

Imperial College London
Department of Earth Science and Engineering

Impulse-based discrete element modelling of rock impact and fragmentation, with applications to block cave mining

Xuhai Tang

December 2013

Supervised by

Dr. Adriana Paluszny and Prof. Robert W. Zimmerman

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Earth Science and Engineering of Imperial College London
and the Diploma of Imperial College London

Declaration

I hereby declare that all work presented in this thesis is my own work, and all else is appropriately referenced.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgements

Completing a PhD is an experience of a lifetime. I would not have been able to accomplish this great challenge without the support of a range of sources. First of all, I would like to thank two my supervisors, Dr. Paluszny Adriana and Professor Robert Zimmerman.

I would like to give a heartfelt, special thanks to Professor Robert Zimmerman, who provides me unflagging support and advice. He is a good teacher, who always encouraged and expected us to think more independently about our research, provide insightful discussions about my project. He works hard with responsibility and his whole heart. His enthusiasm for research is contagious and taught me a lot of things.

Special thanks to Dr. Paluszny Adriana. Her flexibility in scheduling, gentle encouragement and relaxed demeanor made for a good working relationship and the impetus for me to finish. Dr. Paluszny is a responsible and patient supervisor. During the past three years and a half, Dr. Paluszny spent a lot of time to improve my presentations, publications and thesis. She always reads my publications carefully, discusses with me and give me helpful advice for research sentence by sentence.

Next, I would also like to thank my colleagues Morteza Nejati, Philipp Lang, Jasmin Ambrose, and Dr Gaurav Singh. I also have to thank the members of my PhD committee, Professor Munjiza and Dr. Gorman for their helpful advice.

Finally, I'd like to thank the British-Australian multinational metals and mining corporation, Rio Tinto, for supporting my research.

Abstract

Impulse-based methods efficiently and accurately model high-frequency collisions of complex shapes based on the enforcement of non-penetrating constraints. It does not rely on penalty parameters nor requires the computation of penetration between bodies. This work presents a novel necessary condition for energy conservation in impulse-based methods. In previous versions of the impulse methods, such as sequential and simultaneous impulse methods, the relative velocity at the contact points after collision is directly derived from the relative velocity before collision, in a purely simultaneous or sequential manner. This work presents a novel energy tracking method (ETM), in which the relative velocities are iteratively but gradually adjusted, simultaneously modelling their interaction at each iteration. ETM ensures the energy conservation while capturing the propagation of forces during collision. The ETM is applied to model the dynamics of fragment collision in the context of fragmentation. Two approaches of fragmentation are proposed: a finite-discrete element approach, and a low cost, fragmentation pattern-based approach. The first approach models the growth of fractures using the finite element method (FEM) and advanced re-meshing technology. This finite-discrete element approach suffers from the drawback of massive computational cost. The low-cost, fragmentation pattern-based approach separate colliding bodies directly. The fragmentation pattern is generated using Weibull distribution equations, the patterns and size distributions computed using full finite/discrete element simulations and experimental results. This work investigates the influence of fragmentation on the frequency of hang-up events and on the gravity flow of rock fragments within a block caving system. Numerical results indicate that models that do not consider fragmentation tend to overestimate the frequency of hang-up accidents.

Contents

1	Introduction	11
1.1	Simulation of fragmentation	11
1.2	Block caving system	17
1.3	Main contributions and publications	18
1.4	Thesis organisation	19
2	Energy conservative impulse-based discrete element method	23
2.1	Introduction	23
2.2	Impulse Based Method	26
2.2.1	Single-point collision: Contact model	26
2.2.2	Single-point collision: Impulse and work	30
2.2.3	Single-point collision: Friction	31
2.2.4	Multiple Collisions: Sequential impulse method	32
2.3	Numerical Tests	33
2.3.1	Collision of two tetrahedra	34
2.3.2	Sphere falling down a slope	35
2.3.3	Draw point	35
2.4	Conclusions	37
3	Energy tracking method for collision response	39
3.1	Introduction	39
3.2	Generalized form of the impulse based method	40
3.3	Single-point collision: Impulse and work	42
3.4	Single-point collision: Friction	45
3.5	Energy Tracking: Single Collision	47
3.6	Energy tracking method for multiple collisions	49
3.7	Parallel computing in ETM	51
3.8	Numerical tests	53
3.8.1	Single-point collision test	57
3.8.2	Energy conservation	59
3.8.3	Comparison with SMM	59
3.8.4	Comparison with SQM	62
3.8.5	Packing and repositioning of multiple bodies	62

3.9	Conclusions	63
4	Numerical modelling of fragmentation	71
4.1	Introduction	71
4.2	Fragmentation simulation workflow	75
4.3	Geometric representation	76
4.4	Deformation	78
4.5	Fracture propagation	79
4.5.1	Stress intensity factor computation	80
4.5.2	Failure criterion	80
4.5.3	Propagation magnitude	80
4.5.4	Propagation direction	81
4.5.5	Dynamics	81
4.6	Case study: Impact-Induced Fragmentation	82
4.7	Conclusions	83
5	Low-cost Weibull distribution based fragmentation pattern	89
5.1	Introduction	89
5.2	Fragmentation pattern	90
5.3	Methodology: Fragmentation based on the Weibull distribution	91
5.3.1	Local coordinates	93
5.3.2	Cutting boxes	96
5.3.3	Measure performance of fragmentation pattern using Weibull distribution	97
5.3.4	Determine parameters of fragmentation pattern using a table	102
5.4	Case study: Fragmentation of a single sphere	103
5.5	Case study: Fragmentation of a realistically shaped rock	104
5.6	Case study: Fragmentation of 5×5×5 spheres	110
5.7	Conclusions and future work	111
6	Block caving case study on the influence of fragmentation on draw-point hang-up	123
6.1	Introduction	123
6.2	Case study: Hang-up of initially spherical rock fragments	127
6.3	Case study: Hang-up of initially cuboid fragments	132
6.3.1	DPC1 and DPC1f	134
6.3.2	DPC2 and DPC2f	140
6.3.3	DPC3 and DPC3f	146
6.3.4	Surface subsidence associated with block caving mining	146
6.4	Conclusions	152

7	Conclusions and recommendations for future research	155
7.1	Conclusion	155
7.1.1	Modelling particle collisions	155
7.1.2	Simulation of fragmentation	156
7.1.3	Influence of secondary fragmentation in block caving	156
7.2	Recommendations for future work	157
7.2.1	Improve the low-cost Weibull distribution based fragmentation pattern . .	157
7.2.2	Reproduce the realistic shape of ore fragments in block caving	158

List of Tables

2.1	Energy and linear momentum for (a) a moving tetrahedron collides with a static one, (b) two colliding tetrahedra	34
5.1	The relation between impact velocities and the parameters of fragmentation pattern	103
6.1	Parameters for the hang-up modelling of spherical ore fragments.	132
6.2	Surface subsidence above the centre of the undercut.	146

List of Figures

1.1	Simplified block caving system.	22
2.1	Estimation for the time of contact. Each circle represents a moving object approaching the ground at consecutive time steps. A collision is detected between time steps t_i and t_{i+1} . Collision time is estimated as t_c	26
2.2	Collision between bodies b_N and b_M at collision c_i : $k=0$ or 1 , s_0 and s_1 indicate the status before and after applying impulse.	27
2.3	A local coordinate system $(\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i)$ is defined at the contact point between bodies b_M and b_N	29
2.4	Collision of multiple bodies: multiple bodies, $\mathbf{b} = \{b_0, b_1, b_2, \dots, b_m\}$, collide with multiple contact points $\mathbf{c} = \{c_0, c_1, c_2, \dots, c_n\}$	32
2.5	Geometric representation of bodies.	34
2.6	Moving tetrahedra colliding with: (a) one static tetrahedron and (b) a moving tetrahedron, and (c) images of a spherical object falling down a slope.	35
2.7	(a) Details of the slope test. (b) Energy change of a sphere bouncing down a slope.	36
2.8	(a) Details of the single draw point test. (b) Energy curve for the system of fragments falling through a cylindrical draw point with different friction and restitution coefficients. (c) Relative error of total energy for the case of perfectly elastic collisions.	37
2.9	Collision of $7 \times 7 \times 10 = 490$ ore fragments, around a single draw point.	38
3.1	Collision between bodies b_N and b_M at iteration s_k	40
3.2	A local coordinate system $(\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i)$ is defined at the contact point between bodies b_M and b_N	41
3.3	The change of energy and relative normal velocity during collision resolution for a single collision.	47
3.4	Collision of multiple bodies: multiple bodies, $\mathbf{b} = \{b_0, b_1, b_2, \dots, b_m\}$, collide with multiple contact points $\mathbf{c} = \{c_0, c_1, c_2, \dots, c_n\}$	50
3.5	Three dimensional computational domain division for parallel computing.	53
3.6	The computational domain of packing and repositioning of concave concrete crosses is divided into four parallel domains and one overlap domain.	54
3.7	Compare the computational costs of serial computing and parallel computing for the packing and repositioning of concave concrete crosses.	55

3.8	The computational domain for the gravity flow of ore fragments in block caving system is divided into four parallel domains and one overlap domain.	56
3.9	Collision of two tetrahedra.	58
3.10	Progression of relative normal velocity, and the elastic, kinetic and total energy of a system of two colliding tetrahedra.	58
3.11	Comparison of SMM and ETM: Collision of a moving body and a static body: Multiple contact points.	60
3.12	Comparison of SMM and ETM: Angular velocities of a moving body colliding with ground.	60
3.13	Relative error of energy during the packing of (a) 540 and (b) 1080 bodies. . . .	61
3.14	Comparison of SQM and ETM: Collision of a moving body and a static body with multiple contact points.	64
3.15	Comparison of SQM and ETM: Angular velocities of moving body colliding with ground.	64
3.16	Geometry of colliding bodies: (a) sphere, where $a=3$ m, (b) cube, where $b=2.4$ m, and (c) cross, where $3c=3.78$ m.	65
3.17	Packing of multiple bodies with different shapes.	65
3.18	Settling of multiple bodies with different shapes, $\mu = 0.64$ and $e_n = 0.64$	66
3.19	Repositioning of multiple bodies with frictions ranging from 0.05 up to 0.6. . . .	67
3.20	Eight lines are fitted through the ensuing dataset to estimate the repose angle. . .	68
3.21	Friction dependent angle of repose. (a) Variation for each shape as compared to experimental and numerical results reported in the literature, (b) measurement error for each case.	69
4.1	The fracture and fragmentation cycle combines the computation for fracturing and the computation for multi-body interaction.	76
4.2	Geometric data structure of bodies.	77
4.3	The representation of fractures during crack growth.	78
4.4	Fragmentation driven by fracture growth.	79
4.5	Mean stress state contours of sphere at impact.	85
4.6	Fragment shapes for (a) case s.h.2 - low, (b) case s.h.3 - mid, (c) case s.vh.4 - high velocity impact.	86
4.7	Fragment mass distribution.	87
4.8	Cumulative fragment mass distribution.	87
4.9	Collection of fragment shapes for s.h.2, s.h.3, and s.vh.4.	88
5.1	A local coordinate system is defined by the primary contact direction.	94
5.2	Cutting boxes define the overall fragmentation of the object.	97
5.3	Control point for Voroni tessellation.	99
5.4	Voronoi cells when impact velocity is 20 m/s.	100

5.5	Algorithm to find the relation between a special value of v and the parameters of fragmentation pattern, a , b and c .	101
5.6	Cheong's experimental results define the relation between impact velocity and the two parameters of Weibull distribution.	102
5.7	A sphere falls downwards, collides with a rigid ground at with impact velocity 26 m/s, and fragments.	104
5.8	Shape of fragments achieved from the fragmentation of a single sphere, with impact velocity 20 m/s.	105
5.9	Shape of fragments achieved from the fragmentation of a single sphere, with impact velocity 26 m/s.	106
5.10	Volume-based cumulative size distribution resulting from the fragmentation of a single sphere is fitted by two-parameter Weibull distribution equation.	107
5.11	Geometry of a realistic shaped rock.	108
5.12	Shape of fragments resulting from the fragmentation of the realistically shaped rock at impact velocity 26 m/s.	109
5.13	The Weibull distribution is employed to fit volume-based cumulative size distribution of fragments resulting from the fragmentation of realistically shaped rock.	113
5.14	The realistically shaped rock is fragmented into a set of small fragments.	114
5.15	Zoom in on the fragmentation process of a realistically shaped rock.	114
5.16	Geometry of container.	115
5.17	The snapshots for DC1f.	116
5.18	The snapshots for DC2f.	117
5.19	The snapshots DC1f, where the fragments are coloured according to their volume.	118
5.20	The snapshots DC2f, where the fragments are coloured according to their volume.	119
5.21	Size distribution of DC1f ($v=10$ m/s) and DC2f($v=100$ m/s).	120
5.22	The Weibull distribution is used to fit the size distribution of DC1f and DC2f.	120
5.23	The relation among total computation time, computation time of fragmentation, and number of contacts at each simulation step for DC1f and DC2f.	121
5.24	The average computational time per contact for DC1f and DC2f.	122
6.1	Conceptual model of a self-sustained propagating cave	125
6.2	The geometry of a draw point for the hang-up modelling of spherical ore fragments.	127
6.3	DPST1 at step 0, 310, and 4000.	128
6.4	DPST1f at step 0, 310, and 4000.	128
6.5	Fragmentation simulation of one ore fragment in DPST1. After collision, the initial rock is fragmented as a function of speed and contact point location, and its effect on fracture growth.	128
6.6	Percentage of volume of extracted ore as a function of time, as predicted by simulations with and without fragmentation ($r = 5.77$ m).	129

6.7	DPST2 at step 0, 2210 and 4000.	130
6.8	DPST2 at step 0, 2210 and 4000.	130
6.9	DPST2f at step 0, 2210 and 4000.	130
6.10	Percentage of volume of extracted ore as a function of time as predicted by simulations with and without fragmentation ($r = 0.75m$).	131
6.11	Geometry specification of a single draw point.	131
6.12	Different size distribution of cuboid rock fragments. The boundary wall is dis- cretized into 1252 elements.	133
6.13	Front view of DPC1, where no fragmentation happens.	135
6.14	Sectional view of DPC1, where no fragmentation occurs.	136
6.15	Front view of DPC1f, where fragmentation happens.	137
6.16	Sectional view of DPC1f, where fragmentation happens.	138
6.17	The volume of extracted ore fragments in DPC1, the volume of extracted ore fragments and damaged ore fragments in DPC1f.	139
6.18	Front view of DPC2, where no fragmentation happens.	141
6.19	Sectional view of DPC2, where no fragmentation happens.	142
6.20	Front view of DPC2f, where fragmentation happens.	143
6.21	Sectional view of DPC2f with fragmentation.	144
6.22	The volume of extracted ore fragments in DPC2, the volume of extracted ore fragments and crushed ore fragments DPC2f.	145
6.23	Front view of DPC3, where no fragmentation happens.	147
6.24	Sectional view of DPC3.	148
6.25	Front view of DPC1f, where fragmentation happens.	149
6.26	Sectional view of DPC3f, where fragmentation happens.	150
6.27	The volume of extracted ore fragments in DPC3, and the volume of extracted ore fragments and crushed ore fragments in DPC3f.	151
6.28	Surface subsidence of three cases.	152

1 Introduction

Fragmentation results when colliding bodies break. Fragmentation is common in industrial processes, such as tumble mills in agriculture [Mishra, 2003], rock fragmentation in mining engineering [Vyazmensky, 2008, Munjiza and Latham, 2004], projectile collision in military engineering [Tanaka et al., 2002], and explosion in chemical engineering [Kaneko et al., 1999]. Engineers seek to enhance the comminution and fragmentation of raw ores, often for the purpose of increasing surface area and enhancing chemical reactions [Cheong et al., 2004], while sometimes aiming to avoid it, as unwanted fragmentation leads to the loss of particles and to the production of hazardous dust [Shipway and Hutchings, 1993]. During fragmentation, a part of the input energy is used to create new fracture surfaces, while the rest is wasted in the form of noise and heat [Fuerstenau and Abouzeid, 2002]. Laboratory experiments have been shown to succeed in investigating this complex process qualitatively, e.g. by measuring the relationship between size distribution and impact velocity [Wittel et al., 2008], but are also restricted in scale and measurability. The present work focuses on the investigation of fragmentation using numerical tools.

1.1 Simulation of fragmentation

Numerical methods provide means to investigate complex problem settings quantitatively, robustly and under conditions difficult to reproduce in a lab. Take an example, the dream of mining on the moon might become true in the future, however, it is difficult to simulate fragmentation under low gravity conditions. Challenges related to fragmentation modelling include accurate geometric representation of fragments and cracks, accurate mechanical computations of movement and energy transfer, efficient processing of large volumes of data, incorporating dynamically defined boundary conditions, handling the collisions among multiple bodies accurately, and most importantly reducing the computational cost to model the fragmentation of hundreds and thousands of bodies. A brief review of the main numerical methods available to address the issues of collision dynamics, fragmentation and fracture propagation follows. Three main approaches for the simulation the collisions among multiple bodies dominate the literature: the penalty method, discontinuous deformation analysis, and impulse-based methods.

The penalty method, which employs a spring-damper model, transforms constrained optimisation problems into a set of non-constrained optimisation problems by iteratively searching a proper penalty parameter [Babuška, 1973]. The penalty method is widely applied to deal with contact problems due to its simplicity [Bourago and Kukudzhinov, 2005, Hasegawa and Fujii,

2003b]. Numerical treatment with penalty parameters has been used to solve contact problems: the formulation of the geometry, the statement of interface laws, the variational formulation and the development of algorithms were considered [Wriggers, 1996]. The main disadvantage of the penalty method is that it relies in a somewhat arbitrary penalty parameter to describe the relation between geometric body penetration and repulsion force. Particularly for contact detection algorithms, the penalty method needs to be defined locally, as the shapes of the bodies tend to be complex, and penalties have to be evaluated with a local perspective to achieve convergence.

Discontinuous Deformation Analysis (DDA), originally proposed by Shi and Goodman [1984], is an extension of the penalty-based discrete element method (DEM) [Cundall and Strack, 1979]. DDA solves the stress-displacement of bodies similarly to FEM; however, it is focussed on solving the interaction of discrete blocks along discontinuities for problems of fractured and jointed rock masses. DDA is also an implicit discrete element model [Jing, 1998]. Instead of considering the contact response separately, a stiffness matrix is implemented based on minimizing the potential energy. The strain energy from the contact force is represented by a function of penetration [Shi and Goodman, 1984]. The advantages of DDA include that the equilibrium of dynamics is always satisfied; stability can be ensured without any extra damping parameter; and penetration and traction are avoided at each step. The disadvantages are the expensive requirements for computational time and memory. Some extensions of DDA are: DDA with second-order displacement functions [Grayeli and Mortazavi, 2006], and its application to solve dynamics problems [Tsesarsky et al., 2005].

Impulse-based method, which employs impulse to avoid penetration between two colliding bodies, was proposed by [Mirtich and Canny, 1994, 1995]. The impulse-based method has the ability to capture models with frequent collisions and contact modes with frequent changes [Mirtich and Canny, 1994]. Unlike FEM-based methods, which compute explicit constraint forces at contact points, the impulse method captures collision impulses between bodies by solving the Linear Complementarity Problem [Baraff, 1994]. The impulse method is suited for non-linked rigid body environments as it was demonstrated in a multi-body simulator for haptic display [Chang and Colgate, 1997]. Its main advantage is that it does not require full geometric intersection checks during the dynamics simulation, and is therefore computationally less costly than the penalty-based alternative. Thus, it requires less iterations to compute the contact response at single contact points. With the impulse method, all types of contact including sliding and rolling can be computed using a single model. Additionally, in contrast to the penalty and Lagrange-based methods, it does not rely on arbitrary penalty parameters that define repulsion as a function of penetration but instead predicts collisions as a function of impact between proximal bodies.

Simulating the propagation of cracks which split colliding bodies is essential for the simulation of fragmentation. The finite element method (FEM) [Zienkiewicz and Taylor, 2000] is probably the most widely used computational method to resolve engineering problems, including

modelling fragmentation [Paluszny and Zimmerman, 2013, Paluszny et al., 2013]. However, the simulation of fracture propagation and fragmentation is limited by the technology of controlling geometry change of fractures and colliding fragments. Mesh re-generation related to evolving geometry is one of the main difficulties of modelling fracture propagation in 3D FEM. Thus, a myriad of approaches to avoid meshing and geometric handling have been developed over the past decades. In discrete element method (DEM) [Cundall and Strack, 1979] and combined finite-discrete method (FDEM) [Munjiza et al., 1995], the fractures propagate along the boundary of discrete elements, which avoids the mesh generation during simulation. An alternative approach is to simulate crack growth explicitly and use fractures to drive fragmentation. Crack representation is often defined in a sub-grid fashion, as an entity within the mesh to avoid mesh modifications and re-generation. Other methods, e.g. boundary element method, aim to reduce complexity by representing only the boundaries of the bodies in question: these are unable to capture heterogeneity in the matrix. Mesh-free methods bypass meshing completely and define domains as a set of points: these introduce difficulties such as domain interface blurring, and costly computational operations. An overview of numerical methods for crack growth and fragmentation follows.

The Discrete Element Method (DEM) [Cundall and Strack, 1979] is frequently used to simulate fragmentation, as it is able to jointly model handle collisions and fracturing. In DEM, the contacts are updated automatically during the calculation process, the discrete elements are allowed to express finite displacements and rotation [Cundall, 1988]. DEM was initially proposed to solve the collision behaviour of granular material [Cundall and Strack, 1979]. Then it was improved to simulate the multiple crack propagation within solid materials, such as intact rock [Hajiabdolmajid et al., 2002] and concrete [Camborde et al., 2000]. DEM has been successfully implemented to model the impact-induced fragmentation of rocks in 3D [Wang and Tonon, 2011] in the context of impact fragmentation for the simulation of fly-rock and the design of defence structures [Wang and Tonon, 2010]. Ghaboussi [1988] introduced the deformation of individual discrete element into numerically model processes, which is named hybrid FEM and DEM method (FEM-DEM). Recent work presents FEM-DEM applied to rock engineering [Elmo et al., 2013], including the modelling of surface subsidence associated with block caving mining [Vyazmensky et al., 2007], the simulation of dynamic 2D geo-mechanics problems [Onate and Rojek, 2004], and the simulation of rock-fill in 2D [Bagherzadeh et al., 2010]. The deformation of the individual blocks can be considered using the varying of equivalent distance between two discrete blocks [Wang and Tonon, 2011]. Alternatively, Ghaboussi [1988] considered the deformation of the individual blocks in two dimensions, in which the blocks are simulated as single quadrilateral element. DEM suffers from some drawbacks. For example, the simulation of fragmentation using DEM usually relies on debonding growth that follows the structure of the mesh and requires the calibration of micro- to macro-material properties.

In order to resolve problems involving discontinuum and continuum, Munjiza et al. [1995] proposed an alternative approach, combined finite-discrete method (FDEM), which inherits the

finite element-based analysis of continuum and the discrete element-based analysis of contact and collision interaction [Munjiza, 2004, Munjiza et al., 2011]. In the context of FDEM, a computational domain is discretized into a group of separate discrete elements which have separate finite meshes [Munjiza et al., 1999, Munjiza and John, 2002]. With the ability to model the transition from continuum to discontinuum conveniently, FDEM has been widely used to simulate the collision, fracturing and fragmentation, ranging from micro-scale experiments in laboratory to macro-scale engineering problems [Mahabadi et al., 2012]. In laboratory, FDEM was used to investigate the gravitational deposition of identical cubes [Munjiza and Latham, 2004] and track the fracture trajectories in Brazilian disc test [Mahabadi et al., 2009, Mahabadi and Grasselli, 2010]. In coastal engineering, FDEM was used to simulate the packing and repositioning of concrete armors Latham et al. [2008b,a], Munjiza et al. [2010]. Latham et al. [2008a] coupled FDEM with computational fluid mechanics to model massive particulates for breakwater engineering. In hydropower engineering, Tatone et al. [2010] used FDEM to assess the stability of gravity dam. Rockfall, that falls along a vertical or sub-vertical cliff, might cause disasters to pedestrians or buildings under the cliff. Therefore, Lisjak and his co-workers [Lisjak and Grasselli, 2010, Lisjak et al., 2010] used FDEM to understand the impact-induced fragmentation of a rockfall, which helps the design of defense constructions. In addition, the cliff recession was modelled using FDEM [Mahabadi et al., 2012]. FDEM also inherits the disadvantage of DEM, such as requiring the calibration of micro- to macro-material properties for elastic problem.

In the Boundary Element Method (BEM) [Banerjee and Butterfield, 1981, Wrobel and Aliabadi, 2002], only the domain boundary is discretised. This feature makes it attractive for fracture problems [Cruse, 1969]. However, BEM has several disadvantages. For example, it suffers from a lack of volumetric property distribution, since the discretisation is only implemented on the boundary. BEM is not well suited to handle non-functional material variations in the volume. Furthermore, the boundary integral equation degenerates when two surfaces are in close proximity [Cruse, 1972, Mukhopadhyay et al., 1998, 2000]. The dual boundary element method (DBEM) was extended to three dimensions by [Mi and Aliabadi, 1994]. DBEM, for crack growth in 3D, utilises special crack front elements which incorporate the displacement variation by modifying shape functions, a technique which allows continuous elements in the discretisation on crack surfaces [Wilde and Aliabadi, 1999]. BEM has also been applied to grow multiple fatigue cracks in 3D [Cisilino and Aliabadi, 1997]; specifically by using a multi-domain model to analyse fracture problems in non-symmetric media [Blandford et al., 1981]. Furthermore, the dual boundary element method and the time domain method were presented to simulate dynamic fracture problem [Fedelinski et al., 1995]. Although BEM can significantly reduce computational cost [Kolk et al., 2005], and stress intensity factor computation is accurate, the method lacks flexibility in terms of volumetric domain definition.

As an extension of finite element method, the smeared crack method [Jirasek, 1998] is implemented to simulate fracture propagation. In the smeared crack method, cracks are described by

multiple embedded displacement discontinuities with explicit representation [Jirasek and Zimmermann, 2001]. The main advantage of embedding smeared cracks within elements is to avoid the remeshing process [Jirasek and Zimmermann, 1998a,b]. The main disadvantage is that there is no stress singularity and stress intensity factors cannot be directly computed. Additionally, as the domain is not explicitly defined, it requires a refined mesh to be able to represent the emerging smeared crack. Alternation, Beissel et al. [1998] and Tang et al. [1998] extended the FEM to simulate fracture propagation via tagging and deleting broken finite elements. In their approaches, the stiffness and strength of elements are reduced to represent the fractures. Because the geometry of fractures is simplified and remeshing is avoided, their methods are able to handle multiple fracture propagation [Jia and Tang, 2008]. One main drawback of this method is that the stress singularity at the crack tips is not correctly represented and the criterion of fracture propagation is a function of local element strength [Tang et al., 1998]. As compared to the fracture criterion based on linear elastic fracture mechanics (LEFM), the fracture criterion based on stresses at the elements is strongly influenced by the amount and size of elements of the mesh. Furthermore, the mesh is required to be homogeneously refined throughout the simulation to capture the onset of fracturing. Difficulties related to the technical implementation of geometric evolution and automatic remeshing motivated the development of mesh-independent methods.

Fracture propagation has been simulated using enrichment functions using partition of unity methods [Griebel and Schweitzer, 2002]. The Extended Finite Element (XFEM) was first proposed by Belytschko and Black [1999] to overcome the meshing burden associated with FEM methods. In XFEM, fracture is described explicitly and a set of enrichment functions are added into the global approximation of conventional FEM to consider crack tips and crack boundaries [Moes et al., 1999, Fries and Baydoun, 2012, Zhu, 2012]. XFEM was then developed by Daux et al. [2000] to simulate arbitrary branched and intersecting cracks. More contributions include Sukumar et al. [2000] who improved the XFEM to model crack propagation in three dimensions. To consider more special problems, the enrichment function is improved in XFEM to study the presence of internal pressure inside cracks [Lecampion, 2009]. The problems related to contact and friction are also solved in the context of XFEM [Khoei and Nikbakht, 2006]. XFEM was extended to handle multiple crack growth in brittle materials [Budyn et al., 2004]. Although a mesh is not required in XFEM to capture crack surfaces for the construction of shape functions, elements are still required to be divided according to the crack surface for accurate integration. Moreover, to simulate multi-crack propagation and due to the impossibility to anticipate fracture path during growth, the mesh requires quite a high degree of homogeneous refinement in order to accurately capture deformation, and possible stress singularities, in any given region of the model. When the geometry of cracks is complicated, XFEM also has difficulties representing intra-element fracture intersection, multiple fractures in one element, requires of special approaches to handle fracture tips within elements, and is bound to a fixed amount of level-set functions that restrict crack orientation within each element.

Similarly to the XFEM, the cracking particle method adds enrichment functions into the global approximation of conventional FEM to consider the crack tips and crack boundaries. However, in the cracking particle method, fractures are represented implicitly and are represented by a set of point-based enriched functions [Rabczuk et al., 2007b,a, 2004]. Using H-adaptivity, particles can be added and removed without changing the underlying data structure [Rabczuk and Belytschko, 2005]. The cracking particle method was extended to 3D by Rabczuk and Belytschko [2007]. The cracking particle method was proposed to simplify the geometric demonstration for cracks. To simulate fragmentation, fragments are split by sets of growing fracture surfaces. However, in the cracking particle method, there is no explicit information of the fracture surface. Thus, a methodology relying on cracking particle method would require reinterpretation of fracture geometry during growth and at the fragmentation stage.

The mesh-free method was proposed by Belytschko et al. [1994] to avoid the drawbacks of remeshing for the computation of displacement field and stress field [Belytschko et al., 1994, 1996]. Then the kernel function was introduced into mesh-free method to improve numerical stability [Liu et al., 1995, Chen et al., 1996]. When the mesh-free method was first used to simulate fracture analysis, it was aided by a background mesh and a visibility criterion [Belytschko et al., 1995a]. In this case, the construction of shape functions is not based on a background mesh, and the mesh is applied only for integration in the computational domain. Numerical examples show that the mesh-free method can obtain accurate stress intensity factors and crack growth can be modelled conveniently. Since the mesh-free method avoids re-meshing, it is attractive to simulate fracture propagation in 3D [Rabczuk et al., 2007b]. Even though stress intensity factors can be obtained accurately using the mesh free method, stress oscillation around the crack tip requires adequate point density and refinement of the underlying mesh.

Enriched Element-free Galerkin method (EFG) Belytschko et al. [1994] formulation for fracture problems can obtain accurate stress intensity factors with few degrees of freedom in 2D [Fleming et al., 1997] and 3D [Duflo, 2006]. Initiation, growth, coalescence and branching of multiple cracks, as well non-linear material and dynamics were taken into account under the frame of mesh free method with enrichment [Bordas et al., 2006]. The reproducing kernel particle method has also been used to model arbitrary crack propagation in 2D and 3D [Ren and Li, 2010, Ren et al., 2011]. The mesh-free method has several disadvantages. First, the distribution of nodes has a significant influence on accuracy. Second, the time to generate node connectivity is not less than the time spending on generating a mesh for an equivalent element method simulation [Idelsohn and Oate, 2006]. Finally, the support domain of nodes in the mesh free method is round or ball, not controlled by the geometric boundary, rendering it difficult to apply external forces exactly.

The Finite Element Method (FEM) remains the most popular numerical method to analyse the behaviour of solid material in engineering [Zienkiewicz and Taylor, 2000]. The finite element method has the ability to handle nonlinear behaviour and boundary conditions. A salient feature of the FEM is that the continuum computational domain is discretised by a set of dis-

crete elements, which are connected using topological connection. The interpolation of FEM is constructed on these elements to ensure the compatibility of the interpolation. However, the numerical compatibility condition is not the same as the physical compatibility condition of a continuum. In fracture propagation simulation, the mesh is required to be updated and match the fracture surfaces. To simulate the propagation of cracks, there are two main challenges: first, to accurately compute stress intensity factors at fracture tip; second, to capture fracture geometry change due to growth. Thus, advanced, non-interactive, remeshing technology simulations is essential for its implementation. Traditional methods for stress intensity factor computation rely on brick-wise meshes to compute energy around the tips, introducing further constraints to the required meshing. Nonetheless, the method has been widely used for fracture propagation during the past decades. Mesh refinement aimed at enhancing stress field modelling was initially set to be localized at the tips, due to computational time restrictions [Wawrzynek and Ingraffea, 1989]. As meshing became less expensive, full remeshing approaches were proposed as alternatives [Bittencourt et al., 1996] to bypass refinement region identification and merging. However, most available methods, albeit aimed at propagation of quasi-static brittle fracture cracks in 3D, rely on manual interaction during remeshing and resort to cumbersome brick-wise meshes around the tip for accurate stress intensity factor computations [Gurses and Miehe, 2009]. By implementing a methodology for stress intensity factor computation on arbitrary meshes, it becomes feasible to formulate a FEM-based method, which in combination with re-meshing, can simulate fully automatic multiple fracture propagation in 2D [Bouchard et al., 2000, Paluszny and Matthai, 2009] and 3D [Paluszny and Zimmerman, 2011]. The simulation of fragmentation is instrumental as it allows modelling of scenarios for the prediction and control of the process, and adds to the understanding of its controlling mechanisms.

In the present work, the fragmentation simulation involves capturing two main processes: damage and cracking of single bodies, and dynamics/collision between fragments. The analysis of damage and cracking in single bodies includes challenges such as defining initial material properties and rock heterogeneities, crack nucleation, and propagation of multiple cracks. Modelling collisions between fragments includes capturing processes such as collision detection, force transfer due to impact and compression, and energy loss during collision.

1.2 Block caving system

Block caving system [Brown, 2003] is a effective and environmental friendly method to extract massive ore fragments from underground. The caving method for mining is frequently used in the extraction of ore bodies [Brown, 2003]. Widely used caving methods include block caving [Kvapil et al., 1989], panel caving [Trueman et al., 2002], and sub-level caving [Freidin et al., 2008]. The present work focuses on block caving and addresses the fragmentation phenomenon in block caving system (see Fig. 1.1). The block caving system seems to be environmental friendly, because most of the operations are carried out underground, which avoid damaging the

buildings and landscape. Meanwhile, the block caving system is an effective mining technology, which damages a part of ore bed using gravity and collects fragmented ore through an array of undercutting draw points [Brown, 2003]. In the process of block caving, a series of haulage tunnels are constructed to extract ores from different levels [Sainsbury, 2012]. In contrast to open pit caving, there is a large uncertainty related to the advance and dynamics of the cave, and how cave progression can be controlled due to its occluded nature. Despite its elevated cost, block cave mining is sometimes necessary as the ore may be too deep to be mined using an open pit, or the pit may not be allowed for environmental reasons. In a block caving system, the ore bed is initially fractured by explosion and further fracturing fragments of the rock bed are induced by gravity. Draw points and tunnels are then drilled to collect ores. The present work discusses some critical issues related to block caving: the influence of size and shape of ore fragments on the gravity flow of material; secondary fragmentation which occurs during the movements of ore fragments from their initial position to the draw points; and the hang-up phenomenon. Analysis of gravity flow is critical since it strongly influences the speed of extraction. Understanding gravity flow also helps to investigate the interaction among ore fragments, which is the foundation of secondary fragmentation modelling. The present work makes use of the novel energy tracking method to resolve the dynamics of ore fragments. The interlocking among oversized ore fragments frequently causes the hang-up phenomenon around draw points, which might terminate the mining process. Extra explosion and operation are required to further fragment these hang-up over rocks. Secondary fragmentation reduces the size of ore fragments, and delays or avoids hang-up. Secondary fragmentation in this context is consistently used within the present thesis as an application of the developed fragmentation method.

1.3 Main contributions and publications

The present work focused on the simulation of fragmentation, which combines the simulation of fragment dynamics and the simulation of fracture propagation. The geometry and energetic state of thousands of created fragments are investigated. A velocity dependent fragmentation pattern-based approach is developed and then used to model the influence of secondary fragmentation in block caving system. The main contributions of this thesis are the following:

- Impulse based methods are used to resolve the collision forces among multiple ore fragments. A necessary condition for the impulse method to be energy conservative was proposed.
- Energy tracking method improves the overall ability of impulse based methods to capture propagation of forces during dynamics simulations.
- Hybrid impulse-based method and FEM is applied to model fragmentation. Fragment sizes follow a two parameter Weibull distribution.

- To reduce computational cost, a low-cost Weibull distribution based fragmentation pattern is developed to fragment colliding bodies efficiently.
- The fragmentation pattern based method is applied to investigate the fragmentation phenomenon within the context of a block caving system. When the size and position of the ore fragments are the same, the weaker material, which leads to fragmentation, predicts higher extraction rates as opposed to the non-fragmentation experiments.

The publications associated to the above contributions are listed:

- X.H. Tang, A. Paluszny, R.W. Zimmerman, 2013, An impulse-based energy tracking method for collision resolution, *Computer Methods in Applied Mechanics and Engineering*, under review.
- X.H. Tang, A. Paluszny, R.W. Zimmerman, 2013, Energy conservative property of impulse-based methods for collision resolution, *International Journal for Numerical Methods and Engineering*, 95 (6), 529-540.
- A. Paluszny, X. H. Tang, R.W. Zimmerman, 2013, Fracture and impulse based finite-discrete element modeling of fragmentation, *Computational Mechanics*, 52(5), 1071-1084.
- X.H. Tang, A. Paluszny, R.W. Zimmerman, 2013, A study of the influence of fragmentation in ore-pass hang-up phenomena. *ARMA* 13-406.
- X.H. Tang, A. Paluszny, and R.W. Zimmerman, 2012, Impulse-based simulation of particle flow during subsurface block-cave mining. *XXIII Int. Cong. Theor. Appl. Mech.*, 1924 August, Beijing, China.
- A. Paluszny, X.H. Tang, and R.W. Zimmerman, 2012, A multi-modal approach to 3D fracture and fragmentation of rock using impulse-based dynamics and the finite element method. *ARMA* 12-213.

1.4 Thesis organisation

This study has focussed on the simulation of the fragmentation process, from the fragmentation of a single colliding body to the fragmentation of hundreds of ore fragments. The collisions between multiple bodies are resolved using the impulse-based energy tracking method. Two approaches are developed to model the fracture propagation within a single colliding body: first, the finite element method and linear elastic fracture mechanics (LEFM) are used to model the fracture propagation inside ore fragments; second, a low-cost fragmentation based approach is developed to reduce computational cost and model the fragmentation of hundreds of bodies. Finally, the influence of fragmentation on the hang-up of ore fragments around a single draw point is investigated using the fragmentation simulator.

Chapter 2 introduces impulse-based methods, which are able to effectively handle high frequent collisions among hundreds and thousands of bodies. Impulse-based methods make use of non-penetrating constraints, apply impulses to colliding bodies, and change the relative velocities between colliding bodies directly. Impulse-based methods do not need calibration to determine the value of penetration parameters which is required in penalty-based method. The main contribution in Chapter 2 is the discussion about the energy conservation property of impulse-based methods. A necessary condition is proposed for the impulse-based methods to be energy conservative [Tang et al., 2012a]. According to the condition, Guendelman’s impulse-based approach is energy conservative, whereas Hahn’s approach does not appear to. Both Guendelman’s impulse-based approach and Hahn’s impulse-based approach are used to run a group of numerical tests. According to the numerical results, ensuring energy conservation property seems to be able to improve the stability of simulation for collision resolution in a system of non-convex polyhedral objects.

Chapter 3 proposed a novel impulse based method, energy tracking method (ETM). The dominant features of ETM are: first, like all the impulse-based methods, impulse are applied to colliding bodies to avoid penetration between bodies; second, by applying impulses gradually to the contact points, the impulses are able to influence multiple contact points at the same time, enabling propagation of impulse effects, and thereby improving the overall ability of the method to capture propagation of forces during dynamics simulations [Tang et al., 2012b, Paluszny et al., 2012]. In addition, ETM ensures the energy conservation during collision and improves numerical stability.

Chapter 4 introduces an algorithm to model fragmentation process, which couples fracture and fragmentation using a combined finite element, impulse-based approach [Paluszny et al., 2012, 2013]. This work can be regarded as the extension of a 3D fracture propagation method, where the geometry of arbitrary fractures is presented exactly and fracturing propagation is resolved using finite element method and linear elastic fracture mechanics (LEFM) [Paluszny and Zimmerman, 2011, 2013]. The angle and velocity of fracture propagation are calculated by maximum circumferential stress criterion and Paris’ law. The fragmentation of a single sphere is used to validate the present method. The observed fragment volume distribution can be fitted using a two-parameter Weibull distribution, and the average fragment sizes decreases as a function of the impact velocity.

Chapter 5 presents a low-cost fragmentation pattern based approach to reduce computational cost and meet the requirement of the fragmentation simulation of hundreds and thousands of colliding bodies. The colliding bodies are fragmented using fragmentation pattern directly, which avoids FEM-based crack propagation simulation and remeshing and re-creates solid geometry during simulation. The fragmentation pattern, which is controlled by three parameters, is established by learning from existing numerical and experimental results in which three parameters are defined as a function of the colliding velocities directly. The Weibull distribution is used to evaluate the performance of the fragmentation pattern. The performance of this low-cost

fragmentation pattern based approach is validated by a group of numerical tests, which include the fragmentation of single sphere and single realistic shaped rock and the fragmentation of a pack of spheres.

In Chapter 6, the fragmentation simulator is used to investigate the influence of secondary fragmentation on the hang-up of ore fragments around a single draw point in a block caving system. Ten different models are presented. First, the gravity flow and hang-up phenomenon of spherical ore fragments with two different sizes of spherical ore fragments is investigated. Then, the gravity flow and hang-up phenomenon of cuboid ore fragments is discussed. Two initial sizes of cuboid ore fragments populate the simulation models. Three different arrangements of fragments are considered. All the models are simulated with and without fragmentation. Simulations with fragmentation yield significantly different predictions as compared to their static counterparts in terms of amount of extracted volume and frequency of draw point hang-up.

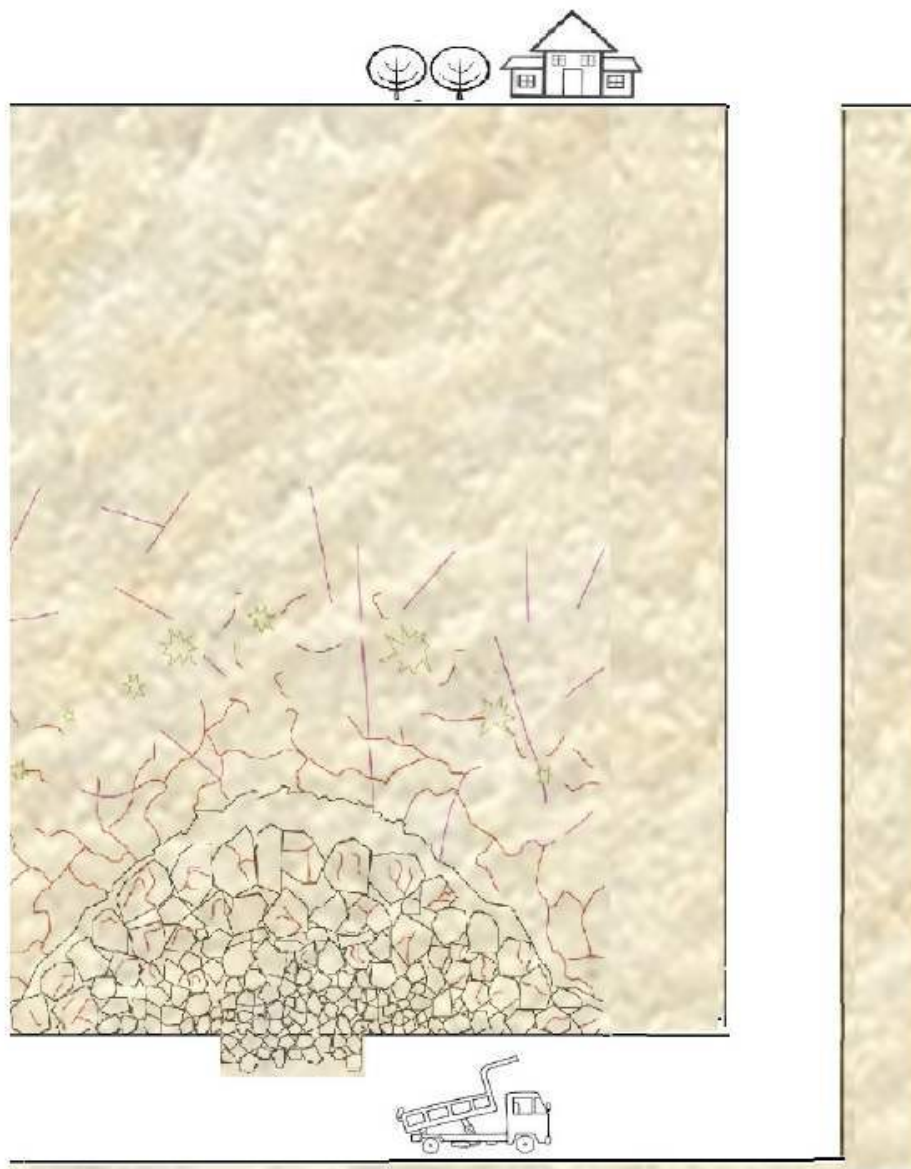


Figure 1.1: Simplified block caving system.

2 Energy conservative impulse-based discrete element method

2.1 Introduction

The simulation of collision-driven interactions in a multi-body system is instrumental in modelling scenarios for the prediction and control of industrial processes, and adds to the understanding of their controlling mechanisms. Often involving hundreds of thousands to millions of interacting bodies, these simulations must constantly weigh numerical stability, physical accuracy, and computational speed, if they are to become industrial tools. This trade-off is especially critical in the simulation of rock fragmentation for mining engineering [Vyazmensky, 2008, Munjiza and Latham, 2004], and for tumble mills in agriculture [Mishra, 2003], for which geometric representation is believed to play a decisive role in the interaction behaviour. The main methods proposed to simulate multiple contacts in a rigid body system include analytic [Baraff, 1994], penalty [Guises et al., 2009], and impulse-based methods [Mirtich, 1996, Guendelman et al., 2003].

Analytical Method Analytical methods compute contact forces by expressing the system as a linear complementary problem (LCP) [Chardonnet et al., 2006]. Analytical methods do not require the determination of a penalty parameter, and do not require a damping parameter in order to remain stable. One drawback of analytical methods arises when there is a need to consider the influence of friction. The system becomes complex when dynamic friction is added, as the defining matrix becomes non-symmetric and non-positive definite e.g. [Drumwright, 2008]. Furthermore, when static friction is considered, the constraints of the two tangent directions at the contact point become nonlinear, and the problem becomes a nonlinear complementarity problem. An alternative method is the Lagrange-multiplier method, which does not need to calibrate the penalty parameter and damping parameter, and is able to exactly implement the boundary constraints. However, the Lagrange-multiplier method introduces extra unknowns, and the Lagrange-multiplier method is only available for bilateral constraint, such as joint constraint; it lacks the ability to solve unilateral constraints [Drumwright, 2008].

Penalty Method The penalty method [Belytschko, 1993, Babuška, 1973] transforms constrained optimisation problems into a set of non-constrained optimisation problems by iteratively computing a proper penalty parameter which enforces collisions as non-penetrating

[Bourago and Kukudzhanov, 2005, Hasegawa and Fujii, 2003a]. The penalty method is simple and is able to simulate the propagation of impact-induced stress waves within deformable colliding bodies [Wang and Tonon, 2011]. The penalty method suffers from some drawbacks. Firstly, the penalty method requires the value of normal and tangential stiffness of a force-spring system, which is difficult to determine and requires numerical and experimental tests to validate [Hentz et al., 2004, McCarthy et al., 2010]. Secondly, if the bodies collide with high relative velocity, immense overlap may be detected, which leads to large forces and instability in simulation. As a result, a somewhat arbitrary damping parameter is usually used to induce energy dissipation and to keep the simulation stable [Laursen and Chawla, 1997, Drumwright, 2008]. To guarantee energy conservation, the midpoint integration rule combined with the Lagrange multiplier method [Haikal and Hjelmstad, 2007] and penalty regularization of contact rates [Noels et al., 2006, Chawla and Laursen, 1998] have been proposed. Thirdly, the penalty method relies on the expensive and geometrically complex computation of penetration distances, volumes and surfaces over intersection of two colliding bodies in order to compute the damping condition [Munjiza and Andrews, 2000]. In engineering, the penalty method is widely used in the context of the discrete element method (DEM) [Cundall and Strack, 1979, Magnier and Donz, 1999], smoothed particle hydrodynamics (SPH) [Lucy, 1977, Gingold and Monaghan, 1977], material point method (MPM), and discontinuous deformation analysis (DDA) [Shi and Goodman, 1984]. These three methods are able to simulate the collision and contact between two bodies, as well as to simulate the deformation and fracturing of deformable bodies. The collision resolution of the previous methods is based on the *a posteriori* penalization of mesh penetration. DEM assembles a set of tetrahedra [Latham et al., 2008b,a] or sphere-conglomerates [Chandramohan and Powell, 2005, Vyazmensky, 2008] in three dimensions. SPH [Liu and Liu, 2003] and MPM [Bardenhagen et al., 2000, Sulsky and Kaul, 2004] are both point-based methods, often applied to simulate explosions and problems involving large deformations. Both apply point-to-point contact detection, which is associated with difficulty in determining the direction of contact force vector [Campbell et al., 2000]. DDA solves for the stresses, displacements and contacts of bodies by minimizing the potential energy, and represents the strain energy arising from the contact force as a function of penetration. Solving for all the constraints in a system of linear equations implies that DDA has the tendency to be extremely time consuming as compared to the impulse method.

Discontinuous Deformation Analysis Discontinuous Deformation Analysis (DDA) is an extension of the discrete element method that was originally proposed by Shi and Goodman [1984]. DDA solves for the stress-displacement of bodies in a manner similar to FEM; however, it is focussed on solving the interaction of discrete blocks along discontinuities for problems of fractured and jointed rock masses. DDA is also an implicit discrete element model [Jing, 1998]. Instead of considering the contact response separately, a stiffness matrix is implemented, based on minimizing the potential energy. The strain energy from the contact force is represented by a function of penetration [Shi and Goodman, 1984]. The advantages of DDA include that

the equilibrium of dynamics is always satisfied, stability can be ensured without any extra damping parameter, and penetration and traction are avoided at each step. The disadvantages are the expensive requirements for computational time and memory. Some extensions of DDA include DDA with second-order displacement functions [Grayeli and Mortazavi, 2006], and its application to solve dynamics problems [Tsesarsky et al., 2005].

Impulse Based Method The impulse method (IM) efficiently and accurately models high-frequency collisions of complex non-convex shapes, based on sequentially enforcing non-penetrating constraints [Hahn, 1988, Schmitt et al., 2005, Mirtich and Canny, 1995]. IM has been shown to be energy conservative [Tang et al., 2013], does not rely on penalty parameters, does not require the computation of penetration between bodies, and benefits from its ability of directly using meso-scale material properties. IM is regularly applied to model haptic interfaces in robotics research [Hollerbach et al., 1999] and to simulate the motion of tendons, joints, and muscles [Sueda et al., 2008, Weinstein et al., 2008]. IM has been used in engineering for simulated sensorless manipulation of objects [Reznik et al., 1997], and, in combination with the finite element method, for the simulation of gravity-driven fragmentation [Paluszny et al., 2012]. Unlike *a posteriori* methods, the impulse-based method does not rely on arbitrary penalty and damping parameters that define repulsion as a function of penetration. Instead, it predicts collisions as a function of predicted impact trajectory and impulse. Thus, IM does not require geometric intersection checks, but instead relies collision time estimation algorithms, e.g. the efficient Lin-Canny algorithm [Lin, 1993, Lin and Canny, 1991]. Contact detection is based on the computation of the relative normal velocity of colliding bodies at each contact point. Newton’s impact law, as defined for a single-point collision, is applied either sequentially (SQM) [Erleben et al., 2005] or simultaneously (SMM) [Baraff, 1994, Coutinho, 2001] to deal with multiple collisions and multiple contact points.

There are at least two versions of SMM. In 1989, Baraff [Baraff, 1994] proposed a simple version of SMM which treated multiple collisions as a linear complementarity problem; it only considered normal impulses and the change of normal velocities, and did not consider friction. The generic SMM [Coutinho, 2001] enforces constraints to all collisions by solving a system of linear equations, taking into account impulses in the normal and tangential directions, and considering friction.

SQM treats multiple concurrent collisions one-by-one. Non-penetrating constraints are enforced to multiple collisions sequentially, iterating over the contact points until no objects collide. In SQM, Newton’s impact law is locally applied to each contact, making it particularly well suited for parallelisation. The analysis of energy dissipation among multiple collisions is a critical issue in engineering, such as the design of comminution devices [Powell and McBride, 2006, Khanal and Powell, 2010, Powell et al., 2008]. In the present work, the physical accuracy of the impulse-based method is validated in the context of energy and momentum conservation. Energy and momentum conservation for frictionless and perfectly elastic collisions is a basic requirement for a numerical method, to ensure that energy loss in simulation is not caused

by numerical error. A necessary condition for an impulse-based method to be energy conservative is proposed below. The present work examines different proposals available in the literature regarding the resolution of collisions using this approach, namely those proposed by Hahn [1988] and Guendelman et al. [2003]. Numerical experiments (described below) indicate that Guendelman’s approach is energy conservative.

2.2 Impulse Based Method

2.2.1 Single-point collision: Contact model

Hahn [1988] was the first to apply the impulse-momentum form of Newton’s second law to compute collision response of multi-body systems [Garstenauer, 2006]. Using this approach, collisions are modelled sequentially, and do not require the computation of penetration depth, area, or volume between the interacting bodies. Collision time is assumed to be of infinitesimal duration; therefore, during the collision, the positions of the bodies are treated as constant. Schmitt et al. [2005] proved that impulse-based solutions for collisions converge towards the exact solution of the dynamics problem as the timestep is reduced. This section introduces the impulse-based approaches proposed by Hahn [1988] and Guendelman et al. [2003]. As time progresses, $\{t_0, t_1, t_2, \dots\}$, positions and velocities of bodies are updated. The non-penetrating strategy [Lin, 1993, Lin and Canny, 1991], based on the fast search of the closest features among convex polyhedra, identifies collision candidates based on the concept of collision time. Accurate estimation of collision time, t_c , is required to apply impulses and avoid geometric overlap due to collision (Fig. 2.1). A body b_N collides with another body b_M at contact point c_i (Fig.

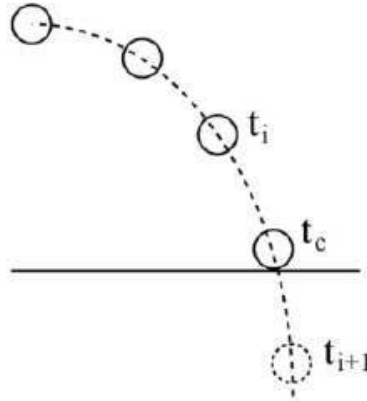


Figure 2.1: Estimation for the time of contact. Each circle represents a moving object approaching the ground at consecutive time steps. A collision is detected between time steps t_i and t_{i+1} . Collision time is estimated as t_c .

2.2). A pair of impulses, $\mathbf{p}_i(s_0)$ and $-\mathbf{p}_i(s_0)$ are applied at the contact point. The velocities of

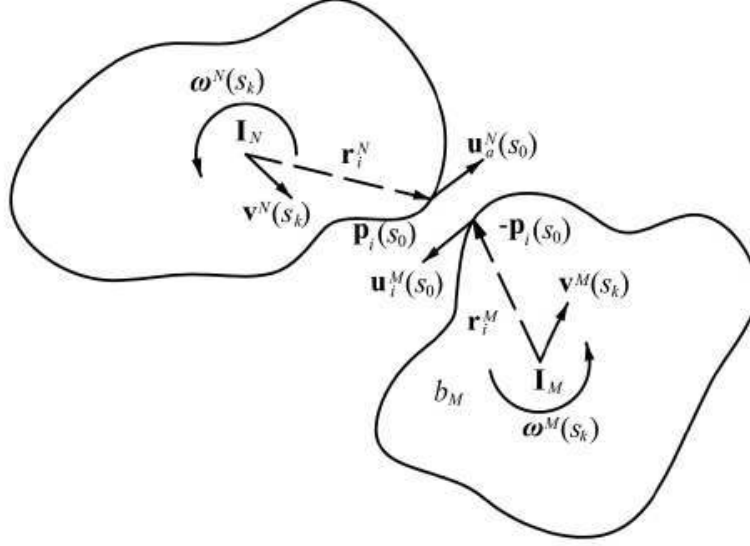


Figure 2.2: Collision between bodies b_N and b_M at collision c_i : $k=0$ or 1 , s_0 and s_1 indicate the status before and after applying impulse. $\mathbf{v}^N(s_0)$ and $\mathbf{v}^M(s_0)$ are linear velocities before collision. $\mathbf{v}^N(s_1)$ and $\mathbf{v}^M(s_1)$ are linear velocities after collision. $\omega^N(s_0)$ and $\omega^M(s_0)$ are angular velocities before collision. $\omega^N(s_1)$ and $\omega^M(s_1)$ are angular velocities after collision. $\mathbf{r}_i^N$ and $\mathbf{r}_i^M$ are the vectors from the centre of mass of bodies to the contact point c_i . $\mathbf{u}_i^N(s_0)$ and $\mathbf{u}_i^M(s_0)$ are the velocities at the contact point before collision. $\mathbf{u}_i^N(s_1)$ and $\mathbf{u}_i^M(s_1)$ are the velocities at the contact point after collision. $\mathbf{p}_i(s_0)$ is the impulse applied to the two colliding bodies to avoid penetration.

colliding bodies at the contact point c_i before and after collision are denoted as

$$\mathbf{u}_i^N(s_0) = \mathbf{v}^N(s_0) + \omega^N(s_0) \times \mathbf{r}_i^N \quad (2.1)$$

$$\mathbf{u}_i^N(s_1) = \mathbf{v}^N(s_1) + \omega^N(s_1) \times \mathbf{r}_i^N \quad (2.2)$$

$$\mathbf{u}_i^M(s_0) = \mathbf{v}^M(s_0) + \omega^M(s_0) \times \mathbf{r}_i^M \quad (2.3)$$

$$\mathbf{u}_i^M(s_1) = \mathbf{v}^M(s_1) + \omega^M(s_1) \times \mathbf{r}_i^M \quad (2.4)$$

where, $\mathbf{r}_i^N = \{r_{i,x}^N, r_{i,y}^N, r_{i,z}^N\}^T$ and $\mathbf{r}_i^M = \{r_{i,x}^M, r_{i,y}^M, r_{i,z}^M\}^T$ are the vectors from the centre of mass of bodies to the contact point c_i , which is assumed to be constant during the collision. $\mathbf{u}_i^N(s_0)$ and $\mathbf{u}_i^N(s_1)$ are the velocities of body b_N at the contact point c_i before and after applying impulse. $\mathbf{u}_i^M(s_0)$ and $\mathbf{u}_i^M(s_1)$ are the velocities of body b_M at the contact point c_i before and after applying impulse, respectively. $\mathbf{v}^N(s_0)$ and $\mathbf{v}^N(s_1)$ are the linear velocities of body b_N at the centre of mass before and after applying impulse. $\mathbf{v}^M(s_0)$ and $\mathbf{v}^M(s_1)$ are the linear velocities of body b_M at the centre of mass before and after applying impulse, respectively. $\omega^N(s_0)$ and $\omega^N(s_1)$ are the angular velocities of body b_N before and after applying impulse. $\omega^M(s_0)$ and $\omega^M(s_1)$ are the angular velocities of body b_M before and after applying impulse.

