POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione
Dipartimento di Scienze e Tecnologie Aerospaziali
Master of Science in Space Engineering

Discrete Models of Hypervelocity Impacts of Orbiting Objects

Relatore: Prof. Michèle Lavagna
Correlatore: Dr. Fabio Ferrari

Tesi di Laurea di:
Davide Melli
Matr. 852595

Anno Accademico 2017-2018

Acknowledgments

Grazie alla mia relatrice, la professoressa Michèle Lavagna. Durante lo svolgimento di questo lavoro e durante i suoi corsi mi ha insegnato a far valere il mio lavoro e difendere quello per cui ho faticato. Grazie per aver sempre rispettato e ascoltato le mie opinioni.

Grazie infinite a Fabio, per essere stato uno dei pochi appigli quando sembrava che nulla funzionasse. Grazie per essere stato sempre disponibile per confronti e chiarimenti.

Grazie alla mia famiglia. Grazie ai miei genitori che hanno sempre supportato le mie scelte. Grazie per avermi dato tutte le risorse per affrontare qualsiasi sfida mi si presentasse. Grazie ai miei fratelli per essere stati un così buon esempio.

Grazie ai miei amici universitari, Nicolas e Damiano. Grazie soprattutto a Diego. Grazie anche per non esserti mai preso troppo sul serio.

Grazie ai miei amici di sempre. Grazie Alma, Luca, Richi, Vale. Grazie per esserci sempre stati. Grazie Mari, che durante lo svolgimento di questa tesi sei stata preziosissima.

Ovviamente, grazie anche a te, Lucia. Semplicemente, grazie di tutto.

Contents

List of Figures

List of Tables

Abstract

Since the beginning of the Space Age, the need of understanding the dynamics linked to hypervelocity impacts has become more and more important. There are more than 170 million debris smaller than 1 cm, more than 670000 from 1 to 10 cm and around 29000 larger debris in orbit around the Earth. The necessity of dealing with collisions with this kind of bodies is paramount.

On the other side, collisions between bodies in the Solar System have always happened. Redirection of asteroids that might collide with Earth has been a subject of research in the last decades. Some missions have been thought, such as AIM, ARM, AIDA, DART, or realized, such as Deep Impact.

This work presents a modeling of continuum based on discrete elements. Similar to a DEM method, a cluster of elementary particles is utilized in order to model a solid body. This modeling is applied to hypervelocity impacts on orbting bodies. The model is implemented in Project Chrono, a multiphysics simulation engine specialized in the management of multibody dynamics.

The code is utilized to recreate two scenarios, an aluminum projectile colliding with an aluminum plate and an asteroid redirection mission based on a kinetic impactor. While the first simulation still presents some criticalities, the outcomes of the second one are in agreement with state-of-the-art literature.

Sommario

Sin dall'inizio dell'Era Spaziale, il bisogno di comprendere la dinamica degli impatti ad iper-velocità è apparso sempre piu importante. In orbita intorno alla Terra ci sono più di 170 milioni di detriti più piccoli di 1 cm, più di 670000 tra gli 1 e i 10 cm e circa 29000 di detriti di dimensione maggiore. La necessità di saper gestire collisioni con questo tipo di oggetti è di fondamentale importanza. Inoltre, ci sono sempre state collisioni tra i corpi del Sistema Solare. Il riorientamento di asteroidi che potrebbero colpire il nostro pianeta è stato soggetto di ricerche negli ultimi decenni. Missioni sono state pensate, come AIM, ARM, AIDA, DART, o realizzate, come Deep Impact.

Questa tesi presenta un modello del continuo basato su elementi discreti. In modo simile al metodo DEM, un ammasso di particelle elementare è utilizzato per modellare un corpo solido. Questo modello è applicato in questa tesi a oggetti orbitanti soggetti ad impatti ad iper-velocità. Il modello è implementato in Project Chrono, un simulatore di multifisica specializzato nella gestione della dinamica multicorpo.

Il codice è utilizzato per ricreare due scenari, un proiettile d'alluminio che impatta contro una piastra di alluminio e una missione di riorientamento di asteroide basata su una sonda-proiettile. Mentre la prima simulazione presenta ancora alcune criticità, i risultati della seconda sono in accordo con lo stato dell'arte.

Chapter 1

Introduction

Understanding how to model and simulate hypervelocity impacts has become a major concern in space engineering or astrodynamics applications. Most of the actual impacts that take place in orbit happen at velocities in the order of magnitude of kilometers per seconds. A solid simulation tool to cope with this kind of problem is needed.

Space debris is a main hazard for any object orbiting Earth. Both artificial bodies and meteoroids have the possibility to hit a satellite in any moment, causing problems to the system itself and, in the worst case, the creation of a debris cloud that would increase the quantity of Space junk. A powerful tool able to simulate the effects of these impacts on artificial objects would be fundamental in order to plan and design countermeasures or collision avoidance manoeuvre.

Another great problem that is being discussed is the redirection of asteroids. Near Earth Objects (NEOs) are asteroid (NEAs) or comets (NECs) with a perihelion at a distance of less of 1.3 au. Potentially Hazardous Asteroids (PHAs) are defined on the asteroid's potential to make close approaches to Earth. All the bodies that might have distances of less than 0.05 au and that are greater than ~ 140 m are considered PHAs. Space community has begun to wonder how to decrease the risk of a PHA impacting the planet. Two main methods exist: distruction and deflection. They are based on nuclear explosive devices or kinetic impactor as direct methods. Indirect methods are about gravity tractors, ion beam sheperds, focused solar energy and others.

The present work is motivated by these observations. The first objective of this study is to propose a modeling method based on discretized elements. It is an approach that is very appropriate when it comes to simulate impacts. It allows the user to study the evolution of every ejecta that might be produced by the collision or the internal characteristics of an object. The second objective is to perform simulations about two scenarios in which hypervelocity impacts are the main feature, a debris against a thin plate and a kinetic impactor against an asteroid. The software utilized to execute all the simulations (creation of the particles, collision and contact dynamics, integration) is Project Chrono, a multi-physics engine.

1.1 Problem definition

The present work aims to define a modeling method for continuum objects such as spacecraft components or asteroids. The proposed model is then simulated and verified with existing studies. The main topics of the thesis are:

- Define a methodology to model a continuum with discretized particles. This part is highly inspired by Discrete Element Method (DEM) but it presents some peculiarities and differences due to the characteristics of the software used for the simulations.

- Define a methodology to model asteroids. This model utilizes the difinition of Rubble Pile as a starting point. The application of cohesive forces brings the Rubble Pile to something more similar to a monolithic body discretized with elementary particles. In this section the solution of a N-body problem is solved in order to create a gravitational aggregate.

- The model for continuum discretization is applied to the case of an aluminum projectile hitting an aluminum plate at hypervelocity. The simulation inputs and parameters are examined. Outcomes are discussed and criticalities highlighted.

- The scenario of a kinetic impactor hitting an asteroid is recreated. DART mission is the main reference in this section. Cohesive forces are utilized in order to get an aggregate that resists to the impact. Post-impact conditions of the asteroid, such as Δv, Δa and ΔT are displayed.

- Guidelines about how to simulate hypervelocity impacts between bodies discretized by elementary particles on Project Chrono are derived.

1.2 State of the Art and previous works

Research about hypervelocity impacts started with the beginning of the Space age in the 20th century. A lot of experiments have been done in order to define material properties in these extreme conditions, both for satellite applications and for asteroid scenarios. For example, at CISAS experiments have been made with porous targets [14] or with spacecraft materials [11]. In [2], the impact behaviour of different shielding panel typologies obtained using a polymeric foam core or using aramidic fibres as reinforcement was evaluated. In [39], the debris cloud created by nonspherical projectile are studied.

About the simulation of such impacts, most of the literature deals with Smoothed Particle Hydrodynamics (SPH). SPH has applications in many fields such as astrophysics, hydrodynamics, magnetohydrodynamics, gas explosions, and granular flows, and has also been extended to simulate bodies with material strength. It is widely applied to impact problems in computational solid mechanics due to its meshless structure. Together with Eulerian methods, it is preferred to Finite Element Methods because of the large deformations that lead to great mesh distortion. SPH are widely used both for Spacecraft impacts and Asteroid impacts. Aluminum sphere against an aluminum plate has been modeled with SPH in [21]; Brittle targets are considered in [19]; SPH has been used to study the debris cloud on spacecraft structures in [23]; an hybrid particle-finite element method has been utilized to simulate orbital debris impact on the Space Shuttle wing leading edge in [42]. Actually, a lot of SPH based works have been done in past years.

Discrete Element Methods (DEM) are relatively new in the simulation of solid bodies. Works about compacting cohesive granular system has been done, like in [16]; low velocity impacts of agglomerates has been simulated in [18]; [24] and [25] are about high velocity impacts and failure dynamics on thin brittle materials discretized by elementary spheres. When it comes to hypervelocity impacts of objects discretized by DEM, the first and (as far as the author knows) the only work is the one by Watson E. and Steinhauser M. O. [48].

Literature concerning asteroid impacts follows more or less the same pattern. A lot of works has been made using SPH models and fewer are based on DEM simulations. An hybrid N-body codes and SPH method are proposed in [8] to model impacts on rubble piles and in [9] to show satellites formation in large impacts; SPH particles are used to characterize momentum transfer in porous targets in [40] and impact erosion model for gravity-dominated planetesimals in [12]. Hybrid between N-body codes and DEM (or quasi-DEM) asteroids are more oriented to low velocity impacts, such as in [17] with spherical particles or in [38] with polihedra; a cohesion study has been made in [43] in the scenario of dense planetary rings.

A lot of research has been done in modelling and simulation of hypervelocity impacts with SPH codes. DEM methods are new in this field of research and a lot of work has to be done in order to evolve this technology.

1.3 Contribution of the present work

The present work has the ambition to outline a method to model solid objects based on discretized elements. Hypervelocity impacts are taken into account in the modeling of the systems. A simulation tool based on Project Chrono is defined and developed. Criticalities and bottlenecks are highlighted and assessed. The code is validated and applied to two cases, an aluminum projectile impacting on an aluminum plate and a DART-like mission in which a kinetic impactor deflects an asteroid.

The thesis is organized in the following way:

- Chapter 2 outlines the hypoteses made in order to develop the model and the simulation. Project Chrono is presented and its features that are relevant to the present work are highlighted and explained. Theorethical concepts about discretization by elementary elements are shown. The proposed discretization method applied to plates and asteroid is then analyzed.

- Chapter 3 presents the simulations. Model construction is made clear and the process of creating an object is shown. Inputs and parameters for the simulations are outlined. Outputs are defined and the outcomes discussed critically.

- Chapter 4 aims to make a point of all the work presented in the previous chapters. The criticalities are discussed, possible improvements are proposed and what has worked well is highlighted.

Chapter 2

Modeling

The objective of this chapter is to present the theoretical framework in which this thesis is carried out.

First of all, in section 2.1, the tool used for the simulations of this work, Project Chrono, is presented. Its functionalities, capabilities and characteristics that are useful for this work are highlighted.

Secondly, the theoretical background for continuum discretization is discussed. This is fundamental in order to justify the hypoteses and simplifications made in chapter 3. Moreover, the simulation that exploits these concept is presented and analyzed.

Finally, the concept of rubble pile asteroid is explained. How this application is linked to discretization of continuum is shown. Then, the asteroid simulation is presented and analyzed.

2.1 Project Chrono presentation

Project Chrono is a multi-physics modelling and simulation infrastructure based on a platform-independent, open-source design. The core of the software is Chrono::Engine, an object-oriented library whose C++ API can be used to perform multi-physics simulations. Other modules are available, such as Chrono::Vehicle, Chrono::FEA and so on. An important module for this work is Chrono::Parallel, a library that enables parallel computing in Chrono. Project Chrono is able to deal with rigid and flexible bodies, costraint, motors, contacts and collision detection. An overview of the architecture of Chrono is offered in figure 2.1.

Project Chrono is a very powerful tool when it comes to large-scale simulation because of its very robust collision detection. The user can define collision shapes using meshes or primitives and Chrono is able to solve for frictional contacts very efficiently.

Chrono has been thought and created for problems of robotics or civil engineering. The way in which it is utilized in this work requires some customization of the engine. What is utilized mainly of Project Chrono are its powerful collision detection algorithm, its contact handling algorithm and its module for CPU parallelization.

2.1.1 Contact dynamics and methods

Project Chrono deals with contacts in two very different ways. The first one is called SMC (smooth contact method) and works with smooth particles. This method is also known as 'penalty method' and it is the one that it is usually used in Discrete Element Method (DEM) simulations. The second one is called NSC (non-smooth contact method) and it operates with completely

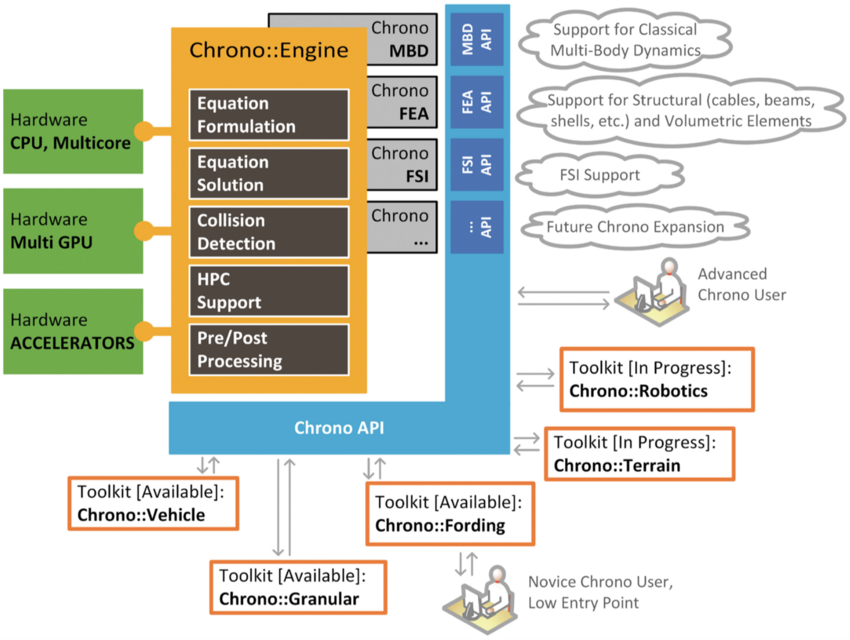


Figure 2.1: An abstraction of the Chrono architecture

rigid bodies. This is a fairly new approach but it offers many pros that makes it very interesting.

