather than elasto-plastic deformable in the present combined FE/DE analysis. The dimensions of the workpiece is $60mm \times 40mm$ and the radius of the shot is $20mm$. The bottom of the workpiece is fixed in both x and y directions.

Fig. 7.8(a) and (b) respectively show the meshes of the single shot system discretised by FEM and combined FEM/DEM. The mesh of the workpiece consists of 2692 unstructured quadrilateral plane strain elements and 2769 nodes, while the shot is discretised into 418 quadrilateral elements and 483 nodes in the FE case.

The workpiece is assumed to be made of an elasto-plastic material with Young's modulus $E = 2.1 \times 10^5 N/mm^2$, Poisson's ratio $\mu = 0.3$, yield stress $\sigma_Y = 349.8N/mm^2$ and nonlinear isotropic working hardening. The hardening curve, $\bar{\sigma}(\bar{\epsilon}_p)$, is defined by linear interpolation between the values listed in Table 7.1. In the FE case, the Young's modulus of the shot, E_s,

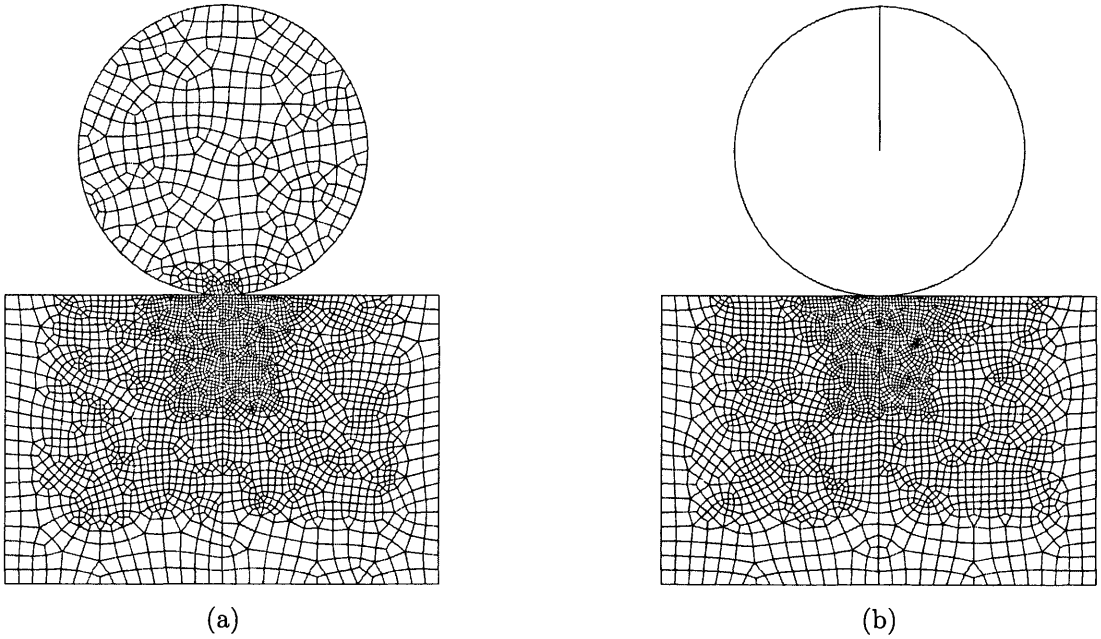


Figure 7.8: Meshes of single shot system – (a) FE analysis; (b) combined FE/DE analysis

Table 7.1: *Hardening Property of Workpiece ($\bar{\sigma}$: N/mm^2)*

$\bar{\epsilon}_p$	0.0	0.0017	0.0039	0.0079	0.0153	0.0288	0.0533	0.0975
$\bar{\sigma}$	349.793	605.419	645.896	689.08	735.152	784.303	836.741	892.684

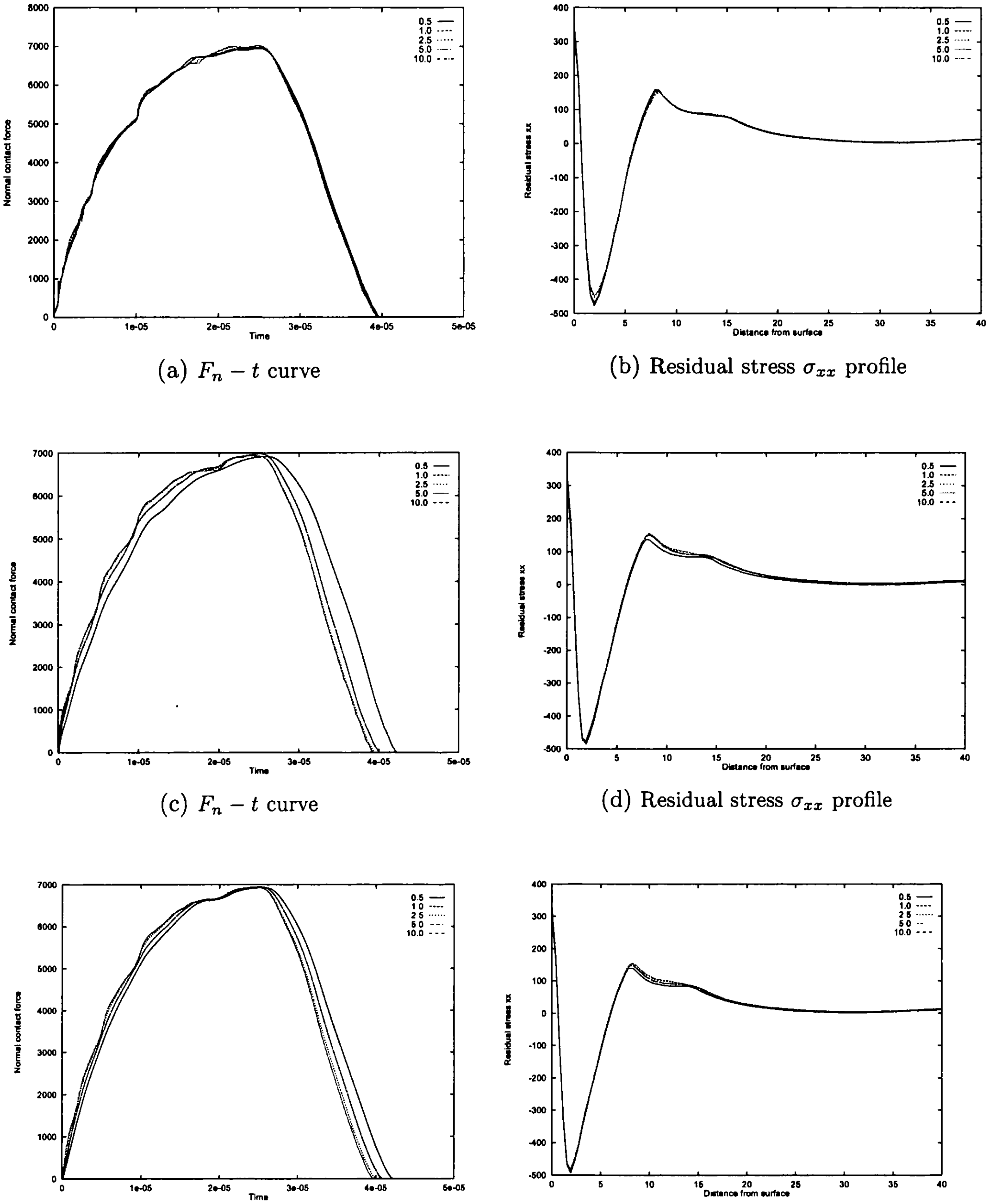
is taken as $E_s = 100E$ to approximately represent a rigid shot.

The impact velocity of the shot is $12.2m/s$ with an angle of impingement of 90^o. No friction is considered between the shot and workpiece during contact, but a critical viscous damping $\xi_n = 0.3$ is applied throughout the simulation. The simulation procedure is terminated only if a reasonable degree of steady-state is established. The actual termination time is set to be $10^{-5}s$, which is about 2.5 times the impact duration.

7.4.1 Comparisons of linear and Hertz models with different values of α

The first set of tests aims to compare the behaviour of the linear and Hertz contact models with reference to the FE analysis, and also to provide information about a suitable value range of parameter α. Five different values of $\alpha = 0.5, 1, 2.5, 5$ and 10 have thus been tested for each analysis.

Figs. 7.9(a)$\sim$(f) respectively plot the time histories of the normal contact force acting on



(a) $F_n - t$ curve

(b) Residual stress σ_{xx} profile

(c) $F_n - t$ curve

(d) Residual stress σ_{xx} profile

(e) $F_n - t$ curve

(f) Residual stress σ_{xx} profile

Figure 7.9: Computed results for the FE and combined FE/DE analyses with 5 different values of α: (a,b) FEM; (c,d) combined FEM/DEM with linear contact model; (e,f) combined FEM/DEM with Hertz contact model

the shot during impact and the final residual stress σ_{xx} profiles along the centre line of the workpiece for the FEM/DEM computations with linear and Hertz models, and the FE analysis with 5 different values of α.

Theoretically, the penalty value should be sufficiently large to satisfy the impenetrability constraint condition to some accuracy, but it appears that when α is taken to be 10, significant oscillation in the history plot of normal contact force has been observed for all the models, suggesting that α should be smaller than 10.

On the other hand, for $\alpha = 0.5$, i.e. the penalty is smaller than the Young's modulus of the workpiece in the FE/DE analysis with two contact models, a longer impact interval has been observed, which means that slightly smaller strain rates may be produced in the workpiece. Although the maximum contact force and the final residual profile are very close to those obtained with the other penalty values, there may have a larger difference if a visco-plastic material property is used for the workpiece.

When α is increased from 0.5 to $1, 2.5$ and 5, the corresponding $F_n - t$ curves become closer and also become less smooth at the same time. Nevertheless, almost identical residual profiles have been achieved, implying that the results are not sensitive to α in this range.

It is also found that the FE analysis has the smallest difference in the contact force history and the residual profiles for different values of α, demonstrating that the FE analysis under the current implementation is not sensitive to the values of α. However, a different degree of oscillation in the $F_n - t$ curves has been noticed for all penalty values considered. This phenomenon may be explained by the fact that the boundary of the disk is discretised into line segments and a node-to-node contact formulation is employed in the FE analysis, which may contribute to those oscillations. On the other hand, the current FE/DE analysis adopts the disk-to-segment model which handles the contact situation better and thus has much smoother $F_n - t$ curves.

To examine if there is significant difference between the FE and FE/DE analysis with different contact models, Fig. 7.10 depicts the time histories of the contact force and the profile of σ_{xx} along the centre line for all three analysis cases with $\alpha = 2.5$. It can be seen that nearly identical results have been attained.

In summary, similar behaviour has been observed for both linear and Hertz contact models in the combined FE/DE analysis in these numerical experiments. However, the performance of the Hertz model is slightly better in terms of overall stability and a larger value range for α.