The relative velocities of two colliding bodies at contact point c_i are defined as

$$\mathbf{u}_i(s_0) = \mathbf{u}_i^N(s_0) - \mathbf{u}_i^M(s_0) \quad (2.5)$$

$$\mathbf{u}_i(s_1) = \mathbf{u}_i^N(s_1) - \mathbf{u}_i^M(s_1) \quad (2.6)$$

where $\mathbf{u}_i(s_0)$ and $\mathbf{u}_i(s_1)$ are relative velocities before and after applying the impulse, respectively. The change of relative velocity is defined as

$$\Delta \mathbf{u}_i(s_0) = \mathbf{u}_i(s_1) - \mathbf{u}_i(s_0) \quad (2.7)$$

When a pair of impulses of opposite signs $\mathbf{p}_i(s_0)$ and $-\mathbf{p}_i(s_0)$ is applied to colliding bodies, these impulses effectively change their linear and angular velocities,

$$\mathbf{p}_i(s_0) = m^N \Delta \mathbf{v}^N(s_0) \quad (2.8)$$

$$\mathbf{r}_i^N \times \mathbf{p}_i(s_0) = \mathbf{I}^N \Delta \boldsymbol{\omega}^N(s_0) \quad (2.9)$$

$$-\mathbf{p}_i(s_0) = m^M \Delta \mathbf{v}^M(s_0) \quad (2.10)$$

$$-\mathbf{r}_i^M \times \mathbf{p}_i(s_0) = \mathbf{I}^M \Delta \boldsymbol{\omega}^M(s_0) \quad (2.11)$$

where $\mathbf{I}^N$ and $\mathbf{I}^M$ are the inertia tensors for body b_N and body b_M , respectively, which are constant during the collision, and are computed using an explicit exact expression [Tonon, 2004], and

$$\Delta \mathbf{v}^N(s_0) = \mathbf{v}^N(s_1) - \mathbf{v}^N(s_0) \quad (2.12)$$

$$\Delta \boldsymbol{\omega}^N(s_0) = \boldsymbol{\omega}^N(s_1) - \boldsymbol{\omega}^N(s_0) \quad (2.13)$$

$$\Delta \mathbf{v}^M(s_0) = \mathbf{v}^M(s_1) - \mathbf{v}^M(s_0) \quad (2.14)$$

$$\Delta \boldsymbol{\omega}^M(s_0) = \boldsymbol{\omega}^M(s_1) - \boldsymbol{\omega}^M(s_0) \quad (2.15)$$

where m^N and m^M are the masses of body b_N and body b_M , respectively. Mirtich [1996] proposed that, once the change of relative velocity is known, impulses are computed from the collision matrix,

$$\mathbf{p}_i(s_0) = \mathbf{K}_i^{-1} \Delta \mathbf{u}_i(s_0) \quad (2.16)$$

where the collision matrix $\mathbf{K}_i$, which is constant, non-singular, symmetric, and positive definite, is defined as

$$\mathbf{K}_i = \left(\frac{1}{m^N} + \frac{1}{m^M} \right) \mathbf{1} - (\tilde{\mathbf{r}}_i^N (\mathbf{I}^N)^{-1} \tilde{\mathbf{r}}_i^N + \tilde{\mathbf{r}}_i^M (\mathbf{I}^M)^{-1} \tilde{\mathbf{r}}_i^M) \quad (2.17)$$

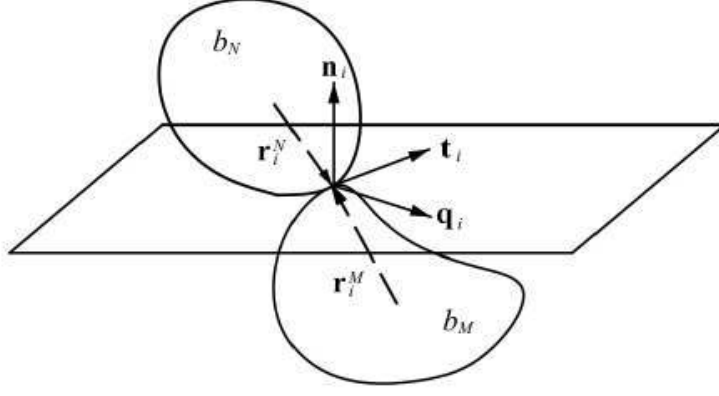


Figure 2.3: A local coordinate system $(\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i)$ is defined at the contact point between bodies b_M and b_N .

where $\mathbf{1}$ is the identity matrix, and

$$\tilde{\mathbf{r}}_i^N = \begin{pmatrix} 0 & -\tilde{r}_{i,z}^N & \tilde{r}_{i,y}^N \\ \tilde{r}_{i,z}^N & 0 & -\tilde{r}_{i,x}^N \\ -\tilde{r}_{i,y}^N & \tilde{r}_{i,x}^N & 0 \end{pmatrix} \quad (2.18)$$

$$\tilde{\mathbf{r}}_i^M = \begin{pmatrix} 0 & -\tilde{r}_{i,z}^M & \tilde{r}_{i,y}^M \\ \tilde{r}_{i,z}^M & 0 & -\tilde{r}_{i,x}^M \\ -\tilde{r}_{i,y}^M & \tilde{r}_{i,x}^M & 0 \end{pmatrix} \quad (2.19)$$

where $\tilde{\mathbf{r}}_i^N$ and $\tilde{\mathbf{r}}_i^M$ are the cross-product matrices of $\mathbf{r}_i^N$ and $\mathbf{r}_i^M$. For the following analysis, a local orthogonal coordinate system in $\{\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i\}$ is defined at the locality of the contact in terms of the normal and tangential components of the collision (Fig. 3.2). It follows:

$$\mathbf{t}_i = \frac{\mathbf{n}_i \times \mathbf{p}_i(s_0)}{|\mathbf{u}_i(s_0)|} \quad (2.20)$$

$$\mathbf{q}_i = \mathbf{t}_i \times \mathbf{n}_i \quad (2.21)$$

where $\mathbf{n}_i$ is the normal direction of collision surface, and $\mathbf{t}_i$ and $\mathbf{q}_i$ are the two tangential directions at the contact point. It follows that the impulse $\tilde{\mathbf{p}}_i(s_0) = \{\tilde{p}_{i,t}(s_0), \tilde{p}_{i,n}(s_0), \tilde{p}_{i,q}(s_0)\}^T$ and relative velocity before and after collision can be formulated in terms of this local coordinate system as

$$\begin{cases} \tilde{\mathbf{u}}_i(s_0) = \mathbf{L}_i \mathbf{u}_i(s_0) \\ \tilde{\mathbf{u}}_i(s_1) = \mathbf{L}_i \mathbf{u}_i(s_1) \\ \tilde{\mathbf{p}}_i(s_0) = \mathbf{L}_i \mathbf{p}_i(s_0) \end{cases} \quad (2.22)$$

where $\mathbf{L}_i = (\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i)^T$ defines the transformation from global to local coordinates. The empirical law is used in the impulse-based method, in which it models response by reversing the

normal relative velocity:

$$\tilde{\mathbf{u}}_n(s_1) = -e\tilde{\mathbf{u}}_n(s_0) \quad (2.23)$$

2.2.2 Single-point collision: Impulse and work

Mirtich [1996] proposed that the work done by contact forces, $\Delta W_i(s_0)$, is path independent and is a function of the relative velocity before and after applying the impulse:

$$\Delta W_i(s_0) = \frac{1}{2} (\mathbf{u}_i(s_1) + \mathbf{u}_i(s_0))^T \mathbf{K}_i^{-1} (\mathbf{u}_i(s_1) - \mathbf{u}_i(s_0)) \quad (2.24)$$

To ensure the conservation of energy of the entire system, this work should be zero. Mirtich [1996] discussed the condition of energy conservation in the impulse-based method, and proposed the following lemma:

Lemma 2.2.1. *If during a collision, the relative contact velocity is exactly reversed, that is to say, $\mathbf{u}_i(s_1) = -\mathbf{u}_i(s_0)$, then the work done by the collision forces is zero, and the energy of the system is therefore unchanged by the collision.*

Note that $\mathbf{u}_i(s_1) = -\mathbf{u}_i(s_0)$ is a potential solution that implies $\Delta W_i(s_0) = 0$. However, it neglects the change of velocity direction caused by the collision. For instance, after an object collides with the ground, the velocity in the normal direction should be reversed, while the velocity in the tangential direction should remain unchanged. Thus, this solution is not often chosen, despite the fact that it conserves energy. An alternative condition for the energy conservative in the impulse-based method is now proposed.

Lemma 2.2.2. *If the impulse is only applied in the normal direction, and the relative velocity in the normal direction is exactly reversed, $\tilde{u}_{i,n}(s_1) = -\tilde{u}_{i,n}(s_0)$, then the energy of the system is conserved during collision.*

Proof. By substituting Eq. (2.16) into Eq. (2.24), it is found that

$$\Delta W_i(s_0) = \frac{1}{2} (\mathbf{u}_i(s_0) + \mathbf{u}_i(s_0))^T \mathbf{p}(s_0) \quad (2.25)$$

The work done by the impulse can be expressed in terms of the defined local coordinate system, as follows:

$$\Delta W_i(s_0) = \frac{1}{2} (\mathbf{L}^{-1} \tilde{\mathbf{u}}_i(s_1) + \mathbf{L}^{-1} \tilde{\mathbf{u}}_i(s_0))^T (\mathbf{L}^{-1} \tilde{\mathbf{p}}(s_0)) \quad (2.26)$$

$$\Delta W_i(s_0) = \frac{1}{2} (\tilde{\mathbf{u}}_i(s_1) + \mathbf{L}^{-1} \tilde{\mathbf{u}}_i(s_0))^T (\mathbf{L}^{-1})^T (\mathbf{L}^{-1}) \tilde{\mathbf{p}}(s_0) \quad (2.27)$$

where $(\mathbf{L}^{-1})^T \mathbf{L}^{-1} = \mathbf{1}$, and so,

$$\begin{aligned} \Delta W_i(s_0) &= \frac{1}{2} (\tilde{\mathbf{u}}_i(s_1) + \tilde{\mathbf{u}}_i(s_0))^T \tilde{\mathbf{p}}(s_0) \\ &= \frac{1}{2} [\tilde{u}_{i,t}(s_1) + \tilde{u}_{i,t}(s_0), \tilde{u}_{i,n}(s_1) + \tilde{u}_{i,n}(s_0), \tilde{u}_{i,q}(s_1) + \tilde{u}_{i,q}(s_0)] \begin{bmatrix} \tilde{p}(s_0)_t \\ \tilde{p}(s_0)_n \\ \tilde{p}(s_0)_q \end{bmatrix} = 0 \end{aligned} \quad (2.28)$$

It assumes that $\tilde{u}_{i,n}(s_1) = -\tilde{u}_{i,n}(s_0)$ and the impulse is only applied to the normal direction. Thus, $\tilde{p}_t(s_0) = 0$ and $\tilde{p}_q(s_0) = 0$, and so

$$\Delta W_i(s_0) = \frac{1}{2} [\tilde{u}_{i,t}(s_1) + \tilde{u}_{i,t}(s_0), 0, \tilde{u}_{i,q}(s_1) + \tilde{u}_{i,q}(s_0)] \begin{bmatrix} 0 \\ \tilde{p}_{i,n}(s_0) \\ 0 \end{bmatrix} = 0 \quad (2.29)$$

□

2.2.3 Single-point collision: Friction

Collision response without friction Hahn [1988] introduced friction into the approach by assuming that the body is static in the tangential directions after collision, and neglecting the tangential velocity after the collision:

$$\tilde{u}_{i,n}(s_1) = -e_n \tilde{u}_{i,n}(s_0) \quad (2.30)$$

which can be rewritten in local coordinates as

$$\begin{bmatrix} \tilde{u}_t(s_1) & \tilde{u}_n(s_1) & \tilde{u}_q(s_1) \end{bmatrix}^T = \begin{bmatrix} 0 & -e \tilde{u}_n(s_0) & 0 \end{bmatrix}^T \quad (2.31)$$

Once the relative velocity after the collision is determined, if the static friction condition is satisfied, then

$$|\tilde{p}_t(s_0)| < \mu |\tilde{p}_n(s_0)| \quad (2.32)$$

then $\mathbf{p}(s_0)$ is computed using Eq. (2.16), where μ is the coefficient of friction. If the static friction condition (2.32) is not satisfied, then, $\mathbf{p}(s_0)$ is computed as follows:

$$\mathbf{p}(s_0) = \mathbf{n} \tilde{p}_n(s_0) + \mu \mathbf{t} |\tilde{p}_n(s_0)|. \quad (2.33)$$

Collision response with friction Guendelman et al. [2003] proposed re-computing the impulse if the static friction assumption is not satisfied. Similarly to Baraff's impulse-based approach, it does so while constraining the relative velocity after the collision to be exactly the inverse of the relative velocity before the collision in the normal direction $\mathbf{n}$, $\tilde{u}_n(s_1) = -\tilde{u}_n(s_0)$. Then, the magnitude of the impulse is computed using the collision matrix, $\mathbf{K}$, as follows:

$$|\mathbf{p}(s_0)| = -\frac{\tilde{u}_n(s_0)(e+1)}{\mathbf{n}^T \mathbf{K}(\mathbf{n} - \mu \mathbf{t})} \quad (2.34)$$

When the static condition is not satisfied, the impulse is defined as

$$\mathbf{p}(s_0) = |\mathbf{p}(s_0)| \mathbf{n} + \mu |\mathbf{p}(s_0)| \mathbf{t}. \quad (2.35)$$

2.2.4 Multiple Collisions: Sequential impulse method

Fig. 2.4 illustrates multiple collisions, $\mathbf{c} = \{c_0, c_1, c_2, \dots, c_n\}$, between multiple bodies, $\mathbf{b} = \{b_0, b_1, b_2, \dots, b_m\}$, detected at a time step t_c , where m is the total colliding bodies and n is the total contact points. At each iteration, the multiple collisions are treated as concurrent collisions and impulses are computed and applied to contact points sequentially. The interaction of multiple collisions is simulated by repeating the computation until all relative normal velocities are non-negative (Algorithm 1).

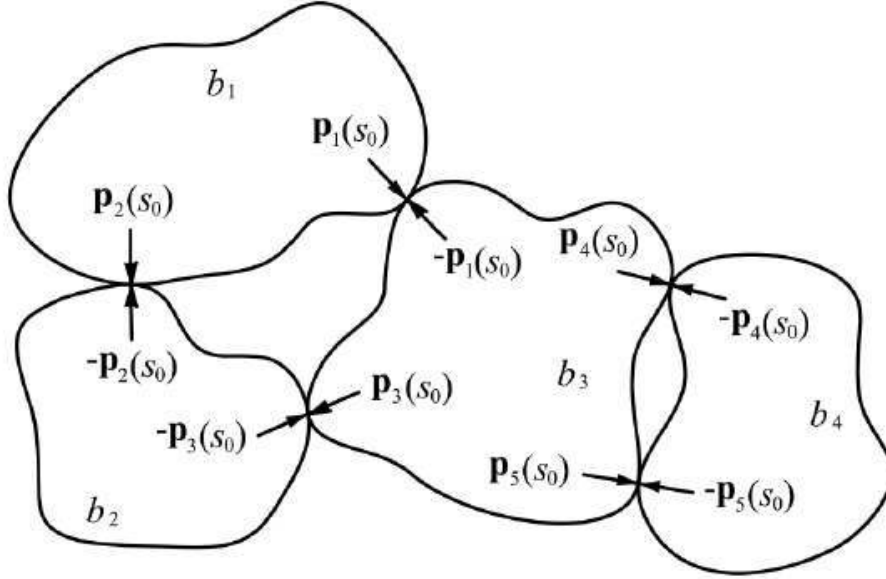


Figure 2.4: Collision of multiple bodies: multiple bodies, $\mathbf{b} = \{b_0, b_1, b_2, \dots, b_m\}$, collide with multiple contact points $\mathbf{c} = \{c_0, c_1, c_2, \dots, c_n\}$, m is the total number of colliding bodies and n is the total number of contact points. A set of impulses, $(\mathbf{p}_0(s_0), \mathbf{p}_1(s_0), \mathbf{p}_2(s_0), \dots, \mathbf{p}_n(s_0))$, are applied to contact points.

Algorithm 1 Sequential impulse method: Collision response for multiple collisions

Require: A set of bodies $\mathbf{b} = \{b_0, b_1, \dots, b_n\}$ collide at contact points $\mathbf{c} = \{c_0, c_1, \dots, c_m\}$

- 1: identify the contact point with smallest relative normal velocity, c_{min}
and its relative normal velocity $\tilde{u}_{min,n}$
 - 2: **while** $\tilde{u}_{min,n} < 0$ **do**
 - 3: compute impulses at c_{min} by Eq. 2.34-2.35
 - 4: update the linear and angular velocity of colliding bodies related to c_{min}
 - 5: recompute the minimum contact point, c_{min} and its relative normal velocity $\tilde{u}_{min,n}$
 - 6: **end while**
-

2.3 Numerical Tests

Numerical tests are now used to investigate energy and linear momentum conservation of the approaches proposed by Hahn [1988] and Guendelman et al. [2003]. In all tests density is 1 kg/m³ and the time step is 1/600 s. The ability to capture energy conservation and linear momentum conservation is essential for a numerical method to simulate collisions correctly. The total system energy, E , is expressed as

$$E = \sum_i^n \left(E_i^p + E_i^k \right) = \sum_i^N \left(m_i g h_i + \int_{V_i} \frac{\mathbf{u}_i^2(\mathbf{x})}{2} dm \right) \quad (2.36)$$

where n is the number of bodies, E_i^p and E_i^k are the potential energy and kinetic energy of body i , m_i is the mass of body i , g is the gravitational acceleration, h_i is the height of the centre of gravity of body i , V_i is the volume, and $\mathbf{u}_i(\mathbf{x})$ is the linear velocity of body i at integration point $\mathbf{x}$. The system is expected to be energy conservative for the special case of perfectly elastic collisions, where $\mu = 0$ and $e_n = 1$. For the case of perfectly elastic collisions, the relative error of E is measured as follows:

$$\varepsilon_e = \left| \frac{E^b - E^f}{E^b} \right| \quad (2.37)$$

where E^b and E^f are the energies before and after collision, and ε_e is the relative error of energy. The total linear momentum of the system is computed as

$$\mathbf{M} = \sum_i^n m_i \mathbf{v}_i \quad (2.38)$$

where $\mathbf{M} = \{M_x, M_y, M_z\}^T$. If there is no external forces acting on a closed system, the linear momentum of the system remains constant. The relative error in the linear momentum of a closed system is defined as

$$\varepsilon_M = \max \left(\left| \frac{M_x^f - M_x^b}{M_x^b} \right|, \left| \frac{M_y^f - M_y^b}{M_y^b} \right|, \left| \frac{M_z^f - M_z^b}{M_z^b} \right| \right) \quad (2.39)$$

where $\mathbf{M}^b$ and $\mathbf{M}^f$ refer to the linear momenta before and after collision. Throughout the above discussion, it is known that IM-Guendelman is able to capture the property of energy conservation, since it is able to ensure that $\tilde{u}_n(s_1) = -\tilde{u}_n(s_0)$. However, IM-Hahn is not able to capture the property of energy conservation, since $\tilde{u}_n(s_1)$ is not exactly the inverse of $\tilde{u}_n(s_0)$. In the following, several collision scenarios are simulated using IM-Guendelman and IM-Hahn, respectively, and the energy and linear momentum before and after collision are measured and compared. These tests include the collision of two tetrahedral fragments, a sphere falling down a slope, and ore fragments flowing through a cylindrical draw point in a block-cave mine [Brown, 2003]. In these tests, each body, whether convex or non-convex, is represented by a conglomerate of tetrahedra (Fig. 2.5).

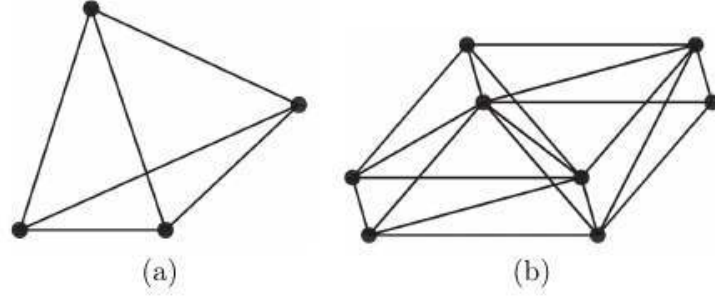


Figure 2.5: Geometric representation of bodies.

2.3.1 Collision of two tetrahedra

The energy- and linear momentum-conservative properties of the collision of two tetrahedra are measured in this test (see Fig. 2.6 (a) and (b)). To eliminate the numerical error from the computation of velocity change caused by gravity, it is assumed that $g=0$ m/s². In the first part of the test, (a), body j is fixed and static, while another, body i , moves with an initial velocity and eventually collides with it (Fig. 2.6). In (b), body j is initially static, but moves after the collision. In (a), the initial linear velocity is $\mathbf{v}_1 = (0, 0, -10)$ m/s, and the initial angular velocity is $\boldsymbol{\omega}_1 = (0, 0, -10)$ m/s. In (b), initial linear velocities are $\mathbf{v}_1 = (0, 0, -10)$ m/s and $\mathbf{v}_2 = (0, 0, 0)$ m/s, and initial angular velocities are $\boldsymbol{\omega}_1 = (5, 10, 3)$ m/s and $\boldsymbol{\omega}_2 = (0, 0, 0)$ m/s. Fig. 2.6(b) shows snapshots of the simulation. In (b), no external force acts on the two tetrahedra. Therefore, this is a closed system. Results, shown in Table 2.1, suggest that both IM-Hahn and IM-Guendelman are linear momentum conservative. However, while IM-Guendelman appears to conserve energy, IM-Hahn does not.

Table 2.1: Energy and linear momentum for (a) a moving tetrahedron collides with a static one, (b) two colliding tetrahedra

ts size	(a) IM-Hahn	(a) IM-Guendelman	(b) IM-Hahn		(b) IM-Guendelman	
	ε_e	ε_e	ε_e	ε_m	ε_e	ε_m
1/60	-2.12	4.37×10^{-8}	-2.11×10^{-2}	0	5.91×10^{-4}	0
1/120	-2.12	-4.15×10^{-9}	-2.11×10^{-2}	0	5.91×10^{-4}	0
1/300	-2.19	-1.18×10^{-9}	-2.57×10^{-2}	0	7.48×10^{-4}	0
1/600	-2.19	-4.06×10^{-8}	-2.57×10^{-2}	0	7.48×10^{-4}	0
1/900	-2.19	3.71×10^{-8}	-2.57×10^{-2}	0	7.48×10^{-4}	0
1/1200	-2.19	-9.06×10^{-9}	-2.63×10^{-2}	0	7.65×10^{-4}	0
1/2400	-2.20	1.44×10^{-8}	-2.63×10^{-2}	0	7.65×10^{-4}	0

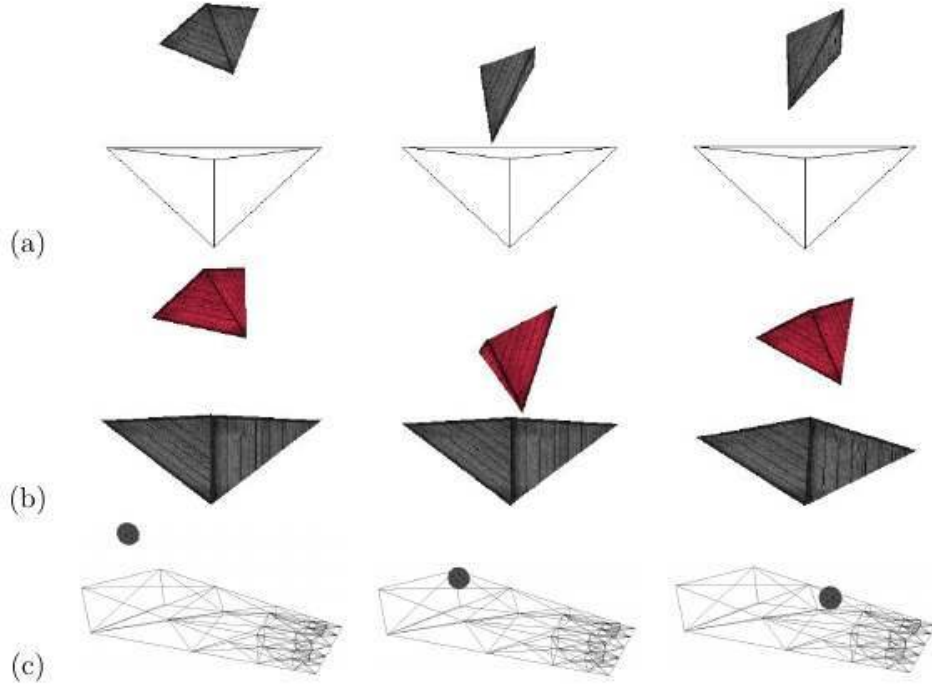


Figure 2.6: Moving tetrahedra colliding with: (a) one static tetrahedron and (b) a moving tetrahedron, and (c) images of a spherical object falling down a slope. Wireframe rendering denotes static bodies. In (a) and (b) objects are defined as: tetrahedron 1, node list: $(-1, 1, 0), (1, 1, 0), (1, -1, 0), (1, 1, -2)$; tetrahedron 2, node list: $(-2, 2, -4), (2, 2, -4), (2, -2, -4), (2, 2, -6)$.

2.3.2 Sphere falling down a slope

The movement of a sphere bouncing down a fixed slope is now investigated (Fig. 2.6(c)). The diameter of the sphere is $d = 2$ m, at an initial height of $h = 4$ m, and the slope is $a : b = 1 : 10$ (Fig. 2.7(a)). In the following tests, $g = 9.81 \text{ m/s}^2$. The sphere is a conglomerate of 106 tetrahedra, whereas the slope is represented by 115 tetrahedra. Perfectly elastic collisions are considered. The energy change during the simulation is shown in Fig. 2.7(b). When IM-Guendelman is used, the average relative error is 2.28×10^{-5} . However, for the IM-Hahn, a relative error in the energy of up to 0.45 is found.

2.3.3 Draw point

Block caving is a methodology to extract massive ore fragments from underground mines in which ore fragments collapse due to gravity and are collected through an undercutting draw point [Brown, 2003]. The gravity-driven fall of 490 ore fragments through a single draw point in a block-cave mine is modelled. The heap width is taken to be $a = 20$ m. The draw point is a cylinder of radius $r = 5.6$ m, and depth $c = 11.6$ m, and the thickness of the wall is $d = 5$ m. The radii of the ore particles is approximately 3 m, and the initial height of the heap is $b \approx 45$

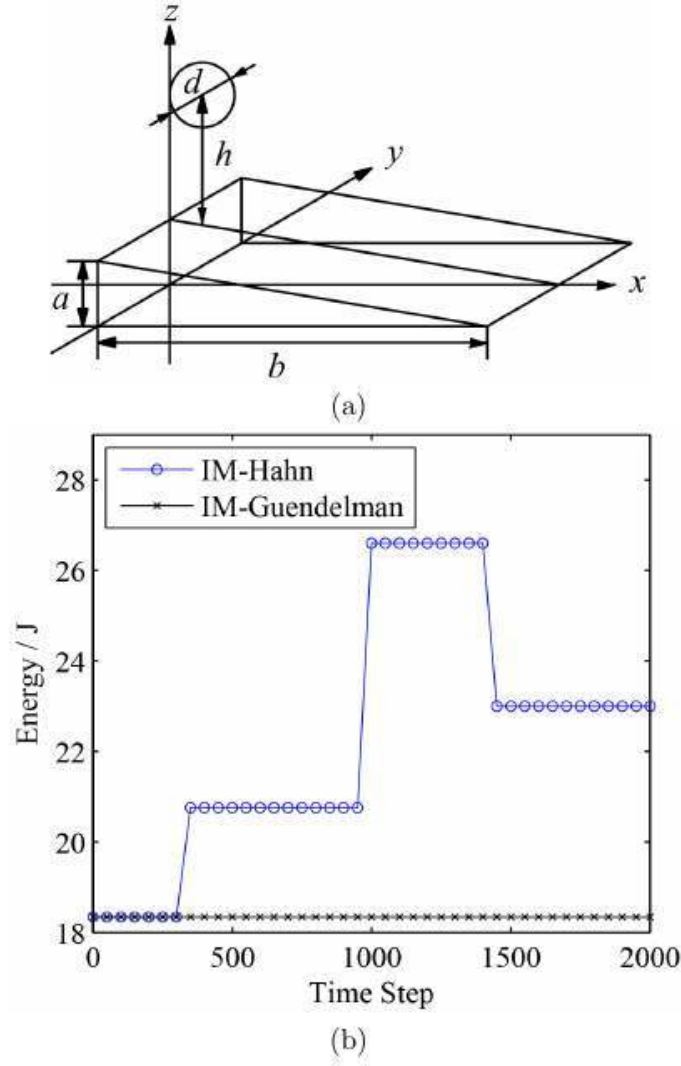


Figure 2.7: (a) Details of the slope test. (b) Energy change of a sphere bouncing down a slope.

m. To simplify the simulation model, only a quarter of the model is considered, as shown in Fig. 2.8(a). Each ore particle has the same shape and contains sixteen tetrahedra. Figure 2.9 shows the initial and final stages of the system, for $e = 0.8$ and $\mu = 0.5$. Energy values as a function of simulation progression are shown in Fig. 2.8(b) for different friction and restitution values, and in more detail in Fig. 2.8(c) for the perfectly elastic case. Results suggest that for $e = 1, \mu = 0$ energy is conservative, with a relative error of less than 1×10^{-3} . For $e \neq 1, \mu \neq 0$, the energy of the system decreases accordingly. Fig. 2.8(b) and (c) that IM-Guendelman is able to ensure energy conservation. Meanwhile, it is also observed that the energy discrepancy increases sharply when IM-Hahn is used.

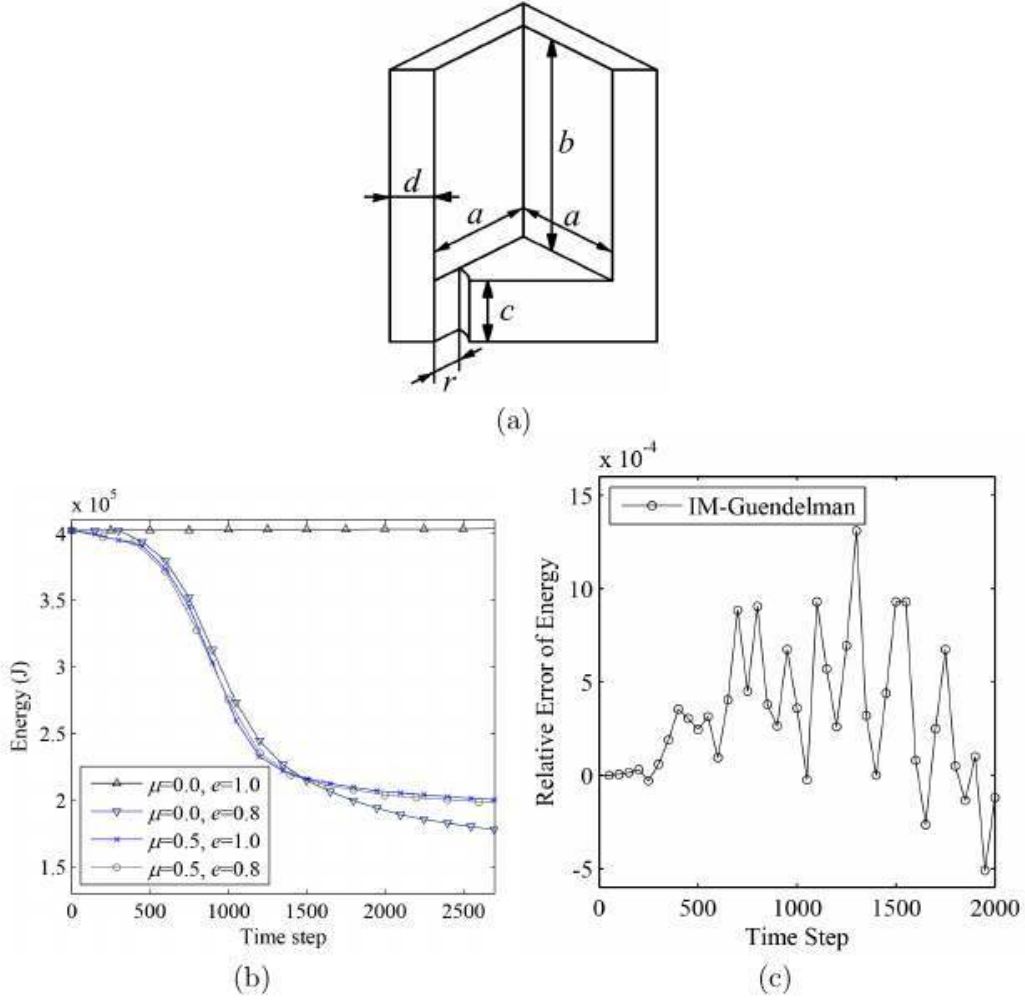


Figure 2.8: (a) Details of the single draw point test. (b) Energy curve for the system of fragments falling through a cylindrical draw point with different friction and restitution values. (c) Relative error of total energy for the case of perfectly elastic collisions.

2.4 Conclusions

In this chapter, the energy conservative property and momentum conservative property of the impulse collision method have been discussed. A necessary condition for the impulse method to be energy conservative has been proposed. According to the condition, Guendelman's impulse-based approach is energy conservative, whereas Hahn's approach does not appear to be. Both approaches were investigated numerically in terms of energy and momentum conservation. Table 2.1 shows that both approaches appear to conserve linear momentum. The impulse method conserves energy, and was shown to be a stable and fast method for collision resolution in a system of non-convex polyhedral objects.

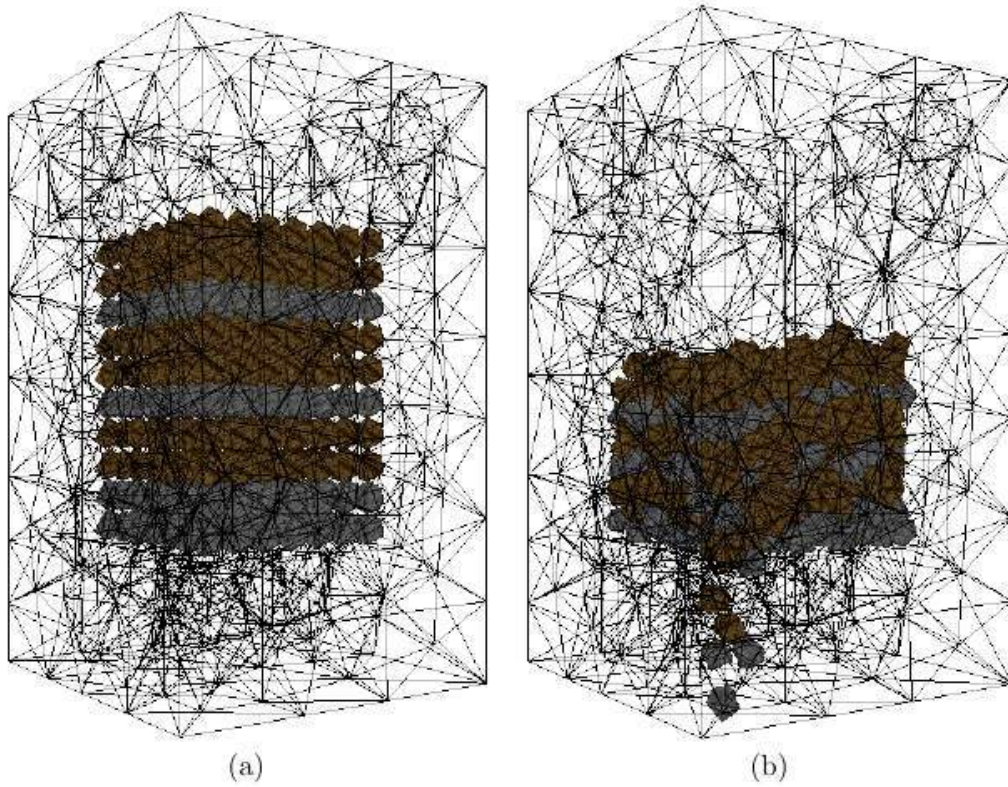


Figure 2.9: Collision of $7 \times 7 \times 10 = 490$ ores, around a single draw point, at (a) initial and (b) intermediate stage. Fragments stop falling through the draw point after $\approx 12k$ simulation steps. Bodies are conglomerates comprising 9208 tetrahedra.

3 Energy tracking method for collision response

3.1 Introduction

Even though the impulse method is able to simulate multiple collision system effectively, this method has several drawbacks. The generic simultaneous impulse method (SMM) [Coutinho, 2001] enforces constraints on all collisions by solving a system of linear equations, taking into account impulses in the normal and tangential directions, while considering friction. Due to the simultaneous evaluation of collisions, SMM cannot capture the propagation of contact forces during a collision. The sequential impulse method (SQM) treats multiple concurrent collisions one-by-one, and non-penetrating constraints are enforced to multiple collisions sequentially, iterating over the contact points until no objects collide. In SQM, Newton's impact law is locally applied to each contact, making it particularly well suited for parallelisation. A drawback of SQM is that it inverts relative normal velocities of collisions one-by-one, yielding results that depend on the order in which collisions are resolved [Erleben et al., 2005, Mosterman, 2007].

In this chapter, a generalised form of the impulse method, called the energy tracking method (ETM), is introduced. ETM addresses the individual shortcomings of SQM and SMM. It is aimed at efficiently and accurately modelling high-frequency collisions of non-convex bodies. ETM treats multiple concurrent collisions as a series of single-point collisions, and introduces an additional level of iteration at each collision time step. At the beginning of the collision analysis the relative velocity is negative, indicating that the bodies are in collision course. An impulse is applied to each body to avoid penetration, in a step-wise manner, incrementally increasing the relative velocity until it reaches zero. At each step, velocities at each collision point are updated. During this process, kinetic energy decreases and elastic energy at the contact points increases. Subsequently, the relative normal velocity continues to increase until all elastic energy is released. As opposed to the standard SQM, ETM gradually applies impulses to change the relative normal velocities of multiple collisions iteratively yet simultaneously. In contrast to SMM, ETM yields low angular velocity errors when dealing with multi-contact collisions, due to its ability to simulate the interaction of forces during collision. ETM is advantageous, as it processes collisions in a simultaneous, albeit iterative, manner. Energy is shown to be conservative for large multi-body systems of spheres, cubes and non-convex crosses. Numerical experiments evaluating the angle of repose of systems of these objects compare well with macroscopic laboratory experiments reported in the literature [Zhou et al., 2002].

3.2 Generalized form of the impulse based method

In SQM and SMM, the relative normal velocity is directly inverted. ETM gradually applies impulses to change the relative normal velocity over a set of iterations, applying impulses to multiple collisions at the same. This procedure is described for a single-point collision. In ETM, the colliding objects are assumed not to deform. A body b_N collides with another body b_M

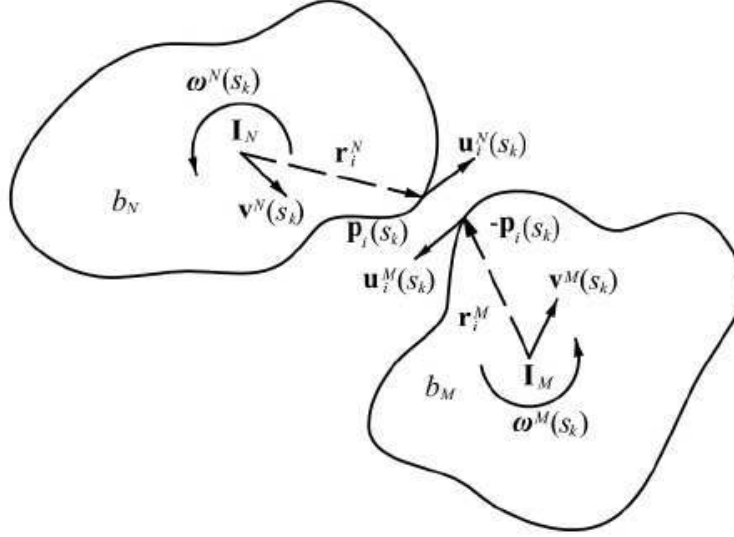


Figure 3.1: Collision between bodies b_N and b_M at iteration s_k : $\mathbf{v}^N(s_k)$ and $\mathbf{v}^M(s_k)$ are linear velocities, $\omega^N(s_k)$ and $\omega^M(s_k)$ are angular velocities, $\mathbf{r}_i^N$ and $\mathbf{r}_i^M$ are the vectors from the centre of mass of bodies to the contact point c_i , $\mathbf{u}_i^N(s_k)$ and $\mathbf{u}_i^M(s_k)$ are the velocities at the contact point, and $\mathbf{p}_i(s_k)$ is the impulse applied to the two colliding bodies to avoid penetration.

at contact point c_i (Fig. 3.1). A set of impulses, $\{\mathbf{p}_i(s_0), \mathbf{p}_i(s_1), \mathbf{p}_i(s_2), \dots\}$, is applied at the contact points to each colliding body sequentially. The subscript i in $\mathbf{p}_i(s_k)$ denotes that the impulse is applied to the contact point c_i , and s_k denotes this impulse is applied at the iteration s_k . The velocities of colliding body b_j ($j = N$ or M) at the contact point c_i at iteration s_k is denoted as:

$$\mathbf{u}_i^j(s_k) = \mathbf{v}^j(s_k) + \omega^j(s_k) \times \mathbf{r}_i^j \quad (3.1)$$

where $\mathbf{r}_i^j = \{r_{i,x}^j, r_{i,y}^j, r_{i,z}^j\}^T$ is the vector from the centre of mass of body b_j to the contact point c_i , which is assumed not to change during the collision. $\mathbf{u}_i^j(s_k)$ is the velocity of body b_j at the contact point c_i . $\mathbf{v}^j(s_k)$ is the linear velocity at the centre of mass, and $\omega^j(s_k)$ is the angular velocity. The relative velocity of two colliding bodies at contact point c_i , $\mathbf{u}_i(s_k)$, is defined as:

$$\mathbf{u}_i(s_k) = \mathbf{u}_i^N(s_k) - \mathbf{u}_i^M(s_k) \quad (3.2)$$

The change of relative velocity from iteration s_k to iteration s_{k+1} is defined as

$$\Delta \mathbf{u}_i(s_k) = \mathbf{u}_i(s_{k+1}) - \mathbf{u}_i(s_k) \quad (3.3)$$

First, an assumption for the change of relative velocity before and after applying an impulse without considering energy loss is described. Energy loss caused by the impulse in the normal direction of contact surface is handled by Stronge's hypothesis. The consideration of friction is introduced in Section 3.5 for single-point collisions and Section 3.6 for multiple collisions. When a pair of impulses of opposite signs $\mathbf{p}_i(s_k)$ and $-\mathbf{p}_i(s_k)$ is applied to colliding bodies, these impulses effectively change their linear and angular velocities,

$$\mathbf{p}_i(s_k) = m^j \Delta \mathbf{v}^j(s_k) \quad (3.4)$$

$$\mathbf{r}_i^j \times \mathbf{p}_i(s_k) = \mathbf{I}^j \Delta \boldsymbol{\omega}^j(s_k) \quad (3.5)$$

where $\mathbf{I}^j$ is the inertia tensor, which is constant during the collision, and is computed using an explicit exact expression [Tonon, 2004], and

$$\Delta \mathbf{v}^j(s_k) = \mathbf{v}^j(s_{k+1}) - \mathbf{v}^j(s_k) \quad (3.6)$$

$$\Delta \boldsymbol{\omega}^j(s_k) = \boldsymbol{\omega}^j(s_{k+1}) - \boldsymbol{\omega}^j(s_k) \quad (3.7)$$

where m^j is the mass of body b_j . Once the change of relative velocity is known, impulses are

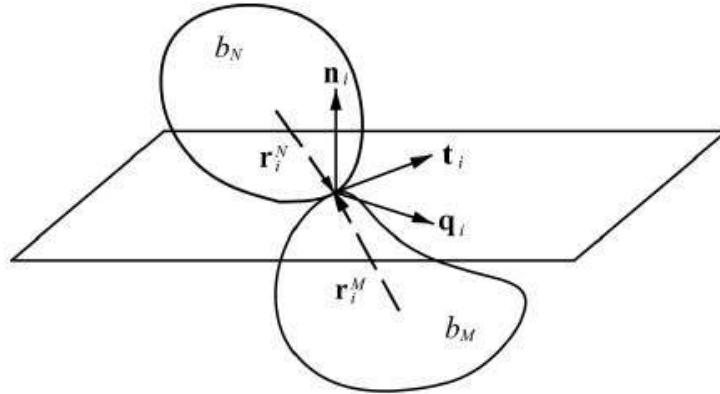


Figure 3.2: A local coordinate system $(\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i)$ is defined at the contact point between bodies b_M and b_N .

computed from the collision matrix [Mirtich, 1996],

$$\mathbf{p}_i(s_k) = \mathbf{K}_i^{-1} \Delta \mathbf{u}_i(s_k) \quad (3.8)$$

where the collision matrix $\mathbf{K}_i$ is constant, non-singular, symmetric, and positive definite. ETM applies Stronge's hypothesis to express the energy dissipation in the normal direction. Let $W_{release}$ be the work done by the normal component of the collision impulse during a collision. Then [Mirtich, 1996]

$$W_{release} = -e_n^2 W_{mc} \quad (3.9)$$

where $e_n \in (0, 1)$ is the normal coefficient of restitution, which is a measurement of bounce, such that $e_n = 1$ implies a perfectly elastic collision and $e_n = 0$ implies that relative normal velocity is reduced to zero after the collision, and W_{mc} is the work done by the normal impulse at the point of maximum compression, where the relative normal velocity changes sign. For the following analysis, a local orthogonal coordinate system in $\{\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i\}$ is defined at the locality of the contact in terms of the normal and tangential components of the collision (Fig. 3.2) in the generalised form of impulse method. It follows:

$$\mathbf{t}_i = \frac{\mathbf{n}_i \times \mathbf{p}_i(s_k)}{|\mathbf{u}_i(s_k)|} \quad (3.10)$$

$$\mathbf{q}_i = \mathbf{t}_i \times \mathbf{n}_i \quad (3.11)$$

where $\mathbf{n}_i$ is the normal direction of collision surface, and $\mathbf{t}_i$ and $\mathbf{q}_i$ are the two tangential directions at the contact point. It follows that the impulse $\tilde{\mathbf{p}}_i(s_k) = \{\tilde{p}_{i,t}(s_k), \tilde{p}_{i,n}(s_k), \tilde{p}_{i,q}(s_k)\}^T$ and relative velocity $\tilde{\mathbf{u}}_i(s_k)$ can be formulated in terms of this local coordinate system as:

$$\begin{cases} \tilde{\mathbf{u}}_i(s_k) = \mathbf{L}_i \mathbf{u}_i(s_k) \\ \tilde{\mathbf{p}}_i(s_k) = \mathbf{L}_i \mathbf{p}_i(s_k) \end{cases} \quad (3.12)$$

where $\mathbf{L}_i = (\mathbf{t}_i, \mathbf{n}_i, \mathbf{q}_i)^T$ defines the transformation from global to local coordinates.

3.3 Single-point collision: Impulse and work

The work done by contact forces, $\Delta W_i(s_k)$, is path independent and is a function of the relative velocity before and after applying the impulse, which can be represented in the form of generalised impulse method as,

$$\Delta W_i(s_k) = \frac{1}{2} (\mathbf{u}_i(s_{k+1}) + \mathbf{u}_i(s_k))^T \mathbf{K}_i^{-1} (\mathbf{u}_i(s_{k+1}) - \mathbf{u}_i(s_k)) \quad (3.13)$$

Work done by the normal and tangential impulses are considered separately.

Lemma 3.3.1. *During a collision, the work done by the impulse can be decomposed into the work done by the impulse in the normal direction, and the work done by the impulse in the tangential direction, as follows:*

$$\Delta W_{i,t}(s_k) = \frac{1}{2} (2\tilde{u}_{i,t}(s_k) + \Delta \tilde{u}_{i,t}(s_k)) \tilde{p}_{i,t}(s_k) \quad (3.14)$$

$$\Delta W_{i,n}(s_k) = \frac{1}{2}(2\tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k))\tilde{p}_{i,n}(s_k) \quad (3.15)$$

$$\Delta W_{i,q}(s_k) = \frac{1}{2}(2\tilde{u}_{i,q}(s_k) + \Delta\tilde{u}_{i,q}(s_k))\tilde{p}_{i,q}(s_k) \quad (3.16)$$

Proof. According to Eq. 3.13, the work done by contact force from iteration s_k to s_{k+1} is:

$$\Delta W_i(s_k) = \frac{1}{2}(\mathbf{u}_i(s_{k+1}) + \mathbf{u}_i(s_k))^T \mathbf{K}_i^{-1}(\mathbf{u}_i(s_{k+1}) - \mathbf{u}_i(s_k)) \quad (3.17)$$

Substituting Eq. 3.8 into the previous,

$$\Delta W_i(s_k) = \frac{1}{2}(\mathbf{u}_i(s_{k+1}) + \mathbf{u}_i(s_k))^T \mathbf{p}_i(s_k) \quad (3.18)$$

Considering the collision in local coordinate system defined by Eq. 3.10 and Eq. 3.11, it follows:

$$\Delta W_i(s_k) = \frac{1}{2}(\mathbf{L}_i^{-1}\tilde{\mathbf{u}}_i(s_{k+1}) + \mathbf{L}_i^{-1}\tilde{\mathbf{u}}_i(s_k))^T \mathbf{L}_i^{-1}\tilde{\mathbf{p}}_i(s_k) \quad (3.19)$$

$$\Delta W_i(s_k) = \frac{1}{2}(\tilde{\mathbf{u}}_i(s_{k+1}) + \tilde{\mathbf{u}}_i(s_k))^T \mathbf{L}_i^{-T} \mathbf{L}_i^{-1} \tilde{\mathbf{p}}_i(s_k) \quad (3.20)$$

Since $\mathbf{L}_i$ is an orthogonal matrix, $\mathbf{L}_i^{-1} = \mathbf{L}_i^T$,

$$\Delta W_i(s_k) = \frac{1}{2}(\tilde{\mathbf{u}}_i(s_{k+1}) + \tilde{\mathbf{u}}_i(s_k))^T \tilde{\mathbf{p}}_i(s_k) \quad (3.21)$$

and

$$\Delta W_i(s_k) = \frac{1}{2} \begin{bmatrix} 2\tilde{u}_{i,t}(s_k) + \Delta\tilde{u}_{i,t}(s_k) \\ 2\tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k) \\ 2\tilde{u}_{i,q}(s_k) + \Delta\tilde{u}_{i,q}(s_k) \end{bmatrix}^T \begin{bmatrix} \tilde{p}_{i,t}(s_k) \\ \tilde{p}_{i,n}(s_k) \\ \tilde{p}_{i,q}(s_k) \end{bmatrix} \quad (3.22)$$

Thus,

$$\Delta W_i(s_k) = \Delta W_{i,t}(s_k) + \Delta W_{i,n}(s_k) + \Delta W_{i,q}(s_k) \quad (3.23)$$

$$\Delta W_{i,t}(s_k) = \frac{1}{2}(2\tilde{u}_{i,t}(s_k) + \Delta\tilde{u}_{i,t}(s_k))\tilde{p}_{i,t}(s_k) \quad (3.24)$$

$$\Delta W_{i,n}(s_k) = \frac{1}{2}(2\tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k))\tilde{p}_{i,n}(s_k) \quad (3.25)$$

$$\Delta W_{i,q}(s_k) = \frac{1}{2}(2\tilde{u}_{i,q}(s_k) + \Delta\tilde{u}_{i,q}(s_k))\tilde{p}_{i,q}(s_k) \quad (3.26)$$

□

Lemma 3.3.2. *During a collision, if the impulse is only applied in the normal direction of the colliding surface, where $\tilde{p}_{i,t}(s_k) = 0$ and $\tilde{p}_{i,r}(s_k) = 0$ in local coordinates, then, the work done by the contact force can be expressed as a function of the change of relative normal velocity:*

$$\Delta W_i(s_k) = \frac{1}{2}(2\tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k))(H_{21}A + H_{22} + H_{23}B)\Delta\tilde{u}_{i,n}(s_k) \quad (3.27)$$

where,

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} = \mathbf{L}_i \mathbf{K}_i^{-1} (\mathbf{L}_i^{-1}) \quad (3.28)$$

$$A = -\frac{H_{23}H_{12} - H_{13}H_{32}}{H_{33}H_{11} - H_{13}H_{31}} \quad (3.29)$$

$$B = -\frac{H_{31}H_{12} - H_{11}H_{32}}{H_{31}H_{13} - H_{11}H_{33}}. \quad (3.30)$$

Proof. According to Eq. 3.31, the work done by the normal contact force is independent of the change of relative velocity in tangential directions, and can be expressed as a function of the change of relative velocity in normal direction:

$$\Delta W_i(s_k) = \frac{1}{2}(2\tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k))\tilde{p}_{i,n}(s_k) \quad (3.31)$$

To replace $\tilde{p}_{i,n}(s_k)$ by a function of $\Delta\tilde{u}_{i,n}(s_k)$, the relationship between $\tilde{p}_{i,n}(s_k)$ and $\Delta\tilde{u}_{i,n}(s_k)$ is expressed as:

$$\mathbf{p}_i(s_k) = \mathbf{K}_i^{-1} \Delta \mathbf{u}_i(s_k) \quad (3.32)$$

$$\mathbf{L}^{-1} \tilde{\mathbf{p}}_i(s_k) = \mathbf{K}_i^{-1} \mathbf{L}^{-1} \Delta \tilde{\mathbf{u}}_i(s_k) \quad (3.33)$$

$$\tilde{\mathbf{p}}_i(s_k) = \mathbf{L}_i \mathbf{K}_i^{-1} \mathbf{L}_i^{-1} \Delta \tilde{\mathbf{u}}_i(s_k) \quad (3.34)$$

where

$$\tilde{\mathbf{p}}_i(s_k) = \mathbf{H} \Delta \tilde{\mathbf{u}}_i(s_k) \quad (3.35)$$

$$\mathbf{H} = \mathbf{L} \mathbf{K}^{-1} (\mathbf{L})^{-1} \quad (3.36)$$

$$\begin{bmatrix} \tilde{p}_{i,t}(s_k) \\ \tilde{p}_{i,n}(s_k) \\ \tilde{p}_{i,q}(s_k) \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} \Delta\tilde{u}_{i,t}(s_k) \\ \Delta\tilde{u}_{i,n}(s_k) \\ \Delta\tilde{u}_{i,q}(s_k) \end{bmatrix} \quad (3.37)$$

$$\begin{bmatrix} \tilde{p}_{i,t}(s_k) \\ \tilde{p}_{i,n}(s_k) \\ \tilde{p}_{i,q}(s_k) \end{bmatrix} = \begin{bmatrix} H_{11}\Delta\tilde{u}_{i,t}(s_k) + H_{12}\Delta\tilde{u}_{i,n}(s_k) + H_{13}\Delta\tilde{u}_{i,q}(s_k) \\ H_{21}\Delta\tilde{u}_{i,t}(s_k) + H_{22}\Delta\tilde{u}_{i,n}(s_k) + H_{23}\Delta\tilde{u}_{i,q}(s_k) \\ H_{31}\Delta\tilde{u}_{i,t}(s_k) + H_{32}\Delta\tilde{u}_{i,n}(s_k) + H_{33}\Delta\tilde{u}_{i,q}(s_k) \end{bmatrix} \quad (3.38)$$

where $\tilde{p}_{i,t}(s_k) = 0$ and $\tilde{p}_{i,q}(s_k) = 0$.

$$\begin{bmatrix} 0 \\ \tilde{p}_{i,n}(s_k) \\ 0 \end{bmatrix} = \begin{bmatrix} H_{11}\Delta\tilde{u}_{i,t}(s_k) + H_{12}\Delta\tilde{u}_{i,n}(s_k) + H_{13}\Delta\tilde{u}_{i,q}(s_k) \\ H_{21}\Delta\tilde{u}_{i,t}(s_k) + H_{22}\Delta\tilde{u}_{i,n}(s_k) + H_{23}\Delta\tilde{u}_{i,q}(s_k) \\ H_{31}\Delta\tilde{u}_{i,t}(s_k) + H_{32}\Delta\tilde{u}_{i,n}(s_k) + H_{33}\Delta\tilde{u}_{i,q}(s_k) \end{bmatrix} \quad (3.39)$$

Regarding $\Delta\tilde{u}_{i,n}(s_k)$ as unknown and solving above equation, $\Delta\tilde{u}_{i,t}(s_k)$ and $\Delta\tilde{u}_{i,q}(s_k)$ can be

expressed as function of $\Delta\tilde{u}_{i,n}(s_k)$:

$$\Delta\tilde{u}_{i,t}(s_k) = -\frac{H_{33}H_{12} - H_{13}H_{32}}{H_{33}H_{11} - H_{13}H_{31}}\Delta\tilde{u}_{i,n}(s_k) = A\Delta\tilde{u}_{i,n}(s_k) \quad (3.40)$$

$$\Delta\tilde{u}_{i,q}(s_k) = -\frac{H_{31}H_{12} - H_{11}H_{32}}{H_{31}H_{13} - H_{11}H_{33}}\Delta\tilde{u}_{i,n}(s_k) = B\Delta\tilde{u}_{i,n}(s_k) \quad (3.41)$$

Then $\tilde{p}_{i,n}(s_k)$ is expressed as a function of $\Delta\tilde{u}_{i,n}(s_k)$,

$$\tilde{p}_{i,n}(s_k) = (H_{21}A + H_{22} + H_{23}B)\Delta\tilde{u}_{i,n}(s_k) \quad (3.42)$$

Substituting the previous into Eq. 3.31:

$$\Delta W_i(s_k) = \frac{1}{2}(2\tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k))(H_{21}A + H_{22} + H_{23}B)\Delta\tilde{u}_{i,n}(s_k) \quad (3.43)$$

□

By manipulating Eq. 3.43, the change of relative normal velocity is defined as follows, if the change of energy is known:

$$\Delta\tilde{u}_{i,n}(s_k) = -\tilde{u}_{i,n}(s_k) \pm \sqrt{C + \tilde{u}_{i,n}^2(s_k)} \quad (3.44)$$

$$C = \frac{2\Delta W_i}{H_{21}A + H_{22} + H_{23}B} \quad (3.45)$$

Since the relative normal velocity increases during the simulation, the positive solution is chosen:

$$\begin{aligned} \Delta u_{i,n}^{max}(s_k) &= -\tilde{u}_{i,n}(s_k) + \sqrt{C + \tilde{u}_{i,n}^2(s_k)} \\ C &= \frac{2\Delta W_i}{H_{21}A + H_{22} + H_{23}B} \end{aligned} \quad (3.46)$$

Lemma 3.2 defines a necessary condition for the impulse-based method to model an energy-conservative collision, implying that there should be no energy loss during impact, and that no artificial source or sink of energy is introduced by the numerical method. The energy-conservative collision is idealised, in that no frictional or other sources of energy loss are taken into account. However, the ability to simulate frictionless and perfectly elastic collisions remains a basic requirement for realistic impact mechanics simulation. Approaches proposed by Baraff [1994] and Guendelman et al. [2003] both satisfy this lemma, indicating that they are energy conservative, whereas the approach proposed by Hahn is not [Hahn, 1988].

3.4 Single-point collision: Friction

In ETM, the change of relative velocity is first computed in the normal direction, after which friction is considered. A static friction condition of ETM is the same as the static friction condition of the impulse method, which is defined by assuming that there is no tangential

velocity after collision, such that the change of the relative normal velocity is

$$\tilde{\mathbf{u}}_i(s_{k+1}) = \begin{bmatrix} 0 & \tilde{u}_{i,n}(s_k) + \Delta\tilde{u}_{i,n}(s_k) & 0 \end{bmatrix}^T \quad (3.47)$$

The impulse in this case is expressed as $\mathbf{p}_i(s_k) = (\mathbf{K}_i)^{-1} \Delta\mathbf{u}_i(s_k)$ (Eq. 3.8). Decomposing this impulse into its normal and tangential directions,

$$\mathbf{p}_{i,n}(s_k) = \mathbf{n}_i^T \mathbf{p}_i(s_k) \mathbf{n}_i \quad (3.48)$$

$$\mathbf{p}_{i,q}(s_k) = \mathbf{p}_i(s_k) - \mathbf{p}_{i,n}(s_k) \quad (3.49)$$

the tangential direction is

$$\mathbf{q}_i = \frac{\mathbf{p}_{i,q}(s_k)}{|\mathbf{p}_{i,q}(s_k)|} \quad (3.50)$$

where the static friction condition is defined as

$$|\mathbf{p}_{i,q}(s_k)| \leq \mu \tilde{p}_{i,n}(s_k) \quad (3.51)$$

If the static friction condition is satisfied, the above impulse is expressed by Eq. 3.8. Otherwise, the impulse is re-computed as follows, to ensure that the change of relative normal velocity is $\Delta\tilde{u}_{i,n}(s_k)$ after applying the impulses. The impulse is expressed as

$$\mathbf{p}_i(s_k) = \tilde{p}_{i,n}(s_k) \mathbf{n}_i + \mu \tilde{p}_{i,n}(s_k) \mathbf{q}_i \quad (3.52)$$

where

$$\tilde{p}_{i,n}(s_k) = \frac{\Delta\tilde{u}_{i,n}(s_k)}{\mathbf{n}_i^T \mathbf{K}_i (\mathbf{n}_i + \mu \mathbf{q}_i)} \quad (3.53)$$

Proof. The tangential impulse is derived as follows.

$$\mathbf{p}_{i,q}(s_k) = \mu \tilde{p}_{i,n}(s_k) \mathbf{q}_i \quad (3.54)$$

So

$$\mathbf{p}_i(s_k) = \mathbf{p}_{i,n}(s_k) + \mathbf{p}_{i,q}(s_k) = \mathbf{p}_{i,n}(s_k) + \mu \tilde{p}_{i,n}(s_k) \mathbf{q}_i \quad (3.55)$$

$$\mathbf{K}_i \mathbf{p}_i(s_k) = \Delta\mathbf{u}_i(s_k) \quad (3.56)$$

$$\mathbf{n}_i^T \mathbf{K}_i \mathbf{p}_i(s_k) = \mathbf{n}_i^T \Delta\mathbf{u}_i(s_k) \quad (3.57)$$

$$\mathbf{n}_i^T \mathbf{K}_i (\mathbf{p}_{i,n}(s_k) + \mathbf{q}_i \mu \tilde{p}_{i,n}(s_k)) = \Delta\tilde{u}_{i,n}(s_k) \quad (3.58)$$

$$\tilde{p}_{i,n}(s_k) = \frac{\Delta\tilde{u}_{i,n}(s_k)}{\mathbf{n}_i^T \mathbf{K}_i (\mathbf{n}_i + \mu \mathbf{q}_i)} \quad (3.59)$$

□

3.5 Energy Tracking: Single Collision

Once collisions are identified, a series of impulses is applied, to avoid penetration. Unlike SQM and SMM, ETM does not instantaneously invert the relative normal velocity directly: $\tilde{u}_{i,n}(s_1) = -e_n \tilde{u}_{i,n}(s_0)$. Instead, it updates the relative normal velocity gradually over a set of iterations, $\{s_0, s_1, s_2, \dots, s_{m_d}\}$, by applying a set of impulses, $\{\mathbf{p}_i(s_0), \mathbf{p}_i(s_1), \mathbf{p}_i(s_2), \dots\}$, sequentially. This is advantageous as it allows to track the change of relative normal velocity and elastic energy at each contact point, and the fluctuation of total and kinetic energy of system during a single collision (Fig. 3.3). The collision response is divided into three stages: compression, in which

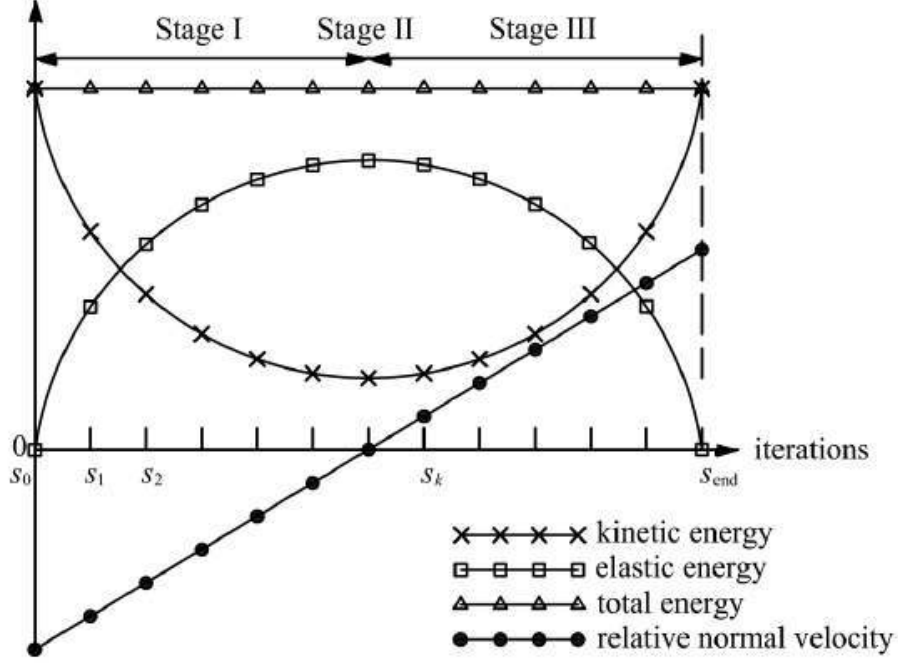


Figure 3.3: The change of energy and relative normal velocity during collision resolution for a single collision. During compression (Stage I), the relative normal velocity increases from negative to zero, elastic energy at the contact point increases, and dynamic energy decreases. During separation (Stage III), the relative normal velocity and dynamic energy increase until all elastic energy is released. The total energy is conservative. In this plot the collision is assumed to be perfectly elastic.