SMC contact method

Penalty method creates a fictitious spring-damper system at every contact point. The stiffness of the spring and the damping coefficient rule how the contact dynamics evolves.

Figure 2.3 shows what happens when two bodies interact with a frictional contact. Normal force $\mathbf{F}_n$ is function of the interpenetration δ_n and the contact velocity $\mathbf{v}_n$. Something very similar happens with the tangential force $\mathbf{F}_t$ that is function of the creep $\boldsymbol{\delta}_t$ and the creep velocity $\mathbf{v}_t$. The equations are:

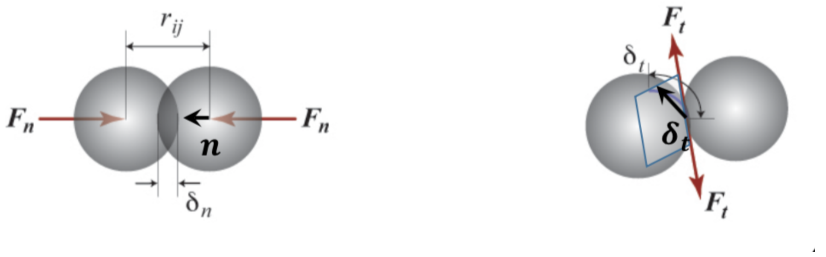


Figure 2.2: Penalty method

$$\mathbf{F}_n = f\left(\frac{\delta_n}{D_{\text{eff}}}\right)(k_n\delta_n\mathbf{n} - \gamma_n m_{\text{eff}}\mathbf{v}_n) \tag{2.1}$$

$$\mathbf{F}_t = f\left(\frac{\delta_n}{D_{\text{eff}}}\right)(k_t\boldsymbol{\delta}_t - \gamma_t m_{\text{eff}}\mathbf{v}_t), \tag{2.2}$$

where m_{eff} and D_{eff} are the effective mass and the effective diameter of curvature of the contacting bodies. Once the contact forces are computed, they are added to the forces and torques container of each body and the Newton-Euler equations of motion are integrated.

Since the SMC contact method is ruled by a spirng-damper couple, the time step selection is subjected to stability condition. The time step h must be chosen in order to have:

$$h < h_{\text{crit}} \sim \sqrt{m_{\text{min}}/k_{\text{max}}}, \tag{2.3}$$

being m_{min} and k_{max} the minimum mass and the maximum stiffness, respectively.

NSC contact method

NSC only operates with rigid bodies. This leads to fundamental costraints: bodies shall not interpenetrate and, if in contact, a friction force shall arise.

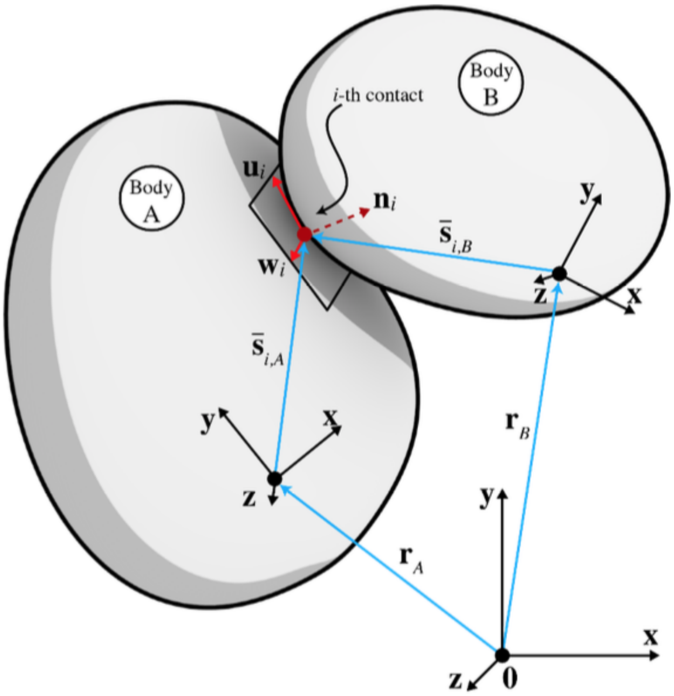


Figure 2.3: Rigid body method

NSC introduces a distance function $\Phi(\mathbf{q}_a(t), \mathbf{q}_b(t))$ such that

$$\Phi(\mathbf{q}_a(t), \mathbf{q}_b(t)) \begin{cases} > 0 & \text{if bodies are separated} \\ = 0 & \text{if bodies are in contact} \\ > 0 & \text{if bodies are interpenetrated,} \end{cases} \tag{2.4}$$

being $\mathbf{q}_a(t)$ and $\mathbf{q}_b(t)$ the generalized coordinates of the two bodies. The *i-th* contact is modeled with the following complementarity problem:

$$\gamma_{i,n} \geq 0 \ , \ \Phi_i(\mathbf{q}) \geq 0 \ , \ \Phi_i(\mathbf{q})\gamma_{i,n} = 0 \ . \tag{2.5}$$

This means that the normal force $\gamma_{i,n}$ and the distance function shall always be ≥ 0. The last equation states that if the force is greater than zero, then the distance must be zero and viceversa.

The friction is modeled with the Coulomb's model:

$$\mu_i \gamma_{i,n} \geq \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2} \tag{2.6}$$

$$\mathbf{F}_{i,t}^{\mathrm{T}} \cdot \mathbf{v}_{i,t} = -||\mathbf{F}_{i,t}|| \ ||\mathbf{v}_{i,t}|| \tag{2.7}$$

$$||\mathbf{v}_{i,t}|| \Big(\mu_i \gamma_{i,n} - \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2} \Big) = 0, \tag{2.8}$$

where $\gamma_{i,j}$ is the *i-th* contact friction force in *h-th* direction. Equation (2.6) shows that friction force is within the friction cone. Equation (2.7) states that friction force and tangential velocity at contact point are collinear and of opposite direction. Finally, equation (2.8) represents the stick-slip condition. If the velocity is greater than zero, it means that the friction force is saturated. On the contrary, if the bodies stick to each other, so the friction force is not saturated, then the tangential velocity must be zero.

This friction model is then inserted in the generalized equations of motion that are:

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q})\mathbf{v} \tag{2.9}$$

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q},\mathbf{v},t) - \mathbf{g}_{\mathbf{q}}(\mathbf{q},t) + \sum_{i \in A} \underbrace{(\gamma_{i,n}\mathbf{D}_{i,n} + \gamma_{i,u}\mathbf{D}_{i,u} + \gamma_{i,w}\mathbf{D}_{i,w})}_{i\text{-}th \text{ frictional contact force}} \tag{2.10}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q},t) \tag{2.11}$$

$$i \in A(\mathbf{q}(t)) : \begin{cases} 0 \leq \Phi_i(\mathbf{q}) \perp \gamma_{i,n} \geq 0 \\ (\gamma_{i,u},\gamma_{i,w}) = \underset{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2} \leq \mu_i \gamma_{i,n}}{\operatorname{argmin}} \quad \mathbf{v}^{\mathrm{T}} \cdot (\gamma_{i,u}\mathbf{D}_{i,u} + \gamma_{i,w}\mathbf{D}_{i,w}) \end{cases} . \tag{2.12}$$

Equation (2.9) relates the time derivative of the generalized positions and velocities through a linear transformation $\mathbf{L}$. Equation (2.10) represents the force balance with inertia forces, external forces $\mathbf{f}$, costraint forces $\mathbf{g}$ and friction forces, where $\mathbf{B}_{i,j}$ are simply projectors in *j-th* direction. Equation (2.11) imposes bilateral costraints on the body. Finally, equation (2.12) summarizes the Coulomb's friction mode. The equations of motion are then discretized and relaxed. After this process, the dynamical equations become a Cone Complementarity Problem.

NSC contact method has no stability costraints on the time step selection since the bodies are rigid and no springs come into play. The limitation comes from the fact that collision detection must work in a proper way. For example, if a sphere of radius of 1 cm travels 1 m every time step, it is very unlikely that the collision detection algorithm detects the contact with another sphere of radius of 1 cm that may cross paths. A good rule of thumb is to select a time step that make the smaller body travel a distance which is comparable with its characteristic length that may be a fraction of the radius for a sphere, a fraction of the side length for a cube and so on.

2.1.2 Collision detection

Chrono::Engine performs collision detection in an efficient way. Searching for contact point for every possible couple of bodies would lead to a problem of N^2 complexity, being N the number of bodies in the simulation. This means that the time required for the collision detection grows very fast with the number of bodies. Chrono, to solve this issue, performs collision detection in two phases.

The first one is the Broad phase. The algorithm detects nearby pairs by a close-neighbour search applied to the bounding volumes of the bodies. This first phase efficiently determines which pairs of object may collide. In this part accuracy is not a major concern but efficiency is. Methods for this phase are Dynamics axis aligned bounding box (AABB) trees, Sweep and Prune (SAP), Hierarchical grids and more.

The second part is the Narrow phase. Pairs for which the Broad phase determined a possible collision are analyzed by more refined algorithms. In this phase the exact shape geometry of the body is utilized. In this section the goal is to accurately select which pairs actually collide and completely characterize the contacts. Methods may be analytical (just between a set of primitive shapes), Separating Axis Theorem, GJK algorithm or MPR.

2.1.3 Integrators and solvers

Time integration in Chrono works differently if the contacts are smooth or non-smooth. Smooth dynamics is the one coming from classical multibody dynamics,rigid and flexible connected through joints, FEA and fluid-solid interaction.

Smooth dynamics is ruled by Differential Algebraic Equations (DAEs).

Usually Linearized Implicit Euler or HHT methods are used as integrators for this kind of problem. Then, the linear system arising from the DAEs is usually solved by MINRES, an iterative solver.

As seen in section 2.1.1, non-smooth equations of motion are formulates as Differential Variational Inequality (DVI). The time-stepper method for these problem is Linearized Implicit Euler. The solver for the Cone Complementary Problem can be chosen between the SOR, Barzilai-Borwein and APGD.

For the interested readers, a large literature is available about numerical methods in Project Chrono (see [45], [46], [47] and [31]).

2.2 Plate scenario

In this section, the main hypoteses and theoretical background for the continuum modeled through discrete particles is introduced and explained. First of all an overview of classical Descrete Element Method (DEM) is offered and then the insight of the model used in this work is presented.

2.2.1 Discrete Element Method

DEM is well-established method for modelling the dynamic behaviour of granular assemblies subjected to a variety of loading scenarios. The fundamental assumption of the method is that the material consists of separate, discrete particles. These particles may have different shapes and properties.

Particles may be subjected to many kind of forces. In macroscopic simulations, forces can be friction, plasticity, gravity, cohesion and so on. At molecular level, particles can interact via Coulomb force, Pauli repulsion, van der Waals forces and many others. Interparticle interactions can be modeled with potentials, such as Lennard–Jones one.

A DEM simulation starts with N different or equal particles with a given initial position and velocity. Then the characteristics of the particles are set. After that, the forces are computed and applied to every particle. Finally, the dynamics are integrated and the new state vector of each particle is found.

Usually, DEM simulation are used to model bulk materials and granular substances. In this thesis, DEM method is utilized to create a solid material, such as an aluminum plate or an aluminum projectile. This work is focused on the simulation of solid materials subjected to hypervelocity impacts, a fairly new application of DEM method.

2.2.2 Parallelization

Discrete Element Method simulation requires the utilization of many particles; this makes the simulation computationally intensive. With the advance in computing power and the development of new more efficient numerical algorithm, millions of particles can be simulated on a single processor.

However, it is possible to parallelize the simulations in order to get a faster simulation or to increase drastically the number of particles. Two basic parallelization methods are available, the CPU and the GPU one.

The CPU parallelization is based on the distribution of the workload of the program over several cores, assuming that the processor has more than one core as in most of the modern CPUs. In order to be split into the different cores, the workload of the program must be distributed over multiple threads. Generally one core receives just one thread. The process of more than one thread linked to a single core is called multithreading or hyper-threading. The work can be divided and distributed manually using threads or tools as OpenMP can be used.

GPU parallelization exploits the highly parallelized architecture of the graphical processing unit to reach as good parallel computational capability as possible. A platform is needed in order to access and manage the GPU computations, softwares as CUDA (made by Nvidia) or OpenGL.

In this work only CPU parallelization has been utilized. OpenMP manages the parallelization of the workload into 4 different cores. The reasons why the GPU has not been exploited is because Project Chrono does not support collision detection and contact dynamics on the graphical processing unit yet. On the contrary, CPU parallelization for granular dynamics in Project Chrono is well established and widely utilized. Chrono::Parallel (the name of the module that enables parallel computation in Chrono) uses custom data structures and tailored algorithms (such as collision detection) while fully exploiting the Chrono::Engine, the core module, modeling capabilities.

