Understanding Collisions in Planetary Systems by Computational Modeling

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Philip Matthias Winter, M. Sc.

aus Villach / Österreich

Tübingen

2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	24.11.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter	PD Dr. Christoph Schäfer
2. Berichterstatter	Prof. Dr. Martin Oettel

Acknowledgments

I express my deepest gratitude to my first supervisor, Christoph Schäfer, for his relentless support and unwavering motivation throughout this decade-long research endeavor. Since my bachelor's degree, his expertise and encouragement have been instrumental in shaping me into a better researcher. My gratitude also extends to my second supervisor, Martin Oettel, for his profound confidence and trust in my work. He played a pivotal role in bringing this thesis to fruition. I sincerely thank the members of my dissertation committee for their valuable time, expertise, and constructive critiques. Their thoughtful input has significantly enhanced the quality and rigor of this work.

Many thanks to my close collaborators from the University of Vienna: Thomas Maindl, Christoph Burger, Rudolf Dvorak, Elke Pilat-Lohinger, and Birgit Loibnegger. Our shared paths have consistently resulted in positive and constructive progress towards common goals. Gratitude also extends to Samuele Crespi, Mattia Galiazzo, and all the other collaborators whose intellectual contributions and collaborative spirit greatly enhanced this research. Special thanks go to my former colleagues from JKU Linz: Sebastian Lehner, Johannes Kofler, Sepp Hochreiter, Günter Klambauer, Bernhard Schäfl, Johannes Lehner, Johannes Schimunek, Lukas Gruber, and Angela Bitto-Nemling. They played pivotal roles in my scientific career and contributed significantly to shaping me into a machine learning researcher. Working alongside such talented individuals has been a privilege. I acknowledge the lab members from the University of Tübingen for their camaraderie, technical assistance, and shared enthusiasm for discovery. Their presence fostered an inspiring and productive research environment. I owe a debt of gratitude to all three universities for providing an exceptional academic and research environment. The resources and opportunities afforded to me have been invaluable. I am also grateful to the administrative staff of the University of Tübingen for their assistance with logistical and procedural matters. Their dedication ensured the smooth progression of this project.

To my dear friends, I offer heartfelt thanks for their unwavering encouragement, understanding, and moments of levity throughout this journey. Their support has been a constant source of mental strength, and I would not have been able to go on without them. My gratitude especially includes: Max Zimmermann, Nikolaus Sulzenauer, Florian Wiedemann, Fabienne Wautsche, Markus Obernosterer, Frank Radosits, Maria Wimmer, David Major, Anja Rautnig, Margit Schmidt, Markus Ambrosch, Sara Wanek, Eva-Maria Krajnc, Jennifer Zierler, Christina Petschnigg, and Anna Wurzer.

I am profoundly grateful to my family for their love, patience, and belief in me. Their encouragement has been pivotal to my perseverance and success. Unreserved gratitude goes to my mother Isolde Brandner-Winter, who is a cornerstone of everything I do.

Thank you!

Philip Matthias Winter
Tübingen, June 27, 2025

Alles habe ich überwunden und alles erkannt,
An nichts mehr haftend und durch nichts befleckt,
Alles habe ich losgelassen,
Und alles Verlangen ist versiegt,
Erlöst bin ich und frei,
Aus mir selbst heraus habe ich dies erkannt,
Wen sollte ich da meinen Lehrer nennen? - Buddha (Dhammapada, 353)

Abstract

Throughout the formation and evolution of planets, collisions with other celestial objects are common, ranging from micrometeorite impacts to major encounters with other protoplanets. This study investigates the conditions under which such events occur, examining the orbital- and physical properties of pre- and post-collision bodies and analyzing the intricate dynamics involved in these interactions. We begin by shortly summarizing key principles of planet formation, as well as introducing the computational methodologies employed in this work. Our astrophysical contributions include an analysis of minor impacts on Jupiter's icy moons, Ganymede and Callisto, an evaluation of post-collision fragments from pairwise planetary collisions, and the compilation of a comprehensive collision catalog spanning a broad range of plausible initial conditions. Methodologically, we advance the field by developing and refining numerical simulation techniques for collision modeling and enhancing large-scale data analysis through the application of modern machine learning approaches. Our findings support established theories of planet formation. We provide refined estimates of impact velocities onto Jupiter's icy moons and study the trajectories of post-collision fragments after major collisions. Utilizing our numerical simulation framework, we conduct extensive parameter studies of planetary collisions. The resulting data is then employed to train modern machine learning methods for fast and accurate collision handling. We conclude that combining numerical simulations with machine learning opens up new pathways in theoretical astrophysics, representing a scalable and versatile framework for elucidating the intricacies of planet formation.

Zusammenfassung

Während der Entstehung und Entwicklung von Planeten kommt es häufig zu Kollisionen mit anderen Himmelsobjekten, die von Mikrometeoriteneinschlägen bis hin zu größeren Zusammenstößen mit anderen Protoplaneten reichen. Diese Arbeit untersucht die Bedingungen, unter denen solche Ereignisse stattfinden, indem sie die orbitalen und physikalischen Eigenschaften von Körpern vor und nach der Kollision untersucht und die komplizierte Dynamik analysiert, die an diesen Wechselwirkungen beteiligt ist. Wir beginnen mit einer kurzen Zusammenfassung der wichtigsten Prinzipien der Planetenentstehung und stellen die in dieser Arbeit verwendeten Methoden vor. Unsere astrophysikalischen Beiträge umfassen eine Analyse kleinerer Einschläge auf den Jupitermonden Ganymed und Kallisto, eine Auswertung von Fragmenten aus Planetenkollisionen und die Zusammenstellung eines umfassenden Kollisionskatalogs, der ein breites Spektrum plausibler Anfangsbedingungen umfasst. Unsere methodischen Beiträge umfassen die Entwicklung und Verbesserung von numerischen Simulationstechniken für die Kollisionsmodellierung und die Analyse von großen Datenmengen durch die Anwendung moderner maschineller Lernverfahren. Unsere Ergebnisse stützen etablierte Theorien der Planetenbildung. Wir liefern verfeinerte Schätzungen der Einschlaggeschwindigkeiten auf Jupiters Eismonden und untersuchen die Flugbahnen von Fragmenten nach größeren Kollisionen. Mithilfe unser eigens entwickelten Simulationsmethode führen wir umfangreiche Parameterstudien von Planetenkollisionen durch. Die daraus resultierenden Daten werden verwendet, um moderne maschinelle Lernmethoden für eine schnelle und genaue Kollisionsbehandlung zu trainieren. Wir kommen zu dem Schluss, dass die Kombination von numerischen Simulationen mit maschinellem Lernen neue Wege in der theoretischen Astrophysik eröffnet und einen skalierbaren und vielseitigen Rahmen für die Erforschung der Feinheiten der Planetenbildung darstellt.

Contents

List of Figures

Notation

Acronyms & Abbreviations

ANN	artificial neural network
AU	astronomical units
byr	billion years
cm	centimeters
d	diameter
DL	deep learning
e.g.	for example (*exempli gratia*)
etc.	and so on (*et cetera*)
FFN	feed-forward neural network
GA	genetic algorithm
GPU	graphics processing unit
i.e.	that is (*id est*)
K	Kelvin
kg	kilograms
km	kilometers
km/s	kilometers per second
ly	light years
m	meters
$M_\odot$	solar masses
ML	machine learning
ODE	ordinary differential equation
pc	parsec
PIM	perfect inelastic merging
ρ	density
SPH	smooth particle hydrodynamics
t	time
T	temperature
w.r.t.	with respect to

Outline

This thesis is organized in two parts.

Part I introduces the overall topic and describes related problems addressed in this work (Chapter 1). This is followed by a chapter that introduces concepts for modeling astrodynamical processes via numerical simulations and analyzing complex, high-dimensional data via machine learning (Chapter 2). I provide important contextual information essential for setting up the second part.

Part II contains all major and minor contributions of this thesis. Specifically, it comprises two major contributions (Chapters 3 and 6) and two minor contributions (Chapters 4 and 5), which investigate various aspects of collisions in planetary systems. All contributions have been published in well-established scientific journals in astrophysics.

This thesis particularly focuses on problems related to the computational modeling of collision processes and examines possible solutions in greater detail. Additionally, I briefly describe the relevance of the individual contributions and their interrelations. Finally, the thesis concludes in Chapter 7, where the most significant findings are discussed in a broader context.

Part I.

Motivation & Background

1. Introduction

What shapes the destiny of a planet? What critical moments in the life of a celestial body determine its trajectory into the future? For centuries, humanity held the belief that celestial bodies were immutable, eternal entities, existing in an unchanging state since the universe's earliest moments. This perspective is understandable, as the sheer scale of a planet's lifespan far outreaches the lifespan of any human being. Consider our own Earth, estimated to be approximately 4.55 byr old [75], an age so vast that it challenges the limits of human comprehension. Such an immense timescale might naturally lead one to assume that planets endure in a state of perpetual stability, untouched by the passage of time.

However, in recent decades, new discoveries and a substantial amount of evidence have overturned these long-held assumptions, revealing a far more chaotic, dynamic, and violent history for celestial bodies than previously imagined [77]. Far from being static, the life of a planet is punctuated by a series of rare but cataclysmic events – moments so brief in the context of a planet's existence that they represent merely a blink of an eye in the bigger picture, yet so profound in their impact that they can reshape the course of its entire history. These events, known as *collisions*, wield extraordinary power. They possess the capacity to erase billions of lives in an instant, dramatically tilt the rotational axis of giant planets [58], or even give rise to the formation of entirely new worlds [59, 60]. These cosmic encounters are defining milestones that leave lasting marks on the evolution of planets and planetary systems.

This thesis is dedicated to exploring the nature and consequences of such collisions. It seeks to shed light onto the underlying mechanisms, thereby challenging our current approaches and understanding of how to recreate and model these processes with modern computational techniques.

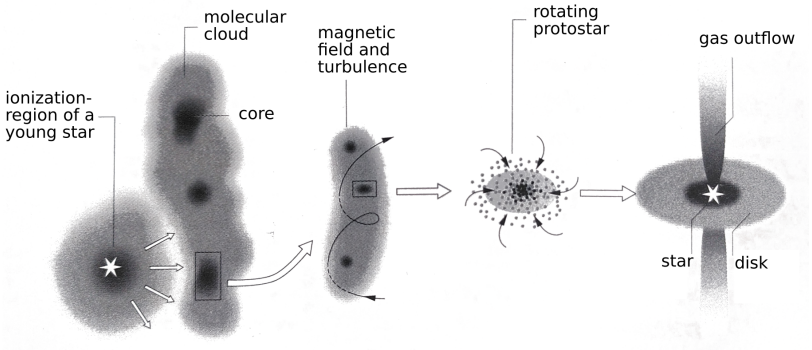


Figure 1.1.: Early stages of star formation [125]. Nearby stellar winds or supernova explosions induce efficient cooling of molecular clouds, thereby leading to the formation of dense cores. Through various phenomena such as magnetic fields and turbulence, cores collapse gravitationally and form a rotating disk with a protostar in its center. Bipolar gas outflows origin from the newly formed star, further taking away angular momentum from the disk.

1.1. Star Formation

To understand the significance of collisions within planetary systems, it is essential to first establish a foundational understanding of the processes governing planet formation and evolution, as detailed in several studies [76, 112, 124, 5]. In this context, the formation of planets is closely linked to the formation of their host stars, necessitating a joint analysis of these processes. The temporal, spatial, and compositional characteristics of planets are strongly influenced by the conditions under which their parent stars form. Typically, star formation occurs within gas-rich environments, such as those found in disk galaxies or irregular galaxies.

More precisely, stars emerge in clustered groups originating from large molecular clouds. These clouds have dimensions on the order of $d \simeq 50\,\mathrm{pc}$, masses around $M \simeq 10^5\,\mathrm{M}_\odot$, relatively low densities of $\rho \simeq 10^2\,\mathrm{cm}^{-3}$, as well as a very low temperature of $T \simeq 15\,\mathrm{K}$ (see Figure 1.1 for a schematic representation of the formation process). The nearest star formation region is located in the direction of the Orion constellation, situated at a distance of approximately 1000 – 1400 ly from Earth.

Star formation can be initiated through various astrophysical processes, including the influence of stellar winds or supernova explosions originating from nearby stars. These events trigger the compression of adjacent molecular clouds, facilitating efficient cooling and thereby supporting conditions leading to star formation. Within molecular clouds, a chaotic interplay of several physical processes governs the dynamics leading to the formation of dense gas and dust cores, characterized by specific physical parameters: diameters between

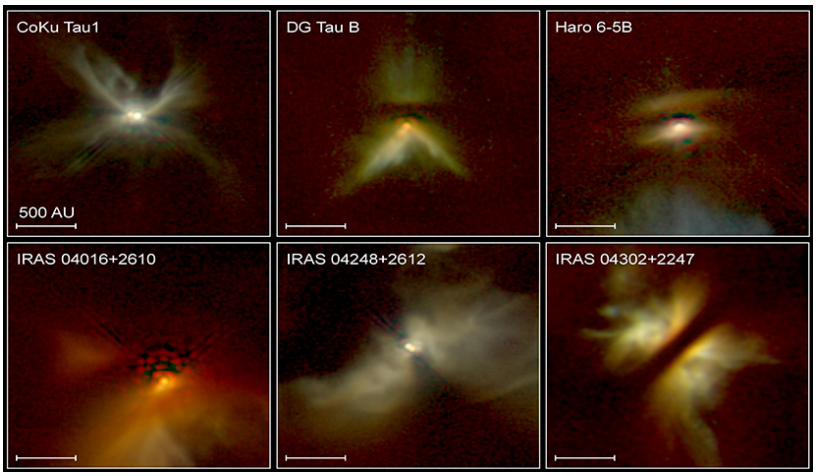


Figure 1.2.: Images of circumstellar disks in infrared (dark components), taken by Hubble space telescope in 1999 [50]. Bright regions above and below the disks are illuminated by the young stars.

$d \simeq 0.1-1$ pc, masses between $M \simeq 100-1000\ \mathrm{M}_\odot$, densities exceeding $\rho > 10^5\ \mathrm{cm}^{-3}$, and temperatures typically ranging between $T \simeq 10-30$ K.

The most relevant forces at play include gravity, which drives contraction, as well as stabilizing factors such as thermal pressure and magnetic fields, which resist contraction. Additionally, turbulence within the cloud can act as both a contracting and stabilizing mechanism, depending on its location, scale, and intensity.

Magnetic fields and turbulence contribute to the formation of dense cores by facilitating the dissipation of angular momentum to surrounding regions, thus enabling the concentration of material necessary for core formation. These dense cores serve as the precursors to stellar and planetary systems. However, for such systems to form, the cores must undergo gravitational collapse. The likelihood of this collapse can be evaluated using the *Jeans criterion*, which provides a theoretical framework for determining the conditions under which a cloud or core becomes gravitationally unstable:

$$d_J = 0.19 \left(\frac{T}{10\ \mathrm{K}}\right)^{1/2} \left(\frac{n_{H_2}}{10^4\ \mathrm{cm}^{-3}}\right)^{-1/2} \mathrm{pc} \tag{1.1}$$

$$M_J = \left(\frac{T}{10\ \mathrm{K}}\right)^{3/2} \left(\frac{n_{H_2}}{10^4\ \mathrm{cm}^{-3}}\right)^{-1/2} \mathrm{M}_\odot \tag{1.2}$$

Here, d_J and M_J are the upper bounds for diameter and the corresponding mass of a core, above which it becomes unstable and collapses. The average count density of H_2 molecules per unit volume in a core is n_{H_2}.

In accordance with the law of conservation of angular momentum, the collapse of a rotating molecular cloud core initiates an increase in rotational velocity as the core contracts, resulting in the formation of a flattened, disk-like structure, commonly referred to as a protoplanetary disk (see Figure 1.2). Within this disk, the region of highest density undergoes further gravitational collapse, leading to the formation of at least one protostar. This protostar subsequently accretes a portion of the surrounding disk material, facilitating further growth over time. Nuclear fusion ignites within the protostar as soon as its core attains sufficient temperature and density.

In contrast to the innermost region of the protoplanetary disk, its outer regions experience additional gravitational instabilities, which drive the coagulation and accumulation of material, ultimately leading to the formation of planetary bodies (see discussion below).

The angular momentum of the system is diverted away through multiple mechanisms. One such mechanism involves the ejection of material through bipolar gas outflows, which are propelled by magnetic activity and stellar winds induced by the newly formed protostar [11, 109]. Alternatively, angular momentum may be transferred into the orbital motion of planets formed within the disk. For example, although the masses of our Jovian planets (Jupiter, Saturn, Uranus, and Neptune) are negligible *w.r.t.* the Sun, these planets contain 98% of the Solar System's total angular momentum distribution through their orbital motions [72]. In multi-star systems, a significant portion of angular momentum is retained within the orbits of stars themselves as they revolve around the system's barycenter.

Circumstellar disks, also referred to as protoplanetary disks, primarily consist of molecular hydrogen (H_2) gas and microscopic dust particles. These disks serve as the reservoirs from which protostars accrete material to facilitate their growth. However, for accretion to proceed effectively, the rotational angular momentum of the protostar must be transported away to prevent rotational breakup. In disk regions where gas is sufficiently ionized, particularly near the protostar, angular momentum can be redistributed outward through magneto-rotational instabilities [120]. Nevertheless, the bulk of the disk material remains non-ionized, necessitating alternative mechanisms for angular momentum transport.

Due to Keplerian motion (also see Section 2.1), the angular velocity of material within the disk is predominantly determined by its radial distance from the central protostar, resulting in a differentially rotating disk. This differential rotation induces shear flows between adjacent regions of the disk, which in turn perturb its dynamical stability. The resulting shear-driven interactions give rise to gas-friction and turbulence [86, 87, 88]. These processes collectively contribute to the complex dynamical evolution of protoplanetary disks, affecting their capacity to support accretion onto the protostar, as well as facilitating planet formation inside the disk.

1.2. Planet Formation

The dynamics of protoplanetary disks play a critical role in the early stages of planet formation, as evidenced by various studies [68, 95, 22, 80]. The process begins with the coagulation of microscopic dust particles, where tiny grains, typically on the order of $d \simeq 10^{-4}$ cm, adhere to one another through mutual interactions driven by van der Waals forces, electrostatic attractions, or other adhesive mechanisms. Since dust particles are typically irregular and sharp (see Figure 1.3), they easily stick to each another.

This coagulation phase, occurring over timescales of approximately $t \simeq 10^3 - 10^4$ yr, results in the formation of larger aggregates, ranging from millimeter- to decimeter-sized pebbles. These pebbles, while still subject to friction with the surrounding gas in the disk, mark a significant step in the hierarchical growth of solid bodies.

As the size of these aggregates increases, their dynamical behavior transitions. Larger objects, typically exceeding the decimeter scale, begin to follow Keplerian orbits more closely due to their increased mass and reduced surface-to-mass ratio. Consequently, these bodies become less prone to perturbations caused by gas drag, friction, and turbulent motions within the disk. This shift in dynamics reduces the efficiency of further growth through simple coagulation, necessitating alternative mechanisms to facilitate the formation of larger bodies.

Further growth involves revisiting gravitational instabilities: Small deviations from the local disk density are induced

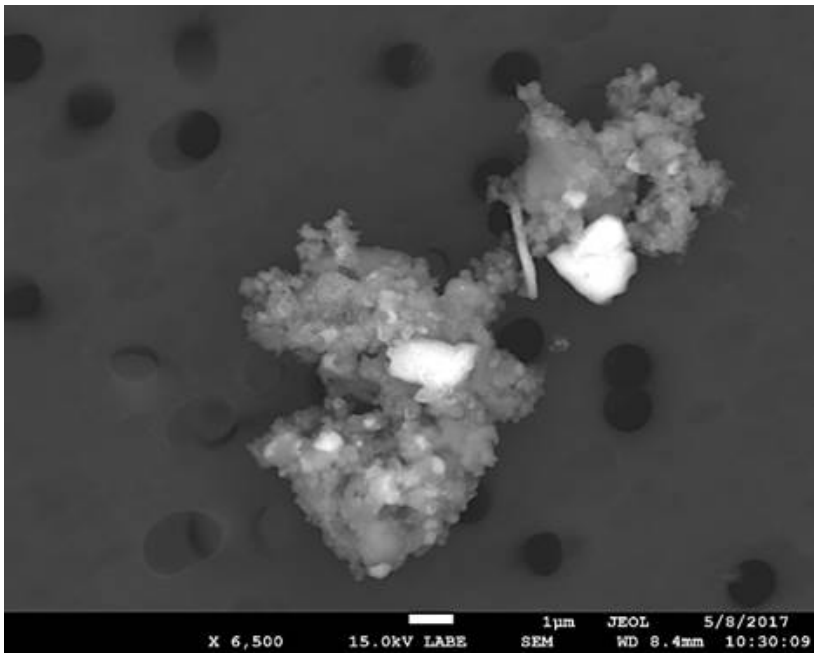


Figure 1.3.: Microscopy image of a cosmic dust particle [82]. Its irregular shape makes it easily stick to other particles.

by turbulent concentration and other hydrodynamic processes. These density perturbations can become gravitationally unstable and trigger gravitational collapse. Unstable regions then initiate a self-reinforcing feedback loop, attracting additional disk material, both solid and gaseous, through gravitational forces. This process, unfolding over approximately $t \simeq 10^5$ yr, results in the formation of planetesimals, which are solid bodies with diameters ranging from $d \simeq 10^8 - 10^9$ cm.

The emergence of planetesimals represents another pivotal transition in disk evolution, as the disk begins to fragment into many individual, discrete objects. The subsequent evolution of these planetesimals drives further growth through mutual collisions and accretion. Over timescales of $t \simeq 10^5 - 10^6$ yr, repeated collisions between planetesimals lead to the formation of planetary embryos, which are larger bodies with diameters on the order of $d \simeq 10^8 - 10^9$ cm.

These embryos serve as the precursors to terrestrial planets and the cores of gas giants, depending on their location within the disk and the availability of gaseous material. The collisional growth process is influenced by a variety of factors, including the relative velocities of planetesimals, the disk's dynamical state, and the surface density of solid material. In addition, orbital resonances start to act as stabilizing or destabilizing factors (see details below). This complex interplay of physical processes ultimately shapes the architecture of planetary systems, setting the stage for further evolution

through dynamical interactions and accretion of residual disk material.

Planetary embryos experience substantial heating driven by radioactive decay of short-lived and long-lived isotopes, tidal interactions with nearby massive bodies, accretion of material during their formation, and energetic collisions with other celestial bodies [41, 136, 67, 26, 106, 141, 20]. These mechanisms collectively induce significant temperature increases within the embryos, leading to partial melting of their constituent materials, particularly within their cores [83]. Partial melting usually goes hand in hand with internal differentiation into layers [118, 1, 38, 139]. This differentiation process is primarily characterized by the segregation of heavy materials which migrate towards the core and light materials which migrate towards the surface.

Furthermore, the self-gravitational forces acting on these bodies are sufficiently strong to overcome the mechanical strength of materials. This facilitates a morphological transition from irregular, asymmetric shapes into more spherical configurations. Large objects with spherical shapes are in a so-called *hydrostatic equilibrium* [19, 52], *i.e.*, when compressive forces of self-gravity balance out internal pressure gradients (also see Figure 1.4).

Considering the volume of a spherical shell $4\pi r^2 dr$ with a thickness of dr, as well as its density $\rho(r)$ at radius r, the total mass of a sphere at radius r is

$$M_r = \int_0^r \rho(r)\, 4\pi r^2 dr. \tag{1.3}$$

Further, for a volume element at depth r and a height of dr, as well as considering $1m^2$ of its surface, its pressure gradient is then

$$\frac{dP}{dr} = -\rho(r)\frac{GM_r}{4\pi r^2}, \tag{1.4}$$

where G is the gravitational constant. This means that the change of pressure is equivalent to the weight of the volume element, thereby effectively balancing out each other. This equilibrium state is a major milestone in the formation of planets, marking their progression to complex inner structures.

Figure 1.4.: Image of asteroid Vesta, taken by the Dawn spacecraft from a distance of about 5200 km [55]. With a diameter of $d = 525$ km [103], Vesta possesses sufficient mass for its self-gravity to overcome rigid body forces, resulting in a more spherical shape *w.r.t.* other asteroids, as well as approaching hydrostatic equilibrium.

The runaway effect of gravitational instabilities significantly influences the accretion dynamics of planetary embryos, enabling earlier embryos to amass substantially more material compared to those forming later in the protoplanetary disk's evolution. This offset accretion process results in the formation of planets with diverse masses and properties, including gas giants, smaller terrestrial planets, as well as countless minor bodies such as dwarf planets, asteroids, and comets. Unlike terrestrial planets, which primarily accrete solid material or lose their gaseous components due to strong solar winds from a nearby parent star, gas giants undergo an additional phase of growth, where they attract and accrete a large amount of gas from the disk [91]. This gas accretion phase occurs over a timescale of about $t \simeq 10^5 - 10^6$ yr, leading to planet diameters of about $d \simeq 10^{10}$ cm.

The interaction between a protoplanet and the protoplanetary disk facilitates a mutual exchange of angular momentum, which drives the planet's migration towards their parent star [71, 127, 85]. As the disk material in the orbital vicinity of a planet depletes (see Figure 1.5), the accretion process largely ceases, effectively halting the planet's growth. However, the complexity of planet formation arises from the interplay of a multitude of interacting processes, necessitating a comprehensive approach to model these phenomena jointly [122].

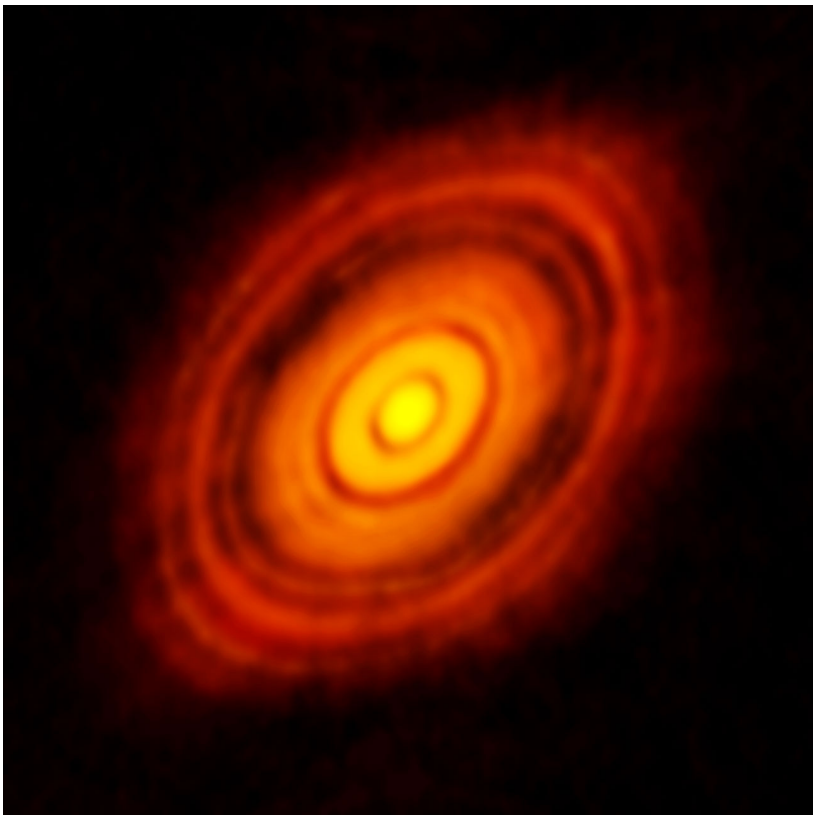


Figure 1.5.: ALMA observation of young star *HL Tauri* [89, 3]. Gaps within the disk indicate ongoing planet formation through accumulation of local disk material. Credit: ALMA (ESO/NAOJ/NRAO)

1.3. Planet Evolution

The orientation of a circumstellar disk plays a pivotal role in determining the dynamical properties of planets formed therefrom. These properties include both the shape and orientation of planetary orbits, as well as the orientation of intrinsic angular momentum, *i.e.*, the planets' rotation axes. However, these orbital parameters are not static. They undergo significant modifications over extended timescales due to gravitational interactions among celestial bodies [116, 126]. Such interactions are particularly pronounced in the presence of orbital resonances, *i.e.*, when massive orbiting bodies exert periodic and recurring gravitational influences on each another. These resonance can lead to complex dynamical evolution, including migration towards or away from their parent star.

The most common form of resonance is the mean motion resonance (MMR), which occurs when two or more celestial bodies have orbital periods that are simple integer ratios of each other. For example, Earth and Venus are in a near 13:8 MMR, while Jupiter and Saturn are in a near 5:2 MMR. Resonances can have either stabilizing or destabilizing effects on the orbits of the involved bodies, depending on the specific

ratio of their orbital periods. Stabilizing resonances, such as 1:1, 3:2, 4:3, or 4:2:1, tend to enhance orbital stability by locking the bodies into predictable, repeating configurations that mitigate chaotic perturbations. Conversely, destabilizing resonances, such as 2:1 or 3:1, can induce significant orbital perturbations, typically leading to changes of orbital parameters (also see Section 2.1), migration, or even ejection from the system. These resonant interactions are critical to understanding the long-term evolution of planetary systems.

In general, stellar and planetary systems comprising more than two interacting bodies exhibit chaotic dynamics [81, 27, 107, 30]. This chaotic behavior can lead to significant orbital perturbations over time. In extreme scenarios, such interactions may result in the ejection of celestial bodies from the system, where one or more bodies achieve escape velocity, often facilitated by close encounters with other planets. In contrast, certain configurations, such as our own Solar System, demonstrate quasi-stable behavior over extended periods of time despite underlying chaotic tendencies. This relative stability is attributed to specific orbital architectures, especially resonant configurations that can mitigate chaotic divergence, as seen in systems like the 5:2 MMR between Jupiter and Saturn.

To investigate the orbital evolution and chaotic dynamics of such systems, researchers commonly use N-body numerical simulations (see 2.1). These simulations model the spatial trajectory of multiple bodies over time, providing insights into the long-term evolution of their orbits and the emergence of chaotic behavior. A widely adopted method for quantifying the degree of chaos in these dynamical systems is the analysis of *Lyapunov Characteristic Exponents* (LCEs) [9]. LCEs measure the rate of exponential divergence of initially nearby trajectories in phase space, offering a robust metric for assessing the stochastic properties of a system under small perturbations in initial conditions. A positive LCE indicates chaotic behavior, with the largest exponent determining the timescale of predictability loss, known as the Lyapunov time. By integrating N-body simulations with LCE analysis, the dynamical stability and chaotic evolution of such systems can be characterized in a comprehensive manner.

Unstable orbital dynamics often lead to significant reconfiguration of a planet's orbit, fundamentally altering its trajectory and position within the system. Its parent star, however, not only affects the orbital path, but also the physical properties of a planet's surface and interior. Factors such as a

planet's distance from the star, its orbital eccentricity, and its rotational characteristics – including the orientation of its rotational axis and its rotational speed – play critical roles in determining surface conditions. For example, orbital instability can cause dramatic variations in surface temperature, atmospheric composition, and other environmental factors, as the planet's exposure to stellar radiation fluctuates. These changes can have profound implications for the planet's habitability (*e.g.*, through effective freezing or vaporization of oceans) and geological activity (*e.g.*, through strong tidal heating), highlighting the intricate interactions between orbital dynamics and stellar influence in shaping planetary systems.

1.4. Problems regarding Collisions

In planetary systems, collisions between celestial bodies can occur at any scale and at any time [42]. This includes a diverse spectrum of events, from low-energy collisions involving microscopic dust particles to intermediate-scale impacts of micrometeorites and small bodies onto planetesimals, up to high-energy, catastrophic collisions between protoplanets. Recently, even collisions between man-made objects and asteroids are becoming more relevant in the context of planetary defense and future asteroid mining operations [53].

Collision processes play a fundamental role in the dynamical evolution and physical development of planetary systems, strongly affecting the formation, growth, and eventual stabilization of planetary bodies across cosmic timescales. Thus, collisions turn planet formation into a fundamentally chaotic and violent process.

Let us now consider a pairwise collision between two distinct objects, denoted as i and j, with their respective masses masses m_i and m_j. Under the assumption of $m_j < m_i$, we define object j as the *projectile* and object i as the *target*. This definition facilitates a clear naming convention to differentiate between objects.

At any given point in time t, both objects occupy a finite volume within three-dimensional Euclidean space. The objects may be composed of a diverse range of materials, which can exist in various physical states, including solid (*e.g.*, ice,

metals, as well as rocky materials like basalt or silicate), liquid (*e.g.*, water or magma), gaseous (*e.g.*, atmospheric gases or volatile ices), and plasma (*e.g.*, auroras, ionospheres, or lightning-induced plasma).

The kinematic and dynamic properties of both objects are defined *w.r.t.* their respective barycenters, which serve as reference points for their motion and physical characteristics. Specifically, each object is described by its position vector $\vec{x}(t)$, which locates the barycenter in three-dimensional space, as well as its respective velocity vector $\vec{v}(t)$ describing translational motion. In addition, each object has a rotational angular momentum vector $\vec{L}(t)$, which defines the orientation and magnitude of its rotation around an axis passing through the barycenter. These properties are critical for accurately modeling the pre-collision, collision, and post-collision dynamics of the system.

The overall impact geometry is determined by $\vec{x}_i$, $\vec{x}_j$, $\vec{v}_i$, $\vec{v}_j$ at the time of contact $t = t_c$. With a relative barycenter distance of $r_{ji} = |\vec{x}_i - \vec{x}_j|$, the impact velocity v_{imp} and the impact angle α_{imp} are defined as

$$v_{imp} = |\vec{v}_i - \vec{v}_j|_{t=t_c} \tag{1.5}$$

$$\alpha_{imp} = \cos^{-1}\left(\frac{\vec{r_{ij}} \cdot \vec{v}_{imp}}{r_{ij} \cdot v_{imp}}\right)_{t=t_c} . \tag{1.6}$$

Therefore, a head-on collision has $\alpha_{imp} = 0°$ while a grazing collision has $\alpha_{imp} = 90°$ (also see Figure 3 in Chapter 6). Note that significant deformations can already happen at $t < t_c$ due to tidal interactions. This is especially true for large or brittle bodies.

In the reference frame of object i, the total kinetic energy of the projectile is

$$E_{kin} = \frac{m_j \cdot v_{imp}^2}{2} . \tag{1.7}$$

Thus, the impact velocity has a much stronger influence on the overall collision outcome compared to the projectile mass. During a collision, this kinetic energy is released, leading to various effects such as strong deformation, compression, fragmentation, ejection, melting, and explosive vaporization.

In general, the temporal evolution of colliding bodies is governed by the principles of classical mechanics, including conservation laws for momentum and angular momentum, as well as the physical interactions between the objects, which may involve elastic, plastic, or inelastic interactions depending on material properties and collision parameters.

Moreover, the objects' internal structure (*e.g.*, differentiated vs. homogeneous or dense vs. light) strongly affects the collision process. For example, a solid object such as a rocky asteroid may exhibit brittle fracture or plastic deformation (*e.g.*, through porosity), whereas gaseous or fluid components such as an atmosphere or a liquid core may largely behave like a Newtonian fluid. Involved materials, as well as their spatial distribution, significantly influence mechanical and dynamical properties during the collision process.

Minor meteorite impacts on Earth are common, with about 50 tons/yr of daily influx [140]. Most meteorites are some millimeters in size, whereas meter-sized ones are far less common. Meteorites usually origin from the main asteroid belt or from debris of comets. After entering Earth's atmosphere at hypersonic velocities, they encounter strong atmospheric resistance, which causes intense frictional heating and ablation. Depending on their composition, this typically results in colored, bright fireballs. Frictional heating can also lead to breakup or explosion before reaching the ground. As iron meteorites are more stable, this effect specifically applies to stony and carbonaceous impactors.

A prominent example for atmospheric disruption and explosion occurred in Tunguska, Central Siberia, in 1908 [21], where a stony asteroid with tens of meters in radius released about 10-20 megatons of energy at an altitude of about 10 km. Despite extensive search, no large fragments have been found. Recently, the formation of Lake Cheko was identified as a potential crater candidate originating from a possible secondary impact [35]. The lake is located about 8 km NNW of the explosion epicenter.

In general, however, minor impacts with $m_j << m_i$ typically result in the formation of craters, which are commonly observed in our Solar System. Crater formation can be roughly divided into three distinct phases: contact & compression, excavation, and modification [77]. Depending on their size and the presence of geological activity, craters can remain in simple, bowl-like shapes or develop complex, multi-component basins during the modification stage (see Figure 1.6).

Figure 1.6.: Left: Barringer Crater in Arizona, USA [119]. It has a diameter of 1.19 km and is about 49000 yr old. The iron-rich impactor had a diameter of about 45 m, a mass of about 3×10^8 kg, and an impact velocity of $1.4 - 2.8\,[v_{esc}]$ [108]. **Right:** Valhalla crater on Callisto, Jupiter's icy moon [129]. The image was taken by Voyager 1 in 1979. While the impact basin measures about 600 km in diameter, the multi-ring structure extends 2600 km across the surface of the moon. Depending on its mass, the projectile had an impact velocity of $12.5 - 63.3\,[v_{esc}]$ (details provided in Chapter 3). **Bottom:** Earth's Moon is covered with countless craters in varying sizes [114]. Their density is an indicator for the geological age of the surface.

Major collisions, on the other hand, can play a critical role in setting the evolutionary path of an object. For example, there is evidence which supports that Earth's Moon was created by a major collision between the young Earth and a Mars-sized impactor [7, 137]. Also, recent studies suggest that Uranus' rotation axis was strongly tilted *w.r.t.* the orbital plane due to one or two major collisions with earth-sized objects [58, 98]. In general, outcomes of major collisions include the potential for fragmentation, merging, and ejection of material from both projectile and target.

Considering the observational evidence of collisions in our Solar System, it is imperative to determine the most effective methodologies and experimental setups for studying these phenomena in general. While small-scale collisions can be studied in the laboratory [49, 48, 115, 61, 25] (also see Figure 1.7), this option is not available in case of large-scale collisions due to several reasons. Major collisions between protoplanets are rare events, involve various complex physical interactions, and usually happen during the relatively short period of planet formation.

From an observational perspective, major collisions can also not be studied directly. These events are usually hidden within a thick disk of gas and dust (see Figure 1.2), which negates direct observation at various wavelengths. Another key limitation in observing planetary collisions is the vast

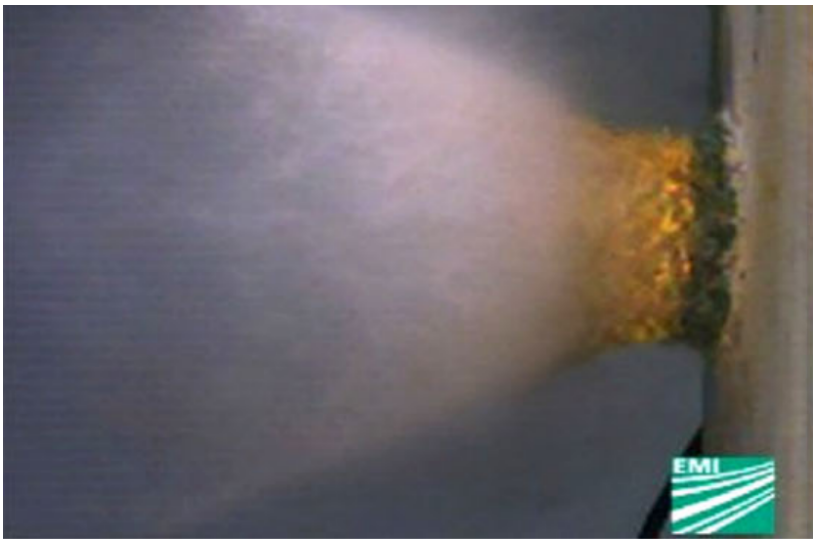


Figure 1.7.: Laboratory experiment of a collision [62, 61] using a two-stage light-gas gun facility. The impact leads to the formation of a crater of about $d \simeq 10$ cm in diameter. The image depicts the ejecta cloud shortly after the impact.

distance to these events combined with the limited angular resolution of current telescopes, which prevents resolving fine details necessary to detect and study such collisions.

In contrast, computational modeling turned out to be a viable approach for studying various types of collision events, as well as their influence on planet formation and long-term planetary evolution [10, 18, 6, 113, 37, 44]. More concretely, collisions can be modeled by the means of both N-body simulations and hydrodynamics simulations (Chapter 2).

In addition to classical modeling approaches, data-driven methods employing machine learning (ML, [78]) are experiencing a renewed surge of interest within various scientific disciplines (Section 2.3). ML provides powerful tools for processing large, complex datasets in a fast and efficient manner while at the same time maintaining high predictive performance. These methods are particularly suited for handling high-dimensional data, which is common in theoretical astrophysics due to the intricate nature of simulated datasets. ML techniques excel in detecting and exploiting subtle patterns and correlations within such datasets, thereby speeding up complex calculations that are often intractable using traditional analytical approaches.

Among the various ML methodologies, deep learning (DL), which leverages artificial neural networks [51], has recently emerged as a particularly promising approach [2, 69]. DL is well-suited for tackling high-dimensional, non-linear prediction tasks, such as modeling dynamical systems or reconstructing physical processes from incomplete or noisy data. The growing adoption of ML and DL underscores their

transformative potential in overcoming computational and analytical challenges inherent in modern astrophysics.

Given the tools for modeling the dynamical processes of minor and major collisions, the scope of this work is to address the following problems:

- **Problem A**: What happens before a collision? What are the initial conditions that lead to a certain outcome?
- **Problem B**: What happens during a collision? How to model the complex physical interactions between gravity and material-specific effects?
- **Problem C**: What happens after a collision? How do the resulting bodies evolve over time?

Solving these problems is by no means a trivial task. Moreover, applied methods often raise significant challenges themselves, thereby introducing an additional layer of complexity.

Computational Methods 2.

2.1. n-Body Simulations

The precise prediction of celestial body movements has historically presented a great challenge for astronomers and mathematicians. For centuries, the complex dynamics of planetary motion was poorly understood, with early models relying heavily on empirical observations and rudimentary geometric approximations. Only through the groundbreaking contributions of Johannes Kepler in the early 17th century, a robust framework, that was grounded in rigorous mathematical principles, was established. This framework allowed to accurately predict the positions of celestial bodies over time. Kepler's seminal work, building upon the heliocentric model proposed by Nicolaus Copernicus, marked a pivotal advancement in astronomy by introducing a set of three fundamental laws that describe the motion of planets in their orbits around the Sun.

1st Law (The Law of Ellipses): Kepler's first law revolutionized the understanding of planetary orbits by asserting that the orbit of every planet is an ellipse, with the host star located at one of its two foci. The mathematical description of an elliptical orbit is given by the polar equation:

$$r = \frac{p}{1 + \varepsilon \cdot cos(\theta)} \,. \tag{2.1}$$

In this equation, r represents the radial distance from the star to the planet, p denotes the semi-latus rectum (a parameter related to the size of the ellipse), ε is the eccentricity of the ellipse (quantifying the degree of elongation, with $\varepsilon = 0$ being a circle and $\varepsilon < 1$ being an ellipse), and θ is the true anomaly, the angle measured from the periastron (the point of closest approach to the star) to the planet's current position in its orbit. This formulation allows for precise calculations of a planet's position at any given time, accounting for the elliptical nature of its trajectory.

Kepler's first law represented a fundamental departure from the previously assumed circular orbits. More specifically,

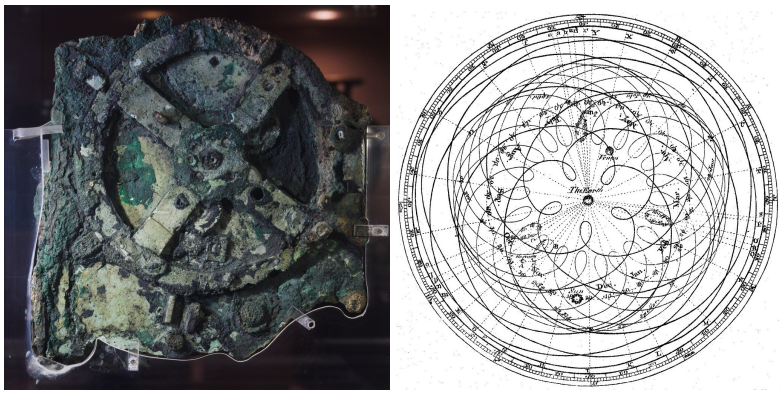


Figure 2.1.: Left: The Antikythera mechanism was an analog computer, consisting of a complex system of wheels and plates [23]. Dating back to about 80 BC, the mechanism represents both masterful and high-precision engineering, but also fundamentally wrong theories originating from the Geocentric model. **Right:** Apparent motion of the Sun, Mercury, and Venus as observed from Earth under the Geocentric model [29]. The complex motions, including recurrent retrograde phases, were usually described via epicycles.

the prominent *Epicycle theory* states that planets move along circles, which themselves move along other circles. Although fundamentally wrong, stacking of many circles in this manner allow for crude approximation of planetary motions. Specifically, epicycles allow for describing recurrent retrograde motion of planets. These motions, however, were merely an artefact resulting from the Geocentric model, which persisted for about 2000 years.

An early example of the deep belief into epicycles is the so-called *Antikythera mechanism* [92, 32] (see Figure 2.1). Although masterfully crafted and highly accurate, it relied on the Geocentric model and epicycles to describe planetary motions. Today, the mechanism is considered as the earliest analog computer in existence.

In contrast, Kepler introduced a physically sound description of observed planetary paths. Kepler's first law not only provided a geometric foundation for understanding planetary motion, but also laid the groundwork for subsequent laws that further refined the predictive power of celestial mechanics.

2nd Law (The Law of Equal Areas): Kepler's second law states that a line segment connecting a planet to its host star, known as the radius vector, sweeps out equal areas in equal intervals of time as the planet traverses its elliptical orbit (see Figure 2.2). This principle implies that a planet moves faster when it is closer to the star (*i.e.*, near the periastron) and slower when it is farther away (*i.e.*, near the apastron), ensuring that the areal velocity remains constant throughout the orbit. This law reflects the dynamic interplay between the planet's position and its orbital speed.

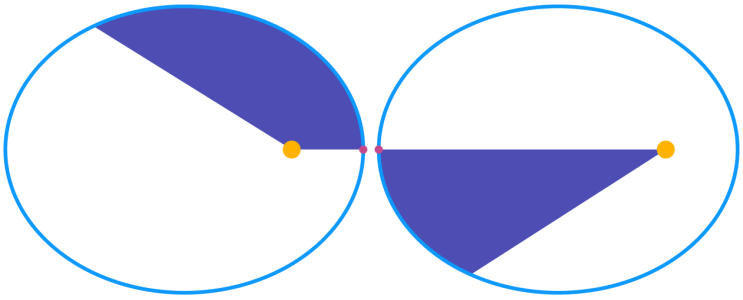


Figure 2.2.: Kepler's 2nd law visualized [12]. Orbital parameters are $\varepsilon = 0.6$ with an orbital period of $t = 4$ yr. When assuming a time interval of $\Delta t = 1$ yr, the traversed area (colored) measures 0.63 AU^2. In the left panel, the planet is located at the periastron ($\theta = 0°$), whereas in the right panel, the planet is located at the apastron ($\theta = 180°$).

3rd Law (The Law of Harmonies): Kepler's third law establishes a fundamental relationship between the orbital characteristics of celestial bodies, stating that the ratio of the cube of the semi-major axis of an orbit to the square of the orbital period remains constant for all objects in orbit around the same central star. This universal constant, which applies across all planets orbiting a given star, reflects a profound insight into the dynamics of celestial mechanics and the underlying gravitational forces governing planetary motion. The mathematical expression of this law is given by:

$$\frac{a^3}{t^2} = \frac{G \cdot (M + m)}{4\pi^2} \,. \tag{2.2}$$

In this equation, a represents the semi-major axis of the elliptical orbit, which is half the length of the orbit's longest diameter. The orbital period t denotes the time required for a planet to complete one full revolution around the star. The constant on the right-hand side involves G, the universal gravitational constant, which quantifies the strength of gravitational attraction, and the masses M and m, corresponding to the mass of the star and the planet, respectively. Note that $M >> m$ simplifies the equation, as the planet's mass has a negligible effect compared to that of the star.

By establishing the three laws as stated above, Kepler provided critical insights into the elliptical, non-uniform motion of celestial bodies and enabled astronomers to model the complex paths of planets with unprecedented accuracy, paving the way for the development of modern orbital dynamics. His fundamental contributions led to the eventual formulation of Isaac Newton's law of universal gravitation in the 17th century:

Given two bodies i and j with masses m_i and m_j, and a relative distance $\vec{r_{ij}} = \vec{x}_j - \vec{x}_i$, the gravitational force F_{ij} acting between the bodies is

$$F_{ij} = G\frac{m_j \cdot m_i}{r_{ij}^2} \,. \tag{2.3}$$

In the case of $n > 2$ bodies with non-zero masses, the total gravitational force acting on i is the sum of forces exerted by all other $n-1$ bodies in the system. Unlike the two-body problem (*i.e.*, $n = 2$), which can be solved analytically, systems with $n > 2$ generally exhibit complex, chaotic behavior that can not be solved via closed-form solutions. Consequently, such systems necessitate the use of numerical simulation techniques to accurately model their dynamics. Let us now investigate how this problem can be formulated as numerical integration of differential equations (also see Figure 2.3).

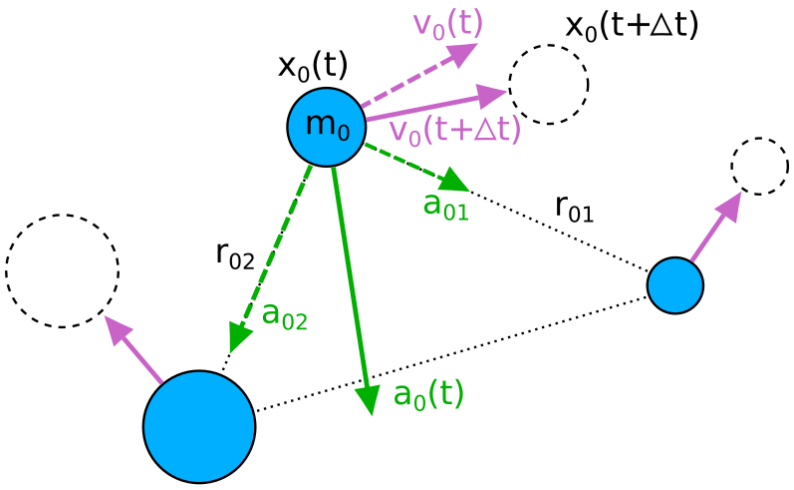


Figure 2.3.: N-body problem visualized. Object $i = 0$ with mass m_0 is being accelerated by two other objects, thereby changing its velocity and position from time $t \rightarrow t + \Delta t$. Other objects are affected in a similar manner and undergo their respective updates in parallel.

We begin with defining some initial conditions for positions $\vec{x}(t_{start})$ and velocities $\vec{v}(t_{start})$ of all bodies at the starting time $t = t_{start}$. Alternatively, initial conditions may be expressed in terms of the so-called *Kepler elements*: the orbit's semi-major axis a, eccentricity ε, inclination i (*w.r.t.* the main plane), argument of the periapsis ω, longitude of the ascending node Ω, and mean anomaly M. Both sets of initial conditions require a total of 6 parameters per body and can be mutually converted between one another.

A barycenter correction is usually applied to initial conditions such that the barycenter rests in the coordinate system's origin. This choice reduces numerical errors during simulations,

thereby enhancing the stability and accuracy of calculated solutions.

The overall objective of n-body simulations is to compute the trajectories of all bodies, represented by their position vectors $\vec{x}_i(t)$ for $i = 1, ..., n$ over a specified time interval $t = [t_{start}, t_{end}]$. The motion of each body is governed by its total acceleration, defined as the second time derivative of its position, $a = \frac{d^2x}{dt^2}$, which arises from the gravitational interactions with all other bodies. Considering an object i, this problem requires solving the following 2nd order ordinary differential equation (ODE):

$$\frac{d^2x_i}{dt^2} = G \sum_{j \neq i} m_j \frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|^3} \,. \tag{2.4}$$

This ODE can be re-formulated as a system of two 1st order ODEs:

$$\frac{d\vec{v}_i}{dt} = G \sum_{j \neq i} m_j \frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|^3} \tag{2.5}$$

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i \,. \tag{2.6}$$

By choosing a discrete time step Δt, the respective updates from $t \longrightarrow t + \Delta t$ are then

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t) + \Delta t \cdot \vec{a}_i(t) \tag{2.7}$$

$$\vec{x}_i(t + \Delta t) = \vec{x}_i(t) + \Delta t \cdot \vec{v}_i(t + \Delta t) \,. \tag{2.8}$$

This update rule is commonly known as the Euler update method, a fundamental approach for numerically solving ODEs by approximating the solution through discrete time steps. In addition to the Euler method, several alternative numerical integrators have been developed, each offering distinct advantages such as enhanced accuracy, improved stability, or energy conservation properties [102, 66, 13, 121, 45, 63]. These advanced methods, including the Runge-Kutta family, Bulirsch-Stoer, and Verlet algorithms, provide more sophisticated techniques for addressing the numerical challenges inherent in solving complex ODEs, particularly in

systems exhibiting nonlinear dynamics or requiring long-term integration. In contrast to the Euler update method, these integrators allow modeling intricate dynamical phenomena with high precision and reliability across a wide range of scientific and engineering applications.

To address the computational demands of modeling gravitational interactions in an n-body system, it is necessary to calculate all pairwise accelerations between bodies i and j for $i = 1, ..., n$ and $j = 1, ..., n$ with $i \neq j$. This results in a computational complexity of $\mathcal{O}(n(n-1))$, which simplifies to $\mathcal{O}(n^2)$ for large n. Consequently, evaluating the right-hand side (RHS) of equation 2.5 becomes computationally prohibitive for systems involving a large number of bodies, as often required in modern astrophysical simulations. The necessity of computing the RHS at each time step further exacerbates the computational burden, rendering direct numerical integration impractical for large n without specialized techniques. As a result, n-body integrations are usually carried out by a computer [47].

To mitigate the $\mathcal{O}(n^2)$ scaling, tree-based approximation methods have been developed [8, 99, 43]. These methods, including the Barnes-Hut algorithm and fast multipole methods, reduce the computational complexity of the RHS to $\mathcal{O}(n \log n)$). They thus offer highly efficient alternatives for simulations involving millions of bodies, such as encountered in planet formation, galactic dynamics, and large-scale cosmological studies.

When expressing the physical properties of celestial bodies, such as their masses, distances, and velocities, in terms of standard SI units, the resulting numerical values are often extremely large, leading to significant numerical errors during computer-aided integration. These errors arise due to the finite precision of floating-point arithmetic, which struggles to accurately represent quantities spanning many orders of magnitude. To address this challenge, astronomers frequently adopt specialized units tailored to the scale of the system under study. In the context of planetary systems, the Gaussian gravitational constant k [36], expressed in terms of the solar mass $M_\odot$ and the astronomical unit (AU), is commonly employed. This constant facilitates more numerically stable computations by scaling the relevant quantities to values that are better suited for computational precision, thereby minimizing rounding errors and improving the reliability and stability of the integration process.

Within the scope of this thesis, n-body simulations serve as a powerful and versatile computational tool for investigating a range of astrophysical phenomena, specifically problems **A**, **B**, and **C**, as defined in Section 1.4. More concretely, we utilize n-body simulations to estimate the initial conditions of collision events, which are critical for understanding the dynamical evolution of the system prior to impact. Additionally, these simulations allow for the study of long-range gravitational interactions between hydrodynamic particles during collision events, capturing the intricate interplay of forces that govern the deformation of bodies. Furthermore, n-body methods are applied to analyze the trajectories of fragments produced as a result of collisions, providing insights into the post-collision dynamics and the eventual fate of the resulting debris. By leveraging the capabilities of n-body simulations, this work aims to advance our understanding of the underlying mechanisms driving these astrophysical processes.

2.2. Smoothed Particle Hydrodynamics

N-body simulations are widely employed in astrophysics to model the long-term orbital evolution of celestial bodies, offering valuable insights into the dynamical behavior of planetary systems. These simulations excel in tracking the gravitational interactions of multiple bodies over extended timescales, providing robust predictions for orbital trajectories and system stability. However, a significant limitation of n-body simulations arises from their treatment of individual objects as point-masses. This simplification makes it impossible to accurately model collisions involving complex physical interactions, such as material deformation, fragmentation, or energy dissipation.

To address collision dynamics, a simple but rudimentary approximation known as perfect inelastic merging (PIM) is often employed [*e.g.*, 94]. In PIM, colliding objects are assumed to merge completely upon contact, conserving total momentum and mass while neglecting detailed physical processes like shock compression or material ejection. A PIM event is triggered when the distance between two objects falls below a specified threshold, such as $d < r_i + r_j$ with r_i and r_j representing the radii of the colliding bodies. These radii are generally estimated by modeling the objects as perfect

spheres with uniform mass densities, a simplification that overlooks variations in internal structure or composition.

