UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Repositorio Digital USM **https://repositorio.usm.cl**

Tesis USM TESIS de Postgrado de acceso ABIERTO

2016

A MATHEMATICAL AND COMPUTATIONAL MODEL FOR MULTIPLE COLLISIONS OF RIGID BODIES: ANEXTENSION OF A-CD2 METHOD

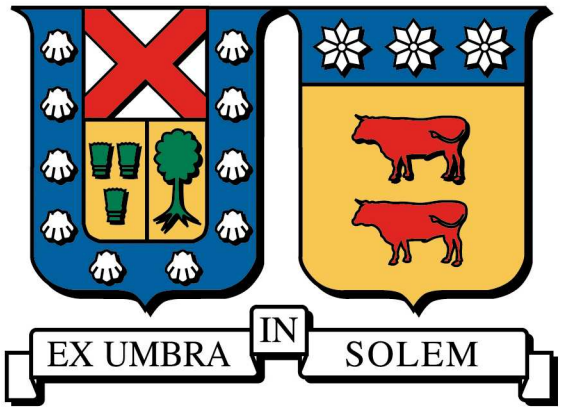
LEON VASQUEZ, ROBERTO JESUS

http://hdl.handle.net/11673/13966

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Universidad Técnica Federico Santa María

Departamento de Informática



A Mathematical and Computational Model for Multiple Collisions of Rigid Bodies: An Extension of A-CD^2 Method

Thesis by

Roberto Jesús León Vásquez

In Partial Fullfillment of the Requeriments for the Degree of

Doctor in Informatic Engineering

Valparaíso, Chile

July, 2016

Title of Thesis

A Mathematical and Computational Model for Multiple Collisions of Rigid Bodies: An Extension of A-CD^2 Method

Author

Roberto Jesús León Vásquez

In Partial Fullfillment of the Requeriments for the Degree of Doctor in Informatic Engineering of Universidad Técnica Federico Santa María

Luis Salinas Ph.D.

advisor, utfsm

Claudio Torres Ph.D.

co-advisor, utfsm

Eric Dimnet Ph.D.

co-advisor, ifsttar

Mauricio Solar Ph.D.

internal member, utfsm

Rodrigo Soto Ph.D.

external national member, uchile

Mario Alberto Storti Ph.D.

external international member, unl

Hernán Astudillo Ph.D.

committe president, utfsm

Valparaíso, Chile July 13th, 2016

This thesis is dedicated to my family:

Karina, Leonor and Lucas.

Acknowledgements

I want to thank my wife for her unconditional love and support in this process. Also for encouraging me when I was fallen and be a mainstay in my life. Thanks to my children, Leonor and Lucas, they are my core and my force to keep going.

Thanks to my mentor Dr. Luis Salinas, for his guide, support and dedication in this work. For his countless advices and suggestions. To Dr. Claudio Torres, for his dedication in this work, for the countless meetings, suggestions and also advices. To Dr. Eric Dimnet, for his dedication in this work, for his suggestions and advices, and also for his support in my stay at Paris and thanks to my collegue, friend and guide Dr. Gonzalo Hernández, for his constant support, friendship and advices.

Thanks to my wife's parents, Marta y Juan Carlos, for his support, advices and being part in this process. To my best friends Leticia and Sebastián, for his support and friendship.

This research was partially supported by: CONICYT grant "Doctorado en Chile", MECESUP grant "Doctorado- Programa Nacional" and the project BASAL FB0821 CCTVal.

Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02).

Abstract

An extension of a mathematical model and a computational simulation for multiple collisions of rigid bodies is presented in this work.

The A-CD^2 method gives a mechanical description for instantaneous collisions between rigid bodies. The solution of a constrained optimization problem is required for obtaining the new velocities of the rigid bodies after collision.

The extension includes the use of the Euler equations for modeling the angular velocities, this produces a non constant angular velocities when the moments of inertia are different. If the moments of inertia are equal, we obtain the constant angular velocity solution.

The extension considers also a reduction in the computational complexity of the original algorithm. The computational complexity has been reduced from an $\mathcal{O}(N^2)$ algorithm to an $\mathcal{O}(N)$ algorithm. This reduction allows to handle problem 20 times larger than the original implementation.

Finally, numerical simulations for a shock-absorber structure, a rockslide real event and granular layers motion are presented.

Keywords: rigid body collision, contact detection, computational complexity, Jacobi elliptic function

Resumen

Una extensión de un modelo matemático y simulaciones computacionales para colisiones múltiples de cuerpos rígidos es presentada en este trabajo

El método A-CD^2 entrega una descripción mecánica para colisiones instantáneas entre cuerpos rígidos. La solución de un problema de optimización con restricciones es requerida para obtener las nuevas velocidades de los cuerpos rígidos después de una colisión.

La extensión a este método incluye el uso de las ecuaciones de Euler para modelar las velocidades angulares, esto produce velocidades angulares no constantes cuando los momentos de inercia son distintos. Si los momentos de inercia son iguales, se obtienen velocidades angulares constantes.

Esta extensión considera también una reducción en la complejidad computacional del algoritmo original. La complejidad computacional ha sido reducido de un algoritmo $\mathcal{O}(N^2)$ a un algoritmo $\mathcal{O}(N)$. Esta reducción permite manejar problemas 20 veces más grandes que la implementación original.

Finalmente se presentan simulaciones numéricas sobre una estructura granular, un evento real de una avalancha y la constitución de capas granulares.

Palabras claves: colisión de cuerpos rígidos, detección de contactos, complejidad computacional, funciones elípticas de Jacobi

Contents

Acknowledgements v

Abstract vii

Resumen ix

List of Tables xv

List of Figures xvii

Chapter 1. Introduction 1

1.1. Granular Material 2

1.2. Collision Model 5

1.3. Contact Detection 17

Chapter 2. A-CD^2 Model 29

2.1. Rigid Bodies Collisions 29

2.2. A-CD^2 Approach 33

Chapter 3. Numerical Model 49

3.1. Matrix Representation 49

3.2. Torque and Force Free Rigid Box Motion 59

3.2.1. First Solution 68

3.2.2. Second Solution 72

3.2.3. Comparison 78

3.2.4. Numerical Results 85

3.3. Contact detection 96

3.3.1. Surface contact detection 99

3.3.2. Face contact detection 100

3.3.3. Edge contact detection 101

3.3.3.1. Starting Situation 102

3.3.3.2. The local reference system 102

3.3.3.3. Checking whether some edges of V intersect box U 104

Chapter 4. Computational Model 107

4.1. Main Procedure 107

4.2. First version 109

4.3. Second version 113

Chapter 5. Computational resources 123

5.1. Levque Cluster 123

5.2. Leftraru Cluster 126

5.3. Visualization Module 128

Chapter 6. Applications: results and comparison 133

6.1. Study of a shock - absorber structure 133

6.1.1. Description of the impact station 133

6.1.2. Impacted structure 134

6.1.3. Instrumentation 135

6.1.4. Impacts 136

6.1.5. Mechanical model 137

6.1.6. Numerical computations and comparison 143

6.2. Study of a rockslide real event 156

6.2.1. Data set acquisition 157

6.2.2. Numerical Simulation 161

6.3. Granular layers 170

Chapter 7. Conclusions 175

Bibliography 177

Appendix A. Subdifferential calculus 193

A.1. Definitions 193

List of Tables

3.1	Jacobi differential equations	69
3.2	Comparing term by term with candidate solutions	70
3.3	Coefficients used to find the analytical solutions for $\omega_1(t)$, $\omega_2(t)$, and $\omega_3(t)$.	72
3.4	Set of initial conditions.	86
5.1	Characteristics of Leftaru.	127
5.2	Modules available in Leftaru	128
6.1	CPU time for impacts.	155

List of Figures

1.1 Complex geometry examples. 4

1.2 Two disc compressed between rigid walls; (a) $t = t_0$, (b) $t = t_1 = t_0 + \Delta t$, (c) $t = t_2 = t_1 + \Delta t$, see [**42**] 6

1.3 Contact force, see [**85**] 9

1.4 The system at collision time: the disks are seen from above, see [**27**]. 10

1.5 The positions of the rigid rod at different times between the first and second impact, see [**152**]. 15

1.6 A simple "potential particle" for two and three dimensions, see [**74**]. 16

1.7 Examples of block mapping to cell space in two dimensions, see [**40**] 18

1.8 Multilevel of representations, see [**123**] 22

1.9 Contact detection problem, see [**114**] 23

1.10 Space decomposition, see [**114**] 25

1.11 Common plane (CP) between two particles: (a) particles in contact, both particles intersect the CP; and (b) particles not in contact, neither particle intersects the CP, see [**119**]. 26

1.12 Ellipsoids A and B are nearest neighbors, and they may collide among them before colliding with their OBP; neither A or B can collide with C during this time, see [**102**] 27

2.1 The point $\boldsymbol{x_1}(t)$ and the solid $\Omega(t)$ move and collide. The outwards normal vector to $\Omega(t)$ is $\boldsymbol{N}$, proj$\boldsymbol{x_1}$ is the projection of $\boldsymbol{x_1}$ on the boundary $\partial\Omega$ of Ω [**55**]. 30

2.2 Point - plane system: a particle with mass m and a fixed plane. 34

2.3 Definition of reactive percussion 37

2.4 Numerical simulation of a landslide. 48

2.5 System with 1225 solids 48

3.1 The global coordinate system $\boldsymbol{O}XYZ$ and the local orthogonal coordinate system $\boldsymbol{G}xyz$ 62

3.2 Evolution from t to $\Delta t = t + 1$ for the local coordinate system. 65

3.3 Cases A, B, C and D solved with numerical integration method and analytical solution 86

3.4 Cases E, F, G and H solved with numerical integration method and analytical solution 88

3.5 Trajectory for a rigid solid with constant rotational velocity. 90

3.6 Trajectory for a rigid solid with non constant rotational velocity. 91

3.7 Solution for $\omega(t)$ in [**182**] 92

3.8 Solution for $\omega(t)$ with our proposal. 92

3.9 Solution for $I_{22} = I_{33}$, where $\omega_1(t)$ is constant. 94

3.10 Solution for $I_{11} = I_{22}$, where $\omega_3(t)$ is constant. 94

3.11 Graph for $\tanh(t)$ and $\text{sech}(t)$. 96

3.12 Types of bounding volumes, see [**53**]. 97

3.13 A box 98

3.14 Ground contact 100

3.15 Contact Edge Detection: first version 101

3.16 Edge contact 106

4.1 Computational model for a numerical simulation. 108

4.2 Block Matrix Information 120

5.1 Levque Cluster Network 124

5.2 Levque hardware 126

5.3 Leftaru Cluster 127

6.1 Tested sandwich structure 134

6.2 CER pendular impact facility. Source [**145**] 134

6.3 Instrumentation of the structure (a: accelerometers, f: force sensors). Source [**145**] 135

6.4 Accelerometers and sensors 136

6.5 Shape of the grains 137

6.6 Gabion with 216 grains 138

6.7 The force applied to the grain is proportional to the part of its volume outside the boundary of the gabion (left); the force is null if the grain is inside the gabion (right) 139

6.8 The initial position of the ball and the 9 gabions. The transparent yellow region is the volume of sand. 140

6.9 Unilateral boundary condition (black zone) 141

6.10 Winkler - Westergaard boundary condition. 142

6.11 Impacts at different level of energy. 143

6.12 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $f_1(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 145

6.13 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $f_2(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 146

6.14 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $f_3(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 147

6.15 Comparison between experimental (blue continuos) and numerical (red dashed) signals of $f_4(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max} 148

6.16 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_0(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 150

6.17 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_2(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 151

6.18 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_3(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 152

6.19 Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_4(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 153

6.20 Left: Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_1(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. Right: Comparision between experimental (blue) and numerical (red hatched) signals peaks of $a_1(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 154

6.21 Comparison between experimental (blue) and numerical (red hatched) displacements for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. 155

6.22 Collapse of the Piton Marcel in Martinique, France. 157

6.23 A sketch of an airborne LiDAR survey 158

6.24 3D LiDAR points before the collapse (IGN survey displayed in height color scale) and after the collapse (Helilmap survey - blue points) 159

6.25 Height difference between both point clouds coded in a polar scale (blue to red) and display in the same perspective as in Figure 6.24. Height

difference comes from 0[m] to 60[m]. Grey points lie outside the collapse area. 160

6.26 Location of the initial set of block within the LiDAR Helimap data set. 160

6.27 A rock represented by a regular polyhedron with dimensions a, b and c. 161

6.28 Initial positions of the numerical rock avalanche. Display made by VPython module visualization. 163

6.29 Kinetic energy with 435 solids. 165

6.30 Dissipative energy with 435 solids. 166

6.31 An avalanche numerical simulation with 435 blocks. 167

6.32 An avalanche numerical simulation with 874 blocks. 168

6.33 Comparison between real and numerical final deposits. The blue curve delimitates the surface considered in the numerical simulation, and the red curve delimitates the greater concentration of material in the final deposit. 169

6.34 Evolution of 500 solids falling. 171

6.35 Evolution of 1000 solids falling. 171

6.36 Evolution of 3000 solids falling. 172

6.37 Evolution of 500 solids forming a heap. 173

6.38 Evolution of 1000 solids forming a heap. 174

6.39 Evolution of 3000 solids forming a heap. 174

7.1 Computational time comparison between the original version (red dashed) and the new one (blue continuous) for different amounts of solids. 176

CHAPTER 1

Introduction

Collision between rigid bodies is an important class of process arising in several natural events. For instances, avalanches, molecules and asteroids collisions, etc. And also in many important engineering and industrial problems e.g., mining, construction, transport, among others.

Thus, it is important to develop computational models for collision simulations in order to avoid experiments, which usually requires resources and equipments that are expensive. A computational simulation allows to considerably reduce the use of special equipments and other resources, and still allowing to obtain a very close description of the dynamics of the phenomenon, with the goal of analyzing and studying the evolution of collisions in time.

This thesis followed the approach presented in **[132, 46]**, named A-CD^2. This method obtain the new velocities of a set of solids, after collision, solving a constrained minimization problem. Currently this method allows us to simulate a system with no more than 1500 solids. The problem is mainly computational: the contact detection algorithm has computatinal complexity $\mathcal{O}(n^2)$, where n is the number of solids in the system, and the algorithm used for the constrained minimization problem, is not the best one for convex optimization. In this thesis an efficient computational method is presented, where an algorithm with linear computational

complexity is achieved, and an interior point method for the constrained optimization problem is used. Now, simulations with more than 5000 solids is achieved, and the computational time has been decreased considerably. Furthermore, the A-CD^2 approach, consideres after collision, the solid evolution with constant rotational velocity, which physically does not obey to classical mechanics equations. In this thesis a set of new equations is presented to obtain the rotational velocities of a solid in torque free motion. Several applications are presented in this work: impacts over gabions (a grid with grains), where the energy curves obtained experimentally and numerically are compared, simulations of an avalanche ocurred in Martinique also is presented, where the final deposit is compared with the real images obtained, and simulations for the constitution of compact grain layers is obtained, where the solids is concentrated in a point with the objective to have a layer of rigid bodies in equilibrium. All these simulations are visualized by means of a Python script developed during this work.

1.1. Granular Material

A granular material is a collection of solid particles or grains, such that most of the particles are in contact with at least some of their neighboring particles. The term "granular materials" and "particulate solids" are interchangeably in the literature. Common examples of granular materials are sand, gravel, food grains, seeds, sugar, coal, and cement for instance [**84**].

Granular materials are commonly encountered in nature and in various industries, that is why the importance of understanding the mechanic and behavior. This is

still an open problem. Nevertheless, some progress has been made during the past few decades.

An interesting scope of granular materials, is when they are in a state of *flowing* (particle - particle interaction are a dominant feature [**23**]). There are several contributions about experimental observations and models related to their behavior. Application of discrete models to problems of granular flow have been initiated by Cundall [**37, 38, 42**]. Here Newton's laws are applied to each particle, and its motion is followed in time. This approach is named as *distinct element method* (DEM). Another applications with these approach appears in [**78, 179, 51**]. In [**68**] individual grains are treated as the "molecules" of a granular "fluid".

In [**84**], is shown that granular material exhibits many features which can not be anticipated from the experience with fluids such as air or water. In fact, unlike water, granular materials are compressible in the sense that the space between the particles often changes during flow.

The theory of granular materials is among the most interesting and intensively developing fields of mechanics [**147**] because the area of its application is very wide. It involves problems of mechanics of geomaterials such as soils and rocks. These problems are related to stability of mine openings, transportation of granular materials of minerals industry and agriculture production, design of storage bunkers and grain tanks, modeling of avalanching, among others.

(a) Hopper

(b) Heap

FIGURE 1.1. Complex geometry examples.

Now, why is it interesting to study granular materials in flow? Dense and fluid, like flows of granular materials, are involved in a huge variety of natural and industrial processes. In industry, one is often concerned with the capacity of grains or pastes to flow without blockage through complex geometries. For instance, hoppers (a storage container used to dispense granular materials through the use of a chute to restrict flow, sometimes assisted by mechanical agitation), pipes (a tubular section or hollow cylinder, usually but not necessarily of circular cross-section, used mainly to convey substances which can flow), among others (see Figure 1.1).

It is important to be clear about the study in granular materials, when there are particles in a certain configuration, as sand in a pipe, arise the concept of a granular material in rest. This implies to study all the factors which are involved in the equilibrium of this system. On the other hand, if the granular material moves, for instance, an extreme of the pipe is opened and the sand start to fall down. In this case arise the concept of a granular material in flow, or a granular flow. An interesting definition is presented in **[25]**, where the adjective "granular" is attributed to materials when they are made of sets of unfastened discrete solid particles (granules)

of a size larger than one micron (micrometre), a length scale where thermal agitation is negligible.

The research about granular flows (materials) is wide [**121, 150, 116, 151, 50, 140, 156, 44, 180, 181, 175, 28, 122, 136, 148, 31, 137**].The main studies is focused in the description of models that explain the behavior of the evolution of the particles; in fact, this behavior continues to be under research, because is not completely understood yet.

1.2. Collision Model

The numerical model for describing the behavior of colliding particles has been studied for many years [**37, 42**]. Cundall proposed the *distinct element method* (DEM), as a numerical model capable of describing the mechanical behavior of assemblies of discs and spheres. The method is based in the use of an explicit numerical scheme. The interaction of the particles is handled contact by contact and the motion of the particles modelled particle by particle, in fact, the computer program BALL was developed for modelling two-dimensional assemblies [**39**].

In DEM, the equilibrium contact forces and displacements of a stressed assembly of discs are found by means of the movements of each particle. These movements are the result of the propagation of disturbances through the medium originated at the boundaries. And the speed of propagation of the disturbances is defined by the physical properties of the medium.

The velocities and accelerations are estimated to be constant over a small period of time, i.e. a time step. The time step chosen may be so small that, perturbations can not propagate from any disc further than its immediate neighbours. Therefore,

the resultant forces on any disc are determined by its interaction with the colliding discs, at any time.

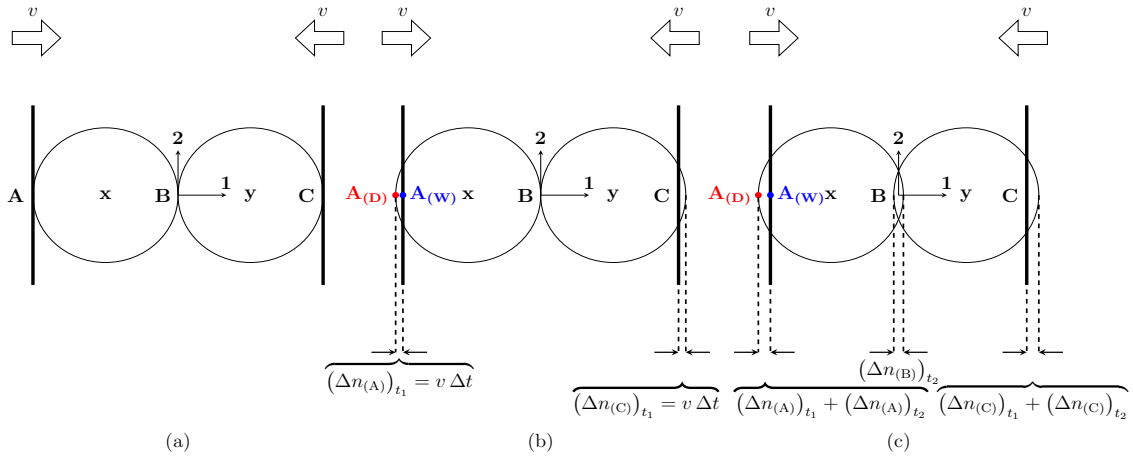


FIGURE 1.2. Two disc compressed between rigid walls; (a) $t = t_0$, (b) $t = t_1 = t_0 + \Delta t$, (c) $t = t_2 = t_1 + \Delta t$, see **[42]**

To illustrate how forces and displacements are determined during a calculation cycle, the case represented in Figure 1.2(a) will be considered. Two discs, labelled as disc **x** and disc **y**, are squashed between a pair of rigid walls. The walls move toward each other at a constant velocity v. Initially, at time $t = t_0$, the walls and discs are touching and no contact forces exist. A time t_1, the walls have moved inward over distances $v\,\Delta t$. In accordance with the assumption that the perturbations can not travel beyond a single disc during one time step, both discs are assumed to maintain their initial positions during the time interval from t_0 to t_1. Overlaps therefore exist at time $t_1 = t_0 + \Delta t$ at contacts **A** and **C** (Figure 1.2(b)) and are of magnitude $\Delta n = v\,\Delta t$.

The points $\mathbf{A_{(D)}}$ and $\mathbf{A_{(W)}}$ (Figure 1.2(b)) are points of the disc and the wall respectively, lying on the line drawn perpendicular to the wall and through the center of the disc. The contact **A** is the point halfway between $\mathbf{A_{(D)}}$ and $\mathbf{A_{(W)}}$. The relative

displacement $\left(\Delta n_{(\mathrm{A})}\right)_{t_1}$ at contact point (overlap) is the displacement of point $\mathbf{A_{(W)}}$ relative to that of point $\mathbf{A_{(D)}}$ ocurring over one time increment.

The relative displacements ocurring at contact points $\mathbf{A}$ and $\mathbf{C}$ at time t_1 are used in a force-displacement law for computing the contact forces. An incremental force-displacement law of the following form is used

$$\Delta F_n = \kappa_n \, (\Delta n)_{t_1} = \kappa_n \, v \, \Delta t \tag{1.1}$$

where κ_n is the normal stiffness and ΔF_n represents the increment in normal force.

The sum of the forces $\mathrm{F}_{(\mathbf{x})_1}$ and $\mathrm{F}_{(\mathbf{y})_1}$ for discs $\mathbf{x}$ and $\mathbf{y}$ at time t_1, taking account the direction 1 (see Figure 1.2) as positive, become

$$\mathrm{F}_{(\mathbf{x})_1} = \kappa_n \, (\Delta n)_{t_1} \tag{1.2}$$

$$\mathrm{F}_{(\mathbf{y})_1} = -\kappa_n \, (\Delta n)_{t_1} \tag{1.3}$$

which are used to obtain new accelerations through Newton's second law

$$\ddot{x}_1 = \frac{\mathrm{F}_{(\mathbf{x})_1}}{m_{(\mathbf{x})}} \tag{1.4}$$

$$\ddot{y}_1 = \frac{\mathrm{F}_{(\mathbf{y})_1}}{m_{(\mathbf{y})}} \tag{1.5}$$

where $m_{(\mathbf{x})}$ and $m_{(\mathbf{y})}$ are the masses of the disc $\mathbf{x}$ and $\mathbf{y}$ respectively. These accelerations (1.4) and (1.5) are assumed to be constant over the time interval $[t_1, t_2]$, then velocities can be obtained

$$[\dot{x}_1]_{t_2} = \left(\frac{\mathrm{F}_{(\mathbf{x})_1}}{m_{(\mathbf{x})}}\right) \Delta t \tag{1.6}$$

$$[\dot{y}_1]_{t_2} = \left(\frac{\mathrm{F}_{(\mathbf{y})_1}}{m_{(\mathbf{y})}}\right) \Delta t \tag{1.7}$$

The relative displacement increments at contact points **A**, **B** and **C** at time t_2 can be obtained

$$\left(\Delta n_{(\mathrm{A})}\right)_{t_2} = \left(v - \left(\frac{\mathrm{F}_{(\mathbf{x})_1}}{m_{(\mathbf{x})}}\right) \Delta t\right) \Delta t \tag{1.8}$$

$$\left(\Delta n_{(\mathrm{B})}\right)_{t_2} = \left(\left(\frac{\mathrm{F}_{(\mathbf{x})_1}}{m_{(\mathbf{x})}}\right) \Delta t - \left(\frac{\mathrm{F}_{(\mathbf{y})_1}}{m_{(\mathbf{y})}}\right) \Delta t\right) \Delta t \tag{1.9}$$

$$\left(\Delta n_{(\mathrm{C})}\right)_{t_2} = \left(\left(\frac{\mathrm{F}_{(\mathbf{y})_1}}{m_{(\mathbf{y})}}\right) \Delta t - (-v)\right) \Delta t \tag{1.10}$$

These calculations are repeated: forces corresponding to the displacements, which are found using force-displacement law (1.2) and (1.3), and the Newton's second law is used to obtain the displacements (1.4) and (1.5).

Cundall's approach has been widely studied [**162, 134, 161, 97, 100, 143, 62, 63**]. The particles are treated as rigid bodies but the contacts between particles follow a viscoelastic behavior in which the local strain variables are the relative particle positions and displacements.

The method called *disc model analysis* is presented in [**85**], where the term "disc" is used instead of "grain". The method is based upon the following assumptions: (i) the elastic behavior of a disc is replaced by the action of springs attached virtually at the contact point in the normal and tangential directions, and the disc itself is displaced and rotated rigidly; (b) discs can overlap each other, and only at that time contact force is produced between them. Usually the amount of overlapping is small enough compared with the radii; (c) for the slippage between discs, the Coulomb's friction law [**76**] is assumed, and the tangential component of the contact force can not exceed the limit determined by it; (d) to attain the equilibrium state of the disc assembly, each disc is displaced and rotated iteratively according to the contact

stiffness matrix defined from the locations of neighboring discs; and (e) the boundary can be either strain - controlled or stress - controlled, and the interaction between boundary elemen and a disc is specified in a similar way as the interaction between discs.

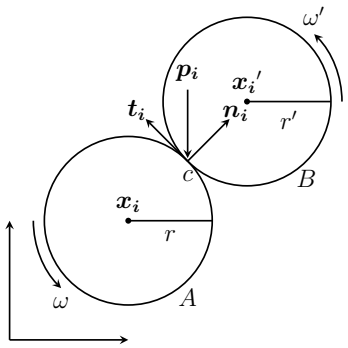


FIGURE 1.3. Contact force, see [**85**]

As shown in Figure 1.3, let two disk A and B, whose centers are located at $\boldsymbol{x_i}$ and $\boldsymbol{x_i}'$ respectively, are in contact with each other. The normal and tangential components of the contact force $\boldsymbol{p_i}$ which is acting on the disc A are denoted as $\boldsymbol{p_N} = -p_i\,\boldsymbol{n_i}$ and $\boldsymbol{p_T} = -p_i\,\boldsymbol{t_i}$, respectively. $\boldsymbol{n_i}$ and $\boldsymbol{t_i}$ are the normal and tangential unit vectors on the disc A. The increment of the relative displacement is given by

$$\Delta\, u_N = \left(\Delta\, \boldsymbol{x_i} - \Delta\, \boldsymbol{x_i}'\right)^T\, \boldsymbol{n_i} \tag{1.11}$$

$$\Delta\, u_T = \left(\Delta\, \boldsymbol{x_i} - \Delta\, \boldsymbol{x_i}'\right)^T\, \boldsymbol{t_i} + r\,\Delta\,\omega + r'\,\Delta\,\omega' \tag{1.12}$$

where r y r' are the radii, and ω and ω' are the angular velocities of the discs A and B respectively.

Between the contact force and the relative displacement, the following incremental force-displacement law is assumed:

$$\Delta \boldsymbol{p_N} = k_N \, \Delta \boldsymbol{u_N} \tag{1.13}$$

$$\Delta \boldsymbol{p_T} = k_T \, \Delta \boldsymbol{u_T} \tag{1.14}$$

where k_N and k_T are stiffnesses. The disc model analysis is performed through an itrative calculation, which is repeated until the resultant force and the moment of each member in the disc assembly become sufficiently small.

The theory formulated by Fremond [**55**], starts considering a point and a rigid body as a system, which is deformable since the distance of the point and the body changes. The basic idea leads to define strain rates and interior forces, these latter are percussions related to collisions and forces related to smooth evolutions, which are defined by their virtual work. This approach presents a new way to obtain the velocities after a collision, through the solution of a system of linear equations. This theory is explained in detail in Chapter 2.

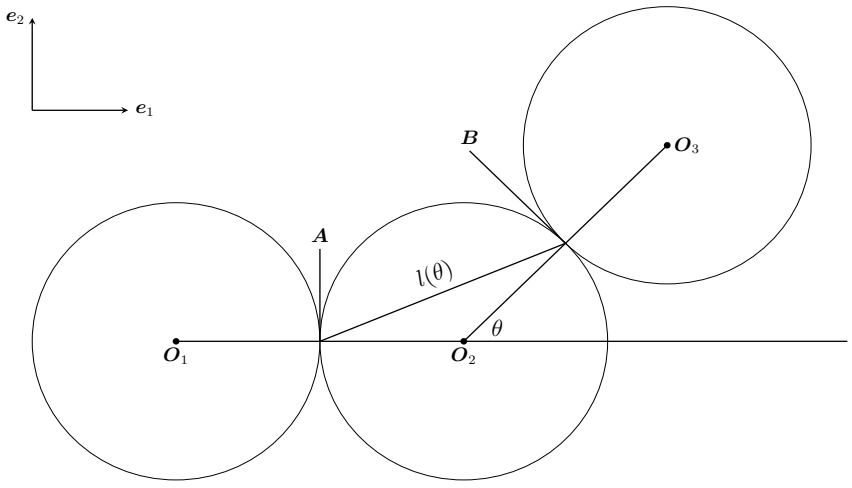


FIGURE 1.4. The system at collision time: the disks are seen from above, see [**27**].

Based in Fremond's approach, [**27**] presents the motion of three rigid balls on a plane and their multiple collisions is investigated. They start with a simplified 2D problem of three rigid disks moving in a plane, considering Fremond's approach: *"the system made of the three disks is deformable"* because the relative distances of material points vary. Therefore, three identical disks with redius R and center O_i, $i = 1, 2, 3$, evolve a plane with normal unit vector $\boldsymbol{e}_3$ (Figure 1.4). The unit vector $\boldsymbol{e}_1$ is parallel to $\boldsymbol{O}_1\boldsymbol{O}_2$ and the unit vector $\boldsymbol{e}_2$ is such that $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ is direct. The angle between $\boldsymbol{e}_1$ and $\boldsymbol{O}_2\boldsymbol{O}_3$ is denoted by θ and belongs to $[0, 2\pi/3]$.

At collision time $\hat{t}$, disks 1 and 2 are in contact at point $\boldsymbol{A}$ and disks 2 and 3 are in contact at point $\boldsymbol{B}$, also the distance between point $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted $l(\theta)$. A rigid system velocity $(\boldsymbol{U}, \Omega)$ is considered such that the distance of any two material points of the system remains constant; then, the velocity of any point $\boldsymbol{C}$ of the system can be expressed by

$$\boldsymbol{U}(\boldsymbol{C}) = \boldsymbol{U}^1(\boldsymbol{O}_1) + \Omega^1\, \boldsymbol{e}_3 \times \boldsymbol{O}_1\boldsymbol{C} \tag{1.15}$$

then, it has $\Omega^i = \hat{\Omega}$, $i = 1, 2, 3$ and

$$\boldsymbol{U}^2(\boldsymbol{O}_2) = \boldsymbol{U}^1(\boldsymbol{O}_1) + \hat{\Omega}\, \boldsymbol{e}_3 \times \boldsymbol{O}_1\boldsymbol{O}_2 \tag{1.16}$$

$$\boldsymbol{U}^3(\boldsymbol{O}_3) = \boldsymbol{U}^1(\boldsymbol{O}_1) + \hat{\Omega}\, \boldsymbol{e}_3 \times \boldsymbol{O}_1\boldsymbol{O}_3 \tag{1.17}$$

After setting the rigid system velocities, the velocities of deformation are defined, which are choose from the local velocities at the contact points $\boldsymbol{A}$ and $\boldsymbol{B}$, i.e., the

relative velocities

$$\boldsymbol{D}_{12}\left(\boldsymbol{U},\Omega\right)=\boldsymbol{U}^{1}(A)-\boldsymbol{U}^{2}(A) \tag{1.18}$$

$$\boldsymbol{D}_{23}\left(\boldsymbol{U},\Omega\right)=\boldsymbol{U}^{2}(B)-\boldsymbol{U}^{3}(A) \tag{1.19}$$

Also, an *at a distance velocity of deformation* [**27**] is chosen

$$\begin{aligned} D_{13}\left(\boldsymbol{U},\Omega\right) &= 2\left(\boldsymbol{U}^{1}(A)-\boldsymbol{U}^{3}(B)\right)\cdot\boldsymbol{BA} \\ &= -2\left(U_{1}^{1}-\left(U_{1}^{3}+R\,\Omega^{3}\sin\theta\right)\right)\,l\,\cos\frac{\theta}{2} \\ &= -2\left(U_{2}^{1}+R\,\Omega^{1}-\left(U_{2}^{3}+R\,\Omega^{3}\cos\theta\right)\right)\,l\,\sin\frac{\theta}{2} \end{aligned} \tag{1.20}$$

which accounts for interaction of disk 3 with disk 1. This velocity is considered as well as the others local velocities of deformation, with the goal to describe the non local interactions. For instance, it could applied in a predictive theory for the behavior of a mixture of sand and long fibers [**58**]. Applying Fremond's approach, the virtual work of the accelerations and interior forces are defined, therefore the principle of virtual work is applied. The internal forces are defined with constitutive laws which have to fulfill the principles of thermodynamics, which means that they have to satisfy the following inequality

$$-\mathcal{T}_{int}\left(\boldsymbol{u}^{\pm},\boldsymbol{\omega}^{\pm}\right)\geq 0 \tag{1.21}$$

where $\boldsymbol{u}^{+}$ and $\boldsymbol{u}^{-}$ are the linear velocity after and before the collision respectively. In the similar fashion $\boldsymbol{\omega}^{+}$ and $\boldsymbol{\omega}^{-}$ are the rotational velocity after and before the collision respectively. $\mathcal{T}_{int}$ is the work of internal forces. Therefore, an important point in Fremond's theory, is the way to satisfy (1.21), which is possible assuming that the

percussion is a function which belongs to the subdifferential of a pseudopotential of dissipation Φ introduced by [**81**] (see Appendix A).

Research based on Fremond's theory has been made, for instance in [**35**], a predictive theory of rigid bodies collisions is developed, where the research is focused in to prove the existence of a solution of the evolution problem of a point above a plane including friction and general constitutive laws during collisions. Also in [**56**], the collision of a wedge with a plane is investigated, the equations of motion involves an interior percussion as a sum of dissipative percussion. In [**57**], a model with fracture is presented, using Fremond's theory, the problem considered is a rock colliding with a wall, damage and collision theory is applied. Furthermore in [**59**], a predictive theory for collisions of a viscous incompressible fluid with solids is presented, this theory is based, as Fremond's approach, on interior percussions which account for the very large stresses and contact forces resulting from the kinematic incompatibilities responsible for the collision. In [**18, 19**], the fracture of a chandelier when falls on the ground is studied at the macroscopic level, the collisions are assumed instantaneous and characterized by a time discontinuity of the velocity field, as Fremond's approach. The Atomized efforts Contact Dynamics respecting the Clausius–Duhem inequality, A-CD^2, approach is presented in [**46, 132, 133, 48**], where a strategy for soil modeling is applied. This is based on a discretization of the media with rigid polygons, a mechanical description of instantaneous collisions is presented. In Chapter 2 is explained with more detail. Other studies based on this theory are crowd movements [**124, 125, 126, 127**], deformable bodies and fluids [**47**], several

applications in civil engineering [**60, 61**], among others. All Fremond's theory in detail, can be found in [**58**].

There are, of course, other approaches –different from Fremond's method– that deal with similar problems, e.g., impact between rigid bodies [**15, 17**], multibody dynamics [**130**], ellastic collisions [**158**], fall of rigid bodies [**152**], particle collision [**101**], nonlinear deformable bodies [**24**], flows with collisions [**155**], among others.

The contact dynamics (CD) method is presented as a discrete element method for the simulation of nonsmooth granular dynamics. This approach emerges from the mathematical formulation of nonsmooth dynamics and the subsequent algorithmic developments by Moreau and Jean [**108, 109, 110, 111, 112, 82, 81, 79, 80**]. The two central ingredients are: (1) the contact laws expressed as complementarity relations between the contact forces and velocities and (2) the nonsmooth motion involving velocity jumps with impulsive unresolved forces as well as smooth motion with resolved static forces; particularly in [**139**], the authors show that a consistent description of the dynamics at the velocity level leads to a numerical scheme together with an explicit treatment of the evolution of the particle configuration, also they discuss the intuitive features of the CD method with regard to collective phenomena involved in the multicontact dynamics of granular media: the role of the coarse-graining time δt, the precision issues and the interpretation of the restitution coefficients. This method has been applied to investigate granular materials [**138, 141, 22, 140, 156, 120, 142, 100, 160, 149, 14, 144**] as well as other mechanical systems composed of rigid bodies with frictional contact interactions.

Recently this method has been applied to simulate the penetration in coarse granular materials [**137**], where the system is simulated through rigid irregular polyhedral particles.

An important remark is that the models described before, consider the simulation of regular rigid bodies, rectangular, disks, balls, etc. Neverthless, there are many models which consider another kind of bodies, for instance in [**152**], a continuous contact impact model has been developed to simulate the multiple impacts of a falling rigid body with the ground (Figure 1.5). Results from the computational model as well as analytic analysis from a discrete contact impact model indicate that subsequent impacts might be larger than the initial impact in some situations.

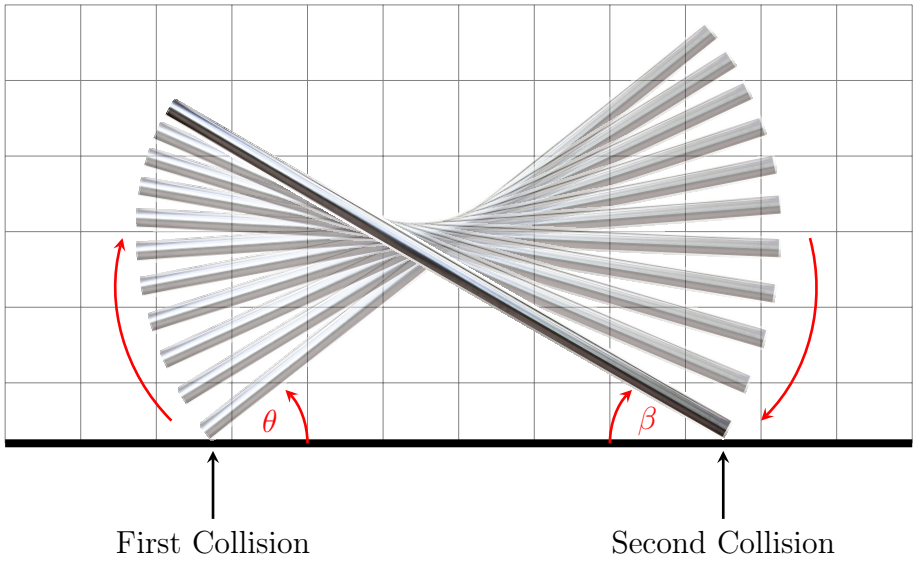


FIGURE 1.5. The positions of the rigid rod at different times between the first and second impact, see [**152**].

The continuous contact model, also known as the compliant contact model [**157**], is used to model the impact contact force, which allows to record specific impacts and forces at any particular moment; and the viscoelastic parameters in the model can be used to describe the energy dissipation and the elastic reconstitution of the

floor. The horizontal ground is modelled as a distributed viscoelastic foundation, which consists of a layer of continuously distributed parallel springs and dampers. Similar ground models have been used in [**157**] to study the impacts of concentrated masses or of rigid bodies with point impact contact.

Another interesting concept is presented in [**74**], denominated the potential particles, which is used to define convex particles with a wide of variety of shapes, from almost polygonal to circular. It is readily applicable in both two and three dimensions. Although contact detection and overlap calculations are not as fast as for circular particles, they are nevertheless sufficiently simple and fast for another implementations. Then, an alternative to the definition of a particle by geometric construction is to define the particle simply in terms of some function of a local coordinate system. The analysis is presented in terms of two dimensions, but the approach can be extended to three dimensions.

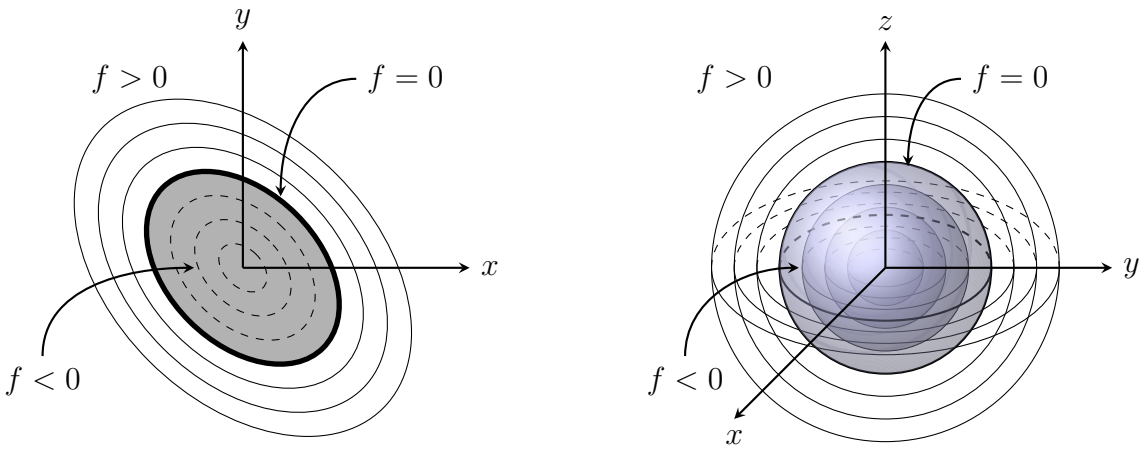


FIGURE 1.6. A simple "potential particle" for two and three dimensions, see [**74**].

The authors define a *potential particle* by means of function $f(x, y) = 0$, where (x, y) are local coordinates. The function is chosen so that: (a) $f = 0$ define the

particle surface; (b) $f < 0$ "inside" the particle; (c) $f > 0$ "outside" the particle; in the Figure 1.6 is possible to observe an example. When two particles is in contact, the model consider just the evaluation of the function f for each particle, considering a minimization or maximization problem.

A rigid bodies collision model also is introduced in [**163**], where a numerical model to generalize limit analysis in plasticity is presented. The system is represented as an assemblage of rigid bodies connected by normal and tangential springs. In [**135**] two rigid bodies are colliding in an incompressible flow, where the coupled system is formulated as a symmetric positive-definite matrix which is solved using the preconditioned conjugate gradient method, the method works for a "wet" and "dry" contact; in [**117**] a novel numerical approach is introduced for the simulation of soft particles interacting via frictional contacts, this approach is based on an implicit formulation of the Material Point Method [**67**], allowing for large particle deformations, combined with the Contact Dynamics method for the treatment of unilateral frictional contacts between particles. In [**70**] a new collision model has been developed to enable multiple-body proximate-flight simulation, where collisions between rigid bodies are modeled using an impulse-based approach with either an iterative propagation method or a simultaneous method.

1.3. Contact Detection

Large scale collision simulations involve to identify the contact of large number of bodies. Any collision model consideres to solve the new velocities of each rigid body after a collision, but before to compute this velocities, it is highly necessary to

know the contact point between a pair of elements, therefore it is very important to develop a robust and efficient contact detection algorithm.

There is a whole range of related works in this matter, for instance, in the formulation of the distinct element method (DEM) [**40, 71**], a robust and rapid method to identify pairs of particle that are touching is presented. A technique for identifying candidate pairs of colliding bodies is developed, in fact, they mention that is prohibitive, in computer time, to check all possible pairs, because the search time increases quadratically with the number of blocks. The space containing the system of blocks is divided into rectangular 3D cells. Each block is mapped into the cell or cells that its "envelope space" occupies. A block's envelope space is defined as the smallest three dimensional box with sides parallel to the coordinate axis that can contain the block. Each cell stores the addresses of all blocks that map into it. Figure 1.7 ilustrates a mapping logic for two dimensional space.

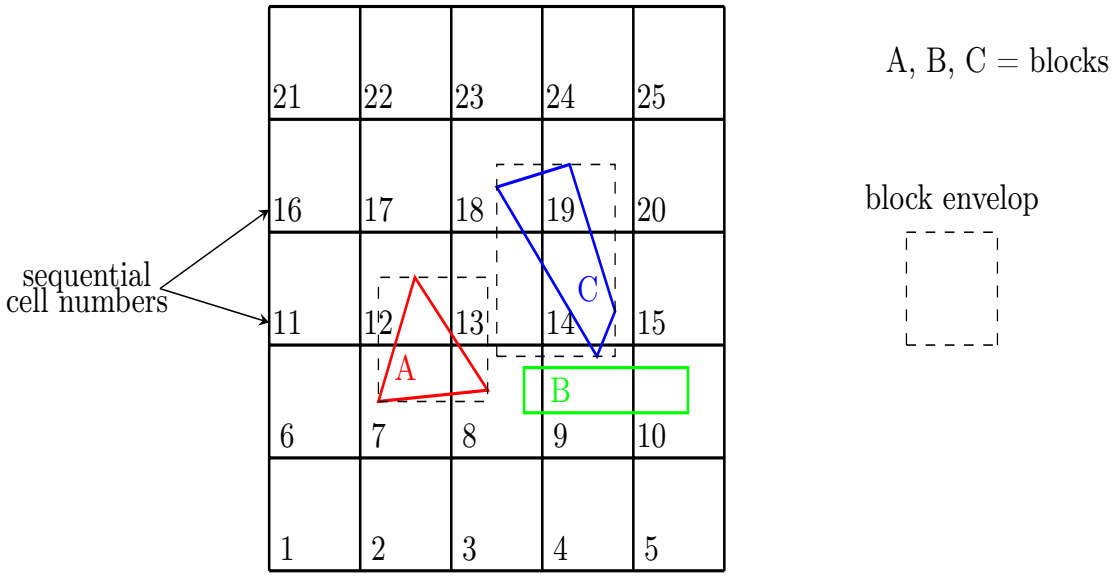


FIGURE 1.7. Examples of block mapping to cell space in two dimensions, see [**40**]

Once all blocks have been mapped into the cell space, it is an easy matter to identify the neighbours to a given block: the cells that correspond to its envelope space contain entries for all blocks that are near. Normally, this "search space" is increased in all directions by a tolerance, so that all blocks within the given tolerance are found. An important remark is that the computation time to perform the map and search functions for each block depends on the size and shape of the block, but not on the number of the blocks in the system. According to the optimum cell size, the authors keep clear that it is difficut to provide a formula for it. They proposed the following: as a block moves during the simulation, the neihbours must be updated, computing an accumulated movement of the block. A variable u_{acc}, set to zero after each re-map, is updated at very timestep, as follows:

$$u_{\text{acc}} := u_{\text{acc}} + \max\left(|du|\right) \tag{1.22}$$

where du is the incremental displacement of a vertex, and the max function is taken over all vertices of the block. When u_{acc} exceeds a tolerance, re-mapping and contact testing is activated. The contact testing is done for a search volume that is twice the tolerance larger in all dimensions than the block envelope. The contact detection problem in this work is splitted into the following two parts:

- determine a "common - plane" that bisects the space between the two blocks; and
- test each block separately for contact with the common - plane.

Then, the task of testing for contact is simplified the following way

(1) Simple vertex-to-plane tests need to be carried out (using dot products). Because the blocks are convex, face and edge contacts are recognized simply by counting the number of vertex-to-plane overlaps for both blocks.

(2) The number of tests depends linearly on the number of vertices. Because the test is over the vertices of block A with the common-plane, and separately test the vertices of block B with the common-plane.

(3) There is no need to test if a potential contact lies within the perimeter of a face. If both blocks touch the common-plane, then they must touch each other.

(4) The unit contact normal is equal to the common-plane unit normal – no additional calculations are necessary.

(5) The determination of minimum gap between two nontouching blocks is trivial: it is simply the sum of the distances of the two blocks from the common-plane.

An important step in contact detection problem, is checking a pair of elements are colliding or not. As Cundall [**40**] remarks, checking all elements in the system implies an algorithm with order N^2, where N is the number of elements. As we see above, it is possible to compute neighbours for each element. In the last example, we observe the cell space mapping technique. However, in [**166**] is introduced a neighbor table (Verlet list), in which those particles j which are nearby particle i are stored. The author pointed out that by using a table containing all particle pairs which are separated by a distance less than $(r_c + r_s)$, where r_c is the cutoff in the range of the interaction potential and r_s is a small skin, significant reductions in computer time

could be achieved. This list has been used in much research, mainly in molecular dynamics [**66, 32, 33, 64, 9, 168**], as the elements are spheres, it could be possible to obtain an important optimization in the contact detection phase.

An object representation scheme, based on a discrete function representation (DFR), is presented in [**123**]. The data structure provides a direct mapping between problem space and a hierarchical representation of the object geometry. This approach achieves a theoretical order of complexity $\mathcal{O}(N)$, for testing contact between two objects, each one defined by N points. This technique can handle general three dimensional shapes, including convex and concave geometries, unlike large systems with geometries relatively simple [**13, 170**].

There are two distinct algorithmic stages in contact detection, namely *spatial search* and *contact resolution*. Spatial search identifies pairs of objects in spatial proximity, which should be further processed for details of contact. Contact resolution provides these geometric details of the contact, such as the extent of the contact surface and the depth of penetration.

The order of algorithmic complexity for the resolution of contact between two objects is $\mathcal{O}(N^2)$, for unstructered boundary representations [**172, 41, 106**]. For tree schemes such as Binary Space Partitioning Trees (BSP), [**164**] or octrees, the cost can be reduced to $\mathcal{O}(N \log N)$ for convex objects. In [**123**] each object is represented at multiple levels of detail and then performing contact resolution at the appropiate level of approximation. The scheme presented uses 4 levels of representation: bounding spheres, bounding box, cellular region and surface facets, see Figure 1.8. A contact

detection schemes for discrete elements with more complex shapes is contained in [**173, 75**].

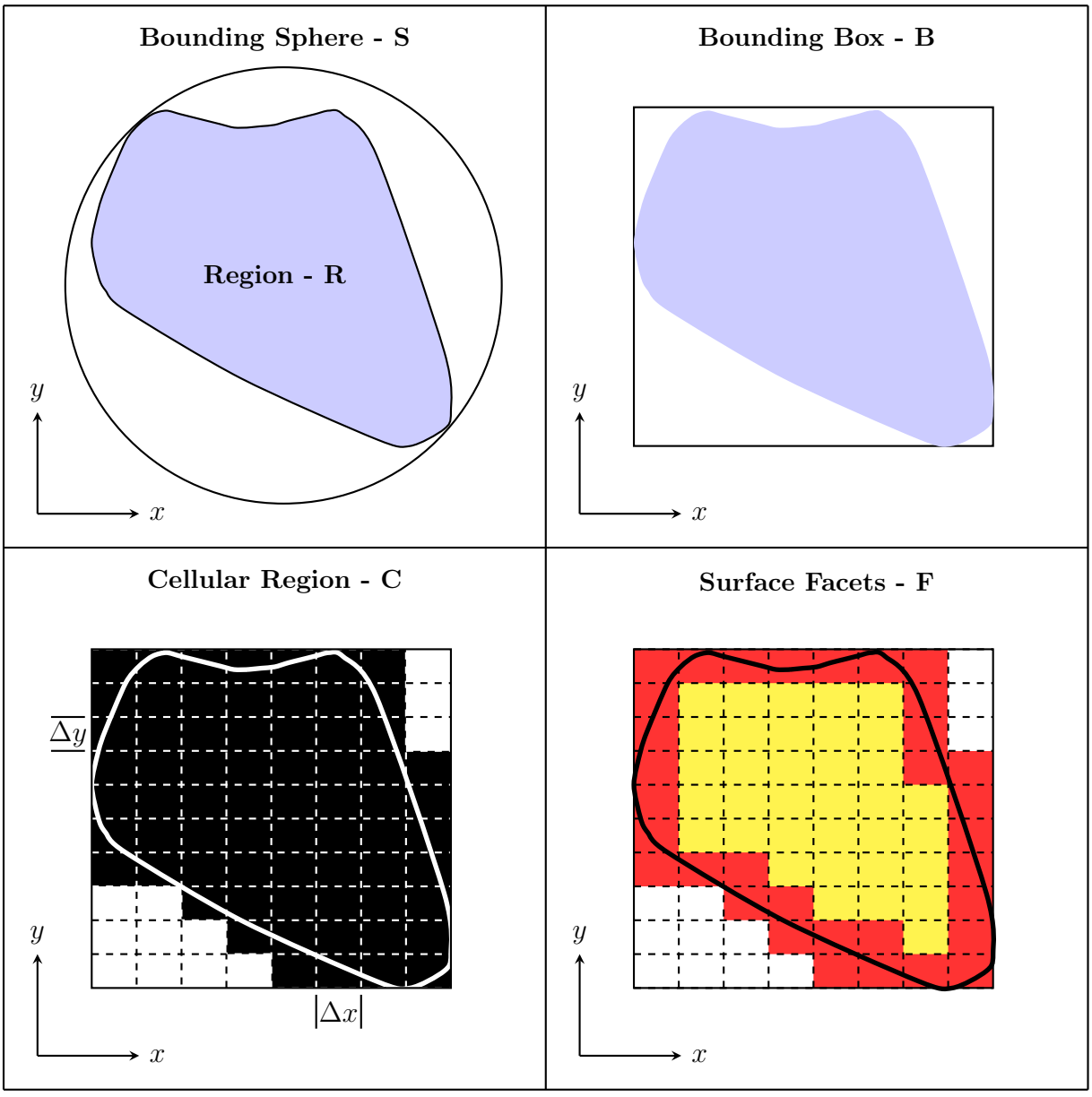


FIGURE 1.8. Multilevel of representations, see [**123**]

As we have seen until now, the contact detection problem is of interest, however there are some aspects to consider in the moment to choose an efficient algorithm. Obviously, while the element shape is complex, the contact detection part will be

expensive in terms of computational cost. For the other hand, if we know that the system have elements with similar size, it could be better. In [**114**], the *no binary search* (NBS) contact detection algorithm is presented, which is based in space descomposition.

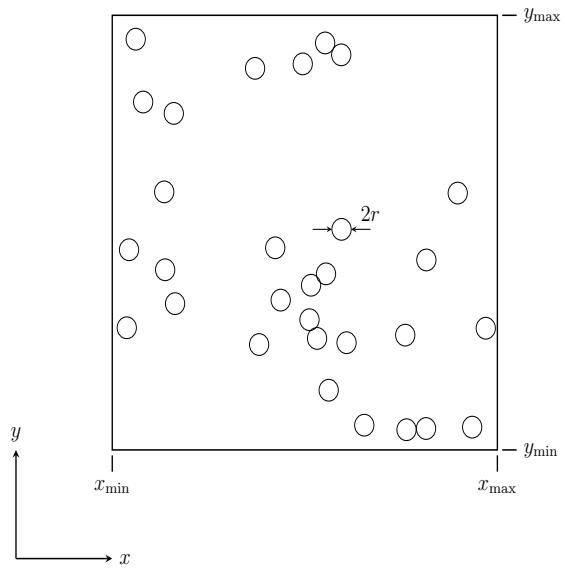


FIGURE 1.9. Contact detection problem, see [**114**]

An important assumption of this approach is that each discrete element can be approximated with a sphere in 3-D or with a circular disc in 2-D. The diameter of an equivalent circular disc $2r$ is obtained from the size of the largest discrete element in the system, i.e. all discrete elements are approximated with identical circular discs. Thus the NBS contact detection algorithm assumes a system comprising N identical discs occupying finite space of rectangular shape, see Figure 1.9. The space boundaries are defined by $x_{\min}$, $x_{\max}$, $y_{\min}$ and $y_{\max}$

The space is subdivided into identical square cells of size $2r$, see Figure 1.10. Each disc is assigned a integer identification number $\{0, 1, 2, ..., N-1\}$. In a similar fashion each cell is assigned an identification couple of integer numbers (ix, iy), where

$ix = 0, 1, 2, ..., \text{ncelx} - 1$ and $iy = 0, 1, 2, ..., \text{ncely} - 1$, where ncelx and ncely are the total number of cells in X and Y direction respectively

$$\text{ncelx} = \frac{x_{\max} - x_{\min}}{2r} \tag{1.23}$$

$$\text{ncely} = \frac{y_{\max} - y_{\min}}{2r} \tag{1.24}$$

Mapping from the set of discs

$$E = \{0, 1, 2, ..., N\} \tag{1.25}$$

to the set of cells

$$\begin{aligned} C = &\{(0,0), (0,1), \cdots, (0, \text{ncely} - 1), (1,0), (1,1), \cdots, (1, \text{ncely} - 1), \\ &\cdots, (\text{ncelx} - 1, 0), (\text{ncelx} - 1, 1), \cdots, (\text{ncelx} - 1, \text{ncely} - 1)\} \end{aligned} \tag{1.26}$$

is defined in such way that each element is assigned to one and only one cell. The detection of contact is therefore accomplished by checking all elements mapped to a particular cell against all elements in neighbouring cells.