2.2.3 Continuum discretization

As previously stated in section 2.2.1, one of the objectives of this work is to model a solid body using discrete elements as "building material". The process begins with the characterization of the volume of the body that has to be discretized. Then, the shape, the dimension and the physical characteristics of the elementary particle shall be selected. After that, the particles must be created and placed accordingly to a desired distribution (regular grid, hexagonal close packing, ...). Finally, the interparticles interactions and costraints are modeled and applied to the particles. In this way, the solid object is discretized completely.

As in can be seen in figure 2.4, after that the volume to be discretized is selected, the particles must be designed. In this case, a cube of side length L is being discretized by elementary spheres. The dimensions of the spheres

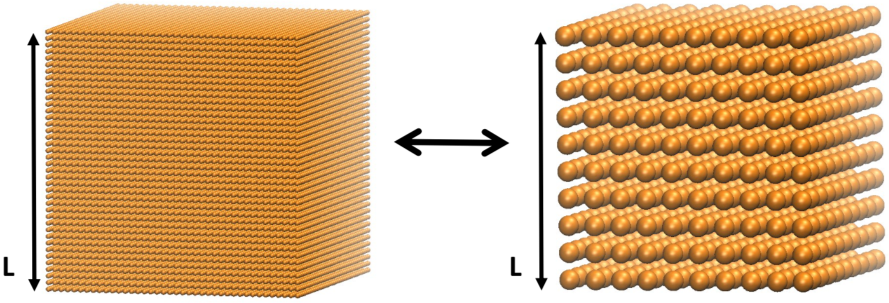


Figure 2.4: Volume discretization

can vary depending on the accuracy level needed in the simulation. One key rule is that the total mass must be conserved. So :

$$M_{\text{cube}} = N_{\text{spheres}} m_i = N_{\text{spheres}} (\frac{4}{3} \pi r^3_{\text{spheres}}) \rho, \tag{2.13}$$

where ρ is the density and r_{spheres} is the radius of the spheres. Density is not only function of the dimensions of the spheres but also of the geometrical distribution of particles. The same volume sampled with grid distribution requires less particles than with a hexagonal close packing distribution. This means that the particles positioned with the regular lattice must be heavier to satisfy mass conservation.

Classical DEM particles have many parameters that rule their interparticle interactions. In the present work the main properties and parameters are smoothness of the particles (or contact rigidity), cohesive forces and friction. The latter is treated with Coulomb's friction model, as already explained in section 2.1.1.

Cohesive actions are modeled as constant forces applied at contact points. This is important because it leads to the shape selection for the elementary particles. Since it is a per-point value, cohesion arise whenever the collision detection algorithm finds a contact point. For example, two flat surface might have from 3 to N contact points. This would render the management of the cohesion within the material very hard. With spherical particles this problem does not exist because there is always just one contact point per sphere pair. This is the reason why in this work only spherical particles are utilized.

Smoothness of particles, or contact rigidity, has already been discussed in section 2.1.1. Particles might be infinitely rigid (with NSC contact method) or they might have user-defined stiffnesses (with SMC contact method).

Classical DEM methods models contact interactions considering both repulsive and attractive actions in the same moment. An example is the Lennard-Jones potential. It is usually utilized in molecular application, but it can be modified in order to simulate macroscopic particles (see [48]). It has a repulsive term that is function of r^{-12}, with r being the distance between the particles. It models short ranges repulsion such as Pauli's. The attractive term is function of r^{-6} and it models long range attraction (such as van der Waals force).

2.3 Asteroid scenario

In this section an overview of the theoretical concepts about rubble pile asteroids and planetoid modeling is described.

A rubble pile body is not a monolithic object but it consists in a multitude of smaller bodies that have coalesced thanks to self-gravitation. It is though that they might have very low cohesion forces between the boulders and that the main force that keep the fragments together is gravity. However, some of this objects, especially the smallest ones, present some degrees of cohesive forces.

The idea that a large percentage of bodies with dimension ranging from $\sim$ 100 m to $\sim$ 100 km might be gravitational aggregates is gaining great acceptance. Evidences coming from observations, experiments and simualations support this theory. The most important clues that confirm the hypotesis are:

- slow spin rate
- low bulk density
- tidal break-ups
- unusual shapes and binaries.

This is the beginning point for the modeling of asteroids in this work. Elementary particles are created and left free to self-attract and coalesce. However, as proven in chapter 3.2, a total absence of cohesion would mean a certain total breack-up in case of an hypervelocity impact. A cohesive forse is then added to make the aggregate more resistant.

So, in this thesis one of the asteroid models is an hybrid between a rubble pile and a monolithic body discretized by elementary particles. The parameter that rules the hybridization is interparticle cohesion. But, since in the present work the objective is to simulate an asteroid redirection and after having shown that gravitational forces are not strong enough to create a resistant aggregate, the cohesive forces are set to high values and they become the main action that hold together the asteroid.

Anyway, also the monolithic asteroids are modeled in the present work. This model takes into account an unbreakable central body surrounded by smaller boulders that simulate the layer of loose material that might cover the asteroid.

In this way, the majority of configurations of asteroids and planetoids are covered and modeled. From loose boulders kept together through self-gravitation to cohesive fragment agglomerates, from discretized monolithic bodies to bodies with hard nucleus surrounded by loose material.

Chapter 3

Simulations

This chapter is about the simulations of impacts in two main scenarios.

In section 3.1 an hypervelocity impact between an aluminum plate and an aluminum projectile is presented. In 3.1.1 an overview of the simulation is given. Then, the next sections are about the set up of the simulation, the implementation and the results obtained.

Section 3.2 is about a completely diffent scenario. An hypervelocity impact between an asteroid and a spacecraft. This simulation is inspired mainly by the Asteroid Impact and Deflection Assessment (AIDA). The concept of the mission proposes an impactor, the Double Asteroid Redirection Test (DART), that hits the smaller body in a binary asteroid system.

3.1 Aluminum Plate Scenario

This section presents the modeling and the simulation of an aluminum plate hit by an aluminum projectile.

3.1.1 Overview

The main objective of this section of the Thesis is to find a constitutive law to build discretized bodies in Chrono::Engine. So it is about finding a relation between the parameters of the program and the physical parameters of the material that the user wants to simulate. One of the key aspect of this section, is that the constitutive law is searched in the event of hypervelocity impacts. This adds a great complexity to the problem.

To do this, an aluminum plate and a spherical aluminum projectile are created. As seen in chapter 3, the discretization particle is a sphere for both the bodies.

To have an experimental and numerical basis for comparison, this scenario is heavily inspired by the work of E. Watson and Martin O. Steinhauser [48].

The inputs of the simulation are:

- Diameter of the projectile D
- Projectile diameter to plate thickness radio t/D
- Dimensions of the plate L
- Impact velocity v_0

This simulation has many parameters that come into play.

Discretization The dimension and the shape of the elementary particle deeply influences the outcomes of the simulation. Smaller particles leads to an higher number of particles, an higher accuracy in the modeling of continuum and an higher complexity of the numerical problem. An higher number of particles leads also to an higher number of contacts and interactions, conditioning the way in which the whole body responds to solicitations.

Contact method: SMC or NSC As seen in chapter 2, Chrono:Engine deals with contact with two different methods.

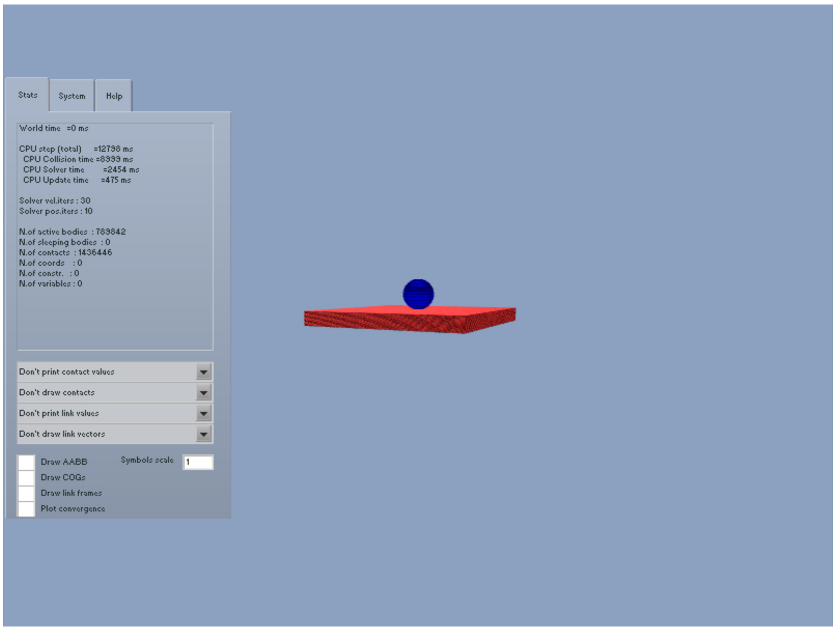
SMC (smooth contact method) creates bodies that interacts with a smooth (penalty) contact method. At the contact point of two bodies there is a fictitious spring-damper system that describes how hard the contact is. With this method, bodies can interpenetrate.

NSC (non-smooth contact method) creates bodies that are completely rigid and solves contact with the solution of a Cone Complementary Problem. With this method, bodies cannot interpenetrate, since this is one of the main hypoteses for a rigid body.

Cohesion Cohesion between elementary particles is one of the main responsibles of how the plate responds to external inputs. Cohesion in Chrono::Engine is a force acting in every contact point of every body. Where the software detects a contact, a cohesion force acts on the bodies that touches. This is one the main parameter that can be used to give a discretized object rigidity. Thus, it is of huge importance for the objective of this section.

Friction Another interparticle interaction is the one due to friction. In Chrono::Engine, the user can set the value of the Coulomb friction coefficient. This parameter is the one that insert energetic dissipation into the system.

Coefficient of restitution A coefficient that rules the ratio of the final to initial relative velocity between two bodies that collide. It is usually in the range $0 \div 1$, where 0 means perfectly inelastic collision and 1 means perfectly elastic collision.



```
World time   =0 ms

CPU step (total)     =12798 ms
  CPU Collision time =8999 ms
  CPU Solver time        =2454 ms
  CPU Update time      =475 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 789842
N.of sleeping bodies  : 0
N.of contacts  : 1436446
N.of coords    : 0
N.of constr.   : 0
N.of variables : 0
```

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence

Symbols scale 1

Figure 3.1: Projectile and plate in Chrono::Engine

Smoothness of the contact for SMC Chrono::Engine allows the selection of the stiffness of the ficticious spring-damper system that is created at the contact of two SMC bodies. This is a key parameter to model and simulate impacts and contacts.

3.1.2 Set up of the simulation

The system is created with a useful tool in Chrono::Engine. Once selected a volume and a distribution of the particles, the software returns the positions. So, for the projectile a spherical volume is selected and for the plate a box volume. For both the bodies, the distribution is a regular grid. The configuration is visible in figure 3.1

Both bodies are discretized with the same elementary particles. Spheres of diameter 0.23686mm and density 5691.43 kg/m^3. The projectile has 40 particles in diameter, while the plate is 17x204x204 particles. The system that comes out is composed by more than 740000 particles. A huge number that is required to discretize the system.

Hypervelocity impacts are a challenge both in terms of modeling and in terms of simulation. Thus, the analysis must start with some hyper-simplified cases. As it can be seen from section 3.1.1, many parameters influences the modeling of the scenario. These over-simplified cases are needed in order to assess the actual effect of every parameter in the best way possible.

Parallel solver

This problem requires, as previously said, a huge number of particles N in order to be consistent with the discretization of continuum. The number of contacts N_c is even higher, because of course every particle has more than one contact point. Thus, the software must solve N_c contacts and then compute and update the state vectors of N particles every time step. This is a huge work for the CPU. In order to decrease the simulation time, the Parallel module of Chrono::Engine is exploided.

This module has many tools to deal with granular dynamics and with system with a very high number of bodies. It has its own high-performance collision detection algorithm and solvers. The CPU parallelization is essential in order to have reasonable simulation times.

SMC analysis

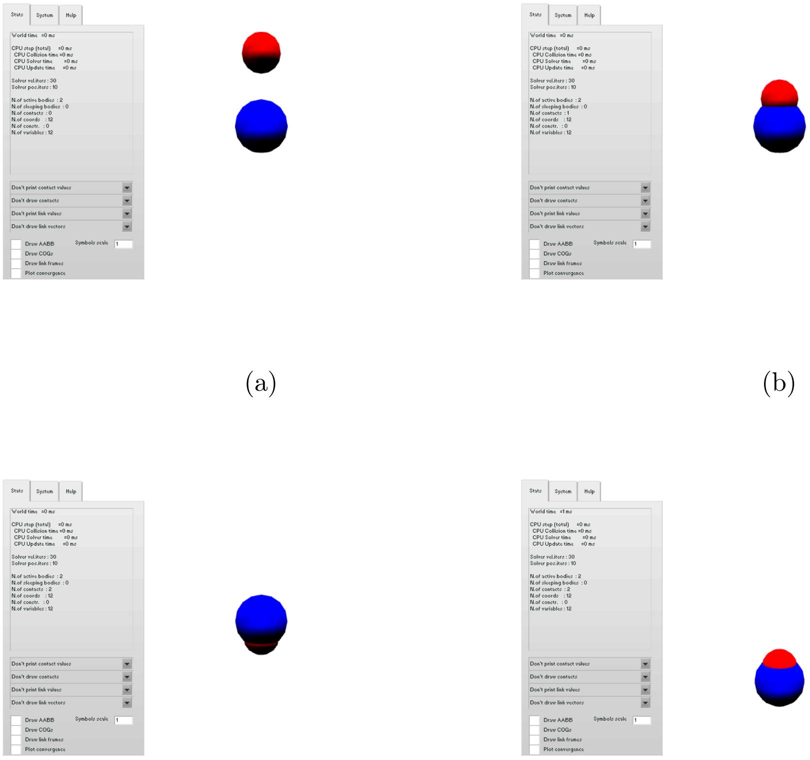
Smooth particles show a fundamental problem that makes them unusable for this application. Two particles of radius r at a distance center-center of $2r$ should be in contact. However, since the SMC particles are smooth, the software does not consider them in contact until their distance d is $d < 2r$. This causes a lot of problems when the cohesion is applied.

