ABSTRACT

Title of Dissertation: MECHANICAL EVOLUTION OF SMALL SOLAR SYSTEM BODIES

Julian C. Marohnic
Doctor of Philosophy, 2023

Dissertation Directed by: Professor Derek C. Richardson
Department of Astronomy

This dissertation presents a series of studies that develop and apply numerical modeling techniques to small bodies in the solar system. We are particularly interested in low-energy deformations, collisions, and disruptions, and our subjects range from near-Earth asteroids to Kuiper belt contact binaries in the farthest reaches of the solar system. We use the *N*-body code `pkdgrav` to investigate these processes and describe our significant additions to its capabilities.

Our first subject is the Kuiper belt object Arrokoth. On January 1, 2019, the *New Horizons* spacecraft flew within 3,550 km of Arrokoth, returning the first in-situ images of a small body in the classical Kuiper belt. Arrokoth was found to be bilobate, with a distinctive contact binary structure. We use `pkdgrav` to investigate the origins of Arrokoth's striking shape and find that plausible formation mechanisms are quite limited. We rule out the possibility of a direct impact between two unbound objects and put forward an alternate scenario in which two cometesimals in a close, synchronous orbit gradually spiral in toward one another before meeting in a gentle

merger. We conclude by exploring implications for the formation of small Kuiper belt objects more generally.

Next, we describe our work modifying `pkdgrav` to accommodate non-spherical particles. Prior work in granular physics has established that particle shape is an important factor governing the behavior of granular bodies like small solar system objects. Irregular particles tend to interlock with one another, inhibiting bulk motion and adding to the shear strength of a medium. We adapt `pkdgrav`'s existing soft-sphere, discrete element contact physics model to allow for modeling of non-spherical grains. We then apply this new capability in three, small-scale proof of concept studies of spin-up, tidal disruption, and the Brazil nut effect. We find a significant difference in behavior when comparing small rubble-pile bodies composed of spherical particles and those composed of non-spherical particles.

Finally, we apply our newly-developed tools to a more comprehensive investigation of particle shape in tidal disruption simulations. We construct small rubble piles from a range of differently-shaped constituents and subject them to simulated tidal encounters with the Earth. We conduct a parameter sweep across different encounter geometries and constituent shapes and conclude that particle shape is a significant contributor to tidal encounter outcomes. The role of particle resolution is also investigated.

MECHANICAL EVOLUTION OF SMALL SOLAR SYSTEM BODIES

by

Julian C. Marohnic

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor Derek C. Richardson, Chair/Advisor
Professor Jessica M. Sunshine
Professor Douglas P. Hamilton
Professor Christine M. Hartzell
Dr. Olivier S. Barnouin

Preface

The work presented in Chapter 2 appeared in the planetary science journal *Icarus* in 2021 under the title "Constraining the final merger of contact binary (486958) Arrokoth with soft-sphere discrete element simulations" (Marohnic et al., 2021). Chapter 3 has been accepted for publication in *The Planetary Science Journal* as "An efficient numerical approach to modeling the effects of particle shape on rubble-pile dynamics" (Marohnic et al., 2023). We include citations to this work several times in this dissertation, with the understanding that this work has not yet been officially published. Finally, Chapter 4 represents an as yet unpublished, but largely complete, study on particle shape and tidal disruptions. We expect that this chapter will be submitted for publication soon after the submission of this dissertation.

Acknowledgments

I feel that I can only begin by expressing my deep gratitude to my research advisor and mentor at the University of Maryland, Derek Richardson. Derek has been an intelligent, thoughtful, patient, and kind mentor throughout my time here and I will always be grateful for that. I feel this all the more acutely after having finished writing this thesis—Derek did not hesitate to offer help and feedback to me on a moment's notice and without a word of complaint, even during his weekends and vacations.

I would also like to thank my colleagues and fellow students in the Astronomy Department. First, to Joe, Harrison, and Carrie, thank you for your companionship and advice in research, at conferences, and in life. I will always be grateful to all of my classmates for their friendship and support, and I will remember my time at the University of Maryland fondly. There are more people that deserve to be mentioned here than I have the space to list, but I want to make a point of thanking those who began their time here with me: Sergio, Harrison, Tomás, Charlotte, Ramsey, Milena, Guangwei, and Yvette. We spent countless hours together struggling through courses, helping each other with research, and of course, having fun. Special thanks also to my close friends Ramsey, Vicente, and Milena and my roommates Liz, Carrie, and Ben. The times I spent with you all over the past six years were sometimes happy and sometimes difficult, but never lonely and certainly never dull.

To my parents Theresa and Brian and my brother Alexander: thank you for your enduring

support and love. I am always, always conscious of how much you have done for me and that I wouldn't be who I am or where I am without our family.

Finally, I am grateful to Zoe and Charlie for their boundless love and patience during my six long years here, during the hard months while I wrote this dissertation, and always.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AMD	Advanced Micro Devices, Inc.
APL	Applied Physics Laboratory
BNE	Brazil Nut Effect
CCKBO	Cold Classical Kuiper Belt Object
COM	Center of Mass
DART	Double Asteroid Redirection Test
DEM	Discrete Element Method
DLR	German Aerospace Center
ESA	European Space Agency
GI	Gravitational Instability
GPU	Graphical Processing Unit
HC	Hierarchical Coagulation
HSDEM	Hard-Sphere Discrete Element Method
JAXA	Japanese Aerospace Exploration Agency
JHU	Johns Hopkins University
KB	Kuiper Belt
KBB	Kuiper Belt Binary
KBO	Kuiper Belt Object
LL	Large Lobe
MASCOT	Mobile Asteroid Surface Scout
MIT	Massachusetts Institute of Technology
NASA	National Aeronautics and Space Administration
NEA	Near-Earth Asteroid
PGM	Platinum Group Metal
PHA	Potentially Hazardous Asteroid
SL	Small Lobe
SL9	Shoemaker-Levy 9
SPH	Smoothed-Particle Hydrodynamics
SSDEM	Soft-Sphere Discrete Element Method
SSSB	Small Solar System Body
STScI	Space Telescope Science Institue
YORP	Yarkovsky–O'Keefe–Radzievskii–Paddack effect

Chapter 1: Introduction

The overarching theme of this dissertation is the mechanical evolution of small solar system objects, particularly those of "rubble pile" composition. Asteroids, comets, and Kuiper belt objects are subjected to a wide variety of physical processes from formation to destruction. We are primarily interested in the use of numerical techniques to understand low-energy deformations and disruptions of small bodies. We will focus in particular on tidal deformation and disruption as well as the origins of the recently visited Kuiper belt contact binary 486958 Arrokoth. In addition to direct applications of numerical approaches to the study of small bodies, we will describe our substantial additions and modifications to the N-body, discrete element method (DEM) code `pkdgrav` that made much of this work possible. This introduction will begin with a survey of small bodies and their importance in our solar system in Section 1.1 and a discussion of their internal structure in Section 1.2. Then, we provide some context for our study of Arrokoth in Section 1.3. In Section 1.4, we discuss the role of planetary tidal encounters in the evolution of small bodies. We trace the progress of efforts to numerically model small solar system bodies in Section 1.5. Section 1.6 is a detailed overview of `pkdgrav`. Finally, 1.7 outlines the overall structure of the remainder of the dissertation.

1.1 Small bodies in the solar system

The population of small solar system bodies (SSSBs, or "small bodies") as a whole is quite diverse and spans the width of the entire solar system, from the classically defined asteroids of the inner solar system to the Kuiper belt and Oort cloud objects beyond the orbit of Neptune. Small bodies are of great interest to planetary scientists, with numerous past, current, and future space missions dedicated to their exploration, in addition to ground-based observation campaigns. Many small bodies in the outer solar system are believed to be primordial, left nearly untouched for billions of years (Morbidelli et al., 2008). These objects can tell us a great deal about the origins and early evolution of our solar system. In the inner solar system, ground-based observations and space missions targeting asteroids have contributed to our understanding of planet formation and the collisional evolution of terrestrial bodies. Potential applications for the study of small bodies go well beyond planetary science. It has been suggested that impacts by comets and asteroids, as well as collisions with larger planetesimals, are a significant source of water on our own planet (Morbidelli et al., 2000; O'Brien et al., 2014). However, others dispute this idea and point to other possible origins for Earth's water (e.g., Piani et al., 2020). It has even been proposed that life itself, or possibly its chemical building blocks, was initially brought to Earth by an extrasolar asteroid or comet (Ginsburg et al., 2018), although this theory is not widely accepted.

Small bodies are also of interest to humanity in more pragmatic terms. Asteroid and comet impacts have posed an existential, if sporadic, threat to life on Earth since its inception. The famous Chicxulub impact that brought about the extinction of all non-avian dinosaurs is believed to have been caused by a large asteroid (Alvarez et al., 1992). In the present day, nations around the planet are increasingly dedicating resources to monitoring small bodies for potential impact

threats. In 2022, the Double Asteroid Redirection Test (DART) spacecraft executed a planned impact into the binary asteroid Didymos as part of an effort to gather data for future impact prevention (Cheng et al., 2021; Daly et al., 2023). There is also growing economic interest in small bodies, particularly in the inner solar system. Asteroids are thought to contain valuable platinum group metals (PGMs) like palladium, iridium, and platinum in far higher concentrations than terrestrial ores. Base metals such as iron, aluminum, and magnesium are also available in abundance and could be used in the future to support space-based construction and manufacturing (Cannon et al., 2023).

1.2 The rubble-pile model of small solar system bodies

Our knowledge of the internal structure of small bodies has evolved substantially in recent decades. For most of human history, the nature and internal structure of small bodies was largely unknown. Asteroids, for example, were believed to be monolithic bodies with bare, rocky surfaces less than a century ago (e.g., Jeffreys, 1947). In contrast, a very large portion of small bodies are now thought to be "rubble piles," sometimes referred to in the literature as "gravitational aggregates" (Richardson et al., 2002). A rubble-pile body is a porous, weakly constituted mass of fragments ranging in size from small grains to large boulders. Rather than internal material strength, rubble piles are almost entirely held together by the mutual self-gravity of their constituent pieces. More recently, there have been investigations into the possible cohesive effects of small van der Walls forces between rubble pile grains (e.g., Rozitis et al., 2014; Sánchez and Scheeres, 2014; Zhang et al., 2018). These forces would likely be quite small (no more than a few hundred pascals), but the additional tensile strength could affect maximum allowable

rotation rates on the margin (Scheeres and Sánchez, 2018). This overall picture of rubble-pile small bodies is quite different from the previously held model of asteroids and comets as coherent, solid objects and has important implications for their behavior and evolution. In this section, we discuss both the historical motivations for the rubble-pile concept and the current state of the evidence supporting it.