However, research has demonstrated that PIM tends to overestimate collision efficiency, leading to unrealistically rapid growth of planetesimals during the early stages of planetary formation [15, 16, 14]. This overestimation introduces significant biases in simulations of protoplanetary disks, affecting predictions of planet formation timescales and the resulting architectures of planetary systems.

To overcome the limitations of PIM and enhance the quality of collision modeling, hydrodynamics simulations offer a powerful alternative. Unlike PIM, which relies on simplified assumptions, hydro simulations are grounded in physically realistic models that account for the thermodynamic and mechanical responses of colliding bodies. This approach significantly reduces systematic errors associated with collision handling, providing more reliable predictions for outcomes such as the formation of moons or the collisional evolution of debris. Improved modeling of collisions in turn allows for developing a better understanding of the intricate processes governing the growth and dynamical evolution of celestial bodies, paving the way for more accurate models of planetary system formation.

The discipline of fluid mechanics traces its origins to the latter half of the 19th century, when William Froude made seminal contributions through his pioneering work on optimizing ship designs [33]. Froude's innovative approach involved systematic experimentation and the use of scale model testing to investigate fluid flow and resistance, laying the groundwork for modern hydrodynamic principles (see Figure 2.4). His methodologies provided a rigorous framework for understanding the behavior of fluids in motion, particularly in the context of naval architecture, and established key concepts that remain important to the field today.

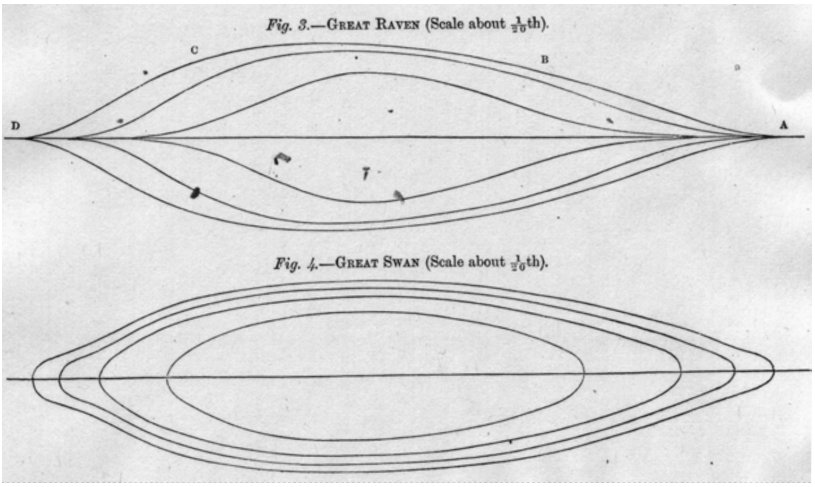


Figure 2.4.: Froude designed two scale models named *Raven* and *Swan* to compare different hull designs [40]. Raven models were consistent with John Scott Russell's *wave-line theory*, which emphasized hull shapes that mimicked wave patterns to reduce resistance. In contrast, Swan models were inspired by water birds.

In modern astrophysics, early hydrodynamic simulations were applied for studying gas dynamics during galaxy formation and star formation [57, 56, 17]. These works introduced computational techniques to model the intricate interplay of gravitational forces, gas pressure, and other dynamical processes. These early simulations provided critical insights into the mechanisms governing the evolution of cosmic structures. The development of these computational approaches marked a significant advancement in astrophysical research, facilitating the study of phenomena that are otherwise challenging or even infeasible to observe directly (also see Section 1.4).

In the study of fluid dynamics, two principal methodologies exist: grid-based methods and particle-based methods [31, 79, 110]. Grid-based methods, often referred to as *Eulerian* approaches, discretize the spatial domain into a structured mesh or grid, where fluid properties such as velocity, pressure, and density are computed at discrete grid points and their associated boundaries. This stationary reference frame enables the solution of governing partial differential equations, such as the Navier-Stokes equations, by leveraging numerical techniques like finite difference, finite volume, or finite element methods. These techniques ensure robust handling of complex flow phenomena, including turbulence and shock waves.

In contrast, particle-based methods, commonly known as *Lagrangian* approaches, model the fluid as a collection of discrete particles, each representing the fluid's local properties such as mass-density, velocity, and internal energy. These particles move with the flow, and their interactions are governed by physical laws. This co-moving reference frame is particularly advantageous for capturing interfaces,

free surfaces, and large deformations in fluid systems, as it naturally tracks the motion of material points without requiring a fixed grid.

Hydrodynamics simulations are not limited to pure fluid dynamics, but combine principles from various different fields in order to achieve a comprehensive and accurate representation of the underlying physical systems. In astrophysical applications, for example, simulations typically include thermodynamics, gravitational interactions, and condensed matter physics to model phenomena ranging from planet formation to galactic dynamics. The inclusion of condensed matter physics introduces material-specific effects, such as viscosity, porosity, internal stress, brittle failure, and phase transitions, which are critical for simulating realistic material behavior. Depending on the specific scientific objectives, simulations may further be extended to include additional processes such as electromagnetic interactions [4] (*a.k.a.* Magnetohydrodynamics) or relativistic effects [96] (*a.k.a.* Relativistic hydrodynamics). These processes are relevant when studying phenomena in accretion disks, plasma dynamics, or the dynamics of compact objects like neutron stars and black holes.

The computational demands of hydrodynamics simulations are substantial, driven by the need for high-resolution spatial and temporal discretizations, sophisticated physical models, and extensive numerical computations to accurately model the intricate dynamics of fluids. Note that gravity, in contrast to other interactions, is a global interaction between all massive components and therefore significantly increases computational complexity, as it necessitates the use of efficient n-body algorithms that scale with a computational cost of at least $\mathcal{O}(n \log n)$) (see Section 2.1). These algorithms are critical for handling the global nature of gravitational interactions in large-scale simulations.

Depending on the complexity of the physical models, the resolution requirements, and the available computational resources, simulations can require weeks to months of wall-clock time to complete, even on modern high-performance computing platforms [111, 64, 123]. These demands underscore the importance of optimizing numerical methods and leveraging advanced hardware architectures, such as parallel computing clusters or graphics processing units (GPUs), to achieve feasible runtimes while maintaining the high fidelity of the simulated processes.

Figure 2.5.: Illustrative snapshot of a moon-forming impact after two hours from first contact [59]. Colors indicate materials for target core (gray), target mantle (orange), projectile core (brown), and projectile mantle (yellow).

In this work, we focus on a particular particle-based simulation method called *Smooth Particle Hydrodynamics* (SPH, [73, 39, 105]). SPH is a Lagrangian method that discretizes a continuous fluid into a set of particles. By solving the equations of hydrodynamics in a flexible, mesh-free framework, SPH is capable of accurately capturing the complex physical interactions that occur during collisions in the context of planet formation. Modeling these collisions include capturing phenomena like shock propagation, major deformations, and fragmentation. More details about the underlying physics, which is being modeled, is provided in Section 2 of my minor contribution (see Chapter 4).

In the context of this thesis, we use SPH simulations for studying problem **B** as defined in Section 1.4. More concretely, we use SPH simulations for creating a hyperparameter study of pairwise planetary collisions, as well as studying fragmentation, water transport, and overall collision dynamics. Using the results from the hyperparameter study, we further train machine learning models for fast and accurate prediction of planetary collision outcomes given some initial conditions.

2.3. Machine Learning

As mentioned in Section 1.4, simulating a collision involves setting the initial conditions for both objects (mass, material structure, spin, atmosphere, *etc.*.), as well as the overall collision geometry (impact velocity, impact angle, *etc.*.). The respective outcome of a collision is greatly affected by the choice of those hyperparameters.

In our Solar System, we are typically limited to observe collision outcomes like the Earth's Moon, various craters, and irregular shaped comets and asteroids. We would naturally like to know which initial conditions might have led

to these specific outcomes in order to obtain a complete picture of the full history. Due to chaotic collision dynamics, however, finding these initial conditions is complex and sometimes non-intuitive. Moreover, long-term geological processes such as sedimentation and plate tectonics can significantly modify post-collision morphology and structural geology. Consequently, the complexity of deducing plausible initial conditions from observations is further exacerbated.

A popular approach for estimating initial conditions involves running the same scenario with slightly adjusted hyperparameters, thereby carefully analyzing and studying the respective outcome in order to further improve hyperparameters. This process is iterated until chosen hyperparameters lead to outcomes that match observations sufficiently well. As one can imagine, this approach is computationally expensive, complicated, and requires constant intervention by a human expert. Moreover, vastly different sets of initial conditions can lead to similar outcomes, thereby transforming the problem into finding not only one plausible set of initial conditions, but all plausible sets along with their respective probabilities.

In practice, iterative hyperparameter search often results in running a parameter-study, *i.e.*, performing a grid-search within the hyperparameter space of initial conditions. This is even more computationally expensive than iterative search since most hyperparameter combinations are implausible and can thus be discarded. Moreover, in case of many hyperparameters, grid search becomes prohibitively expensive and inefficient due to the curse of dimensionality.

However, the importance of parameter studies should not be underestimated since they implicitly contain valuable information about collision dynamics, as well as correlations between all hyperparameters. This implicit information can be automatically extracted and exploited via machine learning (ML, [78]). In contrast to deductive reasoning, ML learns from data and therefore relies on inductive reasoning. A concise definition of ML was provided in [78]: "A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

The capabilities of ML to effectively extract and utilize patterns from data significantly shape the research approach and methodologies adopted in this thesis. In the context

of studying planetary collisions, ML is not only capable of automatically and efficiently finding plausible initial conditions (Chapter 3), but also of accurate and fast prediction of collision outcomes when initial conditions are provided (Chapter 6). In ML terms, these problems are often referred to as inverse and forward problems, respectively. In the context of this thesis, these problems translate to studying problems **A** and **B** as defined in Section 1.4.

Since astrophysicists are not necessarily familiar with the field of ML, I would now like to briefly introduce the relevant information that is necessary for following my scientific contributions presented in Part II.

Genetic Algorithms (GA): GAs are a family of computational models that are inspired by the laws of natural selection in evolutionary biology [117, 128, 93]. Although GAs can be applied to a wide range of problems, they are commonly used for parameter-optimization and search. Solving a specific problem with GAs (*i.e.*, finding near-optimal solutions globally) involves the following steps:

- Encoding and decoding a set of candidate solutions (*population*) into -and from a chromosome-like data structure (*DNA*).
- Evaluating the performance (*fitness*) of candidate solutions on a given problem.
- Applying operators for selection, recombination, and mutation to improve the average performance of the set of candidate solutions.

These steps are repeated in an iterative manner (*evolution*) until convergence or until a minimum performance threshold is met.

GAs are very general, *i.e.* they do not require making strict assumptions about the underlying optimization problem. They can therefore be applied to a wide range of functions, including those that are non-differentiable and have multiple local optima. GAs usually yield robust results even in case of limited domain knowledge. However, due to their lack of exploiting problem-specific properties like gradient information, they often under-perform specialized optimization methods.

In Chapter 3, we apply GAs to find orbital parameters of small bodies that collide with Jupiter's moons. Based on these parameters, we then estimate typical collision velocities and

impact angles that are highly relevant for reconstructing crater formation for specific craters.

Deep Learning (DL): DL [2, 69] is a special type of ML that makes use of artificial neural networks (ANNs). ANNs are composed of simple, non-linear modules referred to as layers. Stacking those layers, *i.e.* increasing depth, results in so-called DL modules. These modules are capable of learning highly abstract representations that are useful for solving specific tasks. There are several types of ANNs, including feed-forward networks [51] (see Figure 2.6), convolutional networks [34, 70], and recurrent networks [28, 54, 90, 46].

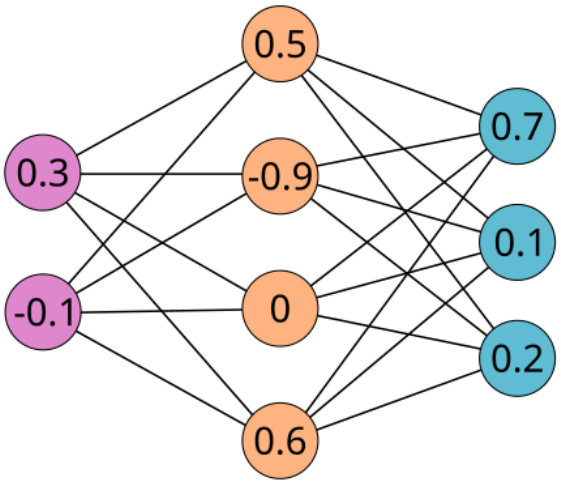


Figure 2.6.: Depiction of a simple example for a two-layered feed-forward neural network (FFN) as directed acyclic graph. The FFN is comprised of 2 input units (purple), 4 hidden units (orange), and 3 output units (blue). Connections w between units are referred to as *weights* and are learned during training. Given some inputs x, respective outputs are calculated as $y = w_1 \cdot \sigma(w_0 \cdot x)$. A non-linear activation function σ is applied to hidden units element-wise in order to make the entire model non-linear.

Unless explicitly introducing randomness, ANNs are deterministic input-output mappings that are characterized by their architecture (hyperparameters), as well as their weights (learnable parameters). Training a DL model, *i.e.*, selecting a specific model from the model class, requires finding a set of weights that meet specific criteria. For example, one would like to find weights such that model predictions (*i.e.*, outputs) meet certain minimum performance criteria when evaluated on some data. ANNs are trained via automated, algorithmic adaptation *w.r.t.* a training dataset. The most prominent optimization algorithm is stochastic gradient descent [97], where weights are iteratively updated in small steps by exploiting their respective loss-gradients. Gradients are calculated via backpropagation [100, 101]. Losses are calculated from a so-called loss function, which acts as optimization objective. The overall goal is generalization, *i.e.*, that the chosen model has the best performance on future, unseen data.

In supervised learning [104], the loss is the distance between model predictions and ground truth targets (also known

as labels) as defined by some distance metric (*e.g.*, mean squared error or cross-entropy error). While early layers usually learn primitive patterns like intensity and edges in images, later layers become sensitive to abstract patterns like specific concepts and objects [138]. Supervised learning methods make the *i.i.d.* assumption, *i.e.*, that training data and future unseen data stem from the same distribution, originating from the same (static) real-world process. Since the *i.i.d.* assumption often does not hold in practice, DL models are usually evaluated on a test dataset that was not used during model development.

Both ANN training and inference can be efficiently implemented by exploiting parallel computing capabilities of graphics processing units (GPUs) [84, 65]. This innovation kick-started the recent revival of DL in research and industry.

Part II.

Understanding Collisions in Planetary Systems by Computational Modeling

3. Analysis of Close Encounters With Ganymede and Callisto Using a Genetic n-Body Algorithm

The first major contribution of this thesis addresses the problem of finding plausible initial conditions for asteroid impacts onto Jupiter's icy moons Ganymede and Callisto (*i.e.*, addressing problem **A** as defined in Section 1.4). By combining n-body simulations with a genetic algorithm, we were able to efficiently search for virtual asteroids that experience a close encounter or even a collision with one of the moons. We initialized orbital parameters of asteroids similar to those from the Centaur family, which are considered as the primary source of impactors.

Our work was published as a journal article in 2018 [134]. Moreover, preliminary results were already used for investigating the formation of Callisto's Valhalla basin with SPH simulations. These studies resulted in my master's thesis [133], a conference talk [130], as well as a conference contribution [132] (see Figure 3.1).

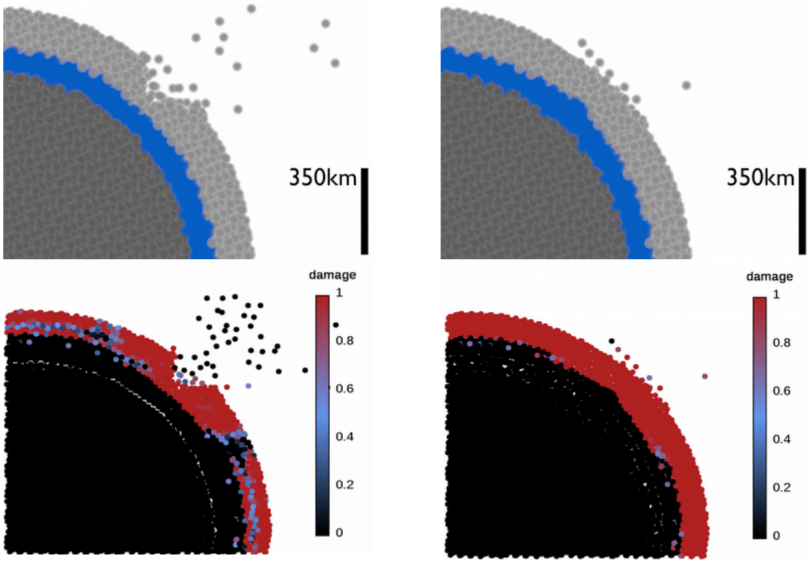


Figure 3.1.: Reconstruction of Callisto's Valhalla basin with SPH simulations [132]. The top panels visualize two snapshots of a plausible crater-forming impact, with a subsurface ocean (in blue) underneath an icy crust (in light gray). The transient crater (left side) quickly flattens out (right side) due to self-gravity. Bottom panels visualize the damaged icy crust due to pressure waves propagating through the body.

frontiers in Astronomy and Space Sciences

ORIGINAL RESEARCH
published: 22 May 2018
doi: 10.3389/fspas.2018.00016

Analysis of Close Encounters With Ganymede and Callisto Using a Genetic n-Body Algorithm

Philip M. Winter, Mattia A. Galiazzo and Thomas I. Maindl*

Department for Astrophysics, University of Vienna, Vienna, Austria

In this work we describe a genetic algorithm which is used in order to study orbits of minor bodies in the frames of close encounters. We find that the algorithm in combination with standard orbital numerical integrators can be used as a good proxy for finding typical orbits of minor bodies in close encounters with planets and even their moons, saving a lot of computational time compared to long-term orbital numerical integrations. Here, we study close encounters of Centaurs with Callisto and Ganymede in particular. We also perform n-body numerical simulations for comparison. We find typical impact velocities to be between $v_{rel} = 20[v_{esc}]$ and $v_{rel} = 30[v_{esc}]$ for Ganymede and between $v_{rel} = 25[v_{esc}]$ and $v_{rel} = 35[v_{esc}]$ for Callisto.

Keywords: Callisto, Ganymede, n-body, close encounters, genetic algorithm, celestial mechanics, numerical simulation, collisions

OPEN ACCESS

Edited by:
Lorenzo Iorio,
Ministry of Education, Universities and Research, Italy

Reviewed by:
Áron László Süli,
Eötvös Loránd University, Hungary
Bojan Novakovic,
University of Belgrade, Serbia

****Correspondence:***
Philip M. Winter
philip.winter@univie.ac.at

Specialty section:
This article was submitted to Fundamental Astronomy, a section of the journal Frontiers in Astronomy and Space Sciences

Received: *19 January 2018*
Accepted: *30 April 2018*
Published: *22 May 2018*

Citation:
Winter PM, Galiazzo MA and Maindl TI (2018) Analysis of Close Encounters With Ganymede and Callisto Using a Genetic n-Body Algorithm. Front. Astron. Space Sci. 5:16. doi: 10.3389/fspas.2018.00016

1. INTRODUCTION

Jupiter's large icy moons such as Ganymede and Callisto show countless impact craters across their surface. Studying these craters gives deep insights into the impactors as well as the moons themselves. This is the first approach in the frame of future works of studying collisions with the outermost moon Callisto. We are especially interested in the Valhalla crater system, Callisto's biggest crater. This impact structure measures several hundred of kilometers in diameter and shows some extraordinary features such as an extensive ring system in the outskirts of the crater (Greeley et al., 2000; Stewart and Allen, 2002). It has been shown that studying the formation of the Valhalla crater reveals new insights regarding Callisto's subsurface composition (Winter et al., 2017).

In this first work we focus on the use of a genetic algorithm[1] (which is described in section 2 and in the Appendix for further details) as a proxy tool to find preliminary orbits of possible close encounters for bodies in the Solar System. In particular, we selected the Centaurs' population. Currently, 423 Centaurs are known[2]. It is estimated that the number of Centaurs with a diameter larger than 1 km lies between $n \sim 10^7$ (Volk and Malhotra, 2008) and about $n \sim 8 \cdot 10^9$ (Di Sisto and Brunini, 2007; Fernández and Sosa, 2015; Napier et al., 2015). These bodies mainly origin from the Trans-Neptunian Objects and have very chaotic orbits. With semi-major axes between $a = 5.5$AU and $a = 30.1$AU, they lie between the giant plantes, from which they are frequently ejected out of the Solar System via close encounters or even impact on one of the planets or their moons. Their lifetime is of the order of 10 Myr (see e.g., Horner et al., 2004a,b; Bailey and Malhotra, 2009; Galiazzo et al., 2016). A GA in combination with standard orbital numerical integrators can be used as a good proxy for typical orbits of minor bodies in close encounters or impacts

[1]Herein "GA".
[2]Data taken from JPL Small-Body Database Search Engine https://ssd.jpl.nasa.gov/sbdb_query.cgi

TABLE 1 | Parameter space for random initial conditions of the test particles.

	Min	Max
a (AU)	4.95	30.33
e (1.0)	0	0.99
i (deg)	0	180
ω (deg)	0	360
Ω (deg)	0	360
M (deg)	0	360

The inclination is weighted by the density distribution of 420 known Centaurs to ensure a realistic distribution of objects.

with planets and even their moons, saving a lot of computational time compared to full orbital numerical integrations (section 3.1). We also perform n-body numerical simulations (section 3.2). We extrapolated the typical orbits in close encounters with their osculating elements and velocities (sections 3.1, 3.2). This kind of orbital analysis can also be very useful for studying the origin of impactors in the last million years, i.e., the Bosumtwi impactor (Galiazzo et al., 2013). We particularly design the genetic algorithm to investigate close encounters with Callisto—and for comparison—Ganymede. Close encounters are the first natural approach to actual impact scenarios which we will study in following works.

2. METHODS

Measuring close encounters or even collisions between minor and major bodies in the context of n-body simulations is computationally demanding. Typically one has to constrain the parameter space of the minor bodies to selected regions within the Solar System (e.g., Kuiper belt objects or specific families of objects). We use a genetic algorithm to find asteroids of the Centaur type family which are likely to have a close encounter with the Jovian moons Ganymede and Callisto. Centaurs mainly origin from TNOs and in particular from the Kuiper Belt (Galiazzo et al., 2016). The method is used to encounter the problem of a large parameter space of initial orbital elements (see **Table 1**). The genetic algorithm boosts the performance compared to classical searching grids by some orders of magnitudes within the given computation time. This allows for measuring a reasonable amount of datapoints to do a statistical analysis of typical close encounter families which marks the first step toward studying actual impact scenarios.

2.1. Genetic Algorithms

A genetic algorithm (Turing, 1950) is an iterative searching algorithm to find solutions for highly complex problems which can have large parameter spaces. GAs origin from the field of biology, therefore we may also use the corresponding terms. Further information is given in the Appendix. GAs are efficient optimization methods for highly complicated functions. The population can overcome local minima quite easily either by the means of crossover or mutation. Moreover, GAs tend to find all possible families of solutions, even if they are unrealistic or uncommon. However, GAs also have negative aspects. The implementation can be quite tricky and there is no general rule how to implement them efficiently because the functions (for fitness, crossover, and mutation) and parameters (number of generations, number of individuals) have to be adapted to the problem as well as to the hardware. If the minimum fitness limit which is needed to solve the problem is not met (due to poor convergence or no learning process), the GA will not find any solutions.

2.2. Genetic N-Body Algorithm

`genbody` is a GA which is being developed to find close approach- and collisional orbits of particles in the context of n-body simulations. Each generation of the GA corresponds to a distinct n-body simulation. During the simulation, the fitness of the population is measured. Afterwards, a new population is created by the means of crossover and mutation. A pseudo code of the genetic n-body algorithm is given in Algorithm 1.

Algorithm 1 Genetic n-body algorithm

```
initialize population
for g in generations do
    while t < t_end do
        take n-body step (t → t + h)
        measure fitness
    crossover
    mutation
```

We use the code to find objects which are likely to collide with either Ganymede or Callisto within a certain time interval. The following list gives an overview how the GA and the n-body code are associated with each other. In analogy to section 2.1, we link the terms between the genetic algorithm and the actual problem:

- Population: Centaur type asteroids with random initial conditions
- DNA: initial Keplerian orbital elements (a, e, i, ω, Ω, M)
- Generation: numerical simulation of the system via an n-body method
- Evolution: iterative process of consecutive simulations
- Fitness: score of each test particle at each simulation given by the fitness function
- Crossover: combination of initial orbital elements of parent particles given by the crossover function
- Child: a test particle with a new set of initial orbital elements
- Mutation: small, random variation of initial orbital elements

The so-called fitness function is the function to be optimized. We use a fitness function of $f = 1/{d_{\rm rel}}^2$ with $d_{\rm rel}$ being the minimum relative distance between a particle and the corresponding moon during each generation. The squared distance was found to be useful if the fitness is used as a probability distribution to sample quite fit parents for crossover, ensuring a high fraction of fit parents and thus increasing the overall performance of

the GA. Note that one can use other fitness functions to study completely different problems. One only has to find a quantity to score objects which show a specific behavior to obtain a population which develops that behavior. The fitness of each body is measured throughout the simulation. If a close approach with the moon occurs, a measurement is taken and the algorithm starts a completely new evolution process. If no close approach happens during the simulation, unfit objects are replaced by new children which are created by objects with a high fitness score. The initial orbital elements $y \in \{a, e, i, \omega, \Omega, M\}$ of the new children are exact 1:1 copies of their corresponding parent. Note that one could use other crossover functions which include two or more parents to create a new child. Next, a small mutation is applied to the whole population in order to avoid to get stuck in local maxima in the learning curve (generation vs. mean fitness). Since our n-body system turns out to be highly sensitive to the mutation rate χ, it is crucial to find suitable values for each generation. A too small χ leads to a solution which gets stuck in a local maximum while a too large χ on the other hand leads to destruction of genetic information. This corresponds to random guessing without any learning process of subsequent generations of populations. Since the trajectories of small bodies in the Solar System are chaotic, the mutation has to be low enough to allow for similar trajectories of subsequent populations (except for the newborn child) within the simulation time. We therefore use an adaptive mutation rate χ which depends on the overall fitness evolution. A scaling factor ζ is introduced to control the learning curve by setting the amplitude of the mutation rate χ: If the mean fitness between two successive generations varies too much, ζ is decreased to prevent a loss of DNA information. If the mean fitness between two successive generations varies too little (by less than 1% in this work) or if the inidividuals of the population are getting too similar to each other, ζ is increased to ensure a healthy population. The mutation rate itself is given by the standard deviation σ of each initial orbital element throughout the population (e.g., $\chi_e = \zeta \cdot \sigma_e$). The initial mutation scaling is set to $\zeta = 0.1$ for all evolutions. At this point it should be noted that the functions for fitness, crossover and mutation are empirical functions found to yield good results (high performance due to learning curves with steep slopes) for this specific problem. Mathematical formulations of the functions we use for fitness, crossover and mutation are given in the Appendix.

The size of the population is $n_{\mathrm{pop}} = 30$ and the total number of bodies in each simulation is $n_{\mathrm{tot}} = 38$, including the Sun, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune. The mass of Mercury was added to the Sun. The orbital elements for the massive bodies are obtained from the JPL HORIZONS system. We perform the simulations in the Jovian-centric system, as we expect only negligible changes of the results because the systematic error produced by the GA is significantly larger than the error made by not using a barycentric frame of reference. We use the Lie-Series n-body integrator as described in Hanslmeier and Dvorak (1984) with a numerical accuracy of $\varepsilon = 10^{-11}$. Note that we do not explicitly include Ganymede and Callisto in the simulations due to a performance gain. In contrast to symplectic n-body integrators, the stepsize of the Lie-integrator is limited by the minimum relative distance between any of the objects,

$$h = \frac{1}{k_G}\left(\frac{\varepsilon \cdot \nu!}{\mathcal{D}}\right)^{\frac{1}{\nu}} \tag{1}$$

where h is the stepsize in days, k_G is the Gaussian gravitational constant and ε is the numerical accuracy. The quantities ν and $\mathcal{D}$ are obtained from the integrator itself during each step, where ν is associated with the number of used Lie-terms and $\mathcal{D}$ depends on the masses and relative distances between the objects (Eggl and Dvorak, 2010). If we include the moons in the calculations, the stepsize would be limited by the distances within the Jupiter system and much higher computational capacities would be needed in order to get the same amount of results. Therefore, we use pseudo-moons which are characterized only by the distance to Jupiter without gravitation in order to be able to calculate distances. A measurement is taken if a particle has its closest approach within the Hill sphere of a moon. The stepsize of the integrator during a Hill sphere crossing has to be low enough to ensure that the maximum spatial distance between two consecutive frames does not exceed the size of the Hill sphere. Otherwise, the particle would overleap the Hill sphere and no measurement would be detected. Therefore, an additional restriction for the upper bound of the stepsize is used. Moreover, if the moons are included in the simulations, their very short orbital periods would lead to significant stability issues with their orbits even at very low stepsizes. Note that since no actual positions of the moons are given during the simulations, the Hill spheres are represented by torus-shaped regions around Jupiter with their first radius being the relative distance to Jupiter and the second radius being the Hill radius of the respective moon. This approach can be used because of the statistical nature of the study, expecting a large number of measurements. Since we are doing statistical studies of typical intersection velocities we found this simple approach to be highly effective. Due to the nature of the GA, we are able to use a large parameter space for the random initial conditions of the test particles (see **Table 1**).

We set the maximum simulation time $t_{\mathrm{max}} = 165\,\mathrm{yr}$ in order to allow at least a full orbital period for each individual. The maximum number of generations g is set to $g_{\mathrm{max}} = 250$. A new iteration is started either if there is no measurement (i.e., close encounter with either Ganymede or Callisto) within these g_{max} iterations or as soon as such a close encounter is recorded. Finally, for a quick check of the results with the GA method, we also check its output with the ones from full integrations of Centaurs (through all their lifetime) taking data kindly provided by Galiazzo et al. (2016), see section 3.2 for more descriptions. We compare these data with the orbits found for Ganymede, as far as close encounters with Jupiter were taken in account up to $d = 0.01\,\mathrm{AU}$.

3. RESULTS

3.1. Analysis of GA Measurements

After about 3300 CPU hours, total number of 531 and 625 measurements was obtained for Ganymede and Callisto, respectively. Each measurement corresponds to an individual

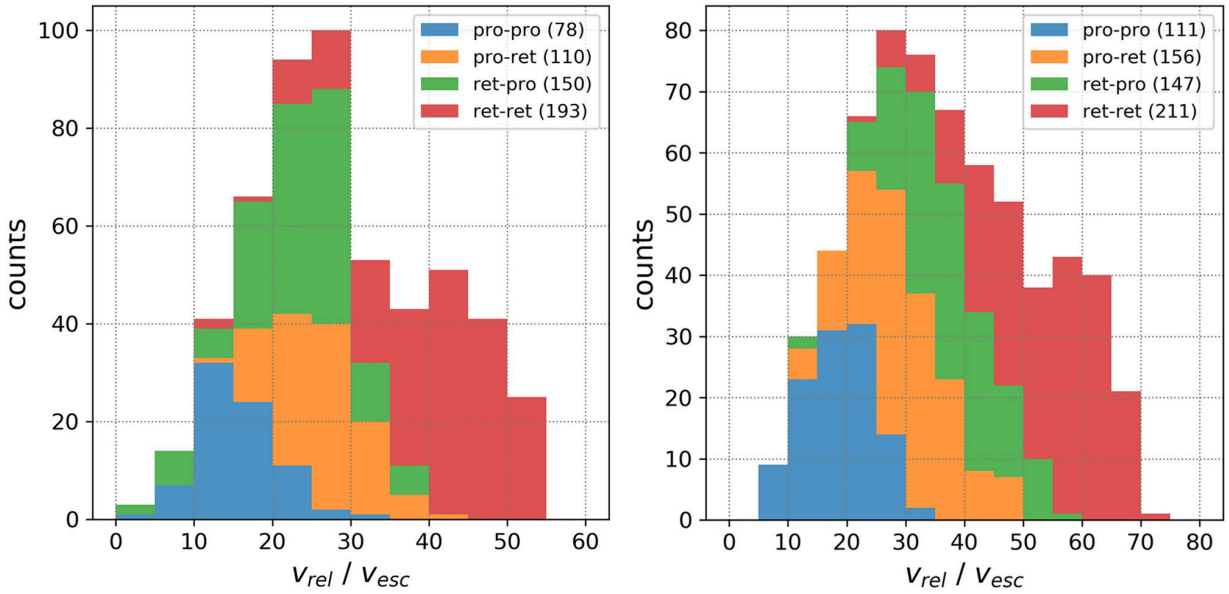


FIGURE 1 | Left: Ganymede, Right: Callisto. The histograms show the relative velocities between the test particles and the corresponding moon at closest approach in units of the escape velocity. The four colorized components represent the possible geometrical encounter scenarios. For example, the orange component *pro-ret* refers to a particle having an inclination smaller than $i = 90°$ (therefore prograde), but encountering the moon retrograde with respect to the Jovian System. The numbers in parentheses correspond to the respective number of measured datapoints. All bins are stacked.

evolution process of the GA and represents an individual close encounter scenario which contains all relevant information about positions, velocities, orbital elements, etc. about the bodies at both the time of closest approach and at the beginning of the respective simulation.

Figure 1 presents the main results of these measurements, showing the relative velocities at closest approach with respect to the moons. We can split the data into four classes, depending on the geometrical properties of the particle trajectories:

1. The low-velocity class (*pro-pro* encounters)
2. The intermediate-velocity class A (*pro-ret* encounters)
3. The intermediate-velocity class B (*ret-pro* encounters)
4. The high-velocity class (*ret-ret* encounters)

The terms *pro-pro*, *pro-ret*, *ret-pro*, and *ret-ret* refer to the corresponding geometries, where the first part describes the prograde or retrograde movement with respect to the Jovian System (with other words: $i < 90°$ is *pro* and $i > 90°$ is *ret*) and the second part describes the direction of flight with respect to the corresponding moon within the Jovian System, moving parallel (*pro*) or antiparallel (*ret*) to the respective moon. More detailed statistics of the four classes are given in **Table 2**.

The most probable close encounter velocities can be seen between $v_{rel} = 20[v_{esc}]$ and $v_{rel} = 30[v_{esc}]$ for Ganymede and between $v_{rel} = 25[v_{esc}]$ and $v_{rel} = 35[v_{esc}]$ for Callisto.

Further observations can be obtained from the results:

- The overall form of the relative velocity histograms can be reproduced by overlapping Gaussian distributions which are represented by the four classes.

TABLE 2 | Mean and standard deviations of close encounter velocities for the four classes, as well as their numbercounts.

	μ(km/s)	σ(km/s)	μ(v_{esc})	σ(v_{esc})	*n*
GANYMEDE					
Class 1, *pro-pro*	15.03	4.55	15.36	5.17	78
Class 2, *pro-ret*	24.38	3.09	26.00	5.33	110
Class 3, *ret-pro*	22.39	5.59	23.45	6.68	150
Class 4, *ret-ret*	39.29	3.84	40.65	8.48	193
CALLISTO					
Class 1, *pro-pro*	11.63	3.40	18.44	5.96	111
Class 2, *pro-ret*	19.41	3.57	29.53	7.91	156
Class 3, *ret-pro*	23.85	3.59	36.84	8.40	147
Class 4, *ret-ret*	34.96	3.12	52.86	10.45	211

- The classes are overlapping stronger for Ganymede than for Callisto, even swapping places (in velocity) when comparing class 2 and class 3 for Ganymede.
- There is a clear trend favouring retrograde encounters (for both Jupiter and the respective moon), with most close encounters being *ret-ret* (36.3% for Ganymede, 33.8% for Callisto).

The first row in **Figure 2** shows the final semi-major axes at the time of the closest approach. While Ganymede shows two distinct peaks at $a \simeq 5$ AU and $a \simeq 20$ AU, Callisto only has one single, large peak at $a \simeq 9$ AU. The other panels shows the respective quantity at starting time of the last generation vs. the same

quantity at the time of the respective measurement. A 1:1 relation corresponds to no change of the orbit during the simulation, while a large scattering means significant changes. As we can see for the eccentricities, there is a significant scattering toward larger values. The plots imply that the scattering for Ganymede close encounters is higher than for Callisto. More than half of the particles even undergo the transition from elliptic ($e < 1$) to hyperbolic ($e > 1$) trajectories. The strong scattering is also apparent in the semi-major axes. The inclination data shows only minor scattering between $i = 50^\circ$ and $i = 150^\circ$, which implies quite stable configurations within the simulation time. However, at low inclinations particles are scattered over the full parameter space and prograde ones tend to get scattered stronger than their retrograde counterparts.

Several other obervations can be obtained from the datasets:

1. The needed number of generations for taking the measurement grows steadily until about $g = 100$ before falling again rapidly.
2. The most dramatic changes of initial orbits happen during the first few tens of generations.
3. It is easier for the GA to find intersection orbits after a short simulation time. Therefore, the time of measurement peaks toward low values with a mean simulation time of $t_{mean} = 45.5$ yr.
4. The intersection probability is approximately twice as high for Callisto because her Hill radius is larger than Ganymede's. For statistical reasons, we therefore assigned more computation time to Ganymede.
5. Two actual collisions are measured for Ganymede (with impact velocities of $v_{\mathrm{rel}} = 34.6$ km/s and $v_{\mathrm{rel}} = 42.2$ km/s, respectively), none for Callisto.

3.2. Preliminary Parent Bodies' Orbits

We take the orbital evolution of Centaurs to do a comparison between the predicted Centaurs' orbits at close encounters and the Jovian moons. We integrate forward a sub-sample of the Centaurs for 30 Myrs and check all the close encounters with Jupiter, using the Lie-integrator. This study considers only close encounters up to a distance of $d = 0.01$ AU from Jupiter similarly to Galiazzo et al. (2016), but with an encounter radius 4 times smaller at the cost of computational time. Thus, the comparison is limited to Ganymede only (since he has a distance of $d \simeq 0.0072$ AU to Jupiter. Callisto has a distance of about $d \simeq 0.0126$ AU to Jupiter)[3].

We take 319 Centaurs with 15 clones in each interval ranging over 5 AU in semi-major axis (for a total of 5104 bodies)[4] in order to quickly get a statistical sample which covers the entire Centaur region from 5 to 30 AU in semi-major axis. This approach is sufficient to give an idea of these kinds of bodies approaching Jupiter and its moons and to have a compareable sample to the orbits produced by the GA.

From the evolution of 5104 objects, a total number of 292 measurements was obtained for Ganymede close encounters. From our sample of Centaurs we find that $\sim$ 22.6% can have close encounters with Jupiter. As the percentage of Centaurs which can cross Ganymede orbits is about 20.1% (8.7% with $e < 1$), almost all the Centaurs' close encounters with Jupiter ($\sim$ 89%) can reach Ganymede's orbit in the range of its Hill sphere. **Figure 3** shows the comparison between GA and Lie integrations for the respective orbital sample output in the Hill sphere of Ganymede.

Figure 3 suggests that the datasets are in quite good agreement, with the GA and the Lie datapoints covering a similar area in the $a - e$ phase space. However, a significant sample at high inclinations is apparent in the GA case, which makes the GA measurements not fully consistent with the measurements obtained by the Lie-integrations. An explanation for this effect is given in section 4. It turns out that these high-inclination datapoints are also responsible for the dense cloud in the $a - e$ phase space between $a = 15 - 30$ AU and $e = 0.7 - 0.8$. Otherwise, the distributions in the $a - i$ space are reasonably in agreement, especially at semi-major axes below $a = 15$ AU. Even the major cloud between $a = 2.5 - 15$ AU and $i = 50 - 140^\circ$ is reflected well. In order to quantify the resemblence of the two distributions obtained via the GA and Lie-integrations, we deploy a two-dimensional Kolmogorov–Smirnov test (Press and Teukolsky, 1988). Due to the reasons mentioned in section 4 we exclude the non-physical high-inclination datapoints with $i > 160^\circ$. For the $a - e$ distributions we get a p-value of $p = 0.27$, for the more diverse $a - i$ distributions $p = 0.058$. While the $a - e$ distributions do not differ significantly, the $a - i$ distributions show significantly less correlation, which indicates the need for a refined GA method as laid out in section 5.

4. CONCLUSIONS

The velocities of the four classes as described in section 3.1 can be explained by geometrical considerations: Class 1 encounters happen if the particles experience deceleration from the Jupiter flyby and additionally "lose" relative velocity due to the parallel direction of flight compared to the respective moon. The intermediate classes 2 and 3 either experience deceleration or acceleration from the Jupiter flyby and additionally "gain" or "lose" relative velocity due to the antiparallel or parallel direction of flight compared to the respective moon. Both acceleration by Jupiter and a "gain" of relative velocity is true for class 4.

Comparing the numbercounts of the classes, a correlation exists with higher fractions of both types of retrograde encounters being more probable. The higher fractions of retrograde encounters can be described by a simple geometrical effect: While a particle moves within the torus-shaped Hill region (the complete area which is accessed by the moon's Hill sphere during a full orbital period), the probability for a retrograde measurement is much higher than a prograde one

[3]We assume the semi-major axes of the moons as a proxy for the distance, neglecting their small eccetricities.

[4]The first region is the one with a semi-major axis between 5 and 10 AU. The second region is between 10 and 15 AU and congruently for the other regions up to 30 AU.

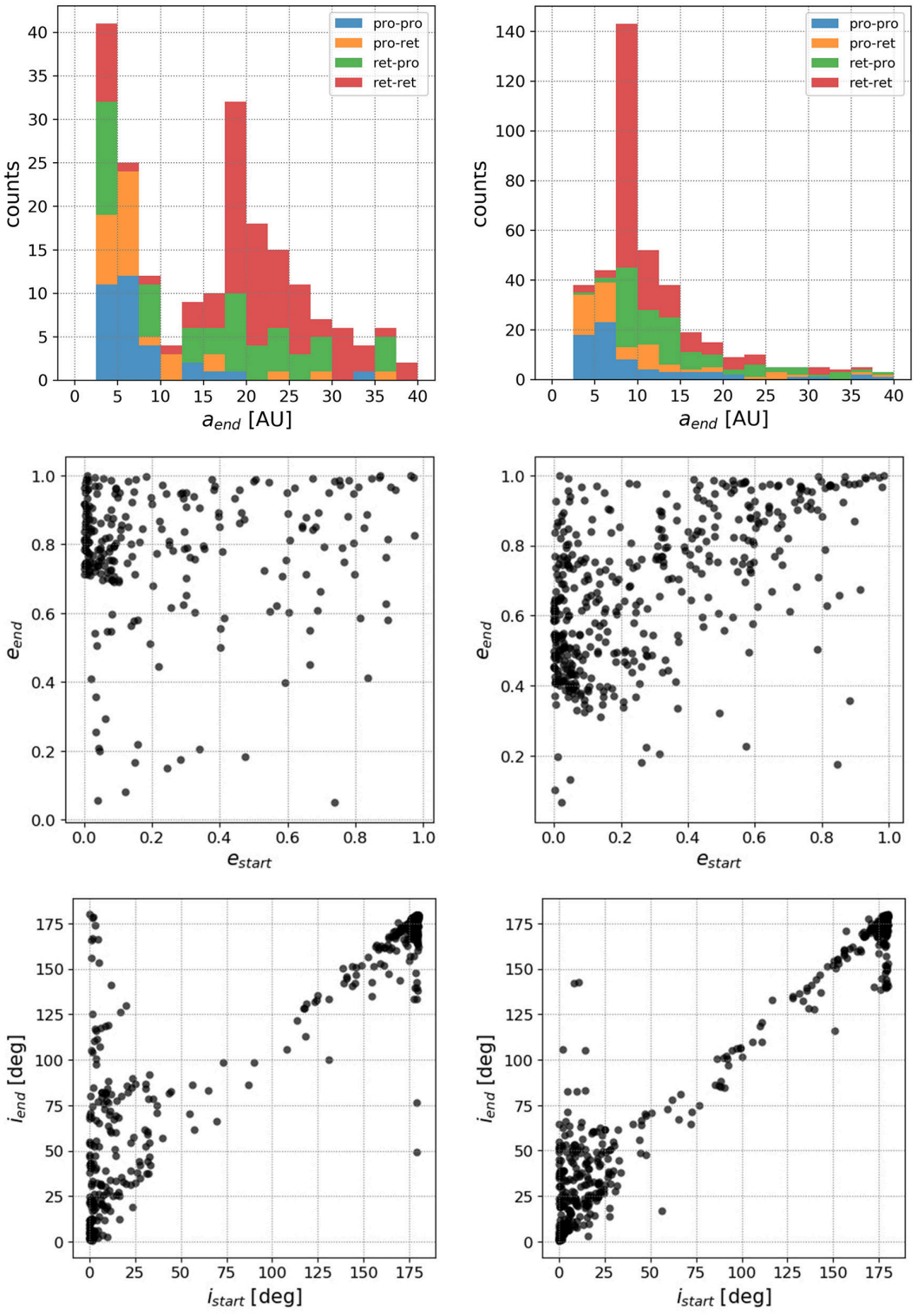


FIGURE 2 | Left column: Ganymede, right column: Callisto. The top row shows the semi-major axes at closest approach. The middle row shows the initial eccentricity of the last generation vs. the eccentricity at the measurement. The bottom row shows the respective inclinations. Each datapoint corresponds to an independent evolution.

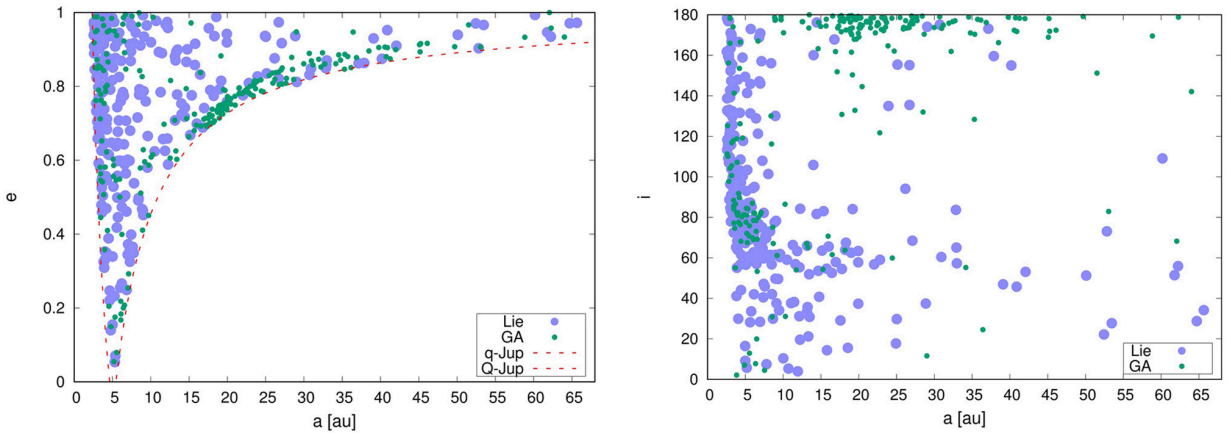


FIGURE 3 | Comparison between measurements of GA and Lie integrations of evoluted orbits for Ganymede close encounters. For the GA, all measurements are taken within Ganymede's Hill sphere. For the Lie integrations, the measurements are taken as described in the text.

since the typical relative velocity between the particle and the moon is high. This finding supports the observation of the heavily cratered front-side of the rotationally bound moons. For example, many large crater systems on Callisto (namely Valhalla, Asgard, Adlinda, Utgard, etc.) are located at the front-side.

Comparing class 2 and class 3, it seems that the influence caused by the flyby at Jupiter and the parallel or antiparallel direction of flight are similar in strength, producing similar numbercounts for these classes. More discreet or weaker effects like the selection of the parameter space and the behavior of the GA may influence the shape of the curves as well as the numbercounts for the classes. However, the overall shape of the individual components can be interpreted as Gaussian distributions, covering a large interval for possible close encounter velocities.

Recalling the distributions of semi-major axes in **Figure 2**, the second peak for Ganymede is caused by the stronger scattering of the classes *ret-pro* and *ret-ret*. This means that Jupiter scatters retrograde orbits stronger than prograde orbits at smaller distances to Jupiter. Many particles end up with high eccentricities, regardless of their initial values. Note that the simulation time of $t_{\max} = 165\,\mathrm{yr}$ indicates that these drastic changes are caused by a single or only a handful of close encounters with Jupiter. The stronger scattering toward higher values of the semi-major axes and eccentricities for Ganymede is intuitive, because particles experience a stronger acceleration by Jupiter during the flyby. The data implies that more than half of all particles even undergo the transition from eccentric to hyperbolic trajectories during the simulations (taking into account we only considered elliptic orbits initially). The comparison between the final semi-major axes indicates that the large peak (at Callisto) vanishes at closer distances to Jupiter (toward Ganymede) and gets scattered over a wider range at typically higher values.

For the inclination, the GA has a selection effect for the sector $a > 15\,\mathrm{AU}$ and $i > 160°$, although the initial setup for the first generation resembles a realistic inclination distribution. This is clearly a selection bias of the GA, as it forces the population to settle either in very low or very high inclination orbits. This settling is due to the fact that statistically the relative distances between the test particles and the Jupiter system are smaller if their orbital planes are aligned. This leads to higher fitness values at very low/high inclinations. However, as a first approximation the GA works well as a predictor for close encounters for prograde orbits. Therefore, we recommend using the GA only for prograde orbits at current state.

The comparison with the numerical integations (see **Figure 3**) reveals that the GA represents the overall close encounter situation quite well in the $a - e$ space and pretty well in the $a - i$ space when excluding retrograde orbits with very high inclinations. Both density distributions cover similar areas in the parameter spaces with almost all datapoints from the Lie measurements having their counterparts in the GA measurements. The statistical analysis via a Kolmogorov–Smirnov test even suggests a correlation between the density distributions for the $a - e$ phase space. Note that the simulations in the GA have a very short integration time of $t_{\max} = 165\,\mathrm{yr}$, which do not allow for modeling the long-term behavior of chaotic orbits. However, the GA still yields a good coverage of parameter space, even with this short simulation time. Interestingly, the GA also finds orbits which lie far beyond the initial parameter space of semi-major axes given in **Table 1**. This indicates, that the implementation of the mutation function of the GA is flexible enough to explore the parameter space beyond its limits. The $a-i$ parameter space is reproduced for inclinations up to $i = 160°$, although the statistical analysis suggests no significant correlation in density. As already stated, the high number of retrograde measurements is a selection bias of the GA.

For future work with the GA we may include the simulation time and further properties of the orbits in the fitness function in order to avoid too short simulations and make high inclinations less likely, for example. We may also include the TNO region as an important extension to the parameter space.

The GA can easily be adapted in order to efficiently measure actual collisions with any given object. In this work we measure only two collisions with Ganymede because as soon as the first particle has its closest approach within the Hill sphere, a completely new evolution process is started.

In summary, the results from the GA are not fully consistent compared to the classical approach because the underlying principles are different. GAs in general tend to find all possible solutions to a given problem rather than only the realistic, physical ones. However, in our implementation this effect can be overcome by optimizing the GA either to avoid clearly non-physical solutions or to enhance realistic solutions by increasing the simulation time, enlarging the parameter space and refining the functions for fitness, crossover and mutation.

5. DISCUSSION

The comparison with the Lie integrations reveal that the genetic n-body algorithm yield both a high number of physical as well as non-physical results. For example, an unrealistic high number of retrograde orbits are found while using a realistic probability distribution for the random initial inclinations. This is clearly a selection effect of the GA, as it finds that the probability for measuring retrograde encounters is significantly higher than for prograde ones. Several reasons such as a too short simulation time, too powerful crossover- and mutation functions or the choice of hyperparameters (such as the population size, number of generations, etc.) can be responsible for causing this high fraction non-physical results. However, it is expected that this high fraction can be significantly reduced by applying one or more of the following improvements:

- A higher simulation time for each generation enables more dynamical effects in general. Therefore, the GA will also tend to produce more physically motivated results.
- The fitness function can be refined to avoid solutions which are clearly non-physical. For example, one may introduce additional terms which depend on the inclination, the time of measurement, etc.
- Since crossover tends to find orbits within the initial random parameter space, this parameter space should be large enough, e.g., 50% larger compared to the parameter space of interest. Regions outside the inital random parameter space are only accessible via mutation.
- We found the mutation function to have a significant effect on the behavior of the GA's learning curve (generation vs. mean fitness). An optimized mutation function can boost the overall performance of the GA drastically, enabling the use of a higher simulation time, a larger population, obtaining more measurements, etc. with the same computation resources.
- A larger population is able to cover the parameter space more homogeneously and may reveal further close encounter families with low probabilities.
- Like in this work, the non-physical results can be efficiently filtered by comparing the results with classical approaches.

However, the GA also yields useable results, especially on the small-scale close encounter dynamics. We find it to be an efficient tool to get a rough idea of the underlying dynamics of the problem and the expected families of solutions before investigating into more detailed analysis with classical approaches. The GA supports the use of existing approaches rather than replacing them. In this work, the GA efficiently finds all possible close encounter geometries even beyond the initial parameter space with low computational effort. The measurements cover all major areas of the parameter spaces in semi-major axis, eccentricity and inclination. Even a weak correlation in the density distributions for the $a - e$ space is apparent when comparing the results of the GA with the long-term Lie integrations. This is quite interesting, as the used methods represent completely different approaches to the underlying problem of close encounters. This encourages more detailed studies with optimized algorithms. In theory, the GA can be used to study a variety of different problems in celestial mechanics, given the appropriate fitness function and adapted functions for crossover and mutation.

Apart from the behavior of the GA itself, relevant information is obtained from the measurements:

- The four classes, which are motivated by geometrical considerations, can be distinguished well in the datasets. The classes allow for a more detailed and structured way for analysing close encounter events in the Jovian System.
- One may distinguish between different impact scenarios depending on the impact velocity. For example, there is no need for analysing retrograde collisions if the particle is classified as class 1 and vice-versa for prograde collisions and class 4.
- There are significantly more retrograde than prograde encounters for both moons. This fact is supported by the heavily cratered front-side of the rotationally bound moons.
- As shown in **Figure 2**, most of the particles get scattered quite drastically in eccentricity during the close encounter. A high fraction may even end up in hyperbolic trajectories.
- Moreover, the distributions of semi-major axes reveal a double-peak structure for Ganymede in contrast to a single-peak structure for Callisto. This can be explained by a stronger scattering of the classes *ret-pro* and *ret-ret* at smaller distances to Jupiter.

AUTHOR CONTRIBUTIONS

PW wrote the code for the genetic algorithm and the transformations for the osculating elements. He wrote about 70% of the paper, especially the part inherent to the genetic algorithm, the core of this paper. MG has contributed performing the full orbital numerical part section, selecting the initial populations for this study, he suggested how to consider the close encounters with the Jupiter moons and he edited several parts of the paper for quality improvements in any section of the paper, apart the

description of the genetic algorithm. He wrote about 15% of the paper. TM helped in editing the quality of the paper and wrote about 15% of the paper. He also helped in the description of the genetic algorithm and performing the Kolmogorov-Smirnov test.

ACKNOWLEDGMENTS

The authors wish to thank the three referees for their comments which helped to significantly improve the manuscript. The authors acknowledge support by the FWF Austrian Science Fund projects S11603-N16 (PW and TM) and P23810-N16 (MG), respectively.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fspas.2018.00016/full#supplementary-material

REFERENCES

Bailey, B. L., and Malhotra, R. (2009). Two dynamical classes of Centaurs. *Icarus* 203, 155–163. doi: 10.1016/j.icarus.2009.03.044

Di Sisto, R. P., and Brunini, A. (2007). The origin and distribution of the Centaur population. *Icarus* 190, 224–235. doi: 10.1016/j.icarus.2007.02.012

Eggl, S., and Dvorak, R. (2010). "An introduction to common numerical integration codes used in dynamical astronomy," in *Dynamics of Small Solar System Bodies and Exoplanets*, eds J. Souchay and R. Dvorak (Berlin; Heidelberg: Springer), 431–480.

Fernández, J. A., and Sosa, A. (2015). Jupiter family comets in near-Earth orbits: are some of them interlopers from the asteroid belt? *Planet. Space Sci.* 118, 14–24. doi: 10.1016/j.pss.2015.07.010

Galiazzo, M. A., Bazsó, Á., Huber, M. S., Losiak, A., Dvorak, R., and Koeberl, C. (2013). A statistical dynamical study of meteorite impactors: a case study based on parameters derived from the Bosumtwi impact event. *Astron. Nachr.* 334, 936–939. doi: 10.1002/asna.201211964

Galiazzo, M. A., Wiegert, P., and Aljbaae, S. (2016). Influence of the Centaurs and TNOs on the main belt and its families. *Astrophys. Space Sci.* 361:371. doi: 10.1007/s10509-016-2957-z

Greeley, R., Klemaszewski, J. E., Wagner, R., and The Galileo Imaging Team (2000). Galileo views of the geology of Callisto. *Planet. Space Sci.* 48, 829–853. doi: 10.1016/S0032-0633(00)00050-7

Hanslmeier, A., and Dvorak, R. (1984). Numerical integration with lie series. *Astron. Astrophys.* 132:203.