When two particles A and B are interpenetrated, a repulsive force (due to the fictitious spring that is created at the contact point) arises. When particle A is moving away from particle B, a cohesive force is created. These opposing actions 'thrust' the particles until a certain moment in which the cohesive force is exceeded and the bond is broken.

This can be shown with a very simple simulation. Two spheres A and B of radius r are in contact, with a distance d slightly smaller than $2r$. A force F is applied to sphere A and an identical force with opposite direction is applied to sphere B. A cohesion force C between A and B is set, such that $C > F$. The bond between the particles should not break up, but it does. At the contact point an instable vibration is created. This problem cannot be overcome not even with a extremely high rigidity of the spheres. The repulsive force would be even higher and the vibration even more instable.

Another issue encountered using SMC particles is that they require the user to set the stiffness parameter for the fictitious spring. If the stiffness is too low the particles may interpenetrate or even pass through each other. If the impact velocity grows, in order to avoid interpenetration, the required stiffness must grow too. At hypervelocities, such as in this case, the stiffness has to grow to a value that makes the computations noisy. When many contacts between many particles are considered, this becomes a true issue. An example of this is visible in figure 3.2. Here the stiffness is selected through the Young's modulus. Its value in this simulation is $69 \cdot 10^9$, the value for aluminum, and the spheres are impacting at 6700 m/s. The red sphere impacts the stationary blue sphere. It can be seen that the red body penetrates the blue one and remains 'stuck'.

These are the reasons why SMC contact method is not suitable when the discretization of a solid material or an impact at hypervelocity are required.



```
World time   =0 ms

CPU step (total)      =0 ms
  CPU Collision time =0 ms
  CPU Solver time        =0 ms
  CPU Update time      =0 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 2
N.of sleeping bodies  : 0
N.of contacts  : 0
N.of coords    : 12
N.of constr.   : 0
N.of variables : 12
```

Symbols scale 1

(a)

```
World time   =0 ms

CPU step (total)      =0 ms
  CPU Collision time =0 ms
  CPU Solver time        =0 ms
  CPU Update time      =0 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 2
N.of sleeping bodies  : 0
N.of contacts  : 1
N.of coords    : 12
N.of constr.   : 0
N.of variables : 12
```

Symbols scale 1

(b)

```
World time   =0 ms

CPU step (total)      =0 ms
  CPU Collision time =0 ms
  CPU Solver time        =0 ms
  CPU Update time      =0 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 2
N.of sleeping bodies  : 0
N.of contacts  : 2
N.of coords    : 12
N.of constr.   : 0
N.of variables : 12
```

Symbols scale 1

(c)

```
World time   =1 ms

CPU step (total)      =0 ms
  CPU Collision time =0 ms
  CPU Solver time        =0 ms
  CPU Update time      =0 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 2
N.of sleeping bodies  : 0
N.of contacts  : 2
N.of coords    : 12
N.of constr.   : 0
N.of variables : 12
```

Symbols scale 1

(d)

Figure 3.2: Impact between particles with too low stiffness

D_{particle}	0.237 mm
Density	5691.43 kg/m^3
$D_{\text{projectile}}$	40 particles
V_{plate}	17x204x204 particles
$v_{\text{projectile}}$	6700 m/s

Table 3.1: Characteristics of the particles and the bodies

NSC analysis

A solution can be found by using rigid bodies instead of smooth ones. These bodies do not have any stiffness parameter and their contacts are just dependent on friction, cohesion and restitution parameters. Moreover, the software detects contacts also when two spheres of radius r are at distance $2r$, differently with respect to SMC bodies. These are the main reasons why the NSC contact method is more suited for the discretization of a rigid body.

3.1.3 Original dimensions

Since the simplified case of hypervelocity impact between two elementary particles works fine, the simulation of the projectile and the plate is conducted. The elementary particles have physical characteristics already presented in section 3.1.2, here summarized in table 3.1.

Time step selection The time step must be capable of sampling the dynamics of the simulation without losing any essential information. SMC systems would require a very small time step in order to make the spring-damper system, that arises at the contact point, stable. NSC does not have this kind of problem. However, the time step must be small enough to avoid interpenetration of particles. A too big time step means that particles travel a great distance every time step. If this distance is too big, there is the risk that the collision detection algorithm does not detect collision at all. The time step is selected in this way:

$$t = \frac{d_{\text{desired}}}{v_{\text{body}}}, \tag{3.1}$$

where d_{desired} is the desired distance that the user wants the body to travel in each time step and v_{body} is the known, or expected, body velocity. In these simulations, d_{desired} is equal to the radius of the particles and v_{body} is equal to the impact velocity of the projectile. In this way the particles travel at most a radius distance every time step. This ensures that the collision detection algorithm works correctly.

Results Impacts between two elementary particles works fine but passing to more than 780000 particles and more than one million contacts might render the process more complicated. The main problem of this simulation is that the particles are too small and the velocities too high to make the solutions trustworthy. Particles move in an unexpected way, with unexpected velocities and in unexpected directions.

The fact is that numerical computations have an intrinsic level of noise. This is acceptable if the numerical error is much smaller than the computed quantities. But when such small particles moves with such an high velocity, a small numerical error is too big to be neglectable.

Physical quantity	New units	Derived quantity
L	10^{-3} m	
M	10^{-9} kg	
t	10^{-6} s	
Density		$1\ \mathrm{L/M^3} = 1 * \mathrm{kg/m^3}$
Velocity		$1\ \mathrm{L/t} = 10^{-3} * \mathrm{m/s}$
Force		$1\ \mathrm{LM/t^2} = 1 * \mathrm{kgm/s^2}$

Table 3.2: Proposed measuring unit modification

3.1.4 Scaled dimensions

The proposed solution is a change in the measurment units in order to render the dimensions higher and the velocities lower. This is possible because Chrono::Engine is unit-less. So, the basic units can be changed. In fact, instead of working with meter and kilograms, one can work with millimiters and grams. Of course, all the other values must be coherent, so, for example, the density must be expressed in [mm/g^3]. In table 3.2 the proposed new basic units are presented.

The new basic units L,M and t (length, mass and time), modifies all the derived units. The new units are selected in order to get bigger particles, with same density, same forces and lower velocities. The new data for the simulation are presented in table 3.3.

This solves many problems. Particles are no more too small and velocity is not too high. In this way the numerical noise is much smaller than the characteristic dimensions of the system. Moreover, according to equation 3.1, the time step also grows.

D_{particle}	0.237 L
Density	5691.43 L/M^3
$D_{\text{projectile}}$	40 particles
V_{plate}	17x204x204x particles
$v_{\text{projectile}}$	6.7 L/t

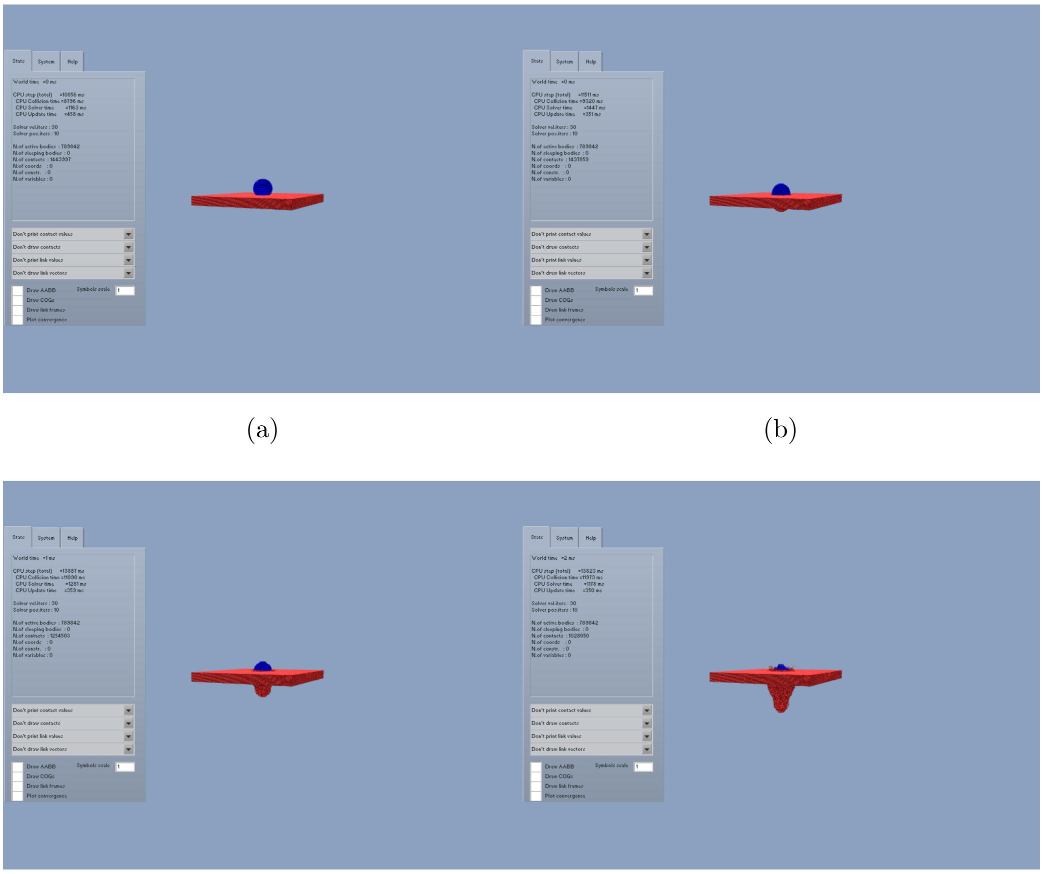
Table 3.3: New data for the simulation

Although many problems are solved thanks to the change in units, the discovery of a constitutive law is still an open problem. Also this simulation of the hypervelocity impact is still not solved. In figure 3.3 the best simulation of the problem is presented. It is obtained with 0 friction coefficient and restitution coefficient and 1500 N of cohesion.

Outcome

The problem of finding a constitutive law for discretized objects is not solved. The simulation of an hypervelocity impact between an aluminum projectile and an aluminum plate is still an open question. This problem presents too many variables and parameters. However, a lot of work in the impact modeling and simulation with Chrono::Engine has been done. The criticalities and the bottlenecks have been highlighted and the possible solutions have been proposed.

All this work, even if not useful to solve the plate impact, has been utilized as a starting point for the simulation of the asteroid impact presented in the next section 3.2.



(a)

```
World time   =0 ms

CPU step (total)      =10856 ms
  CPU Collision time =8796 ms
  CPU Solver time         =1163 ms
  CPU Update time      =458 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 789842
N.of sleeping bodies  : 0
N.of contacts  : 1443997
N.of coords    : 0
N.of constr.   : 0
N.of variables : 0

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
Symbols scale 1
```

(b)

```
World time   =0 ms

CPU step (total)      =11511 ms
  CPU Collision time =9320 ms
  CPU Solver time         =1447 ms
  CPU Update time      =351 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 789842
N.of sleeping bodies  : 0
N.of contacts  : 1437859
N.of coords    : 0
N.of constr.   : 0
N.of variables : 0

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
Symbols scale 1
```

(c)

```
World time   =1 ms

CPU step (total)      =13887 ms
  CPU Collision time =11898 ms
  CPU Solver time         =1281 ms
  CPU Update time      =359 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 789842
N.of sleeping bodies  : 0
N.of contacts  : 1254503
N.of coords    : 0
N.of constr.   : 0
N.of variables : 0

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
Symbols scale 1
```

(d)

```
World time   =2 ms

CPU step (total)      =13823 ms
  CPU Collision time =11973 ms
  CPU Solver time         =1178 ms
  CPU Update time      =350 ms

Solver vel.iters : 30
Solver pos.iters : 10

N.of active bodies  : 789842
N.of sleeping bodies  : 0
N.of contacts  : 1028050
N.of coords    : 0
N.of constr.   : 0
N.of variables : 0

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
Symbols scale 1
```

Figure 3.3: Snapshots of the impact simulation

D_{primary}	780 m
$D_{\text{secondary}}$	160 m
M_{system}	$5.28 \cdot 10^{11}$ kg
$M_{\text{secondary}}$	$4.8 \cdot 10^{9}$ kg

Table 3.4: Physical properties of 65803 Didymos

3.2 Asteroid Impact Scenario

This scenario is inspired by the AIDA mission. An impactor hits an asteroid to deflect its trajectory. The surroundings of Earth orbit are full of bodies that may encounter our planet in the future, the so called NEOs (Near Earth Objects). Redirection of PHOs (Potentially Hazardous Objects) has become an actual necessity for the human race.

3.2.1 Overview

In this section, impacts against asteroids are discussed. The scenario that should be recreated, is about an impactor that redirects an asteroid. The projectile must impart a net Δv to the asteroid, avoiding the creation of a too big plume of debris or the complete distruction of the body.

DART is set to impact the smaller body of the binary asteroid system 65803 Didymos. In tables 3.4 and 3.5, the physical properties of the binary system and the orbital properties are shown.

In table 3.6, the properties of the Spacecraft and the main characteristics of the mission are presented.