Before proceeding, we note that the terminology used to describe these objects and their structures may cause some confusion. As stated, a large proportion of all small bodies in the solar system, including asteroids, comets, and Kuiper belt objects, are believed to be highly porous, shattered, or in some way lacking in significant internal strength. Despite their superficially similar structures, their formation histories may be quite different from each other. Many asteroids are the result of collisions in which larger, coherent objects have been shattered and dispersed. The collections of their coalesced fragments that remain are traditionally referred to as "rubble piles." In contrast, comets and Kuiper belt objects are thought to have initially formed by accumulation from smaller pieces and as a result may have the structure of gravitational aggregates without any history of catastrophic collisions. To further complicate matters, internal structure likely spans a continuum of possible configurations. For example, some asteroids may be composed of many small, similarly sized pieces, while others may consist of a layer of rubble surrounding a core of much larger fragments or boulders. Both could be referred to as rubble piles. Historical circumstance has led to the "rubble pile" moniker being applied more broadly in some contexts, while still carrying the implication of a collisional origin in certain settings. In this dissertation, we attempt to avoid using the term "rubble pile" to refer to comets specifically where possible, with the understanding that modern usage has made it difficult to keep a consistent separation in terminology between collisional and primitive, accreted aggregates. We will use the term "rubble

pile" to refer to gravitational aggregates in the abstract in accordance with common contemporary usage. For a nuanced discussion of this issue, readers may refer to Richardson et al. (2002).

1.2.1 Historical development of the rubble-pile model

In the 19th century, work by Édouard Roche established the distance at which a fluid body held together by self-gravity would be disrupted by the tidal forces exerted by a second body—this is now known as the "Roche limit," and is discussed in greater detail in Section 1.4. The earliest efforts to extend the work of Roche from fluids to solid bodies like asteroids and comets relied on the assumption that these bodies were monoliths. On this premise, Jeffreys (1947) calculated that tidal disruption of rocky bodies by Jupiter would only be possible for objects greater than 440 km in diameter. He similarly finds a threshold of 220 km in the case of disruption by the Earth. Öpik (1950) revised that estimate down by a factor of $\sim$2 by substituting a more realistic estimate of material strength. Still, he concluded that tidal disruption of solid rocky bodies smaller than $\sim$100 km is functionally impossible. Models of asteroid structure continued to evolve through the latter half of the 20th century. The term "rubble pile" was coined by Chapman (1978) to describe asteroids with unconsolidated, strengthless interiors that might be formed as a byproduct of collisional evolution in the main belt.

At the same time, more detailed models of cometary structure were being put forward. In late 1965, observers documented the tidally induced breakup of the sun-grazing comet Ikeya-Seki into three large fragments (Hirayama and Moriyama, 1965). This observation prompted speculation about the internal strength of small bodies (Öpik, 1966). A gravitational aggregate model for comets was put forward less than a decade later by Weissman (1986) to explain the

observed frequency of comet breakup events, though he suggested that in the case of comets the primary fragments would likely be primordial rather than derived from collisions. The breakup of comet Shoemaker-Levy 9 (SL9) lent further observational support to this idea. During a close encounter with Jupiter in 1992, tidal forces pulled SL9 into dozens of fragments which would go on to impact the planet's surface on a subsequent passage (Chodas and Yeomans, 1996). Subsequent calculations of the tidal stress exerted on SL9 by Scotti and Melosh (1993) estimated a magnitude of only a fraction of bar. Despite their differing formation pathways, it is now believed that a large portion of both asteroids and cometary nuclei are low-strength, gravitational aggregates of some kind.

1.2.2 Further evidence for the rubble-pile model of small bodies

The rubble-pile model has gained wide acceptance and is now applied to a variety of small bodies, from near-Earth asteroids (NEAs) to comets in the outer solar system. We have already discussed instances of comets breaking up under the influence of tidal stresses and how these observations pointed to a low-strength, low-density structure. Additional studies and observations have lent further support to the idea of an aggregate structure in small bodies with a wide array of formation histories.

A study of near-Earth, Mars-crossing, and main belt asteroids in the early 2000s revealed a striking trend in their rotation rates. Nearly all of the asteroids observed with diameters less than $\sim$10 km had spin periods of greater than 2.2 hours (see Fig. 1.1). Those few that did not were all less than $\sim$200 m in diameter and tended to be roughly spheroidal in shape (Pravec and Harris, 2000; Pravec et al., 2002). The observed maximum spin rate for objects larger than

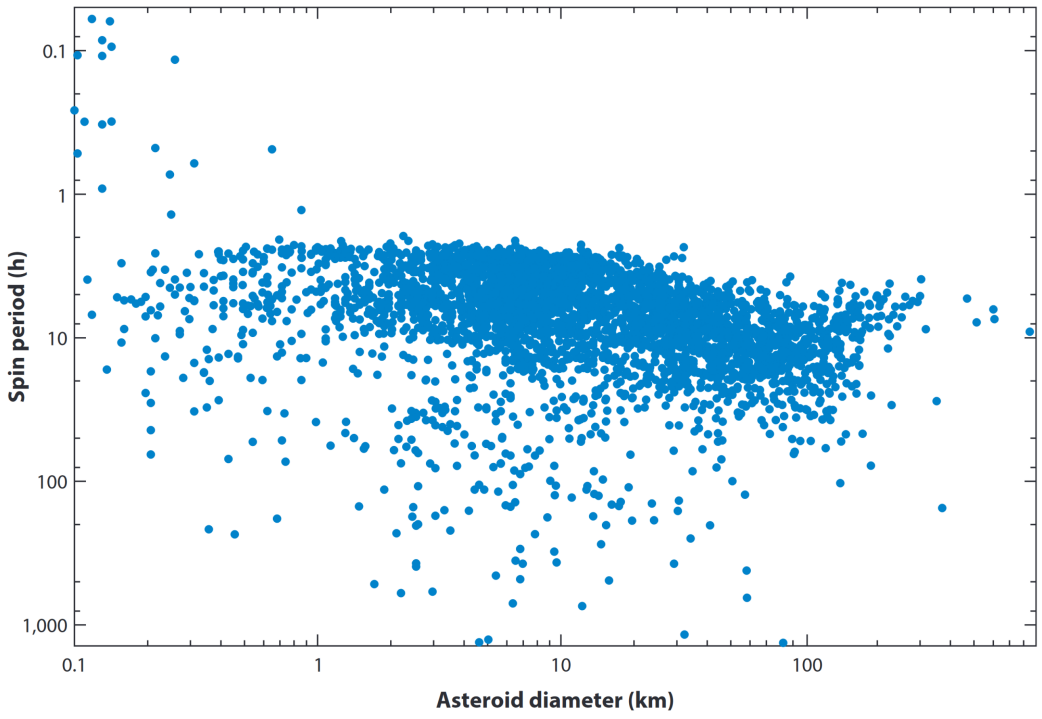


Figure 1.1: A plot of asteroid spin periods as a function of asteroid diameter. Asteroids between ∼1 km and ∼100 km in diameter seem to experience a "spin limit," which is ascribed to low internal strength. This image was taken from a review by Walsh (2018), which in turn used asteroid lightcurves from the Planetary Data System archive to make this plot. The data presented here includes two outlier objects with diameters greater than 200 m and spin periods shorter than 2 hours that were not present in the initial analysis by Pravec and Harris (2000). Readers may refer to that work for more information regarding this figure and to Pravec and Harris (2000) and Pravec et al. (2002) for further discussion of asteroid spin rates.

200 m was found to correspond closely to the theoretical rotational breakup limit for an asteroid with reasonable density and no tensile strength. While not definitive, this observation strongly suggested a rubble-pile structure for asteroids between 200 m and 10 km in diameter. In contrast, the smaller, faster-rotating asteroids in the sample seemed likely to be monolithic boulders with the requisite strength to sustain tension from rotation.

The discovery of "crater chains" or "catenae" in the solar system, particularly Earth's moon and the Galilean satellites, has also bolstered the case for strengthless small bodies. Crater chains

are linear collections of craters that can span significant distances (up to tens of kilometers) on terrestrial bodies and can not be ascribed to local geologic processes or secondary impacts. These features are created when small objects disintegrate prior to impact due to tidal stresses exerted by the primary body. The remnant fragments then impact the target body sequentially along a line, leaving a distinctive signature. It is notable that these features have largely been found on the Moon and the Jupiter-facing sides of Callisto and Ganymede, as all three are satellites about planets where tidal disruptions are known to occur (Asphaug and Benz, 1996). The existence of crater chains and their observed frequency have been cited as evidence of the rubble-pile nature of small bodies (Bottke Jr and Melosh, 1996; Walsh, 2018).

Numerous spacecraft have visited small bodies and have demonstrated their rubble-like properties via in-situ imaging and measurement, at least with regard to the surfaces of these objects. The asteroid 25143 Itokawa, photographed and sampled by the spacecraft *Hayabusa* in 2005, is covered with a layer of gravel, boulders, and fine regolith (Fujiwara et al., 2006). Itokawa was estimated to have high bulk porosity ($\sim$40%), which is at least consistent with a rubble-pile interior (Abe et al., 2006). Images and measurements from missions to the asteroids 162173 Ryugu and 101955 Bennu (see Fig. 1.2) in recent years have clearly indicated that they have rubble-pile surfaces as well (Watanabe et al., 2019; Barnouin et al., 2019).

