

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

DEM Modeling of 3-D Kinematics in Rock Slope Failure

Permalink

<https://escholarship.org/uc/item/7f55b56f>

Author

Keissar, Yuval

Publication Date

2024

Peer reviewed|Thesis/dissertation

DEM Modeling of 3-D Kinematics in Rock Slope Failure

By

Yuval Keissar

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Nicholas Sitar, Co-Chair
Professor Michael Gardner, Co-Chair
Professor Tarek Zohdi
Professor Dimitrios Zekkos

Fall 2024

DEM Modeling of 3-D Kinematics in Rock Slope Failure

Copyright 2024
by
Yuval Keissar

Abstract

DEM Modeling of 3-D Kinematics in Rock Slope Failure

by

Yuval Keissar

Doctor of Philosophy in Engineering - Civil and Environmental Engineering

University of California, Berkeley

Professor Nicholas Sitar, Co-Chair

Professor Michael Gardner, Co-Chair

In geotechnical engineering and rock mechanics, capturing the kinematic response of a particulate medium to loading is fundamental to modeling naturally deposited sediments, tunneling, and rock slope stability, among others. Fundamentally, in discontinuous media such as jointed rock masses, the displacement is governed by pre-existing discontinuities (bedding planes, faults, and joints) and continuum-based methods generally are not able to describe the highly localized nature of displacements along these structural features. Particulate-based numerical models, such as the Discrete Element Method (DEM) and Discontinuous Deformation Analysis (DDA), have been developed to model particulate systems, including fractured rock masses. These approaches are able to directly model the shape of individual particles within a particulate system, and can intrinsically accommodate large displacements associated with discontinuities at particle and block margins.

In this research, we first present a formulation of a new method to calculate the contact point between polyhedral particles that can be used to complement different particulate methods. Contact detection and contact geometry are among the most important steps in DEM and DDA simulations. Accurately representing the contact between two particles is crucial, and when modeling fractured rock using polyhedral particles, the accuracy of the contact point calculation is essential for obtaining realistic and reliable simulation results. We present a new algorithm for accurately calculating the contact point between two colliding polyhedral particles that uses the topology of the particles to assert their interaction with the plane of contact and not directly with each other. The new algorithm provides improved performance in terms of global stability of DEM models by mitigating numerically induced instability associated with errors and sporadic movement in the contact point calculation. We then present a comparative study of contact detection algorithms routinely used for simulating convex polyhedra in particulate methods. We critically assess, in terms of accuracy and

computational efficiency, the advantages and limitations of each algorithm tested in the scope of this research. These algorithms are implemented within the same open-source software framework to allow for an objective assessment of their performance. In this case, DEM is used as the particulate solver; however, the characteristics of the selected contact detection algorithms are independent of this choice. The verification examples presented allow for informed decisions on which class of contact detection algorithm may be most appropriate for specific types of particulate simulations involving convex polyhedra. Lastly, we compare solutions obtained using Limit-Equilibrium (LE) and the Discrete Element Method (DEM), both in 2D and 3D, for analyzing rock slope stability and runout. Specifically, we illustrate the importance of kinematics in modeling of rock slopes and describing the progressive nature by which they displace and ultimately fail. While 3D LE methods provide a measure of the factor of safety against failure, the failure surface and mode are assumed, and the rock mass is typically represented by vertical slices (2D) or columns (3D) in the analysis. Thus, the kinematic response of the rock mass is artificially constrained since the natural structure within the rock is largely ignored, the failure mode is assumed a priori, and thus the factor of safety calculated using LE approaches may not necessarily describe the governing response. In contrast, in DEM a specific mode of failure is not assumed, since natural discontinuities, joints, shears, and fractures, as observed in the field can be used to create a more realistic representation of the rock mass. We also show that with an increasing number of rock blocks in the model (tighter spacing of the joints) without changing the intrinsic material properties, the rock mass is less stable. This has implications for rock slope stability evaluations, as rock that is more fractured will be less kinematically constrained and require more mechanical strength to remain stable. Additionally, during rock slide initiation and propagation, the rock within the sliding mass may fracture and disintegrate, such that it becomes less constrained as it deforms. The outcome is a progressive rock slope failure and accelerating displacements as the rock blocks within the sliding mass become more fractured. Thus, we show that in analyzing the response of fractured rock, it is important to not only consider the mechanical properties required for stability, but also the amount of displacement the sliding mass may undergo, as breakup of the rock due to displacement will have a destabilizing effect.

To my daughters, Ella and Gali, and my wife, Zohar.
You are my everything. I love you to the moon and the stars.
And back.

Contents

Contents	ii
List of Figures	iv
List of Tables	vi
1 Introduction	1
2 A New Algorithm for Accurate Contact Point Calculation Between Convex Polyhedral Particles in DEM	3
2.1 Introduction	3
2.2 Contact Point Calculation with EPA	4
2.3 Contact Polygon Intersection Algorithm for Contact Point Calculation	5
2.4 Examples	13
2.5 Conclusion	19
3 Benchmarking Contact Detection Algorithms Used in Polyhedral Particles System	21
3.1 Introduction	21
3.2 Contact Detection Algorithms	22
3.3 Benchmark Tests	23
3.4 Discussion	37
3.5 Conclusion	39
4 Capturing 3D Kinematics in Rock Slope Failure	41
4.1 Introduction	41
4.2 Numerical Methods	42
4.3 Case Study Site Description	42
4.4 2D Planar Failure Analysis	44
4.5 3D LE and DEM Comparison	47
4.6 3D DEM analysis of Fractured Rock Mass	50
4.7 Conclusion	53

5 Conclusion and Future Research	55
5.1 Future Research	56
Bibliography	57
A 3D edges intersection	62

List of Figures

2.1	Illustration of EPA contact point calculation.	6
2.2	Block resting on a boundary block. The contact point is marked with spheres colored by progression in time.	7
2.3	The common plane between the particles and the distance sign for the vertices that pass it (after Nezami et al. (2004)).	8
2.4	Intersection of the particles with the plane of contact.	9
2.5	Point in or out a convex polygon.	10
2.6	Coincident edges cases.	11
2.7	Edges intersection cases.	12
2.8	Intersection of particles demonstration.	14
2.9	Displacement of block centroid (left) and angle of rotation around a vertical axis (right) for a block sitting on a horizontal plane. GJK-EPA algorithm shows non-physical rotation due to contact point error while intersecting polygon method remains stationary and eliminates spurious torque.	15
2.10	Block drops on a horizontal surface, modeled using GJK-EPA (left) and the IPM (right, orange).	16
2.11	A jointed wedge on an inclined slope.	18
2.12	3D model showing the angular velocity of the blocks.	19
3.1	Polyhedral particles contact types.	25
3.2	Parallel edge-edge contact point as calculated using the different algorithms.	30
3.3	Edge-Face contact contact point as computed using the different contact detection algorithms.	33
3.4	Face-Face contact contact point. Contact points calculated using LP are numbered according to the timestep.	34
3.5	25 cubic particles rest on a horizontal plane.	35
3.6	Face-Face contact with multiple particles when using the GJK-EPA contact algorithm: Plan view of the particles and horizontal forces acting on the particles.	36
3.7	Hexagonal rolling down the slope. Orange points mark the trajectory of the center of mass of the particle.	37
3.8	Trajectory of the centroid of the hexagonal rod as it slides down the slope.	38
4.1	Round Hill open-pit mine failure	43

4.2	Schematic model for Round Hill open-pit mine in New Zealand (Brown et al., 2016). The grid represents the columns used in the LE analysis. The planes are faults bounding the sliding mass.	44
4.3	Cross section used in the 2D LE analysis.	45
4.4	Inter-slice friction angle influence on the stability of the 2D model.	47
4.5	Kinematic analysis of quasi-2D DEM model.	48
4.6	3D LE model.	48
4.7	3D DEM model with columns representing the sliding mass.	49
4.8	Number of particles for F.S. \approx 1.	50
4.9	Average displacement of columnar blocks.	51
4.10	3D DEM model of jointed rock mass.	52
4.11	Average displacement of the rock mass for different friction angles.	53

List of Tables

2.1	Mechanical and numerical properties for a jointed wedge on an inclined slope. . .	17
2.2	Mechanical and numerical properties for the cut slope.	18
3.1	Contact features as calculated using different contact detection algorithms for vertex-vertex collision (ε is a number smaller than machine precision 1^{-16}). . . .	26
3.2	Contact features as calculated using different contact detection algorithms for vertex-edge collision.	27
3.3	Contact features as calculated using different contact detection algorithms for vertex-face collision.	28
3.4	Contact features as calculated using different contact detection algorithms for parallel edge-edge collision.	29
3.5	Contact features as calculated using different contact detection algorithms for crossing edge-edge collision.	31
3.6	Contact features as calculated using different contact detection algorithms for edge-face collision.	32
3.7	Face-Face execution times.	34
3.8	Rolling hexagonal rod execution times.	38

Acknowledgments

First, I would like to thank my Ph.D. advisors, Professor Nicholas Sitar and Professor Michael Gardner. I thank Professor Sitar for the opportunity to pursue my dream of studying in such a wonderful institution. His guidance and support were instrumental in the completion of this dissertation. His valuable insights, creative ideas, and wisdom pushed me to develop and refine my research in ways I never thought possible. I will carry his lessons for the rest of my life. More importantly, his support beyond academic life during difficult times is greatly appreciated. I will never forget that.

Professor Gardner, Mike, introduced me to the world of real coding and open-source software. In our first meeting, I could barely understand anything he was saying, but slowly and steadily, he taught me everything I know about computing and coding. His guidance and patience helped me develop to places that seemed unimaginable to me only a few years ago. I genuinely enjoyed our meetings, in person or on Zoom, and I am hopeful this is only the beginning of a fruitful collaboration.

I greatly appreciate the members of my Qualifying and Thesis committees, Professor Dimitrios Zekkos, Professor Tarek Zohdi, and Professor Per-Olof Persson. I learned a lot in your classes, and your advice on numerical simulations and DEM improved my research greatly.

During my years in Berkeley, I had the privilege of learning from the best faculty in their fields: Professor Robert Kayen, Professor Kenichi Soga, Professor Jonathan Bray, Professor Adda Athanasopoulos Zekkos, Professor Sanjay Govindjee, Professor James Demmel, and Professor Katherine Yelick.

I also want to thank my research group colleagues and friends, Peng Tan and Hasitha Sithadara Wijesuriya, for our discussions and help, and my friends in Berkeley for wonderful times.

This research was supported in part by funding from the Edward G. and John R. Cahill Chair and the Jane Lewis Fellowship. I am grateful for making this research possible. Tagasoft, Wellington, NZ, generously provided an academic license for TSLope, LEM slope stability code.

Finally, I would like to thank my family. To my mom and dad, who always pushed me to pursue my dreams and supported me every step of the way. To my awesome siblings, who are simply wonderful. I missed you every second we have been so far from each other. My biggest thank you goes to my amazing wife, Zohar, who has been my steady rock throughout this journey. I am forever grateful for her infinite support and love. Lastly, my two beautiful daughters, Gali and Ella, who have never stopped filling my life with love and joy. I love you so much.

Chapter 1

Introduction

Capturing the kinematic response of a particulate medium to loading is fundamental to many types of problems in geotechnical engineering, including naturally deposited sediments, tunneling, and rock slope stability. Continuum methods, such as the Finite Element Method (FEM), have been traditionally used to evaluate deformations of slopes (e.g., Duncan et al., 1968; Griffiths and Fenton, 2004), boreholes (e.g., Diersch et al., 2011; Gaede et al., 2012), or underground openings (Chheng and Likitlersuang, 2018). However, continuous-based methods are usually limited to small displacements. In addition, in discontinuous media such as jointed rock mass, the displacement is governed by pre-existing discontinuities (bedding planes, faults, and joints). Continuum-based methods cannot attain the discrete nature of particulate mechanics.

Limit Equilibrium (LE) methods were developed to evaluate the potential for failure, mainly in slopes and earth-fill dams (e.g., Duncan, 1996). These types of methods solve the equations for the equilibrium of forces and moments to assess the factor of safety. The simplicity of the formulations and efficiency of the different LE method software packages make them highly attractive in research and practice. While popular and useful, these methods do not consider the internal structure of the modeled material and do not account for displacements. The failure surface must be assumed and is not derived as part of the analysis.

To study large-scale particulate systems, the Material Point Method (MPM) (Sulsky et al., 1994, 1995) was proposed. This method was shown to be effective for large deformation landslides (Soga et al., 2016) by utilizing the Eulerian-Lagrangian approach. In this particle-based method, the material is represented by an assemblage of material points, and Newton's equations of motion are used to calculate deformations. However, this method does not consider the kinematic nature of natural materials, such as rock blocks and the shape of particles in sediments, since it does not account for the shape of individual particles.

To appropriately model the polyhedral shape of geo-materials, discrete element methods began to be developed some 50 years ago. The Distinct Element Method (DEM) (Cundall and Strack, 1979) and the Discontinuous Deformation Analysis (DDA) (Shi, 1988) were designed to capture the interaction between individual polyhedral blocks without a-priori

assumptions on the mode of failure and the failure surface.

In this research, DEM is employed to study the interaction between neighboring particles to obtain the correct kinematic response of particle assemblages. This thesis is organized as follows:

- Chapter 2 describes an issue with the contact point calculation between two colliding particles in DEM, which has not been well handled in some DEM formulations. Specifically, an accurate and numerically more stable simulation computation of the contact point location is introduced.
- In Chapter 3, contact point detection as such is an important aspect of being able to obtain a kinematically accurate solution. To this end, commonly used contact detection algorithms used in DEM are compared, and the advantages and limitations of each algorithm are presented.
- In Chapter 4 full-scale simulations of jointed rock slope stability are explored to evaluate the influence of kinematics failure initiation and failure progression. The focus is on the limitations of limit equilibrium slope stability analyses in analyzing jointed rock masses and the importance of modeling the blocky nature of the rock is demonstrated.
- Chapter 5 provides a summary and conclusion of the research presented in this thesis. Key findings are highlighted with a focus on recommendations for future research.

Chapter 2

A New Algorithm for Accurate Contact Point Calculation Between Convex Polyhedral Particles in DEM

2.1 Introduction

In fractured rock masses, slope failure occurs along pre-existing discontinuities, such as joints and fractures. Therefore, unlike in soil modeling, the shape of the particles is inherently polyhedral. The Discrete Element Method (DEM) (Cundall and Strack, 1979) and Discontinuous Deformation Analysis (DDA) (Shi, 1988) have been applied to capture the fractured nature of rock masses and to simulate the interaction between discrete rock blocks. For particle-based methods such as DEM and DDA, the most computationally expensive step in the analysis is the determination of contacts between the particles and the computation of the resulting forces, accounting for 70-80% of the total simulation time (Horner et al., 2001).

The contact detection is a two-step procedure: 1) neighbor search and 2) contact resolution. In the neighbor search step, the simulation domain is subdivided to avoid checking every particle against every other particle for contact (a problem that would scale ($\mathcal{O}(N^2)$). Generally, either binning ($\mathcal{O}(N)$) (Munjiza and Andrews, 1998; Williams et al., 2004; Krjgsman et al., 2014) or hierarchical type algorithms ($\mathcal{O}(N \log(N))$) (O'Connor, 1996; Feng and Owen, 2002; Perkins and Williams, 2001) are used for this phase of the computations. In these approaches, only a subset of particles are checked for potential contact based on their proximity. The output from this phase is a list of particle pairs that should be checked further for potential contact.

The contact resolution step is where the physical shape of the particles is considered, i.e. if the particles are in contact, the contact geometry is determined. Many algorithms for contact resolution for polyhedral blocks have been developed in recent years. Among the earliest and most commonly used is the "Common-Plane" approach proposed by Cundall (1988). In this approach, the space between two blocks is bisected by a plane, and the distance is

then calculated from each particle to the plane and not to each other. This algorithm was later augmented by the Fast Common Plane (Nezami et al., 2004) and the Shortest Link Method (G. Nezami et al., 2006). These algorithms are prone to numerical instability when the contact type changes rapidly (Feng et al., 2012). Other types of contact resolution were developed to overcome such issues, such as algorithms based on energy conservation (e.g. Feng and Owen, 2004; Feng et al., 2012), direct search methods (Zhang et al., 2015, 2016; Zheng et al., 2017b,a), and convex optimization (Boon et al., 2012).

A commonly used algorithm, both due to its speed and simplicity, is the Gilbert-Johnson-Kheerti (GJK) algorithm (Gilbert et al., 1988), in which convex particles are checked for intersection using their Minkowski difference. The basic idea is to construct a simplex closing the origin of the Minkowski difference of two blocks in contact using support points and support functions. This algorithm is efficient and robust for any convex shape but does not provide the contact geometry, i.e., contact point, contact normal, and penetration. The Expanding Polytope Algorithm (EPA) (Van Den Bergen, 2001) was developed to find the complete contact information using the output from the GJK algorithm. EPA takes the final simplex from GJK and iteratively expands it to a polytope by adding and removing faces until the face closest to the origin is also a face on the boundary of the Minkowski difference.

However, in the case of polyhedral particles, the location of the contact point determined from EPA often changes sporadically and does not necessarily provide the correct location of the contact point (Feng and Tan, 2020). In this work, it is shown that errors associated with the contact point calculated using GJK-EPA induce numerical instability that produces non-physical torques within the particle assemblage. These non-physical torques induce translations and rotations that could be misinterpreted as the actual physical movements in the simulations of fractured rock. To mitigate this issue, a new algorithm is presented that uses contact properties found using the GJK algorithm to calculate the contact point using a polygon of intersection. This approach provides a more accurate and stable representation of the contact point and eliminates instability associated with the contact point found using GJK-EPA. The improved performance and stability of this approach are showcased in examples that show both the numerical consequences of the contact point error in the GJK-EPA and how this is mitigated using the polygon of intersection approach. By pairing the GJK contact detection algorithm with a more stable and accurate contact point calculation approach, efficient and reliable DEM simulations of fractured rock mass composed of convex polyhedral particles can be performed.

2.2 Contact Point Calculation with EPA

In the EPA, the contact point is calculated in barycentric coordinates. The error in the contact point calculation for convex polyhedral particles is a function of how the point on the expanded polytope closest to the origin is represented in barycentric coordinates. This can be illustrated in a simple 2D example: Block A is resting on Block B, as shown in Fig. 2.1. The penetration between the two blocks is exaggerated for visualization purposes (Fig.

2.1a). Under gravity loading, the blocks are obviously in face-to-face contact. The point c on the face f on the Minkowski difference closest to the origin is $[0, -0.3]$. The direction from the origin to point c gives the contact normal, $[0, -1]$ (Fig. 2.1b). The barycentric coordinates of point c can be calculated using the coordinates of that point and the points on the Minkowski difference face closest to the origin (V_f^i):

$$c = \xi_1 V_f^1 + \xi_2 V_f^2 \quad (2.1)$$

Which gives the barycentric coordinates of point c as $[\xi_1, \xi_2] = [0.4, 0.6]$. The contact points are then calculated using the associated vertices on the boundary of the two blocks $V_A^i \in \partial A$ and $V_B^i \in \partial B$. Using the barycentric coordinates, the contact points are computed as:

$$c_{A,B} = \xi_1 V_{A,B}^1 + \xi_2 V_{A,B}^2 \quad (2.2)$$

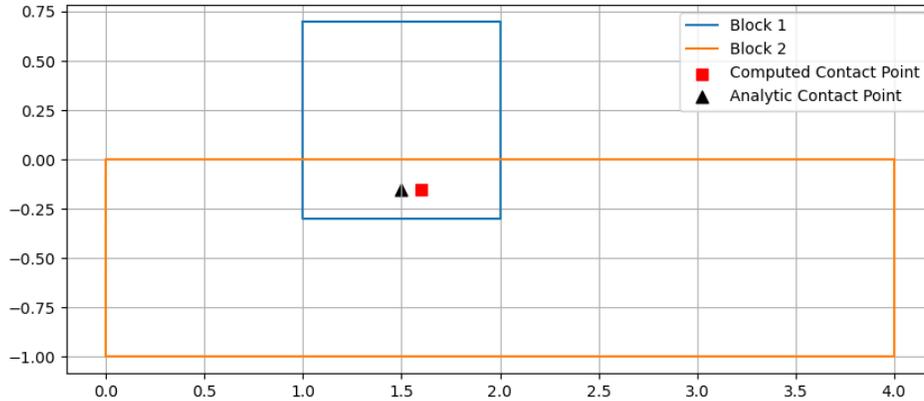
Which gives the following contact points: $c_A = [1.6, -0.3]$ and $c_B = [1.6, 0]$. The common contact point is $c_p = (c_A + c_B)/2 = [1.6, -0.15]$. This computed contact point is slightly off-center with respect to Block A compared with the actual contact point $c_p = [1.5, -0.15]$. This is a difference of 6.7% which becomes significant when moments on the block are computed.

This problem then extends to 3D, as shown in Fig. 2.2. The contact point is off-center with respect to a block resting on a larger boundary block. This generates torque in the block, artificially causing it to rotate. Subsequently, the contact point shifts to the vertices closest to the surface, and the non-physical torque increases in magnitude. This produces several undesirable effects: it induces global instability if the block is in contact with other blocks around it, as will be discussed later (Section 2.4), and it produces spurious moments, which then affect the overall behavior of the block assemblage.