In this section two models for the asterоid are utilized. In section 3.2.3

e	0.03
v_{orbital}	17 cm/s
$v_{\text{heliocentric}}$	23 km/s
T	12.11 h
a	1.18 km

Table 3.5: Orbital properties of the secondary body of 65803 Didymos

M	$\sim$ 500 kg
v_{relative}	$\sim$ 6 km/s
Δv_{didy}	$\sim$ 0.4 mm/s
ΔT_{didy}	$\sim$ 7 min

Table 3.6: Orbital properties of the secondary body of 65803 Didymos

the asteroid is entirely composed by particles. In section 3.2.4 the asteroid is created by a solid nucleus and some boulders covering its surface.

As in section 3.1.2, also for these simulations, the Chrono::Engine CPU parallelization is exploited.

Inputs

The inputs for this scenario are:

- Mass of the impactor M
- Shape of the impactor
- Relative impact velocity
- Discretization of the asteroid
- Contact method NSC

Parameters

In these simulations, the main goal is to create a body that does not shatter when impacted. This means that a certain level of rigidity must be achieved. Chrono::Engine allows the user to work with certain parameters to reach this objective.

Cohesion As in paragraph 3.1.1, the cohesion is the force that is applied by the software at every contact point. Higher cohesion means higher overall strength. The bond between the particles creating a body is harder to be broken.

Friction coefficient Another way to achieve global rigidity, is to set a very high friction coefficient. An high friction coefficient makes the relative motion between particles in contact very hard.

Outputs

The analysis is focused on the redirection capability of an impactor. The outputs of these simulation are:

Velocity change The asteroid gains momentum when the impactor hits it. This velocity change would be completely negligible in terms of heliocentric orbits. This is why the binary system 65803 Didymos is very useful. As it can be seen in table 3.5, the orbital velocity of the moon of the system (from now on Didymoon) is in the order of cm/s. This means that even a small gain in velocity (mm/s) is relevant if the orbital reference frame.

Period change What can be really sensed with observations from the Earth is the change in the orbital period of Didymoon. A little orbital period shift is not visible instantaneously, but after a few days it is measurable from ground based observation.

3.2.2 Set up of the simulation

First of all, it is important to visualize the forces that comes into play when gravitational aggregates are considered. In these scenarios, every particle of the system is attracted by every other particle. These forces are the ones that should contrast the impulse produced by the impactor.

To show these forces, a simple gravitational aggregation of few bodies is implemented. The analysis starts with two bodies, and then 8, 64 and 512

N. of particles	Density[kg/m^3]	Radius [m]
2	2000	40
8	2000	25.2
64	2000	12.6
512	2000	6.3

Table 3.7: Characteristics of the bodies

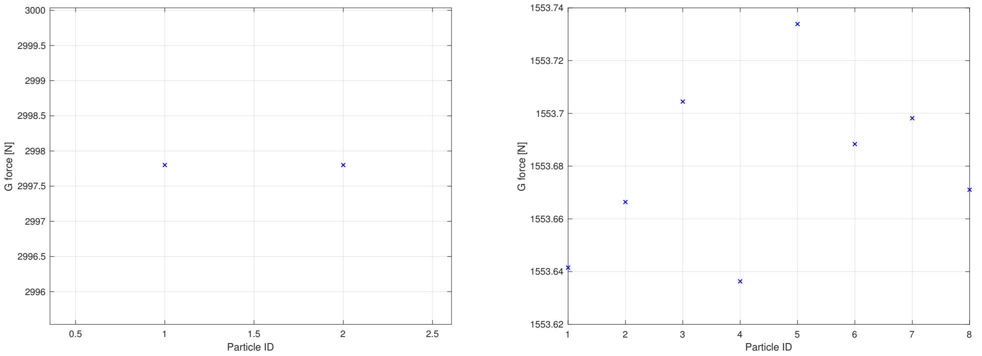
particles. In these four simulations the total mass is conserved. In table 3.7, the characteristics of the particles for every case are shown.

2 particles Two particles are created and they are free to attract each other gravitationally. This is a simple gravitational two-body problem that can be solved analitically.

8 particles The particles are created in a regular grid inside a square box volume. The particles are not in contact at the beginning of the simulation, but they have a little distance in order to allow them to gravitationally aggregate.

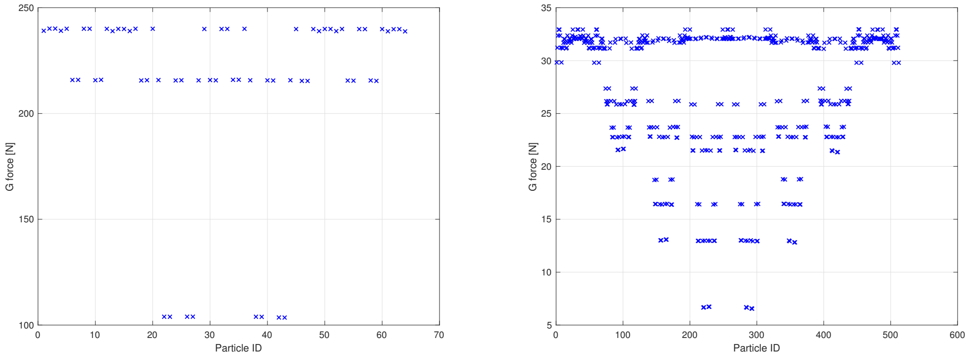
64 and 512 particles Also in these simulations, the particles are in a regular grid within a cubic volume. Again, they are not in contact at the beginning of the simulation and they are free to aggregate.

All the results are presented in figure 3.4. The graph shows the norm of net force acting on every particle. As it can be seen, the more the mass of the particles get smaller, the more the net force on the particles get smaller.



(a) 2 particles

(b) 8 particles



(c) 64 particles

(d) 512 particles

Figure 3.4: Gravitational forces acting on particles

N. particles	10307
Radius	3 m
Density	1600 kg/m^3

Table 3.8: Properties of the aggregate elementary particles

3.2.3 Aggregate impact

In this section, impacts against an asteroid are simulated. The objective is to find parameters that allows the asteroid to be mostly intact when hit by a projectile. Post-impact aggregate velocity and orbital period are investigated. In the following paragraph the construction of the asteroid is analysed. Then, the model of the impactor is treated. Finally, the outputs of the simulation are presented.

How the asteroid is modeled

The way in which the continuum is discretized in this work is by mean of elementary particles. In this section, the asteroid elementary particles are all identical spherical particles. In table 3.8, the physical properties of the elementary particles are presented.

The asteroid is built exploiting the self gravitation of the particles. At the beginning of the simulation, all the particles are created in random positions inside a given spherical volume. All the particles are at rest at the beginning to ease the gravitational aggregation and to create a quasi-spherical asteorid (similar to Didymoon). The initial conditions for the particles are summarized in table 3.9

After the creation of the particles, gravitational forces are computed and

$\mathbf{v}_{\text{initial}}$	$[0, 0, 0]$ m/s
$\boldsymbol{\omega}_{\text{Radius}}$	$[0, 0, 0]$ rad/s
R sphere	95m

Table 3.9: Initial conditions of the aggregate elementary particles

applied to every particles. The software solves at each time step an extact N-body problem. This means that at each time step, the program scans all the particles, gets their positions, their mass (in this case all the mass are equal) and computes the gravitational force. A pseudo code is presented in algorithm 1.

The result is the one presented in the picture 3.5. The physical characteristics are presented in table 3.10.

The simulated asteroid is very similar to Didymoon, both in shape and in the order of magnitude of the mass.

In figure 3.6 it is possible to see the positions of the centers of mass of every particle that makes up the asteroid. In figure 3.7, the gravitational force to which every particle is subjected is presented. These pictures are presented with a color code that links the position in the aggregate to the ID of the particle. This is useful to understand the order of magnitude of the force that acts on every particle in relation to the position.

How the impactor is modeled

The impactor is modeled in a very simple way, it is a spherical body. What it is important in this analysis is to assess the post-impact orbit of the asteroid. A refined discretization of the impactor would not be relevant in this scenario.

```
Result: Net force of every particle
EmptyForcesAccumulator on all bodies
for Every body i do
    for Every body j do
        GetPosition.Body.i
        GetPosition.Body.j
        GetMass.Body.i
        GetMass.Body.j
        ComputeForce
        Move on next body j
        AccumulateForce on Body i
        AccumulateForce on Body j
        Move on next Body j
    end
    Move on next Body i
end
```

Algorithm 1: Computation of gravitation forces

Diameter	~ 160 m
Mass	$1.865 \cdot 10^9$ kg

Table 3.10: Physical characteristics of the asteroid

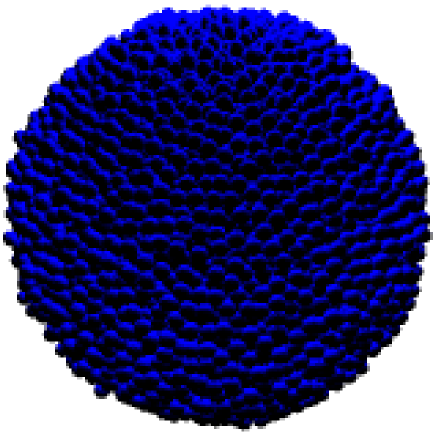


Figure 3.5: Aggregate in Chrono::Engine

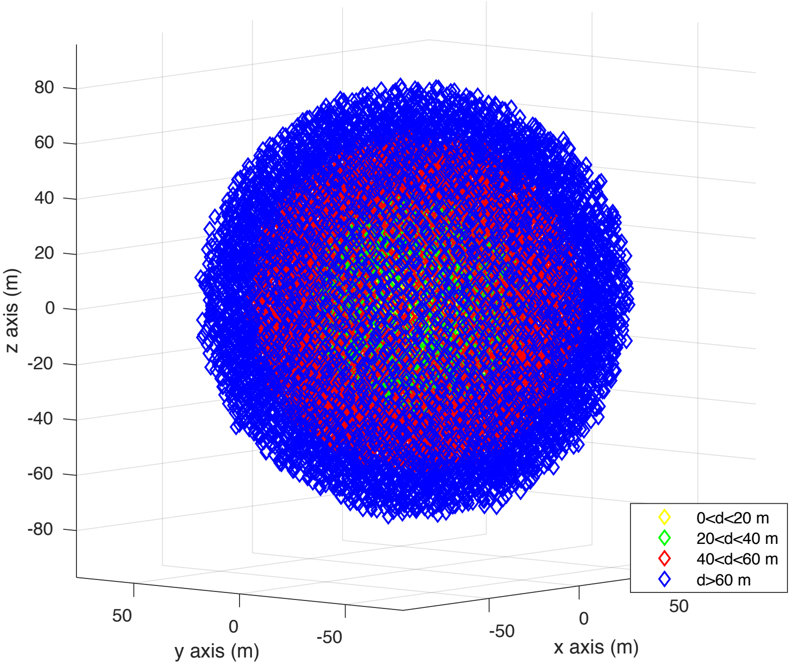


Figure 3.6: Position of the particles in the aggregate

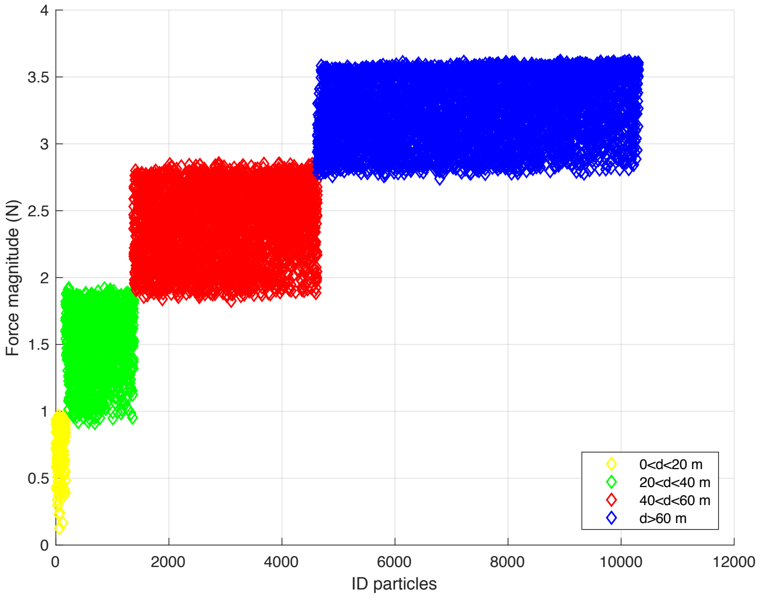


Figure 3.7: Gravitational force acting on the particles in the aggregate

Diameter	0.5 m
Density	1000 kg/m^3
Velocity	4000 m/s

Table 3.11: Properties of the impactor

In table 3.11, the physical properties and the velocity of the impactor are presented.

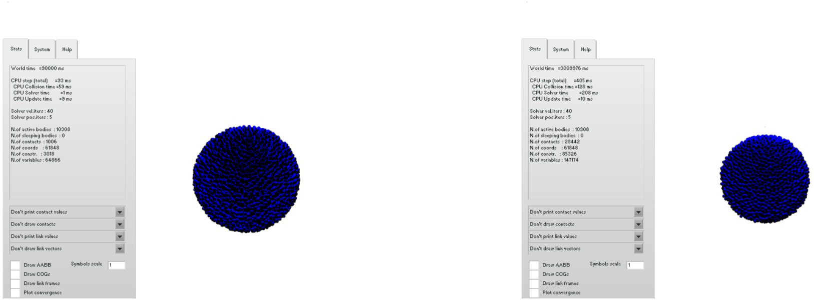
Simulations

As introduced in paragraph 3.2.1, the parameter for this set of simulations is cohesion. The outputs (described in paragraph 3.2.1) and the dependance on the parameter, are investigated. Only the cohesion varies while the friction coefficient is kept at a constant value.