Robotic explorations of comets have yielded similar results. In 2005, the *Deep Impact* spacecraft executed a planned collision with the comet 9P/Tempel (A'Hearn et al., 2005). Subsequent analysis of the ejecta plume and other data returned by the mission estimated a bulk density of approximately 400–500 kg/m^3 and a surface-layer porosity in excess of 70%, at least near the site of impact. Data from the *Rosetta* mission to the comet 67P/Churyumov-Gerasimenko (see Fig. 1.3) suggest similarly low density and high porosity figures for that object. These measure-

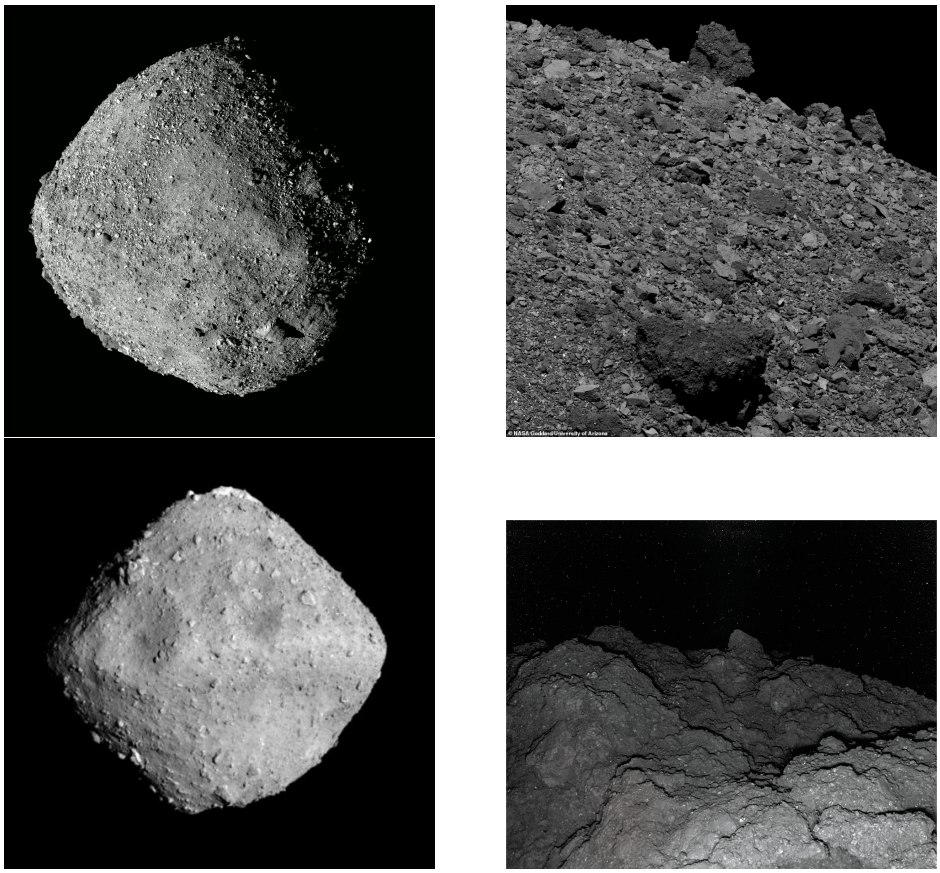


Figure 1.2: The top two panels depict the asteroid Bennu and its surface. The bottom two images are of the asteroid Ryugu. Both asteroids are believed to have rubble-pile structure, with high bulk porosities. Image credits: NASA Goddard/University of Arizona, JAXA/DLR/MASCOT.

ments strongly suggest unconsolidated, low-strength internal structures, although both of these comets are believed to be primordial, rather than collisionally derived, gravitational aggregates (Davidsson et al., 2016). We note that, to date, there have been no direct measurements or observations of the interiors of small bodies. Although the body of evidence we have presented here suggests otherwise, a coherent internal structure cannot be firmly ruled out.

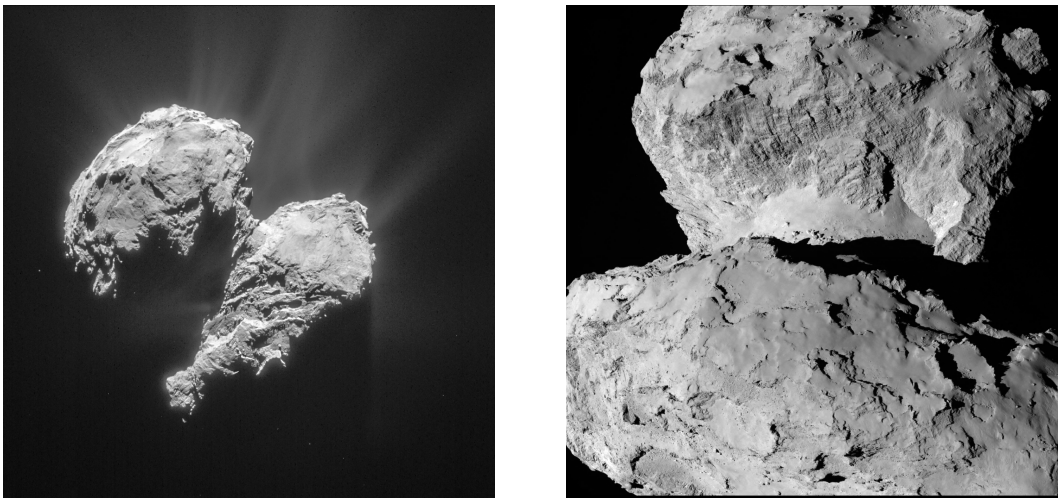


Figure 1.3: Images of the comet 67P/Churyumov-Gerasimenko taken by the *Rosetta* spacecraft. Comet 67P is an example of a highly porous small body, likely due its formation via the accretion of smaller "cometesimals." Image credits: ESA/Rosetta

1.2.3 The origins of rubble-pile asteroids

Another line of support for the rubble-pile model, particularly in the case of asteroids, comes from work on planet formation. Asteroids are in a sense "leftover material" from the birth of our solar system. Thus, it is natural that planet formation models can inform our understanding of asteroids, just as asteroids inform our understanding of planet formation. The dominant formation mechanism for large planetesimals in the young solar system remains uncertain. Traditionally, it was believed that bottom-up coagulation of small grains would gradually produce a population of 100 to 1000 km bodies (Michel et al., 2015). However, this theory has been seriously challenged. Modeling and laboratory work has shown that once silicate particles reach the millimeter size range, it is difficult to induce them to stick. Instead, they have a tendency to bounce off of one another (Güttler et al., 2010). This finding poses a major problem for the hierarchical growth model of planetesimals. More recently, an alternate model of plan-

etesimal growth has gained support. Small grains in the Sun's protoplanetary disk corralled into high-density regions by turbulent gas could collapse directly into large bodies up to a thousand kilometers in diameter (Johansen et al., 2007; Morbidelli et al., 2009). This model implies that asteroids were "born big." Under this new paradigm, objects of diameter greater than $\sim$100 km are primordial. This initial population was then gradually ground down in a process known as "comminution." Repeated collisions shattered large planetesimals, whose fragments would then have been dispersed and reaccumulated into new asteroids. This process continued until Jupiter reached its present size, at which point most remaining planetesimals would have been ejected from the main belt, leaving us with the population of rubble-pile asteroids we see today.

1.3 The contact binary Arrokoth

The Kuiper belt (KB), sometimes known as the Edgeworth-Kuiper belt, is a circumstellar disk of icy material lying beyond the orbit of Neptune at 30 au. Its outer edge is typically established at 50 au. The existence of what we now know as the Kuiper belt was first put forward by Edgeworth (1943) and later by Kuiper (1951), although it seems that Kuiper was not aware of Edgeworth's earlier work at the time (Jewitt, 1999). Pluto is the largest member of the KB, though it was not recognized as part of a broader population at the time of its discovery in 1930 (Leonard, 1930; Tombaugh and Moore, 2017). With the exception of Pluto, the first definitive observation of a KBO came in the early 1990s with the discovery of 1992 QB1 (Jewitt and Luu, 1993). The number of known KBOs grew quickly afterwards.

Prior to the *New Horizons* mission's flyby of Pluto and its five satellites in 2015, our only knowledge of KBOs came from observations made from Earth's vicinity (Stern et al., 2015).

After leaving the Pluto system behind, the *New Horizons* spacecraft was redirected to a close passage with the KBO 2014 MU_{69}, which was later given the name "Arrokoth." On January 1, 2019, *New Horizons* executed the flyby, passing within 3,538 km (Stern et al., 2019). Arrokoth is distinctly bilobate object, consisting of two oblate spheroids joined together by a narrow, neck-like feature (see Fig. 1.4). Its dimensions are roughly 36 km by 20 km by 10 km. Since the lobes are in contact rather than in mutual orbit, Arrokoth's mass and bulk density are not well-constrained. It has a rotation period of $\sim$ 16 hours and the rotation axis is oriented such that the body is flattened in the plane of rotation. If we assume that the neck feature is not in tension, Arrokoth's overall bulk density must be no less than 290 kg/m^3 (Spencer et al., 2020). The overall appearance of Arrokoth is quite striking and is somewhat reminiscent of a two-segment snowman—the two lobes are very well-defined with only a narrow contact plane connecting them. Based on appearance alone, it seems very likely that the two lobes were initially distinct bodies.

As the only KBO that has been directly observed outside of the Pluto system, Arrokoth presents a unique case study that can provide insight into the KBO population and its provenance. In Chapter 2, we apply our *N*-body, DEM code `pkdgrav` to Arrokoth and investigate possible merger scenarios that could explain how its contact binary structure came about. While we have no definitive proof that Arrokoth is a gravitational aggregate, there is some evidence that it is indeed built up from primordial cometesimals (Morbidelli and Nesvorný, 2020; Stern et al., 2023). Given that Arrokoth is likely a primordial body that has been left undisturbed since its formation (McKinnon et al., 2020), it seems quite plausible that it has a rubble-pile structure similar to that of a comet.