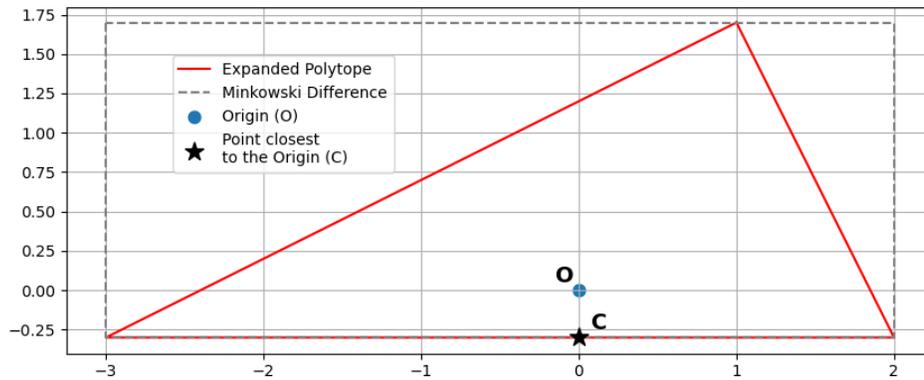
In a fully dynamic problem where polyhedral blocks are colliding, this might not be an issue because faces from neighboring blocks are rarely truly parallel. However, for blocks resting or slowly sliding on big boundary blocks, the contact type are often classified as face-to-face, with two parallel faces from opposing blocks. It is important to note, that this is not a problem when the particles are modeled as spheres since the contact point between spherical particles is established analytically; however, when modeling fractured rock it is necessary to account for the polyhedral block shapes to ensure that the correct kinematic response is captured (Goodman and Kieffer, 2000; Sitar et al., 2005; Gardner and Sitar, 2019).

2.3 Contact Polygon Intersection Algorithm for Contact Point Calculation

To mitigate the issues associated with the contact point calculation in EPA, a new algorithm, the Intersection Polygon Method (IPM), that utilizes a polygon of intersection between two colliding convex polyhedral particles is introduced. The input data needed to compute the contact point is the contact geometry: 1) contact normal, 2) penetration depth, and 3) a



(a) Two colliding blocks with the computed and desired contact points.



(b) EPA contact point calculation process with Minkowski difference and the expanded polytope in 2-D.

Figure 2.1: Illustration of EPA contact point calculation.

point on the plane of contact. This information can be obtained using any contact detection algorithm; however, in this work GJK-EPA algorithm is utilized to provide this initial input and then IPM is used to accurately establish the contact point. In this context, IPM is suitable to be implemented with any other contact detection algorithm.

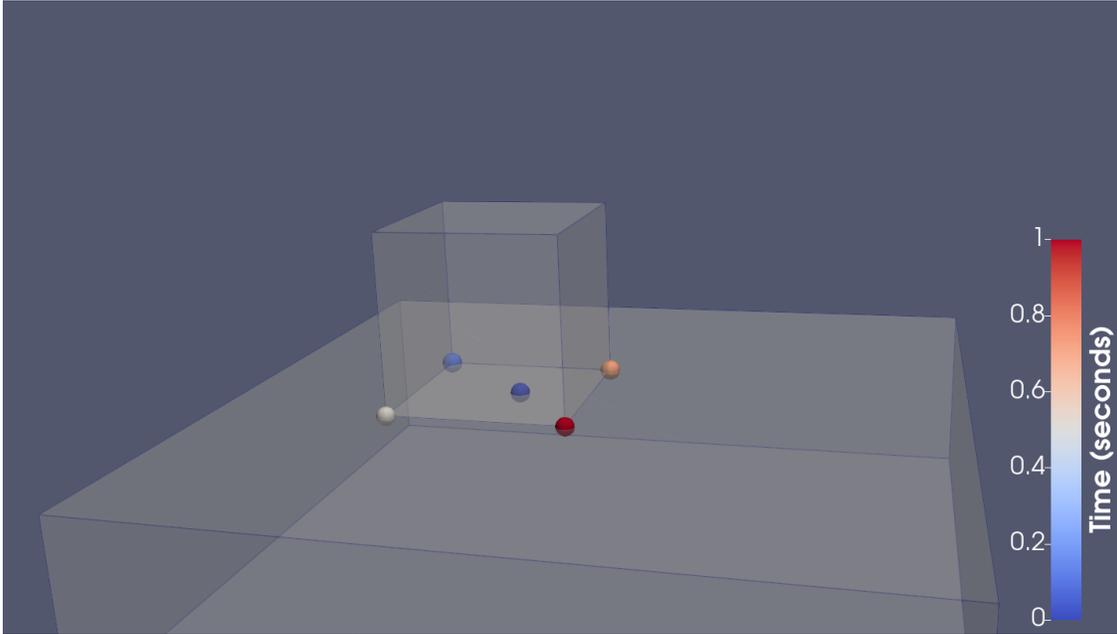


Figure 2.2: Block resting on a boundary block. The contact point is marked with spheres colored by progression in time.

Search for Vertices That Penetrate the Plane of Contact

Following the notation for the Fast Common Plane method introduced by Nezami et al. (2004), the centroids of particles A and B are located on opposing sides of the plane of contact. We first check for vertices that pass the plane of contact, i.e., vertices that are not on the same side of the plane as the centroid. The distance between its vertices and the plane is calculated as follows:

$$d^v = \hat{n} \cdot (V_0 - V) \quad (2.3)$$

where \hat{n} is the unit vector normal to the plane, V_0 is any point on the plane, and V is the vertex of the particle. For a particle with a centroid that is located on the negative side of the plane, if the distance is positive, the vertex passes the plane (Fig. 2.3).

Intersection of particles' edges and the Plane of Contact

Once all vertices that penetrate the plane of contact are identified, we search for intersections of the edges of the block with the plane (Fig. 2.4). An edge intersects the plane if and only if one of its endpoints passes the plane (Condition 1). The equation of the plane of contact can be written as follows:

$$\hat{n} \cdot (P - V_0) = 0 \quad (2.4)$$

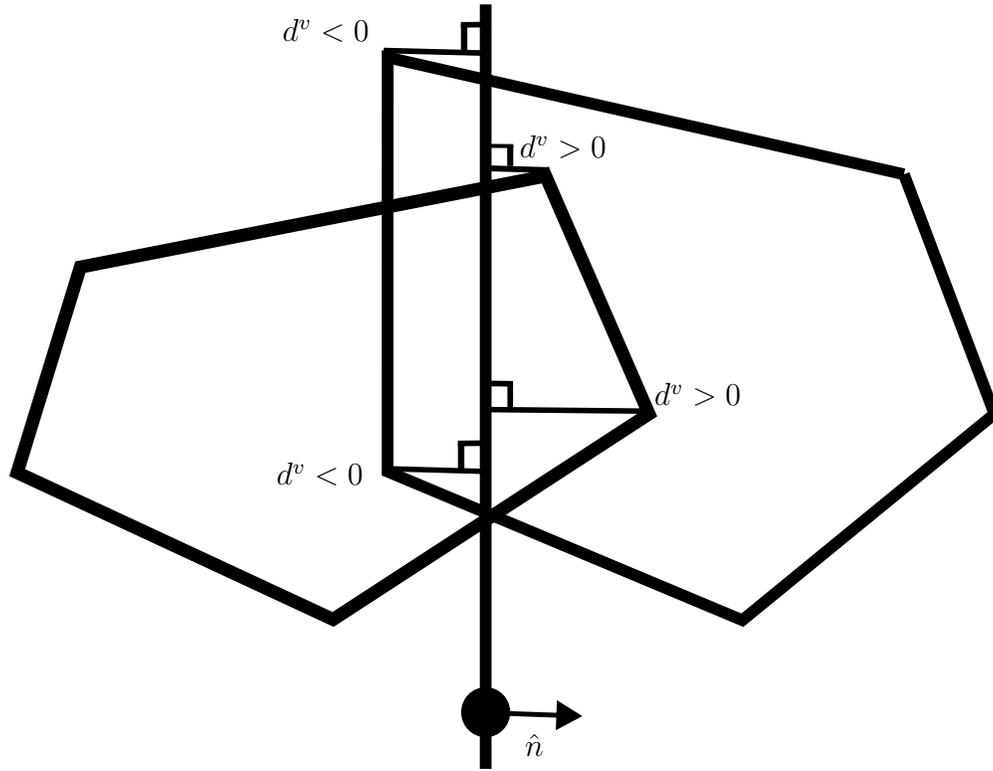


Figure 2.3: The common plane between the particles and the distance sign for the vertices that pass it (after Nezami et al. (2004)).

Where P are points on the plane. The parametric equation of the line going through points P_1 and P_2 is

$$P = P_1 + u(P_2 - P_1) \quad (2.5)$$

An edge defined by P_1 and P_2 intersects the plane if P is equal in both equations. Plugging in Eq. 2.5 in Eq. 2.4 gives:

$$\hat{n} \cdot (P_1 + u(P_2 - P_1) - V_0) = 0 \quad (2.6)$$

Solving for u :

$$u = \frac{\hat{n} \cdot (V_0 - P_1)}{\hat{n} \cdot (P_2 - P_1)} \quad (2.7)$$

The edge intersects the plane if the point of intersection P is between P_1 and P_2 , i.e., $0 \leq u \leq 1$. If the denominator in Eq. 2.7 is 0, the edge is parallel to the plane, and Condition 1 is not satisfied. To find the point of intersection between the edge and the plane, plug in the value for u (Eq. 2.7) in Eq. 2.5.

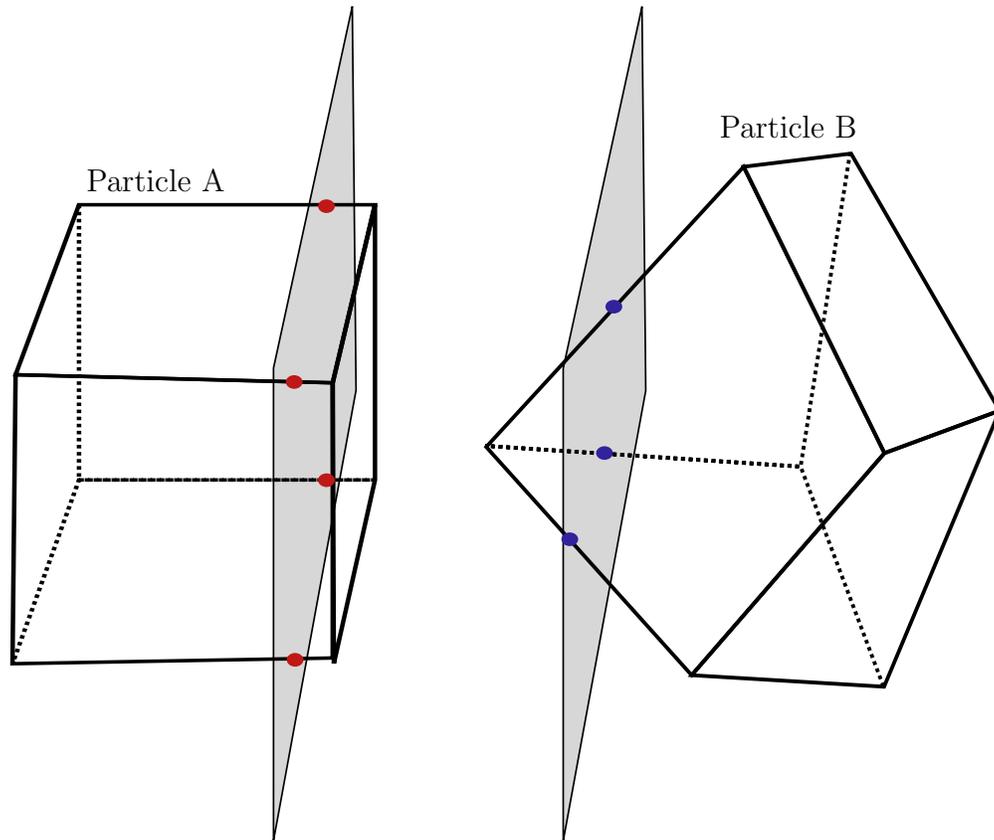


Figure 2.4: Intersection of the particles with the plane of contact.

Intersecting polygons

The intersections of edges and the plane from each particle are used to create two polygons. Note that this is, in fact, a 2D problem, since the two polygons are co-planar. The intersection polygon is composed of:

- Vertices from polygon A located inside polygon B.
- Vertices from polygon B located inside polygon A.
- Intersection of edges of the two polygons.

If all the vertices of polygon A are inside polygon B, then polygon A is the intersection polygon. The same applies to polygon B. If one polygon is not completely contained in the other polygon, one or more edges from each polygon intersect.

Vertices inside polygons

We first check if any vertex in polygon A is located inside polygon B. For convex polygons, a vertex is located inside it if all the inward-pointing normals to its edges point toward the vertex. This can be checked easily with cross and dot products. Given the vertices of polygon A, for example, and the normal to the plane(n), the inward-pointing normal to an edge ($edge_n$) defined by two given vertices v_i and v_{i+1} is:

$$edge_n = \hat{n} \times (v_{i+1} - v_i) \quad (2.8)$$

To check if the normal to the edge points to the point, the dot product is used. A point (p) is inside the polygon (2.5a) if and only if the following condition is satisfied:

$$edge_{ni} \cdot (p - v_i) \geq 0 \quad (2.9)$$

Where v_i are the vertices of the polygon. If this dot product is negative, the angle between the normal and the vector from the endpoint to the point of interest is strictly greater than 90° , and the point is located outside the polygon (Fig. 2.5b).

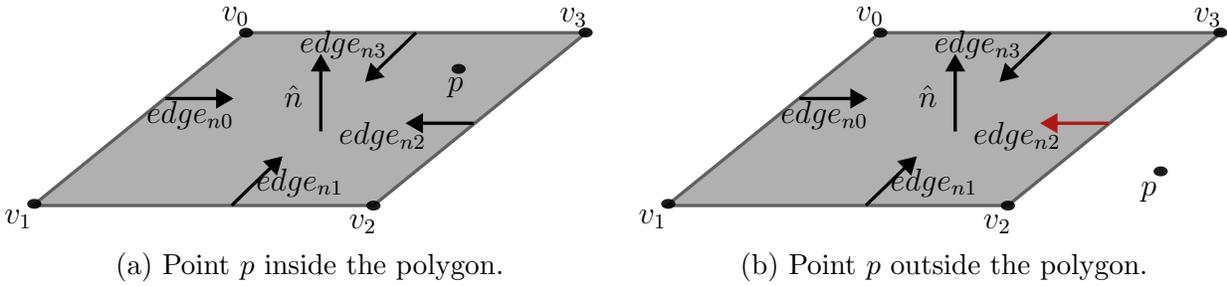


Figure 2.5: Point in or out a convex polygon.

Intersecting edges

To find the intersection of edges, every edge of polygon A is checked against all edges of polygon B. The edges are defined by their endpoints, p_1 and p_2 for $edge_1$, and p_3 and p_4 for $edge_2$. There are a few cases to check for:

The edges are parallel:

If the edges are parallel, their cross-product is the zero vector:

$$|(p_2 - p_1) \times (p_4 - p_3)| = 0 \quad (2.10)$$

The distance between the edges needs to be calculated to check if the edges are parallel or coincident. We can project one endpoint from edge 2 onto edge 1:

$$proj_{edge_1} p_3 = \frac{(p_3 - p_1) \cdot (p_2 - p_1)}{(p_2 - p_1) \cdot (p_2 - p_1)} (p_2 - p_1) \quad (2.11)$$

If the distance between p_3 and its projection onto $edge_1$ is greater than 0, the lines are parallel and do not intersect.

The edges are coincident:

If the distance between p_3 and its projection onto $edge_1$ is 0, the edges are coincident. If the edges are coincident, each endpoint of each edge is checked if it is located between both endpoints of the other edge. Since the edges are co-linear, a simple dot product is sufficient. For example, if p_3 is located between p_1 and p_2 (Fig. 2.6a), the following inequality applies:

$$(p_3 - p_1) \cdot (p_3 - p_2) < 0 \quad (2.12)$$

If the dot product in Eq. 2.12 is greater than zero, p_3 is not considered as a point of intersection (Fig. 2.6b). If both endpoints of the same edge are not considered as a point of intersection, the edges do not intersect.

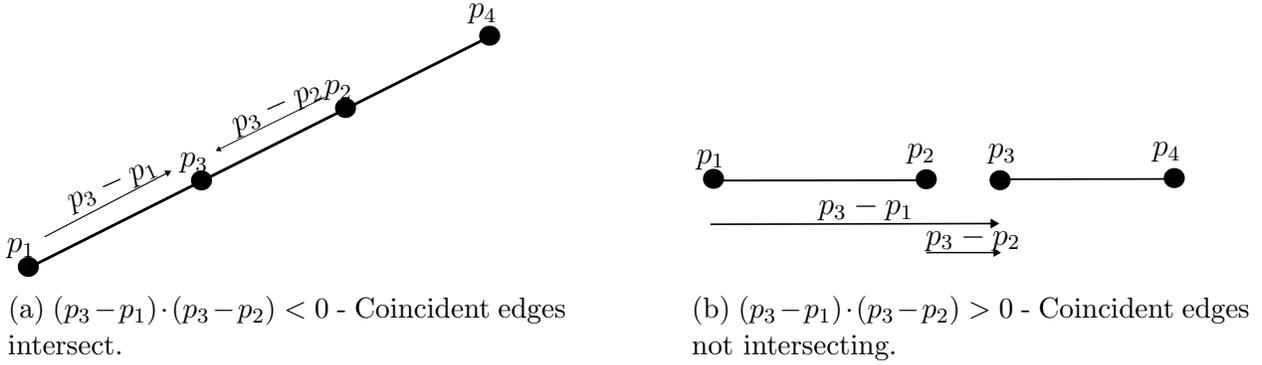


Figure 2.6: Coincident edges cases.

The edges might intersect:

If the edges are not parallel or coincident, we need to check for an intersection. In general, lines in \mathbb{R}^3 rarely intersect. However, both endpoints of the edges are intersections of edges of the original particles with the plane, i.e., the edges are co-planar. Points that lie on the 2 edges have the following equations:

$$p_a = p_1 + t(p_2 - p_1) \quad (2.13a)$$

$$p_b = p_3 + u(p_4 - p_3) \quad (2.13b)$$

The length of the line segment between these two points is:

$$f(t, u) = \|p_a - p_b\|^2 = \|p_1 - p_3 + t(p_2 - p_1) - u(p_4 - p_3)\|^2 \quad (2.14)$$

To find the minimum distance between p_a and p_b , the derivative of f with respect to t and u equals to zero. Solving for t and u gives:

$$u = \frac{(l_{43} \cdot l_{13})(l_{21} \cdot l_{21}) - (l_{43} \cdot l_{21})(l_{43} \cdot l_{21})}{(l_{43} \cdot l_{43})(l_{21} \cdot l_{21}) - (l_{43} \cdot l_{21})(l_{43} \cdot l_{21})} \quad (2.15)$$

$$t = \frac{(l_{21} \cdot l_{43})u - (l_{21} \cdot l_{13})}{(l_{21} \cdot l_{21})} \quad (2.16)$$

Where $l_{13} = p_1 - p_3$, $l_{21} = p_2 - p_1$, $l_{43} = p_4 - p_3$. For brevity, the derivation is presented in A. The point of intersection is then found by substituting t or u in Eq. 2.13. The edges intersect if $t, u \in [0, 1]$ (Fig 2.7). If the denominator of t or u is equal to 0, the edges are parallel or coincident. Note that if the edges intersect $p_a = p_b$.

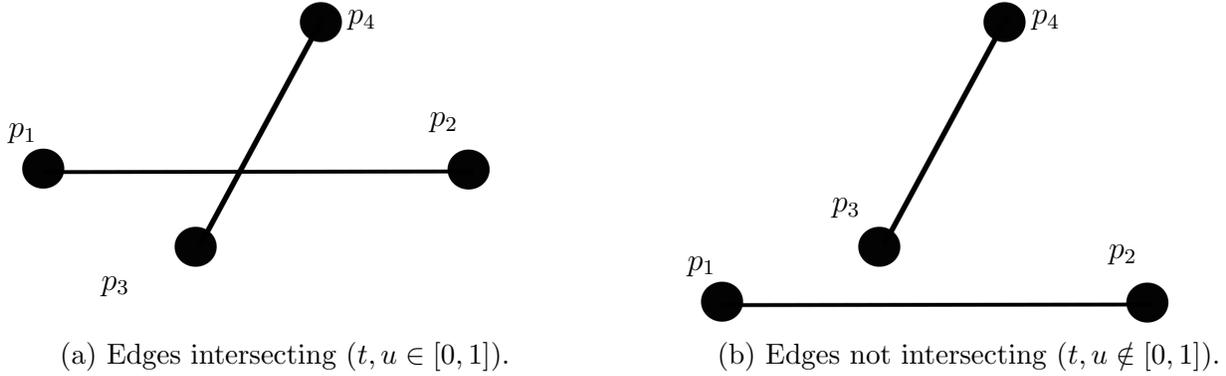


Figure 2.7: Edges intersection cases.

Intersection Polygon

Now that the points of intersection between the two polygons are identified, the contact point is calculated as the center of mass of the intersection polygon (Fig. 2.8) using simplex integration (Shi, 1997). The vertices of the polygon of intersection are given in the local coordinate system on the XY plane. The area of the polygon is calculated as:

$$A = \iint_A dx dy = \frac{1}{2} \sum_{k=0}^n \begin{vmatrix} x_k & y_k \\ x_{k+1} & y_{k+1} \end{vmatrix} \quad (2.17)$$

where n is the number of vertices in the polygon of intersection and $[x_{n+1}, y_{n+1}] = [x_0, y_0]$. The first moment of inertia is calculated as:

$$\begin{aligned} \iint_A x dx dy &= \frac{1}{6} \sum_{k=0}^n \begin{vmatrix} x_k & y_k \\ x_{k+1} & y_{k+1} \end{vmatrix} (x_k + x_{k+1}) \\ \iint_A y dx dy &= \frac{1}{6} \sum_{k=0}^n \begin{vmatrix} x_k & y_k \\ x_{k+1} & y_{k+1} \end{vmatrix} (y_k + y_{k+1}) \end{aligned} \quad (2.18)$$

The center of mass of the polygon of intersection (in local coordinates) is given by:

$$\begin{aligned} x &= \frac{\iint_A x \, dx \, dy}{A} \\ y &= \frac{\iint_A y \, dx \, dy}{A} \end{aligned} \tag{2.19}$$

where the numerator is Equation 2.18 and the denominator is Equation 2.17.