Time step selection

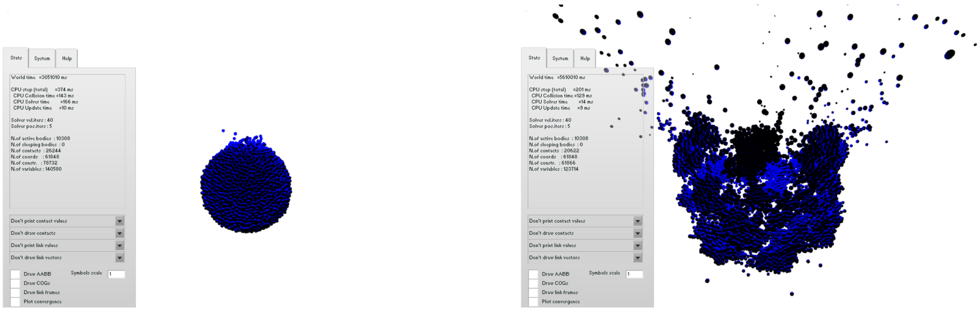
It is important that the dynamics of the system is simulated with precision. Time step selection is a key property of the program. A simulation with a too high time step is likely to produce non physical results. Particles can interpenetrate, the software can miss contacts and so on. Of course, at the contrary, the time step cannot be too small because the simulation would last too long and practicality would be lost.

The simulation has three main phases. The aggregation, the impact and the evolution. For the aggregation, a time step of 10 s has been chosen. Particles are quite big and do not have an high velocity. There are no risks of interpenetration and the this time-step makes the simulation quite fast.



(a) Creation of the bodies

(b) Pre-impact



(c) Just after impact

(d) After impact

Figure 3.8: Evolution of the impact with cohesion 100 *N*

Simulation phase	Time step [s]
Aggregation	10
Impact	0.0001
Evolution	0.1 1

Table 3.12: Time step selection

The impact phase is more critical, since a small particle is moving very fast. This requires a very small time step. A time step of 0.0001 s has been chosen. This means that the impactor travels a distance of 0.4 m every time step (before the impact). This distance is less that the radius length, and so no interpenetration occurs. The post-impact evolution has been divided into two sections. The first one is more dangerous since the particles are close and they gain velocity, while the second one is less problematic because the system is stabilized and the evolution is already begun. For the first evolution phase the time step is set to 0.1 s, while for the second one is 1 s. These informations are collected in table 3.12.

Cohesion

In this section, cohesion is the only parameter that is changing. All other characteristics of the material are kept constant in all the simulations. In table 3.13 they are summarized.

Since the gravitational force that binds the particles in the asteroid is so small (see figure 3.7), a cohesion force is needed in order to make the aggregate rigid enough to not shatter in the impact. The cohesion is set to

Friction coefficient	0.5
Restitution coefficient	0.3

Table 3.13: Surface characteristics of the particles and impactor

100 N, 200 N, 300 N, 400 N, 500 N, 1000 N and 1000 N.

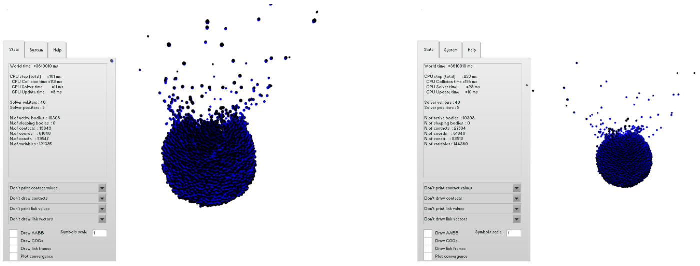
The post-impact aggregates can be seen in figure 3.9. The difference in size of the asteroid is simply due to the different position of the camera, but the asteroids are equal.

In figure 3.10, the distribution of the velocities of the particles after the impact are presented. These values are taken ~ 90 s after the impact, at the same moment for every simulation.

As expected, lower cohesion leads to weaker aggregates. This can be seen both in figure 3.9 and 3.10. Only asteroids with cohesion of 1000 N and 2000 N show no debris. Also the velocity distribution helps with the analysis. At lower cohesion values, the particles have a greater variety in velocity. When cohesion grows, more and more particles have the same order of magnitude of velocity, and the only ones that stand out are the debris. Post-impact velocities of the asteroids with cohesion 1000 N and 2000 N are very similar. All the particles have practically the same velocity and almost no debris are present.

Results

As mentioned in 3.2.1, the analysis is focused on the redirection capability of the kinetic impactor. Since the target shall be intact after the impact and a moderate number of debris are desired, the analysis is focused on the

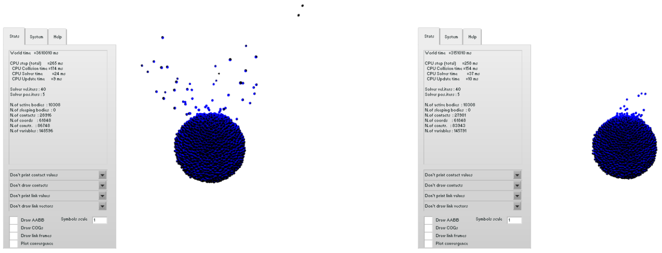


```
World time   =3610010 ms

CPU step (total)     =181 ms
  CPU Collision time =112 ms
  CPU Solver time        =11 ms
  CPU Update time      =9 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 19849
N.of coords    : 61848
N.of constr.   : 59547
N.of variables : 121395

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

```
World time   =3610010 ms

CPU step (total)     =253 ms
  CPU Collision time =156 ms
  CPU Solver time        =28 ms
  CPU Update time      =10 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 27504
N.of coords    : 61848
N.of constr.   : 82512
N.of variables : 144360

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

(a) Cohesion 100 N

(b) Cohesion 200 N

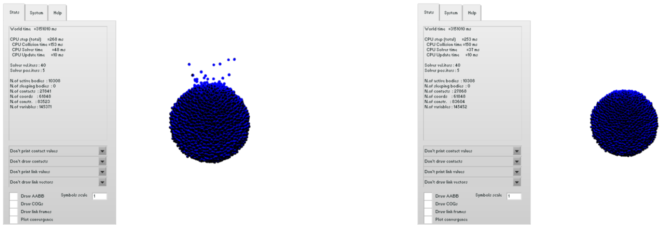


```
World time   =3610010 ms

CPU step (total)     =265 ms
  CPU Collision time =174 ms
  CPU Solver time        =24 ms
  CPU Update time      =9 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 28916
N.of coords    : 61848
N.of constr.   : 86748
N.of variables : 148596

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

```
World time   =3151010 ms

CPU step (total)     =258 ms
  CPU Collision time =154 ms
  CPU Solver time        =37 ms
  CPU Update time      =10 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 27981
N.of coords    : 61848
N.of constr.   : 83943
N.of variables : 145731

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

(c) Cohesion 300 N

(d) Cohesion 400 N

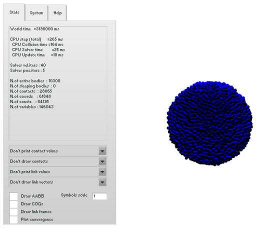


```
World time   =3151010 ms

CPU step (total)     =268 ms
  CPU Collision time =153 ms
  CPU Solver time        =48 ms
  CPU Update time      =10 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 27841
N.of coords    : 61848
N.of constr.   : 83523
N.of variables : 145371

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

```
World time   =3151010 ms

CPU step (total)     =253 ms
  CPU Collision time =150 ms
  CPU Solver time        =37 ms
  CPU Update time      =10 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 27868
N.of coords    : 61848
N.of constr.   : 83604
N.of variables : 145452

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

(e) Cohesion 500 N

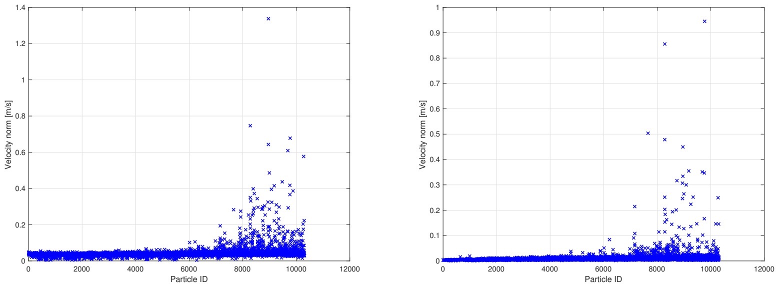
(f) Cohesion 1000 N



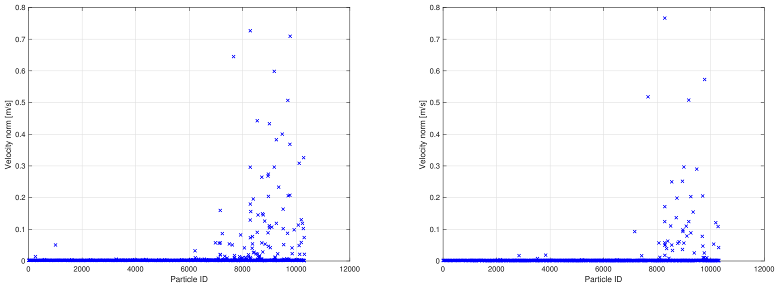
```
World time   =3190000 ms

CPU step (total)     =265 ms
  CPU Collision time =164 ms
  CPU Solver time        =25 ms
  CPU Update time      =10 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 10308
N.of sleeping bodies  : 0
N.of contacts  : 28065
N.of coords    : 61848
N.of constr.   : 84195
N.of variables : 146043

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors

☐ Draw AABB        Symbols scale  1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence
```

(g) Cohesion 2000 N

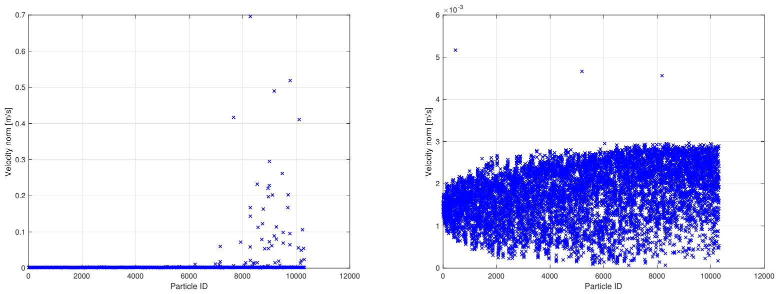
Figure 3.9: Post-impact condition of the asteroid



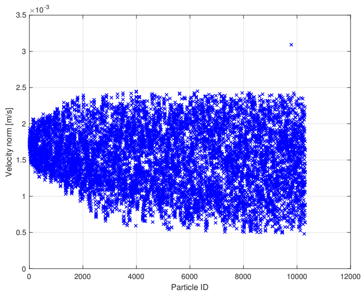
(a) Cohesion 100 N (b) Cohesion 200 N



(c) Cohesion 300 N (d) Cohesion 400 N



(e) Cohesion 500 N (f) Cohesion 1000 N



(g) Cohesion 2000 N

Figure 3.10: Post-impact velocity of the particles

asteroid with cohesion values of 400 N, 500 N, 1000 N and 2000 N.

First of all, the post-impact velocity of the aggregate is computed. Scanning all the particles and excluding the debris, the post-impact velocity is computing avaraging the velocities with the number of the aggregate particles (N).

$$\mathbf{v}_{\text{aggregate}} = \frac{\sum_{i=1}^{N} \mathbf{v}_i}{N} \tag{3.2}$$

This post-impact velocity is the $\Delta\mathbf{v}$ imparted to the asteroid.

The binary asteroid system is simply modeled with the secondary in a circular orbit around the primary with a radius R. Only the component of $\Delta\mathbf{v}$ along the orbital velocity is able to change the orbital period. With these assumptions, it is very easy to compute the post-impact semi-major axis, a_{new}, and the post-impact orbital period, T_{new}.

$$a_{\text{new}} = -\frac{\mu}{2}\left[\frac{1}{\frac{v_{\text{new}}^2}{2} - \frac{\mu}{R}}\right] \tag{3.3}$$

$$T_{\text{new}} = 2\pi\sqrt{\frac{a_{\text{new}}^3}{\mu}}, \tag{3.4}$$

being $v_{\text{new}} = v_{\text{old}} + \Delta v$.

Table 3.14 presents data from the original system and the results of the simulations for cohesion (c) $c = 400$ N, $c = 500$ N, $c = 1000$ N, $c = 1000$ N.

From the table 3.14, it is visible how at $c = 500$ N there is a maximum in the gained velocity Δv. This is explainable thanks to momentum transfer efficiency β. Ejecta that are released back towards the incident direction carry away a section of the momentum. Thus, an impactor with mass m, velocity v, transfers an impulse p higher than mv but it is

$$p = mv + p_{\text{ejecta}} = \beta mv. \tag{3.5}$$

System		Results
Original system	$v_{\text{old}} = 0.17$ m/s $R = 1180$ m $\mu = 34.102$ m^3/s^2 $T_{\text{old}} = 726.878$ min	
$c = 2000$ N	$\Delta v = 0.0014$ m/s	$v_{\text{new}} = 0.1714$ m/s $a_{\text{new}} = 1199.84$ m $T_{\text{new}} = 745.287$ min $\Delta T = 18.409$ min
$c = 1000$ N	$\Delta v = 0.0014$ m/s	$v_{\text{new}} = 0.1714$ m/s $a_{\text{new}} = 1199.84$ m $T_{\text{new}} = 745.287$ min $\Delta T = 18.409$ min
$c = 500$ N	$\Delta v = 0.0019$ m/s	$v_{\text{new}} = 0.1719$ m/s $a_{\text{new}} = 1207.13$ m $T_{\text{new}} = 752.903$ min $\Delta T = 25.215$ min
$c = 400$ N	$\Delta v = 0.0017$ m/s	$v_{\text{new}} = 0.1717$ m/s $a_{\text{new}} = 1204.2$ m $T_{\text{new}} = 749.358$ min $\Delta T = 22.479$ min

Table 3.14: Simulation Results

	$v_{\text{ejecta}}[\text{m/s}]$
$c = 2000$ N	0
$c = 1000$ N	0.004
$c = 500$ N	4.607
$c = 400$ N	4.248

Table 3.15: Total post-impact ejecta velocity in incident direction

In fact, if the ejecta velocity in incident direction grows, also the transfer efficiency grows. The sum of all the post-impact velocities in incident direction v_{ejecta} confirms this theory. Table 3.15 shows the results. The velocity is computed as

$$v_{\text{ejecta}} = \sum_{i=1}^{N_{\text{ejecta}}} v_i^{\text{ejecta}}, \tag{3.6}$$

considering only ejecta that has a positive incident velocity.