An improvement of this procedure, named MR algorithm, is presented in [**115**]. An optimum cell size algorithm is presented in [**105, 104, 103**] to improve the calculation speed. The theoretical formula is proposed by considering the effects of the number of searched cells and contact checks. The optimum cell size was given round 1.5 - 2.0 times of particle radius.

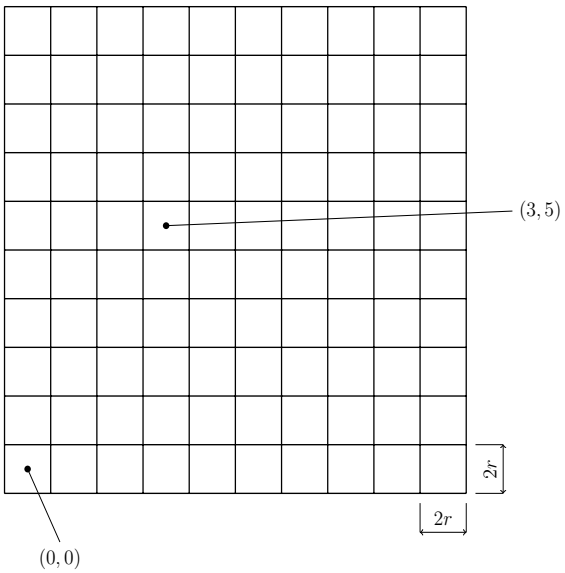


FIGURE 1.10. Space decomposition, see [**114**]

Much research about contact detection has been developed, in [**165**] an efficient algorithm is presented for arbitrarily shaped rigid moving objects. The detection is in two stages: finding candidates for possible collisions by building an octree, and the exact collision detection where a surface octree is built for each object. The algorithm is applied in particle flows simulation. In [**77**] a new boxing algorithm called "level - boxing" is presented, which consists in the definition of a maximum collision distance for each element depending on particle size. This definition allows to check collision for a large particle in a wide range, on the other hand, a smaller particle is limited to a narrower range.

In spite of that some algorithms use space decomposition or bounding box for contact detection, building cell-linked lists has achieved very good results. This technique [**12**] allows to reduce the complexity time for checking detection from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. For this reason, many works is based on this technique [**96, 128, 167, 95, 91, 174, 176, 94, 159**].

A fast contact detection algorithm named *fast common plane* (FCP) is presented in [**118**], where in 2-D the common plane is found by checking only 5 possible candidate planes, and in 3-D, the candidates plane fall within 4 types related to the geometry of the particles and their relative positions. This new approach introduces a new fashion to obtain the common plane for polygonal and polyhedral particles. In [**119**] a new approach to obtain the CP (see Figure 1.11) is presented. The concept of "shortest link" is introduced. Among all the possible line segments that connect any point on the surface of particle A to any point on the surface of particle B, the one with the shortest length defines the shortest link between the two particles. The perpendicular bisector plane of the shortest link fulfils all the conditions of a CP, suggesting that CP can be obtained by seeking the shortest link.

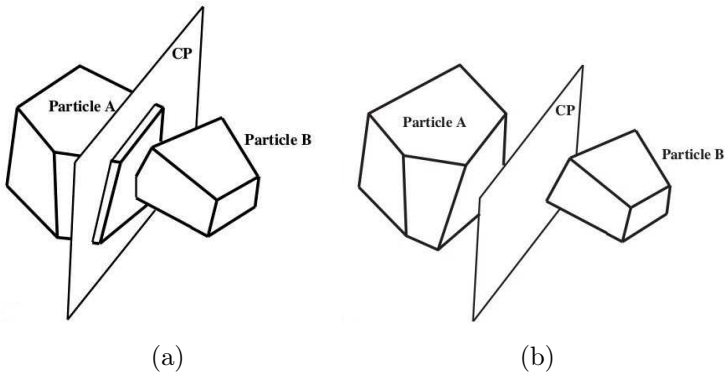


FIGURE 1.11. Common plane (CP) between two particles: (a) particles in contact, both particles intersect the CP; and (b) particles not in contact, neither particle intersects the CP, see [**119**].

The sort moving boxes (SMB) algorithm is presented in [**92**] for large number of moving 2D/3D objects which are represented by their axis-aligned bounding boxes (AABBs). The AABB is one of the most common bounding volumes [**53**], which consists in a rectangular six-sided box (in 3D, four-sided in 2D) categorized by having its faces oriented in such a way that its face normals are at all times parallel with

the axes of the given coordinate system. The main feature of the SMB algorithm is the full exploitation of the temporal coherence (property that some features of an application do not change significantly between two consecutive time steps) of the objects exhibited in a dynamic environment. In the algorithm, the AABBs are first projected to each Cartesian axis. The projected intervals on the axes are separately sorted by the diminishing increment sort (DIS) [**154**] and further divided into subsections. By processing all the intervals within the subsections to check if they overlap, a complete contact list can be built.

A collision detection method using a dual bounding volume hierarchy is presented in [**29, 30**]. The bounding volume hierarchy consists of an oriented bounding box (OBB) [**53**] tree enhanced with bounding spheres. This approach combines the compactness of OBBs and the simplicity of spheres. In [**102**], an oriented bounding parallelepiped (OBP) is suggested to build, instead of an ellipsoid around each body, therefore it is possible to predict collisions between elements that having overlapping parallelepipeds (see Figure 1.12)

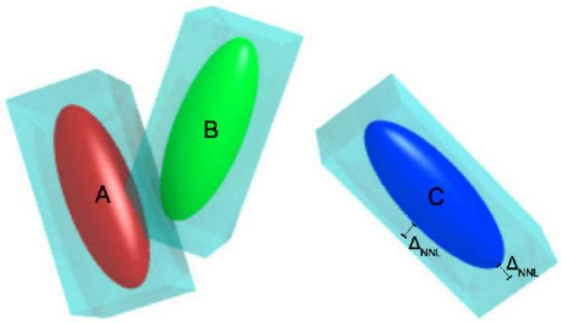


FIGURE 1.12. Ellipsoids A and B are nearest neighbors, and they may collide among them before colliding with their OBP; neither A or B can collide with C during this time, see [**102**]

In **[178]** a three-dimensional discrete element code (BLOKS3D) for efficient simulation of polyhedral particles of any size is developed. A neighbor search algorithm called TLS (two-level search) is described. The elements used in simulations are mainly convex particles, in **[20, 21, 93, 169]** several algorithms for contact detection between convex polygonal and polyhedral particles are presented. For shapes more complicated, several algorithms have been presented recently in **[177, 34, 86]**.

CHAPTER 2

A-CD^2 Model

In this thesis, Fremond's approach is followed [**55**]. The collision problem is addressed as a system deformation in the sense that a set of rigid bodies constitute a system. The system is deformable due to changes in the distance between the bodies. We highlight the origin of this approach, where the equations of motion are presented in the case of a collision. Later, a theory based on Fremond formulation is presented as the A-CD^2 approach, where forces are replaced by a succession of percussions (forces concentrated in time).

2.1. Rigid Bodies Collisions

The theory formulated by Fremond [**55**] starts considering a point and a rigid body as a system. The system is deformable since the distance of the point and the body changes. The basic idea is to define strain rates and interior forces, these latter are percussions related to collisions and forces related to smooth evolutions, which are defined by their virtual work.

Thus, the distance g between the point $\boldsymbol{x_1}$ and the rigid body Ω may change [**55, 58**]

$$g(t) = (\boldsymbol{x_1}(t) - \operatorname{proj} \boldsymbol{x_1}(t)) \cdot \boldsymbol{N} (\operatorname{proj} \boldsymbol{x_1}(t)) \tag{2.1}$$

where proj $\boldsymbol{x_1}$ is the projection of $\boldsymbol{x_1}$ on the boundary $\partial\Omega$ of Ω and $\boldsymbol{N}$ is the outwards normal vector to Ω at proj $\boldsymbol{x_1}$ (see Figure 2.1). Therefore the point does not interpenetrate the solid if and only if $\forall t\, g(t) \geq 0$.

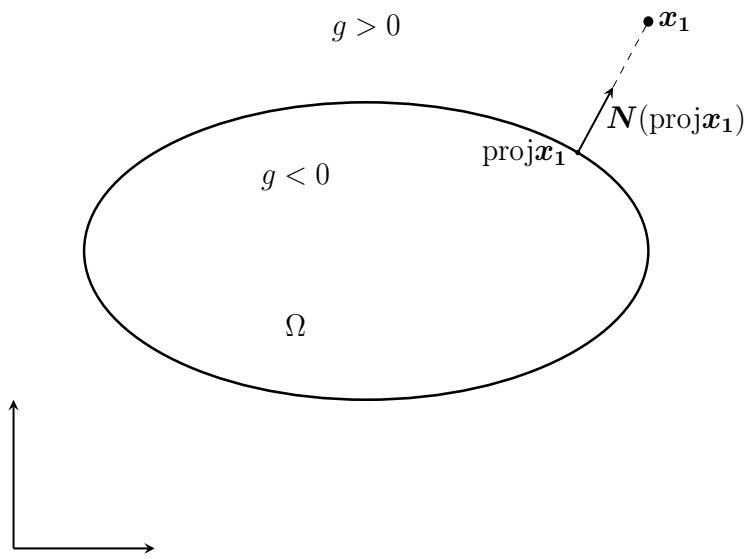


FIGURE 2.1. The point $\boldsymbol{x_1}(t)$ and the solid $\Omega(t)$ move and collide. The outwards normal vector to $\Omega(t)$ is $\boldsymbol{N}$, proj$\boldsymbol{x_1}$ is the projection of $\boldsymbol{x_1}$ on the boundary $\partial\Omega$ of Ω [**55**].

Let V_1 and V_2, be the linear spaces of the virtual velocities of the point (vectors depending on t) and of the rigid body (vectors depending on t and $\boldsymbol{x} \in \Omega(t)$); let R be the linear subset of $V = V_1 \times V_2$ of the rigid system velocities. Then $\boldsymbol{V}_1(t) = \boldsymbol{W}(\boldsymbol{x}_1(t), t), \boldsymbol{V}_2(\boldsymbol{x}, t) = \boldsymbol{W}(\boldsymbol{x}, t)$, where $\boldsymbol{W}(\boldsymbol{x}, t) = \boldsymbol{V}_0(t) + \boldsymbol{\omega}(t) \times (\boldsymbol{x} - \boldsymbol{x}_0)$, defined by the same twist, $\{\boldsymbol{V}_0(t), \boldsymbol{\omega}(t)\}$, because in a movement with such velocities the distances of the points of the system remain constant. Let $D = V/R$, be the linear space of the velocities defined up to a rigid system velocity. It is also the space of the velocities of the point with respect to the rigid body. The strain rate of the deformable system is defined by

$$\boldsymbol{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2) \in V \mapsto \boldsymbol{D}(\boldsymbol{V}) \qquad \text{with} \qquad \boldsymbol{D}(\boldsymbol{V})(t) = \boldsymbol{V}_1(t) - \boldsymbol{V}_2(\boldsymbol{x}_1(t), t) \quad (2.2)$$

Due to collisions, the velocities can be discontinues. The indices $-$ and $+$ refer to values before and after collisions.

Interior forces are defined by their work. There are forces and percussions. The latter are forces concentrated in time. The virtual work of the acceleration forces is [**58**]

$$\mathcal{T}_{acc}\left(\boldsymbol{V},t_1,t_2\right)=\int_{t_1}^{t_2}\left(m\,\frac{d\boldsymbol{U}_1}{dt}(t)\cdot\boldsymbol{V}_1(t)+\int\limits_{\Omega(t)}\rho\frac{d\boldsymbol{U}_2}{dt}(\boldsymbol{x},t)\cdot\boldsymbol{V}_2(\boldsymbol{x},t)\,d\Omega\right)dt$$

$$+\sum_{t\in Z}\left(m[\boldsymbol{U}_1](t)\cdot\frac{\boldsymbol{V}_1^+(t)+\boldsymbol{V}_1^-(t)}{2}+\int\limits_{\Omega(t)}\rho[\boldsymbol{U}_2](\boldsymbol{x},t)\cdot\frac{\boldsymbol{V}_2^+(t)+\boldsymbol{V}_2^-(t)}{2}d\Omega\right)\tag{2.3}$$

where $\boldsymbol{U}=(\boldsymbol{U}_1,\boldsymbol{U}_2)$ are the actual velocities, $[\boldsymbol{U}_1](t)=\boldsymbol{U}_1^+-\boldsymbol{U}_1^-$ and $Z=Z\left(\boldsymbol{V},\boldsymbol{U},t_1,t_2\right)$ is the set of the instants of the time interval $[t_1,t_2]$ where the actual or virtual velocities are discontinuos, m is the mass of the point and ρ the density of the rigid body. This expression is meaningful if the velocities are functions of bounded variation[**113**]. The virtual work of the acceleration forces is such that the actual work is the variation of the kinetic energy between the times t_1 and t_2.

The virtual work of the interior forces $\mathcal{T}_{int}$ is a linear function on D which must be equal to zero for any rigid system velocity [**58**]

$$\mathcal{T}_{\rm int}(\boldsymbol{V},t_1,t_2)=-\int_{t_1}^{t_2}\boldsymbol{R}^{\rm int}(t)\cdot\boldsymbol{D}(\boldsymbol{V})(t)dt-\sum_{t\in Z}\boldsymbol{P}^{\rm int}\cdot\frac{\boldsymbol{D}(\boldsymbol{V}^+)(t)+\boldsymbol{D}(\boldsymbol{V}^-)(t)}{2}\tag{2.4}$$

where $\boldsymbol{R}^{\rm int}(t)$ is a force and $\boldsymbol{P}^{\rm int}(t)$ a percussion. It is obvious that $\mathcal{T}_{\rm int}(\boldsymbol{V},t_1,t_2)=0$ for $\boldsymbol{V}\in R$.

The virtual work of the exterior forces $\mathcal{T}_{\text{ext}}$ is a linear function on V **[58]**

$$\mathcal{T}_{\text{ext}}(\boldsymbol{V}, t_1, t_2) = \int_{t_1}^{t_2} \left(\boldsymbol{f}_1(t) \cdot \boldsymbol{V}_1(t) + \int\limits_{\partial\Omega(t)} \boldsymbol{T}_2(\boldsymbol{x}, t) d\Gamma \right) dt$$

$$+ \sum_{t \in Z} \left(\boldsymbol{P}_1^{\text{ext}}(t) \frac{\boldsymbol{V}_1^+(t) + \boldsymbol{V}_1^-(t)}{2} + \int\limits_{\partial\Omega(t)} \boldsymbol{P}_2^{\text{ext}}(\boldsymbol{x}, t) \cdot \frac{\boldsymbol{V}_2^+(t) + \boldsymbol{V}_2^-(t)}{2} \, d\Omega \right) \tag{2.5}$$

where $\boldsymbol{P}_1^{\text{ext}}(t)$ is the exterior percussion applied to the point and $\boldsymbol{P}_2^{\text{ext}}(\boldsymbol{x}, t)$ the surface density of the exterior percussion applied to the solid, $\boldsymbol{f}_1(t)$ is the exterior force applied to the point and $\boldsymbol{T}_2(x, t)$ the exterior surface force applied to the solid.

The equations of motion result from the principle of virtual work: the virtual work of the accelerations between times t_1 and t_2 is equal to the sum of the virtual works of the interior and exterior forces between the same instants,

$$\forall t_1, \forall t_2, \forall \boldsymbol{V} \in V, \qquad \mathcal{T}_{acc}(\boldsymbol{V}, t_1, t_2) = \mathcal{T}_{\text{ext}}(\boldsymbol{V}, t_1, t_2) + \mathcal{T}_{\text{int}}(\boldsymbol{V}, t_1, t_2) \tag{2.6}$$

The equations of motion of the point are: at any time t, in particular when there is collision,

$$m[\boldsymbol{U}_1](t) = -\boldsymbol{P}^{\text{int}}(t) + \boldsymbol{P}_1^{\text{ext}}(t) \tag{2.7}$$

and almost everywhere,

$$m \, \frac{d\boldsymbol{U}_1}{dt} = -\boldsymbol{R}^{\text{int}} + f_1 \tag{2.8}$$

2.2. A-CD^2 Approach

A theory for multiple collisions in granular flow, based on the representation of the medium as a collection of rigid bodies [**55**] is presented in [**46, 132, 133**]. In this thesis, rigid bodies contacts are studied by means of the principle of virtual work. When rigid bodies get into collision, it is no longer possible to solve the classical equation of motion because velocities are not differentiable: the application of the principle of virtual work in association with appropriate constitutive laws relating internal stress and velocities, allows to obtain a set of equations of motion, valid both for smooth and for non-smooth evolutions. In particular, this approach allows us to overcome the limits of the classical penalty method, it does not require the definition of gap functions, and satisfy the actual physical condition of non-interpenetration of the particles. Moreover, compared to [**112**] or [**81**] the existence and the uniqueness of the solution of the Clausius–Duhem inequality has been showed [**46**]. This approach is called Atomized efforts Contact Dynamics respecting the Clausius–Duhem inequality (A-CD^2). It describes multiple bodies contact dynamics (according to the Clausius–Duhem inequality) by means of an "atomization" of the efforts exerted during contact.

According to obtain the velocities after the collision, this approach, for the sake of simplicity, it exposed for a single point, with mass m, colliding once to a rigid fixed surface in a time interval $[t_1, t_2]$ and having instantaneous velocity discontinuity, before generalizing the formulation to a simulataneous collision of N rigid solids [**55, 46**].

As the collision is assumed instantaneous, the velocity $\boldsymbol{u}$ of the point is discontinous at the instant t_c of the impact, having a left and right limit noted $\boldsymbol{u}^-$ and $\boldsymbol{u}^+$. The contact force is concentrated in time, becoming a percussion $\boldsymbol{P}^{int}$ [**55**]. In Figure 2.2 we observe the interaction between the point and the rigid fixed plane with one collision in the time interval, the description of the point trajectory $\boldsymbol{x}(t)$ in $[t_1, t_2]$ and the collision at time t_c.

Interior forces considered in the particle motion at interval time $[t_1, t_2]$ are defined by their work. Let us consider a contact force $\boldsymbol{f}^{int}$, which is concentrated in time, such as the internal percussion $\boldsymbol{P}^{int}$. The particle velocity is discontinuous [**46**], therefore let us consider a virtual velocity of deformation which has, such as the actual velocity $\boldsymbol{u}$, a left and right limit. This means, we have velocities before and after collision: $\boldsymbol{v}^-$ and $\boldsymbol{v}^+$ respectively. Then, the interior forces virtual work is given by [**132, 46**]

$$W^{int} = -\int_{t_1}^{t_2} \boldsymbol{f}^{int}(t) \cdot \boldsymbol{v}(t)\, dt - \boldsymbol{P}^{int}(t_c) \cdot \left(\frac{\boldsymbol{v}^-(t_c) + \boldsymbol{v}^+(t_c)}{2} \right) \tag{2.9}$$

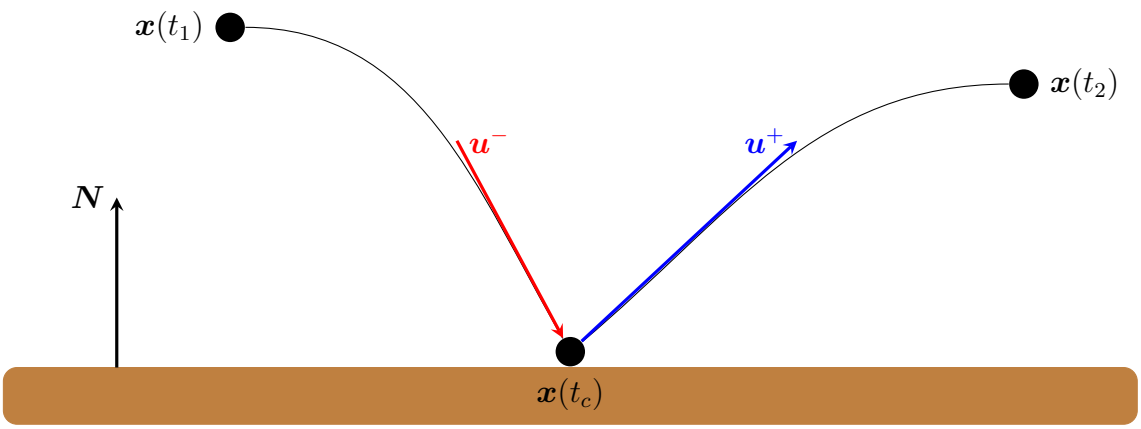


FIGURE 2.2. Point - plane system: a particle with mass m and a fixed plane.

In particular, (2.9) allows us to establish a duality between the internal percussion $\boldsymbol{P}^{int}$ and the quantity $\left(\dfrac{\boldsymbol{v}^- + \boldsymbol{v}^+}{2}\right)$, which can be interpreted as the rate of deformation of the system.

The virtual work of the acceleration efforts is **[132, 46]**

$$W^{acc} = \int_{t_1}^{t_2} m\frac{d\boldsymbol{u}(t)}{dt} \cdot \boldsymbol{v}(t)\, dt + m\left(\boldsymbol{u}^+(t_c) - \boldsymbol{u}^-(t_c)\right) \cdot \left(\frac{\boldsymbol{v}^-(t_c) + \boldsymbol{v}^+(t_c)}{2}\right) \tag{2.10}$$

An exterior percussion $\boldsymbol{P}^{ext}$ could be taken into account, not depending on $\boldsymbol{u}$, and also applied to the point at the instant of collision. This exterior percussion is later used to include the forces not depending on the velocities in the equations of the velocity jumps. The exterior virtual work is of the following form **[132, 46]**

$$W^{ext} = \int_{t_1}^{t_2} \boldsymbol{f}^{ext}(t) \cdot \boldsymbol{v}(t)\, dt + \boldsymbol{P}^{exp}(t_c) \cdot \left(\frac{\boldsymbol{v}^-(t_c) + \boldsymbol{v}^+(t_c)}{2}\right) \tag{2.11}$$

The principle of virtual work implies that for any velocity $\boldsymbol{v}$ at any time t_c, the following expression holds **[58]**

$$W^{acc} = W^{int} + W^{ext} \tag{2.12}$$

According to this principle, the equations of motion assume the following form on $[t_1, t_2]$

$$m\,\frac{d\boldsymbol{u}}{dt} = -\boldsymbol{f}^{int} + \boldsymbol{f}^{ext} \qquad \text{almost everywhere} \tag{2.13}$$

and

$$m\left(\boldsymbol{u}^+(t_c) - \boldsymbol{u}^-(t_c)\right) = -\boldsymbol{P}^{int} + \boldsymbol{P}^{ext} \qquad \text{everywhere} \tag{2.14}$$

The internal percussion $\boldsymbol{P}^{int}$ in (2.14) is generally unknown and depends on the deformation rate, as a consequence of the duality established by (2.9) between $\boldsymbol{P}^{int}$ and $\left(\dfrac{\boldsymbol{v}^- + \boldsymbol{v}^+}{2}\right)$. Therefore it is neccesary to introduce an appropiate set of constitutive

laws describing the behaviour and the interactions between the point and the rigid surface during the collision. This set is formulated by expressing $\boldsymbol{P}^{int}$ as a function of $\left(\dfrac{\boldsymbol{u}^- + \boldsymbol{u}^+}{2}\right)$.

In the case of a contact problem, constitutive laws describe the interactions among particles during the collision and have to assure the non - interpenetration of the solids.

Internal percussion is therefore divided in two parts, a dissipative percussion $\boldsymbol{P}^d$ and a reactive percussion $\boldsymbol{P}^{reac}$ **[132, 46]**

$$\boldsymbol{P}^{int} = \boldsymbol{P}^d + \boldsymbol{P}^{reac} \tag{2.15}$$

Dissipative percussion describes the dissipative interaccions among colliding solids. In general, the associated dissipative interaction can be described introducing a pseudopotential of dissipation Φ^d, which is a convex, positive and null in the origin **[46, 58, 107, 131]**

$$\boldsymbol{P}^d \in \partial\Phi^d \left(\frac{\boldsymbol{u}^- + \boldsymbol{u}^+}{2}\right) \tag{2.16}$$

where $\partial\Phi^d$ indicates the subdifferential (see Appendix A) of the function Φ^d.

Reactive percussion describes the non - interpenetration condition, which implies $u_N^+ = \boldsymbol{u}^+ \cdot \boldsymbol{N} \geq 0$ (see Figure 2.2). This percussion is null if $u_N^+ > 0$, is not null if $u_N^+ = 0$ and implies that the condition $u_N^+ < 0$ can not be verified (see Figure 2.3). All these properties can be written by means of a indicator function of the following form **[132, 46, 55]**

$$\boldsymbol{P}^{reac} \in \partial\mathbb{1}_K \left(\frac{\boldsymbol{u}^- + \boldsymbol{u}^+}{2} \cdot \boldsymbol{N}\right), \quad \text{with } K = \left[\frac{\boldsymbol{u}^- \cdot \boldsymbol{N}}{2}, \infty\right[\tag{2.17}$$

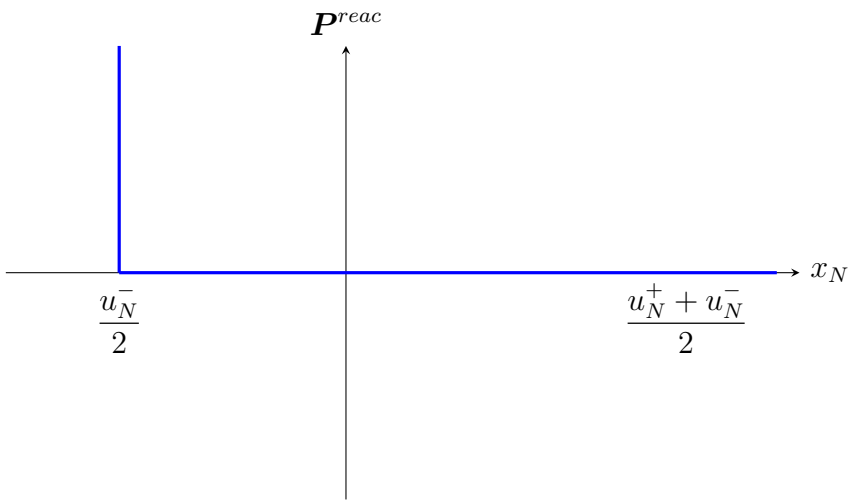


FIGURE 2.3. Definition of reactive percussion

In particular, as K is convex and contains the zero value, the indicator function $\mathbb{1}_K$ is a pseudopotential of dissipation [**107, 55, 131**]. The internal percussion can be written in the following form

$$\boldsymbol{P}^{int} \in \partial\Phi\left(\frac{\boldsymbol{u}^- + \boldsymbol{u}^+}{2}\right) \quad \text{where } \Phi = \Phi^d + \mathbb{1}_K \tag{2.18}$$

i.e. the internal percussion is derived from a pseudopotential of dissipation.

As the reactive percussion is determined by an indicator function, it is possible to express this condition by a constraint, where the relative velocity of the point at collision time must belong to the set $K = \left[\frac{u_N^-}{2}, \infty\right[$. The dissipative percussion can be determined by a normal behavior and a tangential behavior (friction) [**46**]. The former one, through experimental results, can be modeled by a quadratic function with good accuracy, but it is not the same case for the tangential behavior. In [**46**] is explained with more details the description of these behaviors and the constitutive law used for the contact forces

According to the assumptions made above about the evolution of the system, especially the solids moving with constant velocities during a time interval (Δt), the problem to solve at an instant of velocitiy jump has to be considered as a simultaneous collision of several solids. Indeed, all the contacts occurring during Δt are considered to take place at the instant of velocity jump as they are computed according to the positions of the bodies at this time interval. The problem to solve thus becomes similar to an instantaneous collision of several solids with known velocities before the collision.

Therefore, the instantaneous collision model exposed above for a "point - plane" system has to be generalized for N colliding solids, each one with mass m, gravity center $\boldsymbol{g}_i$, and moment of inertia I_i. The kth contact between solid i and solid j takes place at point $\boldsymbol{A}_{ijk}$ and $\boldsymbol{P}_{ijk}^{int}$ is the internal percussion at this point. The lth point where external percussion $\boldsymbol{P}_{il}^{ext}$ is applied to the solid i takes place at point $\boldsymbol{B}_{il}$.

The actual velocity of gravity center $\boldsymbol{g}_i$ is denoted by $\boldsymbol{u}_i$ and the actual rotational velocity by $\boldsymbol{\omega}_i$ of solid i; the virtual velocity is denoted by $\boldsymbol{v}_i$ and the virtual rotational velocity by $\boldsymbol{\varpi}_i$. Thus, the actual relative velocity of solids i and j at the contact point $\boldsymbol{A}_{ijk}$ is given by

$$\boldsymbol{d}_{ij}^{\boldsymbol{u}}\left(\boldsymbol{A}_{ijk}\right) = \left(\boldsymbol{u}_i + \boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{A}_{ijk}\right) - \left(\boldsymbol{u}_j + \boldsymbol{\omega}_j \times \boldsymbol{g}_j\boldsymbol{A}_{ijk}\right) \tag{2.19}$$

and at point $\boldsymbol{B}_{il}$ the actual velocity of solid i is

$$\boldsymbol{e}_i^{\boldsymbol{u}}\left(\boldsymbol{B}_{il}\right) = \left(\boldsymbol{u}_i + \boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{B}_{il}\right) \tag{2.20}$$

Also, the virtual relative velocity of solids i and j at the contact point $\boldsymbol{A}_{ijk}$ is given by

$$\boldsymbol{d}_{ij}^{\boldsymbol{v}}\left(\boldsymbol{A}_{ijk}\right)=\left(\boldsymbol{v}_i+\boldsymbol{\varpi}_i\times\boldsymbol{g}_i\boldsymbol{A}_{ijk}\right)-\left(\boldsymbol{v}_j+\boldsymbol{\varpi}_j\times\boldsymbol{g}_j\boldsymbol{A}_{ijk}\right) \tag{2.21}$$

and at point $\boldsymbol{B}_{il}$ the virtual velocity of solid i is

$$\boldsymbol{e}_i^{\boldsymbol{v}}\left(\boldsymbol{B}_{il}\right)=\left(\boldsymbol{v}_i+\boldsymbol{\varpi}_i\times\boldsymbol{g}_i\boldsymbol{B}_{il}\right) \tag{2.22}$$

From (2.14), and considering a solid, not a point, we have [**58**]:

$$m\left(\boldsymbol{u}^+-\boldsymbol{u}^-\right)=-\boldsymbol{P}^{int}+\boldsymbol{P}^{ext} \tag{2.23}$$

$$I\left(\boldsymbol{\omega}^+-\boldsymbol{\omega}^-\right)=-\boldsymbol{g}\,\boldsymbol{A}\times\boldsymbol{P}^{int}+\boldsymbol{g}\,\boldsymbol{B}\times\boldsymbol{P}^{ext} \tag{2.24}$$

Let $[\boldsymbol{x}]=\boldsymbol{x}^+-\boldsymbol{x}^-$ denote the difference between the right and left limit of the vector $\boldsymbol{x}$. Then, rewritten (2.23) and (2.24), we have

$$m\left[\boldsymbol{u}\right]=-\boldsymbol{P}^{int}+\boldsymbol{P}^{ext} \tag{2.25}$$

$$I\left[\boldsymbol{\omega}\right]=-\boldsymbol{g}\,\boldsymbol{A}\times\boldsymbol{P}^{int}+\boldsymbol{g}\,\boldsymbol{B}\times\boldsymbol{P}^{ext} \tag{2.26}$$

Applying (2.25) and (2.26) for N solids and for each kth contact at point $\boldsymbol{A}_{ijk}$:

$$\sum_{i=1}^{N}m_i\left[\boldsymbol{x}_i\right]=-\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{P}_{ijk}^{int}+\sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}}\boldsymbol{P}_{il}^{ext} \tag{2.27}$$

$$\sum_{i=1}^{N}I_i\left[\boldsymbol{\omega}_i\right]=-\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{P}_{ijk}^{int}+\sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}}\boldsymbol{g}_i\boldsymbol{B}_{il}\times\boldsymbol{P}_{il}^{ext} \tag{2.28}$$

Let $\overline{[x]}=\dfrac{x^-+x^+}{2}$ denote the average of right and left limit of the vector $\boldsymbol{x}$. Thus, we can obtain the virtual kinetic energy of the linear and rotational virtual velocity

by multipliying, respectively, the virtual linear and rotational rate of deformation:

$$\sum_{i=1}^{N} m_i \left[\boldsymbol{x}_i\right] \cdot \overline{\left[\boldsymbol{v}_i\right]} = -\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{P}_{ijk}^{int} \cdot \overline{\left[\boldsymbol{v}_i\right]} + \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext} \cdot \overline{\left[\boldsymbol{v}_i\right]} \tag{2.29}$$

$$\begin{aligned}\sum_{i=1}^{N} I_i \left[\boldsymbol{\omega}_i\right] \cdot \overline{\left[\boldsymbol{\varpi}_i\right]} = &-\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \left(\boldsymbol{g}_i \boldsymbol{A}_{ijk} \times \boldsymbol{P}_{ijk}^{int}\right) \cdot \overline{\left[\boldsymbol{\varpi}_i\right]} \\ &+ \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \left(\boldsymbol{g}_i \boldsymbol{B}_{il} \times \boldsymbol{P}_{il}^{ext}\right) \cdot \overline{\left[\boldsymbol{\varpi}_i\right]}\end{aligned} \tag{2.30}$$

Adding (2.29) and (2.30), we obtain the total virtual kinetic energy,

$$\begin{aligned}\sum_{i=1}^{N} m_i \left[\boldsymbol{u}_i\right] \cdot \overline{\left[\boldsymbol{v}_i\right]} + \sum_{i=1}^{N} I_i \left[\boldsymbol{\omega}_i\right] \cdot \overline{\left[\boldsymbol{\varpi}_i\right]} = &-\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{P}_{ijk}^{int} \cdot \left(\overline{\left[\boldsymbol{v}_i\right]} + \boldsymbol{g}_i \boldsymbol{A}_{ijk} \times \overline{\left[\boldsymbol{\varpi}_i\right]}\right) \\ &+ \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext} \cdot \left(\overline{\left[\boldsymbol{v}_i\right]} + \boldsymbol{g}_i \boldsymbol{B}_{il} \times \overline{\left[\boldsymbol{\varpi}_i\right]}\right)\end{aligned} \tag{2.31}$$

Now, we can obtain the actual kinetic energy of the linear and rotational actual velocity multipliying, respectively, the actual linear and rotational rate of deformation:

$$\sum_{i=1}^{N} m_i \left[\boldsymbol{u}_i\right] \cdot \overline{\left[\boldsymbol{u}_i\right]} = -\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{P}_{ijk}^{int} \cdot \overline{\left[\boldsymbol{u}_i\right]} + \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext} \cdot \overline{\left[\boldsymbol{u}_i\right]} \tag{2.32}$$

$$\begin{aligned}\sum_{i=1}^{N} I_i \left[\boldsymbol{\omega}_i\right] \cdot \overline{\left[\boldsymbol{\omega}_i\right]} = &-\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{g}_i \boldsymbol{A}_{ijk} \times \boldsymbol{P}_{ijk}^{int} \cdot \overline{\left[\boldsymbol{\omega}_i\right]} \\ &+ \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{g}_i \boldsymbol{B}_{il} \times \boldsymbol{P}_{il}^{ext} \cdot \overline{\left[\boldsymbol{\omega}_i\right]}\end{aligned} \tag{2.33}$$

Adding (2.32) and (2.33), we obtain the total actual kinetic energy,

$$\begin{aligned}\sum_{i=1}^{N} m_i \left[\boldsymbol{u}_i\right] \cdot \overline{\left[\boldsymbol{u}_i\right]} + \sum_{i=1}^{N} I_i \left[\boldsymbol{\omega}_i\right] \cdot \overline{\left[\boldsymbol{\omega}_i\right]} = &-\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{P}_{ijk}^{int} \cdot \left(\overline{\left[\boldsymbol{u}_i\right]} + \boldsymbol{g}_i \boldsymbol{A}_{ijk} \times \overline{\left[\boldsymbol{\omega}_i\right]}\right) \\ &+ \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext} \cdot \left(\overline{\left[\boldsymbol{u}_i\right]} + \boldsymbol{g}_i \boldsymbol{B}_{il} \times \overline{\left[\boldsymbol{\omega}_i\right]}\right)\end{aligned} \tag{2.34}$$

Substracting (2.31) and (2.34),

$$
\begin{aligned}
&-\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{P}_{ijk}^{int}\cdot\left(\overline{[\boldsymbol{v}_i]}+\boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\overline{[\boldsymbol{\varpi}_i]}\right)+\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{P}_{ijk}^{int}\cdot\left(\overline{[\boldsymbol{u}_i]}+\boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\overline{[\boldsymbol{\omega}_i]}\right)\\
&+\sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}}\boldsymbol{P}_{il}^{ext}\cdot\left(\overline{[\boldsymbol{v}_i]}+\boldsymbol{g}_i\boldsymbol{B}_{il}\times\overline{[\boldsymbol{\varphi}_i]}\right)-\sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}}\boldsymbol{P}_{il}^{ext}\cdot\left(\overline{[\boldsymbol{u}_i]}+\boldsymbol{g}_i\boldsymbol{B}_{il}\times\overline{[\boldsymbol{\omega}_i]}\right)\\
&=\sum_{i=1}^{N}m_i\left[\boldsymbol{u}_i\right]\left(\overline{[\boldsymbol{v}_i]}-\overline{[\boldsymbol{u}_i]}\right)+\sum_{i=1}^{N}I_i\left[\boldsymbol{\omega}_i\right]\left(\overline{[\boldsymbol{\varpi}_i]}-\overline{[\boldsymbol{\omega}_i]}\right)
\end{aligned}
\tag{2.35}
$$

Then, applying (2.19), (2.20), (2.21) and (2.22):

$$
\begin{aligned}
&\sum_{i=1}^{N}m_i\left[\boldsymbol{u}_i\right]\cdot\left(\overline{[\boldsymbol{v}_i]}-\overline{[\boldsymbol{u}_i]}\right)+\sum_{i=1}^{N}I_i\left[\boldsymbol{\omega}_i\right]\cdot\left(\overline{[\boldsymbol{\varpi}_i]}-\overline{[\boldsymbol{\omega}_i]}\right)\\
&+\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{P}_{ijk}^{int}\left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{v}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{v}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}\right)-\boldsymbol{P}_{ijk}^{int}\left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{u}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}\right)\\
&-\sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}}\boldsymbol{P}_{il}^{exp}\left(\frac{\boldsymbol{e}_{i}^{\boldsymbol{v}^+}\left(\boldsymbol{B}_{il}\right)+\boldsymbol{e}_{i}^{\boldsymbol{v}^-}\left(\boldsymbol{B}_{il}\right)}{2}\right)+\boldsymbol{P}_{il}^{exp}\left(\frac{\boldsymbol{e}_{i}^{\boldsymbol{u}^+}\left(\boldsymbol{B}_{il}\right)+\boldsymbol{e}_{i}^{\boldsymbol{u}^-}\left(\boldsymbol{B}_{il}\right)}{2}\right)=0
\end{aligned}
\tag{2.36}
$$

Now, introducing a set of constitutive laws, we can describe the behavior and interactions among the colliding solids, which using the pseudopotential defined by (2.18)

$$
\boldsymbol{P}_{ijk}^{int}\in\partial\Phi_{ijk}\left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{u}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}\right)
\tag{2.37}
$$

and from the sub-differential definition (see Appendix A), for any $\boldsymbol{v}$, we have that

$$
\begin{aligned}
&\Phi_{ijk}\left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{v}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{v}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}\right)-\Phi_{ijk}\left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{u}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}\right)\geq\\
&\boldsymbol{P}_{ijk}^{int}\cdot\left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{v}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{v}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}-\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}\left(\boldsymbol{A}_{ijk}\right)+\boldsymbol{d}_{ij}^{\boldsymbol{u}^-}\left(\boldsymbol{A}_{ijk}\right)}{2}\right)
\end{aligned}
\tag{2.38}
$$

Thus (2.36) becomes

$$\sum_{i=1}^{N} m_i \left[\boldsymbol{u}_i\right] \cdot \left(\overline{\left[\boldsymbol{v}_i\right]} - \overline{\left[\boldsymbol{u}_i\right]}\right) + \sum_{i=1}^{N} I_i \left[\boldsymbol{\omega}_i\right] \cdot \left(\overline{\left[\boldsymbol{\varpi}_i\right]} - \overline{\left[\boldsymbol{\omega}_i\right]}\right)$$
$$+ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \Phi_{ijk} \left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{v}^+}\left(A_{ijk}\right) + \boldsymbol{d}_{ij}^{\boldsymbol{v}^-}\left(A_{ijk}\right)}{2} \right) - \Phi_{ijk} \left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}\left(A_{ijk}\right) + \boldsymbol{d}_{ij}^{\boldsymbol{u}^-}\left(A_{ijk}\right)}{2} \right)$$
$$- \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{exp} \left(\frac{\boldsymbol{e}_i^{\boldsymbol{v}^+}\left(\boldsymbol{B}_{il}\right) + \boldsymbol{e}_i^{\boldsymbol{v}^-}\left(\boldsymbol{B}_{il}\right)}{2} \right) + \boldsymbol{P}_{il}^{exp} \left(\frac{\boldsymbol{e}_i^{\boldsymbol{u}^+}\left(\boldsymbol{B}_{il}\right) + \boldsymbol{e}_i^{\boldsymbol{u}^-}\left(\boldsymbol{B}_{il}\right)}{2} \right) \geq 0 \tag{2.39}$$

Introducing the scalar product

$$\left\langle \boldsymbol{u}, \boldsymbol{v} \right\rangle = \sum_{i=1}^{N} m_i \left(\boldsymbol{u}_i \cdot \boldsymbol{v}_i\right) + I_i \left(\boldsymbol{\omega}_i \cdot \boldsymbol{\varpi}_i\right) \tag{2.40}$$

and defining the vector $\boldsymbol{b}$ such as

$$\left\langle \boldsymbol{b}, \boldsymbol{u} \right\rangle = \sum_{i=1}^{N} \sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext} \cdot \boldsymbol{u}_i + \left(\boldsymbol{P}_{il}^{ext} \times \boldsymbol{g}_i \boldsymbol{B}_{il}\right) \cdot \boldsymbol{\omega}_i \tag{2.41}$$

and the pseudopotential Φ such as

$$\Phi\left(\boldsymbol{v}\right) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \Phi_{ijk} \left(\boldsymbol{d}_{ij}^{\boldsymbol{v}}\left(\boldsymbol{A}_{ijk}\right)\right) \tag{2.42}$$

then (2.39) becomes

$$\forall \boldsymbol{v}, \left\langle \boldsymbol{u}^+ - \boldsymbol{u}^- - \boldsymbol{b}, \frac{\boldsymbol{v}^+ + \boldsymbol{v}^-}{2} - \frac{\boldsymbol{u}^+ + \boldsymbol{u}^-}{2} \right\rangle + \Phi \left(\frac{\boldsymbol{v}^+ + \boldsymbol{v}^-}{2} \right) - \Phi \left(\frac{\boldsymbol{u}^+ + \boldsymbol{u}^-}{2} \right) \geq 0 \tag{2.43}$$

Given that $\mathbb{R}^{6N}$ has a scalar product defined by $\left\langle \cdot, \cdot \right\rangle$ and given the definition of sub - gradient, the formulation (2.43) is equivalent to following,

$$-\left(\boldsymbol{u}^+ - \boldsymbol{u}^- - \boldsymbol{b}\right) \in \partial\Phi \left(\frac{\boldsymbol{u}^+ + \boldsymbol{u}^-}{2} \right) \tag{2.44}$$

Let be $\boldsymbol{x} = \dfrac{\boldsymbol{u}^+ + \boldsymbol{u}^-}{2}$, then the last equation could be written by

$$2\,\boldsymbol{u}^- + \boldsymbol{b} \in 2\,\boldsymbol{x} + \partial\Phi\left(\boldsymbol{x}\right) \tag{2.45}$$

Equation 2.44 is equivalent to the following minimization problem:

$$\inf_{\boldsymbol{x}} F(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle + \Phi(\boldsymbol{x}) - \left\langle 2\,\boldsymbol{u}^- + \boldsymbol{b}, \boldsymbol{x} \right\rangle \tag{2.46}$$

Let's recall that he function $\Phi(\boldsymbol{x}) = \Phi^d(\boldsymbol{x}) + \mathbb{1}_K(\boldsymbol{x})$, where Φ^d is the sum of dissipative parts given by (2.16) and the indicator function represents the reactive part given by (2.17). Particularly, the reactive percussion can be replaced by the constraints $\varphi_k(\boldsymbol{x}) \leq 0$ defining the set Ω.

The collision problem is now a constrained minimization problem, where the solution gives us the new velocities of solids at some discrete time Δt. The function $F(\boldsymbol{x})$ is quadratic due to the choice of the dissipative percussion Φ^d (quadratic) and the set of constraints $\varphi_k(\boldsymbol{x}) \leq 0$, $k = 1, n_c$ (non - interpenetration condition) where n_c is the total number of contact points.

Thus, the constrained minimization problem is given by:

$$\begin{gathered} \inf_{\boldsymbol{x} \in \Omega} F(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle + \Phi^d(\boldsymbol{x}) - \left\langle 2\,\boldsymbol{u}^- + \boldsymbol{b}, \boldsymbol{x} \right\rangle \\ \text{where} \quad \Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^{6N} \,:\, \varphi_k(\boldsymbol{x}) \leq 0, \quad k = 1, ..., n_c \right\} \end{gathered} \tag{2.47}$$

The constraint $\varphi_k(\boldsymbol{x})$ associated to the contact point $\boldsymbol{A}_{ijk}$ is given by

$$\varphi_k(\boldsymbol{x}) = \left(\boldsymbol{d}_{ij}^{\boldsymbol{u}/\boldsymbol{2}^-}\left(\boldsymbol{A}_{ijk}\right) - \boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right) \right) \cdot \boldsymbol{N}_k \tag{2.48}$$

where $\boldsymbol{N}_k$ is the normal vector at contact point $\boldsymbol{A}_{ijk}$.

The constraint $\varphi_k(\boldsymbol{x})$ is the reactive part of the internal percussion, determined by the indicator function $\mathbb{1}_K$ (see Figure 2.3), where the velocity at contact point must be greater or equal to $\dfrac{u_N^-}{2}$, therefore when there is a simultaneous collision of N solids, the relative velocity at each contact point must to fulfill this reactive

condition, then

$$\boldsymbol{d}_{ij}^{\frac{\boldsymbol{u}^-}{2}}\left(\boldsymbol{A}_{ijk}\right)\cdot\boldsymbol{N}_k \leq \boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right)\cdot\boldsymbol{N}_k$$

$$\boldsymbol{d}_{ij}^{\frac{\boldsymbol{u}^-}{2}}\left(\boldsymbol{A}_{ijk}\right)\cdot\boldsymbol{N}_k - \boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right)\cdot\boldsymbol{N}_k \leq 0$$

$$\left(\boldsymbol{d}_{ij}^{\frac{\boldsymbol{u}^-}{2}}\left(\boldsymbol{A}_{ijk}\right) - \boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right)\right)\cdot\boldsymbol{N}_k \leq 0 \tag{2.49}$$

which is the constraint $\varphi_k(\boldsymbol{x}) \leq 0$. It is simple to note that the constraints are linear, then the problem to solve is a quadratic minimization problem with linear constraints.

The solution to the minimization problem (2.47) is a saddle point of the Lagrangian function [**46**]

$$L\left(\boldsymbol{x},\mu\right) \rightarrow F\left(\boldsymbol{x}\right) + \sum_{k=1}^{n_c}\mu_k\,\varphi_k\left(\boldsymbol{x}\right) \tag{2.50}$$

The domain Ω, such as $F\left(\boldsymbol{x}\right)$ and $\varphi_k\left(\boldsymbol{x}\right)$ is convex, this means that if $\boldsymbol{x}^*$ is a solution to the problem, exists at least a $\boldsymbol{\mu}^* \in \mathbb{R}_+^{n_c}$ such that $(\boldsymbol{x}^*, \boldsymbol{\mu}^*)$ is a saddle point of $L\left(\boldsymbol{x},\boldsymbol{\mu}\right)$.

The existence and the uniqueness of the solution is proved in [**133**], by the following theorem

THEOREM 1. *If all the constitutive laws derive from a pseudopotential of dissipation, then the problem of multiple collision of N rigid solids has a unique solution* $\boldsymbol{u}^+$.

The proof of this theorem is a direct consequence of the strong convexity of the function $F(\boldsymbol{x})$.

The constitutive law (2.37) derives from a pseudopotential of dissipation:

$$\boldsymbol{P}_{ijk}^{int} \cdot \left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}(\boldsymbol{A}_{ijk}) + \boldsymbol{d}_{ij}^{\boldsymbol{u}^-}(\boldsymbol{A}_{ijk})}{2} \right) \geq 0 \tag{2.51}$$

in every point $\boldsymbol{A}_{ijk}$ and

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{P}_{ijk}^{int} \cdot \left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}(\boldsymbol{A}_{ijk}) + \boldsymbol{d}_{ij}^{\boldsymbol{u}^-}(\boldsymbol{A}_{ijk})}{2} \right) \geq 0 \tag{2.52}$$

This inequality, under the hypothesis that the temperature does not vary during the collision, represents the inequality of Clausius - Duhem [**58**]. This inequality is important because it shows that the collision is dissipative if no exterior percussion is applied. In this case, the kinetic energy balance is negative,

$$\sum_{i=1}^{N} \frac{1}{2} m_i \left((\boldsymbol{u}_i^+)^2 - (\boldsymbol{u}_i^-)^2 \right) + \frac{1}{2} I_i \left((\boldsymbol{\omega}_i^+)^2 - (\boldsymbol{\omega}_i^-)^2 \right) = \sum_{i=1}^{N} m_i \left[\boldsymbol{u}_i \right] \overline{\left[\boldsymbol{u}_i \right]} + I_i \left[\boldsymbol{\omega}_i \right] \overline{\left[\boldsymbol{\omega}_i \right]}$$

$$= - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{\boldsymbol{A}_{ijk}} \boldsymbol{P}_{ijk}^{int} \cdot \left(\frac{\boldsymbol{d}_{ij}^{\boldsymbol{u}^+}(\boldsymbol{A}_{ijk}) + \boldsymbol{d}_{ij}^{\boldsymbol{u}^-}(\boldsymbol{A}_{ijk})}{2} \right) \leq 0 \tag{2.53}$$

To compute an approximation of the evolution described by the set of equations over a time interval $[0, T]$, it must handle forces having a density with respect to Lebesgue's measure **[69, 146]** in time and percussions having density with respect to Dirac's measure **[69, 146]** in time. The A-CD^2 method consists of approximating all the forces by a succession of percussions, in order to have all the stresses described as percussions.

The time discretization technique of the forces (atomization) presented makes the evolution a succession of instantaneous velocity discontinuities. Recalling (2.13) and (2.14), we have

$$m \frac{d\boldsymbol{u}}{dt} = -\boldsymbol{f}^{int} + \boldsymbol{f}^{ext} \qquad \text{almost everywhere} \tag{2.54}$$

and

$$m\left[\boldsymbol{u}\right] = -\boldsymbol{P}^{int} + \boldsymbol{P}^{ext} \qquad \text{everywhere} \tag{2.55}$$

From time 0 to t, the accumulated stresses $\boldsymbol{R}^{int}$ and $\boldsymbol{R}^{ext}$ are defined by [**133**],

$$\boldsymbol{R}^{int}(t) = \int_0^t \boldsymbol{f}^{int}(\tau)\, d\tau + \int_0^t \sum_{t_i} \boldsymbol{P}^{int}(t_i)\, \delta_{t_i}(\tau)\, d\tau \tag{2.56}$$

$$\boldsymbol{R}^{ext}(t) = \int_0^t \boldsymbol{f}^{ext}(\tau)\, d\tau + \int_0^t \sum_{t_j} \boldsymbol{P}^{ext}(t_j)\, \delta_{t_j}(\tau)\, d\tau \tag{2.57}$$

where t_i and t_j are the instants when internal and external percussions are exerted.

The percussion method (PM) approximates the regular elements of $\boldsymbol{R}^{int}$ and $\boldsymbol{R}^{ext}$ by the percussions. This procedure is named stress atomization. The approximated evolution is obtained by computing a series of velocities discontinuities separated by constant velocity evolutions. When this procedure is applied to a system formed by solids, is known as A-CD^2 method, and can be summarized as follows:

- The time length $[0, T]$ is discretized in n regular steps $[t_k, t_{k+1}]$ of length $\Delta_n = \dfrac{T}{n}$. In each time step, active forces are "atomized", that is, replaced by percussions exerted at the instant $\theta_k = t_k + \dfrac{n}{2}$.
- All the percussions exerted during the time gap $[t_k, t_{k+1}]$ are also exerted at the instant $\theta_k \in [t_k, t_{k+1}]$. It follows that velocities are discontinuous at the instants θ_k when the percussions are exerted and are constant elsewhere.

The atomization of a regular force $\boldsymbol{f}$ on the time interval $[t_k, t_{k+1}]$ consists of replacing it with the percussion $\boldsymbol{P}$ exerted at the instant θ_k. For instance, if $\boldsymbol{f}$ depends on a time-dependent quantity y, then

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(y(\tau)\right)\, d\tau \simeq \left(t_{k+1} - t_k\right)\, \boldsymbol{f}\left(\frac{y^+\left(\theta_k\right) + y^-\left(\theta_k\right)}{2}\right) \tag{2.58}$$

and thus $\boldsymbol{f}$ can be replaced by the percussion

$$\boldsymbol{P} = \Delta_n \boldsymbol{f} \left(\frac{y^+(\theta_k) + y^-(\theta_k)}{2} \right) \delta_{\theta_k}(t) \tag{2.59}$$

Three cases are considered,

- **Constant force.** The atomization of a constant force $\boldsymbol{f}_0$ consists of approximating it with a percussion of intensity $\boldsymbol{f}_0 \Delta_n$, on any time interval Δ_n. For instance, the action of the weight

  $$\boldsymbol{w}(t_k) = -\int_{t_k}^{t_{k+1}} m \boldsymbol{g} \, d\tau \tag{2.60}$$

  at time θ_k is approximated by

  $$\boldsymbol{p}_w(t) = -m \boldsymbol{g} \Delta_n \tag{2.61}$$

- **Time dependent force.** The force $\boldsymbol{f}(t)$ exerted on the system at time gap $[t_k, t_{k+1}]$ is replaced by a percussion of intensity $\Delta_n \boldsymbol{f}(\theta_k)$ exerted at the instant θ_k.
- **Position dependent force.** The force $\boldsymbol{f}(y(t))$ exerted on the system at the time $[t_k, t_{k+1}]$ is replaced by a percussion of intensity $\Delta_n \boldsymbol{f}(y(\theta_k))$ exerted at the instant θ_k. If we consider, for example, that two points are elastically bounded and note the elongation $x(t)$ at the instant t, the modulus of the elastic force exerted on every point is $\kappa |x(t)|$. This force is therefore replaced by a percussion of intensity $\Delta_n \kappa |x(\theta_k)|$.

A numerical example based on the proposed theory is presented; a package of randomly generated particles has been arranged on a skid, simulating the behaviour of a landslide formed with 300 rigid regular bodies **[133]**, see Figure 2.4. The parameters

are $\kappa_N = \kappa_T = 78\,[\mathrm{kg/m}]$ and the function Φ is

$$\Phi = \kappa_T \left(\left(\frac{\boldsymbol{u}^+ + \boldsymbol{u}^-}{2} \right) \boldsymbol{T} \right)^2 + \kappa_N \left(\left(\frac{\boldsymbol{u}^+ + \boldsymbol{u}^-}{2} \right) \boldsymbol{N} \right)^2 \tag{2.62}$$

where $\boldsymbol{T}$ and $\boldsymbol{N}$ are the tangential and normal directions respectively.