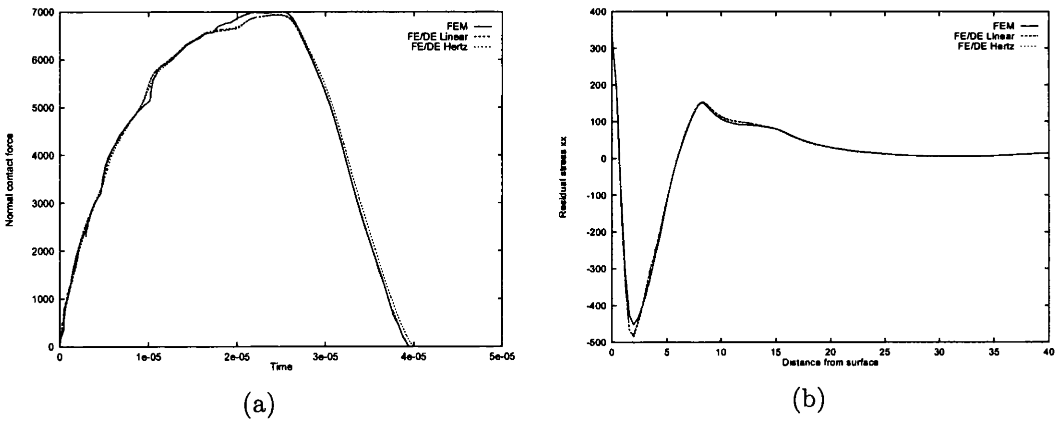


Figure 7.10: Comparison of computed results between FE, FE/DE with linear contact model and FE/DE with Hertz contact model for $\alpha = 2.5$: (a) time histories of normal contact force; (b) residual stress σ_{xx} profile

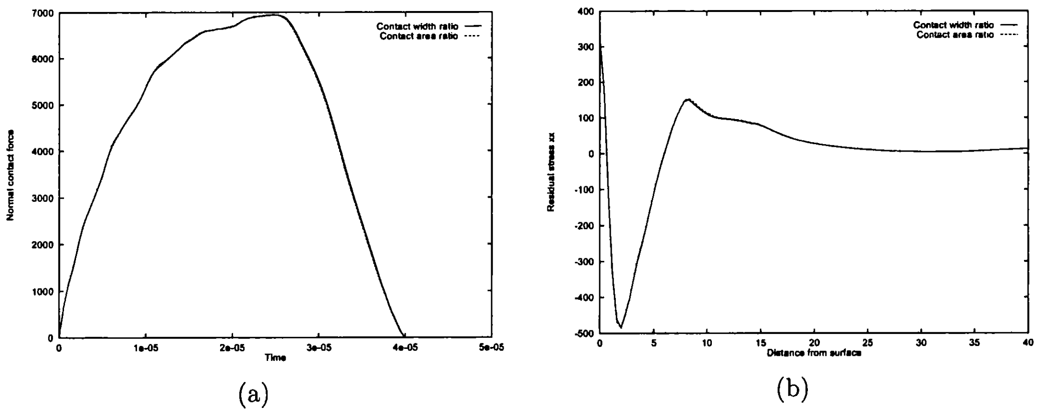


Figure 7.11: Results comparison of two partial contact factor γ evaluation approaches: (a) time histories of normal contact force; (b) residual stress σ_{xx} profile

7.4.2 Comparisons of two partial contact factor evaluation schemes

The second set of tests aims to compare the behaviour of the two different schemes presented in Chapter 3 for the evaluation of the partial contact factor γ. In this case, only the Hertz model with $\alpha = 2.5$ is considered. The corresponding time histories of the normal contact force and the final residual stress σ_{xx} profiles along the centre line of the workpiece are shown in Fig. 7.11. It is evident that almost identical results have been obtained by the two approaches and thus it is confirmed that the contact width ratio can be adopted as an inexpensive and effective approach for the evaluation of the partial contact factor γ in the contact models.

7.5 3D Analysis

In the previous sections, a general algorithmic framework has been established on the combined finite/discrete element analysis of shot peening processes, particularly for the evaluation and comparison of several interaction laws governing the contact between a 2D disk and segment of a solid surface.

The representation of shot as 2D disks, and workpiece as 2D plane strain FE problem is not fully valid for the simulation of shot peening process as has been demonstrated in the numerical tests. The resulting tensile stress on the shot peened surface suggests that the 2D model cannot capture the major fundamental physics and characteristics of the shot peening process, therefore a full 3D simulation is in general required.

Consider a single shot impact system as depicted in Fig. 7.12, where the workpiece occupies a space domain Ω, and the shot of radius R, travels with a speed of $\boldsymbol{v}$ toward the workpiece. In

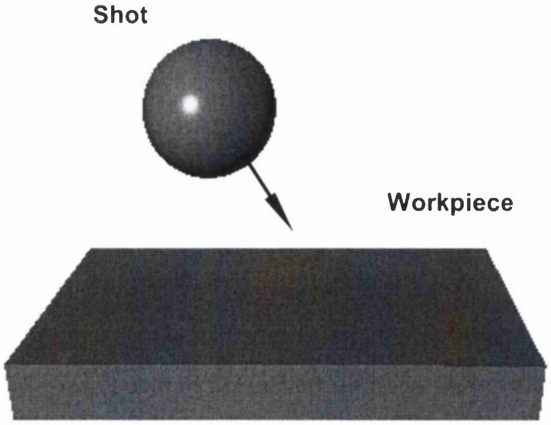


Figure 7.12: Single shot system

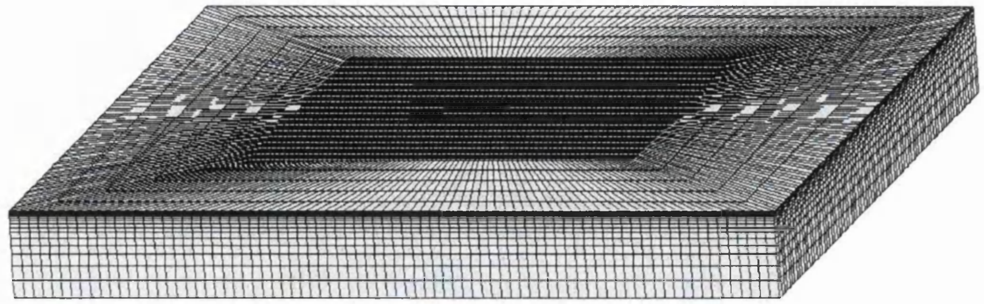


Figure 7.13: 3D FE mesh of the component

the current combined FE/DE scheme, the workpiece to be peened is modelled by conventional FE techniques, while the shot is represented as a rigid sphere by discrete element.

As a result of the FE discretisation using linear elements (tetrahedral or hexahedral) for the domain, the boundary is approximated by a set of 3- or 4-noded facets. For the purpose of simplifying contact detection procedures, each 4-noded facet is further divided into two 3-noded facets. Consequently the contact between the shot and workpiece can be modelled by the contact of a sphere with a 3-noded facet. DE modelling of sphere/facet contact has been established in Chapter 3.

A single and a multiple shot impact model is to be examined. In the former instance, the upper surface of a small component is impacted by a single sphere, whereas it is shot peened by 16(= 4 × 4) regularly packed spheres in the latter case. For both problems, it is assumed that the shot is of radius 0.5 mm and impacts at a velocity of 19 m/s. The component, of dimension 8 × 8 × 0.9, is discretised as 8-noded solid elements with average element size 0.05 × 0.05 × 0.05 mm in the impact area as shown in Fig. 7.13. Its material is assumed to be aluminium alloy with Young's modulus 68 GPa, yield stress 157 MPa and a linear strain hardening rate of 120 MPa/unit strain. The following boundary conditions are imposed: the bottom surface is fixed in all three directions; the 4 side surfaces are fixed in their normal directions; and the top surface is free. A global damping of 0.05 is applied to all degrees of freedom to damp out the kinetic energy.

7.5.1 Comparison with axisymmetric model

The first experiment is provided to assess the accuracy of the normal interaction model proposed in Chapter 3 by comparing with the axisymmetric finite element simulation. The single shot impact problem is considered in this instance. The axisymmetric model is shown

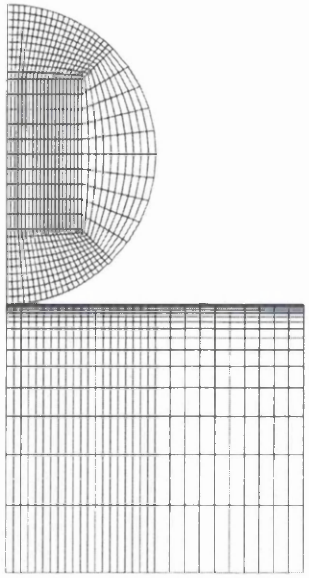
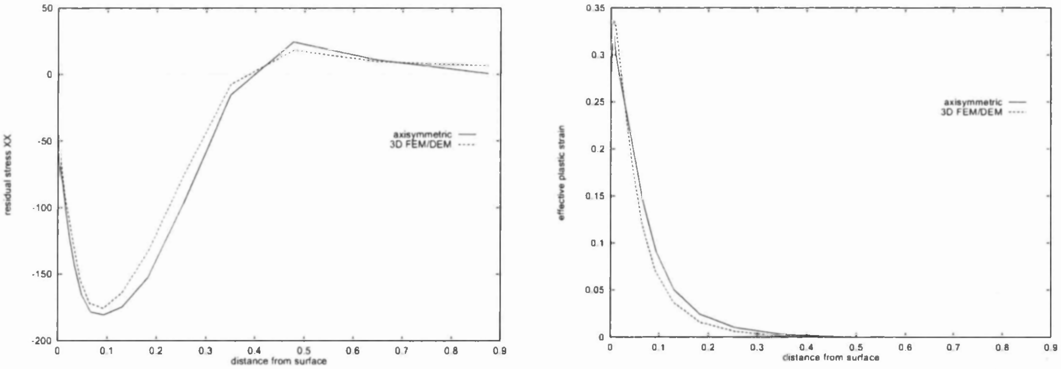


Figure 7.14: Axisymmetric FE model of single shot system

in Fig. 7.14, in which both workpiece and shot are discretised to finite elements, and the standard penalty algorithm with node-to-facet contact is employed to simulate the contact between the workpiece and shot. The same mesh density and boundary conditions as in the 3D model are imposed. The penalty value for both 3D normal contact and axisymmetric models is assumed to be the same and taken as $k_n = 5E$, where E is the Young's modulus of



(a) (b)

Figure 7.15: Comparison of axisymmetric and 3D FEM/DEM results: (a) residual stress σ_{xx}; (b) effective plastic strain $\bar{\epsilon}_p$

the component.