2.4 Examples

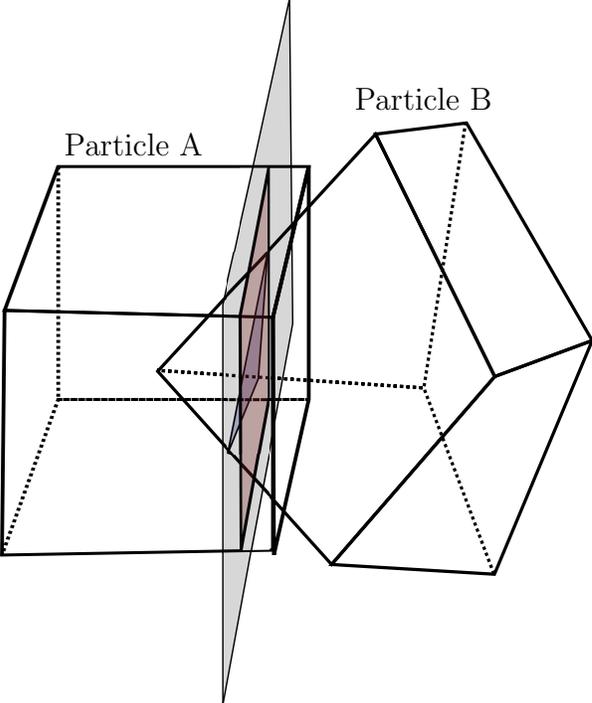
We use several examples to illustrate how the errors associated with the contact point calculation in GJK-EPA can induce numerical instability and non-physical behavior and how the new contact point calculation procedure, IPM, is able to remedy these issues. In doing so, we are able to leverage the efficiency of establishing contact using the GJK-EPA algorithm while improving the stability and reliability of the contact point calculations.

Block Resting on a Horizontal Plane

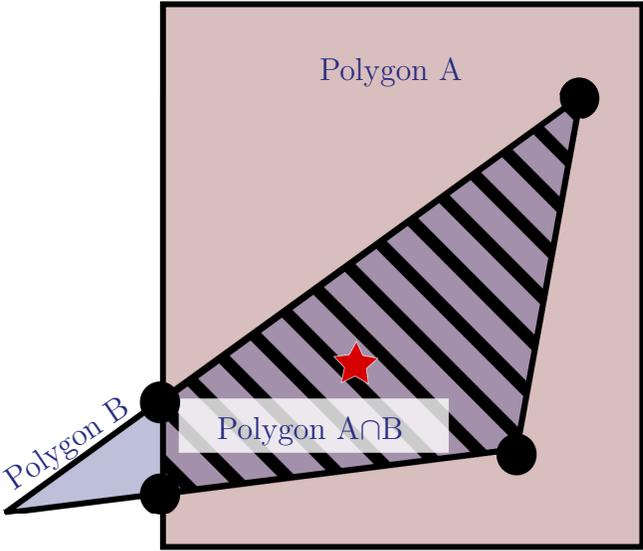
A block is resting on a horizontal, fixed boundary block (Fig. 2.2). In this example, the block should remain stationary as there are no frictional forces and the gravitational force on the block is balanced by the normal force between the block and the stationary boundary. However, as discussed in section 2.2, in some face-to-face contact scenarios, the computed contact point in GJK-EPA is off-center. This induces artificial torque in the block, causing it to rotate, bounce, and slide sideways. Fig. 2.9 shows the displacement of the block's centroid and the rotation angle of the block with respect to a vertical axis in time. The same configuration was modeled using the IPM. The computed contact point remains in the center of the contacting face. No artificial torque is generated, and therefore, the block does not exhibit spurious rotation.

Bouncing Blocks

In this example, two cubic blocks are dropped onto a horizontal surface from the same height. The blocks start from rest, accelerate due to gravity, and then impact the surface (Fig. 2.10a). The contact between the blocks and the surface is a face-to-face contact, and thus should not induce any rotation as the contact force acts through the centroid of the blocks (Fig. 2.10b). The block on the left is modeled using the GJK-EPA, and the block on the right is modeled using the IPM. As discussed in the previous section, the computed contact point using the GJK-EPA is off-center, so the contact force between the block and the surface does not act through the center of mass of the block. This error in the contact point in GJK-EPA computation induces a non-physical torque. After the first impact, the influence of this error causes the block to rotate artificially (Fig. 2.10c) and, upon second



(a) Colliding particles and their intersection with the plane of contact.



(b) Plan view of the intersecting polygons. Points of intersection are marked with black circles. The calculated contact point is marked with a red star.

Figure 2.8: Intersection of particles demonstration.

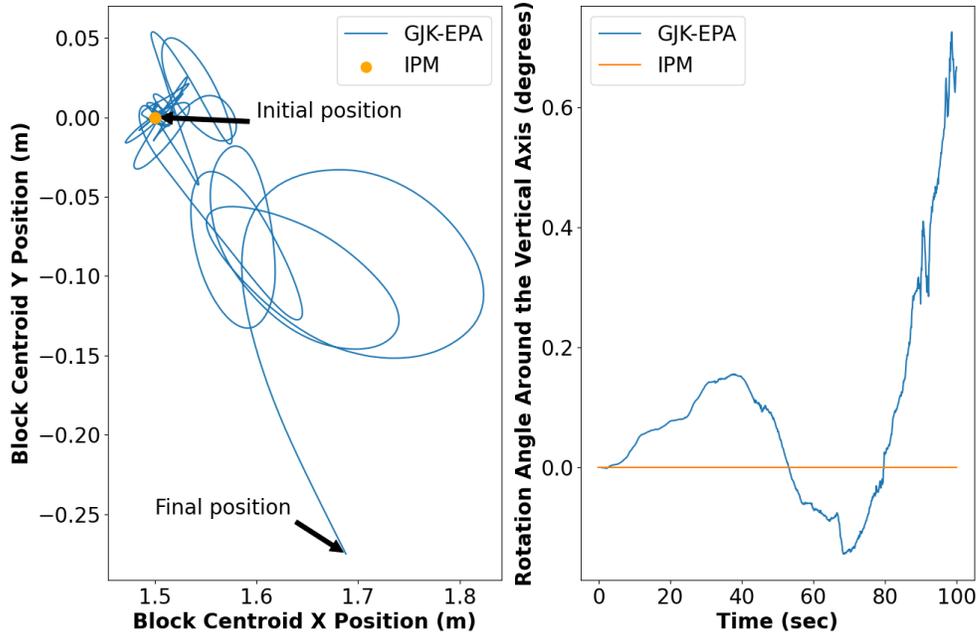


Figure 2.9: Displacement of block centroid (left) and angle of rotation around a vertical axis (right) for a block sitting on a horizontal plane. GJK-EPA algorithm shows non-physical rotation due to contact point error while intersecting polygon method remains stationary and eliminates spurious torque.

impact, the block lands on one of its edges and goes completely off-balance (Fig. 2.10d). In comparison, the motion of the block simulated using the IPM is purely vertical and there is no rotation, as expected and required.

Blocky Sliding Mass

In the previous example, the influence of the contact point calculation was illustrated for a single moving particle that does not interact with other moving particles. However, in most analyses, many particles are interacting with each other and errors in the contact point calculation can induce instability that is non-physical and an artifact of the inaccuracy of the contact point algorithm. To illustrate this aspect of the problem, a wedge composed of several polyhedral particles on a 20° slope is modeled (Fig. 2.11a). The friction angle of the sliding surface and the joints is 25° . The mechanical and numerical parameters used for both models are shown in Table 2.1. The wedge contains vertical joints that divide it into 32 prismatic blocks. The modeled behavior of the fractured wedge is compared for the

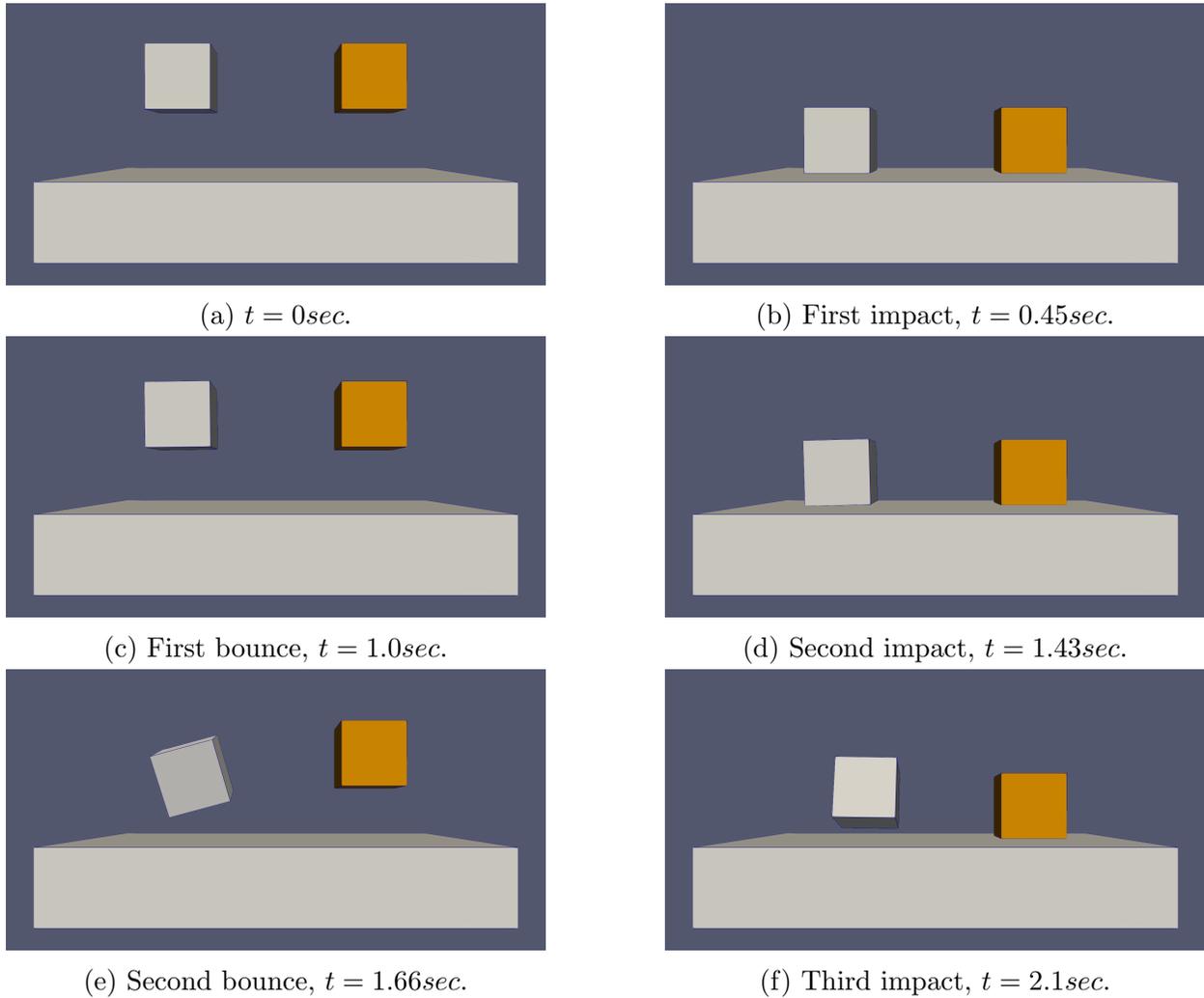


Figure 2.10: Block drops on a horizontal surface, modeled using GJK-EPA (left) and the IPM (right, orange).

case when using the contact point from GJK-EPA and the IPM. Under gravity only and with everything kept equal except for the contact point algorithm, the response of the two simulations is strikingly different. Using the GJK-EPA algorithm (Fig 2.11b), errors in the location of the contact point result in artificial torque that generates rotation in some of the blocks. This induces non-physical displacement of the fractured wedge since the artificial particle rotations push neighboring blocks down the slope. In comparison, using the IPM the blocks remain stable and do not move once sufficient frictional force is mobilized (Fig. 2.11d).

Further, while decreasing the numerical time step size partially mitigates the non-physical torque associated with the contact point from GJK-EPA, the error persists and still induces

Table 2.1: Mechanical and numerical properties for a jointed wedge on an inclined slope.

Friction Angle ($^{\circ}$)	Normal Stiffness (GN/m)	Tangential Stiffness (GN/m)
25	10	5
Density (kg/m^3)	Timestep (sec)	Damping
2000	1×10^{-4}	0.05

artificially large displacements in the fractured mass. For example, decreasing the time step from 1×10^{-4} to 1×10^{-5} seconds in GJK-EPA diminishes the rotation of the blocks (Fig. 2.11b vs. Fig. 2.11c), however; the error persists in a more subtle fashion and, as shown in Section 2.4, can cause global stability issues. Moreover, even though the IPM is more computationally expensive per iteration, the ability to use larger time steps makes it more attractive. For example, a 100-second real-time simulation using the IPM takes ~ 2000 seconds to complete using a time step of 1×10^{-4} seconds. A stable simulation using the traditional GJK-EPA was achieved when the time step was decreased to 1×10^{-5} seconds. This simulation took ~ 5000 seconds to complete while still exhibiting non-physical creep in the fractured mass response.

Jointed Wedge on a Cut Slope

A simulation of a fractured rock slope is used to further demonstrate the implications of a sporadically changing contact point and the improved stability of the IPM. A rock mass is intersected by 3 joint sets to create 53 irregular blocks mimicking a cut slope (Fig. 2.12a). The sliding mass is bounded by three existing fault planes within the rock mass. These fault planes represent the sliding surfaces and the surrounding slope material is modeled using fixed boundary blocks. Table 2.2 summarizes the mechanical and numerical properties used for this simulation. To illustrate the influence of the contact point calculation and potential for misleading numerical results, we use a high friction angle for both the joints (inter-particle friction) and the sliding surface (particle-boundary friction) such that the rock slope should remain stable. When using the IPM, the fractured rock mass remains stable. In comparison, using the GJK-EPA contact point algorithm, the simulation is unstable even when the timestep is an order of magnitude smaller. As shown in Fig. 2.12b the figure, a block at the toe of the sliding mass begins to spontaneously rotate due to the sporadically varying location of the contact point and the associated torque. This block then interacts with neighboring blocks in the fractured rock mass, and eventually exerts large, non-physical forces on its neighbors due to the contact point error. As a result, the angular velocity of this block increases rapidly and destabilizes the entire slope (Fig. 2.12c). While this error can be reduced using a smaller numerical time step, as already noted, the rock slope will

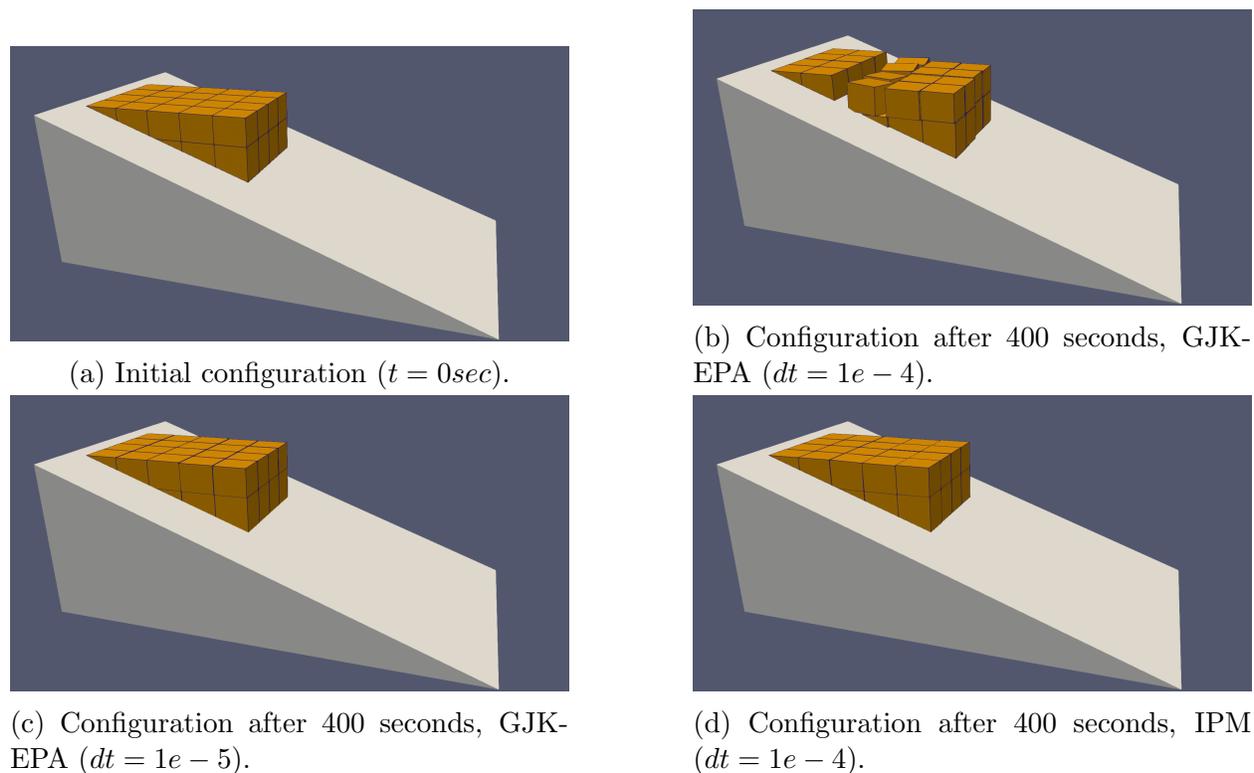


Figure 2.11: A jointed wedge on an inclined slope.

Table 2.2: Mechanical and numerical properties for the cut slope.

Friction Angle ($^{\circ}$)	Normal Stiffness (GN/m)	Tangential Stiffness (GN/m)
35	5×10^3	2.5×10^3
Density (kg/m^3)	Timestep (sec) (IPM / GJK-EPA)	Damping
2000	$1 \times 10^{-5} / 1 \times 10^{-6}$	0.05

still experience non-physical creeping displacement that could be misconstrued as a real, progressive slope failure.

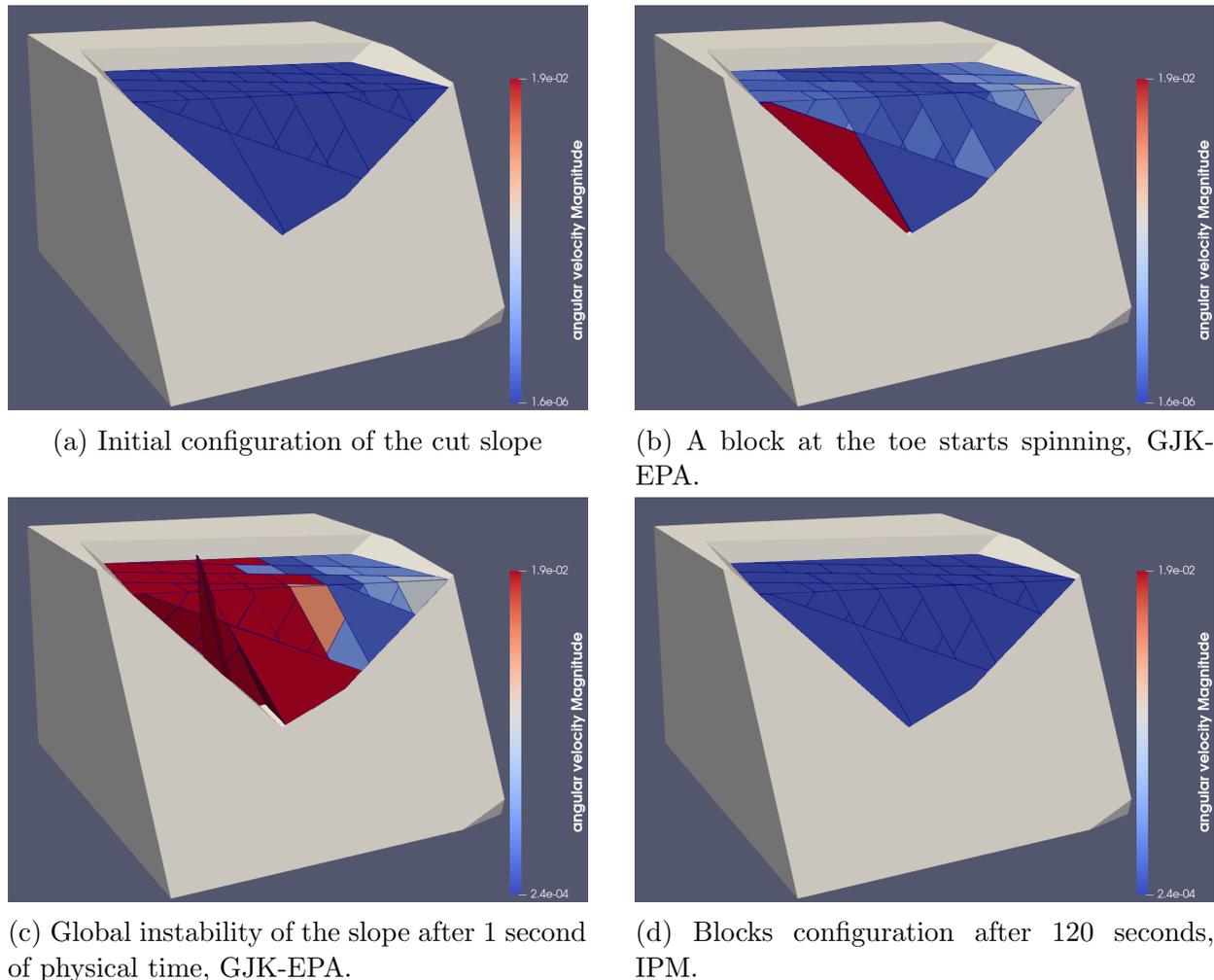


Figure 2.12: 3D model showing the angular velocity of the blocks.

2.5 Conclusion

In a densely packed, fractured rock mass, rock blocks are often positioned in a face-to-face contact. In addition, blocks are typically in contact with more than one neighboring block within the fractured rock. When modeling fractured rock masses using particulate methods, such as DEM, calculating the correct contact point between blocks in contact is essential to capturing the mechanical response appropriately and avoiding numerical instability associated with sporadic changes of the contact. An incorrect contact point might generate an artificial torque and, subsequently, a model that overpredicts displacement. Further, incorrect and non-physical variation of the contact point within a fractured rock mass model may produce irrecoverable numerical instability that could be misinterpreted as instability of the

rock mass.