Horner, J., Evans, N. W., and Bailey, M. E. (2004a). Simulations of the population of Centaurs - I. The bulk statistics. *Mthly Not. R. Astron. Soc.* 354, 798–810. doi: 10.1111/j.1365-2966.2004.08240.x

Horner, J., Evans, N. W., and Bailey, M. E. (2004b). Simulations of the population of Centaurs - II. Individual objects. *Mthly Not. R. Astron. Soc.* 355, 321–329. doi: 10.1111/j.1365-2966.2004.08342.x

Napier, B., Asher, D., Bailey, M., and Steel, D. (2015). Centaurs as a hazard to civilization. *Astron. Geophys.* 56, 6.24–6.30. doi: 10.1093/astrogeo/atv198

Press, W. H., and Teukolsky, S. A. (1988). Kolmogorov-Smirnov test for two-dimensional data. *Comput. Phys.* 2, 74–77. doi: 10.1063/1.4822753

Stewart, S. A., and Allen, P. J. (2002). A 20-km-diameter multi-ringed impact structure in the North Sea. *Nature*, 418, 520–523. doi: 10.1038/nature00914

Turing, A. M. (1950). Computing machinery and intelligence. *Mind* 59, 433–460.

Volk, K., and Malhotra, R. (2008). The Scattered Disk as the Source of the Jupiter Family Comets. *apj*, 687, 714–725. doi: 10.1086/591839

Winter, P. M., Maindl, T. I., Schäfer, C., and Galiazzo, M. A. (2017). Reconstruction of Callisto's Valhalla basin using n-body and SPH simulations. *Eur. Planet. Sci. Congr.* 11, 1–2.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix

The principle of a genetic algorithm can be described in a three-step process:

1. The first generation is initialized in the system. The initial DNA of the individuals should cover the full parameter space in which at least one solution may exist. If there is no solution within the parameter space, no solution may be found by the GA. Note that the final solution of the problem is not only affected by the DNA, but also by other factors like the dynamics of the system itself, the fitness function, the crossover function, the mutation function, and the number of generations.

2. The population evolves within the system and the fitness of each individual is measured. Usually this is the part which is computationally most demanding. It is crucial to score each individual correctly with the fitness function to enhance finding the final solution to the given problem.

3. At the end of each generation the individuals with the least fitness die out and are replaced by new individuals (children) via crossover. Moreover, a small mutation is applied to the whole population. We then have a new population using the DNA of the previous one. Steps 2 and 3 are repeated until the fitness of the population reaches a certain level, which corresponds to the solution of the problem. At least one individual has to die out and there must be at least one new child at each generation. The number of children has to be equal to the number of dying individuals in order to keep the total number of individuals in the population equal for all generations. Note that each new generation is likely to have a higher mean fitness than the previous one which can be interpreted as a learning process.

Population

The number of individuals in the population is crucial for a GA to work. Choosing a too low number, the individuals tend to become all equal after a few generations. Since they are all equal, the learning process can only take place via mutation, which is often insufficient to solve the problem properly. This situation significantly hinders finding solutions within limited computation time. The population may also get stuck in local maxima (of the fitness curve) easily and no good solution will be found.

If the number of individuals is too high on the other hand, many generations are needed to observe a learning effect. One may also encounter performance problems because each individual has to evolve in the system at each generation. However, in principle it is good to prefer a rather high number of individuals for several reasons such as coverage of the parameter space, minimizing the risk of getting stuck in local maxima, and an overall healthier population of competing families using different approaches to solve the given problem. Considering these facts, one may choose an individual number as high as possible for given computation power and memory space. Note that for GAs it is often relatively simple to implement parallel computing with either multiple CPUs or with GPUs. A higher number of individuals may then be used.

DNA

The properties of each individual are summarized in their DNA. For example, the DNA of an asteroid may be its orbital parameters as well as its mass, shape, albedo, composition, etc. The DNA depends on the nature of the problem itself. In general it is good to have as much relevant information in the DNA as possible to include as many possible aspects of the problem.

Fitness Function

The fitness function scores each individual by its behaviour (which itself is based on its DNA) within the system. The better an individual can solve the given problem, the higher the fitness should get. Keep in mind that the problem to be solved has to be imprinted in the fitness function. Since we want to analyse close encounters, our fitness function is given by the inverse of the squared relative distance between a particle and the corresponding Jovian moon. Algorithm 2 shows the implementation used for the genetic n-body algorithm. The fitness for each individual in one generation is given by the maximum fitness at any timestep of the n-body integration.

Algorithm 1: measure fitness

for *i in individuals* **do**
- $d_{rel,i} = ||\vec{x_m} - \vec{x_i}||$
- $f = d_{rel,i}^{-2}$
- $f_i = max(f_i, f)$

Generations and Evolution

During the iterative searching process (evolution), the population ideally gets better and better with each generation to find good solutions to the problem. Nevertheless, there may be generations with lower fitness than the previous ones. If the population settles in a local maximum, it has to overcome it again. This means, that at least one individual has to find a higher maximum to guide other individuals towards the respective direction in the parameter space. Then, during the process of overcoming, the mean fitness of the population may even decrease for a short period before rising again to new maximum values.

Natural Selection and Crossover

Natural selection is another key component of a GA. Relatively fit 'parents' are allowed to pass their DNA to the following generation by creating a new 'child'. The crossover function can take many different forms, as long as it conserves and incorporates the parents' DNA. In many cases, a simple crossover function (mean values between parent A and parent B) is sufficient to produce children which are likely to have a higher fitness than the mean fitness of the population. Typically, the least fit individuals are replaced by new children to keep the population size constant. The crossover and mutation functions have to be adjusted to each other to ensure a steep learning curve. If the crossover is too strong, the population gets stuck in local maxima and mutation is not sufficient to overcome them again. If the mutation is too strong, the information of the

DNA gets destroyed too fast and no learning process can happen. The fitness of the population may even decrease and settle at a very low value. Algorithm 3 shows the implementation used for the genetic n-body algorithm. The vector y contains the 6 initial orbital elements for the respective object. A 1:1 copy is being made using a parent with high fitness.

Algorithm 2: crossover

for *ch in children* **do**
  choose parent pa: $f_{pa} > f_{ch}$
  $\vec{y_{ch}} = \vec{y_{pa}}$

Mutation Function

The mutation function is used to apply small, random variations to the DNA of every individual after every generation. These variations are needed in order to overcome local maxima and to keep the population 'healthy'. This means, that the DNA of the individuals can not degenerate to the same values for the whole population where no learning process can happen any more. Algorithm 4 shows the implementation used for the genetic n-body algorithm. Each orbital element of each individual is mutated separately using random numbers between 0 and 1. The scaling factor ζ controls the amplitude of the mutation rate χ.

Algorithm 3: mutation

if $\overline{f_t} > 20 \cdot \overline{f_{t-h}}$ *or* $\overline{f_t} < 0.2 \cdot \overline{f_{max}}$ **then**
  $\zeta = \zeta \cdot 0.8$
if $(\overline{f_t} > 0.99 \cdot \overline{f_{t-h}}$ *and* $\overline{f_t} < 1.01 \cdot \overline{f_{t-h}})$ *or* $max(\vec{f_t})/\overline{f_t} < 2$ **then**
  $\zeta = \zeta \cdot 2$
calculate σ_a, σ_e, σ_i, σ_ω, σ_Ω, σ_M
$\vec{\chi} = \zeta \cdot \vec{\sigma}$
for *i in individuals* **do**
  $\vec{y_i} = \vec{y_i} + random \cdot \vec{\chi}$

A versatile smoothed particle hydrodynamics code for graphic cards

4.

After investigating initial conditions of impactors, we focused on modeling collision processes themselves. The complex interactions involved during collisions requires the use of visco-elastic fluid simulations. A well-established code framework for performing SPH simulations was published as a journal article in 2020 [105]. This work acts as a minor contribution in the context of this thesis.

Extensive use of the presented SPH code resulted in the creation of a dataset of protoplanetary collisions. It was used for studying water transport [74], as well as trajectories of post-collision fragments [24] (also see Chapter 5). Later, we considered this dataset as the precursor of an improved and extended variant, which was ultimately published in [135] (also see Chapter 6).

Astronomy and Computing 33 (2020) 100410

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

A versatile smoothed particle hydrodynamics code for graphic cards ☆

C.M. Schäfer [a,*], O.J. Wandel [a], C. Burger [a,b], T.I. Maindl [b], U. Malamud [c], S.K. Buruchenko [d], R. Sfair [e], H. Audiffren [a,f], E. Vavilina [a], P.M. Winter [g]

[a] Institut für Astronomie und Astrophysik, Auf der Morgenstelle 10, 72076 Tübingen, Germany
[b] Department of Astrophysics, University of Vienna, Tuerkenschanzstr. 17, 1180 Vienna, Austria
[c] Physics Department, Technion - Israel institute of Technology, Haifa, 3200002, Israel
[d] South Ural State University, 456776 Zababachina 9-115 Snezhinsk, Chelyabinsk Region, Russia
[e] Sao Paulo State University, Guaratingueta, Sao Paulo, CEP 01049-010, Brazil
[f] Aix-Marseille University, 3 Place Victor Hugo, 13003 Marseille, France
[g] Institute for Machine Learning, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria

ARTICLE INFO

Article history:
Received 14 February 2020
Accepted 10 August 2020
Available online 19 August 2020

Keywords:
Smoothed particle hydrodynamics
GPU-computing
Hydrodynamics
Continuum mechanics

ABSTRACT

We present the second release of the now open source smoothed particle hydrodynamics code `miluphcuda`. The code is designed to run on Nvidia CUDA capable devices. It handles one to three dimensional problems and includes modules to solve the equations for viscid and inviscid hydrodynamical flows, the equations of continuum mechanics using SPH, and self-gravity with a Barnes–Hut tree. The covered material models include different porosity and plasticity models. Several equations of states, especially for impact physics, are implemented. The basic ideas of the numerical scheme are presented, the usage of the code is explained and its versatility is shown by means of different applications. The code is hereby publicly available.

1. Introduction

The Lagrangian meshfree particle method smoothed particle hydrodynamics (SPH) was invented by Lucy (1977) and Gingold and Monaghan (1977) to solve the hydrodynamic equations for compressible flows in astrophysical applications. During the last 40 years, it has been widely used in astrophysics to tackle a substantial number of problems. In the last two decades the field of applications have expanded vividly: SPH has been established as one of the major concepts for fluid animation and rigid-fluid coupling in computer graphics (Ihmsen et al., 2014; Akinci et al., 2012), it was extended to continuum mechanics to model high-velocity impacts (Libersky and Petschek, 1991; Stellingwerf and Wingate, 1994), it is widely used in geosciences to model landslides and granular media (Yu et al., 2018; Bui et al., 2008), and the method has been improved and extended substantially to model phase separating fluid mixtures (Zander et al., 2018), to solve the magnetohydrodynamical equations (Price, 2012) and to model relativistic flows (Rosswog, 2015). It has also been used to solve the non-linear Schrödinger equation (Mocz and Succi, 2015). The SPH scheme has proven to be an advantageous method to solve all kind of different partial differential equations (PDEs). Since the most astrophysical applications have to include gravity as the most dominating force, gravitational interactions had been implemented into the SPH scheme from the early development on. Moreover, not only external gravitational sources but also the self-gravity of the simulated flow or solid body can be considered. Especially tree algorithms for the treatment of the long range gravitational force have been combined with SPH (Hernquist and Katz, 1989).

Several SPH codes have been developed in the recent years and we present in the following the codes that have been published under the GNU General Public License. The list is most certainly incomplete and we want to apologize any omitted code. The first popular code GADGET-2 by Springel (2000) is a massively parallel TreeSPH code, which is still widely used in the astrophysical community. A more recent and modern code from the astrophysical community is the PHANTOM code by Price et al. (2017). It is a parallel, modular and low-memory smoothed particle hydrodynamics and magnetohydrodynamics code with focus on stellar, galactic, planetary and high energy astrophysics. The second version of the GASOLINE code, GASOLINE2, has also been published under the GPLv2 license (Wadsley et al., 2017). The code concentrates also on astrophysical problems with a special focus on cosmology. Another versatile SPH code written is SPHERAL++ by Owen (2001). Since the field of applications of the scheme has been broadened thoroughly in the recent decade, modern and versatile codes outside of the astrophysical community have emerged. PySPH is reproducible and high-performance

☆ This code is registered at the ASCL with the code entry ascl:1911.023.

* Corresponding author.
E-mail address: ch.schaefer@uni-tuebingen.de (C.M. Schäfer).

https://doi.org/10.1016/j.ascom.2020.100410
2213-1337/

framework for smoothed particle hydrodynamics is developed by Ramachandran (2016). It is implemented in a mix of Python and Cython. The user writes pure Python code and the high-performance computing (HPC) code is generated automatically. PySPH includes solver for fluid and solid mechanics and supports both OpenMP and MPI. Another code to study free-surface flows is DualSPHysics (Crespo et al., 2015), which is the very first SPH framework featuring a modern design and GUI interface (DesignSPHysics). AQUAgpusph (Cercos-Pita, 2015) is a 3D, OpenCL accelerated SPH solver for GPUs. GPUSPH by Hérault et al. (2010), which has been originally a port of the freely available SPHysics Fortran code to Nvidia's CUDA library, is now a standalone project and supports multiple GPUs. PersianSPH is an open-source code and features elastic–plastic soil behaviour and elastic–plastic solid behaviour (Korzani et al., 2017). SPHERA (Amicarelli et al., 2017) is also publicly available and supports the simulation of flooding including the transport of solid bodies, bed-load transport, damage on electrical substations and fast landslides in rocks and solids. Another versatile code within the astrocommunity is SWIFT by Schaller et al. (2018). Recent simulations with SWIFT included up to 100 million SPH particles (Kegerreis et al., 2019). It can directly read GADGET-2 output files and shows tremendous scaling.

Here, we present the updated version of our GPU SPH code `miluphcuda`.[1] The new version is published under the GPLv3 license and available on github via https://github.com/christophmschaefer/miluphcuda. Originally, the code has been developed to model collisions between solid, self-gravitating objects in an astrophysical context. The basic implementation is described in Schäfer et al. (2016), and until today, the code is solely applied in the astronomy community. The code handles one-, two- and three-dimensional problems. It has been used to study the activation of porous and non-porous asteroids (Haghighipour et al., 2018, 2016), collisions between Earth and its past moons (Malamud et al., 2018), collisional formation of moons (Malamud et al., 2020), impact cratering (Wandel et al., 2017), the formation of Arrokoth (Grishin et al., 2020), transfer and loss of water in hit-and-run collisions (Burger and Schäfer, 2017; Burger et al., 2018) and sampling of granular material (Schäfer et al., 2017) among other applications. Recent extensions include further porosity models, a module to solve the Navier–Stokes equations, and more sophisticated equation of states like ANEOS.

The outline of this paper is as follows. The next section summarizes the characteristics of the code as presented in Schäfer et al. (2016) and provides a comprehensive description of the new extensions. Section 3 shows some of the recent applications and numerical tests to provide an overview of the possibilities of the code. We will conclude in the final section and present details about the code and its structure in the appendix.

2. Physical and numerical method

In this section, we present the different physical models, their numerical implementations, and further aspects of `miluphcuda`.

The basic idea of the SPH scheme is to transform a system of partial differential equations to a system of ordinary differential equations, which can be solved by standard integrators. For this transformation the partial differential equations are approximated in two steps: First, with an average over all spatial field quantities by the help of the convolution with the kernel function and second by a discretization of this average. For an introduction to the fundamental ideas of SPH, we refer to the comprehensive review articles by Benz (1990), Monaghan (1992), Monaghan (2005), Liu and Liu (2010) and Monaghan (2012). In the following, we present the SPH equations for viscid and inviscid hydrodynamics and continuum mechanics as implemented in the code. The different modules implemented in the code allow for the simulation of various different astrophysical objects: By the use of the sole hydrodynamical module, one can model a gaseous accretion disc, both viscous and inviscid. By adding the self-gravity module, the gravitational field produced by the disc is also included. The solid module allows to model brittle, rocky materials, such as smaller, non-porous planetesimals. Adding the porosity module, the code can be used to simulate porous objects like porous asteroids, comets and Kuiper belt objects. Larger planetesimals or dwarf planets can be simulated with a combination of the solid or hydro module with the self-gravity module. The different strength models included in the solid module allow to model brittle or granular materials, e.g., a layer of granular regolith on a brittle and porous asteroid.

2.1. Hydrodynamics

The flow of a fluid is completely described by its velocity field $\mathbf{v}(\mathbf{x}, t)$, and two thermodynamic field quantities, where usually the density $\varrho(\mathbf{x}, t)$ and the pressure $p(\mathbf{x}, t)$ are chosen. The conservation laws for mass, momentum and energy determine the dynamics and lead to a set of PDEs which can be solved with SPH for all locations $\mathbf{x}$ and times t.

The Euler equation describes the flow of an inviscid, ideal fluid. It reads in Lagrangian representation

$$\varrho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\nabla p, \tag{1}$$

with the velocity $\mathbf{v}$, the density ϱ and the pressure p. By the use of the identity

$$\frac{1}{\varrho}\nabla p = \frac{1}{\varrho^{2-\lambda}}\nabla\left(\frac{p}{\varrho^{\lambda-1}}\right) + \frac{p}{\varrho^{\lambda}}\nabla\frac{1}{\varrho^{1-\lambda}}, \tag{2}$$

one can derive the following two equivalent SPH representations,

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = -\sum_b m_b \frac{p_a + p_b}{\varrho_a \varrho_b}\nabla W_{ab} \tag{3}$$

for $\lambda = 1$, and

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = -\sum_b m_b \left[\frac{p_a}{\varrho_a^2} + \frac{p_b}{\varrho_b^2}\right]\nabla W_{ab} \tag{4}$$

for $\lambda = 2$.

2.2. Viscous flows

The extension of the Euler equation for viscous, Newtonian fluids is the Navier–Stokes equation. It reads in Lagrangian representation

$$\varrho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\nabla p + \nabla \cdot T, \tag{5}$$

where T denotes the viscous stress tensor. The viscous stress tensor accounts for the internal friction of the flow. For a Newtonian fluid, where the viscosity depends linearly on the gradient of the velocity field, it is given by

$$T^{\alpha\beta} = \eta\left[\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} - \frac{2}{3}\delta^{\alpha\beta}\frac{\partial v^\gamma}{\partial x^\gamma}\right] + \zeta\delta^{\alpha\beta}\frac{\partial v^\gamma}{\partial x^\gamma} \tag{6}$$

with the dynamic viscosity coefficient η and the bulk viscosity coefficient ζ. The kinematic viscosity coefficient ν is the ratio between the dynamic viscosity coefficient and the density of the fluid, $\nu = \eta/\varrho$. The term in the parentheses in Eq. (6)

[1] `miluph` is pronounced [mailʌv]

represents the traceless part of the viscous stress tensor and is called viscous shear stress tensor. The numerical treatment and solution of the Navier–Stokes equation is demanding because of the second derivatives of the velocities. We follow the approach by Flebbe et al. (1994) and calculate the derivatives of the velocity components for particle a according to

$$\frac{\partial v_a^\alpha}{\partial x^\beta} = \sum_b \frac{m_b}{\varrho_b}\left(v_b^\alpha - v_a^\alpha\right)\frac{\partial W_{ab}}{\partial x^\beta} \equiv v_a^{\alpha\beta}. \tag{7}$$

The viscous stress tensor for particle a can be calculated via

$$T_a^{\alpha\beta} = \eta_a\left[v_a^{\alpha\beta} + v_a^{\beta\alpha}\right] - \frac{2}{3}\delta^{\alpha\beta}v_a^{\gamma\gamma}. \tag{8}$$

In a second step, the accelerations due to the viscous stress are determined via

$$\frac{\mathrm{d}v_a^\alpha}{\mathrm{d}t} = \sum_b m_b \frac{T_a^{\alpha\beta} + T_b^{\alpha\beta}}{\varrho_a\varrho_b}\frac{\partial W_{ab}}{\partial x^\beta}. \tag{9}$$

If the partial derivatives of the velocities are not calculated with Eq. (7), but with the numerical equivalent SPH sum

$$\frac{\partial v_a^\alpha}{\partial x^\beta} = \frac{1}{\varrho_a}\sum_b m_b\left(v_b^\alpha - v_a^\alpha\right)\frac{\partial W_{ab}}{\partial x^\beta}, \tag{10}$$

the accelerations are determined via

$$\frac{\mathrm{d}v_a^\alpha}{\mathrm{d}t} = \sum_b m_b\left[\frac{T_a^{\alpha\beta}}{\varrho_a^2} + \frac{T_b^{\alpha\beta}}{\varrho_b^2}\right]\frac{\partial W_{ab}}{\partial x^\beta}. \tag{11}$$

The bulk part of the Navier–Stokes equation can also be used to define an artificial viscosity that prevents mutual particle penetration and does not yield unwanted spurious shear. A possible ansatz for an artificial bulk viscosity was successfully applied in planet–disc interaction simulations in Schäfer et al. (2004): the bulk part of the viscous stress tensor is $\zeta\nabla\cdot\mathbf{v}$, which leads to an additional acceleration for particle a

$$\left.\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right|_{\text{bulk}} = \sum_b m_b\zeta_{ab}\frac{(\nabla\cdot\mathbf{v})_a + (\nabla\cdot\mathbf{v})_b}{\varrho_a\varrho_b}\nabla_a W_{ab}, \tag{12}$$

or

$$\left.\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right|_{\text{bulk}} = \sum_b m_b\zeta_{ab}\left[\frac{(\nabla\cdot\mathbf{v})_a}{\varrho_a^2} + \frac{(\nabla\cdot\mathbf{v})_b}{\varrho_b^2}\right]\nabla_a W_{ab}, \tag{13}$$

in case of the other SPH representation, respectively. The bulk viscosity coefficient ζ_{ab} for the interaction between particles a and b is

$$\zeta_{ab} = \begin{cases} -f\bar{h}_{ab}^2\overline{(\nabla\cdot\mathbf{v})}_{ab}\bar{\varrho}_{ab} & \text{for } (\mathbf{v}_a - \mathbf{v}_b)\cdot(\mathbf{x}_a - \mathbf{x}_b) \\ 0 & \text{else.} \end{cases} \tag{14}$$

The term $\bar{q}_{ab}$ denotes an abbreviation for the average $\frac{1}{2}(q_a + q_b)$ for any quantity q and f a parameter to scale the effect of the artificial bulk viscosity, usually set to 1.

2.3. Solid body mechanics

The implementation of solid body mechanics with fragmentation in our code follows the ideas and concepts of Stellingwerf and Wingate (1994) and Benz and Asphaug (1994). We refer for the thorough description of our implementation in Schäfer et al. (2016) and present the basic equations in the following.

For a perfect elastic material, the conservation of momentum reads

$$\varrho\frac{\mathrm{d}v^\alpha}{\mathrm{d}t} = \frac{\partial\sigma^{\alpha\beta}}{\partial x^\beta}, \tag{15}$$

where the stress tensor $\sigma^{\alpha\beta}$ is built up by the pressure p and the traceless deviatoric stress tensor $S^{\alpha\beta}$

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + S^{\alpha\beta}. \tag{16}$$

The conservation of the internal energy u is given by

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{p}{\varrho}\frac{\partial v^\alpha}{\partial x^\alpha} + \frac{1}{\varrho}S^{\alpha\beta}\dot{\varepsilon}^{\alpha\beta}. \tag{17}$$

Here, $\dot{\varepsilon}^{\alpha\beta}$ denotes the strain rate tensor for small deformations

$$\dot{\varepsilon}^{\alpha\beta} = \frac{1}{2}\left(\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha}\right). \tag{18}$$

Additionally, an equation for the time evolution of the deviatoric stress tensor $S^{\alpha\beta}$ is required to describe the dynamics of the solid body, which is called the constitutive equation. For a perfectly elastic behaviour, we choose Hooke's law, where the stress depends linearly on the strain, which yields using the Jaumann rate eventually

$$\frac{\mathrm{d}S^{\alpha\beta}}{\mathrm{d}t} = 2\mu\left(\dot{\varepsilon}^{\alpha\beta} - \frac{1}{3}\delta^{\alpha\beta}\dot{\varepsilon}^{\gamma\gamma}\right) + S^{\alpha\gamma}R^{\gamma\beta} - R^{\alpha\gamma}S^{\gamma\beta}, \tag{19}$$

with the shear modulus μ and the rotation rate tensor $R^{\alpha\beta}$

$$R^{\alpha\beta} = \frac{1}{2}\left(\frac{\partial v^\alpha}{\partial x^\beta} - \frac{\partial v^\beta}{\partial x^\alpha}\right). \tag{20}$$

2.4. Strength models

A solid material does not deform perfectly elastic for all experienced strains. Once the applied strain exceeds the elastic limit, the deformation becomes plastic. The transition from an elastic to a plastic state of a substance is characterized by a stress state. The stress at which the material starts deforming plastically is called yield stress, and the condition for plastic deformation is the yield criterion. For an isotropic medium, the yield criterion is a symmetric function of the principle stresses σ_i

$$f(\sigma_1, \sigma_2, \sigma_3) = \text{const.} \tag{21}$$

The constant on the righthandside is material dependent and related to the particular yield limit. Since f is a symmetric function of the principle stresses, the stress components can be expressed by the invariants of the stress tensor

$$f(I_1, J_2, J_3) = \text{const.} \tag{22}$$

In the original code, we have implemented the elementary von Mises yield strength to model plastic behaviour of solid bodies, and noted already the shortcoming of this simplistic model. Following Collins et al. (2004) and Jutzi (2015), we have added two additional strength models suitable for the modelling of granular material and a pressure dependent yield strength model, which is applicable to rocks and interplays with the fragmentation model for brittle materials by Grady and Kipp (Schäfer et al., 2016).

2.4.1. Von Mises yield stress

In the von Mises yield stress model (von Mises, 1913), the deviatoric stress is limited once it reaches a constant yield strength limit Y_M. At first, the second invariant of the stress tensor is computed via

$$J_2 = \frac{1}{2}S^{\alpha\beta}S_{\alpha\beta}, \tag{23}$$

for each individual particle, and the limiting factor f_Y is calculated according to

$$f_Y = \min\left[Y_M^2/3J_2, 1\right]. \tag{24}$$

Then, the deviatoric stress, which was integrated using Eq. (19), is limited

$$S^{\alpha\beta} \to f_Y S^{\alpha\beta}. \tag{25}$$

2.4.2. Drucker–Prager and Mohr–Coulomb yield criterion

The Drucker–Prager yield strength depends on the pressure and is applicable to model rocky materials. The condition for plasticity reads

$$\sqrt{J_2} = A - BI_1, \tag{26}$$

where I_1 denotes the first invariant of the stress tensor ($I_1 = \mathrm{tr}[\sigma^{\alpha\beta}]$) and J_2 the second invariant given by Eq. (23). The two material constants A and B are the so-called Drucker–Prager constants. The criterion is related to the Mohr–Coulomb yield criterion, and the constants A, B can also be expressed in terms of cohesion c and angle of internal friction φ of the material via

$$A = \frac{6c\cos\varphi}{\sqrt{3}\,(3-\sin\varphi)}, \quad B = \frac{2\sin\varphi}{\sqrt{3}\,(3-\sin\varphi)}. \tag{27}$$

2.4.3. Pressure dependent yield criterion in combination with damage model

In our fracture model (Schäfer et al., 2016), the reduction in strength under tensile loading is expressed via a scalar damage variable $0 \leq D \leq 1$, where $D = 0$ corresponds to intact material and $D = 1$ to a totally fractured material that cannot undergo any tension or deviatoric stress. In this sense, a material with $D = 0.5$ experiences half the deviatoric stress and tension of an undamaged material with $D = 0$. In order to relate the fragmentation model with the yield strength for plastic flow, Collins et al. (2004) calculate a damage dependent yield strength in the following way. Undamaged, intact substances have a pressure dependent yield strength of

$$Y_P = c + \frac{\mu p}{1 + \mu p/(Y_M - c)}, \tag{28}$$

where μ is the coefficient of internal friction, given by $\mu = \tan\varphi$, c is the cohesion and Y_M the shear strength at infinite pressure. For totally damaged materials, the deviatoric stress is usually set to zero. However, as pointed out by Jutzi (2015), it is important to take into account the friction in totally damaged material, e.g., shattered rocks. Therefore, for totally damaged materials with $D = 1$, the yield strength is not vanishing but dominated by the pressure and given by

$$Y_D = \mu_D p, \tag{29}$$

where μ_D now denotes the coefficient of internal friction of the totally damaged, granular material. For partially damaged materials, the yield strength is determined according to

$$Y = (1 - D)Y_P + DY_D, \tag{30}$$

while $Y \leq Y_P$ is ensured. As soon as one particle exceeds the yield strength, the deviatoric stress is reduced according to Eq. (25), with the limiting factor

$$f_Y = \min\left[Y/\sqrt{J_2}, 1\right]. \tag{31}$$

2.5. Porosity models

Macroscopic porosity can be easily modelled with SPH by simply adding macroscopic voids on the scale of the particle volume to the initial distribution. However, sub-resolution porosity can be troublesome. Various sub-resolution porosity models exist in the literature and are applied to porous materials from metals to rocks. Although originally the $p - \alpha$-model was developed to study impacts into metal shielding, we have first implemented it for the simulation of porous, rocky objects such as comets and asteroids.

2.5.1. The p-α-model

The p-α-model by Herrmann (1969), which was modified by Jutzi et al. (2009), relates the porous pressure p of a material to the solid pressure p_s and the distention α. The main idea is to separate the volume change in a porous material into the pore collapse of the porous material and the compression of the matrix material. The distention is defined by

$$\alpha = \frac{\varrho_s}{\varrho} = \frac{1}{1-\phi}, \tag{32}$$

with the solid density ϱ_s, the porous density ϱ and the porosity ϕ. The porous pressure is calculated with the empirically found equation using the distention α and the pressure of the solid material p_s (Carroll and Holt, 1972)

$$p = \frac{1}{\alpha} p_s(\varrho_s, u_s) = \frac{1}{\alpha} p_s(\alpha\varrho, u), \tag{33}$$

where ϱ_s and u_s are the density and the energy of the solid, and ϱ and u of the porous material. The crush-curve relates the pressure to the porosity and yields a maximum pressure which the porous material can undergo before crushing. This gives an additional boundary condition for the pressure in the simulation. Two crush-curves are currently implemented. One being quadratic and the other having two regimes with parameters gained by fitting function (34) to the crush-curve of porous pumice, using a compression testing machine. In the plastic regime the distention α is computed as follows

$$\alpha(p) = \begin{cases} (\alpha_e - \alpha_t)\frac{(p_t - p)^{n_1}}{(p_t - p_e)^{n_1}} + (\alpha_t - 1)\frac{(p_s - p)^{n_2}}{(p_s - p_e)^{n_2}} + 1 & \text{if } p_e < p < p_t \\ (\alpha_t - 1)\frac{(p_s - p)^{n_2}}{(p_s - p_e)^{n_2}} + 1 & \text{if } p_t \leq p < p_s \\ 1 & \text{if } p \geq p_s, \end{cases} \tag{34}$$

where p_e is the pressure where the transition from the elastic to the plastic regime happens, p_s the pressure where the porous material is fully compacted and the density equals the matrix density ($\alpha = 1$) and the pressure p_t indicates a transition between the two regimes with different slopes n_1 and n_2 (Jutzi et al., 2009).

The distention is evolved in time depending on the pressure as follows

$$\dot{\alpha} = \frac{\mathrm{d}\alpha}{\mathrm{d}p}\frac{\mathrm{d}p}{\mathrm{d}t}. \tag{35}$$

One of the most important aspects of the p–α-model implementation is the check function for the timestep. The relation between the pressure and the distention is an implicit. However, to solve it explicitly, one has to make sure that the distention is resolved correctly. Hence, the distention has to follow the crush-curve since each distention value relates to a specific maximum pressure value. If the integrated distention yields a pressure beyond the maximum pressure for this specific distention value, the time step has been too coarse, the crush-curve is not reproduced correctly and the timestep has to be reduced, the integration step has to be repeated. To ensure this critical boundary condition our integrators have either an distention or a pressure check function where the change of the previous to the current value is calculated and if these values are too large the timestep is decreased accordingly, and the current step is recalculated with the lowered timestep. In order to validate the porous model, we have simulated the impact of a nylon, spherical projectile into a cube of pumice in three dimensions. Fig. 1 shows the pressure-distention values of the SPH particles from the impact. The distention α lies exactly on the crush-curve and thus resolving the timestep accordingly with crush-curve parameters shown in Table 1. The simulation was a nylon sphere impacting into a pumice cube. The parameters for the Tillotson equation of state (EOS) are those of

Table 1
Parameters for the crush-curve which were used for the plot and the simulation in Fig. 1.

p_e [Pa]	p_t [Pa]	p_s [Pa]	α_0	α_t	n_1	n_2
10^6	6.8×10^7	2.13×10^8	3.45	1.90	12	3

Table 2
Parameters for the crush-curve which were used for the plot and the simulation in Fig. 2.

p_e [Pa]	p_s [Pa]	c_0 [ms^{-1}]	c_e [ms^{-1}]
8×10^7	7×10^8	5.35×10^3	4.11×10^3

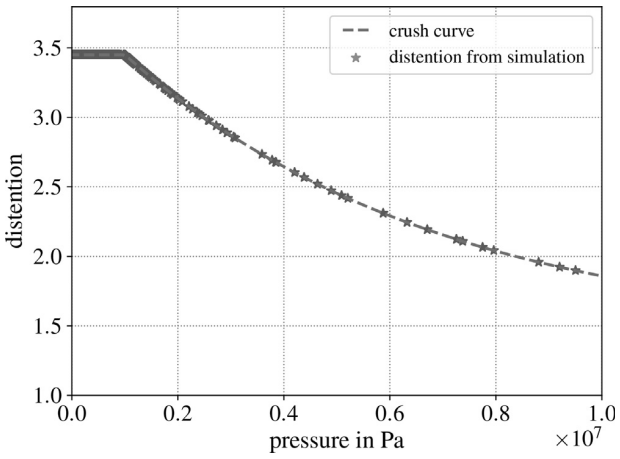


Fig. 1. Relation between the distention α and the pressure p. The initial distention is $\alpha_0 = 3.45$ and until p_e is reached the deformation is elastic and $\frac{d\alpha}{dt} = 0$.

solid rock with a volume $V = 216\,\text{cm}^3$ and a matrix density $\varrho_s = 2325.3\,\text{kg/m}^3$. The impactor sphere has the Tillotson EOS parameters of nylon with a radius $r = 7\,\text{mm}$, a density $\varrho_s = 1185\,\text{kg/m}^3$ and an impact velocity $v_{\text{imp}} = 2.58\,\text{km s}^{-1}$.

A shock wave test was performed which travels in one spatial dimension. The material was porous aluminium with an initial distention of $\alpha_0 = 1.275$. The crush-curve parameters are shown in Table 2 and the Tillotson EOS parameters are those of aluminium. The cylinder has a height $h = 2\,\text{cm}$ and a radius of $r = 0.2\,\text{cm}$. A constant velocity is set in the z-direction with $v_z = -45.8\,\text{cm s}^{-1}$. The quadratic crush-curve

$$\alpha = 1 + (\alpha_e - 1)\frac{(p_s - p)^2}{(p_s - p_e)^2} \tag{36}$$

was used and $\left[\frac{d\alpha}{dp} \neq 0\right]_{\text{elastic}}$ is calculated using

$$\left[\frac{d\alpha}{dp}\right]_{\text{elastic}} = \frac{\alpha^2}{K_0}\left[1 - \left(\frac{1}{h(\alpha)^2}\right)\right] \tag{37}$$

where $K_0 = c_0^2 \varrho_0$ and

$$h(\alpha) = 1 + (\alpha - 1)\frac{c_e - c_0}{c_0(\alpha_e - 1)}, \tag{38}$$

with $\alpha_e = \alpha(p_e)$. The results are shown in Fig. 2. We find perfect agreement with the simulation presented by Jutzi et al. (2009).

2.5.2. The ε-α-model

Another porosity model has been implemented into the code which is the so called ε-α-model (Collins et al., 2011). The main difference between this and the above mentioned p–α-model is

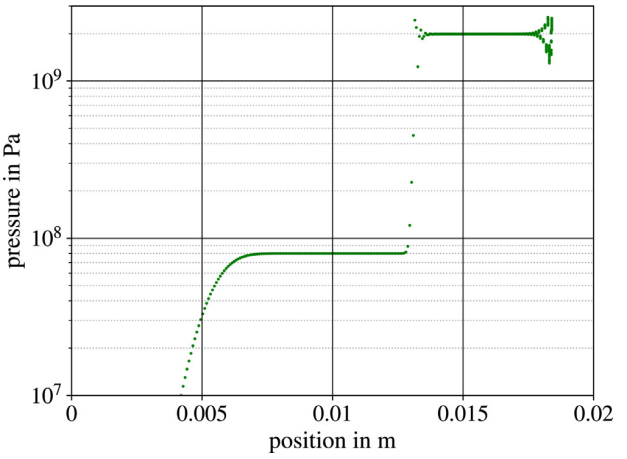


Fig. 2. One dimensional shock wave in a three dimensional porous aluminium cylinder. The pressure is plotted as a function of the vertical position of the SPH particle in the cylinder. The plot shows two waves with an elastic precursor followed by the actual shock wave.

the calculation of the distention. In this model the distention is calculated using the volumetric strain ε_V which is given by the change in volume of an element divided by the initial volume leading to following expression:

$$\varepsilon_V = \int_{V_0}^{V'} \frac{dV}{V} = \ln\left(\frac{V'}{V_0}\right), \tag{39}$$

with V_0 as the initial and V' as the updated volume. The volumetric strain can be related to the distention α

$$\varepsilon_V = \ln\left(\frac{V'}{V_0}\right) = \ln\left(\frac{V'}{V_s}\frac{V_s}{V_0}\right) = \ln\left(\frac{\alpha}{\alpha_0}\right), \tag{40}$$

with an ideally constant matrix volume V_s. The compaction function is defined as

$$\alpha(p) = \begin{cases} \alpha_0 e^{\kappa(\varepsilon_V - \varepsilon_e)} & \text{if } \varepsilon_e > \varepsilon_V \geq \varepsilon_x \\ 1 + \left(\alpha_0 e^{\kappa(\varepsilon_x - \varepsilon_e)} - 1\right)\left(\frac{\varepsilon_c - \varepsilon_V}{\varepsilon_c - \varepsilon_x}\right)^2 & \text{if } \varepsilon_x > \varepsilon_V > \varepsilon_c \\ 1 & \text{if } \varepsilon_V \leq \varepsilon_c, \end{cases} \tag{41}$$

with ε_e as the elastic–plastic transition strain, ε_x the threshold strain for the transition between the two compaction regimes, ε_c the threshold for full compaction, κ the exponential compaction rate and α_0 the initial distention. The parameter ε_c has to be calculated beforehand with following equation

$$\varepsilon_c = \frac{2 - 2\alpha_0 e^{\kappa(\varepsilon_x - \varepsilon_e)}}{\kappa \alpha_0 e^{\kappa(\varepsilon_x - \varepsilon_e)}} + \varepsilon_x. \tag{42}$$

In this model the distention is not calculated directly using the pressure. The change in the distention is calculated as follows

$$\dot{\alpha} = \frac{d\alpha}{d\varepsilon_V}\frac{d\varepsilon_V}{dt}. \tag{43}$$

2.5.3. Sirono-model

The Sirono porosity model implemented in the code is by Sirono (2004) which was later modified by Schäfer et al. (2007), and extended to different material by Geretshauser et al. (2010). It is an isothermal model, which is the main difference to the two above mentioned models. It uses a slightly modified Murnaghan EOS which is an extension of the liquid EOS with a non-linear

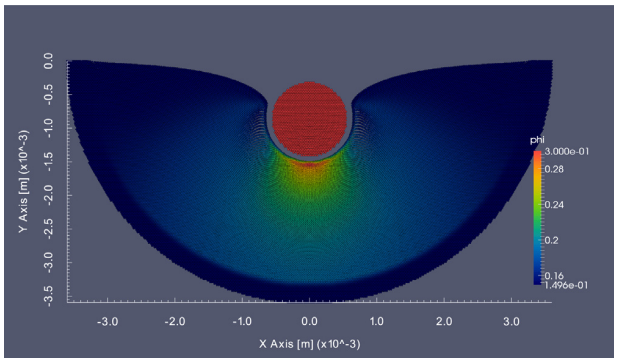


Fig. 3. Crater depth of the glass bead falling into the porous dust cake. The colour mapped is the filling factor $\phi = \varrho/\varrho_s$.

Table 3
Parameter table of the two dimensional crater depth validation simulations for the Sirono model.

Physical Quantity	Value with Unit
Glass bead	
Matrix density ϱ_0	2.54×10^3 kg/m^3
Bulk modulus K_0	5×10^9 Pa
Murnaghan exponent n	4
Radius r	0.55×10^{-3} m
Impact velocity v_0	0.65 ms^{-1}
Dust target	
Initial filling factor ϕ_0	0.15
Matrix density ϱ_s	2000 kg/m^3
Reference density ϱ_0'	300 kg/m^3
Bulk modulus K_0	3×10^5 Pa
Radius r_{dust}	3.3×10^{-3} m
Other parameters	
Artificial viscosity α, β	0.1, 0

dependency of the pressure on the density

$$p(\phi) = K(\phi_0')\left(\frac{\phi}{\phi_0'} - 1\right), \tag{44}$$

where $\phi_0' = \frac{\varrho_0'}{\varrho_s}$ and ϱ_0' is the reference density, i.e. the density at $p = 0$. To calculate the pressure at a given density we only need the bulk modulus $K(\phi)$ which is given by the power law

$$K(\phi) = 2\mu(\phi) = K_0\left(\frac{\phi}{\phi_i}\right)^{\gamma}, \tag{45}$$

with the shear modulus $\mu(\phi)$ and $\phi_i = \varrho_i/\varrho_s$. Since there is no energy dependency in this EOS it is only used for the low velocity regime. The pressure is restricted by the compressive strength $\Sigma(\phi)$ and tensile strength $T(\phi)$. Until the pressure reaches the upper limit (the compressive strength) the deformation is elastic and once it surpasses the compressive strength the deformation process will be plastic. This changes the slope for the calculation of the pressure Eq. (44) and thus deforming the material irreversibly. The pressure is limited in following way

$$p(\phi) = \begin{cases} \Sigma(\phi) & \text{if } \phi > \phi_c^+ \\ T(\phi) & \text{if } \phi < \phi_c^-. \end{cases} \tag{46}$$

The compressive strength is calculated using this equation

$$\Sigma(\phi) = \begin{cases} p_m\left(\frac{\phi_{\max}-\phi_{\min}}{\phi_{\max}-\phi_{\min}-\epsilon} - 1\right)^{\Delta \ln 10} & \text{if } \phi \le \phi_{\min} + \epsilon \\ p_m\left(\frac{\phi_{\max}-\phi_{\min}}{\phi_{\max}-\phi} - 1\right)^{\Delta \ln 10} & \text{if } \phi_{\min} + \epsilon < \phi < \phi_{\max} \\ \infty & \text{if } \phi \le \phi_{\max}, \end{cases} \tag{47}$$

with the material dependent parameters p_m, Δ, ϵ and the minimum and maximum filling factors $\phi_{\min}$ and $\phi_{\max}$. For the test calculation presented in Fig. 3, the following values have been calibrated using (Güttler et al., 2009) the mean pressure $p_m = 260$ Pa, $\Delta = 0.58$, $\epsilon = 0.005$, $\phi_{\min} = 0.12$ and $\phi_{\max} = 0.58$. The tensile strength is given by

$$T(\phi) = -10^{a+b\phi}\ \text{Pa}, \tag{48}$$

with $a = 2.8$ and $b = 1.48$. Fig. 3 shows one of these (in this case two dimensional) simulations where a glass bead falls into the porous dust cake (SiO_2). The used values for the setup are shown in Table 3.

2.6. Equation of states

The equation of state relates the thermodynamic state variables and closes the system of PDEs. It is normally defined as a relation between the density, pressure and temperature or internal energy of a specific material. Depending on the physical regimes, different EOS are applied. There is no universal EOS, which can be used for all materials under all conditions, liquid, gaseous or solid. Therefore, various EOS are implemented and available in the code. In the original version, the available EOS were liquid EOS, Murnaghan EOS, and Tillotson EOS. In the new version, we have added the EOS for polytropic gas, isothermal gas, ideal gas, and ANEOS. The use of porosity models require additional porosity-dependent pressure relations, which are implemented via special EOS for these porous materials. Hence, there are several additional versions for porous materials with Tillotson EOS for the p–α-, the Sirono-, and the ε–α-model. Additional EOS may be included to the code straightforwardly by adding another `EOS_TYPE_*` to `pressure.h` and the calculation of the pressure to the kernel function in `pressure.cu`.

2.6.1. Murnaghan EOS

The Murnaghan EOS can be derived from the assumption that the bulk modulus varies linearly with pressure (Murnaghan, 1937). It reads

$$p = \frac{K_0}{n}\left(\left(\frac{\varrho}{\varrho_0}\right)^n - 1\right), \tag{49}$$

where K_0 is the bulk modulus and ϱ_0 the density in the relaxed state. The Murnaghan EOS can be used to model weakly compressible materials. In this publication, it was applied to model the gravitational collapse of a granular column, see Section 3.4.

2.6.2. Tillotson EOS

A widely used non-linear EOS for simulating high-velocity impacts was introduced by Tillotson (1962) and can be applied over a wide range of physical conditions, while being computationally simple. It provides rudimentary distinction between the solid and vapour phase, but lacks a consistent treatment of phase changes. In the case of compressed regions ($\varrho \ge \varrho_0$) and u lower than the energy of incipient vaporization u_{iv} the equation of state reads

$$p = \left[a_T + \frac{b_T}{1 + u/(u_0\eta^2)}\right]\varrho u + A_T\xi + B_T\xi^2, \tag{50}$$

with $\eta = \varrho/\varrho_0$ and $\xi = \eta - 1$. In case of expanded material (u greater than the energy of complete vaporization u_{cv}) the

equation of state takes the form

$$p = a_T \varrho u + \left[\frac{b_T \varrho u}{1 + u/(u_0 \eta^2)} + A_T \xi \exp\left[-\beta_T \left(\eta^{-1} - 1\right)\right] \right] \tag{51}$$

$$\times \exp\left[-\alpha_T \left(\eta^{-1} - 1\right)^2\right]. \tag{52}$$

The symbols ϱ_0, A_T, B_T, u_0, a_T, b_T, α_T, and β_T are material dependent parameters. In the partial vaporization $u_{iv} < u < u_{cv}$, the pressure is linearly interpolated between the two values obtained via (50) and (51), respectively. The Tillotson EOS has the advantage of being computational very simple, and still sophisticated enough for the application over a wide regime of physical conditions in high pressure ranges, such as high-velocity collisions and impact cratering.

Note, that the Tillotson EOS can also yield negative pressure values for expanded ($\eta < 1$) and cold ($u < u_{iv}$) states. This corresponds to a tension in the solid material. As stated by Melosh (1996), a fluid will fragment into smaller droplets at lower densities, limiting the tension. In order to consider this behaviour, a density limit parameter can be defined for each material type in the simulation. As soon as the ratio η falls below this limit, the pressure is set to zero instead of becoming negative. Additionally, there is also a compile time option which prevents negative pressure values at all.

2.6.3. ANEOS

A more sophisticated treatment of phase changes is possible with the analytic equations of state (ANEOS) package (see, e.g., Melosh, 2007). ANEOS is based on calculating the Helmholtz free energy from first principles, where all relevant quantities can then be deduced from. This also holds for the temperature, which is not directly possible for the Tillotson EOS. However, ANEOS requires more than 40 input parameters (10 for the Tillotson EOS), and is not freely available. It consists of several thousand lines of FORTRAN code, which has to be utilized to produce lookup-tables, which can then be interpolated from. To enhance the thermodynamical aspects of our SPH code's material model, we have recently finished implementing ANEOS to overcome the limitations of the Tillotson EOS (Malamud et al., 2020).

2.7. Fixed and variable smoothing

The code supports three different methods for SPH smoothing: Either a fixed smoothing length for each material type is set or the smoothing length varies for each particle separately. In the latter case, a fixed number of interaction partners can be set individually for each material type or the smoothing length is integrated according to the following evolution equation

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{h}{d} \nabla \cdot \mathbf{v}, \tag{53}$$

where d denotes the dimension. Additionally, an upper and lower limiter can be chosen for each material type. If two particles with differing smoothing lengths h_a and h_b interact, the kernel is calculated with their mean $\frac{1}{2}(h_a + h_b)$. The search for the interaction partners is conducted by the help of the Barnes–Hut tree. For each particle, a tree walk starting at the root of the tree is performed. The search continuous descending only into nodes that intersect with the sphere with the radius of the particle's smoothing length. Finally, individual particles in the lowermost nodes (the leaves) are checked for interaction by calculating the distance and comparing to the smoothing length. Since the interaction partner search for one particle is independent from the search for the partners of another particle, the search algorithm parallelizes perfectly.

2.8. Time integrator

The default time integrator is an embedded second order Runge–Kutta integrator, which uses an additional third step to determine an adaptive time step. Its implementation is based on the description in Butcher (1987). Four more integrators are implemented: a simple Euler integrator for test purposes during new developments, two different predictor–corrector schemes, one with a prediction step of half the time step and a correction step with the whole time step and vice versa, and a special purpose coupled Heun and fourth order Runge–Kutta integrator. The time step size for all integrators is limited by the Courant condition and the artificial viscosity is also taken into account (Monaghan, 2012)

$$\Delta t \leq \mathrm{CFL} \frac{h}{c_s + 1.2(c_s \alpha + \mu_{\max} \beta)}. \tag{54}$$

Here, CFL is the Courant number, c_s denotes the sound speed, h is the smoothing length, α and β are the artificial viscosity parameters and $\mu_{\max}$ is the maximum value of μ_{ab} from all interaction pairs. The term μ_{ab} is an approximation for the velocity divergence and calculated according to

$$\mu_{ab} = \frac{\bar{h}_{ab} (\mathbf{v_a} - \mathbf{v_b}) \cdot (\mathbf{x_a} - \mathbf{x_b})}{(\mathbf{x_a} - \mathbf{x_b})^2 + \varepsilon_v \bar{h}_{ab}^2}. \tag{55}$$

The term $\bar{h}_{ab}^2$ is the average of the smoothing lengths of particle a and b. and the factor $\varepsilon_v \bar{h}_{ab}^2$ in the denominator prevents divergence of μ_{ab}. Apart from the Courant time limit, there exist additional limit check functions to ensure convergence of the crush-curve for simulations including porosity, see Section 2.5, and upper limits for an absolute change of specific values like the damage variable in simulations including brittle fracture can be specified to ensure convergence, i.e.

$$\Delta t \leq \begin{cases} \varepsilon \frac{|\xi| + \xi_{\min}}{\left|\frac{\mathrm{d}\xi}{\mathrm{d}t}\right|} & \text{for } \left|\frac{\mathrm{d}\xi}{\mathrm{d}t}\right| > 0 \\ \Delta t_{\max} & \text{else,} \end{cases} \tag{56}$$

for all quantities ξ which are integrated, where $\varepsilon < 1$ and $\xi_{\min}$ is a measure to set the minimum time step.

In addition, a combined Heun and fourth order Runge–Kutta (RK4) integrator is implemented with intended application for circumbinary discs following the method by Thun and Kley (2018). The Heun part is used to solve the hydrodynamical equations, i.e. to integrate the SPH equations and the RK4 part is used to integrate the orbits of the gravitating objects. For each complete timestep in the Heun integrator, the four steps of the RK4 are performed. In every substep of the RK4 the accelerations for all gravitational point masses are calculated. At the end of timestep, both integrators are at the new time $t + \Delta t$ and the cycle restarts.

2.9. Additional gravitating bodies

An arbitrary number of purely gravitationally interacting (point-)masses may be included in the simulation. The gravitational force from these objects leads to an additional acceleration of the particles. These masses also interact mutually and their motions are integrated with the same integrator that integrates the fluid or solid body. This renders simulations possible that include more than one central object, e.g., collisions of smaller objects in a planetary system or the simulation of accretion discs around circumbinaries with one or more potentially embedded planets. The point masses are stored in additional data files and a minimum and maximum distance for the particle can be specified individually for each point mass. Once a particle comes too close or too far, it is deactivated in the simulation.

2.10. Long-range forces

Currently, `miluphcuda` allows to simulate the self-gravity of the mass distribution given by the particle distribution. The long-range forces due to the mutually gravitational interaction of the particles is either calculated directly via a N^2 algorithm or approximately using the Barnes–Hut tree (Barnes and Hut, 1986) that is also used for the interaction partner search. The latter one may be also a starting point to implement other long-range forces like Coulomb's law. The additional acceleration due to the gravitational interaction is added to the momentum equation of the applied module, i.e. one of Eqs. (1),(5),(15).

The self-gravity module has been validated by the simulation of an isothermal, spherical and initially uniformly rotating collapsing molecular cloud by Schäfer et al. (2016). The module is used in the three applications presented in Sections 3.1 to 3.3.

3. Applications

In this section we intent to display the versatility of `miluphcuda`. We have adapted the code to various differing physical problems including hydro- and solid mechanics, with and without gravitational forces, self-gravity, strength-dominated and strengthless bodies, inviscid and viscous flows. The subsections contain exemplary simulations of various projects which make use of our code. In practice, each numerical investigation with SPH has to include a convergence study, i.e. varying number of particle numbers, varying smoothing lengths, and different values for other parameters of the method such as the artificial viscosity. Since our intention is to present many different simulations with all available modules, we do not present these convergence studies here, and focus only on variety of applications.

3.1. Embedding realistic collision outcomes into long-term planet formation simulations

The formation of terrestrial planets is believed to proceed in several relatively distinct steps, where during the last and final phase, often termed 'late-stage accretion', planetary embryos (∼ Moon- to Mars-sized objects) and remaining smaller planetesimals eventually accrete into planets. This phase is characterized by chaotic interaction and giant collisions among large, similar-sized, gravity-dominated bodies, which shape many of a planet's final characteristics — its orbital parameters, spin and obliquity, and not least its composition, especially also volatile/water contents. Since (giant) collisions are naturally the core agent of accretion, it is crucial to study those events individually in detail. However, to obtain truly self-consistent results especially in the context of a chaotic planet formation environment it is eventually the combination of the long-term dynamical evolution of protoplanets/planetesimals with realistic outcomes of individual encounters that is necessary. We follow two different approaches for this problem, both centred around physically accurate collision simulations with `miluphcuda`. The first one is to use a catalog of collision outcomes, spanning the most relevant parts of parameter space, where the outcome of a particular collision scenario is then interpolated from (Section 3.1.1). The second approach is to run a dedicated SPH simulation for each occurring collision event 'on the fly', whose results are directly re-inserted into the further underlying N-body evolution (Section 3.1.2).

Our main aim in these applications is to study volatile/water transport, which may have been delivered to the accretion zone of Earth by successive collisions among water-carrying planetesimals and planetary embryos from the outer asteroid belt region.

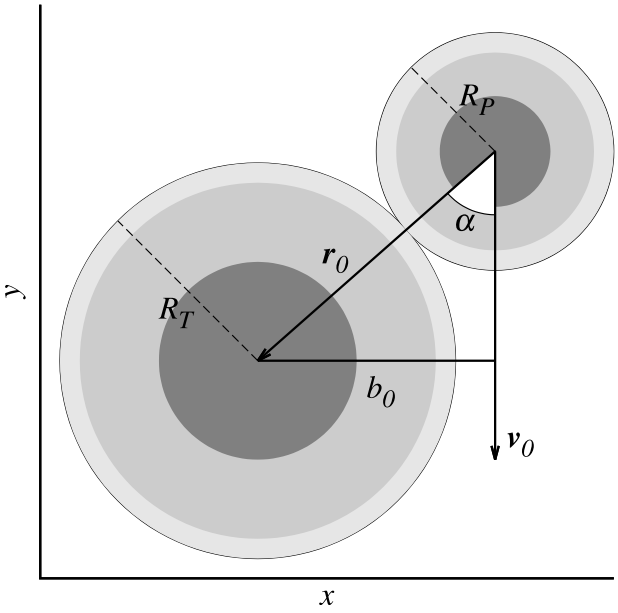


Fig. 4. Collision geometry for similar-sized, differentiated bodies in a target-centric frame. $|\mathbf{v}_0|$ is the impact velocity, α the impact angle (0° for head-on).

Even though semi-analytical models (scaling laws) to approximately predict collision outcomes have been developed in recent years (Leinhardt and Stewart, 2012; Leinhardt et al., 2015; Marcus et al., 2010b), they are currently not able to reliably predict the more subtle consequences for water inventories (Burger et al., 2018), where existing studies have almost exclusively assumed oversimplified perfectly inelastic merging of water-bearing bodies. This is despite the fact that water and other volatile material are particularly susceptible to collisional fragmentation and loss processes, and thus masses as well as water contents of final terrestrial planets are considerably overestimated in most cases.