$\tilde{u}_{i,n}(s_k)$ is negative; restitution, the stage in which the estimated energy loss can be accounted for; and separation, in which the relative normal velocities are positive. The initial change of relative normal velocity, $\Delta \dot{u}_n$, is defined as

$$\Delta \dot{u}_n = \frac{\tilde{u}_{i,n}(s_0)}{m_d} \quad (3.60)$$

where $\tilde{u}_{i,n}(s_0)$ is the initial relative normal velocity, and m_d is a positive integer which controls the rate of change of the relative normal velocity during collision analysis. m_d constitutes

the number of iterations for impulse computation at each collision time step. Without loss of generality, the change of relative normal velocity is initially chosen as $\Delta\dot{u}_n$ for each iteration, and is recomputed at different stages to ensure the energy conservative property (Algorithm 2).

Algorithm 2 Collision response for a single collision

Require: Body b_N and body b_M collide at contact point c_i

- 1: initialise the iteration number $k = 0$
 - 2: initialise $W_{i,n}(s_0) = 0$
 - 3: calculate the initial change of relative normal velocity, $\Delta\dot{u}_n$ (Eq. 3.60)
 - 4: compute relative normal velocity before collision,
 - 5: **while** $\tilde{u}_{i,n}(s_k) < 0$ **do**
 - 6: compute relative normal velocity, $\tilde{u}_{i,n}(s_k)$
 - 7: compute $\Delta\tilde{u}_{i,n}(s_k)$ (Eq. 3.60)
 - 8: compute impulse, $\mathbf{p}_i(s_k)$ (Eq. 3.47-3.52) considering friction
 - 9: calculate work done by impulse (Eq. 3.13) and update the energy at contact point by
 $W_{i,n}(s_{k+1}) = W_{i,n}(s_k) + \Delta W_{i,n}(s_k)$
 - 10: update the linear and angular velocity of b_N and b_M by applying impulses
 - 11: $k=k+1$
 - 12: **end while**
 - 13: compute the energy dissipation using Stronge's hypothesis: $W_{i,n}(s_{k+1}) = e_n^2 W_{i,n}(s_k)$
 - 14: $k=k+1$
 - 15: **while** $W_i(s_k) > 0$ **do**
 - 16: compute the maximum change of relative normal velocity, $\Delta u_{i,n}^{max}(s_k)$, from the residual energy at the contact point (Eq. 3.46)
 - 17: compute $\Delta\tilde{u}_{i,n}(s_k)$ (Eq. 3.64)
 - 18: compute impulse, $\mathbf{p}_i(s_k)$ (Eq. 3.47-3.52) considering friction
 - 19: compute work done by impulse, W_i (Eq. 3.13) and update the energy at the contact points: $W_{i,n}(s_{k+1}) = W_{i,n}(s_k) + \Delta W_{i,n}(s_k)$
 - 20: update the linear velocity and angular velocity of colliding bodies b_N and b_M
 - 21: $k=k+1$
 - 22: **end while**
-

First stage The compression of two colliding bodies is modelled taking friction into account. The assumption of the change of relative normal velocity from iteration s_k to iteration s_{k+1} is expressed as

$$\Delta\tilde{u}_{i,n}(s_k) = \begin{cases} \Delta\dot{u}_n & \text{if } \tilde{u}_{i,n}(s_k) + \Delta\dot{u}_n < 0 \\ |\tilde{u}_{i,n}(s_k)| & \text{if } \tilde{u}_{i,n}(s_k) + \Delta\dot{u}_n > 0 \end{cases} \quad (3.61)$$

The change of relative normal velocity is defined as $\Delta\dot{u}_n$, unless the relative normal velocity becomes positive after applying an impulse. In this case, the change of the relative normal velocity is recomputed to ensure that it reaches zero after applying the impulse for a single collision. Once the assumption for the change of relative normal velocity is established, the impulse is obtained from Eq. 3.47-3.52. Work done by the impulse in normal direction, $\Delta W_{i,n}(s_k)$, is

expressed by Eq. 3.13. The elastic energy at the contact point before collision is set to zero, $W_i(s_0) = 0$. At each step, the energy due to the normal contact force at contact point c_i is updated,

$$W_{i,n}(s_{k+1}) = W_{i,n}(s_k) + \Delta W_{i,n}(s_k). \quad (3.62)$$

Second stage At the end of the first stage, the work done by the normal impulse reaches the maximum value, and the sign of the relative normal velocity changes to positive. Energy dissipation in the normal direction is expressed by Stronge's hypothesis,

$$W_i(s_{k+1}) = e_n^2 W_i(s_k). \quad (3.63)$$

Third stage Separation of two colliding bodies is modelled taking friction into account. The change of relative normal velocity from iteration s_k to iteration s_{k+1} is expressed as

$$\Delta \tilde{u}_{i,n}(s_k) = \begin{cases} \Delta \dot{u}_i & \text{if } \Delta \dot{u}_n \leq \Delta u_{i,n}^{max}(s_k) \\ \Delta u_{i,n}^{max}(s_k) & \text{if } \Delta \dot{u}_n > \Delta u_{i,n}^{max}(s_k) \end{cases} \quad (3.64)$$

The change of relative normal velocity is defined as $\Delta \dot{u}_n$. However, there is a limitation of the change of relative normal velocity as the work done by impulse should be smaller than the energy at contact point. Therefore, there is a maximum change of relative normal velocity, $\Delta u_{i,n}^{max}(s_k)$, which releases the residual energy at the contact point. This maximum change of the relative normal velocity is expressed by Eq. 3.47. In Eq. 3.52 there are two solutions for the change of relative normal velocity. Once the assumption of the change of relative normal velocity is established, the impulse is computed using Eq. 3.47-3.52. The work done by the impulse, $\Delta W_i(s_k)$, is expressed by Eq. 3.13. As in Stage I, the energy at each contact point c_i is updated at each step according to Eq. 3.62.

3.6 Energy tracking method for multiple collisions

Fig. 3.4 illustrates revolution of multiple collisions by ETM which is a generalised form of impulse based method. $\mathbf{c} = \{c_0, c_1, c_2, \dots, c_n\}$, between multiple bodies, $\mathbf{b} = \{b_0, b_1, b_2, \dots, b_m\}$, detected at a time step t_c , where m is the total colliding bodies and n is the total contact points. At each iteration, multiple collisions are treated as concurrent collisions, and impulses are computed and applied to contact points sequentially. The interaction of multiple collisions is simulated by repeating the computation until all relative normal velocities are non-negative (Algorithm 3). The computation for multiple collisions consists of three main stages.

First stage Impulses are applied to eliminate all negative relative normal velocities among contact points. At each iteration s_k , the contact point with the smallest relative normal velocity, the minimum contact point, c_{min} , is identified. If the minimum contact point is colliding (its relative normal velocity is negative), a pair of impulses, obtained using Eq. 3.47-3.52 considering

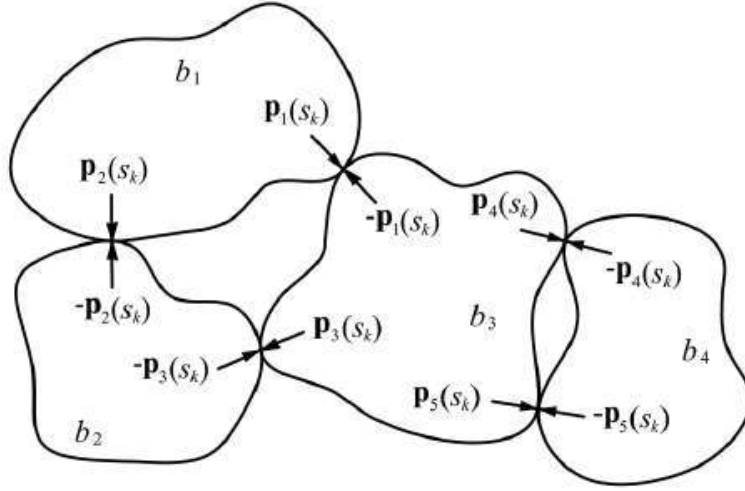


Figure 3.4: Collision of multiple bodies: multiple bodies, $\mathbf{b} = \{b_0, b_1, b_2, \dots, b_m\}$, collide with multiple contact points $\mathbf{c} = \{c_0, c_1, c_2, \dots, c_n\}$, m is the total number of colliding bodies and n is the total number of contact points. A set of impulses, $(\mathbf{p}_0(s_k), \mathbf{p}_1(s_k), \mathbf{p}_2(s_k), \dots, \mathbf{p}_n(s_k))$, are applied to contact points at iteration s_k .

friction, is applied to c_{min} to resolve the collision. The work done by normal impulses is computed using Eq. 3.13, and the energy at the contact point is updated. The linear and angular velocities of the colliding bodies, and the relative velocities of all related contact points, are also updated. The procedure is repeated until the c_{min} is not colliding. At the end of this stage, all relative normal velocities will be non-negative and the total energy among collisions reaches its maximum value.

Second stage Energy dissipation in the normal direction is applied sequentially over all collisions applying Stronge's hypothesis (Eq. 3.63).

Third stage Impulses are applied to progressively release energy at all collision points. At each iteration s_k , contact points $\mathbf{c} = \{c_0, c_1, c_2, \dots\}$ are sequentially evaluated. If the elastic $W_{i,n}(s_k)$ is positive, a pair of impulses, $\mathbf{p}_i(s_k)$ and $-\mathbf{p}_i(s_k)$ is applied to contact point c_i to release energy. This computation is repeated until the energy at all contact points is zero, i.e. $\max\{W_i\} = 0$. For each iteration s_k and each collision c_i , impulses are expressed by Eq. 3.47-3.52. The work done by normal impulses is computed using Eq. 3.13. Subsequently, the energy of the collision, and the velocities of the colliding bodies related to contact point c_i , are updated. The above three stages are repeated until no negative relative normal velocity remains. As in SQM, ETM cannot ensure termination after a certain number of iterations for perfectly elastic collisions. For oscillating cases, termination is enforced after the three stages have been repeated a finite amount of times, e.g. 300, after which the normal restitution is set to zero. Once the restitution is zero, the computation is ensured to terminate after a few

Algorithm 3 Collision response for multiple collisions

Require: A set of bodies $\{b_0, b_1, \dots, b_n\}$ collide at contact points $c_i \in \{c_0, c_1, \dots, c_m\}$

```
1: initialise  $W_i = 0$  for each contact point  $i$ 
2: identify the contact point with smallest relative normal velocity,  $c_{min}$ 
   and its relative normal velocity  $\tilde{u}_{min,n}$ 
3: while  $\tilde{u}_{min,n} < 0$  do
4:   while  $\tilde{u}_{min,n} < 0$  do
5:     compute impulses (Eq. 3.47-3.52)
6:     calculate work done by impulse (Eq. 3.13) and update the energy at the contact point
7:     update the linear and angular velocity of colliding bodies
8:     recompute the minimum contact point,  $c_{min}$  and its relative normal velocity  $\tilde{u}_{min,n}$ 
9:   end while
10:  apply Stronge's hypothesis to all the contact points
11:  search contact point with maximum elastic energy,  $\max\{W_i\}$ 
12:  while  $\max\{W_i\} > 0$  do
13:    for all  $c_i$  do
14:      compute impulses (Eq. 3.47-3.52)
15:      calculate work done by impulse (Eq. 3.13) and update the energy at the contact
        point
16:      update the linear and angular velocity of colliding bodies
17:    end for
18:    recompute contact point with maximum elastic energy,  $\max\{W_i\}$ 
19:  end while
20:  recompute the minimum contact point,  $c_{min}$  and its relative normal velocity  $\tilde{u}_{min,n}$ 
21: end while
```

iterations.

3.7 Parallel computing in ETM

Parallel computing [Chapman et al., 2007] is a methodology in which many calculations are carried out simultaneously and aims to reduce computational time. ETM resolves collision responses locally at every contact point, thus, it is particularly well suited for parallelization. In the present work, the resolution of collision response is parallelized using OpenMP (Open Multi-Processing) [Chapman et al., 2007], a shared-memory application programming interface supporting multi-platform shared memory multiprocessing programming in C++.

The three dimensional domain is divided into four parallel domains for parallel computing and one overlap domain, which is used to coordinate different parallel domains, see Fig. 3.5. First, a local coordinate system is created, whose origin is at the geometry center of all moving bodies and whose axes are parallel to the axes of global coordinates. Then, the computation domain is simply divided into four parallel domains and one overlap domain using the four quadrants of the local coordinates. The colliding bodies whose nodes are all in quadrant i ($i=1, 2, 3, 4$) are inserted into a set PB_i ($i=1, 2, 3, 4$). The colliding bodies whose nodes are in

different quadrants are inserted into a set PB_o . The contact points, which are on the colliding bodies of PB_o , are inserted to a set PC_o . The contact points, which are on the colliding bodies of PB_i and are not on the colliding bodies of PB_o , are inserted into a set PC_i ($i=1, 2, 3, 4$). The influence domain of the contact points in the set PB_i is called the parallel domain i . The influence domain of the contact points in the set PB_i is called overlap domain. It means that changing the state of colliding bodies in parallel domain i will only influence the colliding bodies in parallel domain i , not influence the colliding bodies in other parallel domains. Meanwhile, changing the state of the colliding bodies in overlap domain probably influence the colliding bodies in the four parallel domains.

The resolving of collision responses is parallelized in the three stages of the ETM, as shown in Algorithm 4. At the first stage of ETM, the first step is to apply impulses to eliminate negative relative velocity in four parallel domain, which can be parallelized. The second step is to eliminate the negative relative velocities the overlap domains. Because the colliding bodies in overlap domain probably share contact points with the colliding bodies in parallel domains. When the state of colliding bodies in the overlap domain is changed, negative relative velocities might appear again in parallel domains again. At this time, the first step and the second step are repeated until there is no negative relative velocity in the entire computational domain. When restitution $e < 1$, the convergence of computational is ensured. At the second stage of ETM, the parallel computation is much simpler. Stronge's hypothesis is applied to the contact points in different parallel domains and overlap domain, which can be parallelized as the operations on each contact point do not depend on each other.

At the third stage of ETM, in which impulses are applied to release all the elastic energy at collisions, first impulses are applied to release energy in four different parallel domains, which can be parallelized as the operations on each contact point do not depend on each other. Then, impulses are applied to release energy in overlap domain separately. As shown in Fig. 3.6, the computational domain of the packing and repositioning of 192 concave concrete crosses, which is introduced in Section 3.8.5 (see Fig. 3.18 (c)), is divided into four parallel domains and one overlap domain. In the entire computational domain, there are 193 colliding bodies, including the boundary wall. The numbers of crosses in four parallel domains are 23, 20, 14, 24, respectively. The number of crosses in the overlap domain is 105. Additionally, there are seven separate crosses which do not share contact points with others, which are not considered in the calculation. The crosses in the four parallel domains are colored with blue, green, red and dark gray, respectively. The crosses in the overlap domain and seven separate crosses are colored with light gray and brown, respectively.

The computational efficiency of serial and parallel computing is compared in Fig. 3.7. This test is run on a personal computer which uses intel(R) Core(TM) i3 CPU with 2.13GHz, 4GB installed memory, and four cores. Computational cost of parallel computing is not constantly cheaper than serial computing. However, average computational time of parallel computing is less than that of serial computing. Average computational time to calculate impulses using

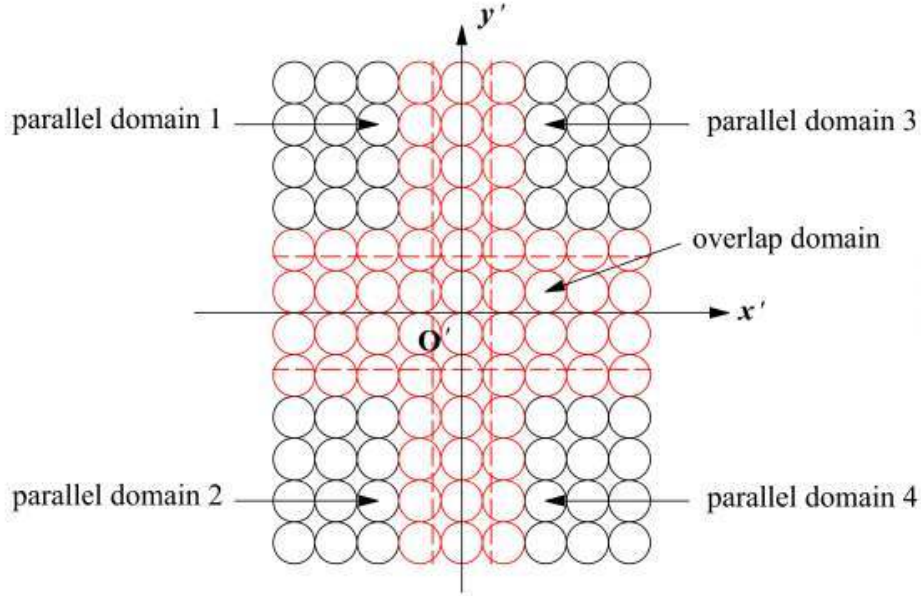


Figure 3.5: Three dimensional computational domain division for parallel computing.

serial computing and parallel computing are 10.65 s and 10.32 s respectively. The computational time is not reduced by four times, which is the number of the parallel domains. The reasons might be that number of colliding bodies in the overlap domain is relatively huge and the numbers of colliding bodies in the four parallel domain are too different. However, the overall computational cost of parallel computing is cheaper than that of serial computing.

The computational domain for the gravity flow of ore fragments in block caving system, introduced in Section 6.3.1 (see see Fig. 6.13 and Fig. 6.14), is divided into four parallel domains and one overlap domain. As shown in Fig. 3.8. There are totally 3757 cuboid ore fragments in the whole computational domain, including the one static boundary wall and static cuboid ore fragments on the boundary. The numbers of ore fragments in four parallel domains are 557, 507, 545 and 516 respectively. The number of ore fragments in the overlap domain is 1280. The number of separate ore fragments, which do not collide with others, is 352. The ore fragments in the four parallel domains are colored with blue, red, drak gray and green respectively. The ore fragments in the overlap domain and the separate ore fragments are colored with light gray and brown respectively.

3.8 Numerical tests

ETM is investigated using selected numerical tests. The collision of two tetrahedra validates the performance of ETM for single-point collisions. A series of multi-contact scenarios are selected to exhibit ETM advantages as compared to SQM and SMM. In some cases gravity is disregarded, certain examples are assumed perfectly elastic to evaluate possible artificial energy losses and

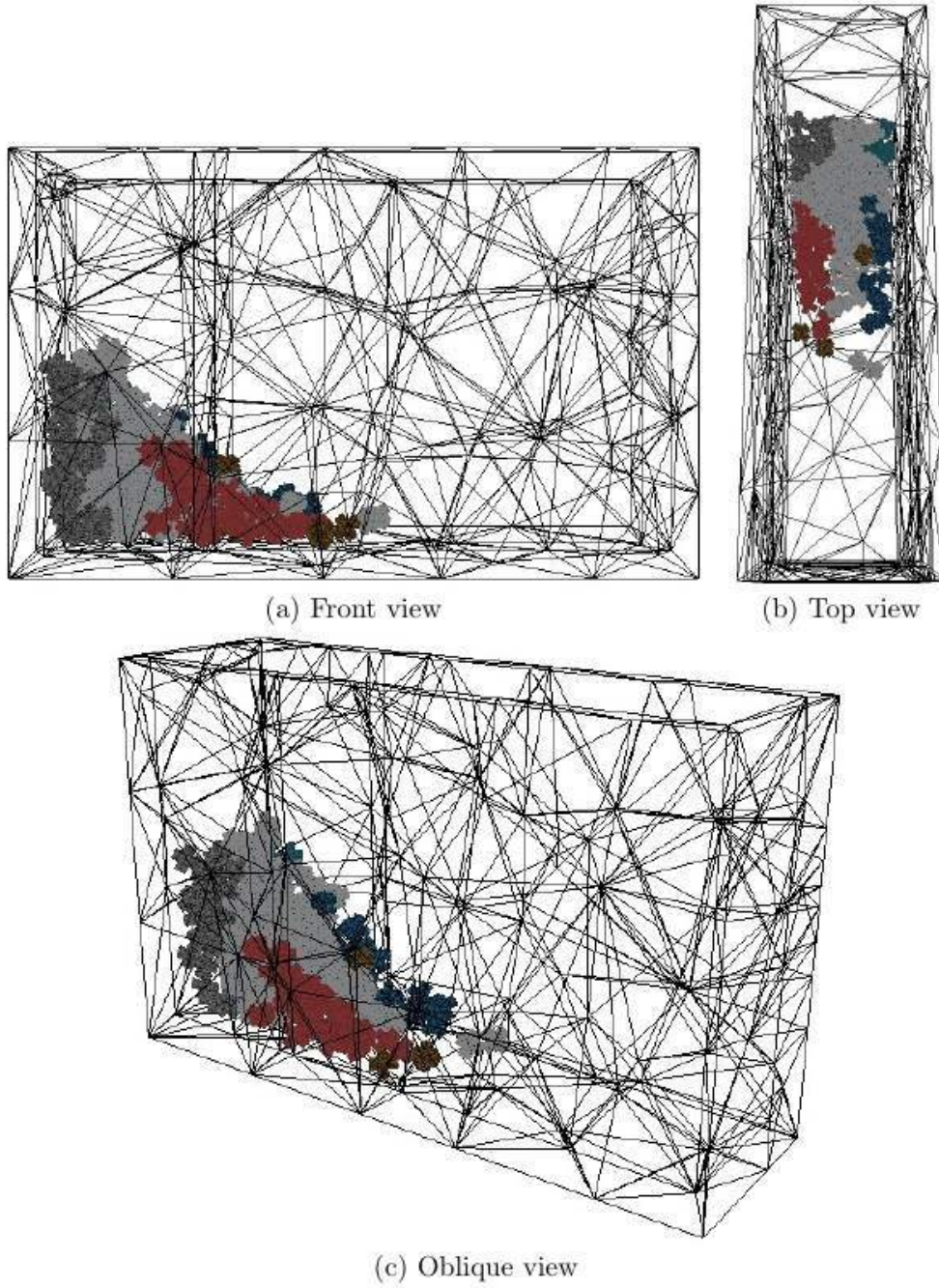


Figure 3.6: The computational domain of packing and repositioning of 192 concave concrete crosses (see Fig. 3.18 (c)) is divided into four parallel domains and one overlap domain. There are totally 193 colliding bodies, including the boundary wall. The numbers of colliding bodies in four parallel domains are 23, 20, 14, 24 respectively. The number of colliding bodies in the overlap domain is 105. The number of separate crosses, which do not collide with others, is seven. The crosses in the four parallel domains are colored with blue, dark gray, red and green respectively. The crosses in the overlap domain and seven separate crosses are colored with light gray and brown respectively.

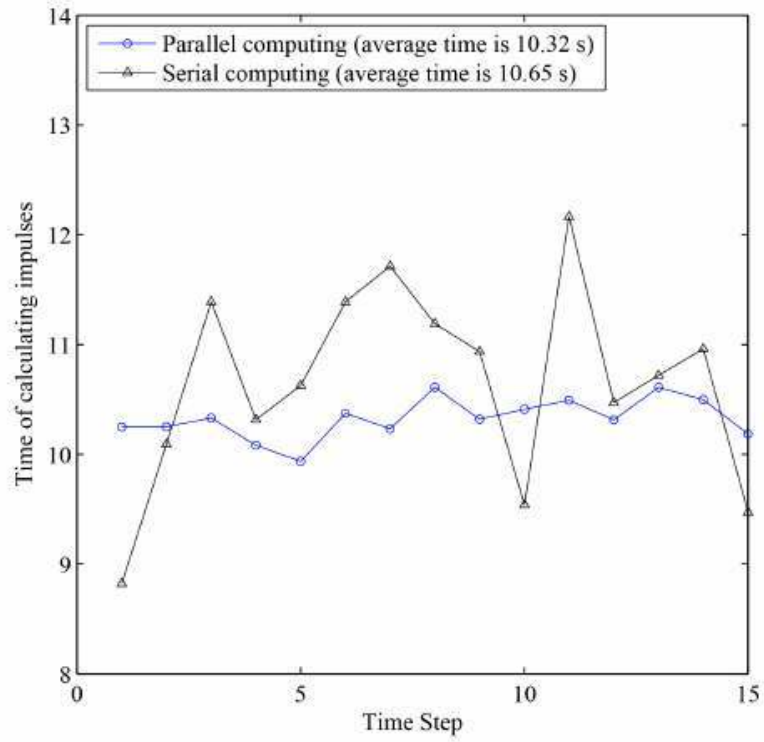
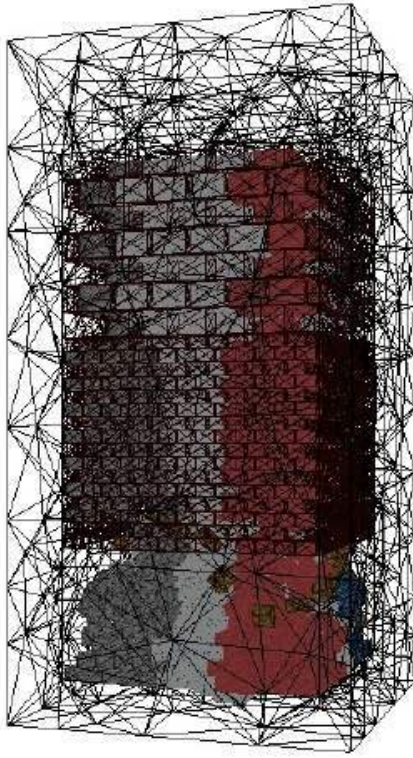
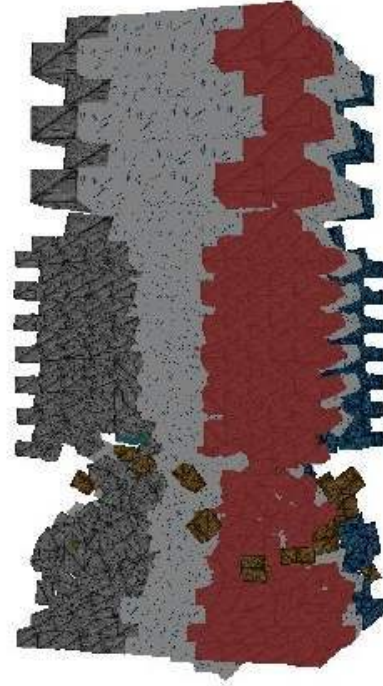


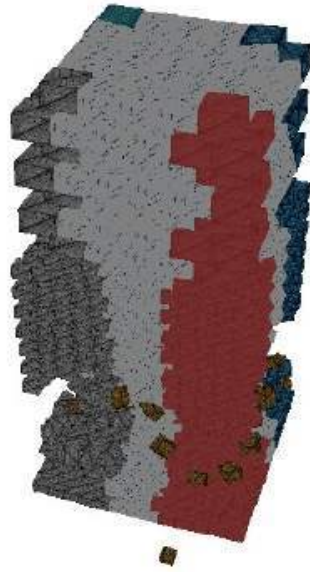
Figure 3.7: Compare computational costs of serial computing and parallel computing for the packing and repositioning of concave concrete crosses. The average computational time to calculate impulses using serial computing and parallel computing are 10.65 s and 10.32 s respectively.



(a) Front view with static boundary wall



(b) Front view without static boundary wall



(c) Top view without static boundary wall

Figure 3.8: The computational domain for the gravity flow of ore fragments in block caving system (see see Fig. 6.13 and Fig. 6.14) is divided into four parallel domains and one overlap domain. There are totally 3756 cuboid ore fragments, including static ore fragments on the boundary. The numbers of ore fragments in four parallel domains are 557, 507, 545 and 516 respectively. The number of colliding bodies in the overlap domain is 1280. The number of separate ore fragments, which do not collide with others, is 352. The ore fragments in the four parallel domains are colored with blue, red, dark gray and green respectively. The ore fragments in the overlap domain and separate ore fragments are colored light gray and brown respectively.

Algorithm 4 Parallel computing for multiple collision.

Require: A set of bodies $\{b_0, b_1, \dots, b_n\}$ collide at contact points $c_i \in \{c_0, c_1, \dots, c_m\}$, contact points in overlap domain PC_o , and contact points in parallel domains PC_i ($i=1, 2, 3, 4$).

- 1: Identify the contact point with smallest relative normal velocity, c_{min}
and its relative normal velocity $\tilde{u}_{min,n}$
 - 2: **while** $\tilde{u}_{min,n} < 0$ **do**
 - 3: **Parallel computing** on contact points in different parallel domains PC_i ($i=1, 2, 3, 4$)
 - 4: Eliminate all the negative relative velocities at contact points in parallel domains i .
 - 5: **End parallel computing**
 - 6: Eliminate all the negative relative velocities in overlap domain.
 - 7: Renew c_{min} and $\tilde{u}_{min,n}$.
 - 8: **end while**
 - 9: **Parallel computing** on contact points in different parallel domains PC_i ($i=1, 2, 3, 4$) and overlap domain.
 - 10: Apply Stronge's hypothesis to the contact points in different parallel domains and overlap domain.
 - 11: **End parallel computing**
 - 12: **Parallel computing** on contact points in different parallel domains PC_i ($i=1, 2, 3, 4$)
 - 13: Release elastic energy at contact points in parallel domains i .
 - 14: **End parallel computing**
 - 15: Release elastic energy at collisions in overlap domain.
-

gains, and in all cases the time step size is taken to be 1/600 seconds. Energy conservation and repose angles are evaluated for a series of packing and repositioning numerical tests involving spheres, cubes, and non-convex cross objects.

3.8.1 Single-point collision test

The energy tracking method is investigated using a single-point collision of two tetrahedra, a and b (Fig. 3.9). Initial linear and angular velocities of body a are $\mathbf{v}_a = (0, 0, -100)$ m/s and $\boldsymbol{\omega}_a = (5, 10, 3)$ rad/s, respectively. Body b is initially static, and only moves after the collision. No external forces act on these bodies. In this test, the collision is assumed to be perfectly elastic and frictionless, i.e. $e_n = 1$ and $\mu = 0$. The experiment constitutes a simple closed system, which is expected to be energy conservative. The progression of the relative normal velocity, elastic and kinetic energy at the contact point, and the total system energy are plotted Fig. 3.10. There are a total of 200 iterations during the collision. During the simulation of collision, the total energy of the system remains constant. During the first stage, the relative normal velocity changes from -211.84 m/s to 0 m/s, the elastic energy at the contact point increases from zero to 60124 J, and the kinetic energy decreases from 83809.2 J to 23685.2 J. During the third stage, the relative normal velocity changes from zero to 211.84 m/s, the elastic energy at the contact point decreases from 60124 J to zero, and the kinetic energy decreases from 23685.2 J to 83809.2 J. Only a single contact point is taken into account here. The influence of m_d is investigated by measuring the relative error of total energy. A

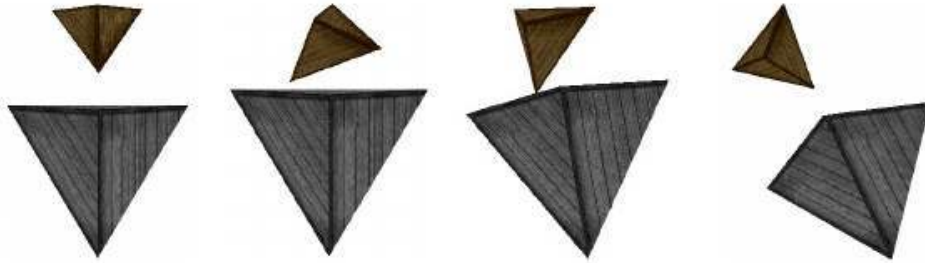


Figure 3.9: Collision of two tetrahedra.

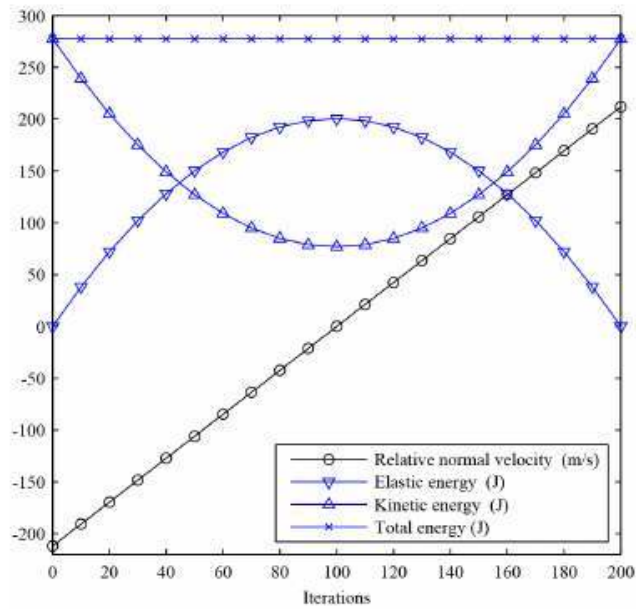


Figure 3.10: Progression of relative normal velocity, and the elastic, kinetic and total energy of a system of two colliding tetrahedra. All but the relative normal velocity are scaled by 1/300 to facilitate plotting.

variation of the error is measured for increasing m_d values of $\{1, 5, 10, 100, 200, 500\}$, yielding errors, $\xi_{E_t} = \{1.73 \times 10^{-16}, 6.94 \times 10^{-16}, 1.21 \times 10^{-15}, 1.73 \times 10^{-16}, 1.51 \times 10^{-14}, 1.42 \times 10^{-14}\}$ that are initially close to the numerical epsilon but increases by two orders of magnitude for 500 iterations. The consistently low errors indicate that the single collision can be resolved exactly. The increase of this error, with increasing m_d , exhibits the accumulation of the numerical error due to the increased amount of iterations and operations. This numerical error is also observed to be low.

3.8.2 Energy conservation

The energy conservative property of multiple bodies is investigated. A necessary condition for the impulse-based method to be energy-conservative has been previously identified [Tang et al., 2013]. Namely, and particularly in the context of an SQM formulation, the relative contact velocity must be exactly reversed, as opposed to only being inverted in the normal direction. In contrast, ETM ensures conservation by constraining iterations based on the tracking of energy during collision resolution. This property is illustrated by quantifying artificial energy loss and gain during collision resolution, in terms of the measurement of the relative error in the computation of energy conservation of the collision. To this end, two perfectly elastic cases are examined, in which 540 and 1080 spherical bodies fall into a static box. Relative energy errors for at each collision step, over 250 time steps, are plotted in Fig. 3.13. Overall, relative errors for both models are consistently below 8×10^{-3} , for both SQM and ETM.

3.8.3 Comparison with SMM

SMM inserts all collisions into a system of linear equations, so that constraints on the relative normal velocities, $\tilde{u}_{i,n}(s_1) = -e_n \tilde{u}_{i,n}(s_0)$, are enforced simultaneously. $\tilde{u}_{i,n}(s_0)$ and $\tilde{u}_{i,n}(s_1)$ are the relative normal velocities before and after collision, respectively. In some cases, this empirical velocity constraint does not match real physical behaviour. The example investigated in this section is one of them. Consider the collision of two box-shaped bodies, where one is static and the other dynamic, and gravity acceleration is ignored. The initial linear velocity of the cube is $(0, 0, -10)$ m/s, and its angular velocity is zero. The moving cube is expected to rotate after collision, as all of its contact points are located on one half of the contact surface, and are repelled upwards after the collision, generating a positive rotational moment that throws the body out of balance. For SMM, since relative normal velocities at all contact points are the same before collision, $\tilde{u}_{i,n}(s_0) = -10$ m/s, then, relative normal velocities at all contact points are the same after collision, $\tilde{u}_{i,n}(s_1) = 10$ m/s. Therefore, SMM does not model the rotation which is a function of the propagation of the impulses during collision (Fig 3.11). Angular velocities in x , y and z directions of the SQM and energy tracking method are plotted in Fig. 3.12. SMM fails to predict post-collision non-zero angular velocities. In contrast, ETM predicts angular velocities that are consistent with the expected rotation.

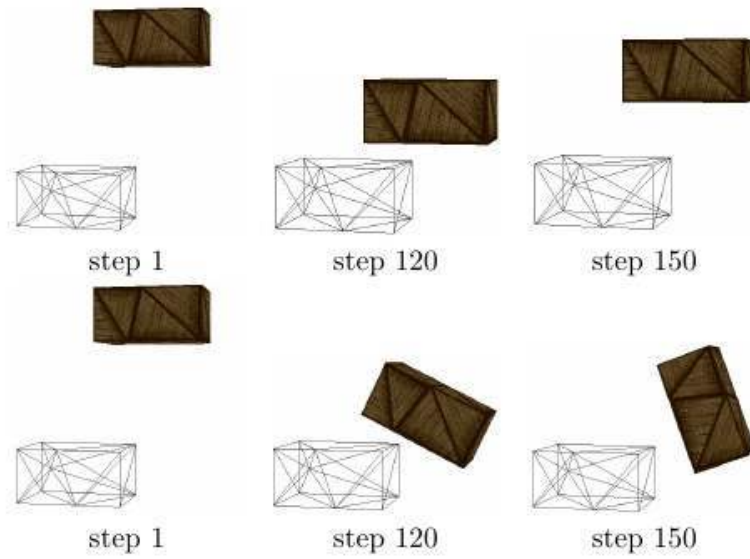


Figure 3.11: Comparison of SMM and ETM: Collision of a moving body and a static body: Multiple contact points.

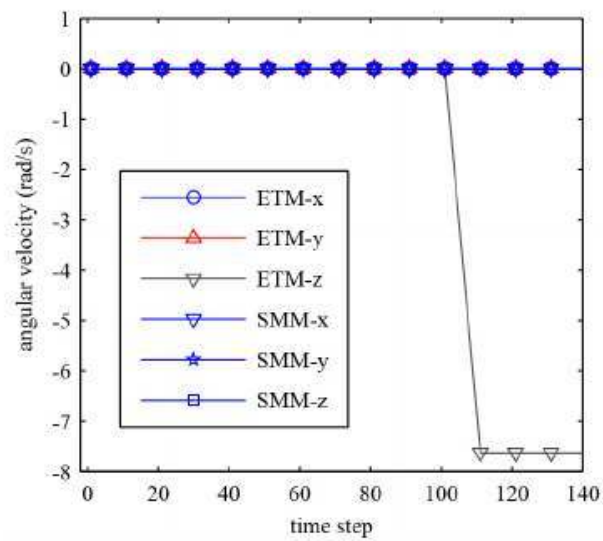
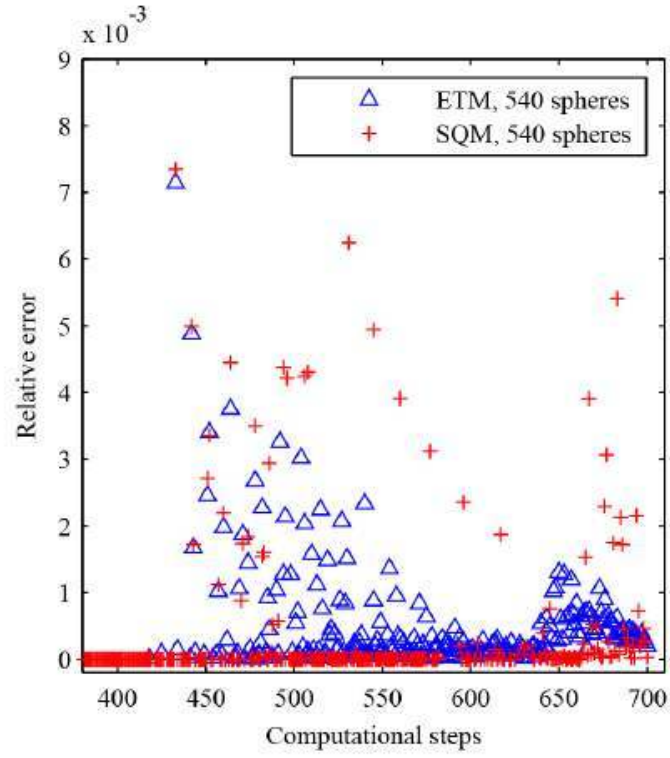
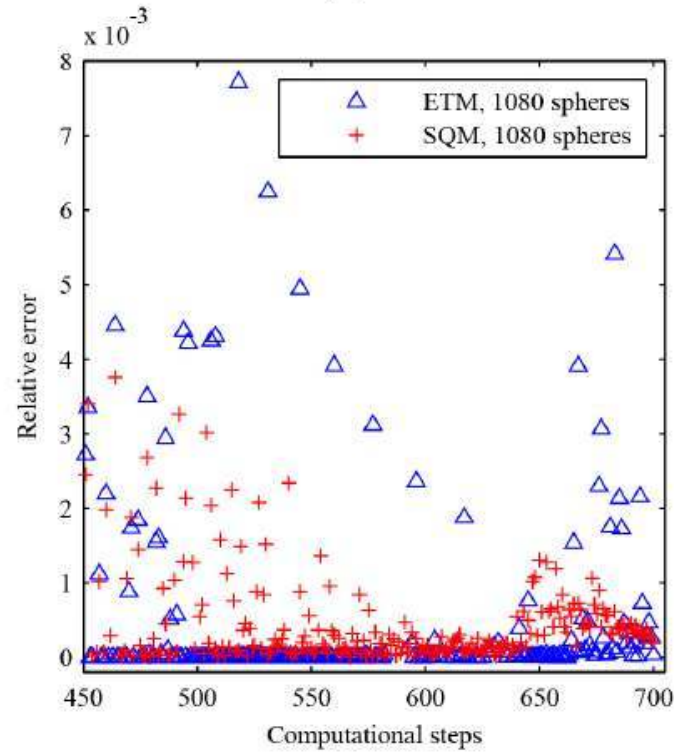


Figure 3.12: Comparison of SMM and ETM: Angular velocities of a moving body colliding with ground.



(a)



(b)

Figure 3.13: Relative error of energy during the packing of (a) 540 and (b) 1080 bodies.

3.8.4 Comparison with SQM

The collision of a $2 \times 2 \times 1$ m box-shaped body and the ground is investigated. Gravitational acceleration is ignored, and the collision is assumed to be perfectly elastic. The initial linear and angular velocities are $(1, 1, -10)$ m/s and zero, respectively. Since there is no initial angular velocity of the cube before the collision, and the ground surface is perfect horizontal, the linear and angular velocities after the collision should be exactly inverted. SQM treats multiple collisions as concurrent collisions and simply inverts the relative normal velocity after applying the impulse. Thus, impulses are applied to collisions one by one, and not simultaneously. In contrast, ETM applies the impulses to multiple collisions simultaneously, but does so by gradually increasing the relative normal velocity during the collision. In this numerical experiment, multiple collisions occur simultaneously between the box-shaped object and the ground, exposing the difficulty of SQM to treat impulse propagation during collisions as compared to ETM. The object collides with the ground at time step 166 (Fig. 3.14). The relative error of the linear velocities constantly remains close to zero, and does not vary with m_d for either method. The relative error of the angular velocity is lower for ETM as compared to SQM from the onset, and further decreases as m_d increases (Fig. 3.15). For higher values of m_d , i.e. $m_d > 50$, the accumulated numerical error becomes measurable and the relative velocity error does not further reduce. In the simulations herein we consider $m_d = 30$.

3.8.5 Packing and repositioning of multiple bodies

The experiment models the interaction between multiple bodies of different shapes, including 326 cubes, 541 spheres, and 192 concave units, discretised using a tetrahedral mesh (Fig. 3.16). Cubes and spheres are typical convex proxies for rock fragments, while specialised three-dimensional concrete crosses are widely used in coastal structure engineering to protect the coast against wave action. Gravitational acceleration is assumed to be -9.8 m/s². Spheres, cubes, and concave units are initially set up in an array, randomly oriented about their axis. These undergo gravity-driven fall and settle into a box with static boundary walls (Fig. 3.17).

The ETM is used to investigate the repose angle of the slope generated by removing the right boundary of the previously generated stacks (Fig. 3.18). Snapshots of the final repose angles for different object shapes and for different frictions are shown in Fig. 3.19. To measure the angle of repose, the floor of the container is subdivided using an 8×16 grid, where the longest side of the container is subdivided sixteen times. Eight lines are fitted, using least squares, through the highest nodes identified in each of the sixteen cells. The slopes of these are regarded as the repose angles in these eight zones (Fig. 3.20). The average repose angle corresponds to the average measured slope. The absolute error is the average difference of the average repose angle as compared to the maximum and minimum measured repose angles. For the spheres, the absolute error ranges from 0.0255 to 0.8 degrees, for cubes from 1.29 to 3.17 degrees, and for crosses from 1.2 up to 12.89 degrees. The angle of repose and respective measurement error is plotted as a function of the friction coefficient in Fig. 3.21. For comparison, angles of repose

obtained experimentally and numerically for dry sand [Nakashima et al., 2011] are plotted along the numerical results. Results are in good agreement with laboratory experiments, which show that the angle of repose increases non-linearly with the friction coefficient [Zhou et al., 2002]. The discrete element method (DEM) has been used to investigate the numerical methodologies to design concrete armour units [Rogers et al., 1996, Latham et al., 2008a]. A numerically generated repose angle for the same cross, computed using 3D DEM [Latham et al., 2008a, Munjiza et al., 2010], is also plotted for comparison. As expected, crosses exhibit more interlocking than spheres and cubes and yield, almost consistently, higher repose angles. For all shapes, repose angles are close to, but greater than measured values for dry sand. Results obtained by ETM are comparable those reported in [Latham et al., 2008a].

3.9 Conclusions

A novel energy tracking method is proposed to solve collision responses in a multi-body system. The ETM belongs to the family of impulse methods, which apply impulses to avoid penetration. In previous versions of the impulse method, i.e. SMM and SQM, the relative velocity at the contact points after collision is directly derived from the relative velocity before collision - in a purely simultaneous or sequential manner, and the impulse is a function of the change of relative normal velocity. In this generalised form of the impulse method, the relative velocity is adjusted gradually over a set of iterations, within each of these iterations their effect is considered simultaneously. By applying velocities gradually to the contact points, the impulses are able to influence multiple contact points at the same time, enabling propagation of impulse effects, and thereby improving the overall ability of the method to capture propagation of forces during dynamics simulations. Furthermore, ETM ensures that no artificial energy loss or gain results from the resolution of the collisions, setting the grounds for a more rigorous analysis of energy lost due to other factors, such as brittle deformation, during impact.

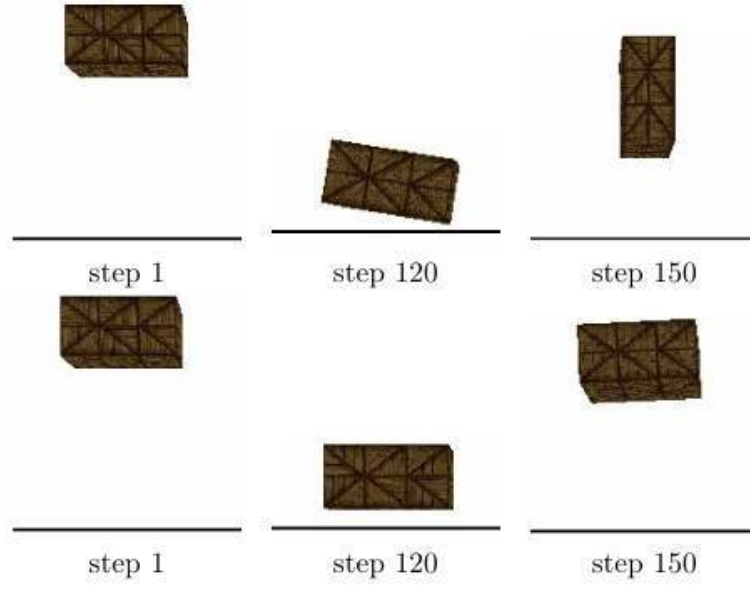


Figure 3.14: Comparison of SQM and ETM: Collision of a moving body and a static body with multiple contact points.

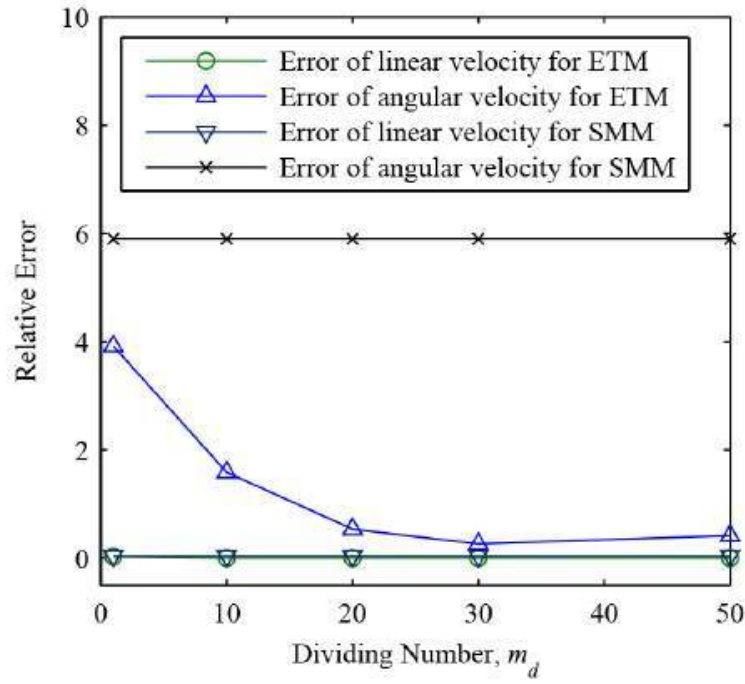


Figure 3.15: Comparison of SQM and ETM: Angular velocities of moving body colliding with ground.

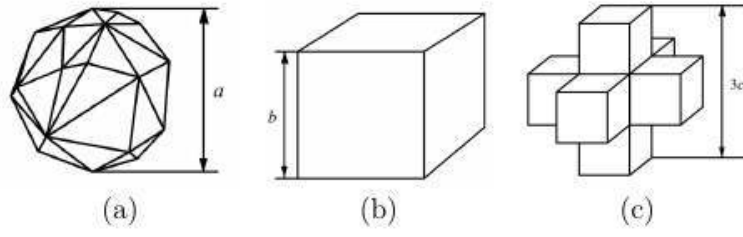


Figure 3.16: Geometry of colliding bodies: (a) sphere, where $a=3$ m, (b) cube, where $b=2.4$ m, and (c) cross, where $3c=3.78$ m.

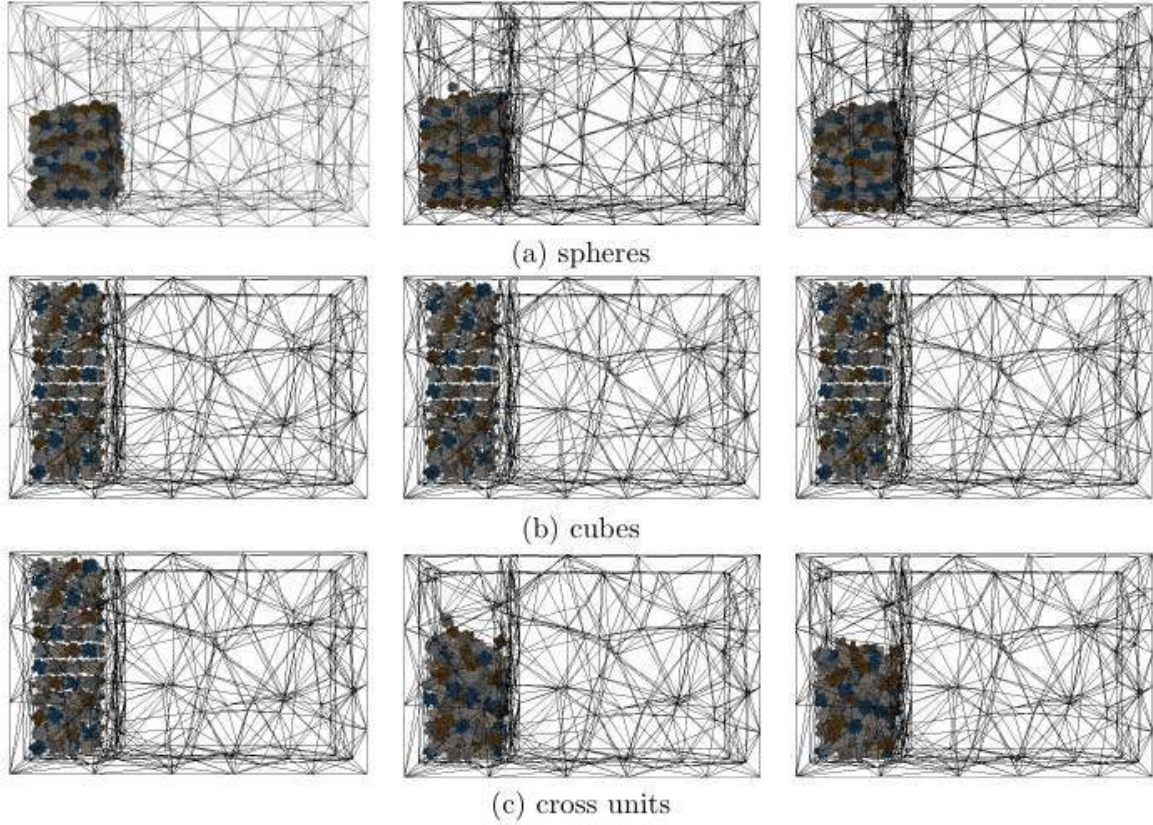


Figure 3.17: Packing of multiple bodies with different shapes, $\mu = 0.3$ and $e_n = 0.3$. In (a), 540 spheres are packed. In (b) and (c), 325 cubes and non-convex cross units are packed. The box is subdivided by a wall, which is later removed to model repositioning. Objects are coloured randomly, and the container mesh is displayed using a wireframe.

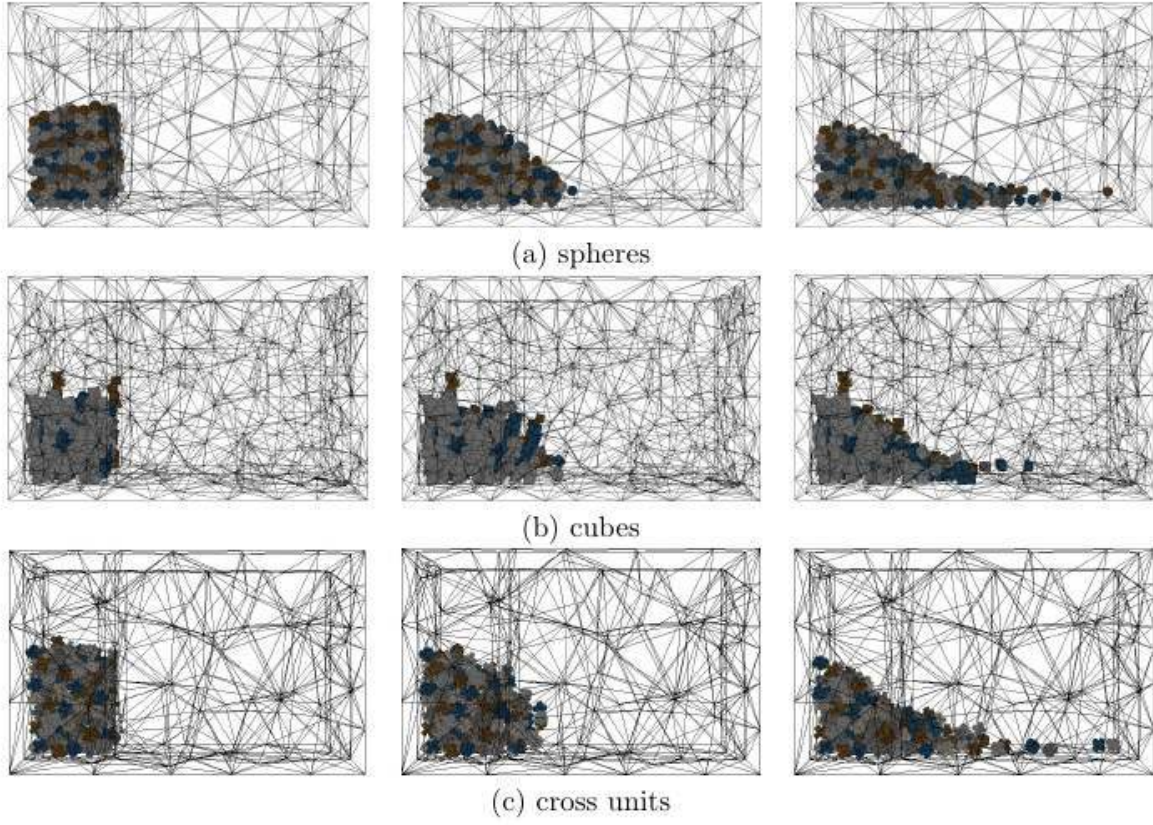


Figure 3.18: Settling of multiple bodies with different shapes, $\mu = 0.64$ and $e_n = 0.64$.

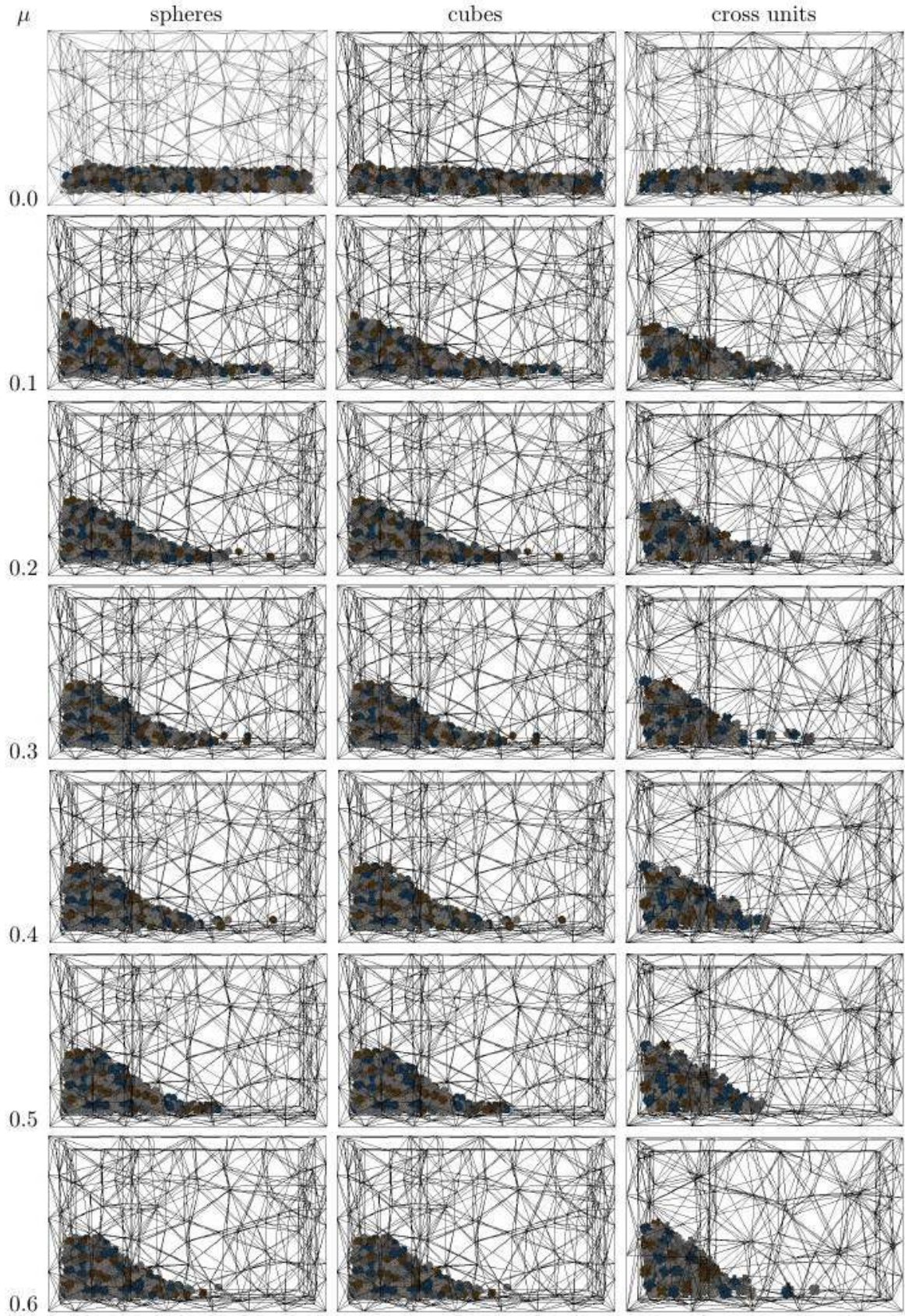


Figure 3.19: Repositioning of multiple bodies with frictions ranging from 0.05 up to 0.6. Normal restitution is assumed $e_n = 0.3$.