Herein we have presented a new algorithm for calculating the contact point between two convex polyhedral particles, the Intersection Polygon Method (IPM). The algorithm uses the vertices and edges of the particles to find the intersection points of each particle with the plane of contact. The intersection points from each particle are then used to find two 2D intersection polygons. The contact point is the centroid of the intersection of these two polygons. This algorithm can be paired with any contact detection algorithm and utilized during contact geometry calculations after contact between particles has been established. The algorithm consistently finds the correct contact point and improves performance, particularly in face-to-face contact between polyhedral blocks. Most importantly, the global stability of the models is improved without artificially induced torque and associated particle movement. Even though the algorithm is more computationally expensive compared to other approaches, the ability to use larger time steps due to stable tracking of the contact point makes it attractive, even for large-scale simulations.

Chapter 3

Benchmarking Contact Detection Algorithms Used in Polyhedral Particles System

3.1 Introduction

In particulate systems, neighboring particles collide and interact with each other. To capture the response of a blocky rock mass in a numerical simulation, and to fully understand the kinematic effect in an assembly of particles, it is necessary to model the particles as polyhedrons. The Discrete Element Method (DEM) (Cundall and Strack, 1979) and Discontinuous Deformation Analysis (DDA) (Shi, 1988) have been extensively used in research to study the response of fractured rock masses in slopes, tunnels, and granular assemblies. In these simulations, the contact detection between polyhedral particles is the most computationally expensive step, which accounts for more than 80% of the total computation time (Horner et al., 2001).

In order to increase computational efficiency and to eliminate unnecessary operations, the contact detection is limited to nearest neighbors or particles deemed close enough to warrant a more thorough check. Algorithms, such as binning algorithms (Munjiza and Andrews, 1998; Williams et al., 2004; Krijgsman et al., 2014) or hierarchical type algorithms (O'Connor, 1996; Feng and Owen, 2002; Perkins and Williams, 2001) are commonly used to limit the search region and establish a list of nearest neighbors that may potentially be in contact.

The nearest neighbors are then checked for actual contact and, for particles found to be in contact, the contact geometry is determined, including the contact normal, the contact point, and the penetration depth. Here, we focus on this stage of the contact detection and contact geometry resolution, as it is more complex and computationally expensive when polyhedral particles are considered, and we review the commonly used contact resolution algorithms. Specifically, we compare the performance in terms of accuracy and efficiency

of several classes of commonly used contact detection algorithms. This comparison within a single simulation framework facilitates the selection of a contact detection algorithm that balances accuracy and performance for different classes of simulations.

3.2 Contact Detection Algorithms

Below we provide a brief summary of the three classes of contact detection algorithms that were evaluated as part of this work. These three classes represent a broad grouping of different contact detection approaches and we have selected representative approaches from within each of these classes. All algorithms were implemented within a single computational framework by using a customized version of the open-source platform `waLBer1a` (Bauer et al., 2021). By using a single simulation framework, all elements of the computations are identical except for the contact detection phase, which allows for an unbiased comparison between the selected algorithms. Within each simulation, everything is kept equal except for the calculation of the contact features.

Topology Methods

Our first class of contact detection algorithms uses a topological approach to establish contact and contact geometry. In this type of algorithm, the shape of the polytope of penetration between two particles is implicitly approximated to calculate the contact features. An example of this approach is the highly efficient and robust Gilbert-Johnson-Keerthi (GJK) algorithm (Gilbert et al., 1988), which is capable of handling any convex shape. The GJK algorithm determines whether two convex shapes intersect by calculating the minimal distance between them using a support function. Its efficiency arises from the fact that it leverages the convexity of the shapes, making it computationally cheaper for convex polyhedra compared to other methods. The Expanding Polytope Algorithm (EPA) (Van Den Bergen, 2001) is coupled with GJK to compute the exact contact features by finding the shortest distance between the overlapping shapes. EPA uses the final simplex returned from GJK, and iteratively expands it by adding and removing faces until the face closest to the origin is also a face on the boundary of the Minkowski difference between the two colliding particles. Herein, the Intersection Polygon Method (IPM) (Keissar et al., in review) is used to augment GJK-EPA to address the accuracy issues reported by Feng and Tan (2020) and Keissar et al. (in review). This method uses the output from GJK-EPA (contact normal and a point on the contact plane) to recalculate the contact point coordinates. The intersections of each particle with the plane of contact are used to form two polygons. The algorithm then searches for intersection of these two polygons to form a polygon of intersection. The centroid of this polygon is the contact point. The Entrance Block Theory (Shi, 2015) is another member of this class. Similar to the GJK algorithm, this method takes advantage of the Minkowski difference between colliding particles, but contact between particles A and B is determined

using a reference point (\mathbf{a}_0) of block A and an entrance block defined by Shi (2015) as:

$$E(A, B) = \cup_{\mathbf{a} \in A, \mathbf{b} \in B} (\mathbf{b} - \mathbf{a} + \mathbf{a}_0) = B - A + \mathbf{a}_0 \quad (3.1)$$

When the reference point is inside the entrance block, the particles penetrate each other. In this work, the GJK-EPA method and IPM approach are used as examples of this class of algorithms.

Common Plane Methods

In Common Plane Methods (CPM), the shape of two polyhedral particles is explicitly used to calculate contact features such as the contact normal and the contact point. This approach was first presented in Cundall (1988). In this method, the particles are checked for contact with a plane that bisects the space between them, and not with each other directly. This method was later augmented by the Fast Common Plane (FCP) (Nezami et al., 2004) and the Shortest Link Method (SLM)(G. Nezami et al., 2006), to find the same common plane more efficiently. In FCP, the authors found that the number of possible common planes is finite and must be parallel to the edges or faces containing the vertex closest to the common plane in each particle. This led to a 12-40 times speed-up in contact calculation (Nezami et al., 2004). In this manuscript, FCP is coupled with the IPM (Keissar et al., in review) to compute the contact point and contact geometry.

Convex Optimization

The final class of algorithms reviewed here uses convex optimization to establish contact. The approach developed by Boon et al. (2012) uses linear programming (LP) to check if two particles are potentially in contact where each particle is represented by a set of inequalities that describe the particle faces. Inner potential particle are used to smooth the vertices and edges of the particle to avoid abrupt changes in the direction of the contact normal and contact type as particles slide past each other (Houlsby, 2009). In the event that particles are found to be in contact, the contact point is found using the log barrier method.

Other classes of contact detection algorithms were designed to address numerical instability in the case of rapid changes in the contact normal based on energy conservation (e.g. Feng and Owen, 2004; Feng et al., 2012) or direct search methods (Zhang et al., 2015, 2016; Zheng et al., 2017b,a); however, these are not investigated in this manuscript.

3.3 Benchmark Tests

A series of benchmark tests is presented to compare the different contact detection algorithms. First, we test each contact type possible for polyhedral particles in contact. Then, static and dynamic simulations are executed to demonstrate the performance of different

algorithms and compare their accuracy and efficiency. Specifically, we compare the performance of GJK-EPA, GJK-EPA-IPM, FCP, and LP to evaluate the relative performance of the different classes of contact detection algorithms discussed in Chapter 3.2.

Polyhedral Particles Contact Type Tests

In polyhedral particles, there are six possible contact types: vertex-vertex, vertex-edge, vertex-face, edge-edge (the edges are crossing or parallel), edge-face, and face-face (Fig. 3.1). The different contact detection algorithms are used to calculate the contact features (contact normal, contact point, and penetration depth). The solution obtained using each algorithm is then compared with the analytical solution.

Vertex-Vertex Contact

Two prismatic particles are placed such that they are in vertex-vertex contact (Fig. 3.1a). The upper particle is perfectly balanced on the fixed prismatic particle below, and reaction forces between the particles should be aligned with their axes (no moments should be generated). According to the FCP algorithm (Nezami et al., 2004), the common plane is the plane that maximizes the gap between the two closest vertices of the opposite particles. In this example, this plane is a face of the upside-down pyramid and not a horizontal plane. As a result, the contact point and the penetration are miscalculated. A similar issue occurs in the GJK-EPA and GJK-EPA-IPM algorithms. The shortest distance that one particle has to move to solve the penetration is inclined and not vertical, and the contact normal is inclined. As a result, the calculated force between the particles pushes the upper particle away from the base, and in subsequent steps, the pyramid slides down the tetrahedron. However, using LP, the contact plane is horizontal. The smooth inner potential particle prevents sudden changes in the contact type, since its faces are curved so that its gradient does not change abruptly (Boon et al., 2012). However, the contact point is slightly different from the analytical solution. The contact features calculated using each algorithm are presented in Table 3.1. Extreme cases when two colliding particles intersect exactly on vertices are clearly not common. However, the smooth faces of the inner potential particle used in LP give it an advantage over the other algorithms. This algorithm is the only one that provides a stable particle when the model runs for multiple steps.

Vertex-Edge Contact

A pyramid is placed on the edge of another particle such that the particles are in vertex-edge contact (Fig. 3.1b). Similarly to Chapter 3.3, using FCP, GJK-EPA, and GJK-EPA-IPM, the calculated contact normal is inclined and the contact point is miscalculated. LP provides the correct contact normal. The smooth faces of the inner particle and the slow changes in its gradient prevent abrupt changes in contact normal and penetration depth, but the contact point is only accurate to the fifth decimal place (Table 3.2). As seen in the previous section,

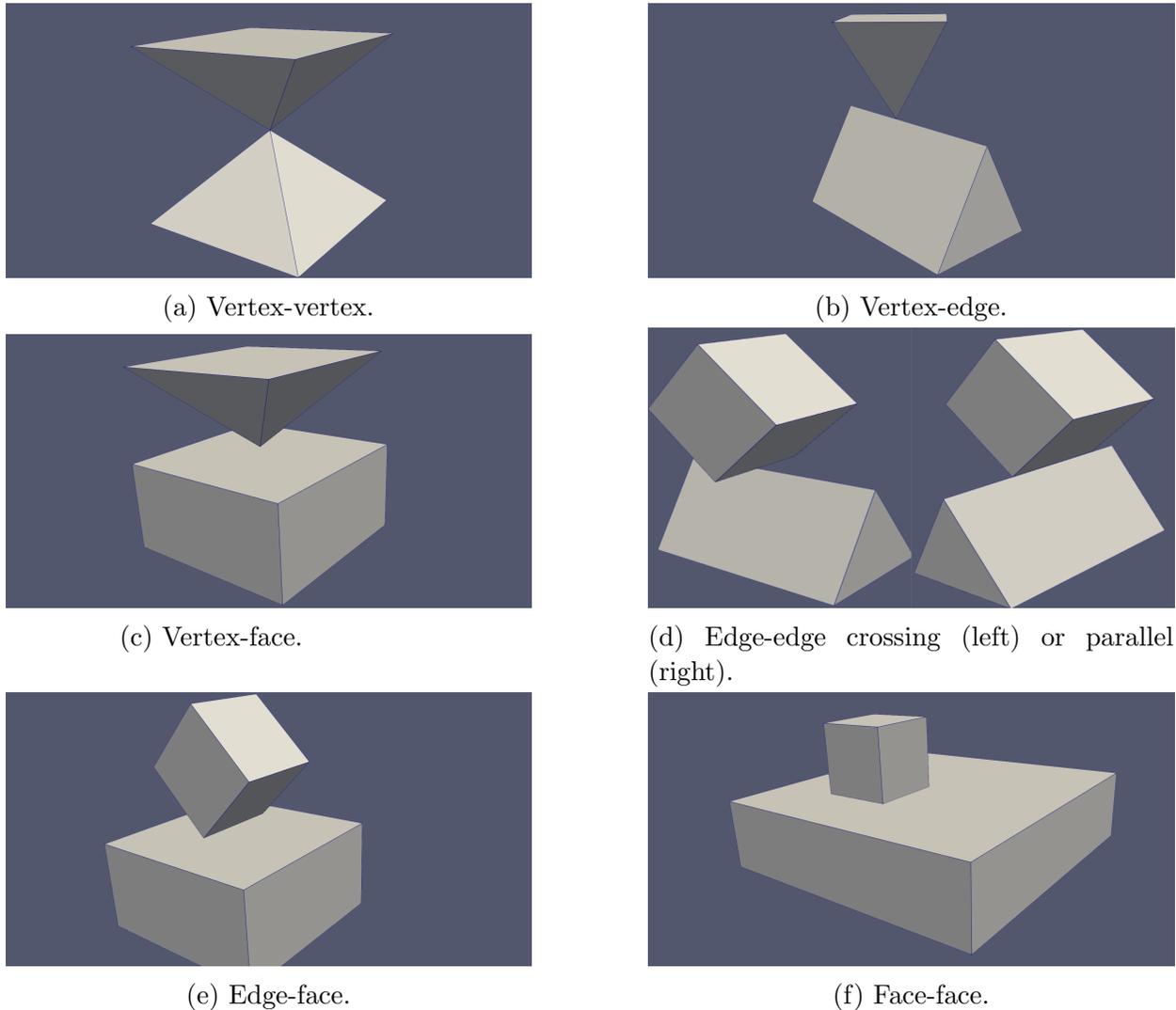


Figure 3.1: Polyhedral particles contact types.

two colliding particles in a vertex-edge contact type are improbable. In a rare case, it does happen, the force acting between the particles will be inclined and acting such that artificial torque will be generated. In subsequent steps, however, the particles may rearrange and the contact type could change to a more stable configuration.

Vertex-Face Contact

A pyramid is placed on a horizontal, fixed boundary particle so that one of its vertices is in contact with the face of the boundary particle (Fig. 3.1c). The pyramid should be perfectly balanced on its vertex. Simulations modeled using FCP provides results matching

Table 3.1: Contact features as calculated using different contact detection algorithms for vertex-vertex collision (ε is a number smaller than machine precision 1^{-16}).

Algorithm	Contact Normal	Contact Point	Penetration Depth
Analytical Solution	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
FCP	$\begin{bmatrix} -0.7071 \\ 0 \\ 0.7071 \end{bmatrix}$	$\begin{bmatrix} -6.08935 \times 10^{-7} \\ 4.69098 \times 10^{-7} \\ -5.60894 \times 10^{-6} \end{bmatrix}$	7.0711×10^{-6}
GJK-EPA	$\begin{bmatrix} -0.2636 \\ 0.5374 \\ 0.8011 \end{bmatrix}$	$\begin{bmatrix} 8.7239 \times 10^{-6} \\ -8.7239 \times 10^{-6} \\ 3.7239 \times 10^{-6} \end{bmatrix}$	6.8266×10^{-7}
GJK-EPA-IPM	$\begin{bmatrix} -0.2636 \\ 0.5374 \\ 0.8011 \end{bmatrix}$	$\begin{bmatrix} -1.36589 \times 10^{-7} \\ 1.12411 \times 10^{-7} \\ -5.79902 \times 10^{-6} \end{bmatrix}$	6.8266×10^{-7}
LP	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \varepsilon \\ \varepsilon \\ -4.2857 \times 10^{-6} \end{bmatrix}$	1×10^{-5}

the analytical solution since FCP uses the faces of the particles to obtain the contact normal. In GJK-EPA, the contact normal is slightly tilted due to the expanding polytope face closest to the origin. The polytope keeps expanding by adding or removing faces until a face in the polytope is only close enough to a face of the Minkowski difference of the colliding particles, but not the face of the particles themselves. As a result, the contact point calculated using GJK-EPA is slightly off-center. The IPM augmentation is not affected by the little tilt in contact normal and returns the correct contact point. The penetration depth calculated using LP is accurate, while the contact normal is slightly tilted and not exactly perpendicular to the boundary surface. The contact point is accurate to its sixth decimal place (Table 3.3). FCP provides the best results in this case; however, the errors related to the other algorithms in this case are small and are not expected to affect the performance of a full-scale model.

Edge-Edge Contact

In most cases, edges of two 3-D colliding particles intersect. In some cases, however, the edges might be parallel, where the contact between them occurs exactly along a line.

Table 3.2: Contact features as calculated using different contact detection algorithms for vertex-edge collision.

Algorithm	Contact Normal	Contact Point	Penetration Depth
Analytical Solution	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
FCP	$\begin{bmatrix} -0.816497 \\ 0 \\ 0.57735 \end{bmatrix}$	$\begin{bmatrix} 3.00008 \times 10^{-7} \\ \varepsilon \\ -4.57573 \times 10^{-6} \end{bmatrix}$	5.77350×10^{-6}
GJK-EPA	$\begin{bmatrix} -0.1426 \\ 3.642 \times 10^{-6} \\ 0.9898 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -2.19504 \times 10^{-10} \\ -5 \times 10^{-6} \end{bmatrix}$	5.52748×10^{-6}
GJK-EPA-IPM	$\begin{bmatrix} -0.1426 \\ 3.642 \times 10^{-6} \\ 0.9898 \end{bmatrix}$	$\begin{bmatrix} 6.96214 \times 10^{-8} \\ -6.30497 \times 10^{-12} \\ -4.98997 \times 10^{-6} \end{bmatrix}$	5.52748×10^{-6}
LP	$\begin{bmatrix} 0 \\ 1.03783 \times 10^{-10} \\ 1 \end{bmatrix}$	$\begin{bmatrix} \varepsilon \\ -2.31481 \times 10^{-12} \\ -3.3333 \times 10^{-6} \end{bmatrix}$	1×10^{-5}

Parallel Edges A rotated cube is placed on a fixed long prismatic particle such that the particles are in edge-edge contact. The edges in contact are perfectly parallel, and the cube slightly penetrates the boundary particle (penetration depth = 1×10^{-5}).

The contact normal is inclined when FCP is used, as seen in Chapter 3.3. As a result, the contact point slightly shifts away from the edge in contact (Fig. 3.2). Using GJK-EPA, when two features from opposing particles (edges in this case) are parallel, the contact point is off-center, as presented in Keissar et al. (in review). Moreover, the calculated contact normal is tilted and does not match the analytical solution. Using the IPM augmentation, the calculated contact point is miscalculated because the input contact normal is tilted. When LP is used, the contact normal is accurate, but the contact point shifts along the edge. Consequently, the reaction forces between the particles will induce artificial torque. Thus, none of the algorithms are able to capture this extreme case and, as the simulation progresses, the cube would slide down the fixed particle in any of the algorithms examined here.

Crossing Edges To examine the results for different contact detection algorithms when the edges cross, a rotated cube is placed on the edge of a prismatic particle, as shown in Fig. 3.1d. The cube is perfectly balanced on the edge of the fixed prismatic particle. Using

Table 3.3: Contact features as calculated using different contact detection algorithms for vertex-face collision.

Algorithm	Contact Normal	Contact Point	Penetration Depth
Analytical Solution	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
FCP	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
GJK-EPA	$\begin{bmatrix} 1.32169 \times 10^{-13} \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -9.97501 \times 10^{-11} \\ -9.03479 \times 10^{-11} \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
GJK-EPA-IPM	$\begin{bmatrix} 1.32169 \times 10^{-13} \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \varepsilon \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
LP	$\begin{bmatrix} 2.21268 \times 10^{-10} \\ 2.21268 \times 10^{-10} \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1.33333 \times 10^{-11} \\ -1.33333 \times 10^{-11} \\ -2 \times 10^{-6} \end{bmatrix}$	1×10^{-5}

FCP, the calculated contact features match the analytical solution. Using GJK-EPA, the contact normal is very slightly tilted since the calculation is terminated when the face of the expanding polytope is only close enough to the Minkowski difference of the colliding particles. As a result, the contact point is a little off-center. The error in the contact normal calculated in GJK-EPA is not large enough to affect the contact point calculation in the IPM. When IPM augmentation is used to complement GJK-EPA, the error at the contact point is smaller. When LP is used, some very small errors in contact normal and contact point are introduced, mostly related to the radius of the inner potential particle. The results are summarized in Table 3.5. The errors related to the different algorithms are very small in this case and are not expected to have a significant effect in a large-scale simulation. With more timesteps, the cube remains perfectly balanced on the fixed particle.

Edge-Face Contact

A cubical particle rests on a horizontal, fixed boundary particle. The boundary particle is shifted so that its centroid is not aligned with the centroid. The particle is rotated so that one of its edges is in contact with the face of the boundary particle (Fig. 3.1e). The center of mass of the particle is above the edge in contact, so the system is balanced and stable under

Table 3.4: Contact features as calculated using different contact detection algorithms for parallel edge-edge collision.

Algorithm	Contact Normal	Contact Point	Penetration Depth
Analytical Solution	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
FCP	$\begin{bmatrix} 0 \\ 0.7071 \\ 0.7071 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -3.3494 \times 10^{-7} \\ -5.3349 \times 10^{-6} \end{bmatrix}$	7.0711×10^{-6}
GJK-EPA	$\begin{bmatrix} 5.2419 \times 10^{-7} \\ 0.2561 \\ 0.9666 \end{bmatrix}$	$\begin{bmatrix} -0.06667 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	6.9914×10^{-6}
GJK-EPA-IPM	$\begin{bmatrix} 5.2419 \times 10^{-7} \\ 0.2561 \\ 0.9666 \end{bmatrix}$	$\begin{bmatrix} 0.00897 \\ 4.55872 \times 10^{-7} \\ -5.1618 \times 10^{-6} \end{bmatrix}$	6.9914×10^{-6}
LP	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0.0406 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}

gravity force alone. Using FCP, the contact features match the analytical values (Fig. 3.3). When GJK-EPA is used, a small error is introduced in the contact normal, as seen before. When two parallel features are in contact (edge and face in this case), the contact point is not in the center of the edge in contact when the cube is not placed right at the center of the boundary particle, inducing artificial torque in the particle, as discussed in Keissar et al. (in review). In subsequent steps, the contact point jumps between the endpoints of the edge. Eventually, the particle rotates and falls on one of its faces. Using GJK-EPA-IPM, the contact point is recalculated, and the error in the contact point is smaller by orders of magnitude. The contact point slightly shifts from the center due to the little tilt of the contact normal. The intersections of the particles with the contact normal form a trapezoid instead of a rectangle and the centroid of the two polygons is slightly off-center compared with the rectangle representing the actual intersecting polygon. Using LP, when the cube is not placed exactly at the center of the boundary particle, the contact point moves along the edge, as seen in Section 3.3. Artificial torque builds up, and eventually, the particle rotates and falls on one of its edges. The calculated contact features are shown in Table 3.6.