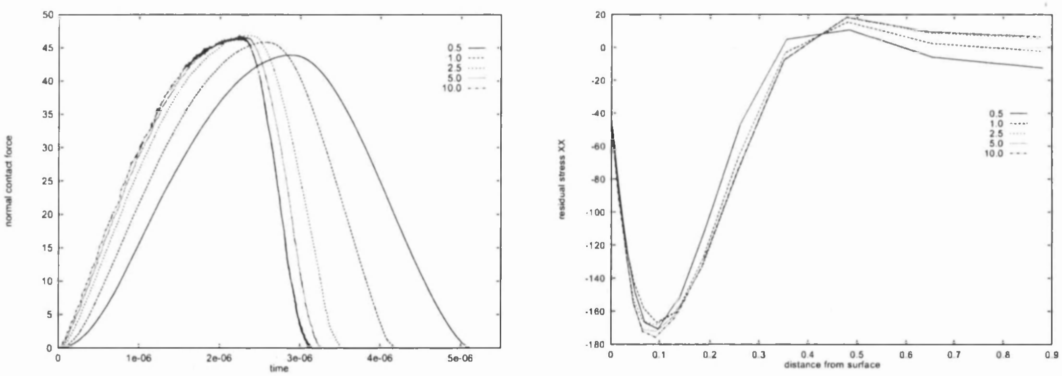
Figs. 7.15(a) and (b), respectively, depict the residual distributions of the stress σ_{xx} and the plastic strain $\bar{\epsilon}_p$ along the centre line of the component. A good agreement between the axisymmetric and 3D FEM/DEM models is clearly demonstrated.

7.5.2 Choices of parameter values α for normal contact models

The second experiment aims to investigate the performance of the normal contact models for different model parameter values α. The single shot impact problem is again considered.

Fig. 7.16(a) and (b), respectively, plot the time histories of the normal contact force acting on the shot during impact and the final residual stress σ_{xx} distribution along the centre line of the component. The penalty value is assumed as $k_n = \alpha E$. Five different values of α =0.5, 1, 2, 5 and 10 are tested.

Basically, the choice of the penalty value is governed by the accuracy and stability conditions required. On one hand, α should be sufficiently large to satisfy the impenetrability constraint condition to some degree as large penetrations will compromise solution accuracy. On the other hand, however, large penalty values may deteriorate the conditioning of the dynamic system which may cause contact oscillation and even stability problems in explicit analysis. It appears that when α is taken to be 10, significant oscillation in the history plot of normal contact force occurs for all models. For $\alpha \leq 0.5$, i.e. the penalty is smaller than the Young's modulus of the impacted material, a longer impact interval has been observed, which means that slightly smaller strain rates may be produced in the component. Therefore $\alpha = (1 \sim 5)$



(a)

(b)

Figure 7.16: Computed results for normal contact linear model with five different values of α =0.5,1,2.5,5,10: (a) time histories of normal contact force; (b) residual stress σ_{xx} profile

is suggested for use in the contact models. Alternatively, an appropriate penalty value can be found based on the energy balance during impact with the condition that a maximum penetration is prescribed (see [10] for details).

Although the maximum contact force and the final residual profile do not differ significantly from those obtained with other penalty values, there may be a larger difference if visco-plastic material properties are used for the component.

7.5.3 Effect of contact friction

The single shot impact problem is used to evaluate the modified Coulomb friction model proposed in Chapter 3. The impact angle is 75^o to the horizontal direction for the tests with five different frictional coefficients μ = 0.0, 0.05, 0.1, 0.2 and 0.5. The resulting residual stresses as shown in Fig. 7.17 indicate that friction plays a significant part in the contact model. The resulting residual stress is increased with increase of the frictional coefficient μ. When $\mu \geq 0.2$, however, further increase of μ has no significant influence on the residual stresses.

7.5.4 Effect of element size

Element size in the impact region affect the solution accuracy of the explicit procedure significantly.

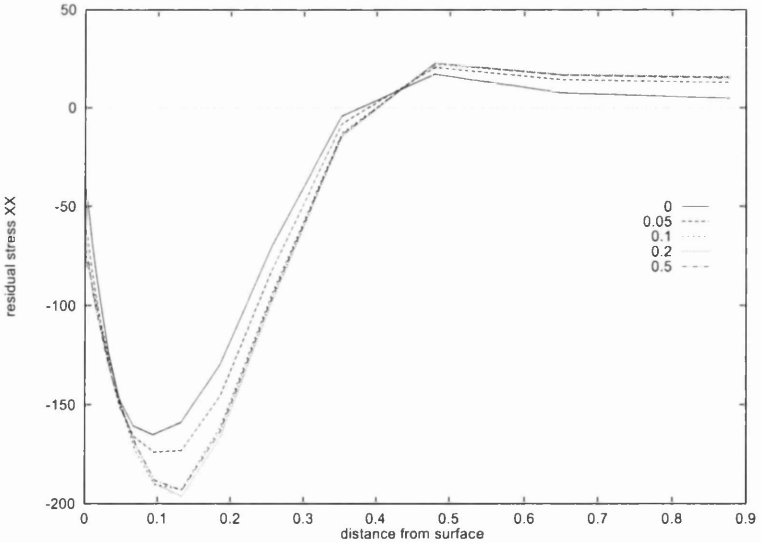


Figure 7.17: Residual stress σ_{xx} profile for five different frictional coefficients μ = 0.0, 0.05, 0.1, 0.2, 0.5.

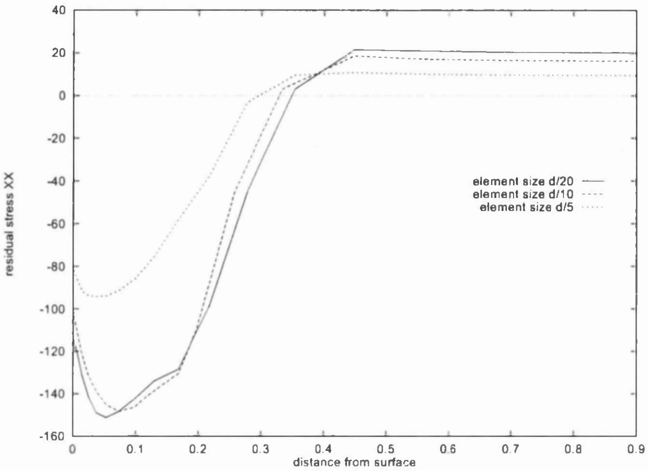


Figure 7.18: Residual stress σ_{xx} profile for three different element sizes $d/5$, $d/10$, $d/20$

Theoretically, smaller element sizes give rise to smaller time steps for the explicit integration, resulting in more accurate solutions. Nevertheless, much reduced element sizes will result in large FE models that may not be solvable in terms of the current available computer resources.

The multiple impact model is chosen to test the effect of element size on residual profile. Three different element sizes for the impacted region of the component are considered, $d/5, d/10$ and $d/20$, where d is the diameter of the shot.

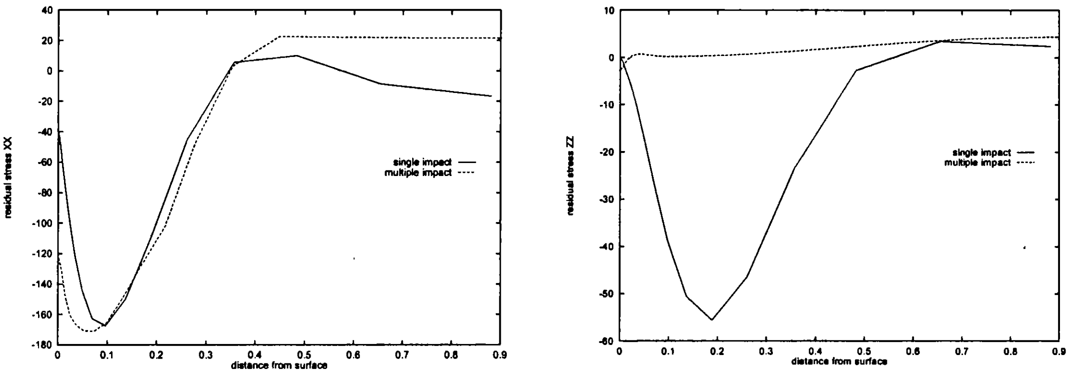
The residual stress σ_{xx} distributions for the three mesh sizes are depicted in Fig. 7.18, which shows that in this case the element size in the impact area should be no larger than $d/10$.

7.5.5 Effect of single and multiple shot impact

In most published work on numerical simulations of shot peening processes, the residual profile is identified from the stress/strain distributions on a section immediately beneath the shot in a *single* shot impact simulation.

To examine the effect of single and multiple shot impact on the residual stress, numerical tests are respectively performed with the above single and multiple models.

The results show that the residual stresses resulting from single and multiple shot impact are different in terms of magnitude and distribution as shown in Figs. 7.19(a) and (b). Consider the stress component σ_{zz}, for instance. Under the assumption that the residual stresses are uniformly distributed at the same depth from the peened surface, it is trivial to draw the conclusion that σ_{zz} must be zero in order to satisfy the global equilibrium in the z direction.



(a)

(b)

Figure 7.19: Comparison of single and multiple shot impact: (a) σ_{xx}; (b) σ_{zz}

In the single shot situation, however, a nonzero σ_{zz} distribution is actually obtained as shown in Fig. 7.19(b), which indicates that the residual profile obtained from single shot simulation may not be fully valid.

7.6 Verification against Experimental Measurements

The experimental work has been carried out and summarised by Evans [14]. It is outlined below.

Three important commercial alloys, Al-2.5Mg-0.25CR (aluminum alloy Al 5052), 316 stainless steel and Ti-6Al-4V, were chosen to give a range of elastic and plastic properties and their melting points range from low to high. The material parameters for all three alloys, as shown in Table 7.2, were obtained by employing testing facilities at the Materials Engineering Department, University of Wales Swansea. In order to establish the rate dependent characteristics of materials compression testing of axi-symmetric cylinders of height 16mm and diameter 12mm has been carried out using a high velocity servo-hydraulic machines. Strain rates have been maintained constant during compression by reducing the crosshead velocity controllably during the test. Specimens were lubricated with PTFE strips ensuring that friction was maintained close to zero and also that homogeneous deformation occurred [13].

Metal Improvement Company (MIC) carried out shot peening of three alloys. Specimens of the aluminum alloy and stainless steel were rectangular blocks of approximate dimensions 23x23x16 mm. Titanium alloy samples were discs of diameter 50 and depth 10 mm. All specimens were peened to a coverage of 200% and further details of the peening conditions, together with sample identification numbers are shown in Table 7.3.