Giant collisions of similar-sized bodies (cf. Fig. 4) are complex and generally super-sonic events, where strong shocks dissipate large amounts of energy, which can lead to large-scale melting and vaporization, or even eject outer layers mechanically. In addition material initially under high pressure can be released from the deep interior when bodies are (partly) disrupted (cf. Fig. 6). Correct modelling of the thermodynamic response with a suitable equation of state is therefore crucial (Section 2.6), while the precise material rheology becomes typically less important once in the gravity-dominated regime. The majority of such encounters is more or less oblique, which makes them naturally fully 3-dimensional events, and gives rise to frequent hit-and-run outcomes (e.g., Fig. 6), where the colliding bodies separate again (gravitationally) after the interaction phase.

3.1.1. Collisions catalog for protoplanetary bodies

In order to arrive at a more realistic estimate of the amount of water transported to Earth's accretion zone, we systematically investigated the rate of water transport in collisions by performing a suite of several hundred collision simulations with `miluphcuda`. Here, we present results in terms of the fraction of water retained after a collision for different parameters of the system including collision velocity, impact angle, and the masses and water-mass fractions of the projectile and target. The resulting catalog of collisional water loss focuses on planetary embryos of masses in the Ceres to Earth range and initial water content. Table 4 lists the parameters chosen for the catalog.

Table 4
Scenario parameters of the collision catalog. The velocity $v_{\rm esc}$ denotes the two-body escape velocity.

Parameter	Values
Total mass $M_{\rm tot} = M_{\rm projectile} + M_{\rm target}$	2 $M_{\rm Ceres}$(1.88×10^{21} kg), 2 $M_{☾}$ (1.47×10^{23} kg), 2 $M_{♂}$(1.28×10^{24} kg), 2 $M_{⊕}$(1.19×10^{25} kg)
Mass ratio $\gamma = M_{\rm projectile}/M_{\rm target}$	0.1, 0.5, 1.0
Projectile water mass fraction $w_{\rm p}$	0.1, 0.2
Target water mass fraction $w_{\rm p}$	0.1, 0.2
Impact velocity v [$v_{\rm esc}$]	1.0, 1.5, 2.0, 3.0, 5.0
Impact angle α [deg]	0, 20, 40, 60

Table 5
Tillotson EOS Parameters adopted from Benz and Asphaug (1999) used in the simulations for the collision catalog. The bulk modulus is set equal to A_T and B_T, shear modulus and the von Mises yield stress are denoted by μ and Y_M, respectively.

Material	ϱ_0 [kg/m^3]	A_T [GPa]	B_T [GPa]	E_0 [MJ/kg]	E_{iv} [MJ/kg]	E_{cv} [MJ/kg]
Basalt	2700	26.7	26.7	487	4.72	18.2
Ice	917	9.47	9.47	10	0.773	3.04
Material	α_T	β_T	a_T	b_T	μ [GPa]	Y_M [GPa]
Basalt	0.5	1.5	5	5	22.7	3.5
Ice	0.3	0.1	10	5	2.8	1

Table 6
Column description of the collision outcome data in the supplement.

Column	Description
v	collision velocity [$v_{\rm esc}$]
alpha	collision angle [deg]
m	$M_{\rm tot} = M_{\rm projectile} + M_{\rm target}$ [kg]
gamma	mass ratio $\gamma = M_{\rm projectile}/M_{\rm target}$
wp	projectile water mass fraction
wt	target water mass fraction
mS1	mass of survivor 1 [kg]
mS2	mass of survivor 2 [kg]
wmfS1	survivor 1 water mass fraction
wmfS2	survivor 2 water mass fraction
wretained	$1 - \frac{\text{water lost to debris [kg]}}{\text{total initial water [kg]}}$

In order to limit the parameter space we assumed a uniform structure consisting of a basaltic silicate rock interior covered by a water ice mantle for all bodies. Our material model is based on the Tillotson equation of state (see Melosh, 1996) and uses the parameters given in Table 4. The Weibull parameters for the damage model are $m = 16$ and $k = 10^{61}\,\mathrm{m}^{-3}$ (Nakamura et al., 2007) for basalt and $m = 9.1$ and $k = 10^{46}\,\mathrm{m}^{-3}$ (Lange et al., 1984) for water ice, respectively (see Table 5).

The outcome in terms of the masses of the largest two surviving fragments and the water remaining on these survivors is given in the supplementary data to this article. The data is given in xlsx format with columns as described in Table 6. Visualizing our high-dimensional data is difficult. In order to properly present our catalog, we thus apply the embedding algorithm UMAP (McInnes et al., 2018a,b), which allows an efficient, non-linear down-projection of our six-dimensional input space. We use the default hyperparameters for the UMAP mapping, except for n_neighbours=15 and min_dist=0.99. Fig. 5 visualizes the embedded dataset. Each of the original six dimensions is contributing some structure to the embedding. There are 17 clusters which are – each for itself – homogeneous in impact angle and projectile/target water mass fraction and composed of different impact velocities and masses. Although the plots are still difficult to interpret, a comparison of the colour gradients indicate that the water retention of the two largest fragments is correlated with both the impact velocity and the impact angle: while the gradients of the velocity closely describe the intra-cluster gradients of the water retention, the impact angle does so with the global gradient across all clusters.

In the near future we plan to create a scheme that allows us to integrate data from an extended catalog into N-body calculations that will use predicted collision outcomes instead of perfect merging. First experiments with multi-dimensional interpolation work satisfactorily regarding the water loss, but fail in terms of predicting correct fragment sizes, positions, and velocity vectors after the impact. Therefore, we currently run a simulation campaign of thousands of collisions — enough to apply machine learning methods to predict detailed collision outcomes as they occur in planet formation simulations (Winter, Winkler, Maindl, and Schäfer, in prep.).

3.1.2. Direct N-body – SPH hybrid simulations

The direct combination of an underlying N-body computation (realized with the versatile REBOUND package, Rein and Liu, 2012) with dedicated individual collision simulations has the advantage of full self-consistency and allows to precisely model all collisions at hand, including more subtle processes like water loss and transfer between colliding bodies in hit-and-run encounters. Such a direct hybrid approach has been applied only rarely and with limited resolution (Genda et al., 2011, 2017), and never to follow water or other volatiles to our knowledge.

In order to ensure the flawless handover from the N-body computation to an SPH collision and back, several well-synchronized steps are necessary, implemented via a Python interface, which calls the necessary subprocesses (including `miluphcuda`) and handles communication between them. Once bodies are found to be on colliding trajectories, the N-body integration continues until they have approached to a well-defined starting distance for the SPH simulation — several times the sum of their radii to allow for the build-up of potential pre-collision tidal deformations. The colliding bodies are initialized already with self-consistent hydrostatic profiles following Burger et al. (2018), which works practically instantaneously and eliminates the need for lengthy numerical relaxation. To ensure a self-consistent (gravity) environment, the star and all other bodies in the N-body system (besides the two colliding ones) are additionally included as point masses for the whole duration of the SPH simulation. Once the collision / interaction phase is over (potential post-collision fragments have clearly separated), we first identify spatially connected fragments via an octree/friends-of-friends algorithm, followed by additional search for gravitationally bound fragments, which are then each considered a single post-collision body. Eventually their (barycenters') positions and velocities are re-inserted into the N-body integration.

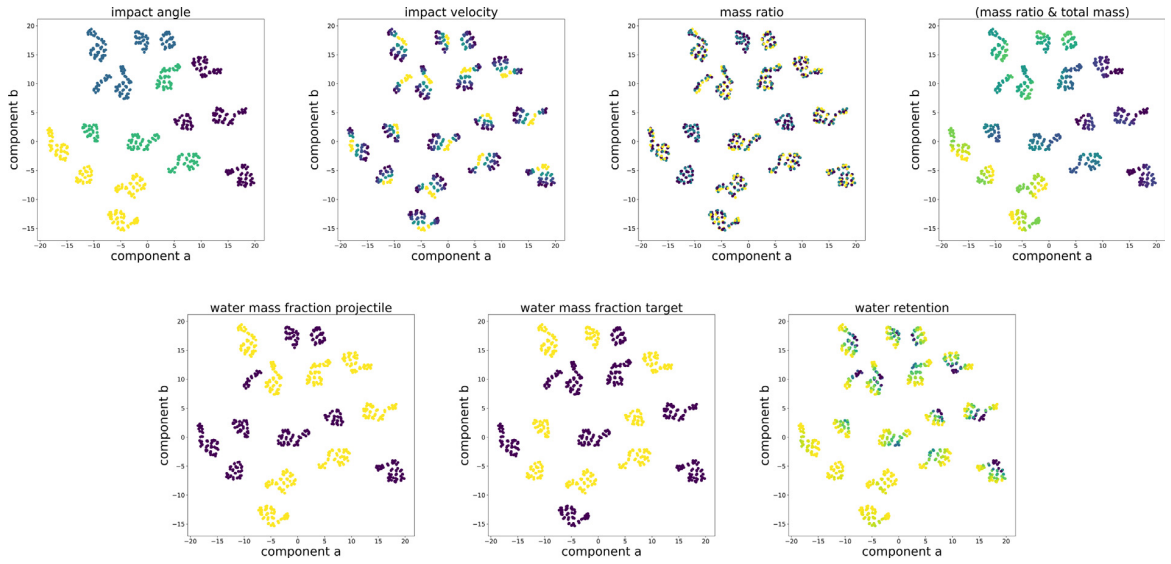


Fig. 5. For visualization, the six-dimensional input space was embedded into two dimensions *component a* and *component b* using UMAP. The titles of the plots indicate the respective colour codings. Note that only the water retention colour code represents a result after evaluating the collision outcomes, the others represent the initial conditions.

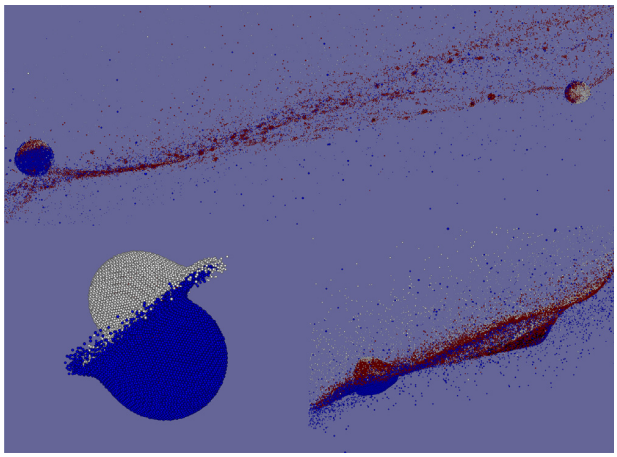


Fig. 6. Snapshots of a hit-and-run collision between protoplanet-sized bodies. The impactor (white) approaches from the left. Both objects are differentiated into an iron core (black), a silicate mantle (red) and a water shell, colour-coded in blue for the target and white for the impactor, to illustrate not only collisional losses but also water transfer between the bodies. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Source: Refer to Burger et al. (2018) for more details.

The vastly different time as well as size scales of individual collisions vs. long-term N-body evolution pose particular challenges. Individual collisions cover ∼hours to days at most, while the whole planetary accretion phase lasts for several 100 Myrs. The spatial domain of individual collisions is usually considerably below 10^6 km, while the whole system covers tens of au ($\sim 10^9$ km). The capabilities of `miluphcuda` allow us to run the necessary multi-material collision simulations (including surface water layers), where the various available strength/material models (Section 2.4) and equations of state (Section 2.6) can be utilized to account for the broad range of physical states of water in an active planet formation environment. The actual conditions at the onset of some currently simulated collision however, depend on many parameters, among them the distance to the star, the body's (especially thermal) history, and the time since the last (large) collision/impact event. For a truly holistic picture all these processes (and likely more) will have to be included eventually, which is our ultimate but not yet accomplished goal, therefore a preliminary solution is to model water inventories as (strengthless) fluid, properties that likely resemble a surface ocean on average. Our SPH code provides the necessary fast and efficient computations to run typically ∼hundreds of dedicated collision simulations per underlying N-body scenario, where individual ones (with resolutions between 25 000 to 75 000 SPH particles) require on the order of an hour to complete (with all pre- and post-processing steps).

In order to model collisional water transport we apply this hybrid framework to an evolving system of several hundred planetary embryos and planetesimals which extends into the water-rich region beyond the ice condensation line (snow line). We include either no (very disruptive), one (accretion or erosion), or two (hit-and-run) post-collision bodies in the further N-body evolution. The rationale is that collisions between large, gravity-dominated bodies generally result in at most two large post-collision bodies besides orders-of-magnitude smaller additional debris, which has either been desiccated directly during the collision, or remaining water is otherwise likely often lost quickly, once vaporized by large-scale energy deposition during impact. An example of such a collision is illustrated in Fig. 6.

3.2. Formation of massive exomoons of super-terrestrial exoplanets

Most simulation studies of giant impacts have focused on the collisional phase space conductive to the formation of Solar system planets and satellites (Barr, 2016). Despite an extensive collision simulation literature, there have only been a few studies that investigated giant impacts relevant to exoplanets that are more massive than the Earth (Marcus et al., 2010a,b; Liu et al., 2015; Barr and Bruck Syal, 2017), and in particular studies that focus on the formation of exosolar satellites (Barr and Bruck Syal, 2017). We also use `miluphcuda` in order to simulate novel scenarios which enable the formation of massive exomoons. We investigate the mass, long-term tidal-stability, composition and origin of material of such exomoons.

Since super-terrestrial planet collisions are often highly energetic, the debris discs which they generate typically require sensitive treatment which goes beyond the capabilities of the more common, Tillotson EOS. We therefore implement the M-ANEOS EOS. Our M-ANEOS parameter input files are derived from Melosh (2007). We consider differentiated impactors and targets composed of 30% iron and 70% dunite by mass. The initial setup of each simulation is calculated via a pre-processing step, in which both impactor and target are generated with relaxed internal structures, i.e. having hydrostatic density profiles and internal energy values from adiabatic compression, following the algorithm provided in appendix A of Burger et al. (2018). All our simulations have a resolution of 10^6 SPH particles.

An example of a particularly interesting simulation is shown in Fig. 7. While most collisions generate a massive disc, which later evolves to coagulate into a massive satellite, we also find this rare and illustrative case in which an exomoon forms through a graze & capture scenario between a super-Earth and an Earth-sized planet. The result is an intact, planet-sized exomoon, containing about half the mass of the Earth. At this size, an exomoon can be potentially detectable using our best available observing instruments. E.g., see the Hunt for Exomoons with Kepler (HEK) (Kipping et al., 2012) initiative. We note that to date, there has not been any confirmed detection of an exomoon, hence this finding may be important.

We validate our model by successfully reproducing the results obtained in a predecessor study (Barr and Bruck Syal, 2017) which uses a different, adaptive mesh refinement (AMR) code. Indeed our results are broadly compatible with theirs, and the data follows similar trends.

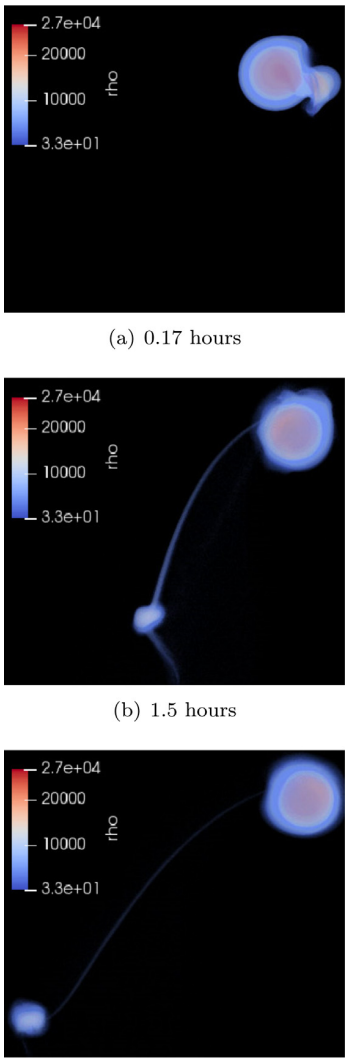
3.3. Ring structures around Chariklo and Haumea

All the four giant planets of our solar system host ring structures in equatorial orbits within their Roche radius, and recently at least two other bodies are found to host ring structures: the Centaur 10199 Chariklo (Braga-Ribas et al., 2014) and the dwarf planet Haumea (Ortiz et al., 2017). The characteristics of the rings vary widely, but in many systems it is possible to find dense and narrow rings mainly composed by particles with sizes ranging from centimetres to a tens of meter bodies. In such case the particles' interaction cannot be neglected, what poses a computational challenge for usual N-body approach usually applied to study the particles dynamics.

Furthermore, dense rings are optically thick and highly collisional systems. The typical impacts might happen at multiple times the metal escape velocity, so the collisions outcome cannot be safely determined from the extrapolation of the usual prescriptions (e.g., Leinhardt and Stewart, 2012).

Some of the issues mentioned above can be solved using SPH simulations. The parallel nature of the technique allows an efficient way to compute the mutual interaction among a large ensemble of particles. Besides being possible to take into account the material properties, SPH simulations also allow the inclusion of fragmentation and porosity effects, so the collisions are treated in a more realistic way.

Our code has been successfully adapted to simulate the system of Chariklo aiming to explain the formation the region depleted of material between the two rings, and to investigate the interaction of the ring particles with a putative satellite that may inhabit this gap. We employed the Tillotson equation of state, and the initial setup consisted in a small section of 10 × 5 km of the rings filled with 20 m bodies, each one being composed by particles spaced by 1 m. The number of bodies were determined to result in an optical depth comparable to the observed value. The self-gravity was taken into account, and all bodies were assumed to be composed by ice particles subject to the gravitational force of Chariklo and the satellite, both treated as external forces. The motion of the particles was followed in a rotating frame that moves with the satellite and periodic boundary conditions were applied in the azimuthal direction.



(a) 0.17 hours

(b) 1.5 hours

(c) 3 hours

Fig. 7. The first three hours of a giant impact between a 0.83 $M_\oplus$ impactor and 6.4 $M_\oplus$ target, obtaining a graze & capture scenario, which generates an intact, 0.5 $M_\oplus$ exomoon. The colour scheme shows the density in units of kg × m^{-3}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To validate the results obtained with `miluphcuda` we compared the simulation outcome with a similar setup integrated using the IAS15 algorithm, a high-order numerical integrator implemented in REBOUND (Rein and Spiegel, 2015). Preliminary results show that both runs are in agreement and the SPH code performed well, even more when we consider that N-body run required some simplification in the effects that can be added to the simulation.

3.4. Collapse of a granular column

To show the applicability of the code to model granular medium, we have simulated the three dimensional collapse of a

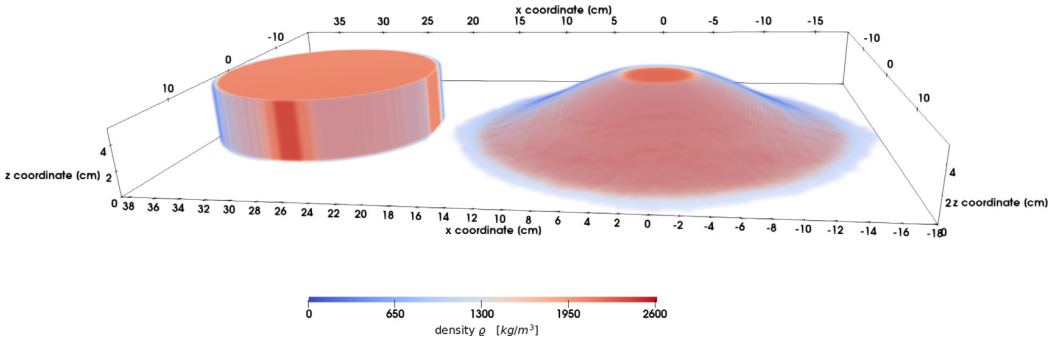


Fig. 8. Gravitational collapse of a granular column. The SPH density is shown in the volume rendered image for the initial state on the left hand side and for the final state on the right hand side. The initial ratio between radius and height was $a = 0.55$.

granular column under gravitation following a validation simulation from Szewc (2016), who compared simulations to experimental results by Lube et al. (2004). Initially, a granular column of radius R_0 and height $H_0 = aR_0$ (with $a = 0.55$ in the simulation presented here) is placed on the ground. The granular material has a density of $\varrho = 2.6\,\mathrm{g/cm^3}$, a sound speed of 100 m/s, an internal angle of friction of 30 deg and zero cohesion. Depending on the ratio a between initial height and radius, Szewc (2016) and Lube et al. (2004) find fitting values for the final runout R_∞ of the medium on the ground. The runout length determined by Szewc (2016) is

$$R_\infty = \begin{cases} 0.72aR_0 + R_0 & a < 1.7 \\ 1.02a^{3/5}R_0 + R_0 & a \geq 1.7 \end{cases}, \tag{57}$$

while the experimental data indicate (Lube et al., 2004)

$$R_\infty = \begin{cases} 1.24aR_0 + R_0 & a < 1.7 \\ 1.6a^{1/2}R_0 + R_0 & a \geq 1.7. \end{cases} \tag{58}$$

The rheology is modelled using the Mohr–Coulomb yield strength. In order to obtain a weakly compressible material, we apply the Murnaghan EOS with $n = 7$, use a bulk modulus of $K = 10^8$ Pa and a shear modulus of $\mu = 10^7$ Pa. The number of particles in the simulation is about 470 000. We find a final runout length of $R_\infty = 17.94$ cm, which is larger than the experimental value of 16.32 cm, found by Lube et al. (2004), a deviation of about 6%. For the same ratio a and initial radius R_0, the simulations of Szewc (2016) yield a runout length of 13.54 cm, a lower value than the experimental outcome. Fig. 8 shows renderings of the initial granular column and the material at the end.

3.5. High-speed dynamics

The code was used to simulate several high-velocity impacts in the context of asteroid target fragmentation by Buruchenko and Schäfer (2017). Moreover, its performance was compared to commercial software. In this section we present exemplary simulations showing the capability of `miluphcuda` to handle high-speed dynamics and shock-wave physics related problems. Piekutowski (2001) studied the experimental outcome of hypervelocity impacts of non-spherical projectiles on zinc plates by looking at specific features in the produced debris clouds. He documented changes in the debris clouds by shooting a zinc sphere, rod, and disc into 0.965 mm-thick zinc plates with a speed of about $5\,\mathrm{km\,s^{-1}}$. The three different shapes of the projectile yield highly distinct debris clouds: The debris cloud produced by the spherical projectile features a hourglass shape, the rod forms a cone-like structure of plate material attached to the remaining intact material of the impacting rod, the disc projectile generates a pillar-shaped debris cloud with only little dispersion. We have simulated the impact with 212 000, 795 000, and 175 000 particles respectively, using the Tillotson EOS with the values as specified in Table 7 in two dimensions, and a constant smoothing length of 2.5 × the initial particle separation. The parameters of the three impact experiments are as follows: the diameter of the sphere is 5.76 mm and the impact speed is $4.98\,\mathrm{km\,s^{-1}}$, the dimensions of the rod are 3.988 mm diameter and 14.148 mm length, and of the disc 13 mm are diameter and 0.795 mm thickness, with impact speeds of $4.97\,\mathrm{km\,s^{-1}}$ and $5\,\mathrm{km\,s^{-1}}$, respectively. The radiographs of the debris clouds produced in the experiment and the result of the SPH simulation are shown in Fig. 9. The simulation is capable to reproduce the resulting debris clouds accurately in shape and size.

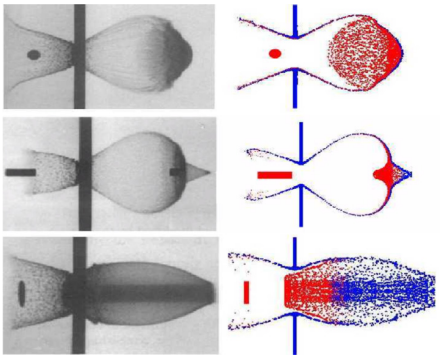


Fig. 9. Debris clouds produced in high-velocity impacts of three differently shaped projectiles into zinc plates. The left panels show the radiographs as published by Piekutowski (2001) and the right panels show the two dimensional simulation outcome using the SPH code. The projectile was (from top to bottom) a sphere, rod, disc.

Table 7
Parameters for the Tillotson EOS which were applied for the high-velocity impact simulations. The shear modulus μ and the von Mises yield stress Y_M are also given.

ϱ_0 [kg/m^3]	A_T [GPa]	B_T [GPa]	E_0 [MJ/kg]	E_{iv} [MJ/kg]	E_{cv} [MJ/kg]
7140	145	115	315	1.138	1.138
α_T	β_T	a_T	b_T	μ [GPa]	Y_M [GPa]
0.5	1.5	10	10	39	0.34

3.6. Shocktube test problem

One standard test for a hydro code is the one dimensional shocktube test problem (see, e.g., Monaghan and Gingold, 1983). The initial values at $t = 0$ are given by

$$\mathbf{U}(x, t = 0) = \begin{cases} \mathbf{U}_l & x \leq x_0 \\ \mathbf{U}_r & x > x_0, \end{cases} \tag{59}$$

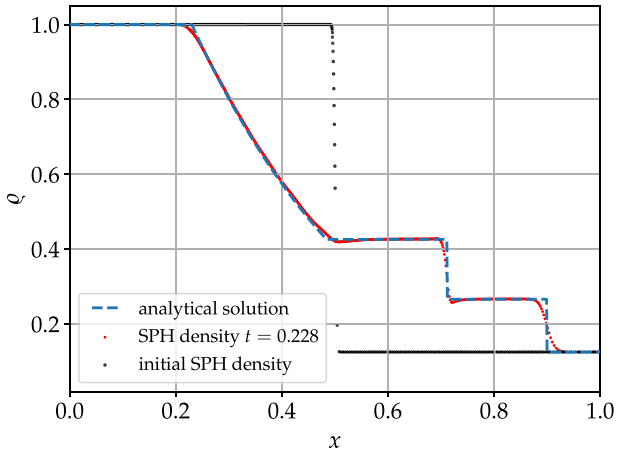


Fig. 10. Shocktube problem. The initial density distribution is indicated by the black dots. The analytical solution at time $t = 0.228$ is given by the dashed blue curve. The density after this time obtained with the SPH simulation is shown by the red dots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

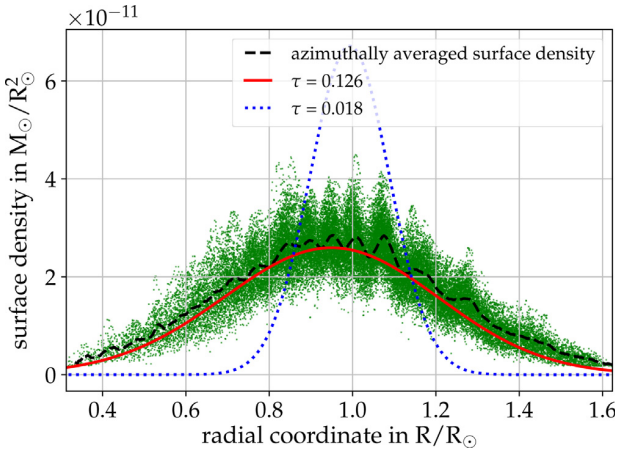


Fig. 11. Radial surface density profile of the viscous dust ring after the viscous time $\tau = 0.126$. The black dashed line denotes the azimuthally averaged surface density, the green dots the density of the particles, and the solid red line the analytical solution given by Eq. (62).

where $\mathbf{U}_{l,r} = (\varrho_{l,r}, v_{l,r}, u_{l,r})$ denotes the constant initial values for density, velocity and internal energy left and right of x_0. The pressure is given by the ideal gas equation

$$p = (\gamma - 1)\varrho u, \tag{60}$$

with $\gamma = 7/5$ in our setup. We place particles in the interval $[-1, 2]$ with $x_0 = 0.5$, and set $\mathbf{U}_l(x, t = 0) = (1, 0, 2.5)$ and $\mathbf{U}_r = (0.125, 0, 2)$. The particles have equal masses with a separation of 4×10^{-3} in the lower density region $[-1, 0.5]$ and a ϱ_r/ϱ_l narrower separation in the higher density region $]0.5, 2]$, leading to a total number of 3376 particles. The smoothing length is 1×10^{-2}. We apply the standard Monaghan artificial viscosity with $\alpha = 1, \beta = 2$. The initial density at $t = 0$, the density at $t = 0.0228$ and the analytical solution is shown in Fig. 10. We find good agreement with the analytical solution.

3.7. Viscously spreading ring

In order to test the Navier–Stokes implementation, we simulate the two dimensional viscously spreading ring with the parameters given in Speith and Kley (2003). The viscously spreading ring is pressure-less and orbits a central point mass on a Keplerian orbit. With the approximation of a small kinematic viscosity coefficient ν that is independent of the surface density, an analytic solution was derived by Lynden-Bell and Pringle (1974). With the initial radial profile of a ring with total mass M at $t = 0$

$$\Sigma(R) = \frac{M}{2\pi R_0}\delta(R - R_0), \tag{61}$$

the surface density evolves according to

$$\Sigma(\tau, x) = \frac{M}{\pi R_0^2}\frac{1}{\tau x^{1/4}} I_{\frac{1}{4}}\left(\frac{2x}{\tau}\right)\exp\left(-\frac{1+x^2}{\tau}\right), \tag{62}$$

with $x = R/R_0$, $\tau = 12\nu t/R_0^2$, and $I_{\frac{1}{4}}$ is the modified Bessel function to the order of 1/4. Following Speith and Kley (2003), we simulate the viscously spreading ring with $N = 40\,000$ particles. The mass of the ring is $10^{-10}\,\mathrm{M}_\odot$, the central mass is $M_c = 1\,\mathrm{M}_\odot$, and the initial radius is $R_0 = 1\,\mathrm{R}_\odot$. The kinematic viscosity coefficient is $\nu = 3 \times 10^{-8}\,\mathrm{R}_\odot^2/\mathrm{s}$. The (constant) smoothing length is set to $h = 0.0501\,\mathrm{R}_\odot$. The initial particle distribution represents the surface density given by Eq. (62) at the viscous time $\tau = 0.018$. The initial velocity for each particle is given by the Keplerian velocity around the central mass

$$v_\varphi = \sqrt{\frac{GM_c}{R}}. \tag{63}$$

Fig. 11 shows the radial surface density profile of the ring after the viscous time $\tau = 0.126$. The simulation matches the analytical solution amply while slightly overestimating the surface density at the outer rim of the ring. We find also the spiral instability as described and analysed by Speith and Kley (2003). The variation of the azimuthally averaged surface density (black dashed line) around the analytical (1D-) solution emerges from this (2D-) instability.

3.8. Circumbinary disc evolution

An arbitrary number of (point-) mass objects can be added to the simulation as described in Section 2.9. We have simulated the evolution of a circumbinary disc in a two dimensional simulation. A circumbinary disc is an accretion disc orbiting a binary system that consists for example of a binary star. The numerical simulation of circumbinary discs may be quite arduous depending on the dynamical parameters of the binary system, e.g., the eccentricity (Thun et al., 2017). The gravitational force acting from the binary stars on the gas may lead to spiral shock waves in the disc and to the formation of a inner gap around the two stars eventually. The code was applied to model the accretion disc around HD104237 using the parameters as described in Dunhill et al. (2015). The two binaries have masses of $2.2\,\mathrm{M}_\odot$ and $1.4\,\mathrm{M}_\odot$, a semi-major axis of $a = 0.22$ au and an eccentricity of $e = 0.6$. The gas density shows the spiral shock waves exerted by violent gravitational interaction with the binary stars. The detailed analysis of our simulations of circumbinary discs are beyond of the scope of this paper and will be published elsewhere (Audiffren et al. in prep.) (see Fig. 12).

4. Conclusion

The numerical particle method SPH has proven to be an appropriate scheme when dealing with PDEs in the context of hydrodynamics and continuum mechanics. The GPU SPH code `miluphcuda` is now publicly available with this new release. We have presented several applications of the code from a wide range of fields, including inviscid and viscous flows, strengthless bodies

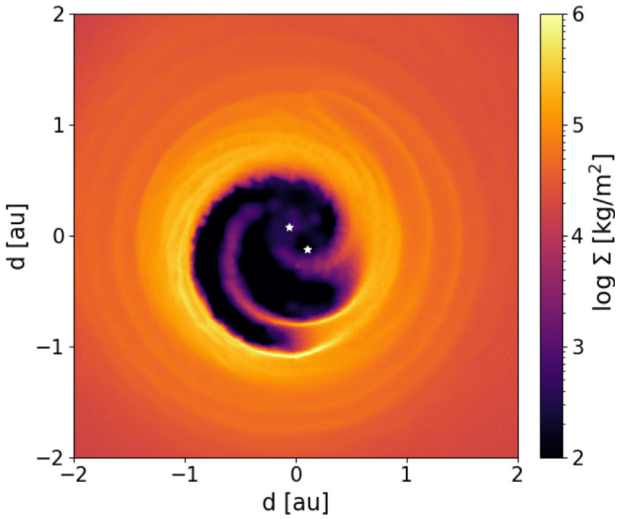


Fig. 12. The surface density distribution in the circumbinary disc around HD104237 after a simulation time of 159 binary orbits on a logarithmic scale. The two white stars indicate the current position of the primary secondary.

and solid bodies, and self-gravitating objects. The main focus of the code lies in collisional and impact studies with different materials and high and low velocities, although it has been extended to handle liquid and gaseous flows from one to three dimensions, e.g., to study circumbinary discs. Another promising application has been the combination of the SPH code with the REBOUND software package to study the long term evolution of water transfer in late stage accretion. The code runs on single Nvidia GPU, which can be plugged in normal workstations running Linux or macOS with ease, and allows for speed-ups compared to a single core CPU code (Schäfer et al., 2016) depending on the specific problem up to 40 in pure SPH simulations and up to 80 in simulations including self-gravity. Since the new modules have only been implemented in the GPU version of our SPH code, we cannot provide speed-up values for simulations including porosity. Compared to non-porous simulations, simulations including porosity demand for much lower time step sizes to obtain convergence of the crush-curve. Since the computational effort per time step remains unchanged overall, we expect comparable speed-ups for porous simulations.

CRediT authorship contribution statement

C.M. Schäfer: Project administration, Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing. **O.J. Wandel:** Methodology, Validation, Writing - original draft. **C. Burger:** Methodology, Validation, Writing - original draft. **T.I. Maindl:** Methodology, Validation, Writing - original draft. **U. Malamud:** Writing - original draft. **S.K. Buruchenko:** Writing - original draft. **R. Sfair:** Writing - original draft. **H. Audiffren:** Writing - original draft. **E. Vavilina:** Writing - review & editing. **P.M. Winter:** Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Several plots in this publication have been made by the use of the matplotlib package by Hunter (2007) or with ParaView (Ayachit, 2015). We thank the anonymous reviewers whose comments and suggestions helped to improve and clarify this manuscript. CB and TIM appreciates support by the FWF Austrian Science Fund project S 11603-N16. CMS and CB appreciate support by the DFG German Science Foundation project 398488521. CMS wants to thank Daniel Thun, Martin Jutzi and Willy Kley for helpful discussions during the course of this project.

The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, Germany, the state of Baden-Württemberg through bwHPC, Germany and the German Research Foundation (DFG) through grant no. INST 37/935-1 FUGG.

Appendix A. The code basics and speedup

The code is written in the C programming language and uses the Nvidia CUDA application programming interface (CUDA version $\geq$ 6.0). The different physical modules and the dimension can be set by compile time switches. The basic tree code is based on the implementation by Burtscher (2011). A description of the source and configuration files is shown in Table A.8. The material configuration files are parsed using `libconfig` (Lindner, 2015). The preferred data format for the data files is HDF5. However, if the library is missing, the simulations can also run with simple ASCII table files, one SPH particle per line. A skeleton `Makefile` is included in the repository and has to be modified according to the local settings and GPU architecture. The code runs on Linux and macOS with a CUDA-capable GPU. The number of threads for each CUDA kernel can be set as a compile time option in `timeintegration.h`. Note, that the best settings depend on your specific hardware and physical model. There is no distinct recommendation we can provide. Usually, a reduction of the computational time can be obtained by changing slightly the number of threads for the kernel with the maximal computational time. It is advisable to tweak the number of threads for the kernels before starting production runs. All compile time options related to the physical and numerical model are set in `parameter.h`. More options like the choice of the integrator, simulation time and number of output files, are parsed on the command line on the execution. All available command line options are explained via the help option `miluphcuda -h`.

If you intend to implement additional physics to the code, the following steps are required: Add new variables to the `Particle` structure in `miluph.h` and include the corresponding memory allocation in `memory_handling.cu`. For each substep of the integrator, the central function `rhs()` is called, which subsequently calls the CUDA kernels to determine all derivatives that are required by the integrator. A skeleton CUDA kernel running over all particles in the simulation is shown in listing 1.

```
1  __global__ void skeleton_function(int *interactions)
2  {
3      int i, inc;
4      inc = blockDim.x * gridDim.x;
5      /* main loop over all sph particles */
6      for (i = threadIdx.x + blockIdx.x * blockDim.x;
       i < numParticles; i += inc) {
7
8          /* loop over all interaction partners of
       particle i */
9          for (j = 0; j < p.noi[i]; j++) {
10             /* index of interacting particle is
       stored in array interactions /*
```

Table A.8
List of files in `miluphcuda`.

Filename	Purpose
anEOS.cu, anEOS.h	functions for initializing and running ANEOS
artificial_stress.cu, artificial_stress.h	functions for artificial stress
boundary.cu, boundary.h	special treatment of boundaries (external forces, etc.)
checks.h	consistency checks of compile time options
coupled_heun_rk4_sph_nbody.cu, coupled_heun_rk4_sph_nbody.h	coupled 2nd order Heun with 4th order RK integrator
cuda_utils.h	CUDA specific functions
damage.cu, damage.h	functions for the fragmentation and damage models
density.cu, density.h	calculate density
device_tools.cu, device_tools.h	functions to print GPU information
euler.cu, euler.h	simple Euler integrator for test purposes
gravity.cu, gravity.h	functions for the calculation of self gravity and gravitational interaction of SPH particles with point masses
internal_forces.cu, internal_forces.h	compute the change of stress tensor (regolith) and/or deviatoric stress tensor (all other solid materials), time evolution of density
io.cu, io.h	functions for input/output, ASCII and HDF5
kernel.cu, kernel.h	calculate SPH kernel values
linalg.cu, linalg.h	helping functions for matrix operations, etc.
little_helpers.cu, little_helpers.h	
Makefile	Makefile for GNU Make. Needs customization!
material.cfg	material configuration and parameters, see `material_data/` subdirectory.
memory_handling.cu, memory_handling.h	functions for memory allocation and copying between host and device
miluph.cu	main
miluph.h	header file with `Particle` structure definition
parameter.h	file includes all important compile time options, physical model and numerical settings
pc_values.dat	minimum absolute values, which are needed for the predictor–corrector integration schemes
plasticity.cu, plasticity.h	material models (rheology): Drucker–Prager, Collins, von Mises
porosity.cu, porosity.h	porosity models
predictor_corrector.cu, predictor_corrector.h	Predictor–Corrector integrator with predictor step dt/2
predictor_corrector_euler.cu, predictor_corrector_euler.h	Predictor–Corrector integrator with predictor step dt
pressure.cu, pressure.h	calculation of the pressure according to the chosen EOS; currently there is, for solids: Tillotson EOS, Murnaghan EOS, Regolith EOS (Drucker–Prager model), ANEOS; for porous solids: Jutzi EOS (Tillotson EOS with $p-\alpha$), Jutzi–Murnaghan EOS (Murnaghan EOS with $p-\alpha$); for gas: Ideal gas EOS, Polytropic gas EOS, Isothermal gas EOS
rhs.cu, rhs.h	right-hand sides, calls the kernels that compute the time derivatives
rk2adaptive.cu, rk2adaptive.h	RK2 integrator with adaptive time step
rk4_pointmass.cu, rk4_pointmass.h	RK4 integrator for gravitating pointmasses
soundspeed.cu, soundspeed.h	calculation of the soundspeed
stress.cu, stress.h	calculation of the stress tensor
timeintegration.cu, timeintegration.h	calculation of all internal forces of the solid or fluid; calculation of artificial viscosity and artificial stress
tree.cu, tree.h	functions for the Barnes–Hut tree, interaction search
velocity.cu, velocity.h	calculate velocities
viscosity.cu, viscosity.h	Navier–Stokes equation
xsph.cu, xsph.h	functions for the XSPH algorithm

```
11              ip = interactions[i *
    MAX_NUM_INTERACTIONS + j];
12              // now p.rho[i] contains the density of
    particle i and
13              // p.rho[ip] the density of its
    interaction partner...
14
15              // insert your sph calculation here
16
17          } // neighbour loop end
18
19
20      } // proceed with next particle
21  }
```

Listing 1: A skeleton CUDA kernel running over all particles and their interaction partners.

The main reason to port the code to CUDA is the expected speedup compared to the CPU-only implementation. Modern SPH codes like SWIFT are designed to run on HPC architectures like clusters with distributed memory and use the Message Passing Interface (MPI) for the process communication. However, access to HPC clusters might be limited and scarce for several reasons. Moreover, long ($\sim$ several weeks) running simulations are normally not possible due to queue regulations. Our code was especially designed to run on standard workstations with Nvidia GPUs to render high resolution SPH simulations possible without the access to special dedicated cluster hardware. Running on single GPUs only, the resolution is limited by the memory of the GPU, which is obviously much less than the distributed memory of a cluster. Hence, with current GPU hardware (12 GB memory), the maximal number of particles in a simulation with `miluphcuda` is 4×10^6 to 10^7 depending on the activated modules in the code.

The speedup for a three dimensional hydro simulation including self-gravity is presented in Fig. A.13. We have compared the runtime for four different tasks: (i) neighbour search, (ii) SPH equations, (iii) self gravity, (iv) total right hand side. The last task combines (i)-(iii) and includes also the time to build the tree and calculate pressure and sound speed for all particles. Task (ii) includes the SPH equations for momentum only, since the test case was isothermal. The highest speed up is achieved in the self-gravity module, followed by the speedup in the neighbour search. These results justify our motivation to port the code to CUDA, since most of the computational time is spent in the self-gravity task for simulations including self-gravity, and in the neighbour-searching task for simulations without self-gravity.

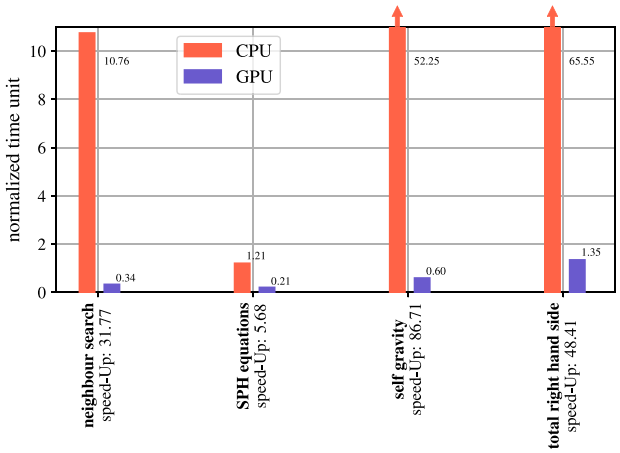


Fig. A.13. Speedup obtained with the CUDA port of the code compared to the CPU-only code (Schäfer et al., 2016). The CPU is an intel i7 clocked at 3.4 GHz. The GPU is a primitive (first release in 2013) standard gaming consumer graphics card Nvidia GTX Titan with a base clock of 837 MHz and 2688 CUDA cores. Shown are the normalized time units spent for different tasks.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at doi:10.1016/j.ascom.2020.100410.

References

Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M., 2012. Versatile rigid-fluid coupling for incompressible sph. ACM Trans. Graph. 31, 62:1–62:8. doi:10.1145/2185520.2185558.

Amicarelli, A., Kocak, B., Sibilla, S., Grabe, J., 2017. A 3d smoothed particle hydrodynamics model for erosional dam-break floods. Int. J. Comput. Fluid Dyn. 31, 413–434. doi:10.1080/10618562.2017.1422731.

Ayachit, U., 2015. The ParaView Guide: A Parallel Visualization Application. Kitware, Inc., USA.

Barnes, J., Hut, P., 1986. A hierarchical O(N log N) force-calculation algorithm. Nature 324, 446–449. doi:10.1038/324446a0.

Barr, A.C., 2016. Formation of exomoons: a solar system perspective. Astron. Rev. 12, 24–52. doi:10.1080/21672857.2017.1279469.

Barr, A.C., Bruck Syal, M., 2017. Formation of massive rocky exomoons by giant impact. Mon. Not. R. Astron. Soc. 466, 4868–4874. doi:10.1093/mnras/stx078.

Benz, W., 1990. Smooth particle hydrodynamics - a review. In: Buchler, J.R. (Ed.), Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects. p. 269.

Benz, W., Asphaug, E., 1994. Impact simulations with fracture. I. Method and tests. Icarus 107, 98–116. doi:10.1006/icar.1994.1009.

Benz, W., Asphaug, E., 1999. Catastrophic disruptions revisited. Icarus 142, 5–20. doi:10.1006/icar.1999.6204.

Braga-Ribas, F., Sicardy, B., Ortiz, J.L., Snodgrass, C., Roques, F., Vieira-Martins, R., Camargo, J.I.B., Assafin, M., Duffard, R., Jehin, E., Pollock, J., Leiva, R., Emilio, M., Machado, D.I., Colazo, C., Lellouch, E., Skottfelt, J., Gillon, M., Ligier, N., Maquet, L., Benedetti-Rossi, G., Gomes, A.R., Kervella, P., Monteiro, H., Sfair, R., Moutamid, M.E., Tancredi, G., Spagnotto, J., Maury, A., Morales, N., Gil-Hutton, R., Roland, S., Ceretta, A., Gu, S.H., Wang, X.B., Harpsøe, K., Rabus, M., Manfroid, J., Opitom, C., Vanzi, L., Mehret, L., Lorenzini, L., Schneiter, E.M., Melia, R., Lecacheux, J., Colas, F., Vachier, F., Widemann, T., Almenares, L., Sandness, R.G., Char, F., Perez, V., Lemos, P., Martinez, N., Jørgensen, U.G., Dominik, M., Roig, F., Reichart, D.E., Lacluyze, A.P., Haislip, J.B., Ivarsen, K.M., Moore, J.P., Frank, N.R., Lambas, D.G., 2014. A ring system detected around the Centaur (10199) Chariklo. Nature 508, 72–75. doi:10.1038/nature13155.

Bui, H.H., Fukagawa, R., Sako, K., Ohno, S., 2008. Lagrangian Meshfree particles method (sph) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model. Int. J. Numer. Anal. Methods Geomech. 32, 1537–1570. doi:10.1002/nag.688.

Burger, C., Maindl, T.I., Schäfer, C.M., 2018. Transfer, loss and physical processing of water in hit-and-run collisions of planetary embryos. Celestial Mech. Dynam. Astronom. 130, doi:10.1007/s10569-017-9795-3.

Burger, C., Schäfer, C.M., 2017. Applicability and limits of simple hydrodynamic scaling for collisions of water-rich bodies in different mass regimes. In: Proceedings of the First Greek-Austrian Workshop on Extrasolar Planetary Systems. pp. 63–81, arXiv:1704.06075.

Burtscher, M., 2011. GPU Computing Gems Emerald Edition.

Buruchenko, S.K., Schäfer, T.I., 2017. Smooth particle hydrodynamics gpu-acceleration tool for asteroid fragmentation simulation. In: 14th Hypervelocity Impact Symposium 2017, HVIS2017, 24-28 April 2017, Canterbury, Kent, UK. Procedia Eng. 204, 59–66. doi:10.1016/j.proeng.2017.09.726.

Butcher, J.C., 1987. The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods. Wiley-Interscience, New York, NY, USA.

Carroll, M., Holt, A.C., 1972. Suggested Modification of the $P-\alpha$ Model for Porous Materials.

Cercos-Pita, J., 2015. Aquagpusph, a new free 3d sph solver accelerated with opencl. Comput. Phys. Comm. 192, 295–312. doi:10.1016/j.cpc.2015.01.026.

Collins, G.S., Melosh, H.J., Ivanov, B.A., 2004. Modeling damage and deformation in impact simulations. Meteorit. Planet. Sci. 39, 217–231. doi:10.1111/j.1945-5100.2004.tb00337.x.

Collins, G., Melosh, J., Wünnemann, K., 2011. Improvements to the epsilon–alpha compaction model for simulating impacts into high-porosity solar system objects. Int. J. Impact Eng. - INT J IMPACT ENG 38, 434–439. doi:10.1016/j.ijimpeng.2010.10.013.

Crespo, A., Domínguez, J., Rogers, B., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A., García-Feal, O., 2015. Dualsphysics: Open-source parallel cfd solver based on smoothed particle hydrodynamics (sph). Comput. Phys. Comm. 187, 204–216. doi:10.1016/j.cpc.2014.10.004.

Dunhill, A.C., Cuadra, J., Dougados, C., 2015. Precession and accretion in circumbinary discs: the case of HD 104237. Mon. Not. R. Astron. Soc. 448, 3545–3554. doi:10.1093/mnras/stv284.

Flebbe, O., Muenzel, S., Herold, H., Riffert, H., Ruder, H., 1994. Smoothed particle hydrodynamics: Physical viscosity and the simulation of accretion disks. Astrophys. J. 431 (754), doi:10.1086/174526.

Genda, H., Iizuka, T., Sasaki, T., Ueno, Y., Ikoma, M., 2017. Ejection of iron-bearing giant-impact fragments and the dynamical and geochemical influence of the fragment re-accretion. Earth Planet. Sci. Lett. 470, 87–95. doi:10.1016/j.epsl.2017.04.035.

Genda, H., Kokubo, E., Ida, S., 2011. Giant impacts and terrestrial planet formation. In: Lunar and Planetary Science Conference, p. 2090.

Geretshauser, R.J., Speith, R., Güttler, M., Blum, J., 2010. Numerical simulations of highly porous dust aggregates in the low-velocity collision regime*. Astron. Astrophys. 513.

Gingold, R.A., Monaghan, J.J., 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. 181, 375–389. doi:10.1093/mnras/181.3.375.

Grishin, E., Malamud, U., Perets, H.B., Wand el, O., Schäfer, C.M., 2020. The wide-binary origin of (2014) MU_{69}-like Kuiper belt contact binaries. Nature 580, 463–466. doi:10.1038/s41586-020-2194-z.

Güttler, C., Krause, M., Geretshauser, R.J., Speith, R., Blum, J., 2009. The physics of protoplanetesimal dust agglomerates. IV. Toward a dynamical collision model. Astrophys. J. 701, 130–141. doi:10.1088/0004-637X/701/1/130.

Haghighipour, N., Maindl, T.I., Schäfer, C., Speith, R., Dvorak, R., 2016. Triggering sublimation-driven activity of main belt comets. Astrophys. J. 830, 22. doi:10.3847/0004-637X/830/1/22.

Haghighipour, N., Maindl, T.I., Schäfer, C.M., Wandel, O.J., 2018. Triggering the activation of main-belt comets: The effect of porosity. Astrophys. J. 855, 60. doi:10.3847/1538-4357/aaa7f3.

Hérault, A., Bilotta, G., Dalrymple, R.A., 2010. Sph on gpu with cuda. J. Hydraul. Res. 48, 74–79. doi:10.1080/00221686.2010.9641247.

Hernquist, L., Katz, N., 1989. TREESPH - A Unification of SPH with the hierarchical tree method. Astrophys. J. Suppl. 70, 419–446. doi:10.1086/191344.

Herrmann, W., 1969. Constitutive equation for the dynamic compaction of ductile porous materials. J. Appl. Phys. 40.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. Comput. Sci. Eng. 9, 90–95.

Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M., 2014. SPH Fluids in computer graphics. In: Lefebvre, S., Spagnuolo, M. (Eds.), Eurographics 2014 - State of the Art Reports. The Eurographics Association, doi:10.2312/egst.20141034.

Jutzi, M., 2015. Sph calculations of asteroid disruptions: The role of pressure dependent failure models. In: VIII Workshop on Catastrophic Disruption in the Solar System. Planet. Space Sci. 107, 3–9. doi:10.1016/j.pss.2014.09.012.

Jutzi, M., Michel, P., Hiraoka, K., Nakamura, A.M., Benz, W., 2009. Numerical simulations of impacts involving porous bodies: Ii. comparison with laboratory experiments. Icarus 201.

Kegerreis, J.A., Eke, V.R., Gonnet, P., Korycansky, D.G., Massey, R.J., Schaller, M., Teodoro, L.F.A., 2019. Planetary giant impacts: convergence of high-resolution simulations using efficient spherical initial conditions and SWIFT. Mon. Not. R. Astron. Soc. 487, 5029–5040. doi:10.1093/mnras/stz1606.

Kipping, D.M., Bakos, Buchhave, L., Nesvorný, D., Schmitt, A., 2012. The hunt for exomoons with kepler (hek). i. description of a new observational project. Astrophys. J. 750, 115. doi:10.1088/0004-637X/750/2/115.

Korzani, M.G., Galindo-Torres, S.A., Scheuermann, A., Williams, D.J., 2017. Parametric study on smoothed particle hydrodynamics for accurate determination of drag coefficient for a circular cylinder. Water Sci. Eng. 10, 143–153. doi:10.1016/j.wse.2017.06.001.

Lange, M.A., Ahrens, T.J., Boslough, M.B., 1984. Impact cratering and spall failure of gabbro. Icarus 58, 383–395. doi:10.1016/0019-1035(84)90084-8.

Leinhardt, Z.M., Dobinson, J., Carter, P.J., Lines, S., 2015. Numerically predicted indirect signatures of terrestrial planet formation. Astrophys. J. 806 (23), doi:10.1088/0004-637X/806/1/23.

Leinhardt, Z.M., Stewart, S.T., 2012. Collisions between gravity-dominated bodies. I. Outcome regimes and scaling laws. Astrophys. J. 745, 79. doi:10.1088/0004-637X/745/1/79.

Libersky, L.D., Petschek, A.G., 1991. Smooth particle hydrodynamics with strength of materials. In: Trease, H.E., Fritts, M.F., Crowley, W.P. (Eds.), Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method. pp. 248–257. doi:10.1007/3-540-54960-9_58.

Lindner, M., 2015. Libconfig@ONLINE. URL http://www.hyperrealm.com/libconfig.

Liu, S.F., Hori, Y., Lin, D.N.C., Asphaug, E., 2015. Giant impact: An efficient mechanism for the devolatilization of super-earths. Astrophys. J. 812, 164. doi:10.1088/0004-637X/812/2/164.

Liu, M.B., Liu, G.R., 2010. Smoothed particle hydrodynamics (sph): an overview and recent developments. Arch. Comput. Methods Eng. 17, 25–76. doi:10.1007/s11831-010-9040-7.

Lube, G., Huppert, H.E., Sparks, R.S.J., Hallworth, M.A., 2004. Axisymmetric collapses of granular columns. J. Fluid Mech. 508, 175–199. doi:10.1017/S0022112004009036.

Lucy, L.B., 1977. A numerical approach to the testing of the fission hypothesis. Astron. J. 82, 1013–1024. doi:10.1086/112164.

Lynden-Bell, D., Pringle, J.E., 1974. The evolution of viscous discs and the origin of the nebular variables. Mon. Not. R. Astron. Soc. 168, 603–637. doi:10.1093/mnras/168.3.603.

Malamud, U., Perets, H.B., Schäfer, C., Burger, C., 2018. Moonfalls: collisions between the earth and its past moons. Mon. Not. R. Astron. Soc. 479, 1711–1721. doi:10.1093/mnras/sty1667.

Malamud, U., Perets, H.B., Schäfer, C., Burger, C., 2020. Collisional formation of massive exomoons of superterrestrial exoplanets. Mon. Not. R. Astron. Soc. 492, 5089–5101. doi:10.1093/mnras/staa211.

Marcus, R.A., Sasselov, D., Hernquist, L., Stewart, S.T., 2010a. Minimum radii of super-earths: Constraints from giant impacts. Astrophys. J. Lett. 712, L73–L76. doi:10.1088/2041-8205/712/1/L73.

Marcus, R.A., Sasselov, D., Stewart, S.T., Hernquist, L., 2010b. Water/icy super-earths: Giant impacts and maximum water content. Astrophys. J. Lett. 719, L45–L49. doi:10.1088/2041-8205/719/1/L45.

McInnes, L., Healy, J., Melville, J., 2018a. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv e-prints arXiv:1802.03426.

McInnes, L., Healy, J., Saul, N., Grossberger, L., 2018b. Umap: Uniform manifold approximation and projection. J. Open Source Softw. 3 (861).

Melosh, H., 1996. Impact Cratering: A Geologic Process. In: Oxford monographs on geology and geophysics, Oxford University Press.

Melosh, H.J., 2007. A hydrocode equation of state for sio2. Meteorit. Planet. Sci. 42, 2079–2098. doi:10.1111/j.1945-5100.2007.tb01009.x.