Here, FCP calculates the contact features that most closely match the analytical solution. GJK-EPA and LP compute an off-center contact point that would very quickly lead to a build-up of artificial torque. The IPM augmentation remedies this issue to some degree, but

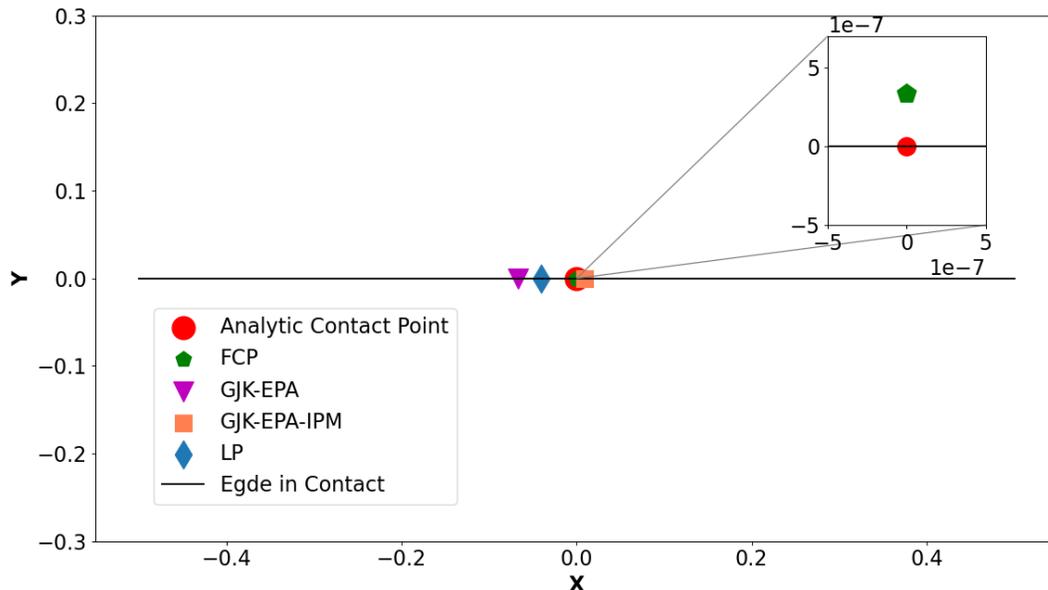


Figure 3.2: Parallel edge-edge contact point as calculated using the different algorithms.

the contact point is still slightly off-center, and the effect of this is investigated later in this manuscript.

Face-Face Contact

A cubical particle rests on a horizontal, fixed boundary particle such that the faces of both particles are in contact (Fig. 3.1f). Using FCP, the contact features match the analytic solution. As discussed in Keissar et al. (in review), when the GJK-EPA is used, the contact point is off-center in the initial time steps when features (faces) of particles in contact are parallel (Fig. 3.4). In subsequent steps, induced torque rotates the particle, and the contact point jumps between the vertices of the face in contact. IPM remedies this malady, and the contact features match the analytic solution. The LP suffers from similar issues as seen in GJK-EPA but at a smaller magnitude. The contact point is slightly off-center, inducing small artificial torque.

Table 3.5: Contact features as calculated using different contact detection algorithms for crossing edge-edge collision.

Algorithm	Contact Normal	Contact Point	Penetration Depth
Analytical Solution	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
FCP	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
GJK-EPA	$\begin{bmatrix} -9.9127 \times 10^{-14} \\ 4.6259 \times 10^{-13} \\ 1 \end{bmatrix}$	$\begin{bmatrix} -4.8151 \times 10^{-10} \\ 9.3707 \times 10^{-10} \\ -5 \times 10^{-6} \end{bmatrix}$	6.9914×10^{-6}
GJK-EPA-IPM	$\begin{bmatrix} -9.9127 \times 10^{-14} \\ 4.6259 \times 10^{-13} \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1.2226 \times 10^{-12} \\ \varepsilon \\ -5 \times 10^{-6} \end{bmatrix}$	6.9914×10^{-6}
LP	$\begin{bmatrix} 0 \\ 1.7875 \times 10^{-10} \\ 1 \end{bmatrix}$	$\begin{bmatrix} \varepsilon \\ -5.2083 \times 10^{-12} \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}

Static Multi-particle Test

The first model used to test the overall performance (accuracy and efficiency) is static. Before each simulation begins, an initialization stage is performed since particles are initially only touching and small displacements are required to establish initial contact forces. This initialization stage eliminates excess oscillations in the contact forces prior to starting the simulation. During this stage, the mechanical damping is increased to 80%, the friction angle is increased to 89° , and particles settle under gravity only until the model reaches initial equilibrium. This process is described in Itasca 3DEC Program Guide (Itasca Consulting Group Inc., 2016). The average force ratio is used to monitor how close the system of particles is to equilibrium. This ratio is defined as the sum of all out-of-balance forces divided by the total force applied to the system:

$$R = \frac{\sum_j^n \langle \sum_i f_i \rangle}{\sum_j^n \sum_i \langle f_i \rangle} \quad (3.2)$$

where The angle brackets $\langle \rangle$ represent the ‘‘Manhattan norm’’:

$$\langle f_i \rangle = |f_x| + |f_y| + |f_z| \quad (3.3)$$

Table 3.6: Contact features as calculated using different contact detection algorithms for edge-face collision.

Algorithm	Contact Normal	Contact Point	Penetration Depth
Analytical Solution	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
FCP	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
GJK-EPA	$\begin{bmatrix} -1.9825 \times 10^{-13} \\ 1.8173 \times 10^{-13} \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0.06667 \\ 6.9396 \times 10^{-10} \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
GJK-EPA-IPM	$\begin{bmatrix} -1.9825 \times 10^{-13} \\ 1.8173 \times 10^{-13} \\ 1 \end{bmatrix}$	$\begin{bmatrix} 3.3042 \times 10^{-9} \\ \varepsilon \\ -5 \times 10^{-6} \end{bmatrix}$	1×10^{-5}
LP	$\begin{bmatrix} 0 \\ 3.0061 \times 10^{-10} \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0.04061 \\ -9.2592 \times 10^{-12} \\ -3.3333 \times 10^{-6} \end{bmatrix}$	1×10^{-5}

The model is considered to reach equilibrium when the average force ratio reaches a threshold value of 1×10^{-5} . In the next sections, simulations begin only after initial equilibrium has been achieved and in-situ stresses have been established.

A model with multiple particles that can interact with each other is used to illustrate the effect of miscalculating contact features. Twenty-five cubic particles rest on a fixed horizontal boundary particle (Fig. 3.5). The particles are in face-to-face contact with the boundary particle and only in touching contact with each other, so no forces should act between them, only reaction forces from the boundary particle. Using FCP, no forces are acting between the particles and each particle only interacts with the lower boundary due to gravity. Using the GJK-EPA, the contact points between each particle and the boundary are off-center. As a result, an artificial torque is generated and causes some rotation of the particles. Due to this rotation, the particles apply a significant force on their neighbors as the contact point jumps between vertices (Fig. 3.6). This model cannot reach an initial equilibrium. Using GJK-EPA-IPM, the contact points are centered with respect to each particle and the horizontal forces are negligible. This model requires the same number of steps to reach initial equilibrium as the FCP model. Using LP the contact points are also off-center, but not as much as the GJK-EPA. The horizontal forces are considerably lower compared to GJK-EPA, and the model is able to reach an initial equilibrium though it takes

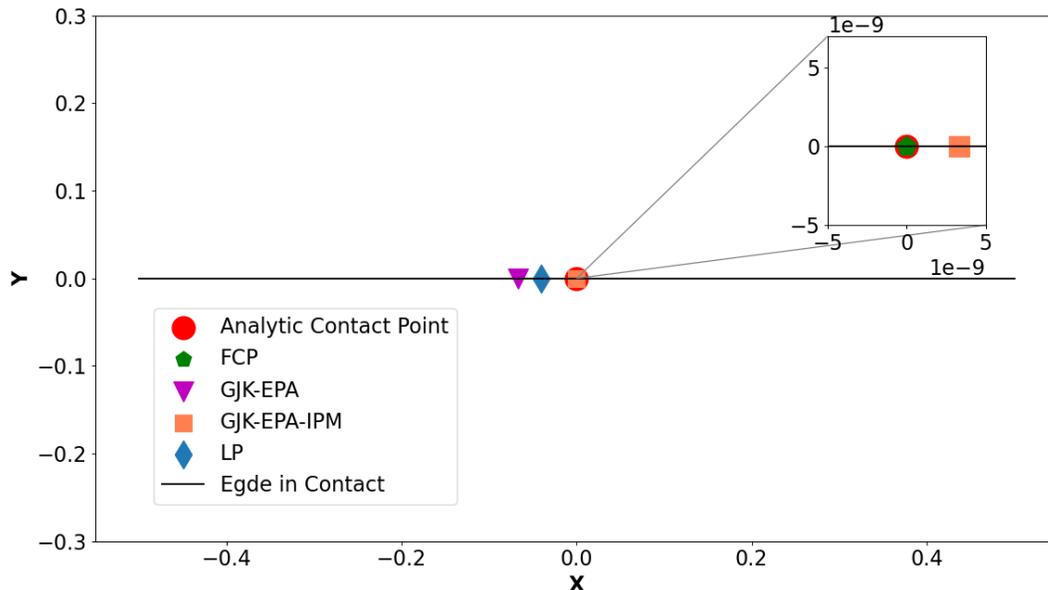


Figure 3.3: Edge-Face contact contact point as computed using the different contact detection algorithms.

considerably more time compared to FCP and GJK-EPA-IPM, as shown in Table 3.7.

Table 3.7 further shows a comparison between all tested contact detection algorithms in terms of computational efficiency. GJK-EPA is by far the fastest algorithm in terms of calculation cycles per second, but the model does not converge to initial equilibrium. Moreover, due to artificially induced torque associated with errors in the contact point calculation, more contacts between particles are checked in each step because of particles are rotating. LP converges to initial equilibrium after the largest number of timesteps. Moreover, this algorithm is the slowest per timestep since a linear program must be solved for each contact check, and a log-barrier problem in the event that the particles are in contact. FCP and GJK-EPA-IPM perform similarly, with the latter slightly faster due to the efficiency in establishing contact inherited from GJK-EPA .

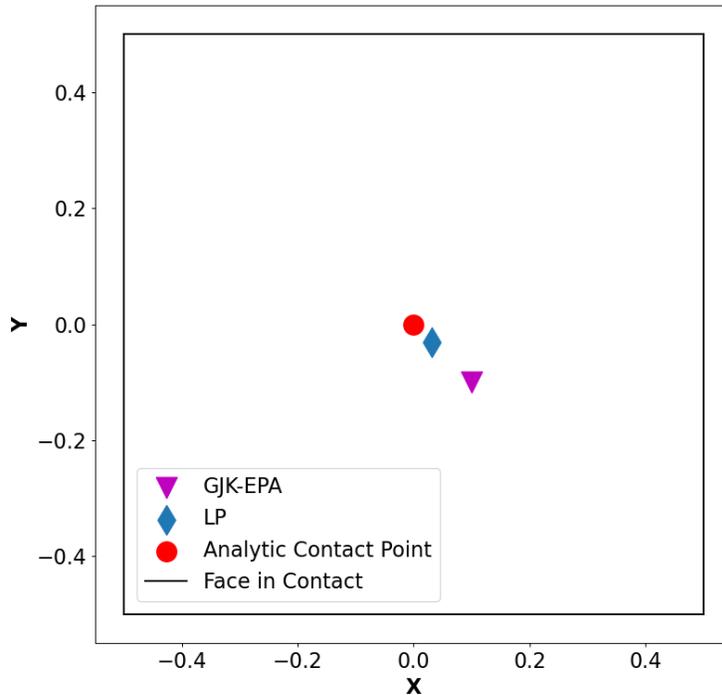


Figure 3.4: Face-Face contact contact point. Contact points calculated using LP are numbered according to the timestep.

Table 3.7: Face-Face execution times.

Algorithm	FCP	GJK-EPA	GJK-EPA-IPM	LP
Number of contact detection steps per second	114,818	569,955	126,753	35,408
Speed up compared to the worst performance	3.2	16.1	3.6	1
Number of steps to reach equilibrium	3,123	FAILED	3,123	468,948
Total time to reach initial equilibrium (seconds)	4.27	FAILED	3.76	2079

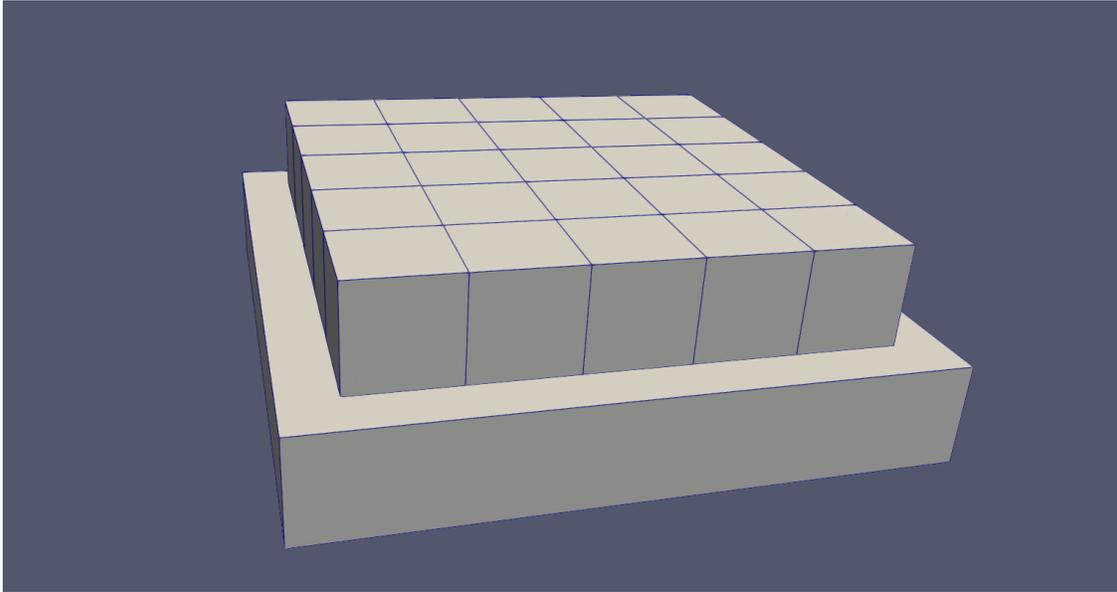


Figure 3.5: 25 cubic particles rest on a horizontal plane.

Dynamic Test

A dynamic problem is presented to further examine the accuracy and efficiency of each contact detection algorithm, and potential implications associated with errors in establishing contact geometry. A rod with a hexagonal cross-section is rolled down a 30° slope. The long axis of the rod is horizontal, so the rod should roll in a straight line in the direction of the dip. In addition, the rod is placed on the slope on one of its edges, so it should roll down the slope rather than slide. The initial position of the rod is shifted along its axis so that it is not in the middle of the boundary particle such that the rod and boundary particle centers of mass are not vertically aligned.

When GJK-EPA is used to detect contact between the rolling particle and the slope, the contact point is off-center, whether the rod is on one of its edges or on one of its faces, as seen in Chapters 3.3 and 3.3. Artificial torque is induced, and the rod bounces between its endpoints. The trajectory of the center of mass of the rod diverges from the long axis of the boundary particle. This issue is partially solved when the contact point is calculated properly using the IPM. The contact point is affected by the slight tilt in the contact normal returned from GJK-EPA. As seen in Chapter 3.3, the intersections of both particles with the plane of contact form a parallelogram or a trapezoid instead of a rectangle. The centroid of the polygon is, therefore, not the same as the centroid of the rectangle, which should represent the polygon of intersection. However, the error in the contact point is smaller than the contact point returned from GJK-EPA. The rod does not bounce from side to side, but gradually slides sideways (Fig. 3.8).

When FCP coupled with IPM is used to compute the contact point, the contact normal

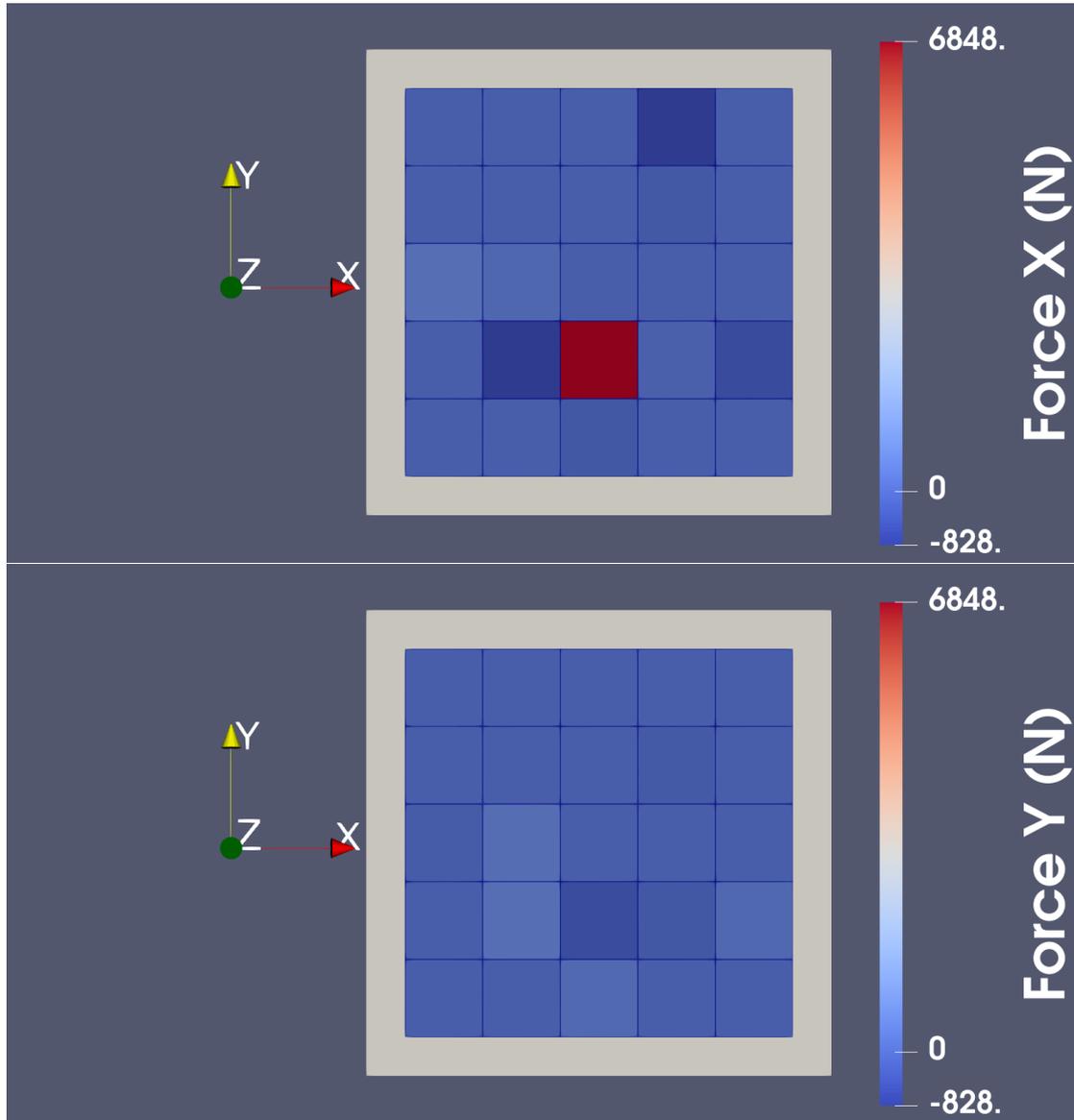


Figure 3.6: Face-Face contact with multiple particles when using the GJK-EPA contact algorithm: Plan view of the particles and horizontal forces acting on the particles.

is defined by the face of the boundary particle, and the hexagonal rod rolls down the slope as expected (Fig. 3.7). As the particle rolls, the contact type alternates between edge-face and face-face, and the contact features (contact normal and contact point) match the analytic solution.

When LP is used, the contact point is off center with respect to the long axis of the rod, as seen in Chapters 3.3 and 3.3. Consequently, torque is induced in the rod and it bounces

between the endpoints. The rod does not roll in a straight line in the direction of the slope, but alternates depending on the shift in the contact point location. The trajectory of the centroid of the hexagonal rod, as simulated using each algorithm, is shown in Fig. 3.8. The trajectories are shown in a plane parallel to the slope.

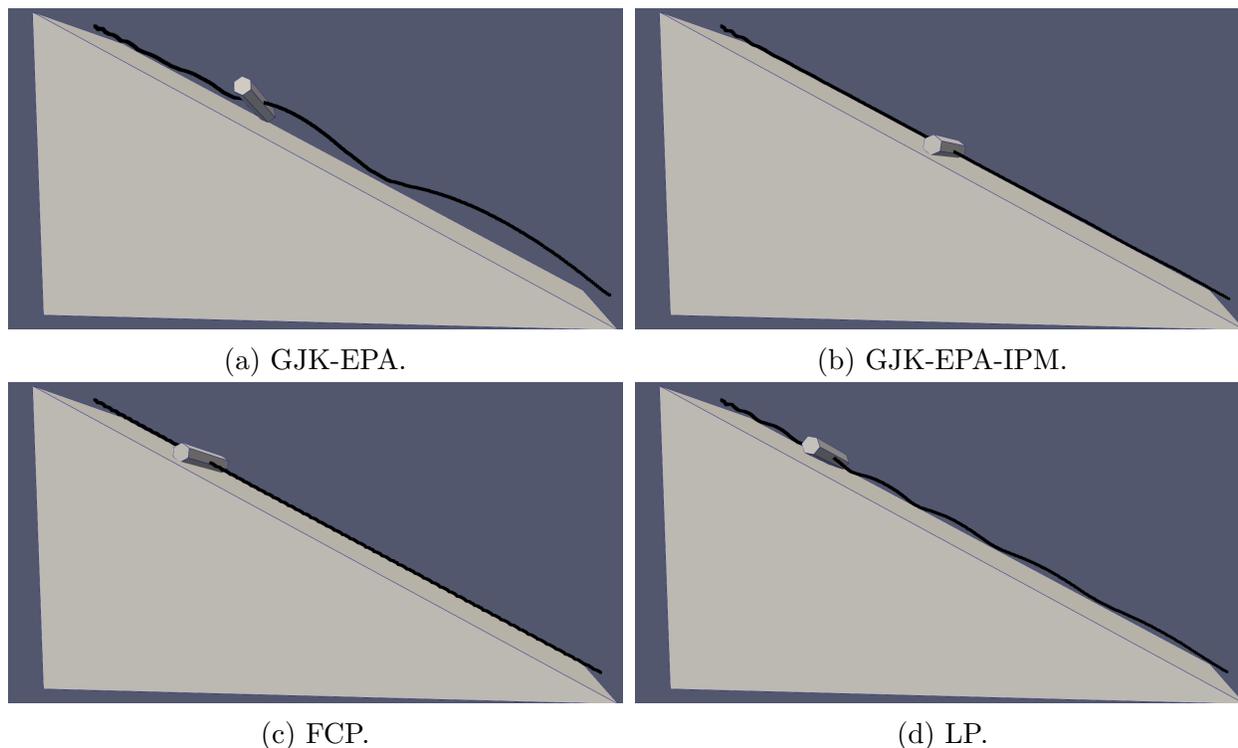


Figure 3.7: Hexagonal rolling down the slope. Orange points mark the trajectory of the center of mass of the particle.