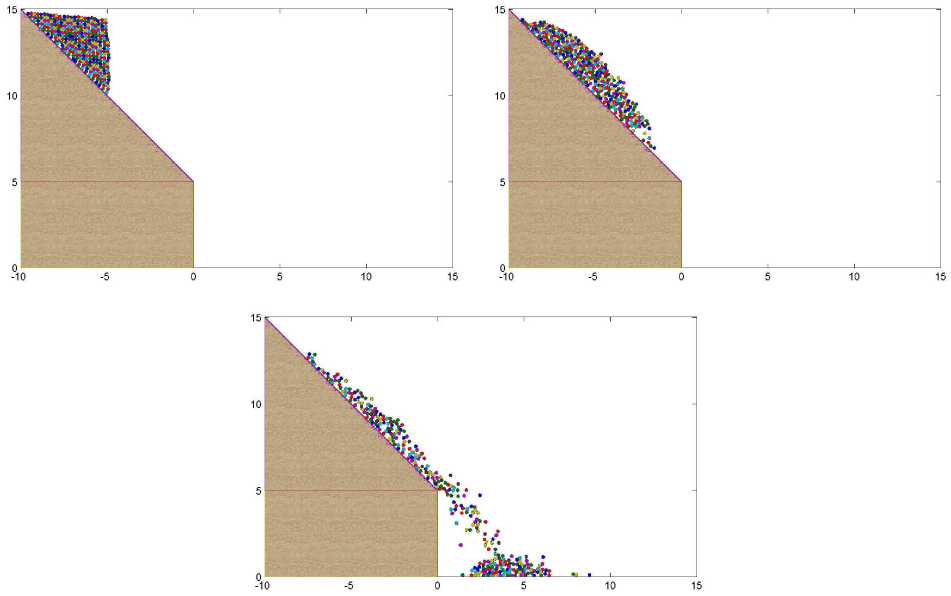


FIGURE 2.4. Numerical simulation of a landslide.

Similar results is obtained in **[132]** with more than 1200 particles, see Figure 2.5.

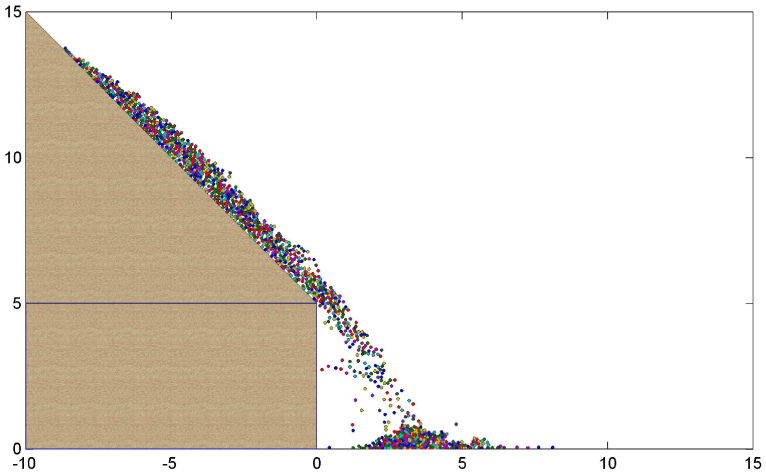


FIGURE 2.5. System with 1225 solids

CHAPTER 3

Numerical Model

The $AC - D^2$ [**132**] method is summarized by the following 3 recursive steps:

(1) Solids displace with constant velocity during Δt.

(2) Contact points are computed with positions of the solids and the end of time step Δt.

(3) New velocities are computed and replace the previous ones in step 1.

New velocities are calculated by means of a constrained minimization problem 2.47

$$\inf_{\boldsymbol{x}\in\Omega} F(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle + \Phi^d(\boldsymbol{x}) - \left\langle 2\,\boldsymbol{u}^- + \boldsymbol{b}, \boldsymbol{x} \right\rangle$$

$$\text{where} \quad \Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^{6N} \,:\, \varphi_k(\boldsymbol{x}) \leq 0, \quad k = 1, ..., n_c \right\} \tag{3.1}$$

The constraint $\varphi_k(\boldsymbol{x})$ associated to the contact point $\boldsymbol{A}_{ijk}$ is given by

$$\varphi_k(\boldsymbol{x}) = \left(\boldsymbol{d}_{ij}^{\boldsymbol{u}/\boldsymbol{2}^-}\left(\boldsymbol{A}_{ijk}\right) - \boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right) \right) \cdot \boldsymbol{N}_k \tag{3.2}$$

where $\boldsymbol{N}_k$ is the normal vector at contact point $\boldsymbol{A}_{ijk}$.

The goal of this chapter is to: (a) expose the matrix representation for $F(\boldsymbol{x})$, (b) show the new equations associated to the non constant rotational velocities of the solids and (c) explain the contact detection method used in this work.

3.1. Matrix Representation

The function $F(\boldsymbol{x})$ is possible to write it in the following quadratic matrix form

$$F\left(\boldsymbol{x}\right) = \frac{1}{2}\,\boldsymbol{x}^T\,\left(2M + \Xi\right)\,\boldsymbol{x} - \boldsymbol{x}^T\,\left(2\,M\,\boldsymbol{u}^- - \boldsymbol{b}^{\text{ext}}\right) \tag{3.3}$$

where $M \in \mathbb{R}^{6N\times 6N}$ is the inertia and mass matrix, $\Xi \in \mathbb{R}^{6N\times 6N}$ is the pseudopotential of dissipation matrix and $\boldsymbol{b}^{\text{ext}}$ is the external percussion vector.

The product $\boldsymbol{x}^T M\, \boldsymbol{x}$ represents the total kinetic energy K_e, which is obtained by

$$K_e = \sum_{i=1}^{N} m_i \cdot \boldsymbol{u}_i^T\, \boldsymbol{u}_i + I_i \cdot \boldsymbol{\omega}_i^T\, \boldsymbol{\omega}_i, \quad m_i \in \mathbb{R},\, I_i \in \mathbb{R}^{3\times 3},\, \boldsymbol{u}_i \in \mathbb{R}^3,\, \boldsymbol{\omega}_i \in \mathbb{R}^3 \tag{3.4}$$

where m_i is the mass of solid i and I_i is the corresponding inertial tensor.

Let the velocity vector be

$$\boldsymbol{u} = [\boldsymbol{u}_1\, \boldsymbol{\omega}_1\, \cdots\, \boldsymbol{u}_i\, \boldsymbol{\omega}_i\, \cdots\, \boldsymbol{u}_N\, \boldsymbol{\omega}_N]^T \in \mathbb{R}^{6N} \tag{3.5}$$

therefore the total kinetic energy can be written as

$$\begin{aligned} K_e(\boldsymbol{u}) &= \sum_{i=1}^{N} m_i\, \boldsymbol{u}_i^T\, \boldsymbol{u}_i + I_i\, \boldsymbol{\omega}_i^T\, \boldsymbol{\omega}_i = \sum_{i=1}^{N} \boldsymbol{u}_i^T \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \boldsymbol{u}_i + \boldsymbol{\omega}_i^T\, I_i\, \boldsymbol{\omega}_i \\ &= \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix}^T \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & & & \\ 0 & 0 & 0 & & I_i & \\ 0 & 0 & 0 & & & \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \\ &= \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix}^T \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & & & \\ 0 & 0 & 0 & & I_i & \\ 0 & 0 & 0 & & & \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \end{aligned} \tag{3.6}$$

Let the diagonal matrix be

$$M_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix}$$

then (3.6) becomes

$$K_e = \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix}^T \begin{bmatrix} M_i & Z_3 \\ Z_3 & I_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \tag{3.7}$$

where Z_3 is a null matrix with size 3×3. Then, expanding the sum, we have

$$
\begin{aligned}
K_e &= \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \end{bmatrix}^T \begin{bmatrix} M_1 & Z_3 \\ Z_3 & I_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \end{bmatrix} + \cdots + \begin{bmatrix} \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}^T \begin{bmatrix} M_N & Z_3 \\ Z_3 & I_N \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} M_1 & Z_3 & \cdots & \cdots & Z_3 \\ Z_3 & I_1 & Z_3 & \cdots & Z_3 \\ \vdots & Z_3 & \ddots & Z_3 & Z_3 \\ \vdots & \vdots & Z_3 & M_N & Z_3 \\ Z_3 & Z_3 & Z_3 & Z_3 & I_N \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \\
&\quad + \cdots + \\
&\quad + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}^T \begin{bmatrix} M_1 & Z_3 & \cdots & \cdots & Z_3 \\ Z_3 & I_1 & Z_3 & \cdots & Z_3 \\ \vdots & Z_3 & \ddots & Z_3 & Z_3 \\ \vdots & \vdots & Z_3 & M_N & Z_3 \\ Z_3 & Z_3 & Z_3 & Z_3 & I_N \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}^T \underbrace{\begin{bmatrix} M_1 & Z_3 & \cdots & \cdots & \cdots & \cdots & \cdots & Z_3 \\ Z_3 & I_1 & Z_3 & \cdots & \cdots & \cdots & \cdots & Z_3 \\ \vdots & Z_3 & \ddots & Z_3 & \cdots & \cdots & \cdots & Z_3 \\ \vdots & \vdots & Z_3 & M_i & Z_3 & \cdots & \cdots & Z_3 \\ \vdots & \vdots & \vdots & Z_3 & I_i & Z_3 & \cdots & Z_3 \\ \vdots & \vdots & \vdots & \vdots & Z_3 & \ddots & Z_3 & Z_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & Z_3 & M_N & Z_3 \\ Z_3 & Z_3 & Z_3 & Z_3 & Z_3 & Z_3 & Z_3 & I_N \end{bmatrix}}_{M} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix} \\
&= \boldsymbol{u}^T M \, \boldsymbol{u}
\end{aligned}
\tag{3.8}
$$

therefore, for any velocity vector $\boldsymbol{x}$ the total kinetic energy is

$$
K_e(\boldsymbol{x}) = \boldsymbol{x}^T M \, \boldsymbol{x} \tag{3.9}
$$

The constitutive law describing the behavior (normal and tangential) of the solids has been chosen quadratic **[133]**. The normal component of the dissipative percussion

applied at each contact point $\boldsymbol{A}_{ijk}$ is given by

$$\Phi_N^d(\boldsymbol{u}) = \sum_{\boldsymbol{A}_{ijk}} \frac{1}{2}\kappa_N \left(\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk}) \cdot \boldsymbol{N}_k\right)^2 \tag{3.10}$$

where $\kappa_N \geq 0$ is the normal coefficient, the relative velocity $\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk})$ is defined in (2.19) and $\boldsymbol{N}_k$ is the normal vector at contact point $\boldsymbol{A}_{ijk}$. Therefore, we have

$$\begin{aligned}
\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk}) \cdot \boldsymbol{N}_k &= \left((\boldsymbol{u}_i + \boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{A}_{ijk}) - \left(\boldsymbol{u}_j + \boldsymbol{\omega}_j \times \boldsymbol{g}_j\boldsymbol{A}_{ijk}\right)\right) \cdot \boldsymbol{N}_k \\
&= (\boldsymbol{u}_i + \boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{A}_{ijk}) \cdot \boldsymbol{N}_k - \left(\boldsymbol{u}_j + \boldsymbol{\omega}_j \times \boldsymbol{g}_j\boldsymbol{A}_{ijk}\right) \cdot \boldsymbol{N}_k \\
&= \boldsymbol{u}_i \cdot \boldsymbol{N}_k + (\boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{A}_{ijk}) \cdot \boldsymbol{N}_k - \boldsymbol{u}_j \cdot \boldsymbol{N}_k + \left(\boldsymbol{\omega}_j \times \boldsymbol{g}_j\boldsymbol{A}_{ijk}\right) \cdot \boldsymbol{N}_k \\
&= \boldsymbol{u}_i \cdot \boldsymbol{N}_k + (\boldsymbol{g}_i\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k) \cdot \boldsymbol{\omega}_i - \boldsymbol{u}_j \cdot \boldsymbol{N}_k + \left(\boldsymbol{g}_j\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k\right) \cdot \boldsymbol{\omega}_j \\
&= \begin{bmatrix} \boldsymbol{N}_k \\ \boldsymbol{g}_i\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k \end{bmatrix}^T \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} - \begin{bmatrix} \boldsymbol{N}_k \\ \boldsymbol{g}_j\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k \end{bmatrix}^T \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \end{bmatrix}
\end{aligned} \tag{3.11}$$

Defining the vector $\boldsymbol{c}_{ik} = \begin{bmatrix} \boldsymbol{N}_k \\ \boldsymbol{g}_i\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k \end{bmatrix} \in \mathbb{R}^6$ and considering the velocity vector defined in (3.5), the vectorial product in (3.11) is given by

$$\begin{aligned}
\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk}) \cdot \boldsymbol{N}_k &= \boldsymbol{c}_{ik}^T \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} - \boldsymbol{c}_{jk}^T \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \boldsymbol{N}_k \\ \boldsymbol{g}_i\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \boldsymbol{N}_k \\ \boldsymbol{g}_j\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k \\ \vdots \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}
\end{aligned} \tag{3.12}$$

Now, defining the following vector

$$\boldsymbol{c}_k = \begin{bmatrix} 0 \cdots 0 & \boldsymbol{N}_k & \boldsymbol{g}_i\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k & 0 \cdots 0 & -\boldsymbol{N}_k & -\boldsymbol{g}_j\boldsymbol{A}_{ijk} \times \boldsymbol{N}_k & 0 \cdots 0 \end{bmatrix}^T \tag{3.13}$$

the expression (3.11) becomes

$$\boldsymbol{d}_{ij}^{\boldsymbol{u}}\left(\boldsymbol{A}_{ijk}\right)\cdot\boldsymbol{N}_k=\begin{bmatrix}0\\ \vdots\\ 0\\ \boldsymbol{N}_k\\ \boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\\ 0\\ \vdots\\ 0\\ -\boldsymbol{N}_k\\ -\boldsymbol{g}_j\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\\ 0\\ \vdots\\ 0\end{bmatrix}^T\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{\omega}_1\\ \vdots\\ \boldsymbol{u}_i\\ \boldsymbol{\omega}_i\\ \vdots\\ \boldsymbol{u}_j\\ \boldsymbol{\omega}_j\\ \vdots\\ \boldsymbol{u}_N\\ \boldsymbol{\omega}_N\end{bmatrix}$$

$$=\boldsymbol{c}_k^T\,\boldsymbol{u} \tag{3.14}$$

Therefore, the normal behavior defined in (3.10) is given by

$$\begin{aligned}\Phi_N^d\left(\boldsymbol{u}\right)&=\frac{1}{2}\,\kappa_N\,\sum_{\boldsymbol{A}_{ijk}}\left(\left(\left(\boldsymbol{u}_i+\boldsymbol{\omega}_i\times\boldsymbol{g}_i\boldsymbol{A}_{ijk}\right)-\left(\boldsymbol{u}_j+\boldsymbol{\omega}_j\times\boldsymbol{g}_j\boldsymbol{A}_{ijk}\right)\right)\cdot\boldsymbol{N}_k\right)^2\\ &=\frac{1}{2}\,\kappa_N\,\sum_{\boldsymbol{A}_{ijk}}\left(\boldsymbol{c}_k^T\,\boldsymbol{u}\right)^2=\frac{1}{2}\,\kappa_N\,\sum_{\boldsymbol{A}_{ijk}}\left(\boldsymbol{c}_k^T\,\boldsymbol{u}\right)^T\,\left(\boldsymbol{c}_k^T\,\boldsymbol{u}\right)\\ &=\frac{1}{2}\,\kappa_N\,\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{u}^T\,\left(\boldsymbol{c}_k\,\boldsymbol{c}_k^T\right)\,\boldsymbol{u}\\ &=\frac{1}{2}\,\kappa_N\,\boldsymbol{u}^T\,\left(\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{c}_k\,\boldsymbol{c}_k^T\right)\,\boldsymbol{u}\end{aligned} \tag{3.15}$$

Let be $\Xi_N=\sum_{\boldsymbol{A}_{ijk}}\boldsymbol{c}_k\,\boldsymbol{c}_k^T$, then the normal behavior for any velocity vector $\boldsymbol{x}$ is given by

$$\Phi_N^d\left(\boldsymbol{x}\right)=\frac{1}{2}\,\kappa_N\,\left(\boldsymbol{x}^T\,\Xi_N\,\boldsymbol{x}\right) \tag{3.16}$$

The tangential component of the dissipative percussion applied at each contact point $\boldsymbol{A}_{ijk}$ is given by

$$\Phi_T^d(\boldsymbol{u}) = \sum_{\boldsymbol{A}_{ijk}} \frac{1}{2}\kappa_T \left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk})\right) - \left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk})\right)\cdot \boldsymbol{N}_k\right)\cdot \boldsymbol{N}_k\right)^2 \tag{3.17}$$

The relative velocity of a contact point A_{ijk} to the solid i is given by

$$\begin{aligned}
\boldsymbol{u}_i + \boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{A}_{ijk} &= \boldsymbol{u}_i + \begin{bmatrix} \boldsymbol{\omega}_i^y\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^z - \boldsymbol{\omega}_i^z\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^y \\ \boldsymbol{\omega}_i^z\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^x - \boldsymbol{\omega}_i^x\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^z \\ \boldsymbol{\omega}_i^x\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^y - \boldsymbol{\omega}_i^y\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^x \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{u}_i^x + \boldsymbol{\omega}_i^y\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^z - \boldsymbol{\omega}_i^z\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^y \\ \boldsymbol{u}_i^y + \boldsymbol{\omega}_i^z\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^x - \boldsymbol{\omega}_i^x\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^z \\ \boldsymbol{u}_i^z + \boldsymbol{\omega}_i^x\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^y - \boldsymbol{\omega}_i^y\ (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^x \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^z & -(\boldsymbol{g}_i\boldsymbol{A}_{ijk})^y \\ 0 & 1 & 0 & -(\boldsymbol{g}_i\boldsymbol{A}_{ijk})^z & 0 & (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^x \\ 0 & 0 & 1 & (\boldsymbol{g}_i\boldsymbol{A}_{ijk})^y & -(\boldsymbol{g}_i\boldsymbol{A}_{ijk})^x & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^x \\ \boldsymbol{u}_i^y \\ \boldsymbol{u}_i^z \\ \boldsymbol{\omega}_i^x \\ \boldsymbol{\omega}_i^y \\ \boldsymbol{\omega}_i^z \end{bmatrix} \\
&= R_{ik} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix}
\end{aligned} \tag{3.18}$$

Then the relative velocity for a contact point A_{ijk}, to both solids i and j, is given by

$$\begin{aligned}
\boldsymbol{d}_{ij}^{\boldsymbol{u}}(\boldsymbol{A}_{ijk}) &= (\boldsymbol{u}_i + \boldsymbol{\omega}_i \times \boldsymbol{g}_i\boldsymbol{A}_{ijk}) - \left(\boldsymbol{u}_j + \boldsymbol{\omega}_j \times \boldsymbol{g}_j\boldsymbol{A}_{ijk}\right) \\
&= R_{ik} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} - R_{jk} \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \end{bmatrix}
\end{aligned} \tag{3.19}$$

Let be Z_3^r a zero matrix with size $3 \times 6r$, therefore

$$R_{ik} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix} - R_{jk} \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \end{bmatrix} = \begin{bmatrix} Z_3^{6(i-1)} & R_{ik} & Z_3^{6(N-i)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix} - \begin{bmatrix} Z_3^{6(j-1)} & R_{jk} & Z_3^{6(N-j)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_j \\ \boldsymbol{\omega}_j \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} Z_3^{6(i-1)} & R_{ik} & Z_3^{6(j-i-1)} & R_{jk} & Z_3^{6(N-j)} \end{bmatrix}}_{R_k^{ij}} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}$$

$$= R_k^{ij} \, \boldsymbol{u} \tag{3.20}$$

where $R_k^{ij} \in \mathbb{R}^{3 \times 6N}$. For any velocity vector $\boldsymbol{x}$, the relative velocity at a contact point $\boldsymbol{A}_{ijk}$ of two solids i and j is then

$$\boldsymbol{d}_{ij}^{\boldsymbol{x}} \left(\boldsymbol{A}_{ijk} \right) = R_k^{ij} \, \boldsymbol{x} \tag{3.21}$$

The second part of the tangential behavior is given by

$$\left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{u}} \left(\boldsymbol{A}_{ijk} \right) \right) \cdot \boldsymbol{N}_k \right) \cdot \boldsymbol{N}_k \tag{3.22}$$

Taking into account (3.11), we have

$$
\begin{aligned}
\left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{u}}\left(\boldsymbol{A}_{ijk}\right)\right)\cdot\boldsymbol{N}_k\right)\cdot\boldsymbol{N}_k &= \left(\begin{bmatrix}\boldsymbol{N}_k\\ \boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\end{bmatrix}^T\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{\omega}_i\end{bmatrix}-\begin{bmatrix}\boldsymbol{N}_k\\ \boldsymbol{g}_j\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\end{bmatrix}^T\begin{bmatrix}\boldsymbol{u}_j\\ \boldsymbol{\omega}_j\end{bmatrix}\right)\cdot\boldsymbol{N}_k\\
&= \left(\begin{bmatrix}\boldsymbol{N}_k\\ \boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\end{bmatrix}^T\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{\omega}_i\end{bmatrix}\right)\cdot\boldsymbol{N}_k-\left(\begin{bmatrix}\boldsymbol{N}_k\\ \boldsymbol{g}_j\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\end{bmatrix}^T\begin{bmatrix}\boldsymbol{u}_j\\ \boldsymbol{\omega}_j\end{bmatrix}\right)\cdot\boldsymbol{N}_k\\
&= \underbrace{\begin{bmatrix}\boldsymbol{N}_k^x\left[\boldsymbol{N}_k\;\boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\right]\\ \boldsymbol{N}_k^y\left[\boldsymbol{N}_k\;\boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\right]\\ \boldsymbol{N}_k^z\left[\boldsymbol{N}_k\;\boldsymbol{g}_i\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\right]\end{bmatrix}}_{S_{ik}}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{\omega}_i\end{bmatrix}-\underbrace{\begin{bmatrix}\boldsymbol{N}_k^x\left[\boldsymbol{N}_k\;\boldsymbol{g}_j\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\right]\\ \boldsymbol{N}_k^y\left[\boldsymbol{N}_k\;\boldsymbol{g}_j\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\right]\\ \boldsymbol{N}_k^z\left[\boldsymbol{N}_k\;\boldsymbol{g}_j\boldsymbol{A}_{ijk}\times\boldsymbol{N}_k\right]\end{bmatrix}}_{S_{jk}}\begin{bmatrix}\boldsymbol{u}_j\\ \boldsymbol{\omega}_j\end{bmatrix}\\
&= S_{ik}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{\omega}_i\end{bmatrix}-S_{jk}\begin{bmatrix}\boldsymbol{u}_j\\ \boldsymbol{\omega}_j\end{bmatrix}\\
&= \begin{bmatrix}Z_3^{6(i-1)} & S_{ik} & Z_3^{6(N-i)}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{\omega}_1\\ \vdots\\ \boldsymbol{u}_i\\ \boldsymbol{\omega}_i\\ \vdots\\ \boldsymbol{u}_N\\ \boldsymbol{\omega}_N\end{bmatrix}-\begin{bmatrix}Z_3^{6(j-1)} & S_{jk} & Z_3^{6(N-j)}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{\omega}_1\\ \vdots\\ \boldsymbol{u}_j\\ \boldsymbol{\omega}_j\\ \vdots\\ \boldsymbol{u}_N\\ \boldsymbol{\omega}_N\end{bmatrix}\\
&= \underbrace{\begin{bmatrix}Z_3^{6(i-1)} & S_{ik} & Z_3^{6(j-i-1)} & S_{jk} & Z_3^{6(N-j)}\end{bmatrix}}_{S_k^{ij}}\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{\omega}_1\\ \vdots\\ \boldsymbol{u}_N\\ \boldsymbol{\omega}_N\end{bmatrix}\\
&= S_k^{ij}\,\boldsymbol{u} \qquad (3.23)
\end{aligned}
$$

where $S_k^{ij}\in\mathbb{R}^{3\times 6N}$. For any velocity vector $\boldsymbol{x}$, the relative velocity for a contact point $\boldsymbol{A}_{ijk}$ of two solids i and j is then

$$\left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right)\right)\cdot\boldsymbol{N}_k\right)\cdot\boldsymbol{N}_k = S_k^{ij}\,\boldsymbol{x} \qquad (3.24)$$

Finally the tangential behavior for any velocity vector $\boldsymbol{x}$, considering (3.21) and (3.24) is given by

$$\begin{aligned}
\Phi_T^d(\boldsymbol{x}) &= \sum_{\boldsymbol{A}_{ijk}} \frac{1}{2}\,\kappa_T\,\left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right)\right) - \left(\left(\boldsymbol{d}_{ij}^{\boldsymbol{x}}\left(\boldsymbol{A}_{ijk}\right)\right)\cdot\boldsymbol{N}_k\right)\cdot\boldsymbol{N}_k\right)^2 \\
&= \sum_{\boldsymbol{A}_{ijk}} \frac{1}{2}\,\kappa_T\,\left(R_k^{ij}\,\boldsymbol{x} - S_k^{ij}\,\boldsymbol{x}\right)^2 = \sum_{\boldsymbol{A}_{ijk}} \frac{1}{2}\,\kappa_T\,\left(\left(R_k^{ij} - S_k^{ij}\right)\,\boldsymbol{x}\right)^2 \\
&= \sum_{\boldsymbol{A}_{ijk}} \frac{1}{2}\,\kappa_T\,\boldsymbol{x}^T\,\left(R_k^{ij} - S_k^{ij}\right)^T\,\left(R_k^{ij} - S_k^{ij}\right)\,\boldsymbol{x} \\
&= \frac{1}{2}\,\kappa_T\,\boldsymbol{x}^T\left(\sum_{\boldsymbol{A}_{ijk}}\left(R_k^{ij} - S_k^{ij}\right)^T\,\left(R_k^{ij} - S_k^{ij}\right)\right)\,\boldsymbol{x}
\end{aligned} \tag{3.25}$$

Let be $\Xi_T = \sum_{\boldsymbol{A}_{ijk}}\left(R_k^{ij} - S_k^{ij}\right)^T\,\left(R_k^{ij} - S_k^{ij}\right)$, then the tangential behavior for any velocity vector $\boldsymbol{x}$ becomes

$$\Phi_T^d(\boldsymbol{x}) = \frac{1}{2}\,\kappa_T\,\left(\boldsymbol{x}^T\,\Xi_T\,\boldsymbol{x}\right) \tag{3.26}$$

Then, the total dissipative percussion is

$$\Phi^d(\boldsymbol{x}) = \Phi_N^d(\boldsymbol{x}) + \Phi_T^d(\boldsymbol{x}) \tag{3.27}$$

The external percussion in (2.41) is defined by the following expression

$$\langle \boldsymbol{b}, \boldsymbol{u}\rangle = \sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext}\cdot\boldsymbol{u}_i + \left(\boldsymbol{P}_{il}^{ext}\times\boldsymbol{g}_i\boldsymbol{B}_{il}\right)\cdot\boldsymbol{\omega}_i \tag{3.28}$$

We can obtain, for this expression, the following

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}} \boldsymbol{P}_{il}^{ext}\cdot\boldsymbol{u}_i + \left(\boldsymbol{P}_{il}^{ext}\times\boldsymbol{g}_i\boldsymbol{B}_{il}\right)\cdot\boldsymbol{\omega}_i &= \sum_{i=1}^{N}\sum_{\boldsymbol{B}_{il}} \begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{\omega}_i\end{bmatrix}^T\,\begin{bmatrix}\boldsymbol{P}_{il}^{ext}\\ \boldsymbol{P}_{il}^{ext}\times\boldsymbol{g}_i\boldsymbol{B}_{il}\end{bmatrix} \\
&= \sum_{\boldsymbol{B}_{il}}\sum_{i=1}^{N} \begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{\omega}_i\end{bmatrix}^T\,\begin{bmatrix}\boldsymbol{P}_{il}^{ext}\\ \boldsymbol{P}_{il}^{ext}\times\boldsymbol{g}_i\boldsymbol{B}_{il}\end{bmatrix}
\end{aligned} \tag{3.29}$$

Taking into account the velocity vector $\boldsymbol{u}$, the external percussion becomes

$$\langle \boldsymbol{b}, \boldsymbol{u} \rangle = \sum_{\boldsymbol{B}_{il}} \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \end{bmatrix}^T \begin{bmatrix} \boldsymbol{P}_{il}^{ext} \\ \boldsymbol{P}_{il}^{ext} \times \boldsymbol{g}_i \boldsymbol{B}_{il} \end{bmatrix} = \sum_{\boldsymbol{B}_{il}} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}^T \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{P}_{il}^{ext} \\ \boldsymbol{P}_{il}^{ext} \times \boldsymbol{g}_i \boldsymbol{B}_{il} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{u}_i \\ \boldsymbol{\omega}_i \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\omega}_N \end{bmatrix}^T \underbrace{\left(\sum_{\boldsymbol{B}_{il}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{P}_{il}^{ext} \\ \boldsymbol{P}_{il}^{ext} \times \boldsymbol{g}_i \boldsymbol{B}_{il} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right)}_{\boldsymbol{b}^{\text{ext}}}$$

$$= \boldsymbol{u}^T \, \boldsymbol{b}^{\text{ext}} \tag{3.30}$$

Finally, the function $F(\boldsymbol{x})$ is obtained in the matricial form. Replacing (3.9), (3.27) and (3.30), we can obtain

$$\begin{aligned} F(\boldsymbol{x}) &= \langle \boldsymbol{x}, \boldsymbol{x} \rangle + \Phi^d(\boldsymbol{x}) - \left\langle 2\,\boldsymbol{u}^- + \boldsymbol{b}^{\text{ext}}, \boldsymbol{x} \right\rangle \\ &= \langle \boldsymbol{x}, \boldsymbol{x} \rangle + \Phi_N^d(\boldsymbol{x}) + \Phi_T^d(\boldsymbol{x}) - \left\langle 2\,\boldsymbol{u}^-, \boldsymbol{x} \right\rangle - \left\langle \boldsymbol{b}^{\text{ext}}, \boldsymbol{x} \right\rangle \\ &= \boldsymbol{x}^T \, M \, \boldsymbol{x} + \frac{1}{2}\,\kappa_N \left(\boldsymbol{x}^T \, \Xi_N \, \boldsymbol{x} \right) + \frac{1}{2}\kappa_T \, \left(\boldsymbol{x}^T \, \Xi_T \, \boldsymbol{x} \right) - 2\,\boldsymbol{x}^T \, M \, \boldsymbol{u}^- - \boldsymbol{x}^T \, \boldsymbol{b}^{\text{ext}} \\ &= \frac{1}{2}\,\boldsymbol{x}^T \left(2M + \underbrace{\frac{1}{2}\,\kappa_N \, \Xi_N + \frac{1}{2}\,\kappa_T \, \Xi_T}_{\Xi} \right) \boldsymbol{x} - \boldsymbol{x}^T \, \left(2M \, \boldsymbol{u}^- - \boldsymbol{b}^{\text{ext}} \right) \\ &= \frac{1}{2}\,\boldsymbol{x}^T \, \left(2M + \Xi \right) \, \boldsymbol{x} - \boldsymbol{x}^T \, \left(2M \, \boldsymbol{u}^- - \boldsymbol{b}^{\text{ext}} \right) \end{aligned} \tag{3.31}$$

which is the expression defined in (3.3).

The linear constraints in (2.47) and (2.49) are defined by

$$\varphi_k(\boldsymbol{x}) = \left(\boldsymbol{d}_{ij}^{\frac{\boldsymbol{u}^-}{2}} \left(\boldsymbol{A}_{ijk} \right) - \boldsymbol{d}_{ij}^{\boldsymbol{x}} \left(\boldsymbol{A}_{ijk} \right) \right) \cdot \boldsymbol{N}_k \leq 0 \tag{3.32}$$

replacing (3.13) and considering (3.14), the constraint $\varphi_k(\boldsymbol{x})$ becomes

$$\begin{aligned} \varphi_k(\boldsymbol{x}) &= \left(\boldsymbol{d}_{ij}^{\frac{\boldsymbol{u}^-}{2}} \left(\boldsymbol{A}_{ijk} \right) - \boldsymbol{d}_{ij}^{\boldsymbol{x}} \left(\boldsymbol{A}_{ijk} \right) \right) \cdot \boldsymbol{N}_k \\ &= \boldsymbol{c}_k^T \frac{\boldsymbol{u}^-}{2} - \boldsymbol{c}_k^T \boldsymbol{x} \end{aligned} \tag{3.33}$$

Finally, the constrained minimization in matricial form has the following expression

$$\begin{aligned} &\min_{\boldsymbol{x} \in \Omega} \frac{1}{2} \boldsymbol{x}^T \left(2M + \Xi \right) \boldsymbol{x} - \boldsymbol{x}^T \left(2M \, \boldsymbol{u}^- - \boldsymbol{b}^{\text{ext}} \right) \\ &\text{where } \Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^{6N} \, : \, \boldsymbol{c}_k^T \frac{\boldsymbol{u}^-}{2} - \boldsymbol{c}_k^T \boldsymbol{x} \leq 0, \; k = 1, ..., n_c \right\} \end{aligned} \tag{3.34}$$

3.2. Torque and Force Free Rigid Box Motion

The A-CD^2 method considers the displacements with constant velocity during Δt. If we take into account a moving solid, we know that the linear velocities are constant, except in the z axis, where the gravity has effect. For the other hand, rotational velocities are not constant due to the inertial motion, which is not considered in the A-CD^2 approach. In this section we present a set of equations associated to the non constant rotational velocities of the solids.

We consider a solid rigid rectangular box (right parallelepiped) of sides a, b, c, flying freely in the absence of torques and external forces, except for gravity.

By Charles' theorem [**65**, p. 188] we know that any general displacement of the box can be represented by a translation plus a rotation, both independent of each other. More precisely, the problem of any rigid body motion can be split into two

separate phases: one concerned solely with the translational motion of the center of mass G of the rigid body, and the other concerning its rotational motion. The trajectory of the center of mass G of the box is described by a position vector $\mathbf{r}$ (*radius vector*) with respect to a fixed **global** coordinate system $\mathbf{O}XYZ$ and it is computed by the standard methods of gravitational mechanics.

We denote the canonical unit basis vectors of $\mathbf{O}XYZ$ by $\boldsymbol{\mathcal{I}}$, $\boldsymbol{\mathcal{J}}$ and $\boldsymbol{\mathcal{K}}$, along the X-, Y- and Z-axis, respectively.

Furthermore, we consider a local moving frame $\mathbf{G}xyz$ *fixed* at the center of mass G of the box. The canonical orthogonal unit basis vectors of $\mathbf{G}xyz$ are denoted by $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$, along the x-, y- and z-axis, respectively. Note that this local frame is fixed on the box, so that, as the box moves and rotates in space, these basis vectors $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ move and simultaneously rotate as well.

The fundamental equations describing the motion of the rigid box are the well known equations from Classical Mechanics:

$$\sum \mathbf{F} = m\,\mathbf{a}\,, \qquad \sum \mathbf{M}_G = \dot{\mathbf{H}}_G\,, \tag{3.35}$$

where $\sum \mathbf{F}$ is the sum of all external forces acting on the box, m is the mass of the box, $\mathbf{a}$ is the instantaneous acceleration of the center of mass G, i.e. $\mathbf{a} = \ddot{\mathbf{r}}$, $\sum \mathbf{M}_G$ is the sum of all external momenta (torques) applied to the box, and $\mathbf{H}_G$ is the angular momentum of the box about its mass center G.

In the continuous case the sums go smoothly over into integrals. From the general theory we know that:

$$\mathbf{H}_G = \mathbf{I}\,\boldsymbol{\omega}\,, \tag{3.36}$$

relative to the OXYZ coordinate system, where $\boldsymbol{\omega}$ is the instantaneous angular speed of the box $\mathbf{I}$ is its inertia tensor whose matrix representation is:

$$\mathbf{I} = \Big[\mathbf{I}_{ij}\Big]_{3\times 3}\,, \tag{3.37}$$

where:

$$\mathbf{I}_{11} = \int (y^2+z^2)\,dm\,, \qquad \mathbf{I}_{22} = \int (z^2+y^2)\,dm\,, \qquad \mathbf{I}_{33} = \int (y^2+x^2)\,dm\,, \tag{3.38}$$

$$\mathbf{I}_{12} = \int xy\,dm\,, \qquad \mathbf{I}_{13} = \int xz\,dm\,, \qquad \mathbf{I}_{23} = \int yz\,dm\,. \tag{3.39}$$

Note that $\mathbf{I}$, as a matrix, is Hermitian and hence, it is diagonalizable by means of a similarity transformation using a real orthogonal matrix. Thus, it is always possible to select the **orthogonal** coordinate axes **G**xyz in such a way that the inertia tensor $\mathbf{I}$ is diagonal:

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_{11} & 0 & 0 \\ 0 & \mathbf{I}_{22} & 0 \\ 0 & 0 & \mathbf{I}_{33} \end{bmatrix}. \tag{3.40}$$

These new **orthogonal** coordinate axes are the so-called *principal axes of inertia* and the corresponding diagonal coefficients $\mathbf{I}_{ii}$ are the *principal moments of inertia*.

With respect to the principal axes of inertia, i.e., with respect to the local coordinate system **G**xyz, equation (3.36) can be written as the set of three equations:

$$H_x \equiv H_1 = I_{11}\,\omega_1\,, \qquad H_y \equiv H_2 = I_{22}\,\omega_2\,, \qquad H_z \equiv H_3 = I_{33}\,\omega_3\,, \tag{3.41}$$

where:

$$H_x = \mathbf{H}_G \bullet \mathbf{i}\,, \qquad H_y = \mathbf{H}_G \bullet \mathbf{j}\,, \qquad H_z = \mathbf{H}_G \bullet \mathbf{k}\,, \tag{3.42}$$

$$\omega_1 = \boldsymbol{\omega} \bullet \mathbf{i}\,, \qquad \omega_2 = \boldsymbol{\omega} \bullet \mathbf{j}\,, \qquad \omega_3 = \boldsymbol{\omega} \bullet \mathbf{k}\,, \tag{3.43}$$

denote the x-, y-, z-components of $\mathbf{H}_G$ and $\boldsymbol{\omega}$ respectively.

In the sequel we will always assume that the moving local orthogonal coordinate system $\mathbf{G}xyz$ is the one formed by the principal axes of inertia and whose origin is fixed at the center of gravity $\mathbf{G}$ of the box.

With the help of the angular momentum $\mathbf{H}_G$ we introduce a special notation adapted to the situation we are dealing with. When we consider $\mathbf{H}_G$ with respect to the global coordinate system $\mathbf{O}XYZ$ we shall write:

$$\big[\mathbf{H}_G\big]_{\mathbf{O}XYZ} = H_X\,\boldsymbol{\mathcal{I}} + H_Y\,\boldsymbol{\mathcal{J}} + H_Z\,\boldsymbol{\mathcal{K}}\,, \tag{3.44}$$

where H_X, H_Y, H_Z denote the X-, Y- and Z-components of $\mathbf{H}_G$. The basis vectors $\boldsymbol{\mathcal{I}}$, $\boldsymbol{\mathcal{J}}$, $\boldsymbol{\mathcal{K}}$, are of course fixed.

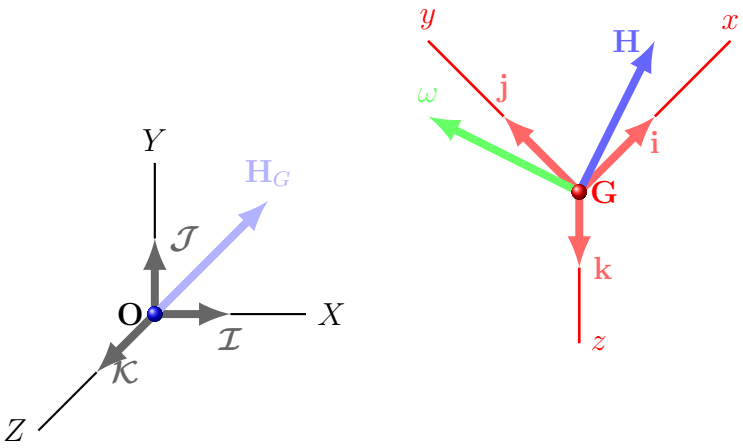


FIGURE 3.1. The global coordinate system $\boldsymbol{O}XYZ$ and the local orthogonal coordinate system $\boldsymbol{G}xyz$

Similarly we can consider $\mathbf{H}_G$ with respect to the moving local coordinate system $\mathbf{G}xyz$ of the principal axes of inertia of the box as well. In this case we write:

$$\big[\mathbf{H}_G\big]_{\mathbf{G}xyz} = H_x\,\mathbf{i} + H_y\,\mathbf{j} + H_z\,\mathbf{k}\,. \tag{3.45}$$

The basis vectors $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, are now fixed on the box and hence move in space and rotate with angular speed $\boldsymbol{\omega}$.

Note that for any given time instant we have:

$$H_X\,\boldsymbol{\mathcal{I}} + H_Y\,\boldsymbol{\mathcal{J}} + H_Z\,\boldsymbol{\mathcal{K}} = \big[\mathbf{H}_G\big]_{\mathbf{O}XYZ} = \big[\mathbf{H}_G\big]_{\mathbf{G}xyz} = H_x\,\mathbf{i} + H_y\,\mathbf{j} + H_z\,\mathbf{k}\,. \quad (3.46)$$

Component equalities do not hold in general, of course. Although the next remark is irrelevant for the classical mechanics considerations in this section, it gives some information about when component equality holds.

REMARK 1. *Recall that the* orthonormal *trihedron* $\{\mathbf{i},\,\mathbf{j},\,\mathbf{k}\}$ *rotates with angular speed* $\boldsymbol{\omega}$ *with respect to the fixed* orthonormal *trihedron* $\{\boldsymbol{\mathcal{I}},\,\boldsymbol{\mathcal{J}},\,\boldsymbol{\mathcal{K}}\}$. *Fix some instant* t, *and hence the* orthonormal *trihedron* $\{\mathbf{i},\,\mathbf{j},\,\mathbf{k}\}$ *as well. Apply a parallel shift to the trihedra* $\{\boldsymbol{\mathcal{I}},\boldsymbol{\mathcal{J}},\boldsymbol{\mathcal{K}}\}$ *and* $\{\mathbf{i},\mathbf{j},\mathbf{k}\}$, *so that all six vectors originate at the center of gravity* G *of the box. Then, whenever* $\boldsymbol{\mathcal{I}}\bullet\mathbf{i} \neq \pm 1$, *respectively* $\boldsymbol{\mathcal{J}}\bullet\mathbf{j} \neq \pm 1$ *and* $\boldsymbol{\mathcal{K}}\bullet\mathbf{k} \neq \pm 1$, *i.e., whenever these vector pairs span planes, we have:*

$$H_X = H_x \quad \Leftrightarrow \quad \mathbf{H}_G = a\,(\boldsymbol{\mathcal{I}} + \mathbf{i}) + b\,\boldsymbol{\mathcal{I}} \times \mathbf{i}\,, \quad a, b \in \mathbb{R}\,, \quad (3.47)$$

$$H_Y = H_y \quad \Leftrightarrow \quad \mathbf{H}_G = a\,(\boldsymbol{\mathcal{J}} + \mathbf{j}) + b\,\boldsymbol{\mathcal{J}} \times \mathbf{j}\,, \quad a, b \in \mathbb{R}\,, \quad (3.48)$$

$$H_Z = H_z \quad \Leftrightarrow \quad \mathbf{H}_G = a\,(\boldsymbol{\mathcal{K}} + \mathbf{k}) + b\,\boldsymbol{\mathcal{K}} \times \mathbf{k}\,, \quad a, b \in \mathbb{R}\,. \quad (3.49)$$

PROOF. We only prove (3.47).

($\Rightarrow$) Let $\mathbf{H}_G$ be such that $H_X = H_x$. Let $\mathbf{H}_G$ originates at G and consider this vector as the radius vector of a point Q in $\mathbb{R}^3$ relative to G. First we observe, that the geometric place of the points P *on the plane* E *spanned by* $\boldsymbol{\mathcal{I}}$ and $\mathbf{i}$, that have equal projections onto $\boldsymbol{\mathcal{I}}$ and $\mathbf{i}$, is the bisector B of the angle subtended by those vectors. The equation of the points P on B is $P = \lambda\,(\boldsymbol{\mathcal{I}} + \mathbf{i})$, $\lambda \in \mathbb{R}$. Now, it is fairly obvious that the geometric place of all points Q *on the space* $\mathbb{R}^3$, that have equal projections

onto $\boldsymbol{\mathcal{I}}$ and $\mathbf{i}$, is the plane F perpendicular to the plane E along the bisector B. Thus, the equation of the points Q on the plane F is $Q = a\,(\boldsymbol{\mathcal{I}} + \mathbf{i}) + b\boldsymbol{\mathcal{I}} \times \mathbf{i}$ with $a, b \in \mathbb{R}$. This proves ($\Rightarrow$).

($\Leftarrow$) We just need to consider the dot product of $\mathbf{H}_G$ with $\boldsymbol{\mathcal{I}}$ and $\mathbf{i}$:

$$H_X = \mathbf{H}_G \cdot \boldsymbol{\mathcal{I}} = a\,(\boldsymbol{\mathcal{I}} + \mathbf{i}) \cdot \boldsymbol{\mathcal{I}} + b\boldsymbol{\mathcal{I}} \times \mathbf{i} \cdot \boldsymbol{\mathcal{I}} = a(1 + \mathbf{i} \cdot \boldsymbol{\mathcal{I}})$$

$$H_x = \mathbf{H}_G \cdot \mathbf{i} = a\,(\boldsymbol{\mathcal{I}} + \mathbf{i}) \cdot \mathbf{i} + b\boldsymbol{\mathcal{I}} \times \mathbf{i} \cdot \mathbf{i} = a(\boldsymbol{\mathcal{I}} \cdot \mathbf{i} + 1)\,.$$

Thus, $H_X = H_x$ as claimed. Similar arguments gives the other two expressions. □

For the time derivative we have a similar but slightly more complicated situation. The partial time derivative, with the standard meaning, will be denoted by the usual symbol $\frac{\partial}{\partial t}$. We introduce the time derivative operator $\big[\ \bullet\ \big]^{\bullet}_{\substack{\text{coordinate}\\ \text{system}}}$, meaning time derivative assuming that the displayed coordinate system is fixed. With the help of the angular momentum $\mathbf{H}_G$ this operator is defined by:

$$\big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{O}XYZ} := \text{time derivative of } \mathbf{H}_G \text{ assuming}$$
$$\text{fixed coordinate system } \mathbf{O}XYZ\,, \tag{3.50}$$

$$\big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{G}xyz} := \text{time derivative of } \mathbf{H}_G \text{ assuming}$$
$$\text{fixed coordinate system } \mathbf{G}xyz\,. \tag{3.51}$$

Thus, in terms of the components:

$$\big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{O}XYZ} = \dot{H}_X\,\boldsymbol{\mathcal{I}} + \dot{H}_Y\,\boldsymbol{\mathcal{J}} + \dot{H}_Z\,\boldsymbol{\mathcal{K}}\,, \tag{3.52}$$

$$\big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{G}xyz} = \dot{H}_x\,\mathbf{i} + \dot{H}_y\,\mathbf{j} + \dot{H}_z\,\mathbf{k}\,. \tag{3.53}$$

Recall, however, that although the trihedron $\{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{K}}\}$ is fixed, i.e., does not change with time, the trihedron $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ rotates, i.e., it does change with respect to time, and hence:

$$\frac{\partial \mathbf{H}_G}{\partial t} = \left[\mathbf{H}_G\right]^{\bullet}_{\mathbf{O}XYZ}, \qquad \frac{\partial \mathbf{H}_G}{\partial t} \neq \left[\mathbf{H}_G\right]^{\bullet}_{\mathbf{G}xyz}, \tag{3.54}$$

in general. For the computation of the dynamics of the system we need the time derivative of $\mathbf{H}_G$ relative to the $\mathbf{O}XYZ$ coordinate system. Since the local basis trihedron $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ rotates with angular speed $\boldsymbol{\omega}$ around an axis which goes through $\mathbf{G}$ and is parallel to $\boldsymbol{\omega}$, we have:

$$\begin{aligned}
\left[\mathbf{H}_G\right]^{\bullet}_{\mathbf{O}XYZ} &= \frac{\partial}{\partial t}\left[\mathbf{H}_G\right]_{\mathbf{O}XYZ} = \frac{\partial}{\partial t}\left[\mathbf{H}_G\right]_{\mathbf{G}xyz} \\
&= \frac{\partial}{\partial t}\left(H_x\,\mathbf{i} + H_y\,\mathbf{j} + H_z\,\mathbf{k}\right) \\
&= \frac{\partial H_x}{\partial t}\,\mathbf{i} + \frac{\partial H_y}{\partial t}\,\mathbf{j} + \frac{\partial H_z}{\partial t}\,\mathbf{k} + H_x\,\frac{\partial \mathbf{i}}{\partial t} + H_y\,\frac{\partial \mathbf{j}}{\partial t} + H_z\,\frac{\partial \mathbf{k}}{\partial t} \\
&= \dot{H}_x\,\mathbf{i} + \dot{H}_y\,\mathbf{j} + \dot{H}_z\,\mathbf{k} + H_x\,\frac{\partial \mathbf{i}}{\partial t} + H_y\,\frac{\partial \mathbf{j}}{\partial t} + H_z\,\frac{\partial \mathbf{k}}{\partial t}\,.
\end{aligned} \tag{3.55}$$

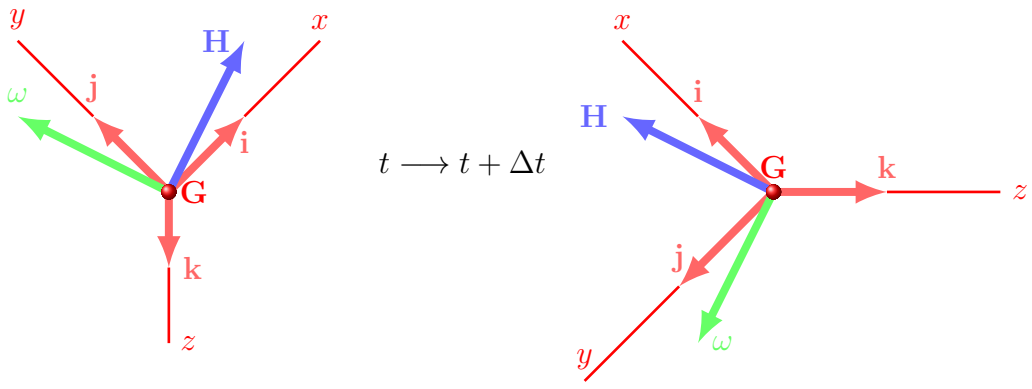


FIGURE 3.2. Evolution from t to $\Delta t = t + 1$ for the local coordinate system.

Approximating the derivatives by the usual finite differences we have:

$$\begin{aligned}
\mathbf{i}\big|_{t+\Delta t} - \mathbf{i}\big|_t &\approx \boldsymbol{\omega} \times \mathbf{i}\big|_t \, \Delta t \,, \\
\mathbf{j}\big|_{t+\Delta t} - \mathbf{j}\big|_t &\approx \boldsymbol{\omega} \times \mathbf{j}\big|_t \, \Delta t \,, \\
\mathbf{k}\big|_{t+\Delta t} - \mathbf{k}\big|_t &\approx \boldsymbol{\omega} \times \mathbf{k}\big|_t \, \Delta t \,,
\end{aligned} \tag{3.56}$$

and hence taking the limit when $\Delta t \to 0$:

$$\frac{\partial \mathbf{i}}{\partial t} = \boldsymbol{\omega} \times \mathbf{i} \,, \qquad \frac{\partial \mathbf{j}}{\partial t} = \boldsymbol{\omega} \times \mathbf{j} \,, \qquad \frac{\partial \mathbf{k}}{\partial t} = \boldsymbol{\omega} \times \mathbf{k} \,, \tag{3.57}$$

Thus, we obtain:

$$\frac{\partial}{\partial t}\Big[\mathbf{H}_G\Big]_{\mathbf{O}XYZ} = \dot{H}_x \, \mathbf{i} + \dot{H}_y \, \mathbf{j} + \dot{H}_z \, \mathbf{k} + \boldsymbol{\omega} \times \mathbf{i} \, H_x + \boldsymbol{\omega} \times \mathbf{j} \, H_y + \boldsymbol{\omega} \times \mathbf{k} \, H_z \,, \tag{3.58}$$

i.e., finally:

$$\big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{O}XYZ} = \big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{G}xyz} + \boldsymbol{\omega} \times \big[\mathbf{H}_G\big]_{\mathbf{G}xyz} \,. \tag{3.59}$$

Now, introducing (3.41) into (3.53) we observe that

$$\big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{G}xyz} = I_{11} \, \dot{\omega}_1 \, \mathbf{i} + I_{22} \, \dot{\omega}_2 \, \mathbf{j} + I_{33} \, \dot{\omega}_3 \, \mathbf{k} \,, \tag{3.60}$$

and hence we get:

$$\sum \mathbf{M}_G = \big[\mathbf{H}_G\big]^{\bullet}_{\mathbf{O}XYZ} = I_{11} \, \dot{\omega}_1 \, \mathbf{i} + I_{22} \, \dot{\omega}_2 \, \mathbf{j} + I_{33} \, \dot{\omega}_3 \, \mathbf{k} + \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \omega_1 & \omega_2 & \omega_3 \\ I_{11} \, \omega_1 & I_{22} \, \omega_2 & I_{33} \, \omega_3 \end{vmatrix} \tag{3.61}$$

Writing this equation componentwise we obtain:

$$\begin{aligned}
\sum M_1 &= I_{11} \, \dot{\omega}_1 + (I_{33} - I_{22}) \, \omega_2 \, \omega_3 \,, \\
\sum M_2 &= I_{22} \, \dot{\omega}_2 + (I_{11} - I_{33}) \, \omega_1 \, \omega_3 \,, \\
\sum M_1 &= I_{33} \, \dot{\omega}_3 + (I_{22} - I_{11}) \, \omega_1 \, \omega_2 \,,
\end{aligned} \tag{3.62}$$

which are (part of) the so called *Euler equations of motion* concerning the rotation motion of the rigid body. The other part of the Euler equations are the three Newton equations concerning the translation motion $\mathbf{F} = m\,\mathbf{a}$.

Now, in our case, since the motion is torque-free, the Euler equation gives:

$$\dot{\mathbf{H}}_G = \sum \mathbf{M}_G = \mathbf{0}\,, \tag{3.63}$$

which means that the direction (and the magnitude) of $\mathbf{H}_G$ remains fixed w.r.t. the fixed coordinate system $\mathbf{O}XYZ$. In this case the equations (3.62) reduce to: (cf. [**65**, p. 205])

$$\begin{aligned} I_{11}\,\dot{\omega}_1 &= (I_{22} - I_{33})\,\omega_2\,\omega_3\,, \\ I_{22}\,\dot{\omega}_2 &= (I_{33} - I_{11})\,\omega_3\,\omega_1\,, \\ I_{33}\,\dot{\omega}_3 &= (I_{11} - I_{22})\,\omega_1\,\omega_2\,, \end{aligned} \tag{3.64}$$

From (3.64) we obtain:

$$\begin{aligned} \left(\omega_1^2\right)^{\bullet} &= \frac{2(I_{22} - I_{33})}{I_{11}}\,\omega_1\,\omega_2\,\omega_3 =: P\,\omega_1\,\omega_2\,\omega_3\,, \\ \left(\omega_2^2\right)^{\bullet} &= \frac{2(I_{33} - I_{11})}{I_{22}}\,\omega_1\,\omega_2\,\omega_3 =: Q\,\omega_1\,\omega_2\,\omega_3\,, \\ \left(\omega_3^2\right)^{\bullet} &= \frac{2(I_{11} - I_{22})}{I_{33}}\,\omega_1\,\omega_2\,\omega_3 =: R\,\omega_1\,\omega_2\,\omega_3\,. \end{aligned} \tag{3.65}$$

It follows that:

$$\left(\omega_2^2\right)^{\bullet} = \frac{Q}{P}\,\left(\omega_1^2\right)^{\bullet}\,, \qquad \left(\omega_3^2\right)^{\bullet} = \frac{R}{P}\,\left(\omega_1^2\right)^{\bullet}\,, \tag{3.66}$$

where:

$$\frac{Q}{P} = \frac{I_{11}(I_{33} - I_{11})}{I_{22}(I_{22} - I_{33})}\,, \qquad \frac{R}{P} = \frac{I_{11}(I_{11} - I_{22})}{I_{33}(I_{22} - I_{33})}\,. \tag{3.67}$$

Thus,

$$\omega_2^2 = a_1^{'} + \frac{Q}{P}\,\omega_1^2\,, \qquad \omega_3^2 = c_1^{'} + \frac{R}{P}\,\omega_1^2\,, \tag{3.68}$$

where $a_1^{'}$ and $c_1^{'}$ are integration constants to be determined. Now the first equation in (3.64) gives:

$$\left(\dot{\omega}_1\right)^2 = \frac{(I_{22}-I_{33})^2}{I_{11}^2}\,\omega_2^2\,\omega_3^3 = \frac{(I_{22}-I_{33})^2}{I_{11}^2}\,\left(a_1^{'} + \frac{Q}{P}\,\omega_1^2\right)\left(c_1^{'} + \frac{R}{P}\,\omega_1^2\right)\,, \tag{3.69}$$

Doing the algebra we get:

$$\left(\dot{\omega}_1\right)^2 = \left(a_1 + b_1\,\omega_1^2\right)\left(c_1 + d_1\,\omega_1^2\right)\,, \tag{3.70}$$

where:

$$a_1 = a_1^{'}\,\frac{I_{22}-I_{33}}{I_{11}} = a_1^{'}\,\frac{P}{2} \qquad b_1 = \frac{I_{33}-I_{11}}{I_{22}} = \frac{Q}{2} \tag{3.71}$$

$$c_1 = c_1^{'}\,\frac{I_{22}-I_{33}}{I_{11}} = c_1^{'}\,\frac{P}{2} \qquad d_1 = \frac{I_{11}-I_{22}}{I_{33}} = \frac{R}{2} \tag{3.72}$$

3.2.1. First Solution. Equation (3.70) is a differential equation related to the Jacobi elliptic functions sn, cn and dn [**171**, p. 492].