Table 7.2: *Mechanical Properties for the Three Experimental Materials*

Material	Young's modulus *GPa*	Poisson ratio	yield stress *MPa*	strain rate exponent	work hardening rate *GPa*
Al 5052	69.3	0.31	148	0.0455	194
316 stainless steel	193	0.328	366	0.0303	2430
Ti-6Al-4V	111	0.33	1193	0.0122	771

Table 7.3: *Experimental Observations on Shot Peened Specimens*

specimen number	shot radius *mm*	Almen A intensity *mm*	indent radius *mm*	standard deviation *mm.*
Al 5052				
A1	0.140	0.121	0.087	0.019
A2	0.140	0.167	0.112	0.017
A3	0.286	0.284	0.1505	0.024
A4	0.286	0.438	0.187	0.0225
A5	0.417	0.449	0.2455	0.0265
A6	0.417	0.601	0.305	0.0665
316 stainless steel				
S1	0.140	0.121	0.0675	0.016
S2	0.140	0.167	0.0865	0.0165
S3	0.286	0.284	0.120	0.0205
S4	0.286	0.438	0.1495	0.019
S5	0.417	0.449	0.1925	0.027
S6	0.417	0.601	0.2375	0.0225
Ti-6Al-4V				
T1	0.140	0.121	0.05	0.011
T2	0.140	0.167	0.0685	0.0115
T3	0.286	0.284	0.088	0.0135
T4	0.286	0.438	0.116	0.014
T5	0.417	0.449	0.1505	0.021
T6	0.417	0.601	0.1855	0.043

The combined FE/DE.method described in the previous section was then performed for all three alloys and with varied process parameters. In order to cover the critical aspects of the process, the following parameters were used in both experimental and numerical studies: (1) shot velocity 50 m/s; (2) shot size with diameter of 0.28mm and 0.84mm; and (3) impingement angle of 90^o and 60^o.

The critical process parameter was considered to be the residual stress profile. Hence the calculated residual stress fields were compared with those obtained from experimentation. The experimental methods available for residual stress determination have been summarised in [15]. Methods involving the removal of metal and the subsequent measurement of dimensional changes are not ideally suited to the present situation where the depth of the residual compressive layers may only be of the order of tens of microns. Of greater usefulness are diffraction methods. The surface is irradiated with beams of suitable wavelength and the resulting diffraction pattern analysed to measure any elastic change in the residual stress. X-radiation, synchrotron radiation and neutron radiation have all been used for this purpose. The penetrating power of X-rays is such that only thin layers at the surface are analysed (for titanium alloys, only a few microns). The technique has to be supplemented by the successive removal of surface layers and further diffraction experiments.

Residual stress measurements have been made on Ti-6Al-4V specimen peened under the conditions corresponding to specimen T5 in Table 7.3. Neutron strain scanning was conducted

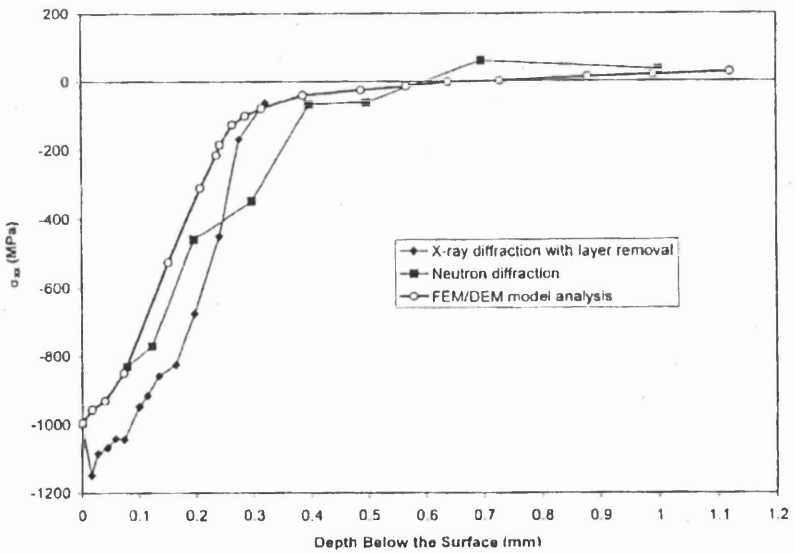


Figure 7.20: Comparison between residual stress profile calculated by FEM/DEM for specimen T4 and those observed in X-ray and neutron diffraction experiments

at the Rutherford Appleton Laboratory and X-ray scanning at Rolls-Royce plc. In both cases, the residual stress in the surface was calculated on the basis of a balanced biaxial stress field. The results of the diffraction experiments are shown in Fig. 7.20 together with the σ_{xx} values calculated from the FE/DE model run under the same peening conditions. The predicted maximum compressive stress agrees well with the measured value as does the depth beneath the surface which has compressive residual stresses. The predicted σ_{xx} profile is also very good bearing in mind the uncertainties associated with the diffraction measurements. The verification is sufficiently good to allow an investigation into how the numerical model might be used to optimise the shot peening process.

7.7 Concluding Remarks

A general algorithmic framework has been established in this work on the combined finite/discrete element analysis of shot peening processes, particularly for the evaluation and comparison of several interaction laws governing the contact between shot and workpiece. It is found, via numerical experiments, that the representation of shot as 2D disks, and workpiece as 2D plane strain FE problem is not fully valid for the simulation of shot peening process. The resulting tensile stress on the shot peened surface suggests that the 2D model cannot capture the major fundamental physics and characteristics of shot peening process, therefore a full 3D simulation is in general required.

The proposed 3D representation of the problem is carefully examined. Numerical tests are performed to evaluate the normal and frictional contact interaction laws with various model parameters. Effects of single and multiple shot impact, as well as element sizes are also numerically investigated. The results indicate that the magnitude and distribution of residual stresses resulting from single shot impact are different from those from multiple shot impact as occur in practical industrial applications, thus multiple shot impact simulations should be employed to provide acceptable residual profiles. Numerical experiments also suggest that the element size in the impact area should be no larger than 1/10 of the shot diameter to guarantee sufficient solution accuracy of the explicit procedure.

The proposed FE/DE model has been verified by experiments on three important commercial alloys, namely Al-2.5Mg-0.25CR (aluminum alloy Al 5052), 316 stainless steel and Ti-6Al-4V. The calculated maximum compressive stress agrees well with the measured value as does the depth beneath the surface which has compressive residual stresses.

Bibliography

[1] *Shot peening applications.* Metal Improvement Company, Inc. (7th Edition).

[2] J. S. Eckersley. Basic techniques and key parameters. In K.J. Marsh, M.E., and D. Phil., editors, *Shot peening: techniques and applications*, pages 19-36. The Chameleon Press Ltd, London, UK, 1993.

[3] S. T. S. Al-Hassani. Mechanical aspects of residual stress development in shot peening. In *ICSP1: 1st International Conference on Shot Peening*, pages 583-602, Paris, France, 1981.

[4] Y. F. Al-Obaid. Shot peening mechanics: experimental and theoretical analysis. *Mechanics of Materials*, 19:251-260, 1995.

[5] D. A. Hills, R. B. Waterhouse, and B. Noble. An analysis of shot peening. *J. Strain Analysis*, 18:95-100, 1983.

[6] L. V. Grasty and C. Andrew. Shot peen forming sheet metal: finite element prediction of deformed shape. *Proc. Instn. Mech. Eng.*, 210:361-366, 1996.

[7] Y. F. Al-Obaid. Three-dimensional dynamic finite element analysis for shot-peening mechanics. *Computers & Structures*, 36:681-689, 1990.

[8] Y. F. Al-Obaid. Multiple shot analysis in shot peening using finite elements. In M.H. Aliabadi and C.A. Brebbia, editors, *Computer Methods and Experimental Measurements for Surface Treatments*, pages 155-168. Computational Mechanics Publications, 1993.

[9] J. Edberg, L. E. Lindgren, and K. I. Mori. Shot peening simulated by two different finite element formulations. In Shen & Dawson (eds), editor, *Simulation of Materials Processing: Theory, Methods and Applications - NUMIFORM 95*, pages 425-430, Balkema, Rotterdam, 1995.

[10] J. Yu. *A contact interaction framework for numerical simulation of multi-body problems and aspects of damage and fracture for brittle materials.* Ph.D. Thesis, University of Wales Swansea, 1999.

[11] N. Petrinić. *Aspects of discrete element modelling involving facet-to-facet contact detection and interaction.* Ph.D. Thesis, University of Wales Swansea, 1996.

[12] O. R. Walton. Numerical simulation of inelastic, frictional particle-particle interactions. In M.C. Roco, editor, *Particulate Two-Phase Flow*, pages 884-911. Butterworths, Boston, USA, 1992.

[13] R. W. Evans and P. J. Scharning. The axi-symmetric compression test and the hot working of alloys. *Mat. Sci. Techn.*, 17:995-1004, 2001.

[14] R. W. Evans. Shot peening process: modelling, verification and optimisation. *Mater. Sci. Tech.*, 18(8):831-839, 2002.

[15] P. J. Withers and H. K. D. H. Bhadeshia. Residual stress. Part I: measurement techniques. *Mat. Sci. Techn.*, 17:355-365, 2001.

Chapter 8

Combined Finite/Discrete Element and Explicit/Implicit Simulations of Peen Forming Processes

8.1 The Peen Forming Process

Peen forming is a dieless forming process performed at room temperature. During the process, the surface of the workpiece is impacted upon by shot. Every piece of shot impacting the surface acts as a tiny peening hammer, producing elastic stretching of the upper surface and the local plastic deformation that manifests itself as a residual compressive stress. The surface force of the residual compressive stress combined with the stretching causes the material to develop a compound, convex curvature on the peened side (Fig. 8.1). The process is ideal for forming large panel shapes where the bend radii is reasonably large without abrupt changes in contour.

Peen forming is often more effective than rolling, stretching or twisting of metal to develop the required curvatures. Saddle-back shapes are possible and by being a dieless process, material allowance for trimming is reduced, and costly development and manufacturing time required to fabricate hard dies is eliminated.

Another distinct advantage with peen forming, unlike most other forming methods, is that all surface stresses generated are of a compressive nature. Although peen formed pieces usually require shot peening on one side only, the final result causes both sides to have compressive stress. These compressive stresses serve to inhibit stress corrosion cracking and to improve fatigue resistance. Some workpieces should be shot peened all over prior to or after peen forming to further improve fatigue and stress corrosion characteristics. Parts which have

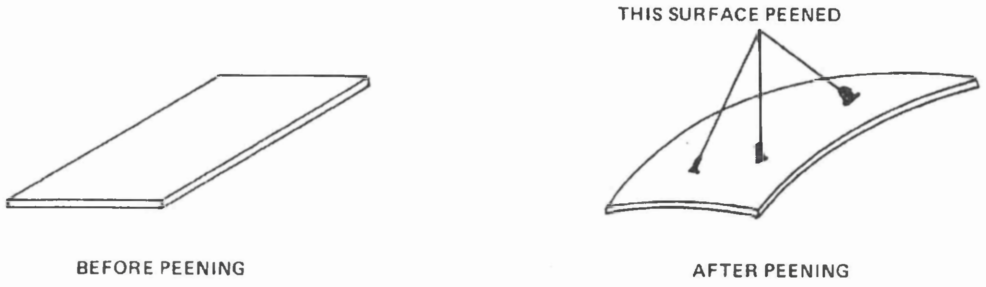


Figure 8.1: (reproduced from [1]): Compound curvature result of tri-axial forces induced by shot peening

been cold formed by other processes are often shot peened to overcome the harmful surface tensile stresses.