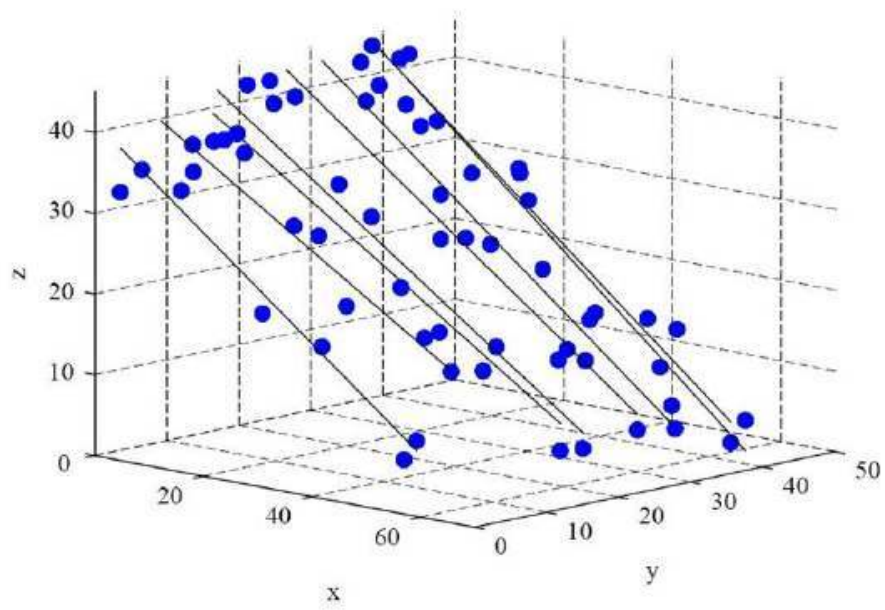
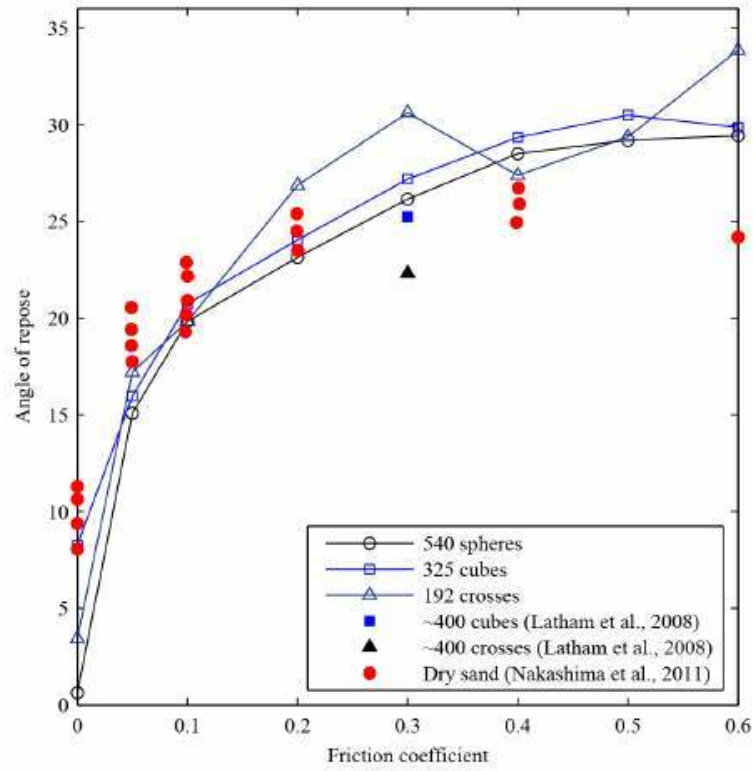
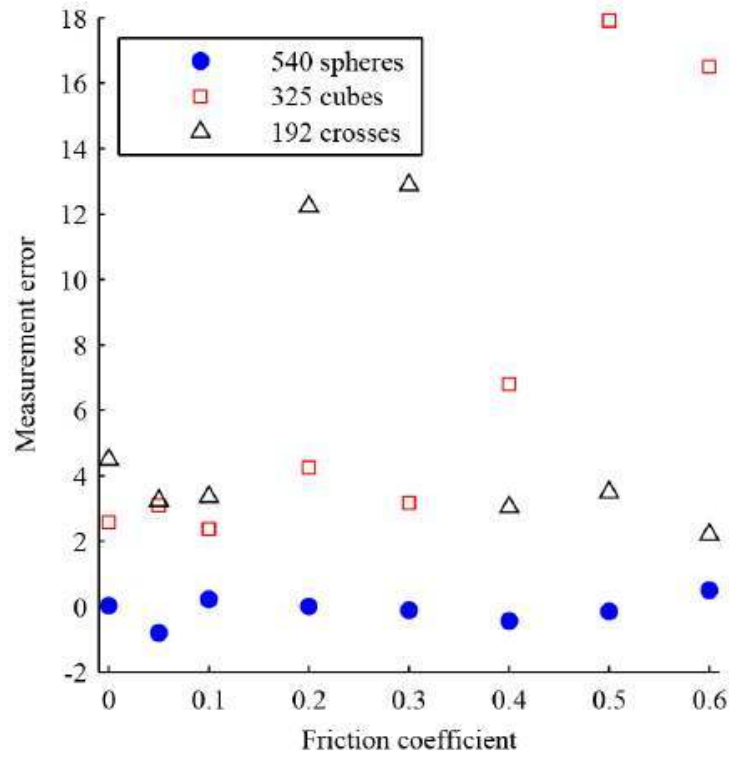


Figure 3.20: Eight lines are fitted through the ensuing dataset to estimate the repose angle.



(a)



(b)

Figure 3.21: Friction dependent angle of repose. (a) Variation for each shape as compared to experimental and numerical results reported in the literature, (b) measurement error for each case.

4 Numerical modelling of fragmentation

4.1 Introduction

This section discusses methods aimed at simulating fragmentation. Namely, the discrete element method (DEM), is often used to model multi-body interaction, and fragmentation, and the hybrid finite/discrete element method, which combines the finite and discrete element methods to capture fragmentation.

FEM with advanced re-meshing technology The finite element method has the ability to handle material heterogeneity, nonlinear behaviour, and boundary conditions [Zienkiewicz and Taylor, 2000, Banks-Sills, 1991]. In FEM, a mesh is used to discretise the computational domain for which a discretised set of equations can be solved. To simulate the propagation of cracks, there are main two challenges: to accurately compute stress intensity factors at fracture tip locations, and to capture fracture geometry changes due to growth. Thus, advanced, non-interactive, re-meshing technology simulations is essential for its implementation. Additionally, traditional methods for stress intensity factor computation rely on brick-like meshes to compute the energy around the tips, introducing further constraints to the required meshing. Nonetheless, the method has been widely used for fracture propagation during the past few decades. Mesh refinement aimed at enhancing stress field modelling was initially set to be localised at the tips, due to computational time restrictions [Wawrzynek and Ingraffea, 1989]. As meshing became less expensive, full re-meshing approaches were proposed as alternatives [Bittencourt et al., 1996] to bypass refinement region identification and merging. However, most available methods, albeit aimed at propagation of quasi-static brittle fracture cracks in 3D, rely on manual interaction during re-meshing and resort to cumbersome brick-like meshes around the tip for accurate stress intensity factor computations [Gurses and Miehe, 2009].

By implementing a methodology for stress intensity factor computation on arbitrary meshes, it becomes feasible to formulate a FEM-based method, which in combination with re-meshing, can simulate fully automatic multiple fracture propagation in 2D [Bouchard et al., 2000, Paluszny and Matthai, 2009] and 3D [Paluszny and Zimmerman, 2011]. Crack propagation is controlled by three criteria: failure, propagation magnitude, and propagation angle. A failure criterion determines whether or not the tip will advance, the propagation magnitude determines the fracture surface gained, and the propagation angle controls the final shape of the fracture. These three criteria take as input the magnitude of the locally computed stress intensity factor, K , and stress intensity factor for opening model (K_I), sliding model (K_{II}) and tearing model (K_{III}).

Thus, the model relies on the accurate computation of the energy ahead of the propagating tip. The three criteria are input to the model, and can be adjusted to satisfy material and in-situ constraints. The implemented criteria includes a sub-critical failure criterion, a Paris-type propagation law, and a maximum circumferential stress angle criterion, and introduce how they are used to change the propagating fracture shape.

FEM without remeshing Finite element method (FEM) [Zienkiewicz and Taylor, 2000] is properly the most popular computational method for solid mechanics in engineering. However, mesh generation becomes the main difficulty for FEM to simulate fracture propagation and fragmentation, in which the geometries of colliding bodies require updating. To avoid remeshing, smeared crack method is proposed, in which FEM is applied without explicit fracture representation [Jirasek, 1998, Jirasek and Zimmermann, 1998a,b]. 2D cracks are described by multiple embedded displacement discontinuities [Jirasek and Zimmermann, 2001]. The main disadvantage is that there is no stress singularity available for the computation of stress intensity factors. Additionally, as the domain is not explicitly defined, it requires a refined mesh to be able to represent the emerging smeared crack.

Alternatively, to avoid remeshing, Beissel et al. [1998] and Tang et al. [1998] model the discontinuity inside material using tagging and deleting of broken mesh elements. used this methodology to model crack propagation. Using this approach, the criterion of crack propagation is a function of local element strength [Tang et al., 1998]. Its main advantage is the simplicity with which the geometry is represented. The disadvantages of this method include: (a) results that are strongly influenced by the amount and size of elements, (b) the mesh is required to be homogeneously refined throughout the simulation to capture the onset of fracturing, and (c) the stress singularity at the crack tips is not correctly represented. Difficulties related to the technical implementation of geometric evolution and automatic re-meshing motivated the development of mesh-independent methods.

Discrete element method The discrete element method (DEM) is a numerical method that is used to solve for dynamics, stress, and displacements of a group of discrete bodies [Cundall, 1988]. DEM often relies on the penalty method to compute collisions, and computes finite rotations as well as the displacements of discrete bodies separately. Contact detection in DEM was applied to arbitrary shapes in 2D [Cundall, 1988] and 3D [Hart et al., 1988], which are composed by conglomerates of lower level particles such as spheres. Thus, internal slits such as those defined by fractures are modelled with difficulty in DEM. DEM is used to model the contact problem [Magnier and Donz, 1999, Williams and Connor, 1999], with the advantage that it is inherently applicable to discontinuous domains. DEM has been used to model fracture propagation [Wittel et al., 2003, 2008], rock fragmentation [Ferenc and Hans, 1996] and molecular dynamics (MD) under shear loads [Tillemans and Herrmann, 1995]. Rock fragmentation in rock fall was simulated using DEM by [Wang and Tonon, 2010]. Instead, fracture propagation is often a function of bond breakage between DEM particles. Furthermore,

a difficulty posed by DEM is the need to define micro-properties including maximum tensile strength and maximum shear strength [Wang, 2009]. These micro-level parameters defined at the scale of the spheres are different from the macro-level, measurable properties, rendering calibration necessary for their identification for each material. Ghaboussi [1988] considered the deformation of individual discrete element to numerically model processes that combine continuous and discontinuous domains in two dimensions, which is named hybrid FEM and DEM method, named as FEM-DEM. FEM-DEM has been applied to simulate rock-fill in 2D [Bagherzadeh et al., 2010]. In that work, collision and movement is modelled using DEM, while fracturing is reduced to partitioning polygon-shaped elements using a simple breakage line. However, most of the work on DEM-FEM for fragmentation is in 2D, and does not implement accurate failure-criterion based propagation. DEM-FEM has been implemented to simulate fracture mechanics studies of heterogeneous media in 2D [Azevedo and Lemos, 2006], transport phenomenon that occurs in highly saturated soil due to seepage forces [Bierawski and Maeno, 2006]. In contrast to FEM, DEM is better suited to model multi-body dynamics; however, its disadvantages include computational expense, difficulty in capturing complex geometry using spherical particles, lack of accuracy in displacement computations as compared to FEM, and difficulty in accurately modelling fracture propagation.

Hybrid finite and discrete element method Munjiza et al. [1995] proposed a combined finite-discrete method (FDEM) which is able to model the transition from continuum to discontinuum conveniently. FDEM discretized a computational domain into a group of separate discrete elements, meanwhile each discrete element has a finite element meshes separately [Munjiza et al., 1999, Munjiza and John, 2002]. Therefore, FDEM is able to inherit the finite element-based analysis of continuum and the discrete element-based analysis of contact and collision interaction [Munjiza, 2004, Munjiza et al., 2011].

FDEM is widely used to model collision, fracturing and fragmentation which involve discontinuum and continuum [Morri et al., 2002, Onate and Rojek, 2004, Morris et al., 2006, Mahabadi et al., 2012]. Munjiza et al. [1999] demonstrated the use of FDEM to model the initiation and propagation of cracks under mode I loads. In his work, FEM is implemented for the hardening part of constitutive law, and is used before the stress in the material reaches the ultimate tensile strength. The opening and propagation of cracks is simulated using DEM. Applying FDEM to simulate the fracturing in Brazilian disc test can be found in references [Mahabadi et al., 2009, Mahabadi and Grasselli, 2010]. Munjiza and John [2002] investigated sensitivity to mesh size of the combined single and smeared crack model, and showed that, for refined meshes, accurate stress and strain fields can be captured around crack tips. Munjiza and Latham [2004] compared the gravitational depositions of cubes by experimental and numerical methods. Their results are sensitive to initial conditions and model parameters.

In practical engineering, FDEM has been successfully used to simulate the packing and repositioning of concrete armors in coastal engineering Latham et al. [2008b,a], Munjiza et al. [2010]. By combining FDEM with computational fluid mechanics, the massive particulates for break-

water engineering was investigated Latham et al. [2008a]. Tatone et al. [2010] used FDEM to assess the stability of gravity dam in hydropower engineering. To predict the behaviour of rockfall, which falls along a vertical or sub-vertical cliff, the impact-induced fragmentation of a single rockfall was investigated in references [Lisjak and Grasselli, 2010, Lisjak et al., 2010]. In addition, the cliff recession was modelled using FDEM [Mahabadi et al., 2012]. FDEM also inherits the disadvantage of DEM, such as requiring the calibration of micro- to macro-material properties for elastic problem. Elmo et al. [2007] investigated the surface subsidence in block caving mining engineering in two dimensions.

Boundary Element Method In the Boundary Element Method (BEM), only the boundary of the computational domain is discretised [Mi and Aliabadi, 1994]. This feature makes it attractive for fracture problems [Cruse, 1969, Kolk et al., 2005]. However, BEM has several disadvantages. For example, it is difficult to deal with the interfaces of different materials, since the discretisation is only implemented on the boundary. BEM has difficulty in considering a situation with multiple materials. Furthermore, the boundary integral equation degenerates when two surfaces are in close proximity [Cruse, 1972, Mukhopadhyay et al., 1998, 2000]. The dual boundary element method (DBEM) was extended to three dimensions by [Mi and Aliabadi, 1994]. DBEM, for crack growth in 3D, utilises special crack front elements which incorporate the displacement variation by modifying shape functions, a technique which allows continuous elements in the discretisation on crack surfaces Wilde and Aliabadi [1999]. BEM has also been applied to grow multiple fatigue cracks in 3D [Cisilino and Aliabadi, 1997]; specifically by using a multi-domain model to analyse fracture problems in non-symmetric media [Blandford et al., 1981]. Furthermore, the dual boundary element method and the time domain method were presented to simulate dynamic fracture problem [Fedelinski et al., 1995].

Extended Finite Element Method (XFEM) The Extended Finite Element (XFEM) was proposed to overcome the meshing burden associated to FEM methods [Moes et al., 1999, Belytschko and Black, 1999]. In this method, enrichment functions are applied to represent fractures at a sub-grid level. XFEM has been applied to grow [Moes et al., 1999] and intersect [Daux et al., 2000] fractures in 2D and 3D [Sukumar et al., 2000], including the modelling of intra fracture friction [Dolbow et al., 2001]. Although a mesh is not required in XFEM to capture crack surfaces for the construction of shape functions, elements are still required to be divided according to the crack surface, for accurate integration. Moreover, to simulate multi-crack propagation, and due to the impossibility of anticipating the fracture path during growth, the mesh requires quite a high degree of homogeneous refinement in order to accurately capture deformation, and possible stress singularities, in any given region of the model. XFEM also has difficulties representing intra-element fracture intersection or multiple fractures in one element, requires of special approaches to handle fracture tips within elements, and is bound to a fixed amount of level-set functions that restrict crack orientation within each element.

Cracking particle method The cracking particle method (CPM) is a numerical method designed to simulate crack propagation, in which crack surfaces are represented by a set of point-based enriched functions [Rabczuk et al., 2007b,a, 2004]. It is similar to XFEM in that the enrichment of test, trial, and sign functions is exploited. In CPM, fractures are represented by a set of nodes. Using H-adaptivity, particles can be added and removed without changing the underlying data structure [Rabczuk and Belytschko, 2005]. CPM was extended to 3D by Rabczuk and Belytschko [2007]. The cracking particle method was proposed to simplify the geometric representation of fractures. However, to simulate fragmentation, fragments are split by sets of growing fracture surfaces. However, in CPM, the enrich functions are structured based on a set of point-based line segments, therefore actually the crack surfaces are represented by a set of line segments and there is no explicit information of fracture surface. Thus, a methodology relying on CPM would require reinterpretation of fracture geometry during growth and at the fragmentation stage.

Mesh-free method The mesh-free method was proposed to avoid the drawbacks of remeshing for the computation of the displacement and stress fields [Belytschko et al., 1994, 1995a,b, 1996]. When the mesh-free method was first used to simulate fracture processes, it was aided by a background mesh [Belytschko et al., 1995a,b, Liu and Tu, 2002]. However, the background is applied only for integration in the computational domain, but not used for the construction of shape functions. Numerical examples show that the mesh-free method can obtain accurate stress intensity factors, and crack growth can be modelled conveniently. Since the mesh-free method avoids re-meshing, it is attractive to simulate fracture propagation in 3D [Rabczuk et al., 2007b]. Even though stress intensity factors can be obtained accurately using the mesh-free method, stress oscillation around the crack tip requires adequate point density and refinement of the underlying mesh. The Enriched Element-free Galerkin method (EFG) formulation for fracture problems can obtain accurate stress intensity factors with few degrees of freedom in 2D [Fleming et al., 1997] and 3D [Duffot, 2006]. Initiation, growth, coalescence and branching of multi cracks, as well non-linear material properties and dynamics were taken into account under the framework of the mesh-free method with enrichment [Bordas et al., 2006]. However, the mesh-free method has several disadvantages. Firstly, the distribution of nodes has a significant influence on accuracy. Secondly, the time to generate node connectivity is not less than the time needed to generate a mesh for an equivalent element method simulation. Finally, the support domain of nodes in the mesh-free method is round or spherical, and is not controlled by the geometric boundary, so it is difficult to apply external forces in this context.

4.2 Fragmentation simulation workflow

Fig. 4.1 shows the technology roadmap proposed for the fragmentation simulation. The methodology is aimed at simulating crack propagation, dynamics, and collision between multiple volumetric bodies in 3D. The cycle describes the interaction between the FEM-based modelling

of discrete fracture propagation and the impulse-based dynamics and collision response. The initial volumetric model, including the ore and its boundaries, is created using off-the-shelf CAD software (Rhino). In most tests, the fragments initially begin to move due to gravity. The dynamics and collision response is computed at each time step. Fracture is simulated selectively in those fragments that are subjected to large contact forces - larger than a pre-defined threshold value. Fragment geometry is updated according to results of fracturing: some are split by fracture surfaces and the resulting fragments are subsequently handled independently, in some fragments the internal fracture pattern of the fragments is updated, and others remain intact. Geometric information, fragment velocities, and boundary conditions are submitted to the input dynamics and collision response computation, which are resolved using the hybrid impulse/penalty method described before. To combine the FEM-based modelling of discrete fracture propagation and the modelling of dynamics and collision response, it is expected to simulate crack propagation more accurately and achieve more accurate results about the energy transfer during collision, comparing with DEM. As compared to the penalty-based methods, the computational cost will be reduced, since the impulse-based methods will be implemented.

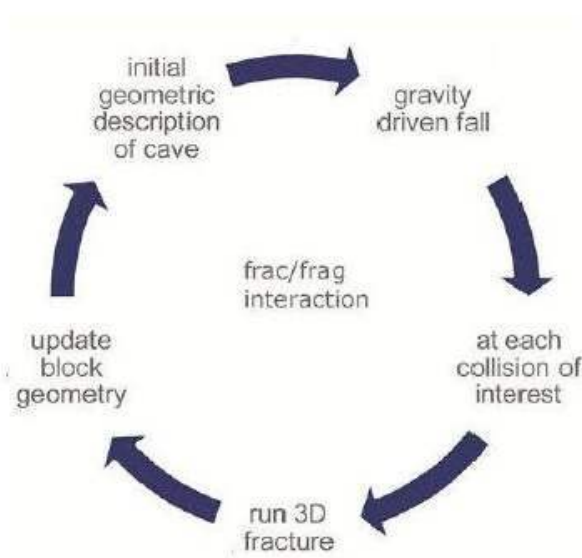


Figure 4.1: The fracture/fragmentation cycle combines the computation of fracturing and the computation of multi-body interaction

4.3 Geometric representation

Each body is represented by a family of points, curves, and surfaces. Both fragments and fractures are represented volumetrically. The first have mass, while the second do not. The standard Boundary Representation (BREP) is used to define water-tight volumetric bodies

before discretisation. Two different discretisation are generated during the simulation: the first is a coarse volumetric representation of fragments used for the dynamics simulation, the second is a refined volumetric representation for fracture propagation. Fragments can, therefore, be convex or non-convex and are always discretised by tetrahedral or hexahedral elements (Fig. 4.2). The points, curves, surfaces that comprise the fracture are grouped into different families. Fracture surfaces are represented by a combination of 3D faceted surfaces and Non-Uniform Rational B-Splines (NURBS) surfaces. Thus, fracture tips are reduced to polyline and NURBS curve descriptions. Families are retained during re-meshing to avoid searching and re-grouping operations. Figure 4.3 shows how fractures are represented before and after discretisation.

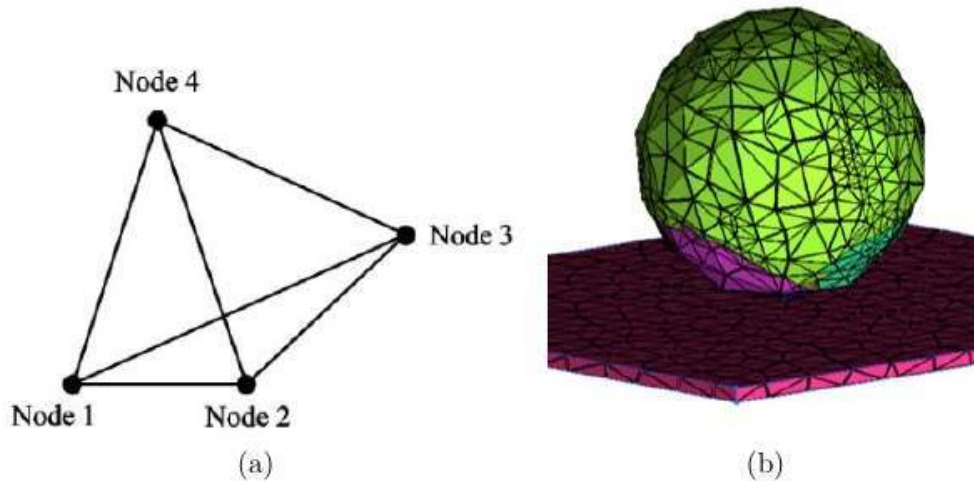


Figure 4.2: Geometric data structure of bodies: (a) tetrahedral element, (c) complex body composed by multiple tetrahedral elements.

After meshing, a group of 3D triangles and segments represent each fracture and its tips. The number of tips and surfaces is arbitrary and increases with propagation. During the simulation of fracture propagation, the geometry is updated using the deformed mesh. Additionally, fractures grow as a function of stress integration at the tips by adding new lofted fracture surface segments between the old tip and the newly created tip (Fig. 4.3 c,d). Specifically, propagation vectors $\vec{\Delta}_n$ are used to determine the location of the new tip nodes,

$$\mathbf{p}_n^{i+1} = \mathbf{p}_n^i + \vec{\Delta}_n \quad (4.1)$$

During the crack growth, Boolean operations are used to determine the intersection and closure of fractures. As shown in Fig. 4.3, once a tip reaches a free surface it stops. Boolean operations are used again to merge the bodies of the participating fractures. Further details of the geometric evolution can be found in reference [Paluszny and Zimmerman, 2011]

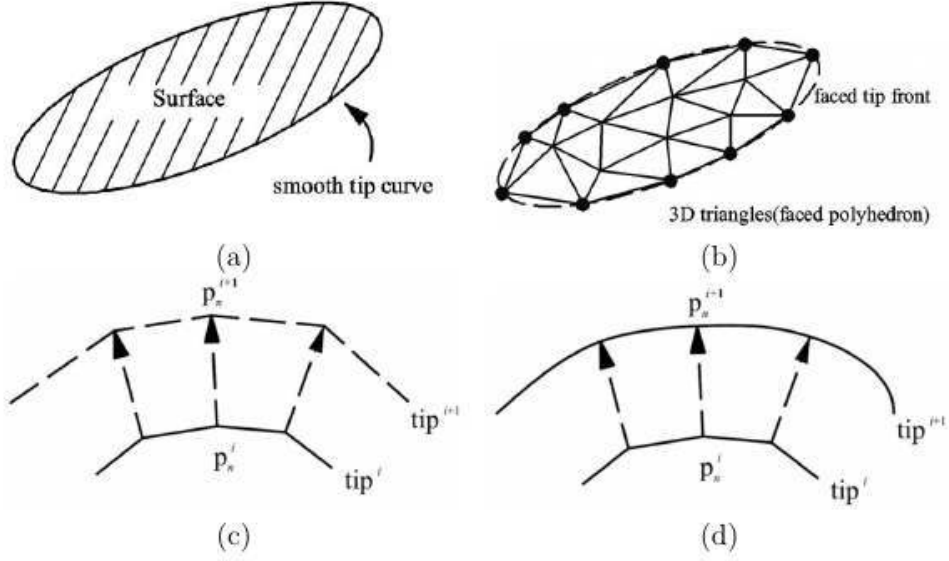


Figure 4.3: The representation of fractures during crack growth: (a) smooth fracture surface before meshing, (b) the mesh of fracture surface, (c) propagation vectors, and (d) new tip fitted through propagated nodes: tip^i and tip^{i+1} are the tip curves of a fracture at steps i and $i + 1$, and are the nodes of fracture n at steps i and $i + 1$.

4.4 Deformation

Fragments are handled as independent FEM domains. Deformation is modelled quasistatically assuming a homogeneous, isotropic, linear elastic medium [Jaeger et al., 2007]:

$$\sigma = \mathbf{D}_e(\varepsilon - \varepsilon_0) + \sigma_0 \quad (4.2)$$

where σ and ε are the strain and stress vectors, σ_0 and ε_0 are the initial strain and stress, and $\mathbf{D}_e$ is a linear elastic stiffness matrix. For a static system

$$\partial\sigma + \mathbf{F} = 0 \quad (4.3)$$

where ∂ is the divergence operator and $\mathbf{F}$ represents the external body forces, i.e. gravity, dilatation, and acceleration. The ensuing system is solved using FEM, for which displacements are the solution variable defined at the nodes, and material properties are defined at the element Gaussian integration points at which stresses and strains are computed. The FEM inversion of the matrix is performed using the Fraunhofer SAMG Solver [Stüben, 2001], which solves for vectorial fields using the multi-grid method.

The fragment domain is discretised using a three-dimensional mesh composed of a set of isoparametric quadratic bars, triangles and tetrahedra. Elements around the crack tip are quarter-point tetrahedra, which better capture the singularities at the fracture tips, and improve accuracy in

stress intensity factor calculations [Banks-Sills and Sherman, 1986, Banks-Sills, 1991]. Figure 4.4 illustrates the stress contours computed during the collision of a sphere with the ground, the non-symmetric nature of the field is attributed to the effect of growing internal fractures. The insides of the fractures are not meshed. These exist as negative volumes surrounded by matrix elements. Meshes are generated based on the geometry, and their optimisation is based on geometry, whereby

$$t_r = O(d_t) \quad (4.4)$$

where t_r is the triangle resolution, O is a quadratic function, and d_t is the distance to the closest fracture NURBS surface tip.

4.5 Fracture propagation

The simulation of crack propagation is limited by the technology available for controlling geometry and mesh generation. Thus, a myriad of approaches to avoid meshing and geometric handling have arisen over the past decades. Crack representation is often defined in a sub-grid fashion, as an entity within the mesh to avoid mesh modifications and re-generation. Other methods, e.g. the boundary element method, aim to reduce complexity by representing only the boundaries of the bodies in question: these are unable to capture heterogeneity in the matrix. Mesh-free methods bypass meshing completely and define domains as a set of points: these introduce difficulties such as domain interface blurring, and costly computational operations. In this section, a brief overview of the available methods to model fracture propagation, multi-body interaction and fragmentation are presented. Figure 4.4 illustrates the results of the three-dimensional fracture propagation approach presented herein.

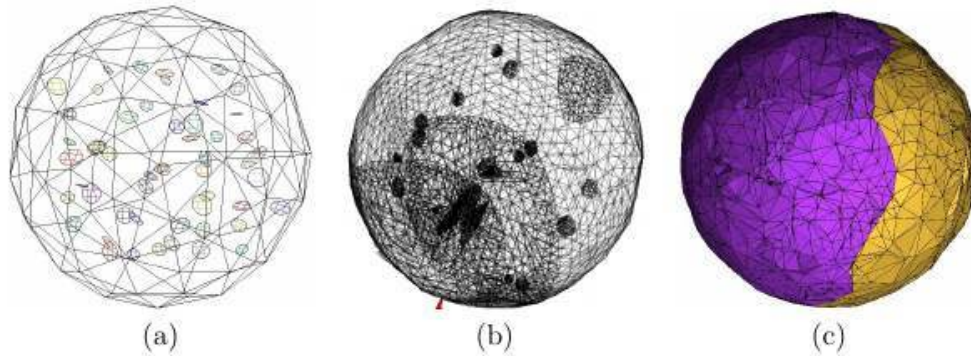


Figure 4.4: Fragmentation driven by fracture growth. (a) A set of initially planar, elliptic flaws grow as a function of stress intensity factor measurement at the tip locality. (b) Fractures advance at a relative speed controlled by the propagation factor exponent. (c) Once fractures reach the solid's boundary, the fragment domain is automatically sub-divided. In (b) the red cone at the bottom-left represents the point of impact, in (c) the object is rotated to the left by 45 degrees.

4.5.1 Stress intensity factor computation

The amount of energy release by a unit crack extension in the direction of the crack plane can be described in the form of area integration using the J integral [Cherepanov, 1967, Rice, 1968]:

$$J_v = \frac{1}{A_c} \int_V (\sigma_{i,j} \frac{\partial u_j}{\partial} x_k - W \varepsilon_{ik}) \frac{\partial u_j}{\partial x_k} dv \quad (4.5)$$

where A_c is the area of the J integration, $\sigma_{i,j}$ is the Kronecker delta, q is an arbitrary weighting vector function representing the virtual crack extension, W is the strain energy, and ε and σ are the strain and stress ahead of the tip, respectively. Quarter point elements [Henshell and Shaw, 1975, Barsoum, 1976] are used to better capture the singularity at the fracture tips. By decomposing the stress and displacement fields into symmetric and anti-symmetric parts, mixed-mode stress intensity factors can be extracted using the interaction energy integral [Cervenka and Saouma, 1997]. Details on how to integrate J a three-dimensional an arbitrary mesh can be found in reference [Paluszny and Zimmerman, 2013]. It follows that for isotropic, homogeneous, linear elastic materials $J = G$, where G is the strain energy release rate, and G is related to stress intensity factor K by

$$\begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} = \frac{1}{E_{eff}} \begin{bmatrix} K_I^2 \\ K_{II}^2 \\ \frac{E_{eff}}{2\mu} K_{III}^2 \end{bmatrix} \quad (4.6)$$

where K_I , K_{II} , and K_{III} are the modal stress intensity factors. Once stress intensity factors ahead of the crack have been computed, the approximate propagation can be captured.

4.5.2 Failure criterion

Once K is computed, a sub-critical failure criterion is used for tensile failure, $K_{IC} < K < K_C$, where K_C is the material toughness and K_{IC} is a sub-critical value, which is lower than the material toughness to account for the lower strength of the damage zone, and other tip processes. Additionally, a Coulomb-based failure criterion is used for failure in compression.

4.5.3 Propagation magnitude

The computation of the propagation magnitude is based on a Paris-type law, derived from experimental observation [Charles, 1958]:

$$\frac{da}{dN} = C(\Delta K)^m \quad (4.7)$$

where a is the length of crack, N is the number of cycles (the number of simulation steps of crack propagation), ΔK is the change of the stress intensity factor during crack propagation,

and C and m are constants. It follows that [Renshaw and Pollard, 1994]:

$$l_{adv} = l \left(\frac{G}{G_{max}} \right) \quad (4.8)$$

where l_{max} is the maximum crack advance, l_{adv} is the tip advance magnitude and α is an experimentally-derived exponent.

4.5.4 Propagation direction

The direction of crack propagation is determined in the plane perpendicular to the crack front (Erdogan and Shi, 1963). In the classical maximum circumferential propagation angle approach, the propagation angle, θ_0 , is assumed to be independent of the mode *III* stress intensity factor K_{III} , and is in turn, formulated as a function of the local K_I and K_{II} ahead of the tip,

$$\cos\theta_0 = -\sin(K_{II}) \left[\frac{6K_{II}^2 + 2K_I^2 \sqrt{1 + 8 \left(\frac{K_{II}}{K_I} \right)^2}}{18K_{II}^2 + 2K_I^2} \right] \quad (4.9)$$

Alternatively, the propagation angle can be defined as a function of all modes, as described by Dhondt (2003). The fracture engine was developed by Dr. Paluszny and her coworkers in C++ and it relies on a volumetric mesher (Ansys ICEM Tetra), an efficient algebraic solver (Fraunhofer SAMG), and a robust Computer Aided Design tool (Rhinoceros) to operate [Paluszny and Matthai, 2009, Paluszny and Zimmerman, 2011, Paluszny et al., 2012, Paluszny and Zimmerman, 2013, Paluszny et al., 2013].

4.5.5 Dynamics

Calculating the dynamics motion of blocks and the collision velocity and collision force is important, as the resulting damage pattern is strongly related to the collision velocity and collision force in the process of fragmentation (Wittel et. al., 2008). This section introduces the methods for calculating collisions and dynamics of the system. Assume that two bodies collide, where $x_a(t)$ is the position of body A, $v_a(t)$ is its velocity, $u_a(t)$ is the velocity of body A relative to body B, and $u_b(t)$ is the velocity of body B relative to body A at the contact location. Thus,

$$\begin{aligned} \mathbf{v}_a(t) &= \dot{\mathbf{x}}_a(t) \\ \mathbf{a}_a(t) &= \dot{\mathbf{v}}_a(t) \end{aligned} \quad (4.10)$$

where $\mathbf{a}(t)$ is the linear acceleration of body A. From the Newton-Euler equation, the following equations are obtained:

$$\begin{aligned}\mathbf{F}(t) &= m_a \mathbf{a}_a(t) \\ -\mathbf{F}(t) &= m_b \mathbf{a}_b(t) \\ \mathbf{r}_a \times \mathbf{F}(t) &= \mathbf{I}_a \beta_a(t) + \omega_a(t) \times \mathbf{I}_a \omega_a(t) \\ -\mathbf{r}_b \times \mathbf{F}(t) &= \mathbf{I}_b \beta_b(t) + \omega_b(t) \times \mathbf{I}_b \omega_b(t)\end{aligned}\tag{4.11}$$

where $\beta_a(t)$ is the angular acceleration of body A relative to body B, $\beta_b(t)$ is the angular acceleration of body B relative to body A, $\mathbf{F}(t)$ is the collision force, m_a and m_b the mass of rigid body, $\mathbf{I}_a$ and $\mathbf{I}_b$ are the inertia tensor. $\mathbf{r}_a$ and $\mathbf{r}_b$ are the vector from the centre of body to the contact point. During the simulation, the velocities of the objects are updated at each time-step.

4.6 Case study: Impact-Induced Fragmentation

The methodology is now demonstrated by numerically fragmenting a brittle sphere. The sphere has a radius of 100 mm, and collides with a flat surface at 90 degrees, with a velocity at impact of 100, 173, 190, or 202 m/s. The sphere's Young's modulus is 20 GPa, and its Poisson's ratio is 0.25. A low value of K_{IC} of 1.5 MPa·m^{1/2} is assigned, to encourage fragmentation. The volume based cumulative fragment size distribution is computed, and the shape of the fragments and their final geometric distribution are studied. The sphere initially contains fifty planar disk flaws, randomly located, with a normal distribution of radii having a mean of 5 mm and a standard deviation of 0.5 mm (Fig. 4.4). Fig. 4.5 illustrates the mean stress contours of the sphere at impact. A compressive zone is observed around the impact location, with tensile stress concentrations at flaws located away from the impact location. By adjusting the propagation exponent, α , the relative speed at which fractures grow at impact is controlled. For low velocities, $\alpha > 10$ (100 m/s), fractures with a higher stress intensity factors at their tips will grow considerably faster than will their neighbours. Fig. 4.4 shows growth for $\alpha = 10$: the fracture at the impact location grows significantly faster than the others. The sphere is fragmented into two pieces in this case (Fig. 4.4 c). For $\alpha=5$ (173 m/s) and $\alpha=2$ (190 m/s), thirty-five and seventy-five fragments are obtained, respectively, as shown in Fig. 4.6 a, b. Finally, for $\alpha=0.35$ (202 m/s), modelling a high velocity impact, 775 fragments are obtained (Fig. 4.6 c). These experiments are referred to as case s.l.1 ($\alpha = 10$), s.h.2 ($\alpha = 5$), s.h.3 ($\alpha=2$), and s.vh.4 ($\alpha=0.35$). In all cases and images, the point of impact is at the bottom of the sphere.

These results indicate that cumulative fragment mass distributions are characterised by a two-parameter cumulative Weibull distribution, as has been observed in laboratory experiments [Cheong et al., 2004]. The two-parameter cumulative Weibull distribution is commonly used to characterise brittle behaviour [Shih, 1980]. It can be expressed as

$$y = 1 - \exp \left[- \left(\frac{x}{x_c} \right)^m \right]\tag{4.12}$$

where x is the fragment volume, y is the corresponding volume fraction, m is the shape factor, and x_c is the scaling factor.

Fragment mass distributions resulting from the previous experiment are measured and compared to experimental data. Fragment sizes decrease as the velocity increases. In the graphs, cases s.h.2 and s.h.3 correspond to a lower velocity and a higher propagation exponent, and case s.vh.4 corresponds to a higher velocity and a lower propagation exponent. For the fragment mass distribution, $F(m)$, in Fig. 4.7, a power law distribution with an exponential cut-off [Wittel et al., 2008] is observed. This corresponds to a power-law distribution of the small fragment size, and an exponential regime for the larger fragments.

In terms of the two-parameter Weibull distribution, Cheong et al. [2004] observed that x_c reduces linearly as a function of increased velocity, whereas m reduces exponentially with velocity. In the present experiments, both x_c and m reduce as the velocity increases

$$(x_c : 0.47 \rightarrow 0.164, m : 3.08 \rightarrow 1.5) \quad (4.13)$$

In Fig. 4.9, the distribution presented by Wittel et al. [2008] is presented, in normalised form, for comparison. The distribution resulting from the low velocity case fits well with the fragment size distributions presented therein, for the chosen function $\alpha = f(v)$. For the higher velocity case, a reduced shape and scaling factor for the fitted cumulative Weibull distribution is obtained. Fragment shape distribution has been shown to depend on scale, material, velocity of impact, and angle of impact [Cheong et al., 2004, Gorham and Salman, 2005, Chunsheng et al., 2002]. Here, the focus is on reproducing the fragment size variation as a function of the velocity of impact, while scale, material and angle of impact are kept constant. The idea is to reproduce, not only fragment size distributions, but also final fragment shape characteristics as a function of velocity. For case s.h.2, most fragmentation occurs around the impact location. Conical fragmentation areas around the zone of impact are reproduced, along with wedge and cap shaped fragments (see Fig. 4.5). One large fragment dominates the distribution, and fractures tend to grow radially out of the centre of the model. For case s.h.3, a wider distribution of fragment sizes occurs, and the formation of wedges and caps is observed. Case s.vh.4, which models higher velocity impact, exhibits a wider distribution of fragment sizes. Significant crushing around the impact location occurs in this case, and prominent wedge fragments are created radially. Wedges are formed due to the growth of meridian plane fractures. These have been observed to form as an effect of the compressive conical region on the sphere's stress state [Gorham and Salman, 2005].

4.7 Conclusions

Simulation capabilities to model coupled fracture and fragmentation processes using a combined finite element, impulse-based approach, have been presented. The method can be implemented as an extension of a 3D fracture propagation code, provided that the representation of the cracks

is explicit and geometry-based. Geometric collision is based on polyhedral tracing, divided into broad and narrow phase contact detection for increased efficiency. Thus, objects are not allowed to penetrate during the collision. Instead, object trajectories are used to estimate time-of-impact, and contact between bodies is modelled by collisions at contact locations. This results in a smaller geometric computation burden, as penetration distances and penetration points, as would be required by the penalty method, need not be computed, and material properties are allowed to be defined at the macro-scale. The advantages of the proposed method include:

- Fragmentation that is controlled by fracturing rather than driven by bond-breakage.
- Explicit geometric representation of fractures and fragments via geometric modelling of surfaces and solids.
- Direct definition of macroscopic material properties.

Fragments can assume polyhedral or spherical shapes, depending on their size, and fracture propagation is modelled using an energy-based growth approach. The Paris law propagation exponent is defined as a function of impact velocity. This methodology has been demonstrated on the problem of fragmenting a sphere, with the resulting fragment size distribution quantified. The observed fragment volume distribution can be fitted using a two-parameter Weibull distribution, and the average fragment sizes decreases as a function of the impact velocity.

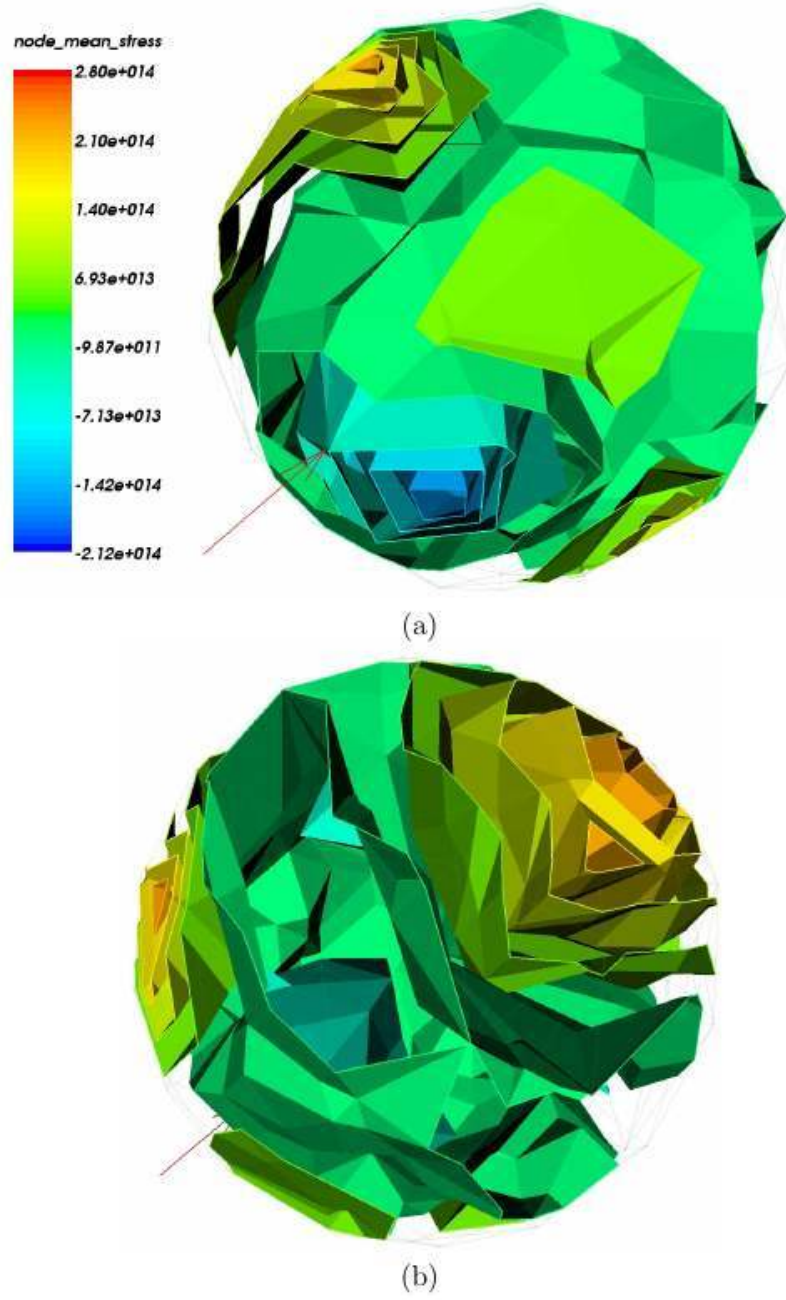


Figure 4.5: Mean stress state contours of sphere at impact. The red arrow indicates the location of impact. The images are two rotations of the same 3D contour distribution. Notice the compression area ensuing around the point of impact, and the tensile fields emerging around growing fractures.

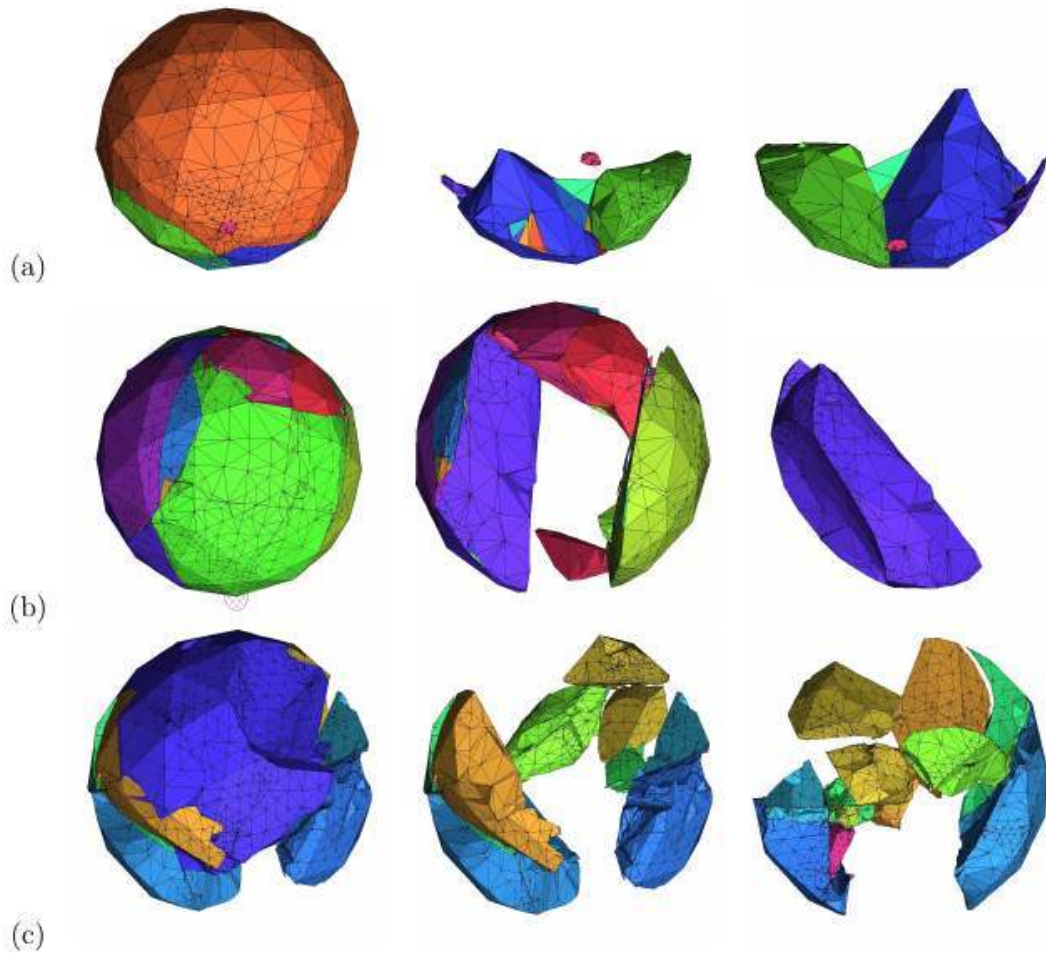


Figure 4.6: Fragment shapes for (a) case s.h.2 - low, (b) case s.h.3 - mid, (c) case s.vh.4 - high velocity impact.

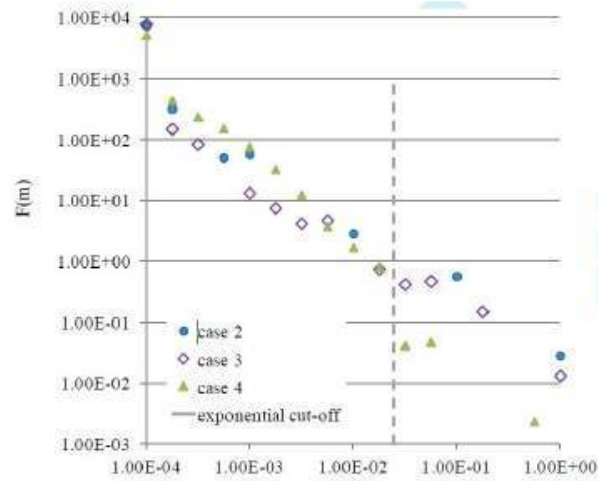


Figure 4.7: Fragment mass distribution, where v is the volume of the fragment, and $F(m)$ is the frequency of fragments of a specific volume. Notice that the simulation generates fragments below the volume threshold, $f_{threshold} = 1 \times 10^{-3} m^3$, these are not further fragmented, they are accumulated as the simulation advances but do not interact with the remaining fragments.

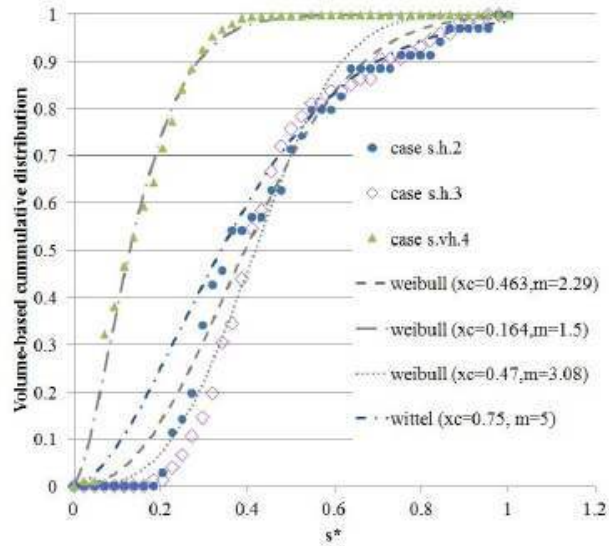


Figure 4.8: Cumulative fragment mass distribution, where s^* is the normalised fragment volume, and the y-axis plots the volume-based cumulative distribution of the fragments.

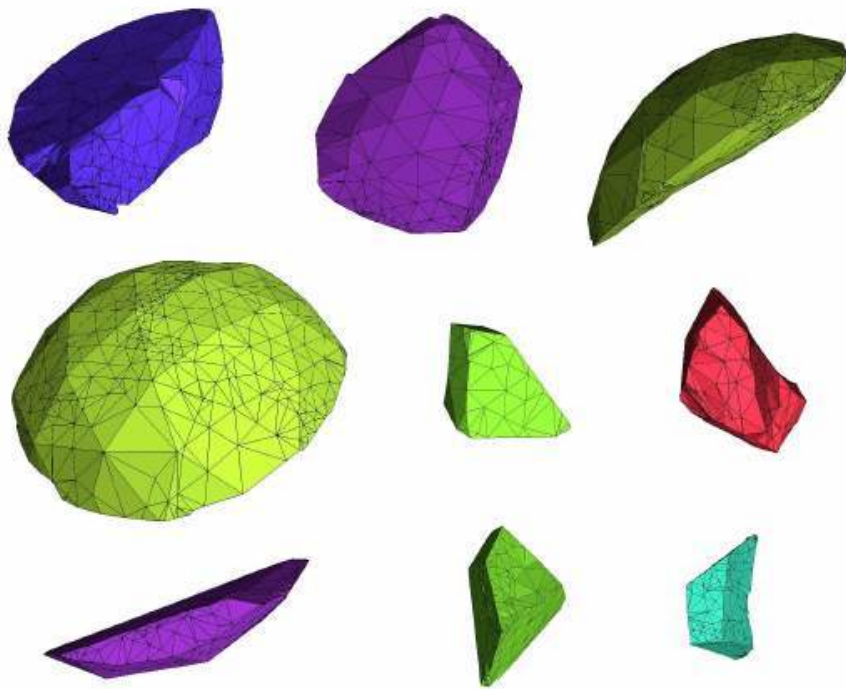


Figure 4.9: Collection of fragment shapes for s.h.2, s.h.3, and s.vh.4.

5 Low-cost Weibull distribution based fragmentation pattern

5.1 Introduction

The fragmentation phenomenon occurs in a variety of materials, ranging from the breakage of a single micro galena grain, of size 1- 2.5 mm [Ali and Bradshaw, 2011], as part of cumminution in mining, up to huge geological faults in the Earth's brittle crust, with fragmentation occurring in the kilometre scale [Holland et al., 2006, Holland, 2004]. Fragmentation in nature is usually a complicated process that relates to a large number of collisions among hundreds and thousands of bodies and the fragmentation that ensues among these bodies. In particular, fragmentation is relevant to various geological and geotechnical processes. Fault gouges in the crust [Holland, 2004, Zhao et al., 2012] are often filled by a huge number of rock fragments. The evolution of active fault gouges may provoke earthquakes which cause injury or property damage to a large population. Understanding the collision and fragmentation of rock fragments is instrumental to predicting and reproducing the evolution of fault gouges. Rock fall [Marsal, 1967] is another widely existing natural hazard strongly related to the impact induced fragmentation process [Agliardi and Crosta, 2003]. Huge falling rocks may fragment into several large rock fragments and many small fragments, which increases the difficulty of predicting the trajectories of falling rocks. Fragmentation changes the size, mass, shape and trajectory of falling rocks which are critical controlling parameters for the design of defence structures [Wang, 2009]. Investigating the fragmentation of rock mass as a function of the quality of the impact can be used to improve comminution equipment, such as high pressure grinding rolls (HPGR) [Klymowsky et al., 2002, Napier-Munn et al., 1996, Djordjevic and Morrison, 2006], which are widely employed to reduce the size distribution of multiple ore fragments [Tromans, 2008]. It has been shown that only 0.1 – 1% of the input energy is used to create new fracture surfaces in conventional grinding, and the rest is wasted in the form of noise and heat [Fuerstenau and Abouzeid, 2002]. Therefore, it is worthwhile to try to contribute to the understanding of the mechanisms that control ore fragmentation.

This work proposes a methodology to study fragmentation using numerical simulations. All of the aforementioned fragmentation related processes combine the collision among and fragmentation of multiple colliding bodies. However, the simulation of the fragmentation of multiple bodies is difficult, mainly due to the intensive computational cost of fracture modelling. Advanced computer technologies, such as GPU-based parallel computation, can be employed to

handle the computational cost [Cleary and Sawley, 2002]. However, speeding up the calculations using computational techniques is not enough. The intensity of the process requires a strategy that reduces the computational cost by developing a cheaper, upscaled algorithm of fragmentation simulation. The present work develops a fragmentation pattern based on the Weibull distribution of fragment sizes of colliding bodies, so as to reduce computational cost. This fragmentation pattern has three controlling parameters, and is derived from existing numerical tests and experiments. This fragmentation-pattern-based approach does not require the re-meshing for explicit fracturing propagation simulation. Instead, colliding bodies are fragmented based on the application of fragmentation patterns directly. The size distribution and size of the largest ensuing bodies due to fragmentation are controlled by the two Weibull parameters, and therefore always satisfy the Weibull distribution that is widely accepted to describe the fragmentation behaviour of brittle materials [Shih, 1980, Cheong et al., 2004, Chunsheng et al., 2002, Wittel et al., 2008].

Associated references about fragmentation are introduced in Section 5.2. The fragmentation pattern based methodology used in present work is discussed in Section 5.3. The validations of the present methodology include the fragmentation of a single sphere (Section 5.4), the fragmentation of a realistically shaped rock (Section 5.5) and the fragmentation of multiple spheres (Section 5.6).

5.2 Fragmentation pattern

Fragmentation process and the ensuing geometric properties are characterised in the literature via laboratory experiments, analytical methods, and numerical methods. This section briefly introduces the studies associated to the ensuing fragmentation patterns, including the fragmentation of single and multiple bodies. Wu et al. [2004] investigated the fragmentation pattern of spheres under double impact test. Different impact velocities, impact angle, strengths and sizes are considered in experiments. Some useful fragmentation patterns observed under static compression are:

- Spheres are crushed into two or three slices, which like the shape of orange slices.
- Two conical fragments fall from the contact zone.

Because fragmentation behaviour under quasi-static and dynamic load conditions is different [Kutter and Fairhurst, 1971]. Wu et al. [2004] also investigates the fragmentation of 141 brittle plaster spheres between two rigid platens under dynamic impacts. A MotionScopeR PCI system was used to capture the process of breakage. Twelve failure patterns were identified, according to the intensity of crushing around the axis of compression, and the total number of fragments. Some useful fragmentations patterns observed under dynamic impact are:

- A meridional primary crack is initiated at the early stage of fragmentation. This primary fracture passes through the vertical axis of spheres, which is the loading axis. At the

second stage, secondary cracks initiate and propagate off the vertical loading axis.

- The crushed fragments can be classified into two classes, the larger fragments and smaller fragments. The larger fragments resulting from the evolution of primary cracks and the smaller fragments resulting from the evolution of secondary cracks.
- The number of fragments increase with increasing impact energy.
- The cumulative mass of size larger is roughly fitted into a Gates-Gaudin-Schuhmann distribution.

5.3 Methodology: Fragmentation based on the Weibull distribution

In Section 4.6, the fracture propagation during collision is modelled using the finite element method (FEM). Without representing fractures implicitly as a set of particles [Rabczuk et al., 2007b,a] or damaged element [Jirasek, 1998], the method used in Chapter 4 describes the fractures explicitly and exactly on their own geometry. The geometry of computational model and mesh are updated at each simulation step. This method, which combines FEM and re-meshing, is able to simulate arbitrary multiple-crack propagation, and consequently it has high accuracy. However, its computational cost is enormous, and it is difficult to use to simulate fragmentation of hundreds or thousands of colliding bodies. Instead of simulating fracture propagation using FEM and re-meshing, the present work establishes a velocity-based fragmentation pattern and uses it to subdivide the colliding bodies directly. This fragmentation approach uses a set of cutting boxes to subdivide colliding bodies, and has three parameters, as shown in Fig. 5.2. For a given impact velocity, the three parameters describing the fragmentation pattern can be looked up in Table 5.1, which avoids massive FEM computational and re-meshing processes. The fragmentation pattern is established by learning from existing experimental and numerical results. The performance of the task of fragmentation is measured by comparing the shape and size distribution of fragments obtained from numerical simulation, to the fragmentation pattern achieved from existing experimental and numerical results. The fragmentation of spheres subjected to loads in one dimension is the basic step to understand the fragmentation mechanism of arbitrarily shaped colliding bodies. As discussed in Section 5.1, many researchers have discussed the fragmentation pattern of spheres [Wu et al., 2004, Cheong et al., 2004] and observed some useful fragmentation patterns. The present work tries to capture some features of the fragmentation of spheres subjected to loads in one dimension. The existing observations, which should be captured by the fragmentation pattern, include:

- It is assumed that a primary direction axis exists during the fragmentation process. Wu et al. [2004] investigated the fragmentation pattern of a set of spheres under double impact tests, and found that there is a primary direction axis during simulation. It was found that

at the early stage of fragmentation, a meridional primary crack is initiated and propagates along a primary direction. Next, a number of secondary cracks initiate and propagate off the vertical loading axis. Cheong et al. [2004] fragmented a set of brittle spheres using 16-mm bore pressurised gas gun in order to understand fragmentation mechanism. The primary direction axis of the fragmentation pattern was also observed. In order to capture this property of a primary direction axis during fragmentation, a primary direction is found based on the maximum impact velocity. A local coordinate system is established using this primary direction, in Section 5.3.1. The following steps to generate the fragmentation pattern is based on this local coordinate system.

- It is assumed that the fragments can be classified into two classes: a small number of larger fragments, and a large number of smaller fragments. In Wu's experiment [Wu et al., 2004], it was found that, when the impact velocity is low, the spheres are fragmented into several larger fragments. When the impact velocity is high, the fragments resulting from fragmentation can be classified into smaller fragments, which initiate from the evolution of secondary cracks, and several larger fragments, which initiate from the propagation of primary cracks. As shown in reference [Wu et al., 2004, Cheong et al., 2004], with high impact velocity, the smaller fragments are mainly distributed in the region in the vicinity of the contact point. This property is captured by using two different rules to generate cutting boxes in the region near the contact point, and in the region far from the contact point, which is introduced in Section 5.3.2. The cutting boxes are a part of the algorithm to generate fragmentation pattern. In the region near the contact point, the cutting boxes have three parameters, a , b , and c , which have the ability to control the number and size of smaller fragments after fragmentation. In the region far from the contact point, the cutting boxes have two parameters, a and b , which have the ability to control the number and size of larger fragments after fragmentation.
- Classifying the fragments into smaller and larger fragments is not sufficient; the size and number of smaller and larger fragments are further controlled using Weibull distribution further. Many researchers have found that the size distribution of fragment achieved after a fragmentation process can be fitted using a two-parameter Weibull distribution [Chunsheng et al., 2002, Wu et al., 2004, Cheong et al., 2004]. For a given impact velocity v , the two parameters of the Weibull distribution can be found from existing experimental and numerical results. This Weibull distribution, which is obtained from existing experimental and numerical results, is defined as the target Weibull distribution. The parameters of the present fragmentation pattern are trained by comparing the achieved size distribution to the target Weibull distribution. The target function is the difference between achieved size distribution and the target Weibull distribution, which should be minimised. This learning process is introduced in Section 5.3.3.

- The number of fragments should increase with an increase of impact energy. When the third property of Weibull distribution is captured, this property will be satisfied automatically.

The relation between impact velocity and the parameters of fragmentation pattern is shown in Table 5.1. The implementation of the fragmentation pattern is simple, and has four main steps:

1. Get impact velocity from the dynamics simulator and use it to generate the fragmentation pattern.
2. The maximum change of relative velocity in the normal direction is found and is then used to generate a local coordinate system.
3. For a given impact velocity v , the parameters of the fragmentation pattern can be found in the Table 5.1.
4. Subdivide the rock fragment using the fragmentation pattern with the obtained parameters. Update the geometry of the colliding bodies and give the updated geometry to the dynamics simulator.

This fragmentation pattern based method has several advantages. Firstly, this methodology reduces the computational cost substantially, because it avoids updating the geometry and the re-meshing process at each simulation step to track the fracture propagation. Secondly, this methodology does not increase the number of elements in colliding bodies at each simulation of fragmentation, which makes the modelling more stable. Thirdly, the fragmentation pattern is trained by learning from existing experimental and numerical results, using the target function of the Weibull distribution. Therefore, the accuracy of size distribution after fragmentation can be ensured.

5.3.1 Local coordinates

Under compression, normal splitting is the primary mode of failure, as observed in the crushing of granular materials [Lee and Farhoomand, 1997, Mesri and Vardhanabhutib, 2009], the fragmentation of falling rocks [Marsal, 1967], the compression of drained sand [Yamamuro and Lade, 1996, Yamamuro et al., 1996, Nakata et al., 2001], the compression of elastic spheres [Sternberg and Rosenthal, 1952] and the compression of irregularly shaped rocks [Hiramatsu and Oka, 1966]. According to the results of Wu's double impact tests discussed in Section 5.2, the size and shape of the larger fragments are strongly related to the primary cracks, which propagate along the loading axis, and the primary direction of fragmentation [Wu et al., 2004]. In the present fragmentation pattern, a primary direction of fragmentation is assumed. This primary direction is derived from the maximum change of relative velocity in the normal direction at each collision step. Subsequently, this primary direction is used to establish a local coordinate system, as shown in Fig. 5.1, which is then implemented to fragment the colliding bodies. The generation of the local coordinate system is introduced in the following parts of this section.