Mocz, P., Succi, S., 2015. Numerical solution of the nonlinear Schrödinger equation using smoothed-particle hydrodynamics. Phys. Rev. E 91, 053304. doi:10.1103/PhysRevE.91.053304.

Monaghan, J.J., 1992. Smoothed particle hydrodynamics. Annu. Rev. Astron. Astrophys. 30, 543–574. doi:10.1146/annurev.aa.30.090192.002551.

Monaghan, J.J., 2005. Smoothed particle hydrodynamics. Rep. Progr. Phys. 68, 1703–1759. doi:10.1088/0034-4885/68/8/r01.

Monaghan, J., 2012. Smoothed particle hydrodynamics and its diverse applications. Annu. Rev. Fluid Mech. 44, 323–346. doi:10.1146/annurev-fluid-120710-101220.

Monaghan, J., Gingold, R., 1983. Shock simulation by the particle method sph. J. Comput. Phys. 52, 374–389. doi:10.1016/0021-9991(83)90036-0.

Murnaghan, F.D., 1937. Finite deformations of an elastic solid. Amer. J. Math. 59, 235–260.

Nakamura, A.M., Michel, P., Setoh, M., 2007. Weibull Parameters of yakuno basalt targets used in documented high-velocity impact experiments. J. Geophys. Res. (Planets) 112, E02001. doi:10.1029/2006JE002757.

Ortiz, J.L., Santos-Sanz, P., Sicardy, B., Benedetti-Rossi, G., Bérard, N., Duffard, R., Braga-Ribas, F., Hopp, U., Ries, C., Nascimbeni, V., Marzari, F., Granata, V., Pál, A., Kiss, C., Pribulla, T., Komžík, R., Hornoch, K., Pravec, P., Bacci, P., Maestripieri, M., Nerli, L., Mazzei, L., Bachini, M., Martinelli, F., Succi, G., Ciabattari, F., Mikuz, H., Carbognani, A., Gaehrken, B., Mottola, S., Hellmich, S., Rommel, F.L., Fernández-Valenzuela, A.C., Cikota, S., Cikota, A., Lecacheux, J., Vieira-Martins, R., Camargo, J.I.B., Assafin, M., Colas, F., Behrend, R., Desmars, J., Meza, E., Alvarez-Candal, A., Beisker, W., Gomes-Junior, A.R., Morgado, B.E., Roques, F., Vachier, F., Berthier, J., Mueller, T.G., Madiedo, J.M., Unsalan, O., Sonbas, E., Karaman, N., Erece, O., Koseoglu, D.T., Ozisik, T., Kalkan, S., Guney, Y., Niaei, M.S., Satir, O., Yesilyaprak, C., Puskullu, C., Kabas, A., Demircan, O., Alikakos, J., Charmandaris, V., Leto, G., Ohlert, J., Christille, J.M., Szakáts, A.T., Varga-Verebélyi, E., Marton, G., Marciniak, A., Bartczak, P., Santana-Ros, T., Butkiewicz-Bąk, M., Dudziński, V., Gazeas, K., Tzouganatos, L., Paschalis, N., Tsamis, V., Sánchez-Lavega, A., Pérez-Hoyos, S., Hueso, R., Guirado, J.C., Peris, V., Iglesias-Marzoa, R., 2017. The size, shape, density and ring of the dwarf planet haumea from a stellar occultation. Nature 550, 219–223. doi:10.1038/nature24051.

Owen, J.M., 2001. An open-source project for modeling hydrodynamics in astrophysical systems. Comput. Sci. Eng. 3, 54–59. doi:10.1109/5992.963428.

Piekutowski, A.J., 2001. Debris clouds produced by the hypervelocity impact of nonspherical projectiles. Int. J. Impact Eng. 26, 613–624. doi:10.1016/S0734-743X(01)00122-1.

Price, D.J., 2012. Smoothed particle hydrodynamics and magnetohydrodynamics. J. Comput. Phys. 231, 759–794. doi:10.1016/j.jcp.2010.12.011.

Price, D.J., Wurster, J., Nixon, C., Tricco, T.S., Toupin, S., Pettitt, A., Chan, C., Laibe, G., Glover, S., Dobbs, C., Nealon, R., Liptai, D., Worpel, H., Bonnerot, C., Dipierro, G., Ragusa, E., Federrath, C., Iaconi, R., Reichardt, T., Forgan, D., Hutchison, M., Constantino, T., Ayliffe, B., Mentiplay, D., Hirsh, K., Lodato, G., 2017. PHANTOM: Smoothed particle hydrodynamics and magnetohydrodynamics code. Astrophys. Source Code Libr. arXiv:1709.002.

Ramachandran, Prabhu, 2016. PySPH: a reproducible and high-performance framework for smoothed particle hydrodynamics, editor=sebastian benthall and scott rostrup. In: Proceedings of the 15th Python in Science Conference. pp. 122–129. doi:10.25080/Majora-629e541a-011.

Rein, H., Liu, S.F., 2012. REBOUND: an open-source multi-purpose N-body code for collisional dynamics. Astron. Astrophys. 537, A128. doi:10.1051/0004-6361/201118085.

Rein, Hanno, Spiegel, David S., 2015. Ias15: a fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits. Mon. Not. R. Astron. Soc. 446 (2), 1424–1437. doi:10.1093/mnras/stu2164.

Rosswog, S., 2015. Sph methods in the modelling of compact objects. Liv. Rev. Comput. Astrophys. 1, 1. doi:10.1007/lrca-2015-1.

Schäfer, C., Riecker, S., Maindl, T.I., Speith, R., Scherrer, S., Kley, W., 2016. A smooth particle hydrodynamics code to model collisions between solid, self-gravitating objects. Astron. Astrophys. 590, A19. doi:10.1051/0004-6361/201528060.

Schäfer, C.M., Scherrer, S., Buchwald, R., Maindl, T.I., Speith, R., Kley, W., 2017. Numerical simulations of regolith sampling processes. Planet. Space Sci. 141, 35–44. doi:10.1016/j.pss.2017.04.015.

Schäfer, C., Speith, R., Hipp, M., Kley, W., 2004. Simulations of planet-disc interactions using smoothed particle hydrodynamics. Astron. Astrophys. 418, 325–335. doi:10.1051/0004-6361:20034034.

Schäfer, C., Speith, R., Kley, W., 2007. Collisions between equal-sized ice grain agglomerates. Astron. Astrophys. 470, 733–739. doi:10.1051/0004-6361:20077354.

Schaller, Matthieu, Gonnet, Pedro, Draper, Peter W., Chalk, Aidan B. G., Bower, Richard G., Willis, James, Hausammann, Loïc, 2018. Swift: sph with inter-dependent fine-grained tasking. ascl:1805.020, arXiv:1805.020.

Sirono, S.i., 2004. Conditions for collisional growth of a grain aggregate. Icarus 167.

Speith, R., Kley, W., 2003. Stability of the viscously spreading ring. Astron. Astrophys. 399, 395–407. doi:10.1051/0004-6361:20021783.

Springel, V., 2000. GADGET-2: A code for cosmological simulations of structure formation. Astrophysics Source Code Library arXiv:0003.001.

Stellingwerf, R.F., Wingate, C.A., 1994. Impact modelling with SPH (invited paper). Mem. Soc. Astron. Ital. 65 (1117).

Szewc, K., 2016. Smoothed particle hydrodynamics modeling of granular column collapse. Granul. Matter 19, 3. doi:10.1007/s10035-016-0684-3.

Thun, D., Kley, W., 2018. Migration of planets in circumbinary discs. Astron. Astrophys. 616, A47. doi:10.1051/0004-6361/201832804.

Thun, D., Kley, W., Picogna, G., 2017. Circumbinary discs: Numerical and physical behaviour. Astron. Astrophys. 604, A102. doi:10.1051/0004-6361/201730666.

Tillotson, J.H., 1962. Metallic Equations of State for Hypervelocity Impact. Technical Report General Atomic Report GA-3216. General Dynamics. San Diego, CA.

von Mises, R., 1913. Mechanik der festen Körper im plastisch deformablen Zustand. Göttin. Nachr. Math. Phys. 1, 582–592.

Wadsley, J.W., Keller, B.W., Quinn, T.R., 2017. Gasoline2: a modern smoothed particle hydrodynamics code. Mon. Not. R. Astron. Soc. 471, 2357–2369. doi:10.1093/mnras/stx1643.

Wandel, O.J., Schäfer, C.M., Maindl, T.I., 2017. Collisional fragmentation of porous objects in planetary systems. In: Proceedings of the First Greek-Austrian Workshop on Extrasolar Planetary Systems, vol. 22. pp. 5–242.

Yu, M., Huang, Y., Deng, W., Cheng, H., 2018. Forecasting landslide mobility using an sph model and ring shear strength tests: a case study. Natural Hazards Earth Syst. Sci. 18, 3343–3353. doi:10.5194/nhess-18-3343-2018.

Zander, C., Hopp-Hirschler, M., Nieken, U., 2018. Mesoscopic simulation and characterization of the morphological evolution in phase separating fluid mixtures. Comput. Mater. Sci. 149, 267–281. doi:10.1016/j.commatsci.2018.03.019.

Protoplanet collisions: Statistical properties of ejecta

5.

The capabilities of our GPU-accelerated SPH code enabled us to perform compute-intensive hyperparameter studies. In order to cover the high-dimensional space of possible initial conditions, we ended up simulating a total of 880 unique, low-resolution collisions. Individual outcomes of these collisions can be roughly categorized into three major scenarios: mergers, hit-and run types, as well as erosion types.

Using the collision-catalog, we worked on analyzing water retention and water transport between projectiles and targets, resulting in a conference contribution [74]. Having available such a collision-catalog also opened up the possibility to derive some statistical conclusions about post-collision fragments. A thorough analysis of those fragments resulted in a journal publication in 2021, which is a minor contribution in the context of this thesis [24].

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **508,** 6013–6022 (2021) https://doi.org/10.1093/mnras/stab2951
Advance Access publication 2021 October 13

Protoplanet collisions: Statistical properties of ejecta

Samuele Crespi,[1,2]★ Ian Dobbs-Dixon,[1,2,3] Nikolaos Georgakarakos,[1,2] Nader Haghighipour,[4,5] Thomas I. Maindl,[6,7] Christoph M. Schäfer[8] and Philip Matthias Winter[9]

[1]*New York University Abu Dhabi, PO Box 129188 Abu Dhabi, United Arab Emirates*
[2]*Center for Astro, Particle and Planetary Physics (CAP*^{3}*), New York University Abu Dhabi, PO Box 129188 Abu Dhabi, United Arab Emirates*
[3]*Center for Space Science, New York University Abu Dhabi, PO Box 129188 Abu Dhabi United Arab Emirates*
[4]*Planetary Science Institute, 1700 East Fort Lowell, Tucson, AZ 85719, USA*
[5]*Institute for Astronomy, University of Hawaii-Manoa, Honolulu, HI 96822, USA*
[6]*Department of Astrophysics, University of Vienna, A-1180 Vienna, Austria*
[7]*SDB Science-driven Business Ltd, 6025 Larnaca, Cyprus*
[8]*Institut für Astronomie und Astrophysik, Eberhard Karls Universität Tübingen, Auf der Morgenstelle 10, D-72076 Tübingen, Germany*
[9]*Institute for Machine Learning, Johannes Kepler University Linz, Altenberger Straße 69, A-4040 Linz, Austria*

Accepted 2021 October 5. Received 2021 October 3; in original form 2021 July 18

ABSTRACT

The last phase of the formation of rocky planets is dominated by collisions among Moon- to Mars-sized planetary embryos. Simulations of this phase need to handle the difficulty of including the post-impact material without saturating the numerical integrator. A common approach is to include the collision-generated material by clustering it into few bodies with the same mass and uniformly scattering them around the collision point. However, this approach oversimplifies the properties of the collision material by neglecting features that can play important roles in the final structure and composition of the system. In this study, we present a statistical analysis of the orbital architecture, mass, and size distributions of the material generated through embryo–embryo collisions and show how they can be used to develop a model that can be directly incorporated into the numerical integrations. For instance, results of our analysis indicate that the masses of the fragments follow an exponential distribution with an exponent of -2.21 ± 0.17 over the range of 10^{-7} to 2×10^{-2} Earth-masses. The distribution of the post-impact velocities show that a large number of fragments are scattered towards the central star. The latter is a new finding that may be quite relevant to the delivery of material from the outer regions of the asteroid belt to the accretion zones of terrestrial planets. Finally, we present an analytical model for the 2D distribution of fragments that can be directly incorporated into numerical integrations.

Key words: planets and satellites: formation – planets and satellites: terrestrial planets.

1 INTRODUCTION

Based on the standard model for planet formation, the last phase of rocky planet formation is dominated by collisions between Moon- to Mars-sized solid bodies, known as protoplanets or planetary embryos. To a first approximation, this phase can be simulated as the evolution of a system of objects that interact exclusively via gravity (e.g. Ida & Makino 1993; Raymond, Quinn & Lunine 2006; Barnes et al. 2009; Morishima 2015; Clement et al. 2019). These simulations are generally carried out using N-body integrators with the largest constraint being the number of bodies, usually kept below a few hundred to avoid lengthy computations.

As the time of N-body integrations increases with the number of bodies, in order to ensure that formation simulations are carried out in a reasonable amount of time, collisions are often treated as perfectly inelastic. That is, the two colliding bodies merge completely and no fragments and debris are produced (e.g. Raymond, Quinn & Lunine 2004; O'Brien, Morbidelli & Levison 2006; Morishima 2015). This approach has been proven to be useful in demonstrating the formation process, especially as a proof of concept. However, it tends to overestimate the collision efficiency, leading to shorter formation time and more massive planets. The latter causes the final planetary system to have smaller number of bodies with planets being farther away from the central star (Bazsó & Schäfer 2020; Dugaro; de Elía & Darriba 2020).

While simulating collisions has been an integral part of planetary science for decades (e.g. Dormand & Woolfson 1977; Benz, Slattery & Cameron 1986; Wetherill 1988), inclusion of collisional fragments in formation simulations has not been straightforward. The reason lies in the fact that collisions produce million and millions of fragments, and including all these in an N-body integration is impractical. Many researchers have tried to circumvent this issue by including only a few of the impact-generated bodies (Chambers 2013; Clement et al. 2019). Another approach is to use a collision catalogue to identify the post-impact bodies that have the larger contributions (i.e. those with larger mass) and ignore the rest (Burger et al. 2020).

Although the above approach has advanced the simulations of terrestrial planet formation closer to realistic ones, it includes only a small portion of the impact-generated bodies. A comprehensive

★ E-mail: sc6459@nyu.edu

formation model requires all this material to be included. However, as explained above, it is impractical to include all post-impact fragments as individual objects. Fortunately, the total mass of these bodies presents a promising path forward.

Because the effect of post-collision objects on the rest of the system is through their gravitational interactions and accretion, using the collective mass of the objects produced after a collision allows for both these factors to be taken into account. The only question will be, how to model this mass. Clement et al. (2019) suggested to distribute the total post-impact mass as a collection of lunar-mass bodies. Carter et al. (2015) included them as unresolved debris by uniformly distributing fragments into circular rings. While both these approaches have been promising, their underlying assumptions are too simplistic. In real collisions, fragments vary considerably in size and velocity, and their distribution is not uniform. A comprehensive model needs to take all this into account. In this paper, we investigate the statistical properties of the material ejected by collisions during the collision dominated phase of the rocky planet formation. In particular, we will use a catalogue of SPH (Smoothed-Particles Hydrodynamics) simulations of collisions to determine the distributions of the mass, velocity, and orbits of the ejecta.

This paper is arranged as follows. We present in Section 2 the two catalogues used in this study, namely the catalogue of N-body impacts and the catalogue of SPH impact simulations. In Section 3, we present and analyse the results of our simulations, and in Section 4, we conclude this study by summarizing the results and discussing their implications.

2 COLLISION CATALOGUES

In this section, we present the two collision catalogues used in our study. The first catalogue consists of 1356 protoplanet collisions and is obtained by N-body simulations of the late stage of terrestrial planet formation in 10 different systems. This catalogue, hereafter referred to as the PC (protoplanetary collision) catalogue, contains the physical properties of the colliding bodies prior to the time of each collision. The second catalogue contains the outcome of the SPH simulations of protoplanetary bodies for different initial conditions. We refer to this catalogue as CO (collision outcome) catalogue. In the following, we explain these two catalogues in more detail.

2.1 Protoplanets collisions catalogue

To generate a collection of protoplanetary collisions, we simulated the late stage of terrestrial planet formation for 10 different distributions of planetary embryos. We considered a system consisting of the Sun, Jupiter in a circular orbit at 5.2 AU, and a disc of protoplanetary bodies extending from 0.5 AU to 4.7 AU. The protoplanetary disc contained around 200 planetary embryos with masses ranging from 0.02 to 0.1 Earth-masses. Following Kokubo & Ida (2000), planetary embryos were placed at distances of 5–10 mutual Hill's radii, and the surface density of the disc was set to followed a $\Sigma = \Sigma_1 r^{-3/2}$ profile with $\Sigma_1 = 10\ \mathrm{g\,cm^{-3}}$. The 10 different simulations differed only in the initial mass and orbits of protoplanets. We integrated each system for 1 Myr using the hybrid routine in the N-body integration package `mercury` (Chambers 1999). The integration time-step was set to 6 d.

The collisions in the PC catalogue are uniquely identified by three parameters: the total mass of the colliding bodies ($m_1 + m_2 = m_{\rm tot}$), their mass-ratio ($\gamma = m_1/m_2$), and their impact velocity (v_0) normalized to the mutual escape velocity (hereafter, normalized impact velocity).[1] Here, m_1 and m_2 are the masses of the two colliding bodies at the moment of impact. Other parameters, such as the collision angle, can also play a role in defining the collisional outcome. However, to avoid computational complexities, they are not investigated in this study. Instead, we consider the averaged value of these parameters, as described in Section 3.3.

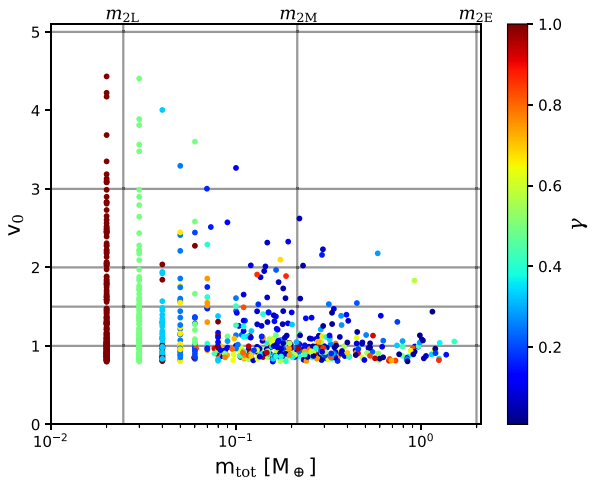


Figure 1. Distribution of all 1356 collisions in the PC catalogue. The x and y axes show the total mass and the normalized impact velocity, respectively. The colour-coding indicates the mass-ratio of the colliding bodies. The horizontal and vertical lines correspond to the specific set of values sampled in the CO catalogue.

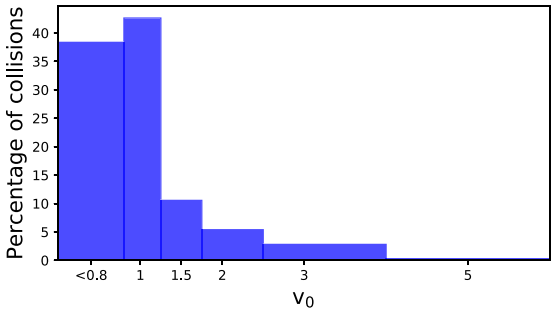


Figure 2. Histogram of the normalized impact velocity (v_0) for the collisions of the PC catalogue.

Fig. 1 shows the distribution of all 1356 collisions in term of $m_{\rm tot}$, γ and v_0. As shown here, high velocity collisions ($v_0 > 3$) are extremely rare (less than 2 per cent of all collisions) and usually occur among smaller bodies ($m_{\rm tot} < 7 \times 10^{-2}\ \mathrm{M_\oplus}$). The dominance of low velocity impacts is more evident in Fig. 2 where we show a histogram of collision velocities. As many as 893 collisions (approximately 2/3 of all collisions) have impact velocities smaller than the mutual escape velocities of the colliding bodies ($v_0 < 1$).

[1]The mutual escape velocity is given by $v_{\rm esc} = [2G(m_1 + m_2)/(r_1 + r_2)]^{1/2}$, where r_1 and r_2 are the radii of the two colliding bodies and G is the gravitational constant.

Table 1. Parameter values sampled in the 880 simulations of the CO catalogue.

Parameter	Value
Total mass ($M_\oplus$)	3.14×10^{-4}, 2.46×10^{-2}, 0.215, 2
Mass-ratio	0.1, 0.5, 1.0
Impact velocity (v_{esc})	1, 1.5, 2, 3, 5
Impact angle (deg)	0, 20, 40, 60
Body 1 water mass fraction w_1	0.1, 0.2
Body 2 water mass fraction w_2	0.1, 0.2

2.2 Collision output catalogue

Our catalogue of collision outcomes consists of 880 low-resolution SPH simulations (about 20 000 particles[2] per scenario) of two colliding protoplanets. The collision simulations were performed using the GPU SPH code MILUPHCUDA (Schäfer et al. 2016, Schäfer et al. 2020).

The initial setup of projectile and target includes SPH particles on hexagonal close-packed grids in hydrostatic equilibrium. Target and projectile are initially set as not rotating and placed at a mutual distance of $r_{ini} = 5(r_1 + r_2)$, where r_1 and r_2 denote the radius of the target and projectile, respectively.

Both the projectile and target are modelled as silicates surrounded by ice shells to account for their water-mass fraction. We use the Tillotson equation of state with input parameters of basalt and ice (Schäfer et al. 2016) to determine the pressure. The material strength is also included where the shear strength is limited by the von Mises yield criterion. To model tensile fractures, we follow Benz & Asphaug (1994) and use the Grady-Kipp fragmentation model and the usual standard artificial viscosity by Monaghan & Pongracic (1985). The gravitational forces between the particles are calculated using a Barnes-Hut tree.

Table 1 shows the range of initial conditions for the collision simulations. The values of the mass correspond to the total colliding mass of two Ceres-sized bodies (hereafter labelled m_{2C}), two Moon-sized bodies (labelled m_{2L}), two Mars-sized bodies (labelled m_{2M}), and two Earth-sized bodies (labelled m_{2E}). To maintain generality, two different values of water-mass fraction are considered for each body. However, the composition of the ejected material is not investigated.

2.2.1 *Collision-type criteria*

Each SPH simulation produces the positions, velocities, and masses of all bodies formed ∼9 to 12 h after the impact. To determine the type of the collision, the post-impact bodies must be examined for possible gravitationally bound objects and also to separate the main bodies from fragments.

To determine if two bodies are gravitationally bound, we compare their relative velocity (v_{rel}) with their mutual escape velocity. If v_{rel} is smaller than the mutual escape velocity, we consider the two bodies gravitationally bound and combine them into a single object at their centre of mass. We continue this procedure until no pairs of gravitationally bound ejecta are left. Results indicated that 15 per cent–30 per cent of the post-impact material are gravitationally bound to their main bodies and is accreted back on to those objects.

To separate the main bodies (meaning the remnants of the colliding bodies if not destroyed) from fragments, we use the following criterion: if the impacting body loses more than 90 per cent of its original mass, it is considered to have been destroyed and is no longer a main body (Leinhardt & Stewart 2012).

Through these two processes, we identify five possible types of collisions:

(i) *Perfect merging* (PM): the two colliding bodies merge and produce one main body with no fragments;
(ii) *quasi-perfect merging* (qPM): more than 90 per cent of one of the two colliding bodies is accreted by the other producing one main body and a small number of fragments;
(iii) *Partial accretion* (PA): less than 90 per cent of one of the two colliding bodies is accreted by the other resulting in one or two main bodies and fragments;
(iv) *Erosive collision* (EC): the biggest body loses less than 90 per cent of its mass during the collision resulting in one or two main bodies and fragments;
(v) *Catastrophic collision* (CC): both colliding bodies are destroyed and all mass is converted into fragments.

We would like to note that the above five collision outcomes are rather different from those described by Leinhardt & Stewart (2012). In particular, these authors divide PA and EC collisions into smaller categories including pure hit-and-run, erosive hit-and-run, cratering and grazing collisions. We decided not to follow this categorization as, in our data set, it is impossible to track which bodies are the two colliding bodies at the end of a simulation.

Results of our SPH simulations indicated that the collision type strongly depends on the impact velocity and the impact angle while the effect of the other parameters such as porosity and material composition is almost negligible. In particular:

(i) for collisions with impact velocities comparable with the escape velocity ($v_0 = 1$) the result is always a PM or qPM,
(ii) collisions with impact velocities at least three times larger than the escape velocity ($v_0 > 3$) never result in PM,
(iii) collisions with impact velocities five times larger than the escape velocity ($v_0 = 5$) either result in EC or CC.

These findings are consistent with the predicted collision outcomes in Leinhardt & Stewart (2012). Fig. 3 shows the occurrence of these collisions as a function of the impact velocity v_0 for all our SPH simulations.

3 ANALYSIS AND DISCUSSION

In this section, we use the SPH results presented in the CO catalogue to present an analysis of the possible outcomes of the collisions presented in the PC catalogue. To begin with, we use the collision classification introduced in the last section to identify the type of collisions that can emerge from the PC catalogue. Fig. 4 shows the results. As shown here, a large majority of the collisions (84.9 per cent) result in perfect or quasi-perfect merging. Other types of collisions also contribute, but at much smaller extent; partial accretion appears at 7.7 per cent, erosion occurs at 7.2 per cent, and

[2]Burger, Maindl & Schäfer (2018) analysed collision outcomes for resolutions between a few 10k and 2.25M SPH particles. They found that even for highly energetic collisions of Earth- and Mars-mass bodies, the global outcomes (kinematics and masses of fragments) show only minor deviations below 10 per cent with varying resolutions, hence only marginally impacting the global outcome classification. The water mass fraction was found to be accurate within only a few percent for the largest surviving body and within about 25 per cent for the second largest fragment. Being mainly interested in the overall surviving fragment masses and kinematics, we settle for a resolution of 20k SPH particles to balance performance and accuracy.

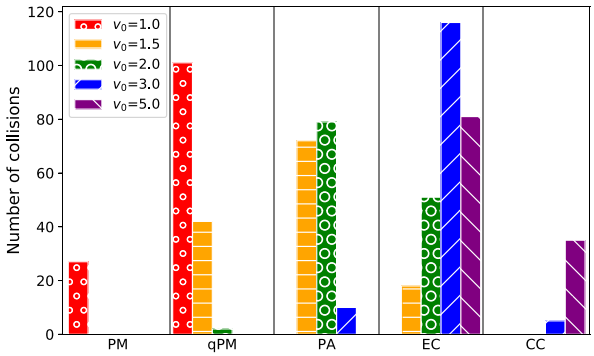


Figure 3. Occurrence of different collision outcomes in all the SPH simulations in the CO catalogue. The colour coding shows the impact velocity in units of mutual escape velocity.

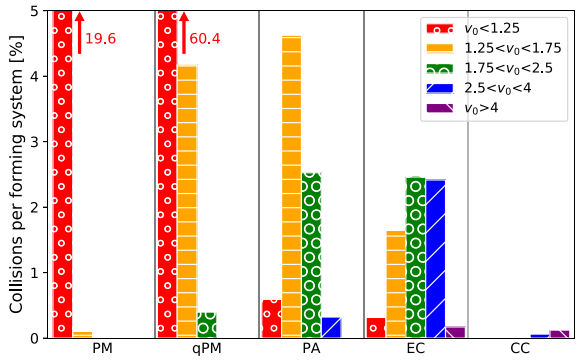


Figure 4. Occurrence of the five different collision types per forming system (PC catalogue). The colour and texture indicates the impact velocity interval in units of the mutual escape velocity between the two colliding bodies.

complete shattering of the colliding bodies (catastrophic collision) happens only 0.18 per cent of the time.

3.1 Total mass of fragments

Using the results of the SPH simulations, we determined the total mass of the fragments that could be produced in each PC collision. We found that impacts with partial accretion would have the biggest contribution, producing most of the fragments' mass. The largest post-impact total mass produced in all collisions in the PC catalogue is that of a PA collision between two embryos with masses of $0.29M_{\oplus}$ and $0.26M_{\oplus}$. Despite their relatively low impact velocity ($v_0 = 1.8$), these bodies can produce approximately $0.1M_{\oplus}$ of fragments, equivalent to 17 per cent of the initial mass of their corresponding system. Results indicate that on average, three such collisions appear in each formation simulation where the collective contribution of these collisions to the total mass of produced fragments could be as large as $\sim 0.067M_{\oplus}$.

Our analysis also indicates that although catastrophic collisions convert all colliding masses into debris, their contributions to the total mass of the fragments are small. Because these collisions happen only for high impact velocities ($v_0 > 4$) and relatively small total colliding mass ($m_{\rm tot} < 0.07$ M$_{\oplus}$), they are rare. Only 0.14 per cent of all collisions in the PC catalogue are of CC type. For that reason, these collisions do not have significant contributions to the total mass of fragments.

Results showed that on average, in each simulation, 0.192 M$_{\oplus}$, corresponding to 4.2 per cent of the total initial mass of a system is converted into debris. This value is smaller than those reported in the study by Burger et al. (2020) where 18 per cent – 24 per cent of the initial mass is converted into fragments. We believe this discrepancy has roots in the differences between the initial setups in our simulations and those by these authors. For instance, these authors included Saturn which can further perturb the orbits of planetary embryos and increase their impact velocities. They also included more bodies which increases the mutual interactions among the embryos, subsequently increasing their impact velocities. The 4.2 per cent total fragments mass in our simulations should, therefore, be taken as a conservative lower limit which is still significant enough to emphasize the importance of including fragments in formation simulations. The latter is especially important when studying the composition of the final bodies.

We would like to note that for the sake of completeness and generality, we also carried out 10 more PC-type simulations where we included Saturn as well. We will analyse these collisions and the degree to which Saturn contributed to the collision outcomes in Section 3.3.1.

3.2 Fragment mass distribution

The mass distribution of the post-impact material is often neglected in simulations that include fragmentation. It has been customary to include all post-impact material into a few bodies, in some instances, even with equal masses (e.g. Clement et al. 2019; Poon et al. 2020; Scora et al. 2020). However, as pointed out by some authors (Mustill, Davies & Johansen 2018; Dugaro et al. 2020), fragments of different masses can affect the evolution of the system differently, both from a physical and a chemical perspective. In this section, we present an analysis of the mass distribution of post-impact bodies and we derive an equation that can be incorporated into formation simulations.

Fig. 5 shows a histogram of the masses of the fragments in all simulations of the CO catalogue. The coloured graphs correspond to the mass distribution of fragments produced in the four subsets of collisions with the total colliding masses of $m_{\rm 2C}$, $m_{\rm 2L}$, $m_{\rm 2M}$, and $m_{\rm 2E}$ as given in Table 1. As shown here, no significant differences exist in the mass distributions of these impacts. In all four cases, the region corresponding to $(10^{-3.5} - 10^{-2})\,m_{\rm tot}$ shows a linear trend in log–log space with a slope varying from -1.09 for the $m_{\rm 2M}$ collision to -1.32 for the system of $m_{\rm 2L}$. Fig. 5 also shows that except for the small interval of (3.1–7.8)$10^{-6}M_{\oplus}$ (which is due to the fact that $m_{\rm 2C}$ masses are two orders of magnitude smaller than those of $m_{\rm 2L}$), the distribution of final fragments masses show an overlap (see the bottom panel). The latter suggests that, in the range of $(10^{-3.5} - 10^{-2})\,m_{\rm tot}$, the post-impact distribution of mass in all these four collisions can be studied collectively. We found that, together, these fragments present an exponential distribution given by

$$n(m)\,{\rm d}m \propto m^{-2.21\pm0.17}\,{\rm d}m, \quad (1)$$

valid in the mass range $(10^{-7} - 2 \times 10^{-2})M_{\oplus}$. Here, $n(m)$ is the number of fragments with mass between m and $m + {\rm d}m$. It is important to mention that the choice of the mass range of $(10^{-3.5} - 10^{-2})\,m_{\rm tot}$ was to prevent the effects of computational uncertainties. For instance, for smaller values of fragments' masses, the distribution strongly depends on the resolution of the SPH simulations, and for larger values, the distribution is dominated by the largest survivors. In particular, the five peaks between -1.04 and 0 correspond to hit-and-run scenarios for which the mass of two colliding bodies remain nearly unchanged after the collision.

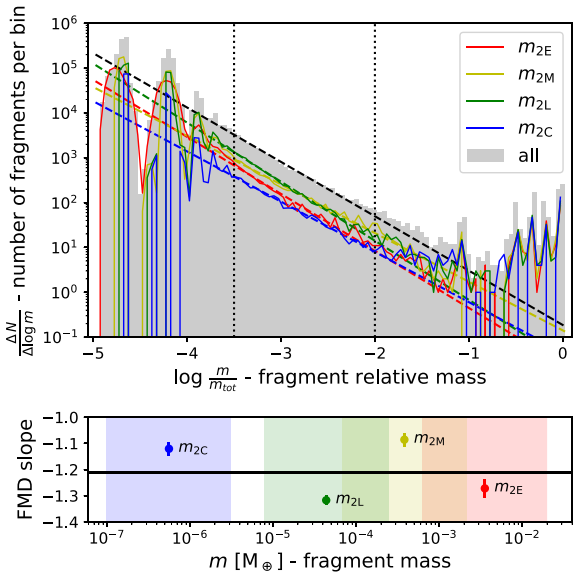


Figure 5. Top: Normalized fragments mass distribution. The grey area corresponds to all the simulations of the CO catalogue. The four coloured lines correspond to simulations with different total masses: red for m_{2E}, yellow for m_{2M}, green for m_{2L}, and blue for m_{2C}. The dashed lines indicate the best linear fit for each distribution in the range $10^{-3.5}$–10^{-2} (i.e. within the two vertical dotted lines). Bottom: Distribution of the slopes of fits in the top panel. The coloured areas correspond to the mass range used in each fit and their corresponding slope are shown at the centre of each range. The black line indicates the slope of the overall distribution.

In order to obtain the error on the slope value, we performed a similar analysis for sets of collisions grouped according to the mass ratio γ and the normalized impact velocity v_0. We found that the slope of the mass distribution varies between -2.58 and -1.96 with a standard deviation of 0.17 for collisions with $v_0 = 5$ and 3, respectively. We also noticed that as the mass ratio γ increases from 0.1 to 0.5 and 1, the absolute value of the slope of the mass distribution decreases from -2.51 to -2.22 and -2.12.

Assuming constant bulk density and spherical shape for all fragments, equation (1) can be written as

$$N(D)\,\mathrm{d}D \propto D^{-4.63\pm0.52}\,\mathrm{d}D, \tag{2}$$

where D represents a fragment's diameter and $n(D)$ is the number of fragments with size between D and $D + \mathrm{d}D$. The distribution slope, that in the log–log space becomes -3.63 ± 0.52, is in full agreement with the median slope of -3.8 reported by Leinhardt & Stewart (2012). Equations (1) and (2) can be used to model fragmentation more realistically and to improve the mechanics of the inclusion of fragments in formation simulations.

3.3 Distribution of fragments

In this section, we determine the distribution of the fragments that can be produced in the collisions among planetary embryos. Because N-body integrations do not resolve embryo–embryo collisions, we use the orbital parameters of colliding bodies at the moment of their impact (i.e. their position in the parameter space of the PC catalogue) to identify their possible outcome using the results of the SPH simulations (i.e. their location in the parameter space of the CO catalogue).

In general, for each N-body collision, the knowledge of the total colliding mass ($m_{\rm tot}$), mass-ratio (γ), and normalized impact velocity (v_0) would be sufficient to uniquely identify that collision in the PC catalogue. However, because the two catalogues are independent from each other, each parameter set ($m_{\rm tot}$, γ, v_0) from the PC catalogue may not necessarily correspond to an exact initial condition for an SPH simulation in the CO catalogue. It is, therefore, necessary to develop a methodology to determine the degree to which the results of an SPH simulation can be used for a given set of embryo–embryo collision. This can be done by weighting the difference between the initial condition of an embryo–embryo collision in the PC catalogue and its closest initial condition in the parameter space of the CO catalogue. In the following, we explain our approach in developing such a weighting measure.

Recall that the initial conditions of the SPH simulations were divided into grids along all parameters. Let's assume that a parameter x from a set of initial conditions of a collision in the PC catalogue falls in the i-th grid in the parameter space of the CO catalogue where the value of the same quantity is given by x_i. We introduce the function K, hereafter referred to as the weight function, as a measure of the difference between the values x and x_i,

$$K(d) = \begin{cases} 1 & \text{if } d_i < \delta \\ 2 - (d/\delta) & \text{if } \delta < d_i < 2\delta \\ 0 & \text{if } d_i > 2\delta \end{cases}. \tag{3}$$

Here, $d_i = |x - x_i|$. We define the quantity $\delta = (x_i - x_{i+1})/3$ as the scaling length of the weight function corresponding to one-third of the difference between two consecutive intervals of the quantity x in the parameter space of the CO catalogue. We choose this finite length because, as demonstrated by the results of the SPH simulations, the outcome of a collision can vary considerably from one grid point to the next. From equation (3), if for a parameter x in the PC catalogue, $d_i > 2\delta$, we consider that parameter to be too far from x_i and, therefore, the results of the SPH simulations corresponding to the initial condition x_i cannot be used for that embryo–embryo collision in the PC catalogue.

In applying equation (3) to the results of the PC catalogue, it is important to note that for each set of ($m_{\rm tot}$, γ, v_0) parameters, SPH simulations were carried out for four different values of the impact angle and four different combinations of water-mass fraction. That means, for each set of the initial conditions in the PC catalogue, there are 16 SPH simulations with different impact angles and water content. Because the collisions in the PC catalogue do not include these parameters, we use the following averaging procedure to incorporate their total effects into account.

To remove the effects of water-mass fraction and impact angle for a given embryo–embryo collision, we calculate the total fragment mass ($m_{\rm fr}$) of that collision by averaging over all its 16 values and determine its corresponding location in the CO parameter space using equation (3);

$$m_{\rm fr}(m_{\rm tot}, \gamma, v_0) = \frac{1}{16\, m_{\rm tot}} \sum_{i,j,k} K_i K_j K_k \sum_{l,m,n} m_{\rm fr}(m_{\rm tot}, \gamma, v_0, w_{1i}, w_{2j}, \alpha_k) \tag{4}$$

While this procedure removes information about the mass of individual fragments, it gives the probability for having a fragment with a specific mass and initial conditions ($m_{\rm tot}$, γ, v_0).

We apply equations (3) and (4) to all 1356 collisions in the PC catalogue. To ensure that for each embryo–embryo collision, the quantities representing lengths are compatible in both catalogues, we

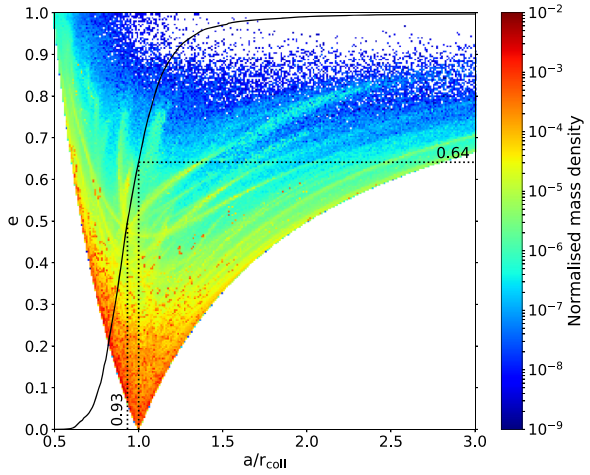


Figure 6. Fragments mass distribution in the $a - e$ plane. The semimajor axis is normalized by the radial distance between the star and the collision point $r_{\rm coll}$. The overplotted black line corresponds to the normalized cumulative mass of the fragments and the dotted lines highlight the points at $a/r_{\rm coll} = 1$ and $m_{\rm cumulative} = 0.5$. The streaks represent fragments produced during high-velocity impacts ($v_0 > 4$).

scale semimajor axis (a) and radial distance (r) in the PC catalogue by the distance of the location of the collision from the star ($r_{\rm coll}$). The final result consists of 9373 158 fragments with minimum mass of $10^{-6}M_{\oplus}$ for which we know the normalized semimajor axis ($a/r_{\rm coll}$), eccentricity (e), and argument of the periastron (ω). In the following, we use this approach to determine the distribution of post-impact material in the $a - e$ parameter space, and in the plane of the system.

3.3.1 *Fragments distribution on the* a − e *plane*

Fig. 6 shows the distribution of fragments on the $a - e$ plane. The colour-coding represents the amount of the mass confined in a grid-cell, per unit area of the cell. The solid black curve shows the cumulative normalized value of fragments mass. As expected, fragments are distributed between two branches $a(1 - e) < 1 < a(1 + e)$ originated at (1,0). We note that all lengths have been scaled by the radial distance of the collision point to the Sun, $r_{\rm coll} = 1$.

An interesting result depicted by Fig. 6 is the region where most of the fragments mass is concentrated. As shown here, after an impact, the majority of the mass stays in the vicinity of the collision point in low eccentricity orbits $(a, e) = (1, 0)$. To explain this feature, we show in Fig. 7 the x and y components of the velocity of a fragment relative to the velocity of the centre of mass of the system immediately after an impact, $\mathbf{v}_{\rm rel} = (\mathbf{v} - \mathbf{v}_{\rm CoM})/v_{\rm CoM}$. In this equation, $\mathbf{v}$ is the fragment's velocity and $\mathbf{v}_{\rm CoM}$ is the velocity of the centre of mass. As shown by the top panel of Fig. 7, the distribution of the relative velocity is nearly spherical and with magnitude generally smaller than 0.3 which confirms the concentration of the mass in the vicinity of the collision point in Fig. 6. As demonstrated by Jackson et al. (2014), this is consistent with the distribution of fragments that are generated in collisions between low-eccentricity bodies.

Fig. 7 also shows streaks of high-velocity fragments stretching away from the point of impact. These streaks, also seen in Fig. 6, are produced in high-velocity impacts ($v_0 > 2$) on high-eccentric orbits ($e_{\rm CoM} > 0.33$). The peculiar fragments distribution produced

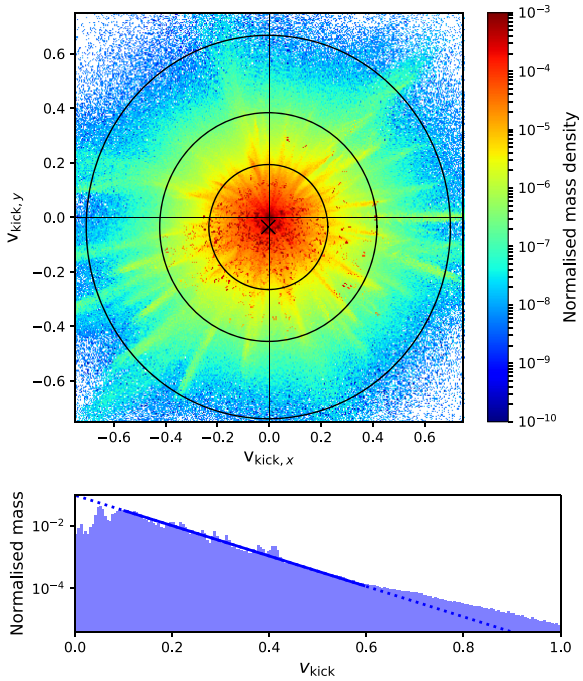


Figure 7. Top: Distribution of the relative velocity $v_{\rm rel}$ of fragments with respect to the centre of mass of the system. The x and y axes are oriented such that for each collision, the velocity of the centre of mass (shown by X) has components of $\hat{\mathbf{v}}_{\rm CoM} = (0, 1)$ and the total orbital angular momentum of the system points towards the reader. The three concentric circles correspond to the regions containing 90 per cent, 99 per cent, and 99.9 per cent of the total fragmented mass. Bottom: Histogram of the distribution of fragments velocities. The blue line corresponds to the exponential fit in equation (5). The dotted part of the line has been excluded from the fitting process.

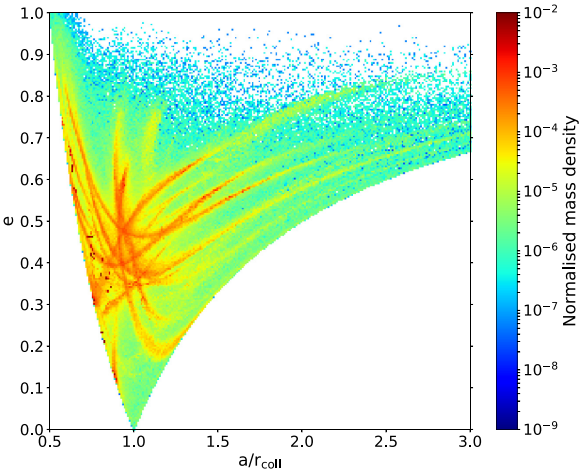


Figure 8. Similar to Fig. 6 but including only the collisions with normalized impact velocity $v_0 > 2$ and eccentricity $e_{\rm CoM} > 0.33$.

during these collisions, shown in Fig. 8, is consistent with the distribution obtained by Jackson et al. (2014) for collisions on high-eccentric orbits and it strongly depends on the eccentricity vector of the two colliding bodies as well as the direction in which the

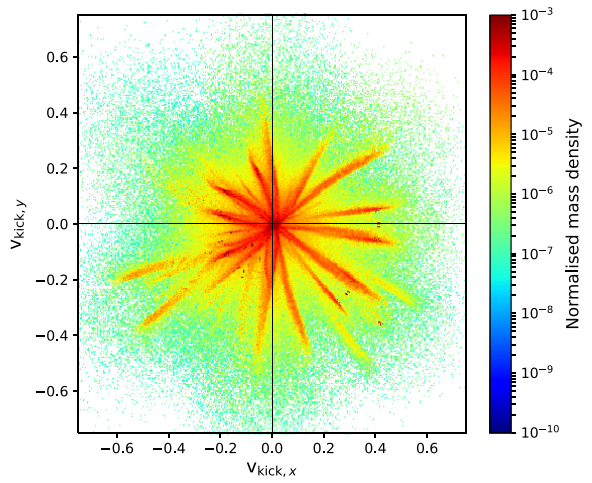


Figure 9. Similar to Fig. 7 but including only the collisions with normalized impact velocity $v_0 > 2$ and eccentricity $e_{CoM} > 0.33$.

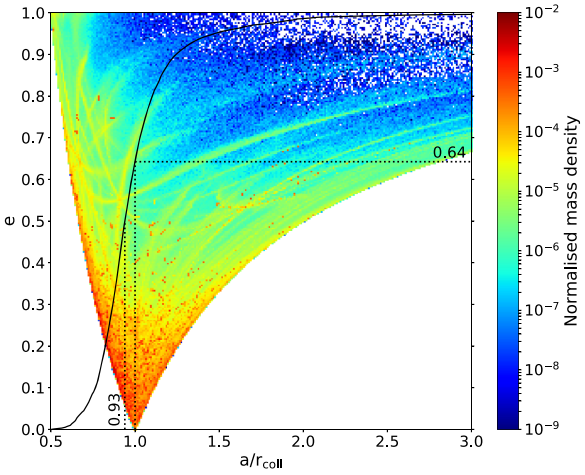


Figure 10. Similar to Fig. 6 with Saturn included.

fragments are ejected (streaks in Fig. 9). Even though the fragments distribution produced by these high-velocity and high-eccentric collisions significantly deviates from the overall distribution, the total amount of fragments produced accounts for only 4.1 per cent of the total mass of fragments from the 10 simulations studied. We therefore decided to keep these collisions in our analysis for completeness.

A detailed inspection of Fig. 7 shows that the centre of the velocity distribution lies slightly away from the impact location. While this displacement is negligible along the x-axis ($v_{\mathrm{kick},\,x} = -0.0037$), along the y-axis, it is rather significant ($v_{\mathrm{kick},\,y} = -0.036$), indicating a slight loss of angular momentum during the impact. This change in orbital angular momentum, that is partially due to the change in rotation of the main post-collisional bodies and partially due to the loss of energy during the impact, causes the distribution of the fragments to slightly deviate from the collision point and manifests itself in the form of small overpopulation towards one of the two boundaries. This can be seen in Fig. 6 where the left branch ($a < r_{\mathrm{coll}}$) is slightly more populated than the right branch ($a > r_{\mathrm{coll}}$).

The slight loss of energy and angular momentum causes the fragments to have slightly smaller semimajor axes. The graph of the cumulative mass distribution of these objects, shown by a solid black curve in Fig. 6, show that 64 per cent of the total mass of fragments are in orbits with semimajor axes smaller than the radial position of the collision point (r_{coll}). In other words, more than half of the fragments have semimajor axes smaller than $0.93\,r_{\mathrm{coll}}$. The magnitude of this net inward distribution of the post-impact material may not seem substantial for a single collision. However, when considered in the grand context of the giant impact phase, where there are numerous collision, its cumulative effect in bringing material closer to the central star may be significant. Furthermore, we investigated the possibility that this effect was produced by a specific class of collisions. However, we found that, apart from small variations in amplitude, this is not the case. For example, the high-velocity and high-eccentric collisions shown in Figs 8 and 9 produce a distribution displacement of $v_{\mathrm{kick},\,y} = -0.042$.

The bottom panel of Fig. 7 shows the histogram of the post-impact velocity of fragments in terms of their masses. As shown here, for the values of $v_{\mathrm{rel}} > 0.1$, the decrease in the amplitude of the relative velocity seems to show an exponential trend. For the values of $v_{\mathrm{rel}} < 0.1$, the distribution deviates from this trend and shows a decrease. This could be attributed to the re-accretion of low-velocity fragments by larger bodies. Our simulations indicate that the large majority of the fragments fall in the range of $0.1 < v_{\mathrm{kick}} < 0.6$. An exponential fit to this range of the distribution indicates

$$m(v_{\mathrm{kick}})\,\mathrm{d}v_{\mathrm{kick}} \propto \mathrm{e}^{-\tau v_{\mathrm{kick}}}\,\mathrm{d}v_{\mathrm{kick}}, \tag{5}$$

where scaling factor $\tau = 11.3 \pm 0.3$.

3.3.2 *Effect of Saturn*

For the sake of comparison, we carried out 10 simulations with the same initial condition as those of our PC catalogue (section 2.1) but included Saturn in a circular orbit. In this case, we obtained 1342 new collisions. Fig. 10 shows the distribution of their fragments. A comparison between this figure and Fig. 6 shows that the population on the left branch extends to higher values of eccentricities (∼0.8), and the distribution with Saturn contains more high-velocity streaks. This could be attributed to Saturn's secular resonance which strongly affects the orbit evolution of planetary embryos (Haghighipour & Winter 2016). A comparison with simulations without Saturn showed that the perturbation of Saturn almost doubled the number of high-velocity impacts (i.e. those with $v_0 > 4$). As important as the effect of Saturn is, such collisions make a small fraction of the total number of collisions and, statistically, their contribution to the average properties of the ejected material is negligible. For instance, similar to the system of Fig. 6, 64 per cent of the total fragmented mass has semimajor axes smaller than the radial position of collision r_{coll}.

3.3.3 *Fragments distribution on the* x − y *plane*

To construct the distribution of fragments on the $x - y$ plane, we calculated the probability that a fragment with a mass m_i would be in a specific grid-cell (Δx, Δy). Because fragments move, we weighted this probability by multiplying it by the time (Δt_i) that a fragment spends inside the cell, modulo the fragment's orbital period (P_i). That is,

$$\frac{\Delta m}{\Delta x \Delta y} = \sum_i \frac{m_i\,\Delta t_i / P_i}{\Delta x \Delta y}. \tag{6}$$

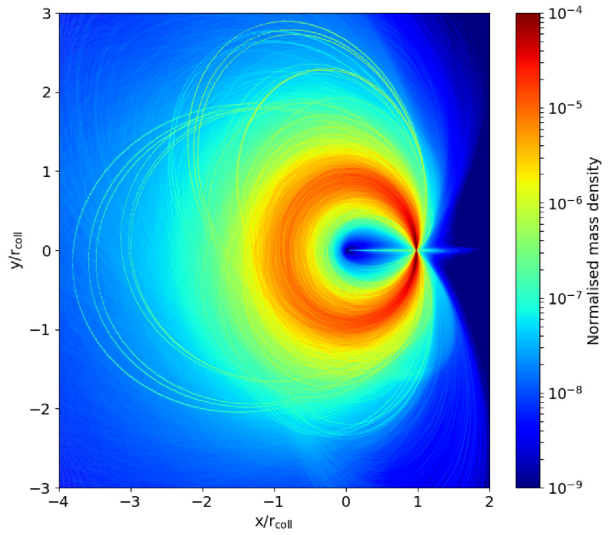


Figure 11. Spatial distribution of fragments mass in the $x - y$ plane. The collision point is at (1,0).

The distribution was then constructed by summing the contributions of all fragments. Fig. 11 shows the results.

It is important to note that the distribution in Fig. 11 did not appear immediately after the collision. As had been reported in other works (e.g. Jackson et al. 2014; Watt, Leinhardt & Su 2021), the post-impact material first formed spiral arms, which in a time-scale approximately equal to a hundred Keplerian orbital at $r_{\rm coll}$, evolve into the distribution shown in the figure. As expected, fragments distributed around the collision point at (1,0) and gradually spread along the azimuthal angle, almost symmetrical with respect to the line connecting the point of impact to the central star. Further examination of the distribution indicated that in the radial direction, the width of the distribution gradually increased with distance from the impact point with a maximum at the antipode of the collisions point with respect to the central star.

3.4 Analytical modelling

As mentioned in the beginning, our goal is to develop a (preferably analytical) model that can be used to demonstrate the outcome of collisions. In this section, we use the distribution shown in Fig. 11 to develop such a model.

Considering a polar-coordinate system with the central star at the origin and the impact point at ($r = 1, \theta = 0$) (the angular momentum vector pointing towards the reader), we find that the distribution shown in Fig. 11 can be modelled by the function

$$\rho\,(r,\theta) = A\,(\theta)\,\frac{r^{\,\beta(\theta)}}{\sqrt{1+\left[\frac{r}{r_0(\theta)}\right]^{\delta(\theta)}}}, \tag{7}$$

where the amplitude A and the exponents β and δ can be obtained using the equation

$$f(\theta) = p_0 - p_1 \cdot \left[\frac{\ln\,(p_2 + |\theta/\pi - 1|)}{p_2 + |\theta/\pi - 1|} + \frac{\ln\,(p_2 - |\theta/\pi - 1|)}{p_2 - |\theta/\pi - 1|}\right]. \tag{8}$$

The values for the parameters p_i, ($i = 0 - 2$) are given in Table 2, and the length scale r_0 (approximately the maximum in the radial mass distribution for a given value of θ) is given by a sinusoidal function

$$r_0\,(\theta) = 0.95 + 0.05 \cdot \cos\,(\theta + 0.013)\,. \tag{9}$$

Table 2. Best-fitting parameters for the functions A, β, and δ with equation (8) as fit function.

$f(\theta)$	p_0	p_1	p_2
$A(\theta)$	1.49×10^{-5}	3.89×10^{-6}	1.018
$\beta(\theta)$	2.98	1.10	1.012
$\delta(\theta)$	16.6	4.99	1.015

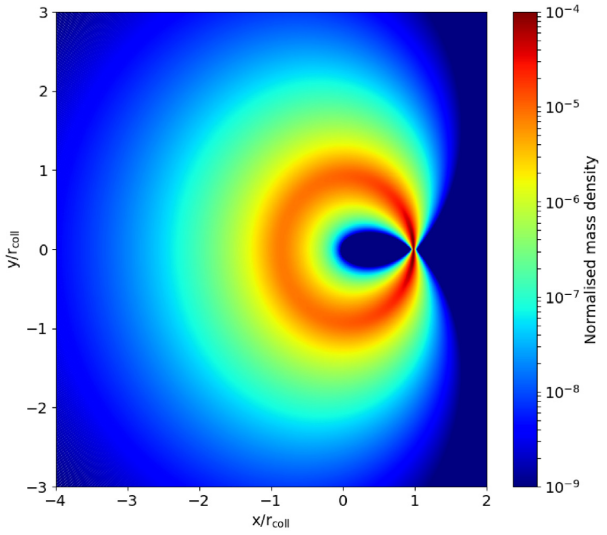


Figure 12. Model of the mass distribution of the post-impact material of the collision of Fig. 11 when using equation (7).