8.2 Review of Modelling of the Peen Forming Process

Compared to the increasing research interest in the determination of the residual stress profile under particular peening conditions by using FEM or DEM or their combinations (see [2, 3] and the references therein), peen forming has received little considerations in terms of numerical simulations. Some limited relevant research on the topic can be broadly classified into two categories: *direct simulation* and *equivalent loading.*

The direct simulation approach attempts to model real shot peening and forming processes by directly employing FEM and/or DEM. Theoretically, this would be an ideal approach since more realistic peening conditions could be taken into account in the numerical simulation process. In practice, however, a large number ($> 10^5$) of shot may need to be simulated and very large scale 3D dynamic FE systems also need to be solved. The solution procedures involved prove to be extremely expensive in terms of computer CPU time and memory requirements, which makes it very difficult to undertake numerical simulations even for a moderate scale problem with the currently available computing resources.

Grasty and Andrew [4] proposed an equivalent loading system in which the upper layers of the workpiece were subjected to a 'squeezed pressure' such that they underwent a small plastic deformation. A range of squeeze pressures was tried, and the one that gave the same curvature as the indentation case was determined. A problem arises in the choice of the magnitudes of both squeeze pressure and the depth of the squeeze layer if no indentation test is available for comparison. In fact, it is difficult to conduct an indentation test on most practical components.

Another approach for applying equivalent loading was described in [5] by Levers and Prior, in which having identified a shot peening residual stress profile, mostly from existing experiment results, an equivalent thermal loading was applied to the workpiece discretised by multi-layer shell elements. In combination with the coefficient of thermal expansion, it induced non-uniform thermal strains into the component, which then further cause global deformation for parts with thin thickness such as aircraft wing panels.

A *combined FE/DE and explicit/implicit simulation strategy* is proposed in this work, in which a residual stress/strain profile is initially identified by simulating the shot peening of a small region of the workpiece with explicit dynamic analysis. The resulting deformation and stress distribution of the entire workpiece are then obtained by employing this residual stress/plastic strain profile as initial conditions for a subsequent implicit static solution of the complete component. In this combined approach, the FE formulation is employed to model the behaviour of the workpiece including material, such as elasto-plasticity with (non-)linear hardening, and/or geometric nonlinearity. As the FE techniques involved are relatively mature and well documented elsewhere (see e.g. [6]), the following discussion will focus on the procedures specially designed for the simulation of peen forming processes.

8.3 Combined FE/DE and Explicit/Implicit Scheme: a General Description

The fundamental difficulty associated with the full scale numerical simulation is the unrealistic demand of computer resources in terms of memory and CPU times arising from the following factors:

- *Necessity of 3D analysis.* A number of numerical experiments have been performed with 2D plane strain simulations as illustrated in the previous chapter. The results demonstrated that 2D analysis could not capture the fundamental features of the shot peening process, and therefore a full 3D simulation is necessary.

- *Accuracy requirements.* The numerical tests performed in the previous chapter also suggest that in order to achieve a moderate level of solution accuracy, the element size in the impact area should be no larger than $d/10$, where d is the diameter of the shot. As d is typically in the range of $0.1 \sim 1$mm, the FE discretisation of, for instance, an Almen "A"-strip ($76 \times 19.05 \times 1.2954$ mm) with shot size $d = 1$ mm will lead to a FE model with over 1.8 million elements, and thus a memory requirement of around 9 GB in the explicit analysis.

- *Stability considerations.* In the explicit solution procedure, the marching time step must not be greater than the critical time step of the dynamic system. For the requirement of using sufficiently fine meshes as mentioned above, the time step used is usually very small which leads to a large number of time increments required to complete the solution. In addition, it also takes a long time to reach steady-state conditions, as the analysis is truly dynamic.

Consequently, it is impractical to employ only the explicit technique to accomplish a full scale peen forming simulation. Reviewing the underlying physics of the process suggests that the following two-stage combined explicit and implicit solution procedure could be a solution.

This procedure involves, at the first stage, the identification of the residual stress/strain profile under particular peening conditions by employing the combined FE/DE scheme on a *small scale sample problem*, and then as the second stage, the application of this profile to the *entire workpiece* to obtain the final deformation and stress distribution using an implicit static analysis.

The small scale shot peening model may be justified by the following considerations:

- Effect of shot impact is local, hence the residual stress/strain profile along a specific section under the surface is contributed to mainly from those shot that impact in its vicinity;
- It is not necessary to peen the complete workpiece whose geometry may be complex. It is sufficient, in terms of accuracy of the residual profile, to peen a small sample model with a regular domain and the same material subject to the same peening parameters;
- It is only necessary to discretise the region of interest into a sufficiently fine mesh with the remainder of the model being coarsely discretised.

The above considerations lead to a moderate sized FE model for the small scale problem to be considered at the first stage. At the second implicit analysis stage, a much coarser mesh can be applied, in particular in the horizontal directions due to the fact that horizontal stresses are relatively uniformly distributed for well-controlled shot peening. Consequently, the practical peen forming problems can be analysed with reasonable computer resources within reasonable times.

To facilitate the discussion, the workpiece is assumed to be a flat plate with its upper surface to be peened. A coordinate system with the x-y plane coincident with the lower surface and the z axis upward is assumed as shown in Fig. 8.2.

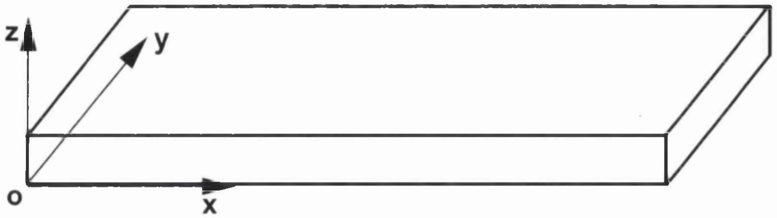


Figure 8.2: Default coordinate system

8.4 Residual Stress Profile: Combined FE/DE Explicit Approach

In chapter 7, the main building blocks for the combined FE/DE explicit procedure has been established. See [2, 3] for more details.

In order to identify a residual profile due to shot peening, a combined FE/DE explicit simulation is performed for a small sample model with a cubic geometry. The domain is discretised into 4-noded or 8-noded 3D solid finite elements with finer meshes in the impact area so as to guarantee sufficient accuracy of the explicit solution, while the shot are represented as rigid spheres by discrete elements.

As shown in the previous chapter, the single impact may not be able to capture the major fundamental physics and characteristics of shot peening processes. it is thus necessary to perform multiple shot impact, and the residual profiles of stress/strain and effective plastic strain along a section are obtained by averaging those of all the elements within the impacted region. Typically, the values at the nodes or Gauss points across the section are selected to make up the profile and these discrete values are then taken as the initial stress/strain conditions for the next analysis stage. We denote the computed residual stress, strain and effective plastic strain distributions from the explicit analysis respectively as $\bar{\sigma}(l), \bar{\varepsilon}(l)$, and $\bar{\epsilon}_p(l)$, where l is the distance from the peened surface.

Note that the boundary conditions imposed on the small scale problem are normally different from the real forming conditions, but these additional constraints will be removed at the implicit analysis stage.

8.5 Global Deformation and Stress Distribution: Implicit Static Solution

In the second stage, or implicit solution, the residual profiles, $\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\varepsilon}}$, along with some history dependent state variables (e.g. plastic strain) obtained from the explicit simulation are first applied as the initial stress/plastic strain distribution to the entire workpiece, then the implicit procedure is employed to solve the problem statically to attain the final deformation and stress states.

This procedure is based on the following assumptions:

1. the stresses/strains are uniformly distributed in horizontal directions;
2. the stresses/strains at the same depth from the peened surface are the same;

Therefore, the stress and strain states $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ at a point (x, y, z) are only functions of z, i.e. $\boldsymbol{\sigma}(x, y, z) = \boldsymbol{\sigma}(z)$ and $\boldsymbol{\varepsilon}(x, y, z) = \boldsymbol{\varepsilon}(z)$. We further assume that the residual profiles, $\bar{\boldsymbol{\sigma}}$ and $\bar{\boldsymbol{\varepsilon}}$, obtained from the explicit analysis can be considered as the initial distributions of $\boldsymbol{\sigma}(z)$ and $\boldsymbol{\varepsilon}(z)$.

Due to different constraint conditions between the sample model and the actual problem, the stress/strain field will not be self-equilibrating, and therefore when initialised with this pattern of residual distribution, the workpiece will be subjected to a non-equilibrating stress field that causes deformation and re-distribution so that equilibrium conditions are reached.

The direct consequence of the above assumptions is that a coarse FE mesh in horizontal directions can be used, which will significantly reduce the size of any FE model so that realistic peen forming problems can be solved. In addition, the element size distribution along the thickness is not necessarily the same as that of the sample problem in the explicit analysis. It may be more suitable to slightly increase element sizes near the peened surface and decrease sizes near the unpeened surface. This is motivated by the observation that after stress re-distribution, the compressive stress level near the unpeened surface for very thin plates can be significantly increased and in some cases, may undergo plastic yielding.

It is important to point out that because the upper layer of the workpiece has undergone plastic deformation during the impacting process, some history-dependent state variables should be applied to the workpiece along with the stress/strain components. Otherwise, an incorrect final residual distribution may result.

The main steps associated with the implicit analysis stage include: (1) automatic application of the residual profile, identified from the explicit analysis, as initial stress/strain to the implicit FE model of the workpiece, followed by an optional equilibrium iteration; (2) the computation of the equivalent loading induced from the applied initial stresses; and the use of incremental analysis if necessary.

8.5.1 Residual profile application

Since the two FE meshes for explicit and implicit analyses may be different along the thickness of the workpiece, some transformation is necessary when imposing the residual profile obtained from explicit solution to the whole workpiece. Here a linear interpolation method is used to determine the corresponding value of all stress/strain components at each node/Gauss point according to its distance from the peened surface. In the case of a curved plate, the position of the point is determined by its shortest distance to the peened surface.

There are several options for the application of the identified residual profile. Theoretically, both stress and strain components should be applied at the same time, together with some state variables. Nevertheless, as the displacement cannot be precisely transformed from the explicit model to the implicit in the current scheme, simultaneous application of stress and strain will not result in equilibrium. Consequently, either $(\boldsymbol{\sigma}, \epsilon_p)$ or $(\boldsymbol{\varepsilon}, \epsilon_p)$ can be chosen as the initial condition, following by an equilibrium iteration procedure.