From now on, we will consider that $I_{11} < I_{22} < I_{33}$. Although this may look as an over-simplification, it is not since we are free to choose the *name* of the unknowns function $\omega_1(t)$, $\omega_2(t)$ and $\omega_3(t)$ and the name will be selected such that we always satisfy $I_{11} < I_{22} < I_{33}$.

This simplification help us to find the signs of three of the four constants appearing in equation (3.70), thus $a < 0$, $b > 0$ and $d < 0$. Unfortunately, c could be either negative of positive. This will depend of the moments of inertias I_{11}, I_{22} and I_{33} and the initial conditions for $\omega_1(0)$, $\omega_2(0)$ and $\omega_3(0)$. So, we must analyze both cases. We will denote by *Case 1* when $c > 0$ and *Case 2* when $c < 0$.

Before continue, we need to come back to the Jacobi elliptic functions mentioned before. The reason they appear here is because they provide an explicit solution to the equation (3.70) depending on the value of the coefficients a, b, c and d, respectively.

In Table 3.1 we provide the form of the differential equations that are solved by the Jacobi elliptic functions sn, cn and dn, respectively.

Differential Equation	Solution
$(y'(x))^2 = (1-y^2(x))\,(1-k^2\,y^2(x))$ $= 1-(1+k^2)\,y^2(x)+k^2\,y^4(x)$	$y(x)=\mathrm{sn}(x,k^2)$
$(y'(x))^2 = (1-y^2(x))\,(1-k^2+k^2\,y^2(x))$ $= 1-k^2+(-1+2\,k^2)\,y^2(x)-k^2\,y^4(x)$	$y(x)=\mathrm{cn}(x,k^2)$
$(y'(x))^2 = (y^2(x)-1)\,(1-k^2-y^2(x))$ $= -1+k^2+(2-k^2)\,y^2(x)-y^4(x)$	$y(x)=\mathrm{dn}(x,k^2)$

TABLE 3.1. Jacobi differential equations

where k^2 is called the *modulus* and here it is considered to be $0<k^2<1$. Thus, now we need to make a suitable change of variables such that we can transform the equation (3.70) into one of the described differential equations in Table 3.1. To achieve this task, we propose to build the non-dimensional function $\widetilde{\omega}_1(\tau)$, such that $\omega_1(t)=\pm\sqrt{\gamma}\,\widetilde{\omega}_1(\tau)$ and $t=\eta\,\tau-t_0$. Thus we obtain,

$$\gamma\,\eta^2\left(\overset{\bullet}{\widetilde{\omega}}_1\right)^2=\left(a_1+b_1\,\gamma\,\widetilde{\omega}_1^2\right)\left(c_1+d_1\,\gamma\,\widetilde{\omega}_1^2\right)\,,$$

expanding the right-hand-side and moving scaling factors also to right hand side we obtain,

$$\left(\overset{\bullet}{\widetilde{\omega}}_1\right)^2=a_1\,c_1\,\gamma^{-1}\,\eta^{-2}+\eta^{-2}\,(a_1\,d_1+b_1\,c_1)\,\widetilde{\omega}_1^2+\gamma^{-1}\eta^{-2}\,b_1\,d_1\,\widetilde{\omega}_1^4 \tag{3.73}$$

where in this the *over-dot* means derivative with respect to the non-dimensional time τ.

To decide which Jacobi elliptic function will be the solution of (3.70) we need to compare each term of the right-hand-side of equation (3.73) and the right-hand-side of the differential equations in Table 3.1.

To simplify the comparison we provide Table 3.2.

Equal terms	Candidate solution
$a_1\, c_1\, \gamma^{-1}\, \eta^{-2} = 1$ $\eta^{-2}\,(a_1\, d_1 + b_1\, c_1) = -(1+k^2)$ $\gamma^{-1}\eta^{-2}\, b_1\, d_1 = k^2$	$y(x) = \mathrm{sn}(x, k^2)$
$a_1\, c_1\, \gamma^{-1}\, \eta^{-2} = 1 - k^2$ $\eta^{-2}\,(a_1\, d_1 + b_1\, c_1) = -1 + 2\,k^2$ $\gamma^{-1}\eta^{-2}\, b_1\, d_1 = -k^2$	$y(x) = \mathrm{cn}(x, k^2)$
$a_1\, c_1\, \gamma^{-1}\, \eta^{-2} = -1 + k^2$ $\eta^{-2}\,(a_1\, d_1 + b_1\, c_1) = 2 - k^2$ $\gamma^{-1}\eta^{-2}\, b_1\, d_1 = -1$	$y(x) = \mathrm{dn}(x, k^2)$

TABLE 3.2. Comparing term by term with candidate solutions

The next step is to find the constants γ, η and k^2. It is interesting to point out that 2 unknowns of the three equations are provided by the original problem and one unknown is provided by the Jacobi elliptic functions. The problem now has been translated to solution of 3 system of nonlinear equations. Fortunately, these can be solved explicitly.

Before continuing, we would like to point out that we are only interested in real solutions of the nonlinear systems of equations and also require that $0 < k < 1$. Although the latter could be removed, but it is kept to avoid the use of identities of the Jacobi elliptic functions later on and obtain the solution in their *standard* form directly.

The solutions are summarized as follows:

$$\left(\widetilde{\omega}_1(\tau),\eta,\gamma,k^2\right)=\begin{cases}\left(\mathrm{dn}(\tau,k^2),\frac{1}{\sqrt{|a_1\,d_1|}},\left|\frac{a_1}{b_1}\right|,1-\left|\frac{b_1\,c_1}{a_1\,d_1}\right|\right),\\ \text{if } c_1>0 \text{ and } |a_1\,d_1|>|b_1\,c_1|\\ \left(\mathrm{dn}(\tau,k^2),\frac{1}{\sqrt{|b_1\,c_1|}},\left|\frac{c_1}{d_1}\right|,1-\left|\frac{a_1\,d_1}{b_1\,c_1}\right|\right),\\ \text{if } c_1>0 \text{ and } |a_1\,d_1|\le|b_1\,c_1|\\ \left(\mathrm{cn}(\tau,k^2),\frac{1}{\sqrt{|a_1\,d_1|+|b_1\,c_1|}},\left|\frac{a_1}{b_1}\right|,\frac{|a_1\,d_1|}{|a_1\,d_1|+|b_1\,c_1|}\right),\\ \text{if } c_1\le 0\end{cases}\tag{3.74}$$

We can easily find $\omega_1(t)=\sqrt{\gamma}\,\widetilde{\omega}_1(\frac{t+t_0}{\eta})$, where t_0 is obtained directly from the initial condition $t_0=\eta\,\widetilde{\omega}_1^{-1}\left(\frac{\omega_1(0)}{\sqrt{\gamma}}\right)$, respectively. Following the same procedure, we can find $\omega_2(t)$ and $\omega_3(t)$. These are:

$$\left(\widetilde{\omega}_2(\tau),\eta,\gamma,k^2\right)=\begin{cases}\left(\mathrm{sn}(\tau,k^2),\frac{1}{\sqrt{|a_1\,d_1|}},\left|\frac{c_2}{d_2}\right|,\left|\frac{b_2\,c_2}{a_2\,d_2}\right|\right) & \text{if } |a_2\,d_2|>|b_2\,c_2|\\ \left(\mathrm{sn}(\tau,k^2),\frac{1}{\sqrt{|b_2\,c_2|}},\left|\frac{a_2}{b_2}\right|,\left|\frac{a_2\,d_2}{b_2\,c_2}\right|\right) & \text{if } |a_2\,d_2|\le|b_2\,c_2|\end{cases}\tag{3.75}$$

and

$$\left(\widetilde{\omega}_3(\tau),\eta,\gamma,k^2\right)=\begin{cases}\left(\mathrm{dn}(\tau,k^2),\frac{1}{\sqrt{|a_3\,d_3|}},\left|\frac{a_3}{b_3}\right|,1-\left|\frac{b_3\,c_3}{a_3\,d_3}\right|\right),\\ \text{if } a_3>0 \text{ and } |a_3\,d_3|>|b_3\,c_3|\\ \left(\mathrm{dn}(\tau,k^2),\frac{1}{\sqrt{|b_3\,c_3|}},\left|\frac{c_3}{d_3}\right|,1-\left|\frac{a_3\,d_3}{b_3\,c_3}\right|\right),\\ \text{if } a_3>0 \text{ and } |a_3\,d_3|\le|b_3\,c_3|\\ \left(\mathrm{cn}(\tau,k^2),\frac{1}{\sqrt{|a_3\,d_3|+|b_3\,c_3|}},\left|\frac{c_3}{d_3}\right|,\frac{|b_3\,c_3|}{|a_3\,d_3|+|b_3\,c_3|}\right),\\ \text{if } a_3\le 0\end{cases}\tag{3.76}$$

where t_0 for $\omega_2(t)$ and $\omega_3(t)$ can be found in the same fashion as we did for $\omega_1(t)$. Finally, we provide a table with the explicit definition of the coefficients used for the analytical solutions.

Coefficient	Value
a_1	$-\dfrac{Q\,\omega_1(0)^2 + \lvert P\rvert\,\omega_2(0)^2}{2}$
b_1	$Q/2$
c_1	$\dfrac{\lvert R\rvert\,\omega_1(0)^2 - \lvert P\rvert\,\omega_3(0)^2}{2}$
d_1	$R/2$
a_2	$-a_1$
b_2	$P/2$
c_2	$\dfrac{\lvert R\rvert\,\omega_2(0)^2 + Q\,\omega_3(0)^2}{2}$
d_2	d_1
a_3	$-c_1$
b_3	b_2
c_3	$-c_2$
d_3	b_1

TABLE 3.3. Coefficients used to find the analytical solutions for $\omega_1(t)$, $\omega_2(t)$, and $\omega_3(t)$.

We can clearly see from table that the sign of c_1 could be positive or negative, and this effect is also valid for a_3 but it is the opposite sign of c_1. All the other coefficients has well defined sign.

3.2.2. Second Solution. Recalling equation (3.70) which is a differential equation related to the Jacobi elliptic functions, Mathematica **[6]** readily gives solutions in terms of these functions,

$$\omega_1(t) = \pm i \sqrt{\frac{a_1}{b_1}}\, \mathrm{sn}\left(i\sqrt{b_1\, c_1}\ (t \mp k_1)\ |\, m_{\omega_1} \right), \tag{3.77}$$

where $i^2 = -1$, k_1 is an integration constant and,

$$m_{\omega_1} = \frac{a_1\, d_1}{b_1\, c_1} = a_1^{'}\ \frac{P}{2}\ \frac{R}{2}\ \frac{2}{Q}\ \frac{2}{P}\ \frac{1}{c_1^{'}} = \frac{a_1^{'}}{c_1^{'}}\ \frac{R}{Q}. \tag{3.78}$$

where P,R and Q are determined by (3.65) and a_1', c_1' by (3.68). Similar results can be obtained for $\omega_2(t)$ and $\omega_3(t)$. It follows from (3.64) that,

$$\left(\omega_1^2\right)^{\bullet} = \frac{P}{Q}\,\left(\omega_2^2\right)^{\bullet}\,, \qquad \left(\omega_1^2\right)^{\bullet} = \frac{R}{Q}\,\left(\omega_3^2\right)^{\bullet}\,, \tag{3.79}$$

Thus,

$$\omega_1^2 = a_2^{'} + \frac{P}{Q}\,\omega_2^2\,, \qquad \omega_3^2 = c_2^{'} + \frac{R}{Q}\,\omega_2^2\,, \tag{3.80}$$

where $a_2^{'}$, $c_2^{'}$ are integration constants to be determined. Now the second equation in (3.64) gives,

$$\left(\dot{\omega}_2\right)^2 = \frac{(I_{33}-I_{11})^2}{I_{22}^2}\,\omega_1^2\,\omega_3^2 = \frac{Q^2}{4}\,\left(a_2^{'} + \frac{P}{Q}\,\omega_2^2\right)\left(c_2^{'} + \frac{R}{Q}\,\omega_2^2\right)\,, \tag{3.81}$$

Doing the algebra we get,

$$\left(\dot{\omega}_1\right)^2 = \left(a_2 + b_2\,\omega_2^2\right)\left(c_2 + d_2\,\omega_2^2\right)\,, \tag{3.82}$$

where,

$$a_2 = a_2^{'}\,\frac{Q}{2}\,, \qquad b_2 = \frac{P}{2}\,, \qquad c_2 = c_2^{'}\,\frac{Q}{2}\,, \qquad d_2 = \frac{R}{2}\,. \tag{3.83}$$

Analogously as (3.77), we get

$$\omega_2(t) = \pm i\,\sqrt{\frac{a_2}{b_2}}\,\mathrm{sn}\left(i\sqrt{b_2\,c_2}\,\left(t \mp k_2\right)\,|\,m_{\omega_2}\right)\,, \tag{3.84}$$

where $i^2 = -1$, k_2 is an integration constant and,

$$m_{\omega_2} = \frac{a_2\,d_2}{b_2\,c_2} = a_2^{'}\,\frac{Q}{2}\,\frac{R}{2}\,\frac{2}{P}\,\frac{2}{Q}\,\frac{1}{c_2^{'}} = \frac{a_2^{'}}{c_2^{'}}\,\frac{R}{P}. \tag{3.85}$$

It follows from (3.64) that,

$$\left(\omega_1^2\right)^{\bullet} = \frac{P}{R}\,\left(\omega_3^2\right)^{\bullet}\,, \qquad \left(\omega_3^2\right)^{\bullet} = \frac{Q}{R}\,\left(\omega_2^2\right)^{\bullet}\,, \tag{3.86}$$

Thus,

$$\omega_1^2 = a_3^{'} + \frac{P}{R}\,\omega_3^2\,, \qquad \omega_2^2 = c_3^{'} + \frac{Q}{R}\,\omega_3^2\,, \tag{3.87}$$

where $a_3^{'}$, $c_3^{'}$ are integration constants to be determined. Now the third equation in (3.64) gives,

$$\left(\dot{\omega}_3\right)^2 = \frac{(I_{11}-I_{22})^2}{I_{33}^2}\,\omega_1^2\,\omega_2^2 = \frac{R^2}{4}\,\left(a_3^{'} + \frac{P}{R}\,\omega_3^2\right)\left(c_3^{'} + \frac{Q}{R}\,\omega_3^2\right)\,, \tag{3.88}$$

Doing the algebra we get,

$$\left(\dot{\omega}_3\right)^2 = \left(a_3 + b_3\,\omega_3^2\right)\left(c_3 + d_3\,\omega_3^2\right)\,, \tag{3.89}$$

where,

$$a_3 = a_3^{'}\,\frac{R}{2}\,, \qquad b_3 = \frac{P}{2}\,, \qquad c_3 = c_3^{'}\,\frac{R}{2}\,, \qquad d_3 = \frac{Q}{2}\,. \tag{3.90}$$

Analogously as (3.77) and (3.84), we get

$$\omega_3(t) = \pm i\,\sqrt{\frac{a_3}{b_3}}\,\mathrm{sn}\left(i\sqrt{b_3\,c_3}\;(t \mp k_3)\;|\,m_{\omega_3}\right)\,, \tag{3.91}$$

where $i^2 = -1$, k_3 is an integration constant and

$$m_{\omega_3} = \frac{a_3\,d_3}{b_3\,c_3} = a_3^{'}\,\frac{R}{2}\,\frac{Q}{2}\,\frac{2}{P}\,\frac{2}{R}\,\frac{1}{c_3^{'}} = \frac{a_3^{'}}{c_3^{'}}\,\frac{Q}{P}. \tag{3.92}$$

Summarizing, we obtain the following equations,

$$\omega_1(t) = \pm i\,\sqrt{\frac{a_1}{b_1}}\,\mathrm{sn}\left(i\sqrt{b_1\,c_1}\;(t \mp k_1)\;|\,m_{\omega_1}\right)\,, \tag{3.93}$$

$$\omega_2(t) = \pm i\,\sqrt{\frac{a_2}{b_2}}\,\mathrm{sn}\left(i\sqrt{b_2\,c_2}\;(t \mp k_2)\;|\,m_{\omega_2}\right)\,, \tag{3.94}$$

$$\omega_3(t) = \pm i\,\sqrt{\frac{a_3}{b_3}}\,\mathrm{sn}\left(i\sqrt{b_3\,c_3}\;(t \mp k_3)\;|\,m_{\omega_3}\right)\,, \tag{3.95}$$

Analyzing these functions, not always are real, in fact, depend of P, Q and R values. Then, we obtain the following combinations depending of the inertia values,

CASE	P	Q	R
1. $I_{11} < I_{22} < I_{33}$	$-$	$+$	$-$
2. $I_{11} < I_{33} < I_{22}$	$+$	$+$	$-$
3. $I_{22} < I_{11} < I_{33}$	$-$	$+$	$+$
4. $I_{22} < I_{33} < I_{11}$	$-$	$-$	$+$
5. $I_{33} < I_{11} < I_{22}$	$+$	$-$	$-$
6. $I_{33} < I_{22} < I_{11}$	$+$	$-$	$+$

(3.96)

For example, we observe that in the case 1, $P < 0, Q > 0$ and $R < 0$, and for each function in (3.93),(3.94) and 3.95, we have:

$$\begin{aligned}
&a_1 < 0 \quad b_1 > 0 \quad c_1 > 0 \quad \Rightarrow \quad \tfrac{a_1}{b_1} < 0 \quad b_1 c_1 > 0 \quad \Rightarrow \quad \omega_1(t) \notin \mathbb{R} \\
&a_1 < 0 \quad b_1 > 0 \quad c_1 < 0 \quad \Rightarrow \quad \tfrac{a_1}{b_1} < 0 \quad b_1 c_1 < 0 \quad \Rightarrow \quad \omega_1(t) \in \mathbb{R} \\
&a_2 > 0 \quad b_2 < 0 \quad c_2 > 0 \quad \Rightarrow \quad \tfrac{a_2}{b_2} < 0 \quad b_2 c_2 < 0 \quad \Rightarrow \quad \omega_2(t) \in \mathbb{R} \\
&a_3 < 0 \quad b_3 < 0 \quad c_3 < 0 \quad \Rightarrow \quad \tfrac{a_3}{b_3} > 0 \quad b_3 c_3 > 0 \quad \Rightarrow \quad \omega_3(t) \notin \mathbb{R} \\
&a_3 > 0 \quad b_3 < 0 \quad c_3 < 0 \quad \Rightarrow \quad \tfrac{a_3}{b_3} < 0 \quad b_3 c_3 > 0 \quad \Rightarrow \quad \omega_3(t) \notin \mathbb{R}
\end{aligned} \tag{3.97}$$

Thus, only $\omega_2(t)$ is real unique for this case, therefore, explicitely is the equation (3.94),

$$\omega_2(t) = \pm i \sqrt{\frac{a_2}{b_2}} \operatorname{sn}\left(i\sqrt{b_2\, c_2}\ (t \mp k_2) \mid m_{\omega_2} \right), \tag{3.98}$$

but we know that $\frac{a_2}{b_2} < 0$ and $b_2 c_2 < 0$, therefore,

$$\omega_2(t) = \mp\sqrt{-\frac{a_2}{b_2}}\,\mathrm{sn}\left(-\sqrt{-b_2\,c_2}\;(t \mp k_2)\,|\,m_{\omega_2}\right), \tag{3.99}$$

$$= \mp\alpha_2\,\mathrm{sn}\left(-\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right), \tag{3.100}$$

$$= \pm\alpha_2\,\mathrm{sn}\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right), \tag{3.101}$$

Now, we need to obtain an explicit expression for the others two velocities $\omega_1(t)$ and $\omega_3(t)$. The first equation in (3.80) gives:

$$\begin{aligned}
\omega_1^2 &= a_2^{'} + \frac{P}{Q}\omega_2^2 = a_2^{'} + \frac{P}{Q}\left(\pm\alpha_2\,\mathrm{sn}\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right)\right)^2 \\
&= a_2^{'} + \frac{P}{Q}\;\alpha_2^2\,\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) = a_2^{'} + \frac{P}{Q}\;\left(-\frac{a_2}{b_2}\right)\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) \\
&= a_2^{'} - \frac{P}{Q}\;\frac{a_2}{b_2}\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) = a_2^{'} - \frac{P}{Q}\;a_2^{'}\frac{Q}{2}\;\frac{2}{P}\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) \\
&= a_2^{'}\left(1 - \mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right)\right) = a_2^{'}\;\mathrm{cn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right)
\end{aligned}$$

Finally we get

$$\omega_1 = \pm\sqrt{a_2^{'}}\;\mathrm{cn}\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) \tag{3.102}$$

Then we have that $\alpha_1 = \sqrt{a_2^{'}}$. Now the second equation in (3.80) gives:

$$\begin{aligned}
\omega_3^2 &= c_2^{'} + \frac{R}{Q}\omega_2^2 = c_2^{'} + \frac{R}{Q}\left(\pm\alpha_2\,\mathrm{sn}\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right)\right)^2 \\
&= c_2^{'} + \frac{R}{Q}\;\alpha_2^2\,\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) = c_2^{'} + \frac{R}{Q}\;\left(-\frac{a_2}{b_2}\right)\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) \\
&= c_2^{'} - \frac{R}{Q}\;\frac{a_2}{b_2}\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) = c_2^{'} - \frac{R}{Q}\;a_2^{'}\frac{Q}{2}\;\frac{2}{P}\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) \\
&= c_2^{'} - a_2^{'}\;\frac{R}{P}\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) = c_2^{'} - c_2^{'}\;m_{\omega_2}\;\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right) \\
&= c_2^{'}\left(1 - m_{\omega_2}\mathrm{sn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right)\right) = c_2^{'}\;\mathrm{dn}^2\left(\beta_2\;(t \mp k_2)\,|\,m_{\omega_2}\right)
\end{aligned}$$

Finally we get

$$\omega_3 = \pm\sqrt{c_2'}\,\mathrm{dn}\left(\beta_2\ (t \mp k_2)\ |\, m_{\omega_2}\right) \tag{3.103}$$

Then we have that $\alpha_3 = \sqrt{c_2'}$.

An important point is mention that m_{ω_2} is always positive:

$$m_{\omega_2} = \frac{a_2'}{c_2'}\,\frac{R}{P} > 0 \tag{3.104}$$

Then, if $0 < m_{\omega_2} < 1$, the velocities are:

$$\begin{aligned}
\omega_1(t) &= \pm\sqrt{a_2'}\,\mathrm{cn}\left(\beta_2\ (t \mp k_2)\ |\, m_{\omega_2}\right) \\
\omega_2(t) &= \pm\alpha_2\,\mathrm{sn}\left(\beta_2\ (t \mp k_2)\ |\, m_{\omega_2}\right), \\
\omega_3(t) &= \pm\sqrt{c_2'}\,\mathrm{dn}\left(\beta_2\ (t \mp k_2)\ |\, m_{\omega_2}\right)
\end{aligned} \tag{3.105}$$

Now, if $m_{\omega_2} > 1$, the velocities are [**11**, p. 573]:

$$\begin{aligned}
\omega_1(t) &= \pm\sqrt{a_2'}\,\mathrm{dn}\left(\sqrt{m_{\omega_2}}\ \beta_2\ (t \mp k_2)\ \middle|\ \frac{1}{m_{\omega_2}}\right) \\
\omega_2(t) &= \pm\alpha_2\,\frac{1}{\sqrt{m_{\omega_2}}}\,\mathrm{sn}\left(\sqrt{m_{\omega_2}}\beta_2\ (t \mp k_2)\ \middle|\ \frac{1}{m_{\omega_2}}\right), \\
\omega_3(t) &= \pm\sqrt{c_2'}\,\mathrm{cn}\left(\sqrt{m_{\omega_2}}\ \beta_2\ (t \mp k_2)\ \middle|\ \frac{1}{m_{\omega_2}}\right)
\end{aligned} \tag{3.106}$$

where choosing the sign depends on the initial values.

3.2.3. Comparison. The solutions with the form $\omega(t) = \pm\sqrt{\gamma}\,\widetilde{\omega}\left(\dfrac{t+t_0}{\eta}\right)$, are summarized as follows:

$$\left(\widetilde{\omega}_1(\tau), \eta, \gamma, k^2\right) = \begin{cases} \left(\mathrm{dn}(\tau, k^2), \frac{1}{\sqrt{|a_1\,d_1|}}, \left|\frac{a_1}{b_1}\right|, 1 - \left|\frac{b_1\,c_1}{a_1\,d_1}\right|\right), \\ \text{if } c_1 > 0 \text{ and } |a_1\,d_1| > |b_1\,c_1| \\ \left(\mathrm{dn}(\tau, k^2), \frac{1}{\sqrt{|b_1\,c_1|}}, \left|\frac{c_1}{d_1}\right|, 1 - \left|\frac{a_1\,d_1}{b_1\,c_1}\right|\right), \\ \text{if } c_1 > 0 \text{ and } |a_1\,d_1| \leq |b_1\,c_1| \\ \left(\mathrm{cn}(\tau, k^2), \frac{1}{\sqrt{|a_1\,d_1| + |b_1\,c_1|}}, \left|\frac{a_1}{b_1}\right|, \frac{|a_1\,d_1|}{|a_1\,d_1| + |b_1\,c_1|}\right), \\ \text{if } c_1 \leq 0 \end{cases} \tag{3.107}$$

,

$$\left(\widetilde{\omega}_2(\tau), \eta, \gamma, k^2\right) = \begin{cases} \left(\mathrm{sn}(\tau, k^2), \frac{1}{\sqrt{|a_1\,d_1|}}, \left|\frac{c_2}{d_2}\right|, \left|\frac{b_2\,c_2}{a_2\,d_2}\right|\right) & \text{if } |a_2\,d_2| > |b_2\,c_2| \\ \left(\mathrm{sn}(\tau, k^2), \frac{1}{\sqrt{|b_2\,c_2|}}, \left|\frac{a_2}{b_2}\right|, \left|\frac{a_2\,d_2}{b_2\,c_2}\right|\right) & \text{if } |a_2\,d_2| \leq |b_2\,c_2| \end{cases} \tag{3.108}$$

and

$$\left(\widetilde{\omega}_3(\tau), \eta, \gamma, k^2\right) = \begin{cases} \left(\mathrm{dn}(\tau, k^2), \frac{1}{\sqrt{|a_3\,d_3|}}, \left|\frac{a_3}{b_3}\right|, 1 - \left|\frac{b_3\,c_3}{a_3\,d_3}\right|\right), \\ \text{if } a_3 > 0 \text{ and } |a_3\,d_3| > |b_3\,c_3| \\ \left(\mathrm{dn}(\tau, k^2), \frac{1}{\sqrt{|b_3\,c_3|}}, \left|\frac{c_3}{d_3}\right|, 1 - \left|\frac{a_3\,d_3}{b_3\,c_3}\right|\right), \\ \text{if } a_3 > 0 \text{ and } |a_3\,d_3| \leq |b_3\,c_3| \\ \left(\mathrm{cn}(\tau, k^2), \frac{1}{\sqrt{|a_3\,d_3| + |b_3\,c_3|}}, \left|\frac{c_3}{d_3}\right|, \frac{|b_3\,c_3|}{|a_3\,d_3| + |b_3\,c_3|}\right), \\ \text{if } a_3 \leq 0 \end{cases} \tag{3.109}$$

where t_0 for $\omega_2(t)$ and $\omega_3(t)$ can be found in the same fashion as we did for $\omega_1(t)$. Taking into account the first equation in (3.108) and the right equation of (3.80), we

have

$$\begin{aligned}
\omega_3^2 &= c_2^{'} + \frac{R}{Q}\,\omega_2^2 = c_2^{'} + \frac{R}{Q}\,\left(\sqrt{\left|\frac{c_2}{d_2}\right|}\mathrm{sn}(\tau,k^2)\right)^2 = c_2^{'} + \frac{R}{Q}\,\left|\frac{c_2}{d_2}\right|\mathrm{sn}^2(\tau,k^2) \\
&= c_2^{'} + \frac{R}{Q}\,\left|\frac{c_2^{'}\,\frac{Q}{2}}{\frac{R}{2}}\right|\mathrm{sn}^2(\tau,k^2) = c_2^{'} + \frac{R}{Q}\,\left|\frac{c_2^{'}\,Q}{R}\right|\mathrm{sn}^2(\tau,k^2) \\
&= c_2^{'} - \left|\frac{R}{Q}\right|\,\left|\frac{c_2^{'}\,Q}{R}\right|\mathrm{sn}^2(\tau,k^2) = c_2^{'} - \left|\frac{R}{Q}\,\frac{c_2^{'}\,Q}{R}\right|\mathrm{sn}^2(\tau,k^2) \\
&= c_2^{'} - |c_2^{'}|\mathrm{sn}^2(\tau,k^2) = c_2^{'}\left(1-\mathrm{sn}^2(\tau,k^2)\right)
\end{aligned}$$

Finally,

$$\omega_3^2 = c_2^{'}\,\mathrm{cn}^2(\tau,k^2) \tag{3.110}$$

Then, we have obtained the third function in equation (3.109). The problem which arises now, is to verify if the coefficients for both functions are equals. First, it verifies if γ coefficients are equal.

In (3.110), γ coefficient is given by

$$c_2^{'} = \omega_3^2 - \frac{R}{Q}\,\omega_2^2 \tag{3.111}$$

On the other hand, in (3.109), γ coefficient is given by

$$\left|\frac{c_3}{d_3}\right| = \left|\frac{c_3^{'}\,\frac{R}{2}}{\frac{Q}{2}}\right| = \left|\frac{c_3^{'}\,R}{Q}\right| = -\frac{c_3^{'}\,R}{Q} = -\left(\omega_2^2 - \frac{Q}{R}\,\omega_3^2\right)\frac{R}{Q} = \omega_3^2 - \frac{R}{Q}\,\omega_2^2 \tag{3.112}$$

Therefore, γ coefficient is the same. Second, it verifies if k^2 coefficients are equal. In (3.108), k^2 is given by

$$\left|\frac{b_2\,c_2}{a_2\,d_2}\right| = \left|\frac{\frac{P}{2}\,\frac{c_2^{'}\,Q}{2}}{\frac{a_2^{'}\,Q}{2}\,\frac{R}{2}}\right| = \frac{P\,c_2^{'}}{a_2^{'}\,R} = \frac{P}{R}\,\frac{\left(\omega_3^2 - \frac{R}{Q}\,\omega_2^2\right)}{\left(\omega_1^2 - \frac{P}{Q}\,\omega_2^2\right)}$$

On the other hand, in (3.109), k^2 is given by

$$\begin{aligned}
\frac{|b_3\,c_3|}{|a_3\,d_3|+|b_3\,c_3|} &= \frac{\left|\frac{P}{2}\,\frac{c_3^{'}R}{2}\right|}{\left|\frac{a_3^{'}R}{2}\,\frac{Q}{2}\right|+\left|\frac{P}{2}\,\frac{c_3^{'}R}{2}\right|} = \frac{P\,c_3^{'}}{P\,c_3^{'}-Q\,a_3^{'}} = \frac{P\left(\omega_2^2-\frac{Q}{R}\,\omega_3^2\right)}{P\left(\omega_2^2-\frac{Q}{R}\,\omega_3^2\right)-Q\left(\omega_1^2-\frac{P}{R}\,\omega_3^2\right)} \\
&= \frac{P\left(\omega_2^2-\frac{Q}{R}\,\omega_3^2\right)}{P\,\omega_2^2-\frac{P\,Q}{R}\,\omega_3^2-Q\,\omega_1^2+\frac{P\,Q}{R}\,\omega_3^2} = \frac{P\left(\omega_2^2-\frac{Q}{R}\,\omega_3^2\right)}{P\,\omega_2^2-Q\,\omega_1^2} \\
&= \frac{P\left(\frac{R}{Q}\,\omega_2^2-\omega_3^2\right)}{P\,\frac{R}{Q}\,\omega_2^2-R\,\omega_1^2} = \frac{-P\left(\omega_3^2-\frac{R}{Q}\,\omega_2^2\right)}{-R\left(\omega_1^2-\frac{P}{Q}\,\omega_2^2\right)} = \frac{P\left(\omega_3^2-\frac{R}{Q}\,\omega_2^2\right)}{R\left(\omega_1^2-\frac{P}{Q}\,\omega_2^2\right)}
\end{aligned}$$

Therefore, k^2 parameter is the same. Third, it verifies if η coefficients are equal. In (3.108), η is given by

$$\frac{1}{\sqrt{|a_1\,d_1|}} = \frac{1}{\sqrt{\left|a_1^{'}\,\frac{P}{2}\,\frac{R}{2}\right|}} = \frac{2}{\sqrt{a_1^{'}\,P\,R}} = \frac{2}{\sqrt{\left(\omega_2^2-\frac{Q}{P}\,\omega_1^2\right)\,P\,R}} = \frac{2}{\sqrt{R\,(P\,\omega_2^2-Q\,\omega_1^2)}}$$

On the other hand, in (3.109), η is given by

$$\begin{aligned}
\frac{1}{\sqrt{|a_3\,d_3|+|b_3\,c_3|}} &= \frac{1}{\sqrt{\left|a_3^{'}\,\frac{R}{2}\,\frac{Q}{2}\right|+\left|c_3^{'}\,\frac{P}{2}\,\frac{R}{2}\right|}} = \frac{1}{\sqrt{c_3^{'}\,\frac{P\,R}{4}-a_3^{'}\,\frac{R\,Q}{4}}} \\
&= \frac{2}{\sqrt{c_3^{'}\,P\,R-a_3^{'}\,R\,Q}} = \frac{2}{\sqrt{\left(\omega_2^2-\frac{Q}{R}\omega_3^2\right)\,P\,R-\left(\omega_1^2-\frac{P}{R}\omega_3^2\right)\,R\,Q}} \\
&= \frac{2}{\sqrt{P\,R\,\omega_2^2-P\,Q\,\omega_3^2-R\,Q\,\omega_1^2+P\,Q\,\omega_3^2}} \\
&= \frac{2}{\sqrt{R\,(P\,\omega_2^2-Q\,\omega_1^2)}}
\end{aligned}$$

Therefore, η parameter is the same. Now, taking into account the first equation in (3.108) and the left equation of (3.80), we have

$$\begin{aligned}\omega_1^2 &= a_2^{'} + \frac{P}{Q}\,\omega_2^2 = a_2^{'} + \frac{P}{Q}\,\left(\sqrt{\left|\frac{c_2}{d_2}\right|}\mathrm{sn}(\tau,k^2)\right)^2 = a_2^{'} + \frac{P}{Q}\,\left|\frac{c_2}{d_2}\right|\mathrm{sn}^2(\tau,k^2)\\ &= a_2^{'} + \frac{P}{Q}\,\left|\frac{c_2^{'}\,\frac{Q}{2}}{\frac{R}{2}}\right|\mathrm{sn}^2(\tau,k^2) = a_2^{'} + \frac{P}{Q}\,\left|\frac{c_2^{'}\,Q}{R}\right|\mathrm{sn}^2(\tau,k^2)\\ &= a_2^{'} - \left|\frac{P}{Q}\right|\,\left|\frac{c_2^{'}\,Q}{R}\right|\mathrm{sn}^2(\tau,k^2) = a_2^{'} - \left|\frac{P}{R}\,c_2^{'}\right|\mathrm{sn}^2(\tau,k^2)\end{aligned}$$

We know that,

$$k^2 = \left|\frac{b_2\,c_2}{a_2\,d_2}\right| = \left|\frac{\frac{P}{2}\,\frac{c_2^{'}\,Q}{2}}{\frac{a_2^{'}\,Q}{2}\,\frac{R}{2}}\right| = \frac{P\,c_2^{'}}{a_2^{'}\,R}$$

which implies that

$$a_2^{'}\,k^2 = \frac{P\,c_2^{'}}{R}$$

Replacing in ω_1^2, it results that

$$\omega_1^2 = a_2^{'} - \left|\frac{P}{R}\,c_2^{'}\right|\mathrm{sn}^2(\tau,k^2) = a_2^{'} - \left|a_2^{'}\,k^2\right|\mathrm{sn}^2(\tau,k^2) = a_2^{'}\,\left(1 - \,k^2\mathrm{sn}^2(\tau,k^2)\right)$$

Finally,

$$\omega_1^2 = a_2^{'}\,\mathrm{dn}^2(\tau,k^2)$$

However, we can notice that there are two functions dn in (3.107). If we analyze the second function dn, we can note a condition, which is $c_1 > 0$ and $|a_1\,d_1| \leq |b_1\,c_1|$, therefore we have the following

$$|a_1\,d_1| \leq |b_1\,c_1| \Rightarrow \left|a_1^{'}\,\frac{P}{2}\,\frac{R}{2}\right| \leq \left|\frac{Q}{2}\,c_1^{'}\,\frac{P}{2}\right| \Rightarrow \left|\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)\,\frac{P}{2}\,\frac{R}{2}\right| \leq \left|\left(\omega_3^2 - \frac{R}{P}\,\omega_1^2\right)\,\frac{Q}{2}\,\frac{P}{2}\right|$$

$$
\begin{aligned}
\left|\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)\,\frac{P}{2}\,\frac{R}{2}\right| &\leq \left|\left(\omega_3^2 - \frac{R}{P}\,\omega_1^2\right)\,\frac{Q}{2}\,\frac{P}{2}\right| \\
\left|\left(\omega_2^2 + \frac{Q}{|P|}\,\omega_1^2\right)\,\frac{|P|}{2}\,\frac{|R|}{2}\right| &\leq \left|\left(\omega_3^2 - \frac{|R|}{|P|}\,\omega_1^2\right)\,\frac{Q}{2}\,\left(\frac{-|P|}{2}\right)\right| \\
\left(\omega_2^2 + \frac{Q}{|P|}\,\omega_1^2\right)\,\frac{|P|}{2}\,\frac{|R|}{2} &\leq \left|\left(\frac{|R|}{2}\,\omega_1^2 - \frac{|P|}{2}\omega_3^2\right)\,\frac{Q}{2}\right| \\
\left(\frac{|P|}{2}\,\omega_2^2 + \frac{Q}{2}\,\omega_1^2\right)\,\frac{|R|}{2} &\leq \left(\frac{|R|}{2}\,\omega_1^2 - \frac{|P|}{2}\omega_3^2\right)\,\frac{Q}{2} \\
\frac{|P|}{2}\,\frac{|R|}{2}\,\omega_2^2 + \frac{Q}{2}\,\frac{|R|}{2}\,\omega_1^2 &\leq \frac{Q}{2}\,\frac{|R|}{2}\,\omega_1^2 - \frac{Q}{2}\,\frac{|P|}{2}\omega_3^2
\end{aligned}
$$

Finally we obtain

$$
\frac{|P|}{2}\,\frac{|R|}{2}\,\omega_2^2 \leq -\frac{Q}{2}\,\frac{|P|}{2}\omega_3^2 \tag{3.113}
$$

This result can be analyzing by two scenarios: (a) $P = 0$, in this case, $\omega_1(t)$ is constant, therefore is $\omega_1(0)$, and (b) $P \neq 0$, in this case, we have $|R|\,\omega_2^2 \leq -Q\,\omega_3^2$, which is not possible, because $Q > 0$.

Therefore, we must choose the first dn function and verify if η, γ and k^2 parameters fit.

In the case of η parameter, we can note that for the first dn function is the same η as in (3.108).

In the case of γ parameter for $\omega_1(t)$, we have,

$$
\gamma = a_2^{'} = \omega_1^2 - \frac{P}{Q}\,\omega_2^2
$$

On the other hand, γ parameter in (3.107) is given by

$$
\begin{aligned}
\left|\frac{a_1}{b_1}\right| &= \left|\frac{a_1^{'}\,\dfrac{P}{2}}{\dfrac{Q}{2}}\right| = \left|\frac{a_1^{'}\,P}{Q}\right| \\
&= \left|\frac{P}{Q}\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)\right| = \frac{-|P|}{Q}\,\left(\omega_2^2 + \frac{Q}{|P|}\,\omega_1^2\right) = \omega_1^2 - \frac{P}{Q}\,\omega_2^2
\end{aligned}
$$

Therefore, γ parameter is the same. In the case of k^2, we have that sn function is given by

$$\left|\frac{b_2\,c_2}{a_2\,d_2}\right| = \frac{P}{R}\,\frac{\left(\omega_3^2 - \frac{R}{Q}\,\omega_2^2\right)}{\left(\omega_1^2 - \frac{P}{Q}\,\omega_2^2\right)} \tag{3.114}$$

On the other hand, dn function is given by

$$\begin{aligned}
1 - \frac{|b_1\,c_1|}{|a_1\,d_1|} &= 1 - \frac{\left|\frac{Q}{2}\,c_1^{'}\,\frac{P}{2}\right|}{\left|a_1^{'}\,\frac{P}{2}\,\frac{R}{2}\right|} = 1 - \frac{\left|c_1^{'}\,Q\right|}{\left|a_1^{'}\,R\right|} = 1 - \left(\frac{-|c_1^{'}|\,Q}{-|R|\,a_1^{'}}\right) = 1 - \left(\frac{|c_1^{'}|\,Q}{|R|\,a_1^{'}}\right) \\
&= 1 - \frac{Q\,\left(\omega_3^2 - \frac{R}{P}\,\omega_1^2\right)}{R\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)} = \frac{R\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right) - Q\,\left(\omega_3^2 - \frac{R}{P}\,\omega_1^2\right)}{R\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)} \\
&= \frac{R\,\omega_2^2 - \frac{Q\,R}{P}\,\omega_1^2 - Q\,\omega_3^2 + \frac{Q\,R}{P}\,\omega_1^2}{R\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)} = \frac{R\,\omega_2^2 - Q\,\omega_3^2}{R\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)} \\
&= \frac{\frac{P}{Q}}{\frac{P}{Q}}\,\frac{R\,\omega_2^2 - Q\,\omega_3^2}{R\,\left(\omega_2^2 - \frac{Q}{P}\,\omega_1^2\right)} = \frac{\frac{P\,R}{Q}\,\omega_2^2 - P\,\omega_3^2}{R\,\left(\frac{P}{Q}\omega_2^2 - \omega_1^2\right)} = \frac{-P\,\left(\omega_3^2 - \frac{R}{Q}\,\omega_2^2\right)}{-R\,\left(\omega_1^2 - \frac{P}{Q}\omega_2^2\right)} \\
&= \frac{P\,\left(\omega_3^2 - \frac{R}{Q}\,\omega_2^2\right)}{R\,\left(\omega_1^2 - \frac{P}{Q}\omega_2^2\right)}
\end{aligned}$$

Therefore, k^2 parameter is the same. Summarizing we have $\omega(t) = \pm\sqrt{\gamma}\,\widetilde{\omega}\left(\frac{t+t_0}{\eta}\right)$ represented by $(\widetilde{\omega}(\tau), \eta, \gamma, k^2)$, where $t = \eta\,\tau - t_0$ and k^2 is the modulus,

$$\begin{aligned}
\left(\widetilde{\omega}_1(\tau), \eta, \gamma, k^2\right) &= \left(\mathrm{dn}(\tau, k^2), \frac{1}{\sqrt{|a_1\,d_1|}}, a_2^{'}, \left|\frac{b_2\,c_2}{a_2\,d_2}\right|\right) \\
\left(\widetilde{\omega}_2(\tau), \eta, \gamma, k^2\right) &= \left(\mathrm{sn}(\tau, k^2), \frac{1}{\sqrt{|a_1\,d_1|}}, \left|\frac{c_2}{d_2}\right|, \left|\frac{b_2\,c_2}{a_2\,d_2}\right|\right) \\
\left(\widetilde{\omega}_3(\tau), \eta, \gamma, k^2\right) &= \left(\mathrm{cn}(\tau, k^2), \frac{1}{\sqrt{|a_1\,d_1|}}, c_2^{'}, \left|\frac{b_2\,c_2}{a_2\,d_2}\right|\right)
\end{aligned} \tag{3.115}$$

It is important to remark that these results are for the case $k^2 = \left|\frac{b_2\,c_2}{a_2\,d_2}\right|$ where $|b_2\,c_2| < |a_2\,d_2|$ therefore $0 < k^2 < 1$; in fact, we can obtain in the same fashion as

we did before, the velocities for the case $k^2 = \left|\frac{a_2\, d_2}{b_2\, c_2}\right|$ where $|b_2\, c_2| \geq |a_2\, d_2|$, and also $0 < k^2 < 1$,

$$\begin{aligned}
\left(\widetilde{\omega}_1(\tau), \eta, \gamma, k^2\right) &= \left(\mathrm{cn}(\tau, k^2), \frac{1}{\sqrt{|b_2\, c_2|}}, a_2^{'}, \left|\frac{a_2\, d_2}{b_2\, c_2}\right|\right) \\
\left(\widetilde{\omega}_2(\tau), \eta, \gamma, k^2\right) &= \left(\mathrm{sn}(\tau, k^2), \frac{1}{\sqrt{|b_2\, c_2|}}, \left|\frac{a_2}{b_2}\right|, \left|\frac{a_2\, d_2}{b_2\, c_2}\right|\right) \qquad (3.116) \\
\left(\widetilde{\omega}_3(\tau), \eta, \gamma, k^2\right) &= \left(\mathrm{dn}(\tau, k^2), \frac{1}{\sqrt{|b_2\, c_2|}}, c_2^{'}, \left|\frac{a_2\, d_2}{b_2\, c_2}\right|\right)
\end{aligned}$$

As final remark, these results are consistents with the Jacobi elliptic functions described in (3.105) and (3.106) corresponding to ω_1, ω_2 and ω_3 respectively. It is possible to rename the parameters in (3.105), to obtain the equivalent parameters in 3.116, for instance we can observe $\omega_2(t)$,

$$\omega_2(t) = \pm\alpha_2\, \mathrm{sn}\left(\beta_2\ (t \mp k_2)\ |\, m_{\omega_2}\right) = \pm\sqrt{\gamma}\, \mathrm{sn}\left(\frac{1}{\eta}\ (t \mp t_0)\ \Big|\, k^2\right) \qquad (3.117)$$

therefore, we obtain $\alpha_2 = \sqrt{\gamma}$, $\beta_2 = \frac{1}{\eta}$ and $m_{\omega_2} = k^2$. If we analyze each equivalence we obtain the following

$$\begin{aligned}
\alpha_2 &= \sqrt{-\frac{a_2}{b_2}} = \sqrt{\left|\frac{a_2}{b_2}\right|} \text{since } a_2 > 0,\ b_2 < 0 \\
\sqrt{\gamma} &= \sqrt{\left|\frac{a_2}{b_2}\right|} \\
\beta_2 &= \sqrt{-b_2\, c_2} = \sqrt{|b_2\, c_2|} \text{since } c_2 > 0,\ b_2 < 0 \\
\frac{1}{\eta} &= \frac{1}{\frac{1}{\sqrt{|b_2\, c_2|}}} = \sqrt{|b_2\, c_2|}
\end{aligned}$$

$$m_{\omega_2} = \frac{a_2^{'}}{c_2^{'}}\frac{R}{P} = \frac{a_2^{'}R}{P\,c_2^{'}} = \frac{R}{P}\frac{\left(\omega_1^2 - \frac{P}{Q}\,\omega_2^2\right)}{\left(\omega_3^2 - \frac{R}{Q}\,\omega_2^2\right)}$$

$$k^2 = \left|\frac{a_2\,d_2}{b_2\,c_2}\right| = \left|\frac{\frac{a_2^{'}Q}{2}\,\frac{R}{2}}{\frac{P}{2}\,\frac{c_2^{'}Q}{2}}\right| = \frac{a_2^{'}R}{P\,c_2^{'}} = \frac{R}{P}\frac{\left(\omega_1^2 - \frac{P}{Q}\,\omega_2^2\right)}{\left(\omega_3^2 - \frac{R}{Q}\,\omega_2^2\right)}$$

According to the offset t_0 and k_2, the sign of the function, outside and inside, will depend on the initial condition, in fact, this is an interesting property of Jacobi elliptic functions [**11**].

The rest of parameters could be check in the same fashion as before, therefore, we can obtain all rotational velocities from two different analysis.

3.2.4. Numerical Results. The comparison between the numerical solution of Euler equations with free torque and our analytical solution is presented. An experiment which describe the difference between a rigid box with free torque motion and without also is shown.

The Algorithm 1 is the main procedure, which compute the rotational velocities with a numerical integration method using Algorithm 5, and the analytical solution using Algorithm 2, 3 and 4. The parameter t_0 is obtained by the inverse of sn function [**99**],

$$\operatorname{inv}\operatorname{sn}(x\,|\,m) = \int_0^x \frac{dt}{\sqrt{1-t^2}\,\sqrt{1-k^2\,t^2}} \tag{3.118}$$

The inputs in the main procedure are $I_{11} = 5$, $I_{22} = 10$ and $I_{33} = 12$, also the timestep is $\Delta t = 10^{-3}$ with $t_F = 5$ seconds. It has been computed 8 cases with different set of initial conditions:

Case	Initial Conditions		
A	$\omega_1(0)$	$\omega_2(0)$	$\omega_3(0)$
B	$\omega_1(0)$	$\omega_2(0)$	$-\omega_3(0)$
C	$\omega_1(0)$	$-\omega_2(0)$	$\omega_3(0)$
D	$-\omega_1(0)$	$\omega_2(0)$	$\omega_3(0)$

Case	Initial Conditions		
E	$\omega_1(0)$	$-\omega_2(0)$	$-\omega_3(0)$
F	$-\omega_1(0)$	$\omega_2(0)$	$-\omega_3(0)$
G	$-\omega_1(0)$	$-\omega_2(0)$	$\omega_3(0)$
H	$-\omega_1(0)$	$-\omega_2(0)$	$-\omega_3(0)$

TABLE 3.4. Set of initial conditions.

where $\omega_0 = [\omega_1(0), \omega_2(0), \omega_3(0)] = [1, 10, 2]$.

For each case, (see Figure 3.3 and 3.4) the dashed lines are the rotational velocities computed by the numerical integration method, and the continuous lines, computed with elliptic Jacobi functions, are the analytical solutions.