For higher cohesions, there is almost no contribution. Asteroid with $c = 2000$ N has no ejecta with positive velocity (even very low) at all. Asteroid with $c = 500$ N and $c = 400$ N show the highest v_{ejecta} of the set. The aggregate with $c = 500$ N has the maximum value because its ejecta are slighly more directed in the incident direction, while ejecta of the aggregate with $c = 400$ N are more spread in the other directions.

Thus, ejecta are one of the main resposible of the impulse tranfer enhancement. Beyond that, a plume of debris may be observable from Earth-based telescopes, offering another way to advance the understanding of impact processes on asteroids.

Diameter nucleus	~ 140 m
Diameter boulders	3.4 m
N boulders	12428
Density	1600 kg/m^3

Table 3.16: Properties of the particles

3.2.4 Hard nucleus impact

This section deals with a different modeling for the asteroid. Everything except for the aggregate works in the same way as in section 3.2.3. Inputs, parameters, time steps are identical.

How the asteroid is modeled

The objective is again to create a quasi-circular asteroid with mass and dimension similar to Didymoon. In section 3.2.3 the asteroid is entirely composed by 'elementary' particles; identical spheres that through interparticle interactions produce a solid body. In this section the asteroid is made by an hard nucleus and some boulders covering the surface of the inner body.

The nucleus has a radius of 70 m, the boulders are 3.4 m in diameter and all of them has a diameter of 1600 kg/m^3 (schematized in table 3.16). The final properties of the asteroid are presented in table 3.17. In figure 3.11 the position of the particles are represented in a 3D graph; a section of the boulders is not shown in order to make the red nucleus visible.

Practically, to build the asteroid, the central nucleus is created and the boulders are free to be gravitationally attracted to central body. To be sure that cohesion does not influence the aggregation, for this phase the value for

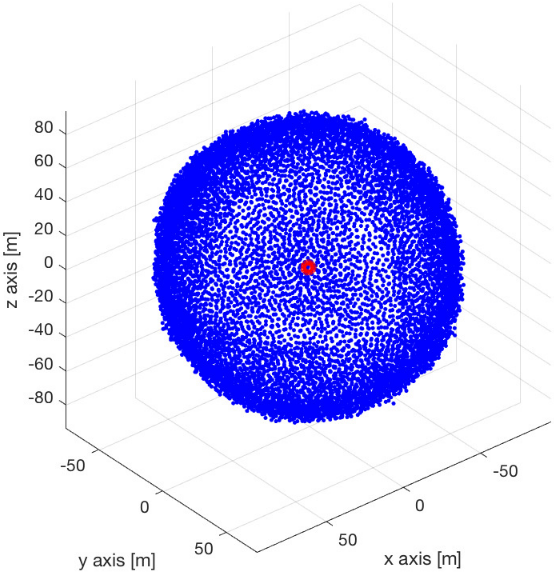


Figure 3.11: Particles position. In red the nucleus

Diameter	~ 155 m
Mass	$1.523 \cdot 10^9$ kg

Table 3.17: Physical properties of the asteroid

cohesion is set to 0 N. Only when a stable condition is reached, the cohesion is set to the value required by the simulation.

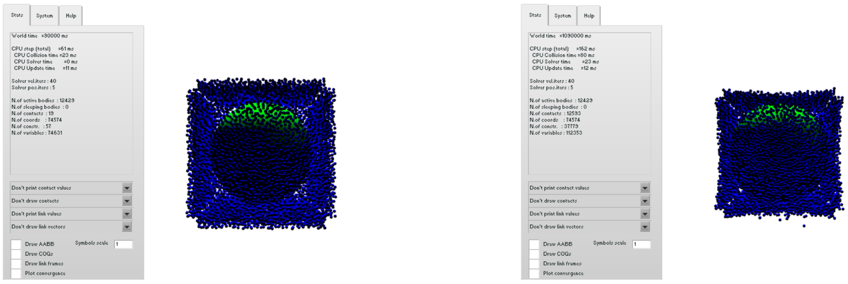
Chrono::Engine has a powerful tool to create bodies in a given volume with a given spatial distribution. This is very useful when an high number of particles has to be produced. Unluckily, Chrono::Engine does not allow the selection of a hollow sphere as a volume to be sapled. The hollow sphere is required in order to not have interpenetration between the nucleus and the particles that are created inside the volume of the central body. This is solved by sampling 6 parallelepipeds that are placed in the 6 directions of the 3D space. After the creation of the bodies, a N-body algorithm (as in 1) computes the gravitational forces that act on every body and the system evolves in the desired nucleus covered by boulders with an almost spherical final shape. The process of the creation of the asteroid is shown in figure 3.12.

Results

As in section 3.2.3, the parameter investigated is cohesion and the other material characteristics that Chrono::Engine makes available to select are kept constant. Thus, friction coefficient and restitution coefficient are, respectively, 0.5 and 0.3 (as in table 3.13). The cohesion values that are investigated are the ones that gives an acceptable solution in section 3.2.3. So, $c = 400$ N, $c = 500$ N, $c = 1000$ N and $c = 2000$ N.

In this section, also the computation of the output is easier since the post-impact velocity is not to be calculated but it is enough to know the velocity of the nucleus.

Figure 3.13 shows the post-impact conditions of the asteroid for the different values of cohesion. In can be seen that for $c = 500$ N there is a maximum



```
World time   =90000 ms

CPU step (total)      =61 ms
  CPU Collision time =23 ms
  CPU Solver time         =0 ms
  CPU Update time      =11 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12429
N.of sleeping bodies  : 0
N.of contacts  : 19
N.of coords    : 74574
N.of constr.   : 57
N.of variables : 74631
```

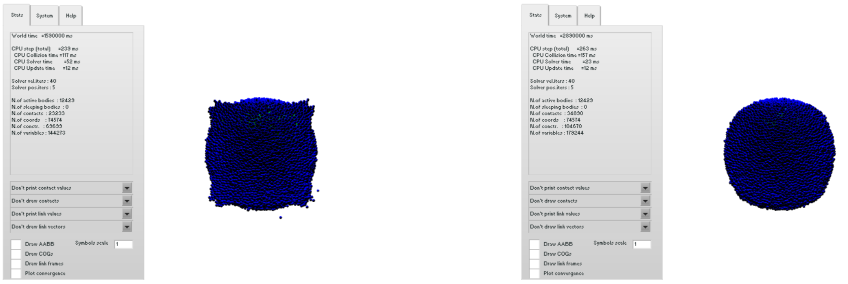
Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
☐ Draw AABB Symbols scale 1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence

```
World time   =1090000 ms

CPU step (total)      =162 ms
  CPU Collision time =80 ms
  CPU Solver time         =23 ms
  CPU Update time      =12 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12429
N.of sleeping bodies  : 0
N.of contacts  : 12593
N.of coords    : 74574
N.of constr.   : 37779
N.of variables : 112353
```

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
☐ Draw AABB Symbols scale 1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence

(a) Initial set-up

(b) Particles being attracted by nucleus



```
World time   =1590000 ms

CPU step (total)      =239 ms
  CPU Collision time =117 ms
  CPU Solver time         =52 ms
  CPU Update time      =12 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12429
N.of sleeping bodies  : 0
N.of contacts  : 23233
N.of coords    : 74574
N.of constr.   : 69699
N.of variables : 144273
```

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
☐ Draw AABB Symbols scale 1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence

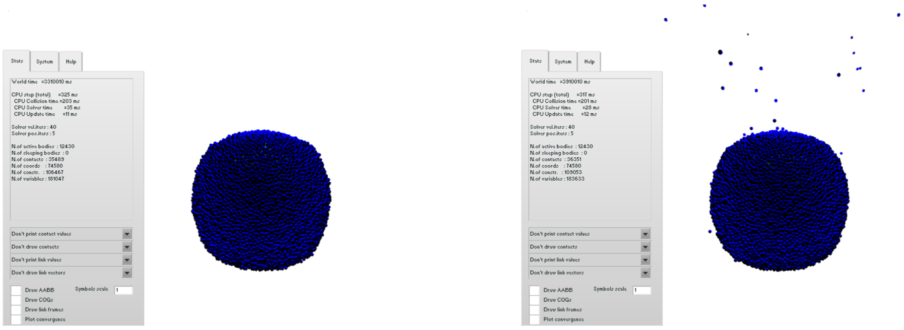
```
World time   =2890000 ms

CPU step (total)      =263 ms
  CPU Collision time =157 ms
  CPU Solver time         =23 ms
  CPU Update time      =12 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12429
N.of sleeping bodies  : 0
N.of contacts  : 34890
N.of coords    : 74574
N.of constr.   : 104670
N.of variables : 179244
```

Don't print contact values
Don't draw contacts
Don't print link values
Don't draw link vectors
☐ Draw AABB Symbols scale 1
☐ Draw COGs
☐ Draw link frames
☐ Plot convergence

(c) Particles almost set

(d) Final aggregate

Figure 3.12: Creation of the asteroid with a solid nucleus

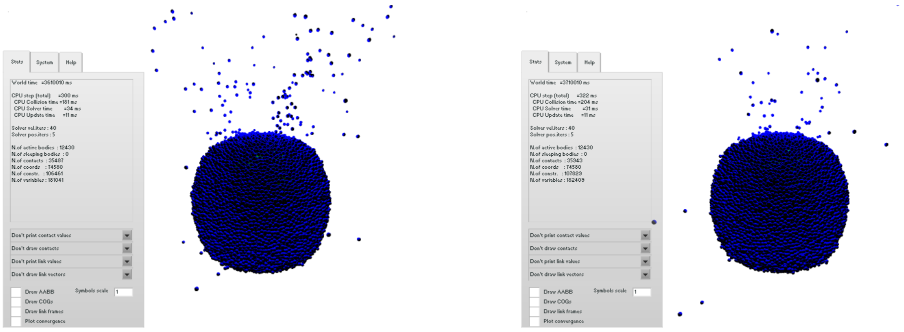


```
World time   =3310010 ms

CPU step (total)      =325 ms
  CPU Collision time =203 ms
  CPU Solver time        =35 ms
  CPU Update time      =11 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12430
N.of sleeping bodies  : 0
N.of contacts  : 35489
N.of coords    : 74580
N.of constr.   : 106467
N.of variables : 181047
```

```
World time   =3910010 ms

CPU step (total)      =317 ms
  CPU Collision time =201 ms
  CPU Solver time        =28 ms
  CPU Update time      =12 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12430
N.of sleeping bodies  : 0
N.of contacts  : 36351
N.of coords    : 74580
N.of constr.   : 109053
N.of variables : 183633
```

(a) $c = 2000$ N

(b) $c = 1000$ N



```
World time   =3610010 ms

CPU step (total)      =300 ms
  CPU Collision time =181 ms
  CPU Solver time        =34 ms
  CPU Update time      =11 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12430
N.of sleeping bodies  : 0
N.of contacts  : 35487
N.of coords    : 74580
N.of constr.   : 106461
N.of variables : 181041
```

```
World time   =3710010 ms

CPU step (total)      =322 ms
  CPU Collision time =204 ms
  CPU Solver time        =31 ms
  CPU Update time      =11 ms

Solver vel.iters : 40
Solver pos.iters : 5

N.of active bodies  : 12430
N.of sleeping bodies  : 0
N.of contacts  : 35943
N.of coords    : 74580
N.of constr.   : 107829
N.of variables : 182409
```

(c) $c = 500$ N

(d) $c = 400$ N

Figure 3.13: Post-impact conditions of the asteroid

of ejecta in the incident direction while for $c = 2000$ N there are no debris at all.

As in the previous section, a table can be created to collect all the results (table 3.18).

It is visible how this modeling of the asteroid makes the momentum transfer generally less effective. All the Δv gained are lower than the ones computed in section 3.2.3. The results for the asteroid with $c = 2000$ N matches pretty well the one expected in [6] for impacts with $\beta = 0$ (which means no

System		Results
Original system	$v_{\text{old}} = 0.17$ m/s $R = 1180$ m $\mu = 34.102$ m^3/s^2 $T_{\text{old}} = 726.878$ min	
$c = 2000$ N	$\Delta v = 0.00056$ m/s	$v_{\text{new}} = 0.17056$ m/s $a_{\text{new}} = 1187.84$ m $T_{\text{new}} = 734.134$ min $\Delta T = 7.256$ min
$c = 1000$ N	$\Delta v = 0.00066$ m/s	$v_{\text{new}} = 0.17066$ m/s $a_{\text{new}} = 1189.25$ m $T_{\text{new}} = 735.444$ min $\Delta T = 8.566$ min
$c = 500$ N	$\Delta v = 0.0015$ m/s	$v_{\text{new}} = 0.1715$ m/s $a_{\text{new}} = 1201.29$ m $T_{\text{new}} = 746.641$ min $\Delta T = 19.736$ min
$c = 400$ N	$\Delta v = 0.0010$ m/s	$v_{\text{new}} = 0.1710$ m/s $a_{\text{new}} = 1194.09$ m $T_{\text{new}} = 739.936$ min $\Delta T = 13.058$ min

Table 3.18: Simulation Results

	v_{ejecta}[m/s]
$c = 2000$ N	0.5988
$c = 1000$ N	2.2168
$c = 500$ N	22.5372
$c = 400$ N	13.9917

Table 3.19: Total post-impact ejecta velocity in incident direction

ejecta). As in previous section, asteroid with $c = 500$ N receives the biggest Δv. Table 3.19, similar to 3.15, presents the v_{ejecta} as a function of asteroid cohesion.