The applied initial stress/strain state together with history-dependent variables in the implicit FE model may differ from the actual situation in the explicit stage due to the interpolation and other numerical procedures employed. Therefore local equilibrium will no longer be preserved and even the plastic yield condition may also be violated due to the numerical errors introduced. The first source of error arises from the computation of the residual distributions from multiple impact by averaging the results at the explicit stage. If the nodal values are chosen, additional errors may result since they are derived from the values at Gauss points. Further numerical errors may be introduced in the interpolation process if the two meshes are different in the z direction. Some local iteration may be performed to restore equilibrium at this stage. However, the numerical errors introduced in the profile transformation from explicit to implicit are not significant and the resulting loading induced from this error will be much smaller than the unbalanced force arising from the change of geometry and boundary conditions imposed. Therefore this equilibrium iteration can be avoided in general.

8.5.2 Incremental implicit analysis

Once the residual stress/strain and plastic strain are applied to the entire workpiece as the initial stress/strain state, the implicit static technique is then performed to determine the final deformation and residual stress distribution. This technique is briefly reviewed below for the initial stress case.

In the current modelling, both geometric and material nonlinearities can be taken into account in the FE formulation. If no external loading is applied (as the gravity force may be ignored), the global equilibrium equation of the workpiece can be expressed as

$$\int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}\, d\,\Omega = 0 \tag{8.1}$$

with an initial value $\boldsymbol{\sigma}_0 = \bar{\boldsymbol{\sigma}}$; and a nontrivial solution $\boldsymbol{\sigma}^* \neq 0$ is sought. In the large strain case, Ω is the final deformed configuration, while for small strain it is the original configuration, i.e. $\Omega = \Omega_0$. Note that the final stress distribution $\boldsymbol{\sigma}^*$ must be in self-equilibrium. For an Almen strip-like free supported plate, the distribution of σ^*_{xx} along a section should satisfy the following two conditions:

$$\int_{-h/2}^{h/2} \sigma_{xx} d\,z = 0$$

$$\int_{-h/2}^{h/2} \sigma_{xx} z' d\,z = 0$$

where h is the thickness of the plate and z' the local coordinate. The solution $\boldsymbol{\sigma}^*$ can be written as

$$\boldsymbol{\sigma}^* = \boldsymbol{\sigma}_0 + \Delta\boldsymbol{\sigma} \tag{8.2}$$

with $\Delta\boldsymbol{\sigma} \neq -\boldsymbol{\sigma}_0$ to ensure a nontrivial solution. Substitution of the above expression into Eq. (8.1) gives

$$\int_{\Omega} \boldsymbol{B}^T (\boldsymbol{\sigma}_0 + \Delta\boldsymbol{\sigma})\, d\,\Omega = 0 \tag{8.3}$$

or

$$\int_{\Omega} \boldsymbol{B}^T \Delta\boldsymbol{\sigma}\, d\,\Omega = -\int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}_0\, d\,\Omega \tag{8.4}$$

For small deformation, the matrix $\boldsymbol{B}$ is constant and thus

$$\int_{\Omega_0} \boldsymbol{B}^T \Delta\boldsymbol{\sigma}\, d\,\Omega = \boldsymbol{F} \tag{8.5}$$

in which $\boldsymbol{F} = -\int_{\Omega_0} \boldsymbol{B}^T \boldsymbol{\sigma}_0\, d\,\Omega$. This means that the proposed method may also be considered as an equivalent loading scheme.

In either small or large strain case, the standard Newton-Raphson solution approach is employed to solve equation (8.4) or (8.5), with emphasis that the exact linearisation for nonlinear material models/constitutive laws and/or geometric nonlinearity should be employed to ensure the asymptotically quadratic convergence of the Newton-Raphson algorithm.

In some cases the Newton-Raphson iteration may encounter convergence problems, for instance, if the geometry of the workpiece to be formed is very complex, or if the workpiece may be very flexible and/or may undergo large deformation after peen forming. In particular, when geometric nonlinearity is considered, the solution iterations may converge slowly or even diverge.

There are two methods to achieve better convergence of the Newton-Raphson algorithm. The first approach is to initially impose extra constraint conditions, as similar to the problem considered in explicit analysis as possible, so that the workpiece being subjected to initial stress/strain is nearly in equilibrium and thus an equilibrium iteration can be achieved easily. The extra constraints will then be gradually released in an incremented manner so as to ensure the convergence of the Newton-Raphson algorithm.

The second option is to apply the initial residual stress incrementally. By introducing a pseudo-loading factor, λ, the initial stress to be applied is expressed by

$$\boldsymbol{\sigma}_0(\lambda) = \lambda\bar{\boldsymbol{\sigma}} \qquad \lambda \in [0,1] \tag{8.6}$$

In small deformation situations, suppose that at the $(n-1)$th increment, the initial stress $\boldsymbol{\sigma}_0^{n-1} = \lambda_{n-1}\bar{\boldsymbol{\sigma}}$ is applied and then a new equilibrium is established, i.e. the re-distributed stress $\boldsymbol{\sigma}_{n-1}^* = \boldsymbol{\sigma}_0^{n-1} + \Delta\boldsymbol{\sigma}_{n-1}$ satisfies

$$\int_{\Omega_0} \boldsymbol{B}^T \boldsymbol{\sigma}_{n-1}^* \, d\Omega = 0 \tag{8.7}$$

At the nth increment, the initial stress to be applied will be

$$\boldsymbol{\sigma}_0^n = \lambda_n\bar{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_0^{n-1} + \Delta\lambda_n\bar{\boldsymbol{\sigma}}$$

where $\Delta\lambda_n = \lambda_n - \lambda_{n-1}$. Further assume that the solution to be sought is $\boldsymbol{\sigma}_n^*$ which can be expressed as

$$\boldsymbol{\sigma}_n^* = \boldsymbol{\sigma}_0^n + \Delta\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_0^{n-1} + \Delta\lambda_n\bar{\boldsymbol{\sigma}} + \Delta\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_{n-1}^* + \Delta\lambda_n\bar{\boldsymbol{\sigma}} + \Delta(\Delta\boldsymbol{\sigma})_n$$

where $\Delta(\Delta\boldsymbol{\sigma})_n = \Delta\boldsymbol{\sigma}_n - \Delta\boldsymbol{\sigma}_{n-1}$. Then it follows from Eq. (8.7) that

$$\int_{\Omega_0} \boldsymbol{B}^T \boldsymbol{\sigma}_n^* \, d\Omega = \int_{\Omega_0} \boldsymbol{B}^T (\Delta(\Delta\boldsymbol{\sigma})_n + \Delta\lambda_n\bar{\boldsymbol{\sigma}}) \, d\Omega = 0 \tag{8.8}$$

i.e.

$$\int_{\Omega_0} \boldsymbol{B}^T \Delta(\Delta\boldsymbol{\sigma})_n \, d\Omega = -\int_{\Omega_0} \boldsymbol{B}^T \Delta\lambda_n\bar{\boldsymbol{\sigma}} \, d\Omega = \Delta\lambda_n\boldsymbol{F} \tag{8.9}$$

Clearly, this equation can be solved more efficiently than the original equation (8.5) due to the reduced degree of applied equivalent loading and thus an improved convergence behaviour can be achieved by the proper control of $\Delta\lambda$ at each step.

Compared to the first measure, the second one takes advantage of any existing advanced loading scheme in the FE procedure. In addition, users have more control over the behaviour of the solution procedure, and is therefore more preferable for practical use.

8.5.3 Generalised implementation

The above discussion assumes that a residual profile under a particular peening condition is applied to the whole workpiece. However, the implicit analysis has been implemented in such a way that much more complex situations can be simulated. Basically, the simulation can be performed by applying different profiles under different peening parameters to different parts of the workpiece at the same or different time steps. Furthermore, even different boundary conditions can be introduced at each time step. With this generalised implementation, many practical peen forming problems may be successfully simulated.

8.6 Numerical Example

The objective of the numerical example is to demonstrate the ability of the proposed solution strategy to simulate the saturation curve of forming a test strip.

The example is based on the experimental data of Airbus UK for peen forming a test strip of $150 \times 50 \times 4$ mm [7]. The material of the strip is aluminium alloy Al 7050 T7651 with Young's modulus 72 GPa, yield stress 450 MPa and a linear strain hardening rate of 120 MPa/unit strain.

The shot size used is S550 with an average diameter of 1.4 mm. Two sets of air pressure are applied: 35 PSI and 55 PSI, which (through use of empirical relations) correspond to impact velocities of 36 m/s and 46.5 m/s respectively. The impingement angle is 90°. The maximum deflections along the 150 mm edge for the two impact velocities are measured to be 2.1 mm and 3 mm respectively.

8.6.1 The first stage analysis: residual profile

To identify a residual profile in the first explicit analysis stage, a sample model of $8 \times 8 \times 4$ mm, with the same material properties as the strip, is established for shot peening as shown in Fig. 8.3(a), where the impact area of $5.6 \times 5.6 \times 4$ mm is discretised into a fine mesh (average element size $0.05 \times 0.05 \times 0.05$ mm). The resulting structured FE model consists of 128,000 8-noded solid elements with 138,737 nodes. Fig. 8.3(b) shows the zoomed FE mesh in which different element sizes along the thickness and within the impacted region and beyond are clearly illustrated. Around 600 MB of memory is required for the explicit solution phase.

The following boundary conditions are imposed: the bottom surface is fixed in all three directions; the 4 side surfaces are fixed in their normal directions; the top surface is free.