Computing times using the different algorithms are shown in Table 3.8. As shown previously, GJK-EPA is the fastest algorithm tested here. The solution, however, does not match the analytical solution. The error using GJK-EPA-IPM is smaller, but the simulation time is about 5 times slower than GJK-EPA. FCP coupled with IPM captured the expected behavior of the rolling hexagonal rod but was significantly slower than GJK-EPA-IPM. LP is the slowest and did not capture the expected behavior.

3.4 Discussion

For polyhedral particles, the task of finding the correct contact geometry that are used to apply forces on the particles is not straightforward. Throughout this paper, we illustrate some limitations and advantages of each contact detection algorithm. While GJK-EPA is

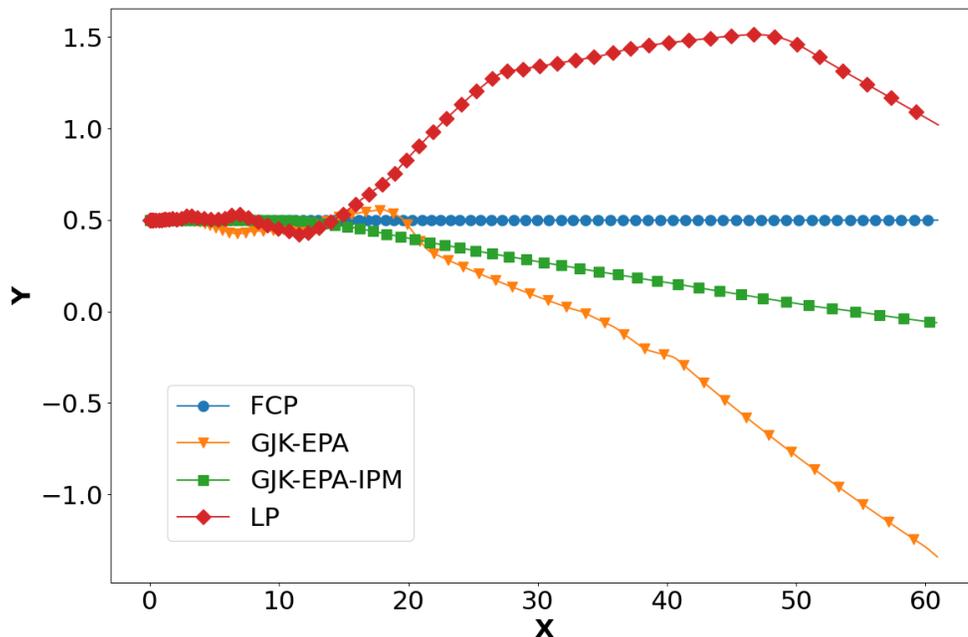


Figure 3.8: Trajectory of the centroid of the hexagonal rod as it slides down the slope.

Table 3.8: Rolling hexagonal rod execution times.

Algorithm	FCP	GJK-EPA	GJK-EPA-IPM	LP
Number of contact detection steps per second	26,044	458,237	91,944	22,105
Speed up compared to the worst performance	1.2	20.7	4.2	1

robust, can deal with any shape of convex particles, and is the fastest algorithm checked in this research, it suffers from an incorrect contact point calculation that provides misleading and potentially numerically unstable results. When features of two colliding particles are parallel and the particles are shifted such that the contacting features are not aligned, the contact point is off-center which induces artificial torque that affects the overall stability of the model. This issue is expected to have the most impact on static or quasi-static models in which displacement is not expected. This issue will have a lesser effect in very dynamic models, where particles collide frequently and faster.

IPM is designed to be coupled with GJK-EPA, or any other contact detection algorithm,

to find the correct contact point, and it benefits from the fast calculation of contact normal and penetration depth using GJK-EPA. This algorithm is affected by the slight tilt in contact normal returned from GJK-EPA, but for static models it provides accurate results with high efficiency.

LP provides good results for most contact types, but solving a linear program to detect each contact is very time-consuming. This algorithm also suffers from inaccuracies when the particles are not centered with respect to each other (as shown in Chapters 3.3 and 3.3).

FCP is robust in terms of finding the correct contact normal and penetration depth for common contact types. The IPM augmentation is well suited for FCP to compute the contact point, using its accurate contact normal defined by the edges and faces of the particles. Although FCP is less efficient than GJK-EPA, the improved stability due to finding the correct contact features also allows the user to use larger time steps.

3.5 Conclusion

In this paper, some commonly used contact detection algorithms from different classes are compared for accuracy and efficiency. First, benchmark tests for every possible contact type in convex polyhedral particles are carried out to verify the results of each contact detection algorithm (Chapter 3.3). In extreme cases when contact between two particles is reduced to a point or a line (vertex-vertex, vertex-edge, and parallel edge-edge), the contact is not well defined using GJK-EPA, GJK-EPA-IPM, and FCP. The contact normal does not match the analytical solution and as a result, the contact point is miscalculated. LP provides the correct contact normal and penetration depth, with only a small error in the contact point, mainly due to the use of smooth potential particles and the fact that the gradient to the smooth faces does not change abruptly. Note that in a full-scale simulation where particles are colliding, it is unlikely that the contact between neighboring particles is exactly on the edges and vertices of both particles.

In other contact types, when features from two colliding particles are parallel (edge-edge, edge-face, or face-face), the contact point may be miscalculated using GJK-EPA, as discussed in Feng and Tan (2020) and (Keissar et al., in review). A similar issue occurs in the LP, but to a smaller magnitude. In both algorithms, miscalculating the contact point causes artificial torque to build up, and the simulation is less numerically stable. The latter augmentation to GJK-EPA, the IPM, was introduced to recalculate the contact point using the output from GJK-EPA. In the augmented GJK-EPA-IPM, the contact point is accurate and those instabilities are averted. FCP provides good results as well.

Next, static and dynamic simulations were executed to illustrate the effect of miscalculating contact features and to compare the efficiency of the different contact detection algorithms. The artificial torque generated as a result of the miscalculation of the contact point is shown to affect the stability of both static and dynamic simulations. In a static simulation, the particles apply forces on their neighbors due to rotation and displacement caused by an artificial torque using the GJK-EPA and LP. In a dynamic simulation, the

direction of sliding and rotation may be incorrect when contact features are not obtained properly.

Chapter 4

Capturing 3D Kinematics in Rock Slope Failure

4.1 Introduction

In fractured rock masses, slope failure occurs along pre-existing discontinuities, such as joints and fractures (e.g. Goodman and Kieffer, 2000), and the interaction between the discontinuities and the geometry of the slope plays an important role in the displacement of the blocky rock mass, both for failure initiation and runout. While limit equilibrium (LE) solutions can deal with the potential for instability to occur, they cannot be used to evaluate the potential for start-stop motion and, ultimately, the potential for runout once failure occurs.

Although larger rock slides have been modeled as equivalent fluids, an alternative that preserves the characteristics of the jointed rock mass is to use methods such as the discrete element method (DEM) (Cundall and Strack, 1979; Cundall, 1988; Hart et al., 1988) or discontinuous deformation analysis (DDA, Shi (1988)) that explicitly model the kinematics of block movement within a blocky rock mass. For example, the large rock avalanche in the Vajont Dam reservoir was modeled using DDA by Sitar et al. (2005) to estimate the strength properties of the failure surface, the velocity of the sliding mass, and the kinematic response that led to the catastrophic failure. In a similar effort, Boon et al. (2014) used DEM to explicitly define the geometry of the failure surface and the sliding mass to back-calculate the friction along the basal plane. While these and other similar analyses have amply demonstrated the potential and usefulness of modeling the discrete nature of fractured rock masses, the adoption of these methods and, in particular, the adoption of 3D versions of these methods has lacked due to the large computational burden.

In this work, we compare the limit equilibrium method with the discrete element method in both two- and three-dimensional dimensions to better understand the influence of kinematics on the stability of slopes, the initiation of failure, different possible failure modes, and, ultimately, the runout of the failing rock mass. To address the computational burden,

we use a modern HPC (High Performance Computing) implementation of DEM, which opens opportunities for efficient and affordable full-scale dynamic analyses of failure initiation and runout.

4.2 Numerical Methods

Limit Equilibrium Methods

LE methods are widely used in slope stability analyses (e.g., Duncan, 1996; Cheng et al., 2007; Choudhury et al., 2007; Zornberg et al., 1998). Their prevalence, simplicity, and speed make them attractive for research and practice. In these methods, the sliding mass is divided into slices in typical 2D analyses and columns in 3D analyses. The most basic assumption in limit equilibrium methods of analysis is the need to assume the location and geometry of the failure surface for each iteration in the analyses. While these methods are typically designed to perform a search for the "critical failure surface" the shape of this surface is predetermined at the outset. The factor of safety is then calculated by dividing the shear strength along the failure surface by the driving forces acting on the sliding mass.

Discrete Element Method

The Discrete Element Method (DEM) is used to numerically simulate the mechanical behavior of a rock mass subjected to loading. In contrast to LE methods, DEM provides an opportunity to consider the kinematics of the problem as well as the displacements and velocities of the individual blocks and the block interactions within the fractured rock mass. In this method, the rock mass can be subdivided into an assemblage of blocks using field data: orientation of discontinuities, joints, fractures, bedding planes, and foliation (e.g. Cundall and Strack, 1979; Cundall, 1988; Hart et al., 1988). The Newton-Euler equations are then solved explicitly for each polyhedral block so the kinematic response of the rock mass can be captured. The computationally expensive step in this analysis is the determination of contacts between the blocks and the computation of the resulting contact forces, e.g. Horner et al. (2001). Hence, the solution of problems with hundreds to thousands of blocks becomes computationally demanding, which can be alleviated by parallel implementation of the algorithms. In this work, we used a customized version of the open-source code `waLBerla` (Bauer et al., 2021).

4.3 Case Study Site Description

Our case study uses data from the 1996 slope failure at the Round Hill open pit, which was a part of Macraes Mining Company Ltd (MMCL) gold mining operations in eastern Otago, New Zealand. The mining district is located along the Hyde-Macraes Shear Zone, with extensive faults, folds, and fractures in the rock mass. The rocks composing the pit



Figure 4.1: Round Hill open-pit mine failure

wall belong to the Haast Schist Group and consist of highly deformed psammitic and pelitic schists. A large failure of the northern wall of the pit (Fig. 4.1) prompted a more detailed investigation of the geologic setting and the mechanical properties of the rock mass (Chapple, 1998). The failure wedge in the pit, designated RH28, was bounded by four sets of faults and it was internally fractured by different sets of joints, smaller faults, and schistosity. A schematic of the 3D geometry of the failure wedge is shown in Fig.4.2.

Chapple (1998) performed extensive tests on the rock from the Round Hill pit and from the failed rock mass. For the purposes of the analyses presented here, we focus only on the properties of joints and shears. As is typical in these settings, the laboratory data give results different from those obtained from back analyses, because the laboratory samples are essentially point data relative to the size of the failure plane. In addition, ring shear samples do not account for the larger scale asperities along the failure plane. In this context, laboratory ring shear data gave residual friction coefficient values in the range of $3 - 6^\circ$, whereas the values back-calculated from 2D limit equilibrium wedge failure analyses gave values in the range of $9 - 14^\circ$. In comparison, the results of back-analyses gave apparent friction angle on the joint surface on the order $41 - 55^\circ$ with no cohesion, and Chapple (1998) opined that assuming the apparent cohesion along wedge intersection in the range 6-20 kPa, the apparent friction angle would fall into a more realistic range of $25 - 35^\circ$.

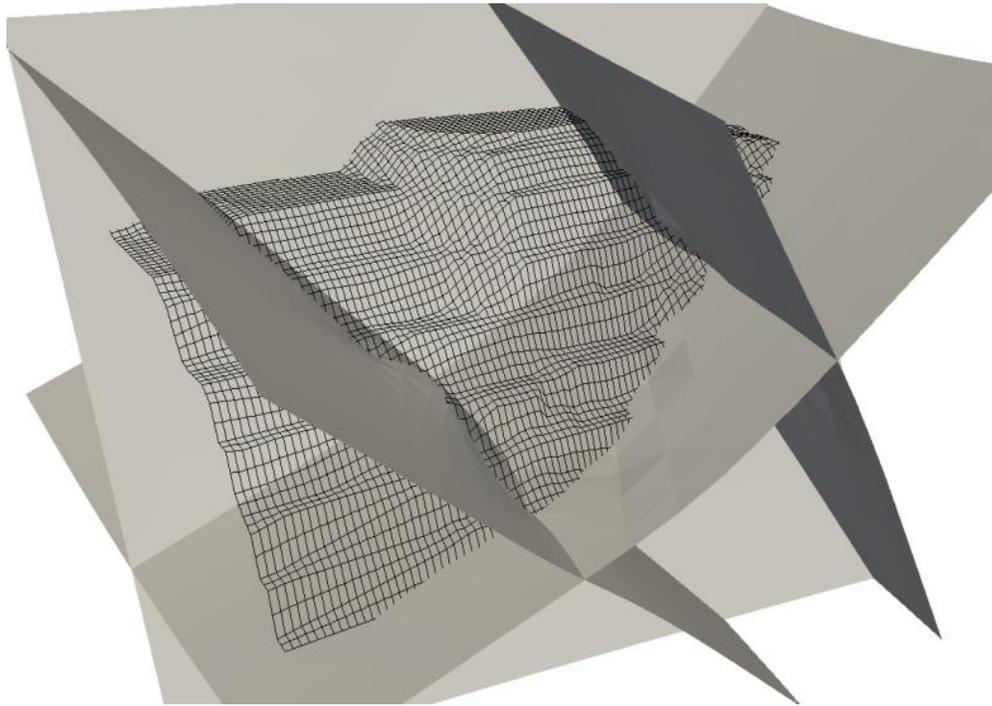


Figure 4.2: Schematic model for Round Hill open-pit mine in New Zealand (Brown et al., 2016). The grid represents the columns used in the LE analysis. The planes are faults bounding the sliding mass.

4.4 2D Planar Failure Analysis

2D Limit Equilibrium and Quasi-2D DEM Comparison

For the purposes of the comparison, we performed a series of 2D LE analyses to evaluate the influence of various parameters on the mode of failure in a jointed rock mass. Specifically, we used a 2D analysis of a section mimicking the Round Hill pit failure to perform a back analysis in order to estimate the strength parameters along the failure surface (Fig. 4.3). The back of the assumed sliding surface dips 62° toward the free surface. The friction angle of the back fault is 40° . The circular basal surface daylights and dips about 3° away from the free surface. Subdividing the rock mass into vertical slices, the computed factor of safety of approximately 1 was achieved when the friction angle on the potential failure surface was 20° (for dry conditions).

To accommodate non-circular potential failure surface, we used the Spencer's Method (Spencer, 1967) in which the factor of safety is calculated with respect to moments equilibrium and with respect to horizontal force equilibrium. The inter-slice shear force (T) to normal force ratio, N , is then iteratively updated until the two values of the factor of safety converge. That computed ratio, N is the inclination of the inter-slice resultant force. The

equivalent friction angle is then calculated as:

$$N = T \tan \phi \tag{4.1}$$

When the factor of safety is 1, the inclination of the resultant inter-slice force is equal to the friction angle. In the case used for the comparison analyses in this study, the equivalent inter-slice force angle, β , for the 2D LE model was 32° .

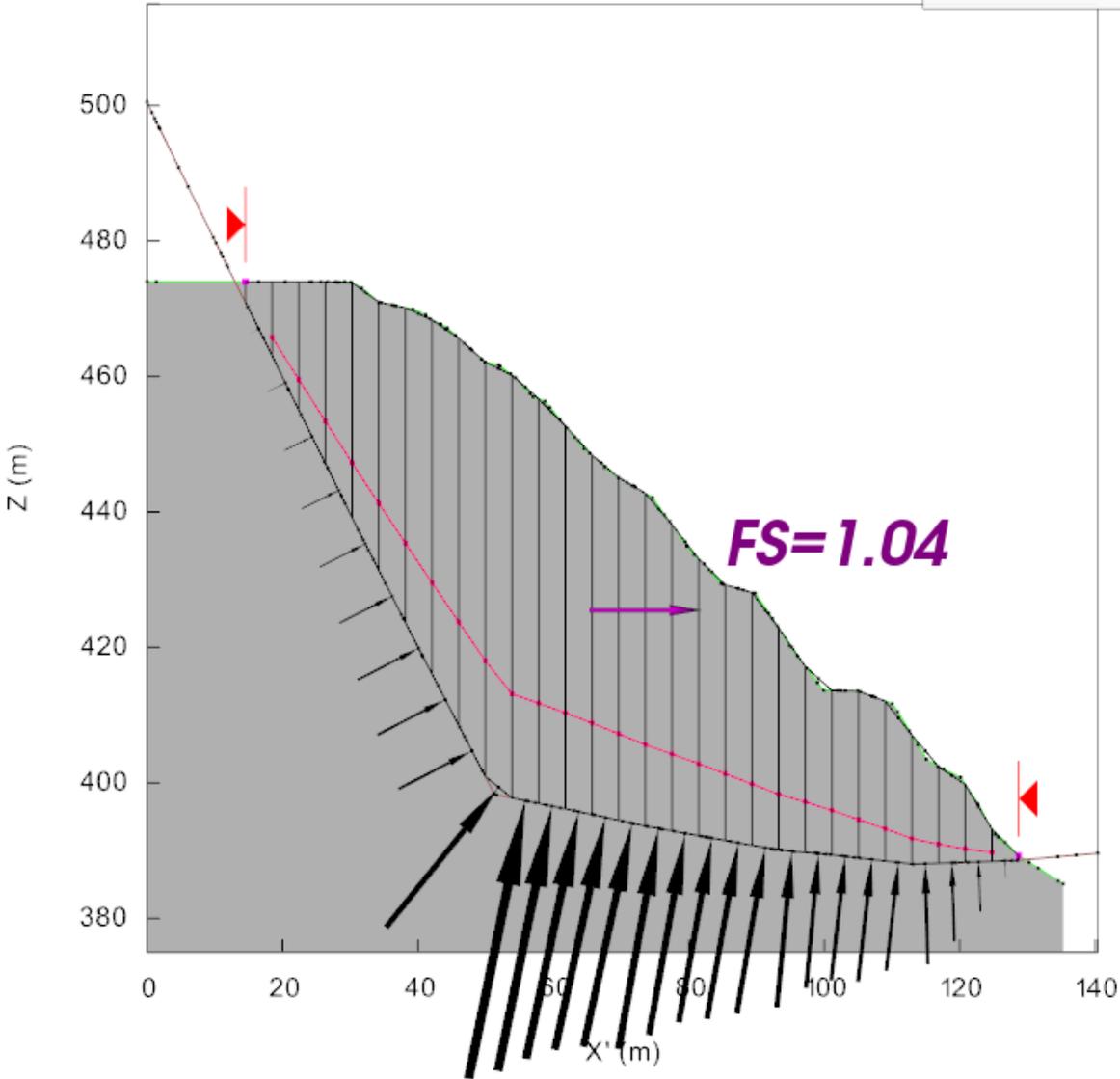


Figure 4.3: Cross section used in the 2D LE analysis.

The same model was constructed in a quasi-2D DEM model, as shown in Fig. 4.4. The sliding mass is represented by slices, or tabular blocks, with a thickness of 4 m, to mimic

the discretization used in the 2D LE analysis. Both models have the same number of sliding blocks. The joints are assumed to be dry. The DEM model allows for the evaluation of the mode of failure for different mechanical properties of the interfaces between the slices and the base of the sliding mass, since the discontinuous nature of the fractured rock mass is explicitly incorporated.

As illustrated in Fig. 4.4, the stability of the model changes as a function of the strength of the inter-slice friction angle. The blocks resting on the back fault apply force on the down-slope blocks, pushing them toward the free face. If the friction angle between the slices is high, the sliding mass acts more as a rigid body, and the mass is stable since interslice friction is too high for the blocks to slide along their interface. When the friction angle between the slices is low, this results in toppling of the slices.

In these analyses, LE and DEM provide comparable results. The equivalent interslice friction angle for a factor of safety of approximately 1 in the LE analysis is 32° . In DEM analysis, the blocks topple when the interslice friction angle is lower than 30° . For interslice friction angle greater than 30° , there is some sliding on the back fault, but there is enough resistance between the blocks, so that they eventually stop. LE analysis, however, cannot capture the transition in failure mode if the friction angle in the rock mass is reduced, as LE methods assume a sliding failure by definition. Moreover, LE cannot obtain a solution when the rock mass slides for some distance and then comes to a halt due to changing block orientations and mobilization of frictional resistance as the rock mass displaces.