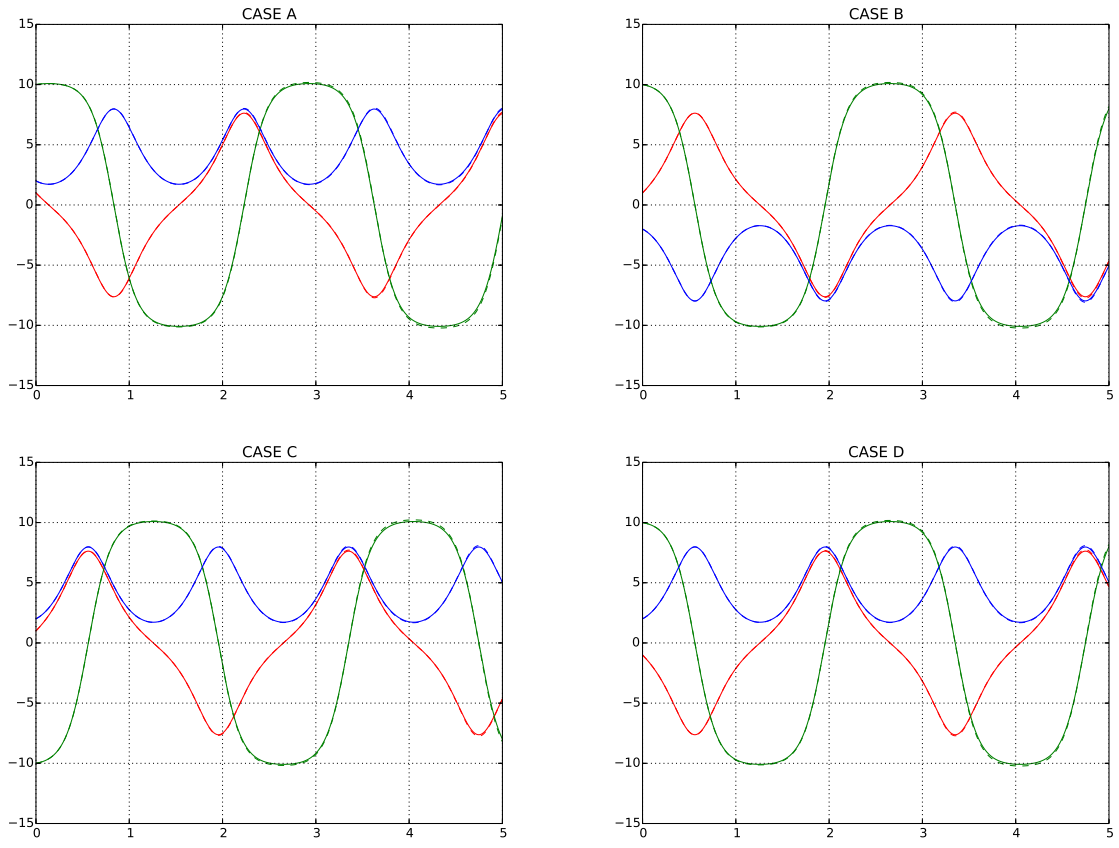


FIGURE 3.3. Cases A, B, C and D solved with numerical integration method and analytical solution

Algorithm 1 Free_Torque_Motion

Require: $I_{11}, I_{22}, I_{33}, \text{tol}, t_F, \Delta t$
Ensure: omg_1, omg_2, omg_3, omg_j1, omg_j2, omg_j3
  $P, Q, R \leftarrow$ JACOBI_VALUES(I_{11}, I_{22}, I_{33})
  a2p, c2p, a1, d1, a2, b2, c2, d2 $\leftarrow$ JACOBI_PARAMETERS$(P, Q, R, \omega(0))$
  omg_1, omg_2, omg_3 $\leftarrow$ EULER_NUM$(\omega(0), P, Q, R, t_F, \Delta t)$
  η, k2 $\leftarrow 1/|\text{a1}\,\text{d1}|, |(\text{b2}\,\text{c2})/(\text{a2}\,\text{d2})|$
  $\gamma_1, \gamma_2, \gamma_3 \leftarrow \text{a2p}, |\text{c2}/\text{d2}|\,, \text{c2p}$
  $t_0 \leftarrow \eta\,\text{inv sn}(\omega_2(0)/\sqrt{\gamma_2}\,|\,\text{k2})$
  omgp2 $\leftarrow (Q/2)\ \omega_1(0) * \omega_3(0)$
  omgp2f $\leftarrow (1/\eta)\,\sqrt{\gamma_2}\,\text{cn}(t_0/\eta\,|\,\text{k2})\,\text{dn}(t_0/\eta\,|\,\text{k2})$
  if |omgp2 − omgp2f| < tol **then**
    sign2 $\leftarrow 1$
  else
    $t_0 \leftarrow \eta\,\text{inv sn}(-\omega_2(0)/\sqrt{\gamma_2}\,|\,\text{k2})$
    sign2 $\leftarrow -1$
  end if
  omgp1 $\leftarrow (P/2)\ \omega_2(0) * \omega_3(0)$
  omgp3f $\leftarrow (-\text{k2}/\eta)\,\sqrt{\gamma_1}\,\text{sn}(t_0/\eta\,|\,\text{k2})\,\text{cn}(t_0/\eta\,|\,\text{k2})$
  if |omgp1 − omgp1f| < tol **then**
    sign1 $\leftarrow 1$
  else
    sign1 $\leftarrow -1$
  end if
  omgp3 $\leftarrow (R/2)\ \omega_1(0) * \omega_2(0)$
  omgp3f $\leftarrow (-1/\eta)\,\sqrt{\gamma_3}\,\text{sn}(t_0/\eta\,|\,\text{k2})\,\text{dn}(t_0/\eta\,|\,\text{k2})$
  if |omgp3 − omgp3f| < tol **then**
    sign3 $\leftarrow 1$
  else
    sign3 $\leftarrow -1$
  end if
  for $t \leftarrow t_I + \Delta t, t_F$ **do**
    omg_j1 $\leftarrow$ OMEGA_DN$(t, \gamma_1, \eta, \text{k2}, \text{sign1}, t_0)$
    omg_j2 $\leftarrow$ OMEGA_SN$(t, \gamma_2, \eta, \text{k2}, \text{sign2}, t_0)$
    omg_j3 $\leftarrow$ OMEGA_CN$(t, \gamma_3, \eta, \text{k2}, \text{sign3}, t_0)$
  end for

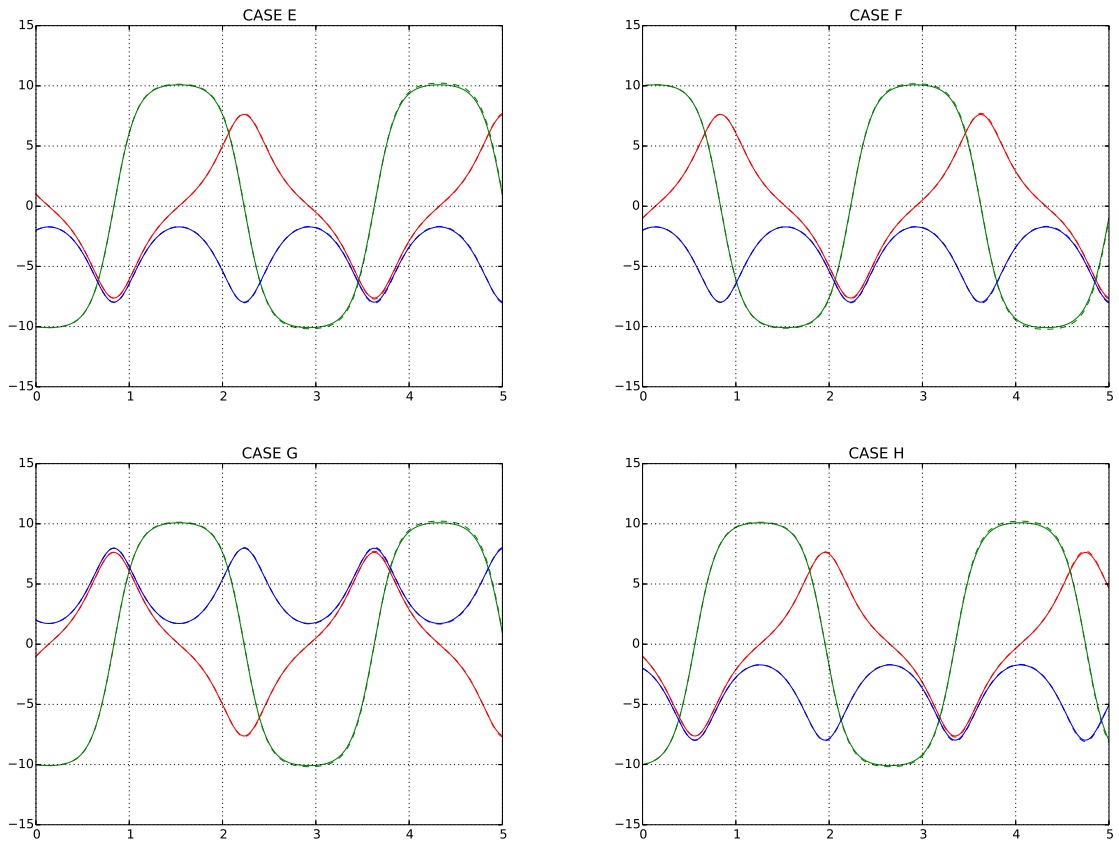


FIGURE 3.4. Cases E, F, G and H solved with numerical integration method and analytical solution

Algorithm 2 Jacobi_Values

function JACOBI_VALUES(I_{11}, I_{22}, I_{33})
$P \leftarrow 2\,(I_{22} - I_{33})\,/I_{11}$
$Q \leftarrow 2\,(I_{33} - I_{11})\,/I_{22}$
$R \leftarrow 2\,(I_{11} - I_{22})\,/I_{33}$
return P, Q, R
end function

Algorithm 3 Jacobi_Parameters

function JACOBI_PARAMETERS($P, Q, R, \omega(0)$)
  a2p $\leftarrow \omega_1(0)^2 - (P/Q)\,\omega_2(0)^2$
  c2p $\leftarrow \omega_3(0)^2 - (R/Q)\,\omega_2(0)^2$
  a1 $\leftarrow -\left(Q\,\omega_1(0)^2 + |P|\,\omega_2(0)^2\right)/2$
  d1 $\leftarrow (R/2)$
  a2 $\leftarrow -a1$
  b2 $\leftarrow (P/2)$
  c2 $\leftarrow \left(|R|\,\omega_2(0)^2 + Q\,\omega_2(0)^2\right)/2$
  d2 $\leftarrow d1$
  return a2p, c2p, a1, d1, a2, b2, c2, d2
end function

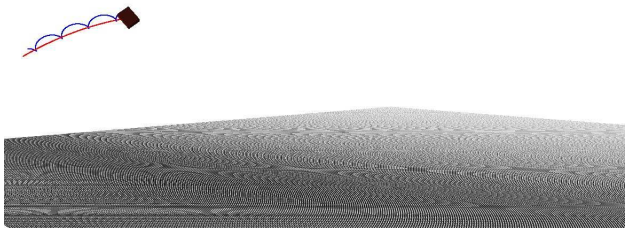
Algorithm 4 Elliptic Jacobi Functions

function OMEGA_DN(t, γ, η, k2, sign, t_0)
  return sign $\sqrt{\gamma}\,\mathrm{dn}\left((t+t_0)/\eta \,|\, \mathrm{k2}\right)$
end function
function OMEGA_SN(t, γ, η, k2, sign, t_0)
  return sign $\sqrt{\gamma}\,\mathrm{sn}\left((t+t_0)/\eta \,|\, \mathrm{k2}\right)$
end function
function OMEGA_CN(t, γ, η, k2, sign, t_0)
  return sign $\sqrt{\gamma}\,\mathrm{cn}\left((t+t_0)/\eta \,|\, \mathrm{k2}\right)$
end function

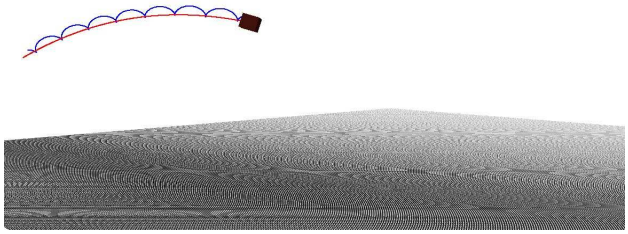
Algorithm 5 Euler Numerical Integration

function EULER_NUM($\omega(0), P, Q, R, t_F, \Delta t$)
  omg_euler1(0) $\leftarrow \omega_1(0)$
  omg_euler2(0) $\leftarrow \omega_2(0)$
  omg_euler3(0) $\leftarrow \omega_3(0)$
  for $i \leftarrow t_I + \Delta t, t_F$ **do**
    omg_euler1(i) $\leftarrow \omega_1(i-\Delta t) + \Delta t\,(P/2)\,\omega_2(i)\,\omega_3(i)$
    omg_euler2(i) $\leftarrow \omega_2(i-\Delta t) + \Delta t\,(Q/2)\,\omega_1(i)\,\omega_2(i)$
    omg_euler3(i) $\leftarrow \omega_3(i-\Delta t) + \Delta t\,(R/2)\,\omega_1(i)\,\omega_2(i)$
  end for
  return omg_euler1, omg_euler2, omg_euler3
end function

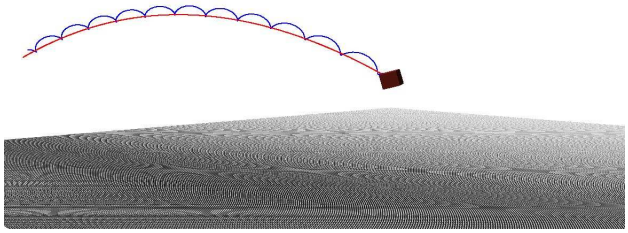
Another interesting result has been obtained: the difference of a solid trajectory considering constant (Figure 3.5) and non constant (Figure 3.6) rotational velocities. The gravity center trajectory (red line) is the same in both cases, but a vertex trajectory (blue line) is loudly different.



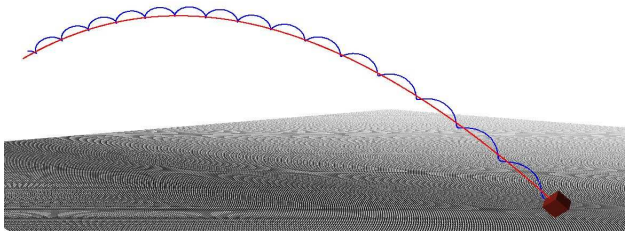
(a) 0.375[s]



(b) 0.75[s]

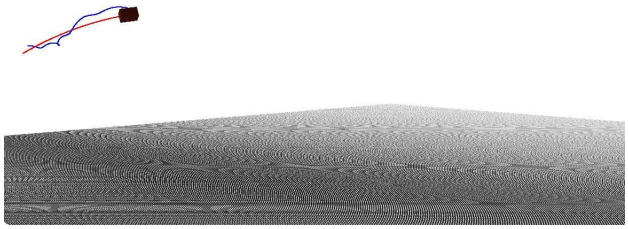


(c) 1.125[s]

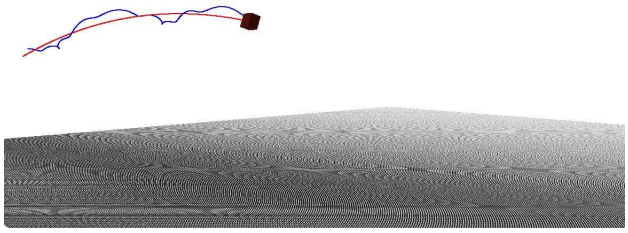


(d) 1.5[s]

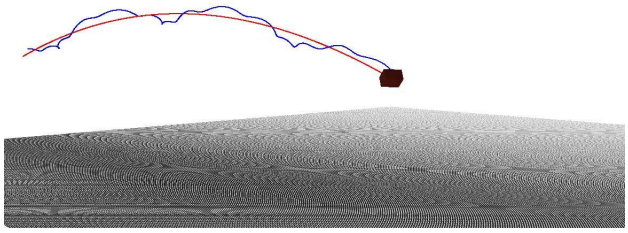
FIGURE 3.5. Trajectory for a rigid solid with constant rotational velocity.



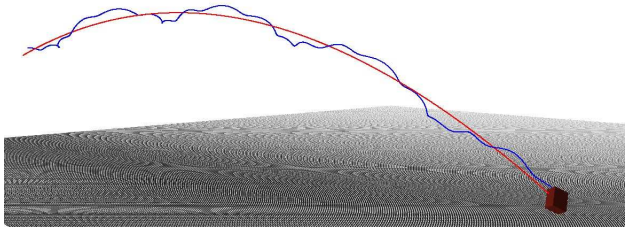
(a) 0.375[s]



(b) 0.75[s]



(c) 1.125[s]



(d) 1.5[s]

FIGURE 3.6. Trajectory for a rigid solid with non constant rotational velocity.

A comparison with another research made in **[182]** has been considered as weel. The inertia values are $I_{11} = 10, I_2 = 20$ and $I_3 = 26$. The initial conditions are $\omega_0 = [1, 15, 1]$. In the Figure 3.7 is shown the example of the exact solution obtained by the authors in **[182]**. In the Figure 3.8, we can observe the same results obtained by our proposal.

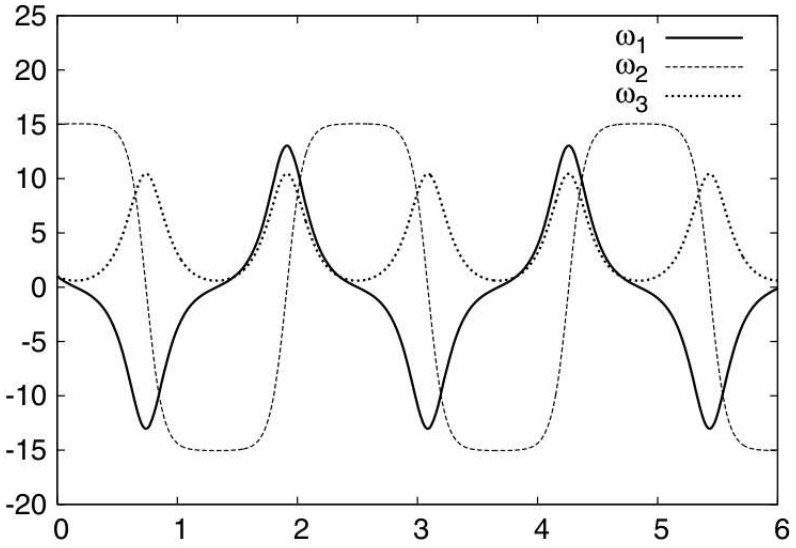


FIGURE 3.7. Solution for $\omega(t)$ in **[182]**

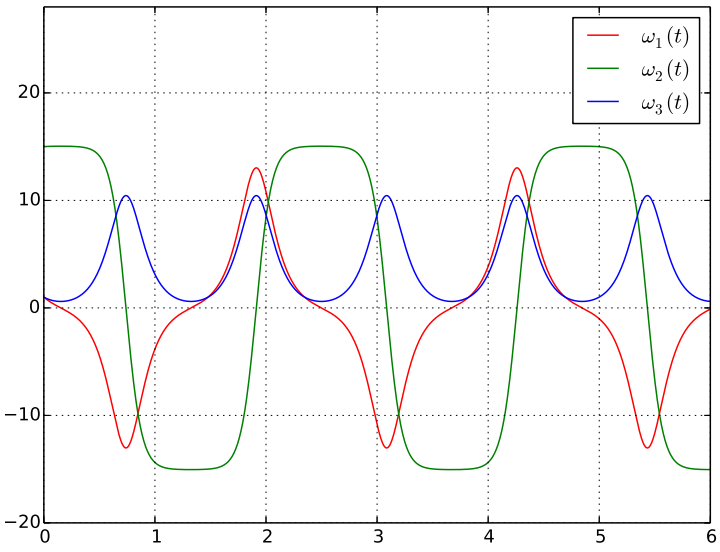


FIGURE 3.8. Solution for $\omega(t)$ with our proposal.

According to Euler equations, when two inertias are equal, a rotational velocity must be constant. For instance, if $I_{22} = I_{33}$, the first equation in (3.64) is now $\dot{\omega}_1 = 0$, therefore ω_1 is constant. Taking into account the solutions, with the condition

$I_{22} = I_{33}$, and considering from (3.65) that $P = 0$, we have the following:

$$k^2 = \left| \frac{b_2\, c_2}{a_2\, d_2} \right| \overset{\text{Table 3.3}}{=} \left| \frac{\frac{P}{2}\, c_2}{a_2\, d_2} \right| = 0 \tag{3.119}$$

When $k^2 = 0$, the Jacobi Elliptic functions take the following forms [**11**]:

$$\begin{aligned} \mathrm{sn}(x \,|\, 0) &= \sin(x) \\ \mathrm{cn}(x \,|\, 0) &= \cos(x) \\ \mathrm{dn}(x \,|\, 0) &= 1 \end{aligned} \tag{3.120}$$

therefore, the solutions in (3.115) now are

$$\begin{aligned} \omega_1(t) &= \sqrt{a_2'} = \sqrt{\omega_1(0)^2 - \frac{P}{Q}\omega_2(0)^2} = \omega_1(0) \\ \omega_2(t) &= \pm\sqrt{\left| \frac{c_2}{d_2} \right|}\, \sin\left(\frac{t + t_0}{\eta} \right) \\ \omega_3(t) &= \pm\sqrt{c_2'}\, \cos\left(\frac{t + t_0}{\eta} \right) \end{aligned} \tag{3.121}$$

where the function sign depends on the initial condition.

In the top chart of the Figure 3.9 is shown the numerical solution for Euler equations with $I_{22} = I_{33}$. In this case, clearly we can note the constant velocity $\omega_1(t)$ (red line) and the corresponding $\omega_2(t)$ and $\omega_3(t)$ as the trigonometric functions sin and cos respectively. In the bottom chart we can observe the analytical solutions presented in (3.121). We can note that all curves fit with the numerical solutions.

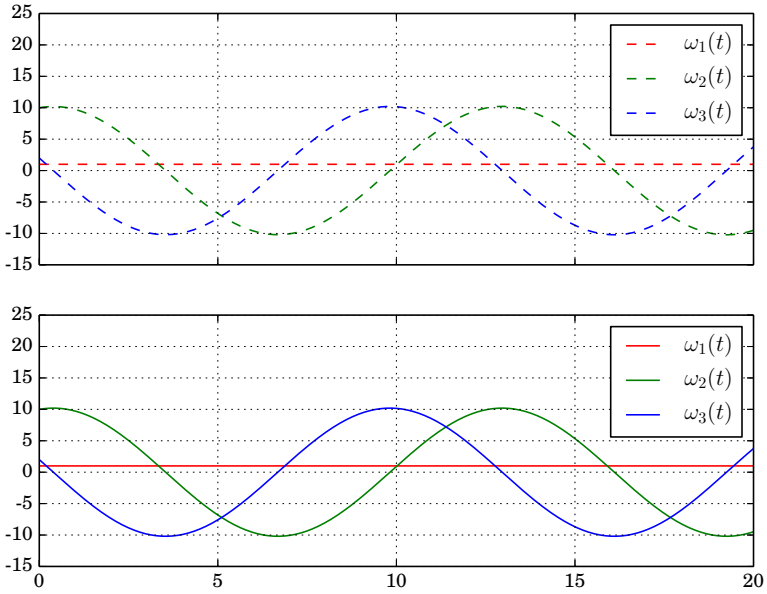


FIGURE 3.9. Solution for $I_{22} = I_{33}$, where $\omega_1(t)$ is constant.

In the same fashion we can obtain the velocity equations if $I_{11} = I_{22}$ (see Figure 3.10).

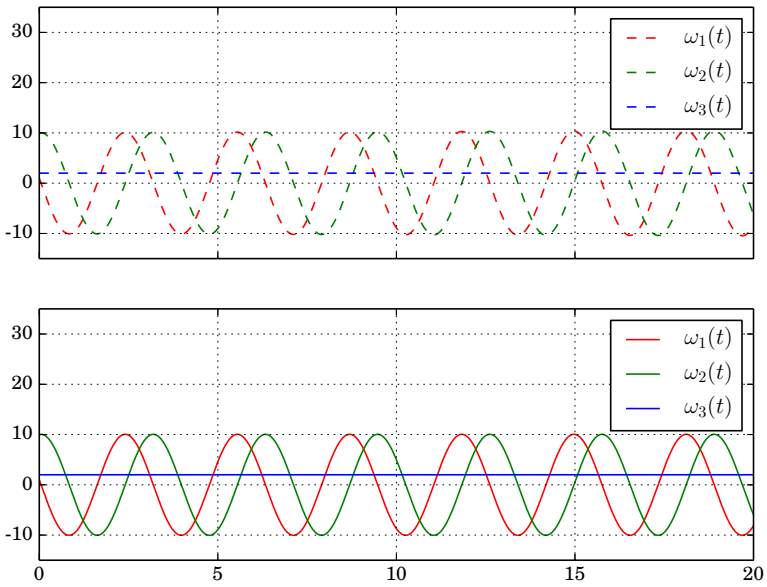


FIGURE 3.10. Solution for $I_{11} = I_{22}$, where $\omega_3(t)$ is constant.

It is important to remark, that the parameter k^2 also can take the value 1. In fact, the Jacobi Elliptic functions have the following identities in this case **[11]**:

$$\begin{aligned} \operatorname{sn}(x \,|\, 1) &= \tanh(x) \\ \operatorname{cn}(x \,|\, 1) &= \operatorname{sech}(x) \\ \operatorname{dn}(x \,|\, 1) &= \operatorname{sech}(x) \end{aligned} \tag{3.122}$$

which are not periodic functions (see Figure 3.11).

If we analyze the case that $Q = 0$, which means that $I_{11} = I_{33}$, we have the following

$$\begin{aligned} k^2 = \left| \frac{b_2\, c_2}{a_2\, d_2} \right| &\overset{\text{Table 3.3}}{=} \left| \frac{\dfrac{P}{2}\ \dfrac{|R|\,\omega_2(0)^2 + Q\,\omega_3(0)^2}{2}}{\dfrac{Q\,\omega_1(0)^2 + |P|\,\omega_2(0)^2}{2}\ \dfrac{R}{2}} \right| \\ &= \left| \frac{P\ (|R|\,\omega_2(0)^2)}{R\ (|P|\,\omega_2(0)^2)} \right| = 1 \end{aligned} \tag{3.123}$$

The problem is, when $I_{11} = I_{33}$ and assuming that $I_{11} < I_{22} < I_{33}$, which is our case, all moments of inertia must be the same, therefore all velocities are constant. According to this, there is not alternative where $k^2 = 1$, in fact, it is not physically possible, because the functions are not periodic.

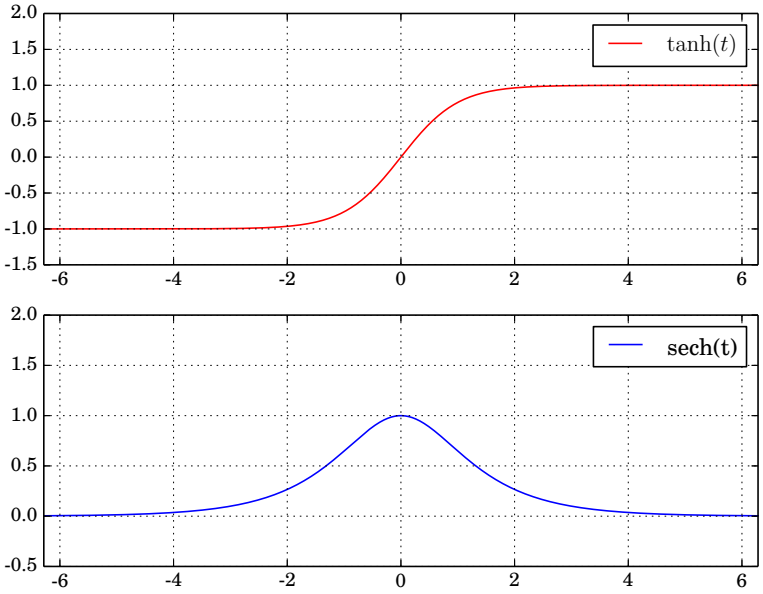


FIGURE 3.11. Graph for $\tanh(t)$ and $\operatorname{sech}(t)$.

3.3. Contact detection

An important step in this method, is the contact detection part, which determines the points where the solids are in contact. In [**53**] is mentioned a *Bounding Volume* (BV) which is a volume encapsulating one or more objects of more complex nature. The idea is, for the simple shapes, such as boxes or spheres, to have cheaper overlap tests than the complex objects they bound. Using bounding volumes fast overlap rejection test is allowed, because it is only necessary a test against the complex bounded geometry when the initial overlap query for the bounding volumes gives a positive result. In [**53**] also mentioned that not all geometric objects serve as effective bounding volumes; desirable properties for bounding volumes include:

- Inexpensive intersection tests
- Tight fitting
- Inexpensive to compute

- Easy to rotate and transform
- Use little memory

The main idea is to apply a technique called "early out", which consists in to procede with less expensive tests first before to entry in the more specific overlap tests. Therefore, the idea of the bounding volumes is that they must have a simple geometry shape, allowing a first fast overlapping test. The principle types of bounding volumes used in real time collision detection are (see Figure 3.12):

- SPHERE
- AABB: Axis-aligned Bounding Boxes
- OBB: Oriented Bounding Boxes
- 8-DOP: Eight - direction discrete orientation polytope
- CONVEX HULL

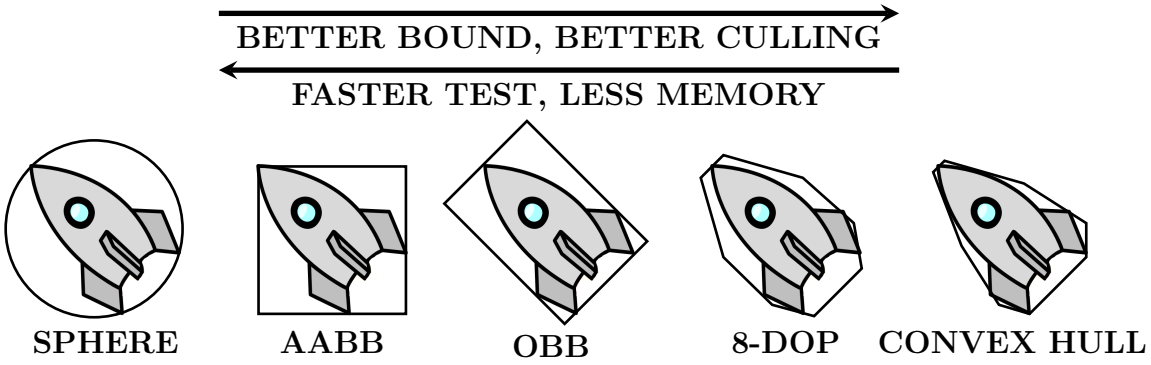


FIGURE 3.12. Types of bounding volumes, see [**53**].

The contact detection procedure has been separated in two steps: a faster overlap detection and a contact point detection. The first step is made using spheres as BV, therefore the overlap detection is focused in detect if a polyhedron is close enough to another one. The second part is performed when both bounding spheres are overlapped, then, is applied a technique to search the points where the polyhedrons

are in contact. Three contact point detections are made: surface, face and edge detection.

Before beginning to review each type of detection, let discuss about the polyhedron representation, which are an important element in the simulation. Each polyhedron is represented by their vertices, edges and faces. To build a polyhedron is necessary to know their size, orientation and position in the space. With this information a polyhedron is easily represented as unique in the space, in fact, their vertices, edges and faces are always defined in their local coordinate system. Therefore, just the position and orientation of each polyhedron is updated from one timestep to another. For instance, let consider the polyhedron of the Figure 3.13, with size a, b and c and orientation $\hat{i}, \hat{j}$ and $\hat{k}$.

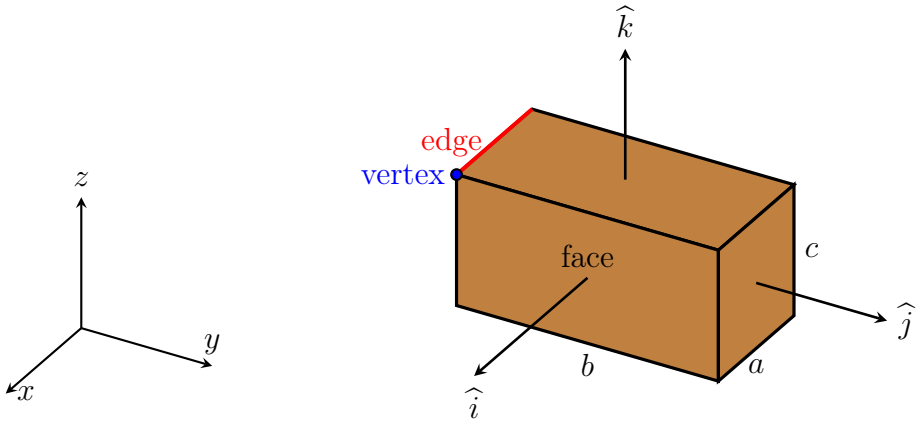


FIGURE 3.13. A box

A box, then, has

- 8 vertices, which are represented in function of the size parameteres a, b and c,
- 12 edges, which are represented in function of two vertices, where the edge begins and ends. For instance, the edge 0 begins at vertex 0 and ends at vertex 1, and
- 6 faces, which are represented with plane equations in function of the polyhedron edges, for instance, the face 0 is represented by the plane equation $[1\ 0\ 0] \cdot [x\ y\ z]^T - \frac{a}{2} = 0$.

3.3.1. Surface contact detection. The surface contact detection is focused, and active, where in the simulations there are one or more rigid surfaces which interacting with polyhedrons. Mainly the ground surface always is in a simulation, which is represented with the plane $x - y$.

The structure of the surface can be determined by two forms: a plane and triangles. The first type of structure is easier to treat, due to is just a plane equation which represent the surface. A point of a polyhedron is in contact with the surface if it is under the plane. For instance, the ground in a simulation is easily represented by the plane $\Pi : z = 0$, then if any vertex has z value negative, then it is in contact with the ground.

When a surface is more complex, then it is represented by a set of points which are concatenated by a triangularization process. A set of triangles represents then a complex surface; in this case, a polyhedron is in contact, if some of their vertices is under some triangle of the surface. When a surface is not the ground but less

complex, for instance, a slide, then it is not necessary the triangularization. A slide can be represented by a plane with a certain inclination angle, even so, the method to detect interpenetration is the same, it must be detected if some vertex of the solid is under the plane.

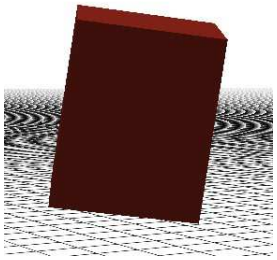


FIGURE 3.14. Ground contact

3.3.2. Face contact detection. Each face k of a solid is represented by a plane equation $\Pi_k(x, y, z) : c_x\, x + c_y\, y + c_z\, z - c_k$. A point $\boldsymbol{p} = (p_x, p_y, p_z)$ in the local reference system of the solid, will be inside the solid if $\Pi_k(p_x, p_y, p_z) < 0\, \forall k = 0, ..., 5$. The point is inside the solid if is under all faces, which means a negative value in the plane evaluation. The important step in this method is about the evaluation which it is necessary to make, because a fast output could be obtain if the point is outside the solid. The face contact detection is not finished yet, in fact, the computation described before is just the way to know if a point is inside or not of the solid. The information which the algorithm really need is about the face which is in contact with the point, for this step it computes a double check, which it is described below

- Check 1: computing closer face to the point.
- Check 2: the velocity direction of the solid is inward the closer face found in the check 1.

Therefore, the contact vector direction is the normal vector of the face, and the contact point is the vertex which is inside the solid.

3.3.3. Edge contact detection. Edge contact detection is the most computationally expensive check. The first version considered the detection as the intersection of two planes formed by the points of the edge and the center of gravity of a solid. Considering the situation in Figure 3.15, we have two solids (red and blue) which are in contact with their edges respectively. The solid 1 contains the edge $\boldsymbol{v}_1\,\boldsymbol{v}_2$, forming a triangle with its center of gravity $\boldsymbol{g}_1$. Analogously, solid 2 contains the edge $\boldsymbol{u}_1\,\boldsymbol{u}_2$, forming a triangle with its center of gravity $\boldsymbol{g}_2$. The edge detection is made in two steps: (1) check if both planes formed by the triangles intersect between them, this check can generate a line l, and (2) check if the line l is lying in some solid. This detection is made between each pair of edges. If we take into account just two solids, each one with 12 edges, it must compute 144 edge detection. We can imagine the amount of edge detections that must be performed for a neighborhood of solids, is much computing!

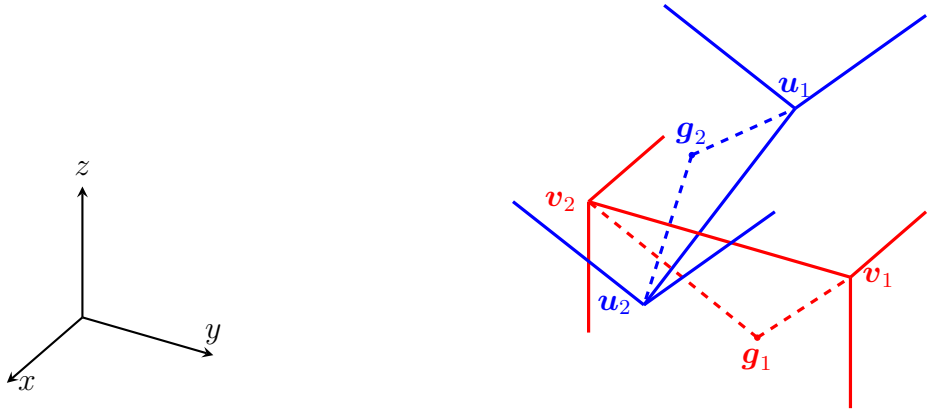


FIGURE 3.15. Contact Edge Detection: first version

In the second version, we introduce another technique. The idea is taking into account the edge projection over a solid, and check if this projection lies partially or completly.

3.3.3.1. *Starting Situation.* The new algorithm considers rectangular boxes V, with sizes $0 < a < b < c$, which are moving and colliding in the 3-dimensional space $\mathbb{R}^3$, which is equipped with a fixed origin $\mathcal{O}$ an the fixed canonical orthonormal standard vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$.

These boxes are considered as including the volume that they enclose. That these boxes are topologically closed means that they also include their boundary. Each box V has eight vertices V_j, $j = 0 : 7$, with position vector $\mathbf{V}_j$ relative to the fixed canonical orthonormal vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. Note that these position vectors $\mathbf{V}_j$ are functions of time t. In addition, box V has twelve edges V_{jk}, joining vertices V_j and V_k, and six plane sides V_{ijkl} bounded by the closed path $V_{ij}V_{jk}V_{kl}V_{li}$ formed by the edges V_{ij}, V_{jk}, V_{kl} and V_{li}.

3.3.3.2. *The local reference system.* For each time step Δt we must sequentially select one box after the other and determine all the other boxes which have nonempty intersection with the selected one.

We call U the selected box. Recall that for each time step U must run over the whole set of boxes V. The vertices of box U are denoted by U_j, $j = 0 : 7$, having position vector $\mathbf{U}_j$ relative to the fixed canonical orthonormal vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. Similarly, U_{jk} denotes the edge joining the vertices U_j and U_k, and U_{ijkl} the plane side with boundary $U_{ij}U_{jk}U_{kl}U_{li}$.

Given a maximal linear speed $v_{\max}$ of the boxes and timestep Δt, it is clear that only those boxes V lying inside a ball $\mathcal{B}(U)$ centered at (the center of gravity of) U and radius, say $2v_{\max}\,\Delta t$, have a chance of colliding with U. Thus, in order to find the boxes which collide with U we do not have to search over the whole set of boxes V but only those lying in $\mathcal{B}(U)$.

First of all, we arbitrarily fix a corner of box U and label it U_0. Then we label U_1, U_2, U_3 the three adjacent vertices to U_0 in such a way that the vectors $\mathbf{U}_1 - \mathbf{U}_0$, $\mathbf{U}_2 - \mathbf{U}_0$ and $\mathbf{U}_3 - \mathbf{U}_0$ make a *right* or *positive trihedron* at U_0. Using this trihedron we define the local orthonormal basis for $\mathbb{R}^3$:

$$\mathbf{b}_1 = \frac{\mathbf{U}_1 - \mathbf{U}_0}{\|\mathbf{U}_1 - \mathbf{U}_0\|}\,, \quad \mathbf{b}_2 = \frac{\mathbf{U}_2 - \mathbf{U}_0}{\|\mathbf{U}_2 - \mathbf{U}_0\|}\,, \quad \mathbf{b}_3 = \frac{\mathbf{U}_3 - \mathbf{U}_0}{\|\mathbf{U}_3 - \mathbf{U}_0\|}\,. \tag{3.124}$$

We use these moving basis vectors to constitute a moving rectangular local coordinate system. Written componentwise in terms of the fixed global canonical basis $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ the moving basis vectors are:

$$\mathbf{b}_1 = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix}\,, \quad \mathbf{b}_2 = \begin{bmatrix} b_{12} \\ b_{22} \\ b_{32} \end{bmatrix}\,, \quad \mathbf{b}_3 = \begin{bmatrix} b_{13} \\ b_{23} \\ b_{33} \end{bmatrix}\,. \tag{3.125}$$

We define the position vectors $\mathbf{V}_j$ of the vertices V_j of the boxes $V \in \mathcal{B}(U)$, relative to the local position vector $\mathbf{U}_0$:

$$\mathbf{V}_j^* := \mathbf{V}_j - \mathbf{U}_0\,, \quad j = 0:7\,. \tag{3.126}$$

Note that these vectors are still represented in terms of the fixed canonical basis vectors $\mathbf{e}_k$. We want to write them in terms of the local basis vectors $\mathbf{b}_k$.

For this, consider any vector $\mathbf{v} = [v_1, v_2, v_3]^\top$ from $\mathbb{R}^3$ written in terms of the canonical basis $\mathbf{e}_k$. In order to write $\mathbf{v}$ in terms of the local basis $\mathbf{b}_k$ we need to find

coefficients $\alpha_1, \alpha_2, \alpha_3$ such that $\alpha_1 \mathbf{b}_1 + \alpha_2 \mathbf{b}_2 + \alpha_3 \mathbf{b}_3 = \mathbf{v}$. We of course need to use the change of basis matrix $\mathbf{B}$:

$$\overbrace{\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}}^{\mathbf{B}} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_2 \end{bmatrix} . \tag{3.127}$$

Thus, the coefficients $\alpha_1, \alpha_2, \alpha_3$ are given by:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} v_1 \\ v_2 \\ v_2 \end{bmatrix} . \tag{3.128}$$

In particular, the coordinates of $\mathbf{V}_j^* = \left[V_{1j}^*, V_{2j}^*, V_{3j}^*\right]^\top$ relative to the $\mathbf{b}_k$ basis are given by:

$$\begin{bmatrix} \alpha_1^* \\ \alpha_2^* \\ \alpha_3^* \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} V_{1j}^* \\ V_{2j}^* \\ V_{3j}^* \end{bmatrix} \tag{3.129}$$

Then we clearly have:

$$V_j \in U \quad \Leftrightarrow \quad 0 \leq \alpha_{1j} \leq a \ \wedge \ 0 \leq \alpha_{2j} \leq b \ \wedge \ 0 \leq \alpha_{3j} \leq c \, . \tag{3.130}$$

This allows us to determine whether vertex V_j lies in box U or not. We must check this for all eight vertices V_j of V. If *at least one* of the vertices V_j of V lies in U, then the two boxes have collided and we must go over to the schock routine. Note that now, in principle, one would have to switch the roles of U and V and check whether some of the vertices of U lie in V, but this is not necessary since we must sequentially select *every* V as the box U, and hence we eventually will do this as well. If *none* of the vertices of V lies in U, it is still possible that at least one or more *edges* of V has nonempy intersection with U. We must check this.

3.3.3.3. *Checking whether some edges of V intersect box U.* We get into this section of the algorithm only when no vertex of box V has been found in box U.

Consider then edge V_{jk} with end vertices V_j and V_k of box V. Relative to the coordinate system located at U_0, but still with respect to the canonical $\mathbf{e}_k$ basis vectors, this edge is described by the position vector:

$$\mathbf{r}^*(s) = (1-s)\,\mathbf{V}_j^* + s\,\mathbf{V}_k^*\,, \quad 0 \le s \le 1\,, \tag{3.131}$$

where –at the risk of being redundant– vectors $\mathbf{r}^*(s)$, $\mathbf{V}_j^*$ and $\mathbf{V}_k^*$ are all written in terms of the fixed canonical basis vectors $\mathbf{e}_1 = [1,0,0]^\top$, $\mathbf{e}_2 = [0,1,0]^\top$,$\mathbf{e}_3 = [0,0,1]^\top$. Note that $\mathbf{r}^*(s)$ is a function of time t as well. Now we apply the change of basis matrix $\mathbf{B}$ —or, more precisely, $\mathbf{B}^{-1}$— to (3.131) and we obtain the representation of the edge V_{jk} in terms of the local ortonormal basis $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$:

$$\boldsymbol{\rho}(s) := \mathbf{B}^{-1}.\,\mathbf{r}^*(s) = (1-s)\,\mathbf{B}^{-1}.\mathbf{V}_j^* + s\,\mathbf{B}^{-1}.\mathbf{V}_k^*\,, \quad 0 \le s \le 1\,. \tag{3.132}$$

Now the solution of the problem of finding whether the edge V_{jk} lies completly or partially inside the box U is easy. Let $\boldsymbol{\rho} = [\rho_1, \rho_2, \rho_3]^\top$ and define:

$$\begin{aligned} J := &\{s \in [0,1] \mid 0 \le \rho_1(s) \le a\} \\ &\cap \{s \in [0,1] \mid 0 \le \rho_2(s) \le b\} \cap \{s \in [0,1] \mid 0 \le \rho_3(s) \le c\} \end{aligned} \tag{3.133}$$

J is clearly a closed subinterval of $[0,1]$. This interval allows us to compute *edge interpenetration*. If for at least one *edge* of box V we have $J \neq \emptyset$, then this V-edge has penetrated the U-box and we must go over to the schock routine. If for all *edges* of box V we have $J = \emptyset$, then we are sure that boxes U and V are disjoint and we have finished testing the current V-box and we can go over to test the next box $V \in \mathcal{B}(U)$. When we have finished testing all $V \in \mathcal{B}(U)$ we can go over and select a new distinguished box U and repeat the whole procedure until

we exhaust the supply of boxes and hence complete the computations for one time step. Besides edge interprenetration, many other useful geometric conclusions can be drawn from (3.133), in particular how deep box V intepenetrates box U. For instance, let $J_0 := \inf J$ and $J_1 = \sup J$, so that $0 \leq J_0 \leq J_1 \leq 1$ and $J = [J_0\,, J_1]$. Then we clearly have:

$$\boldsymbol{\rho}(s) \in U \;\Leftrightarrow\; s \in J = [J_0\,, J_1]\;, \tag{3.134}$$

and moreover:

$$V_j \in U \;\Leftrightarrow\; J_0 = 0 \qquad \text{and} \qquad V_k \in U \;\Leftrightarrow\; J_1 = 1\,. \tag{3.135}$$

Moreover, having represented vectors with respect to the $\mathbf{b}_k$-basis localized at vertex U_0, it is easy to obtain projections onto the sides of box U without any computation: for instance, if we want to project some vector $\mathbf{v}$ –already represented in such a way– onto a box side which is perpendicular to, say, $\mathbf{b}_1$, all we have to do is to delete the $\mathbf{b}_1$-component from $\mathbf{v}$.

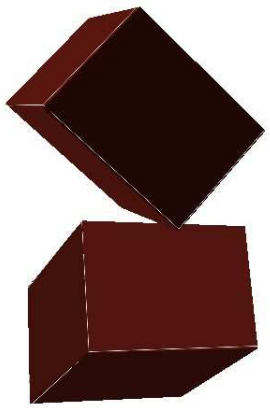


FIGURE 3.16. Edge contact

CHAPTER 4

Computational Model

The computational model performed in the numerical simulations has two versions: (1) the computational improvements made in the A-CD^2 approach, mainly the acceleration of the Uzawa method by the Conjugate Gradient Method, and, (2) the performance improvements in the entire algorithm.

4.1. Main Procedure

The main procedure implemented for numerical simulations is described in the Figure 4.1. In the module CONTACT DETECTION three kind of contact detection: surface, face and edge are considered. Then, if some contact point is detected, it must be performed the module OPTIMIZATION, where the constrained minimization problem is build as a quadratic function in matrix form. The optimization solver (CPLEX) obtains the solution. Finally, the updating velocity module is performed, where the new velocities of each solid is computed. When the simulation ends ($t = T$), an evolution file is generated (`evolution.data`). This file contains all the positions of the solids from $t = 0$ to $t = T$. A Python script receives this file for the 3D visualization.

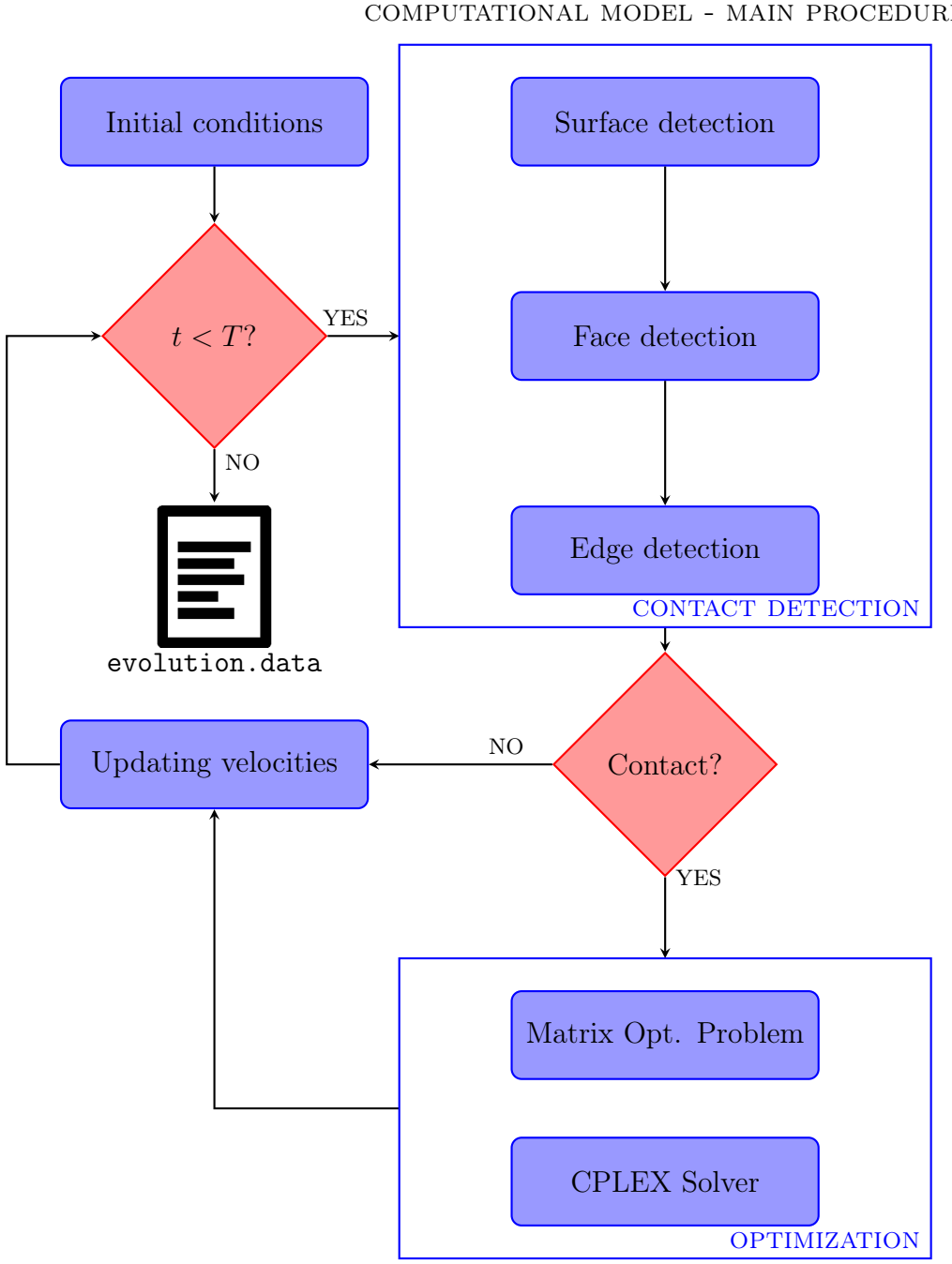


FIGURE 4.1. Computational model for a numerical simulation.

4.2. First version

The A-CD^2 method has been performed with the main procedure described before. In the first version, the collision detection module was implemented by inside-point technique for surface and face detection, and by plane-intersection technique for edge detection. The constrained minimization problem was solved by the Uzawa method [**36**, p. 360]. This saddle point problem, $\min F(\overrightarrow{x})$ subject to a set of constraints $\phi_l(\overrightarrow{x}) \leq 0$, is solved by means of an iterative method, which approximates the solution of the dual problem (Algorithm 7). Also, it has been studied the constraint matrix. When a solid has more than three contact points, the constraints are not linearly independent, then it is possible to keep only the constraints which are linear independent. For this, it was implemented a reduced row echelon form procedure (Algorithm 6), which reduce the dimensionality of the constraint matrix and keep only the linear independent rows, i.e. the rank of the constraint matrix [**26**, p. 44], and reduce the computational time as well, because it is not necessary to evaluate all the constraints.

Algorithm 6 Reduced Row Echelon Form

Require: $M \in \mathbb{R}^{n \times n}$ is a matrix.
  $lead \leftarrow 0$
  $row_{count} \leftarrow$ number of rows in M
  $col_{count} \leftarrow$ number of columns in M
  for $r \leftarrow 0, row_{count} - 1$ **do**
    if $col_{count} \leq lead$ **then**
      STOP
    end if
    $i \leftarrow r$
    while $M[i, lead] \leftarrow 0$ **do**
      $i \leftarrow i + 1$
      if $row_{count} == i$ **then**
        $i \leftarrow r$
        $lead \leftarrow lead + 1$
        if $col_{count} == lead$ **then**
          STOP
        end if
      end if
    end while
    Swap row i and row r
    if $M[r, lead] \neq 0$ **then**
      $r \leftarrow \frac{r}{M[r, lead]}$
    end if
    for $i \leftarrow 0, row_{count} - 1$ **do**
      if $i \neq r$ **then**
        Row $i \leftarrow$ Row $i - M[r, lead] \times$ Row r
      end if
    end for
    $lead \leftarrow lead + 1$
  end for

An important step in the Uzawa method is to obtain the minimum of the Lagrangian function (line 4 of Algorithm 7). This step is computed by the well know conjugate gradient method (Algorithm 8). The application of this method is possible because the constrained minimization problem is convex, then the Lagrangian function is convex as well. Therefore, to obtain a global minimum of this function, it is considered the derivative, equal to zero, which is a system of linear equations,

given that the Lagrangian function is quadratic. Then the solution of this system of linear equations, will be the global minimum.

Algorithm 7 Uzawa method

Require: $\boldsymbol{\mu}_0 \in \mathbb{R}_p^+$:Lagrange multipliers initial vector, $\boldsymbol{x}_0$: starting point.

1: $\gamma_0 \leftarrow 1$
2: $n \leftarrow 0$
3: **while** $|\gamma_n| \leq \varepsilon$ **do**
4: $\quad \boldsymbol{x}_{n+1} \leftarrow \operatorname{argmin} \left\{ F(\boldsymbol{x}) + \sum_{l=1}^{p} \mu_l^n \phi_l(\boldsymbol{x}), \quad x \in \mathbb{R}^{6N} \right\}$
5: $\quad \mu_{n+1}^l \leftarrow \max \left\{ 0, \mu_n^l + \varrho \, \phi^l \left(\boldsymbol{x}_{n+1} \right) \right\}, \quad \forall l = 1, p$
6: $\quad \gamma_{n+1} \leftarrow \sum_{l=1}^{p} \mu_{n+1}^l \phi^l \left(\boldsymbol{x}_{n+1} \right)$
7: $\quad n \leftarrow n + 1$
8: **end while**

Algorithm 8 Conjugate gradient method

Require: $\overrightarrow{x_0} \in \mathbb{R}^{6N}$ is the initial vector.

1: $\overrightarrow{r_0} \leftarrow 2M\overrightarrow{U^-} + \overrightarrow{T^{ext}} + \sum_{l=1}^{p} \mu_l \overrightarrow{C_l} - 2(M + \Xi)\overrightarrow{x_0}$
2: $\overrightarrow{p_0} \leftarrow \overrightarrow{r_0}$
3: $k \leftarrow 0$
4: **while** $||\overrightarrow{r_0}|| \geq \varepsilon$ **do**
5: $\quad \alpha_k \leftarrow \frac{\overrightarrow{r_k}^T \overrightarrow{p_k}}{\overrightarrow{p_k}^T (2M + 2\Xi) \overrightarrow{p_k}}$
6: $\quad \overrightarrow{x_{k+1}} \leftarrow \overrightarrow{x_k} + \alpha_k \overrightarrow{p_k}$
7: $\quad \overrightarrow{r_{k+1}} \leftarrow \overrightarrow{r_k} - \alpha \left(2M + 2\Xi \right) \overrightarrow{p_k}$
8: $\quad \beta_k \leftarrow \frac{\overrightarrow{r_{k+1}}^T \overrightarrow{r_{k+1}}}{\overrightarrow{r_k}^T \overrightarrow{r_k}}$
9: $\quad \overrightarrow{p_{k+1}} \leftarrow \overrightarrow{r_{k+1}} + \beta_k \overrightarrow{p_k}$
10: $\quad k \leftarrow k + 1$
11: **end while**

The main problem in the first version was the poor convergence properties of Uzawa method. Particularly in this case, the parameter ϱ (line 5 of Algorithm 7) is very hard to set, even considering the proposed parameter [**36**, p. 362]. In our case, all the experiments computed with Uzawa method required to stop in some timestep and restart the computation with different ϱ. This occurs several times, complicating the final computation and obtaining the result in various stages and

not in one execution. Another complication concerning ϱ parameter, was the search of a new parameter value, spending a great amount of computation time.

According to the contact detection module, the first version was built without dynamical memory allocation with respect to the amount of contacts points which are detected. The maximum value of contact points is estimated in function of the amount of solids, therefore, a fixed size of memory is allocated for the contacts, independently the contact points which would be detected. For instance, if we have 500 solids to simulate, then we know that as maximum 8 contact points will be detect for each solid, then it is allocated 4000 contact points in memory.

After the contact detection, the constrained minimization problem must be built. A tridimensional matrix is generated, where the normal percussion matrix, the tangential percussion matrix and the contact vector is stored, therefore, the dimension of this matrix is (number of contacts) $\times$ 7 (3 rows for normal percussion, 3 rows for tangential percussion and one row for contact vector) $\times$ 12 (the six velocities for each solid). The problem arises when the total percussion matrix must be computed. For each contact point an access to this tridimensional matrix is done, therefore, there is a high computational cost. Furthermore, the constraint matrix is computed in the same fashion, increasing the computational time.

4.3. Second version

The main procedure of second version is presented in Algorithm 9.

Algorithm 9 Main Procedure

Require: T: simulation time, Δt: timestep, t_{prox}: proximity time.
Require: file_size : size data, file_position: position data.
Require: κ_N: normal coefficient parameter, κ_T: tangential coefficient parameter.

```
current ← []
t ← 0
get_size_solids()
get_position_solids()
computing_mass()
computing_inertia()
computing_radius()
computing_inertiaMatrix()
while t < T do
    computing_points()
    if t mod t_prox == 0 then
        computing_proximity()
    end if
    current ← contact_detection()
    if current ≠ [] then
        minimization(current)
    else
        updating_velocity()
    end if
    updating_position()
    updating_axis()
    computing_inertiaMatrix()
    t ← t + Δt
end while
```

In first place, it is obtained the size and position for each solid, which are stored in the files size_position and file_position, respectively. After this, the mass, inertia

matrix and radius is computed for each matrix and store in the solid structure (Listing 4.1).

```
1  struct solid_structure{
2    int id;//identification number.
3    double mass;//mass of the solid.
4    double size[3]//size of the solid.
5    double pos[3];//current position.
6    double vel[3];//transation velocity.
7    double omg[3];//rotation velocity.
8    double axis[3][3];//axis i,j,k.
9    double diag_inertia[3];//diagonal inertia matrix.
10   double inertia_matrix[3][3];//inertia matrix.
11   double vertices[NVERT][3];//vertices.
12   double points[NPOINTS][3];//general reference.
13   double faces[NFACE][4];//planes equation for each face.
14   int tabprox[NPROX];//NPROX closest solids.
15   int n_test;
16   double C1,C2,C3;
17   double rad;//radius
18   int idBlock[100];
19   int cantBlock;
20 } solid[N];
```

LISTING 4.1. Struct for a solid.

In the loop, the time variable is set in 0 and it is increased until total time T. For each loop, it is computed the points for each solids, i.e., the current position of all vertices. If $t = 0$ or a multiple of t_{prox} iterations have passed, the computing_proximity method is called. This method computes the neighborhood for each solid using a cell mapping technique [**40**], applied to a three dimensional space. This technique consists in dividing the space in boxes or cubes, and calculating the cube where each solid belongs, for instance, if a solid i is in the position (x_i, y_i, z_i), then belongs to the cube with coordinate

$$\left(\frac{x_i - \min_x}{\text{size}_x}, \frac{y_i - \min_y}{\text{size}_y}, \frac{z_i - \min_z}{\text{size}_z}\right) \tag{4.1}$$

where $\min_x, \min_y$ and $\min_z$ are the minimum coordinates x, y and z respectively, and size_x, size_y and size_z are the each size of the cube. Each cube has three coordinates, which is the position in the space, consequently has a corresponding cube. The neighborhood of the solid will be then, all the solids which are in the closer cubes, i.e. for the cube CUBE(a, b, c) with coordinate (a, b, c), the neighborhood will be formed by the following 26 cubes around:

CUBE$(a, b, c-1)$	CUBE$(a-1, b, c)$	CUBE$(a+1, b, c)$
CUBE$(a, b, c+1)$	CUBE$(a-1, b, c-1)$	CUBE$(a+1, b, c-1)$
CUBE$(a, b-1, c)$	CUBE$(a-1, b, c+1)$	CUBE$(a+1, b, c+1)$
CUBE$(a, b-1, c-1)$	CUBE$(a-1, b-1, c)$	CUBE$(a+1, b-1, c)$
CUBE$(a, b-1, c+1)$	CUBE$(a-1, b-1, c-1)$	CUBE$(a+1, b-1, c-1)$
CUBE$(a, b+1, c)$	CUBE$(a-1, b-1, c+1)$	CUBE$(a+1, b-1, c+1)$
CUBE$(a, b+1, c-1)$	CUBE$(a-1, b+1, c)$	CUBE$(a+1, b+1, c)$
CUBE$(a, b+1, c+1)$	CUBE$(a-1, b+1, c-1)$	CUBE$(a+1, b+1, c-1)$
	CUBE$(a-1, b+1, c+1)$	CUBE$(a+1, b+1, c+1)$

The solids which belong to the neighborhood are stored in an array named `tabprox` in the solid structure (Listing 4.1). This procedure clearly is not an algorithm with quadratic computational time, as in the first version, for several reason: (a) the computation of the cube where a solid is positioned, is linear, because the procedure is

performed for each solid, then we have $\mathcal{O}(N)$; (b) the computation of the neighborhood for each solid is just to collect the solids in the 26 cubes around, also linear computational time, and (c) the neighborhood is not updated at each iteration, every t_{prox} steps, then is a linear computational time procedure as well. An important remark is that, the neighborhood is stored in ascendent order.