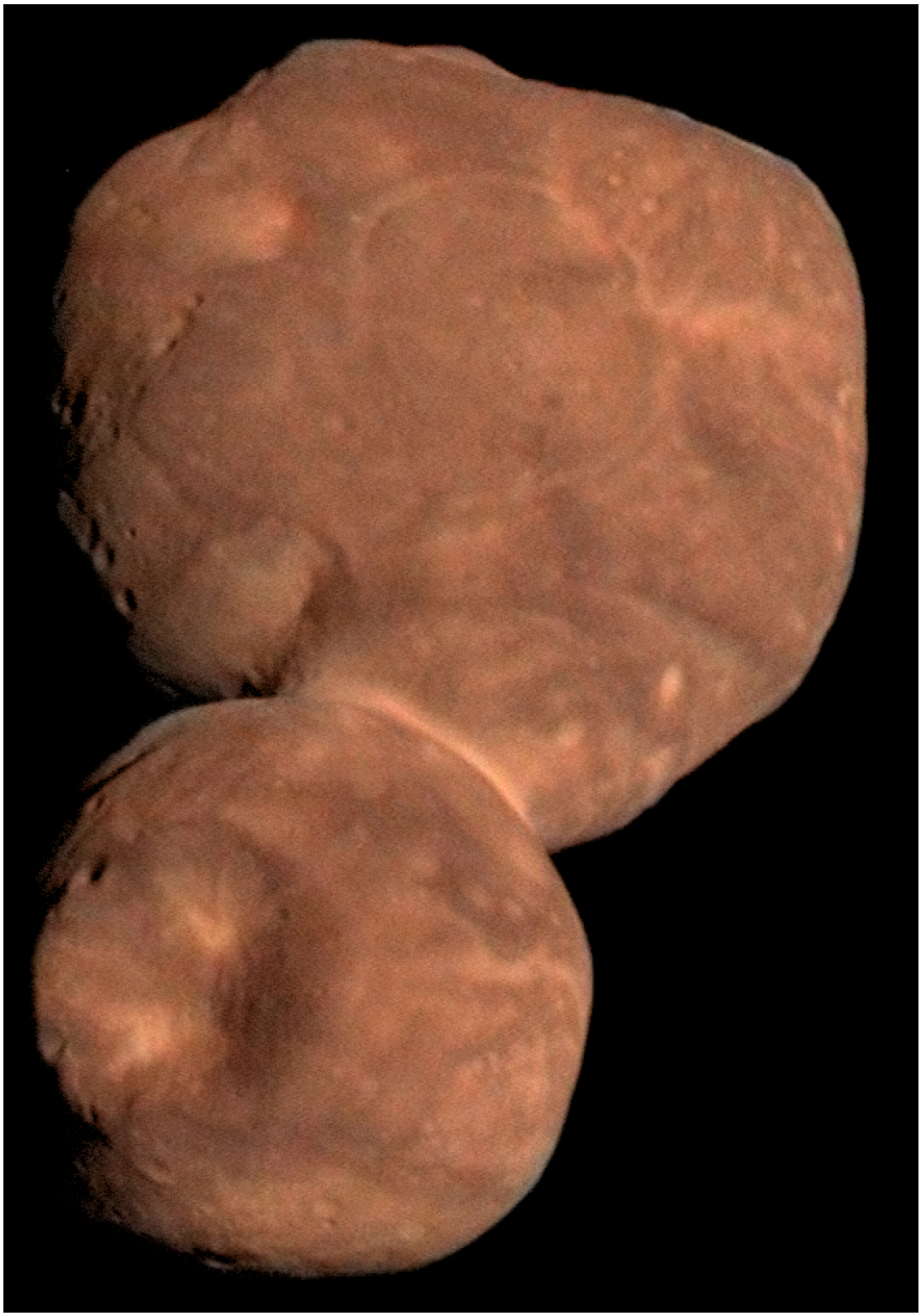


Figure 1.4: The Kuiper belt contact binary Arrokoth as seen by *New Horizons*. 2 discusses the origins of Arrokoth in detail, with particular focus on its notable bilobate shape. Image credit: NASA/JHU/APL

1.4 Planetary tidal encounters

Tidal encounters with planets are an example of a relatively low-energy process that can dramatically reshape small bodies. A tenuous encounter may induce resurfacing in a rubble-pile body (Binzel et al., 2010; Kim et al., 2023) and stronger tidal stresses can cause wholesale reshaping of small bodies, with a tendency to produce elongated objects. In the most extreme cases, planetary encounters can disrupt small bodies completely. The specific outcome will be influenced by the encounter geometry, the relative densities of the two objects, rotation states, and other factors. Disruptions of small bodies by planetary tides have been observed in our solar system, as we have already seen in the cases of comets SL9 and Ikeya-Seki (see Fig. 1.5).

The formal study of tidal disruption began with Roche (1847). Roche derived an expression for the distance within which an inviscid, fluid body in synchronous orbit about a larger object cannot maintain itself. If its orbit is any smaller, it must disintegrate. The classical Roche radius d can be expressed as follows:

$$d = 2.455 R_p \sqrt[3]{\frac{\rho_p}{\rho_s}},$$

where R_p is the radius of the primary and ρ_p and ρ_s are the bulk densities of the primary and secondary bodies, respectively. The Roche limit is often applied generally in the context of solar system bodies. However, the limit that Roche calculated came with some stipulations regarding the bodies in question: an ellipsoidal secondary with specific axis ratios, a spherical primary, a circular orbit, and, crucially, the requirement of a fluid secondary body. This was recognized as a shortcoming in the theory, and we have already discussed the work of Jeffreys (1947) and Öpik

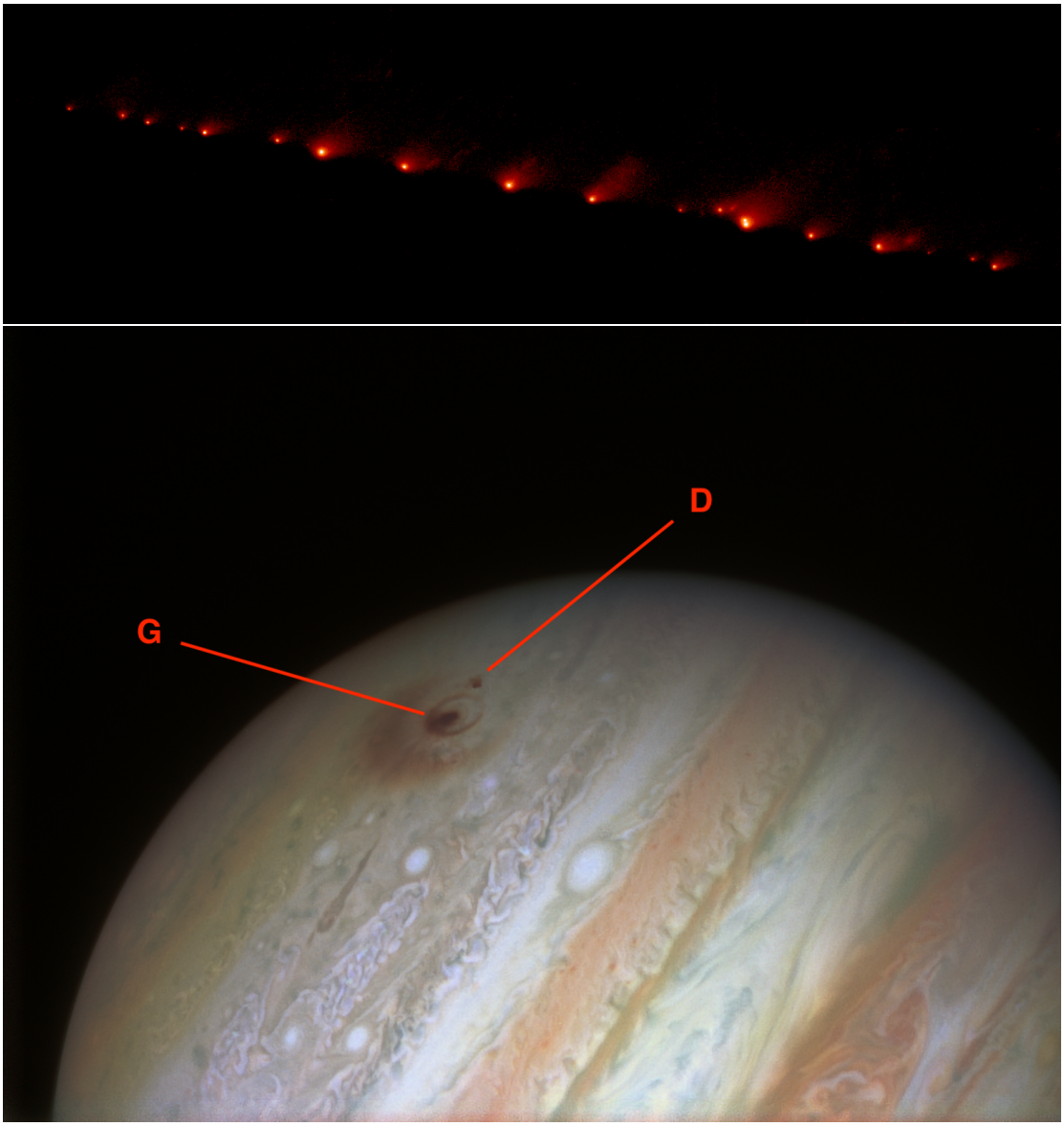


Figure 1.5: Tidally disrupted fragments of the comet Shoemaker-Levy 9 prior to their impact with Jupiter in 1994 (top). Later, the impact sites of fragments D and G are clearly visible on Jupiter's southern hemisphere (bottom). The red labels indicate impact scars left by the two fragments. Observations of the breakup informed models of the internal structure of comets. Image credits: NASA/STSci, NASA/MIT/H. Hammel

(1950) that expanded Roche's theory to consider the effects of material strength and spherical secondaries. Analytical models of tidal disruption continued to grow more sophisticated throughout the 20th and early 21st centuries alongside advances in models of asteroidal and cometary structure (e.g., Sekiguchi, 1970; Aggarwal and Oberbeck, 1974; Davidsson, 2001; Holsapple and Michel, 2006). Although analytical theories of tidal disruption are quite useful in some contexts, their applications have strict limits. Just as was the case for Roche's theory, these models are concerned with viable equilibrium shapes and spins and shapes for given tidal stresses. They are static theories that can tell us at what point a body will begin to disrupt, but not what will happen after a deformation or disruption has begun (Holsapple and Michel, 2008).

1.5 Numerical modeling of solar system rubble piles

As analytical models of tidal disruption were being refined, others were exploring numerical approaches to studying the problem. Using computational rather than analytical methods generally allows for easier study of dynamic phenomena, where small bodies and their constituents are allowed to evolve and change in time. The two dominant computational approaches to studying processes involving small bodies are smoothed-particle hydrodynamics (SPH) and the discrete element method (occasionally referred to as the distinct element method).

1.5.1 Smoothed-particle hydrodynamics

SPH methods were designed to simulate continuum media, typically fluids or solid media that can flow and deform (Gingold and Monaghan, 1977). While initially developed with astrophysical applications in mind, SPH methods have found uses in a variety of fields includ-

ing volcanology, oceanography, and glaciology (Mullet et al., 2023; Tran-Duc et al., 2018; Pan et al., 2013). While technically a particle-based method, SPH "particles" represent the state of a continuum in a given location. Rather than directly resolving interactions between individual constituents in a granular medium, SPH codes use constitutive models to track and simulate the macroscopic properties of a medium like temperature, porosity, and material strength.