Kinematic analysis

In DEM, the mode of failure is not assumed, instead it is derived as part of the analysis and is a function of the strength of the discontinuities, their relative orientation, and the geometry of the slope face. This allowed us to explore possible failure modes for different interslice and bottom failure surface friction angles while maintaining the friction angle on the back fault constant at $\phi = 40^\circ$.

The results show that when the friction angle along the bottom failure surface and between the slices is high, the resistance to sliding is proportionally higher. When the friction angle between the slices and on the bottom failure surface is set to $\phi = 45^\circ$, blocks slide on the back fault, but the sliding mass around the toe of the slope experiences little displacement or rotation due to the high strength and the blocks come to a halt (Fig. 4.5a). In contrast, when the friction angle along the bottom is very low ($\phi = 5^\circ$) and the friction angle between the slices is high ($\phi = 45^\circ$), there is only a little resistance to sliding. The blocks around the toe of the slope can no longer support the weight of the blocks sliding down the back fault and they slide up and over the inclined slope with no noticeable rotation (Fig. 4.5b). In contrast, when the friction angle on the bottom failure surface is high, i.e. $\phi = 45^\circ$ and the inter-slice friction angle is relatively low, $\phi = 25^\circ$, the blocks at the toe of the slope resist sliding, while the blocks further up in the slope topple (Fig. 4.5c). This ability of DEM to capture the transition between potential failure modes along different portions of

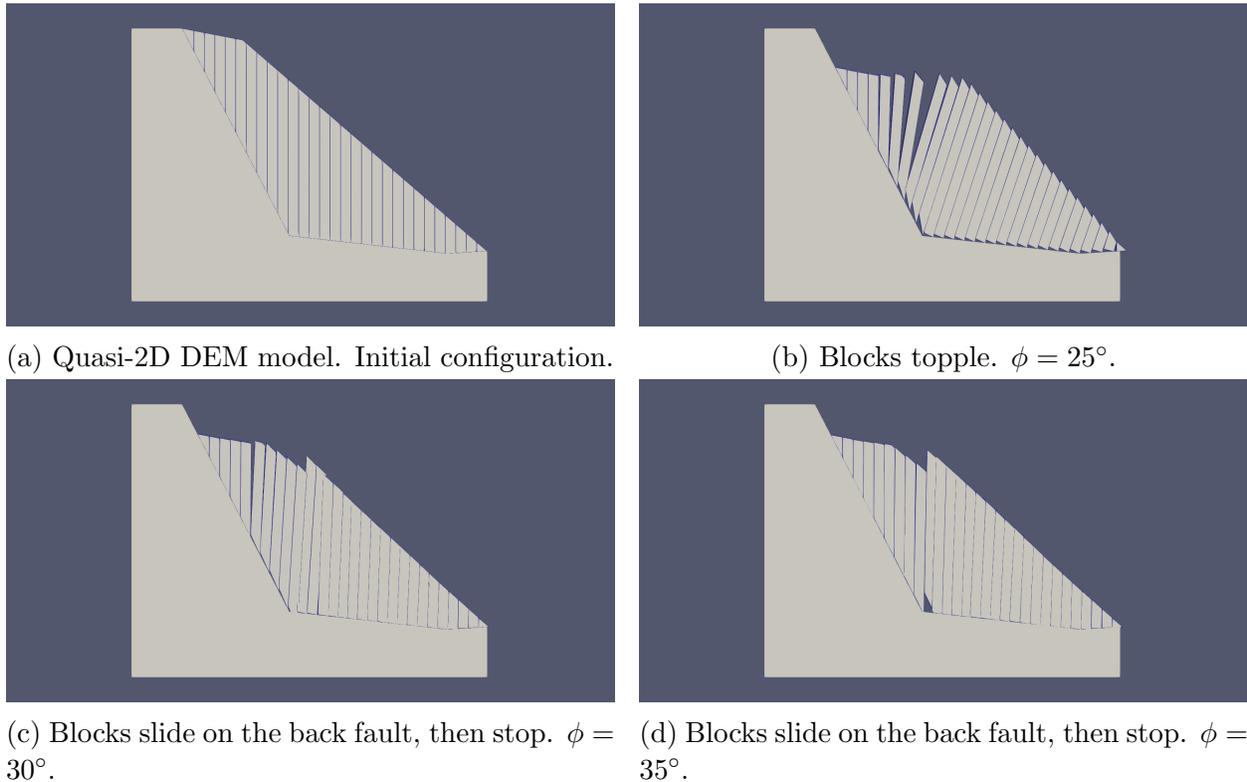


Figure 4.4: Inter-slice friction angle influence on the stability of the 2D model.

the slope sets it apart from LE methods, as sliding failure may not necessarily be the mode that ultimately drives slope stability.

4.5 3D LE and DEM Comparison

The open pit mine wall failure described in Chapter 4.3 was bounded by three faults to form a wedge. 3D LE analyses were performed to investigate the 3D effects of the intersection of the three faults on the stability of the rock mass. In the 3D limit equilibrium analyses, the sliding mass was divided into columns with a square cross section (Fig. 4.6a) for use in Spencer's Method and the failure surface was configured to represent the bounding fault planes (Fig. 4.6b). Using this approach, the required friction angle to maintain stability (factor of safety of unity) was only 9° .

In previous work Sitar et al. (2005) and Boon et al. (2014) showed that comminution of a sliding rock mass into smaller blocks decreased its stability. Therefore, this effect was explored using 3D DEM model, constructed with columns representing the sliding mass to mimic the 3D LE analyses (Fig. 4.7). The spacing between the vertical joints creating the columns was then gradually decreased, such that the number of particles was increased with

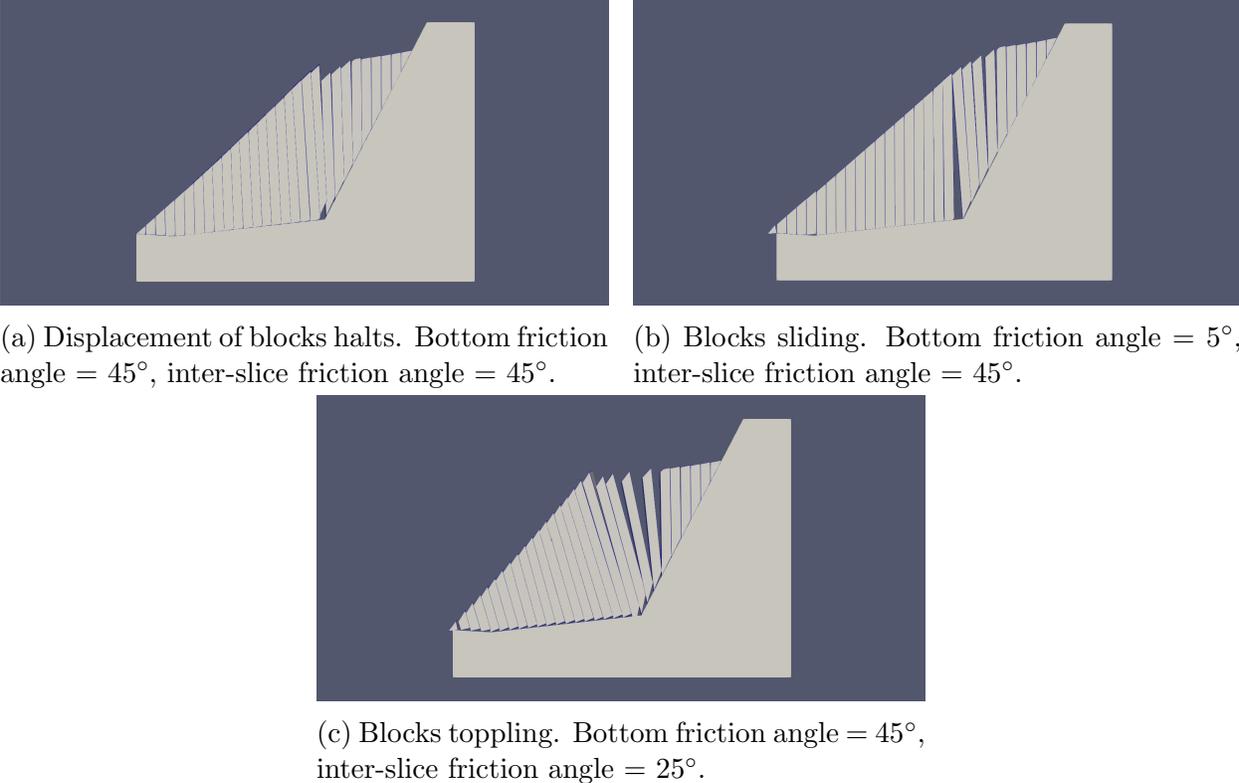


Figure 4.5: Kinematic analysis of quasi-2D DEM model.

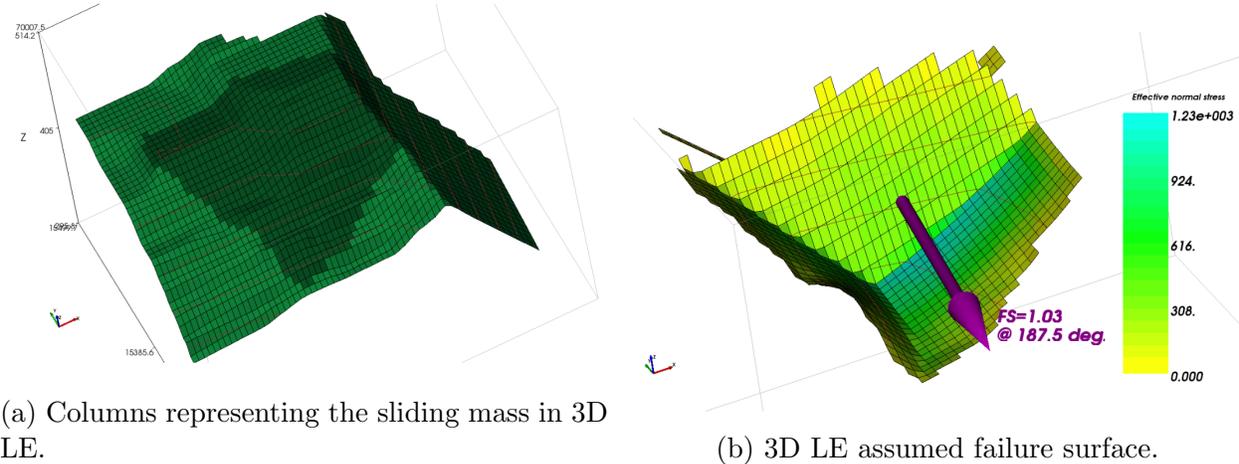


Figure 4.6: 3D LE model.

decreasing joint spacing. The friction angle between the blocks and along the boundaries

required to keep the fractured rock mass stable was then evaluated as a function of the number of columns.

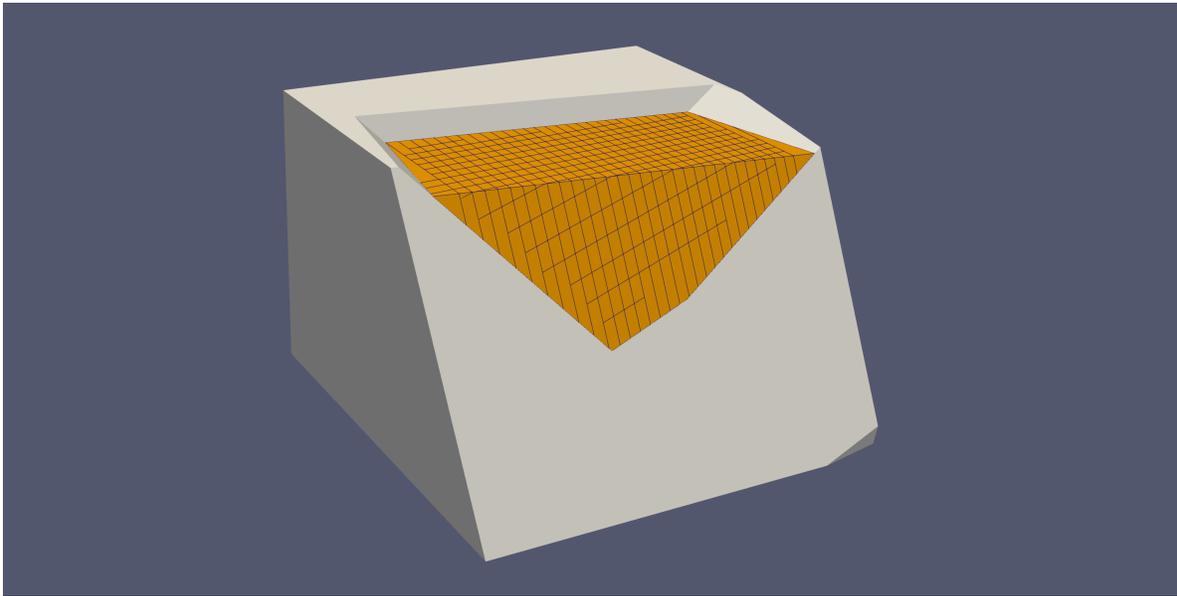


Figure 4.7: 3D DEM model with columns representing the sliding mass.

As expected, the results show that as the number of particles increases, the required friction angle to bring the factor of safety to approximately one increases (Fig. 4.8). A factor of safety of 1 was evaluated by the amount of displacement of the rock mass and its average velocity, that is the rock mass was considered static when its velocity dropped below a threshold value ($1 \times 10^{-4} m/s$). The friction angle required for a factor of safety about 1 in the LE model is also plotted in Fig. 4.8. Compared with the results obtained with DEM, the friction angle required for stability in the LE model is noticeably lower. Specifically, as the number of degrees of freedom in the system increases with increasing number of columns in the DEM models, the sliding mass is less constrained to a particular failure mode. Locally, with tighter spacing of joints, blocks can more easily translate and rotate, exerting forces and moments on their neighbors and consequently affecting the overall stability of the slope.

Additionally, the average displacement of the rock mass was monitored to explore the effect of the number of blocks on the velocity of the rock mass. For a constant friction angle between the blocks and along the boundaries ($\phi = 10^\circ$), as the number of blocks increases, the average displacement of the sliding mass increases. At the limit, models with a few tens of blocks show constant velocity when failing, or creeping, while blocks with hundreds of blocks accelerate quickly (Fig. 4.9).

These findings are consistent with the work by Sitar et al. (2005), who examined the mechanism of the failure of the Vajont Dam using DDA. They found that with an increasing number of particles in the model, the sliding mass is less stable and the velocity of the

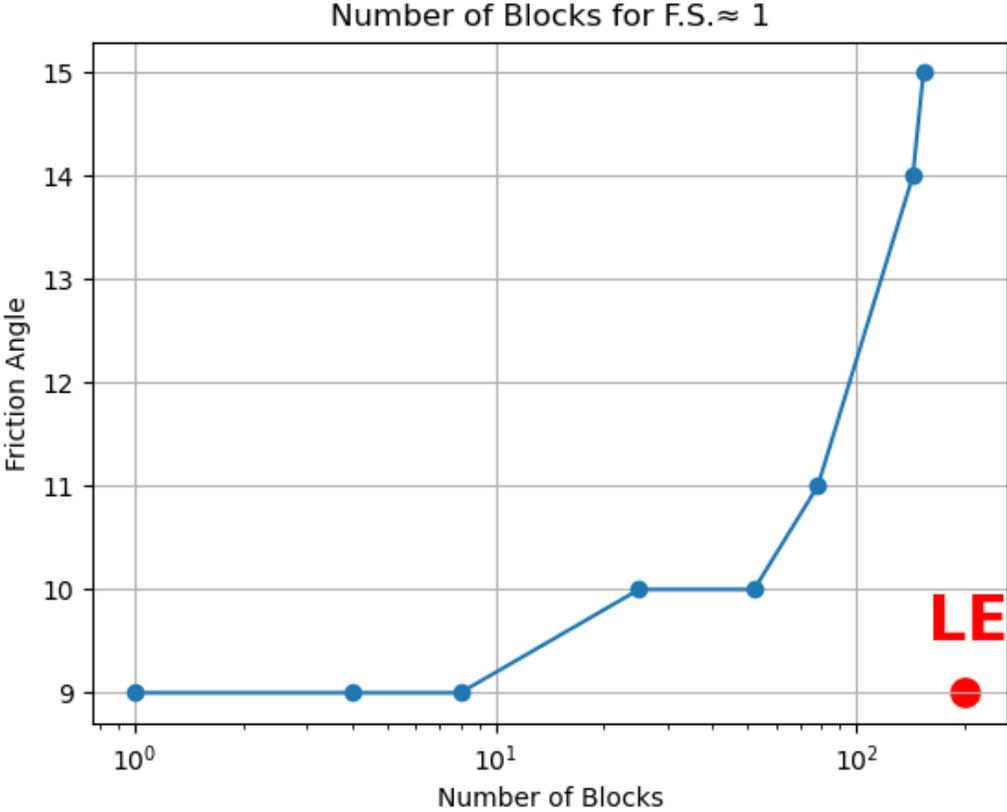


Figure 4.8: Number of particles for F.S.≈1.

sliding mass is greater. This implies that as the sliding mass breaks apart as it displaces, the required frictional resistance to arrest motion will increase. For rocky slopes that experience incremental displacements, it is likely that they will become less and less stable as individual particles within the sliding mass fracture during each displacement increment. Most importantly, no assumptions about the reduction of strength of the discontinuities during sliding are necessary to produce this effect.

4.6 3D DEM analysis of Fractured Rock Mass

In a natural rock mass, the orientation and spacing of the joints are more complex than shown previously. Blocks are different in size and shape and, typically, rock blocks are irregular. To study the influence of natural jointing on the stability of the slope and the initiation of failure, joint orientations based on field measurements were used to create a more realistic representation of the rock mass Chapter 4.5). The spacing between the joints was

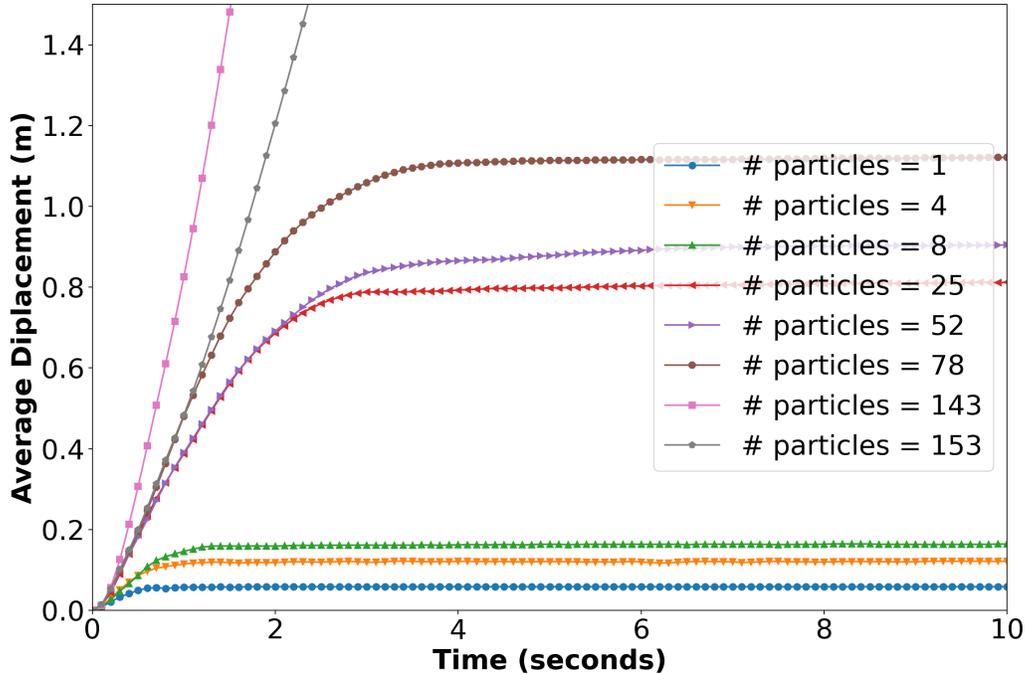


Figure 4.9: Average displacement of columnar blocks.

adjusted to create a model with a number of blocks comparable to the simulations with rock mass divided into columns with the orientation of the joints modeled as reported by Chapple (1998).

In this configuration, which accounts for the natural jointing and structure within the fractured rock mass, the friction angle required to maintain stability is now 15° . In comparison the friction angle required to maintain stability for the same number of columnar blocks in was only 10° . This illustrates how the kinematics, or the shape of the blocks and the interaction of the geometry of the joints and the face of the slope affect the initiation of failure and the overall stability of the slope. With columnar joints, each block rests on the slightly inclined boundary block. With more natural jointing of the rock mass, blocks may be placed on steeper joints, inducing local instabilities that lead to global failure.

in the case of the jointed rock mass with natural joint orientations, the rock mass is stable (Figs. 4.10a & 4.10b) when the friction angle is high enough. However, the amount of displacement required to mobilize sufficient frictional resistance and inter-particle locking depends on the frictional resistance between the blocks. For example, when the friction angle between blocks is 20° , minimal displacement occurs before the sliding mass self-arrests, while for a friction angle of 15° , the average displacement is on the order of 4 m before the sliding

stops. When the friction angle is low (10°), the blocks slide with little resistance, and the rock mass accelerates quickly and slides down the slope to the bottom of the pit (Fig. 4.10c).

Figure 4.11)) shows the average displacement as a function of the friction angle and illustrates the temporal changes in stability and hence the velocity of the slide mass. Here, when the friction angle is high enough for stability (greater than 15°), the slope initially displaces, but then stops. As the friction angle decreases but remains above 11° , the slope initially displaces, but then these displacements begin to slow down. Nevertheless, the slope still creeps and eventually displacements accelerate again until the slope fails. For friction angles below 11° , blocks continue to accelerate and displacements grow steadily. Considering potential fracturing and breakup of the sliding mass and the implications of the number of particles, as shown in Chapter 4.5, cases with a friction angle that allow for substantial displacement may also become unstable if sufficient breakup of particles within the sliding mass does occur. Thus, in DEM careful consideration should be given not just to the amount of frictional resistance required to maintain stability, but also to the amount of displacement that may occur before the slide stops moving.