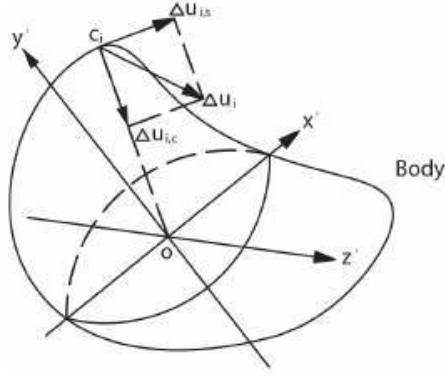


Figure 5.1: A local coordinate system is defined by the primary contact direction. $\mathbf{o}$ is the geometry centre of colliding bodies. y' –axis is the primary direction of fragmentation and x' –axis is the secondary direction of fragmentation. z' –axis is perpendicular to the plan resulting from x' – and y' –. $\Delta \mathbf{u}_i$ is the change of relative velocity at contact point i . $\Delta u_{i,s}$ and $\Delta u_{i,c}$ are the shearing component and compression component of $\Delta \mathbf{u}_i$, respectively.

Primary direction y' The change of the relative velocities at each contact point $\Delta \mathbf{u}_{i,c}$ is decomposed into two components: one compression component and one shearing component. Taking an example, as shown in Fig. 5.1, the change of relative velocity $\Delta \mathbf{u}_i$ is decomposed into compression component $\Delta \mathbf{u}_{i,c}$ and shearing component $\Delta \mathbf{u}_{i,s}$. The compression component is calculated as

$$\Delta \mathbf{u}_{i,c} = \frac{\Delta \mathbf{u}_i \mathbf{r}_i}{|\mathbf{r}_i|} \quad (5.1)$$

and the shearing component of $\Delta \mathbf{u}_i$ is calculated by

$$\Delta \mathbf{u}_{i,s} = \Delta \mathbf{u}_i - \Delta \mathbf{u}_{i,c} \quad (5.2)$$

where, $\mathbf{r}_i$ is the vector from contact point $\mathbf{c}_i$ to the centre of colliding body $\mathbf{o}$, which is calculated by

$$\mathbf{r}_i = \mathbf{o} - \mathbf{c}_i \quad (5.3)$$

The above computation of the compression component is repeated on all contact points. The contact point with the maximum change of relative velocity in compression direction is found. The maximum change of relative velocity in the compression direction is defined as $\Delta \mathbf{u}_c^{max}$. The vector from the contact point with $\Delta \mathbf{u}_c^{max}$ to the geometry centre of the colliding body is denoted as $\mathbf{y}'$, which is primary direction. Then, the primary direction $\mathbf{y}'$ is used to establish the local coordinate system for fragmentation. The detection of the primary direction is summarised in Algorithm 5.

Secondary direction x' Three directions are required to establish a local coordinates system. After the primary direction $\mathbf{y}'$ is determined, the secondary direction $\mathbf{x}'$ is found from the

Algorithm 5 Search the primary direction

Require: A set of contact points $c_i \in \{c_0, c_1, \dots, c_m\}$

- 1: Initialise the maximum change of relative velocity in compression $\Delta \mathbf{u}_c^{max} = 0$.
 - 2: **for** all c_i **do**
 - 3: Calculate the change of relative velocity in compression, $\Delta \mathbf{u}_{i,c}$ using Eq. (5.1).
 - 4: **If** ($\Delta \mathbf{u}_{i,c} > \Delta \mathbf{u}_c^{max}$)
 - 5: $\Delta \mathbf{u}_c^{max} = \Delta \mathbf{u}_{i,c}$
 - 6: Calculate $\mathbf{r}_i$ using Eq. (5.3).
 - 7: Set $\mathbf{r}_i$ as the primary direction, $\mathbf{y}' = \mathbf{r}_i$.
 - 8: **End if**
 - 9: **end for**
-

candidates which are roughly perpendicular to the primary direction. Calculate the vector from the contact point to the geometric centre of the colliding body, $\mathbf{r}_i$, over all contact points. When $\mathbf{r}_i$ is roughly perpendicular to the primary direction, this contact point is inserted to the candidates of the secondary direction. Then the candidate with maximum change of relative velocity in the compression direction is found and is used to establish the secondary direction. For example, for contact point c_i , the change of relative velocity at this contact point is $\Delta \mathbf{u}_i$. The compression component of the change of relative velocity at this contact point is $\Delta \mathbf{u}_{i,c}$, which is calculated using Eq.(5.2). The direction from this contact point to the geometric centre of the colliding body is the compression force $\mathbf{r}_i$, which is calculated using Eq.(5.3). The angle between $\mathbf{r}_i$ and primary direction $\mathbf{y}'$ is calculated using:

$$\theta_{i,p} = \text{acos}\left(\frac{\mathbf{r}_i \mathbf{y}'}{|\mathbf{r}_i| |\mathbf{y}'|}\right) \quad (5.4)$$

When $\theta_{i,p} > 0.25\pi$, it means that $\mathbf{r}_i$ is roughly perpendicular to the primary direction. This contact point is inserted into the set of the candidates of the secondary direction. Then the candidate with maximum change of relative velocity in the compression direction is selected to define the secondary direction. The algorithm that is to search secondary direction is shown in Algorithm 6.

Construct local coordinates After the primary direction $\mathbf{y}'$ and the secondary direction $\mathbf{x}'$ are identified, they are used to establish a local coordinate system $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$. The third direction of the local coordinates $\mathbf{z}'$ is perpendicular to plane of $\mathbf{x}'$ and $\mathbf{y}'$:

$$\mathbf{z}' = \mathbf{x}' \times \mathbf{y}' \quad (5.5)$$

Consider a point $\mathbf{P} = (x_p, y_p, z_p)$ in the global coordinate system. The point in terms of this local coordinate system, $\tilde{\mathbf{P}} = (\tilde{x}_p, \tilde{y}_p, \tilde{z}_p)$, is calculated by

$$\tilde{\mathbf{P}} = \mathbf{L} \mathbf{P} \quad (5.6)$$

Algorithm 6 Search the secondary direction

Require: A set of contact points $c_i \in \{c_0, c_1, \dots, c_m\}$ and the primary direction $\mathbf{y}'$

```
1: Initialise the maximum change of relative velocity in the compression direction  $\Delta \mathbf{u}_c^{sec} = 0$ .
2: for all  $c_i$  do
3:   Calculate the angle  $\theta_{i,p}$ , which is the angle between  $\mathbf{r}_i$  and primary direction  $\mathbf{y}'$ , by Eq.
   5.1.
4:   If ( $\theta_{i,p} > 0.25\pi$ )
5:     Calculate the change of relative velocity in the compression direction  $\Delta \mathbf{u}_{i,c}$ .
6:     If ( $\Delta \mathbf{u}_{i,c} > \Delta \mathbf{u}_c^{sec}$ )
7:        $\Delta \mathbf{u}_c^{sec} = \Delta \mathbf{u}_{i,c}$ 
8:       Set  $\mathbf{r}_i$  as the secondary direction,  $\mathbf{x}' = \mathbf{r}_i$ .
9:     End if
10:  End if
11: end for
```

where $\mathbf{L}$ is the transfer matrix.

5.3.2 Cutting boxes

The fragmentation pattern uses a set of cutting boxes to subdivide colliding boxes directly. These cutting boxes are based on the local coordinates $\{\mathbf{x}', \mathbf{y}', \mathbf{z}'\}$ and have three parameters. As mentioned in Section 5.2, the fragments resulting from double impact tests can be classified to two different sizes: larger fragments and smaller fragments [Wu et al., 2004]. The larger fragments result from primary fractures, which mainly pass through the primary direction, whereas smaller fragments result from secondary fractures and distribute in the vicinity of contact points. Therefore, the present work uses two different rules to generate cutting boxes in the region near the contact point and the region far from the contact point, respectively. Thus, the cutting boxes are able to generate a number of smaller fragments and several larger fragments. Moreover, cutting boxes are able to ensure that smaller fragments mainly are located in the region around the contact point. As shown in Fig. 5.2(a), in the lower half of the colliding body, cutting boxes are created in three dimensions. The number of cutting boxes in three dimension are a , b and c respectively. As shown in Fig. 5.2(b), in the upper half of the colliding body, cutting boxes are created in two dimensions. The number of cutting boxes in two dimension are a and b , respectively. To simplify the algorithm, the values of a and b are same in both the upper half and lower half of the body.

There are many approaches to generate cutting boxes. The present work uses the Voronoi tessellation to generate cutting boxes. To generate a Voronoi tessellation, a set of seeds is needed. The distribution of seeds is achieved in Algorithm 7. Here, six control points, as shown in Fig. 5.3, are used to assist the generating of Voronoi tessellation seeds. Then, a group of Voronoi tessellation is generated using a three-dimensional Voronoi cell library, Voro++ [Rycroft, 2009]. One Voronoi tessellation is shown in Fig. 5.4. It is seen that, according to the rule of generating Voronoi tessellation [Okabe et al., 2000], when the distribution of the seeds

of Voronoi tessellation is orthogonal, the Voronoi cells will be boxes. It has to be admitted that using a Voronoi tessellation to generate cutting boxes is not the simplest approach. The reason for implementing a Voronoi tessellation is to remain the structure of the code used in the rock mechanics group at Imperial College.

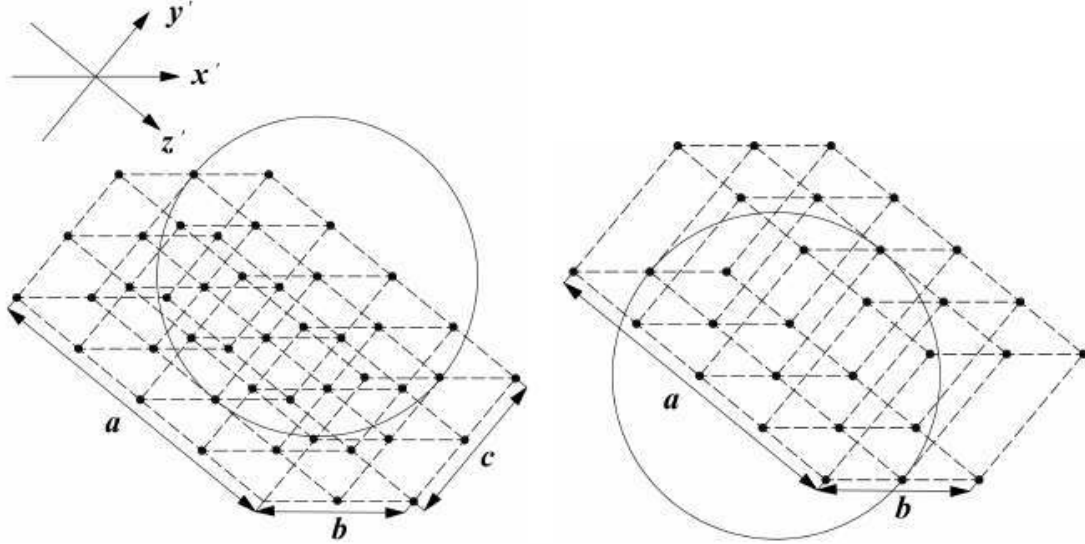


Figure 5.2: Cutting boxes define the overall fragmentation of the object. The colliding body is equally divided into two domains. The domain near the contact point is defined as the lower half of the body. (a) In the lower half of the body, cutting boxes are generated in three dimensions to generate smaller fragments. The number of cutting boxes in three dimension are a , b and c , respectively. For example, in this graph, $a=4$, $b=2$ and $c=2$. (b) In the upper half of the body, cutting boxes are fragmented in two dimensions to generate the larger fragments. The number of cutting box is two dimensions are a and b , respectively. For example, in this graph, $a=4$ and $b=2$. To simplify algorithm, the value of a and b are same in both upper half and lower half of body.

5.3.3 Measure performance of fragmentation pattern using Weibull distribution

In this section, the parameters of cutting boxes are further trained by learning from existing experimental and numerical results based on the Weibull distribution equation. Many researchers have found that the size distribution of fragments achieved from a fragmentation process is able to be fitted using a two-parameter Weibull distribution using a least square fit. The Weibull distribution equation is

$$y = 1 - \exp\left[-\left(\frac{x}{x_c}\right)^m\right] \quad (5.7)$$

where, x is the fragment size and y is the volume fraction corresponding to fragment size. The Weibull distribution equation has two parameters, m and x_c . m is corresponding to the width

Algorithm 7 Creating cutting boxes

Require: The domain of colliding body is denoted as Ω . The nodes in colliding body are denoted as $(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n)$. n is the total number of nodes. In global coordinates system and local coordinates system, the coordinates of node $\mathbf{p}_i$ are (x_i, y_i, z_i) and $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$, respectively. The seeds of Voronoi tessellation are stored in Ω_{voronoi}

```
1: Initialise  $\tilde{x}_{\min}, \tilde{y}_{\min}, \tilde{z}_{\min}, \tilde{x}_{\max}, \tilde{y}_{\max}$  and  $\tilde{z}_{\max}$ .
2: for all nodes in domain  $\Omega$  do
3:   Calculate the coordinates of  $\mathbf{p}_i$  in local coordinates system, and is denoted as  $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ 
4:   If  $(\tilde{y}_i < \tilde{y}_{\min})$ 
5:      $\tilde{y}_{\min} = \tilde{y}_i, \mathbf{c}_1 = \mathbf{p}_i$ .
6:   End if
7:   If  $(\tilde{y}_i > \tilde{y}_{\max})$ 
8:      $\tilde{y}_{\max} = \tilde{y}_i, \mathbf{c}_2 = \mathbf{p}_i$ .
9:   End if
10:  If  $(\tilde{z}_i < \tilde{z}_{\min})$ 
11:     $\tilde{z}_{\min} = \tilde{z}_i, \mathbf{c}_3 = \mathbf{p}_i$ .
12:  End if
13:  If  $(\tilde{x}_i > \tilde{x}_{\max})$ 
14:     $\tilde{x}_{\max} = \tilde{x}_i, \mathbf{c}_4 = \mathbf{p}_i$ .
15:  End if
16:  If  $(\tilde{z}_i > \tilde{z}_{\max})$ 
17:     $\tilde{z}_{\max} = \tilde{z}_i, \mathbf{c}_5 = \mathbf{p}_i$ .
18:  End if
19:  If  $(\tilde{x}_i < \tilde{x}_{\min})$ 
20:     $\tilde{x}_{\min} = \tilde{x}_i, \mathbf{c}_6 = \mathbf{p}_i$ .
21:  End if
22: end for
23: Initialise  $ip_x = 1, ip_y = 1$  and  $ip_z = 1$ .
24: while  $ip_x < (b - 1)$  do
25:   while  $ip_y < (c - 1)$  do
26:    while  $ip_z < (a - 1)$  do
27:       $\mathbf{p}_{\text{voronoi}} = \mathbf{c}_1 + ip_y 0.5(\mathbf{c}_2 - \mathbf{c}_1) + ip_x(\mathbf{c}_5 - \mathbf{c}_3) + ip_z(\mathbf{c}_6 - \mathbf{c}_4)$ 
28:      Insert  $\mathbf{p}_{\text{voronoi}}$  into  $\Omega_{\text{voronoi}}$ .
29:       $ip_z = ip_z + 1$ .
30:    end while
31:     $ip_y = ip_y + 1$ .
32:  end while
33:   $ip_x = ip_x + 1$ .
34: end while
35: A group of Voronoi tessellation is generated using a three-dimensional Voronoi cell library,
    Voro++, according to the seeds in  $\Omega_{\text{voronoi}}$ .
36: Voronoi tessellation is used to fragment the colliding body.
```

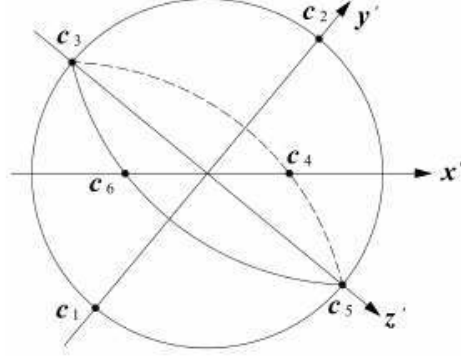


Figure 5.3: Control point for Voroni tessellation. These six control points are the limits of colliding body in local coordinates system.

of distribution [Shih, 1980] and x_c is the fragment size corresponding to a volume-fraction of $1-1/e$ [Cheong et al., 2004]. For a given impact velocity v , the two parameters can be found from existing experimental and numerical results. The Weibull distribution that satisfies the experimental and numerical data is defined as the target Weibull distribution. The parameters of cutting boxes, a , b and c , are trained by comparing the achieved size distribution to the target Weibull distribution. To define the learning process, three features of this problem are defined:

1. Task: Fragment a colliding body.
2. Performance measure: Comparing the achieved size distribution to the target Weibull distribution. For a given impact velocity v , the size distribution under this impact velocity can be fitted by a Weibull distribution with calculated parameters achieved from existing experimental and numerical results, using Eq. (5.9) and Eq. (5.8). The Weibull distribution with these two parameters, which are obtained from Eq. (5.9) and Eq. (5.8), is the target.
3. Target function: The difference between the achieved size distribution and the target Weibull distribution is defined as the Target function. The parameters of cutting boxes are chosen to minimise the value of the Target function, which is to say to minimise the difference between the size distribution that is obtained from the present numerical simulation, and the target Weibull distribution. The algorithm to find a , b and c is shown in Fig. 5.5, which is introduced in detail in the following parts of this section.

Target Weibull distribution The Weibull distribution is widely used to fit the size distribution of fragments resulting from fragmentation process [Shih, 1980]. The Weibull distribution has two parameters m and x_c . m corresponds to the width of distribution of ore fragments [Shih, 1980], whereas x_c is the fragment size corresponding to a volume-fraction of $1-1/e$ in this

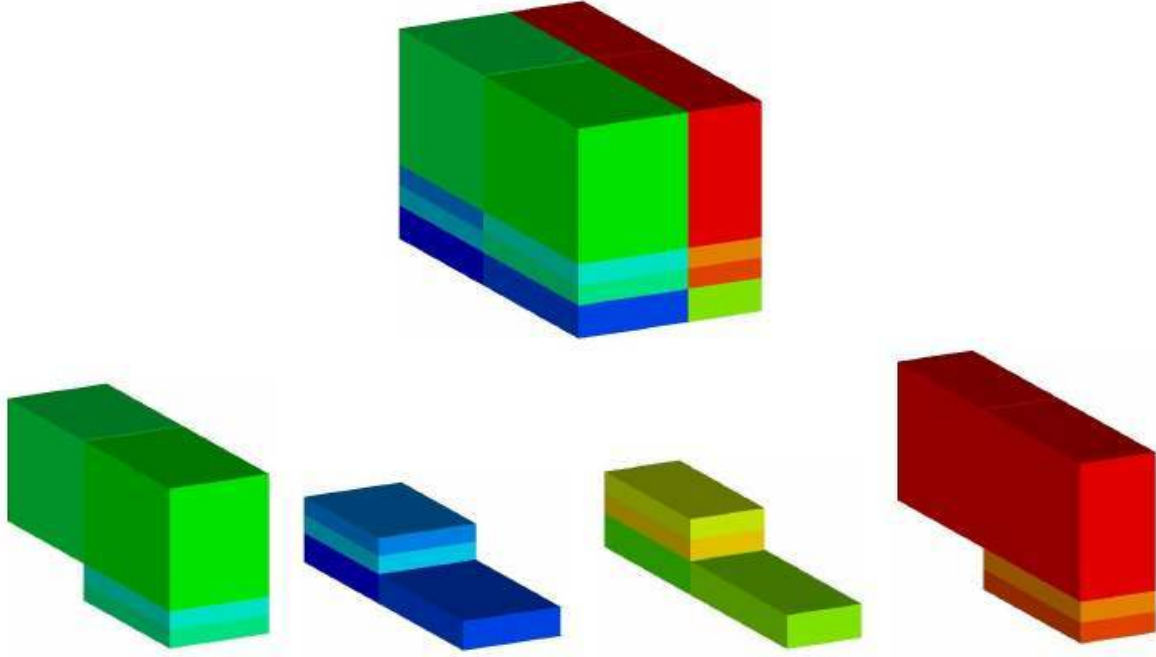


Figure 5.4: Voronoi cells for generating cutting boxes when impact velocity is 20 m/s. The fragmentation pattern parameters are chosen as $a=3$, $b=4$, $c=3$. It is noticed that the voronoi cells are all box-shape.

case [Cheong et al., 2004]. According to the existing experimental results, the number and size of smaller fragments and larger fragments correspond to the impact velocity. It means that, for a given impact velocity, the m and x_c can be determined using a function of the impact velocity. The present work uses Cheong's experimental results [Cheong et al., 2004] to establish the relation between the impact velocity v and the Weibull distribution parameters. Cheong et al. [2004] fragmented a set of brittle spheres using 16-mm bore pressurised gas gun in order to understand the fragmentation mechanism. The experimental data obtained by Cheong et al. [2004] is shown in Fig. 5.6(a) and Fig. 5.6(b). The data points are fitted using the following equations:

$$x_c = 3.9130 - 0.6897(v - 25.9973); \quad (5.8)$$

$$\begin{cases} \text{if } v > 19.54316 & m = 29.1252 - 9.9783(v - 17.0251) \\ \text{if } v \leq 19.54316 & m = 4 \end{cases} \quad (5.9)$$

Hence, for a given impact velocity v , m and x_c can be obtained using the above two equations. The Weibull distribution, with the two parameters obtained from Eq. 5.8 and Eq. 5.9, is defined as the target Weibull distribution, and is denoted as $f_w^e(l)$, where l is the size of fragments.

Measure the performance of fragmentation pattern For a given impact velocity v , without remeshing, regenerating geometry, and computing stress and strain, the cutting boxes

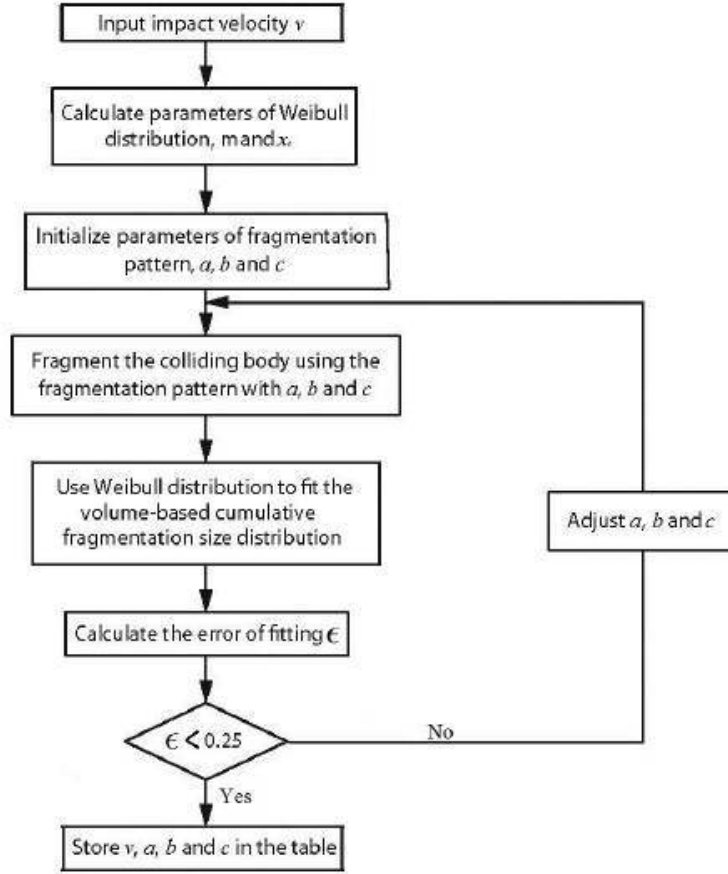


Figure 5.5: Algorithm to find the relation between a special value of v and the parameters of fragmentation pattern, a , b and c .

of fragmentation pattern is used to fragment colliding bodies directly. The property of cutting boxes is controlled by three parameters a , b and c . The size distribution of fragments achieved by fragmentation pattern is called the pattern-based Weibull distribution $f_w^p(l)$. The performance of the fragmentation pattern is measured by comparing the pattern-based Weibull distribution to the target Weibull distribution. The target function, ϵ , which is the error between pattern-based Weibull distribution and the target Weibull distribution, can be expressed as

$$\epsilon = \int_0^{l_{max}} |f_w^p(l) - f_w^e(l)| dl \quad (5.10)$$

For a given impact velocity, the exhaustion method is used to find the parameters of fragmentation. The algorithm is shown in Fig. 5.5, which contains six main steps: First, for a given impact velocity v , two parameters of the Weibull distribution, x_c and m , are calculated using Eq. 5.8 and Eq. 5.9. This Weibull distribution achieved from Eq. 5.8 and Eq. 5.9 is the

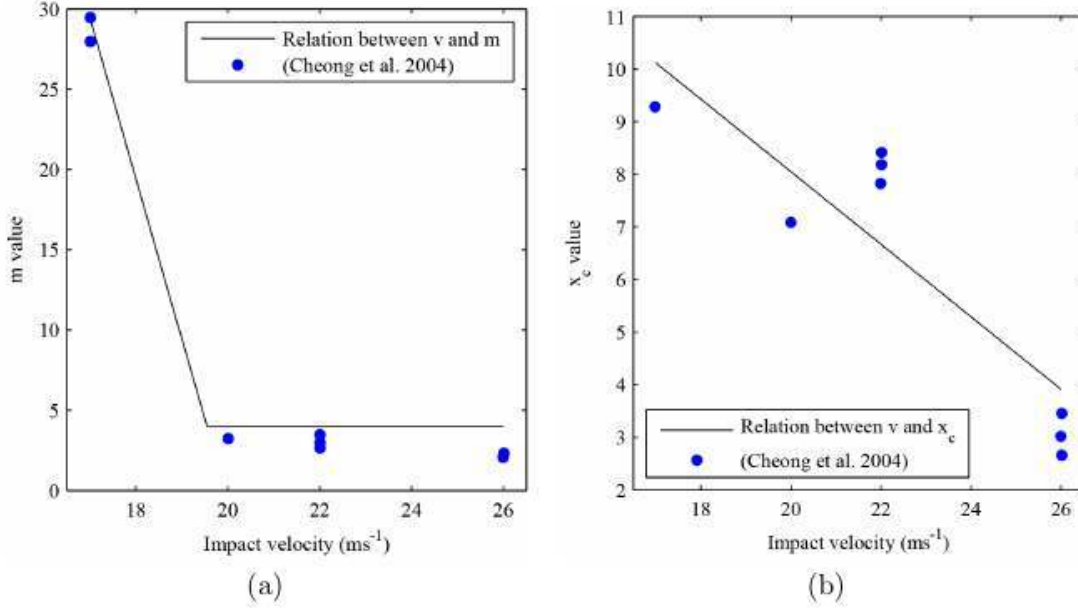


Figure 5.6: Cheong's experimental results define the relation between impact velocity and the two parameters of the Weibull distribution. (a) The experimental relation between impact velocity v and parameter m is fitted using Eq. (5.9); (a) The experimental relation between impact velocity v and parameter x_c is fitted using Eq. (5.8).

target Weibull distribution. Secondly, the parameters of fragmentation pattern, a , b and c , are initialised empirically. Thirdly, use the fragmentation pattern with a , b and c to create cutting boxes and fragment the colliding body. Fourthly, use the Weibull distribution to fit the volume-based cumulative fragment size distribution resulting from fragmentation process using a least squares fit. The Weibull distribution obtained from the present fragmentation pattern is called the pattern-based Weibull distribution. Fifthly, calculate the difference between the pattern-based Weibull distribution and the target Weibull distribution equation, ϵ , using the target function (see Eq. 5.10). Lastly, if $\epsilon < 0.25$, the parameters, a , b and c for impact velocity v are found and stored in Table 5.1. If not, the parameters are adjusted and steps three, four and five are repeated until ϵ is smaller than 0.25.

5.3.4 Determine parameters of fragmentation pattern using a table

Repeat the algorithm above for different impact velocities and insert the found proper parameters of fragmentation pattern into Table 5.1. Because the relation between the impact velocity and the parameters of fragmentation is expressed discretely in Table 5.1, it is impossible to cover all the impact velocities. When a given impact velocity is not covered by Table 5.1, a best approximation is found from Table 5.1 and is used to establish the fragmentation pattern. For example, if the impact velocity is 20 m/s, the best approximated impact velocity in the Table 5.1 is 21.47 m/s, and the parameters of the fragmentation pattern are $a=3$, $b=4$, $c=3$.

Table 5.1: The relation between impact velocities and the parameters of fragmentation pattern

Impact velocity (m/s)	x_c (m)	m	a	c	b	Error
16.66	10.353	32.7691	3	2	2	0.0046
18.38	9.1668	15.6064	4	2	2	0.0921
21.47	7.0356	4	3	4	3	0.2146
23.68	5.5113	4	4	6	4	0.0858
24.66	4.8354	4	4	8	5	0.0353
25.32	4.3802	4	5	10	5	0.0386
25.42	4.3112	4	5	60	5	0.1624
25.87	4.0008	4	6	60	5	0.1385
26.26	3.7318	4	6	60	6	0.1222

5.4 Case study: Fragmentation of a single sphere

The performance of the fragmentation pattern is validated by modelling the fragmentation of a single sphere, as shown in Fig. 5.7. The diameter of the sphere is 12.7 mm, and its density is 100 kg/m³. The restitution coefficient is zero, and friction is ignored. The sphere falls downwards and collides with a rigid ground at different impact velocities of 17 m/s, 20 m/s, and 26 m/s. The snapshots of the collision and fragmentation process for an impact velocity of 26 m/s are shown in Fig. 5.7. The restitution coefficient is zero and the friction is ignored, which means that the relative velocities are set to zero after the collision, therefore the absolute values of the change of the relative velocities and impact velocities are the same. Using Table 5.1, the simulator finds the corresponding parameters of the fragmentation pattern automatically. For an impact velocity of 17 m/s, the best-approximated velocity in Table 5.1 is 16.66 m/s and the parameters of the fragmentation pattern are $a=3$, $b=2$, and $c=2$. For an impact velocity of 20 m/s, the best-approximated velocity is 21.47 m/s, and the parameters of the fragmentation pattern are $a=3$, $b=4$, and $c=3$. For an impact velocity of 26 m/s, the best-approximated velocity is 26.26 m/s and the parameters of fragmentation pattern are $a=6$, $b=60$, and $c=6$. These parameters are used to generate cutting boxes for different impact velocities, and to fragment the sphere.

Fig. 5.8 shows the shape of the fragments resulting from the fragmentation at impact velocity of 20 m/s. The sphere shatters into eight fragments. Figure 5.9 shows the shape of the fragments resulting from fragmentation at an impact velocity of 26 m/s. As shown in Fig. 5.9, when the impact velocity reaches 26 m/s, the region around the contact points is fragmented into a large number of small fragments. The rest of the sphere is fragmented into several main fragments. This distribution of small fragments and main fragments is in agreement with the experimental observations in [Cheong et al., 2004]. Comparing Fig. 5.8 to Fig. 5.9, the number of fragments increases with increasing impact velocity. The two-parameter Weibull distribution equation is then used to fit the volume-based cumulative fragment size distribution, by a least-squares fit. As shown in Fig. 5.10, the size distributions obtained from the simulation are marked by

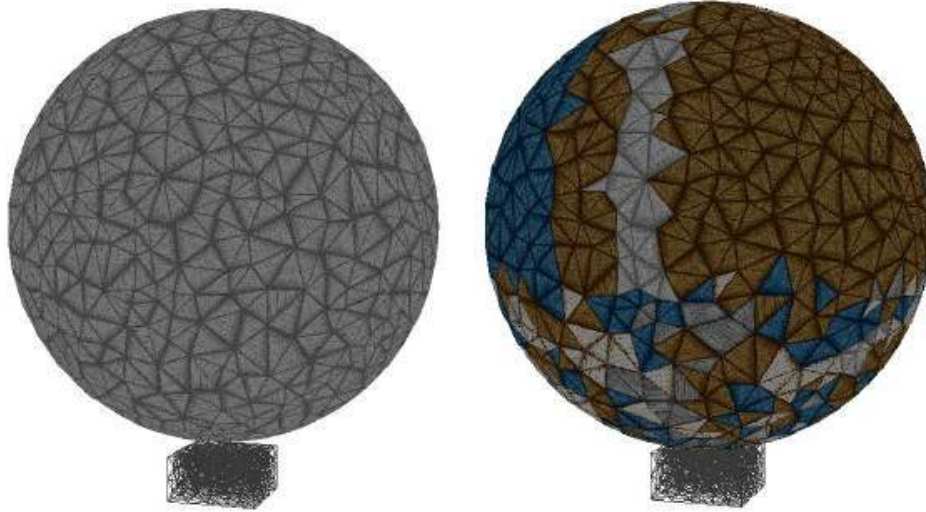


Figure 5.7: A sphere falls downwards, collides with a rigid ground at with impact velocity 26 m/s, and fragments into a large number of fragments. Each fragment is coloured arbitrarily for visualisation purposes.

circles, regular triangles and inverted triangles, for impact velocities of 17 m/s, 20 m/s, and 26 m/s respectively. Then the numerical data are fitted using the Weibull distribution equation. Three fitting curves for different impact velocities are shown in Fig. 5.10. The volume-based cumulative fragment size distribution can be fitted very well by the Weibull distribution. Cheong et al. [2004] used a gas gun to crush the spheres, whose diameters were 12.7 mm, with different impact velocities of 17 m/s, 20 m/s and 26 m/s. The two two-parameter Weibull distribution equation is then used to fit the volume-based cumulative fragment size distribution obtained from his experiments. The fitting curves drawn by Cheong are also plotted in Fig. 5.10 for comparison. The Weibull distributions obtained from the numerical results are close to the Weibull distributions obtained from the experimental data.

5.5 Case study: Fragmentation of a realistically shaped rock

The performance of the fragmentation pattern is further validated by modelling the fragmentation of a realistically-shaped rock. This rock falls downwards, collides with a rigid ground, and breaks up. The geometry of this rock is shown in Fig. 5.11, and the size of this rock is approximately $3\text{m} \times 4\text{m} \times 5\text{m}$. The density is 1 kg/m^3 , and the restitution coefficient and friction coefficient are both set to be zero. A total of 1458 tetrahedral elements are used to discrete this realistically-shaped rock. Three different impact velocities are considered: 17 m/s, 20 m/s, and 26 m/s. The restitution coefficient and friction coefficient are both zero, it means that the relative velocities is vanished to be zero after collision, therefore the absolute of initial impact velocities are same with the change of relative velocities. Using Table 5.1, the simulator finds

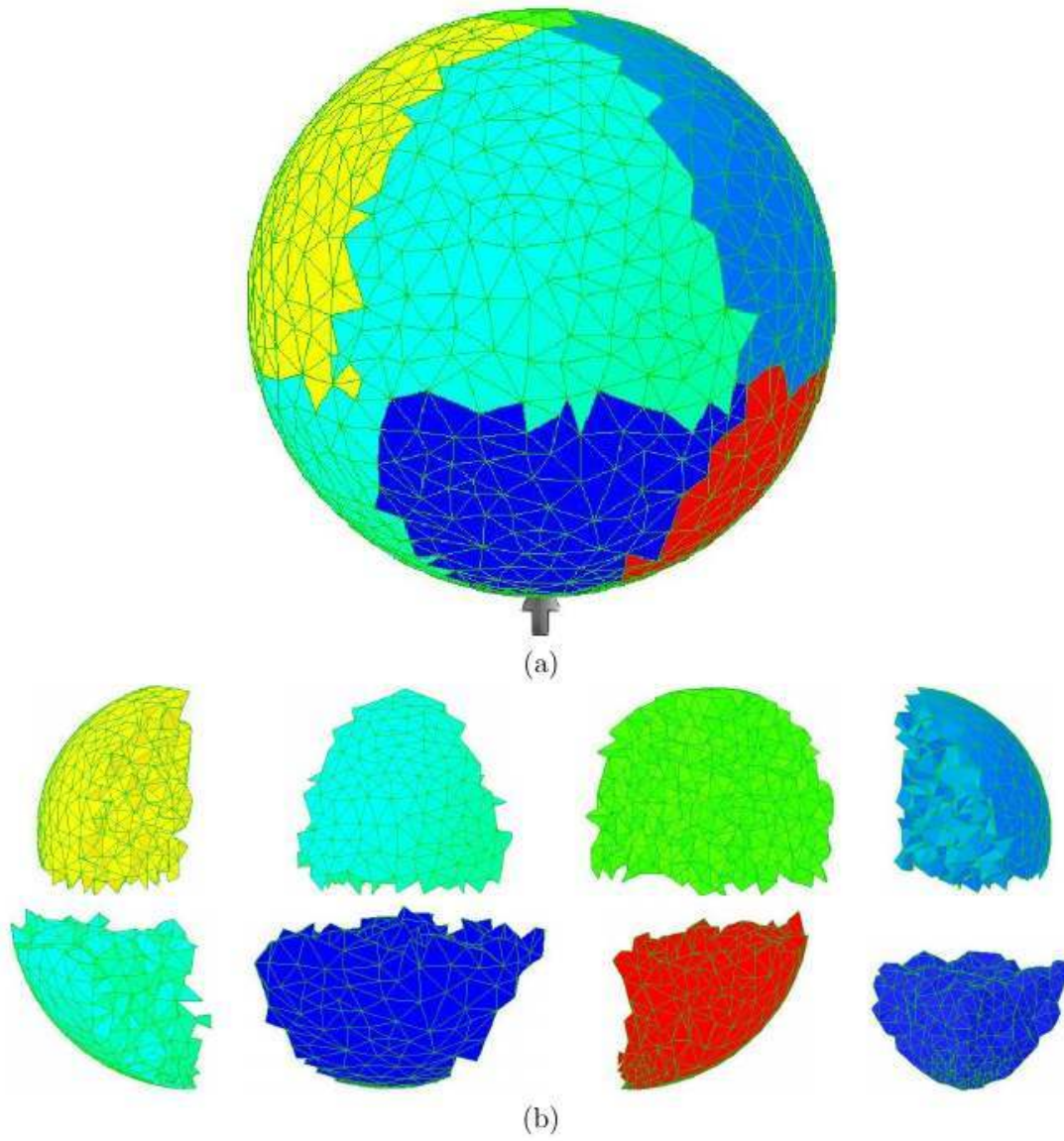


Figure 5.8: Shape of fragments achieved from the fragmentation of a single sphere, with impact velocity 20 m/s. (a) shows the fragments as a whole, where the fragments are coloured arbitrarily to distinguish each other. (b) shows these eight fragments separately. A total of eight fragments are obtained.

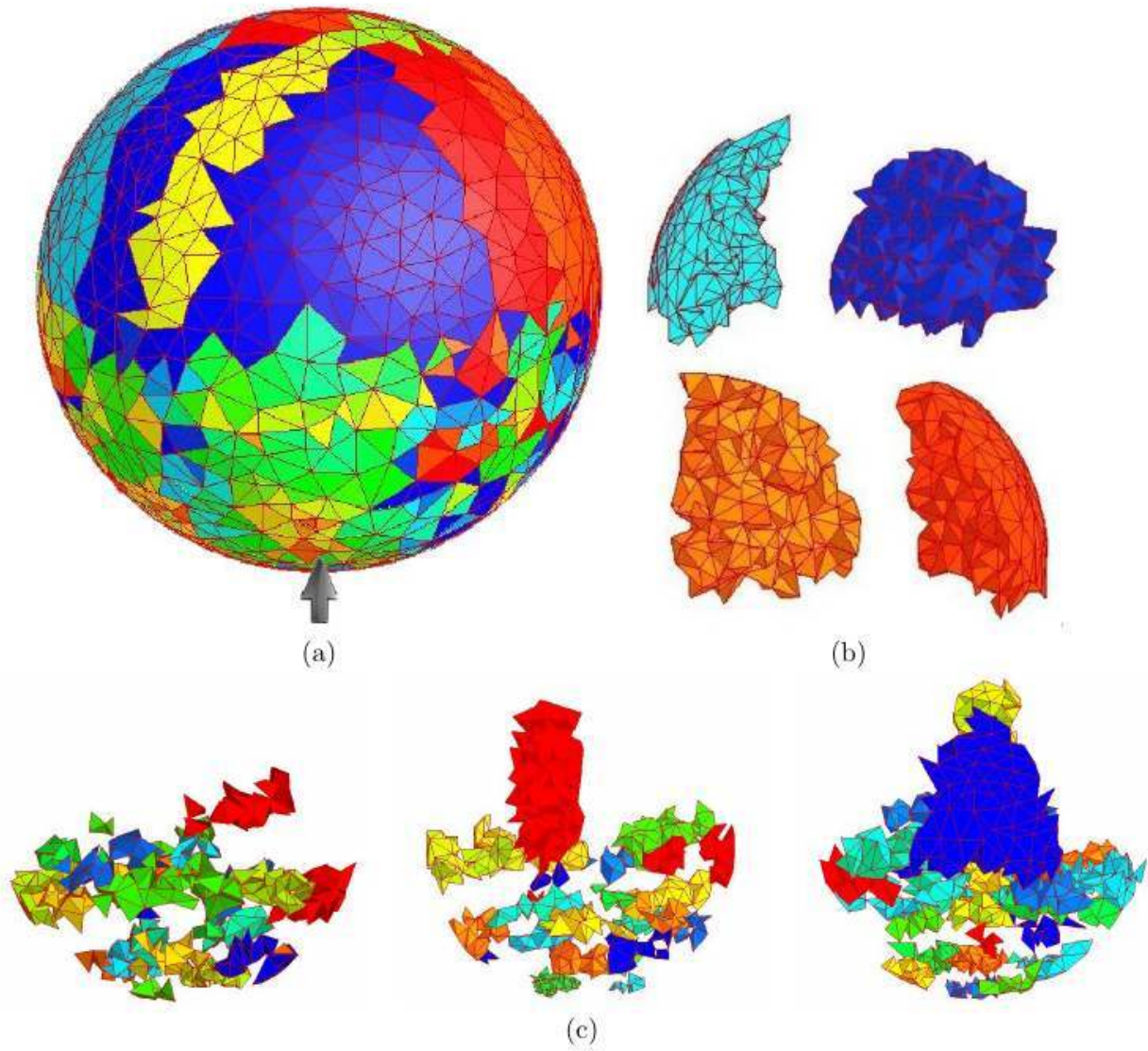


Figure 5.9: Shape of fragments achieved from the fragmentation of a single sphere, with impact velocity 26 m/s. (b) and (c) show these fragments separately. The smaller fragments are in the vicinity of the contact point, while larger fragments form away from the contact point.

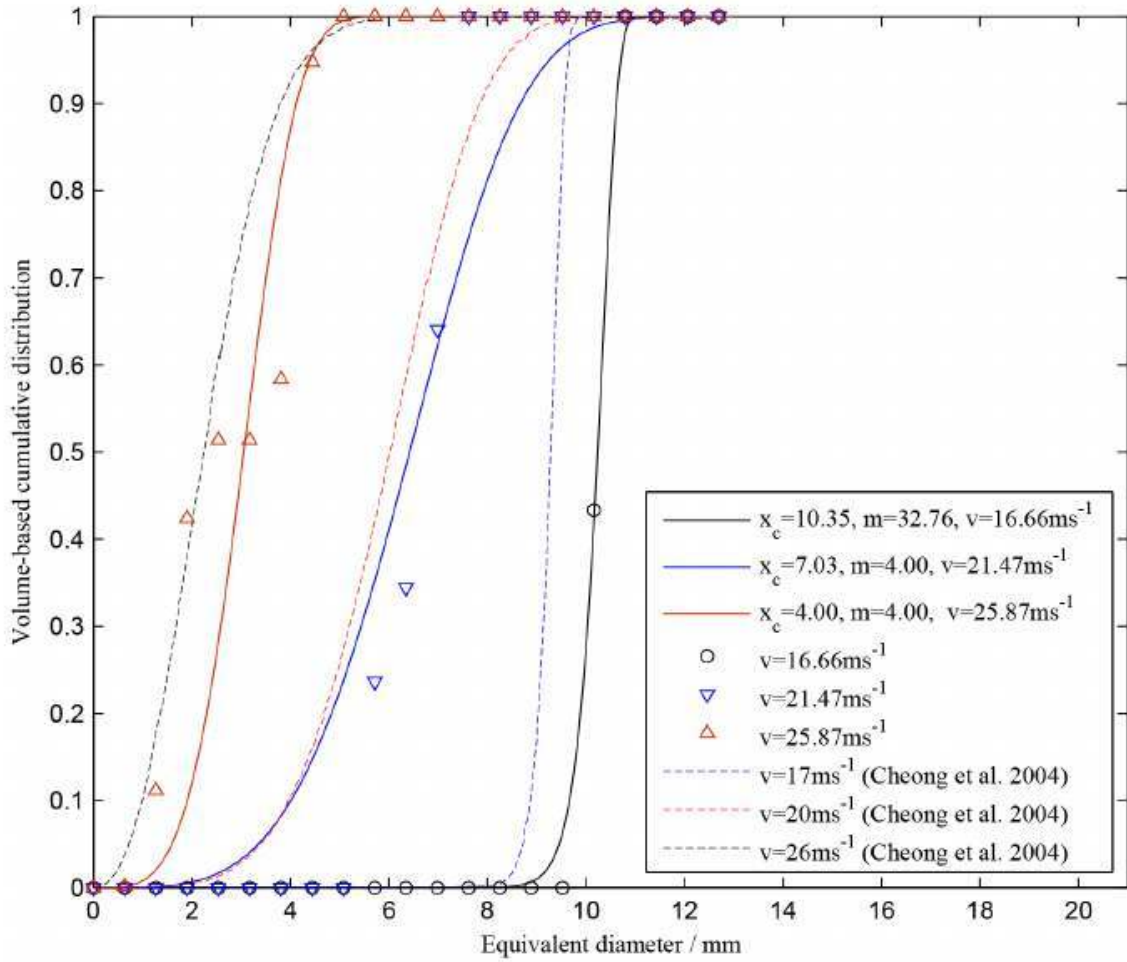


Figure 5.10: Volume-based cumulative size distribution resulting from the fragmentation of a single sphere is fitted by two-parameter Weibull distribution equation. The size distributions obtained from simulation are marked by circles, regular triangles and inverted triangles for impact velocities of 17 m/s, 20 m/s, and 26 m/s respectively. The numerical data are then fitted using the Weibull distribution. Cheong et al. [2004] used a gas gun to crush the spheres, whose diameters were 12.7 mm, with impact velocities of 17 m/s, 20 m/s and 26 m/s. The Weibull distributions that were fit to Cheong's experimental data are plotted for comparison. The Weibull distributions obtained from the numerical results are close to the Weibull distributions obtained from the experimental data.

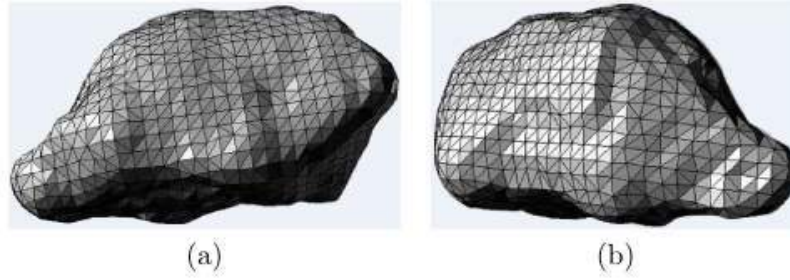


Figure 5.11: Geometry of a realistic shaped rock.

the corresponding parameters of the fragmentation pattern automatically. For an impact velocity of 17 m/s, the best-approximated velocity in Table 5.1 is 16.66 m/s, and the parameters of the fragmentation pattern are $a=3$, $b=2$, and $c=2$. For an impact velocity of 20 m/s, the best-approximated velocity is 21.47 m/s, and the parameters of the fragmentation pattern are $a=3$, $b=4$, and $c=3$. For an impact velocity of 26 m/s, the best-approximated velocity is 26.26 m/s, and the parameters of the fragmentation pattern are $a=6$, $b=60$, $c=6$. These parameters are used to generate cutting boxes for different impact velocities, and to fragment the sphere.

The shape of fragments resulting from the fragmentation at an impact velocity of $v=26$ m/s is shown in Fig. 5.12. This rock is fragmented into a number of smaller fragments, and several larger fragments. The smaller fragments are mostly distributed in the vicinity of the contact point. A similar distribution of smaller fragments and larger fragments are also observed experiments of the fragmentation of a single sphere, which has been discussed in Section 5.4. Then, two-parameter Weibull distribution is used to fit the volume-based cumulative fragment size distribution of fragments with the least-squares fit, which is shown in Fig. 5.13. It is seen that the size distribution of fragments resulting from fragmentation can be well fitted by a two-parameter Weibull distribution.

To test the stability of the simulator, a hammer is allowed to fall downwards, and to fragment this realistically-shaped rock, which is placed in a cuboid container with rigid boundary walls. As shown in Fig. 5.15, the width of the container is $6\text{m} \times 6\text{m}$. The density of the hammer is 100 kg/m^3 . Six elements are used to discretise the hammer, and 3448 elements are used to discretise the rigid container. The threshold of fragmentation initialisation is set to be 0 m/s, which means that the fragmentation process will never stop until all of the fragments are small fragments. The fragmentation process is shown in Fig. 5.14 and Fig. 5.15. It is shown that after 2000 simulation steps, the realistically-shaped rock has fragmented into a set of small fragments, most of which are single-element fragments. This test shows the ability of the simulator to handle huge number of small fragments in a stable manner.

This test is run on a personal computer which uses Intel Xeon E5640 CPU with 2660MHz, 12GB installed memory, and four cores. This test spent approximately two minutes.

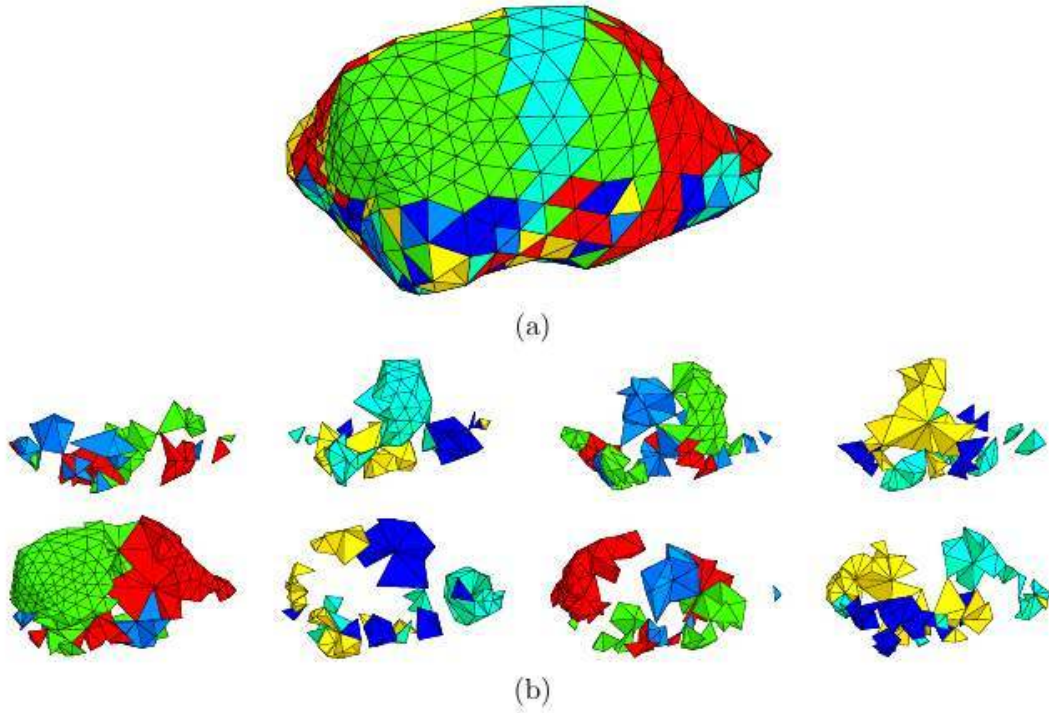


Figure 5.12: Shape of fragments resulting from the fragmentation of the realistic shaped rock at impact velocity 26 m/s. This realistically shaped rock fragments into a number of smaller fragments and several larger fragments. The smaller fragments mostly are distributed in the vicinity of the contact point. The similar distribution of smaller fragments and larger fragments are also observed experiments of the fragmentation of a single sphere, which has been discussed in Section 5.4.

5.6 Case study: Fragmentation of $5 \times 5 \times 5$ spheres

This fragmentation pattern is used to fragment $5 \times 5 \times 5$ ore particles, in order to validate the performance of the code for multiple colliding bodies. The fragmentation of multiple bodies is critical to control the fragmentation process in mining engineering. An example is the high pressure grinding rolls (HPGR) that are widely used to comminute ore particles. Studying confined multiple-particle breakage is critical to reduce comminution energy and enhance downstream mineral recovery in pressure grinding rolls [Ali and Bradshaw, 2009, 2011, Fitzgibbon and Veasy, 1990, Walkiewicz et al., 1993].

In this test, the breakage of a confined particle bed, which consists of $5 \times 5 \times 5$ ore particles, is investigated. These seventy-five ore particles are placed in a static and rigid container, which has static and rigid boundary walls in five directions, as shown in Fig. 5.16. The dimension of the static container are $60 \text{ m} \times 60 \text{ m} \times 110 \text{ m}$. The height, width and length of the particle bed are all 60 m . The ore particles are spherical, and their diameter is 12 m . A cuboid hammer moves downwards and compresses the ore particles. The dimensions of the hammer are $60 \text{ m} \times 60 \text{ m} \times 25 \text{ m}$. The ore particles are constrained in the other five directions, and no ore particle can escape from the container. The density of the ore particles is 1 kg/m^3 , and each ore particle is discretised using 550 tetrahedral elements. The container is discretised into 4025 tetrahedral elements, and the hammer is discretised into six tetrahedral elements. The restitution coefficient and friction coefficient of the ore particles are set to 0.8 and 0.4, respectively. The restitution coefficient and friction coefficient of the container and hammer are 0.7 and 0.4, respectively. To study the influence of the crushing velocity, two tests DC1f and DC2f are conducted: in DC1f, the crushing velocity is 10 m/s , and in DC2f, the crushing velocity is 100 m/s . The resulting images for DC1f are shown in Fig. 5.17. The ore particles are coloured to enhance visualization. The resulting fragmentation of DC2f is shown in Fig. 5.18. For a crushing velocity of 10 m/s , the hammer starts to compress the ore particles at step 598. When the impact velocity is 100 m/s , the first collision between the hammer and ore particles is detected at step 541. Fragmentation develops from the top of the ore particles to the bottom with as the simulation progresses. To show the change of volume of the ore particles, the ore particles are coloured based on their volume, as shown in Fig. 5.19 and Fig. 5.20 for DC1f and DC2f, respectively. In these two graphs, the intact ore particles, of volume 868 m^3 , are in orange and small ore fragments whose volume is under 200 m^3 are in blue. For both DC1f and DC2f, it can be seen that at early stages, catastrophic damage mainly occurs at the top layer of particles.

As the simulation progresses, the fragmentation develops from the top to the bottom of the sphere pack. Meanwhile, the size distribution of ore particles of DC1f at step 658 and the size distribution of ore particles of DC2f at step 650 are both plotted in Fig. 5.21. The $-x$ axis is the normalised fragmentation volume and y axis is the percent of total volume. The Weibull distribution is used to fit the size distribution of crushed fragments, whose volume is smaller than 868 m^3 (the volume of intact ore particles is 868 m^3), as shown in Fig. 5.22. It is found that size distribution in DC1f and DC2f are similar, even the numerical of damage fragments

increase significantly with the increasing of crushing velocity. It is shown that the fragmentation increases with the increasing of impact velocities. For instance, for DC2f, crushing velocity is 10 m/s, approximately 43% of fragments are smaller than 600 m^3 . For DC1f, the crushing velocity is 100 m/s, and approximately 18% of fragments are smaller than 600 m^3 . This means that if 600 m^3 is the maximum limit to collect ore fragments, the qualified ore fragments produced in DC1f is 25% more than the qualified ore fragments produced in DC2f. The above observation is agree with existing numerical results [Ali and Bradshaw, 2011], which models the ore particle bed breakage using the discrete element method. In reference [Ali and Bradshaw, 2011], it is shown that there is catastrophic damage the top-layered ore particles. This is also found in other existing experimental and numerical results [Khanal et al., 2007, Liu et al., 2005, Tsoungui et al., 1999].

The relation among total computation time, computation time of fragmentation, and number of contacts at each simulation step for DC1f and DC2f are shown in Fig. 5.23. The number of contacts is scaled by $1/50$ in Fig. 5.23(a), and by $1/40$ in Fig. 5.23(b). The total computational time of each simulation step increases significantly with the increase in the number of contacts. More fragments are generated in DC2f ($v=100 \text{ m/s}$) than in DC1f($v=10 \text{ m/s}$); therefore, the total computational time of each simulation step increases more in DC2f rather than the total computational time of each simulation step in DC1f. In Fig. 5.23(b) the total computational time and the number of contacts at each simulation step increase at the same. However, the time of fragmentation does not significantly increase with the increasing of number of contacts. The average computational times per contact in in DC1f and in DC2f are shown in Fig. 5.24. The number of contacts for DC1f is scaled by $1/1 \times 10^6$, and is shown in Fig. 5.24(a). The number of contacts for DC2f is scaled by $1/2 \times 10^6$, and is shown in Fig. 5.24(b). The average computational time per contact increases with the increasing of number of contacts, the time of collision detection increases with the increasing of connection among contacts.

These two tests are run on a personal computer which uses Intel Xeon E5640 CPU with 2660MHz, 12GB installed memory, and four cores. Both two tests spent approximately six hours to run 650 simulation steps.

5.7 Conclusions and future work

In order to meet the requirement of the fragmentation simulation of hundreds and thousands of colliding bodies, a low-cost fragmentation pattern based approach is proposed. Using the fragmentation pattern, the colliding bodies are fragmented directly, without simulating crack propagation, thereby avoiding re-meshing and re-creating solid geometry during the simulation. The parameters of the fragmentation pattern are derived from impact velocities. The relation between impact velocities and fragmentation pattern parameters is described discretely. The fragmentation pattern is established by learning from existing numerical and experimental results. The performance of fragmentation pattern is controlled using the Weibull distribution,

fitted to the volume-based cumulative distribution.

This fragmentation pattern was used to fragment a single sphere. It was found that the present methodology is able to capture some phenomenon that are observed in experiments. For example, most of the smaller fragments are located close to the contact point. The volume-based cumulative distribution achieved using present method agrees to the experimental results well. Then, fragmentation pattern is used to crush a realistically shaped rock. Similar to the previous simulation, the volume-based cumulative distribution that is obtained, follows the Weibull distribution very well. Finally, the presented method is used to crush $5 \times 5 \times 5$ spheres in a static and rigid container. Numerical results show that the average size of fragments decreases with the increasing of impact velocity. The above results agree with numerical and experimental results reported in the literature.

However, the present methodology can still be improved. This fragmentation pattern is relatively simple, and ignores many factors influencing fragmentation. For example, the present fragmentation pattern only learns from the existing fragmentation of spheres. However, in mining engineering, the shape of the rocks is arbitrary. Many researchers have shown that the shape of colliding bodies has a significant impact on the fragmentation behaviour [Kevin, 2002, Mair et al., 2002, Anthony and Marone, 2005, Abe and Mair, 2009, Morgan and Boettcher, 1999, Grady and Kipp, 1980, Margolin, 1983, Denoual and Hild, 2000]. An obvious extension to this algorithm is to consider the fragmentation of multiple shapes of fragments for the upscaling of pattern formation during impact.

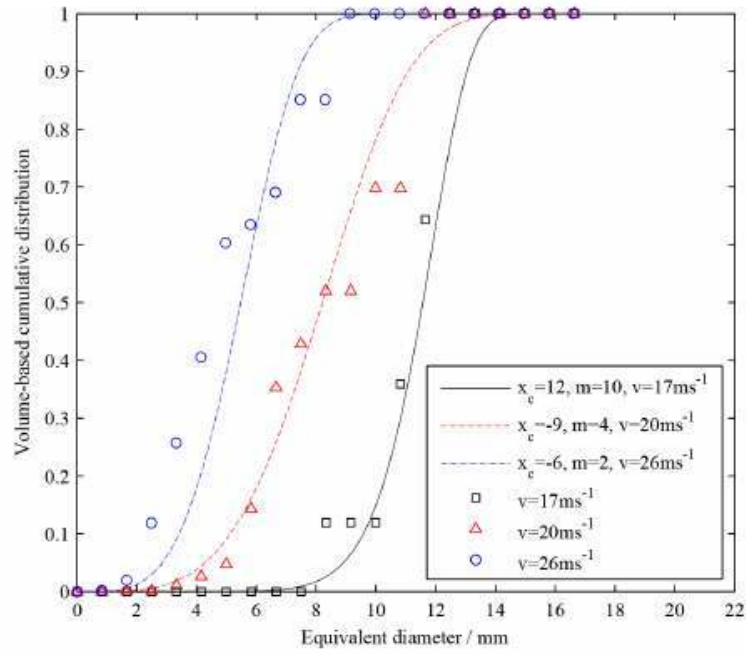


Figure 5.13: The Weibull distribution is employed to fit volume-based cumulative size distribution of fragments resulting from the fragmentation of realistically shaped rock. The size distributions for three different impact velocities all can be well fitted by the Weibull distribution.

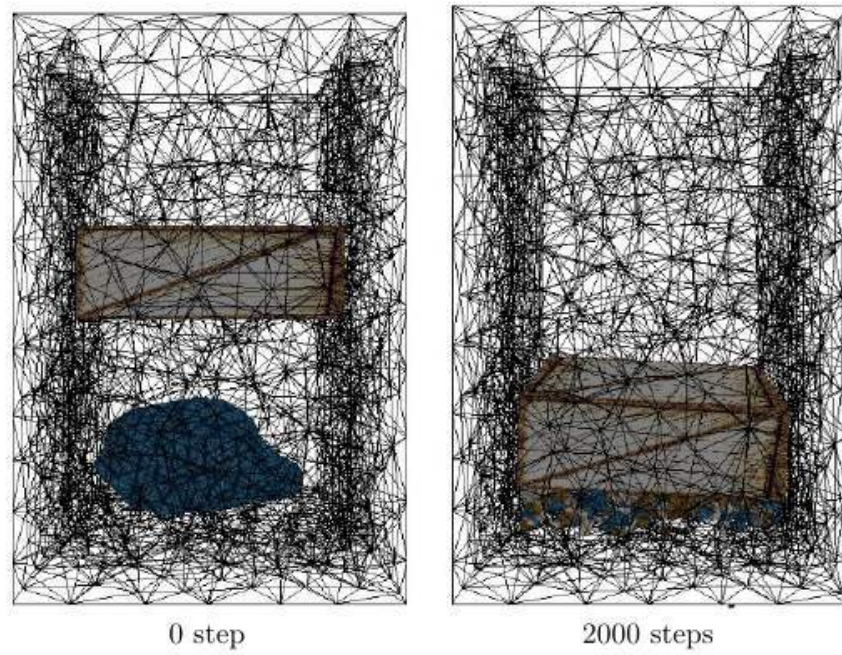


Figure 5.14: A hammer falls downwards and shatters the realistically shaped rock, which is placed in a container with static boundary walls. The static container is shown by the grid. After 2000 simulation steps, this rock has fragmented into a set of small fragments. Most of the fragments are single-element fragments.