SPH codes have been and still are applied to the study of small rubble-pile bodies. Boss et al. (1991) used an SPH code to investigate tidal disruption of large planetesimals and found that the Earth's tidal forces could prompt spin-up, elongation, and mass loss. Asphaug and Benz (1996) studied the disruption of the comet SL9 and used an SPH model to constrain its pre-encounter size and density. Jutzi et al. (2019) applied an SPH code equipped with both N-body and porosity models to asteroid collisions between large asteroids. They found that large, porous targets have significant resistance to collisional disruption, behaving more like monolithic bodies than rubble piles. SPH is most advantageous relative to DEM when studying high-energy impacts and fragmentation. In these contexts, the extremely small time steps required to resolve the evolution of the medium would be prohibitively expensive for DEM simulations of any reasonable resolution. In contrast, DEM explicitly models individual grains and their contacts and is generally preferred when dealing with relatively low-energy, quasi-static phenomena. Since these processes are the focus of this dissertation, we will now set aside SPH and discuss DEM and some of its applications to small rubble-pile bodies in greater detail.

1.5.2 The discrete element method

DEM was initially developed by Cundall (1971). Readers may also refer to Cundall and Strack (1979) for a more detailed treatment of this influential implementation of DEM. In contrast to SPH, DEM models granular media as a collection of fully distinct constituents or particles. Particles are allowed to come into direct contact with one another and reaction forces are computed explicitly. The calculated forces are then applied to each particle and their velocities are updated accordingly. Depending on the details of the implementation, various frictional forces may be applied as well. Self-gravity is also a crucial element when modeling small bodies due to their rubble-pile structure. If we want to apply DEM codes to small bodies, they must include a method of calculating interparticle gravitational forces. The fundamental problem of combining N-body capability with a DEM code is relatively difficult, both from the perspective of implementation and in terms of the computational resources required. This topic will be discussed further in Section 1.6, although it is not in and of itself a primary focus of Section 1.6 or this dissertation.

An important distinction is drawn between "hard-sphere" DEM (HSDEM) and "soft-sphere" DEM (SSDEM). Under HSDEM schemes, particles are impenetrable and collisions are considered instantaneous (Richardson et al., 2000). This approach has since been replaced in many applications by soft-sphere friction models (e.g., Wada et al., 2006; Sánchez and Scheeres, 2011; Schwartz et al., 2012). This was motivated by a number of drawbacks that HSDEM suffers from in contrast with SSDEM. Typically, HSDEM requires each individual contact event to be resolved chronologically. A post-collision trajectory is calculated for each involved particle in any given contact before the simulation can continue. Since collisions are modeled as instantaneous, point-

like contacts, HSDEM is also incapable of handling static, rolling, or twisting friction interactions that by their nature require persistent contact between particles. In the case of a self-gravitating assembly of potentially hundreds of thousands or even millions of components, such an approach is no longer viable. While SSDEM typically requires a smaller time step to resolve particle interactions, it is generally considered the more physically realistic approach and is therefore more desirable. It is also of note that we use the terms hard-"sphere" and soft-"sphere" here to describe these different contact physics schemes. While most DEM codes do use spherical particles for ease of computation, this is not always the case. There are cases in which particle shape is an important variable in modelling small bodies, and we will take a much more careful look at some of these cases in Chapters 3 and 4.

Since its inception, DEM has seen wide application to the study of small bodies. Richardson et al. (1998) used an HSDEM model (a more limited N-body code than `pkdgrav`) to explore tidal disruptions of rubble piles in the vicinity of the Earth across a wide parameter space. Walsh and Richardson (2006) a hard-sphere version of `pkdgrav` to the problem of binary formation among NEAs, finding that tidal disruption can produce binaries similar to those observed. However, Walsh et al. (2008) showed that the thermal Yarkovsky-O'Keefe-Radzievskii-Paddack effect (YORP) is a much more likely formation mechanism for the NEA and main belt binary populations. Sánchez and Scheeres (2014) used an SSDEM code to explore the role of cohesive strength in rubble-pile bodies, concluding that some fraction of the rapidly rotating population of asteroids may actually be rubble piles, although the distinction between between rubble-pile bodies and non-rubble-pile bodies becomes somewhat blurred once sufficient cohesive strength is involved. Finally, this dissertation itself is largely concerned with using a soft-sphere DEM model to study a variety of processes involving small solar system bodies.

1.6 The *N*-body code `pkdgrav`

This section describes `pkdgrav` in some detail, since the work presented in the remainder of the thesis makes use of it. If the reader is not interested in the details of the code, they may skip directly to Section 1.7.

In this dissertation, we use and make substantial modifications to the numerical gravity code `pkdgrav`. `pkdgrav` was initially developed at the University of Washington for the purpose of performing cosmological N-body simulations (Richardson et al., 2000; Stadel, 2001). The code was subsequently altered to handle physical collisions between particles for use in planetesimal modeling. At the time, the approach used an HSDEM scheme. This has since been superseded by a soft-sphere friction model (see Section 1.6.2).

`pkdgrav` uses a hierarchical tree algorithm to conduct a neighbor search before resolving contacts, which reduces the cost of locating neighboring particles to an $\mathcal{O}(N \log N)$ operation, where N is the number of particles. Interparticle gravity is also computed using a tree to speed up the calculations, by replacing $\mathcal{O}(N^2)$ sums over individual particles with multipole expansions of the gravitational potential contributed by small or distant cells. The multipole expansions are taken to hexadecapole order as a middle ground between speed and accuracy. This does introduce some error and asymmetry into the force calculation, typically $\ll 1\%$, depending on how much speed-up is desired. Both the neighbor search and gravity calculation are also parallelized, allowing `pkdgrav` to distribute the work across an arbitrary set of processors for further speed optimization.

1.6.1 Integration scheme

`pkdgrav` uses a fixed-step, second-order leapfrog method to integrate gravity and contact interactions. Briefly, the leapfrog algorithm works as follows. First, initial positions and velocities are provided and resulting accelerations are calculated. The positions and velocities are then subjected to "kicks" and "drifts" in which velocities and then positions are updated alternately. These kicks and drifts follow the scheme laid out in Eq. 1.1, where $\mathbf{r}_{i,n}$ denotes the vector position of particle i at step n. Eq. 1.1a is the opening kick step, wherein particle velocities are updated by a half-step while keeping the positions constant; Eq. 1.1b is the drift step, wherein particle positions are advanced a full step with the velocities held constant; and Eq. 1.1c is the closing kick step, to ensure particle positions and velocities are synchronized at the end of the step. Here h is the time-step interval, typically a small fraction of the shortest dynamical time being modeled in the system. New accelerations $\ddot{\mathbf{r}}_{i,n+1}$ are calculated using the tree code once each time step, after the drift.

$$\dot{\mathbf{r}}_{i,n+\frac{1}{2}} = \dot{\mathbf{r}}_{i,n} + \frac{h}{2}\ddot{\mathbf{r}}_{i,n}, \tag{1.1a}$$

$$\mathbf{r}_{i,n+1} = \mathbf{r}_{i,n} + h\dot{\mathbf{r}}_{i,n+\frac{1}{2}}, \tag{1.1b}$$

$$\dot{\mathbf{r}}_{i,n+1} = \dot{\mathbf{r}}_{i,n+\frac{1}{2}} + \frac{h}{2}\ddot{\mathbf{r}}_{i,n+1}. \tag{1.1c}$$

Leapfrog integration in the gravity problem is symplectic for fixed time steps if the accelerations are a function of position only. This means that the Hamiltonian of the system is preserved from step to step (Saha and Tremaine, 1992; Quinn et al., 2010). Neglecting any dissipation resulting from collisions between particles, the Hamiltonian of an assembly of mutually

self-gravitating particles should remain constant. Thus, the symplectic nature of the leapfrog scheme is desirable for our application. Even when accounting for deviations from the ideal, non-dissipative case, the leapfrog method is advantageous due to its ease of implementation. Recall however that small errors are present in the gravity force calculations due to the tree approximation. Also, the tree structure is not strictly momentum conserving. An "opening angle" parameter allows users to control this behavior, trading accuracy for computational cost. The ideal choice will depend on the physical scenario being modeled.

A complication arises with the soft-sphere collision model. As alluded to above, leapfrog integrators typically are restricted to systems of the form $\ddot{\mathbf{r}} = \mathbf{f}(\mathbf{r})$, i.e., those for which accelerations are a function of particle positions alone. In the soft-sphere model, however, this condition no longer applies. Restoring forces between particles in contact usually have a damping term that is velocity dependent. During the kick phase, a given particle's calculated velocity is "out of sync" with its position by one half time step. But since the soft-sphere forces are velocity dependent, we need an updated estimate of the velocity. To correct for this, we calculate "predicted" velocities and spins for each particle, using a simple (first-order) Euler step:

$$\dot{\mathbf{r}}^{\text{pred}}_{i,n+1} = \dot{\mathbf{r}}_{i,n+\frac{1}{2}} + \frac{h}{2}\ddot{\mathbf{r}}_{i,n}, \tag{1.2a}$$

$$\boldsymbol{\omega}^{\text{pred}}_{i,n+1} = \boldsymbol{\omega}_{i,n+\frac{1}{2}} + \frac{h}{2}\dot{\boldsymbol{\omega}}_{i,n}. \tag{1.2b}$$

(Spins are needed for the contact forces under friction; see Section 1.6.2.2.) A more accurate approach would be to write a modified Hamiltonian that accounts for this velocity dependence in the motion equations and then derive a corresponding leapfrog scheme. However, any improvements in accuracy from this more complex approach are small compared to the energy lost to damping

and friction, especially given the relatively small time steps that we typically use (Schwartz et al., 2012).