As illustrated in Fig. 8.3(a), the upper surface of the sample problem is repeatedly impacted by a set of $16(= 4 \times 4)$ regularly packed shot which covers an area of 5.6×5.6 mm. The

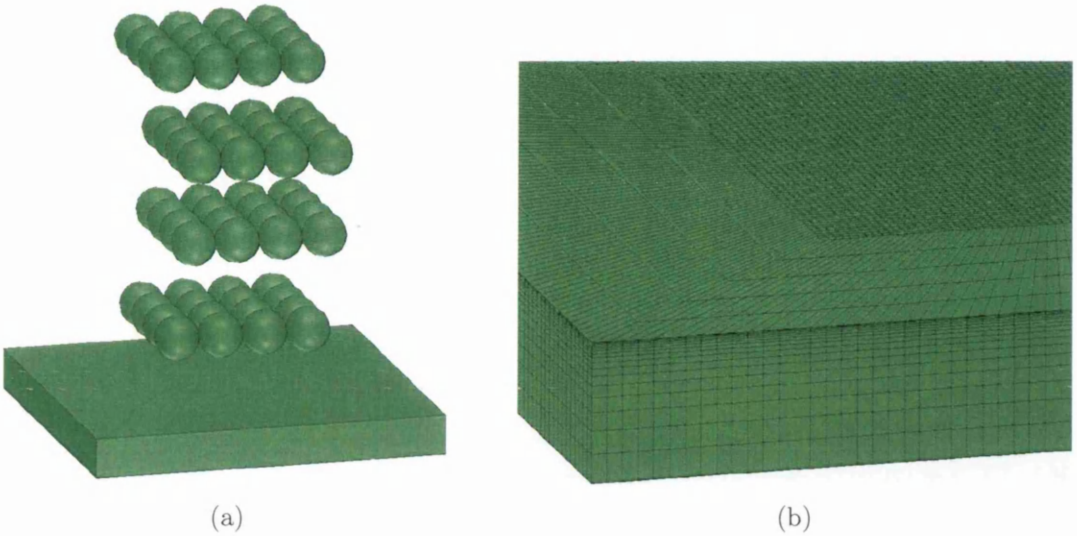


Figure 8.3: Sample model: (a) multiple shot and pass impact; (b) zoom of the FE mesh

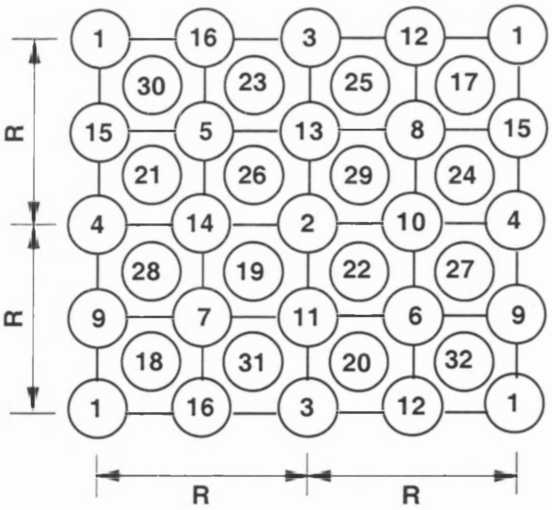


Figure 8.4: Layout of the impact position for each pass

impact positions of 32 passes are carefully offset to one another, as shown in Fig. 8.4, so that the best possible coverage can be achieved after each pass.

The residual stress, strain and plastic strain distributions after each multiple impact are obtained by averaging the contributions from all the elements in the impact region, as mentioned earlier, when the dynamic system reaches a steady-state condition and before the next impact starts. Figs. 8.5(a)~(f) respectively depict the residual stress $\bar{\sigma}_{xx}$, strain $\bar{\epsilon}_{xx}$, and effective plastic strain $\bar{\epsilon}_p$ profiles through the thickness for the selected passes with the two impact velocities. It is evident that the values at the surface and the maximum magnitudes, as well as the depth of compressive stress, increase with the number of passes and gradually converge. It also demonstrates that, with the same shot diameter, a larger impact velocity may result in a larger residual compressive stress distribution and a deeper compressive layer. However the maximum magnitudes of residual compressive stress for different impact velocities becomes almost the same when the peening process tends towards saturation.

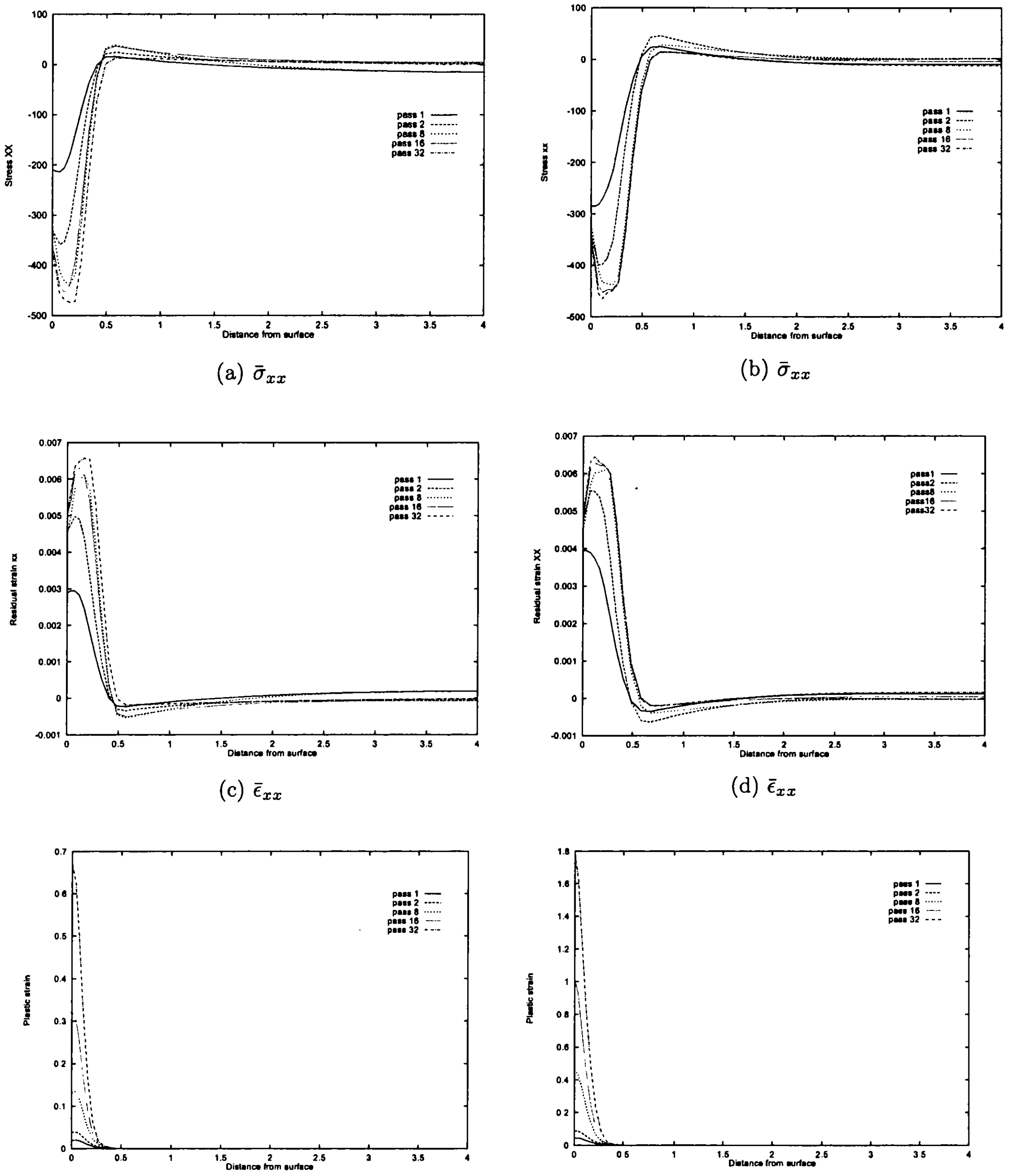
8.6.2 The second stage analysis: deformation and stress re-distribution

In the second analysis stage, an implicit FE model of the entire strip is developed. The FE mesh consists of only 4500 8-noded solid elements with 5456 nodes as a result of the much larger element size (5 × 5 mm) used in the horizontal directions, although the same number of element layers as the sample problem is employed along the thickness. The total memory requirement is less than 300 MB. The minimum constraints are applied to restrain rigid body movements but allow the strip to deform freely. The procedure described in the previous section is then applied to the strip.

Fig. 8.6 shows the final deformation of the strip after 32 passes for the impact velocity 46.5 m/s, from which a curvature shape is clearly demonstrated. The maximum displacements in the z direction, or the so called arc height, after several selected passes for the two impact velocities are depicted in Figs. 8.7(a) and (b) respectively. It is evident from both figures that the arc height difference between pass 16 and pass 32 is less than 10%, indicating that a saturation has been established. These computed maximum arc heights (2,2 mm and 2.45 mm) are in reasonably good agreement with the experiment (2.1 mm and 3 mm).

The re-distributed stress profiles (σ_{xx}) for the two impact velocities are illustrated in Fig. 8.8. The main difference from Fig. 8.5 is the decrease of compressive stress near the peened surface. A compressive stress layer is also introduced near the unpeened surface, a benefit existing only in the peen forming process.

In addition, both small strain and large strain cases have been considered and no significant difference has been observed. It should also be noted that an incremental approach was



(a) $\bar{\sigma}_{xx}$

(b) $\bar{\sigma}_{xx}$

(c) $\bar{\epsilon}_{xx}$

(d) $\bar{\epsilon}_{xx}$

(e) $\bar{\epsilon}_p$

(f) $\bar{\epsilon}_p$

Figure 8.5: Residual profiles obtained from explicit simulations for two impact velocities: v = 36 m/s (left column) and v = 46.5 m/s (right column)

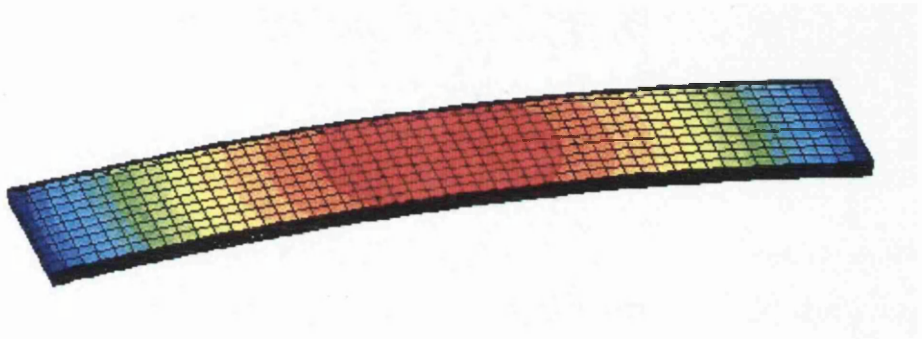
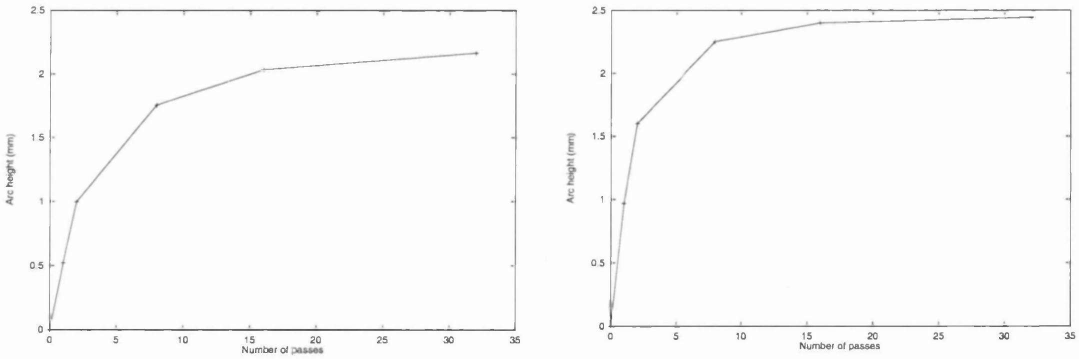