Fig. 12 shows the $x - y$ distribution obtained using equation (7). A comparison between this figure and Fig. 11 shows that the model has captured all essential elements of the distribution. For instance, that most of the post-impact material is closer to the central star than the collision point can clearly be seen in this figure. The latter can also be seen in Fig. 13 where the θ-dependence of the four functions A, β, δ, and r_0 are shown. The fact that r_0 oscillates between 1 (at the collision point $\theta = 0$) and 0.9 (close to the antipode $\theta = \pi$), indicates that the post-impact material is preferentially scattered into orbits closer to the central star. Moreover, the minimum of r_0 is directly related to the kick distribution displacement illustrated in Section 3.3.1. In fact, the predominance of kicks in the negative y-direction ($v_{\rm kick,\,y} = -0.036$) causes the argument of periastron for the fragments to centre around $\theta = \pi$, while the small displacement of kicks in the negative x direction ($v_{\rm kick,\,x} = -0.0037$) causes the argument of the periastron to slightly move clockwise inducing a phase of 0.013 in equation (9).

Lastly, we investigated how the spatial distribution of the fragments is altered when individual subsets of the PC catalogue are considered. In particular, we divided the collisions into subsets depending on the normalized impact velocity. We obtained the spatial distribution of fragments for these subsets of collisions using the same analysis performed for the overall spatial distribution (Section 3.3.3) and we fitted the results using equation (7). We obtained that no significant difference can be observed when the spatial distributions from the subsets are compared with the overall distribution in Fig. 11. However, we noticed that the model in

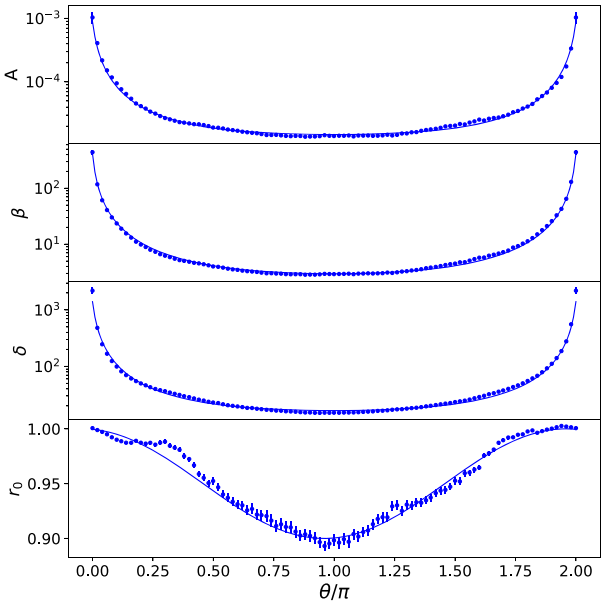


Figure 13. From top to bottom, graphs of A, β, δ, and r_0 in terms of θ. In all panels, $\theta = 0$ represent the position of the impact. The solid curves are the best fits corresponding to equation (8) for A, β, δ, and equation (9) for r_0.

equation (7) is extremely sensitive to individual fragments with high mass. In particular, the length scale r_0 in equation (9) tends to match the orbit of the biggest fragments instead of matching the actual maximum of the distribution along the radial direction. This issue is easily solved when enough collisions are considered, as in the case of the spatial distribution in Fig. 11, obtained by considering all the 1356 collisions in our PC catalogue. Possible future work could increase the size of the PC catalogue to better characterize the spatial distribution of fragments as a function of the collision parameters.

4 CONCLUSIONS

In this study, we present the results of a statistical analysis of the material generated through collisions between protoplanetary bodies during the last stage of the formation of terrestrial planets. Using a catalogue of 1365 impacts and results of the SPH simulations of 880 embryo–embryo collisions, we find that approximately 4.2 per cent of the total mass of the colliding embryos will be converted into debris. This value can increase up to 24 per cent (Burger et al. 2020) when the masses of the colliding bodies are smaller and a second giant planet is included.

One important result of our study is the negligible contribution of catastrophic collisions to the total mass of the collisional fragments. Our simulations show that, in general, the rate of the occurrence of these collisions is small. For that reason, even though they convert the entire masses of the colliding bodies into debris, they do not contribute much to the total mass of fragments. This result suggests that when simulating collisions to study the growth of planetesimals or the late stage of the formation of terrestrial planets, one can ignore catastrophic collisions to a first approximation.

In our simulations, we did not consider the distribution of water after each impact. However, volatile materials, such as water, are particularly prone to be lost during these events and to be redistributed across the system (Kegerreis et al. 2020). A sophisticated and robust methodology for including the production of collisional fragments is, therefore, paramount in simulations of planetary formation in general, and in particular in those that aim to track chemicals.

Our analysis indicated that for fragments in the range of $10^{-7}\ M_\oplus$ to $2 \times 10^{-2}\ M_\oplus$, the mass distribution can be approximated by an exponential function with an exponent of -2.21 ± 0.17. Assuming a constant bulk density in this range, this mass distribution translates into an exponential size distribution with an exponent of -3.63 ± 0.52. This value is consistent with the median value of 3.8 obtained by Leinhardt & Stewart (2012).

Our analysis of the spatial distribution of fragments indicated that the orbits of the large majority of post-impact bodies have eccentricities smaller than 0.4, implying that most of the collisional fragments have low velocities. We also found that many of these fragments ($\sim$ 64 per cent) are closer to the central star than the point of the impact. This inward scattering of post-impact material is observed for the first time and can be attributed to the loss of angular momentum and energy during an impact.

Finally, we developed an analytical formula for the distribution of fragments (equation 7). This equation proves useful in predicting and analysing results of giant impacts removing the need for carrying out SPH simulations of such events.

ACKNOWLEDGEMENTS

We are thankful to the referee, John Chambers, for his critically reading of our paper and his useful comments that improved our manuscript. This work was supported by the NYU Abu Dhabi Global Ph.D. Student Fellowship Program. TIM acknowledges the support from the Austrian Science Fund (FWF): P33351-N. CMS appreciates support by the DFG German Science Foundation projects 398488521, 446102036, 285676328. NH acknowledges support from NASA grant 80NSSC21K1050.

The following software facilitate the advancement of this work: MATPLOTLIB (Hunter 2007), NUMPY (Van Der Walt, Colbert & Varoquaux 2011).

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Barnes R., Quinn T. R., Lissauer J. J., Richardson D. C., 2009, Icarus, 203, 626
Benz W., Asphaug E., 1994, Icarus, 107, 98
Benz W., Slattery W. L., Cameron A. G. W., 1986, Icarus, 66, 515
Burger C., Maindl T. I., Schäfer C. M., 2018, Celest. Mech. Dyn. Astron., 130, 2
Burger C., Bazsó Á., Schäfer C. M., 2020, A&A, 634, A76
Carter P. J., Leinhardt Z. M., Elliott T., Walter M. J., Stewart S. T., 2015, ApJ, 813, 72
Chambers J. E., 1999, MNRAS, 304, 793
Chambers J. E., 2013, Icarus, 224, 43
Clement M. S., Kaib N. A., Raymond S. N., Chambers J. E., Walsh K. J., 2019, Icarus, 321, 778
Dormand J. R., Woolfson M. M., 1977, MNRAS, 180, 243
Dugaro A., de Elía G. C., Darriba L. A., 2020, A&A, 641, A139
Haghighipour N., Winter O. C., 2016, Celest. Mech. Dyn. Astron., 124, 235
Hunter J. D., 2007, Comput. Sci. Eng., 9, 90
Ida S., Makino J., 1993, Icarus, 106, 210
Jackson A. P., Wyatt M. C., Bonsor A., Veras D., 2014, MNRAS, 440, 3757

Kegerreis J. A., Eke V. R., Catling D. C., Massey R. J., Teodoro L. F. A., Zahnle K. J., 2020, ApJ, 901, L31
Kokubo E., Ida S., 2000, Icarus, 143, 15
Leinhardt Z. M., Stewart S. T., 2012, ApJ, 745, 79
Monaghan J., Pongracic H., 1985, Appl. Numer. Math., 1, 187
Morishima R., 2015, Icarus, 260, 368
Mustill A. J., Davies M. B., Johansen A., 2018, MNRAS, 478, 2896
O'Brien D. P., Morbidelli A., Levison H. F., 2006, Icarus, 184, 39
Poon S. T. S., Nelson R. P., Jacobson S. A., Morbidelli A., 2020, MNRAS, 491, 5595
Raymond S. N., Quinn T., Lunine J. I., 2004, Icarus, 168, 1
Raymond S. N., Quinn T., Lunine J. I., 2006, Icarus, 183, 265
Schäfer C. M. et al., 2020, Astron. Comput., 33, 100410
Schäfer C., Riecker S., Maindl T. I., Speith R., Scherrer S., Kley W., 2016, A&A, 590, A19
Scora J., Valencia D., Morbidelli A., Jacobson S., 2020, MNRAS, 493, 4910
Van Der Walt S., Colbert S. C., Varoquaux G., 2011, Comput. Sci. Eng., 13, 22
Watt L., Leinhardt Z., Su K., 2021, MNRAS, 502, 2984
Wetherill G. W., 1988, in Vilas F., Chapman C. R., Matthews M. S., eds, Accumulation of Mercury from Planetesimals. University of Arizona Press, Tucson, AZ, p. 670

This paper has been typeset from a TEX/LATEX file prepared by the author.

6. Residual neural networks for the prediction of planetary collision outcomes

We quickly realized that systematic analysis and knowledge extraction from our collision-catalog was a non-trivial task and required a more sophisticated approach. In the broader context of the recent advancements in deep learning (DL), we decided to significantly shift our focus towards this particular direction.

Using our SPH code, we were in a unique position for generating high-quality, ground truth collision data. We vastly extended and improved upon our existing dataset by simulating a total of 10164 pairwise planetary collisions, which is the largest collision dataset as of 2025.

We then developed and trained a DL model to predict collision outcomes directly from initial conditions. A major goal was to speed up predictions by several orders of magnitude, thereby enabling handling of collisions in a matter of milliseconds rather than minutes. Our model achieved state of the art results in terms of predictive performance while at the same time being more efficient than existing approaches.

In 2022, our work resulted in a conference contribution [131], as well as a journal publication, which is a major contribution in the context of this thesis [135].

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **520,** 1224–1242 (2023) https://doi.org/10.1093/mnras/stac2933
Advance Access publication 2022 October 14

Residual neural networks for the prediction of planetary collision outcomes

Philip M. Winter,[1★] Christoph Burger,[1★†] Sebastian Lehner,[2†] Johannes Kofler,[2] Thomas I. Maindl[3,4] and Christoph M. Schäfer[1]

[1]*Institut für Astronomie und Astrophysik, Eberhard Karls Universität Tübingen, Auf der Morgenstelle 10, D-72076 Tübingen, Germany*
[2]*Johannes Kepler University Linz, Altenberger Straße 69, A-4040 Linz, Austria*
[3]*SDB Science-driven Business Ltd, 85 Faneromenis Avenue, Ria Court 46, Suite 301, 6025 Larnaca, Cyprus*
[4]*Department of Astrophysics, University of Vienna, Türkenschanzstraße 17, A-1180 Vienna, Austria*

Accepted 2022 September 29. Received 2022 September 23; in original form 2022 April 21

ABSTRACT

Fast and accurate treatment of collisions in the context of modern *N*-body planet formation simulations remains a challenging task due to inherently complex collision processes. We aim to tackle this problem with machine learning (ML), in particular via residual neural networks. Our model is motivated by the underlying physical processes of the data-generating process and allows for flexible prediction of post-collision states. We demonstrate that our model outperforms commonly used collision handling methods such as perfect inelastic merging and feed-forward neural networks in both prediction accuracy and out-of-distribution generalization. Our model outperforms the current state of the art in 20/24 experiments. We provide a data set that consists of 10164 Smooth Particle Hydrodynamics (SPH) simulations of pairwise planetary collisions. The data set is specifically suited for ML research to improve computational aspects for collision treatment and for studying planetary collisions in general. We formulate the ML task as a multi-task regression problem, allowing simple, yet efficient training of ML models for collision treatment in an end-to-end manner. Our models can be easily integrated into existing *N*-body frameworks and can be used within our chosen parameter space of initial conditions, i.e. where similar-sized collisions during late-stage terrestrial planet formation typically occur.

Key words: hydrodynamics – methods: numerical – astronomical data bases: miscellaneous – celestial mechanics – planets and satellites: composition – planets and satellites: formation.

1 INTRODUCTION

1.1 Planet formation background

Planet formation is inherently connected to collisions on all scales, from μm-sized dust grains up to planet-sized bodies. The precise mechanisms of early planetary growth generally depend on the current conditions in the protoplanetary disc and the amount and (dominant) size of available building blocks (e.g. Kokubo & Ida 2002; McNeil, Duncan & Levison 2005; Johansen & Lambrechts 2017). Particularly for terrestrial planets, our current understanding suggests that their final phase of accretion comprises growth via pairwise collisions of up to planet-sized bodies, lasting on the order of tens to hundreds of Myr (e.g. Chambers & Wetherill 1998; Kokubo & Ida 1998; Agnor, Canup & Levison 1999; Chambers 2001; Kokubo, Kominami & Ida 2006). This is supported by the long accretion times of terrestrial planets in the Solar System, as well as features like Mercury's high bulk density, Earth's large moon, or Mars' hemispheric dichotomy, all believed to be the consequences of large-scale collisions of roughly similar-sized bodies. Indirect evidence for such encounters has also been found in extrasolar systems (e.g. Wyatt & Jackson 2016) in the form of observed infrared excess caused by warm dust, interpreted as collision debris. These large collision events are of particular interest as they shape the final characteristics of terrestrial planets, and likely contribute to the broad compositional diversity of observed low-mass exoplanets (Marcus et al. 2009, 2010; Inamdar & Schlichting 2016; Bonomo et al. 2019). This phase of planet formation naturally also leads to radial mixing of material and allows for (dynamical and collisional) transport of volatiles, such as water to the inner parts of the system, and especially to potential planets forming in the habitable zone (Morbidelli et al. 2000; Izidoro et al. 2013; O'Brien et al. 2014, 2018; Haghighipour & Winter 2016; Burger, Bazsó & Schäfer 2020b).

Modelling of this final phase of planet formation is typically based on *N*-body simulations, where mainly the gravitational interaction of hundreds to thousands of bodies is followed for up to few hundred Myr (e.g. Chambers 2013; Fischer & Ciesla 2014; O'Brien et al. 2014; Quintana & Lissauer 2014; Quintana et al. 2016, as some of the more recent work). As planet formation models become more sophisticated and aim to study more than the most basic outcome quantities, collision modelling has to keep up in order to avoid systematic errors caused by too crude approximations of the underlying physics.

★ E-mail: winter@murena.io (PMW); christoph.burger@uni-tuebingen.de (CB)
† Equal contribution.

1.2 The collision treatment problem

Accurate modelling of major collisions among large, up to planet-sized bodies plays an important role in understanding the formation, evolution, and diversity of planetary systems. The prediction task for two-body collisions is well-defined: Given the initial conditions, such as collision geometry and object properties, we ask for the outcome state at a specific later point in time.

Up to relatively recently, collisions in planet formation scenarios were typically modeled by assuming complete accretion in all encounters (e.g. Raymond, Quinn & Lunine 2004, 2007; Haghighipour & Raymond 2007; Izidoro et al. 2013; Fischer & Ciesla 2014; O'Brien et al. 2014; Quintana & Lissauer 2014), often referred to as perfect inelastic merging (PIM). This approach is simple and fast, but gives reasonably accurate predictions only for the lower end of the spectrum of characteristic collision velocities, or for large target-to-impactor mass ratios. In general, collisions between large and roughly similar-sized bodies can result in a diverse range of outcomes (e.g. Leinhardt & Stewart 2012), and often include significant material losses (Haghighipour & Maindl 2022). This can affect bulk and chemical composition (e.g. Carter et al. 2015; Dwyer, Nimmo & Chambers 2015; Carter et al. 2018), and even more so for volatile constituents, especially at or close to the surface (Marcus et al. 2010; Maindl et al. 2014, 2017; Burger, Maindl & Schäfer 2018; Kegerreis et al. 2020; Burger et al. 2020b). In addition, collisions among similar-sized bodies frequently result in two large and gravitationally unbound survivors, instead of a single dominant one, as exemplified in Fig. 1. These so-called hit-and-run events constitute up to half of all collision outcomes (e.g. Chambers 2013; Clement et al. 2019; Burger et al. 2020b). This can prolong planetary accretion considerably, naturally leading to a higher overall number of collisions, and resulting in very different behaviour in terms of material loss and transfer between colliding objects (Burger et al. 2018; Burger, Maindl & Schäfer 2020a; Burger et al. 2020b).

Several approaches have been developed to account for this diverse range of possible collision outcomes. Genda et al. (2017) developed scaling laws for collisional erosion with a focus towards smaller projectile-to-target mass ratios down to 1:10 000, where outcomes are generally dominated by a single large survivor. Zhou, Dvorak & Zhou (2021) propose an approach that also exclusively assumes a single survivor, but includes randomly picked material losses, based on statistics of a large number of Smooth Particle Hydrodynamic (SPH) collision simulations. Crespi et al. (2021) suggest an approach based on a catalogue of SPH collision outcomes, focusing on the distribution of smaller-scale collision fragments. A recent framework based on semi-analytical scaling laws (Leinhardt & Stewart 2012; Stewart & Leinhardt 2012; Leinhardt et al. 2015) has been applied in various planet formation studies (e.g. Chambers 2013; Bonsor et al. 2015; Carter et al. 2015; Quintana et al. 2016; Carter et al. 2018; Clement et al. 2019). Albeit fast and relatively straightforward to implement, its prediction accuracy for more complex behaviour, like the fate of surface volatiles, or individual material losses and transfer in hit-and-run, is naturally limited (Burger et al. 2018). Genda, Kokubo & Ida (2011), Genda et al. (2017), and Burger et al. (2020b) resolve collisions in *N*-body planet formation simulations by running dedicated SPH simulations for each event on the fly, which is the most accurate approach, but computationally complex and expensive.

To summarize, depending on the problem at hand and the available computational resources, one has to make design choices which method to use. Both, simple problems and/or sufficient computational resources allow the use of sophisticated collision treatment methods, whereas complex problems and/or limited resources require certain trade-offs between prediction accuracy and computation time. For many applications, it would be desirable to choose and adjust this trade-off more flexibly. Although analytic and heuristic approaches are efficient, they are typically neither very accurate, nor allow adjusting the accuracy-speed trade-off. In contrast, full hydrodynamic simulations for individual collisions are much more costly, but yet very accurate. In this paper, we aim to combine all three properties, yielding an efficient, still accurate and flexible approach, where flexible means that it can be easily adapted to different accuracy-speed trade-offs.

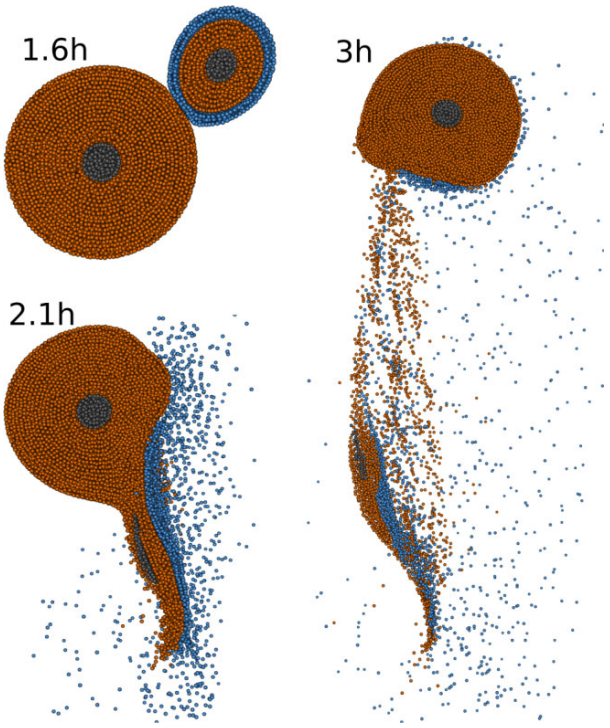


Figure 1. Exemplary snapshots at three different times of a SPH simulation of a planet–scale collision. The Mars-sized projectile hits the Earth-sized target at an impact angle of 43° and an impact velocity of 1.3 times the mutual escape velocity, resulting in a hit-and-run outcome. Colours indicate the different materials – an iron core, a silicate mantle, and a water/ice shell. Bodies are cut into halves for visualization. We perform 10 164 collision simulations, covering a large parameter space of initial conditions.

1.3 Machine learning for planetary collisions

The recent progress of cheap and efficient hardware caused a renaissance of machine learning (ML), enabling to solve complex tasks in different fields such as computer vision (Krizhevsky, Sutskever & Hinton 2017) and natural language processing (Brown et al. 2020) with unprecedented accuracy and speed. Recently, Tamayo et al. (2020) and Cranmer et al. (2021) applied ML for predicting long-term stability and dissolution of compact multiplanet systems, indicating that ML may serve as an efficient tool for fast and accurate approximation of astrodynamical processes.

Recurrent neural networks (RNNs; Jordan 1986; Pearlmutter 1989; Elman 1990) have been applied for approximating hydrodynamical simulations (Wiewel, Becher & Thürey 2019) and astrophysical simulations such as 2D mantle convection (Agarwal et al.

2021). Several works successfully demonstrated the applicability and usefulness of ML for planetary collision treatment, opening up a promising research direction for computational astrophysics: Valencia, Paracha & Jackson (2019) apply gradient boosting regression trees (Breiman et al. 1984; Friedman 2001), Gaussian processes (GPs; Rasmussen & Williams 2005), and a nested method for classifying collision scenarios and regressing the largest remnant mass. Cambioni et al. (2019) use a multiclass support-vector machine (Cortes & Vapnik 1995; Hearst et al. 1998) for the classification of different collision scenarios. They apply a small, three-layered feed-forward neural network (FFN; Rosenblatt 1961; Ivakhnenko & Lapa 1965) to regress accretion efficiencies, i.e. the mass of the largest remnant. Cambioni et al. (2021) extend this work and include surrogate models for predicting core mass-fractions of the largest and second-largest remnants. Emsenhuber et al. (2020) extend the work from Cambioni et al. (2019) to additionally predict orbital parameters of the two largest remnants with a separate regressor, resulting in a set of models that can be directly incorporated into *N*-body frameworks for collision treatment. However, this approach is limited to the main collision plane and does not allow prediction of orbital inclinations and longitudes of ascending nodes. The above-mentioned works use the SPH data from Reufer et al. (2012) that consists of collisions between non-rotating, differentiated iron-silicate bodies.

Timpe et al. (2020b) establish a high-quality data set that consists of 14 856 collisions between differentiated, rotating bodies (Timpe et al. 2020a). They apply a two-step classification-regression approach to predict post-collision properties. They study several different methods for collision treatment and find data-driven methods to outperform non-data driven methods. Gradient-boosted decision trees and FFNs are used for both classification and regression, whereas polynomial chaos expansion (Wiener 1938) and GPs are studied for regression only. They train different regressors for each individual post-impact property, and predict a variety of properties of the largest and second-largest remnant, and the remaining debris. FFNs and XGBoost (Chen & Guestrin 2016) perform best amongst data-driven methods. We regard that study as our closest related work.

The overall goal of our work is to improve the prediction of planetary collision outcomes via ML models. In particular, this includes minimizing systematic prediction errors as much as possible by outperforming the current state of the art. We improve upon the works above by providing a more general data set, reframing the ML task as a multi-task problem, and employing a simple, but problem-adapted ML model for the prediction of planetary collision outcomes. We train our model to predict masses, material fractions, positions, and velocities of the two largest post-collision remnants, and the remaining debris. Our contributions are summarized as follows:

(i) We perform extensive *N*-body simulations to determine realistic initial conditions for planetary collisions. We base the choice of the parameter space for our SPH data set on the outcome of the *N*-body simulations. To that end, we provide a comprehensive data set that consists of 10 164 SPH simulations of pairwise planetary collisions. We use between 20 and 50k SPH particles, which is relatively low resolution compared to state-of-the-art simulations in astrophysics with up to several million SPH particles. Our data set covers typical collision setups and is the first of its kind to combine all essential elements for a comprehensive treatment of collisions, including realistic object models (differentiated and rotating bodies), detailed pre- and post-collision geometries, and temporal information. The data set allows to study several generic topics, such as collision treatment in a broad range of scenarios, inverse problems (e.g. the Moon-forming impact), and collisional accretion during planet formation. While our data set is in general comparable to the one provided by Timpe et al. (2020b), it additionally includes volatile (water) layers, which opens up studies regarding collisional water/volatile transfer and loss, even though this is intended rather as a proof of concept in this work, mainly because of the difficulty to accurately resolve such surface layers.

(ii) In contrast to existing work, we follow a multi-task learning approach in the sense of multidimensional regression, in which a single ML model learns to predict the entire post-collision state rather than only specific, individual aspects of the state. Our ML task generalizes the collision treatment problem to 3D space, while at the same time avoiding the need for manual definition of class boundaries for different collision scenarios. Existing approaches often formulate the task as a classification problem, requiring somewhat arbitrary class definitions. We demonstrate that our multi-task learning approach leads to simple and computationally efficient models, while remaining relatively accurate compared to single-task learning.

(iii) We propose an ML model that helps modelling of temporal dynamics by evolving system states in an autoregressive manner. This closely resembles the data generation process, i.e. classical numerical solvers that iteratively solve the underlying hydrodynamic equations. This includes handling both, the properties of the colliding bodies and the spatio-temporal evolution of the system. Our model allows for flexible prediction of post-collision states at different times, and can be employed for collision treatment within existing *N*-body frameworks. We demonstrate superior prediction accuracy in comparison to commonly used baseline methods and the current state of the art. Moreover, our model requires little computational costs, reducing the prediction speed by approximately four orders of magnitude compared to the SPH simulations.

With our work, we aim to provide high-quality data and an ML model that is useful for various downstream applications. The paper is organized as follows. In Section 2, we describe our data generation pipeline, as well as the ML model used for collision treatment. In Section 3, we present our experiments and their results. Section 4 summarizes and concludes the paper.

2 METHODS

2.1 Data generation

2.1.1 N*-body simulations*

Burger et al. (2020b) developed a hybrid framework, based on extensive *N*-body simulations in combination with realistic collision treatment by direct SPH simulations. These results and collision statistics are also used to inform the choice of initial conditions for the SPH simulations performed in this study. In addition, we provide a cleaned and extended[1] version of their data set of approximately 10k collisions, which we refer to as '*N*-body data set'.[2]

[1]Based on new (yet unpublished) *N*-body + SPH simulations in a similar dynamical environment.

[2]The '*N*-body data set' based on the simulations by Burger et al. (2020b) provides data on collision parameters before contact, and basic data on the final state after the collision, like masses and composition of the two largest remnants, but no dynamical information (positions and velocities) and no data on intermediate states. Along with our other data and tools, it is available at https://github.com/littleblacksheep/csv/tree/main/misc.

The scenarios in Burger et al. (2020b) are based on an evolving disc of (∼Mars-mass) planetary embryos + smaller bodies (planetesimals). Their dynamical and collisional evolution is followed over several hundred Myr of terrestrial planet formation in an environment akin to the early Solar System. The embryos and planetesimals are modelled as differentiated, three-layered, self-gravitating bodies, similar to the SPH simulations in this work. The rotation state is not tracked across multiple collisions. The approach of on-the-fly SPH simulations allows not only for accurate treatment of each individual collision, but also a relatively straightforward re-integration of collision outcomes into the overall *N*-body dynamics (for our ML approaches this is discussed in Section B3). It also includes individual tracking of both large survivors in hit-and-run collisions, which comprise up to 50 per cent of outcomes between similar-sized bodies. Therefore, this data set also provides reliable collision (input parameter) statistics for the scenarios in this work.

2.1.2 *SPH simulations*

SPH is a numerical method for modelling visco-elastic fluid flows. The method was first proposed by Gingold & Monaghan (1977) and Lucy (1977) and has since been applied extensively to model various aspects of astrophysical collision processes, including planetary collisions. In this work, we use the SPH code `miluphcuda`[3] (Schäfer et al. 2016, 2020) to generate a planetary collision data set. An example is illustrated in Fig. 1. The SPH code solves the continuum mechanics equations for hydrodynamic flow, can handle 3D, multimaterial problems, and includes self-gravity. It also includes modules for the simulation of elasto-plastic solid-body physics based on several available material models and equations of state.

In this work, we perform pure hydro simulations, i.e. only solving the Euler equation with scalar pressure, instead of full tensorial treatment of material strength. Since we perform a large number of simulations, we trade some physical accuracy for numerical stability and more data (due to faster computation). However, this design choice is still a reasonably good proxy within the scope of our scenarios (Burger & Schäfer 2017; Burger et al. 2018). For actual collisions in an active planet formation environment, it can be assumed that the physical state of the colliding bodies – and hence their material (strength) response – varies over a broad range, even for otherwise identical scenarios in terms of masses, compositions, and collision parameters. This may be a function of their collision history, thermal state, and possibly various other factors. Considering those ambiguities, our rather simple material model allows the data set to remain as general as possible and at the same time consistent over our whole parameter space. We use the Tillotson equation of state (Tillotson 1962; Melosh 1989) for all simulations. Technical details are given in Section A1.

The SPH simulation pipeline is fully automated and includes all steps to initialize, run, and post-process individual simulations (see Fig. 2). For each run, a specific parameter set is sampled from the parameter space (Table 1). The chosen parameters cover a broad range of possible collision scenarios during terrestrial planet formation. The particular choices of parameter ranges are additionally informed by the robust statistics of our *N*-body data set (see Section 2.1.1). Note that we use the *N*-body data set exclusively for choosing meaningful parameter intervals representative of late-stage terrestrial planet formation. For creating the SPH data set, our parameter space of initial conditions is sampled randomly within the chosen intervals.

[3]The SPH code `miluphcuda` is in active development and publicly available at https://github.com/christophmschaefer/miluphcuda

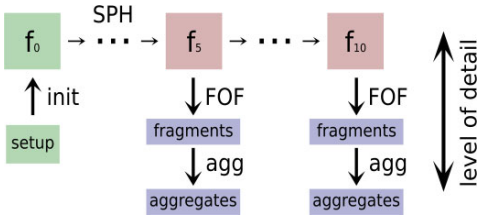


Figure 2. Data generation pipeline. For each individual simulation, a setup is sampled from the parameter space. The SPH particle distribution and its properties are set up in 'init', resulting in the input frame f_0. The SPH code then evolves the system, leading to a number of output frames. All output frames are post-processed with a friends-of-friends algorithm to compute all spatially connected material 'fragments'. Finally, 'aggregates' are identified, defined as gravitationally bound collections of fragments.

Table 1. Parameter space of initial conditions for our SPH simulations, covering a wide range of typical scenarios for rocky planet formation. See the text for detailed definitions. All parameters are randomly sampled.

Parameter	Min	Max	Description
$M_{\rm tot}(kg)$	$2 \times M_{\rm Ceres}$	$2 \times M_{\rm Earth}$	Total mass
$\gamma(1)$	0.05	1	Mass ratio m_p/m_t
$\zeta_{\rm iron}(1)$	0.01	0.25	Iron (core) fraction
$\zeta_{\rm water}(1)$	0; 0.1	0.25	Water (shell) fraction
$v_{\rm imp}(v_{\rm esc})$	1	8	Impact velocity
α(deg)	0	90	Impact angle
$P_{\rm rot}(P_{\rm rot,crit})$	0	0.2	Rotation period
$\theta_{\rm rot}$(deg)	0	180	Rotation axis polar
$\phi_{\rm rot}$(deg)	0	360	Rotation axis azimuthal
$f_i(1)$	3	7	Initial distance factor
$f_t(1)$	40	60	Simulation time factor
$N_{\rm tot}(1)$	20k	50k	Number of SPH particles

For initializing self-gravitating bodies in hydrostatic equilibrium, we adopt the approaches and tools from Burger et al. (2018), who calculate realistic density and pressure profiles for multilayered bodies. The colliding objects are referred to as *projectile* and *target*, the latter being the more massive body. They are initialized at a certain distance, on the order of several times the sum of their radii, to allow for pre-collision tidal deformation, relaxation of rotating configurations, and settling of residual numerical artefacts (e.g. at material boundaries). Based on the desired impact velocity and impact angle at 'touching-ball' distance (cf. Fig. 3), initial positions are calculated via backtracking the analytical two-body trajectories up to a distance of $d_{\rm initial} = f_i \times (R_{\rm t} + R_{\rm p})$. $R_{\rm t}$ and $R_{\rm p}$ are the target and projectile radii, and the initial distance factor f_i is a parameter. The total simulation time is calculated via $T_{\rm sim} = \tau_{\rm col} \times (f_i + f_t)$ and rounded up to the next full hour. $\tau_{\rm col}$ is the collision time-scale $\tau_{\rm col} = (R_{\rm t} + R_{\rm p})/v_{\rm imp}$. The impact velocity $v_{\rm imp}$ and the impact angle α are specified at touching-ball distance $R_{\rm t} + R_{\rm p}$, where $\alpha = 0°$ corresponds to head-on collisions and $v_{\rm imp}$ is the absolute value of the relative velocity vector v at touching-ball distance (cf. Fig. 3). The minimum number of SPH particles is set such that the resulting water shell has a thickness of at least 2 SPH particles at $\zeta_{\rm water} = 0.1$ (Burger 2019). Note that this resolution may be too low to accurately simulate the water layers' response for a range of scenarios

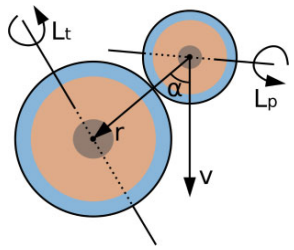


Figure 3. Collision geometry for planetary collisions. The impact angle α is measured between the relative position $\mathbf{r}$ and the relative velocity $\mathbf{v}$ between target and projectile at 'touching-ball' distance. Both objects comprise a core-mantle-shell structure and have random rotation axes $\mathbf{L}_t$ and $\mathbf{L}_p$, which can lie outside the plane spanned by $\mathbf{r}$ and $\mathbf{v}$.

(parameter combinations). This can be particularly problematic for the second-largest post-collision remnant, while reliable predictions are possible for the largest remnant already at resolutions similar to ours, as demonstrated by Burger et al. (2018). Nevertheless, results for water mass fractions on post-collision remnants should be taken with a grain of salt, and consequently we consider our ML model predictions for this aspect rather a proof of concept and not generally accurate at this point. For other basic outcome properties, on the other hand, like masses and kinematics of the two largest remnants, Burger et al. (2018) found resolution convergence for similar collision scenarios within 10 per cent for their 100k particles simulations. Our simulations contain either 2 or 3 materials, depending on ζ_{water}, where we remove the water shell if $\zeta_{\text{water}} < 0.1$ was sampled. The total colliding mass covers a range from $2 \times M_{\text{Ceres}}$ up to $2 \times M_{\text{Earth}}$. The mantle (basalt) mass-fraction is defined by $\zeta_{\text{basalt}} = 1 - \zeta_{\text{iron}} - \zeta_{\text{water}}$. Since the hydrostatic initialization routine is based on non-rotating objects, we set our maximum rotation period $P_{\text{rot,max}} = 0.2 \times P_{\text{rot,crit}}$ for both target and projectile in order to avoid excessive initial oscillations and instabilities, which typically occur once P_{rot} approaches $P_{\text{rot,crit}}$. The critical rotation period $P_{\text{rot,crit}}$ is defined such that material at the surface of the (idealized spherical) body is weightless according to Kepler's third law. Rotation axes are chosen randomly for both target and projectile. We refer to Section A2 for more details.

During simulation, the SPH code periodically produces output frames, which contain the state of all SPH particles at the respective time. We keep the first, the last, and intermediate frames for post-processing, where intermediate frames are saved at 5-h intervals (simulated time). All frames undergo the same post-processing procedure:

(i) Spatially connected collision fragments are calculated by the friends-of-friends algorithm (Geller & Huchra 1983).

(ii) Barycentres, orbital angular momentum, and spin angular momentum are calculated for each fragment, as well as for the entire system.

(iii) The two[4] largest aggregates of fragments are calculated. An aggregate is defined as a collection of gravitationally bound fragments, determined by an iterative procedure, which starts from the most-massive fragment as seed (see Burger et al. 2020b, for details). In the remainder of the paper, these aggregates are referred to as 'remnants' for clarity.

(iv) Basic visualization is done for the large fragments. A fragment is considered significant if it consists of at least 5 SPH particles.

(v) In this work, we focus on the prediction of macroscopic system states, requiring information on the level of remnants only. Moreover, we aim to keep memory requirements of the final data set low. Therefore, SPH output frames are subsampled, keeping 1 out of 10 SPH particles.

Keeping intermediate frames enables in-depth studies of temporal properties of the collision process. Moreover, they allow for the development of sophisticated ML models, i.e. models that not only predict the final state of the system, but the entire temporal evolution in detail. Note that since we sample our parameter space randomly, inputs to ML models do not require initial conditions that are similar to those in Burger et al. (2020b).

2.2 Machine Learning for collision treatment

From an ML perspective, the collision treatment task requires learning physical laws (e.g. conservation laws, material deformations, gravitational interactions, etc.) and handling the temporal evolution of the system (e.g. via time-series modelling). Various ML approaches can be applied in different contexts, mostly depending on which level of detail one is interested in. Therefore, we design our SPH data set such that it can be used at different levels of detail. For example, one can use remnant or fragment information ('macro states') rather than SPH particle representations ('micro states') for learning certain aspects (e.g. predicting certain quantities such as the mass of the largest remnant or the thermal energy of the system). Depending on which level of detail ML is applied to, different aspects may be able to be learned more or less efficiently. In this work, we focus on macro states because this setup is the most relevant one in order to incorporate ML models into *N*-body simulations for planet formation and evolution (see Table 2. and Section B3 for more details). In contrast, ML models operating on micro states may be a better choice if one is interested in studying details of the hydrodynamic flow and physical interactions in simulations such as SPH.

2.2.1 Collision treatment as a multi-task regression problem

Supervised learning is the task of selecting (learning) a specific model from a certain model class by using example input-target pairs. The difference between model outputs and desired target outputs results in an error, which is used to improve a model. We train our ML models in a supervised manner to predict several different quantities (mass, material fractions, position, and velocities) at once, which turns the problem into a multi-task problem. Our multi-task problem can be interpreted as a multidimensional regression problem of different physical quantities, since we use shared representations to predict different modalities. We motivate formulating and solving the problem as a multi-task problem due to inherent dependencies and correlations between the individual subtasks (e.g. trajectories of individual fragments highly depend on the overall mass distribution). Since all of our sub-tasks are highly correlated with each other, we hope that the multi-task setting supports generalization due to shared representations within ML models, acting as a form of regularization. Shared representations naturally allow for exploiting dependencies and correlations between different tasks, potentially improving the ML model's predictive performance. Note that in

[4]Gravity-dominated collisions of roughly similar-sized bodies generally result in either none (if highly destructive), one, or two (in hit-and-run scenarios) large surviving bodies, along with orders-of-magnitude smaller debris.

Table 2. Non-redundant ML features and normalization hyperparameters for feature normalization. Units indicate different physical quantities. All data x_{phys} is normalized during pre-processing. Note that since material fractions sum up to 1, only core (iron) and shell (water) fractions are required. Initial rotation speeds of the colliding bodies are encoded via the norms of their respective rotation axes.

State	Feature	Dim	Description	σ (ours)	σ (Timpe)
Initial	$N_{\text{tot}}(1)$	1	Number of SPH particles	5e + 4	2.3e + 5
	$M_{\text{tot}}(kg)$	1	Total mass	1e + 25	100
	$\gamma(1.)$	1	Mass ratio m_p/m_t	1	1
	$\zeta_p(1.)$	2	Material fractions projectile	1	1
	$\zeta_t(1.)$	2	Material fractions target	1	1
	$rot_p(\text{rad s}^{-1})$	3	Rotation axis projectile	6.5e-05	100
	$rot_t(\text{rad s}^{-1})$	3	Rotation axis target	6.5e-05	100
	$x_p(\text{m})$	3	Barycentre position projectile	[5e+07, 2e+08, 2e + 07]	[6, 41, 3]
	$v_p(\text{m s}^{-1})$	3	Barycentre velocity projectile	[2e+03, 1e+04, 6e + 02]	[2, 16, 0.5]
	$x_t(\text{m})$	3	Barycentre position target	[5e+07, 2e+08, 2e + 07]	[6, 41, 3]
	$v_t(\text{m s}^{-1})$	3	Barycentre velocity target	[2e+03, 1e+04, 6e + 02]	[2, 16, 0.5]
final	$m_1(\text{kg})$	1	Mass largest remnant	1e + 25	100
	$m_2(\text{kg})$	1	Mass 2nd-largest remnant	1e + 25	100
	$m_r(\text{kg})$	1	Mass rest	1e + 25	100
	$\zeta_1(1.)$	2	Material fractions largest remnant	1	1
	$\zeta_2(1.)$	2	Material fractions 2nd-largest remnant	1	1
	$\zeta_r(1.)$	2	Material fractions rest	1	1
	$x_1(\text{m})$	3	Barycentre position largest remnant	[5e+07, 2e+08, 2e + 07]	[6, 41, 3]
	$v_1(\text{m s}^{-1})$	3	Barycentre velocity largest remnant	[2e+03, 1e+04, 6e + 02]	[2, 16, 0.5]
	$x_2(\text{m})$	3	Barycentre position 2nd-largest remnant	[5e+07, 2e+08, 2e + 07]	[6, 41, 3]
	$v_2(\text{m s}^{-1})$	3	Barycentre velocity 2nd-largest remnant	[2e+03, 1e+04, 6e + 02]	[2, 16, 0.5]
	$x_r(\text{m})$	3	Barycentre position rest	[5e+07, 2e+08, 2e + 07]	[6, 41, 3]
	$v_r(\text{m s}^{-1})$	3	Barycentre velocity rest	[2e+03, 1e+04, 6e + 02]	[2, 16, 0.5]

contrast to our multi-task setting, Cambioni et al. (2019, 2021), Emsenhuber et al. (2020), and Timpe et al. (2020b) use individual ML models for predicting either single or subsets of collision outcome quantities, i.e. following single-task approaches. We believe that single-task approaches introduce unnecessary restrictions to the generalization capabilities of ML models, because individual models are exclusively able to specialize in their respective regime. Also, using individual regression-models for individual outcome scenarios (i.e. erosion, accretion, and hit-and-run) may lead to various issues caused by data-scarcity due to class-imbalances, which are common in planetary collision data sets. Moreover, using a single ML model for solving several subtasks at once may allow much better accuracy-speed trade-offs, especially in the presence of many subtasks. In our work, the importance of the individual subtasks are implicitly given via data pre-processing, effectively weighting loss terms of the respective subtasks. However, the importance can be explicitly adjusted depending on specific use cases.

To our knowledge, this is the first work to fully formulate the ML task as a regression problem. Our formulation allows simple, yet efficient training in an end-to-end manner and facilitates easy integration of ML models into existing *N*-body frameworks. Our regression objective does not explicitly optimize for classification performance, but rather for regression of macroscopic properties of the system. At the same time, our objective avoids the need for complicated approaches, which require two separate ML models for regression and classification, respectively.

We believe that it is not beneficial to train classifiers that explicitly discriminate between different outcome scenarios such as accretion or hit-and-run, because such classification can be easily performed as a post-processing step on top of regression-model predictions. We favour performing classification via a post-processing step rather than training dedicated classifiers because the former can be used in combination with variable class definitions, whereas the latter is bound to fixed class definitions. When training dedicated classifiers, changing class definitions would require re-training the classifiers, which can be quite cumbersome in practice. Moreover, pure classifiers cannot be used as a full replacement for collision treatment in *N*-body simulations.

In general, we believe that defining and learning fixed classification schemes is not optimal due to continuous transitions between classes and the associated arbitrariness of class definitions. We believe that training dedicated classifiers is only reasonable if one is explicitly interested in accurate classification under the restriction of fixed class definitions.

The integration of ML models for collision treatment into *N*-body simulations might require additional post-processing steps on top of ML predictions. This includes restricting predictions to conserve certain quantities such as the total mass, e.g. by re-scaling predicted masses and/or distributing debris material across the two largest remnants. For the actual application in *N*-body simulations, it is especially important how (if at all) the remaining collision debris is treated, which typically consists mostly of physically non-connected and gravitationally unbound fragments. This naturally opens up many possibilities depending on the respective use case (i.e. the precise physical and numerical model). In our experiments, we do not apply any additional post-processing steps in order to remain as general as possible and to obtain conservative performance estimates.

Fragments that are formed from collisions between non-rotating objects mostly remain in the collision's main symmetry plane (the x–y plane in our case) with only marginal z-components. However, collisions between rotating objects generally break this symmetry, and may produce large fragments with significant z-components. This symmetry breaking is also confirmed by the data from Timpe et al. (2020b). We thus generalize the prediction task from Emsenhuber et al. (2020) to 3D space, treating all dimensions equally to consistently handle deviations from the main collision plane.

2.2.2 Autoregressive ML models for temporal evolution

The use of autoregressive ML models for predicting collision outcomes can be motivated by studying the data generation process, i.e. the SPH simulations. We know that the data generation process has the Markov property, i.e. states s_{t+h} at a time $t + h$ depend entirely on their previous states s_t at time t. We assume continuous transitions between states for infinitesimal stepsizes h. The transition from s_t to s_{t+h} is described by a transition function g.

$$s_{t+h} = g(s_t, h) \tag{1}$$

Historically, g refers to a set of hand-crafted equations that incorporate certain physical laws (e.g. gravity, friction, etc.), as well as a procedure to evolve the system in time (e.g. numerical integration) by means of differential equations. In practice, these approaches often suffer from limitations, such as the requirement to use small stepsizes when using classical solvers. Too large stepsizes typically introduce large systematic errors, often leading to diverging or unstable solutions.

In this work, we aim to approximate solutions obtained using hand-crafted transition functions via an ML model that is learned from data. We believe that ML models are – once trained – efficient, powerful, and flexible transition functions for modelling the underlying physical processes in planetary collisions over time. In contrast to FFNs, our proposed model class exploits the Markov property of the data generation process, i.e. taking multiple, autoregressive steps to predict system states at a desired time T. There are several arguments that support the use of autoregressive ML models:

(i) Neural networks are universal function approximators (Hornik, Stinchcombe & White 1989) that allow learning highly complex functions. This property allows direct prediction of system states at various times T, entirely circumventing the need for time-series modelling. ML models should thus be much more computationally efficient compared to numerical simulations. The most extreme case would be to predict the final state directly, as typically achieved in the literature. Depending on the choice for the stepsize, our model allows a flexible accuracy-speed trade-off. Small stepsizes can be expected to better model the physical processes and lead to more accurate predictions at the cost of computational resources, whereas large stepsizes lead to less accurate, but faster predictions.

(ii) Autoregressive models subdivide the prediction of system states by taking multiple iterative steps. Since the universal function approximation theorem also applies to autoregressive ML models, they can use magnitudes larger, more complex time-steps compared to classical transition functions (numerical solvers) before getting unstable. This property typically makes these ML models much more efficient in terms of computational costs compared to hand-crafted transition functions.

(iii) Learned transition functions allow context-dependent time-steps, i.e. adjusting the transition function automatically, based on data-specific information. This property avoids algorithmic design decisions, making ML-based transition functions more general and flexible compared to hand-crafted transition functions.

(iv) Using autoregressive ML models allows for improved interpretability by enabling analysis of intermediate states. Such an analysis is not possible for ML models like FFNs or regression trees, which typically try to predict final post-collision states directly.

(v) Due to their design, we believe that autoregressive ML models can achieve better generalization compared to methods that try to predict final states directly, allowing more accurate predictions and improved o.o.d. generalization capabilities. Autoregressive ML models that learn fixed time intervals (i.e. taking multiple steps with the same stepsize) are effectively time-invariant per design, in the sense that they have to learn physical processes at only a single timescale. Thereby, the models are not forced to spend their parameters[5] for learning to become time-invariant. This property can potentially also lead to improved parameter efficiency compared to non-time-invariant ML approaches.

(vi) Longer physical interactions typically lead to the emergence of more complex dynamical processes during planetary collision events. Using autoregressive models naturally accounts for these effects by allocating computational resources that linearly scale with time, which is consistent with fluid-flow approaches.

Gated architectures (Hochreiter & Schmidhuber 1997; Cho et al. 2014) and regularized RNNs (Schmidt et al. 2021) are able to produce chaotic dynamics, but often suffer from the exploding gradient problem (Metz et al. 2021), typically leading to diverging sequences (Monfared, Mikhaeil & Durstewitz 2021). On the other hand, non-chaotic sequences have bounded loss gradients and converge to fixed points. Thus, training autoregressive ML models is typically non-trivial and often very sensitive to hyperparameters, especially when the data-generating process is itself chaotic. Exploding gradients and diverging sequences in LSTMs[6] can be mitigated via the forget gate (Gers, Schmidhuber & Cummins 1999), thereby reintroducing the vanishing gradient problem (Hochreiter 1998). The vanishing gradient problem can prohibit efficient training of deep neural networks. For our autoregressive ML model, we find regularization of hidden states to be a robust strategy against diverging sequences. Moreover, we find that gradient descend with backpropagation through time (Robinson & Fallside 1987; Werbos 1988; Mozer 1995) works fine for training.

2.2.3 Residual neural network for planetary collision handling

Our proposed model for prediction of collision outcomes can be interpreted as a residual neural network (ResNet; He et al. 2016). ResNets were originally introduced to ease the training of deep neural networks. The 'residual' aspect refers to reformulating neural network layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. Comparable to LSTMs without forget gate, ResNets efficiently mitigate the vanishing gradient problem.

We refer to our architecture as RES herein. In contrast to a classical ResNets, learned parameters in our model are shared across different steps (see Fig. 4). Our architecture treats temporal dynamics consistently by evolving system states in an autoregressive manner. Individual steps of a trained model can be interpreted as evolving the system for a fixed, but learned time interval. This approach is comparable to explicit iterative methods such as the Euler method or the Runge–Kutta method (Runge 1895; Kutta 1901). Ideally, smaller stepsizes should allow for better modelling of physical processes and should thus lead to better performance at the cost of computational resources. Our architecture allows for flexible prediction of system states at different times T by taking the respective number of update steps. Our architecture is autoregressive, i.e. only requiring the initial state y^0 and the number of steps $n_{\text{steps}} = s_h \times T$ as input. The

[5]Note that herein term 'parameters' can either refer to individual parameters of initial conditions for our SPH simulations, or to learnable parameters of an ML model. Both cases should be apparent from the respective text passages.
[6]Long Short-Term Memory, a special kind of RNN that is widely used for processing sequential data such as written texts, time series, or DNA sequences.

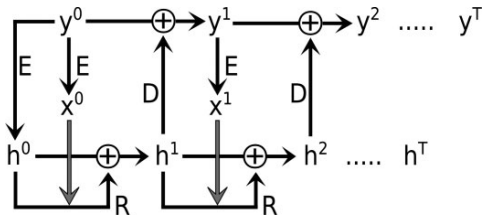


Figure 4. Operations of our weight-tied residual neural network architecture. The neural network modules E, R, and D are learned from data and shared across individual steps. The initial state y^0 is encoded into the initial hidden state h^0 and the input x^0 of R at time $t = 0$. R then predicts additive updates to h^t by using h^t and x^t. At each step, D outputs relative updates, which are used to evolve system states y via Euler integration. Sequences y^t and h^t are calculated iteratively, where the number of steps T correlate linearly to simulation time.

hyperparameter s_h allows to take different accuracy-speed trade-offs by adjusting the temporal resolution (i.e. the stepsize).

We want to stress that predicted sequences y^t and h^t [with $t \in (0, n_{\text{steps}})$] should not be considered as time sequences per se, but may nevertheless be closely related/correlated to time sequences, especially when considering our task of predicting system states at different points in time. We incorporate this close temporal correlation by choosing the number of steps to correlate linearly to the simulated time of the respective SPH simulations. This strong assumption may require additional and more detailed consideration in future work.

Our entire model architecture can be formalized as follows:

$$(h^t, x^t) = E(y^t, \phi_E) \tag{2}$$

$$h^t = h^{t-1} + R(h^{t-1}, x^{t-1}, \phi_R) \tag{3}$$

$$y^t = y^{t-1} + D(h^t, \phi_D) \tag{4}$$

E, R, and D correspond to the encoder module, residual module, and decoder module, respectively. Hidden states that are predicted by E are only used for the initial hidden state h^0. Our architecture avoids the vanishing gradient problem via additive updates to hidden states h and physical states y. Additive updates to the physical states y can be interpreted as Euler discretization of a continuous transformation and is closely related to the works of Chen et al. (2018), He et al. (2016), and Srivastava, Greff & Schmidhuber (2015). E, R, and D are FFN networks[7] (Rosenblatt 1961; Ivakhnenko & Lapa 1965) with learnable parameters ϕ_E, ϕ_R, and ϕ_D.

2.2.4 *Baseline models*

We choose three baseline methods for comparison with our newly proposed RES model class. Existing work (Cambioni et al. 2019, 2021; Emsenhuber et al. 2020; Timpe et al. 2020b) use FFNs as regressors for collision outcomes. We thus choose an FFN as our first baseline. We choose a linear regression model (LIN) as our second baseline to study the benefit of deep learning models compared to a simple, data-driven model. The third baseline is PIM, which is still widely used in astrophysical problems involving collisions because the method is purely analytic and fast. PIM assumes a perfect inelastic collision of target and projectile, always leading to a single surviving body, and conserving mass and momentum of the system by design.

[7]Historically also referred to as multilayer perceptrons

To enable learning non-linear mappings, artificial neural networks require so-called activation functions, which are applied element-wise to individual neurons usually after calculating the matrix-vector product for the respective layers. Due to its sound theoretical advantage compared to other activation functions, we use the scaled exponential linear unit (SELU) activation function (Klambauer et al. 2017) for hidden layers and linear activation functions for output layers of our deep learning models. SELU activations have self-normalizing properties, where neuron activations automatically converge towards zero mean and unit variance in the case of many hidden layers, leading to substantial advantages for training, regularization, and robustness when compared to other approaches. An optional rotation module can be incorporated into the ML models as additional pre- and post-processing steps, rendering the models rotation-equivariant (see Section B1 for details).

Although our SPH results naturally contain approximations and assumptions about the simulated physical processes, and are also subject to typical numerical inaccuracies, we define the SPH data as our ground truth. This definition is generally motivated by the fact that hydrodynamical simulations are currently considered the most accurate method for planetary collision treatment.

2.2.5 *ML experiment setup*

We split our data into a development set and a test set. The development set includes approximately 88 per cent of the data (8927 data points) and consists of training and validation splits, whereas the test set covers the remaining 12 per cent (1237 data points). The entire data set contains 10 164 data points. Using the development set, we perform fivefold crossvalidation[8] (Hastie, Tibshirani & Friedman 2017) for all experiments, allowing to calculate confidence intervals for our results. All training and validation splits share the same data distribution. Note that validation data are inappropriate for estimating performance on future data because validation data are used for hyperparameter optimization, which can be a source of information leakage. A holdout test set is required to estimate performance on completely new, unseen data.

Let us recap that our data set covers the parameter space as defined in Table 1. Although the parameter space is carefully chosen, parameters of real collisions are naturally not strictly limited to our defined parameter ranges (Quionero-Candela et al. 2009), i.e. so-called o.o.d. data points. In practice, ML models often fail to generalize to such o.o.d. data points. In order to study o.o.d. generalization of our ML models, we establish an o.o.d. test set. We expect that problem-specific models have better o.o.d. generalization capabilities compared to general-purpose models (Mitchell 1980). It is widely known that the impact velocity and the impact angle are two of the most important parameters in the context of planetary collisions. Thus, we manually select four regions in the impact angle – impact velocity space that compose our o.o.d. test set (see Table 3). We use this o.o.d. test set as our default test set in experiments unless stated otherwise.

Our data set D consists of $N = 10\,164$ tuples (y_i^0, z_i^T), $i \in [1, N]$, representing initial system states (at $t = 0$) and final system states (at $t = T$). For our supervised learning task, y_i^0 and T are used as model inputs, whereas z_i^T are used as ground truth labels. For intermediate

[8]Folds are non-intersecting, same-sized subsets of a data set. For five-fold crossvalidation, a total of five models are trained independently. Each training session consists of training on four-folds, whereas the remaining fifth fold is used for validation.

Table 3. Selected regions for the out-of-distribution (o.o.d.) test set. We select regions that may result in qualitatively different outcomes compared to the development set. The test set contains about 12 per cent of all data points.