1.6.2 Soft-sphere friction model

`pkdgrav` has been modified to include a soft-sphere discrete element method (SSDEM) scheme for treating particle interactions (Schwartz et al., 2012). Unlike HSDEM, SSDEM resolves collisions temporally, allowing particles to interpenetrate slightly as a proxy for surface deformation. `pkdgrav`'s SSDEM implementation uses a spring-dashpot model, in which overlaps between neighboring particles are detected and normal and tangential restoring forces are then modeled as damped springs following Hooke's law with user-adjustable spring and damping constants. The spring forces capture the effects of deformation at particle contacts, while the damping forces capture the effects of kinetic friction. `pkdgrav` uses this approach to track forces and torques from twisting, rolling, and sliding friction. Particle overlaps are tracked for as long as particles remain in contact, and reaction forces depend not only on the degree of overlap at the current time step but also on the contact history. Here, we give an overview of how deformation and sliding friction are implemented in `pkdgrav`. This brief summary draws heavily on Schwartz et al. (2012) and Zhang et al. (2018), and further details can be found in those works. We omit a detailed description of rolling and twisting friction in `pkdgrav`—readers may consult Zhang et al. (2017) or Zhang et al. (2018) for more information on this subject. SSDEM is a substantially more realistic model of the granular physics we are interested in, able to capture multiple simultaneous and persistent contacts self-consistently. These are exactly the sort of interactions at work in small rubble-pile bodies; using an SSDEM model thus allows us to more

accurately simulate the interaction of systems made up of hundreds of thousands of particles.

1.6.2.1 Particle deformation

When computing contact forces between particles in a simulation, a neighbor search is performed. The "overlap" x between neighboring particles is then computed as $x = s_p + s_n - |\mathbf{d}|$, where s_p and s_n are, respectively, the radii of the particle in question and one of its neighbors. $\mathbf{d} = \mathbf{r}_n - \mathbf{r}_p$ is the displacement between the particle and the neighbor center-of-mass positions, so $|\mathbf{d}|$ is the scalar distance between their centers. A nonnegative value of x then indicates that the particle and its neighbor are in contact and will experience a normal restoring force in our implementation following Hooke's law,

$$\mathbf{F}_{N,\text{restoring}} = -k_n x \hat{\mathbf{n}}, \tag{1.3}$$

where $\hat{\mathbf{n}} = \mathbf{d}/|\mathbf{d}|$ is the unit normal vector pointing from the center of the particle toward the center of its neighbor and k_n is the normal spring constant. While k_n is user-adjustable, we typically choose a value that will limit particle overlaps to $\sim$1% of the smaller particle's radius in a given contact, based on the maximum expected impact speeds between particles. This 1% limit is a trade-off between accuracy and performance. In reality, k_n also carries information about the material properties of the constituent particles, including the effective Young's modulus. A more realistic choice of k_n would limit the extent of the overlaps even further, but would require a smaller time step. For further discussion of Young's modulus and k_n in `pkdgrav`, see DeMartini et al. (2019). Particles in contact can experience a relative displacement in the tangential direction

as well. This will give rise to a tangential restoring force given by

$$\mathbf{F}_{T,\text{restoring}} = k_t \mathbf{S}, \tag{1.4}$$

where k_t is the tangential spring constant. This value is user-adjustable, though it is set to a default value of $\frac{2}{7}k_n$ (see Schwartz et al. (2012) for details). $\mathbf{S}$ is the tangential displacement from the equilibrium contact point defined by

$$\mathbf{S} = \int_{\text{overlap}} \mathbf{u}_t(t)dt + \mathbf{S}_0. \tag{1.5}$$

Here, $\mathbf{u}_t(t)$ is the relative tangential velocity of the particles at the contact point and $\mathbf{S}_0$ is the initial tangential extension in a static overlap. The tangential extension is always zero when two particles first come into contact, but if slipping occurs at the contact point, this quantity can take on a non-zero value and will encode information about the contact history.

1.6.2.2 Friction at particle contacts

In addition to particle overlap as a proxy for deformation during particle contacts, a model for kinetic and static friction is included. The kinetic friction between two particles depends on the relative velocity at the contact point (see Eq. 1.7), which must account for both the center-of-mass (COM) velocities and the relative spins of the two particles. The "lever arm" l_p of a particle involved in a contact is defined to be the distance between the center of the particle and the center of the overlap region. The lever arm is not a fixed characteristic of a particle, but depends on the radii of the two particles in contact and the degree of overlap they experience on a given time

step. This distance is given by

$$l_p = \frac{s_p^2 - s_n^2 + |\mathbf{d}|^2}{2\,|\mathbf{d}|}. \tag{1.6}$$

A neighboring particle in contact will have a corresponding lever arm given by $l_n = |\mathbf{d}| - l_p$. We can then define the relative velocity ($\mathbf{u}$) at the contact point,

$$\mathbf{u} = \mathbf{v}_n - \mathbf{v}_p + l_n(\hat{\mathbf{n}} \times \boldsymbol{\omega}_n) - l_p(\hat{\mathbf{n}} \times \boldsymbol{\omega}_p), \tag{1.7}$$

where $\mathbf{v}_p$ and $\boldsymbol{\omega}_p$ are the COM velocity and angular velocity of the particle and $\mathbf{v}_n$ and $\boldsymbol{\omega}_n$ are the same quantities for the neighboring particle in question.

We apply damping forces to both normal and tangential relative motion of overlapping particles,

$$\mathbf{F}_{N,\text{damping}} = C_n \mathbf{u}_n, \tag{1.8}$$

$$\mathbf{F}_{T,\text{damping}} = C_t \mathbf{u}_t, \tag{1.9}$$

where $\mathbf{u}_n = (\mathbf{u} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ is the normal component of relative velocity and $\mathbf{u}_t = \mathbf{u} - \mathbf{u}_n$ is the corresponding tangential component. C_n and C_t are the normal and tangential damping coefficients, respectively, discussed further below. Combining our deformation and kinetic friction contact forces (Eqs. 1.3, 1.4, 1.8, and 1.9), we have

$$\mathbf{F}_N = -k_n x \hat{\mathbf{n}} + C_n \mathbf{u}_n, \tag{1.10}$$

$$\mathbf{F}_T = k_t \mathbf{S} + C_t \mathbf{u}_t. \tag{1.11}$$

Equal and opposite forces are then applied to the neighboring particle. While the damping forces are written here in terms of C_n and C_t, we may also wish to know how these values relate to the normal and tangential coefficients of restution, ε_n and ε_t. C_n can be expressed neatly as a function of ε_n, k_n, and the reduced mass μ of the two particles in contact, defined as $\mu = m_p m_n/(m_p + m_n)$:

$$C_n = -2 \ln \varepsilon_n \sqrt{\frac{k_n \mu}{\pi^2 + \ln^2 \varepsilon_n}} \tag{1.12}$$

Unfortunately, there is no such simple correspondence between C_t and ε_t. Instead, the effective ε_t will depend on the characteristic oscillation frequencies of both the normal and tangential spring in a complicated function that cannot be solved analytically for C_t. In practice, we use an empirically determined value for ε_t that produces physical behavior in our simulations. For a more detailed discussion of the relation between damping and restitution coefficients, see Schwartz et al. (2012).

In addition to the kinetic friction scheme, a model of static friction is also included. The distinction between static and kinetic friction is important when a contact between two particles alternates between sticking and slipping states. This is a result of the interlocking that occurs between microscopic surface features when two particles in contact are subjected to shearing. We assume that the static friction force is proportional to the normal force, with the proportionality constant being the coefficient of static friction, μ_s:

$$|\mathbf{F}_{T,\max}| = \mu_s |\mathbf{F}_N| . \tag{1.13}$$

This is the maximum tangential force that can be supplied by static friction. If this force is

exceeded, the contact slips and we reset the tangential displacement $\mathbf{S}$ to zero. The complete tangential force can then be written as

$$\mathbf{F}_T = \min \left\{ \mu_s \left| \mathbf{F}_N \right| \hat{\mathbf{S}}; k_t \mathbf{S} + C_t \mathbf{u}_t \right\}, \tag{1.14}$$

where $\hat{\mathbf{S}} \equiv \mathbf{S}/\left|\mathbf{S}\right|$. This particle and its neighbor will both feel (equal and opposite) tangential forces acting on their COM, as well as corresponding torques at the contact point. The torques will depend on the lever arms of the respective particles.

1.7 This dissertation

This subject of this dissertation is the mechanical evolution of small solar system bodies. In particular, we are interested in studying how low-energy deformations, disruptions, and even collisions can shape small objects of rubble-pile composition. These processes are quite common in the solar system and, as we have seen in this chapter, are of significant interest to planetary science. Our primary tool in this investigation is the SSDEM-equipped *N*-body code `pkdgrav`, which we described in the preceding section.

Chapter 2 of this thesis describes our application of `pkdgrav` to the Kuiper belt contact binary Arrokoth, which we discussed in Section 1.3. This chapter was published in the planetary science journal *Icarus* and is presented here with minimal modification. Our aim was to understand how such an object could form in the Kuiper belt from two previously unconnected planetesimals. We conduct simulations modeling a variety of merger scenarios and find that only a very gentle collision following a slow, inspiraling orbit could produce an object with Arrokoth's current shape.

In Chapter 3, we describe our modifications to `pkdgrav` that allow for the use of non-spherical particles in SSDEM simulations. While previous iterations of `pkdgrav` had added this feature in the hard-sphere context (Richardson et al., 2009), significant changes were required to apply realistic soft-sphere contact physics to non-spherical particles. We also include three small-scale proof-of-concept studies to demonstrate the value of our additions to the code and describe some additional modifications that greatly improve the efficiency of integrating our non-spherical particles. This work also forms the basis for the study of tidal disruption in Chapter 4. As of this writing, this chapter has been accepted for publication in *The Planetary Science Journal*. We had intended to include an additional chapter leveraging these improvements to study particle shape in rubble-pile spin-up processes. Unfortunately, time constraints pushed this project out of the scope of this dissertation.

Chapter 4 applies the code modifications from Chapter 3 to the tidal disruption of small bodies. We discuss the importance of particle shape in granular media generally and use `pkdgrav`'s new capabilities to investigate the role of particle shape in tidal deformation and disruption. We conduct $\sim$1,900 tidal disruption simulations and find that particle shape has a significant effect on simulation outcomes in a number of different contexts. We plan to submit this chapter for publication soon after this dissertation is accepted, pending some expected contributions from a collaborator.

Finally, Appendix B documents `pkdtools`, the sizable software package developed to handle and analyze the terabytes of data produced by the study described in Chapter 4. `pkdtools` has been distributed to the wider community of `pkdgrav` users and has already been used in production work outside of this dissertation.