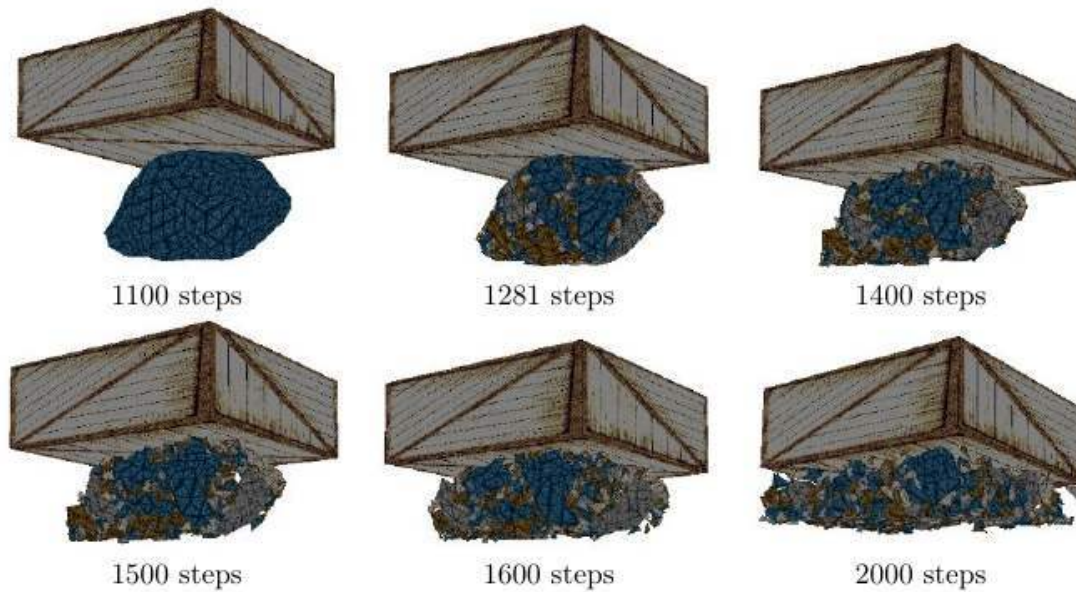


Figure 5.15: Zoom in on the fragmentation process of a realistically shaped rock. After 2000 simulation steps, this rock has fragmented into a set of small fragments. Most of the fragments are single-element fragments.

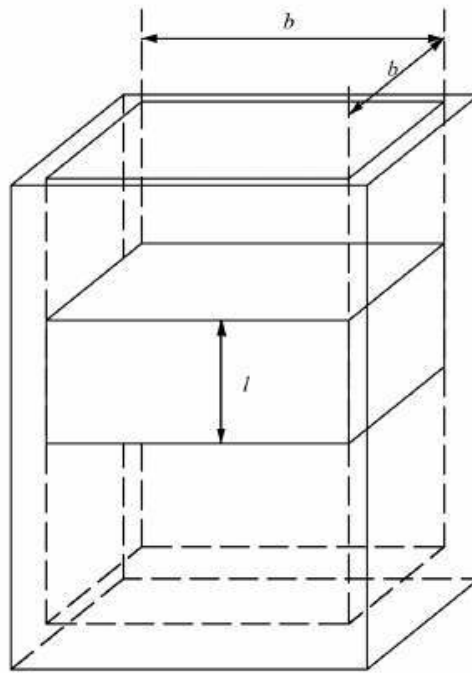


Figure 5.16: Geometry of container. The dimension of the container is $b \times b$, where $b=60$ m. The depth of the hammer is $l=25$ m.

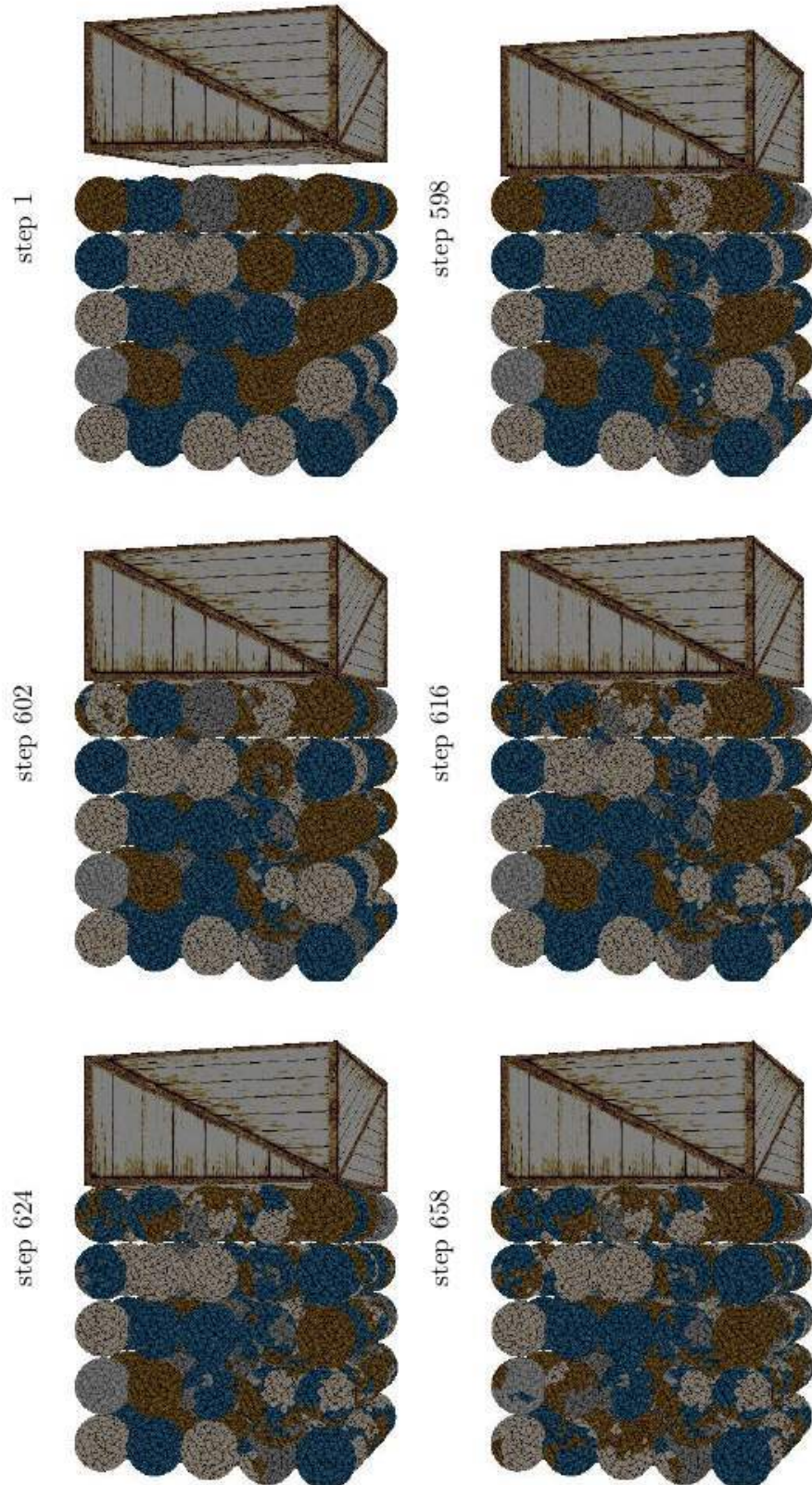


Figure 5.17: The snapshots for DC1f. The first collision between hammer and ore particles is detected at step 598. Fragmentation develops from the tops of the ore particles to the bottoms of the ore particles, with the increase of simulation steps.

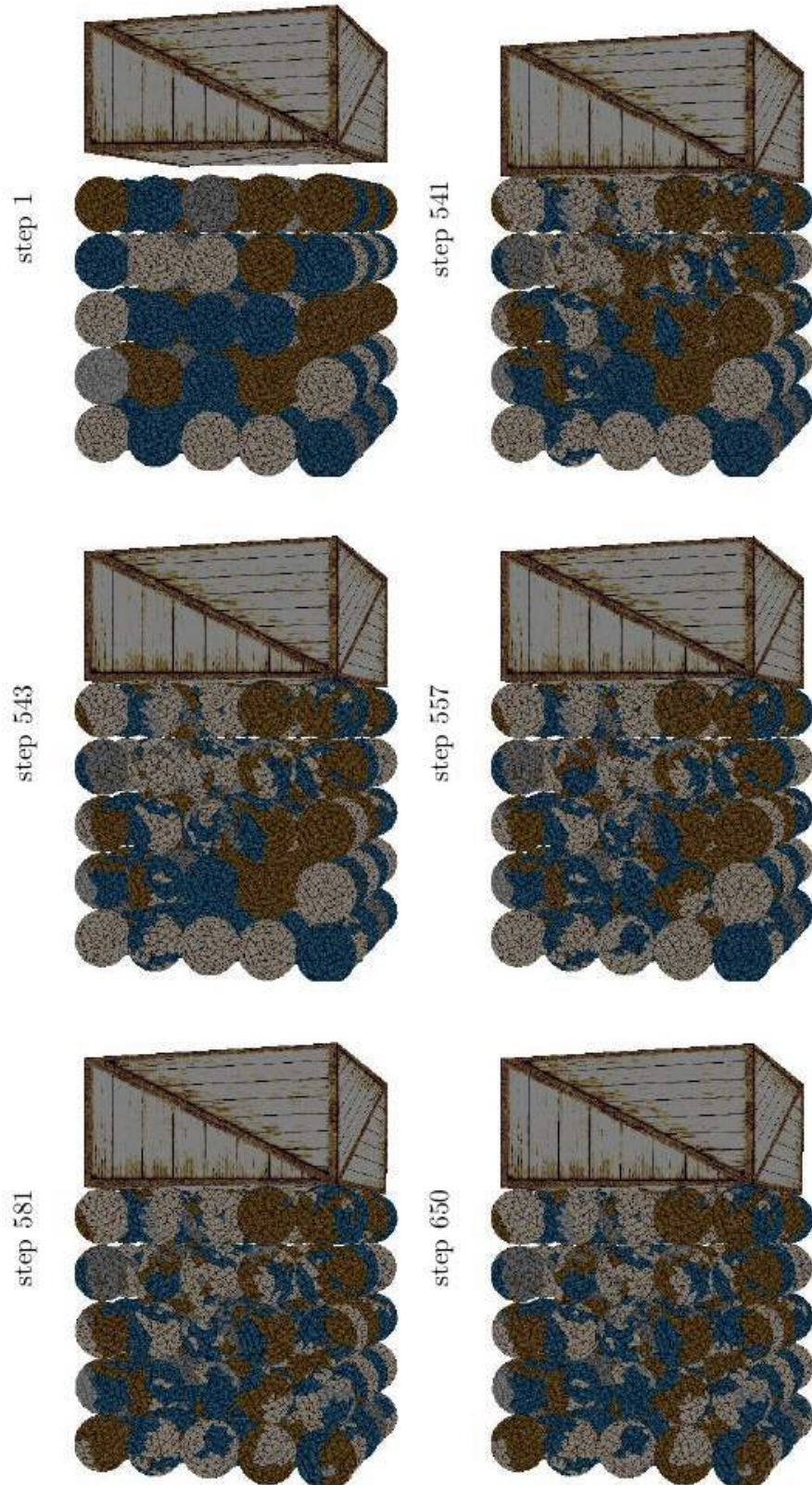


Figure 5.18: The snapshots DC2f. The first collision between hammer and ore particles is detected at step 541. Fragmentation develops from the tops of the ore particles to the bottom of the ore particles, with the increase of simulation steps.

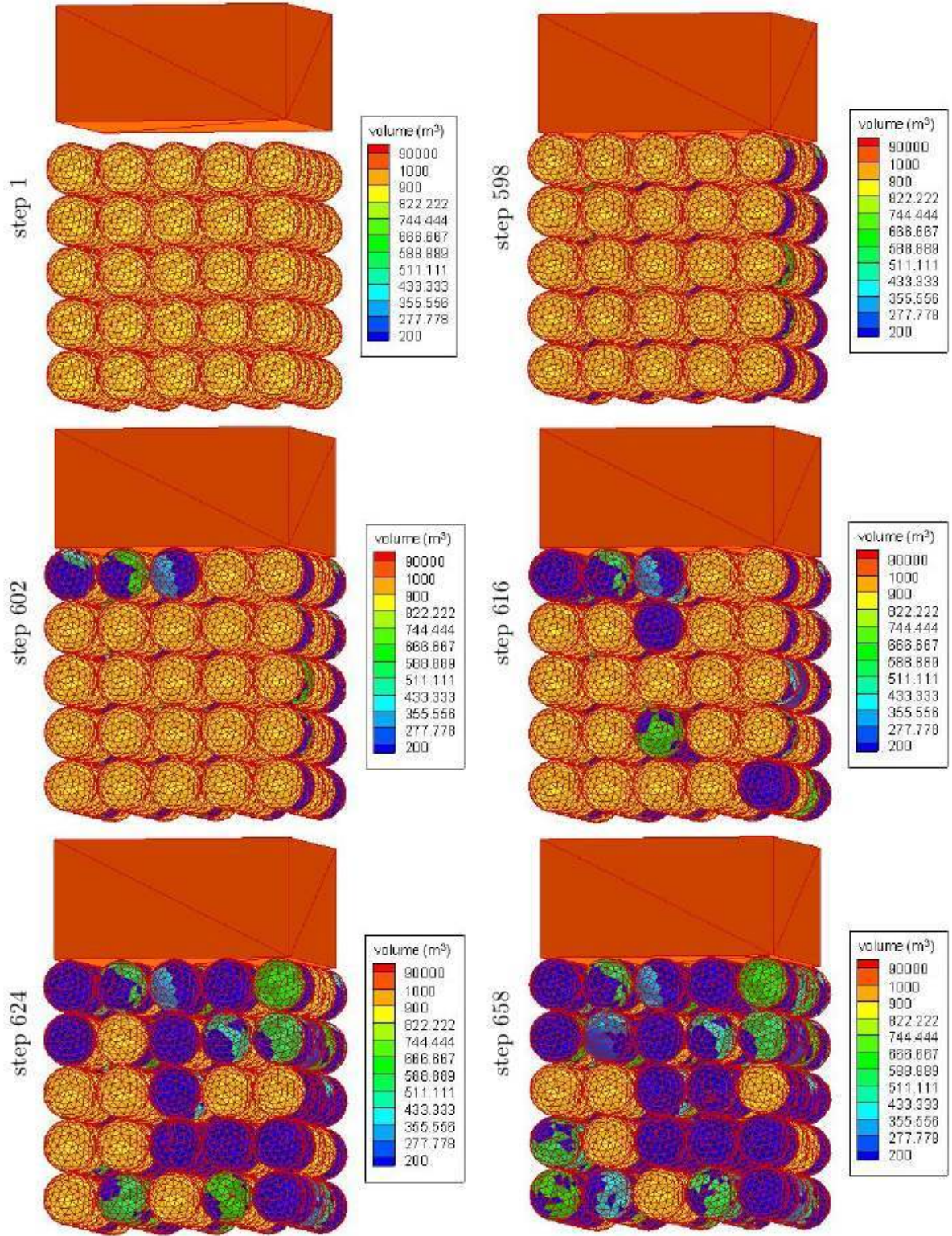


Figure 5.19: The snapshots DC1f, where the fragments are coloured according to their volume. At the early stage, the smaller fragments, which are blue, only exist at the top layered particles. With the evolution of simulation steps, fragmentation develops from the top to the bottom of spheres.

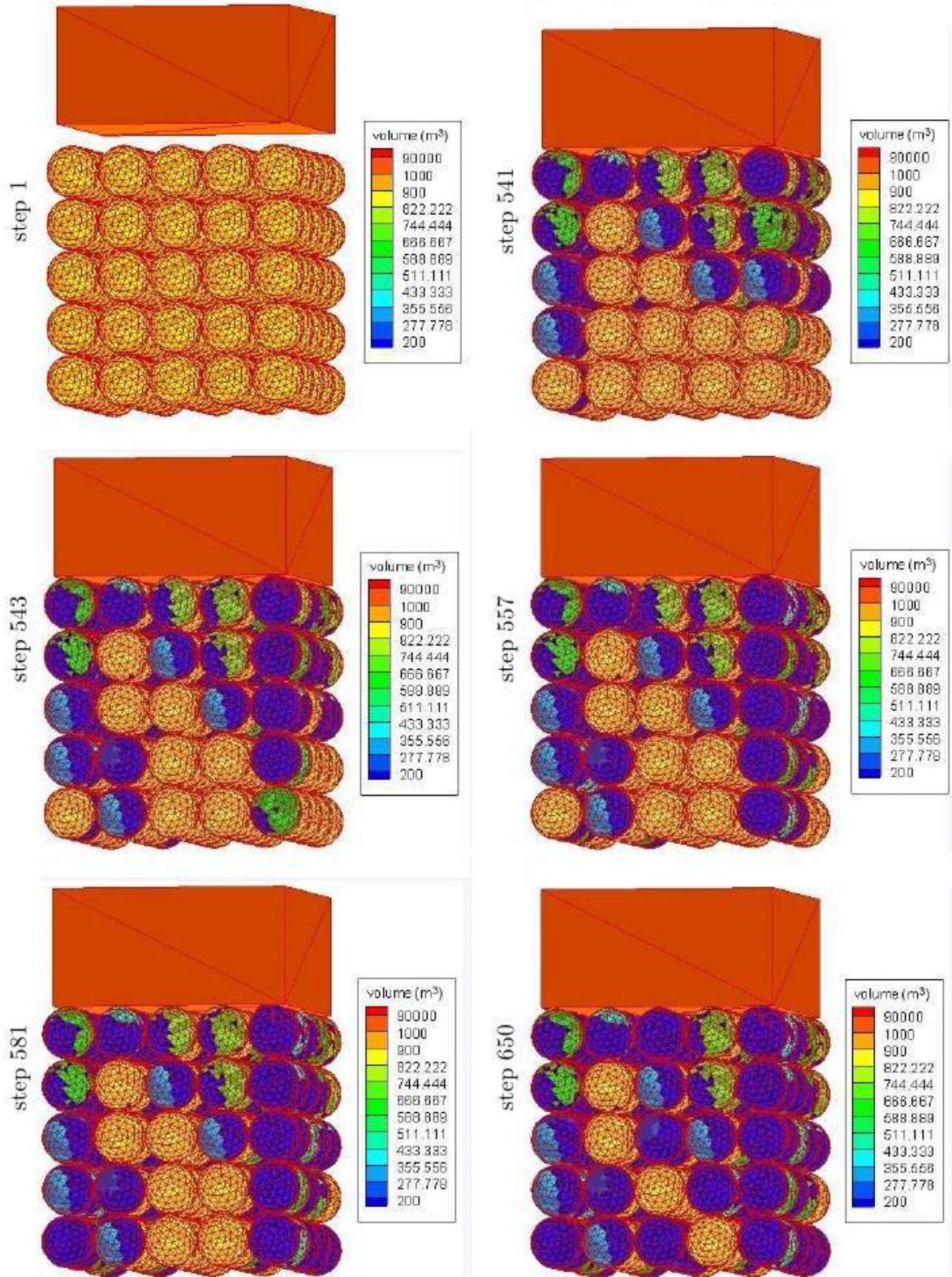


Figure 5.20: The snapshots DC2f, where the fragments are coloured according to their volume. At the early stage, the smaller fragments, which are blue, only exists at the top layered particles. With the evolution of simulation step, the fragmentation develops from the top to the bottom of spheres.

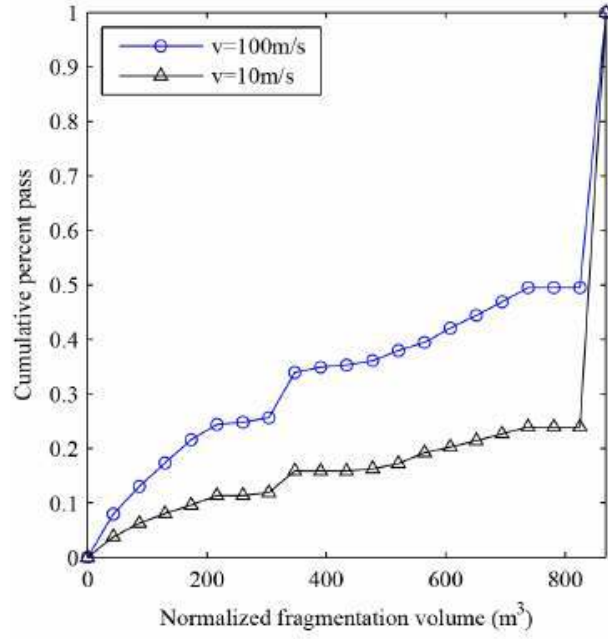


Figure 5.21: Size distribution of DC1f ($v=10$ m/s) and DC2f($v=100$ m/s). The size of fragments achieved from fragmentation decreases with the increasing of impact velocities.

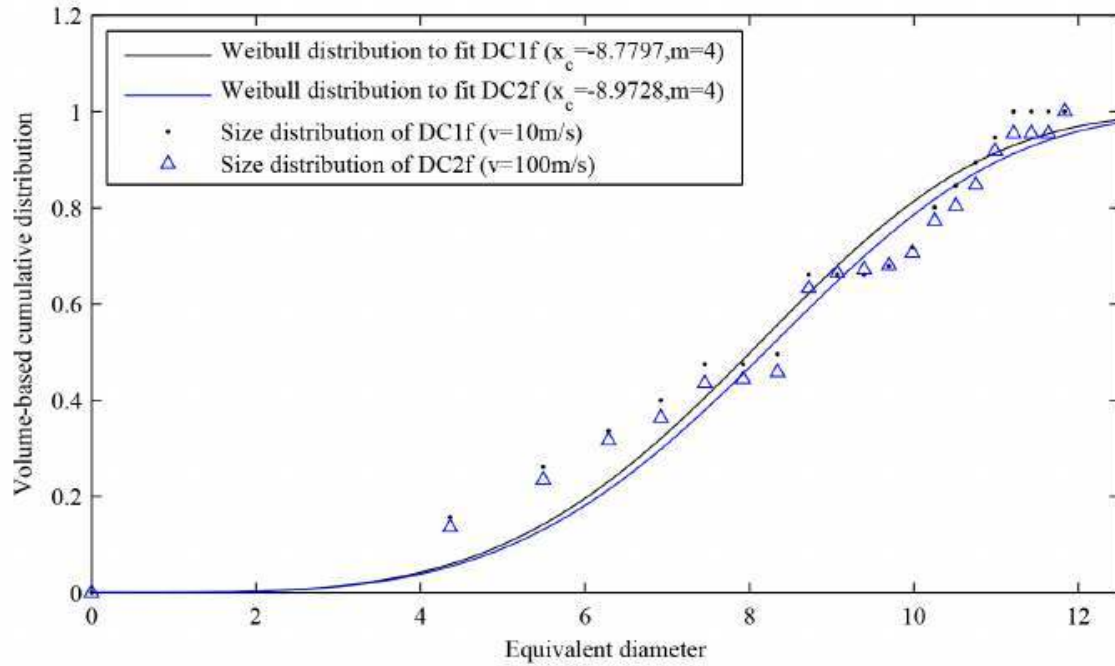
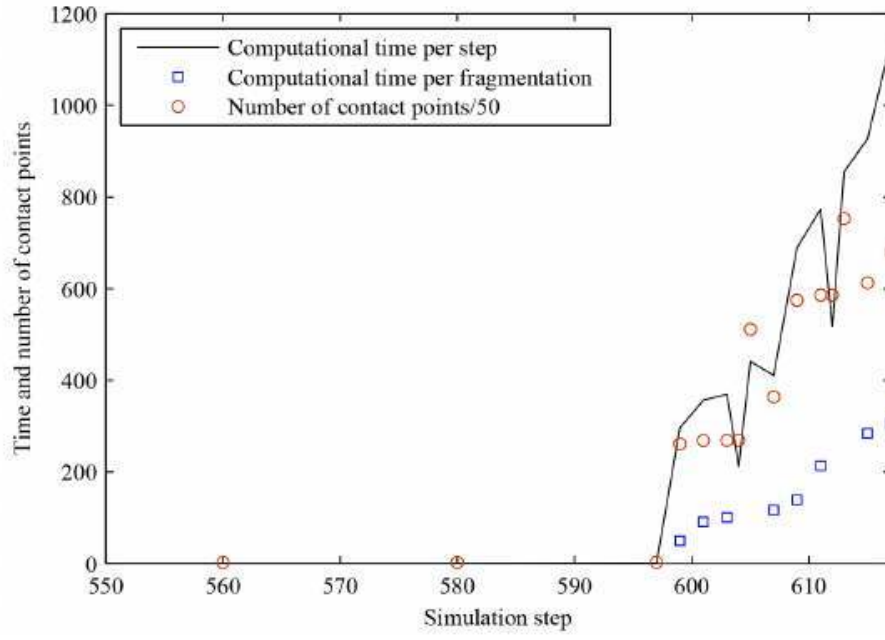
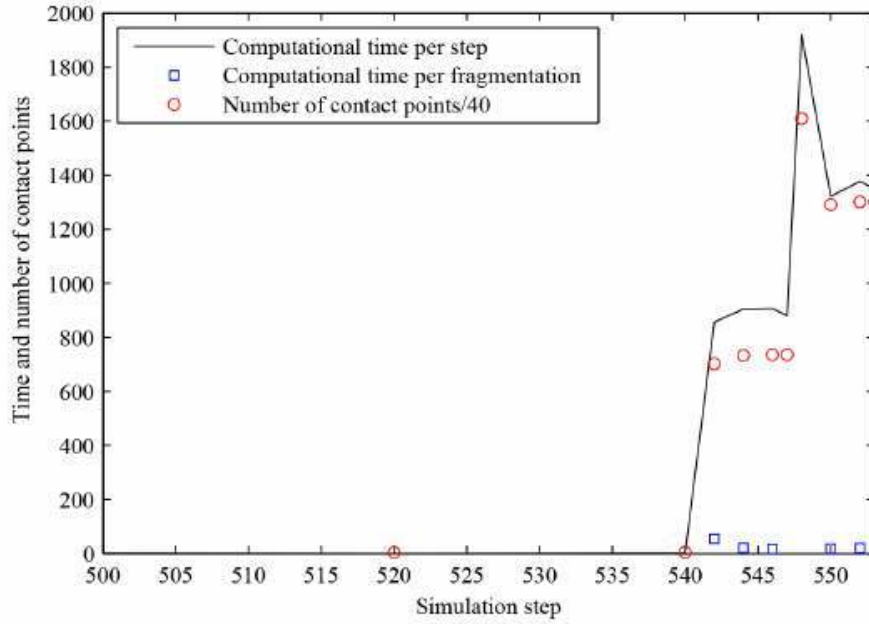


Figure 5.22: The Weibull distribution is used to fit the size distribution of DC1f and DC2f. Size distributions in DC1f and DC2f are similar. The numerical of damage fragments increase significantly with the increasing of crushing velocity.

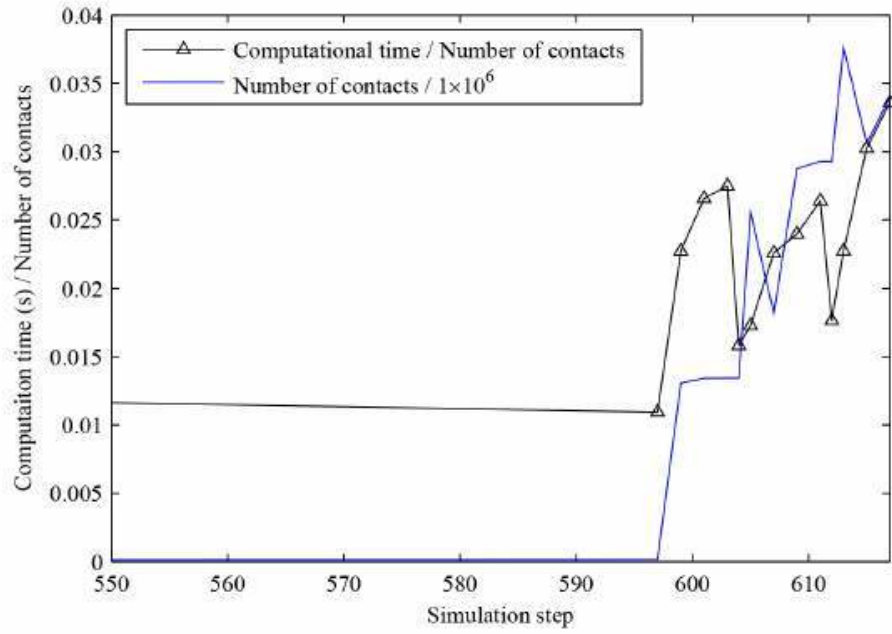


(a) DC1f ($v=10$ m/s)

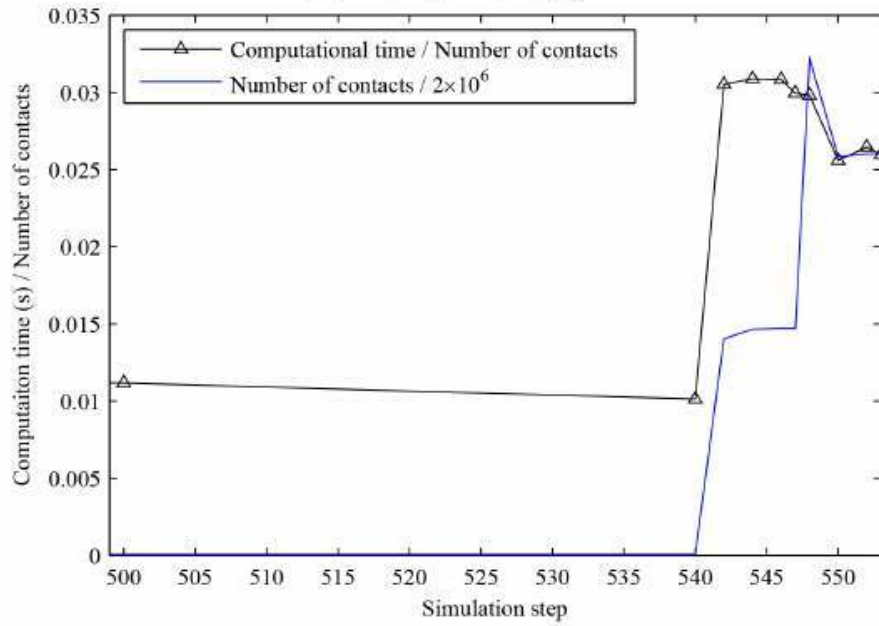


(b) DC2f ($v=100$ m/s)

Figure 5.23: The relation among total computation time, computational time of fragmentation, and number of contacts at each simulation step for two different crushing velocities. The number of contacts is scaled by $1/50$ in (a), and is scaled by $1/40$ in (b). The total computational time of each simulation step increases significantly with the increase in the number of contacts. More fragments are generated at a crushing velocity of $v=100$ m/s as compared to the amount when the crushing velocity is $v=10$ m/s. The computational time of fragmentation does not increase significantly with the number of contacts points, and has weaker impact on the total computational time.



(a) DC1f ($v=10$ m/s)



(b) DC2f ($v=100$ m/s)

Figure 5.24: The average computational time per contact for DC1f and DC2f. The number of contacts for $v=10$ m/s is scaled by $1/1 \times 10^6$ and is shown in (a). The number of contacts for $v=100$ m/s is scaled by $1/2 \times 10^6$ and is shown in (b). The average computational time per contact increases with the number of contacts.

6 Block caving case study on the influence of fragmentation on draw-point hang-up

6.1 Introduction

In recent years, there has been increased interest in underground mining methods such as block caving, sub-level caving and panel caving [Brown, 2003]. Underground mining methods are able to extract massive volume of rock from underground, and therefore they are perceived as being more environmental friendly. The present work uses the impulse-based method and the impact-velocity-based fragmentation pattern to investigate the influence of fragmentation on the gravity flow and hang-up phenomenon in a block caving system. The block caving method does not completely blast the ore body to small ore fragments, but blasts initially large intact ore bodies, and allows the ore bodies to progressively fragment under their own weight [Melo et al., 2009]. Because the block caving method uses gravity to fragment and transport a large amount of rock fragments, including ore and waste fragments, it provides a low cost, economical, and effective approach for underground mining. The dynamics of multiple ore fragments is simulated by the energy tracking method (ETM), and ore fragments are fragmented directly using the Weibull distribution based fragmentation pattern, which was described in Chapter 5. The influence of fragmentation is investigated by comparing the volumes of extracted ore fragments in strong materials and weak materials. The fragmentation happens within the ore fragments of a weak material, whereas no fragmentation occurs in the ore fragmentation of a strong material. The hang-up phenomenon, which is a major problem in block caving, requires extra explosions and operations to crush the oversized ore fragments. Moreover, the cave front propagates upwards due to gravity, which leads to the problem of surface subsidence [Vyazmensky, 2008, Sainsbury, 2012]. The present work investigates the influence of shape and size distribution on gravity flow and hang-up phenomenon.

Gravity flow in block caving system Understanding the gravity flow of ore fragments and waste fragments contributes to understanding and controlling the ore recovery and dilution in design and production [Laubscher, 1994]. For example, the entry of waste into valuable ore fragments significantly reduces the efficiency of block caving systems, and should be avoided. The hang-up phenomenon is a frequently occurring problem around draw points. Some oversized ore fragments may block the draw points, often requiring further fragmented by blasting and rock breakers. Hydraulic breaking, and placing a concussion bag below or between oversize

rocks, are also widely used to deal with the blockage around draw points [Ngidi and Pretorius, 2011]. The present work uses the energy tracking method (ETM) to handle the collisions of ore fragments and the gravity flow in block caving system is discussed. The hang-up phenomenon around a single draw point is also discussed with and without considering fragmentation process during simulation.

Gravity flow has been studied by many researchers [Kvapil, 1992, Kvapil et al., 1989]. They identify the gravity flow of material as occurring in two zones: the extraction zone and movement zone. The extraction zone is the area from which material is removed, and the movement zone is formed by the material that is flowing. Kvapil [Kvapil, 1992, Kvapil et al., 1989] and other researchers [Chen, 1997, Power, 2004] suggested that the geometries of the extraction zone and the movement zone are ellipsoidal. Susaeta [2004] employed large three-dimensional models to study the geometry of isolated movement zone, which is represented as a cylindrical shape. The width of the isolated movement zone has a maximum value after the start of drawing. Laubscher [2000], Vyazmensky et al. [2007] used experimental models to propose a design chart that is able to predict the cave angle via the MRMR (Mining Rock Mass Rating), the density of the caved rocks, the height of the caved rocks, and the minimum and maximum span of a footprint. Castro et al. [2007] employed a large three-dimensional physical model to investigate the gravitational transport of cohesionless gravel, so as to understand the gravity flow of coarse ore fragments [Power, 2004]. The material is drawn from an isolated draw point and forms an isolated extraction zone. Castro discussed some critical parameters of gravity flow in experiments, which include height of draw, particle size, draw point dimension and the effect of model scale. The experimental data statistically demonstrates that the total mass drawn and the height of draw have significant impact on the shape of extract zones and movement zones. However, draw point width and particle size have only a slight impact on the geometry of draw zones. Castro also shown that stress arch zones, which collapse as material is drawn, have significant influence on the height of the isolated movement zones.

Secondary fragmentation in block caving system To analyse fragmentation and gravity flow in a block caving system, a conceptual model of a self-sustained propagating cave is proposed, using seismic investigations and underground instrumentation [Duplancic, 2003]. As shown in Fig. 6.1, the ore bed around a single draw point is divided into four regions: the elastic zone, the seismogenic zone, the yielded zone, and the mobilised zone.

- **Elastic zone** This zone is far from the draw points, where caving activities have only a slight influence. The rock mass in this zone can be treated approximately as a continuous material. Natural and pre-existing faults in this zone can be ignored.
- **Seismogenic zone** Due to the redistribution of stress and strain caused by caving activities, some pre-existing fractures, such as joints and faults, slip along fracture surfaces, and some new faults are initiated. Microseismic or seismic phenomenon can be detected by underground instrumentation in this zone.

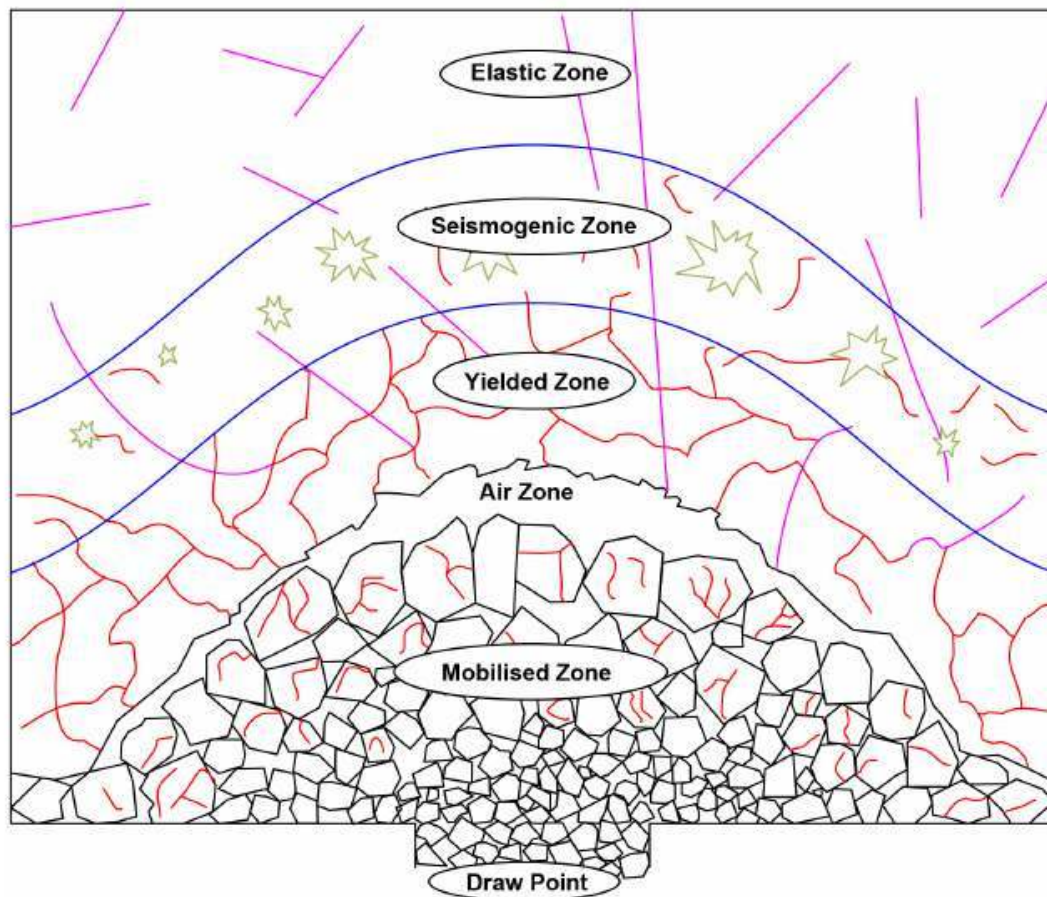


Figure 6.1: The conceptual model of a self-sustained propagating cave [Sainsbury, 2012]

- **Yielded zone** Comparing to the seismogenic zone, in this zone the displacement of the rock mass increases. Existing fractures evolve further, and more new fractures are initiated. The developing fracture surfaces split the rock mass, and consequently hundreds and thousands of ore fragments are generated and prepare to move down.
- **Mobilised zone** A large number of rock fragments are generated in the yielded zone, move downwards into this mobilised zone, and are collected through the draw points. The failure to control the movement of the ore fragments might cause blockage around the draw points.

During a block caving operation, the fragmentation phenomenon involves primary and secondary phases. The primary fragmentation occurs in the yielded zone due to the propagation of fracture surfaces. The secondary fragmentation occurs within the air gap and the mobilised region, where ore fragments are further fragmented during transportation. The secondary fragmentation process further reduces the size of rock fragments resulting from the primary fragmentation. Secondary fragmentation is strongly influenced by the size distribution of rock fragments, and

the heights of the air gap and the mobilised zone [Laubscher, 1994]. Understanding the influence of secondary fragmentation on the size distribution of ore fragments is critical to improving the production effectiveness of block cave mining [Pierce, 2010].

Hang-up phenomenon in ore pass system An ore pass system is used in block cave mining to transport rock fragments [Lessard and Hadjigeorgiou, 2006]. The low cost ore pass system is designed for gravitational transportation of ore fragments along long vertical distances. The performance of ore pass systems strongly influence the economic performance of mining production [Hagan and Acheampong, 1999]. The failure of gravitational transportation in an ore pass system can disrupt ore production [Stacey and Swart, 1997, Ferguson, 1991]. According to the statistical data from U.S. Mine Safety and Health Administration (MSHA), ore pass related hazards widely exist, and cause thousands of injuries. Approximately 75% of injuries happen when hand tools are employed to pull or free the ore pass chutes in ore passes [Beus et al., 1997].

Two kinds of hang-up phenomena are observed in block cave mining: interlocking arches and cohesive arches. The interlocking of large size boulders is referred to as interlocking arches, and the cementation of fine and sticky particles is referred to as cohesive arches [Nazeri et al., 2002]. A survey investigated the performance of ore pass system in ten Quebec underground mines and demonstrated that interlocking hang-up phenomenon of coarse material are much more common, compared to the cohesive arching phenomenon [Lessard and Hadjigeorgiou, 2007]. The interlocking hang-up problems were observed in eight out of the ten surveyed mines. Some researchers have employed risk assessment methodologies to investigate the main controlling factors in ore pass failure [Orace et al., 2011].

Physical modelling and particle flow analysis are also employed to evaluate ore-pass design [Beus et al., 1997]. The performance of the ore pass system is influenced by the size distribution of rock fragments, and the geometry of the ore pass system. The shape of the ore pass system can be straight or curved. The geometry of screening devices can include grizzlies, scalpers and mantles. The ratio between the ore pass dimension and the largest rock block size, e , is used to ensure the free flow of ore fragments. This parameter is calculated as $e = \frac{D}{d}$, where D is the ore pass dimension and (d) is the dimension of largest rock fragment. Jenike [Jenike, 1961] suggests that $e > 2$ be used in order to ensure the free flow of rock material. Lessard and Hadjigeorgiou [2007] used 1065 numerical runs to identify the relationship between maximum rock block size and the dimension of the ore pass system. These numerical results then helped to avoid the failure of ore pass system. Lessard and Hadjigeorgiou [2007] discussed the difference of vertical and inclined ore pass systems. The difference between square and circular pass systems, and the difference between spherical and cuboid particles were also discussed. Numerical results showed that the results of cuboid shaped rocks are closer to the realistic results, as compared to the results of simulations of spherical objects. The friction coefficient is another critical parameter influencing the hang-up phenomenon.

6.2 Case study: Hang-up of initially spherical rock fragments

In this section, the influence of fragmentation on interlocking hang-up of spherical particles in the vicinity of a single draw point is investigated. As shown in Fig. 6.2, the geometry of the draw point is characterised as $a=12$ m, $b=1.15$ m, and $c=10$ m. The depth of the draw point is $d=10$ m. The ore fragment is discretised using a series of tetrahedra. Above the draw points, a cuboid boundary wall is created to constrain the movement of ore fragments. The width of the cuboid boundary wall is $30\text{ m} \times 30\text{ m}$.

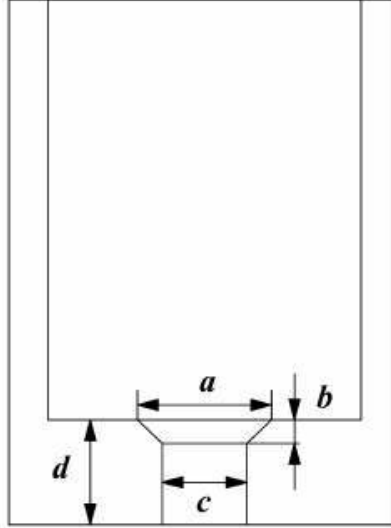


Figure 6.2: The geometry of a draw point for the hang-up modelling of spherical ore fragments.

The percent of volume of extracted ore fragments before the occurrence of hang-up is strongly related to the efficiency of a block caving system. The percentage of volume of extracted ore fragments, e , is calculated by,

$$e = \frac{V_{extract}}{V_{total}} \quad (6.1)$$

where $V_{extract}$ is the volume of extracted ore fragments whose geometric centre is below the bottom of the draw point (-10 m), and V_{total} is the total volume of all ore fragments. Four tests, with varying diameters of ore fragments, with and without considering fragmentation, are modelled. As shown in Table 6.1, DPST1 and DPST1f simulate the gravity flow of small spheres whose diameters are 5.775 m. DPST1 and DPST1f simulate the gravity flow of small spheres whose diameters are 7.5 m. In DPST1 and DPST2, the material of ore fragments is strong, and so no fragmentation is expected. In DPST1f and DPST2f, the material comprising the ore fragments is weak, and so fragmentation happens during flow. The density of the ore fragments is 2400 kg/m^3 . The total volume of ore fragments in the four tests is similar. The restitution coefficient is 0.3 , and the friction coefficient is 0.75 , both of which values are appropriate for a variety of mining rocks [Cleary and Sawley, 2002]. Simulation results of DPST1 are shown in

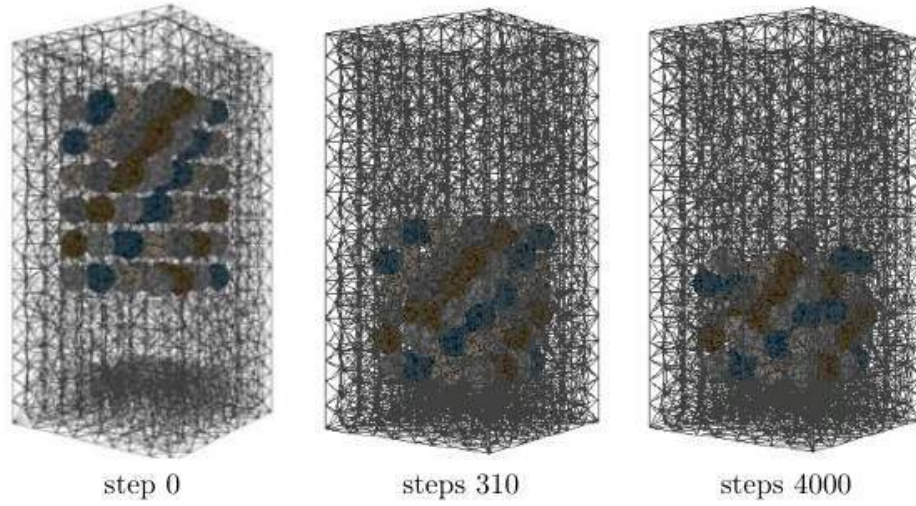


Figure 6.3: DPST1 at step 0, 310, and 4000. The image includes the mesh representation of the container and the draw point.

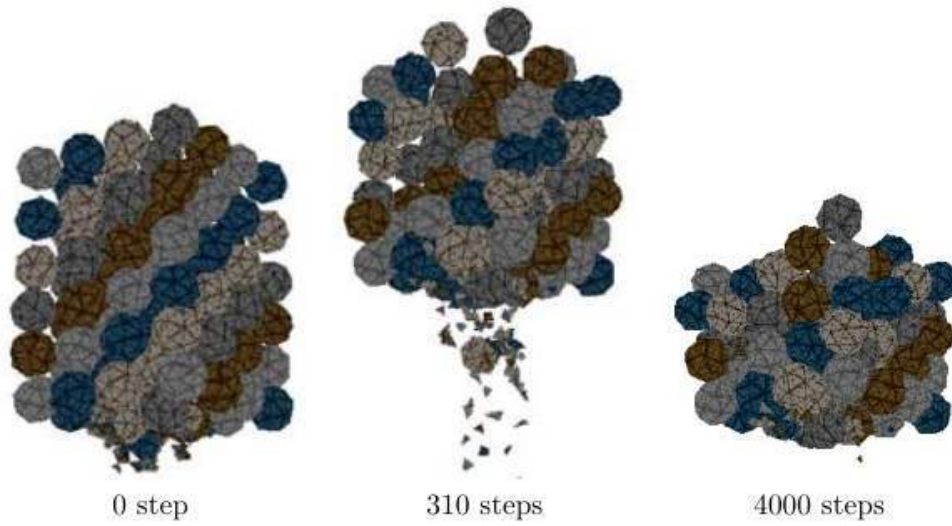


Figure 6.4: DPST1f at step 0, 310, and 4000.

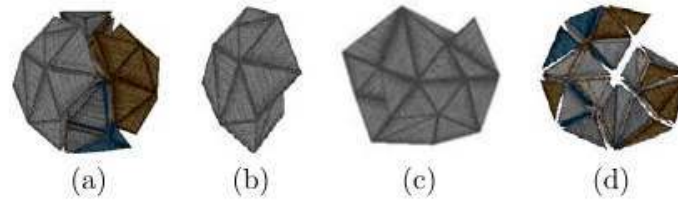


Figure 6.5: Fragmentation simulation of one ore fragment in DPST1. After collision, the initial rock is fragmented as a function of speed and contact point location, and its effect on fracture growth.

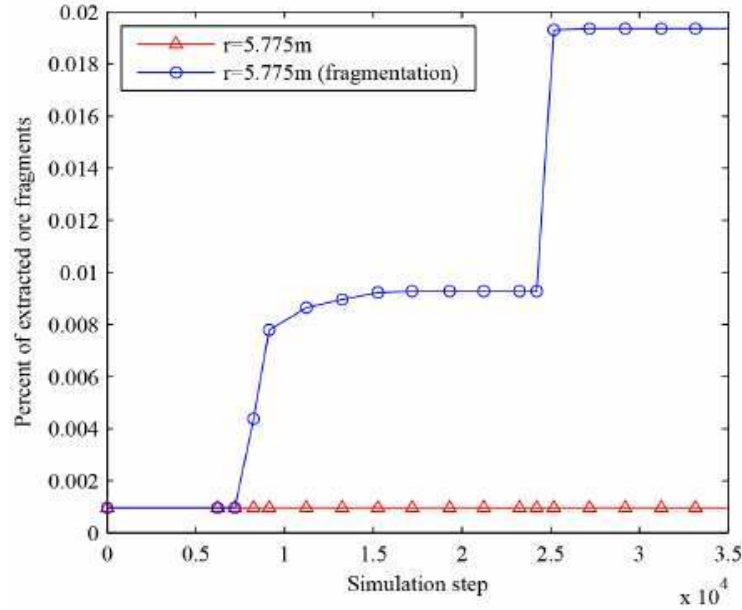


Figure 6.6: Percentage of volume of extracted ore as a function of time, as predicted by simulations with and without fragmentation ($r = 5.77 \text{ m}$). Considering the influence of fragmentation is important to predict the efficiency of extracting. In this test, if there is no fragmentation, there is no extraction.

Fig. 6.3, where the boundary wall is displayed via a mesh, and ore fragments are displayed via coloured solid tetrahedra. DPST1f is shown in Fig. 6.4; in this case, the display of the model boundaries is turned off. Sample fragmented shapes are shown in Fig. 6.5, where different colours correspond to separate final fragments. Fragmentation occurs during the impact of a fragment with other ore fragments, and with the lower boundary wall. The rock has been subdivided into 43 smaller fragments, as shown in Fig. 6.5(d). Minimum fragment size is capped by the original refinement of the mesh, and can be adjusted depending on available computational time and memory for the simulation. This cap is a highly desirable feature of the methodology, as high-speed impact may yield fragmentation patterns in which thousands, or indeed up to millions, of new fragments are created. The comparative evolution of the percentage of volume of extracted ore fragments during simulation is shown in Fig. 6.6.

Images of DPST2 and DPST2f are presented in Fig. 6.7 and 6.8, respectively. Sample fragment shapes are shown in Fig. 6.9. In this test, the most strongly fragmented piece of ore broke up into 34 smaller fragments. The comparative progress of extraction of ore volume during the simulation is plotted in Fig. 6.10.

According to Fig. 6.7, without considering the influence of fragmentation, some ore fragments initially fall through the draw point. However, in DPST1, which uses a smaller diameter for the ore fragments, no ore fragment is extracted (see Fig. 6.3). This test demonstrates that the appearance of hang-up phenomenon is not only related to the size of the ore fragments, but also related to the initial position of ore fragments, and the relative position of the draw

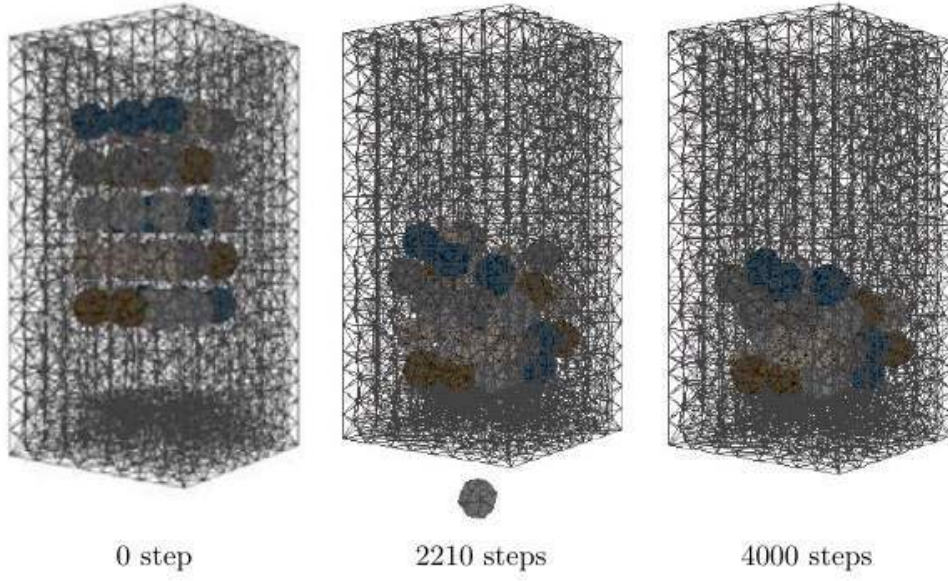


Figure 6.7: DPST2 at step 0, 2210 and 4000.

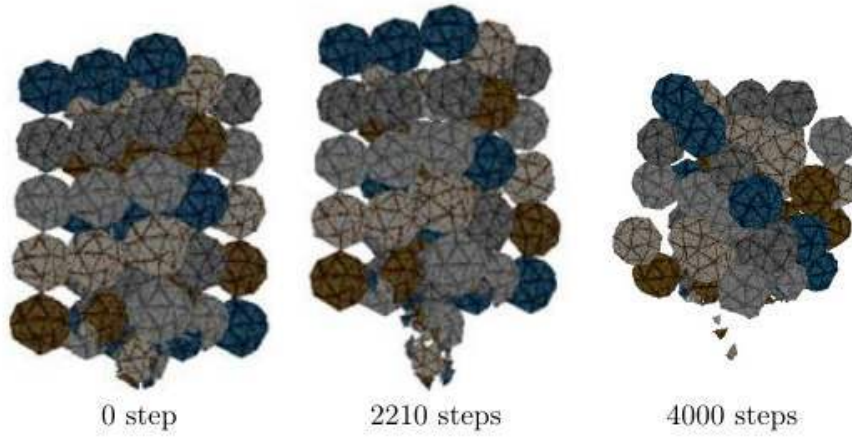


Figure 6.8: DPST2 at step 0, 2210 and 4000.

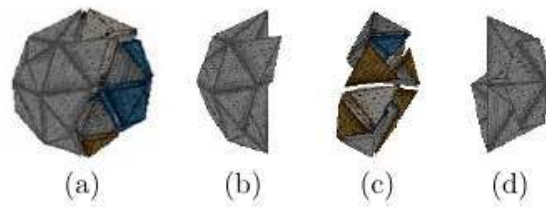


Figure 6.9: DPST2f at step 0, 2210 and 4000.

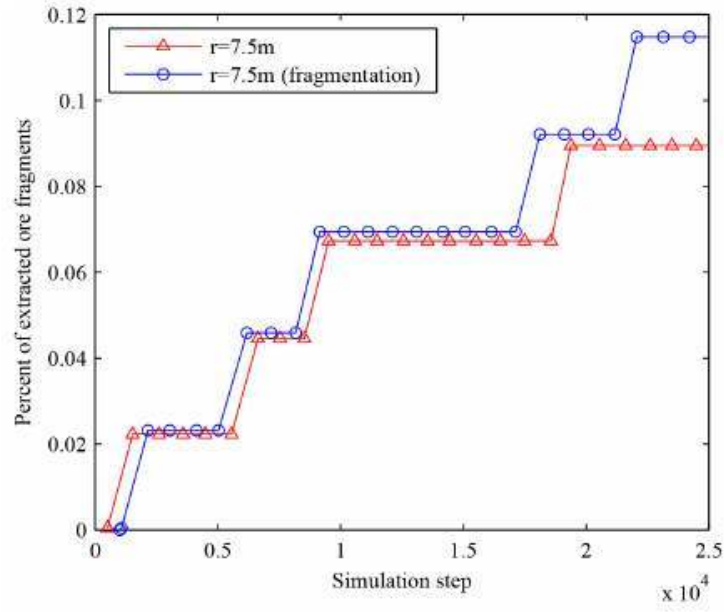


Figure 6.10: Percentage of volume of extracted ore as a function of time as predicted by simulations with and without fragmentation ($r = 0.75m$).

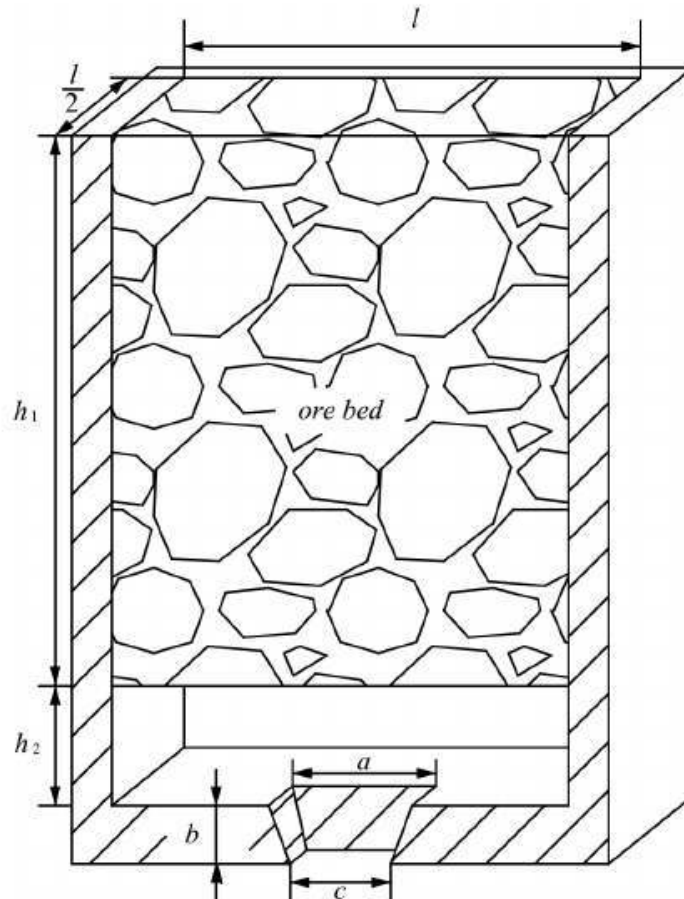


Figure 6.11: Geometry specification of a single draw point. $l=20$ m, $h_1=20$ m and $h_2=10$ m. The dimensions of the draw point are: $a=7.16$ m, $b=3$ m, and $c=3$ m.

Table 6.1: Parameters for the hang-up modelling of spherical ore fragments.

	Diameter r (m)	Total volume of fragments (m^3)	Initial number of fragments	Fragmentation happens
DPST1	5.775	9246.44	96	No
DPST1f	5.775	9246.44	96	Yes
DPST2	7.5	9493.88	45	No
DPST2f	7.5	9493.88	45	Yes

point. For DPST2, it is observed that 8.89% ore volume of ore fragments falls though the draw point before hang-up. In contrast, in DPST2f, 11.46% ore volume of ore fragments fall though the draw point before hang-up. In total, 2.57% more ore volume of rock is extracted before hang-up, implying that the extraction process, albeit with larger initial fragment sizes, in this case was more effective. In the following section, the gravity flow of cuboid ore fragments is investigated. The shape of the ore fragments has significant impact on the gravity flow of system. Wang et al. [2011] used the discrete element method to simulate the gravity flow of spheres, octahedra, tetrahedra and a mixture thereof, and showed that sphere flowability is greater than that of tetrahedra. Meanwhile, the tetrahedra flowability is significantly greater than that of the octahedra.