Parameter	Region 1	Region 2	Region 3	Region 4
$\alpha_{\rm min}$(deg)	10	65	80	0
$\alpha_{\rm max}$(deg)	30	75	90	20
$v_{\rm imp,min}(v_{\rm esc})$	1.5	2	1	6
$v_{\rm imp,max}(v_{\rm esc})$	2.5	4	2	8

states $0 < t < T$ holds. D is split into a development set $D_{\rm dev}$ and a test set $D_{\rm test}$. Our training and validation splits are derived from $D_{\rm dev}$ depending on the respective crossvalidation fold.

$$D = \left\{ \left(y_1^0, z_1^T\right), \left(y_2^0, z_2^T\right), ..., \left(y_N^0, z_N^T\right) \right\} \tag{5}$$

$$D_{\rm dev} \subset D, \quad D_{\rm test} \subset D, \quad D_{\rm dev} \cap D_{\rm test} = \{\} \tag{6}$$

We perform data pre-processing to transform features into appropriate value ranges for ML. We apply feature-wise normalization $x_{\rm ML} = \frac{(x_{\rm phys} - \mu)}{\sigma}$ to transform data $x_{\rm phys}$ given in SI units into data $x_{\rm ML}$, whose value ranges are better suited for ML. We set $\mu = 0$ for all features. Note that the barycentre of the system still remains at the origin of the coordinate system for all data points even after normalization. Table 2. summarizes our ML features, along with normalization hyperparameters σ. Note that μ and σ implicitly define the importance between different subtasks during ML model training. Importance of subtasks can be further adjusted via introducing dedicated weights for the corresponding loss terms. The detailed pre-processing pipeline can be found in the provided source code (see Section 1.1).

Note that although accurate tracking of the rotation state is important for many aspects of planet formation and evolution modelling, we do not include rotation in our model predictions. This is because it is non-trivial to derive physically reliable (and unique) post-collision rotation states within our post-processing chain for SPH collision simulations. A major point to consider is that our definition of a remnant is not restricted to a single physically connected fragment, but also includes all gravitationally bound fragments in addition. In reality, these fragments may or may not be actually accreted at some later point in time, or interact otherwise with each other. Nevertheless, we consider including these fragments into the definition of *remnant* as the best possible option, considering the alternative of simply ignoring them. This comes on top of the general issue that approximate rotational equilibrium has to be achieved after the collision in order to extract a reliable rotation state, which is highly scenario-dependent in terms of the relevant dynamics and time-scales. Considering those difficulties, we decided not to include the rotation state in our ML model predictions. Therefore, while pre-collision rotation is fully accounted for, we do not attempt to predict post-collision rotation states in this work.

Our optimization objective is to minimize the mean absolute error (MAE) between model predictions y_i^T and ground truth labels z_i^T over our training data:

$$y_i^T = f\left(y_i^0, T, \phi\right) \tag{7}$$

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \xi \cdot ||y_i^T - z_i^T|| \tag{8}$$

$f : \mathbb{R}^d \rightarrow \mathbb{R}^k$ refers to an ML model that regresses final states when given initial states. In this work, we focus on handling macroscopic system states for both model inputs and outputs, resulting in $d = 25$ and $k = 27$. f has learnable parameters ϕ that we aim to optimize. M refers to the size of the training split for individual crossvalidation folds.

We treat every unit of mass (i.e. every kilogram) as equally important in our ML task. We account for this treatment by calculating mass-dependent weights ξ in order to re-weight errors for output features that correspond to the largest remnant, second-largest remnant, and the rest (of material), respectively.

$$\xi = [\xi_{\rm lr}, \ \xi_{\rm 2lr}, \ \xi_{\rm rest}] \tag{9}$$

$$\xi = \left[\frac{m_{\rm lr}}{m_{\rm tot}}, \frac{m_{\rm 2lr}}{m_{\rm tot}}, \frac{m_{\rm rest}}{m_{\rm tot}}\right] \tag{10}$$

$$m_{\rm tot} = m_{\rm lr} + m_{\rm 2lr} + m_{\rm rest} \tag{11}$$

We consider this re-weighting to be essential to accurately reflect the prediction problem, especially if $m_{\rm 2lr} << m_{\rm lr}$ or $m_{\rm rest} << m_{\rm lr}$. Early experiments without re-weighting lead to poor prediction performance originating from small objects (remnants and fragments). In general, small objects are more difficult to predict compared to large objects. Moreover, the assignment of the second-largest remnant tends to jump in the presence of many small objects, making it almost impossible to predict robustly. This labelling noise then leads to large error gradients, hampering learning significantly. This problem is ameliorated with our re-weighting approach.

We use identical training hyperparameters for our deep learning models (FFN and RES). Training is performed via stochastic gradient descend (Robbins & Monro 1951), utilizing the backpropagation algorithm (Kelley 1960; Rumelhart, Hinton & Williams 1986). We use the adamax optimizer (Kingma & Ba 2014) with default hyperparameters, a minibatch size of $bs = 128$, a constant learning rate of $\eta = 0.0005$, and a weight decay of $wd = 0.0001$. We apply gradient-norm clipping (Pascanu, Mikolov & Bengio 2013), allowing for maximum gradient norms of $n_{\rm grad} = 50$. Moreover, we use exponential moving average models (Ruppert 1988; Polyak 1990; Tarvainen & Valpola 2017) with a rate of $r_{\rm ema} = 0.999$ for validation and testing. We find that the mean-squared error leads to worse validation performance than the MAE, which is more robust to outliers. To alleviate the exploding gradient problem as described by Metz et al. (2021), we additionally penalize too large activations of hidden states h_t in our RES model. We train each of our models for 5000 epochs, which is sufficient to reach convergence.

Since different ML architectures are inherently difficult to compare, we try to find the best architectures and respective models in terms of validation performance for each model class (i.e. FFN and RES) separately. We optimize hyperparameters manually in an iterative manner (i.e. always optimizing one hyperparameter at a time while keeping others fixed, and repeating the procedure until convergence in validation performance) and dedicate approximately the same amount of time and computational resources to optimize each set of hyperparameters for FFN and RES. Table B1 summarizes all optimized hyperparameters for FFN and RES, while LIN has no model-class hyperparameters. In order to prevent information leakage and misleading test performance, we solely perform hyperparameter finetuning based on validation performance. Test performance is measured after model development was completed. In principle, RES allow using intermediate states as additional learning signals. Unless stated otherwise, we only use final states for training to ensure a fair comparison with our baselines.

We use the root mean squared error (RMSE) as our validation metric. For certain applications, prediction speed may play a significant role. We note that the RMSE metric does not account for this aspect and thus purely focuses on prediction accuracy. All performance results reported below are obtained by first taking the best RMSE

(minimum) over all epochs for every fold individually, then averaging them over the folds. Errors indicate the minimum and maximum (over folds) of best RMSE values. The same procedure holds for classification accuracies, except for first taking the maximum instead of the minimum. Measuring and interpreting RMSE in the data space is unintuitive in our multi-task setting due to vastly different value ranges of individual quantities. Thus, RMSE is measured in ML feature-space. Moreover, since RMSE values have to be interpreted in consideration of the overall ML task, we recommend comparing reported results relative to each other.

We use the balanced accuracy score for validating classification performances of our models. Balanced accuracy in the multiclass classification setting is defined by taking the average of true-positive rates for individual classes. The true-positive rate is also referred to as sensitivity or recall. Considering the strong class-imbalances that are typically present in planetary collision data, we consider the balanced accuracy score to be much more problem-focused and more applicable compared to the unbalanced accuracy score.

2.2.6 *Efficiency considerations of ML*

In practice, researchers are interested at which point using ML starts to pay off compared to classical approaches for a fixed computational budget. Consider the goal of predicting m collision outcomes. For classical approaches such as PIM or direct SPH simulations, we only need to consider inference times, which scale linearly with m. On the other hand, ML requires consideration of data generation, training, and inference. In general, the required computation times for these three components are mostly independent from each other. In our case, data generation requires by far the most time, followed by training. Finally, ML inference requires only a tiny fraction of the overall computation budget. Thus, we recommend using ML approaches in case of extensive inference, i.e. large m. Let us define the three computation times τ_d for generating one data point, τ_t for ML model training, and τ_i for inference of one data point. In our case, we consider τ_t as the total wall-clock training time for fivefolds, each having 5000 epochs. N refers to the training data set size. We can calculate for which m the use of ML pays off (i.e. $T_\mathrm{ML} < T_\mathrm{CL}$) when comparing the overall computation times T_CL for classical approaches with computation times T_ML for ML approaches:

$$T_\mathrm{CL} = m \times \tau_\mathrm{i,CL} \tag{12}$$

$$T_\mathrm{ML} = N \times \tau_\mathrm{d} + \tau_\mathrm{t} + m \times \tau_\mathrm{i,ML} \tag{13}$$

$$m > \frac{N \times \tau_\mathrm{d} + \tau_\mathrm{t}}{\tau_\mathrm{i,CL} - \tau_\mathrm{i,ML}} \tag{14}$$

3 RESULTS

3.1 SPH collision data

The provided SPH data serves as the basis for ML models in order to solve the collision treatment problem accurately and fast. To our knowledge, this data set is the first of its kind to combine different aspects such as object rotation, realistic object models including water layers, and providing time-series data. All data can be freely accessed (see Section 1.1).

Our results are consistent with Leinhardt & Stewart (2012) and Stewart & Leinhardt (2012) in identifying three major outcome regimes, erosion, accretion, and hit-and-run. We define these regimes as

(i) erosion: $m_{lr} < m_t$
(ii) accretion: $m_{lr} > m_t \wedge m_{2lr} \leq 0.1 m_p$
(iii) hit-and-run: $m_{lr} > m_t \wedge m_{2lr} > 0.1 m_p$

where subscripts indicate the largest remnant, second-largest remnant, target, and projectile, respectively. Each of these regimes can be further divided into subclasses, based on thresholds for remnant masses, as defined in Table 4, and plotted in Fig. 5. Erosion typically results from high impact velocities and/or low impact angles, whereas accretion mostly emerges for lower impact velocities. Hit-and-run either results from high impact angles, or from a combination of lower impact angles, large-enough projectile-to-target mass ratios, and impact velocities that are low enough to avoid global disruption but high enough to avoid an accretion-type outcome.

Table 4. Class counts for different outcome regimes, and for random and realistic collision parameters. The three major outcome regimes are each further divided into subclasses, based on remnant masses. Random parameters are obtained from uniform, random sampling, whereas realistic conditions are obtained from dynamically consistent N-body simulations.

Class	Subclass	Random	Realistic
Erosion	$m_{lr} < m_t/2$	18.3 % (1856)	3.0 % (151)
Erosion	$m_{lr} > m_t/2$	55.1 % (5600)	58.5 % (2968)
Accretion	$m_{lr} < m_t + m_p/2$	1.3 % (135)	2.0 % (102)
Accretion	$m_{lr} > m_t + m_p/2$	5.2 % (526)	5.5 % (281)
Hit-and-run	$m_{2lr} < m_p/2$	0.4 % (40)	4.4 % (221)
Hit-and-run	$m_{2lr} > m_p/2$	19.7 % (2007)	26.7 % (1353)

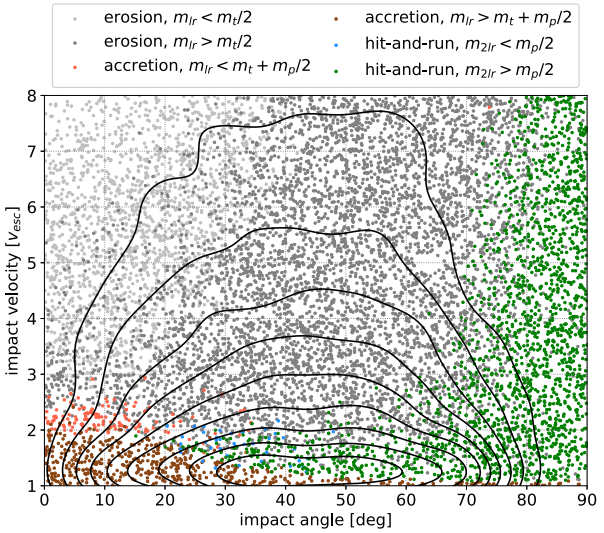


Figure 5. Overview of collision outcomes in impact angle – impact velocity space. Each data point represents a simulation in our SPH data set. Colours indicate major outcome regimes: erosion (grayish), accretion (reddish), and hit-and-run (greenish). Each regime is further divided into subcategories, depending on remnant masses. The contour overlay indicates collision statistics in a realistic dynamical environment, obtained from N-body simulations by Burger et al. (2020b), see Section 2.1.1. Contour levels correspond to iso-proportions of the density (in 10 per cent steps).

Table 5. Test performance (RMSE and balanced accuracy, both measured on the final state) of different approaches for planetary collision treatment. Classification is performed as a post-processing step on top of predicted masses. We assume the SPH simulation data to be the ground truth. For single-task learning (last six rows), each entry corresponds to the performance of individually trained ML models and column headers indicate optimized tasks, respectively. We use data from Timpe et al. (2020b) to obtain the results in the last four rows. Best results are indicated in bold, whereas $*$ indicate statistically significant results according to a Wilcoxon test (comparing FFN and RES). Our proposed RES model outperforms the other baseline methods in most experiments.

Method	Notes	Mass	Material	Position	Velocity	Total	Accuracy
PIM	o.o.d. test	$0.1501^{+0.0000}_{+0.0000}$	$0.0605^{+0.0000}_{+0.0000}$	$0.3046^{+0.0000}_{+0.0000}$	$0.2229^{+0.0000}_{+0.0000}$	$0.2450^{+0.0000}_{+0.0000}$	$0.1667^{+0.0000}_{+0.0000}$
LIN	o.o.d. test	$0.0460^{+0.0009}_{-0.0012}$	$0.0250^{+0.0002}_{-0.0002}$	$0.1880^{+0.0007}_{-0.0007}$	$0.1509^{+0.0015}_{-0.0014}$	$0.1479^{+0.0009}_{-0.0010}$	$0.2340^{+0.0043}_{-0.0057}$
FFN	o.o.d. test	$0.0121^{+0.0008}_{-0.0008}$	$0.0129^{+0.0006}_{-0.0003}$	$0.0487^{+0.0014}_{-0.0012}$	$0.0433^{+0.0010}_{-0.0010}$	$0.0408^{+0.0008}_{-0.0009}$	$\mathbf{0.4964^{+0.0480}_{-0.0538}}$
RES	o.o.d. test	$\mathbf{*0.0108^{+0.0003}_{-0.0005}}$	$\mathbf{0.0127^{+0.0002}_{-0.0002}}$	$\mathbf{*0.0386^{+0.0015}_{-0.0018}}$	$\mathbf{0.0428^{+0.0004}_{-0.0003}}$	$\mathbf{*0.0364^{+0.0009}_{-0.0010}}$	$0.4887^{+0.0160}_{-0.0139}$
FFN	o.o.d. test, + labels	$0.0122^{+0.0009}_{-0.0010}$	$\mathbf{0.0136^{+0.0002}_{-0.0004}}$	$0.0500^{+0.0015}_{-0.0020}$	$0.0433^{+0.0005}_{-0.0004}$	$0.0416^{+0.0009}_{-0.0010}$	$0.5309^{+0.0673}_{-0.1678}$
RES	o.o.d. test, + labels	$\mathbf{*0.0107^{+0.0008}_{-0.0009}}$	$0.0136^{+0.0002}_{-0.0002}$	$\mathbf{*0.0372^{+0.0012}_{-0.0014}}$	$\mathbf{0.0426^{+0.0011}_{-0.0006}}$	$\mathbf{*0.0358^{+0.0007}_{-0.0009}}$	$\mathbf{0.5311^{+0.0548}_{-0.0311}}$
FFN	o.o.d. test, single	$\mathbf{0.0083^{+0.0003}_{-0.0004}}$	$0.0098^{+0.0002}_{-0.0002}$	$0.0422^{+0.0009}_{-0.0010}$	$\mathbf{0.0409^{+0.0007}_{-0.0008}}$	–	$0.4720^{+0.0417}_{-0.0470}$
RES	o.o.d. test, single	$0.0083^{+0.0008}_{-0.0004}$	$\mathbf{*0.0090^{+0.0003}_{-0.0003}}$	$\mathbf{*0.0364^{+0.0024}_{-0.0017}}$	$0.0422^{+0.0004}_{-0.0010}$	–	$\mathbf{*0.5165^{+0.0349}_{-0.0228}}$
PIM	i.i.d. test, Timpe	$0.2088^{+0.0000}_{+0.0000}$	$0.1925^{+0.0000}_{+0.0000}$	$0.3691^{+0.0000}_{+0.0000}$	$0.2880^{+0.0000}_{+0.0000}$	–	$0.1667^{+0.0000}_{+0.0000}$
LIN	i.i.d. test, Timpe	$0.0518^{+0.0003}_{-0.0002}$	$0.0473^{+0.0003}_{-0.0005}$	$0.2105^{+0.0005}_{-0.0002}$	$0.1693^{+0.0002}_{-0.0005}$	–	$0.3160^{+0.0025}_{-0.0015}$
FFN	i.i.d. test, Timpe	$0.0132^{+0.0002}_{-0.0001}$	$0.0144^{+0.0002}_{-0.0001}$	$0.0877^{+0.0017}_{-0.0012}$	$0.0631^{+0.0006}_{-0.0004}$	–	$0.4675^{+0.0039}_{-0.0021}$
RES	i.i.d. test, Timpe	$\mathbf{*0.0126^{+0.0003}_{-0.0004}}$	$\mathbf{0.0141^{+0.0006}_{-0.0006}}$	$\mathbf{*0.0840^{+0.0037}_{-0.0036}}$	$\mathbf{0.0628^{+0.0007}_{-0.0009}}$	–	$\mathbf{0.4690^{+0.0026}_{-0.0023}}$

3.2 ML experiments

Below we present our results for the experiments described in Section 2.2.5. Results indicated by $*$ are statistically significant ($p < 0.05$) according to a Wilcoxon test when comparing FFN with RES for the respective experiments.

3.2.1 Performance

We compare commonly used methods for planetary collision treatment with our proposed RES model and summarize the results in Tables 5 and B2. Our o.o.d. test set consists of data points within manually selected regions in the impact angle – impact velocity space (dashed regions in Fig. 7).

All deep learning models outperform the PIM and LIN baselines by a large margin. Improved performance over PIM was expected, since it is an analytic model that applies very simplistic assumptions on collision dynamics. However, we still regard PIM as a useful method in case of limited computational resources. Improved results over the LIN baseline were also expected since it assumes that the data are linearly dependent. RES perform best amongst the deep learning approaches that we studied, significantly outperforming the FFN baseline, also illustrated in Fig. 6. In general, our deep learning models generalize well to the o.o.d. test set, indicating that they might even generalize beyond our covered parameter space (Table 1). RES consistently outperforms the FFN baseline in terms of RMSE on the o.o.d. test set.

We do not observe performance gains when using intermediate states as additional labels during training. We observe a shrinkage effect (regression to the data set mean) in model predictions when using intermediate states, which may harm performance measured on the final state. Moreover, non-converged intermediate states may have a relatively high labelling noise due to the discretization into remnants, potentially making intermediate states more difficult to predict compared to final states. We also believe that intermediate macro states are somewhat redundant, unless operating on a microscopic scale, i.e. directly learning from SPH particle representations or similar.

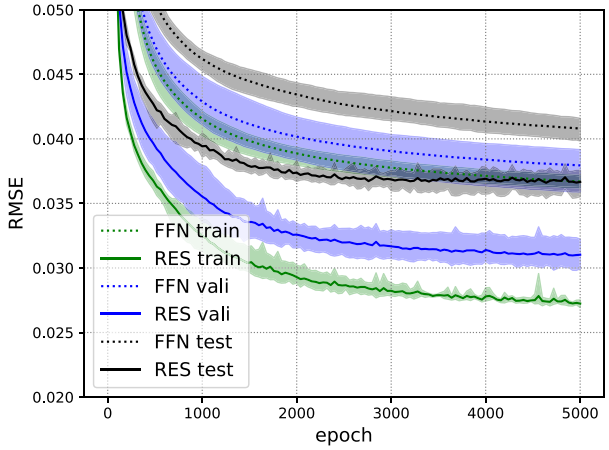


Figure 6. Learning curves depicting the training, validation, and test performance for our multi-task objective. RMSE is measured on the final state ($t = T$). Shaded regions indicate the minimum and maximum performance over the five crossvalidation folds. Performance increases significantly during the first 1000 epochs before converging. RES significantly outperforms the FFN baseline.

Direct prediction of final states with FFNs only allows analysis of hidden representations, which are typically too abstract to interpret due to their high-dimensional, learned nature. In addition to the analysis of hidden representations, our RES architecture allows analysing predicted intermediate states, opening up an entry point for interpreting model predictions in more detail. This is also illustrated in Fig. B1. Although our learning objective contains no incentive for predicted remnant trajectories to align with ground truth trajectories, we observe spatio-temporally continuous transitions from initial to final states. This indicates that steps in RES may correlate to the temporal evolution of the physical system to a certain extent, even though our results do not allow for strong conclusions.

We also train our ML models on the planetary collision data set provided by Timpe et al. (2020a, b; single-task setting) and report

Table 6. Number of learnable parameters and number of optimized hyperparameters of our methods, as well as typical computation times for data generation, training, and inference on the same hardware (GPU: Nvidia GTX 1080Ti). τ_d and τ_i are the average computation times for generating one data point and model inference for one data point, whereas τ_t is the total wall-clock training time for fivefolds, each having 5000 epochs. The required time for data generation is $\tau_D = N \times \tau_d$ and takes the largest part of the overall computational budget. s_h is the number of RES model steps taken per simulated hour.

Method	#Param	#Hyper	τ_d	τ_t	τ_i
PIM	0	0	–	–	<1 s
LIN	729	0	0.75 h	<1 s	<1 s
FFN	22203	2	0.75 h	11.95 h	<1 s
RES, $s_h = 1$	64417	3	0.75 h	33.51 h	<1 s
RES, $s_h = 2$	64417	3	0.75 h	44.78 h	<1 s
RES, $s_h = 3$	64417	3	0.75 h	56.45 h	<1 s
RES, $s_h = 4$	64417	3	0.75 h	66.74 h	<1 s
SPH	0	∼5	–	–	0.75 h

Table 7. RMSE of deep learning models on the fixed-size o.o.d. test set when using 100 per cent, 50 per cent, and 25 per cent of training data. RES requires 50 per cent less data to achieve similar performance compared to FFN.

Method	100 per cent data	50 per cent data	25 per cent data
FFN	$0.0408^{+0.0008}_{-0.0009}$	$0.0442^{+0.0004}_{-0.0005}$	$0.0516^{+0.0012}_{-0.0012}$
RES	$\mathbf{*0.0364^{+0.0009}_{-0.0010}}$	$\mathbf{*0.0398^{+0.0009}_{-0.0013}}$	$\mathbf{*0.0469^{+0.0013}_{-0.0010}}$

performance results on the independent and identically distributed (i.i.d.) test set in Table 5. Without performing any additional hyperarameter finetuning, we observe that RES outperforms the other baseline methods in all sub-tasks, verifying that our RES model is data set-agnostic. However, our results are not statistically significant for 2/4 subtasks.

3.2.2 *Efficiency*

We report the required computational resources of our methods in Table 6. As expected, PIM marks one extremum of the accuracy-speed trade-off, requiring the least amount of computational resources. On the other hand, SPH marks the other extremum, scaling badly with m (see Section 2.2.6). ML models cover intermediate trade-off regions, depending on the model class and hyperparameter choices. ML models are approximately 4 magnitudes faster in inference compared to SPH (both running on a single GPU), allowing ML models to be efficiently used in large-scale planetary evolution simulations. Considering the results from Table 6, for our deep learning models it holds that $N \times \tau_d \gg \tau_t \gg N \times \tau_i$ for $N = 10164$. This indicates that the most effective way to save computation when using ML is by requiring less training data, i.e. small N. However, using less data will inevitably lead to worse performance in terms of RMSE and to degrading generalization.

Therefore, we perform ablation studies using different training data set sizes to investigate data-efficiency of our deep learning models and report the corresponding o.o.d. test performances in Table 7. The results indicate that RES requires 50 per cent less data to achieve comparable performance to the FFN baseline. In other words, using RES is much more efficient than the FFN baseline in the case of comparable performance. Once trained, ML methods are practically as fast as PIM, while maintaining high prediction accuracy.

Table 6 also summarizes the number of learnable parameters and the number of model class hyperparameters of all collision treatment methods we studied in this work. ML training requires additional hyperparameters such as bs, η, wd, and n_grad. The number of learnable parameters for LIN is fully determined by the ML task, i.e. the dimensions of input and label vectors. For deep learning methods, the number of learnable parameters depends on the hyperparameter choices that ultimately define model architectures. We optimize hyperparameters w.r.t. validation performance and perform extensive ablation studies for both FFN and RES to verify their optimal hyperparameters. We consider the reported model capacities (i.e. the number of learnable parameters) to be optimal in terms of validation performance for our data. We find that the optimal FFN architecture requires less learnable parameters compared to the optimal RES architecture. In particular, increasing FFN's size does not improve its prediction performance anymore. We refer to Section B2 for more details about hyperparameters. The number of hyperparameters for the direct-SPH method accounts for the most important method-specific aspects such as the smoothing length and settings related to the equation of state of the simulated material.

We study the effects of multi-task learning and single-task learning on model performance and report ablation studies in Table 5. Multi-task learning is computationally more efficient by design (i.e. prediction speed and required parameters), requiring only a single model for predicting several different modalities. However, we observe a performance decrease when comparing multi-task learning (one model with four tasks) to single-task learning (four models, each with one task) in favour of single-task learning for both FFN and RES. We perform statistical significance tests (Wilcoxon) between single-task and multi-task experiments. We find single-task significantly ($p < 0.05$) outperforms multi-task for almost all subtasks for both FFN and RES. Two exceptions are performances for the position and velocity tasks on the o.o.d. test set for the RES model. We conclude that our hypothesis of improved generalization due to mutual benefit via exploiting shared representations does not hold in our experiments. Since we optimized model hyperparameters for the multi-task setting, single-task models might be even better with further finetuning. RES outperforms FFN in 4/4 subtasks for multi-task learning, but only in 2/4 for single-task learning, indicating that RES benefits a bit more from multi-task learning.

We verify that our regression-approach is suited to perform classification of different collision scenarios as a post-processing step, avoiding the need for two-step classification-regression approaches. Classification accuracies are calculated on top of regression results w.r.t. the six different classes as defined in Table 4. Balanced accuracies are reported in Table 5. Fig. 7 visualizes predicted collision outcomes based on predicted masses of the largest and second-largest remnants.

We observe that our ML models typically tend to mispredict actual accretion scenarios as hit-and-run, ultimately resulting in misclassifications in post-processing. This is pronounced for low-velocity (close to v_esc) collisions, and particularly for lower impact angles ($\lesssim 30^\circ$), and directly visible when comparing Figs 7 and 5. We assume that this mistreatment stems from the relative under-representation of accretion scenarios in our data (see Table 4), which results in models having poor classification performance for the respective scenarios. This is a common problem in imbalanced classification tasks and can be tackled with different approaches such as generating more data for under-represented classes, oversampling of under-represented classes, regularizing the model during training, or introducing problem-specific model architectures. Note that the balanced accuracy score succeeds in reflecting the reduced classi-

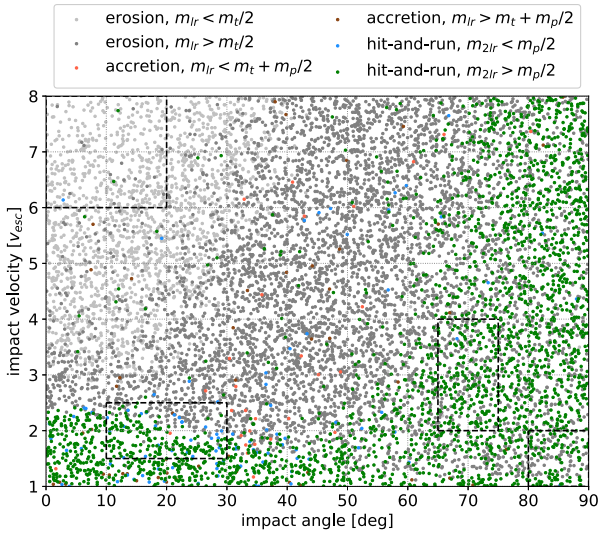


Figure 7. Classification results on validation and o.o.d. test data using predicted remnant masses of the RES model. Data points for validation are obtained by combining the respective validation splits that originate from different crossvalidation folds, whereas o.o.d. test set regions are marked with dashed boxes. The model learned to differentiate between typical collision scenarios/outcomes and generalizes to the o.o.d. test set.

fication performance on under-represented classes. In contrast, the unbalanced accuracy score typically tends to be much higher, since under-represented classes are not accounted for properly.

4 CONCLUSION

We perform *N*-body and SPH simulations to tackle the problem of accurate and fast treatment of planetary collisions with ML methods. We use the SPH data to employ a simple but problem-adapted ML model for predicting masses, material fractions, positions, and velocities of the two largest post-collision remnants and the remaining debris. Our model helps the modelling of temporal dynamics by evolving system states in an autoregressive manner, which closely resembles the data-generating process. The model allows for flexible prediction of post-collision states at different times and can be employed for collision treatment within existing *N*-body simulation frameworks.

We summarize our experiment results by comparing the performance of our two best methods, the FFN baseline and our proposed RES model, for all experiments on the respective test sets. We count a total of 24 comparisons. RES outperforms FFN in 20/24 cases. In 13/20 cases for RES and 0/4 cases for FFN, improvements are statistically significant ($p < 0.05$). Moreover, RES is also more data-efficient, achieving similar performance while requiring approximately half as much data compared to FFN. Although multi-task learning is more computationally efficient than single-task learning, we do not observe improved generalization induced by shared representations.

We demonstrate that the FFN baseline is outperformed by our RES model due to its problem-adapted algorithmic bias (i.e. its autoregressive structure), which is better suited for modelling the underlying physical processes in planetary collisions over time (see also Section 2.2.2). The optimal RES architecture requires more learnable parameters than the optimal FFN architecture because RES consists of three neural network modules, whereas FFN consists of only one module. However, we find that increasing FFN's size does not increase its performance anymore. This finding indicates that the performance improvement from RES originates from its better algorithmic bias rather than from its relatively larger number of learnable parameters. Moreover, the superiority of RES is apparent in both our own data and data from Timpe et al. (2020b), indicating a general trend rather than a data-specific effect.

Measuring the actual effects of systematic errors induced by ML model predictions still remains an open topic in the context of planetary formation and evolution modelling. Thus, a natural follow-up work could be to test various ML methods for collision treatment in such simulations.

Beyond studying the basic task of collision outcome prediction, our data and methods also open up further interesting lines of research related to planet formation in general. This includes studying inverse problems or focusing on specific collision scenarios (Canup, Barr & Crawford 2012; Chau et al. 2018). Other possible directions are the extension of our methods to different regimes such as small bodies or objects including (proto-)atmospheres, probably requiring to extend the underlying physics model. The latter can include more sophisticated equations of state for more realistic thermodynamics and advanced models for material strength to accurately simulate solid-body behaviour.

ML models might benefit from learning directly based on microscopic representations, i.e. fragments or even down to the SPH particle level, and thereby improve aspects regarding generalization and interpretability. Incorporating certain aspects like symmetries, conserved quantities, and sophisticated numerical approaches that have been developed in recent years (Chen et al. 2020; Alet et al. 2021; Hoedt et al. 2021; Satorras, Hoogeboom & Welling 2021; Brandstetter et al. 2022) could be promising directions for further improvements of ML architectures. For example, graph neural networks (GNNs; Scarselli et al. 2009; Defferrard, Bresson & Vandergheynst 2016; Kipf & Welling 2017) and regression forests (Ho 1995) have been successfully applied to the approximation of numerical simulations (Ladický et al. 2015; Martinkus, Lucchi & Perraudin 2020; Pfaff et al. 2020; Sanchez-Gonzalez et al. 2020; Mayr et al. 2021). Efficient ML approaches begin to replace traditional PDE solvers in the context of hydrodynamic simulations (Li et al. 2020a,b). Residual neural networks (He et al. 2016) showed promising results in modelling complex dynamical processes by formulating the neural network layer-structure as a continuous-depth model (Queiruga et al. 2020) in the context of neural ODEs (Chen et al. 2018; Kidger 2022). ML methods allow accurate collision modelling at scale, while at the same time being orders of magnitude faster compared to classical, non data-driven approaches.

ACKNOWLEDGEMENTS

Christoph Burger (CB) and Christoph M. Schäfer (CMS) appreciate support by the German Research Foundation (DFG) project 398488521. CB acknowledges support by the Austrian Science Fund (FWF) project S11603-N16. Thomas I. Maindl (TIM) acknowledges support from the Austrian Science Fund (FWF): P33351-N. The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no. INST 37/935-1 FUGG.

We thank Sepp Hochreiter, Günter Klambauer, Johannes Brandstetter, Pieter-Jan Hoedt, Max Zimmermann, Rudolf Dvorak, Kajetan Schweighofer, Andreas Mayr, Miles Timpe, and Christian Reinhardt

for contributing valuable feedback and comments. Simulations for our data sets made use of the REBOUND code (Rein & Liu 2012). We thank the developers of PYTHON (Van Rossum & Drake 2009), PYTORCH (Paszke et al. 2019), MATPLOTLIB (Hunter 2007), and NUMPY (Harris et al. 2020), whose contributions facilitate the advancement of our work.

DATA AVAILABILITY

As extensive simulation data sets for ML begin to emerge (Villaescusa-Navarro et al. 2022), we advocate making these data sets publicly available in order to optimize the benefit from network effects among the research community. Data sharing helps to reduce spending computational resources for data generation, making ML approaches more efficient and sustainable in the long term.

We provide our data, source code, and pre-trained ML models to encourage independent researchers to reproduce our results and to incorporate our methods into their own work. The data underlying this article are available in the Phaidra repository at https://phaidra.univie.ac.at/o:1206181, and can be accessed with the handle 11353/10.1206181. Our source code and pre-trained ML models can be accessed at https://github.com/littleblacksheep/csv/tree/main. We kindly ask users to report possible bugs to winter@murena.io.

REFERENCES

Agarwal S., Tosi N., Kessel P., Breuer D., Montavon G., 2021, Phys. Rev. Fluids, 6, 113801
Agnor C. B., Canup R. M., Levison H. F., 1999, Icarus, 142, 219
Alet F., Doblar D., Zhou A., Tenenbaum J. B., Kawaguchi K., Finn C., 2021, preprint (arXiv:2112.03321)
Bonomo A. S. et al., 2019, Nat. Astron., 3, 416
Bonsor A., Leinhardt Z. M., Carter P. J., Elliott T., Walter M. J., Stewart S. T., 2015, Icarus, 247, 291
Brandstetter J., Hesselink R., van der Pol E., Bekkers E. J., Welling M., 2022, preprint (arXiv:2110.02905)
Breiman L., Friedman J. H., Olshen R. A., Stone C. J., 1984, Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA
Brown T. B. et al., 2020, in Larochelle H., Ranzato M., Hadsell R., Balcan M. F., Lin H., eds, Language Models are Few-Shot Learners, vol. 33, Curran Associates, Inc., p. 1877, https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
Burger C., 2019, Dissertation. University of Vienna, Austria
Burger C., Schäfer C. M., 2017, The First Greek-Austrian Workshop on Extrasolar Planetary Systems, Proceedings of the First Greek-Austrian Workshop on Extrasolar Planetary Systems. p. 63
Burger C., Maindl T. I., Schäfer C. M., 2018, Celest. Mech. Dyn. Astron., 130, 2
Burger C., Maindl T. I., Schäfer C., 2020a, in Elmegreen B. G., Tóth L. V., Güdel M., eds, IAU Symposium, Vol. 345, IAU Symposium. p. 287 preprint (arXiv:2002.00231), doi:10.1017/S1743921318008621
Burger C., Bazsó Á., Schäfer C. M., 2020b, A&A, 634, A76
Cambioni S., Asphaug E., Emsenhuber A., Gabriel T. S. J., Furfaro R., Schwartz S. R., 2019, ApJ, 875, 40
Cambioni S., Jacobson S. A., Emsenhuber A., Asphaug E., Rubie D. C., Gabriel T. S. J., Schwartz S. R., Furfaro R., 2021, Planet. Sci. J., 2, 93
Canup R. M., Barr A. C., Crawford D. A., 2012, Icarus, 222, 200
Carter P. J., Leinhardt Z. M., Elliott T., Walter M. J., Stewart S. T., 2015, ApJ, 813, 72
Carter P. J., Leinhardt Z. M., Elliott T., Stewart S. T., Walter M. J., 2018, Earth Planet. Sci. Lett., 484, 276
Chambers J. E., 2001, Icarus, 152, 205
Chambers J., 2013, Icarus, 224, 43
Chambers J. E., Wetherill G. W., 1998, Icarus, 136, 304
Chau A., Reinhardt C., Helled R., Stadel J., 2018, ApJ, 865, 35
Chen T., Guestrin C., 2016, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16. ACM, New York, NY, USA, p. 785
Chen R. T. Q., Rubanova Y., Bettencourt J., Duvenaud D., 2018, Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18. Curran Associates Inc., Red Hook, NY, USA, p. 6572, http://doi.acm.org/10.1145/2939672.2939785
Chen Z., Zhang J., Arjovsky M., Bottou L., 2020, in 8th International Conference on Learning Representations, ICLR 2020, April 26-30, OpenReview.net, 2020,. Addis Ababa, Ethiopia, https://openreview.net/forum?id = BkgYPREtPr
Cho K., van Merriënboer B., Bahdanau D., Bengio Y., 2014, in Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. Association for Computational Linguistics, Doha, Qatar, p. 103, https://aclanthology.org/W14-4012
Clement M. S., Kaib N. A., Raymond S. N., Chambers J. E., Walsh K. J., 2019, Icarus, 321, 778
Cortes C., Vapnik V., 1995, Mach. Learn., 20, 273
Cranmer M., Tamayo D., Rein H., Battaglia P., Hadden S., Armitage P. J., Ho S., Spergel D. N., 2021, Proc. Natl. Acad. Sci., 118, e2026053118
Crespi S., Dobbs-Dixon I., Georgakarakos N., Haghighipour N., Maindl T. I., Schäfer C. M., Winter P. M., 2021, MNRAS, 508, 6013
Defferrard M., Bresson X., Vandergheynst P., 2016, in Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16. Curran Associates Inc., Red Hook, NY, USA, p. 3844
Dong X., Thanou D., Rabbat M., Frossard P., 2019, IEEE Signal Process. Mag., 36, 44
Dwyer C. A., Nimmo F., Chambers J. E., 2015, Icarus, 245, 145
Elman J. L., 1990, Cogn. Sci., 14, 179
Emsenhuber A., Cambioni S., Asphaug E., Gabriel T. S. J., Schwartz S. R., Furfaro R., 2020, ApJ, 891, 6
Fischer R. A., Ciesla F. J., 2014, Earth Planet. Sci. Lett., 392, 28
Friedman J. H., 2001, Ann. Stat., 29, 1189
Geller M. J., Huchra J. P., 1983, ApJS, 52, 61
Genda H., Kokubo E., Ida S., 2011, 42nd Lunar and Planetary Science Conference. p. 2090
Genda H., Fujita T., Kobayashi H., Tanaka H., Suetsugu R., Abe Y., 2017, Icarus, 294, 234
Genda H., Iizuka T., Sasaki T., Ueno Y., Ikoma M., 2017, Earth Planet. Sci. Lett., 470, 87
Gers F. A., Schmidhuber J., Cummins F., 1999, Learning to forget: Continual Prediction with LSTM, Vol. 2, . p. 850
Gingold R. A., Monaghan J. J., 1977, MNRAS, 181, 375
Haghighipour N., Maindl T. I., 2022, ApJ, 926, 197
Haghighipour N., Raymond S. N., 2007, ApJ, 666, 436
Haghighipour N., Winter O. C., 2016, Celest. Mech. Dyn. Astron., 124, 235
Harris C. R. et al., 2020, Nature, 585, 357
Hastie T., Tibshirani R., Friedman J., 2017, The Elements of Statistical Learning. Springer, New York, NY
He K., Zhang X., Ren S., Sun J., 2016, Deep Residual Learning for Image Recognition, IEEE, Las Vegas, NV, USA, p. 770
Hearst M., Dumais S., Osuna E., Platt J., Scholkopf B., 1998, IEEE Intell. Syst. Appl., 13, 18
Ho T. K., 1995, Proceedings of 3rd international conference on document analysis and recognition, IEEE, Montreal, QC, Canada, p. 278
Hochreiter S., 1998, Int. J. Uncertain. Fuzziness Knowl.-Based Syst., 6, 107
Hochreiter S., Schmidhuber J., 1997, Neural Comput., 9, 1735
Hoedt P.-J., Kratzert F., Klotz D., Halmich C., Holzleitner M., Nearing G. S., Hochreiter S., Klambauer G., 2021, Proceedings of the 38 th International Conference on Machine Learning, PMLR, p. 139
Hornik K., Stinchcombe M., White H., 1989, Neural Netw., 2, 359
Hunter J. D., 2007, Comput. Sci. Eng., 9, 90
Inamdar N. K., Schlichting H. E., 2016, ApJ, 817, L13
Ivakhnenko A. G., Lapa V. G., 1965, Cybernetic Predicting Devices. CCM Information Corporation.U.S. department of commerce. Clearinghouse for federal scientific and technical information, Washington, D.C.
Izidoro A., de Souza Torres K., Winter O. C., Haghighipour N., 2013, ApJ, 767, 54

Johansen A., Lambrechts M., 2017, Ann. Rev. Earth Planet. Sci., 45, 359
Jordan M. I., 1986, Technical report AD-A-173989/5/XAB; ICS-8604, Serial order: A Parallel Distributed Processing Approach.US, CA
Kegerreis J. A., Eke V. R., Massey R. J., Teodoro L. F. A., 2020, ApJ, 897, 161
Kelley H. J., 1960, Ars J., 30, 947
Kidger P., 2022, On neural differential equations, University of Oxford
Kingma D. P., Ba J., 2014, in Bengio B., LeCun Y., eds, 3rd International Conference on Learning Representations, San Diego, CA, preprint (arXiv:1412.6980)
Kipf T., Welling M., 2017, preprint (arXiv:1609.02907)
Klambauer G., Unterthiner T., Mayr A., Hochreiter S., 2017, preprint (arXiv:1706.02515)
Kokubo E., Ida S., 1998, Icarus, 131, 171
Kokubo E., Ida S., 2002, ApJ, 581, 666
Kokubo E., Kominami J., Ida S., 2006, ApJ, 642, 1131
Krizhevsky A., Sutskever I., Hinton G. E., 2017, Commun. ACM, 60, 84
Kutta W., 1901, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, B.G Teubner, Leipzig, https://www.bibsonomy.org/bibtex/24ee695c671c31151cc9816e314dfa6c2/brouder
Ladický L., Jeong S., Solenthaler B., Pollefeys M., Gross M., 2015, ACM Trans. Graph., 34, 199
Leinhardt Z. M., Stewart S. T., 2012, ApJ, 745, 79
Leinhardt Z. M., Dobinson J., Carter P. J., Lines S., 2015, ApJ, 806, 23
Li Z., Kovachki N., Azizzadenesheli K., Liu B., Bhattacharya K., Stuart A., Anandkumar A., 2020a, preprint (arXiv:2010.08895)
Li Z., Kovachki N., Azizzadenesheli K., Liu B., Bhattacharya K., Stuart A., Anandkumar A., 2020b, Workshop on Integration of Deep Neural Models and Differential Equations, ICLR, Virtual Event, preprint (arXiv:2003.03485)
Lucy L. B., 1977, AJ, 82, 1013
Maindl T. I., Dvorak R., Schäfer C., Speith R., 2014, IAU Symp. 310, Complex Planetary Systems, Cambridge University Press, p. 138
Maindl T. I., Schäfer C. M., Haghighipour N., Burger C., Dvorak R., 2017, Proceedings of the First Greek-Austrian Workshop on Extrasolar Planetary Systems, CreateSpace Independent Publishing Platform, South Carolina, United States, p. 137
Marcus R. A., Stewart S. T., Sasselov D., Hernquist L., 2009, ApJ, 700, L118
Marcus R. A., Sasselov D., Stewart S. T., Hernquist L., 2010, ApJ, 719, L45
Martinkus K., Lucchi A., Perraudin N., 2020, preprint (arXiv:2010.06948)
Mayr A., Lehner S., Mayrhofer A., Kloss C., Hochreiter S., Brandstetter J., 2021, preprint (arXiv:2105.01636)
McNeil D., Duncan M., Levison H. F., 2005, AJ, 130, 2884
Melosh H. J., 1989, Impact Cratering: A Geologic Process, Oxford University Press ; Clarendon Press New York, Oxford
Metz L., Freeman C. D., Schoenholz S. S., Kachman T., 2021, preprint (arXiv:2111.05803)
Mitchell T. M., 1980, Technical report, The Need for Biases in Learning Generalizations, Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ., New Jersey, p. 184
Monfared Z., Mikhaeil J. M., Durstewitz D., 2021, preprint (arXiv:abs/2110.07238)
Morbidelli A., Chambers J., Lunine J. I., Petit J. M., Robert F., Valsecchi G. B., Cyr K. E., 2000, Meteorit. Planet Sci., 35, 1309
Mozer M. C., 1995, Complex Syst., 3, 349
O'Brien D. P., Walsh K. J., Morbidelli A., Raymond S. N., Mandell A. M., 2014, Icarus, 239, 74
O'Brien D. P., Izidoro A., Jacobson S. A., Raymond S. N., Rubie D. C., 2018, Space Sci. Rev., 214, 47
Pascanu R., Mikolov T., Bengio Y., 2013, Proceedings of the 30th International Conference on Machine Learning, Vol. 28, PMLR, Atlanta, USA, p. 1310
Paszke A. et al., 2019, in Advances in Neural Information Processing Systems 32. Curran Associates, Inc., p. 8024
Pearlmutter B. A., 1989, Neural Comput., 1, 263
Pfaff T., Fortunato M., Sanchez-Gonzalez A., Battaglia P. W., 2020, preprint (arXiv:2010.03409)
Polyak B., 1990, Avtom. Telemekh., 7, 98
Press, William H. F. B. P., Teukolsky S. A., Vetterling W. T., 1992, Numerical Recipes in Fortran 77: The Art of Scientific Computing. Cambridge Univ. Press, Cambridge
Queiruga A. F., Erichson N. B., Taylor D., Mahoney M. W., 2020, preprint (arxiv:2008.02389)
Quintana E. V., Lissauer J. J., 2014, ApJ, 786, 33
Quintana E. V., Barclay T., Borucki W. J., Rowe J. F., Chambers J. E., 2016, ApJ, 821, 126
Quionero-Candela J., Sugiyama M., Schwaighofer A., Lawrence N. D., 2009, Dataset Shift in Machine Learning. The MIT Press, Cambridge, Massachusetts, USA
Rampásek L., Wolf G., 2021, preprint (arXiv:2107.07432)
Rasmussen C. E., Williams C. K. I., 2005, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge, Massachusetts, USA
Raymond S. N., Quinn T., Lunine J. I., 2004, Icarus, 168, 1
Raymond S. N., Quinn T., Lunine J. I., 2007, Astrobiology, 7, 66
Rein H., Liu S. F., 2012, A&A, 537, A128
Reufer A., Meier M. M., Benz W., Wieler R., 2012, Icarus, 221, 296–299
Robbins H., Monro S., 1951, Annals Math. Stat., 22, 400
Robinson A. J., Fallside F., 1987, Technical Report CUED/F-INFENG/TR.1, The Utility Driven Dynamic Error Propagation Network. Engineering Department, Cambridge Univ., Cambridge, UK
Rosenblatt F., 1961, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, Washington DC
Rumelhart D., Hinton G. E., Williams R., 1986, Nature, 323, 533
Runge C., 1895, Math. Ann., 46, 167
Ruppert D., 1988, Technical Report, Efficient Estimations from a Slowly Convergent Robbins-Monro Process, Cornell University Operations Research and Industrial Engineering
Sanchez-Gonzalez A., Godwin J., Pfaff T., Ying R., Leskovec J., Battaglia P. W., 2020, preprint (arXiv:2002.09405)
Satorras V. G., Hoogeboom E., Welling M., 2021, preprint (arXiv:2102.09844)
Scarselli F., Gori M., Tsoi A. C., Hagenbuchner M., Monfardini G., 2009, IEEE Trans. Neural Netw., 20, 61
Schäfer C., Riecker S., Maindl T. I., Speith R., Scherrer S., Kley W., 2016, A&A, 590, A19
Schäfer C. et al., 2020, Astron. Comput., 33, 100410
Schmidt D., Koppe G., Monfared Z., Beutelspacher M., Durstewitz D., 2021, Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies, ICLR, Virtual Event, Austria
Srivastava R. K., Greff K., Schmidhuber J., 2015, Highway Networks, 32nd International Conference on Machine Learning, Lille, France
Stewart S. T., Leinhardt Z. M., 2012, ApJ, 751, 32
Stoer J., Bulirsch R., 1980, Introduction to Numerical Analysis, Vol. §7.2.14. Springer-Verlag, New York
Tamayo D. et al., 2020, Proc. Natl. Acad. Sci., 117, 18194
Tarvainen A., Valpola H., 2017, in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, p. 1195
Tillotson J. H., 1962, Metallic Equations of State For Hypervelocity Impact, General Atomic Report GA-3216. 1962. Technical Report.Air Force Weapons Lab, San Diego, California
Timpe M. L., Han Veiga M., Knabenhans M., Stadel J., Marelli S., 2020a, Machine learning applied to simulations of collisions between rotating, differentiated planets, Springer International Publ.
Timpe M. L., Han Veiga M., Knabenhans M., Stadel J., Marelli S., 2020b, Comput. Astrophys. Cosmol., 7, 2
Valencia D., Paracha E., Jackson A. P., 2019, ApJ, 882, 35
Van Rossum G., Drake F. L., 2009, Python 3 Reference Manual. CreateSpace, Scotts Valley, CA
Villaescusa-Navarro F., et al., 2022, ApJ, 929, 132
Werbos P. J., 1988, Neural Networks, 1, 339
Wiener N., 1938, Am. J. Math., 60, 897
Wiewel S., Becher M., Thürey N., 2019, Comput. Graph. Forum, 38, 71
Wyatt M. C., Jackson A. P., 2016, Space Sci. Rev., 205, 231
Zhou L., Dvorak R., Zhou L.-Y., 2021, MNRAS, 505, 4571

APPENDIX A: SPH DATA SET DETAILS

A1 Technical details

We use three Nvidia GeForce GTX1080 Ti (11 GB) GPUs for the generation of SPH collision data. The total computation time can be split up into pre-processing, SPH simulations, and post-processing. Pre-processing and post-processing are performed on the CPU, whereas SPH simulations are performed on the GPU. The average computation times per simulation (average over 10 164 simulations) in these categories are 169 s, 42 min, and 4.8 s using a single GPU, which resulted in approximately 317 GPU days for the entire SPH data set. Similar computation times hold for the *N*-body data set (see Section 2.1.1) from Burger et al. (2020b).

The relatively large number of data points (one data point corresponds to one simulation) does not allow for high-resolution runs due to hardware limitations. We perform data reduction on the fly to significantly reduce the memory footprint from approximately 200 TB of raw data. After post-processing, individual simulation runs require less than 2 MB of storage space. The complete data-set requires about 12.2 GB of storage space.

We performed a total of 10 794 SPH simulations. 630 of those runs failed, mostly because they ran into numerical issues either in the setup script or during the actual simulation with `miluphcuda`. The density of failed runs is higher in the disruption regime compared to other regimes due to excessively small time-steps (via the adaptive time integration). This tends to be more likely in the disruption regime, where very high pressures are more common. We end up with 10 164 valid simulations for our data set. Note that due to the inhomogeneous distribution of invalid simulations, the distribution of valid simulations is also somewhat inhomogeneous, leading to a slightly worse coverage of high-velocity, low-impact-angle scenarios (see Fig. A1). We keep configuration files of invalid simulations for possible future data analysis.

A2 Pre-collision spin

Let us consider a single rotating body, consisting of n SPH particles. The rotation axis can either be defined in spherical coordinates (radius r, azimuth ϕ, polar angle θ) or in Cartesian coordinates (x, y, z):

$$x = r\ \cos(\phi)\ \sin(\theta) \tag{A1}$$

$$y = r\ \sin(\phi)\ \sin(\theta) \tag{A2}$$

$$z = r\ \cos(\theta) \tag{A3}$$

In our case, r is either the length of the angular momentum vector $\mathbf{L}$, or the rotation period P_{rot}.

$$\vec{L} = \sum_{i=1}^{n} \vec{r}_i \times (m_i\, \vec{v}_i) \tag{A4}$$

Here, m_i, $\vec{r}_i$, and $\vec{v}_i$ refer to the masses, positions, and velocities of individual SPH particles w.r.t. the barycentre of the object. The critical rotation period $P_{\mathrm{rot,crit}}$ is defined such that a test mass at the surface of the (idealized spherical) body is weightless according to Kepler's third law:

$$P_{\mathrm{rot,crit}} = \sqrt{\frac{4\pi^2 r^3}{G\,m}} = \frac{2\pi}{\omega_{\mathrm{crit}}} \tag{A5}$$

Here, r, m, and G refer to the object radius, its mass, and the gravitational constant. For data set generation, the rotation speed (angular velocity) ω is randomly sampled between $\omega = 0$ and $\omega = 0.2 \times \omega_{\mathrm{crit}}$, whereas the rotation axis is randomly sampled in Cartesian coordinates.

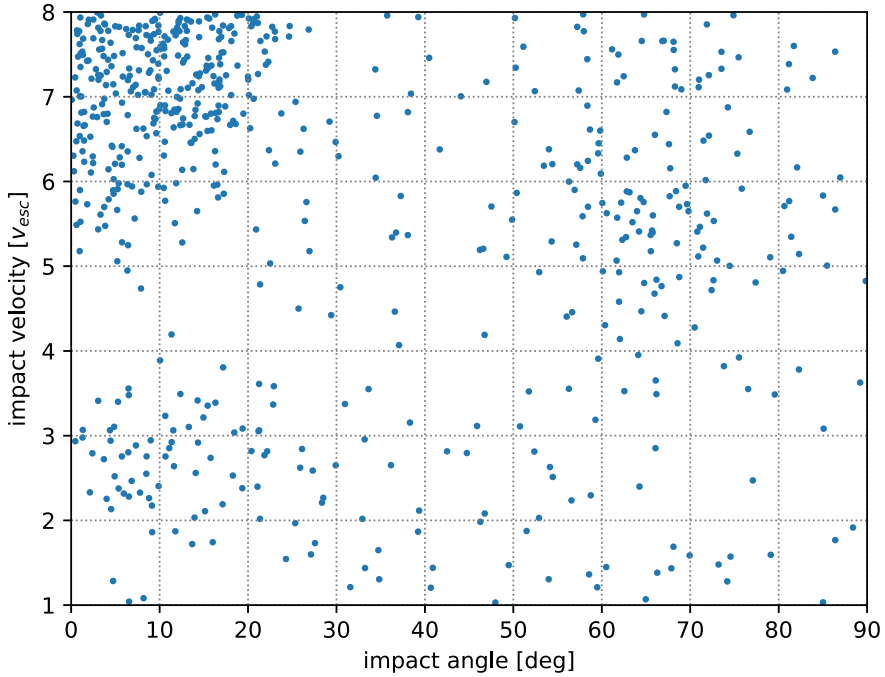


Figure A1. Distribution of invalid simulation runs. The chance that `miluphcuda` suffers from numerical issues is increased for collision scenarios with high velocities and low impact angles.

APPENDIX B: MACHINE LEARNING

B1 Rotation equivariance

An optional rotation module can be used as a pre- and post-processing step. The module rotates the system in the 3D domain, making the ML models equivariant to rotations. Thus, the models are geometrically consistent and do not require to spend their capacity on learning to be rotation equivariant. We propose to rotate and de-rotate the entire system before and after applying learnable modules.

$$y_i^0 = rot\left(\hat{y}_i^0, R_i\right) \tag{B1}$$

$$y_i^T = f\left(y_i^0, T, \Phi\right) \tag{B2}$$

$$\hat{y}_i^T = rot\left(y_i^T, R_i^{-1}\right) \tag{B3}$$

rot refers to the rotation module, f is an ML model with its respective learnable parameters Φ, and $\hat{y}$ refers to systems with any orientation, whereas y have a fixed orientation. We calculate a rotation matrix R_i for each data point such that the system is rotated into a fixed basis. The pre-rotation basis is given via the impact geometry and is calculated as follows (see Fig. 3):

$$\vec{e}_0 = \frac{\vec{v}}{|\vec{v}|} \qquad \vec{e}_1 = \frac{\vec{e}_0 \times \vec{r}}{|\vec{e}_0 \times \vec{r}|} \qquad \vec{e}_2 = \frac{\vec{e}_0 \times \vec{e}_1}{|\vec{e}_0 \times \vec{e}_1|} \tag{B4}$$

v and r are the relative velocity and relative distance vectors between the projectile and the target. The rotation matrix R_i is chosen as the inverse of the pre-rotation basis:

$$R_i = \begin{pmatrix} e_{00} & e_{01} & e_{02} \\ e_{10} & e_{11} & e_{12} \\ e_{20} & e_{21} & e_{22} \end{pmatrix}^{-1} \tag{B5}$$

The rotation module can be easily embedded into ML frameworks for collision treatment.

B2 Miscellaneous

Table B1 summarizes optimized hyperparameters for our deep learning models. Formally, model architectures can also be interpreted as hyperparameters. We tested a handful of different possible architectures and performed ablation studies before settling on the best-performing architecture, which is presented in the main text. For FFN, we tested a residual neural network approach and different activation functions.