Chapter 2: Origins of the Kuiper belt contact binary Arrokoth

2.1 Preface

This work was motivated by the *New Horizons* flyby of Arrokoth in 2019. The *New Horizons* Science team was interested in using `pkdgrav` simulations to try to understand how a contact binary like Arrokoth could be formed in the Kuiper belt. I was tasked with conducting simulations in service of this project, and the work I did there formed the basis for my Master's thesis. This chapter was published in the journal *Icarus* with the title "Constraining the final merger of contact binary (486958) Arrokoth with soft-sphere discrete element simulations" (Marohnic et al., 2021). The simulations and conclusions also make up a substantial portion of McKinnon et al. (2020), published in *Science*.

The great majority of the work presented here is my own. Two other graduate students contributed work that is included here. First, one of the simulations described in the chapter was conducted by Harrison Agrusa. Second, Joseph DeMartini helped me in using the visualization software Paraview to add colors to the acceleration maps we show. I carried out the remainder of the simulations and analysis and wrote the manuscript, with copious editing and insight from my advisor, Derek Richardson. This work is presented largely as it appeared in *Icarus*, save for formatting and the Acknowledgments section.

2.2 Introduction

2.2.1 (486958) Arrokoth

After exploring the Pluto system in 2015, the New Horizons mission was redirected to include a visit to the Kuiper belt object (486958) Arrokoth. On January 1, 2019, the New Horizons spacecraft passed within 3550 kilometers of the body (Stern et al., 2019). Arrokoth lies in the cold classical Kuiper belt and is a bilobate contact binary—we will refer to the larger of the two lobes as "LL" and to the smaller as "SL," following McKinnon et al. (2020) (see Figure 2.1). LL and SL have measured diameters of 19.5 km and 14.2 km, respectively. Both lobes approximate flattened or oblate spheres in shape. They are joined by a narrow contact region, which we will refer to as the "neck." As is typical for a cold classical Kuiper Belt Object (CCKBO) (Gulbis et al., 2006; Noll et al., 2008), Arrokoth is red in color (Benecchi et al., 2019a), though the neck is slightly less red and noticeably brighter than most of the rest of the object. Its rotation period is 15.92 ± 0.1 hours. Thus far, no evidence of satellites larger than $\sim$2 km in diameter has been found, nor has any sign of gas or dust in the vicinity of Arrokoth (Stern et al., 2019). Given its heliocentric distance, its surface temperature is likely somewhere between 30 and 60 K (Prialnik et al., 2008). Its orbit has semimajor axis, eccentricity, and inclination of 44.08 au, 0.035, and 2.4°, respectively (Stern et al., 2018; Benecchi et al., 2019b).

2.2.2 The origins of Arrokoth

As a CCKBO, Arrokoth is likely among the most primitive bodies in the solar system. These bodies are thought to have formed in situ from the solar nebula approximately 4.5 Gyr ago

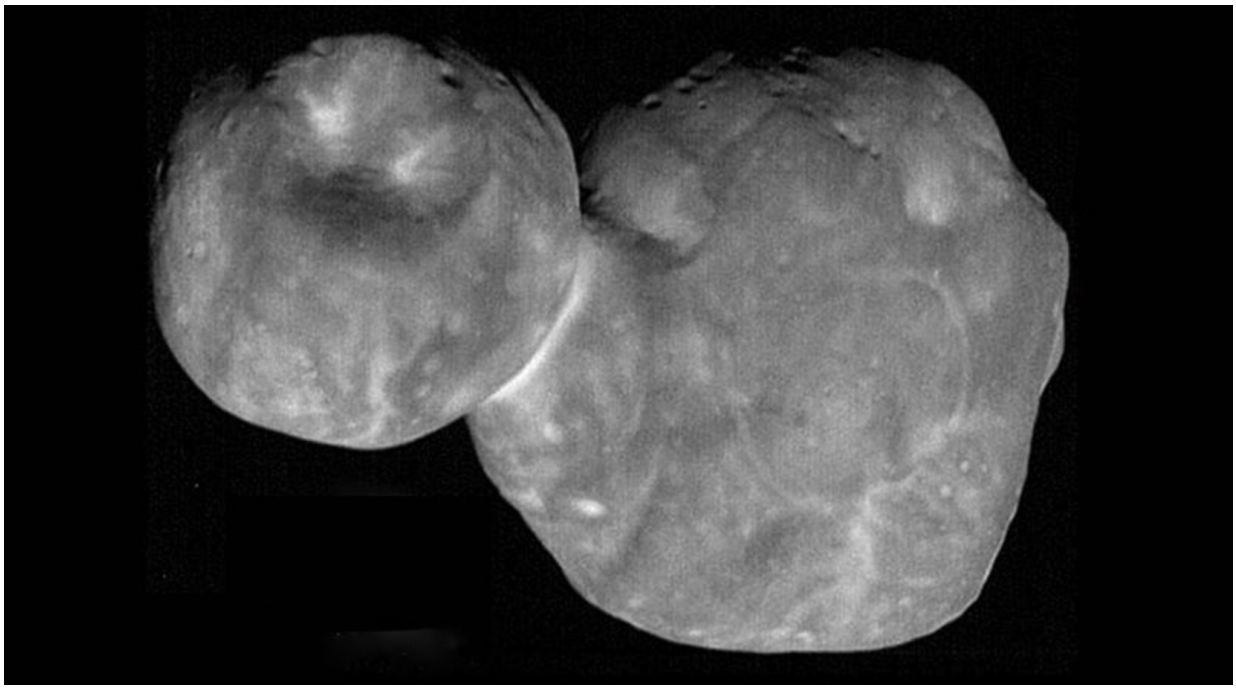


Figure 2.1: An image of Arrokoth captured by New Horizons during its flyby. (Credit NASA, JHU APL, Southwest Research Institute, ESA)

and remained nearly undisturbed since (Batygin et al., 2011). Considering its unusual shape, it seems likely that the present-day Arrokoth was created by the merger of two progenitor Kuiper Belt Objects (KBOs), which could have comprised a binary system prior to the merger or could have collided by chance. The present configuration and characteristics of Arrokoth may be able to give insight into the formation of Arrokoth in particular and into the formation of KBOs and Kuiper belt binaries (KBBs) in general.

Numerous possible formation mechanisms for KBOs have been proposed, which can be divided into two broad categories. The first is hierarchical coagulation (HC), in which successive two-body collisions lead to the gradual accretion of large objects. Competition between mergers on one hand and fragmentation on the other determine the outcome and the time required to form KBOs. Studies by Stern and Kenyon, among others, conclude that in order to form the Kuiper belt as it exists today in $\lesssim 10^8$ years via HC, orbits in the primordial belt would need to have

had $e \sim 10^{-4}$ and $i \sim 10^{-2}$ (Stern, 1996; Kenyon, 2002; Nesvorný et al., 2010). The question of how binary KBOs might form under an HC model has been approached from many directions. Three-body encounters could leave behind a binary when a third object is ejected, removing angular momentum from the system (Goldreich et al., 2002). Two KBOs might enter an unstable orbit by entering each other's Hill spheres or via chaos-assisted temporary capture and transition to a stable orbit once dynamical friction from background planetesimals removes excess energy (Goldreich et al., 2002; Astakhov et al., 2005; Lee et al., 2007; Schlichting and Sari, 2008). Additionally, two bodies might collide within the Hill sphere of a third body, dissipating enough energy to form a stable binary with a mass ratio near one (Weidenschilling, 2002), a common characteristic of KBBs (Noll et al., 2007; Schlichting and Sari, 2008). Such collisions could also produce a binary with a much larger mass ratio that then undergoes a series of exchange reactions with other KBOs, leaving behind a much larger secondary (Funato et al., 2004).

An alternative to HC is gravitational instability (GI), in which gravitationally bound clouds of particles form in the thick midplane of the disk or via the streaming instability and collapse into large objects on short timescales (Nesvorný et al., 2010). Any angular momentum this system has will encourage the formation of two or more bound objects of similar sizes. Forming binaries via GI also gives a natural explanation for the observed correlation in color of KBBs (Benecchi et al., 2009), as binaries formed locally would have the same composition and thus the same color. More recently, Nesvorný et al. (2019) showed that GI can reproduce the observed KBB inclination distribution, while HC models cannot. While the simulations presented in this study model only the final stage of the contact binary formation, understanding the origins of the progenitor bodies helps to put the merger in context.

In spite of the merger that likely formed Arrokoth, LL and SL individually appear to be

quite intact. This suggests that the contact binary was created by a low-speed collision between two primordial CCKBOs. Given their relatively small sizes, great distance from the Sun, and the fact that they lie in the dynamically cold segment of the Kuiper belt, CCKBOs are unlikely to have undergone any significant processing after their initial formation. As such, they can provide us with a unique probe into conditions in the outer solar nebula, and Arrokoth in particular should be able to give us insight into the formation of KBBs and of planetary bodies in general (Stern et al., 2019; McKinnon et al., 2020).

2.2.3 Objectives

We detail and expand on the numerical simulations introduced in McKinnon et al. (2020) to match the morphology and dynamical state of Arrokoth following the merger of its contact binary components. We constrain the merger circumstances by running a suite of simulations with different initial conditions and comparing the outcomes with observations made by the New Horizons spacecraft. Specifically, we consider the overall shape and final spin state of Arrokoth. We apply these methods exclusively to the final stages of the merger and do not model the genesis of the binary system prior to the merger event.

Prior work has used numerical models to study the formation of bilobate comets and asteroids via mergers and collisions. Jutzi and Asphaug (2015) used an SPH code to simulate collisions between relatively small objects (0.1-1 km) at or just above the mutual escape speed, producing some bilobate objects. Subsequent studies have focused on higher-speed impacts, at 200 m/s or above, of small impactors hitting much larger target bodies (Jutzi and Benz, 2017; Jutzi, 2019; Sugiura et al., 2020), though Schwartz et al. (2018) considers impacts as slow as 20

m/s. Most of this work was done with objects like comet 67P/Churymov-Gerasimenko in mind. They do not attempt to produce anything like the narrow neck and intact lobes that we see in Arrokoth, and focus on catastrophic or sub-catastrophic collisions that produce far more disruption than must have been the case in the merger that produced the Arrokoth contact binary. Nesvorný et al. (2010) considers the possibility that 67P began as a binary and collapsed. They simulate a variety of merger scenarios and produce some very gentle collisions, but focus primarily on the dynamics of the binary collapse and do not carefully examine the merger itself. While we consider the possibility of a collision between unbound bodies in the case of Arrokoth, we also include a variety of simulations in which Arrokoth forms as the result of a very gentle merger between two bound objects.