In the next step, the contact detection method is called. First a double check is performed: (a) as the neighborhood for each solid is sorted in ascendent order, a solid with greater ID (line 2 of Listing 4.1) is just considered, because, a solid with lower ID was already checked, and (b) an overlapping testing is performed, which involves comparing the distance between both solids and the sum of their radius [**53**, p. 102]; if the distance between a solid a and b is less than the sum of their radius $(r_a + r_b)$, then they are colliding. In any another case, the next step contact detection algorithm is not performed. If the solids could be colliding, then a vertex-face and edge-edge contact detection is performed.

```
1  typedef struct contact{
2    int sol_one;//ID solid 1
3    int sol_two;//ID solid 2
4    //vector from center of solid 1 to contact point.
5    double gp_one[3];
6    //vector from center of solid 2 to contact point.
7    double gp_two[3];
8    //normal vector in the contact point.
9    double normal_vec[3];
10   struct contact *next;
11 }contact;
```

LISTING 4.2. Struct for a contact point

The information is stored in the linked list `current` (Listing 4.2). This structure considers the ID of the solids, the contact vector of each one, i.e. the vector from the center of mass of each solid to the contact point and the normal vector in the

contact point. This linked list is generated dynamically, therefore we use only the memory according to the amount of contact points for each timestep.

Algorithm 10 Contact Detection

Ensure: contact_p

```
1: contact_p ← []
2: contact_p ← surface_contact_detection(contact_p)
3: contact_p ← solid_contact_detection(contact_p)
```

The main procedure of the contact_detection() method is presented in the Algorithm 10. This procedure is composed by two stages: (a) the surface contact detection algorithm and (b) the solid contact detection algorithm.

Algorithm 11 Surface Contact Detection

Require: contactList : a contact structure linked list.
Ensure: contactList : linked list with new contact points added.

```
1: for each solid i do
2:     for each vertex j in the solid i do
3:         if vertex j is below surface then
4:             if solid i moves inward the surface then
5:                 contactList ← add_new_contact()        ▷ contact point is added.
6:             end if
7:         end if
8:     end for
9: end for
```

The procedure of the surface contact detection is presented in the Algorithm 11. In this method, for each solid it verifies if one of the vertices is under the surface (for instance, the ground or a slope). Then, it verifies if the solid moves inward the surface, if it does, the vertex is a contact point where the normal vector will be the normal direction of the surface.

The procedure of the solid contact detection is presented in the Algorithm 12. The method computed two detections: (1) a vertex - face detection and (a) a edge - edge detection.

Algorithm 12 Solid Contact Detection

Require: contactList : a contact structure linked list.
Ensure: contactList : linked list with new contact points added.

```
1: for each solid i do
2:     for each neighbor solid j of solid i do
3:         if solid i and solid j are overlapped then
4:             for each vertex k of the solid i do    ▷ vertex - face contact detection
5:                 if vertex k is inside the solid j then
6:                     nearestFace ← nearest face of solid j to vertex k
7:                     if solid i moves inward the nearestFace then
8:                         contactList ← add_new_contact()
9:                     end if
10:                end if
11:            end for
12:            for each edge k of the solid j do        ▷ edge - edge contact detection
13:                if a section of edge k is inside solid i then
14:                    if solid i moves inward the solid j then
15:                        contactList ← add_new_contact()
16:                    end if
17:                end if
18:            end for
19:        end if
20:    end for
21: end for
```

If two solids are overlapped, then it verifies if any vertex is under the face of his neighbour and if they are getting even closer, on the contrary, the contact point is not verified. In the case of an edge - edge detection, it verifies if the edge of the solid is inside of his neighbour, we take in account as “inside”, even if a interval of the edge is inside of the neighbour solid volume. Then, if they are getting even closer,

the contact point is computed, which it will be the half point of the "inside" section of the edge.

After the contact detection method, if there are contact points detected, the minimization method is called. In this method a constrained minization problem is solved. The problem is given by:

$$\min_{\boldsymbol{x}\in\Omega} \frac{1}{2}\,\boldsymbol{x}^T\,\left(2M+\Xi\right)\,\boldsymbol{x} - \,\boldsymbol{x}^T\,\left(2M\,\boldsymbol{u}^- - \boldsymbol{b}^{\text{ext}}\right)$$
$$\text{where } \Omega = \left\{\boldsymbol{x}\in\mathbb{R}^{6N}\,:\,\boldsymbol{c}_k^T\,\boldsymbol{x} \geq \boldsymbol{c}_k^T\,\frac{\boldsymbol{u}^-}{2},\ k=1,...,n_c\right\} \tag{4.2}$$

The quadratic objetive function matrix is composed by two matrices: mass – inertia and pseudopotential of dissipation. These matrices have the following form:

$$M = \begin{bmatrix} M_1 & Z_3 & \cdots & \cdots & \cdots & \cdots & \cdots & Z_3 \\ Z_3 & I_1 & Z_3 & \cdots & \cdots & \cdots & \cdots & Z_3 \\ \vdots & Z_3 & \ddots & Z_3 & \cdots & \cdots & \cdots & Z_3 \\ \vdots & \vdots & Z_3 & M_i & Z_3 & \cdots & \cdots & Z_3 \\ \vdots & \vdots & \vdots & Z_3 & I_i & Z_3 & \cdots & Z_3 \\ \vdots & \vdots & \vdots & \vdots & Z_3 & \ddots & Z_3 & Z_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & Z_3 & M_N & Z_3 \\ Z_3 & Z_3 & Z_3 & Z_3 & Z_3 & Z_3 & Z_3 & I_N \end{bmatrix}$$

where Z_3 is a null matrix with size 3×3,

$$\Phi_N^d\left(\boldsymbol{x}\right) = \frac{1}{2}\,\kappa_N\,\left(\boldsymbol{x}^T\,\Xi_N\,\boldsymbol{x}\right)$$
$$\Phi_T^d\left(\boldsymbol{x}\right) = \frac{1}{2}\,\kappa_T\,\left(\boldsymbol{x}^T\,\Xi_T\,\boldsymbol{x}\right)$$

Each of these matrices are sparse. Taking advantage of the above, a Block Matrix Information (BMI) was generated. This structure consists in a $30\times N$ matrix (30 is choose as the maximum number of colliding bodies for a solid). For each solid, which corresponds to a column, the number of the solid which is in contact with is

stored. For instance, if the solid i is colliding with solid j and k, therefore, in the column i of the BMI is stored both numbers, j and k. An important fact to remark is that, each column is in ascendent order, but it is not necessary to computed a sort algorithm, because we must to recall that the neighborhood of each solid is already sorted in ascendent form, then, each column of BMI is going to be sorted as well. The aim of building a BMI, is to avoid a $6N \times 6N$ matrix product with a vector. With a BMI it will be computed only the block where the contact was detected. For instance, in the BMI of the Figure 4.2 it possible to observe that the column 2 has the following numbers: 2, 3, 5 and 8, therefore, in the main matrix (quadratic), it will be computed only the blocks with coordinate (2,2),(2,3),(2,5) and (2,8) avoiding the other six blocks which are null matrices.

0	1	2	3	4	5	6	7	8	9
0	1	2	0	1	2	0	6	2	9
3	4	3	2	4	5	6	7	4	
6		5	3	8		7		8	
		8							

FIGURE 4.2. Block Matrix Information

The procedure of the minimization method is presented in the Algorithm 13. In this method, it is computed the main matrix $H = (2M + \Xi)$, the linear part $P = \left(2M\,\boldsymbol{u}^{-} - \boldsymbol{b}^{\text{ext}}\right)$, and the constraints. The solution is computed by the library CPLEX. This solver is set to use a Barrier Optimization Method (interior point method [**43**]) to find the optimal solution. An important remark is that the matrix

structure which CPLEX requires, must be in Compressed Column Storage form (CCS) [**52**], therefore, the BMI is very useful in this case, because we already have the column information of the sparse matrix. Then, the solution is obtained and the velocities for each solid are updated.

Algorithm 13 Minimization - Part I

Require: contactList : a contact structure linked list.
Require: $\boldsymbol{u}^-$: current velocity vector.
Ensure: $\boldsymbol{u}^+$: new velocity vector.

$H \leftarrow 0$: quadratic objetive function matrix,$\boldsymbol{P} \leftarrow 0$: linear part matrix.
C: constraint matrix, $\boldsymbol{B}$: constraint right side vector.
for each solid i **do**
    $H[6i, 6i:6i+2, 6i+2] \leftarrow 2\,m_i$ ▷ (mass matrix)
    $H[6i+3, 6i+3:6i+5, 6i+5] \leftarrow 2\,I_i$ ▷ (inertia matrix)
end for
for each solid i **do**
    $\boldsymbol{P}[6i:6i+2] \leftarrow 2\,m_i\,\boldsymbol{u}^-[6i:6i+2]$
    $\boldsymbol{P}[6i+3:6i+5] \leftarrow 2\,I_i\,\boldsymbol{u}^-[6i+3:6i+5]$
    $\boldsymbol{P}[6i+2] \leftarrow \boldsymbol{P}[6i+2] - m_i\,G\,\Delta t$ ▷ (gravity percussion)
end for
for each contact k in contactList **do**
    ▷ normal pseudopotential
    $H[6i, 6j:6i+5, 6j+5] \leftarrow H[6i, 6j:6i+5, 6j+5] + \kappa_N\,\Phi_N(i,j)$
    $H[6j, 6i:6j+5, 6i+5] \leftarrow H[6j, 6i:6j+5, 6i+5] + \kappa_N\,\Phi_N(j,i)$
    $H[6i, 6i:6i+5, 6i+5] \leftarrow H[6i, 6i:6i+5, 6i+5] + \kappa_N\,\Phi_N(i,i)$
    $H[6j, 6j:6j+5, 6j+5] \leftarrow H[6j, 6j:6j+5, 6j+5] + \kappa_N\,\Phi_N(j,j)$
    ▷ tangential pseudopotential
    $H[6i, 6j:6i+5, 6j+5] \leftarrow H[6i, 6j:6i+5, 6j+5] + \kappa_T\,\Phi_T(i,j)$
    $H[6j, 6i:6j+5, 6i+5] \leftarrow H[6j, 6i:6j+5, 6i+5] + \kappa_T\,\Phi_T(j,i)$
    $H[6i, 6i:6i+5, 6i+5] \leftarrow H[6i, 6i:6i+5, 6i+5] + \kappa_T\,\Phi_T(i,i)$
    $H[6j, 6j:6j+5, 6j+5] \leftarrow H[6j, 6j:6j+5, 6j+5] + \kappa_T\,\Phi_T(j,j)$
end for

Algorithm 13 Minimization - Part II

for each contact k in contactList **do**
    ▷ constraint matrix
    $C[k, 6i : 6i + 2] \leftarrow \boldsymbol{n}_k$ ▷ see 3.14
    $C[k, 6i + 3 : 6i + 5] \leftarrow \boldsymbol{g}_i \boldsymbol{A}_{ijk} \times \boldsymbol{n}_k$ ▷ see 3.14
    $C[k, 6j : 6j + 2] \leftarrow -\boldsymbol{n}_k$ ▷ see 3.14
    $C[k, 6j + 3 : 6j + 5] \leftarrow -\boldsymbol{g}_j \boldsymbol{A}_{ijk} \times \boldsymbol{n}_k$ ▷ see 3.14
end for
for each contact k in contactList **do**
    ▷ constraint right side vector
    $\boldsymbol{B}[k] \leftarrow \boldsymbol{B}[k] + \frac{1}{2} \left(C[k, 6i : 6i + 5] \, \boldsymbol{u}^-[6i : 6i + 5] \right)$
    $\boldsymbol{B}[k] \leftarrow \boldsymbol{B}[k] + \frac{1}{2} \left(C[k, 6j : 6j + 5] \, \boldsymbol{u}^-[6j : 6j + 5] \right)$
end for
▷ solution from cplex library
$\boldsymbol{x} \leftarrow$ cplex_solution(H, P, C, B)
▷ updating velocities
for each solid i **do**
    $\boldsymbol{u}^+[6i : 6i + 5] = 2 \, \boldsymbol{x}[6i : 6i + 5] - \boldsymbol{u}^-[6i : 6i + 5]$
end for

If there are not contact points, then the updating_velocity method is called, where the linear velocities are updated taking in account the equations from classical mechanics, and the rotational velocities are updated taking in account the expressions from Euler equations. Then, the position, axis and inertia matrix for each solid are updated as well.

CHAPTER 5

Computational resources

5.1. Levque Cluster

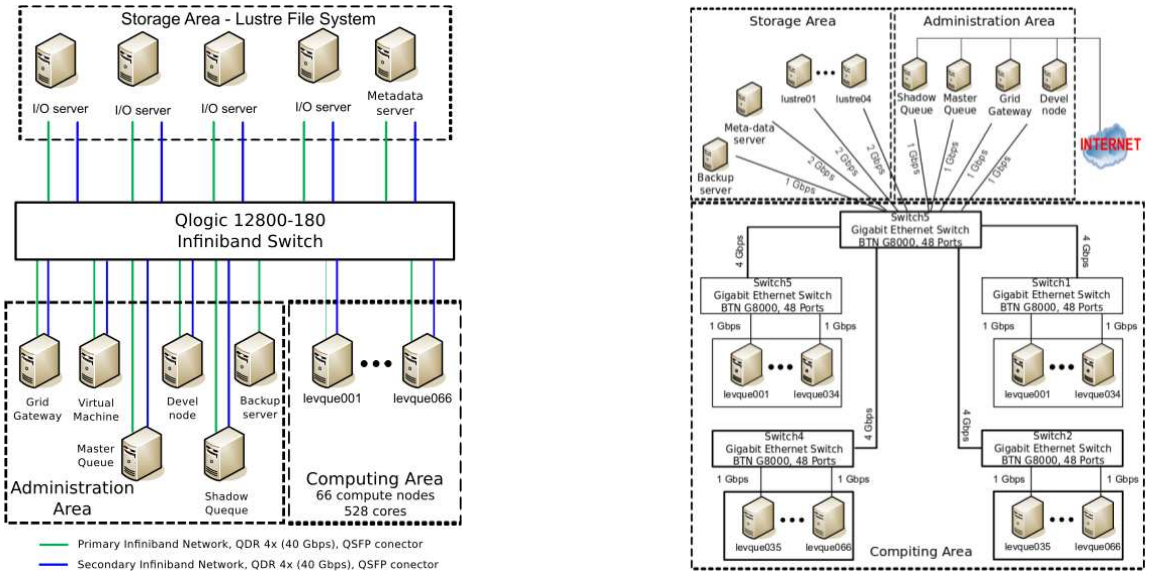
The first version of the numerical simulation program has been developed in FORTRAN 90. This program was performed in the cluster LEVQUE [**98**] as part of the NLHPC (National Laboratory for High Performance Computing) project commanded by the Center of Mathematical Modelling (CMM) at University of Chile and where University Federico Santa María is part as well.

Levque Cluster is an IBM iDataplex machine with 536 compute cores dedicated to satisfy the demand for scientific computing. This machine, funded by the BASAL project PFB-03, offers to researchers a computing power of about 6TFlops. This computational power is achieved thanks to the combination of Intel Nehalem processors, a Qlogic Director Infiniband switch and an appropriate software and hardware integration.

The cluster architecture can be divided into three main areas: the Computing Area, the Storage Area and the Administration Area. These areas are interconnected by means of two separated networks, each one playing different roles within the HPC infrastructure. The computing area is used to perform the scientific computations and it is composed by 67 compute nodes, each one equipped with multicore CPUs, where users run their applications. The storage area provides a scalable persistent layer to the data required by the computing area. The administration area is used to

facilitate the interaction with the computing area and to perform the monitoring of the whole infrastructure as well as the correctness of the users' jobs. These areas are bonded through the interconnection network and the administration network. The former one is used for computing purposes (I/O and IPC for example) and the latter one is used to operate, maintain and monitor the HPC infrastructure.

In particular, the Levque Cluster architecture, composed by a computing area of 67 nodes, representing 536 cores exclusively dedicated to run user jobs. The storage area is composed by 5 nodes, four dedicated only to I/O operations summarizing 8 TB of available space, which is managed by one server known as the meta-data server. Finally, the administration area is composed by four nodes: two acting as the head of the cluster (two master nodes in fail over configuration) and two acting as the interface of the cluster (one for users and one for grid computing).



(a) Interconnection network (b) Ethernet network

FIGURE 5.1. Levque Cluster Network

The interconnection network is a packet switched network based on the Infiniband (IB) technology capable of reaching a throughput of 40Gb/s by port with a very low latency end-to-end. Each node in the cluster is equipped with an IB Host Card Adapter (HCA) with two ports, which are both connected to a switch capable of growing up to 432 Infiniband ports (adding leaf modules). The administration network is composed by 5 Ethernet switches providing a link rate up-to 4Gb/s, from which users can log into the cluster to run their jobs and recover their results. Figure 5.1 (a) depicts the how the interconnection network bond the above defined areas and Figure 5.1 (b) depicts the administration network layout.

The computing area is composed by 66 compute nodes serving the users requirements and one compute node for testing purposes. The first 66 compute nodes are equipped with two quad-core Intel Xeon X5550 processors running at 2.67GHz (each compute node has 8 cores) with 24GB RAM each one. They are monitored and administrated through the Ethernet network and interconnected with the other clusters areas through the Infiniband network. We highlight that each compute node has a dual port IB HCA, which are both connected. This means that each node can establish a high speed communication (40Gb/s) with two compute nodes at the same time. The matlab compute node is equipped with two quad core Intel Xeon E5520 processors running at 2.27GHz with 24GB of RAM. This is equipped with the Ethernet network only. In summary, the Levque cluster offers to users a theoretical compute power of 6TFlops and more than 1.5TB of distributed RAM.

(a) IBM dx360 M2 Compute Node

(b) IBM x3650 M3 Storage Server

FIGURE 5.2. Levque hardware

Levque cluster implements a *Lustre Parallel Filesystem*, which is well known in the HPC community as one of the best open source solutions to provide a storage back-end for HPC clusters. In particular, the Lustre architecture implemented in Levque is composed by 4 I/O servers, one meta-data server and one backup server. The I/O servers are accessed by the compute nodes through the Infiniband network. This interconnection network ensures a high throughput when applications are performing input/output file system operations, since it is capable of performing load balancing and stripping of data among the I/O servers. Each one of these servers is equipped with 5 hard disk of 500GB, providing an aggregated storage space of 8TB with a $n+1$ redundancy of data. The addressing and allocation of space within this storage area is managed by the Lustre meta-data server, which knows the location and redundancy of each object stored on the pool of disks.

5.2. Leftraru Cluster

The second version of the numerical simulation program has been developed in C and performed in the cluster LEFTRARU [**4**], as part of the NLHPC (National Laboratory for High Performance Computing) project commanded by Center of Mathematical Modelling (CMM) at University of Chile and where University Federico Santa María is part as well.

Equipment	Thin Nodes	Thick Nodes	Xeon Phi
Model	HP ProLiant SL230s Gen8	HP ProLiant SL250s Gen8	Intel Xeon Phi
Amount	128	4	12
Processor	2 x Intel Xeon E5-2660 10 cores each	2 x Intel Xeon E5-2660 10 cores each	5110P 240 cores each
RAM	48 GB	64 GB	8 GB
Infiniband	Infiniband FDR 4X Mellanox	Infiniband FDR 4X Mellanox	Shared with thick nodes
Ethernet	2 Ethernet 1 GB	2 Ethernet 1 GB	Shared with thick nodes

TABLE 5.1. Characteristics of Leftaru.

Leftraru has 2640 processors supporting science and research in Chile. The computers are organized in a cluster which contains the characteristics of Table 5.1.

FIGURE 5.3. Leftaru Cluster

The coprocessors Xeon Phi are distributed among the four thick nodes (3 MICS each). Leftraru is based on Linux, which operates through SSH and a queue manager, where Slurm is used. Slurm is a job manager that provides a framework for queueing work, allocation of compute nodes, reservation of CPU time, execution and management of works within the cluster. The compute nodes are divided into partitions Slurm, within which users can run their jobs.

Leftaru offers a variety of modules, where researchers can use. In Table 5.2 are all modules available.

adf/2014.04	gromosxx/1.3.0	libzmq/4.2.0	openmpi_intel/1.6.5
alps/2.2	gsl/1.9	lsst/9.2	openmpi_intel/1.8.1
ampl/20141216	gurobi/6.0.0	mcr/2012a	openmpi_intel/1.8.3
ampl/20141231	gurobi/6.0.3	meme/4.10.0	openmpi_intel/1.8.5
astro/0.1	gurobi/6.0.4	metis/5.1.0	orca/3.0.3
blas/3.5.0	hdf5/1.8.13	mpich/3.1.3	parmetis/4.0.3
cbc/2.8	hdf5/1.8.15	mpich_intel/3.1.3	python/3.4.2
cplex/12.6.1	hopspack/2.0.2	namd/2.9	R/3.1.2
espresso/5.1.1	hwlock/1.9	namd_intel/2.10	R/3.2.0
espresso/5.1.2	hwlock_intel/1.9	namd_intel/2.9	siesta/3.2
ferret/6.93	impi/4.1	nco/4.4.8	siesta/trunk-462
fftw/3.3.4	impi/5.0.0	ncview/1.93g	stata/13
fftw_intel/2.1.5	impi/5.0.2	netcdf/4.3.2	stata/14
fftw_intel/3.3.4	impi/5.0.3	netcdf/4.3.3	symphony/5.6.6
g09/D01	intel/14.0.2	openbabel/2.3.2	weka/3.6.11
gcc/4.8.2	intel/15.0.0	openfoam/2.3.1	wrf/3.6.1
gromacs/5.0.4	intel/15.0.1	openmpi/1.8.1	wrf/3.7
gromos++/1.3.0	lammps/55sep14	openmpi/1.8.3	

TABLE 5.2. Modules available in Leftaru

5.3. Visualization Module

The 3D visualization module has been built in Python. The Visual Python [**8**] is a module which allows us to build 3D graphics. This module was originated by David Scherer in 2000. With VPython it is possible to create navigable 3D display and animations, even for those with limited programming experience.

In this thesis, the visualization module contains three components: (i) the solid class, (ii) the surface function and (iii) the evolution procedure. The last one, is the main program, which read the `evolution.data` file as input and show the evolution of the system.

The `solid class` considers a constructor where the main attributes of each solid is declared: size, position, axis, color, vertices and edges. In the Listings 5.1 we can observe the constructor for the solid class.

```
1  class solid:
2      def __init__(self, size, center, axis, num):
3          self.size = vector(size)
4          self.center = vector(center)
5          self.i = vector(axis[0]).norm()
6          self.j = vector(axis[1]).norm()
7          self.k = vector(axis[2]).norm()
8          self.color = (0.62,0.16,0.11)
9          self.list_of_vertices = []
10         self.list_of_edges = [[],[],[],[]]
```

LISTING 5.1. Solid class

The methods associated to solid class are:

- `draw()`: which generates a box in the space.
- `get_vertices()`: which computes the initial vertices of the solid.
- `set_position()`: which set the position and axis of the solid after an evolution in time.
- `set_vertices()`: which set the vertices of the solid after an evolution in time.

This functions are shown in the Listings 5.2.

```
1  def draw(self):
2          self.obj = box(pos = self.center,axis = self.i,up
               = self.j,size = self.size,color = self.color,
             make_trail = True)
3
4  def get_vertices(self):
5          for i in range(8):
6              vertex_i = self.coef[i][0]*(self.size[0]/2)*
                   self.i
7              vertex_j = self.coef[i][1]*(self.size[1]/2)*
                   self.j
8              vertex_k = self.coef[i][2]*(self.size[2]/2)*
                   self.k
9              vertex =  self.center + vertex_i + vertex_j +
                    vertex_k
10             self.list_of_vertices.append(vertex)
11
12 def set_position(self, center, axis):
13         self.center = vector(center)
14         self.obj.pos = vector(center)
15
16         self.i = vector(axis[0]).norm()
17         self.j = vector(axis[1]).norm()
18         self.k = vector(axis[2]).norm()
19
20         self.obj.axis = self.size[0]*self.i
21         self.obj.up = self.j
22
23 def set_vertices(self):
24         for i in range(8):
25             vertex_i = self.coef[i][0]*(self.size[0]/2)*
                   self.i
26             vertex_j = self.coef[i][1]*(self.size[1]/2)*
                   self.j
27             vertex_k = self.coef[i][2]*(self.size[2]/2)*
                   self.k
28             vertex =  self.center + vertex_i + vertex_j +
                    vertex_k
29             self.list_of_vertices[i] = (vertex)
```

LISTING 5.2. Methods of solid class

The `surface function` generates a set of lines, that formed a plane. The input for this function is a corner of the surface and the height and width which the user wants. This corner (a reference point) is a dictionary with two keys: the point which contains a tuple with the coordinates in the $x - y$ plane, and the size of the surface (width and height) for each coordinate. Then, the object `curve` from Visual Python is used for drawing the vertical and horizontal lines respectively. The first `for` cycle generates the parallel lines to the y axis, and the second one to the x axis. The parameter `delta_h` indicates the gap between two consecutive lines. The code is shown in the Listings 5.3.

```
1 #Input: reference = {'point':(x,y),'delta':(delta_x,
     delta_y)}, delta_h
2 def surface(reference,delta_h):
3     for i in arange(reference['point'][0],reference['
        point'][0] + reference['delta'][0] + 2*delta_h,
        delta_h):
4         curve(pos = [(i,reference['point'][1],0.),(i,
            reference['point'][1] + reference['delta'][1] +
             delta_h,0.)],color=color.black)
5
6     for j in arange(reference['point'][1],reference['
        point'][1] + reference['delta'][1] + 2*delta_h,
        delta_h):
7         curve(pos = [(reference['point'][0],j,0.),(
            reference['point'][0] + reference['delta'][0] +
             delta_h,j,0.)],color=color.black)
```

LISTING 5.3. Surface function

CHAPTER 6

Applications: results and comparison

6.1. Study of a shock - absorber structure

The purpose of this application is to investigate the mechanical behavior of a sandwich structure impacted by a steel ball [**49**]. These structures are used as protection devices agains rock falls and made of a front wall of gabions and an inside layer of sand. Such a structure has been built, instrumented and experimentally tested using a pendular impact facility. Four successive impacts with increasing energy level is applied. For each impact, forces, accelerations and displacements in different locations are measured. All the experimental information is compared with the numerical simulations.

6.1.1. Description of the impact station. The facility used to perform the experimental impacts is a pendulum system, designed and built by CER (Centre d'Expérimentation et de Recherche [**1**]) for the purpose of performing experimental impacts on structures. It is a metallic structure which enables to swing steel ball (254[mm] of diameter, 260[kg] of weight), maintained by two slings, in order to make it impact the tested structure with horizontal speed. The ball is lifted to a chosen elevation thanks to a winch and a steel cable fixed to the backside of the ball. The maximum dropping height allowed by the facility is 4[m], corresponding to a 10[kJ] impact energy named E_{max}. A 3[m] high wall has been built in reinforced concrete and stabilized by a compacted embankment in order to maintain the tested structures

during the impacts and to enable the measurement of the forces generated by the impact at several positions on the backside of the structure (Figure 6.2 - 6.1).

FIGURE 6.1. Tested sandwich structure

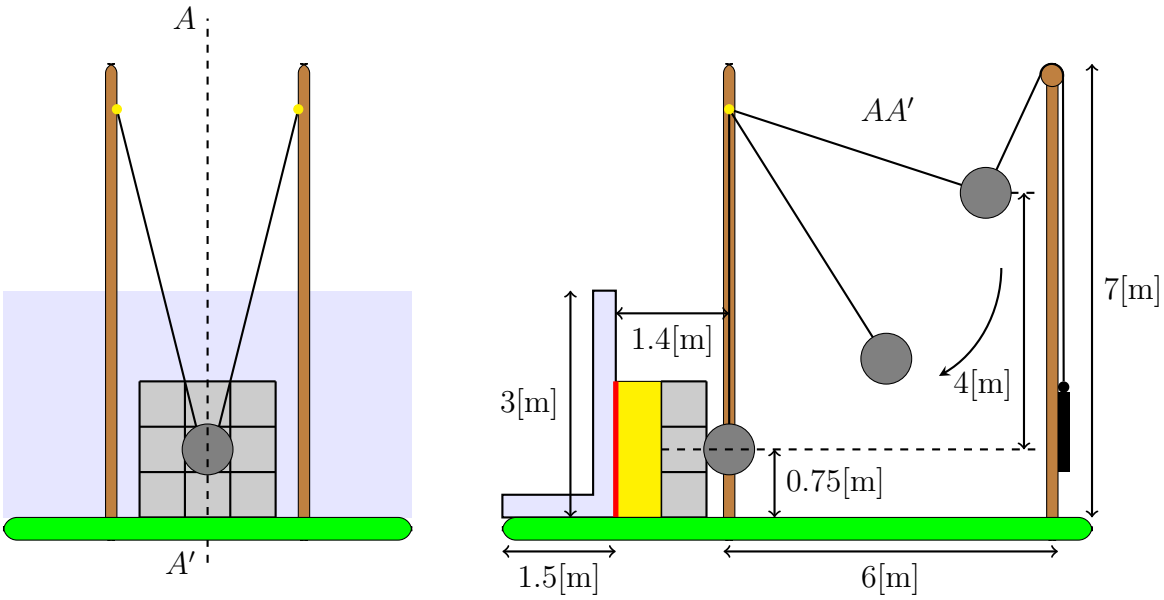


FIGURE 6.2. CER pendular impact facility. Source **[145]**

6.1.2. Impacted structure. The tested structure consists of a front wall of nine gabions and a sand nucleus. The cubic gabions ($\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m]) are made of limestone blocks confined by a steel wire net. Each gabion consists of about 216

blocks. A column of gabions is added on each side of this front wall in order to maintain and stabilize it during the impact. The nucleus is made of Seine sand, $D1$ classified with $w = 7.9\%$ and $\gamma_h = 18[\text{kN/m}^3]$ **[10]**, where w is the water content (quantity of water contained in a material) and γ_h is the force density.

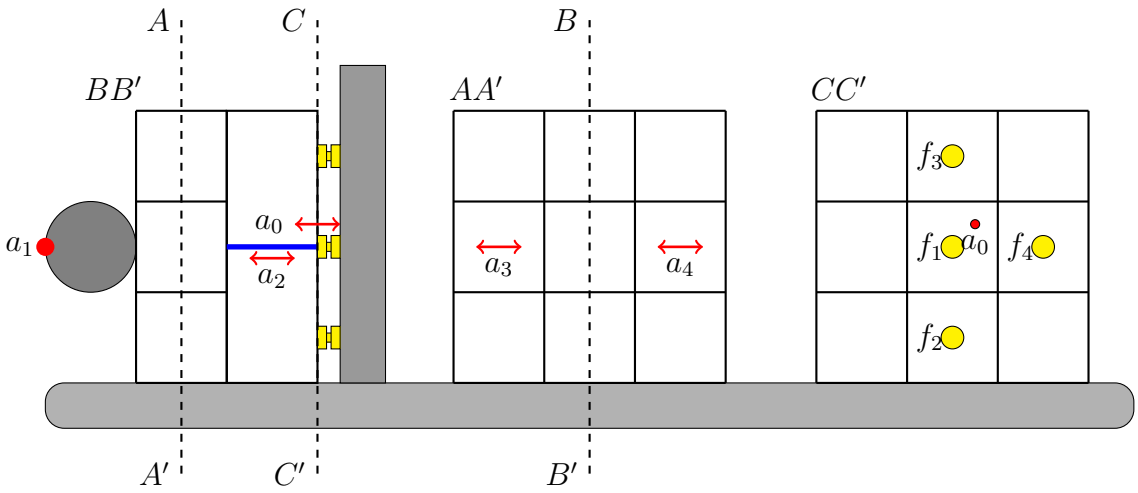


FIGURE 6.3. Instrumentation of the structure (a: accelerometers, f: force sensors). Source **[145]**

6.1.3. Instrumentation. The impacted structure is instrumented with sensors measuring accelerations, forces and displacements during the impact (Figure 6.3). Accelerometers are used to measure the acceleration of the ball (Figure 4(a)) and the acceleration of some of the blocks inside the gabions (Figure 4(c)). For this purpose, "artificial" blocks made of concrete and instrumented with accelerometers are installed inside the gabions. A steel plate containing 4 strain sensors is located vertically between the backside of the nucleus and the concrete wall in order to measure the forces generated by the impacts on the backside of the nucleus **[72]**. Additionally, a displacement sensor is placed between the steel plate and the back

(a) Accelerometer on the iron ball

(b) Displacement sensor

(c) Accelerometer inside a gabion

FIGURE 6.4. Accelerometers and sensors

of the central gabion in order to measure the displacement of the gabion-wall during the impact (Figure 4(b)).

6.1.4. Impacts. The sandwich structure was impacted four times with increasing energy (20%, 40%, 80% and 100% of E_{max} (maximum impact energy)). During each impact, the data of all the sensors were recorded simultaneously and the signals were processed by a second order Butterworth filter with a cut frequency of 600[Hz] [**87**].

6.1.5. Mechanical model. The mechanical complexity of the tested structures makes some assumptions necessary for the numerical simulation of the impacts. The impacted structure is regarded as a granular media (the gabion - wall) with unilateral boundary conditions at its bottom and at both lateral sides, elastic boundary conditions for its contacts with the sand nucleus and "internal" elastic forces due to the confinement by the steel wire net surrounding the gabions.

Thus, the mechanical modeling of the structure requires to create a tridimensional granular media representing the collection of limestone blocks (or grains) of the gabion - wall, to model the interaction of these "grains" with the sand nucleus and the steel wire nettings and to compute the impact of the steel ball by taking into account the unilateral contact boundary conditions between grains, grains and steel ball as well as between grains and the bottom and lateral sides of the gabion - wall.

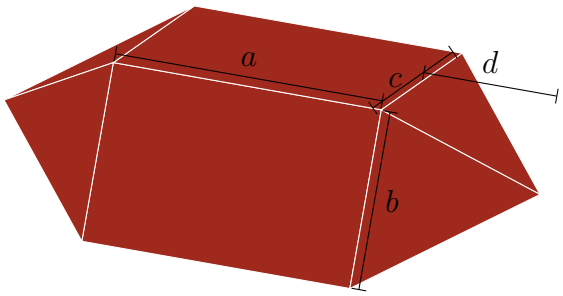


FIGURE 6.5. Shape of the grains

The gabion - wall of the sandwich structure is modeled by a granular media in which each limestone block is a polyhedral grain. The action of the steel wire net is modeled by a confinement force applied to the grains belonging to the boundary of each gabion. The sand nucleus is modeled as a Winkler - Westergard solid **[54]** in contact with the grains of the backside of the gabion - wall. In order to represent

the geometry of the limestone blocks of the gabion, a 12 - sides polyhedral shape is used (Figure 6.5).

Each gabion consists of an average number of 216 blocks. A sample of blocks was measured to obtain the values of a, b, c and d to use for modeling them as described in Figure 6.5. The average values of these parameters are 7.8[cm] for a, 5.3[cm] for b, 6.5[cm] for c and 3.8[cm] for d with a dispersion of 23%.

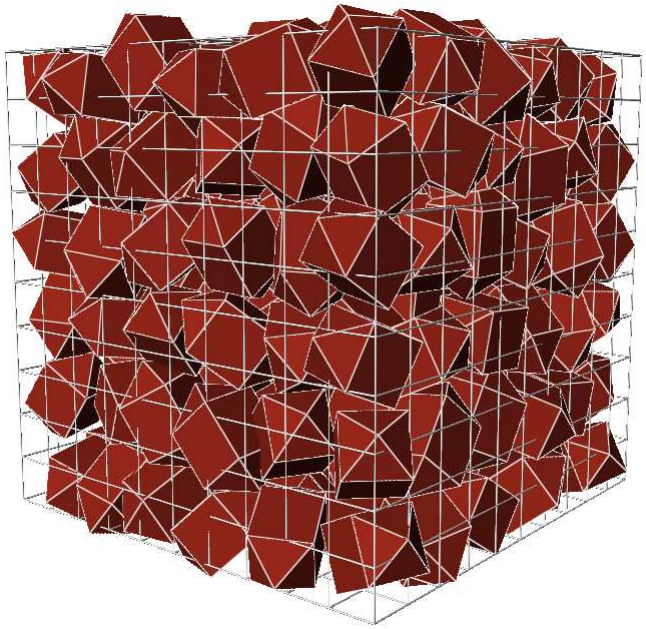


FIGURE 6.6. Gabion with 216 grains

Thus, in order to make a numerical gabion, 216 grains of dimensions a, b, c and d randomly chosen within the range of $\pm 23\%$ of their average values, are randomly disposed and oriented inside a $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] box, following the algorithm recommended in **[83]**. This collection of grains is then compacted by numerical simulation until the containing box reaches the dimensions of the actual gabions (Figure 6.6). This compaction process is achieved considering boundary conditions:

a random amount of grains are dropped to an empty gabion. A collision process is initiated until the compactness is obtained.

The numerical compaction is done by applying, to each side of the box, inwards percussion on each grain in contact with any side of the box. Nine different "numerical" gabions were computed this way. The grid drawn on Figure 6.6 represents the $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] dimensions of a gabion.

The steel wire net surrounding a gabion and insuring its confinement is then modeled by a force applied to each grain having a part of itself outside the $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] $\times$ $\frac{1}{2}$[m] gabion and proportional to the volume of the grain that is outside this domain (Figure 6.7).

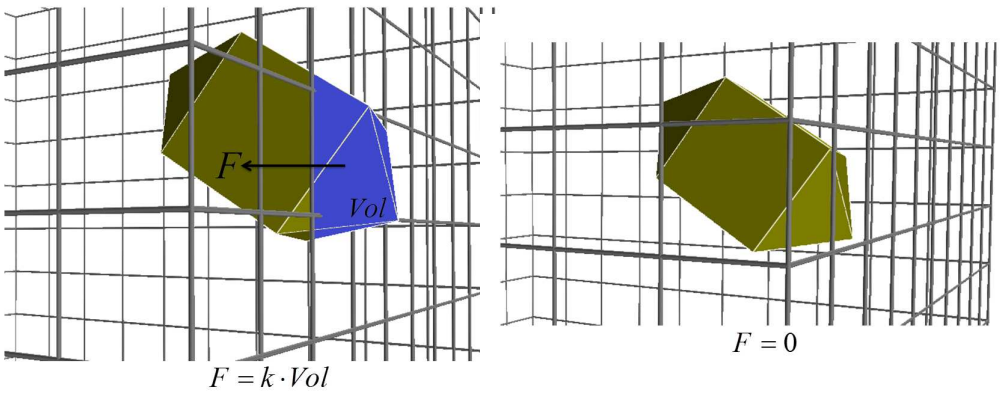


FIGURE 6.7. The force applied to the grain is proportional to the part of its volume outside the boundary of the gabion (left); the force is null if the grain is inside the gabion (right)

The use of $k = 10^{11}$[N/m^3] allowed to obtain nine size - and shape - stabilized numerical gabions which are placed like in the experimental structure, in order to build the gabion - wall.

Thus it obtains a system of 1944 polyhedral grains in unilateral contact and submitted to elastic confinement forces at the boundary of the gabions. Let be precise

that, a grain from the boundary of a gabion which is submitted to the confinement force (if a part of it crosses the boundary of the gabion he belongs to), is submitted to contact forces due to its contacts with grains of the gabion he belongs to, and also to contact forces due to its contacts with grains from the neighbor gabion.

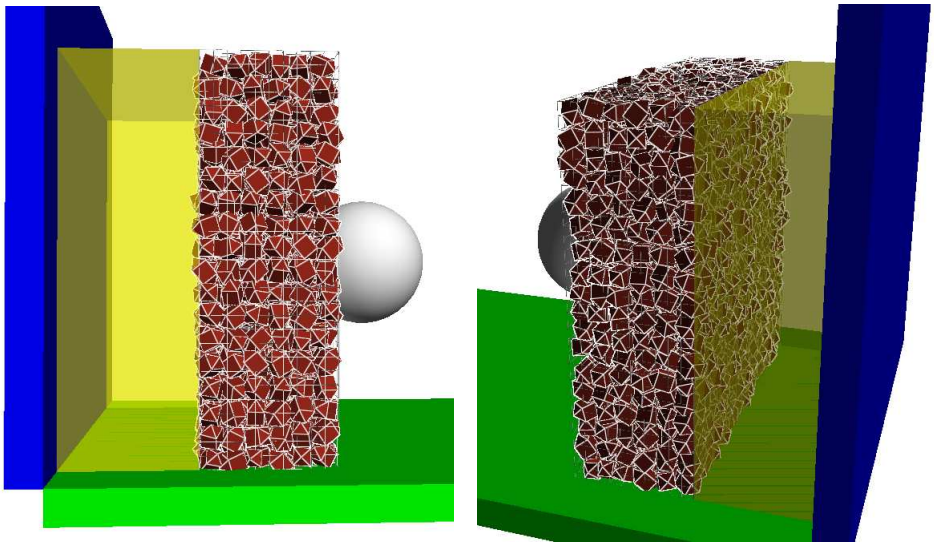


FIGURE 6.8. The initial position of the ball and the 9 gabions. The transparent yellow region is the volume of sand.

As the experimental impacts do not show any significant displacement of the two extra columns of gabions disposed at each lateral side of the nine gabions wall in order to stabilize it (Figure 6.9), they are not modeled by a collection of grains, but by unilateral contact boundary conditions for the grains. Unilateral contact boundary conditions are also applied to the grains of the bottom of the wall to model the interaction with the soil.

The backside of the gabion wall is in contact with the nucleus of the structure, its most deformable part, made of sand and backed on to a rigid concrete wall. Experimental measurements of Young's modulus of the used sand have values around 300[MPa](±7%). This part of the structure is modeled by a Winkler - Westergaard (Figure 6.10) elastic solid **[54]** in the direction of the impact. In this direction, a displacement Δz corresponds to an elastic force $F = k_{ww}\Delta z$.

This boundary condition is to be applied to each grain of the backside of the gabion wall during the impact. According to the measurements of Young's modulus of the sand, k_{ww} is set to 3×10^4[N/m].

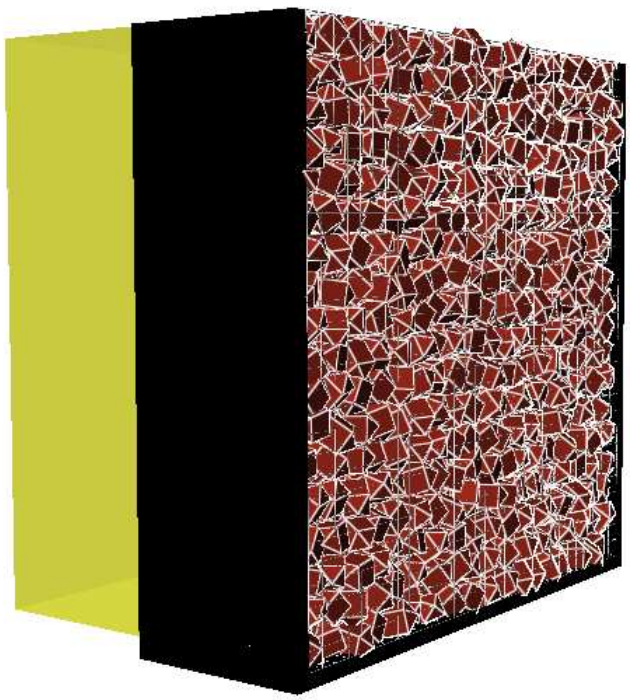


FIGURE 6.9. Unilateral boundary condition (black zone)

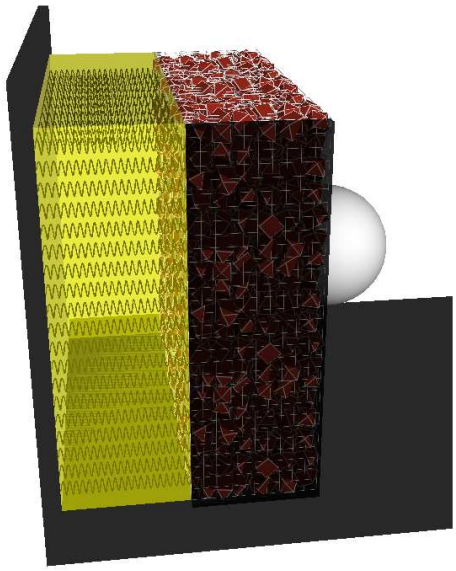
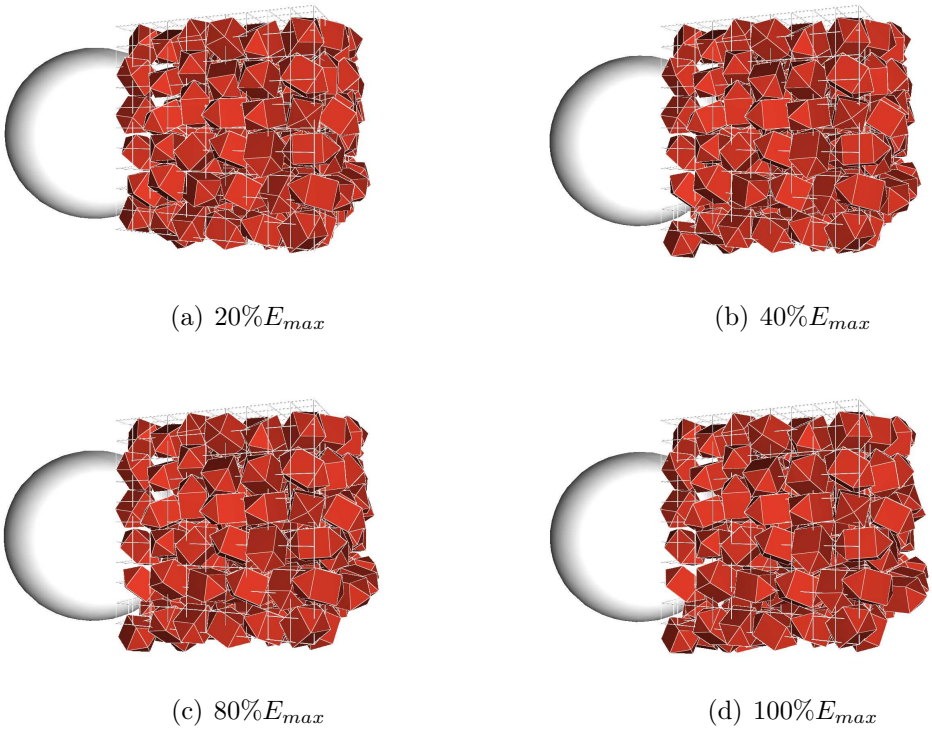


FIGURE 6.10. Winkler - Westergaard boundary condition.

The numerical simulation of the impact on the sandwich structure has been performed by the A-CD^2 method which has the purpose of computing 3D polyhedral granular media dynamics [**133, 45, 47, 46**].

As in the non - smooth Contact Dynamics method [**80**], the grains are in unilateral contact. The A-CD^2 method is applied for expressing the contact forces, having for consequence a different formulation of the equations to solve at each time step.

For the numerical simulation of the impact on the mechanical model of the sandwich structure, the forces not depending on the velocities are the gravity force and the elastic forces applied to the grains by the Winkler - Westergaard [**54**] solid and the steel wire netting. The gravity force does not depend on the positions and its expression is easy to get for any grain. The mentioned elastic forces depend on the positions and therefore need to be computed taking into account the positions of the grains.



(a) $20\%E_{max}$ (b) $40\%E_{max}$

(c) $80\%E_{max}$ (d) $100\%E_{max}$

FIGURE 6.11. Impacts at different level of energy.

6.1.6. Numerical computations and comparison. Similarly to the experimental testing of the structure, four successive impacts with increasing level of energy are computed with the mechanical model of the sandwich structure, as is shown in Figure 6.11.

The time step used for the computation was $\Delta t = 10^{-3}$[s], making the friction force between grains a linear viscous force with a viscosity coefficient $\upsilon = 0.3$[Ns/m]. To cover the duration of the signals recorded during the experimental collisions, 100 time steps are computed for each collision. At each step, the number of contacts to take into account is between 6200 and 8600 in the system.

For each impact, forces, accelerations and displacements are computed in the mechanical model at the places where they have been measured during the experiment.

Thus, we obtain the numerical signals to compare with the experimental ones for the validation of the model.

The forces are measured and computed at the positions f_1, f_2, f_3 and f_4 (right picture in Figure 6.3). The computation of the forces is made thanks to the deformation of the Winkler - Westergaard solid. In the Figures 6.12 - 6.15 are presented the comparison of the experimental and numerical signals at each one of these spots for the four successive impacts of 20%, 40%, 80% and 100% of E_{max}.

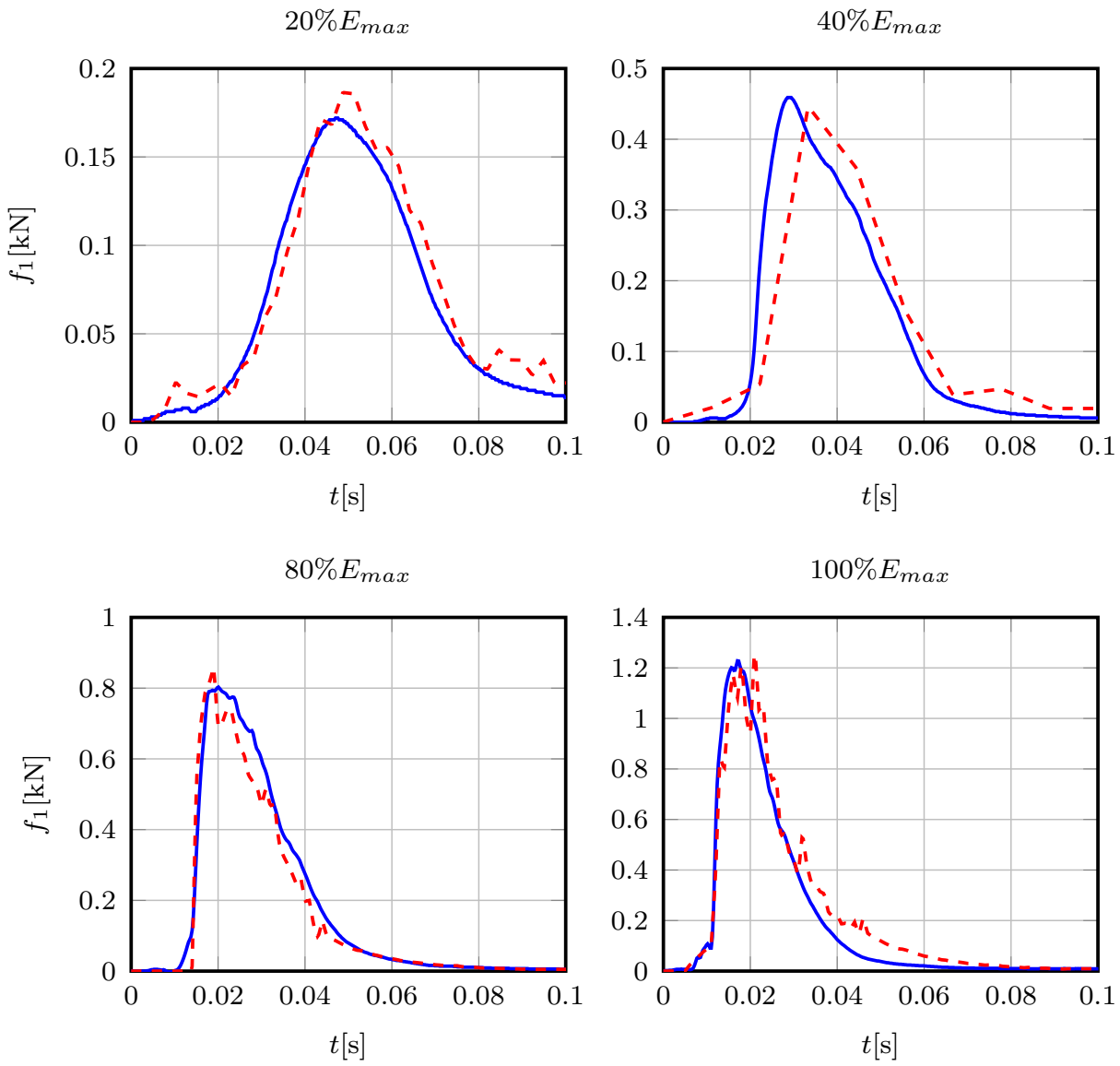


FIGURE 6.12. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $f_1(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

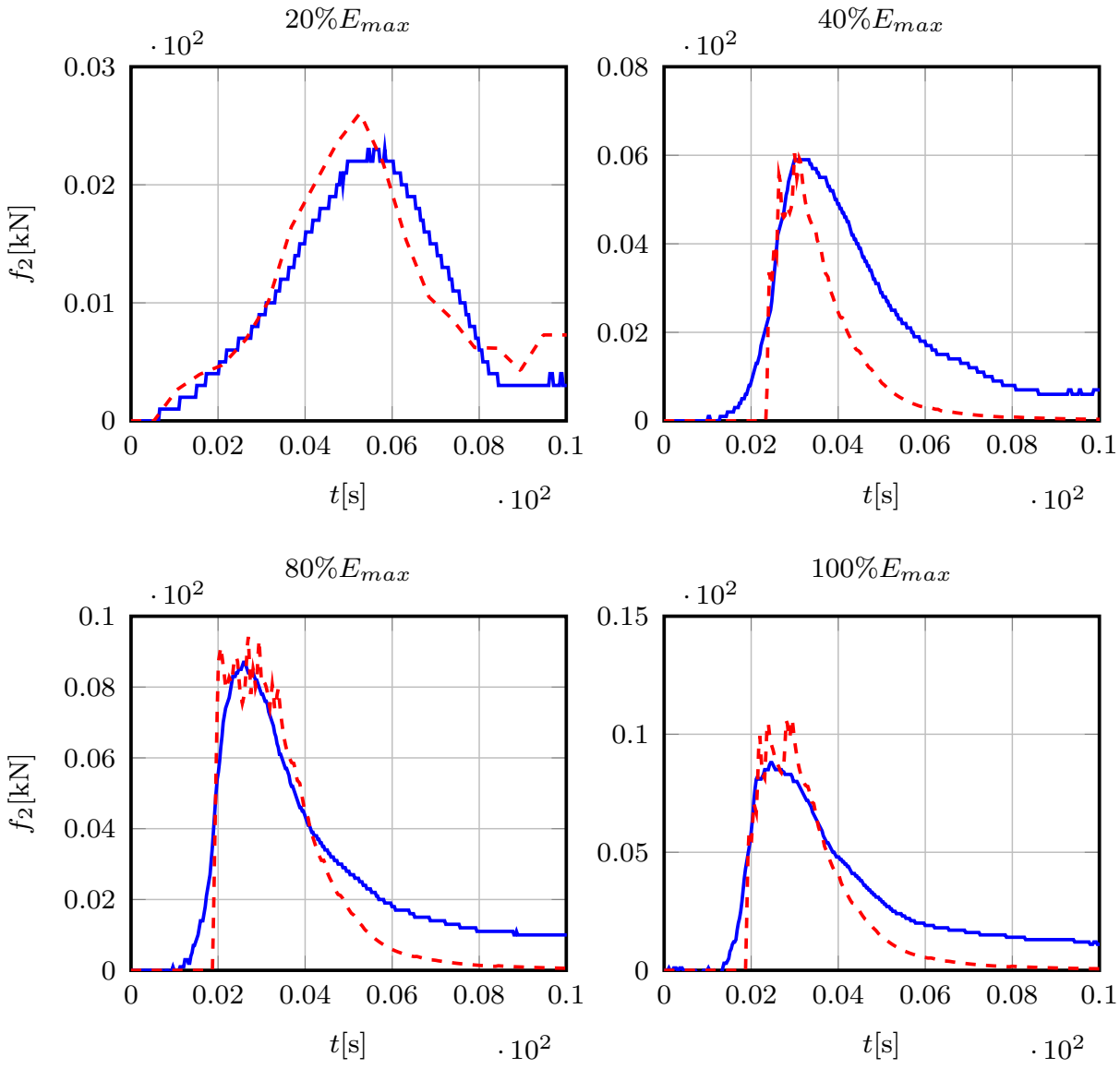


FIGURE 6.13. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $f_2(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

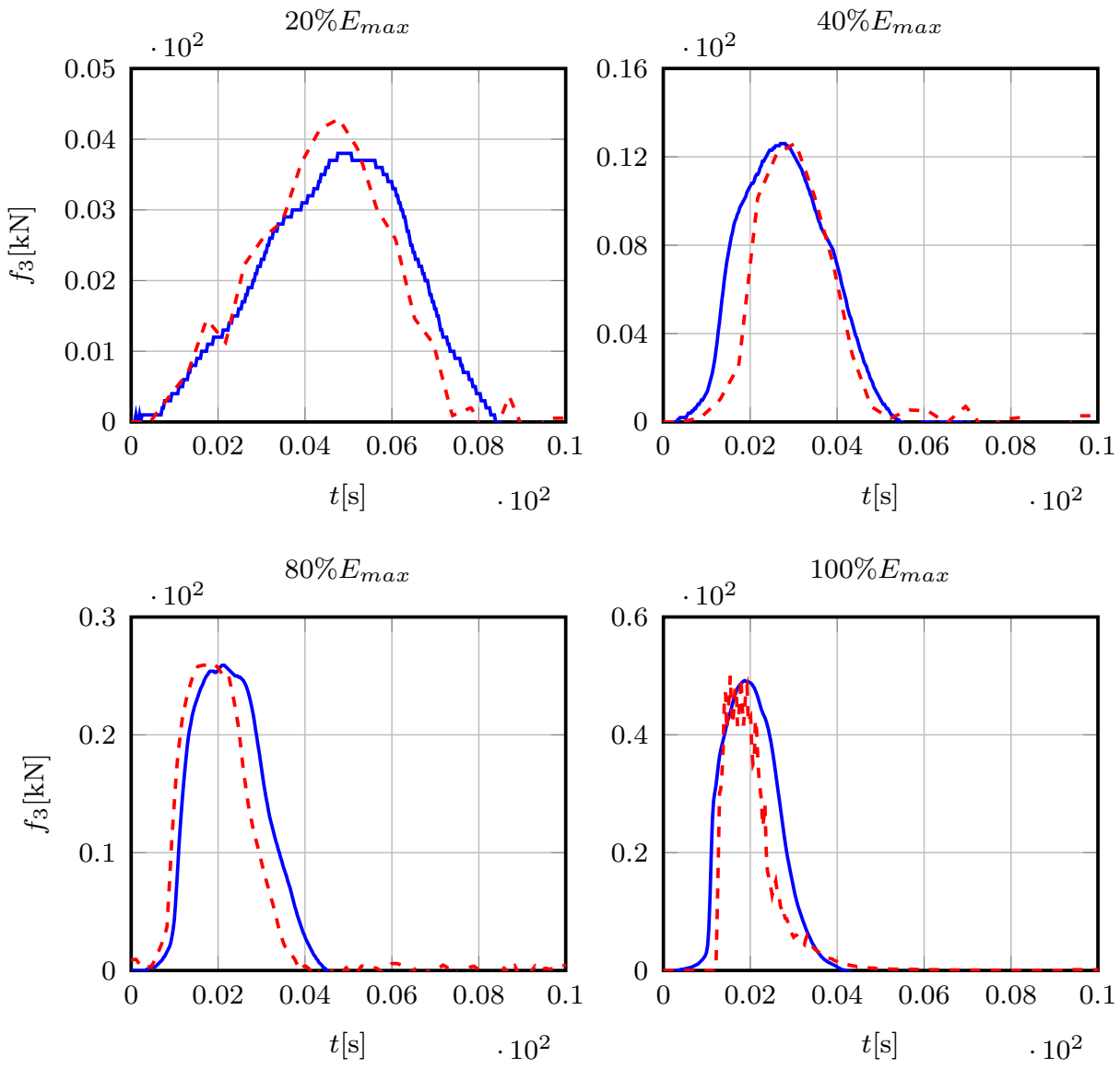


FIGURE 6.14. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $f_3(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

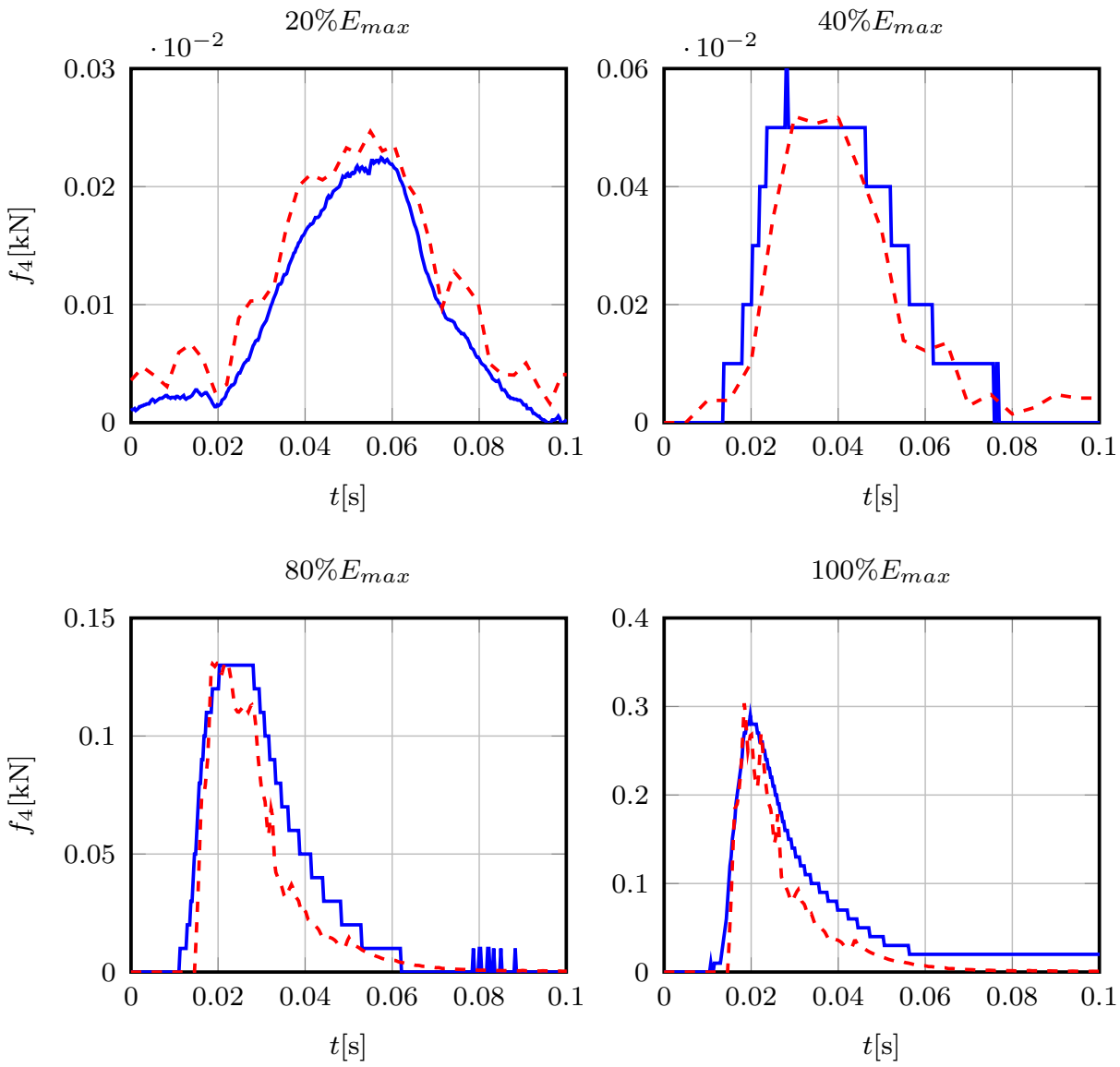


FIGURE 6.15. Comparison between experimental (blue continuos) and numerical (red dashed) signals of $f_4(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}

There are five points where the accelerations were measured: a_0, a_1, a_2, a_3 and a_4 (Figure 6.3). The point a_1 corresponds to the steel ball horizontal acceleration and the other points correspond to horizontal acceleration at specific positions for three grains. The acceleration a_2 corresponds to a grain located in the back of the gabion, and it has measured the acceleration of the motion toward the boundary of the structure; whereas that the acceleration a_0 corresponds to the motion at back side of the sand wall (left picture in Figure 6.3). The acceleration a_3 and a_4 correspond to two grains located in the middle of the structure (middle picture in Figure 6.3) and it has measured the acceleration of the motion toward the left and the right of the central gabion respectively. In the Figures 6.16 - 6.19 are presented the comparisons between the experimental and numerical accelerations signals for the grains in the points a_0, a_2, a_3 and a_4.