Figure 8.6: Final deformation with z-displacement contour after 32 passes: v=46.5m/s



(a)

(b)

Figure 8.7: Saturation curves: (a) v=36m/s; (b) v=46.5m/s

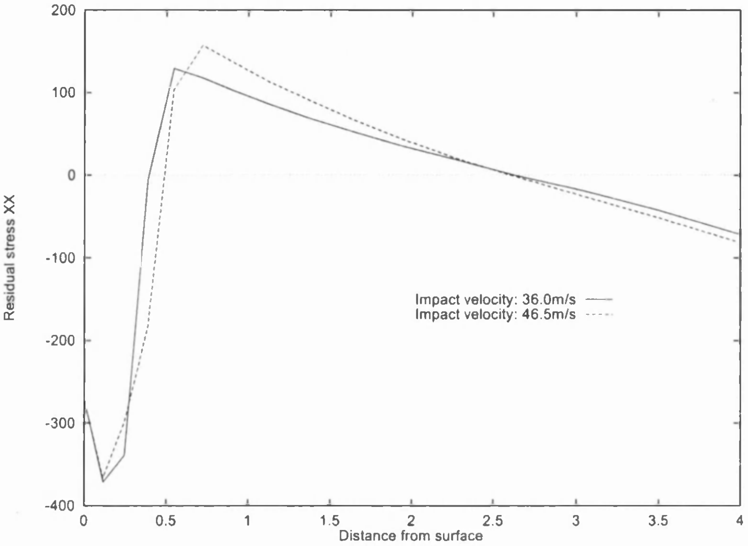


Figure 8.8: Re-distributed stress σ_{xx} for two impact velocities

not necessary for this example, and stress redistribution on a deformed configuration was achieved in a single increment.

8.7 Concluding Remarks

A two stage combined FE/DE and explicit/implicit solution procedure for the simulation of peen forming process has been developed, in which the residual profiles including stress, strain and effective plastic strain, are identified first by shot peening a small region of a workpiece with explicit dynamic analysis. The final deformation and stress distribution of the entire workpiece are then obtained by introducing these residual profiles as initial conditions in the subsequent implicit static solution. The applicability of the methodology developed has been illustrated through the peen forming of a test strip and the numerical results are in good agreement with the experimental results.

The direct benefit of the proposed strategy is that it results in FE/DE models of acceptable size which permits practical peen forming problems to be simulated within acceptable computer costs. Furthermore, the solution procedure does not rely on any empiricism with all input data being based on physically related quantities. Significant improvement in terms of computer CPU requirement can be achieved by applying parallelisation techniques to the simulation process by recognising the inherent concurrent nature of the computations involved in the explicit analysis in particular.

Bibliography

[1] *Shot peening applications.* Metal Improvement Company, Inc. (7th Edition).

[2] K. Han, D. Perić, A. J. L. Crook, and D. R. J. Owen. A combined finite/discrete element simulation of shot peening process. part I: studies on 2D interaction laws. *Eng. Comp.*, 17(5):593–619, 2000.

[3] K. Han, D. Perić, D. R. J. Owen and J. Yu. A combined finite/discrete element simulation of shot peening process. part II: 3D interaction laws. *Eng. Comp.*, 17(6/7):684-702, 2000.

[4] L. V. Grasty, and C. Andrew. Shot peen forming sheet metal: finite element prediction of deformed shape. *Proc. Instn. Mech. Eng.*, 210:361-366,1996.

[5] A. Levers, and A. Prior. Finite element analysis of shot peening. *J. Mater. Process. Technol.*, 80-81:304-308, 1999.

[6] D. Perić and D. R. J. Owen. Finite element applications to the nonlinear mechanics of solids. *Rep. Prog. Phys.*, 61:1495-1574, 1998.

[7] A. Levers. Private communication, 2000.

Chapter 9

Conclusions and Further Research

9.1 Conclusions

The present work has addressed several major algorithmic and modelling issues on the discrete element method, and in particular, established a general framework for the combined finite/discrete element simulation of shot peening and peen forming processes. The main contributions made are summarised as follows.

- On the basis of a general review of the two types of the commonly used contact detection algorithms (tree based and cell/grid based), and extensive numerical tests conducted, the dynamic cell ('D-cell') search algorithm has been established as the most efficient contact detection approach among the algorithms tested for large scale problems (over 1 million discrete objects). The parametric study of the 'D-cell' algorithm itself has suggested a (nearly) optimal cell size for general applications.

- A comprehensive discussion on the contact resolution for circular and spherical discrete objects has been presented. It has concluded that the normal interaction laws commonly employed in the DEM community can be unified under a general power law form, and the linear and Hertz interaction laws are normally sufficient for most applications.

- For the purpose of modelling contact between discrete and finite elements, several disk/segement and sphere/facet contact models, which can account for partial and corner contact situations, have been developed. They have been successfully applied to the subsequent combined FE/DE simulation of shot peening processes.

- A modified classic Coulomb friction model for sphere contact has been proposed, in which the analogue of the classic friction law with perfectly elasto-plastic theory is applied. The model has been extensively used subsequently and behaved very robustly.

- The stability issue of the central difference time integration scheme has been investigated, which not only confirms a previous research that the critical time step commonly used may lead to unstable solutions, but also reveals some new features. A new time stepping scheme is therefore developed, which can ensure both short and long term stability of the contact models.

- An efficient and robust approach has been proposed for contact resolution of 2D superquadrics, which is accomplished by a novel combination of several advanced numerical techniques, including firstly representing any superquadric with a convex polygon through adaptive sampling; then clipping two polygons to identify the overlap zone; and employing an efficient linear algorithm to search for intersections and overlap area of the clipped polygons; finally utilising an established polygon/polygon contact framework to determine the contact forces and directions.

- Significant effort has been made in the development of effective numerical procedures for generating an initial packing of different sizes of discrete objects within different shapes of geometric domains. In particular, an advancing front technique based approach is described which can efficiently generate a random packing of disks, ellipses and polygons; whilst a geometric based compression algorithm is developed to generate a random packing of spheres.

- A general algorithmic framework has been established for the combined finite/discrete element analysis of shot peening processes, particularly for the evaluation and comparison of several interaction laws governing the contact between shot and workpiece.It is found, via numerical experiments, that the representation of shot as 2D disks, and workpiece as 2D plane strain finite element problem is not fully valid for the simulation of a shot peening process. The resulting tensile stress on the shot peened surface suggests that the 2D model cannot capture the major fundamental physics and characteristics of the shot peening process, therefore a full 3D simulation is in general required.

- The proposed 3D representation of the problem is carefully examined. Numerical tests are performed to evaluate the normal and frictional contact interaction laws with various model parameters. Effects of single and multiple shot impact, as well as element sizes are also numerically investigated. The results indicate that the magnitude and distribution of residual stresses resulting from single shot impact are different from those from multiple shot impact as occur in practical industrial applications, thus multiple shot impact simulations should be employed to provide acceptable residual profiles. Numerical experiments also suggest that the element size in the impact area should be no larger than one tenth of the shot diameter to guarantee sufficient solution accuracy of the explicit procedure.

- The proposed FE/DE model has been verified by experiments conducted elsewhere on three important commercial alloys. The calculated maximum compressive stress agrees well with the measured value as does the depth beneath the surface which has compressive residual stresses.

- A two stage combined finite/discrete element and explicit/implicit solution procedure for the simulation of peen forming process has been developed, in which the residual profiles including stress, strain and effective plastic strain, are identified first by shot peening a small region of a workpiece with explicit dynamic analysis. The final deformation and stress distribution of the entire workpiece are then obtained by introducing these residual profiles as initial conditions in the subsequent implicit static solution. The applicability of the methodology developed has been illustrated through the peen forming of a test strip and the numerical results are in good agreement with the experimental results. The direct benefit of the proposed strategy is that it results in finite/discrete element models of acceptable size that permits practical peen forming problems to be simulated within acceptable computer costs. Furthermore, the solution procedure does not rely on any empiricism with all input data being based on physically related quantities. Significant improvement in terms of computer CPU requirement can be achieved by applying parallelisation techniques to the simulation process by recognising the inherent concurrent nature of the computations involved in the explicit analysis in particular.

9.2 Recommendations for Further Research

Based on the experience gained in this research, the following are identified as topics which may merit further research.

9.2.1 Modelling of friction

In many practical applications, correct modelling of friction is the key for capturing the actual physical phenomena involved. In particular, it has been increasingly recognised that rolling friction plays a fundamental role in the behaviour of particle systems which are represented by circular disks/spheres. Although the mechanism associated with the rolling resistance of disks has been briefly discussed, no rolling friction model has been developed in this work. Further research on this topic is worthwhile.

The proposed 3D friction model for sphere contact is reliable as has been proved, but the associated computational cost is fairly expensive. It is highly desirable if some improvement can be made to the model or a more efficient model can be developed.

9.2.2 Contact resolution for polyhedrons and 3D superquadrics

A substantial effort has been made in this work for the development of an efficient and robust contact model for polygons and 2D superquadrics. It is however still the case that contact resolution for non-spherical 3D objects, such as polyhedrons and 3D superquadrics, is a very expensive operation. Although it is possible to extend the methodology for dealing with 2D superquadrics to 3D superquadrics by tessellating the surfaces to triangular facets, more effective strategies are required to be developed.

9.2.3 Packing of irregular shaped discrete objects

Packing of different shaped discrete objects has received particular attention in this work. It is still a challenge to develop a sufficiently efficient packing procedure for irregular shaped objects. This may become increasingly important when more complex discrete element shapes are introduced in the DEM.

9.2.4 Alternative solution strategies

The current employed solution procedures for the dynamic equations of discrete element systems are almost all confined to the explicit time stepping algorithms. The time step constraint imposed to the algorithms leads to a large number of time increments necessary for the simulation of even a very short period of real time (seconds). In applications such as powder compaction, the system is mainly of a quasi-static nature. Even in a truly dynamic situation, it is often the case that certain regions/parts of the system may behave quasi-statically within a certain time interval. For a quasi-static system, implicit solution procedures may be more efficient since a much larger time step can be applied, though a (non-)linear system of equations needs to be solved at each time step. Therefore it may worth exploiting the possibility of a combined, but intelligent, explict/implicit solution strategy for discrete systems.

9.2.5 Parallelisation

Besides the large number of time increments associated with the explicit method as mentioned above, the number of discrete elements required to realistically represent a practical problem may be very large or even beyond the limitation of currently available computer resources. Therefore the computational cost involved is extremely intensive. The most feasible solution to the problem is to increase the available computational capacity by introducing parallel computing. Significant improvement in terms of CPU requirement can be achieved by applying parallelisation techniques to the simulation process by recognising the inherent concurrent nature of the computations involved in the explicit analysis in particular.