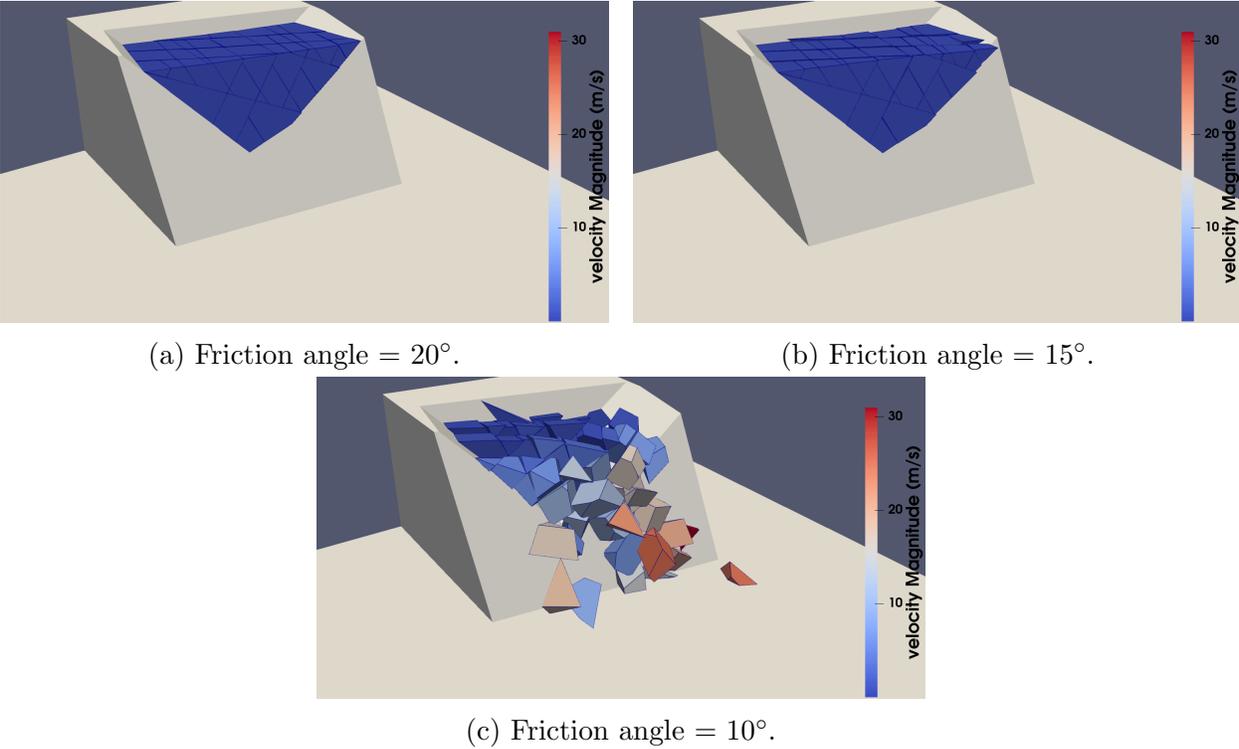


Figure 4.10: 3D DEM model of jointed rock mass.

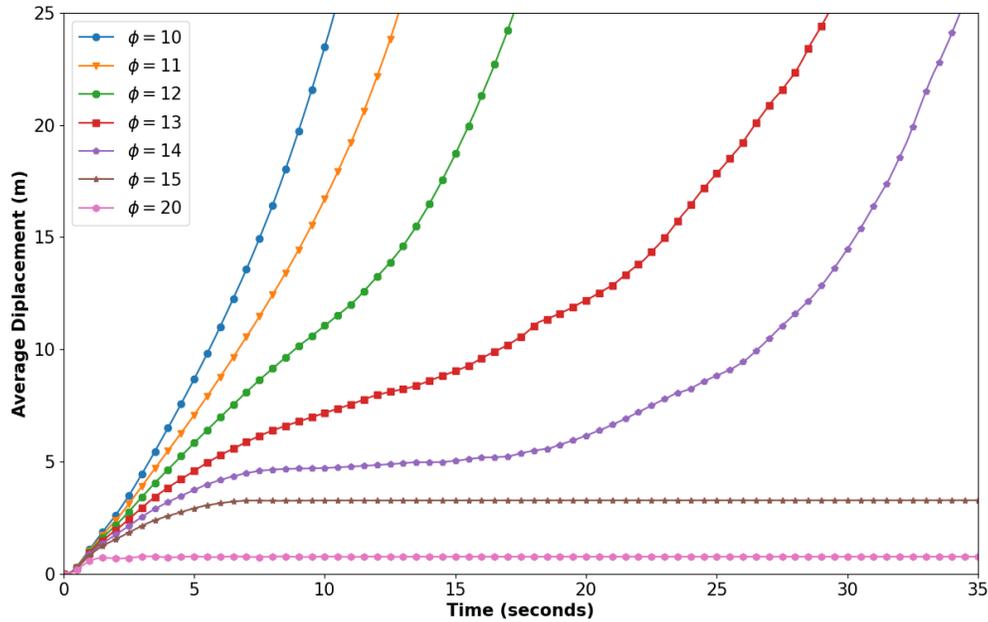


Figure 4.11: Average displacement of the rock mass for different friction angles.

4.7 Conclusion

We investigated rock slope stability and kinematics using both Limit Equilibrium (LE) and the Discrete Element Method (DEM). Our analyses considered both 2D and 3D approaches and using an open-pit mine failure in Round Hill, New Zealand as an example.

In 2D analyses, LE and DEM provided comparable results. The sliding mass remained stable (factor of safety of approximately 1) for the same frictional properties of the failure surface and the equivalent friction between the vertical slices. However, the results illustrate that LE cannot capture the correct failure mode when the mode of failure changes as the friction between the slices changes, since it is constrained to a slide failure mode by definition. In DEM, the mode of failure changes for different friction values along the assumed failure surface and between the slices. Thus, the DEM approach is naturally able to capture the transition between failure modes. This is an essential feature of modeling the kinematic response of a fractured rock mass.

While in 2D LE and DEM gave similar results, in 3D the solutions diverge quite significantly. Again, in LE analysis the failure mode is assumed and it is not derived as part of the analysis. In contrast, in DEM, blocks are allowed to rotate and translate in any direction, applying forces on their neighbors, thus inducing instability. Furthermore, we show that with an increasing number of blocks, the sliding mass is less constrained and requires more

friction to remain stable. This implies that the rock mass will naturally accelerate as blocks break apart during sliding. We also show that for a constant friction angle, a rock mass divided into more columns undergoes more displacement.

Finally, we illustrate the importance of kinematics in rock slope stability by dividing the rock mass into polyhedral blocks defined by joint sets measured in the field. Compared with analyses using columnar blocks, the natural rock mass is less stable since the natural jointing within the rock introduces local planes of instability along which individual particles can displace. The columns utilized in LE analysis and reproduced in DEM for comparison artificially constrain the ways in which the natural rock can move.

Together, these findings illustrate the importance of considering the natural structure within fractured rock and the progressive nature by which rock slopes may fail. We show that when a rock mass is composed of many blocks, the rock mass can accelerate rapidly to failure, while when the number of blocks with the same properties is low, the rock mass displaces slightly but may come to a halt. For an intermediate number of blocks, the rock mass first accelerates, and then the sliding mass creeps down the slope at a relatively constant velocity before accelerating to failure. This highlights an important aspect for rock slope stability analyses in that not only the friction angle for stability should be considered, but the amount of displacement that may occur. If displacements are sufficiently large to induce breakup within the fractured rock mass, careful consideration must be given to how the increased number of particles in the sliding mass may progressively induce failure and an acceleration in slope movement. Most importantly, this behavior is not tied to changes in material properties as such, which was often assumed in analyses using equivalent media approaches.

Chapter 5

Conclusion and Future Research

The primary objective of the research presented in this thesis was to explore the role of kinematics on the stability of jointed rock slopes and the initiation of failure and subsequent runout. Natural joints in a fractured rock mass form irregular polyhedral rock blocks. The interaction of neighboring blocks and with the free surface, determines the response of the rock mass to loading.

First, we have introduced a new algorithm, the Intersection Polygon Method (IPM), for accurately calculating contact points between convex polyhedral particles in Discrete Element Method (DEM) simulations. The primary goal was to address the challenges associated with detecting and calculating precise contact points between colliding polyhedral particles, which are common in many granular material systems and geological settings. The new algorithm offers a significant improvement over existing methods by ensuring higher accuracy in the determination of the contact point. Through rigorous testing, we demonstrate that the algorithm enhances the precision of contact calculations and improves the stability and reliability of DEM simulations involving polyhedral particles. These improvements are crucial for accurately modeling interactions in materials with complex geometries, such as rock blocks, granular media, and other irregularly shaped particles. While more computationally demanding than other algorithms (e.g., GJK-EPA), the improved stability allows the use of larger time steps, making the new algorithm attractive, even for large-scale simulations.

In the next stage of this research, we performed an extensive evaluation of existing contact detection algorithms for polyhedral particles used in DEM, identifying the strengths and weaknesses of each method and comparing their efficiency on a single computational platform. Our comparative analysis, which includes a series of benchmark tests (static and dynamic), was designed to help DEM users make an informed choice of the algorithm most appropriate for their needs.

Finally, we addressed a practical rock mechanics context by exploring the advantages of using DEM for analysis and simulations of rock slope failures. A comparative analysis of Limit Equilibrium (LE) and DEM was performed in two and three dimensions. By capturing the 3D kinematic response of particles involved in slope failures, we show that the rock mass can accelerate as a function of the number of blocks in the sliding mass. Furthermore, we

show that kinematics plays a significant role in the initiation of failure by modeling natural joint sets, as observed in the field. This is critical for applications in geotechnical engineering and rock mechanics, where accurate simulations of particle behavior are essential to predict failure mechanisms and to design safe infrastructure.

5.1 Future Research

While computationally expansive and often not accessible, the implementation of DEM in a modern HPC (High-Performance Computing) environment opens opportunities for efficient and affordable full-scale analyses of failure initiation and runout. The computations can be further accelerated by taking advantage of GPU's. With increasing computing power, we can create more complex models with tens or hundreds of thousands of particles. Stochastic realization of the rock mass can be used to study how a change in the initial configuration of the rock mass affects the velocity, travel distance, and shape of the runout. Furthermore, non-persistent joint sets can be implemented to provide a more realistic representation of the rock mass.

Bibliography

- Martin Bauer, Sebastian Eibl, Christian Godenschwager, Nils Kohl, Michael Kuron, Christoph Rettinger, Florian Schornbaum, Christoph Schwarzmeier, Dominik Thönnies, Harald Köstler, et al. walberla: A block-structured high-performance framework for multiphysics simulations. *Computers & Mathematics with Applications*, 81:478–501, 2021.
- Chia Weng Boon, GT Houlsby, and Stefano Utili. A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method. *Computers and Geotechnics*, 44:73–82, 2012.
- Chia Weng Boon, GT Houlsby, and Stefano Utili. New insights into the 1963 vajont slide using 2d and 3d distinct-element method analyses. *Geotechnique*, 64(10):800–816, 2014.
- IR Brown, PJ Wood, and MK Elmouttie. Estimation of in situ strength from back-analysis of pit slope failure. In PM Dight, editor, *APSSIM 2016: Proceedings of the First Asia Pacific Slope Stability in Mining Conference*, pages 331–338. Australian Centre for Geomechanics, 2016. doi: 10.36487/ACG_rep/1604_18_Brown. URL https://papers.acg.uwa.edu.au/p/1604_18_Brown/.
- Aaron P Chapple. An engineering geological investigation into pit slope stability at macraes gold mine, macraes flat, otago, new zealand. 1998.
- Yung Ming Cheng, T Lansivaara, and WB Wei. Two-dimensional slope stability analysis by limit equilibrium and strength reduction methods. *Computers and geotechnics*, 34(3): 137–150, 2007.
- Chhunla Chheng and Suched Likitlersuang. Underground excavation behaviour in bangkok using three-dimensional finite element method. *Computers and Geotechnics*, 95:68–81, 2018.
- Deepankar Choudhury, Somdatta Basu, and Jonathan D Bray. Behaviour of slopes under static and seismic conditions by limit equilibrium method. In *Embankments, Dams, and Slopes: Lessons from the new orleans levee failures and other current issues*, pages 1–10. 2007.

- Peter 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. In *International journal of rock mechanics and mining sciences & geomechanics abstracts*, pages 107–116. Elsevier, 1988.
- Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *geotechnique*, 29(1):47–65, 1979.
- H-JG Diersch, D Bauer, W Heidemann, Wolfram Rühaak, and Peter Schätzl. Finite element modeling of borehole heat exchanger systems: Part 1. fundamentals. *Computers & Geosciences*, 37(8):1122–1135, 2011.
- James Michael Duncan. State of the art: limit equilibrium and finite-element analysis of slopes. *Journal of Geotechnical engineering*, 122(7):577–596, 1996.
- James Michael Duncan, Richard E Goodman, et al. Finite element analysis of slopes in jointed rock. 1968.
- YT Feng and DRJ Owen. An augmented spatial digital tree algorithm for contact detection in computational mechanics. *International Journal for Numerical Methods in Engineering*, 55(2):159–176, 2002.
- YT Feng and DRJ Owen. A 2d polygon/polygon contact model: algorithmic aspects. *Engineering Computations*, 21(2/3/4):265–277, 2004.
- YT Feng and Yuanqiang Tan. On minkowski difference-based contact detection in discrete/discontinuous modelling of convex polygons/polyhedra: Algorithms and implementation. *Engineering Computations*, 37(1):54–72, 2020.
- YT Feng, K Han, and DRJ Owen. Energy-conserving contact interaction models for arbitrarily shaped discrete elements. *Computer methods in applied mechanics and engineering*, 205:169–177, 2012.
- Erfan G. Nezami, Youssef MA Hashash, Dawei Zhao, and Jamshid Ghaboussi. Shortest link method for contact detection in discrete element method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(8):783–801, 2006.
- Oliver Gaede, Florian Karpfinger, Jeroen Jocker, and Romain Prioul. Comparison between analytical and 3d finite element solutions for borehole stresses in anisotropic elastic rock. *International journal of rock mechanics and mining sciences*, 51:53–63, 2012.
- Michael Gardner and Nicholas Sitar. Modeling of dynamic rock–fluid interaction using coupled 3-d discrete element and lattice boltzmann methods. *Rock Mechanics and Rock Engineering*, 52:5161–5180, 2019.

- Elmer G Gilbert, Daniel W Johnson, and S Sathiya Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, 1988.
- Richard E Goodman and D Scott Kieffer. Behavior of rock in slopes. *Journal of geotechnical and geoenvironmental engineering*, 126(8):675–684, 2000.
- DV Griffiths and Gordon A Fenton. Probabilistic slope stability analysis by finite elements. *Journal of geotechnical and geoenvironmental engineering*, 130(5):507–518, 2004.
- R Hart, Peter 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. In *International journal of rock mechanics and mining sciences & geomechanics abstracts*, volume 25, pages 117–125. Elsevier, 1988.
- David A Horner, John F Peters, and Alex Carrillo. Large scale discrete element modeling of vehicle-soil interaction. *Journal of engineering mechanics*, 127(10):1027–1032, 2001.
- GT Houlsby. Potential particles: a method for modelling non-circular particles in dem. *Computers and Geotechnics*, 36(6):953–959, 2009.
- Itasca Consulting Group Inc. 3dec—theory and background, 2016.
- Y Keissar, M Gardner, and N Sitar. A new algorithm for accurate contact point calculation between convex polyhedral particles in dem. in review.
- Dinant Krijgsman, Vitaliy Ogarko, and Stefan Luding. Optimal parameters for a hierarchical grid data structure for contact detection in arbitrarily polydisperse particle systems. *Computational particle mechanics*, 1(3):357–372, 2014.
- Andrews Munjiza and KRF Andrews. Nbs contact detection algorithm for bodies of similar size. *International Journal for Numerical Methods in Engineering*, 43(1):131–149, 1998.
- Erfan G Nezami, Youssef MA Hashash, Dawei Zhao, and Jamshid Ghaboussi. A fast contact detection algorithm for 3-d discrete element method. *Computers and geotechnics*, 31(7):575–587, 2004.
- Ruaidhrí Manfred O’Connor. *A distributed discrete element modeling environment: algorithms, implementation and applications*. PhD thesis, Massachusetts Institute of Technology, 1996.
- Eric Perkins and John R Williams. A fast contact detection algorithm insensitive to object sizes. *Engineering Computations*, 18(1/2):48–62, 2001.
- Gen-Hua Shi. Discontinuous deformation analysis—a new model for the statics and dynamics of block systems (ph. d. thesis). *University of California, Berkeley, USA*, 1988.

- Gen-hua Shi. *Working forum on manifold method of material analysis*. 1997.
- GenHua Shi. Contact theory. *Science China Technological Sciences*, 58:1450–1496, 2015.
- Nicholas Sitar, Mary M MacLaughlin, and David M Doolin. Influence of kinematics on landslide mobility and failure mode. *Journal of geotechnical and geoenvironmental engineering*, 131(6):716–728, 2005.
- Kenichi Soga, E Alonso, Alba Yerro, K Kumar, and Samila Bandara. Trends in large-deformation analysis of landslide mass movements with particular emphasis on the material point method. *Géotechnique*, 66(3):248–273, 2016.
- Ef Spencer. A method of analysis of the stability of embankments assuming parallel interslice forces. *Geotechnique*, 17(1):11–26, 1967.
- Deborah Sulsky, Zhen Chen, and Howard L Schreyer. A particle method for history-dependent materials. *Computer methods in applied mechanics and engineering*, 118(1-2):179–196, 1994.
- Deborah Sulsky, Shi-Jian Zhou, and Howard L Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer physics communications*, 87(1-2):236–252, 1995.
- Gino Van Den Bergen. Proximity queries and penetration depth computation on 3d game objects. In *Game developers conference*, volume 170, page 209, 2001.
- John R Williams, Eric Perkins, and Ben Cook. A contact algorithm for partitioning n arbitrary sized objects. *Engineering Computations*, 21(2/3/4):235–248, 2004.
- Hong Zhang, Guangqi Chen, Lu Zheng, Zheng Han, Yingbin Zhang, YanQiang Wu, and Shuguang Liu. Detection of contacts between three-dimensional polyhedral blocks for discontinuous deformation analysis. *International Journal of Rock Mechanics and Mining Sciences*, 78:57–73, 2015.
- Hong Zhang, Shu-guang Liu, Zheng Han, Lu Zheng, Ying-bin Zhang, Yan-qiang Wu, Yan-ge Li, and Wei Wang. A new algorithm to identify contact types between arbitrarily shaped polyhedral blocks for three-dimensional discontinuous deformation analysis. *Computers and Geotechnics*, 80:1–15, 2016.
- Fei Zheng, Yu-Yong Jiao, Michael Gardner, and Nicholas Sitar. A fast direct search algorithm for contact detection of convex polygonal or polyhedral particles. *Computers and Geotechnics*, 87:76–85, 2017a. ISSN 0266-352X. doi: <https://doi.org/10.1016/j.compgeo.2017.02.001>. URL <https://www.sciencedirect.com/science/article/pii/S0266352X17300277>.

- Fei Zheng, Yu-Yong Jiao, Xiu-Li Zhang, Fei Tan, Long Wang, and Qiang Zhao. Object-oriented contact detection approach for three-dimensional discontinuous deformation analysis based on entrance block theory. *International Journal of Geomechanics*, 17(5): E4016009, 2017b.
- Jorge G Zornberg, Nicholas Sitar, and James K Mitchell. Limit equilibrium as basis for design of geosynthetic reinforced slopes. *Journal of Geotechnical and Geoenvironmental Engineering*, 124(8):684–698, 1998.

Appendix A

3D edges intersection

The equations for the two edges are:

$$p_a = p_1 + t(p_2 - p_1) \quad (\text{A.1a})$$

$$p_b = p_3 + u(p_4 - p_3) \quad (\text{A.1b})$$

Where p_a and p_b are some points on the lines. The distance between these two points is:

$$f = \|p_a - p_b\|^2 \quad (\text{A.2})$$

Substituting the equations of the lines gives:

$$f = \|p_1 - p_3 + t(p_2 - p_1) + u(p_4 - p_3)\|^2 \quad (\text{A.3})$$

$$= \|l_{13} + tl_{21} + ul_{43}\|^2 \quad (\text{A.4})$$

Where $l_{13} = p_1 - p_3$, $l_{21} = p_2 - p_1$, $l_{43} = p_4 - p_3$. Define:

$$F \stackrel{\text{def}}{=} l_{13} + tl_{21} + ul_{43} \quad (\text{A.5})$$

$$f = F \cdot F \quad (\text{A.6})$$

To find the point of intersection between the two line segments, the derivative of f with respect to t and u is zero:

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial t} (F \cdot F) = 2 \frac{\partial F}{\partial t} \cdot F = 0 \quad (\text{A.7})$$

$$l_{21} \cdot (l_{13} + tl_{21} + ul_{43}) = 0 \quad (\text{A.8})$$

$$t = \frac{(l_{21} \cdot l_{43})u - (l_{21} \cdot l_{13})}{(l_{21} \cdot l_{21})} \quad (\text{A.9})$$

$$\frac{\partial f}{\partial u} = \frac{\partial}{\partial u} (F \cdot F) = 2 \frac{\partial F}{\partial u} \cdot F = 0 \quad (\text{A.10})$$

$$-l_{43} \cdot (l_{13} + tl_{21} - ul_{43}) = 0 \quad (\text{A.11})$$

To find u , substitute t :

$$u(l_{43} \cdot l_{43}) - (l_{43} \cdot l_{21}) \left(\frac{(l_{43} \cdot l_{21})u}{(l_{21} \cdot l_{21})} - \frac{(l_{21} \cdot l_{13})}{(l_{21} \cdot l_{21})} \right) = (l_{43} \cdot l_{13}) \quad (\text{A.12})$$

$$u = \frac{(l_{43} \cdot l_{13})(l_{21} \cdot l_{21}) - (l_{43} \cdot l_{21})(l_{43} \cdot l_{21})}{(l_{43} \cdot l_{43})(l_{21} \cdot l_{21}) - (l_{43} \cdot l_{21})(l_{43} \cdot l_{21})} \quad (\text{A.13})$$