6.3 Case study: Hang-up of initially cuboid fragments

As ore fragment geometry has a significant impact on the flowability of ore fragments [Wang et al., 2011], the present section simulates the jointed ore bed by a set of cuboid ore fragments, instead of spherical particles. Because it is relatively convenient to fill the ore bed in experiments using cuboid fragments, and the interlocking among tightly placed cuboid fragments can simulate the interlocking among nature ore fragments, McNearny and Abel [1993] used cuboid half-bricks to investigate the gravity flow of ore fragments around draw points in large-scale two-dimensional block caving model tests. The geometry of the draw point and the ore fragments are shown in Fig. 6.11. The density of the ore bed is 10 kg/m^3 . This ore bed is a cuboid area. The bottom of this ore bed is a square, whose width is $l=20 \text{ m}$. The height of ore bed is $h_1=20 \text{ m}$. Therefore, the total volume of the ore bed is 27000m^3 . A trapezoidal draw point is used to collect the ore fragments. The top width of this trapezoidal draw point is $a=7.16 \text{ m}$, the bottom width of this draw point is $c=3 \text{ m}$, and the depth of this draw point is $b=3 \text{ m}$. There is a void space between the ore bed and the bottom of draw point, the height of which is $h_2=10 \text{ m}$. The ore fragments are constrained in a rigid and box-like boundary wall. Therefore, the ore fragments can be collected through the draw point. The restitution coefficients of the boundary wall and the ore fragments are both 0.6. The friction coefficients of boundary wall and

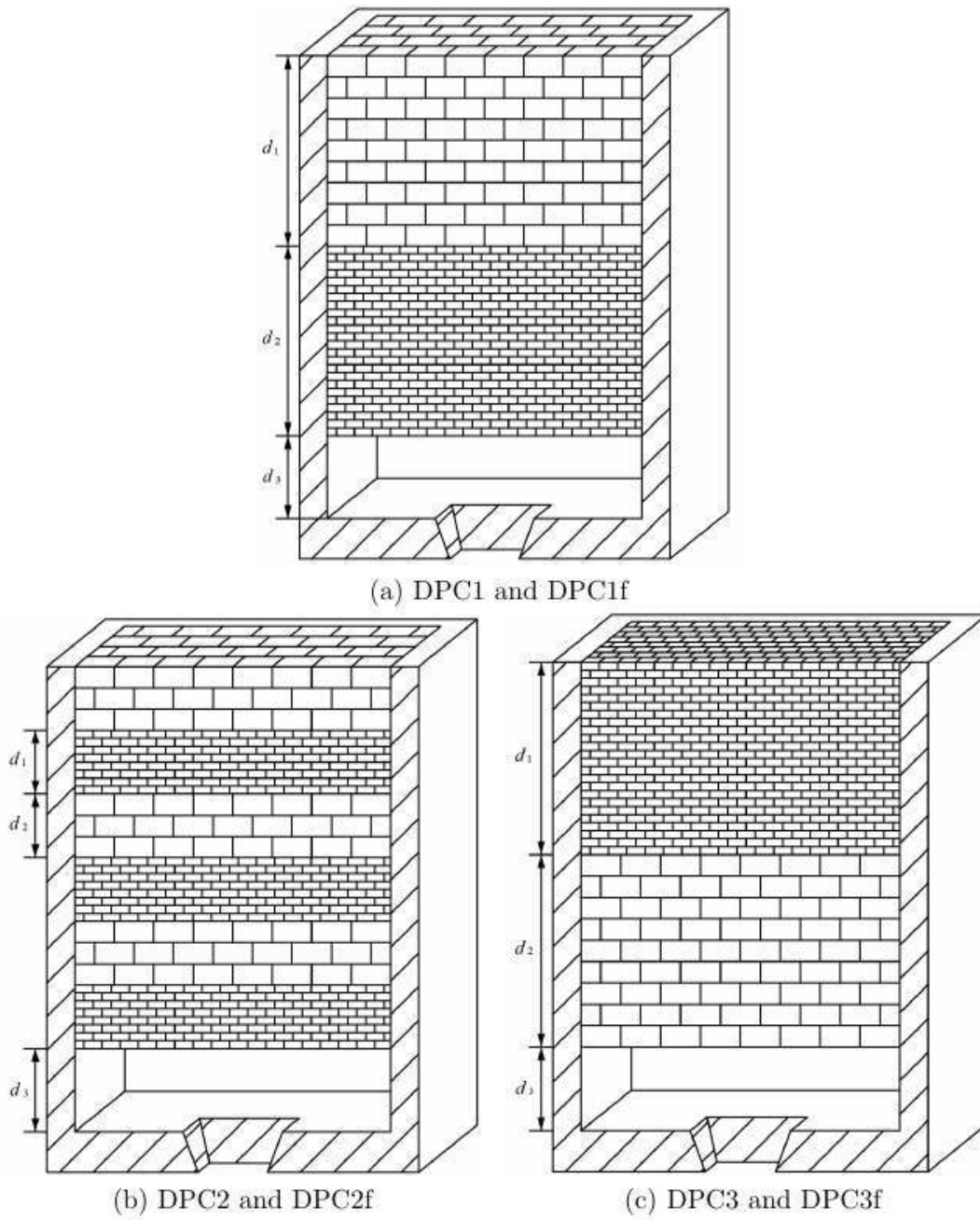


Figure 6.12: Different size distribution of cuboid rock fragments.

ore fragments are both 0.77. These restitution coefficients and friction coefficients allow energy loss during collision, and enhance the interlocking among ore fragments. The jointed ore bed is simulated by cubes which are placed side-by-side and end-to-end, as shown in Fig. 6.12. These ore fragments are packed tightly and interlocked, to simulate weathered and broken rocks. To simulate different size distributions of ore fragments, two different sizes of bricks are used: small fragments and large fragments. Both the small and large fragments are shorter in one dimension than in the other two. The size of the small fragments is $1.5\text{ m} \times 1.5\text{ m} \times 1\text{ m}$, and the size of the large fragments is $4\text{ m} \times 4\text{ m} \times 2.5\text{ m}$. Each fragment is discretised using six tetrahedral elements. Different size distributions are simulated by placing the small fragments and large fragments in different layers, as shown in Fig. 6.12. Three different cases are considered:

- DPC1: As shown in Fig. 6.12(a), the upper half of ore bed is fragmented by a set of large cuboid ore fragments, and the lower half of ore bed is fragmented by a set of small cuboid ore fragments. There are totally 3756 cuboid ore fragments initially. The material of ore fragments is assumed to be strong. Therefore, no fragmentation occurs during the simulation.
- DPC1f: The distribution of small and large ore fragments is the same as the distribution in DPC1. The difference is that material of the ore fragments is weaker, therefore fragmentation is more likely to occur during the simulation.
- DPC2: As shown in Fig. 6.12(b), the layers of small fragments and large fragments are mixed. There are totally 3400 cuboid ore fragments initially. The material comprising the ore fragments is assumed to be strong. Therefore, no fragmentation occurs during the simulation.
- DPC2f: The arrangement of small and large ore fragments is the same as the arrangement in DPC2. The difference is that material comprising the ore fragments is weaker, and so no fragmentation occurs during the simulation.
- DPC3: As shown in Fig. 6.12(c), the upper half of the ore bed is fragmented by a set of small cuboid ore fragments, and the lower half of the ore bed is fragmented by a set of large cuboid ore fragments. There are totally 3400 cuboid ore fragments initially. The material of ore fragments is assumed to be strong. Therefore, no fragmentation occurs during the simulation.
- DPC3f: The arrangement of small and large ore fragments is the same as in the arrangement in DPC3. The difference is that material of the ore fragments is weaker, and therefore fragmentation is more likely to occur during the simulation.

6.3.1 DPC1 and DPC1f

As shown in Fig. 6.1, the size of the ore fragments decreases with the decrease of the distance of the ore fragments to the draw point. In DPC1 and DPC1f, small ore fragments are set up to

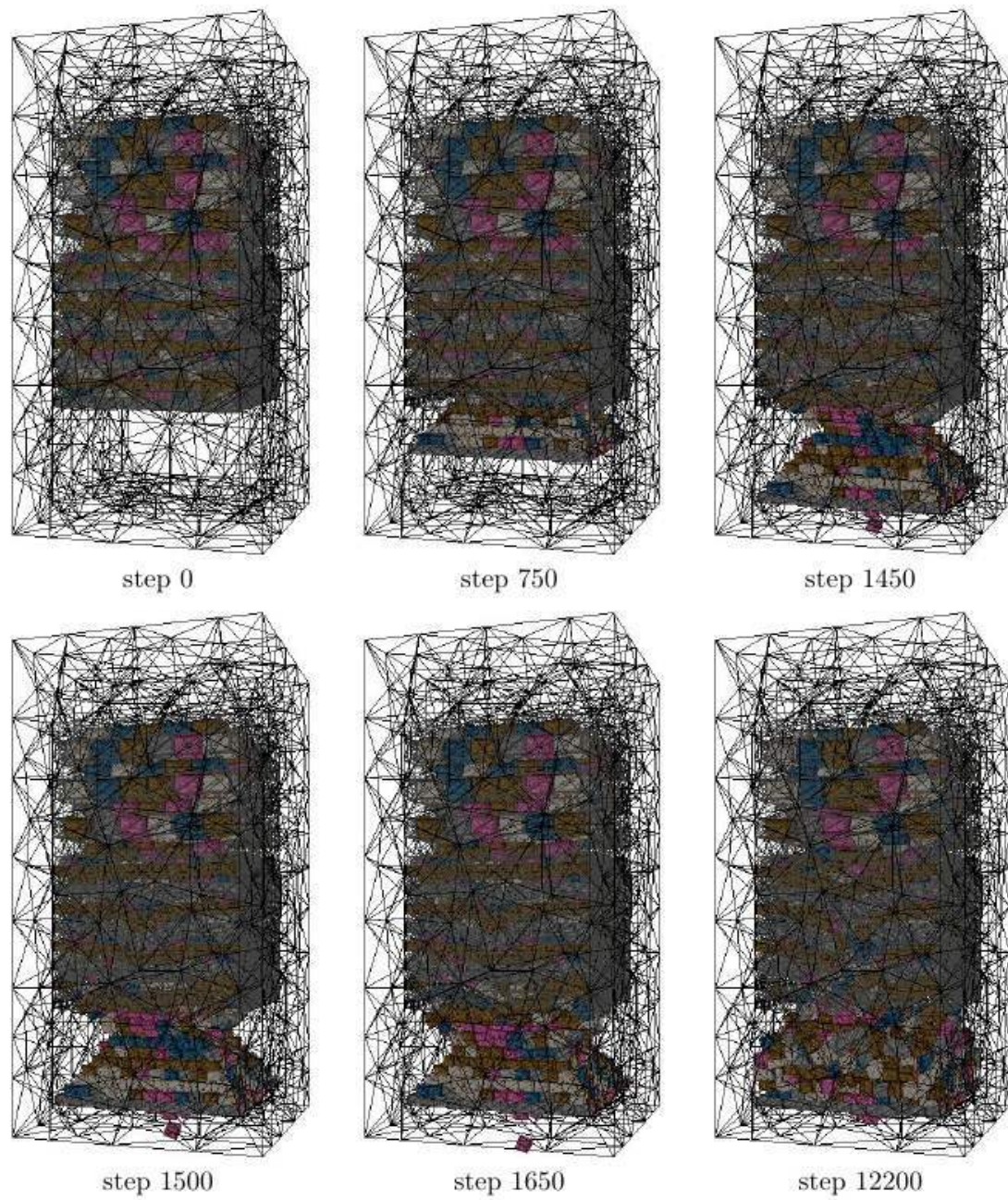


Figure 6.13: Front view of DPC1, where no fragmentation occurs. Ore fragments are coloured arbitrarily to distinguish with each other. Few small ore fragments pass through the draw point.

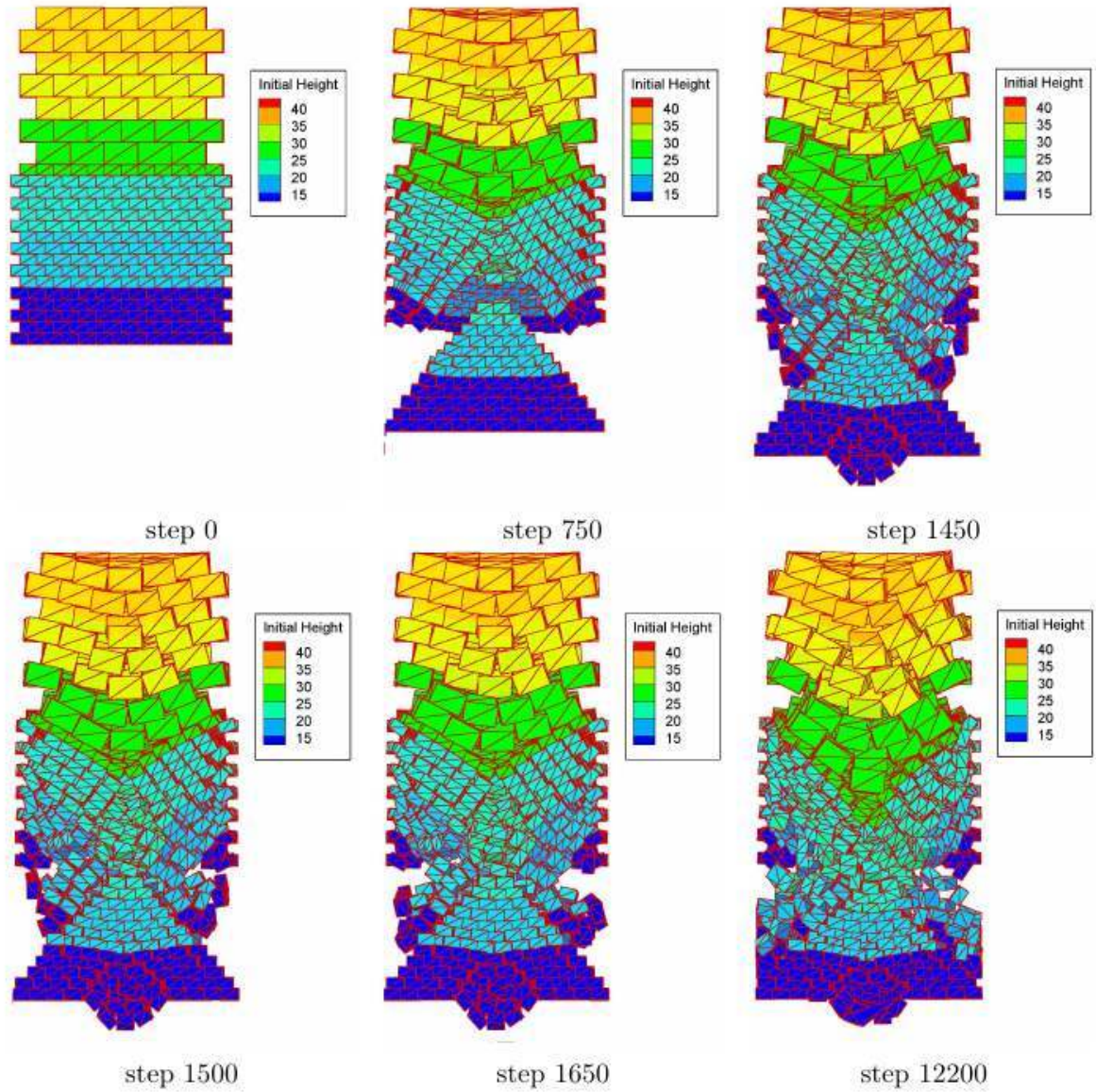


Figure 6.14: Sectional view of DPC1, where no fragmentation occurs. Ore fragments are coloured using their initial height. The subsidence of ore fragments increases with the decrease of their distances to the central axis of draw point.

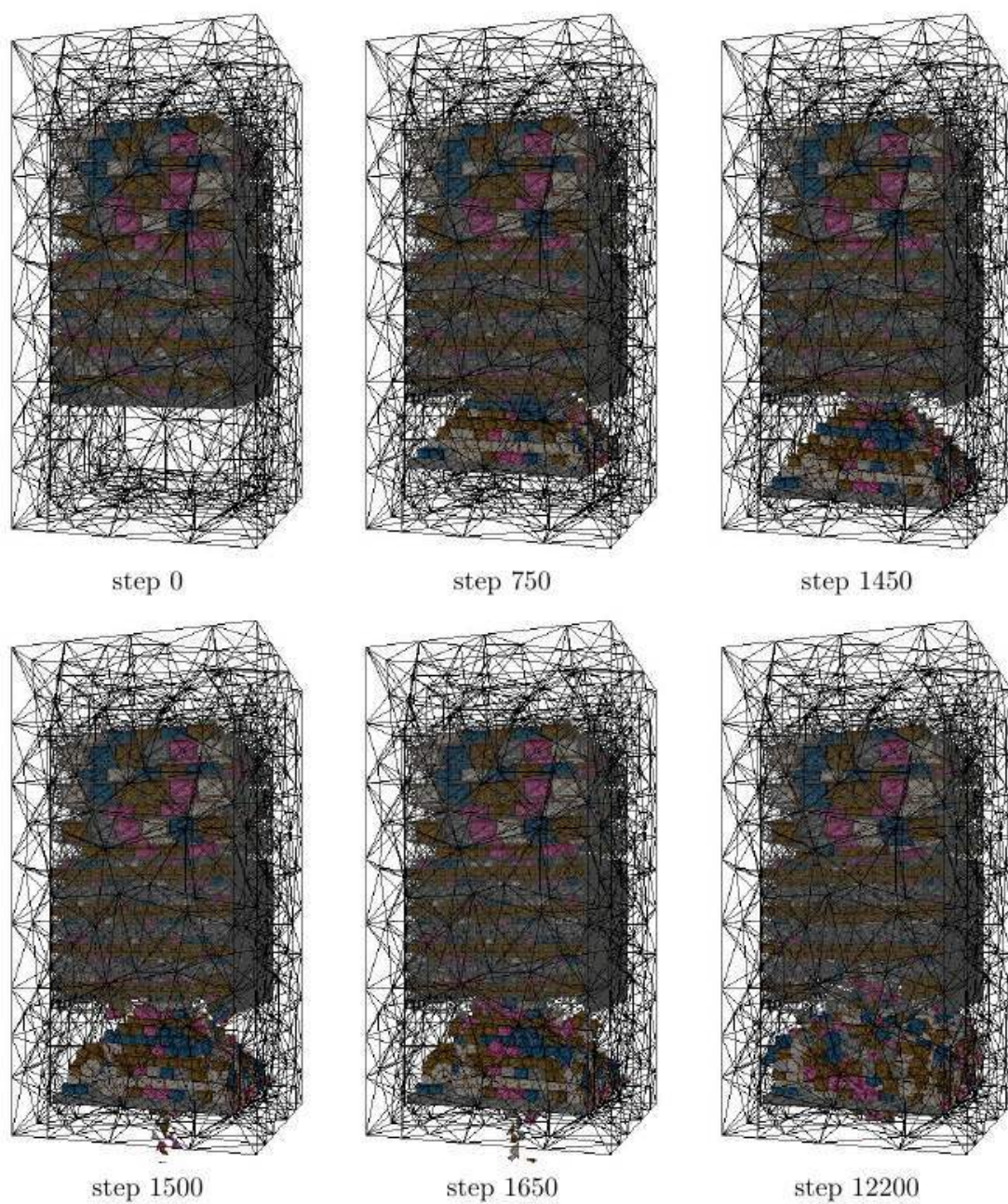


Figure 6.15: Front view of DPC1f, where fragmentation happens. Ore fragments are coloured arbitrarily to distinguish them from each other.

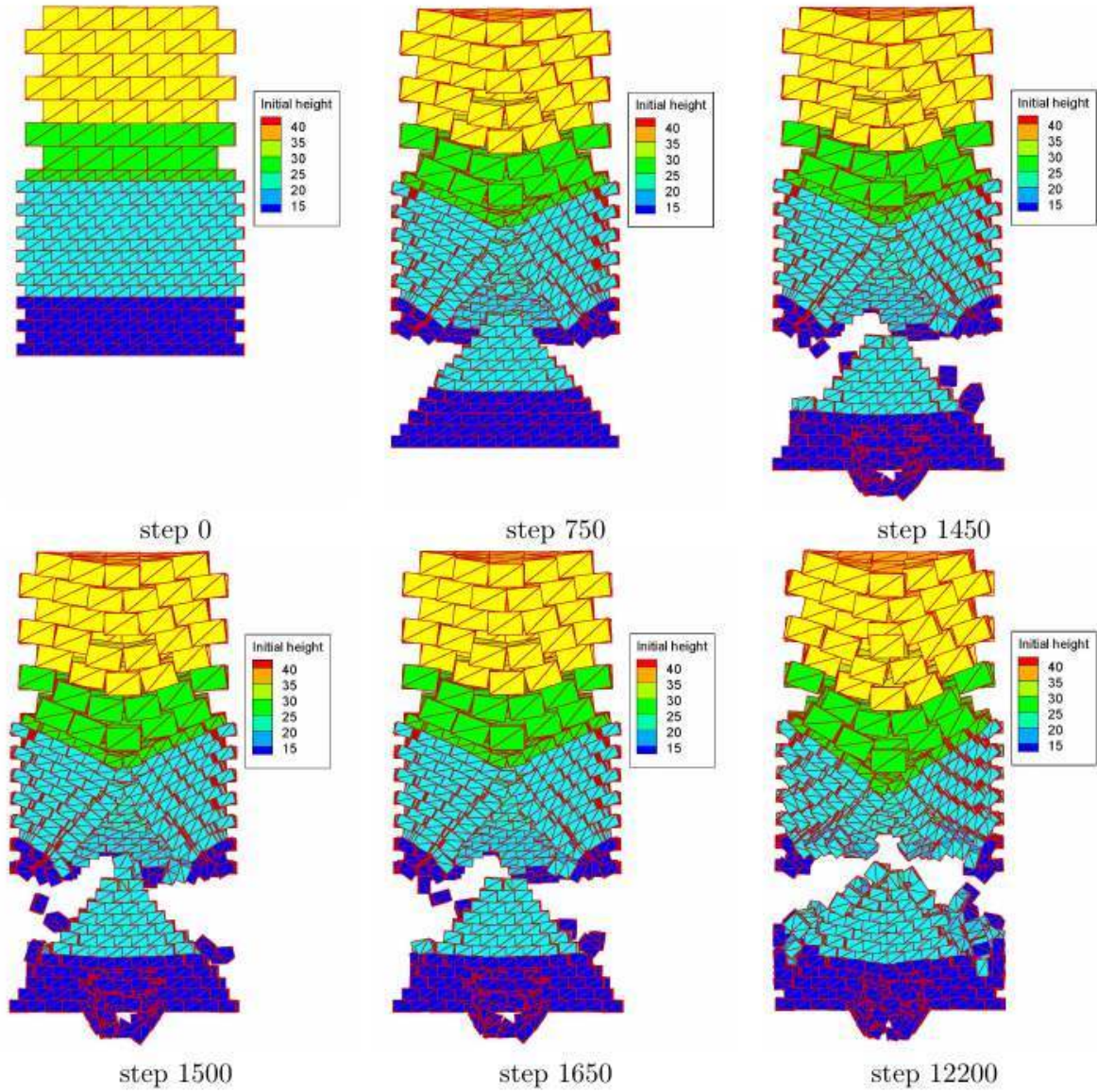


Figure 6.16: Sectional view of DPC1f, where fragmentation happens. Ore fragments are coloured using their initial height. The subsidence of ore fragments increases with the decrease of their distances to the central axis of the draw point.

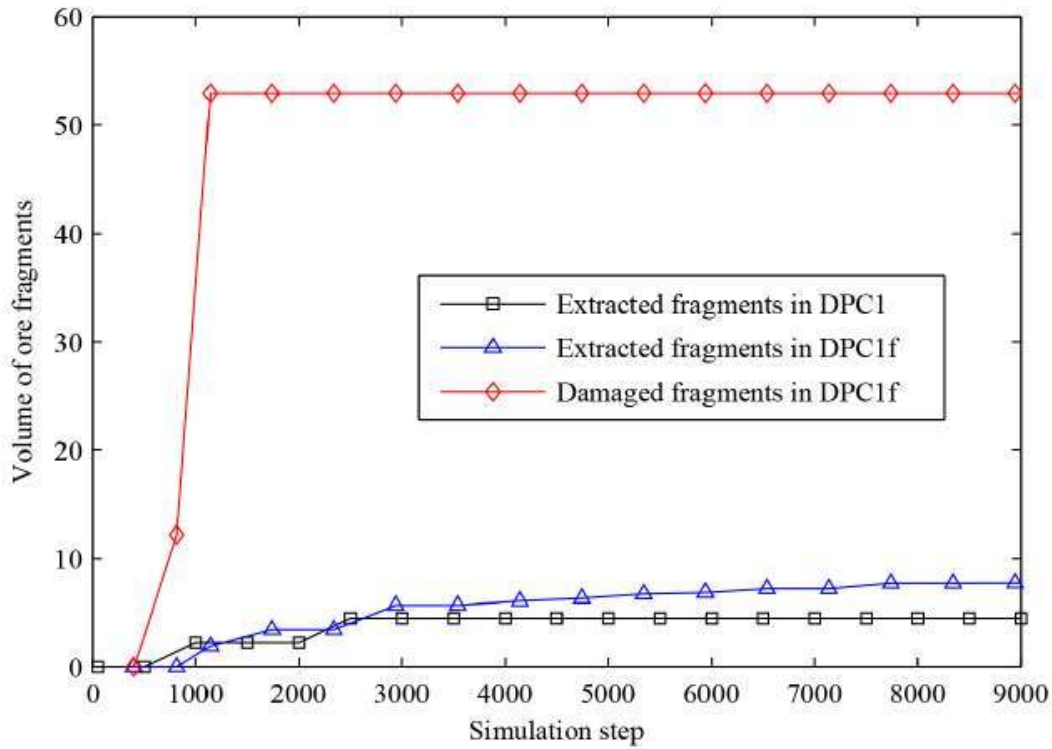


Figure 6.17: The volume of extracted ore fragments in DPC1, the volume of extracted ore fragments and damaged ore fragments in DPC1f. In DPC1, no fragmentation, ore fragments hang-up in the vicinity of draw point at step 2400 (approximately). Approximately 5 m³ of ore fragments are collected below the draw point. In DPC1f, fragmentation happens, and hang-up is delayed to approximately step 7800. Approximately 9 m³ ore fragments are collected; the fragmentation process has increased the volume of extracted ore fragments. Fragmentation mostly happens around step 1000. The volume of damaged ore fragments reaches to 54 m³.

be nearer than the large fragments, as shown in Fig. 6.12(a). The depth of small fragments is $d_1=16$ m, and the depth of large fragments is $d_2=14$ m. To simulate the interlocking between intact rocks and broken rocks, the ore fragments next to the boundary wall are set to be static during simulation. The rest of ore fragments are initially static, but start to move downward under gravity. The front view for DPC1, where no fragmentation occurs, is illustrated in Fig. 6.13. In this figure, the boundary wall is displayed via the mesh, and the ore fragments are coloured arbitrarily to distinguish each other. Few ore fragments fall through the draw point. The sectional view of DPC1 is shown in Fig. 6.14. To display the movement of ore fragments clearly, these ore fragments are coloured using their initial height. It is found that the displacements of the ore fragments around the central axis are larger than the displacements of the ore fragments that are far away from the central axis. A cluster of ore fragments, which is triangular, starts to move down before the others, because of the interlock among ore fragments and interlock between the ore fragments and the boundary wall. Fig. 6.15 illustrates the front

view for DPC1f, which involves the fragmentation process. Few ore fragments fall through the draw point. The sectional view of DPC1f is shown in Fig. 6.16. To display the movement of ore fragments clearly, these ore fragments are coloured using their initial height. It also can be found that the displacements of the ore fragments around the central axis are larger than the displacements of the ore fragments that are far from the central axis. The volume of extracted ore fragments in DPC1, and also the volume of extracted ore fragments and crushed ore fragments in DPC1f, are shown in Fig. 6.17. In DPC1, without fragmentation, ore fragments hang up in the vicinity of the draw point at step 2400 (approximately). Approximately 5 m^3 of ore fragments are collected below the draw point. In DPC1f, fragmentation happens, and hang-up is delayed to approximately step 7800. Approximately 9 m^3 of ore fragments are collected; the fragmentation process has increased the volume of extracted ore fragments. Fragmentation mostly happens around step 1000, and the volume of damaged ore fragments reaches to 54 m^3 . Comparing Fig. 6.14 to Fig. 6.16, it is seen that the influence of fragmentation is slight on the displacement of most of the ore fragments, because the fragmentation only occurs around the draw point.

There are totally 3756 cuboid ore fragments initially. In DPC1f, at step 12200, the number of ore fragments reaches to 4017. DPC1 and DPC1f are both run on a personal computer which uses Intel Xeon E5690 CPU with 3.45GHz (two processors), 12GB installed memory, and 12 cores. DPC1 and DPC1f spent approximately 19 days and 30 days to run 12200 simulation steps respectively.

6.3.2 DPC2 and DPC2f

In many ore beds, the size distribution varies in different horizontal layers. As an example, coal is a sedimentary rock which is created from decaying plants. Over millions of years, the plants are pushed down by layers of overburden rock, and the size distributions vary in the layers of coal. In DPC2 and DPC2f, some layers of the ore bed are fragmented into small fragments, and some layers of ore bed are fragmented into larger fragments. The layers of small fragments and the layers of large fragments are mixed, as shown in Fig. 6.12(b). The depth of large fragment layers and the depth of small fragment layers are the same, $d_1=4 \text{ m}$ and $d_2=4 \text{ m}$. To simulate the interlocking between intact rocks and broken rocks, the ore fragments next to the boundary wall are set to be static during simulation. The ore fragments are initially static, and start to move downward under gravity.

The front view of DPC2 is illustrated in Fig. 6.18, where the boundary wall is displayed as a mesh. It is seen that the large fragments block the draw point immediately after the start of the simulation. Therefore, no ore fragments pass through the draw point. The sectional view of DPC2 is shown in Fig. 6.19. Because of interlocking, most of the ore fragments remains nearly frozen in their initial place. For large fragments, only the lowest layers fall down. For the simulation in which fragmentation happens, the front view for DPC2f is illustrated in Fig. 6.20. Some ore fragments fragment around the draw point, and several ore fragments pass through

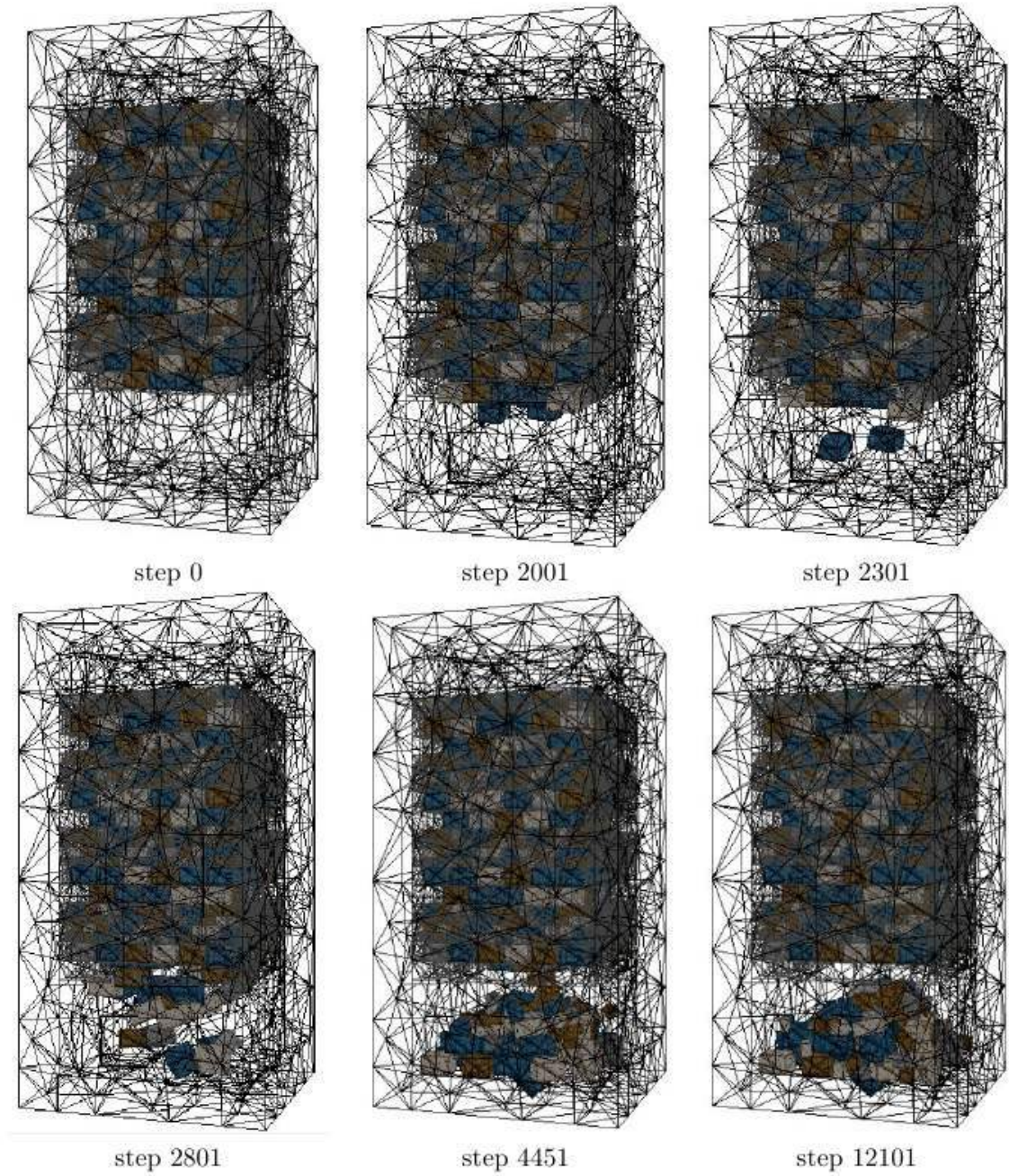


Figure 6.18: Front view of DPC2, where no fragmentation happens. Ore fragments are coloured arbitrarily to distinguish them from each other. No ore fragments pass through the draw point.

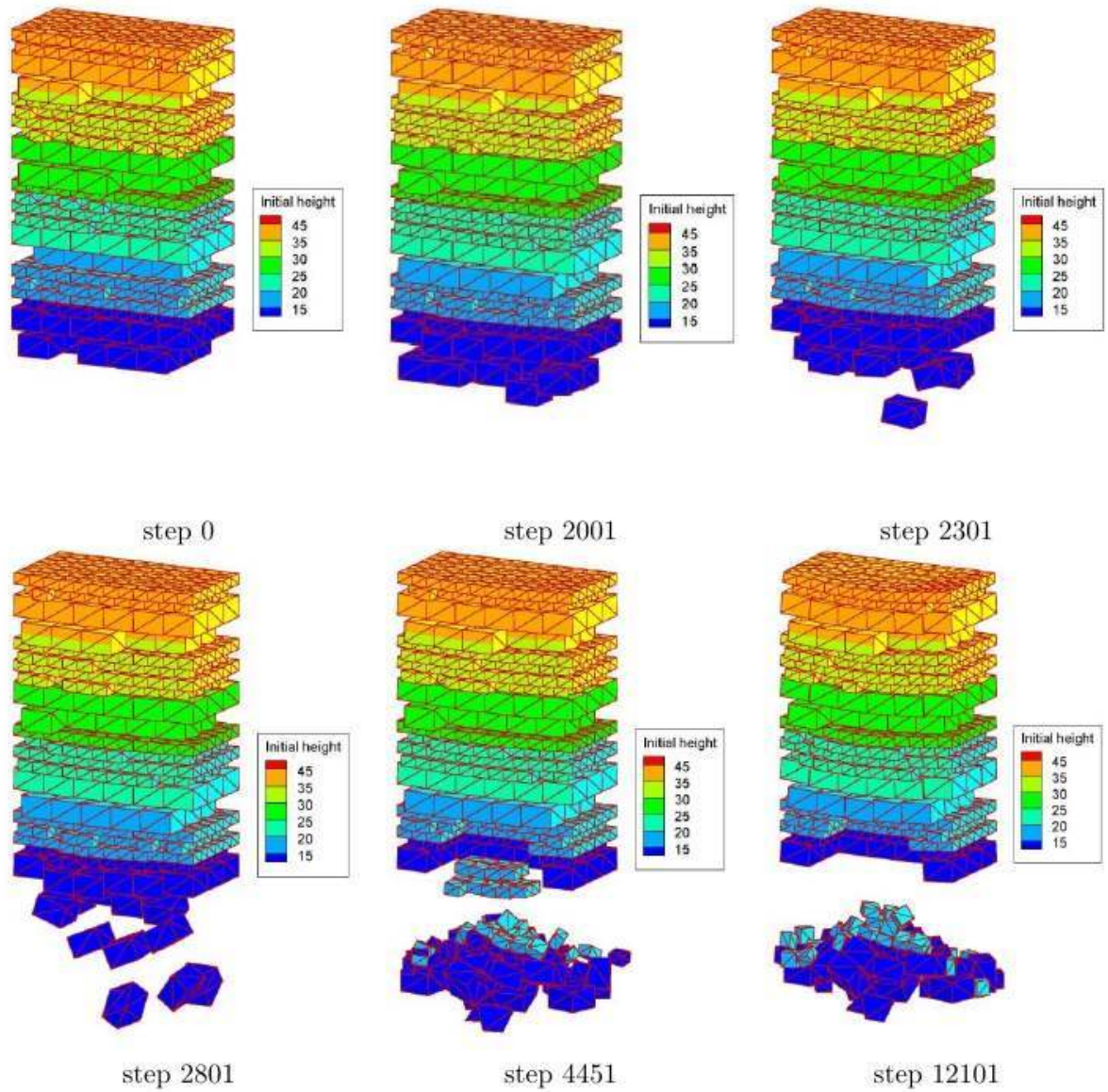


Figure 6.19: Sectional view of DPC2, where no fragmentation happens. Ore fragments are coloured using their initial height.

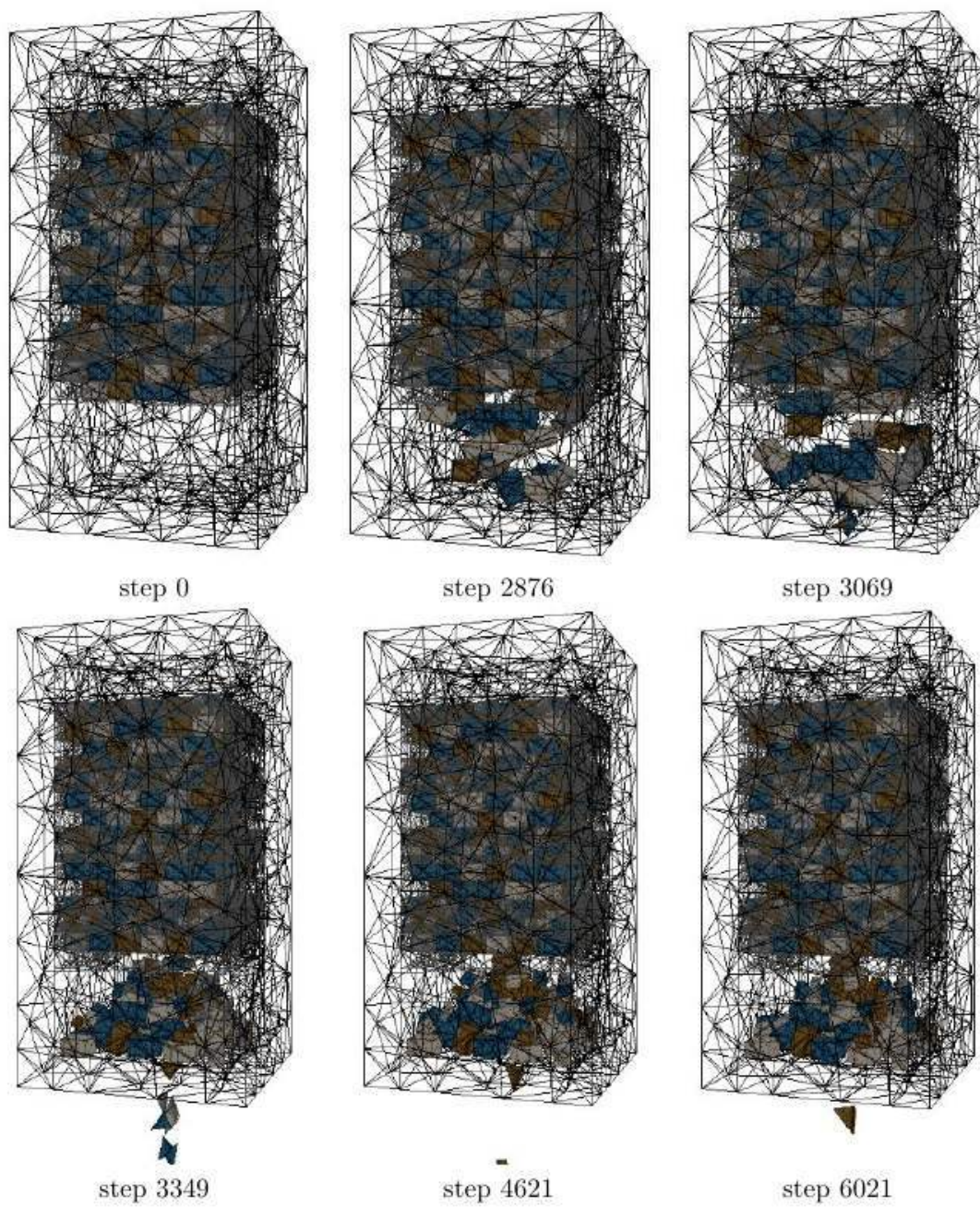


Figure 6.20: Front view of DPC2f, where fragmentation happens. Ore fragments are coloured arbitrarily to distinguish them from each other.

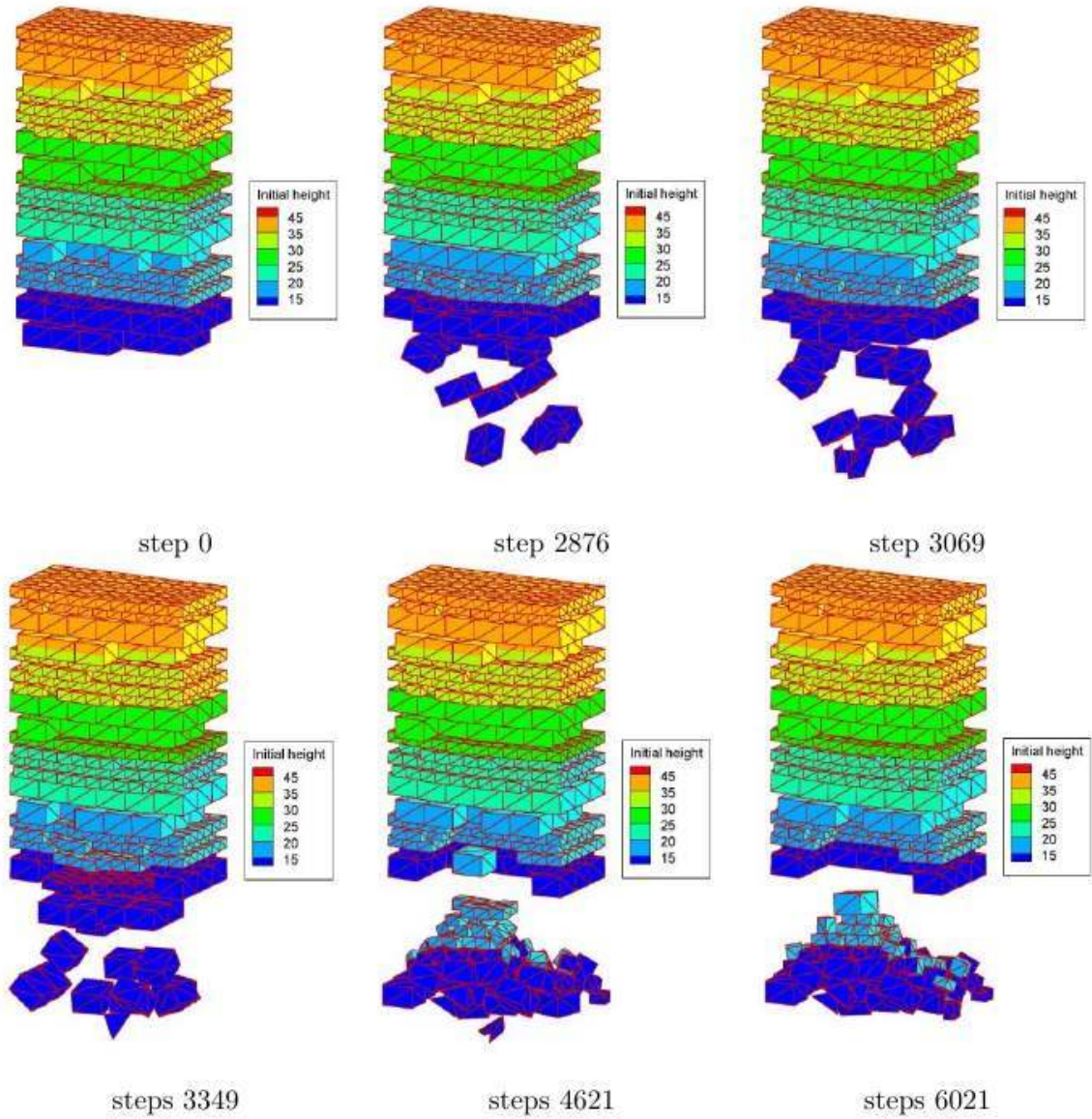


Figure 6.21: Sectional view of DPC2f. Ore fragments are coloured using their initial height. The displacements of the ore fragments around the central axis are larger than the displacements of the ore fragments that are far from the central axis.

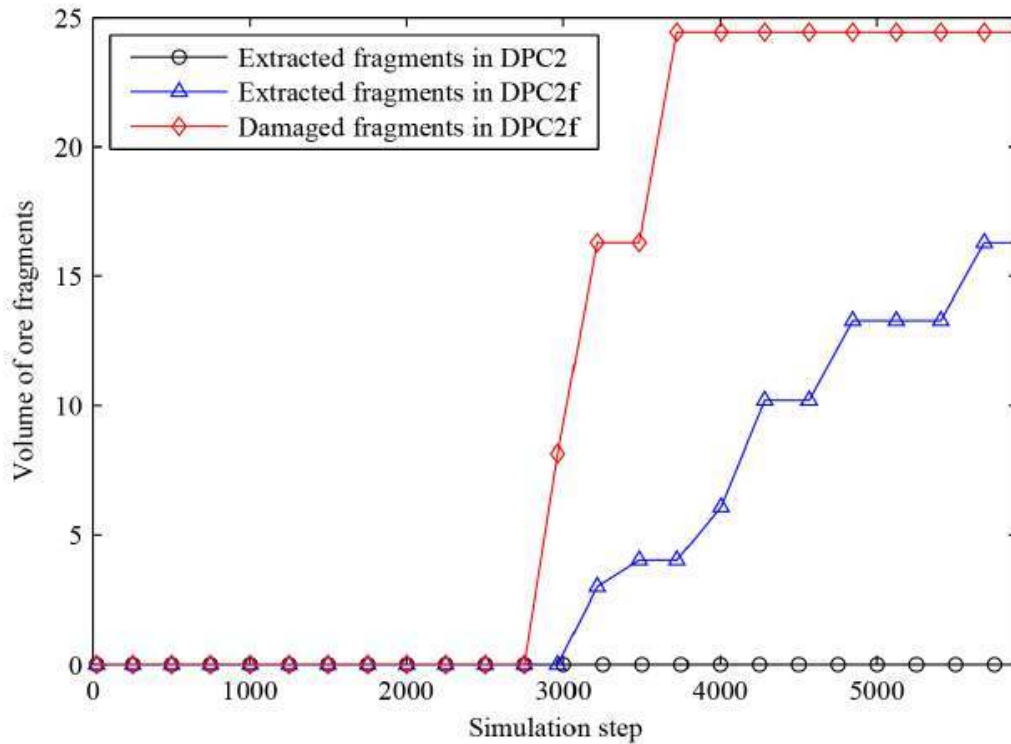


Figure 6.22: The volume of extracted ore fragments in DPC2, the volume of extracted ore fragments and damaged ore fragments in DPC2f. In DPC2, no fragmentation, no ore fragments falls through the draw point. However, in DPC2f, fragmentation happens, approximately 17m³ ore fragments are collected below draw point. The ore fragments hang-up at step 5900 approximately. Fragmentation mostly happens between step 2800 and step 3800, the volume of damaged ore fragments reaches 25 m³.

the draw point. The sectional view of DPC2f is shown in Fig. 6.21. It is found that, because of interlocking, most of the ore fragments remains frozen in their initial place, which was also found in DPC2. This means that the fragmentation process does not have a significant impact on the region far from the draw points. Compared to DPC1 and DPC1f, flowability is significantly influenced by the arrangement of small and large fragments. When the large fragments are at the bottom of ore bed, they tend to interlock with each other, and hold up the rest of the fragments. The volume of extracted ore fragments in DPC2, and the volume of extracted ore fragments and damaged ore fragments in DPC2f, are shown in 6.22. In DPC2, no fragmentation occurs, and no ore fragments pass through the draw point. However, in DPC2f, fragmentation happens, and approximately 17 m³ ore fragments are collected below the draw point. The ore fragments hang up at step 5900 (approximately). Fragmentation mostly happens between step 2800 and step 3800, and the volume of damaged ore fragments reaches 25 m³.

There are totally 3400 cuboid ore fragments initially. In DPC2f, at step 6021, the number of ore fragments reaches to 3415. DPC2 and DPC2f are both run on a personal computer which

uses Intel Xeon E5690 CPU with 3.45GHz (two processors), 12GB installed memory, and 12 cores. DPC2 and DPC2f spent approximately 26 days to run 6025 simulation steps respectively.

6.3.3 DPC3 and DPC3f

In DPC3 and DPC3f, the small fragments are distributed in the upper half of ore bed, and the large fragments are distributed in the lower half of the ore bed, as shown in Fig. 6.12 (c). The ore fragments are initially static, and start to move downward under gravity. The front view of DPC3 is illustrated in Fig. 6.23, where the boundary wall is displayed via the mesh. Similar to the snapshots of DPC3, without fragmentation, no ore fragment passes through the draw point. The large fragments block the draw point immediately after the start of the simulation. The sectional view of DPC3 is shown in Fig. 6.24, where ore fragments are coloured based on their initial height. The front view for DPC3f, where fragmentation happens, is shown in Fig. 6.25 and the sectional view of DPC3f is shown in Fig. 6.26, where the ore fragments are coloured based on their initial height. With fragmentation, some ore fragments pass through the draw point. Similar to DPC2 and DPC2f, because interlocking, most of the ore fragments remain nearly frozen in their initial place. Only the ore fragments in the bottom of ore bed fall down. The volume of extracted ore fragments in DPC3, and the volume of extracted ore fragments and damaged ore fragments in DPC3f are shown in 6.27. In DPC3, without fragmentation, no ore fragments falls through the draw point. However, in DPC3f, where fragmentation happens, approximately 2 m^3 of ore fragments are collected through the draw point. The ore fragments hang-up at step 2900, approximately. Fragmentation mostly happens between step 2200 and step 3, and the volume of damaged ore fragments reaches to 24 m^3 . Comparing Fig. 6.24 to Fig. 6.26, The influence of fragmentation on the displacement of most of the ore fragments is small, because fragmentation only exists around the draw point.

There are totally 3756 cuboid ore fragments initially. In DPC3f, at step 11301, the number of ore fragments reaches to 3415. DPC3 and DPC3f are both run on a personal computer which uses Intel Xeon E5690 CPU with 3.45GHz (two processors), 12GB installed memory, and 12 cores. DPC3 and DPC3f spent approximately 18 days and 13 days to run 5050 simulation steps respectively.

6.3.4 Surface subsidence associated with block caving mining

Table 6.2: Surface subsidence above the centre of the undercut.

	DPC1 at step 12000	DPC2 at step 12100	DPC3 at step 11300
surface subsidence	-3.1889m	-0.6743m	-1.5708

The block caving system is characterised by extracting a large amount of ore fragments underground. Ore fragments are fragmented and collected through draw points under their own

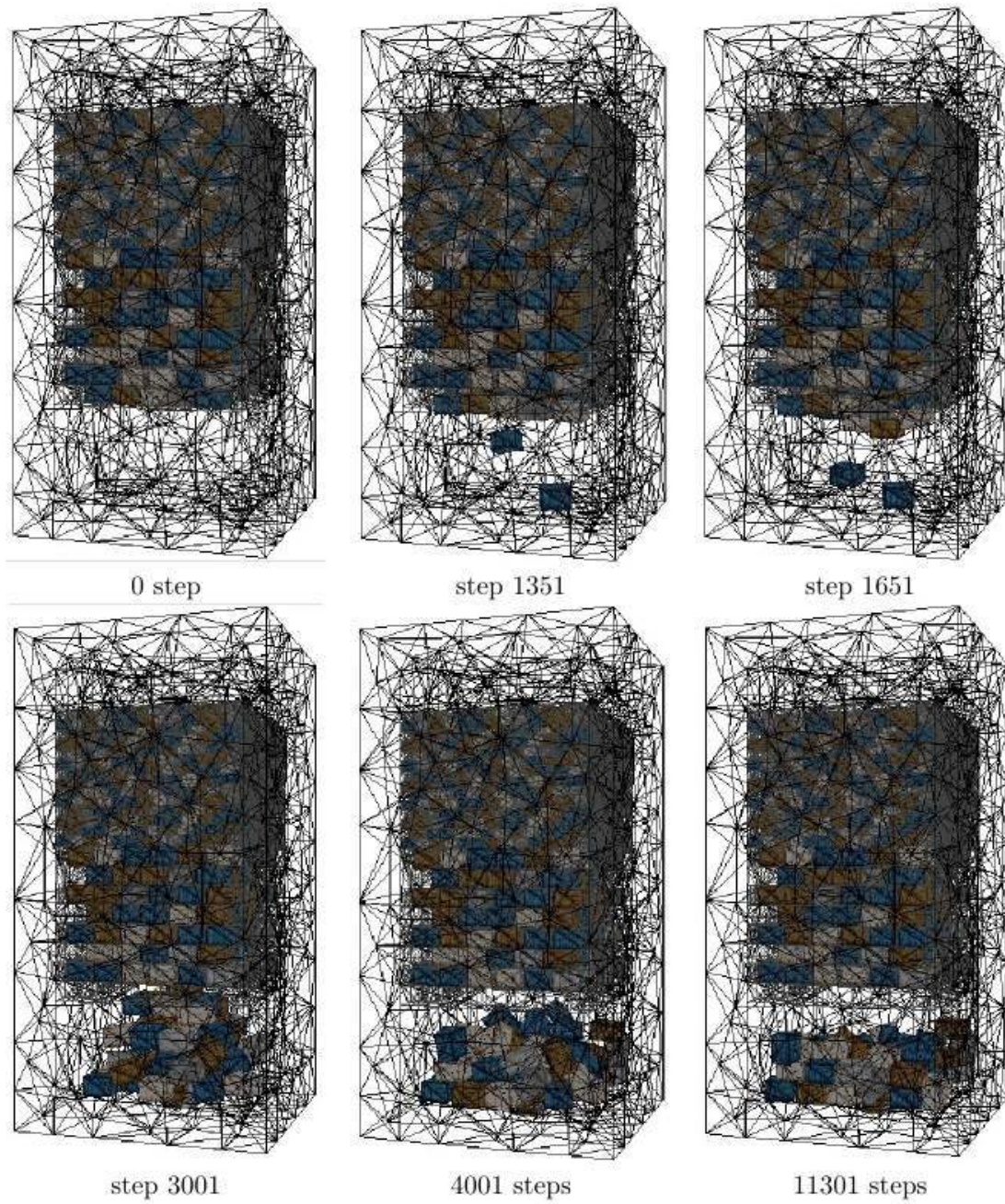


Figure 6.23: Front view of DPC3, where no fragmentation happens. Ore fragments are coloured arbitrarily to distinguish them from each other. No ore fragments pass through the draw point

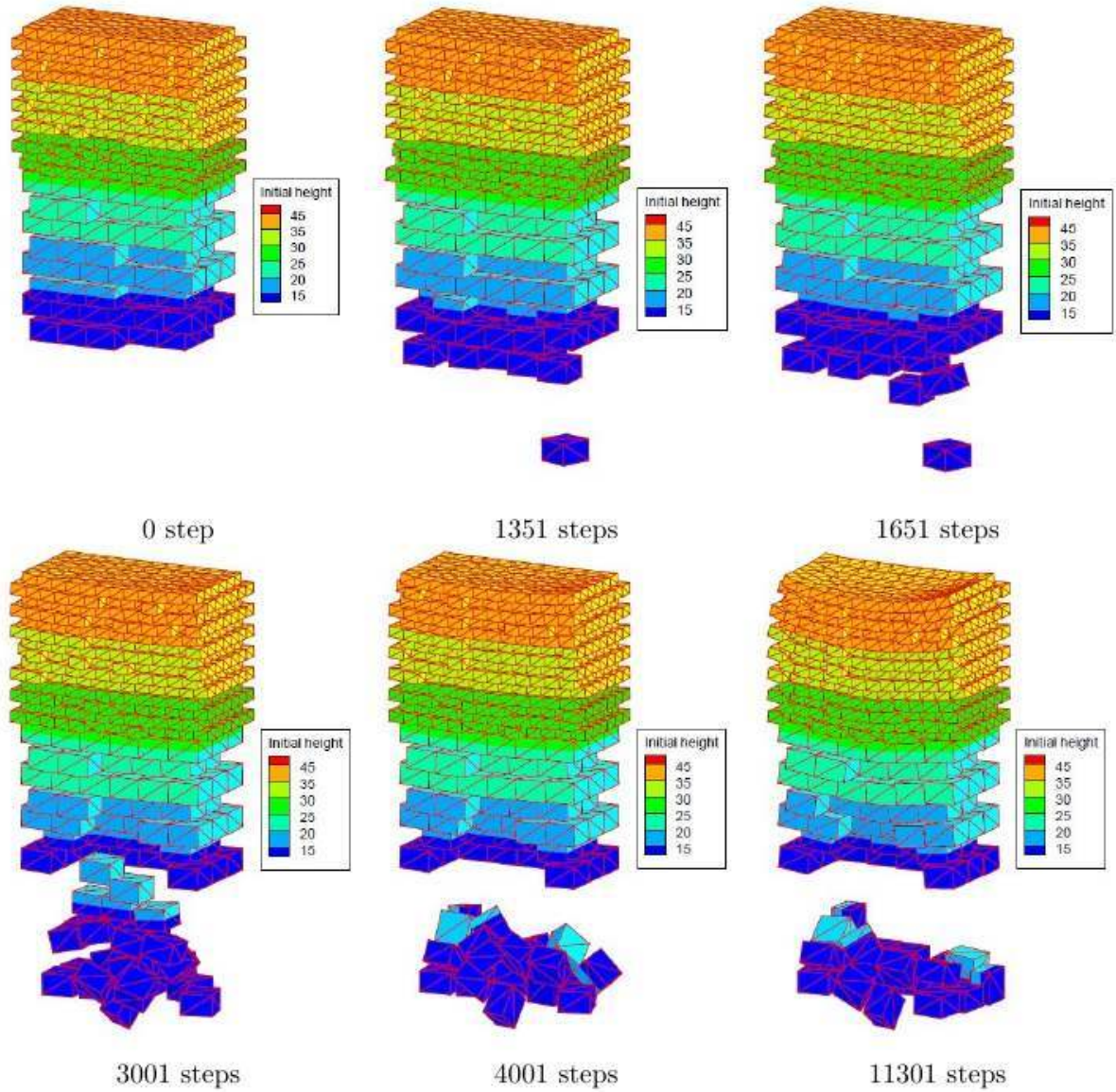


Figure 6.24: Sectional view of DPC3, where no fragmentation happens. Ore fragments are coloured based on their initial height.

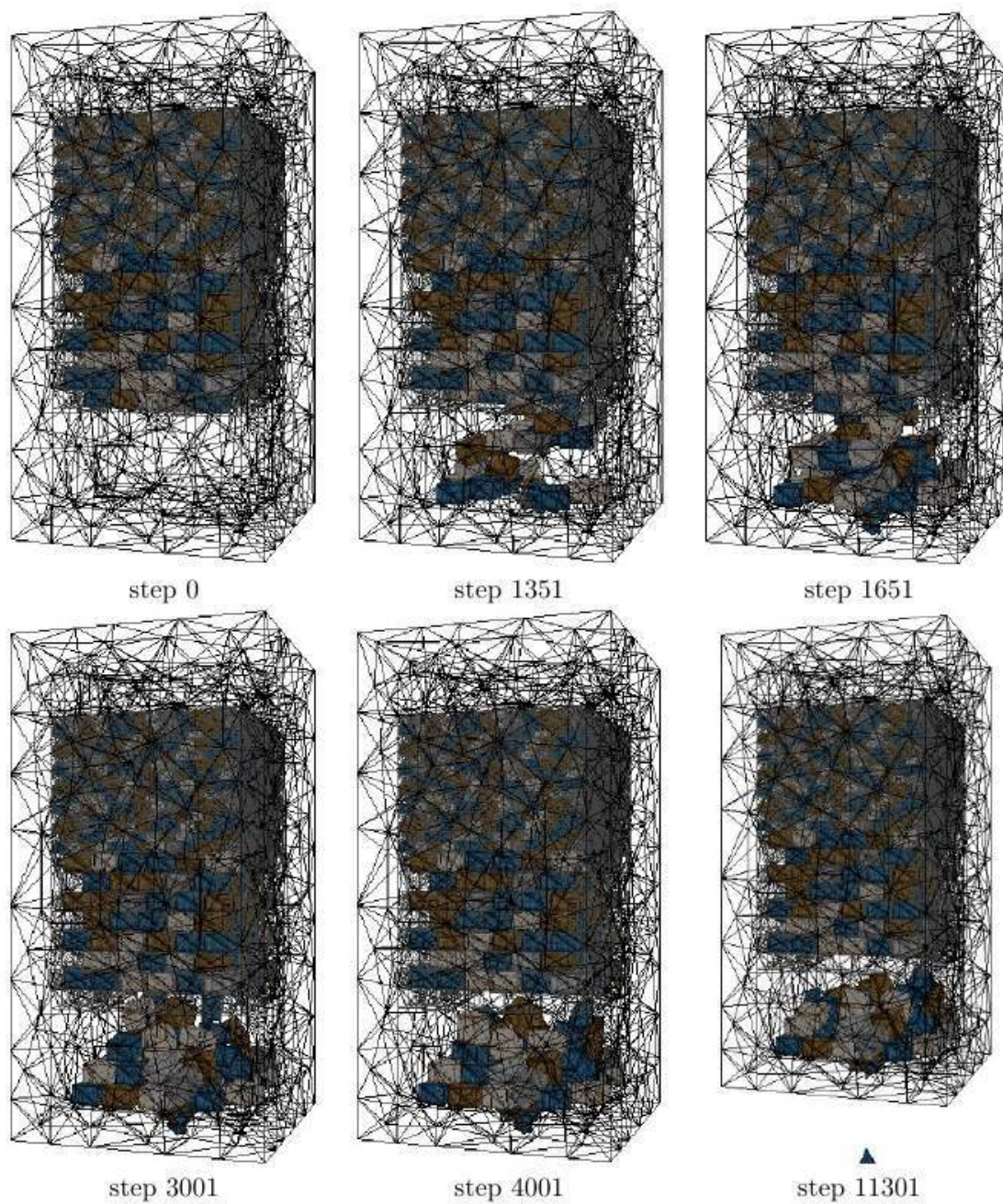


Figure 6.25: Front view of DPC3f, where fragmentation happens. Ore fragments are coloured arbitrarily to distinguish them from each other.