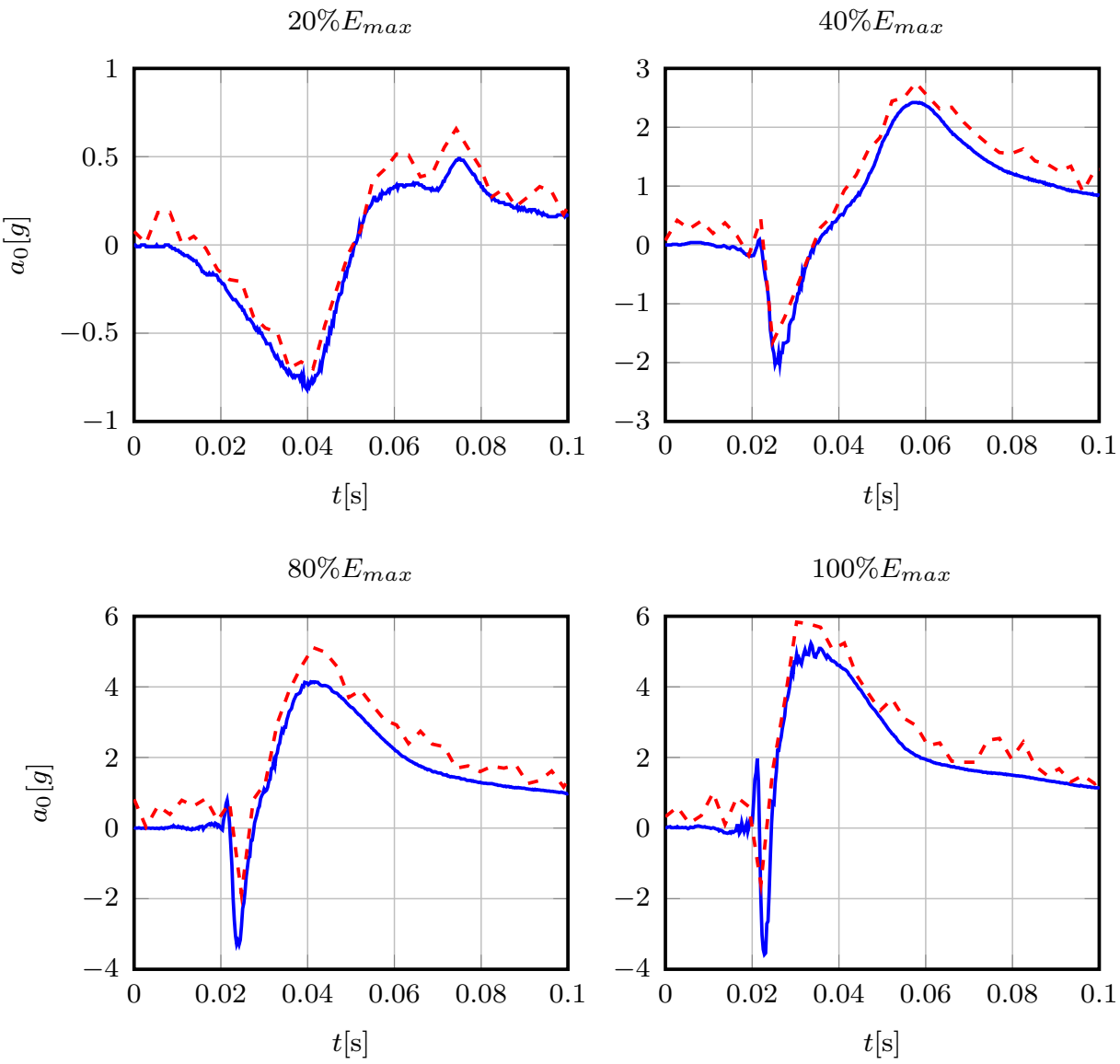


FIGURE 6.16. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_0(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

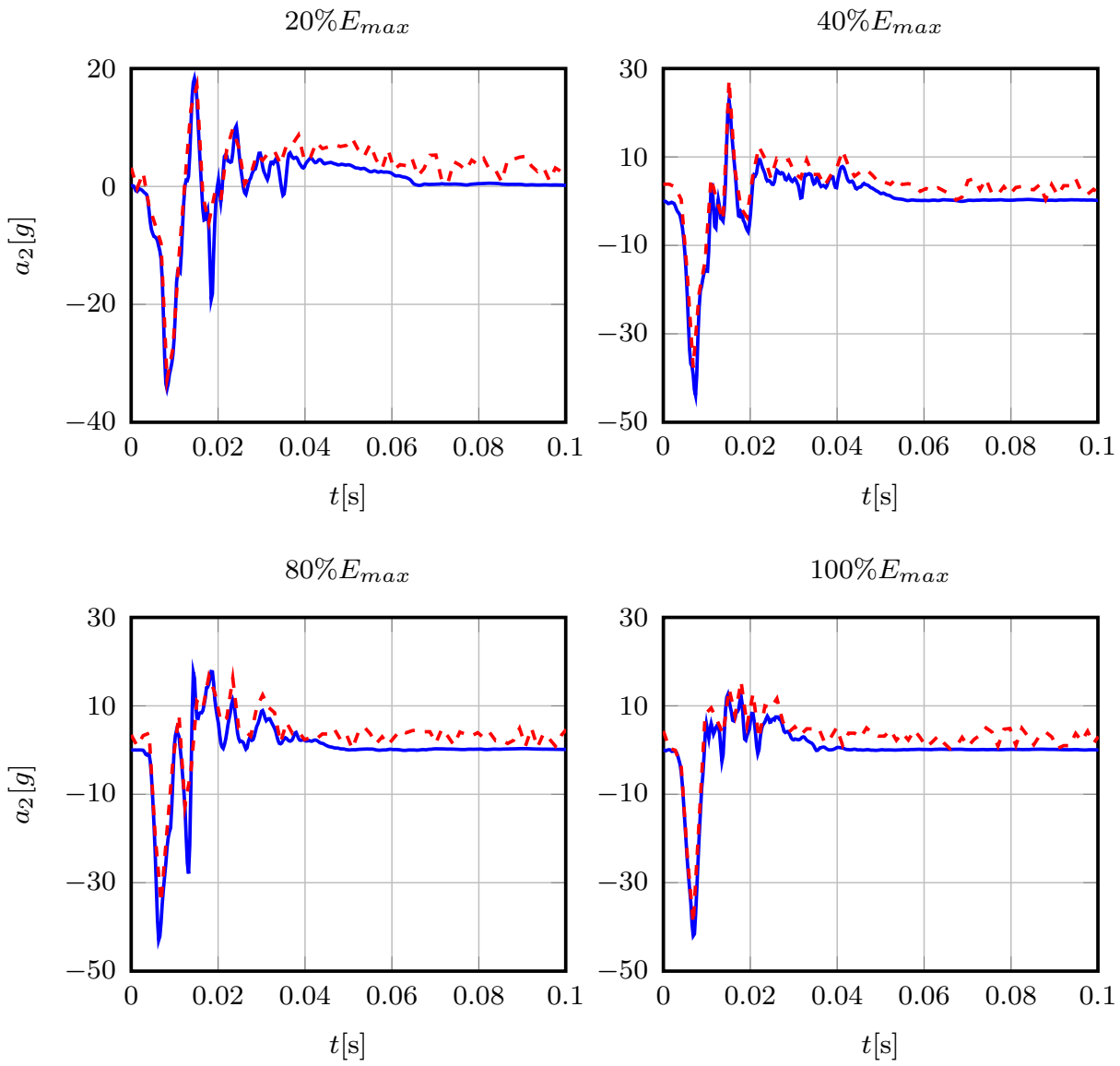


FIGURE 6.17. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_2(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

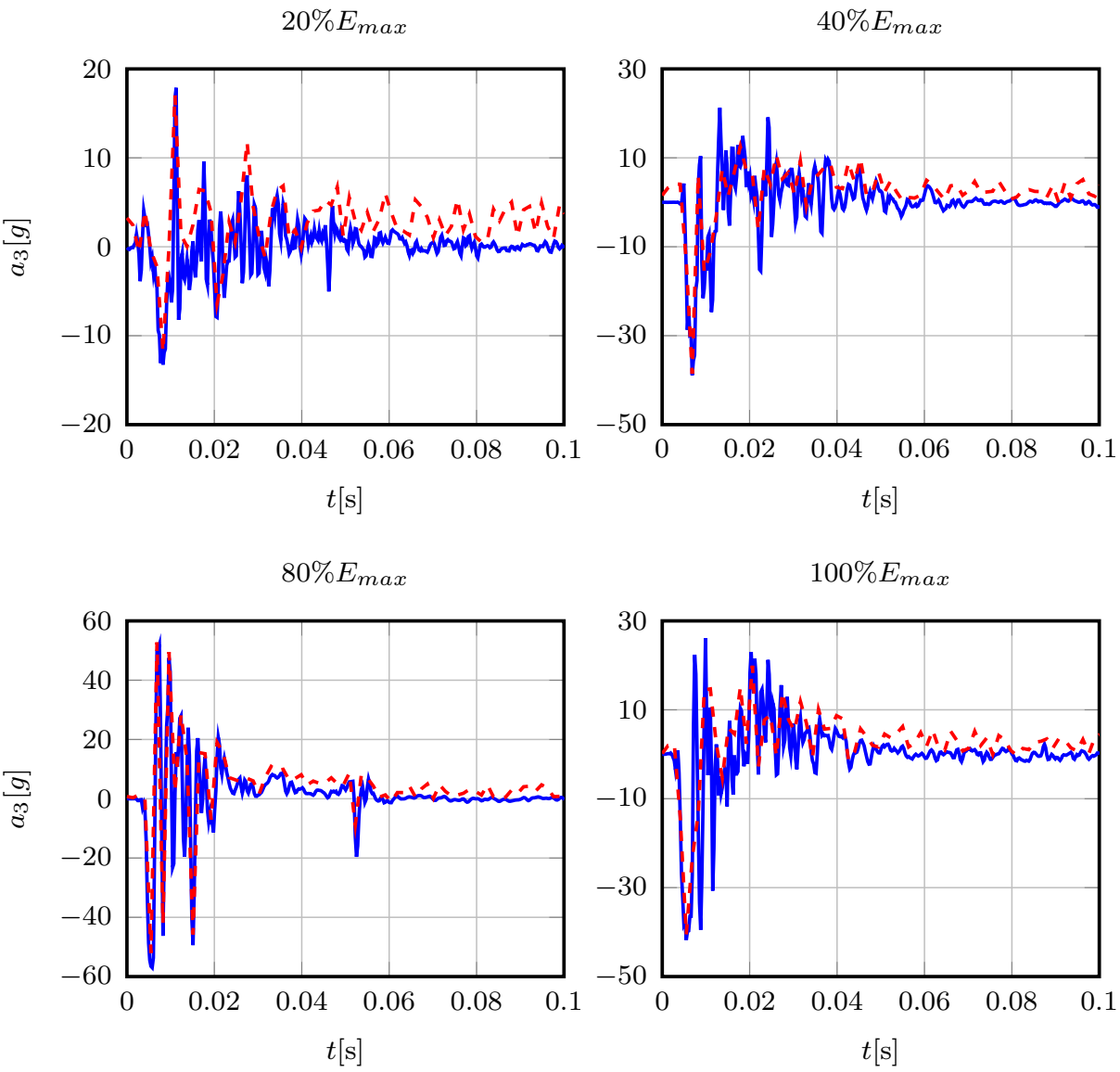


FIGURE 6.18. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_3(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

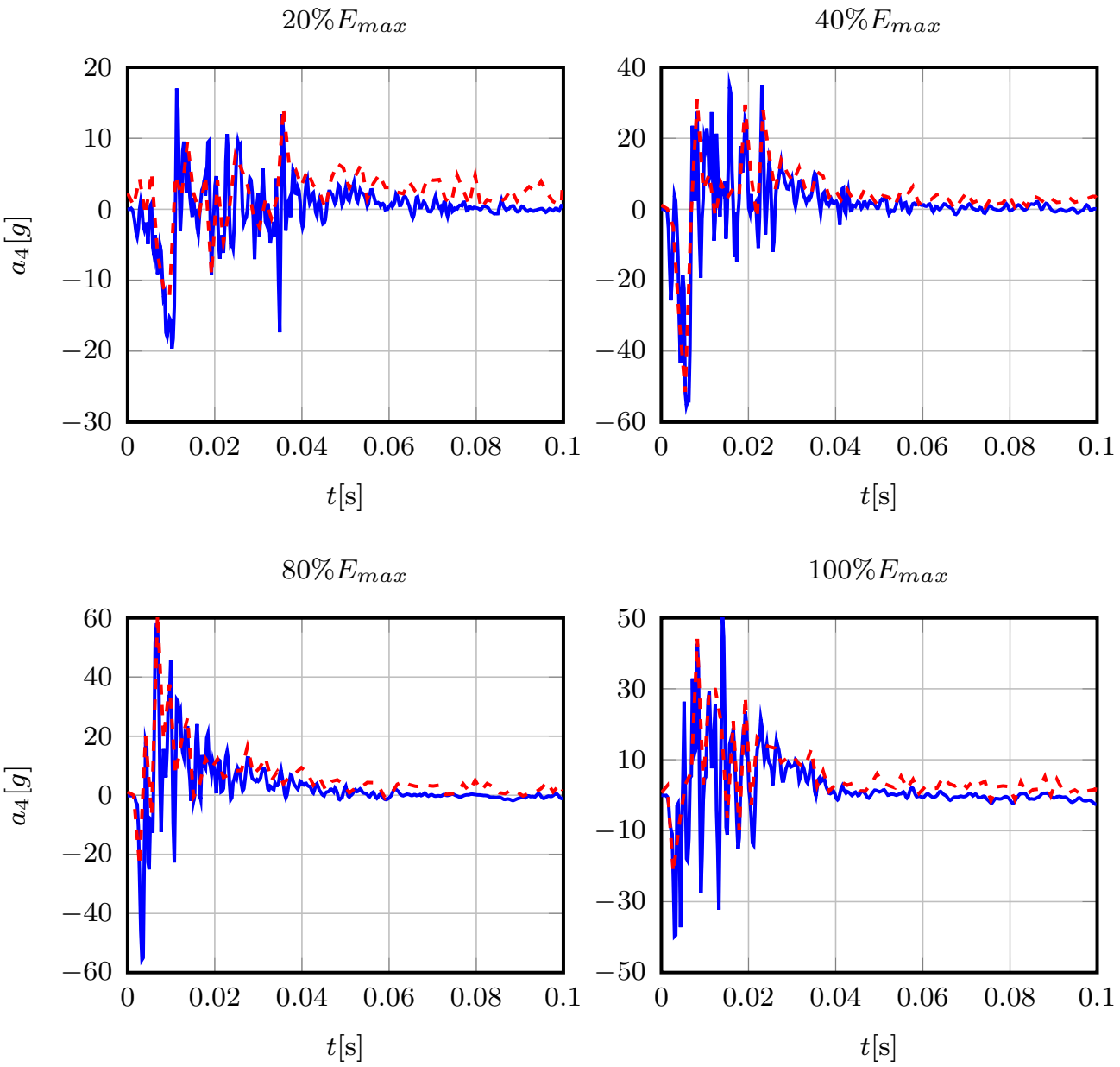


FIGURE 6.19. Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_4(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

Concerning the measurements of the steel ball acceleration, the accelerometer could only register the signal peaks for the first three impacts. The problem could be fixed for the last impact only. For this reason, it was only possible to obtain the experimental acceleration signal during the impact with 100% of E_{max}, which is presented in Figure 6.20.

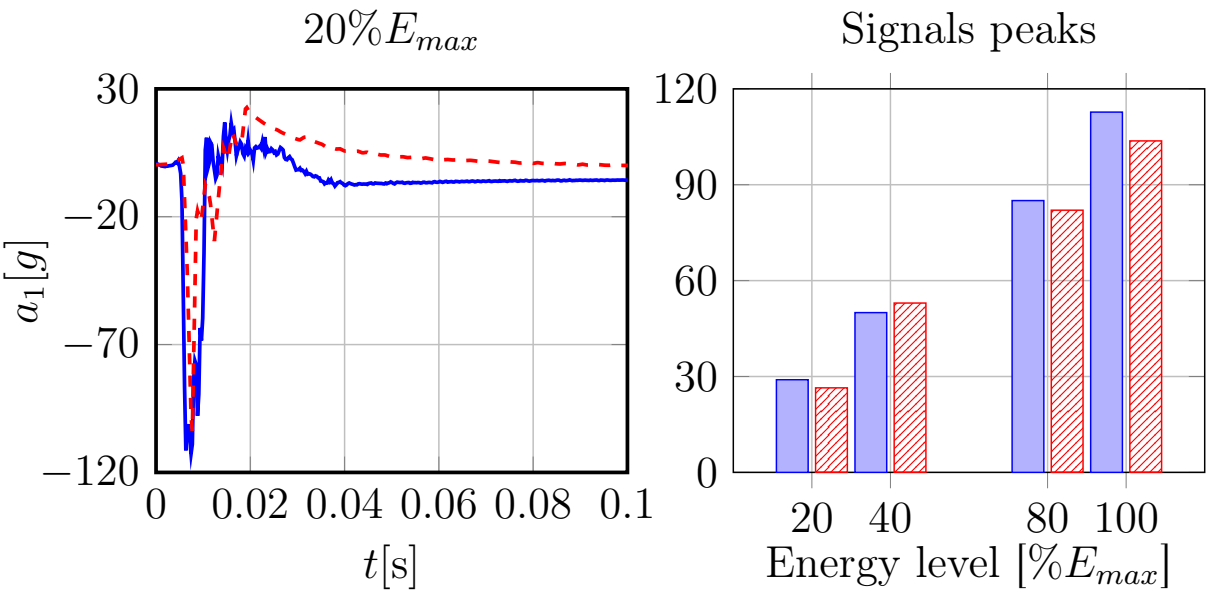


FIGURE 6.20. Left: Comparison between experimental (blue continuous) and numerical (red dashed) signals of $a_1(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}. Right: Comparision between experimental (blue) and numerical (red hatched) signals peaks of $a_1(t)$ for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

The displacement sensors measured the maximal displacement of the blocks belonging to the backside of the gabion - wall during each impact. In the numerical simulation, this displacement is computed by comparing the positions of the corresponding grains in the backside of the gabion - wall to their positions before the first impact. In Figure 6.21 is presented the comparison between maximum displacements for the four impacts with different energy levels.

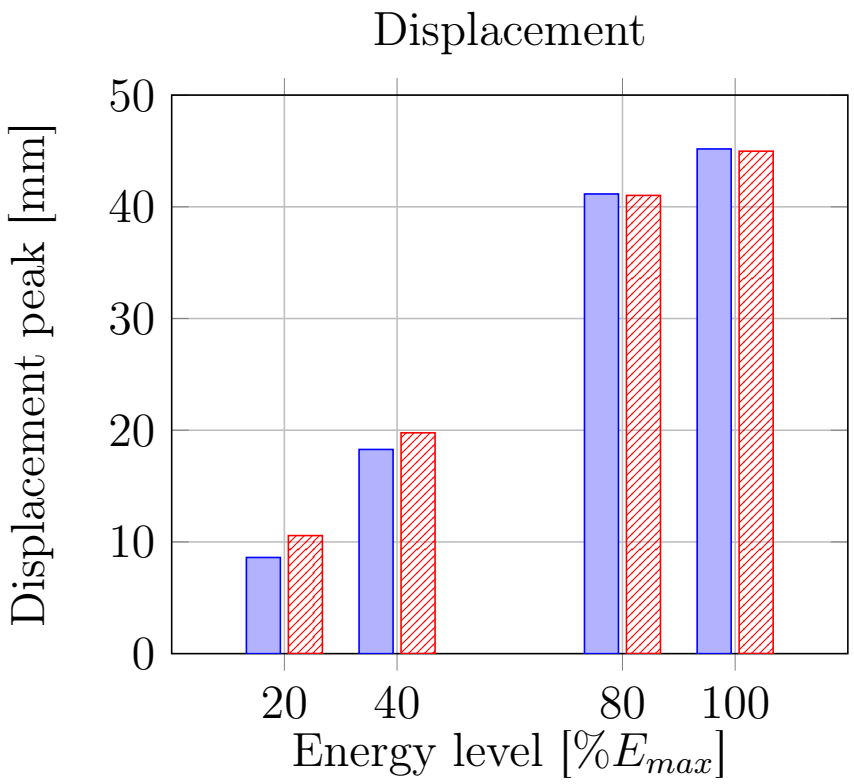


FIGURE 6.21. Comparison between experimental (blue) and numerical (red hatched) displacements for the 4 impacts of $20\%, 40\%, 80\%$ and 100% of E_{max}.

Concerning the CPU time (in seconds) for the four impacts with different energy level, see Table 6.1.

Impacts			
$20\%E_{max}$	$40\%E_{max}$	$80\%E_{max}$	$100\%E_{max}$
4527.525	5718.016	3435.088	7768.229

TABLE 6.1. CPU time for impacts.

6.2. Study of a rockslide real event

A major collapse of the Piton Marcel in Martinique, France, has occurred during the night on May 11th 2010. The materials which were at the top of the cliff for a couple of days, has been remobilized on May 17th 2010, and it caused a first mudflow along the Rivière du Prêcheur (River Preacher). From June 25th 2010, there was no trace left of the collapse at the top of the cliff, all sediments had been remobilized. In the Figure 6.22 it is possible to observe the deposit of material at the bottom of the cliff after the collapse.

The main study of this event is to replay the particular collapse with a numerical simulation; for achieving this, it has been necessary to obtain the information of the materials which were in the top of the cliff. The Airborne LiDAR data have been acquired before and after the collapse. Hence, two accurate tridimensional point clouds have been available to reconstruct the exact location of the missing material and to generate realistic initial conditions for the simulation. The first set of points has been used to obtain the initial position of the material which it is necessary to estimate the volume which has fallen; and the second one to obtain the final deposit of the material which it is necessary to make a comparison with numerical results. Finally, another important aspect obtaining these two point clouds, is the relieve surface where the numerical simulations have been done; the difference between the initial position of the material and the final one, allows to build the relieve surface.

FIGURE 6.22. Collapse of the Piton Marcel in Martinique, France.

6.2.1. Data set acquisition. Airborne LiDAR (Light Detection And Ranging) is a system which provides the topography of a landscape as geo – referenced 3D point clouds by the measurement of the time - of - flight of a short laser pulse (generally 1064[nm]) once reflected on the Earth surface as is shown in Figure 6.23 **[16]**. They operate at high pulse repetition rates from 50[kHz] to 400[kHz] depending of the flying conditions. Supporting vector attitudes are calculated using the synergy of both GPS and inertial measurements (IMU) providing at the top end process a height accuracy less than 0.05[m] and a planimetric accuracy about 0.4[m] or less depending on the flying conditions as well as on the surveyed topography (tridimensional structure) **[153]**.

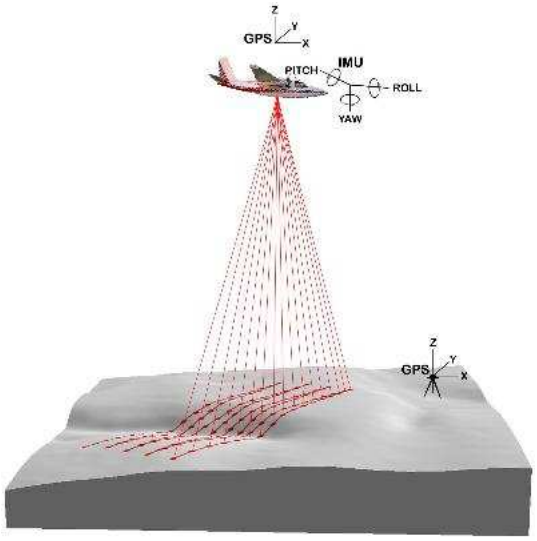


FIGURE 6.23. A sketch of an airborne LiDAR survey

Moreover, such active systems, called multiple echos LiDAR, allow detecting several return signals for a single laser shot. It is particularly relevant when surveying vegetation areas since a single LiDAR survey allows acquiring not only the canopy (upper part of the vegetation layer), but also points inside the vegetation layers as well as on the ground underneath. In the framework of the French national project Litto3D [**5**], LiDAR data have been acquired both on the sea side around the Martinique Island as well as over the entire island surface. This survey was operated by the French National Survey (ING) in March 2010, with the Optech ALTM 3100 LiDAR device [**7**]. After the Piton Marcel collapse and the successive mudflows during May and June 2010, local authorities have decided to survey with LiDAR system the entire Rivière du Prêcheur, from the Piton Marcel to the seashore on July 22nd 2010. This survey was operated by the Swiss company Helimap [**3**] from a helicopter. In the Figure 6.24 is shown both the IGN (color scale, red points are the

highest ones) and the Helimap data set (blue points) in a perspective front view of the collapse area.

FIGURE 6.24. 3D LiDAR points before the collapse (IGN survey displayed in height color scale) and after the collapse (Helilmap survey - blue points)

The height difference between both point clouds (before and after the collapse) has been calculated with the open source software CloudCompare (CC) **[2]** as is shown in Figure 6.25. Clearly it is possible to observe the deficit of material along the cliff (red area) which reaches 130[m] straight down.

The height difference becomes smaller in white areas to null in blue ones. Figure 6.25 confirms that all collapsed materials have been remobilized during the mudflow events since height difference along the thalweg ("valley way", it refers to a line drawn to join the lowest points along the entire length of a valley in its downward slope, defining its deepest channel) is null, contrary to what has been observed just after the collapse (Figure 6.22).

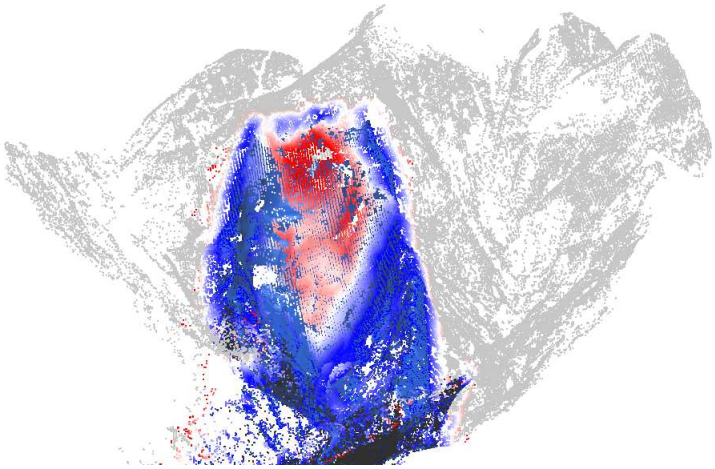


FIGURE 6.25. Height difference between both point clouds coded in a polar scale (blue to red) and display in the same perspective as in Figure 6.24. Height difference comes from 0[m] to 60[m]. Grey points lie outside the collapse area.

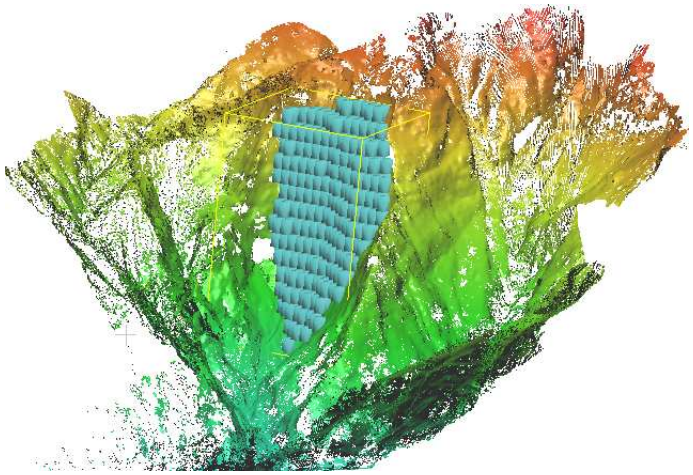


FIGURE 6.26. Location of the initial set of block within the LiDAR Helimap data set.

In order to prepare the initial position for the numerical simulation, it has been isolated a region of interest around collapsed area (red to white area in Figure 6.25). It has been calculated a triangulation of the upper (IGN data set) and of the bottom (Helimap data set) point clouds with the end to delineate the volume of the collapsed material. Then, this volume is filled with similar parallelepiped finite elements where

dimensions are set both to make a paving of the 3D bounding box and to belong to the delineated volume. A fair estimate of the total volume is $8\times10^5[\mathrm{m}^3]$. Considering the geometry of the collapse, finite elements have been oriented along the z direction and coplanar to the orientation of the remaining cliff. Figure 5 shows the inclusion of these blocks in the Helimap used for the numerical simulation and can be considered as the initial position.

6.2.2. Numerical Simulation. The total material which has fallen from the top of the cliff has been simulated by a volumetric partition, this means, the total volume has been split in discrete elements with the objective to complete as much as possible the area which has been released. Then, each one of these discrete elements is considered as a numerical rock, or block, which is represented by a regular polyhedron of 6 faces with fixed dimensions.

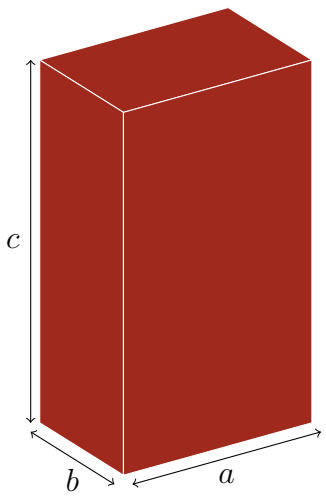
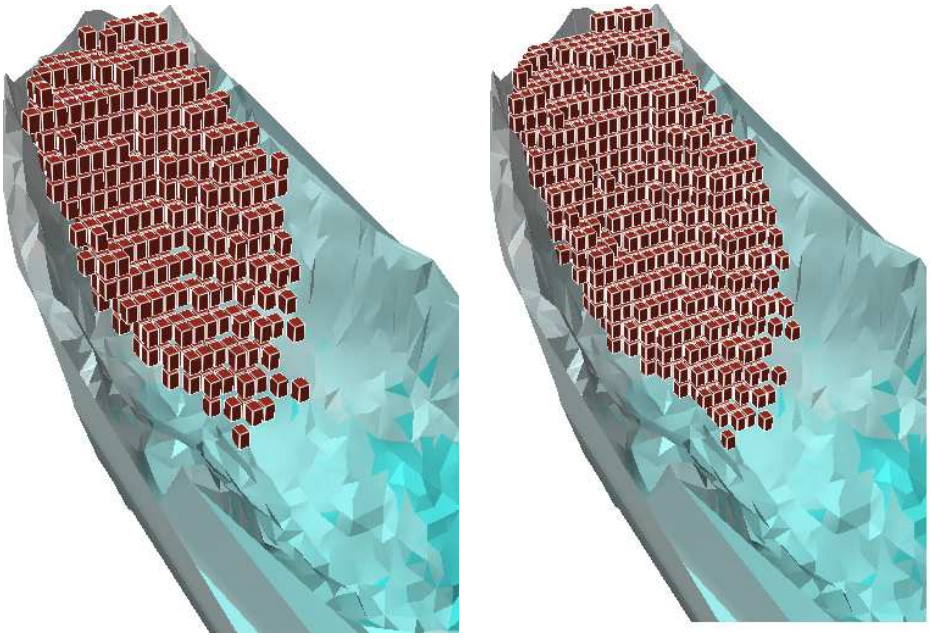


FIGURE 6.27. A rock represented by a regular polyhedron with dimensions a, b and c.

As it has been mentioned before, the numerical rocks have been positioned in the orientation z and coplanar to the remaining cliff. In the Figure 6.27 it is possible to observe a representation of each numerical rock with width a, length b and height c.

The rock avalanche has been simulated considering two set of blocks: 435 and 874. In the first one, each block has dimensions $10,935$[m], $9,636$[m] and $15,775$[m] respectively, with a volume of $1662,21$[m^3]. In the second set of blocks, each one has dimensions $8,852$[m], $7,801$[m] and $12,771$[m] respectively, with a volume of $881,85$[m^3].

In the Figure 6.28 it is possible to observe the two set of blocks with their respective initial position; on the left with 435 blocks and on the right with 874 blocks. A visualization module has been developed to obtain the results of the numerical simulation. This module has been developed in VPython which allows building several geometric elements such as boxes, triangles, spheres, cylinders, pyramids, etc. In this numerical simulation, with this visualization module has been possible to build the surface of the remaining cliff with the point cloud given by LiDAR Helimap; this construction has been possible with the triangulation of these points and each set of blocks is representing by the regular polyhedrons explained before. The visualization module also has allowed building an evolution program, which consists in to observe the behavior of the blocks falling by the cliff after to obtain the numerical results. Furthermore, it is possible to obtain instantaneous position of the numerical avalanche evolution with the objective to compare the final deposit, or final position, of the blocks versus the real final deposit.



(a) 435 blocks (b) 874 blocks

FIGURE 6.28. Initial positions of the numerical rock avalanche. Display made by VPython module visualization.

The numerical simulation of the rock avalanche then, considers a set of blocks positioned in the top of the cliff, a time step for the simulation, which is the time interval where the blocks move with constant velocity, this parameter has been set in 0.1[s] with total time of simulation of 35[s], and the surface of the cliff which is determined by 2371 points and 4709 triangles.

Several results can be obtained with this numerical simulation: it is possible to obtain the position of the blocks during the entire evolution, checking if the behavior corresponds to the real case; furthermore it is possible to obtain the kinetic energy curve, translational and rotational, and the dissipation energy curve as well. In the Figure 6.29 the curves are from kinetic energy, the first one in the top is the total kinetic energy, in the middle there is the translational kinetic energy curve and in the bottom the rotational kinetic energy curve. According to these curves,

it is observed that in the interval $5[s] - 10[s]$ the total energy is in the maximal value, therefore in this interval the system has the maximal velocity due to the increasing collisions between the blocks. An interesting result is at the beginning of the energy curve; approximately at 2[s] there is a small decreasing, this is due to that the solids in the bottom start to collide with the relief, then there is a small drop in velocity but it is rapidly recovered in the next 10[s]. Finally the energy curve decreases to zero due to the reduction of the system velocity since the amount of collisions between the blocks declines. In the Figure 6.30 it is observed the instantaneous dissipative energy curve at top, the accumulative dissipative energy curve at middle and the potential energy at bottom, all these curves, including from kinetic energy, has been computed with 435 blocks. According to the dissipative energy, the instantaneous curve shows that the maximal dissipation is reached at around 12[s] which is consistent with the maximal value of kinetic energy, also it is possible to observe that at the end of the time evolution, the dissipative energy decreases to zero, which also is consistent with the real behavior of the avalanche. The dissipative accumulative curve is monotonically increasing, in fact, around 12[s] there is a characteristic impulse, this is due to the same impulse in the instantaneous energy; at the end, the accumulative curve obtains a constant behavior, due to the decrease of energy.

The potential energy curve is in constant decrease, which is consistent, due to the blocks are falling, then the system loses potential energy during the evolution; at the end, the potential curve obtain a constant behavior due to the blocks is in the bottom of the cliff, therefore, they do not continue falling down.

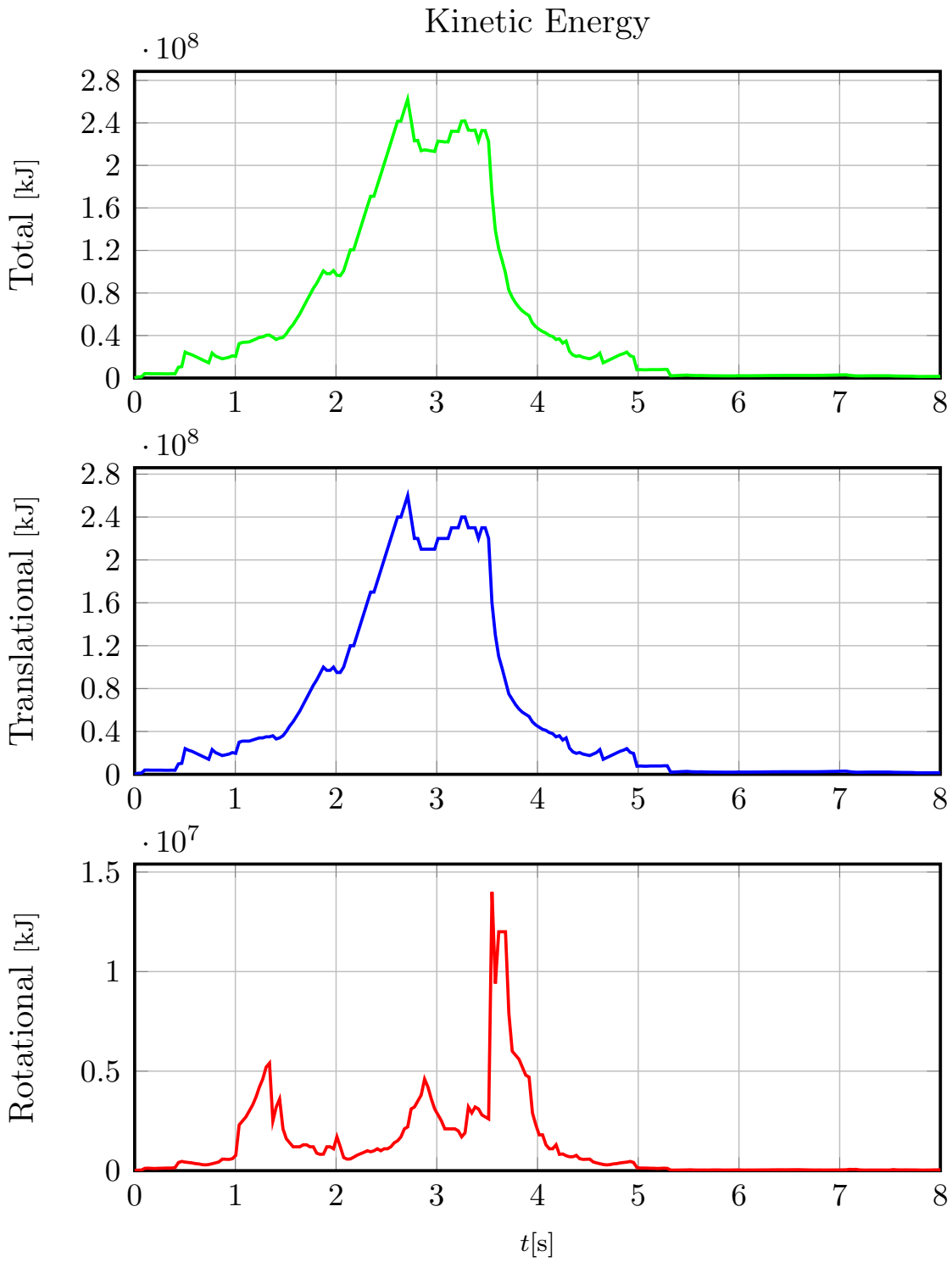


FIGURE 6.29. Kinetic energy with 435 solids.

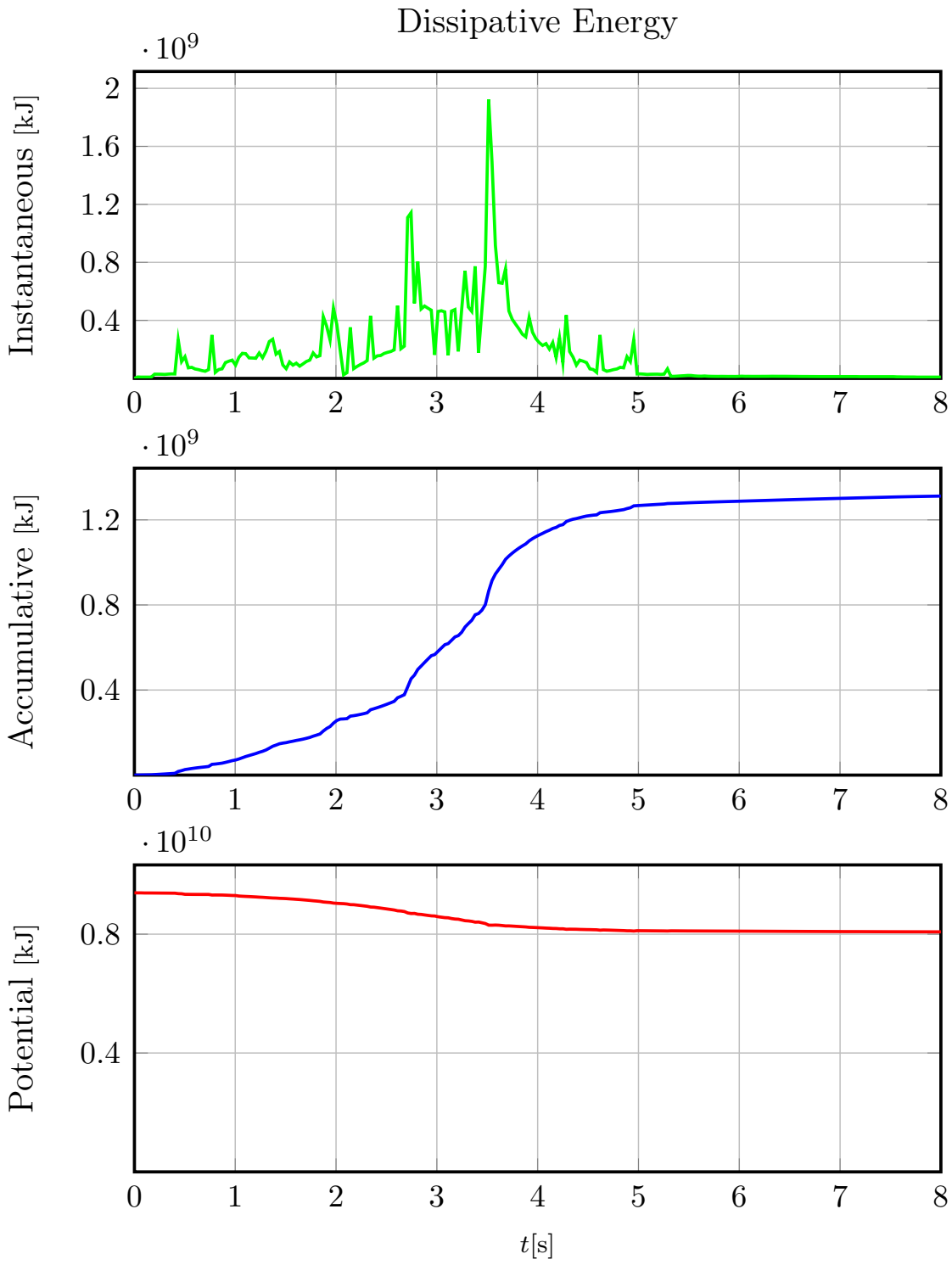
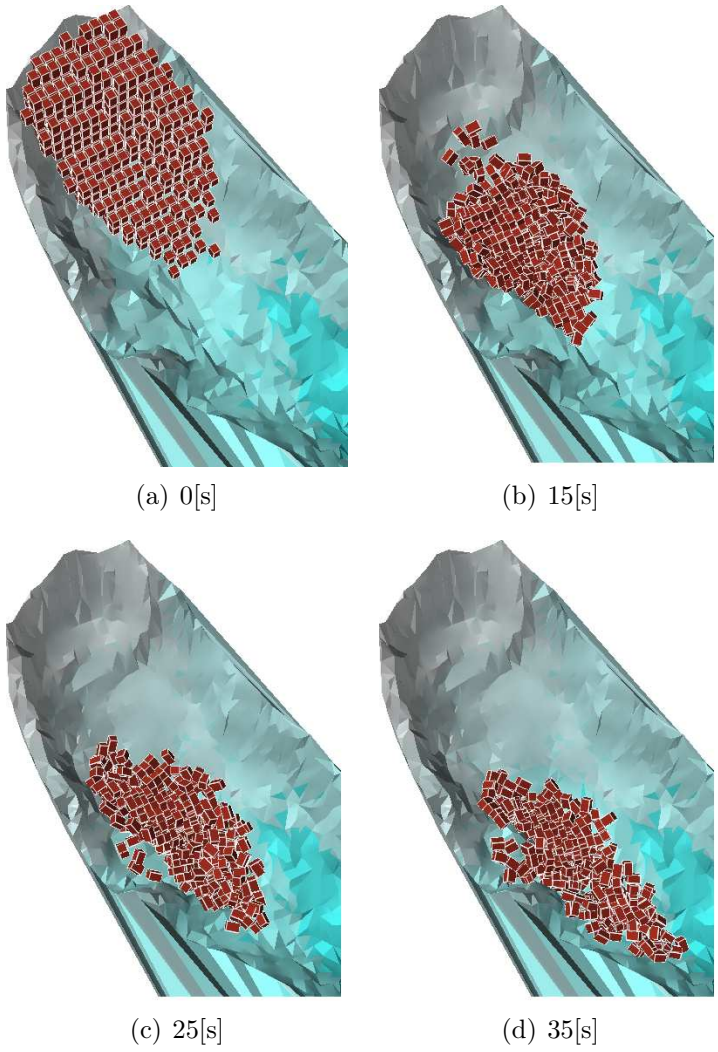


FIGURE 6.30. Dissipative energy with 435 solids.

In Figure 6.31 the evolution of the numerical avalanche with 435 blocks and for different time step is observed.



(a) 0[s] (b) 15[s]

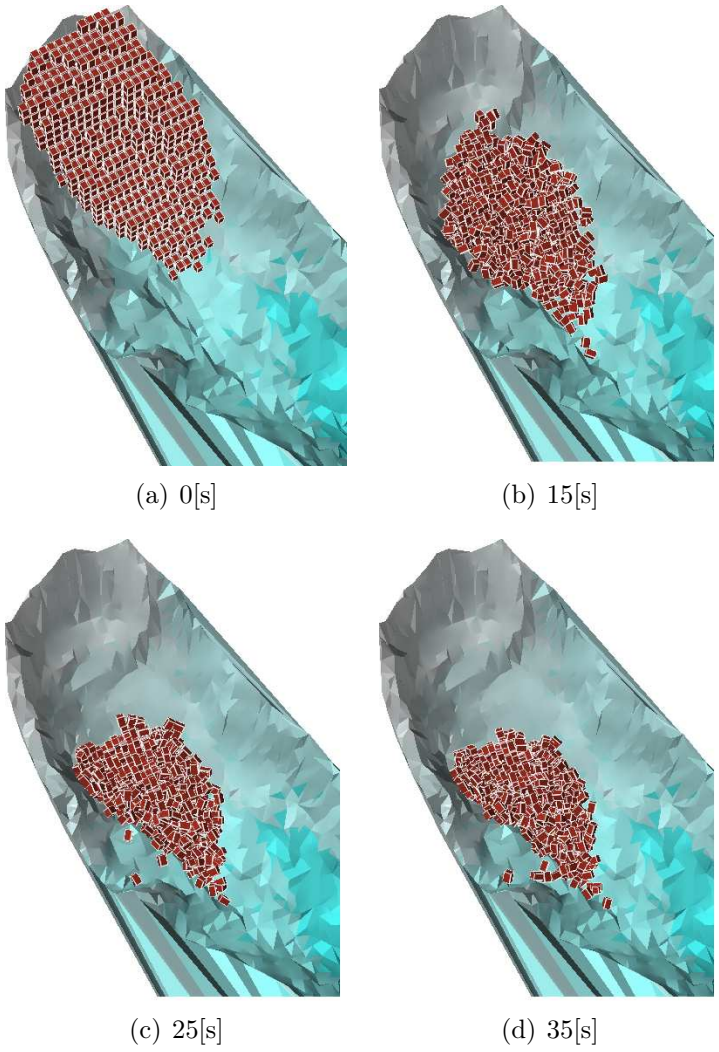
(c) 25[s] (d) 35[s]

FIGURE 6.31. An avalanche numerical simulation with 435 blocks.

At the beginning (Figure 31(a)) it is shown the initial position of the blocks, then in the right (Figure 6.31(b)), after 15[s] the system presents a considerable deformation. In Figure 6.31(c) it is observed the same deformation and collision between the blocks, in addition of a concentration taking the form as a decline curvature in the bottom of the relief. In the final position at 35[s] (Figure 6.31(d)),

the system is in a quiet state, the blocks take even more the form of the relief and the velocity of the system decreases to zero.

In Figure 6.32 the evolution of the numerical avalanche with 874 solids for the same time step as the other simulation is observed.



(a) 0[s] (b) 15[s]

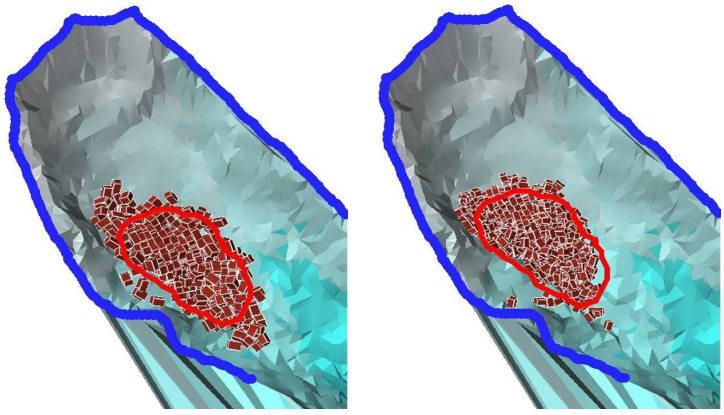
(c) 25[s] (d) 35[s]

FIGURE 6.32. An avalanche numerical simulation with 874 blocks.