We studied LSTMs (Hochreiter & Schmidhuber 1997; Hochreiter 1998; Gers et al. 1999) in-depth and investigated the following setups:

(i) Direct prediction of final states without Euler updates for studying performance of direct predictions versus relative predictions.
(ii) LSTM without forget gate for mitigating the vanishing gradient problem.

We also investigated several different GNN architectures (Scarselli et al. 2009; Defferrard et al. 2016; Kipf & Welling 2017). Our best architecture was a latent GNN (Dong et al. 2019), which slightly outperformed the FFN baseline. For the GNN, we investigated several architecture variations that use the initial SPH representation in combination with clustering approaches. These variations were mostly inspired by the architecture of Sanchez-Gonzalez et al. (2020). Increasing the number of graph nodes may require hierarchical graph structures (Martinkus et al. 2020; Rampásek & Wolf 2021) to account for learning long–range interactions such as gravity.

For the weight-tied residual network RES, we tested state updates via the Bulirsch–Stoer method (Stoer & Bulirsch 1980; Press, Teukolsky & Vetterling 1992) instead of the classical Euler update, but we did not observe any improvements.

Preliminary investigation of predicted water mass fractions did not allow for in-depth analysis of generalization to the low-water-content regime (i.e. $\zeta_{\text{water}} < 0.1$), because model predictions were too inaccurate.

We summarize the training and validation performance of our experiments in Table B2 . Results are consistent with the corresponding test results, which can be found in Table 5.

We provide pre-trained FFN and RES models for direct integration into existing N-body frameworks. We train these models using the entire development set (i.e. training + validation data) and report their performance in Table B3.

Table B1. Model-class hyperparameters for deep learning collision treatment approaches. Hyperparameters are optimized w.r.t. validation performance.

Method	Description	Value
FFN	Number of layers	8
FFN	Hidden state size	56
RES	Number of steps per simulated hour	4
RES	Number of layers for each module E, R, and D	3
RES	Hidden state size	64

B3 Incorporation of ML models into N-body frameworks

All our models can in principle be employed for collision treatment within existing N-body simulation frameworks. In the following, we list some important aspects related to implementation and technical details thereof:

(i) As our data are highly imbalanced (see Fig. 7 and Table 4), different use cases might require re-training the ML model using different data subsets. This benefit might be especially true for classification accuracy, which is sensitive to class-imbalances. Note that using the entire data set might lead to miss-classifications of collision outcome types (e.g. compare Fig. 7 versus Fig. 6). Also, depending on the use case, one might want to incorporate additional restrictions (such as imposing conservation laws) as post-processing step on top of model predictions. Further details are elaborated in Section 2.2.1.

(ii) Our models require consistent input features, produced by collisions that lie within our parameter space as defined in Table 1. Having consistent input features also includes applying the identical pre-processing pipeline (e.g. the numerical scaling of features) in analogy to the pre-processing pipeline that is used for model training. We believe that producing consistent input features is the most error-prone process when implementing ML methods for collision treatment. For sophisticated frameworks or if o.o.d. data points are expected to be common, one may even consider applying anomaly detection methods to check for potential invalid inputs.

(iii) Note that we define the onset of the collision process before objects come in direct contact in order to include tidal effects. This should be considered for triggering collision events in N-body frameworks. We propose taking two measurements, one when tidal interaction begins, and another one once the objects come into direct contact (cf. Fig. 3). We initialize the colliding bodies at a distance of $d_{\text{initial}} = f_i \times (R_t + R_p)$. R_{t} and R_{p} are the target and projectile radii, while the initial distance factor f_i is a hyperparameter (between $f_i = 3$ and $f_i = 7$ for our data). Therefore, reasonable estimates of R_{t} and R_{p} might be required to define the first measurement in N-body simulations.

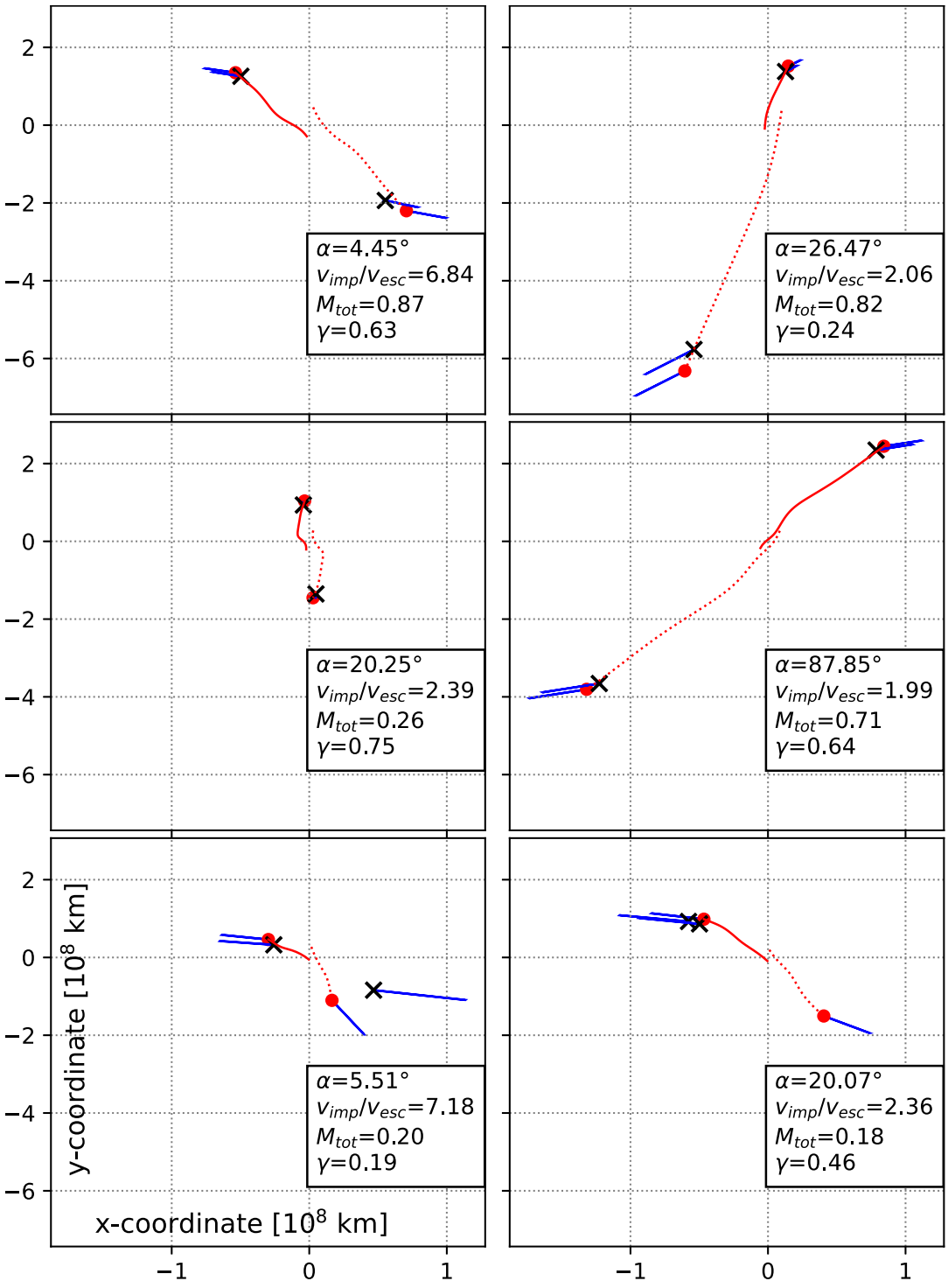


Figure B1. Visualization of predicted intermediate states (in red) of our RES model for four different data points of the o.o.d. test set. Each plot depicts two predicted trajectories, where one corresponds to the largest remnant (solid) and the other one to the second-largest remnant (dotted). Final positions are indicated by red dots (predictions) and black crosses (ground truth). Ground truth initial positions are indicated by the starting point of the respective predicted trajectories. Final velocity vectors (for both prediction and ground truth) are indicated by blue arrows for all objects. Note that projectiles are flying into the negative y-direction initially and the x–y plane is the main collision plane. The labels indicate the initial conditions (mass in units of 10^{25} kg, cf. Table 1). We find that z-components (which are caused by rotating bodies) were quite significant in many simulations. Prediction performance in the z-component is comparable to the x- and y-components. We observe spatio-temporally continuous transitions from initial to final states, possibly indicating a correlation to the temporal evolution of the physical system to a certain extent. The first four panels were cherry-picked for predicted positions in the x–y plane. The last two panels visualize typical failure cases of model predictions, where positions and velocities of the second-largest remnants are poorly approximated.

Table B2. Training and validation performance (RMSE and balanced accuracy, both measured on the final state) of different approaches for planetary collision treatment. Classification is performed as a post-processing step on top of predicted masses. We assume the SPH simulation data to be the ground truth. For single-task learning (lower half of the table), each entry corresponds to the performance of individually trained ML models and column headers indicate optimized tasks, respectively. We use data from Timpe et al. (2020b) to obtain results in the lower half. Best results are indicated in bold, whereas ∗ indicate statistically significant results according to a Wilcoxon test (comparing FFN and RES). Our proposed RES model outperforms the other baseline methods in most experiments.

Method	Split	Mass	Material	Position	Velocity	Total	Accuracy
PIM	Training	$0.1388^{+0.0008}_{-0.0012}$	$0.0619^{+0.0002}_{-0.0001}$	$0.4055^{+0.0010}_{-0.0008}$	$0.4411^{+0.0022}_{-0.0018}$	$0.3619^{+0.0010}_{-0.0007}$	$0.1667^{+0.0000}_{+0.0000}$
LIN	Training	$0.0441^{+0.0003}_{-0.0003}$	$0.0280^{+0.0001}_{-0.0001}$	$0.1617^{+0.0005}_{-0.0006}$	$0.1579^{+0.0011}_{-0.0015}$	$0.1375^{+0.0006}_{-0.0007}$	$0.2296^{+0.0068}_{-0.0088}$
FFN	Training	$0.0060^{+0.0003}_{-0.0004}$	$0.0115^{+0.0002}_{-0.0001}$	$0.0403^{+0.0007}_{-0.0007}$	$0.0446^{+0.0006}_{-0.0007}$	$0.0367^{+0.0005}_{-0.0005}$	$0.4422^{+0.0444}_{-0.0416}$
RES	Training	$\mathbf{*0.0047^{+0.0001}_{-0.0001}}$	$\mathbf{*0.0108^{+0.0002}_{-0.0003}}$	$\mathbf{*0.0272^{+0.0002}_{-0.0003}}$	$\mathbf{*0.0339^{+0.0003}_{-0.0006}}$	$\mathbf{*0.0272^{+0.0002}_{-0.0002}}$	$\mathbf{*0.5234^{+0.0089}_{-0.0182}}$
PIM	Validation	$0.1388^{+0.0048}_{-0.0032}$	$0.0619^{+0.0004}_{-0.0008}$	$0.4056^{+0.0031}_{-0.0041}$	$0.4411^{+0.0071}_{-0.0087}$	$0.3619^{+0.0030}_{-0.0041}$	$0.1667^{+0.0000}_{+0.0000}$
LIN	Validation	$0.0443^{+0.0012}_{-0.0009}$	$0.0280^{+0.0005}_{-0.0008}$	$0.1619^{+0.0024}_{-0.0018}$	$0.1581^{+0.0066}_{-0.0042}$	$0.1377^{+0.0033}_{-0.0021}$	$0.2380^{+0.0485}_{-0.0220}$
FFN	Validation	$0.0066^{+0.0003}_{-0.0004}$	$0.0117^{+0.0002}_{-0.0001}$	$0.0420^{+0.0016}_{-0.0017}$	$0.0458^{+0.0023}_{-0.0032}$	$0.0379^{+0.0012}_{-0.0021}$	$0.4558^{+0.0328}_{-0.0334}$
RES	Validation	$\mathbf{*0.0052^{+0.0003}_{-0.0004}}$	$\mathbf{*0.0111^{+0.0003}_{-0.0004}}$	$\mathbf{*0.0320^{+0.0019}_{-0.0012}}$	$\mathbf{*0.0382^{+0.0015}_{-0.0022}}$	$\mathbf{*0.0309^{+0.0014}_{-0.0010}}$	$\mathbf{*0.5381^{+0.0438}_{-0.0631}}$
PIM	Training, Timpe	$0.2080^{+0.0003}_{-0.0003}$	$0.1933^{+0.0014}_{-0.0009}$	$0.3598^{+0.0019}_{-0.0014}$	$0.2897^{+0.0017}_{-0.0017}$	–	$0.1667^{+0.0000}_{+0.0000}$
LIN	Training, Timpe	$0.0528^{+0.0005}_{-0.0004}$	$0.0493^{+0.0004}_{-0.0003}$	$0.2040^{+0.0013}_{-0.0014}$	$0.1691^{+0.0009}_{-0.0007}$	–	$0.3019^{+0.0063}_{-0.0037}$
FFN	Training, Timpe	$0.0124^{+0.0003}_{-0.0002}$	$0.0145^{+0.0002}_{-0.0002}$	$0.0757^{+0.0017}_{-0.0009}$	$0.0589^{+0.0002}_{-0.0003}$	–	$0.4172^{+0.0040}_{-0.0040}$
RES	Training, Timpe	$\mathbf{*0.0112^{+0.0004}_{-0.0004}}$	$\mathbf{*0.0138^{+0.0005}_{-0.0006}}$	$\mathbf{*0.0658^{+0.0010}_{-0.0008}}$	$\mathbf{*0.0548^{+0.0007}_{-0.0007}}$	–	$\mathbf{*0.4252^{+0.0063}_{-0.0059}}$
PIM	Validation, Timpe	$0.2080^{+0.0013}_{-0.0011}$	$0.1933^{+0.0035}_{-0.0058}$	$0.3598^{+0.0056}_{-0.0077}$	$0.2897^{+0.0066}_{-0.0070}$	–	$0.1667^{+0.0000}_{+0.0000}$
LIN	Validation, Timpe	$0.0529^{+0.0015}_{-0.0009}$	$0.0495^{+0.0014}_{-0.0005}$	$0.2044^{+0.0046}_{-0.0064}$	$0.1695^{+0.0043}_{-0.0046}$	–	$0.3010^{+0.0181}_{-0.0105}$
FFN	Validation, Timpe	$0.0137^{+0.0010}_{-0.0008}$	$0.0155^{+0.0009}_{-0.0006}$	$0.0834^{+0.0056}_{-0.0025}$	$\mathbf{0.0642^{+0.0024}_{-0.0019}}$	–	$0.4196^{+0.0163}_{-0.0116}$
RES	Validation, Timpe	$\mathbf{*0.0130^{+0.0013}_{-0.0011}}$	$\mathbf{0.0152^{+0.0005}_{-0.0003}}$	$\mathbf{0.0826^{+0.0054}_{-0.0035}}$	$0.0649^{+0.0028}_{-0.0017}$	–	$\mathbf{0.4265^{+0.0059}_{-0.0049}}$

Table B3. Performance of provided pre-trained models.

Method	Split	Mass	Material	Position	Velocity	Total	Accuracy
FFN	Development	0.0055	0.0111	0.0399	0.0442	0.0364	0.4535
RES	Development	0.0046	0.0108	0.0273	0.0341	0.0273	0.5035
FFN	o.o.d. test	0.0119	0.0120	0.0483	0.0432	0.0405	0.5593
RES	o.o.d. test	0.0104	0.0125	0.0375	0.0425	0.0357	0.5091

(iv) Collisions can happen at arbitrary orientations. We provide a rotation module to account for rotation-equivariance of our ML models. We recommend re-training ML models using data augmentation (i.e. randomly rotating the systems) in combination with the rotation module as a sanity check for guaranteeing rotation-equivariance. Note that all ML models in this paper were trained without using the rotation module, i.e. projectiles flying into the negative y-direction initially, where the main collision plane is the x–y plane. Our data pre-processing is also specifically adjusted for this fixed-orientation setup, i.e. different axes getting scaled differently in order to account for the variability in the respectiv e directions.

(v) Our ML models can be ran on CPUs if using GPUs is unfeasible or none are available. We estimate that CPU inference times $\tau_{\mathrm{i,ML}}$ are in the order of 1–10 s.

(vi) As discussed in Section 2.2.5, extracting reliable post-collision rotation states from our data remains an open issue, potentially restricting the full usability of our ML models in *N*-body simulations, specifically for full tracking of rotation states over several collisions. However, once post-collision rotation states can be extracted reliably from collision simulations, they can be easily incorporated into our multi-task regression objective.

This paper has been typeset from a TEX/LATEX file prepared by the author.

7. Conclusion

7.1. Summary & Interpretation

In this thesis, we conducted an exhaustive study of collisions within planetary systems by employing advanced computational modeling techniques. We developed and utilized methods that effectively manage large-scale data in a systematic and insightful manner. Key findings from this research include precise estimations of collision geometries, detailed analyses of collision outcomes, and the exploration of complex relationships through parameter studies. Additionally, we advanced the field by enhancing computational modeling techniques for numerical simulations and machine learning, while also generating extensive, high-quality datasets.

Our results imply that modern computational methods provide a scalable and efficient framework for both generating and analyzing large-scale data in the field of theoretical astrophysics for planetary collisions. These methods have proven adept at accurately capturing the complex physical interactions that are inherent in such collisions. Our results support the efficacy of modern ML and DL approaches, affirming that data-driven models have secured a significant role within a field traditionally driven by analytic modeling methods.

From an astrophysical perspective, our findings confirm and support well-established theories and interpretations regarding planet formation and evolution. In contrast to previous work, we underlined the importance of large-scale parameter studies, which, when distilled through modern ML methods, reveal insights not previously accessible with traditional approaches. However, we acknowledge that the integration of these advanced computational methods introduces an additional layer of complexity, which might not be immediately intuitive or accessible to many astronomers.

We acknowledge that our work presented in Chapter 3 has some conceptual weaknesses. We did not thoroughly validate the distribution of measured initial conditions obtained from our genetic algorithm (GA) by utilizing results from

long-term n-body simulations. Moreover, we did not perform ablation studies for how strongly the overall outcomes were affected by the dynamics of the GA versus the n-body simulations (for measuring fitness). We thus can not guarantee that our approach led to an unbiased estimation of plausible initial conditions. It is possible that the GA induced a selection bias towards trivial solutions while somewhat neglecting orbital dynamics of projectiles.

7.2. Implications & Outlook

Although numerical simulations are well-established in modern astrophysics, usage of ML just started to gain traction recently. In contrast to other fields like computer vision and medicine, ML in astrophysics has still a long way to go in order to catch up with the latest methodological innovations.

Our work can be extended to different scales ranging from micro-meteorites to collisions involving gas giants or stars. Moreover, the usefulness of our approach presented in Chapter 6 still has to be verified in the context of planetary evolution studies. Depending on available hardware and the specific domain, ML methods can be further improved and adapted to meet certain criteria such as uncertainty estimation of predictions or improved predictive accuracy. Our data can also be used for generative modeling, thereby serving as a foundation for a new kind of simulation environment.

Adaptation to handling ever-growing amounts of data requires more interdisciplinary cooperation, as well as specialized education of individual researchers. The insight that completely different fields suffer from identical data-analysis problems might act as a worthwhile stepping stone for future innovation.

I am confident that the research presented herein is able to inspire the next generation of scientists to explore the hidden treasures of planet formation. While I may not view myself as a leading figure in astrophysics, I do consider myself as one of the few people who have ventured to integrate machine learning with the aim of providing a fresh perspective on new research pathways. Although my personal journey had significant detours and felt hopeless at times, my deep fascination for nature always kept me going towards something meaningful.

Bibliography

[1] Yutaka Abe. "Thermal Evolution and Chemical Differentiation of the Terrestrial Magma Ocean". In: *Evolution of the Earth and Planets*. American Geophysical Union (AGU), 1993, pp. 41–54. ISBN: 9781118666692. DOI: https://doi.org/10.1029/GM074p0041. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/GM074p0041. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/GM074p0041.

[2] I. Aizenberg, N. N. Aizenberg, and J. P. Vandewalle. *Multi-Valued and Universal Binary Neurons – Theory, Learning and Applications*. Boston, MA, USA: Springer US, 2000. ISBN: 978-1-4757-3115-6. DOI: 10.1007/978-1-4757-3115-6.

[3] ALMA (ESO/NAOJ/NRAO) / Brogan et al. *ALMA image of the protoplanetary disc around HL Tauri*. https://www.eso.org/public/images/eso1436a/. image. Accessed: 2025-05-30. 2014.

[4] H. Alfvén. "Existence of Electromagnetic-Hydrodynamic Waves". In: *Nature* 150.3805 (Oct. 1942), pp. 405–406. DOI: 10.1038/150405d0.

[5] Philip J. Armitage. *Astrophysics of Planet Formation*. 2nd ed. Cambridge University Press, 2020.

[6] Erik Asphaug, Craig B. Agnor, and Quentin Williams. "Hit-and-run Planetary Collisions". In: *Nature* 439.7073 (2006), pp. 155–160. ISSN: 1476-4687. DOI: 10.1038/nature04311. URL: https://doi.org/10.1038/nature04311.

[7] Erik Asphaug et al. "Collision Chains among the Terrestrial Planets. III. Formation of the Moon". In: *The Planetary Science Journal* 2.5, 200 (Oct. 2021), p. 200. DOI: 10.3847/PSJ/ac19b2. arXiv: 2110.00222 [astro-ph.EP].

[8] Josh Barnes and Piet Hut. "A hierarchical O(N log N) force-calculation algorithm". In: *Nature* 324.6096 (1986), pp. 446–449. ISSN: 1476-4687. DOI: 10.1038/324446a0. URL: https://doi.org/10.1038/324446a0.

[9] G. Benettin et al. "Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems; a Method for Computing All of Them. Part 1: Theory". In: *Meccanica* 15.1 (1980). This paper provides a foundational method for numerically computing Lyapunov exponents, widely used in chaos analysis for dynamical systems., pp. 9–20. DOI: 10.1007/BF02128236.

[10] Willy Benz and Erik Asphaug. "Catastrophic Disruptions Revisited". In: *Icarus* 142.1 (1999), pp. 5–20. ISSN: 0019-1035. DOI: https://doi.org/10.1006/icar.1999.6204. URL: https://www.sciencedirect.com/science/article/pii/S0019103599962048.

[11] R. D. Blandford and D. G. Payne. "Hydromagnetic flows from accretion discs and the production of radio jets". In: *Monthly Notices of the Royal Astronomical Society* 199.4 (Aug. 1982), pp. 883–903. ISSN: 0035-8711. DOI: 10.1093/mnras/199.4.883. eprint: https://academic.oup.com/mnras/article-pdf/199/4/883/2880142/mnras199-0883.pdf. URL: https://doi.org/10.1093/mnras/199.4.883.

[12] Jeff Bryant and Oleksandr Pavlyk. *Kepler's Second Law*. image. Contributed by: Jeff Bryant, Stephen Wolfram, Oleksandr Pavlyk. Open content licensed under CC BY-NC-SA, Accessed: 2024-11-22. 2007. URL: https://demonstrations.wolfram.com/KeplersSecondLaw/.

[13] Roland Bulirsch and Josef Stoer. "Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods". In: *Numerische Mathematik* 8 (1966), pp. 1–13.

[14] C. Burger, Á. Bazsó, and C. M. Schäfer. "Realistic collisional water transport during terrestrial planet formation. Self-consistent modeling by an N-body-SPH hybrid code". In: *Astronomy & Astrophysics* 634, A76 (2020), A76. DOI: 10.1051/0004-6361/201936366.

[15] C. Burger, T. I. Maindl, and C. M. Schäfer. "Transfer, loss and physical processing of water in hit-and-run collisions of planetary embryos". In: *Celestial Mechanics and Dynamical Astronomy* 130.1, 2 (Jan. 2018), p. 2. DOI: 10.1007/s10569-017-9795-3. arXiv: 1710.03669 [astro-ph.EP].

[16] Christoph Burger, Thomas I. Maindl, and Christoph Schäfer. "Water delivery to dry protoplanets by hit-and-run collisions". In: *IAU Symposium*. Ed. by Bruce G. Elmegreen, L. Viktor Tóth, and Manuel Güdel. Vol. 345. IAU Symposium. 2020, pp. 287–288. DOI: 10.1017/S1743921318008621. arXiv: 2002.00231 [astro-ph.EP].

[17] Renyue Cen and Jeremiah P. Ostriker. "Galaxy Formation and Physical Bias". In: *Astrophysical Journal Letters* 399 (Nov. 1992), p. L113. DOI: 10.1086/186620.

[18] John E. Chambers. "Planetary accretion in the inner Solar System". In: *Earth and Planetary Science Letters* 223.3 (2004), pp. 241–252. ISSN: 0012-821X. DOI: https://doi.org/10.1016/j.epsl.2004.04.031. URL: https://www.sciencedirect.com/science/article/pii/S0012821X04002791.

[19] Subrahmanyan Chandrasekhar. *An Introduction to the Study of Stellar Structure*. Chicago: University of Chicago Press, 1939.

[20] Chrenko, O., Brož, M., and Lambrechts, M. "Eccentricity excitation and merging of planetary embryos heated by pebble accretion". In: *A&A* 606 (2017), A114. DOI: 10.1051/0004-6361/201731033. URL: https://doi.org/10.1051/0004-6361/201731033.

[21] Christopher F. Chyba, Paul J. Thomas, and Kevin J. Zahnle. "The 1908 Tunguska explosion: atmospheric disruption of a stony asteroid". In: *Nature* 361.6407 (Jan. 1993), pp. 40–44. ISSN: 1476-4687. DOI: 10.1038/361040a0. URL: https://doi.org/10.1038/361040a0.

[22] Matthew S. Clement et al. *Formation of Terrestrial Planets*. 2024. arXiv: 2411.03453 [astro-ph.EP]. URL: https://arxiv.org/abs/2411.03453.

[23] Wikimedia Commons. *Antikythera Mechanism Fragment A (Front)*. Accessed: 2025-06-05. 2025. URL: https://en.wikipedia.org/wiki/Antikythera_mechanism#/media/File:Antikythera_Fragment_A_(Front).webp.

[24] Samuele Crespi et al. "Protoplanet collisions: Statistical properties of ejecta". In: *Monthly Notices of the Royal Astronomical Society* 508.4 (Oct. 2021), pp. 6013–6022. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2951. eprint: https://academic.oup.com/mnras/article-pdf/508/4/6013/41062970/stab2951.pdf. URL: https://doi.org/10.1093/mnras/stab2951.

[25] *The JHUAPL Planetary Impact Lab (PIL): Capabilities and initial results*. Vol. 2019 15th Hypervelocity Impact Symposium. Hypervelocity Impact Symposium. Apr. 2019, V001T06A004. DOI: 10.1115/HVIS2019-084. eprint: https://asmedigitalcollection.asme.org/hvis/proceedings-pdf/HVIS2019/883556/V001T06A004/6551009/v001t06a004-hvis2019-084.pdf. URL: https://doi.org/10.1115/HVIS2019-084.

[26] Stanley F. Dermott and Carl Sagan. "Tidal effects of disconnected hydrocarbon seas on Titan". In: *Nature* 374.6519 (1995), pp. 238–240. ISSN: 1476-4687. DOI: 10.1038/374238a0. URL: https://doi.org/10.1038/374238a0.

[27] R. Dvorak, F. Freistetter, and J. Kurths. *Chaos and Stability in Planetary Systems*. Lecture Notes in Physics. Springer Berlin Heidelberg, 2009. ISBN: 9783540814078. URL: https://books.google.at/books?id=StARswEACAAJ.

[28] Jeffrey L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: https://www.sciencedirect.com/science/article/pii/036402139090002E.

[29] James Ferguson et al. *The Complexity to be Described by the Geocentric Model*. Engraved for the Encyclopaedia Britannica (1st Edition, 1771; facsimile reprint 1971), Volume 1, Fig. 2 of Plate XL facing page 449. Public Domain, Uploaded: 2010-07-14, Accessed: 2025-06-05. 1771. URL: https://upload.wikimedia.org/wikipedia/commons/0/0e/Cassini_apparent.jpg.

[30] Sylvio Ferraz-Mello. *Chaotic Dynamics in Planetary Systems*. 1st ed. Springer Praxis Books. 1 b/w illustrations. Springer Cham, 2023, pp. XI, 167. ISBN: 978-3-031-45815-6. DOI: 10.1007/978-3-031-45816-3. URL: https://doi.org/10.1007/978-3-031-45816-3.

[31] Joel H. Ferziger, Milovan Perić, and Robert L. Street. *Computational Methods for Fluid Dynamics*. 4th ed. Cham: Springer, 2020. ISBN: 978-3-319-99691-2. DOI: 10.1007/978-3-319-99693-6.

[32] Tony Freeth et al. "A Model of the Cosmos in the Ancient Greek Antikythera Mechanism". In: *Scientific Reports* 11.5821 (2021). DOI: 10.1038/s41598-021-84310-w. URL: https://www.nature.com/articles/s41598-021-84310-w.

[33] William Froude. "The Laws of Fluid Resistance". In: *Science Lectures at South Kensington*. Vol. II. London: Macmillan & Company, 1879, pp. 88–121.

[34] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251. URL: https://doi.org/10.1007/BF00344251.

[35] L. Gasperini et al. "A possible impact crater for the 1908 Tunguska Event". In: *Terra Nova* 19.4 (2007), pp. 245–251. DOI: https://doi.org/10.1111/j.1365-3121.2007.00742.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-3121.2007.00742.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-3121.2007.00742.x.

[36] Carl Friedrich Gauss. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum*. Latin. Hamburgi: Frid. Perthes et I. H. Besser, 1809.

[37] Hidenori Genda et al. “Ejection of iron-bearing giant-impact fragments and the dynamical and geochemical influence of the fragment re-accretion”. In: *Earth and Planetary Science Letters* 470 (2017), pp. 87–95. DOI: 10.1016/j.epsl.2017.04.035. arXiv: 1704.05251 [astro-ph.EP].

[38] Amitabha Ghosh and Harry Y McSween Jr. “A thermal model for the differentiation of asteroid 4 Vesta, based on radiogenic heating”. In: *Icarus* 134.2 (1998), pp. 187–206.

[39] R. A. Gingold and J. J. Monaghan. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389. ISSN: 0035-8711. DOI: 10.1093/mnras/181.3.375. eprint: https://academic.oup.com/mnras/article-pdf/181/3/375/3104055/mnras181-0375.pdf. URL: https://doi.org/10.1093/mnras/181.3.375.

[40] Glasgow (Scotland). *Lectures on Naval Architecture and Engineering: With Catalogue of the Exhibition*. The catalogue (104 p.) has a special title page: Catalogue with supplement, Glasgow, Printed by R. Anderson, 1880. London and Glasgow: W. Collins, Sons, & Co. (Limited), 1881, pp. iv, 2, 245, 104. URL: https://catalog.hathitrust.org/Record/002016455,https://hdl.handle.net/2027/mdp.39015006391257.

[41] Peter Goldreich and Steven Soter. “Q in the solar system”. In: *Icarus* 5.1 (1966), pp. 375–389. ISSN: 0019-1035. DOI: https://doi.org/10.1016/0019-1035(66)90051-0. URL: https://www.sciencedirect.com/science/article/pii/0019103566900510.

[42] R. Gomes et al. “Origin of the cataclysmic Late Heavy Bombardment period of the terrestrial planets”. In: *Nature* 435.7041 (2005), pp. 466–469. ISSN: 1476-4687. DOI: 10.1038/nature03676. URL: https://doi.org/10.1038/nature03676.

[43] L. Greengard and V. Rokhlin. “A Fast Algorithm for Particle Simulations”. In: *Journal of Computational Physics* 73 (1987), pp. 325–348. DOI: 10.1016/0021-9991(87)90140-9.

[44] Alex N. Halliday and Robin M. Canup. “The Accretion of Planet Earth”. In: *Nature Reviews Earth & Environment* 4.1 (2023), pp. 19–35. ISSN: 2662-138X. DOI: 10.1038/s43017-022-00370-0. URL: https://doi.org/10.1038/s43017-022-00370-0.

[45] A. Hanslmeier and R. Dvorak. “Application of Lie-Series to Numerical Integration in Celestial Mechanics”. In: *Applications of Modern Dynamics to Celestial Mechanics and Astrodynamics*. Ed. by Victor Szebehely. Dordrecht: Springer Netherlands, 1982, pp. 345–345. ISBN: 978-94-009-7793-8.

[46] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[47] Sebastian von Hoerner. “Die numerische Integration des n-Körper-Problems für Sternhaufen”. In: *Zeitschrift für Astrophysik* 50 (1960), pp. 184–214.

[48] K A Holsapple. “The Scaling of Impact Processes in Planetary Sciences”. In: *Annual Review of Earth and Planetary Sciences* 21.Volume 21, 1993 (1993), pp. 333–373. ISSN: 1545-4495. DOI: https://doi.org/10.1146/annurev.ea.21.050193.002001. URL: https://www.annualreviews.org/content/journals/10.1146/annurev.ea.21.050193.002001.

[49] K. A. Holsapple and R. M. Schmidt. *Centrifuge Crater Scaling Experiment II: Material Strength Effects*. Tech. rep. DNA 4999 0Z. Interim Report for Period 31 January 1978-31 January 1979. Seattle, Washington: Boeing Aerospace Company, 1979. URL: https://apps.dtic.mil/sti/tr/pdf/ADA106424.pdf.

[50] D. Padgett (IPAC/Caltech) et al. *Young Stellar Disks in Infrared*. image. Release ID: 1999-05. Public domain image from NASA, Accessed: 2024-11-15. 1999. URL: https://hubblesite.org/contents/media/images/1999/05/771-Image.html?news=true.

[51] A. G. Ivakhnenko and V. G. Lapa. *Cybernetic Predicting Devices*. New York, NY, USA: CCM Information Corporation, 1965. ISBN: N/A.

[52] Harold Jeffreys. *The Earth: Its Origin, History and Physical Constitution*. 3rd. Cambridge: Cambridge University Press, 1952.

[53] Johns Hopkins University Applied Physics Laboratory. *Double Asteroid Redirection Test (DART) Mission*. https://dart.jhuapl.edu/Mission/. Accessed: 2025-06-03. 2022. URL: https://dart.jhuapl.edu/Mission/.

[54] Michael I. Jordan. "Chapter 25 - Serial Order: A Parallel Distributed Processing Approach". In: *Neural-Network Models of Cognition*. Ed. by John W. Donahoe and Vivian Packard Dorsel. Vol. 121. Advances in Psychology. North-Holland, 1997, pp. 471–495. DOI: https://doi.org/10.1016/S0166-4115(97)80111-2. URL: https://www.sciencedirect.com/science/article/pii/S0166411597801112.

[55] NASA / JPL / MPS / DLR / IDA / Björn Jónsson. *Asteroid Vesta in Natural Color*. https://planetary.s3.amazonaws.com/assets/images/9-small-bodies/2013/20130819_vesta_rgb_20110724_0835.png. image. Accessed: 2025-05-29. 2011.

[56] Neal Katz. "Dissipational Galaxy Formation. II. Effects of Star Formation". In: *Astrophysical Journal* 391 (June 1992), p. 502. DOI: 10.1086/171366.

[57] Neal Katz and James E. Gunn. "Dissipational Galaxy Formation. I. Effects of Gasdynamics". In: *Astrophysical Journal* 377 (Aug. 1991), p. 365. DOI: 10.1086/170367.

[58] J. A. Kegerreis et al. "Consequences of Giant Impacts on Early Uranus for Rotation, Internal Structure, Debris, and Atmospheric Erosion". In: *The Astrophysical Journal* 861.1 (2018), p. 52. DOI: 10.3847/1538-4357/aac725. URL: https://dx.doi.org/10.3847/1538-4357/aac725.

[59] Jacob A Kegerreis et al. "Immediate Origin of the Moon as a Post-impact Satellite". In: *The Astrophysical Journal Letters* 937.2 (2022). Figure adapted from http://icc.dur.ac.uk/giant_impacts/moon_wide_orbit_slice.mp4, Accessed: 2025-01-24, p. L40.

[60] Jacob A Kegerreis et al. "Origin of Mars's moons by disruptive partial capture of an asteroid". In: *arXiv preprint arXiv:2407.15936* (2024).

[61] Thomas Kenkmann et al. "Experimental impact cratering: A summary of the major results of the MEMIN research unit". In: *Meteoritics & Planetary Science* 53.8 (2018), pp. 1543–1568. DOI: https://doi.org/10.1111/maps.13048. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/maps.13048. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/maps.13048.

[62] Thomas Kenkmann et al. *Image from: Experimental impact cratering: A summary of the major results of the MEMIN research unit*. Image from article published in Meteoritics & Planetary Science. Adapted from Figure 1e from the article. 2018. DOI: 10.1111/maps.13048. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/maps.13048.

[63] Hiroshi Kinoshita, Haruo Yoshida, and Hiroshi Nakai. "Symplectic integrators and their application to dynamical astronomy". In: *Celestial Mechanics and Dynamical Astronomy* 50.1 (1990), pp. 59–71. ISSN: 1572-9478. DOI: 10.1007/BF00048986. URL: https://doi.org/10.1007/BF00048986.

[64] Anatoly A. Klypin, Sebastian Trujillo-Gomez, and Joel Primack. "Dark Matter Halos in the Standard Cosmological Model: Results from the Bolshoi Simulation". In: *The Astrophysical Journal* 740.2 (2011), p. 102. DOI: 10.1088/0004-637X/740/2/102.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[66] Wilhelm Kutta. "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen". In: *Zeitschrift für Mathematik und Physik* 46 (1901), pp. 435–453.

[67] Kurt Lambeck. *The Earth's Variable Rotation: Geophysical Causes and Consequences*. Cambridge Monographs on Mechanics. Cambridge University Press, 1980.

[68] Helmut Lammer and Michel Blanc. "From Disks to Planets: The Making of Planets and Their Early Atmospheres. An Introduction". In: *Space Science Reviews* 214.2 (2018), p. 60. ISSN: 1572-9672. DOI: 10.1007/s11214-017-0433-x. URL: https://doi.org/10.1007/s11214-017-0433-x.

[69] Y. LeCun, Y. Bengio, and G. E. Hinton. "Deep Learning". In: *Nature* 521 (2015), pp. 436–444. DOI: 10.1038/nature14539. URL: https://doi.org/10.1038/nature14539.

[70] Yann LeCun and Yoshua Bengio. "Convolutional networks for images, speech, and time series". In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998, 255–258. ISBN: 0262511029.

[71] D. N. C. Lin and John Papaloizou. "On the Tidal Interaction between Protoplanets and the Protoplanetary Disk. III. Orbital Migration of Protoplanets". In: *Astrophysical Journal* 309 (Oct. 1986), p. 846. DOI: 10.1086/164653.

[72] Jack J. Lissauer. "Planet formation." In: *Annual Rev. Astron. Astrophys.* 31 (Jan. 1993), pp. 129–174. DOI: 10.1146/annurev.aa.31.090193.001021.

[73] L. B. Lucy. "A numerical approach to the testing of the fission hypothesis." In: *Astronomical Journal* 82 (Dec. 1977), pp. 1013–1024. DOI: 10.1086/112164.

[74] Thomas Maindl et al. "A survey of collision outcomes during planet formation: water transport and loss". In: *European Planetary Science Congress*. Sept. 2018, EPSC2018-995, EPSC2018–995.

[75] Gérard Manhes et al. "Lead isotope study of basic-ultrabasic layered complexes: Speculations about the age of the earth and primitive mantle characteristics". In: *Earth and Planetary Science Letters* 47.3 (1980), pp. 370–382. ISSN: 0012-821X. DOI: https://doi.org/10.1016/0012-821X(80)90024-2. URL: https://www.sciencedirect.com/science/article/pii/0012821X80900242.

[76] Christopher F McKee and Eve C Ostriker. "Theory of star formation". In: *Annu. Rev. Astron. Astrophys.* 45.1 (2007), pp. 565–687.

[77] H. J. Melosh. *Impact Cratering: A Geologic Process*. Vol. 11. Oxford Monographs on Geology and Geophysics. Lunar and Planetary Laboratory, University of Arizona. Oxford University Press, 1989. ISBN: 0195042840.

[78] Tom M. Mitchell. *Machine Learning*. 1st. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN: 0070428077.

[79] J. J. Monaghan. "Smoothed Particle Hydrodynamics". In: *Annual Review of Astronomy and Astrophysics* 30 (1992), pp. 543–574. DOI: 10.1146/annurev.aa.30.090192.002551.

[80] Christoph Mordasini and Remo Burn. *Planet formation – observational constraints, physical processes, and compositional patterns*. 2024. arXiv: 2404.15555 [astro-ph.EP]. URL: https://arxiv.org/abs/2404.15555.

[81] Carl D. Murray and Stanley F. Dermott. *Solar System Dynamics*. Cambridge University Press, 2000.

[82] NASA Astromaterials Acquisition and Curation Office. *Cosmic Dust Collection Image*. https://curator.jsc.nasa.gov/dust/. image. Accessed: 2025-05-29. 2024.

[83] Francis Nimmo and Thorsten Kleine. "Early Differentiation and Core Formation". In: *The Early Earth*. American Geophysical Union (AGU), 2015. Chap. 5, pp. 83–102. ISBN: 9781118860359. DOI: https://doi.org/10.1002/9781118860359.ch5. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/9781118860359.ch5. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/9781118860359.ch5.

[84] Nvidia Corporation. *Method and Apparatus for Processing Graphics Primitives*. U.S. Patent. Issued. 1999.

[85] Sijme-Jan Paardekooper and Anders Johansen. "Giant Planet Formation and Migration". In: *Space Science Reviews* 214.1 (2018), p. 38. ISSN: 1572-9672. DOI: 10.1007/s11214-018-0472-y. URL: https://doi.org/10.1007/s11214-018-0472-y.

[86] J. C. B. Papaloizou and J. E. Pringle. "The dynamical stability of differentially rotating discs with constant specific angular momentum". In: *Monthly Notices of the Royal Astronomical Society* 208 (June 1984), pp. 721–750. DOI: 10.1093/mnras/208.4.721.

[87] J. C. B. Papaloizou and J. E. Pringle. "The dynamical stability of differentially rotating discs – II". In: *Monthly Notices of the Royal Astronomical Society* 213.4 (Apr. 1985), pp. 799–820. ISSN: 0035-8711. DOI: 10.1093/mnras/213.4.799. eprint: https://academic.oup.com/mnras/article-pdf/213/4/799/2793210/mnras213-0799.pdf. URL: https://doi.org/10.1093/mnras/213.4.799.

[88] J. C. B. Papaloizou and J. E. Pringle. "The dynamical stability of differentially rotating discs – III". In: *Monthly Notices of the Royal Astronomical Society* 225.2 (Mar. 1987), pp. 267–283. ISSN: 0035-8711. DOI: 10.1093/mnras/225.2.267. eprint: https://academic.oup.com/mnras/article-pdf/225/2/267/2961273/mnras225-0267.pdf. URL: https://doi.org/10.1093/mnras/225.2.267.

[89] ALMA Partnership et al. “THE 2014 ALMA LONG BASELINE CAMPAIGN: FIRST RESULTS FROM HIGH ANGULAR RESOLUTION OBSERVATIONS TOWARD THE HL TAU REGION*”. In: *The Astrophysical Journal Letters* 808.1 (2015), p. L3. DOI: 10.1088/2041-8205/808/1/L3. URL: https://dx.doi.org/10.1088/2041-8205/808/1/L3.

[90] Barak A. Pearlmutter. “Learning State Space Trajectories in Recurrent Neural Networks”. In: *Neural Computation* 1.2 (1989), pp. 263–269. DOI: 10.1162/neco.1989.1.2.263.

[91] James B. Pollack et al. “Formation of the Giant Planets by Concurrent Accretion of Solids and Gas”. In: *Icarus* 124.1 (1996), pp. 62–85. DOI: 10.1006/icar.1996.0190.

[92] Derek J. de Solla Price. “Gears from the Greeks: The Antikythera Mechanism — A Calendar Computer from ca. 80 B.C.” In: *Transactions of the American Philosophical Society* 64.7 (1974), pp. 1–70. DOI: 10.2307/1006146. URL: https://www.jstor.org/stable/1006146.

[93] Alain Pétrowski and Sana Ben-Hamida. *Evolutionary Algorithms*. John Wiley & Sons, 2017, p. 30. ISBN: 978-1-119-13638-5.

[94] S. N. Raymond, T. Quinn, and J. I. Lunine. “Making other earths: dynamical simulations of terrestrial planet formation and water delivery”. In: *Icarus* 168 (Mar. 2004), pp. 1–17. DOI: 10.1016/j.icarus.2003.11.019. eprint: astro-ph/0308159.

[95] Sean N. Raymond and Alessandro Morbidelli. “Planet Formation: Key Mechanisms and Global Models”. In: *Demographics of Exoplanetary Systems: Lecture Notes of the 3rd Advanced School on Exoplanetary Science*. Ed. by Katia Biazzo et al. Cham: Springer International Publishing, 2022, pp. 3–82. ISBN: 978-3-030-88124-5. DOI: 10.1007/978-3-030-88124-5_1. URL: https://doi.org/10.1007/978-3-030-88124-5_1.

[96] Luciano Rezzolla and Olindo Zanotti. *Relativistic Hydrodynamics*. Oxford University Press, Sept. 2013. ISBN: 9780198528906. DOI: 10.1093/acprof:oso/9780198528906.001.0001. URL: https://doi.org/10.1093/acprof:oso/9780198528906.001.0001.

[97] Herbert Robbins and Sutton Monro. “A stochastic approximation method”. In: *The annals of mathematical statistics* (1951), pp. 400–407.

[98] Zeeve Rogoszinski and Douglas P. Hamilton. “Tilting Uranus: Collisions versus Spin–Orbit Resonance”. In: *The Planetary Science Journal* 2.2 (2021), p. 78. DOI: 10.3847/PSJ/abec4e. URL: https://dx.doi.org/10.3847/PSJ/abec4e.

[99] V Rokhlin. “Rapid solution of integral equations of classical potential theory”. In: *Journal of Computational Physics* 60.2 (1985), pp. 187–207. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(85)90002-6. URL: https://www.sciencedirect.com/science/article/pii/0021999185900026.

[100] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Cornell Aeronautical Laboratory. Report no. VG-1196-G-8. Spartan Books, 1962. URL: https://books.google.at/books?id=7FhRAAAAMAAJ.

[101] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: https://doi.org/10.1038/323533a0.

[102] Carl Runge. "Über die numerische Auflösung von Differentialgleichungen". In: *Mathematische Annalen* 46 (1895), pp. 167–178.

[103] C. T. Russell et al. "Dawn at Vesta: Testing the Protoplanetary Paradigm". In: *Science* 336.6082 (May 2012), p. 684. DOI: 10.1126/science.1219381.

[104] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.

[105] C.M. Schäfer et al. "A versatile smoothed particle hydrodynamics code for graphic cards". In: *Astronomy and Computing* 33 (2020), p. 100410. ISSN: 2213-1337. DOI: https://doi.org/10.1016/j.ascom.2020.100410. URL: https://www.sciencedirect.com/science/article/pii/S2213133720300640.

[106] William D. Sears. "Tidal Dissipation in Oceans on Titan". In: *Icarus* 113.1 (1995), pp. 39–56. ISSN: 0019-1035. DOI: https://doi.org/10.1006/icar.1995.1004. URL: https://www.sciencedirect.com/science/article/pii/S0019103585710044.

[107] Ivan I. Shevchenko. *Dynamical Chaos in Planetary Systems*. 1st ed. Vol. 1. Astrophysics and Space Science Library. 93 b/w illustrations, 46 illustrations in colour. Springer Cham, 2020, pp. XII, 376. ISBN: 978-3-030-52143-1. DOI: 10.1007/978-3-030-52144-8. URL: https://doi.org/10.1007/978-3-030-52144-8.

[108] Eugene M. Shoemaker. "Meteor Crater, Arizona". In: *Geological Society of America, Centennial Field Guide – Rocky Mountain Section*. Vol. 2. Boulder, Colorado: Geological Society of America, 1987, pp. 399–404. ISBN: 0-8137-5406-2.

[109] Frank Shu et al. "Magnetocentrifugally Driven Flows from Young Stars and Disks. I. A Generalized Model". In: *Astrophysical Journal* 429 (July 1994), p. 781. DOI: 10.1086/174363.

[110] Volker Springel. "E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh". In: *Monthly Notices of the Royal Astronomical Society* 401.2 (2010), pp. 791–851. DOI: 10.1111/j.1365-2966.2009.15715.x.

[111] Volker Springel et al. "Simulations of the formation, evolution and clustering of galaxies and quasars". In: *Nature* 435.7042 (2005), pp. 629–636. DOI: 10.1038/nature03597.

[112] Steven Stahler and Francesco Palla. *The Formation of Stars*. 1st ed. Wiley-VCH, Jan. 2005, p. 865. ISBN: 978-3527405596.

[113] Sarah T. Stewart and Zoë M. Leinhardt. "COLLISIONS BETWEEN GRAVITY-DOMINATED BODIES. II. THE DIVERSITY OF IMPACT OUTCOMES DURING THE END STAGE OF PLANET FORMATION". In: *The Astrophysical Journal* 751.1 (2012), p. 32. DOI: 10.1088/0004-637X/751/1/32. URL: https://dx.doi.org/10.1088/0004-637X/751/1/32.

[114] Nikolaus Sulzenauer. *Earth's Moon*. image. Canon EOS 1000D, exposure: 1/640 sec, ISO speed rating: 1600, time: 15.5.2016, 20:23:54. 2016.

[115] K. Thoma et al. "Experimental impact cratering: The MEMIN project". In: *EGU general assembly, Vienna, (April 2-7, 2006)*. Vol. 8. Geophysical Research Abstracts. 2006, p. 02381.

[116] K. Tsiganis et al. “Origin of the orbital architecture of the giant planets of the Solar System”. In: *Nature* 435.7041 (2005), pp. 459–461. ISSN: 1476-4687. DOI: 10.1038/nature03539. URL: https://doi.org/10.1038/nature03539.

[117] A. M. Turing. “Computing Machinery and Intelligence”. In: *Mind* 59.236 (1950), pp. 433–460. ISSN: 00264423, 14602113. URL: http://www.jstor.org/stable/2251299.

[118] Harold C. Urey and Serge A. Korff. “The Planets: Their Origin and Development”. In: *Physics Today* 5.8 (Aug. 1952), pp. 12–12. ISSN: 0031-9228. DOI: 10.1063/1.3067687. eprint: https://pubs.aip.org/physicstoday/article-pdf/5/8/12/8310758/12_1_online.pdf. URL: https://doi.org/10.1063/1.3067687.

[119] D. Roddy USGS. *Barringer Meteor Crater, Arizona*. image. Copyright: USGS Copyright Free Policy, Accessed: 2024-11-01. 2017. URL: https://solarviews.com/cap/earth/meteor.htm.

[120] E. P. Velikhov et al. “Magneto-rotational instability in the accreting envelope of a protostar and the formation of the large-scale magnetic field”. In: *Astronomy Reports* 56.2 (2012), pp. 84–95. DOI: 10.1134/S106377291201009X. URL: https://doi.org/10.1134/S106377291201009X.

[121] Loup Verlet. “Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules”. In: *Phys. Rev.* 159 (1 1967), pp. 98–103. DOI: 10.1103/PhysRev.159.98. URL: https://link.aps.org/doi/10.1103/PhysRev.159.98.

[122] Voelkel, Oliver et al. “Exploring multiple generations of planetary embryos”. In: *A&A* 666 (2022), A90. DOI: 10.1051/0004-6361/202141830. URL: https://doi.org/10.1051/0004-6361/202141830.

[123] Mark Vogelsberger et al. “Introducing the Illustris Project: Simulating the coevolution of dark and visible matter in the Universe”. In: *Monthly Notices of the Royal Astronomical Society* 444.2 (2014), pp. 1518–1547. DOI: 10.1093/mnras/stu1536.

[124] Hans-Heinrich Voigt. *Abriss der Astronomie*. Ed. by Hermann-Josef Röser and Werner Tscharnuter. 6th ed. Weinheim: Wiley-VCH, 2012, p. 1170. ISBN: 978-3-527-40736-1.

[125] Hans-Heinrich Voigt. *Abriss der Astronomie*. Ed. by Hermann-Josef Röser and Werner Tscharnuter. 6th ed. Adapted from Figure on p. 436, Chapter VII: Sternaufbau und Sternentwicklung. Weinheim: Wiley-VCH, 2012. ISBN: 978-3-527-40736-1.

[126] Kevin J. Walsh et al. “A low mass for Mars from Jupiter’s early gas-driven migration”. In: *Nature* 475.7355 (2011), pp. 206–209. ISSN: 1476-4687. DOI: 10.1038/nature10201. URL: https://doi.org/10.1038/nature10201.

[127] William R. Ward. “Protoplanet Migration by Nebula Tides”. In: *Icarus* 126.2 (1997), pp. 261–281. ISSN: 0019-1035. DOI: https://doi.org/10.1006/icar.1996.5647. URL: https://www.sciencedirect.com/science/article/pii/S001910359695647X.

[128] Darrell Whitley. “A Genetic Algorithm Tutorial”. In: *Statistics and Computing* 4.2 (1994), pp. 65–85. ISSN: 1573-1375. DOI: 10.1007/BF00175354. URL: https://doi.org/10.1007/BF00175354.

[129] Dave Williams, Jay Friedlander, and NASA. *Valhalla Crater on Callisto*. image. NASA Image ID: P-21287. Public domain image from NASA, Accessed: 2024-11-15. 1979. URL: https://nssdc.gsfc.nasa.gov/imgcat/html/object_page/vg1_p21287.html.

[130] P. M. Winter et al. “Reconstruction of Callisto’s Valhalla basin using n-body and SPH simulations”. In: *European Planetary Science Congress*. Sept. 2017, EPSC2017-795, EPSC2017–795.

[131] P. M. Winter et al. “Residual Neural Networks for the Prediction of Planetary Collision Outcomes”. In: *53rd Lunar and Planetary Science Conference*. Vol. 2678. LPI Contributions. Mar. 2022, 1920, p. 1920.

[132] P. M. Winter et al. “The Origin of Callisto’s Valhalla Basin: First Results of SPH Impact Simulations and the Search for the Impactor’s Origin”. In: *49th Lunar and Planetary Science Conference*. Houston: Lunar and Planetary Institute, 2018, Abstract #1519. URL: https://www.hou.usra.edu/meetings/lpsc2018/pdf/1519.pdf.

[133] Philip Winter et al. “Reconstruction of Callisto’s Valhalla basin using n-body and SPH simulations”. In collaboration with Eberhard Karls Universität Tübingen. MA thesis. Wien, Austria: Universität Wien, Institut für Astrophysik, 2017. URL: https://ucrisportal.univie.ac.at/de/publications/4b70a4fe-c8be-4aa5-adac-9a9ca6703d6a.

[134] Philip M. Winter, Mattia A. Galiazzo, and Thomas I. Maindl. “Analysis of Close Encounters With Ganymede and Callisto Using a Genetic n-Body Algorithm”. In: *Frontiers in Astronomy and Space Sciences* 5 (2018). ISSN: 2296-987X. DOI: 10.3389/fspas.2018.00016. URL: https://www.frontiersin.org/articles/10.3389/fspas.2018.00016.

[135] Philip M Winter et al. “Residual neural networks for the prediction of planetary collision outcomes”. In: *Monthly Notices of the Royal Astronomical Society* 520.1 (Oct. 2022), pp. 1224–1242. ISSN: 0035-8711. DOI: 10.1093/mnras/stac2933. eprint: https://academic.oup.com/mnras/article-pdf/520/1/1224/49103575/stac2933.pdf. URL: https://doi.org/10.1093/mnras/stac2933.

[136] Charles F Yoder. “How tidal heating in Io drives the Galilean orbital resonance locks”. In: *Nature* 279.5716 (1979), pp. 767–770.

[137] Qian Yuan et al. “Moon-forming impactor as a source of Earth’s basal mantle anomalies”. In: *Nature* 623.7985 (2023), pp. 95–99. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06589-1. URL: https://doi.org/10.1038/s41586-023-06589-1.

[138] Matthew D. Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 818–833. ISBN: 978-3-319-10590-1.

[139] Zhongtian Zhang, David Bercovici, and Jacob S. Jordan. “A Two-Phase Model for the Evolution of Planetary Embryos With Implications for the Formation of Mars”. In: *Journal of Geophysical Research: Planets* 126.4 (2021). e2020JE006754 2020JE006754, e2020JE006754. DOI: https://doi.org/10.1029/2020JE006754. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2020JE006754. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020JE006754.

[140] M Zolensky et al. “Meteorites and the Early Solar System II”. In: *Univ. Ariz. Press, United States* (2006), pp. 869–888.

[141] Ondřej Šrámek et al. “Thermal evolution and differentiation of planetesimals and planetary embryos”. In: *Icarus* 217.1 (2012), pp. 339–354. ISSN: 0019-1035. DOI: https://doi.org/10.1016/j.icarus.2011.11.021. URL: https://www.sciencedirect.com/science/article/pii/S0019103511004489.