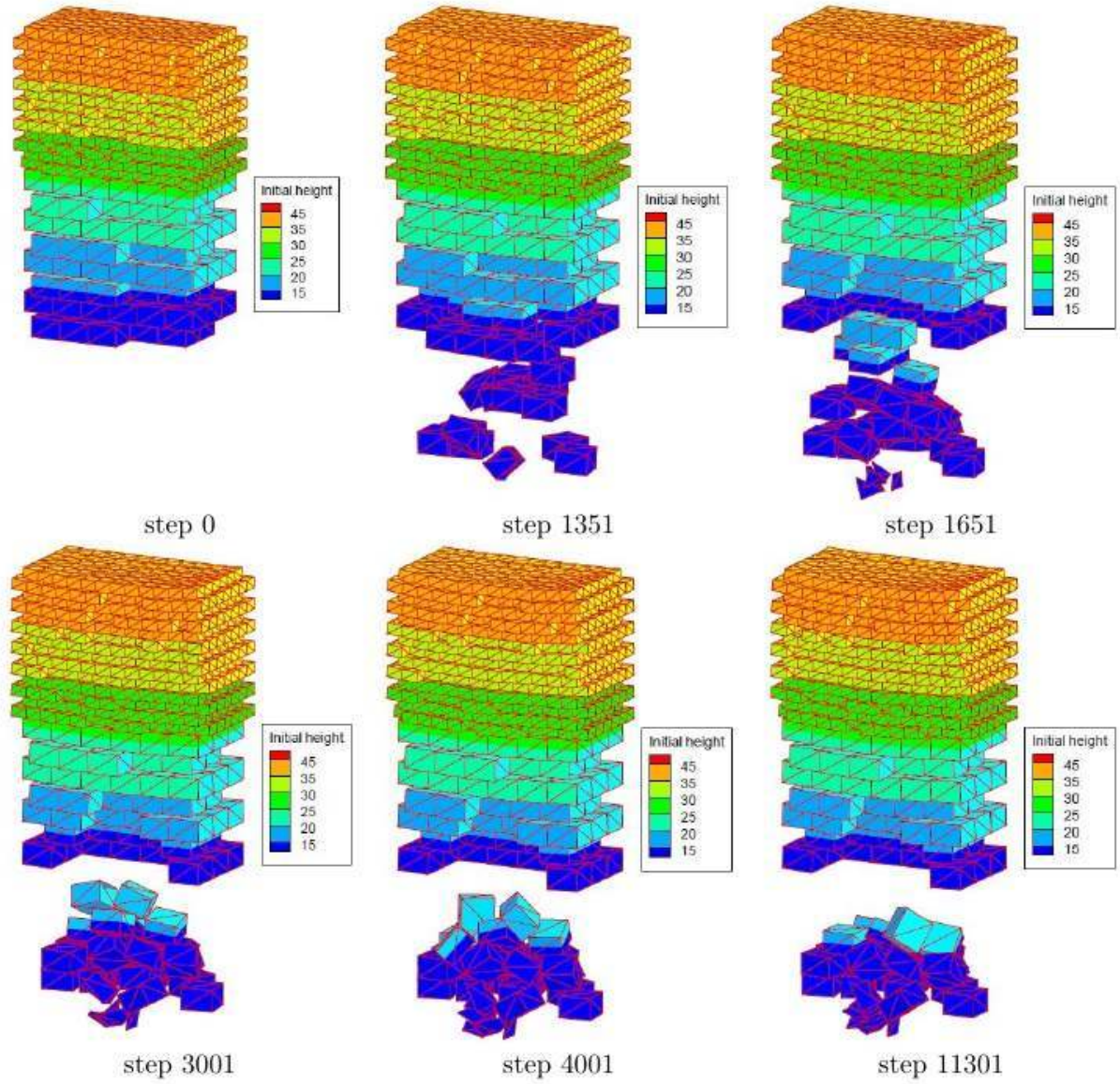


Figure 6.26: Sectional view of DPC3f, where fragmentation happens. Ore fragments are coloured based on their initial height.

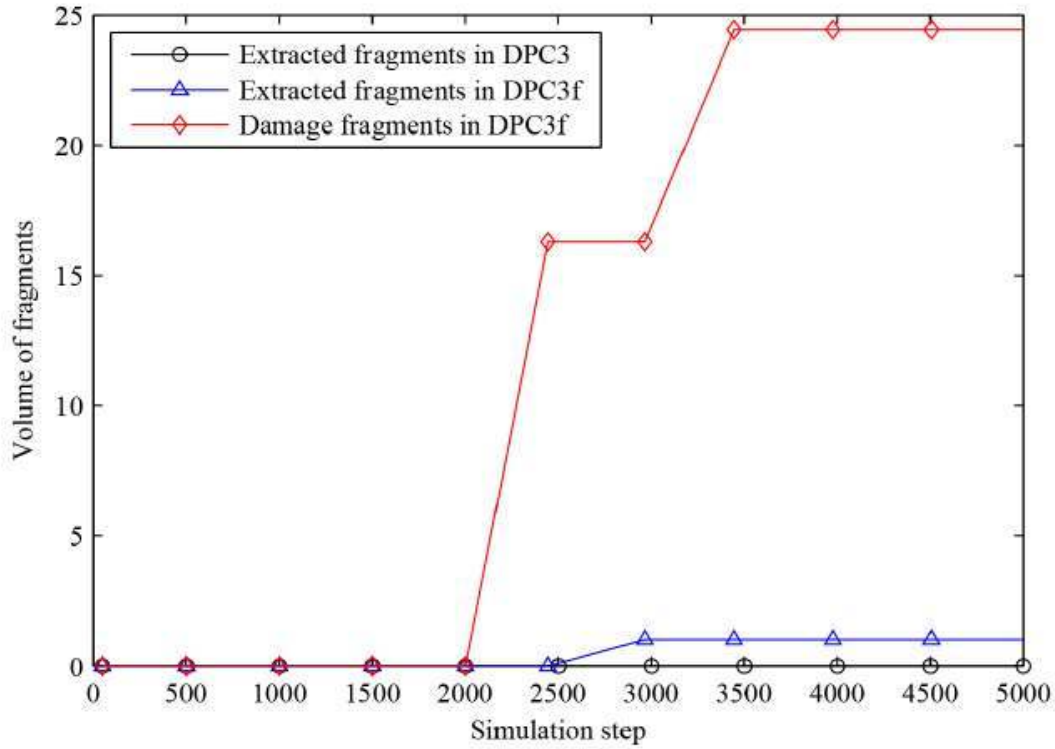


Figure 6.27: The volume of extracted ore fragments in DPC3, the volume of extracted ore fragments and crushed ore fragments in DPC3f. In DPC3, no fragmentation occurs, and no ore fragments fall through the draw point. However, in DPC3f, fragmentation happens, and approximately 2 m³ of ore fragments are collected below draw point. The ore fragments hang up at step 2900, approximately. Fragmentation mostly happens between step 2200 and step 3500, and the volume of damaged ore fragments reaches to 24 m³.

gravity. Extracting large amounts of ore fragments might trigger surface subsidence [Vyazmensky, 2008, Vyazmensky et al., 2007] which is a danger for the mining infrastructure and operational hazard assessments. Surface subsidence is also another important issue for both environmental impact and the efficiency of the mining process. Hoek [1974] used an analytical model to investigate the progressive hanging wall failure sequence with the increasing of depth of a mine. Numerical methods also provide opportunities to predict the surface subsidence phenomenon and govern the subsidence development [Vyazmensky, 2008, Vyazmensky et al., 2007]. As shown in Fig. 6.28, surface subsidence is found in these three tests after those ore fragments close to the draw point have fallen down. Because fragmentation is only considered in the vicinity of the draw point, fragmentation does not have a significant impact on the gravity flow in the region far from draw points, and only the results without fragmentation are considered here. The maximum subsidence above the centre of the undercut, for three tests at approximately same simulation step, are shown in Table 6.2. It is seen that the surface subsidence phenomenon is related to the distribution of small and large fragments. The surface

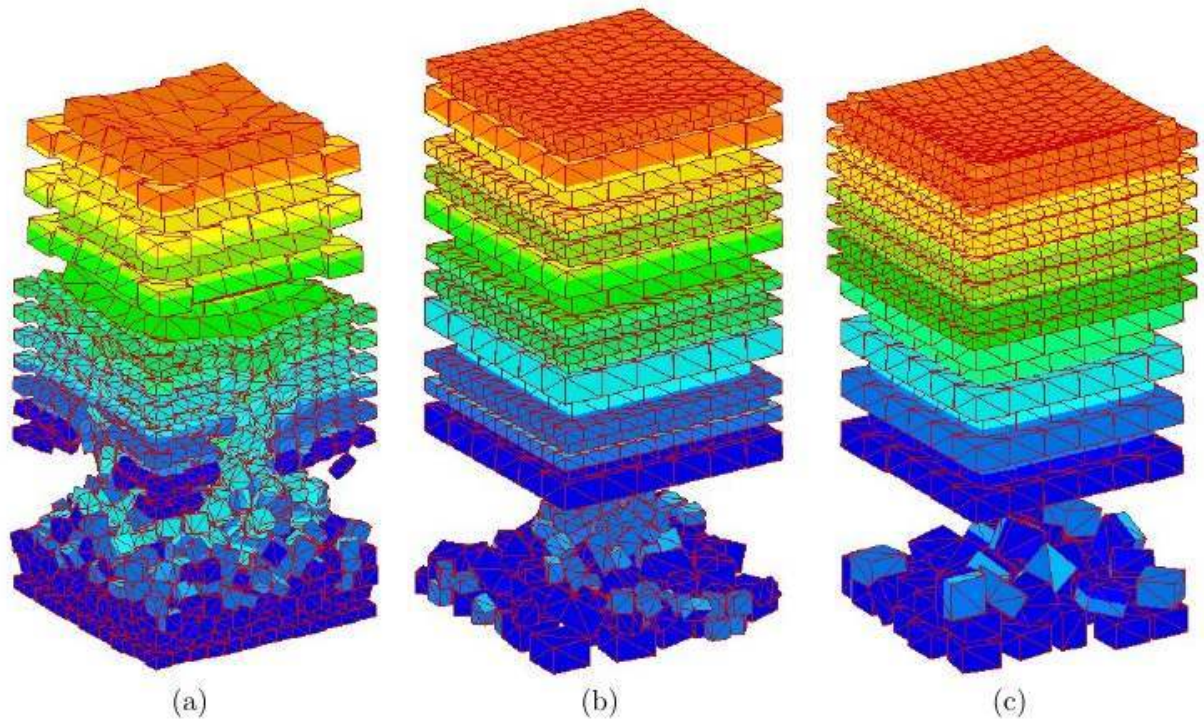


Figure 6.28: Surface subsidence of three tests.

subsidence of DPC1 is -3.1559 m, which is most severe. It can be seen from Fig. 6.16 and Fig. 6.14 that the displacement of ore fragments in DPC1 are much more than the displacement of ore fragments in DPC2 and DPC3. The value of surface subsidence of DPC3 (-1.5708 m) is slightly greater than the surface subsidence of DPC2 (0.6743 m).

6.4 Conclusions

Fragmentation was taken into account within the framework of a numerical investigation of the interlocking hang-up phenomena in the vicinity of a single draw point. First, the gravity flow of spherical ore fragments was investigated. Numerical results were compared between simulations that do or do not consider fragmentation during the fall through the draw point. It was observed that, with fragmentation, 1.93% more ore volume is extracted before hang-up when the initial diameter of ore fragments is 5.775 m, and 2.57% more ore volume of ore is extracted with fragmentation before hang-up when the initial diameter of ore fragments 7.5 m. This implies that the simulation with fragmentation predicts higher extraction rates than its non-fragmentation counterpart. These numerical tests illustrate the importance of modelling fragmentation during the simulation of draw point extraction as part of block caving mine planning.

Then, the gravity flow of cuboid ore fragments was investigated. Two different sizes of ore fragments were considered, small ore fragments and large ore fragments. Moreover, three

different arrangements of small fragments and large fragments were considered. In test DPC1 and DPC1f, all the large ore fragments are located higher than the small ore fragments. In DPC2 and DPC2f, the layers of small fragments and the layers of large fragments are mixed. In DPC3 and DPC3f, all the small ore fragments are higher than the large ore fragments. The fragmentation process does not occur in DPC1, DPC2 and DPC3, whereas fragmentation does occur in DPC1f, DPC2f and DPC3f. Numerical results show that DPC1f extracts 4 m³ more ore fragments than DPC1. DPC2 and DPC3 do not extract any ore fragments, whereas DPC2f and DPC3f extract 17 m³ and 2 m³ of ore fragments, respectively, before the hang-up of ore fragments occurs. This means that when the size and position of the ore fragments are same, the weaker material, which leads to fragmentation, predicts higher extraction rates than does its non-fragmentation counterpart. Furthermore, the arrangement of ore fragments has a significant impact on the gravity flow of ore fragments. Greater flowability is found in DPC1 and DPC1f, where the big fragments are all higher than the small fragments, compared to the other four tests. In addition, greater flowability leads to greater surface subsidence.

7 Conclusions and recommendations for future research

7.1 Conclusion

The present work focussed on the simulation of fragmentation, which must simultaneously resolve fragment dynamics and fracture propagation, while keeping track of the geometry and energetic state of thousands of created fragments. The main contributions of this thesis are the following:

- (a) a rigorous discussion of the energy conservation property of the impulse based method;
- (b) the development of a fragmentation approach that combines the finite element method (FEM) and the impulse-based method;
- (c) a low-cost fragmentation approach based on the Weibull distribution has been implemented to simulation fragmentation;
- (d) an analysis and discussion of the influence of secondary fragmentation on block caving.

7.1.1 Modelling particle collisions

The energy conservative property and momentum conservative property of the impulse-based method was investigated, and a necessary condition for the impulse method to be energy conservative was proposed. In the context of the impulse-based method, Guendelman and Hahn proposed two versions of algorithms to handle collisions in numerical tests. Guendelman's impulse-based approach, which conserves energy, was shown to be a stable and fast method for collision resolution in a system of non-convex polyhedral objects. Only Guendelman's approach is able to ensure momentum and energy conservation during simulations, whereas Hahn's approach only conserves momentum.

Previously existing impulse-based methods suffer from some drawbacks. For example, in the simultaneous impulse-based method (SMM) and sequential impulse-based method (SQM), the relative velocity at the contact points after collision is directly derived from the relative velocity before the collision, and the impulse is a function of the change of relative normal velocity. Since the change of relative velocity at the contact points is implemented directly, it leads to instability in the simulation. A novel energy tracking method (ETM) was proposed to compute collision responses in a multi-body system, which changes the relative velocity gradually among multiple collisions. In the ETM, the relative velocity is adjusted gradually over a set of iterations, and within each of these iterations their effect is considered simultaneously. By applying velocities

gradually to the contact points, the impulses are able to influence multiple contact points at the same time, enabling the propagation of impulse effects, and thereby improving the overall ability of the method to capture propagation of forces during dynamic simulations.

7.1.2 Simulation of fragmentation

Two approaches to modelling fragmentation, a finite-discrete element approach, and a low cost, velocity dependent fragmentation pattern-based approach, were presented for fragmentation simulation. The former approach is based on advanced re-meshing techniques and FEM, in which cracks are represented explicitly based on their geometry. The path of crack propagation in this approach can be arbitrary, and is not necessarily forced to follow the element boundaries. Fragmentation is controlled by fracturing, rather than driven by bond breakage. This approach suffers from the drawback of massive computational cost. In order to meet the requirement of the fragmentation simulation of hundreds and thousands of colliding bodies, the latter approach used a low-cost fragmentation pattern based on the Weibull distribution to separate colliding bodies directly. This approach makes use of patterns and size distributions computed during full finite/discrete element simulations. It generates informed cutting boxes that create a spatial subdivision, which reproduces the fragment size distribution, while generating a low-cost subdivision of the analysed fragment. This method is fast, and ensures physical accuracy by learning existing fragmentation patterns from experiments and numerical results. Validations of the low-cost fragmentation method included the impact-induced fragmentation of a single spherical rock fragment and a realistically shaped rock fragment, and the crushing of an assemblage of spherical rock fragments.

7.1.3 Influence of secondary fragmentation in block caving

Fragmentation was taken into account within the framework of a numerical investigation of the interlocking hang-up phenomena in the vicinity of a single draw point. Numerical results of both spherical and cuboidal ore fragments showed that the simulations with fragmentation predict higher extraction rates than do their non-fragmentation counterparts. Results indicate that the spatial arrangement of poly-disperse ore fragments has a significant impact on the gravity flow of ore fragments. Thus, simulating the flow of randomly placed spherical fragments arranged around one or more draw points, even if considering fragmentation, may not accurately represent the complexity of the effects of breakage and spatial arrangement on draw point flow. The presented numerical tests contribute to showing the importance of modelling fragmentation and gravity flow in the simulation of draw point extraction as part of block caving mine planning.

Enhancing the physical accuracy of simulations has been shown to improve the numerical stability of the presented fragment dynamics numerical method. The consistent energy-based modelling is critical to ensuring physical accuracy, as it affects behaviour by controlling the velocity of the fragments, and influencing fragmentation. Comparing Guendelman's and Hahn's

impulse-based approaches, the performance of Guendelman's impulse-based approach, which is able to ensure the energy conservation, is able to achieve reasonable results when it is used to model the gravity flow of ore fragments around a single draw point test (see section 2.3.3), however Hahn's approach fails. In addition, by modelling the propagation of contact forces more accurately, the results of ETM are more accurate than those of the other two previously used impulse-based methods. Moreover, it is important to consider the limitation of computational cost, especially for simulations that involve hundreds and thousands of ore fragments in engineering. In order to address this limitation, the thesis proposes to shift the paradigm of fragmentation simulation moving away from FEM and linear fracture mechanics based simulation towards fragmentation pattern based simulation. Using the fragmentation pattern, which is established based on both experimental results, and on FEM and linear fracture mechanics based simulations, the computational cost is reduced sharply. The proposed pattern-based simulation provides an attractive prospect to simulate the fragmentation of large numbers of colliding rock blocks.

7.2 Recommendations for future work

Although detailed work has been done to resolve collision responses and simulate fragmentation process in this work, the understanding of the complicated process of collision and fragmentation is still limited. In order to enhance this understanding, the following additional work is recommended:

7.2.1 Improve the low-cost Weibull distribution based fragmentation pattern

In order to reduce computational costs and improve numerical stability, the present work makes use of the low-cost Weibull distribution based fragmentation pattern to fragment colliding bodies directly, thereby avoiding the difficulties of re-meshing and regenerating the geometry during the simulations. Even though this fragmentation pattern has been validated by a series of numerical tests, it is still in its infancy. To meet the requirements of modelling complicated fragmentation processes in engineering, more factors that influence the fragmentation pattern should be taken into account.

- **Shape** In the present work, the experiments associated to the fragmentation of spheres are used to establish the fragmentation pattern. However, in a real block caving system, the ore fragments have arbitrary and complicated shapes. To simulate the fragmentation of ore fragments with arbitrary shapes, more geometrical factors, which have significant influence on fragmentation, should be taken into account when generating the fragmentation pattern. For example, the results of experiments provide strong evidence for the impacts of aspect ratio and angularity on fragmentation potential under compression. The ore fragments with higher aspect ratio are expected to be susceptible to bending loads [McDowell and Humphries, 2002]. The impact of angularity has been discussed in the

model by [Hardin, 1985] and other existing experiments [Lee, 1992, Lee and Farhoomand, 1997, Lee and Seed, 1967]. The model of [Hardin, 1985] suggests that fragmentation potential increase with increasing angularity. In addition, this model also suggests that higher void ratio or porosity also increases the potential for fragmentation. The reason for this is that, considering the fragmentation in a group of ore fragments, higher porosity decreases the number of neighbouring contacts and increases the shearing-induced tensile stress in fragments.

- **Strength** The fragmentation pattern in the present work summarises the results of experimental works carried out with uniform materials. However, the mechanical material properties have significant influence on the induced dynamic response of ore fragments [Wu and Chau, 2006]. The initial distribution of flaws and fracture toughness obviously influences the propagation of cracks and affects the final fragmentation pattern.
- **Impact energy and impacting angle** The critical role of impact energy has been shown in many experiments [Giacomini et al., 2009, Wu et al., 2004]. Further studies are required to enhance the understanding of the relationship between the impacting energy and other collision parameters, including the restitution coefficient, the friction coefficient, and the resultant change of relative velocity. Moreover, the impacting angle plays a key role in the fragmentation process. Giacomini et al. [2009] suggests that compared to the effect of the impact angle, the effect of the impact energy tends to be of second order.

7.2.2 Reproduce the realistic shape of ore fragments in block caving

The present work modelled the gravity flow and secondary fragmentation in a block caving system. However, the current simulator has some difficulties in reproducing the gravity flow in block caving.

- **Reproduce more accurate gravity flow around a single draw point** Although spherical particles [Pierce, 2010] and cubic half-bricks [McNearny and Abel, 1993] have been used to successfully predict the character of gravity flow in the isolated movement zone (IMZ) [Castro et al., 2007] and the isolated extraction zone (IEZ) [Castro et al., 2007], it is still worthwhile to generate the ore fragments using realistic shapes and initial size distribution. The realistic shape and initial size distribution can be exploited in future work, to improve the accuracy of the simulation of interlocking, friction, and fragmentation of broken rocks. Moreover, joints and fractures measured from on site monitoring should be utilised to fragment the ore bed and generate initial ore fragments.
- **Multiple interactive draw points** The present work has investigated gravity flow in the vicinity of a single draw point. In the process of block caving, multiple draw points are normally exploited to extract ore fragments underground at the same time [Brown, 2003]. Neighbouring draw points have an influence on each other, and change the behaviour

of gravity flow. Obviously, increasing the number of draw points will increase the cost of mining. However, increasing the spacing between two neighbouring draw points, and reducing the number of draw points, might lead to the abandonment of large volumes of potentially caved rock. Therefore, understanding the interaction of multiple draw points is crucial to the improvement of the design of the spacing between the draw points.

References

- S. Abe and K. Mair. Effects of gouge fragment shape on fault friction: New 3D modeling results. *Geophysical Research Letters*, 36(23):L23302, 2009.
- F. Agliardi and G. Crosta. High resolution three-dimensional numerical modelling of rockfalls. *International Journal of Rock Mechanics and Mining Sciences*, 40:455–471, 2003.
- A. Ali and S. Bradshaw. Quantifying damage around grain boundaries in microwave treated ores. *Chemical Engineering and Processing: Process Intensification*, 48(11-123):1566–1573, 2009.
- A. Ali and S. Bradshaw. Confined particle bed breakage of microwave treated and untreated ores. *Minerals Engineering*, 24(14):1625–1630, 2011.
- J. Anthony and C. Marone. Influence of particle characteristics on granular friction. *Journal of Geophysical Research: Solid Earth*, 110(B8):DOI: 10.1029/2004JB003399, 2005.
- N. Azevedo and J. Lemos. Hybrid discrete element finite element method for fracture analysis. *Computer Methods in Applied Mechanics and Engineering*, 195(33-36):4579–4593, 2006.
- I. Babuška. The finite element method with penalty. *Mathematics of Computation*, 27(122):221–228, 1973.
- K. Bagherzadeh, A. Mirghasemi, and S. Mohammadi. Numerical simulation of particle breakage of angular particles using combined DEM and FEM. *Powder Technology*, 205(1-3):15–29, 2010.
- P.K. Banerjee and R. Butterfield. *Boundary Element Methods in Engineering Science*. McGraw-Hill Inc.,US, 1981.
- L. Banks-Sills. Application of the finite element method to linear elastic fracture mechanics. *Applied Mechanics Reviews*, 44:447–461, 1991.
- L. Banks-Sills and D. Sherman. Comparison of methods for calculating stress intensity factors with quarter-point elements. *International Journal of Fracture*, 32:127–140, 1986.
- D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. *Computer Graphics (Proc. SIGGRAPH)*, Orlando, FL, 28:23–34, 1994.

- S.G. Bardenhagen, J.U. Brackbill, and D. Sulsky. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering*, 187(3-4):529–541, 2000.
- R. Barsoum. Further application of quadratic isoparametric finite elements to linear fracture mechanics of plate bending and general shells. *International Journal for Numerical Methods in Engineering*, 11:167–169, 1976.
- S.R. Beissel, G.R. Johnson, and C.H. Popelar. An element-failure algorithm for dynamic crack propagation in general directions. *Engng Fract Mech*, 61(3-4):407–425, 1998.
- T. Belytschko. The splitting pinball method for contact-impact problems. *Computer Methods in Applied Mechanics and Engineering*, 105(3):375–393, 1993.
- T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.
- T. Belytschko, Y.Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994.
- T. Belytschko, Y.Y. Lu, and L. Gu. Crack propagation by element-free galerkin methods. *Engineering Fracture Mechanics*, 51(2):295–315, 1995a.
- T. Belytschko, Y.Y. Lu, and L. Gu. Element-free galerkin methods for static and dynamic fracture. *International Journal of Solids and Structures*, 32:2547–2570, 1995b.
- T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47, 1996.
- M.J. Beus, S. Iverson, and B. Stewart. Application of physical modelling and particle flow analysis to evaluate ore-pass design. volume 1, page 112, The Institute of Mining and Metallurgy, 1997.
- L.G. Bierawski and S. Maeno. DEM-FEM model of highly saturated soil motion due to seepage force. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 132(5):401–409, 2006.
- T. N. Bittencourt, P. A. Wawrzynek, A. R. Ingraffea, and J. L. Sousa. Quasi-automatic simulation of crack propagation for 2D lefm problems. *Engineering Fracture Mechanics*, 55:321–324, 1996.
- G.E. Blandford, A.R. Ingraffea, and J.A. Liggett. Two-dimensional stress intensity factor computations using the boundary element method. *International Journal for Numerical Methods in Engineering*, 17(3):387–404, 1981.
- S. Bordas, T. Rabczuk, and G. Zi. Three-dimensional crack initiation, propagation, branching and junction in non-linear materials by an extended mesh-free method without asymptotic enrichment. *Engineering Fracture Mechanics*, 75(5):943–960, 2006.

- P.O. Bouchard, F. Bay, Y. Chastel, and I. Tovenar. Crack propagation modelling using an advanced remeshing technique. *Computer Methods in Applied Mechanics and Engineering*, 189(723-742), 2000.
- N. Bourago and V. Kukudzhanov. A review of contact algorithms. *The institute for problems in mechanics of RAS, Moscow, Izv. RAN. MTT.*, 1:45–87, 2005.
- E.T. Brown. *Block caving geomechanics*. Julius Kruttschnitt Mineral Research Centre, University of Queensland, 2003.
- E. Budyn, G. Zi, N. Mos, and T. Belytschko. A method for multiple crack growth in brittle materials without remeshing. *International Journal for Numerical Methods in Engineering*, 61(10):1741–1770, 2004.
- F. Camborde, C. Mariotti, and F. Donz. Numerical study of rock and concrete behaviour by discrete element modelling. *Computers and Geotechnics*, 27(4):225–247, 2000.
- J. Campbell, R. Vignjevic, and L. Libersky. Computer methods in applied mechanics and engineering. *Computer Methods in Applied Mechanics and Engineering*, 184(1):49–65, 2000.
- R. Castro, R. Trueman, and A. Halim. A study of isolated draw zones in block caving mines by means of a large 3D physical model. *International Journal of Rock Mechanics and Mining Sciences*, 44(6):860–870, 2007.
- J. Cervenka and V. Saouma. Numerical evaluation of 3-D sif for arbitrary finite element meshes. *Engineering Fracture Mechanics*, 57(5):541–563, 1997.
- R. Chandramohan and M.S. Powell. Measurement of particle interaction properties for incorporation in the discrete element method simulation. *Minerals Engineering*, 18(12):1142–1151, 2005.
- B. Chang and J. E. Colgate. Real-time impulse-based simulation of rigid body systems for haptic display. In *Proceedings of the 1997 ASME International Mechanical Engineering Congress and Exhibition*, pages 1–8, 1997.
- B. Chapman, G. Jost, and R. Pas. *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*. MIT Press, 2007.
- J.R. Chardonnet, S. Miossec, A. Kheddar, H. Arisumi, H. Hirukawa, F. Pierrot, and K. Yokoi. Dynamic simulator for humanoids using constraint-based method with static friction. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pages 1366–1371, Kunming, China, December 2006.
- R.J. Charles. Static fatigue of glass. *Journal of Applied Physics*, 29:1549–1560, 1958.

- V. Chawla and T.A. Laursen. Energy consistent algorithms for frictional contact problems. *International Journal for Numerical Methods in Engineering*, 42(5):799–827, 1998.
- G. Chen. Stochastic modeling of rock fragment flow under gravity. *International Journal of Rock Mechanics and Mining Science*, 34:323–31, 1997.
- J.S. Chen, C. Pan, C.T. Wu, and W.K. Liu. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):195–227, 1996.
- Y.S. Cheong, G.K. Reynolds, A.D. Salman, and M.J. Hounslow. Modelling fragment size distribution using two-parameter weibull equation. *International Journal of Mineral Processing*, 74(0):S227–S237, 2004.
- G.P. Cherepanov. The propagation of cracks in a continuous media. *Journal of Applied Mathematics and Mechanics*, 31:503–512, 1967.
- L. Chunsheng, D. Robert, and F. Dieter. Fracture statistics of brittle materials: Weibull or normal distribution. *Physical Review E*, 65:067102, Jun 2002.
- A.P. Cisilino and M.H. Aliabadi. Three-dimensional bem analysis for fatigue crack growth in welded components. *International Journal of Pressure Vessels and Piping*, 70(2):135–144, 1997.
- P.W. Cleary and M.L. Sawley. DEM modeling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge. *Applied Mathematical modeling*, 26:89–111, 2002.
- M. Coutinho. *Dynamic simulations of multibody systems*. Springer, 2001.
- T.A. Cruse. Numerical solutions in three dimensional elastostatics. *International Journal of Solids and Structures*, 5:1259–1274, 1969.
- T.A. Cruse. Numerical evaluation of elastic stress intensity factors by the boundary integral equation method. In *Swedelow JL (ed.) The surface Crack: Physical Problems and Computational Solutions ASME*, pages 153–170, New York, 1972.
- P. A. Cundall. Formulation of a three-dimensional distinct element model. Part I: A scheme to detect and represent contacts in a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 25:107–116, 1988.
- P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29:47–65, 1979.

- C. Daux, N. Moes, J. Dolbow, N. Sukumar, and T. Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48(12):1741–1760, 2000.
- C. Denoual and F. Hild. A damage model for the dynamic fragmentation of brittle solids. *Computer Methods in Applied Mechanics and Engineering*, 183(3-4):247–258, 2000.
- N. Djordjevic and R. Morrison. Exploratory modelling of grinding pressure within a compressed particle bed. 19(10):995–1004, 2006.
- J. Dolbow, N. Moes, and T. Belytschko. An extended finite element method for modelling crack growth with frictional contact. *Computer Methods in Applied Mechanics and Engineering*, 190:6825–6846, 2001.
- E. Drumwright. A fast and stable penalty method for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):231–240, 2008.
- M. Duflot. A meshless method with enriched weight functions for three-dimensional crack propagation. *International Journal for Numerical Methods in Engineering*, 65(12):1970–2006, 2006.
- P. Duplancic. *Characterisation of Caving Mechanism Through Analysis of Stress and Seismicity*. University of Western Australia, 2003.
- D. Elmo, A. Vyazmensky, and D. Stead. A hybrid FEM/DEM approach to model the interaction between open-pit and underground block-caving mining. In *Proceedings of the 1st Canada-US Rock Mechanics Symposium*, 27-31 May 2007 2007.
- D. Elmo, D. Stead, E. Eberhardt, and A. Vyazmensky. Applications of finite/discrete element modeling to rock engineering problems. *International Journal of Geomechanics*, 13(5):565–580, 2013.
- K. Erleben, J. Sporring, K. Henriksen, and H. Dohlmann. *Physics Based Animation*. Charles River Media, 2005.
- P. Fedelinski, M.H. Aliabadi, and D.P. Rooke. A single-region time domain bem for dynamic crack problems. *International Journal of Solids and Structures*, 32(24):3555–3571, 1995.
- K. Ferenc and J.H. Hans. A study of fragmentation processes using a discrete element method. *Computer Methods in Applied Mechanics and Engineering*, 138(1-4):3–18, 1996.
- Ferguson. *Ore pass design guidelines*. Report to Mining Research Directorate, 1991.
- K.E. Fitzgibbon and T.J. Veasy. Thermally assisted liberation - a review. *Minerals Engineering*, 3(1/2):181–185, 1990.

- M. Fleming, Y.A. Chu, B. Moran, and T. Belytschko. Enriched element-free galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering*, 40(8):1483–1504, 1997.
- A. Freidin, S. Neverov, A. Neverov, and P. Filippov. Mine stability with application of sublevel caving schemes. *Journal of Mining Science*, 44(1):82–91, 2008.
- T. Fries and M. Baydoun. Crack propagation with the extended finite element method and a hybrid explicit-implicit crack description. *International Journal for Numerical Methods in Engineering*, 89(12):1527–1558, 2012.
- D.W. Fuerstenau and A.Z.M. Abouzeid. The energy efficiency of ball milling in comminution. *International journal of Mineral Processing*, 67(1-4):161–185, 2002.
- H. Garstenauer. *A Unified Framework for Rigid Body Dynamics*. PhD thesis, Johannes Kepler Universitaet Linz, 2006.
- J. Ghaboussi. Fully deformable discrete element analysis using a finite element approach. *Comput. Geotech.*, 5:175–195, 1988.
- A. Giacomini, O. Buzzi, B. Renard, and G.P. Giani. Experimental studies on fragmentation of rock falls on impact with rock surfaces. *International Journal of Rock Mechanics and Mining Sciences*, 46(4):708–715, 2009.
- R. Gingoid and J. Monaghan. Smoothed particle hydrodynamics theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.
- D.A. Gorham and A.D. Salman. The failure of spherical particles under impact. *Wear*, 258(1-4):580–587, 2005.
- D.E. Grady and M.E. Kipp. Continuum modeling of explosive fracture in oil shale. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 17(3):147–157, 1980.
- R. Grayeli and A. Mortazavi. Discontinuous deformation analysis with second-order finite element meshed block. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(15):1545–1561, 2006.
- M. Griebel and M. A. Schweitzer. *Meshfree Methods for Partial Differential Equations*, volume 22, chapter A particle-partition of unity method. Part IV: Parallelization. Springer Verlag, 2002.
- E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with staking. *ACM Transaction on Graphics*, 22(3):871–878, 2003.

- R. Guises, J. Fand J. Latham, and A. Munjiza. Granular packing: numerical simulation and the characterisation of the effect of particle shape. *Granular Matter*, 11(5):281–292, 2009.
- Ercan Gurses and Christian Miehe. A computational framework of three-dimensional configurational-force-driven brittle crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 198(15-16):1413–1428, 2009.
- T.O. Hagan and E. Acheampong. Current design, support and maintenance of rockpasses and assessment of practices applicable at depth. in: Hagan to, editor. In *Proceedings of the 2nd Southern African rock engineering symposium*, pages 62–79, South Africa: ISRM, 1999.
- J. K. Hahn. Realistic animation of rigid bodies. *Computer Graphics*, 22(4):299–308, 1988.
- G. Haikal and Hjelmstad. A finite element formulation of non-smooth contact based on oriented volumes for quadrilateral and hexahedral elements. *Computer Methods in Applied Mechanics and Engineering*, 196:4690–4711, 2007.
- V. Hajiabdolmajid, P. Kaiser, and C. Martin. Modelling brittle failure of rock. *International Journal of Rock Mechanics and Mining Sciences*, 39(6):731–741, 2002.
- B.O. Hardin. Crushing of soil particles. *Journal of Geotechnical Engineering ASCE*, 111(10):1177–1192, 1985.
- R. Hart, P.A. Cundall, and J. Lemos. Formulation of a three-dimensional distinct element model - part II. mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences*, 25(3):117–125, 1988.
- S. Hasegawa and N. Fujii. Real-time rigid body simulation based on volumetric penalty method. In *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 326–332, 2003a.
- S. Hasegawa and N. Fujii. Real-time rigid body simulation based on volumetric penalty method. *Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2003, 11th Symposium on.*, 2003b.
- R. D. Henshell and K. G. Shaw. Crack tip finite elements are unnecessary. *International Journal for Numerical Methods in Engineering*, 9:495–507, 1975.
- S. Hentz, L. Daudeville, and F.V. Donze. Identification and validation of a discrete element model for concrete. *Journal of Engineering Mechanics (ASCE)*, 130:709–719, 2004.
- Y. Hiramatsu and Y. Oka. Determination of the tensile strength of rock by a compression test of as irregular test piece. *International Journal of Rock Mechanics and Mining Sciences*, 3: 89–99, 1966.

- E. Hoek. Progressive caving induced by mining an inclined ore body. *Trans. Inst. Mining and Metallurgy, London*, 83:A133–A139, 1974.
- M. Holland. *Structure of fault zones in cohesive volcanic sequences*. PhD thesis, Endogene Dynamik - RWTH Aachen, 2004.
- M. Holland, J. Urai, and S. Martel. The internal structure of fault zones in basaltic sequences. *Earth and Planetary Science Letters*, 248(1-2):301, 2006.
- J.M. Hollerbach, W.B. Thompson, and P. Shirley. The convergence of robotics, vision, and computer graphics for user interaction. *International Journal of Robotics Research*, 18, 1999.
- S.R. Idelsohn and E. Oate. To mesh or not to mesh. that is the question. *Computer Methods in Applied Mechanics and Engineering*, 195(37-40):4681–4696, 2006.
- J. C. Jaeger, N. G. W. Cook, and R. W. Zimmerman. *Fundamentals of Rock Mechanics*. Blackwell Publishing, 2007.
- A.W. Jenike. *Gravity flow of bulk solids. Bulletin no. 108*. PhD thesis, Endogene Dynamik - RWTH Aachen, 1961.
- P. Jia and C.A. Tang. Numerical study on failure mechanism of tunnel in jointed rock mass. *Tunnelling and underground Space Technology*, 23(5):500–507, 2008.
- L. Jing. Formulation of discontinuous deformation analysis (dda)an implicit discrete element model for block systems. *Engineering Geology*, 49(3-4):371–381, 1998.
- M. Jirasek. Nonlocal models for damage and fracture: comparison of approaches. *International Journal of Solids and Structures*, 35:31–32, 1998.
- M. Jirasek and T. Zimmermann. Embedded crack model: I. basic formulation. *International Journal for Numerical Methods in Engineering*, 50(6):1269–1290, 2001.
- Z. P. Jirasek and T. Zimmermann. Analysis of the rotating crack model. *Journal of Engineering Mechanics*, 124:277–284, 1998a.
- Z. P. Jirasek and T. Zimmermann. Rotating crack model with transition to scalar damage. *Journal of Engineering Mechanics*, 124:277–284, 1998b.
- Y. Kaneko, T. Shiojima, and M. Horio. DEM simulation of fluidized beds for gas-phase olefin polymerization. *Chemical Engineering Science*, 54(24):5809–5821, 1999.
- M.F. Kevin. The effect of particle dimensionality on granular friction in laboratory shear zones. *Geophysical Research Letters*, 29(19):22–1–22–4, 2002.
- M. Khanal and M. Powell. Energy representation for the unified comminution model. *Particulate Science and Technology: An International Journal*, 28(6):529–538, 2010.

- M. Khanal, W. Schubert, and J. Tomas. Discrete element method simulation of bed comminution. *Minerals Engineering*, 20:179–187, 2007.
- A. Khoei and M. Nikbakht. Contact friction modeling with the extended finite element method (X-FEM). *Journal of Materials Processing Technology*, 177(1-3):58–62, 2006.
- R. Klymowsky, N. Petzelt, J. Knecht, and E. Burchardt. Selection and sizing of high pressure grinding rolls. In: *Mular, A.L., Halbe, D.N., Barratt, D.J. (Eds.), Mineral Processing Plant Design Practice and Control, vol. 1. Soc. for Mining, Metallurgy and Exploration, Inc., Littleton, Colorado., 2002.*
- K. Kolk, W. Weber, and G. Kuhn. Investigation of 3D crack propagation problems via fast bem formulations. *Computational mechanics*, 37(1):32–40, 2005.
- H.K. Kutter and C. Fairhurst. On the fracture process in blasting. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 8(3):181–188, 1971.
- R. Kvapil. *Sublevel caving*. In: *SME mining engineering handbook*. Littleton, CO: Soc Min Metall Explor, 1992.
- R. Kvapil, L. Baeza, J. Rosenthal, and G. Flores. Block caving at el teniente mine, chiles. *Trans Inst Min Metall A*, 98(43-56):147–157, 1989.
- J.P. Latham, A. Munjiza, J. Mindel, J.S. Xiang, R. Guises, X. Garcia, C. Pain, G. Gorman, and M. Piggott. Modelling of massive particulates for breakwater engineering using coupled FEMDEM and CFD. *Particuology*, 6(6):572–583, 2008a.
- L.P. Latham, A. Munjiza, X. Garcia, J.S. Xiang, and R. Guisesa. Three-dimensional particle shape acquisition and use of shape library for DEM and FEM/DEM simulation. *Minerals Engineering*, 21:797–805, 2008b.
- D. Laubscher. *Block caving manual. Report for the international caving study*. Brisbane: JKMRC and Itasca Consulting Group, 2000.
- D.H. Laubscher. Cave mining-the state of the art. *Journal of the South African Institute of Mining and Metallurgy*, 2:279–293, 1994.
- T.A. Laursen and V. Chawla. Design of energy conserving algorithms for frictionless dynamic contact problems. *International Journal for Numerical Methods in Engineering*, 40(5):863–886, 1997.
- B. Lecampion. An extended finite element method for hydraulic fracture problems. *International Journal for Numerical Methods in Engineering*, 25:121–133, 2009.
- D.M. Lee. *The angles of frictions of granular fills*. PhD thesis, University of Cambridge, 1992.

- K.J. Lee and I. Farhoomand. Compressibility and crushing of granular soils in isotropic triaxial compression. *Canadian Geotechnical Journal*, 4(1):68–86, 1997.
- K.J. Lee and H. Seed. Drained strength characteristics of sands. *Journal of the soil and mechanics and foundations division, ASCE*, 93(SM6):117–141, 1967.
- J.F. Lessard and J. Hadjigeorgiou. Ore pass database: Quebec underground metal mines. In *Canadian Institute of Mining Bulletin*, March/April 2006.
- J.F. Lessard and J. Hadjigeorgiou. Numerical investigations of ore pass hang-up phenomena. *International Journal of Rock Mechanics and Mining Sciences*, 44(6):820–834, 2007.
- M.C. Lin. *Efficient Collision Detection for Animation and Robotics*. PhD thesis, University of California, Berkeley, 1993.
- M.C. Lin and J.F. Canny. A fast algorithm for incremental distance calculation. In *Proceeding of the IEEE International Conference on Robotics and Automation, IEEE*, Sacramento, CA, USA, April 1991.
- A. Lisjak and G. Grasselli. Rock impact modelling using fem/dem. In *Proc., 5th Int. Conf. on Discrete Element Methods, School of Engineering and Materials Science*, Queen Mary Univ. of London, London, 2010.
- A. Lisjak, M. Spadari, A. Giacomini, and G. Grasselli. Rock fall numerical modelling using a combined finite-discrete element approach. In *Proc., Rock Slope Stability Symp.*, 2010.
- G.R. Liu and M.B. Liu. *Smooth particle hydrodynamics: a meshfree particle method*. Singapore: World Scientific, 2003.
- G.R. Liu and Z.H. Tu. An adaptive procedure based on background cells for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 191(17-18):1923–1943, 2002.
- H.Y. Liu, S.Q. Kou, and P.A. Lindqvist. Numerical studies on the inter-particle breakage of a confined particle assembly in rock crushing. *Mechanics of Materials*, 37:935–954, 2005.
- W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995.
- L. B. Lucy. A numerical approach to the testing of the fission thesis. *Astronomical Journal*, 82: 1013–1024, 1977.
- S.A. Magnier and F.V. Donz. Numerical simulations of impacts using a discrete element method. *Mechanics of Cohesive-frictional Materials*, 3(3):257–276, 1999.
- O.K. Mahabadi and G. Grasselli. Implementation of a rock joint shear strength criterion inside a combined finite-discrete element method (fem/dem) code. In *Proc., 5th Int. Conf. on*

- Discrete Element Methods*, School of Engineering and Materials Science, Queen Mary Univ. of London, London, 2010.
- O.K. Mahabadi, G. Grasselli, and A. Munjiza. Numerical modelling of a brazilian disc test of layered rocks using the combined finite-discrete element method. In *Rock Engineering in Difficult Conditions; Proc., 3rd Canada-US Rock Mechanics Symp., M. Diederichs and G. Grasselli, eds., Canadian Rock Mechanics Association., Toronto, Canada, 2009.*
- O.K. Mahabadi, A. Lisjak, A. Munjiza, and G. Grasselli. Y-geo: New combined finite-discrete element numerical code for geomechanical applications. *International Journal of Geomechanics*, 12(6):676–688, 2012.
- K. Mair, K.M. Frye, and C. Marone. Influence of grain characteristics on the friction of granular shear zones. 107(B10):ECV 4–1 – ECV 4–9, 2002.
- L.G. Margolin. Elastic moduli of a cracked body. *International Journal of Fracture*, 22(1): 65–79, 1983.
- R.J. Marsal. Large scale testing of rockfill materials. *Journal of Soil Mechanics and Foundations Division ASCE*, 93(2):27–43, 1967.
- J.J. McCarthy, V. Jasti, M. Marinack, and C.F. Higgs. Quantitative validation of the discrete element method using an annular shear cell. *Powder Technology*, 203:70–77, 2010.
- G. McDowell and A. Humphries. Yielding of granular materials. *Granular matter*, 4:1–8, 2002.
- R.L. McNearny and J.F. Abel. Large-scale two-dimensional block caving model tests. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 30(2): 93–109, 1993.
- F. Melo, F. Vivanco, and C. Fuentes. Calculated isolated extracted and movement zones compared to scaled models for block caving. *International Journal of Rock Mechanics and Mining Sciences*, 46(4):731–737, 2009.
- G. Mesri and B. Vardhanabhutib. Compression of granular materials. *Canadian Geotechnical Journal*, 46(4):369–392, 2009.
- Y. Mi and M.H. Aliabadi. Three-dimensional crack growth simulation using bem. *Computers and Structures*, 52(5):871–878, 1994.
- B. Mirtich and J. Canny. Impulse-based dynamic simulation. *Workshop on Algorithmic Foundations of Robotics*, K. Goldberg, P. Halperin, J. C. Latombe and R. Wilson, ed., A.K. Peters, Boston, MA., 1994.
- B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. *I3D '95 Proceedings of the 1995 symposium on Interactive 3D graphics*, 1995.

- B.V. Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California at Berkeley, 1996.
- B.K. Mishra. A review of computer simulation of tumbling mills by the discrete element method: Part II: Practical applications. *International Journal of Mineral Processing*, 71(1-4):95–112, 2003.
- N. Moes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.
- J. K. Morgan and M. S. Boettcher. Numerical simulations of granular shear zones using the distinct element method: II effects of particle size distribution and interparticle friction on mechanical behavior. *Journal of Geophysical Research: Solid Earth*, 104:2721–2732, 1999.
- J.P. Morri, L.A. Glenn, and S.C. Blair. The distinct element method-application to structures in jointed rock. In: *Lecture notes in computational science and engineering: mesh-free methods*, 26:291–306, 2002.
- J.P. Morris, M.B. Rubinb, G.I. Blocka, and M.P. Bonner. Simulations of fracture and fragmentation of geologic materials using combined FEM/DEM analysis. *International Journal of Impact Engineering*, 33(1/2):463–473, 2006.
- P.J. Mosterman. On the normal component of centralized frictionless collision sequences. *Journal of applied mechanics*, 2007.
- N.K. Mukhopadhyay, Maiti S.K., and A. Kakodkar. BEM based evaluation of SIFs using modified crack closure integral technique under remote and/or crack edge loading. *Engineering Fracture Mechanics*, 61(5-6):656–671, 1998.
- N.K. Mukhopadhyay, S.K. Maiti, and A. Kakodkar. A review of sif evaluation and modelling of singularities in bem. *Computational Mechanics*, 25(4):358–375, 2000.
- A. Munjiza. *The combined finite-discrete element method*. John Wiley and Sons, Ltd, 2004.
- A. Munjiza and K.R.F. Andrews. Penalty function method for combined finite-discrete element systems comprising large number of separate bodies. *International Journal for Numerical Methods in Engineering*, 49:1377–1396, 2000.
- A. Munjiza and N. John. Mesh size sensitivity of the combined FEM/DEM fracture and fragmentation algorithms. *Engineering Fracture Mechanics*, 69(2):281–295, 2002.
- A. Munjiza and J.P. Latham. Comparison of experimental and FEM/DEM results for gravitational deposition of identical cubes. *Engineering Computations*, 21(2-3):249–264, 2004.
- A. Munjiza, D.R.J. Owen, and N. Bicanic. A combined finite-discrete element method in transient dynamics of fracturing solids. *Engineering Computations*, 12(2):145–174, 1995.

- A. Munjiza, K.R.F. Andrews, and J.K. White. Combined single and smeared crack model in combined finite-discrete element analysis. *International Journal for Numerical Methods in Engineering*, 44(1):41–57, 1999.
- A. Munjiza, J. Xiang, X. Garcia, J. Latham, G. D’Albano, and N. John. The virtual geoscience workbench, vgw: Open source tools. *Particuology*, 8(2):100–105, 2010.
- A. Munjiza, E. Knight, and E. Rougier. *Computational Mechanics of Discontinua*. Wiley, 2011.
- H. Nakashima, Y. Shioji, T. Kobayashi, S. Aoki, H. Shimizu, J. Miyasaka, and K. Ohdoi. Determining the angle of repose of sand under low-gravity conditions using discrete element method. *Journal of Terramechanics*, 48(1):17–26, 2011.
- Y. Nakata, M. Hyodo, A.F.L. Hyde and Y. Kato, and H. Murata. Microscopic particle crushing and sand subjected to one-dimensional compression. *Soils and Foundations*, 41(1):69–82, 2001.
- T.J. Napier-Munn, S. Morrel, R.D. Morrison, and T. Kojovic. Mineral comminution circuits: their operation and optimization. *Julius Kruttschnitt Mineral Research Centre*, 1996.
- H. Nazeri, G. Mustoe, T.G. Rozgonyi, and C.J. Wienecke. Implementation of a discrete element methodology for the modeling of gravity flow of ore in ore passes. In *Hammah, et al., editors. NARMS-TAC 2002.*, pages 1307–12, University of Toronto, 2002.
- S. Ngidi and D. D. Pretorius. A fast algorithm for incremental distance calculation. In *IMPACT OF POOR FRAGMENTATION ON CAVE MANAGEMENT*, pages 111–122, Sacramento, CA , USA, 2011.
- L. Noels, L. Stainier, and J.P. Ponthot. An energy momentum conserving algorithm using the variational formulation of visco-plastic updates. *International Journal for Numerical Methods in Engineering*, 65(6):904–942, 2006.
- A. Okabe, B. Boots, K. Sugihara, and S. Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, Inc., New York, NY, 2000.
- E. Onate and J. Rojek. Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 193(27-29):3087–3128, 2004.
- K. Oraee, M.H. Basiri, M.H. Sajidan, and M. Hayati. Risk assessment in block cave mining. In *Dearborn, MI: SME. 2011 SME Annual Meeting and Exhibit and CMA 113th National Western Mining Conference, Shaping a Strong Future Through Mining*, Denver, Colorado, USA, 2011.

- A. Paluszny and S. K. Matthai. Numerical modeling of discrete multi-crack growth applied to pattern formation in geological brittle media. *International Journal of Solids and Structures*, 46:3383–3397, 2009.
- A. Paluszny and R. W. Zimmerman. Numerical simulation of multiple 3D fracture propagation using arbitrary meshes. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12): 953–966, 2011.
- A. Paluszny and R.W. Zimmerman. Numerical fracture growth modeling using smooth surface geometric deformation. *Engineering Fracture Mechanics*, 108:19–36, 2013.
- A. Paluszny, X.H. Tang, and R.W. Zimmerman. A multi-modal approach to 3D Fracture and Fragmentation of Rock using Impulse-Based Dynamics and the Finite Element Method. In *The 46th US Rock Mechanics/Geomechanics Symposium*, pages 24–27, Chicago, IL, USA, June 2012.
- A. Paluszny, X.H. Tang, and R.W. Zimmerman. A fracture- and impulse-based FDEM approach for fragmentation. *Computational Mechanics*, 52(5):1071–1084, 2013.
- M.E. Pierce. *A model for gravity flow of fragmented rock in caving mines*. PhD thesis, The University of Queensland, 2010.
- M.S. Powell and A.T. McBride. What is required from DEM simulations to model breakage in mills? *Minerals Engineering*, 19(10):1013–1021, 2006.
- M.S. Powell, I. Govender, and A.T. McBride. Applying DEM outputs to the unified comminution model. *Minerals Engineering*, 21:744–750, 2008.
- G. Power. *Modelling granular flow in caving mines: large scale physical models and full scale experiments*. PhD thesis, University of Queensland, 2004.
- T. Rabczuk and T. Belytschko. Adaptivity for structured mesh-free particle methods in 2D and 3D. *International Journal for Numerical Methods in Engineering*, 63 (11):1559–1582, 2005.
- T. Rabczuk and T. Belytschko. A three-dimensional large deformation mesh-free method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*, 196: 2777–2799, 2007.
- T. Rabczuk, T. Belytschko, and S.P. Xiao. Stable particle methods based on lagrangian kernels. *Computer Methods in Applied Mechanics and Engineering*, 193(26):1035–1063, 2004.
- T. Rabczuk, P.M.A. Areias, and T. Belytschko. A simplified mesh-free method for shear bands with cohesive surfaces. *International Journal for Numerical Methods in Engineering*, 69: 993–1021, 2007a.

- T. Rabczuk, S. Bordas, and G. Zi. A three-dimensional meshfree method for continuous multiple-crack initiation, propagation and junction in statics and dynamics. *Comput Mech*, 40:473–495, 2007b.
- B. Ren and S. Li. Meshfree simulation of plugging failure in high-speed impacts. *Computers and Structures*, 88:909–923, 2010.
- B. Ren, S. Li, J. Qian, and X. Zeng. Meshfree simulations of spall fracture. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):797–811, 2011.
- C. Renshaw and D. Pollard. Numerical simulation of fracture set formation: a fracture mechanics model consistent with experimental observations. *Journal of Geophysical Research*, 99:9359–9372, 1994.
- D. Reznik, S. Brown, and John Canny. Dynamic simulation as a design tool for a microactuator array. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1675–1680, 1997.
- J. R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.
- J. Rogers, B. Hamer, A. Brampton, S. Challinor, M. Glennerster, P. Brenton, and A. Bradbury. Beach management manua (second edition). Technical report, Robotics Institute, Carnegie Mellon University, 1996.
- C.H. Rycroft. Voro++: A three-dimensional voronoi cell library in C++, lawrence berkeley national laboratory. *Chaos*, pages LBNL–1430E, 2009.
- B.A. Sainsbury. *A model for cave propagation and subsidence assessment in jointed rock masses*. PhD thesis, The University of New South Wales, 2012.
- A. Schmitt, J. Bender, and H. Prautzsch. On the convergence and correctness of impulse-based dynamic simulation. In *Internal Report 17, Institut fr Betriebs-und Dialogsysteme*, 2005.
- G.H. Shi and R.E. Goodman. Discontinuous deformation analysis. *The 25th U.S. Symposium on Rock Mechanics (USRMS)*, pages 84–0269, 1984.
- T.T. Shih. An evaluation of the probabilistic approach to brittle design. *Engineering Fracture Mechanics*, 13(2):257–271, 1980.
- P.H. Shipway and I.M. Hutchings. Philosophical magazine. A. *Fracture of brittle sphere under compression and impact loading: II. Results for lead-glass and sapphire spheres*, 67(6):1389–1404, 1993.
- T. Stacey and A. Swart. Investigation into drawpoints, tips orepasses and chutes. report to the safety in mines research advisory committee. In *Steffen, Robertson and Kirsten*, volume 1, page 112, 1997.

- E. Sternberg and F. Rosenthal. The elastic sphere under concentrated loads. *Journal of Applied Mechanics*, 19:413–421, 1952.
- K Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 281-309:128, 2001.
- Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. Musculotendon simulation for hand animation. *ACM Trans. Graph.*, 27(3), 2008.
- N. Sukumar, N. Moes, B. Moran, and T. Belytschko. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering*, 48:1549–1570, 2000.
- D. Sulsky and A. Kaul. Implicit dynamics in the material-point method. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1137–1170, 2004.
- A. Susaeta. Theory of gravity flow (part 1). In *In: Proc MassMin*, pages 167–72, Santiago, Chile, 2004.
- K. Tanaka, M. Nishida, T. Kunimochi, and T. Takagi. Discrete element simulation and experiment for dynamic response of two-dimensional granular matter to the impact of a spherical projectile. *Powder Technology*, 124(1-2):160–173, 2002.
- C.A. Tang, W.T. Yang, Y.F. Fu, and X.H. Xu. A new approach to numerical method of modelling geological processes and rock engineering problems continuum to discontinuum and linearity to nonlinearity. *Engineering Geology*, 49(3/4):207–214, 1998.
- X.H. Tang, A. Paluszny, and R.W. Zimmerman. Impulse-based simulation of particle flow during subsurface block-cave mining. In *XXIII ICTAM, 19-24*, Beijing, China, August 2012a.
- X.H. Tang, A. Paluszny, and R.W. Zimmerman. An impulse-based energy tracking method for collision resolution. *Computer Methods in Applied Mechanics and Engineering*, under review, 2012b.
- X.H. Tang, A. Paluszny, and R.W. Zimmerman. Energy conservative property of impulse-based methods for collision resolution. *International Journal for Numerical Methods in Engineering*, 95(6):529–540, 2013.
- B. Tatone, A. Lisjak, O. Mahabadi, G. Grasselli, and C. Donnelly. A preliminary evaluation of the combined finite-discrete element method as a tool to assess gravity dam stability. In *Proc., Canadian Dam Association Annual Conf., Canadian Dam Association*, Niagara Falls, Ontario, Canada, 2010.
- H.-J. Tillemans and H. J. Herrmann. Simulating deformations of granular solids under shear. *Physica A*, 217:261–288, 1995.

- F. Tonon. Explicit Exact Formulas for the 3-D Tetrahedron Inertia Tensor in Terms of its Vertex Coordinates. *Journal of Mathematics and Statistics*, 1(1):8–11, 2004.
- D. Tromans. Mineral comminution: energy efficiency considerations. *Minerals Engineering*, 21(8):613–620, 2008.
- R. Trueman, M. Pierce, and R. Wattimena. Quantifying stresses and support requirements in the undercut and production level drifts of block and panel caving mines. *International Journal of Rock Mechanics and Mining Sciences*, 39(5):617–632, 2002.
- M. Tsesarsky, Y.H. Hatzor, and N. Sitar. Dynamic displacement of a block on an inclined plane: Analytical, experimental and dda results. *Rock Mechanics and Rock Engineering*, 38(2):153–167, 2005.
- O. Tsoungui, D. Vallet, J.C. Charmet, and S. Roux. Size effects in single grain fragmentation. *Granular Matter*, 2:19–27, 1999.
- A. Vyazmensky. *Numerical modelling of surface subsidence associated with block cave mining using a finite element/discrete element approach*. PhD thesis, Fraser University, 2008.
- A. Vyazmensky, D. Elmo, and D. Stead. Combined finite-discrete element modelling of surface subsidence associated with block caving mining. In *Proceedings of the 1st Canada-US Rock Mechanics Symposium*, pages 467–475, Vancouver, Canada, 27-31 May 2007 2007.
- J. Walkiewicz, D. Lindrothand, and A. Clark. Grindability of taconite rod mill feed enhanced by microwave induced cracking. In *Society of Mining Engineering Annual Meeting, 1-4*, 1993.
- J. Wang, H.S. Yu, P. Langston, and F. Fraige. Particle shape effects in discrete element modelling of cohesive angular particles. *Granular Matter*, 13:1–12, 2011.
- Y. Wang and F. Tonon. Calibration of a discrete element model for intact rock up to its peak strength. *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(5):447–469, 2010.
- Y. Wang and F. Tonon. Discrete element modeling of rock fragmentation upon impact in rock fall analysis. *Rock Mechanics and Rock Engineering*, 44:23–35, 2011.
- Y. N. Wang. *Three-dimensional rock-fall analysis with impact fragmentation and fly-rock modelling*. PhD thesis, University of Texas at Austin, 2009.
- P. A. Wawrzynek and A. R. Ingraffea. An interactive approach to local remeshing around a propagation crack. *Finite Elements in Analysis and Design*, 5:87–96, 1989.
- R. Weinstein, E. Guendelman, and R. Fedkiw. Impulse-based control of joints and muscles. *IEEE Transactions on Visualization and Computer Graphics*, 14(1), 2008.

- A.J. Wilde and M.H. Aliabadi. A 3-D dual bem formulation for the analysis of crack growth. *Computational Mechanics*, 23(3):250–257, 1999.
- J.R. Williams and R.O. Connor. Discrete element simulation and the contact problem. *Archives of Computational Methods in Engineering*, 6(4):279–304, 1999.
- F.K. Wittel, J. Schulte-Fischedick, F. Kun, B. Krplina, and M. Frie. Discrete element simulation of transverse cracking during the pyrolysis of carbon fibre reinforced plastics to carbon/carbon composites. *Computational Materials Science*, 28(1):1–15, 2003.
- F.K. Wittel, H.A. Carmona, F. Kun, and H.J. Herrmann. Mechanisms in impact fragmentation. *International Journal of Fracture*, 154 (1-2):105–117, 2008.
- P. Wriggers. Finite element methods for contact problems with friction. *Tribology International*, 29(8):651–658, 1996.
- L.C. Wrobel and M.H. Aliabadi. *The Boundary Element Method: Volumes 1 and 2*. Wiley-Blackwell, 2002.
- S.Z. Wu and K.T. Chau. Dynamic response of an elastic sphere under diametral impacts. *Mechanics of Materials*, 38(11):1039–1060, 2006.
- S.Z. Wu, K.T. Chau, and T.X. Yu. Crushing and fragmentation of brittle spheres under double impact test. *Powder Technology*, 143-144(, PAGES = 41-55.), 2004.
- J.A. Yamamuro and P.V. Lade. Drained sand behaviour in axisymmetric tests at high pressures. *Journal of Geotechnical Engineering, ASCE*, 122(2):109–119, 1996.
- J.A. Yamamuro, P.A. Bopp, and P.V. Lade. One-dimensional compression of sands at high pressures. *Journal of Geotechnical Engineering, ASCE*, 122(2):147–154, 1996.
- Z. Zhao, L. Jing, and I. Neretnieks. Particle mechanics model for the effects of shear on solute retardation coefficient in rock fractures. *International Journal of Rock Mechanics and Mining Sciences*, 52:92–102, 2012.
- Y.C. Zhou, B.H. Xu, A.B. Yu, and P. Zulli. An experimental and numerical study of the angle of repose of coarse spheres. *Powder Technology*, 125(1):45–54, 2002.
- Q. Zhu. On enrichment functions in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 91(2):186–217, 2012.
- O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, Volume 1, The Basis*. Butterworth-Heinemann, fifth edition, 2000.