The final position of the numerical avalanche is compared with the final deposit of the real one. In Figure 6.33(a) the shape of the real avalanche deposit is observed and in the Figure 6.33(b) and Figure 6.33(c) the final numerical avalanche deposit with 435 and 874 blocks respectively are shown. The first one, marked in red, has

a shape of an arrowhead where the tip is at the bottom, in the numerical case, the final deposit has a similar shape, a sort of an arrowhead pointing down. In fact the location where the deposits are situated, both are around the point where the slope of the hill changes considerably, therefore the numerical model simulates in good way the behavior of the real avalanche.

(a) Real avalanche final deposit



(b) Final deposit with 435 blocks

(c) Final deposit with 874 blocks

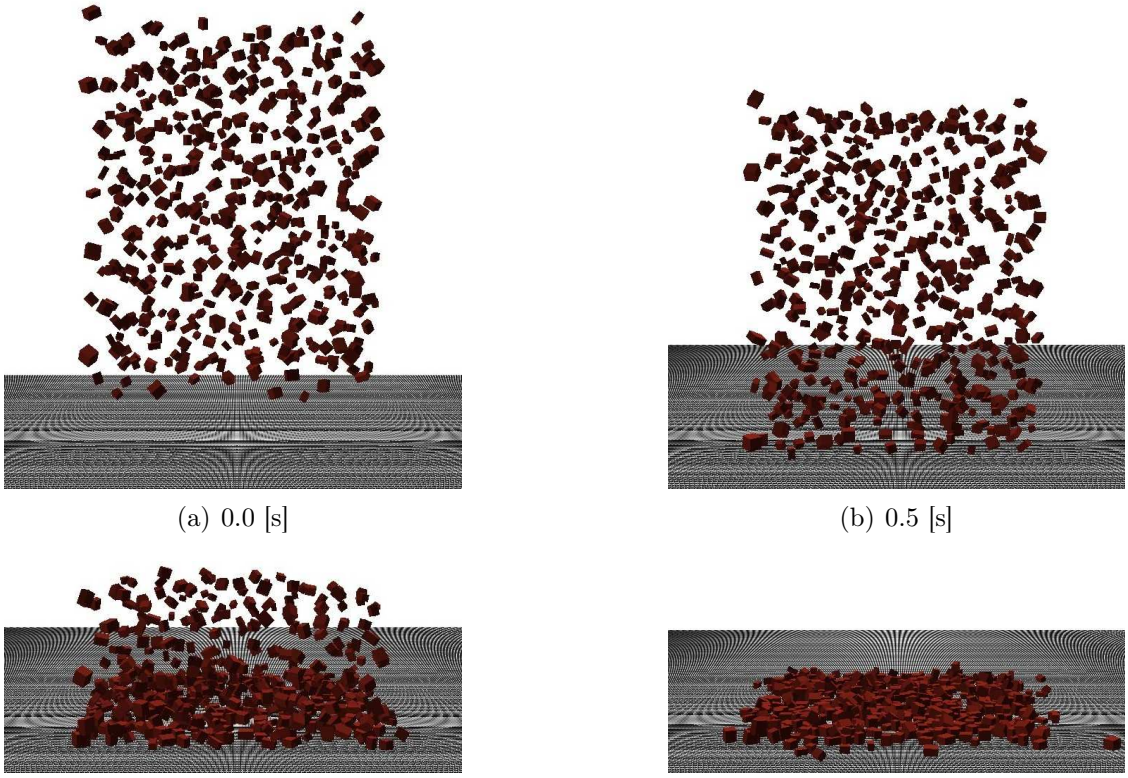
FIGURE 6.33. Comparison between real and numerical final deposits. The blue curve delimitates the surface considered in the numerical simulation, and the red curve delimitates the greater concentration of material in the final deposit.

6.3. Granular layers

A granular material is a collection of solid particles or grains, such that most of the particles are in contact with at least some of their neighboring particles. The term granular layer could be used to describe a type of granular material, in fact, a granular layer is simply a collection of solid particles which is in repose in some surface. In this case it will be considered a granular layer in a plane. The problem arises in building this layer, because the idea is to obtain a very compact granular heap.

An interesting result is that the computational time for N solids, have a linear behavior. It is important to remark that the optimization part was a black case, which it is not possible to change, but all the rest of the computations: contact detection, optimization problem building, velocity updating and geometry setting were computed. There are two experiments from an initial configuration for N solids: in the first one, all solids start in repose and fall; and in the second, all solids beging at rest in a random position but directed towards the origin, therefore after some simulation time, solids start to collide forming a heap.

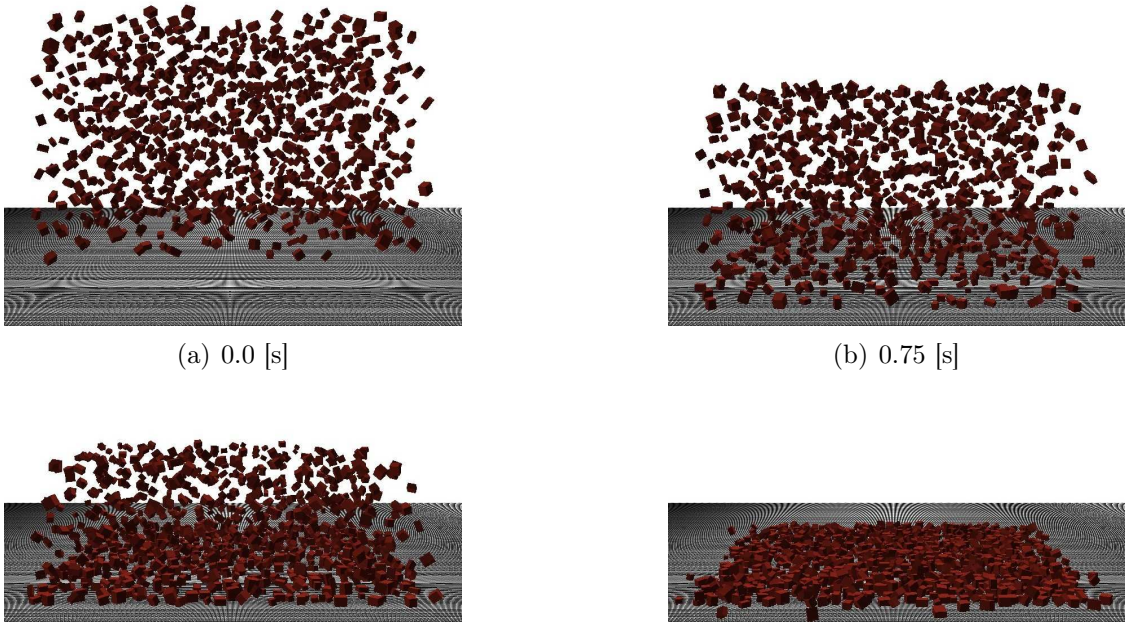
The computational time for the first type of experiment is fast, because there are not a great amount of contacts in the simulation, therefore, the more expensive part of the computation does not occur at each timestep.



(a) 0.0 [s] (b) 0.5 [s]

(c) 1.2 [s] (d) 2.0 [s]

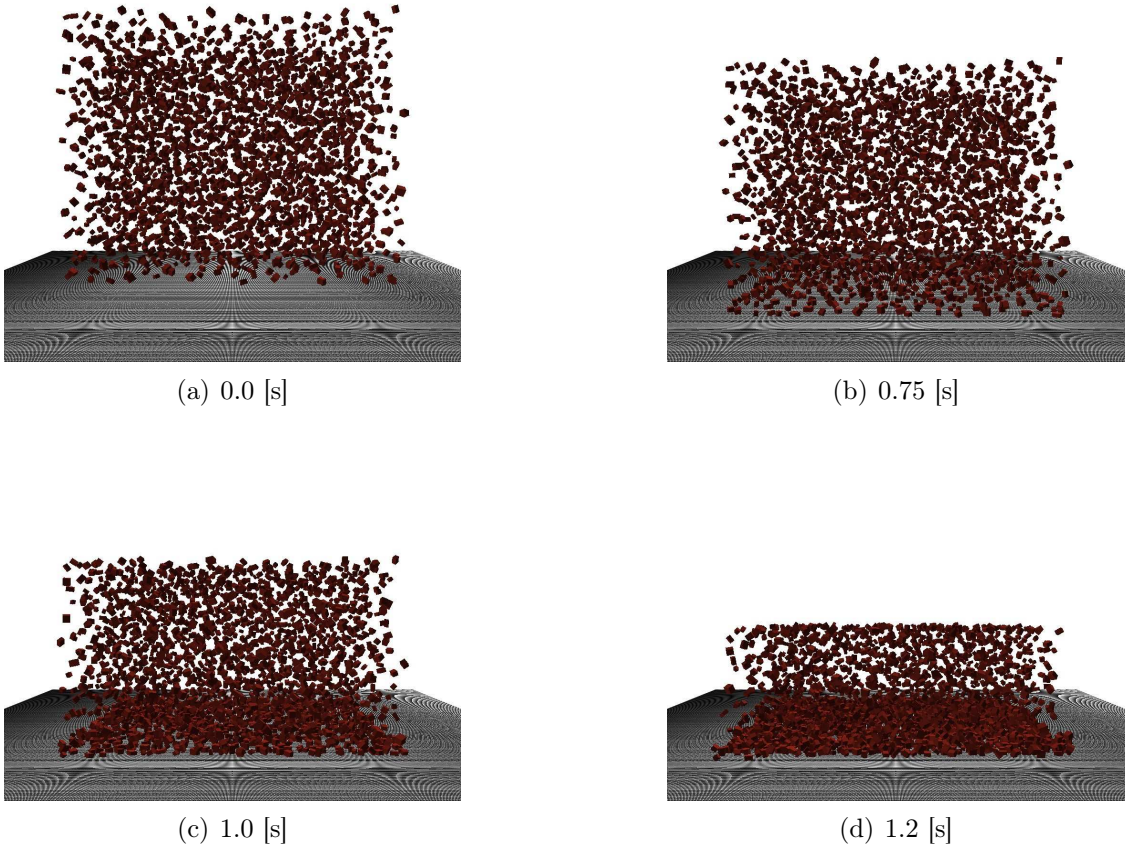
FIGURE 6.34. Evolution of 500 solids falling.



(a) 0.0 [s] (b) 0.75 [s]

(c) 1.0 [s] (d) 1.2 [s]

FIGURE 6.35. Evolution of 1000 solids falling.



(a) 0.0 [s]

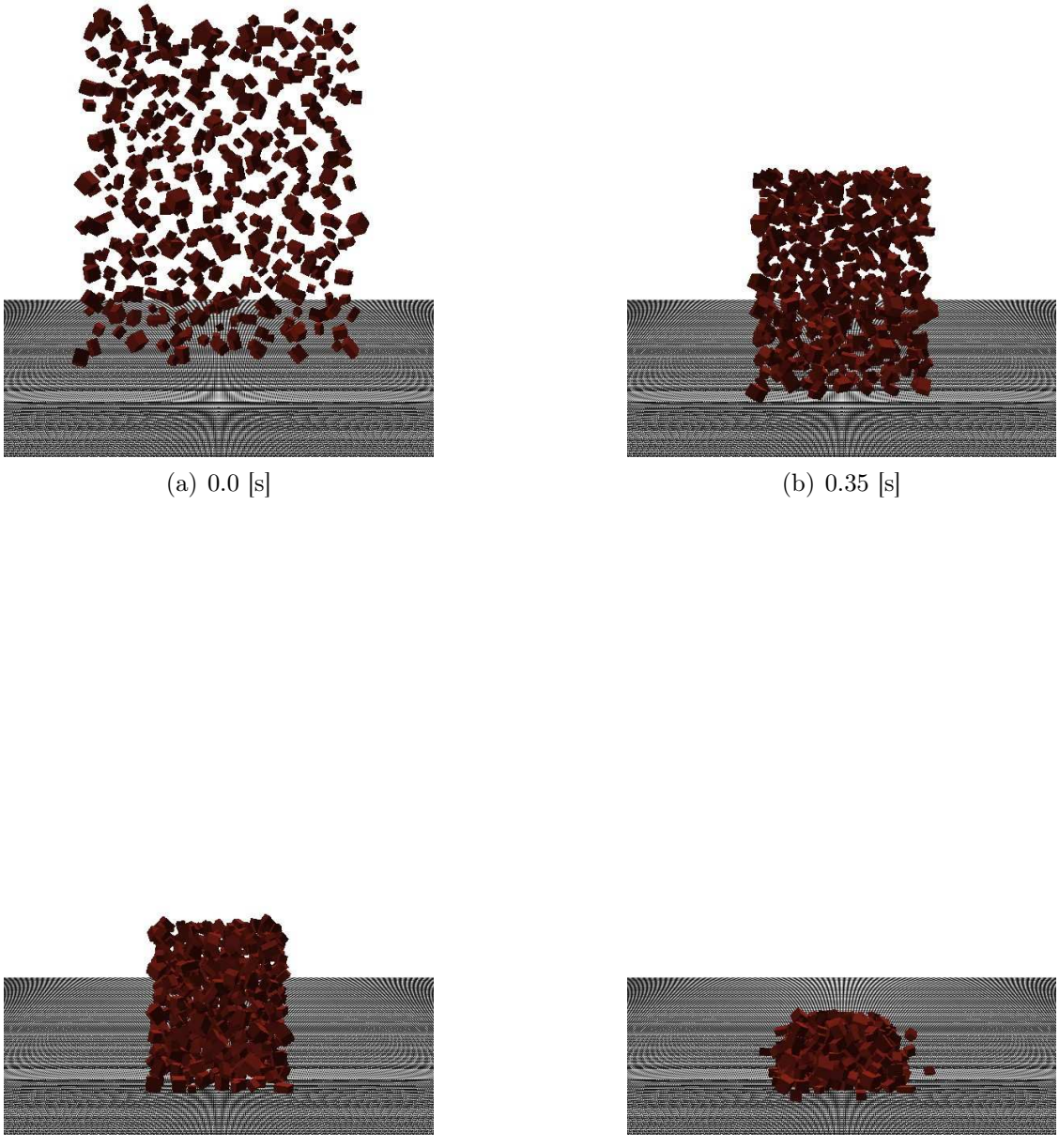
(b) 0.75 [s]

(c) 1.0 [s]

(d) 1.2 [s]

FIGURE 6.36. Evolution of 3000 solids falling.

The evolution with 500, 1000 and 3000 solids is shown in Figure 6.34, 6.35 and 6.36 respectively. For the other hand, the second kind of experiment is slower, because the amount of contacts increase in time, therefore, at each timestep is necessary to solve a constrained minimization problem. The evolution with 500, 1000 and 3000 solids is shown in Figure 6.37, 6.38 and 6.39 respectively. In this case, the initial rotational velocity for each solid is random, and the initial linear velocity is directed towards the origin.



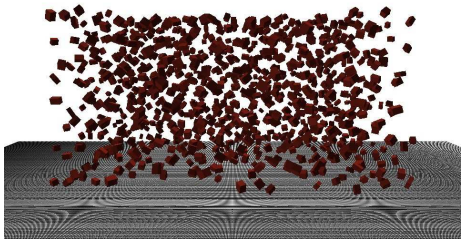
(a) 0.0 [s]

(b) 0.35 [s]

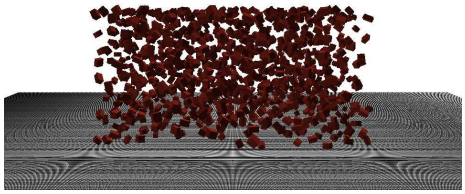
(c) 0.5 [s]

(d) 0.9 [s]

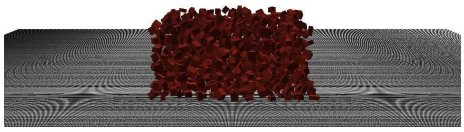
FIGURE 6.37. Evolution of 500 solids forming a heap.



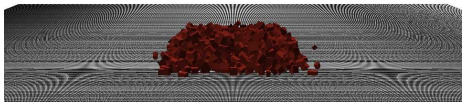
(a) 0.0 [s]



(b) 0.2 [s]

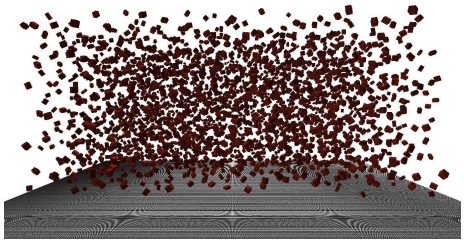


(c) 0.5 [s]

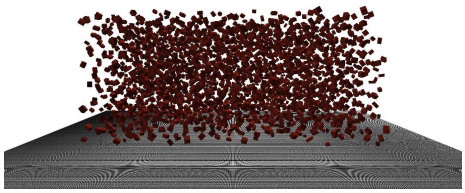


(d) 0.8 [s]

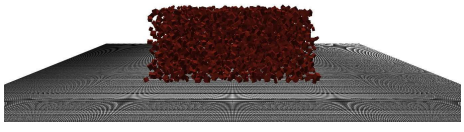
FIGURE 6.38. Evolution of 1000 solids forming a heap.



(a) 0.0 [s]



(b) 0.2 [s]



(c) 0.5 [s]

(d) 0.8 [s]

FIGURE 6.39. Evolution of 3000 solids forming a heap.

CHAPTER 7

Conclusions

The main goal of this thesis was to develop a mathematical and computational model for multiple collision of rigid bodies, by means of an extension of the A-CD^2 method. The new elements addressed in this work was the computational issues detected in the original version and the theoretical development for the rotational velocities of the solids. Theses issues comprise (a) inefficient use of computational memory, (b) convergence problem in the constrained minimization problem method and (c) inadequate mechanical behavior according to the rotational velocity of the solids. For the first issue a dynamical algorithm was presented, where the efficient use of computational memory has been achieved. Mainly in the contact detection module, a dynamical neighborhood algorithm was developed accompained by a linked list technique for contact points storing. Regarding the convergence problem, the original version uses Uzawa method to find the solution for the constrained minimization problem. In this work an interior-point method (barrier) has been used, by means of the CPLEX library. The convergence problem has been solved, because the high convexity of the function to minimize the barrier method best suits. Finally, the original approach consider, in a timestep simulation, all velocities constant. In our case, a new set of equations has been presented particularly for the rotational velocity non constant behavior.

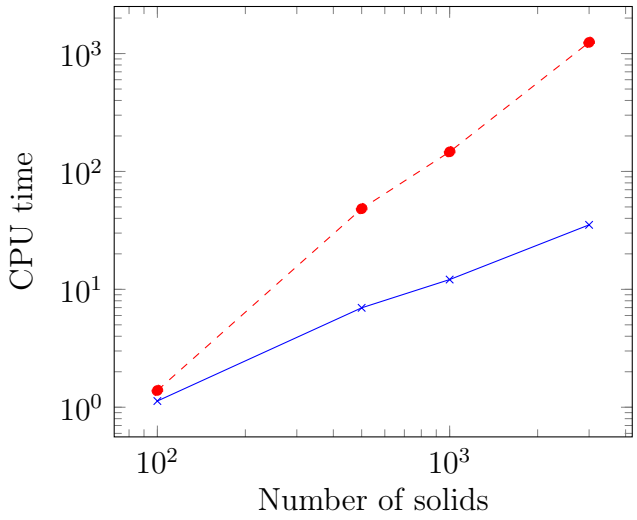


FIGURE 7.1. Computational time comparison between the original version (red dashed) and the new one (blue continuous) for different amounts of solids.

All these improvements allow us to compute until 3000 solids in a reasonable computational time. In the Figure 7.1, the cpu time is shown for one iteration, without the CPLEX computational cost. Clearly we observe an improvement between the original version and the new one, where the computational order has been reduced. An important remark is that with the original approach, no more than 1000 solids simulation can be achieved in a reasonable computational time. As final conclusion, this extension offers the possibility to simulate large scale problem, even so, we can continue to obtain more improvements.

The publications achieved in this thesis are **[49, 88, 89, 90]**.

The entire code of this work is accessible at `https://github.com/rleonvas/phd_thesis`.

Bibliography

[1] Centre d'expérimentation et de recherche (cer). http://www.normandie-centre.cerema.fr/centre-d-experimentation-et-de-recherche-cer-r340.html.

[2] Cloud compare. http://www.danielgm.net/cc/.

[3] Helimap. http://www.helimap.ch/en_index.html.

[4] Leftraru. http://usuarios.nlhpc.cl/.

[5] Litto 3D. http://www.shom.fr/les-activites/projets/modele-numerique-terre-mer/.

[6] Mathematica. Wolfram Research Inc.

[7] OpTech. http://www.optech.ca/prodaltm.htm.

[8] Visual Python. http://vpython.org/.

[9] Neighbor-list reduction: Optimization for computation of molecular Van der Waals and solvent-accessible surface areas. *Journal of Computational Chemistry*, 19:797 – 808, 1997.

[10] Guidelines for soil description, 4th edition. Technical report, Food and agriculture organization of the United Nations, 2006.

[11] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*, volume 55 of *Applied Mathematics Series*. National Bureau of Standards, Washington, D.C., 1972. Issued June 1964, Tenth Printing December 1972, with corrections.

[12] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, 1989.

[13] G. Astfalk. *Applications on Advance Architecture Computers*. SIAM, 1996.

[14] E. Azéma, F. Radjaï, and R. Peyroux. Force transmission in a packing of pentagonal particles. *Physical Review E*, 76:011301, 2007.

[15] L. Baeza. *Simulación de choque entre elementos rígidos - Memoria Ingeniero Civil Mecánico*. PhD thesis, Universidad de Concepción, 2000.

[16] E.P. Baltsavias. Airborne laser scanning: basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:199 – 214, 1999.

[17] G. Barrientos and L. Baeza. Simulation of impact between rigid elements. *International Journal of Solids and Structures*, 40:4943 – 4954, 2003.

[18] E. Bonetti and M. Fremond. Collisions and fracture. *Vietnam Journal of Mathematics*, 32:167 – 186, 2004.

[19] E. Bonetti and M. Fremond. Collisions and fracture: a 1-D theory. how to tear off a chandelier from the ceiling. *Journal of Elasticity*, 74:47 – 66, 2004.

[20] C.W. Boon, G.T. Houlsby, and S. 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.

[21] C.W. Boon, G.T. Houlsby, and S. Utili. A new contact detection algorithm for three-dimensional non-spherical particles. *Powder Technology*, 248:94 – 102, 2013.

[22] I. Bratberg, F. Radjaï, and A. Hansen. Dynamic rearrangements and packing regimes in randomly deposited two-dimensional granular beds. *Physical Review E*, 66:031303, 2002.

[23] C.E. Brennen. *Fundamentals of Multiphase Flow*. Cambridge University, 2005.

[24] F.S. Buezas, M.B. Rosales, and C.P. Filipich. Collisions between two nonlinear deformable bodies stated within continuum mechanics. *International Journal of Mechanical Sciences*, 52:777 – 783, 2010.

[25] G. Capriz, P. Giovine, and P. Mariano. *Mathematical Models of Granular Matter*. Springer, 2008.

[26] D. Meyer Carl. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[27] F. Caselli and M. Fremond. Collision of three balls on a plane. *Computational Mechanic*, 43:743 – 754, 2009.

[28] G. Chambon, R. Bouvarel, D. Laigle, and M. Naaim. Numerical simulations of granular free-surface flows using smoothed particle hydrodynamics. *Journal of Non-Newtonian Fluid Mechanics*, 166:698 – 712, 2011.

[29] J. Chang, W. Wang, and M. Kim. Efficient collision detection using a dual bounding volume hierarchy. *Lecture Notes in Computer Science*, 4975:143 – 154, 2008.

[30] J. Chang, W. Wang, and M. Kim. Efficient collision detection using a dual obb-sphere bounding volume hierarchy. *Computer-Aided Design*, 42:50 – 57, 2010.

[31] G. Chen, L. Zheng, Y. Zhang, and J. Wu. Numerical simulation in rockfall analysis: A close comparison of 2-d and 3-d dda. *Rock Mechanics and Rock Engineering*, 46:527 – 541, 2013.

[32] A.A. Chialvo and P.G. Debenedetti. On the use of the verlet neighbor list in molecular dynamics. *Computer Physics Communications*, 60:215 – 224, 1990.

[33] A.A. Chialvo and P.G. Debenedetti. On the performance of an automated verlet neighbor list algorithm for large systems on a vector processor. *Computer Physics Communications*, 64:15 – 18, 1991.

[34] J. Choi, S. Rhim, and J. Choi. A general purpose contact algorithm using a compliance contact force model for rigid and flexible bodies of complex geometry. *International Journal of Non-Linear Mechanics*, 53:13 – 23, 2013.

[35] C. Cholet. Collisions d'un point et d'un plane. *Problémes mathématiques de la mécanique*, pages 455 – 458, 1999.

[36] P. Ciarlet. *Introduction to Numerical Linear Algebra and Optimisation*. Cambridge, 1989.

[37] P.A. Cundall. A computer model for simulating progressive, large-scale movements in blocky rock systems. *Proceedings of the Symposium of the International Society of Rock Mechanics*, 1:132 – 150, 1971.

[38] P.A. Cundall. A computer model for rock-mass behavior using interactive graphics for the input and output of geometrical data. Technical report, Missouri River Division of US Army Corps of Engineers, 1974.

[39] P.A. Cundall. Ball-a program to model granular media using the distinct element method. Technical report, Advanced Technology Group, Dames & Moore, London, 1978.

[40] P.A. Cundall. Formulation of a three-dimensional distinct element model - Part I. a scheme to detect and represent contacts in a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics*, 25:107 – 116, 1988.

[41] P.A. Cundall and R. Hart. Numerical modelling of discontinua. *Engineering Computations*, 9:101 – 113, 1989.

[42] P.A. Cundall and O.D.L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29:47 – 65, 1976.

[43] G.B. Dantzig and M.N. Thapa. *Linear Programming 2: Theory and Extensions*. Springer, 2003.

[44] A. de Ryck, R. Condotta, and J.A. Dodds. Shape of a cohesive granular heap. *Powder Technology*, 157:72 – 78, 2005.

[45] E. Dimnet. Evolution of system of multiple solids. International conference on non-smooth / non-convex mechanics with applications in the engineering, 2002.

[46] E. Dimnet. *Mouvement et collisions de solides rigides ou déformables*. PhD thesis, ENPC, 2002.

[47] E. Dimnet. Collisions of rigid bodies, deformable bodies and fluids. 2nd MIT Conference on Computational Fluid and Solid Mechanics, 2003.

[48] E. Dimnet. Theory and numerical simulation of instantaneous collisions of solids. *International Journal of Solids and Structures*, 43:6100–6114, 2006.

[49] E. Dimnet, E. Haza-Rozier, G. Vinceslas, R. León, and G. Hernández. Experimental and numerical study of a shock-absorbing structure. *Acta Mechanica*, 224:3037 – 3055, 2013.

[50] S. Douady, B. Andreotti, A. Daerr, and P. Cladé. From a grain to avalanches: on the physics of granular surface flows. *Comptes Rendus Physique*, 3:177 – 186, 2002.

[51] T.G. Drake and O.R. Walton. Comparison of experimental and simulated grain flows. *Journal of Applied Mechanics*, 62:131 – 135, 1995.

[52] V. Eijkhout. Lapack working note 50 distributed sparse data structures for linear algebra operations. Technical report, Department of Computer Science, University of Tennessee, Knoxville, 1992.

[53] C. Ericson. *Real-Time Collision Detection*. Elsevier, 2005.

[54] Q. Fletcher and L.R. Hermann. Elastic foundation representation of continuum. *Journal of Engineering Mechanics*, pages 95–100, 1971.

[55] M. Fremond. Rigid bodies collisions. *Physics Letters*, 204:33–41, 1995.

[56] M. Fremond. Collision of a wedge with a plane. *Computational and Applied Mathematics*, 19:127 – 136, 2000.

[57] M. Fremond. Collisions and fractures of solids. *Tendencias em Matemática Aplicada e Computacional*, 3:1 – 4, 2002.

[58] M. Fremond. *Non-Smooth Thermomecanics*. Springer, 2002.

[59] M. Fremond, R. Gormaz, and J. San Martín. Collision of a solid with an incompresible fluid. *Theoretical and Computational Fluid Dynamics*, 16:405 – 420, 2003.

[60] M. Fremond and F. Maceri. *Novel Approaches in Civil Engineering*. Lecture Notes in Applied and Computational Mechanics - Springer, 2003.

[61] M. Fremond and F. Maceri. *Mechanical Modelling and Computational Issues in Civil Engineering*. Lecture Notes in Applied and Computational Mechanics - Springer, 2005.

[62] X. García and E. Medina. Acoustic response of cemented granular sedimentary rocks: Molecular dynamics modeling. *Physical Review E*, 75:061308, 2007.

[63] F.A. Gilabert, J.N. Roux, and A. Castellanos. Computer simulation of model cohesive powders: Influence of assembling procedure and contact laws on low consolidation states. *Physical Review E*, 75:011303, 2007.

[64] E. Glikman, I. Kelson, N.V. Doan, and H. Tietze. An optimized algorithm for molecular dynamics simulation of large-scale systems. *Journal of Computational Physics*, 124:85 – 92, 1996.

[65] H. Goldstein. *Classical Mechanics*. Addison-Wesley, 3rd edition, 1980.

[66] G.S. Grest, B. Dünweg, and K. Kremer. Vectorized link cell Fortran code for molecular dynamics simulations for a large number of particles. *Computer Physics Communications*, 55:269 – 285, 1989.

[67] J.E. Guilkey and J.A. Weiss. Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method. *International journal for numerical methods in engineering*, 57:1323 – 1338, 2003.

[68] P.K. Haff. Grain flow as a fluid-mechanical phenomenon. *Journal of Fluid Mechanics*, 134:401–430, 1983.

[69] P.R. Halmos. *Measure Theory*. Springer-Verlag, 1974.

[70] R.E. Harris, P.A. Liever, E.A. Luke, and J.G. Dudley. Multiple-body collision algorithms for computational simulation of high-speed air-delivered systems. *Communications in Computational Physics*, 17:564 – 593, 2015.

[71] R. Hart, P.A. Cundall, and J. Lemos. Formulation of a three-dimensional distinct element model - Part II. mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics*, 3:117 – 125, 1988.

[72] A. Heymann, S. Lambert, P. Gotteland, M. Collombet, and M. Douaillat. Expérimentations grandeur réelle sur merlons de protection contre les chutes de blocs rocheux. Journées Nationales de Géotechnique et de Géologie, 2010.

[73] J.B. Hiriart-Urruty and C. Lemarechal. *Fundamentals of Convex Analysis*. Springer, 2004.

[74] G.T. Houlsby. Potential particles: a method for modelling non - circular particles in DEM. *Computers and Geotechniques*, 36:953 – 959, 2009.

[75] P. Hubbard. *Collision Detection for Interactive Graphics Applications*. PhD thesis, Department of Computer Science, Brown University, 1995.

[76] D. Inman. *Engineering Vibration*. Prentice Hall, 2014.

[77] T. Iwai, C.W. Hong, and P. Greil. Fast particle pair detection algorithms for particle simulations. *International Journal of Modern Physics C*, 10:823 – 837, 1999.

[78] K. Iwashita and M. Oda. Rolling resistance at contacts in simulation of shear band development by dem. *Journal of engineering mechanics*, 124:285 – 292, 1998.

[79] M. Jean. Frictional contact in collections of rigid or deformable bodies: numerical simulation of geomaterial motions. *Studies in Applied Mechanics*, 42:463 – 486, 1995.

[80] M. Jean. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177:235 – 257, 1999.

[81] M. Jean and J.J. Moreau. Unilaterality and dry friction in the dynamics of rigid body collections. *Proceedings of Contact Mechanics International Symposium*, 1992.

[82] M. Jean and E. Pratt. A system of rigid bodies with dry friction. *International Journal of Engineering Science*, 23:497 – 513, 1985.

[83] J-F. Jerier, D. Imbault, F.V. Donze, and P. Doremus. A geometry algorithm based on tetrahedral meshes to generate dense polydisperse sphere packing. *Granular Matter*, 1:43–52, 2009.

[84] K. Kesava and P. Nott. *An Introduction to Granular Flow*. Cambridge University Press, 2008.

[85] Y. Kishino. Disc model of analysis of granular media. *in Micromechanics of Granular Materials*, 1988.

[86] M. Kodam, J. Curtis, Bruno Hancock, and C. Wassgren. Discrete element method modeling of bi-convex pharmaceutical tablets: Contact detection algorithms and validation. *Chemical Engineering Science*, 69:587 – 601, 2012.

[87] S. Lambert, A. Heymann, P. Gotteland, M. Douaillat, E. Haza-Rozier, and G. Vinceslas. Comparaison expérimentale semi - vraie grandeur du comportement de trios structures pare - blocs. Journées Nationales de Géotechnique et de Géologie, 2010.

[88] R. León, E. Dimnet, and G. Hernández. Numerical computation for a granular flow. X Jornadas de Mecánica Computacional, 2011.

[89] R. León, E. Dimnet, and G. Hernández. Numerical simulation for a granular flow. XIX Congreso sobre Métodos Numéricos y sus aplicaciones, 2011.

[90] R. León, L. Salinas, and C. Torres. Torque free rigid motion: an elliptic function primer. LXXXIV Encuentro Anual - Sociedad de Matemática de Chile, 2015.

[91] J.S. Leszczynski and M. Ciesielski. Effective algorithm for detection of a collision between spherical particles. *in Computational Science - ICCS*, 3037:348 – 355, 2004.

[92] C.F. Li, Y.T. Feng, and D.R.J. Owen. Smb: Collision detection based on temporal coherence. *Computer Methods in Applied Mechanics and Engineering*, 195:2252 – 2269, 2006.

[93] S. Mack, P. Langston, C. Webb, and T. York. Experimental validation of polyhedral discrete element model. *Powder Technology*, 214:431 – 442, 2011.

[94] D.R. Mason. Faster neighbour list generation using a novel lattice vector representation. *Computer Physics Communications*, 170:31 – 41, 2005.

[95] M.L. Matin, P.J. Daivis, and B.D. Todd. Cell neighbor list method for planar elongational flow: rheology of a diatomic fluid. *Computer Physics Communications*, 151:35 – 46, 2003.

[96] W. Mattson and B.M. Rice. Near-neighbor calculations using a modified cell-linked list method. *Computer Physics Communication*, 119:135 – 148, 1999.

[97] H.G. Matuttis, S. Luding, and H.J. Herrmann. Discrete element simulations of dense packings and heaps made of spherical and non-spherical particles.

[98] J.C. Maureira, C. Baeza, and T. Perez Acle. Levque cluster user manual. Technical report, NLHPC, 2011.

[99] H. McKean and V. Moll. *Elliptic Curves: Function Theory, Geometry, Arithmetic*. Cambridge University Press, Cambridge CB2 2RU, UK, 1999. First published 1997.

[100] S. McNamara and H. Herrmann. Measurement of indeterminacy in packings of perfectly rigid disks. *Physical Review E*, 70:061303, 2004.

[101] C.J. Meyer and D.A. Deglon. Particle collision modeling – A review.

[102] C. De Michele. Simulating hard rigid bodies. *Journal of Computational Physics*, 229:3276 – 3294, 2010.

[103] H. Mio, A. Shimosaka, Y. Shirakawa, and J. Hidaka. Cell optimization for fast contact detection in the discrete element method algorithm.

[104] H. Mio, A. Shimosaka, Y. Shirakawa, and J. Hidaka. Optimum cell condition for contact detection having a large particle size ratio in the discrete element method.

[105] H. Mio, A. Shimosaka, Y. Shirakawa, and J. Hidaka. Optimum cell size for contact detection in the algorithm of the discrete element method.

[106] M. Moore and J. Wilhelms. Collision detection and response for computer animation.

[107] J.J. Moreau. Fonctionnelles convexes. Séminaire sur les équations aux dérivées partielles, 1966.

[108] J.J. Moreau. Evolution problem associated with a moving convex set in a hilbert space. *Journal of Differential Equations*, 26:347 – 374, 1977.

[109] J.J. Moreau. Bounded variation in time. *In Topics in Nonsmooth Mechanics*, 1988.

[110] J.J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. *In Nonsmooth Mechanics and Applications*, 1988.

[111] J.J. Moreau. New computation methods in granular dynamics. *In Powders & Grains*, 1993.

[112] J.J. Moreau. Some numerical methods in multibody dynamics: application to granular materials. *European Journal of Mechanics - A/Solids*, 13:93 – 114, 1994.

[113] J.J. Moreau, P.D. Panagiotopoulos, and G. Strang. *Topics in Nonsmooth Mechanics*. Birkhäuser Verlag, 1988.

[114] A. Munjiza and K.R.F. Andrews. NBS contact detection algorithm for bodies of similar size. *International journal for numerical methods in engineering*, 43:131 – 149, 1998.

[115] A. Munjiza, E. Rougier, and N.W.M. John. MR linear contact detection algorithm. *International journal for numerical methods in engineering*, 66:46 – 71, 2006.

[116] R.M. Nedderman, U. Tüzün, S.B. Savage, and G.T. Houlsby. The flow of granular materials: Discharge rates from hoppers. *Chemical Engineering Science*, 37:1597 – 1609, 1982.

[117] S. Nezamabadi, F. Radjaï, J. Averseng, and J. Delenne. Implicit frictional-contact model for soft particle systems. *Journal of the Mechanics and Physics of Solids*, 83:72 – 87, 2015.

[118] E.G. Nezami, Y.M.A. Hashash, D. Zhao, and J. Ghaboussi. A fast contact detection algorithm for 3-d discrete element method. *Computers and Geotechnics*, 31:575 – 587, 2004.

[119] E.G. Nezami, E. Rougier, and N.W.M. John. Mr linear contact detection algorithm. *International journal for numerical methods in engineering*, 66:46 – 71, 2006.

[120] C. Nouguier-Lehonn, B. Cambou, and E. Vincens. Influence of particle shape and angularity on the behaviour of granular materials: a numerical analysis. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27:1207 – 1226, 2003.

[121] J. Novosad and K. Surapati. Flow of granular materials: Determination and interpretation of flow patterns. *Powder Technology*, 2:82 – 86, 1968.

[122] S. Oliveira and D.E. Stewart. Physically accurate granular flow simulation. *International Conference on Computational Science*, 9:286 – 291, 2012.

[123] R. O'Connor and J.R. Williams. A three dimensional geometric representation scheme for contact detection in discrete element simulation. International Journal Computer Aided Engineering & Software Engineering Computation, 1996.

[124] P. Pecol, P. Argoul, S. Dal Pont, and S. Erlicher. Modelling crowd - structure interaction. *Mécanique & Industries*, 11:495 – 504, 2010.

[125] P. Pecol, P. Argoul, S. Dal Pont, and S. Erlicher. Smooth/non-smooth contact modeling of human crowds movement: numerical aspects and application to emergency evacuations. *Annals of Solid and Structural Mechanics*, 2:69 – 85, 2011.

[126] P. Pecol, P. Argoul, S. Dal Pont, and S. Erlicher. A new crowd movement modeling for pedestrians who hold hand. XVIIIth Symposium Vibrations, Chocs et Bruit & ASTELAB, 2012.

[127] P. Pecol, P. Argoul, S. Dal Pont, and S. Erlicher. The non-smooth view for contact dynamics by michel fremond extended to the modelling of crowd movements. *Discrete and continuous dynamical systems series*, 6:547 – 565, 2013.

[128] E. Perkins and J.R. Williams. A fast contact detection algorithm insensitive to object sizes. *Engineering Computations*, 18:48 – 61, 2001.

[129] J. Peypouquet. *Convex Optimization in Normed Spaces: Theory, Methods and Examples*. Springer, 2014.

[130] F. Pfeiffer and Ch. Glocker. *Multibody dynamics with unilateral contacts*. Wiley Series, 1996.

[131] F.G. Pfeiffer. Theme issue: Non-smooth mechanics. *Philosophical Transaction of the Royal Society*, 359:2307–2630, 2001.

[132] S. Dal Pont and E. Dimnet. A theory for multiple collisions of rigid solids and numerical simulation of granular flow. *International Journal of Solids and Structures*, 43:6100 – 6114, 2006.

[133] S. Dal Pont and E. Dimnet. Theoretical approach and numerical simulation of instantaneous collisions in granular media using the $A - CD^2$ method. *Communications in Applied Mathematics and Computational Science*, 3:1 – 24, 2008.

[134] T. Pöschel and V. Buchholtz. Molecular dynamics of arbitrarily shaped granular particles. *Journal of Physics I*, 5:1431 – 1455, 1995.

[135] L. Qiu, Y. Yu, and R. Fedkiw. On thin gaps between rigid bodies two-way coupled to incompressible flow. *Journal of Computational Physics*, 292:1 – 29, 2015.

[136] J.C. Quezada, P. Breul, G. Saussine, and F. Radjai. Stability, deformation, and variability of granular fills composed of polyhedral particles. *Physical review E*, 86:031308, 2012.

[137] J.C. Quezada, P. Breul, G. Saussine, and F. Radjai. Penetration test in coarse granular material using contact dynamics method. *Computers and Geotechnics*, 55:248 – 253, 2014.

[138] F. Radjaï, M. Jean, J.J. Moreau, and S. Roux. Force distributions in dense two-dimensional granular systems. *Physical Review Letters*, 77:274 – 277, 1996.

[139] F. Radjaï and V. Richefeu. Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41:715 – 728, 2009.

[140] F. Radjaï and S. Roux. Turbulentlike fluctuations in quasistatic flow of granular media. *Physical Review Letters*, 89:064302, 2002.

[141] F. Radjaï, D.E. Wolf, M. Jean, and J.J. Moreau. Force distributions in dense two-dimensional granular systems. *Physical Review Letters*, 77:274 – 277, 1996.

[142] M. Renouf, F. Dubois, and P. Alart. A parallel version of the non smooth contact dynamics algorithm applied to the simulation of granular media. *Journal of Computational and Applied Mathematics*, 168:375 – 382, 2004.

[143] V. Richefeu, F. Radjaï, and M.S. El Youssouf. Stress transmission in wet granular materials. *European Physical Journal E*, 21:359 – 369, 2006.

[144] A. Ries and D.E. Wolf. Shear zones in granular media: Three-dimensional contact dynamics simulation. *Physical Review E*, 76:051301, 2007.

[145] E. Haza Rozier. Project anr rempare - station d'impact pendulaire du cer - merlons pare - blocs. Technical report, LCPC, 2010.

[146] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987.

[147] O. Sadovskaya and V. Sadovskii. *Mathematical Modelling in Mechanics of Granular Materials*. Springer, 2012.

[148] T.G. Sano and H. Hayakawa. Simulation of granular jets: Is granular flow really a perfect fluid? *Physical review E*, 86:041308, 2012.

[149] G. Saussine, C. Cholet, P.E. Gautier, F. Dubois, C. Bohatier, and J.J. Moreau. Modelling ballast behaviour under dynamic loading. Part 1: A 2D polygonal discrete element method approach. *Computer Methods in Applied Mechanics Engineering*, 195:2841 – 2859, 2006.

[150] S.B. Savage. Gravity flow of cohesionless granular materials in chutes and channels. *Journal of Fluid Mechanics*, 92:53 – 96, 1979.

[151] S.B. Savage. The mechanics of rapid granular flows. *Advances in applied mechanics*, 24:289 – 366, 1984.

[152] H. Shan, J. Su, F. Badiu, J. Zhu, and L. Xu. Modeling and simulation of multiple impacts of falling rigid bodies. *Mathematical and Computer Modelling*, 43:592 – 611, 2006.

[153] J. Shan and C. Toth. *Topographic laser ranging and scanning: principle and processing*. CRC Press, 2009.

[154] D.L. Shell. A high-speed sorting procedure. *Communications of the ACM*, 7:30 – 32, 1959.

[155] P. Singh, T.I. Hesla, and D.D. Joseph. Distributed lagrange multiplier method for particulate flows with collisions. *International Journal of Multiphase Flow*, 29:495 – 509, 2003.

[156] L. Staron, J.P. Vilotte, and F. Radjaï. Preavalanche instabilities in a granular pile. *Physical Review Letters*, 89:204302, 2002.

[157] D. Stoianovici and Y. Hurmuzlu. A critical study of the applicability of rigid-body collision theory. *Journal of Applied Mechanics*, 63:307 – 316, 1996.

[158] W.J. Stronge. Rigid body collisions with friction. *Proceedings: Mathematical and Physical Sciences*, 431:169 – 181, 1990.

[159] G. Sutmann and V. Stegailov. Optimization of neighbor list techniques in liquid matter simulations. *Journal of Molecular Liquids*, 125:197 – 203, 2006.

[160] A. Taboada, K. Chang, F.Radjaï, and F. Bouchette. Rheology, force transmission, and shear instabilities in frictional granular media from biaxial numerical tests using the contact dynamics method. *Journal of Geophysical Research*, 110:B09202, 2005.

[161] C. Thornton. Coefficient of restitution for collinear collisions of elasticperfectly plastic spheres. *Journal of Applied Mechanics*, 64:383 – 386, 1997.

[162] C. Thornton and K.K. Yin. Impact of elastic spheres with and without adhesion. *Powder Technology*, 65:153 – 166, 1991.

[163] T.Yagi and N. Takeuchi. An explicit dynamic method of rigid bodies-spring model. *International Journal of Computational Methods*, 12:1540014, 2015.

[164] G. Vanecek. Towards automatic grid generation usnig binary space partition trees. Technical report, Department of Computer Science, Purdue University, 1990.

[165] B. C. Vemuri, Y. Cao, and L. Chen. Fast collision detection algorithms with applications to particle flow. *Computer Graphics*, 17:121 – 134, 1998.

[166] L. Verlet. Computer "experiments" on classical fluids I, thermodynamical properties of lennard-jones molecules. *Physical Review*, 159:98–103, 1967.

[167] J.H. Walther and P. Koumoutsakos. Molecular dynamics simulation of nanodroplet evaporation. *Journal of Heat Transfer*, 123:741 – 748, 2001.

[168] D. Wang, F. Hsiao, C. Chuang, and Y. Lee. Algorithm optimization in molecular dynamics simulation. *Computer Physics Communications*, 177:551 – 559, 2007.

[169] J. Wang, S. Li, and C. Feng. A shrunken edge algorithm for contact detection between convex polyhedral blocks. *Computers and Geotechnics*, 63:315 – 330, 2015.

[170] M. Warren and J.K. Salmon. A parallel hashed oct-tree n-body algorithm. Proceedings of Supercomputing, 1993.

[171] E.T. Whittaker and G.N. Watson. *A Course of Modern Analysis*. Cambridge University Press, 4th (1927) edition, reprint 1963.

[172] J.R. Williams. Contact analysis of large numbers of interacting bodies using discrete modal methods for simulating material failure on the microscopic scale. *Engineering Computations*, 5:198 – 209, 1988.

[173] J.R. Williams and R. O'Connnor. A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries. *Engineering Computations*, 12:185 – 201, 1995.

[174] J.R. Williams, E. Perkins, and B. Cook. A contact algorithm for partitioning N arbitrary sized objects. *Engineering Computations*, 21:235 – 248, 2004.

[175] H. Yada, T. Kawaguchi, and T. Tanaka. Relation between segregation patterns and granular flow modes in conical rotating drum. *Flow Measurement and Instrumentation*, 21:207 – 211, 2010.

[176] Z. Yao, J. Wang, G. Liu, and M. Cheng. Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method. *Computer Physics Communications*, 161:27 – 35, 2004.

[177] H. Zhang, G. Chen, L. Zheng, Z. Han, Y. Zhang, Y. Wu, and S. Liu. Detection of contacts between three-dimensional polyhedral blocks for discontinuous deformation analysis. *International Journal of Rock Mechanics & Mining Sciences*, 78:57 – 73, 2015.

[178] D. Zhao, E.G. Nezami, Y.M.A. Hashash, and J. Ghaboussi. Three-dimensional discrete element simulation for granular materials. *International Journal for Computer-Aided Engineering and Software*, 23:749 – 770, 2006.

[179] Y.C. Zhou, B.D. Wright, R.Y. Yang, B.H. Xu, and A.B. Yu. Rolling friction in the dynamic simulation of sandpile formation. *Physica A*, 269:536 – 553, 1999.

[180] H.P. Zhu, Y.H. Wu, and A.B. Yu. Discrete and continuum modelling of granular flow. *China particuology*, 3:354 – 363, 2005.

[181] T.I. Zohdi. Modeling and direct simulation of near-field granular flows. *International Journal of Solids and Structures*, 42:539 – 564, 2005.

[182] R. Van Zon and J. Schofield. Numerical implementation of the exact dynamics of free rigid bodies. *Journal of Computational Physics*, 225:145–164, 2007.

APPENDIX A

Subdifferential calculus

The main goal of this appendix is to review the notions of subdifferential calculus.

A.1. Definitions

DEFINITION 1. *A function* $f : S \mapsto (-\infty, \infty]$, *defined on a convex set* **[73, 129]** $S \subset \mathbb{R}^n$, *is said to be* **convex on** $\boldsymbol{S}$ *if for every* $x_1, x_2 \in S$ *and every* $\lambda \in [0, 1]$

$$f\left(\lambda\, x_1 + (1-\lambda)\, x_2\right) \leq \lambda\, f(x_1) + (1-\lambda)\, f(x_2) \tag{A.1}$$

The same function is said to be **strictly convex** *if for every* $x_1, x_2 \in S$, $x_1 \neq x_2$, *and for every* $\lambda \in (0, 1)$

$$f\left(\lambda\, x_1 + (1-\lambda)\, x_2\right) < \lambda\, f(x_1) + (1-\lambda)\, f(x_2) \tag{A.2}$$

By a "sign-mirror treatment" the above definition can be turned into the following: a function $f : S \mapsto [-\infty, \infty)$, defined on a convex set $S \subset \mathbb{R}^n$, is said to be **[strictly] concave on** $\boldsymbol{S}$ if the function $-f$ is [strictly] convex, as defined above.

DEFINITION 2. *The* **domain** *of a function* $f : \mathbb{R}^n \mapsto (-\infty, \infty]$ *is the set dom* f, *given by*

$$dom\ f \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : f(x) < +\infty\} \tag{A.3}$$

It is clear that for every $\boldsymbol{x}_0 \in \mathbb{R}^n$ the following equivalence holds: $\boldsymbol{x}_0 \in \mathrm{dom} f$ if and only if $f(\boldsymbol{x}_0) \in \mathbb{R}$. Note also that if $f : \mathbb{R}^n \mapsto (-\infty, +\infty]$ is a convex function (Def. 1), then dom f is a convex set **[73, 129]**.

DEFINITION 3. *Given $S \subset \mathbb{R}^n$, consider the following function $\mathbb{1}_S : R^n \mapsto \{0, +\infty\}$*

$$\mathbb{1}_S \coloneqq \begin{cases} 0 & \text{if } \boldsymbol{x} \in S \\ +\infty & \text{if } \boldsymbol{x} \notin S \end{cases} \tag{A.4}$$

This function is called the **indicator function** *of the set S.*

It is easy to see that $S \subset \mathbb{R}^n$ is a convex set if and only if its indicator function $\mathbb{1}_S$ is a convex function.

DEFINITION 4. *The* **epigraph** *of a function $f : \mathbb{R}^n \mapsto (-\infty, +\infty]$ is the subset epi f of $\mathbb{R}^n \times \mathbb{R}$ defined by*

$$epi\ f \coloneqq \{(\boldsymbol{x}, y) \in \mathbb{R}^n \times \mathbb{R} : f(\boldsymbol{x}) \leq y\} \tag{A.5}$$

DEFINITION 5. *A* **subgradient** *of a function $f : \mathbb{R}^n \mapsto (-\infty, +\infty]$, $f \not\equiv +\infty$, at the point $\boldsymbol{x}_0 \in \mathbb{R}^n$ is a vector $\boldsymbol{\xi} \in \mathbb{R}^n$ such that*

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \boldsymbol{\xi}^T (\boldsymbol{x} - \boldsymbol{x}_0) \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^n \tag{A.6}$$

The set $\partial f(\boldsymbol{x}_0)$ (possibly empty) of all such subgradients is called the **subdifferential** of f at the point $\boldsymbol{x}_0$. Observe that this definition is only nontrivial if $\boldsymbol{x}_0 \in$ dom f : if $\boldsymbol{x}_0 \in \mathbb{R}^n$ dom f, then $f(\boldsymbol{x}_0) = +\infty$, so $\partial f(\boldsymbol{x}_0) = \emptyset$.

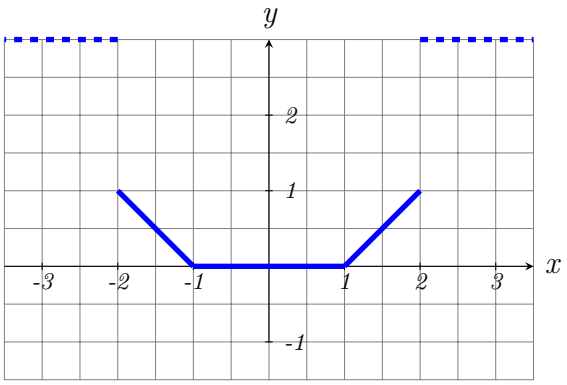
For convex functions, subgradients form a generalization of the classical notion of gradient:

PROPOSITION 1. *Let $f : \mathbb{R}^n \mapsto (-\infty, +\infty]$ be a convex function that is differentiable at the point $\boldsymbol{x}_0 \in$ int dom f. Then $\partial f(\boldsymbol{x}_0) = \{\nabla f(\boldsymbol{x}_0)\}$.*

Let's consider an example of subdifferential,

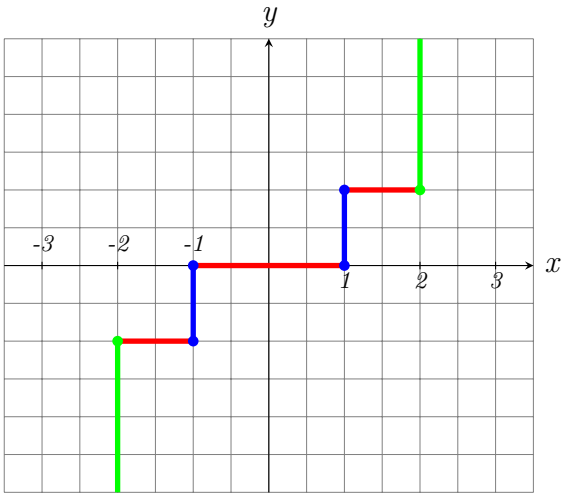
EXAMPLE 1. *Consider the function* $f : \mathbb{R} \to]-\infty, +\infty]$, *defined by*

$$f(x) = \begin{cases} 0 & \text{if } x \in [-1, +1] \\ |x-1| & \text{if } x \in [-2, -1[\cup]1, 2] \\ +\infty & \text{if } x \in]-\infty, -2[\cup]2, +\infty[\end{cases}$$



Therefore, the subdifferential of f *is given by*

$$\partial f(x) = \begin{cases} \{0\} & \text{if } x \in (-1, +1) \\ [-1, 0] & \text{if } x = -1 \\ [0, 1] & \text{if } x = 1 \\ \{-1\} & \text{if } x \in (-2, -1) \\ \{1\} & \text{if } x \in (1, 2) \\ (-\infty, -1] & \text{if } x = -2 \\ [1, +\infty] & \text{if } x = 2 \\ \textit{undefined} & \text{if } x \in (-\infty, -2) \cup (2, +\infty) \end{cases}$$



THEOREM 2 (Moreau-Rockafellar). *Let* $f, g : \mathbb{R}^n \mapsto (-\infty, +\infty]$ *be convex functions. Then for every* $\boldsymbol{x}_0 \in \mathbb{R}^n$

$$\partial f(\boldsymbol{x}_0) + \partial g(\boldsymbol{x}_0) \subset \partial(f+g)(\boldsymbol{x}_0) \tag{A.7}$$

Moreover, suppose that $\text{int dom } f \cap \text{dom } g \neq \emptyset$. *Then for every* $\boldsymbol{x}_0 \in \mathbb{R}^n$ *also*

$$\partial(f+g)(\boldsymbol{x}_0) \subset \partial f(\boldsymbol{x}_0) + \partial g(\boldsymbol{x}_0). \tag{A.8}$$

As a precursor to the Karush-Kuhn-Tucker theorem, we have now the following application of the Moreau-Rockafellar theorem.

THEOREM 3. *Let $f : \mathbb{R}^n \mapsto R$ be a convex function and let $S \subset \mathbb{R}^n$ be a nonempty convex set. Consider the optimization problem*

$$(P) \quad \inf_{\boldsymbol{x} \in S} f(\boldsymbol{x}) \tag{A.9}$$

Then $\overline{\boldsymbol{x}} \in S$ is an optimal solution of (P) if and only if there exists a subgradient $\overline{\boldsymbol{\xi}} \in \partial f(\overline{\boldsymbol{x}})$ such that

$$\overline{\boldsymbol{\xi}}^T (\boldsymbol{x} - \overline{\boldsymbol{x}}) \geq 0 \qquad \textit{for all } \boldsymbol{x} \in S \tag{A.10}$$