As before, the asteroid that shows the biggest Δv is the one that has the maximum v_{ejecta} (always agreeing with the momentum transfer efficiency). Even if the ejecta velocities with the hard nucleus are higher that the ones coming from the aggregate impact, the Δv is smaller. This is probably due to a less efficient way to transmit momentum.

Chapter 4

Conclusion

Having a tool to simulate hypervelocity impacts is fundamental in order to understand the dynamics of such a complex scenario. Experiments about this kind of impact can be very expensive and complicated in the case of spacecraft materials and of course they would be impossible applied to asteroids.

A reliable simulation tool is key to design structural parts of satellites or for spacecraft shields. The threat represented by space debris makes the definition of efficient countermeasures. Understanding how to utilize simulations to characterize the most relevant aspects of the dynamics of an hypervelocity impact would allow engineers to research new solutions for collision protection.

Characterizing the dynamics of impacts on asteroids and small irregular bodies, such at comets, is crucial for the future Space missions. Engineers and physicists have the possibility to predict with more accuracy the outcomes of missions that involve collision with small bodies. They could be hard landings, kinetic impactors to deflect asteroids or to create an ejecta plume. Moreover, this tool can be used to simulate impacts between asteroids and planetoid in order to simulate the dynamics of the formation of the small bodies of the Solar System.

4.1 Summary of plate scenario

The plate scenario, presented in section 3.1, has proved to be the most challenging one. The management of every aspect of the simulation, small particles, the very high impact velocity, the great number of parameters and the packing of the particles, has turned out to be too ambitious.

No reliable results have been found and the impact simulation is still to be completed. Collisions still presents non-phyical behaviours when many particles interact.

4.1.1 What can be learned from this work

However, a lot has been accomplished. Many criticalities have been found and highlighted. Some important guidelines useful to simulate hypervelocity impacts are summarized in the following paragraphs.

Contact Method Soft bodies (SMC) has proved not to be suitable for this application. The difficulty to create a lattice and to exploit correctly cohesion properties of the materials makes soft particles unappealing. Moreover, because of the high velocity collisions, stiffness of the particles must be set to a value that is too high to provide reliable results.

Better outcomes can be obtained with the use of rigid particles (NSC). They are able to fit properly in a lattice and the cohesion forces are steady and works as expected. Also, no stiffness has to be set and no problems in the collision arises.

Unit System Choice Extremely small particles shall be avoided to make the collision detection algorithm work properly. For example, instead of working with meters, the user can utilize millimiters to have 'bigger' bodies. This process can be applied to other quantities like masses. Having a big body that has a very small mass can trigger issues with the solvers. However, this process of using non-standars units can help to exploit the ranges of values in which the software works in an optimal way.

4.1.2 Future works

A general constitutive law that links physical material parameters to software parameters has to be found yet. This would be fundamental to render this tool universal and applicable to any configuration of impacts between any material at any velocity.

Another improvement that would make the simulations much faster is the GPU parallelization of the solution of the contact dynamics. From the point of view of the results nothing would change, but the time needed in order to solve a system made of more than 780000 particles and more than 1 million contacts would decrease significantly.

Once that these improvements are set, a more complex scenario could be simulated, such as bumpers or a pressurized tank or a solar panel. In addition to fractures and debris clouds, also the acceleration field that a satellite gains after an impact could be studied.

4.2 Summary of asteroid scenario

The asteroid scenario in section 3.2 presents the creation of a gravitational aggregate with cohesive forces and simulates the redirection capabilities of a kinetic impactor with characteristics similar to DART. The simulation behaves well and the outcomes are in agreement with state-of-the-art studies.

4.2.1 Future works

The GPU parallelization of the N-body code or its Barnes-Hut approximation would be a way to increase the number of bodies that take place in the simulation or to decrease their sizes, making a finer continuum approximation. However, the increase of the number of bodies would require a GPU parallelization also for the contact dynamics, in order to not slow down too much the simulation.

The tool can be expanded with collisions between asteroids, both rubble piles and discretized monoliths. This could be applied to asteroid families formation and the collision evolution of the Solar System. The time development of post-impact system is fundamental in the understanding the creation of binary systems or asteroid families. More ambitious would be the simulation of asteroid impacts on planets or moons.

Another extension could be a systematic characterization of the inner structure of the asteroids, pre and post impacts. The mass distribution, the distribution of cohesive forces, the internal void mapping and so on. This could be useful also in the case in which the simulation of a knowns asteroid is meant to be done. If the shape and the mass are known, the discretized asteroid can be recreated. This can be utilized to study the gravitational field in the case of missions that perform flybys or stay in orbit around these irregular bodies.

Bibliography

[1] Holsapple K. A. and Housen K. R. “Momentum transfer in asteroid impacts. I. Theory and scaling”. In: *Icarus* 221 (2012), pp. 875–887.

[2] Anghileri M. et Al. “Development of Orbital Debris Impact Protection Panels”. In: *5th European LS-DYNA Users Conference* Methods and Techniques (4) (2014).

[3] Asphaug E. et Al. “Disruption of kilometre-sized asteroids byenergetic collisions”. In: *Nature* 393 (1998), pp. 437–440.

[4] Ballouz R. L. et Al. “Numerical simulations of collisional disruption of rotating gravitational aggregates: Dependence on material properties”. In: *Planetary and Space Science* 107 (2015), pp. 29–35.

[5] Cheng A. et Al. “AIDA/DART: Double Asteroid Redirection Test”. In: *11th Low Cost Planetary Missions Conference*. 2015.

[6] Cheng A.F. et al. “Asteroid Impact & Deflection Assessment mission: Kinetic impactor”. In: *Planetary and Space Science* 121 (2016), pp. 27–35.

[7] Collins G. S. et Al. “Numerical modelling of impact processes”. In: *Impact Cratering: Processes and Products* (2013), pp. 254–269.

[8] Deller J. F. et Al. “A new approach to modelling impacts on rubble pile asteroid simulants”. In: *Monthly Notices of the Royal Astronomical Society* 455 (2016), pp. 3752–3762.

[9] Durda D. D. et Al. “The formation of asteroid satellites in large impacts: results from numerical simulations”. In: *Icarus* 170 (2004), pp. 243–257.

[10] Ferrari F. et Al. “N-body gravitational and contact dynamics for asteroid aggregation”. In: *Multibody Syst Dyn* 39 (2017), pp. 3–20.

[11] Francesconi A. et Al. “Experimental study of spacecraft material ejected upon hypervelocity impact”. In: (), pp. 175–180.

[12] Genda H. et Al. “Impact erosion model for gravity-dominated planetesimals”. In: *Icarus* 000 (2017), pp. 1–13.

[13] Giacomuzzo C. et Al. “Hypervelocity experiments of impact cratering and catastrophic disruption of targets representative of minor bodies of the Solar System”. In: *Advances in Space Research* 40 (2007), pp. 244–251.

[14] Giacomuzzo C. et Al. “Investigation of impact cratering processes into porous targets through hypervelocity experiment and simulations”. In: (), pp. 175–180.

[15] Jutzi M. et Al. “Modeling asteroid collisions and impact processes”. In: (2015).

[16] Kadau D. et Al. “Contact dynamics simulations of compacting cohesive granular systems”. In: *Computer Physics Communication* 147 (2002), pp. 190–193.

[17] Leinhardt Z. M. et Al. “Direct N-body Simulations of Rubble Pile Collisions”. In: *Icarus* 146 (2000), pp. 133–151.

[18] Liu L. et Al. “Impact breakage of spherical, cuboidal and cylindrical agglomerates”. In: *Powder Technology* 199 (2010), pp. 189–196.

[19] Michel Y. et Al. “Hypervelocity impacts on thin brittle targets: Experimental data and SPH simulations”. In: *International Journal of Impact Engineering* 33 (2006), pp. 441–451.

[20] Otsubo M. et Al. “Empirical assessment of the critical time increment in explicit particulate discrete element method simulations”. In: *Computers and Geotechnics* 86 (2017), pp. 67–79.

[21] Plassard F. et Al. “Hypervelocity impact of aluminium sphere against aluminium plate : experiment and LS-DYNA correlation”. In: *8th European LS-DYNA Users Conference* (2011).

[22] Roig F. et Al. “Interacting ellipsoids: a minimal model for the dynamics of rubble-pile bodies”. In: *Icarus* 165 (2003), pp. 355–370.

[23] Ryan S. et Al. “Numerical simulation of hypervelocity impact on CFRP/Al HC SP spacecraft structures causing penetration and fragment ejection”. In: *International Journal of Impact Engineering* 33 (2006), pp. 703–712.

[24] Steinhauser M. O. et Al. “Impact failure of granular materials ? Non-equilibrium multiscale simulations and high-speed experiments”. In: *International Journal of Plasticity* 25 (2009), pp. 161–182.

[25] Steinhauser M. O. et Al. “Numerical Simulation of Fracture and Failure Dynamics in Brittle Solids”. In: *The 12th Int. Symposium on Plasticity and Its Current Applications* (2006).

[26] Stickle A. M. et Al. “Modeling Momentum Transfer from Kinetic Impacts: Implications for Redirecting Asteroids”. In: *Procedia Engineering* 103 (2015), pp. 577–584.

[27] Tasora A. et Al. “Chrono: An Open Source Multi-physics Dynamics Engine”. In: *High Performance Computing in Science and Engineering* (2016), pp. 19–49.

[28] Tasora A. et Al. *Large-Scale Parallel Multibody Dynamics with Frictional Contact on the Graphical Processing Unit.* Technical Report. University of Parma, University of Winsonsin-Madison, Argonne National Laboratory, 2008.

[29] Yu Y. et Al. “Structural analysis of rubble-pile asteroids applied to collisional evolution”. In: *Astrodynamics* 1 (2017), pp. 57–69.

[30] Zhang Y. et Al. “Creep stability of the proposed AIDA mission target 65803 Didymos: I. Discrete cohesionless granular physics model”. In: ().

[31] M. Anitescu and A. Tasora. “An iterative approach for cone complementary problems for nonsmooth dynamics”. In: *Computational Optimization and applications* 47 (2010), pp. 207–235.

[32] Chung Y. C. and Ooi J. Y. “Benchmark tests for verifying discrete element modelling codes at particle impact level”. In: *Granular Matter* 13 (2011), pp. 643–656.

[33] O’Sullivan C. and Bray J. D. “Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme”. In: *Engineering Computations* 21 (2004), pp. 278–303.

[34] Heyn T. D. “On the modeling, simulation, and visualization of many-body dynamics problems with friction and contact”. PhD in Mechanical Engineering. University of Winsonsin-Madison, 2013.

[35] Kwarta M. andNegrut D. *Using the Complementarity and Penalty Methods for Solving Frictional Contact Problems in Chrono:Validation for the Cone Penetration Test*. Technical Report. University of Winsonsin-Madison, 2017.

[36] Negrut D. and Serban R. *Posing Multibody Dynamics with Friction and Contact as a Differential Complementarity Problem*. Technical Report. University of Winsonsin-Madison, 2017.

[37] Blazquez E. "Numerical Simulation of N-body Asteroid Aggregation Dynamics". MSc in Space Engineering. Politecnico di Milano, 2017.

[38] Korycansky D. G. and Asphaug E. "Low-speed impacts between rubble piles modeled as collections of polyhedra, 2". In: *Icarus* 204 (2009), pp. 316–329.

[39] Piekutowsky A. J. "Debris clouds produced by the hypervelocity impact of nonspherical projectiles". In: *International Journal of Impact Engineering* 26 (2001), pp. 631–624.

[40] Jutzi M. and Michel P. "Hypervelocity impacts on asteroids and momentum transfer I. Numerical simulations using porous targets". In: *Icarus* 229 (2014), pp. 247–253.

[41] Leinhardt Z. M. and Richardson D. C. "N-Body Simulations of Planetesimal Evolution: Effect of Varying Impactor Mass Ratio". In: *Icarus* 159 (2002), pp. 306–313.

[42] Fahrenthold E. P. and Hernandez R. J. "Simulation of orbital debris impact on the Space Shuttle wing leading edge". In: *International Journal of Impact Engineering* 33 (2006), pp. 231–243.

[43] Perrine R. P. and Richardson D. C. “N-body simulations of cohesion in dense planetary rings: A study of cohesion parameters”. In: *Icarus* 219 (2012), pp. 515–533.

[44] Sanchez P. and Scheeres D. J. “The Strength of Regolith and Rubble Pile Asteroids”. In: *Meteoritics and Planetary Science* 2 (2014).

[45] A. Tasora and M. Anitescu. “A convex complementarity approach for simulating large granular flows”. In: *Journal of Computational and Nonlinear Dynamics* 5 (2010), pp. 1–10.

[46] A. Tasora and M. Anitescu. “A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 200 (2011), pp. 439–453.

[47] D. Negrut Tasora A. and M. Anitescu. “Large-scale parallel multi-body dynamics with frictional contact on the graphical processing unit”. In: *Journal of Multi-body Dynamics* 222 (2008), pp. 315–326.

[48] Steinhauser M. O. Watson E. “Discrete Particle Method for Simulating Hypervelocity Impact Phenomena”. In: *Materials* 10.379 (2017).