2.3 Method

2.3.1 `pkdgrav`

To model the merger of Arrokoth, we use `pkdgrav`, a parallel N-body tree code (Stadel, 2001) with an implementation of the soft-sphere discrete element method (SSDEM) for collisions between spherical particles (Schwartz et al., 2012). The SSDEM model works by allowing particles to interpenetrate slightly as a proxy for surface deformation, with restoring forces implemented as damped springs with a user-adjustable spring constant. The model includes a detailed implementation of static, rolling, and twisting friction (Zhang et al., 2017) along with interparticle cohesion (Zhang et al., 2018).

Given that `pkdgrav` is a parallel tree code, it can simulate systems of hundreds of thousands of particles quickly. This allowed us to run a relatively large number of high-resolution

simulations to explore the parameter space. Furthermore, its explicit treatment of particle contacts makes `pkdgrav` well-suited to this energy regime. We are aware of at least one other numerical approach to modeling the final stage of the formation of Arrokoth (Wandel et al., 2019), but not enough detail has been published at the time of this writing to allow a meaningful comparison between these approaches.

When used in this study, cohesive forces were only applied between particles of the same progenitor body. In other words, a contact between a particle from LL and a particle from SL would be treated as cohesionless. This choice was motivated by some of our early simulations of the binary merger with cohesion included. After an initial contact between bodies, the size of the neck would continue to grow as particles near the contact point stuck together and pulled others along with them. Because we wanted to use cohesion to capture the effect of material strength, we judged this behavior to be unphysical and made this adjustment to our model. The effects of cohesion in our simulations are discussed in greater detail in Section 2.4.

2.3.2 Acceleration mapping

To assess the extent to which the final merger disturbs surface material, we measured the net accelerations felt by particles on the surface of LL and SL throughout their mutual approach and impact. Specifically, we tracked accelerations due exclusively to contact forces between particles and recorded the maximum such acceleration experienced up to each point in the simulation by each particle. The aim of recording contact accrelerations was to observe shocks caused by the collision of the two bodies and to determine if this might have left a signature on the surface of Arrokoth.

2.3.3 Setup of LL and SL

We model each lobe of the present contact binary system separately, under the assumption that they formed separately and merged at some point in the past. We generate spherical "rubble piles" out of many smaller particles. To create the oblate spheroids used in some of our simulations, we begin with the spherical bodies and remove particles as needed to "carve out" the appropriate shapes. Thus, all oblate progenitor objects in this study have volumes less than their corresponding spherical progenitor bodies. The specific parameters used to generate LL and SL respectively (size, shape, number of particles, particle density) depend on the impact case being modeled (discussed in the following section). In the case of spherical components,[1] we use approximately 135,000 and 63,000 particles to model LL and SL, respectively. In our simulations, we adopted spherical radii of R_{LL} = 8.97 km for LL and R_{SL} = 6.82 km for SL. Because composing a body from equal-size particles adds excess shear strength that a real rubble pile would not likely have, we use a distribution of particle sizes (Quillen et al., 2016; DeMartini et al., 2019). Particle radii are normally distributed with a mean radius of 136 m and a standard deviation of 27 m. The largest and smallest radii included are 163 m and 109 m, respectively. Particles in a given simulation have a uniform mass density, though different densities are used in different runs. Most of the simulations presented here use a bulk density of 0.5 g/cm^3, though we include some simulations with lower densities (see Table 2.1). Note that we measure the mass density of particles in a given simulation to be approximately twice the bulk density, due to the macroporosity of the rubble piles. In simulations with oblate lobes, the number of particles is in proportion to the decrease in volume from the original spherical shape. After generating LL and SL we run

[1] When this study began, the components were thought to be spherical. Although we now know they are oblate, the spherical case is useful for comparison purposes. See Section 2.4.

simulations with each body separately to allow the particles to settle into an equilibrium between self-gravity and repulsive contact forces.

Particles are either fricitionless or given "gravel" friction parameters—with static, rolling, and twisting coefficients of friction of 1.0, 1.05, and 1.3, respectively. These parameters are representative of a typical rough material surface, and can be considered the opposite of the frictionless case. They correspond to material with a friction angle of approximately 39° for a random, polydisperse packing of spheres, similar to coarse, rocky materials on Earth (Zhang et al., 2017). It should be noted here that the exact interior composition of Arrokoth is not known, although it is likely some combination of icy and rocky material (Stern et al., 2019). The friction model in `pkdgrav` is largely designed to capture the effects of particle shape and internal friction rather than composition. On the assumption that Arrokoth is made up of irregular particles, we chose friction parameters that correspond to material with a reasonably high internal friction. We use a time step of 0.037 seconds (chosen to sample the soft-sphere springs adequately) and run each simulation for one million steps, or about 10.3 hours. We aim to keep soft-sphere particle overlaps to $\sim$1% of the smallest particle radius, so we take into account expected particle collision speeds and pressures and set the contact force spring constant accordingly—the value used in the simulations presented here is 8.0 x 10^{10} kg/s^2. We use a shape parameter of 0.5 and normal and tangential coefficients of restitution are both set to 0.2. The shape parameter describes how out-of-round the particles are treated for the purpose of computing friction and cohesion; smaller values indicate rounder particles and 0.5 has been found to be a good representation for gravel-like material. Our choice of restitution coefficients in this work results in more dissipation than for a canonical gravel-like material (with restitution coefficients of 0.55), but serves as a proxy

for some crushing that is not otherwise captured in our code.[2] Consequently, our results have the caveat that there may be more or less dissipation than is modeled here, which could alter the critical impact speed somewhat (Leinhardt et al., 2000). The increased dissipation likely favors less body shape deformation at impact and a narrower neck. See Zhang et al. (2017) for more discussion of the parameters described in this section and their implementation in `pkdgrav`.

[2]Estimates of the compressive strength of comet 67P/Churyumov-Gerasimenko are about 1 kPa (Heinisch et al., 2019). A rough calculation suggests that the impact pressure felt by the two components of Arrokoth colliding at 5 m/s would be several orders of magnitude larger than this. On the other hand, the pressure cannot be so large that it visually distorts the bodies at the point of contact, as this is not evident in observations of Arrokoth. Thus, we can reasonably expect a moderate, but not excessive, amount of crushing.

	Impact speed (m/s)	Impact angle	Friction	ρ (g/cm^3)	Cohesion (Pa)	c_{LL}	c_{SL}	Spins
Slow inspiral								
Inspiral 1	2.9	80°	Gravel	0.5	0	1	1	Synchronous
Inspiral 2	2.9	80°	Gravel	0.5	275	1	1	Synchronous
Inspiral 3	2.9	80°	Gravel	0.5	2750	1	1	Synchronous
Inspiral 4	2.9	80°	Gravel	0.5	27500	1	1	Synchronous
Inspiral 5	2.1	80°	Gravel	0.25	0	1	1	Synchronous
Inspiral 6	2.1	80°	Gravel	0.25	275	1	1	Synchronous
Inspiral 7	2.1	80°	Gravel	0.25	2750	1	1	Synchronous
Inspiral 8	2.1	80°	Gravel	0.25	27500	1	1	Synchronous
Inspiral 9	1.7	80°	Gravel	0.16	0	1	1	Synchronous
Inspiral 10	1.7	80°	Gravel	0.16	275	1	1	Synchronous
Inspiral 11	1.7	80°	Gravel	0.16	2750	1	1	Synchronous
Inspiral 12	1.7	80°	Gravel	0.16	27500	1	1	Synchronous
Inspiral 13	2.9	80°	None	0.5	0	1	1	Synchronous
Inspiral 14	2.9	80°	Gravel	0.5	0	1	1	None
Inspiral 15	2.9	80°	Gravel	0.5	1000	1	1	Equal, opposite
Oblate								
Oblate 1	2.5	80°	Gravel	0.5	0	1.5	1	Synchronous
Oblate 2	2.5	80°	Gravel	0.5	0	2	1.2	Synchronous
Oblate 3	2.5	80°	Gravel	0.5	0	2	1	Synchronous
Oblate 4	2.5	80°	Gravel	0.5	0	2	2	Synchronous
Oblate 5	2.5	80°	Gravel	0.5	0	3	1.4	Synchronous
Impacts								
Impact 1	5.0	45°	Gravel	0.5	1000	1	1	None
Impact 2	5.0	65°	Gravel	0.5	1000	1	1	None
Impact 3	5.0	75°	Gravel	0.5	1000	1	1	None
Impact 4	5.0	85°	Gravel	0.5	1000	1	1	None
Impact 5	10.0	45°	Gravel	0.5	1000	1	1	None
Impact 6	10.0	65°	Gravel	0.5	1000	1	1	None
Impact 7	10.0	75°	Gravel	0.5	1000	1	1	None
Impact 8	10.0	85°	Gravel	0.5	1000	1	1	None
Impact 9	1.0	45°	None	0.5	0	1	1	None
Impact 10	2.0	75°	None	0.5	0	1	1	None

Table 2.1: A summary of the simulations performed. ρ is the mass density and c_{LL} and c_{SL} are ratios of the long axes to the short (flattened) axes of LL and SL, respectively. Values listed in the sixth column are all *interparticle* cohesive strengths.

2.3.4 Impacts and merger events simulated

As the initial returns from New Horizons suggested a contact binary composed of two *spherical* lobes, we began with a suite of runs modeling mergers of two spherical bodies. While the shapes of LL and SL are now understood to be more similar to oblate spheroids, the spherical cases still serve as useful end members of a distribution of possible shapes and illustrate the effects of different parameters on the outcome of the merger event. Further, we include a number of scenarios in which the progenitor bodies are modeled as oblate spheroids, with one simulation matching the best-available shape model at the time of writing. While the actual shapes of LL and SL are not perfect spheroids, for the sake of simplicity we restrict ourselves to symmetric progenitors. Our range of flattened shapes was informed by the uncertainty measure of the observations. As this study was conducted, the best estimates of the shapes were evolving, and this is reflected in our choice of progenitor shapes.

To better understand the merger scenarios that could have led to the creation of Arrokoth as it exists today, we carried out a series of simulations, varying impact angle and impact speed, progenitor material cohesion, shape, density, and spin. We divide our simulations into three general groups: "slow inspiral", "oblate", and "impacts". Each group consists of a grid search in which we vary different parameters to determine their effects on the final contact binary. See Table 2.1 for a summary of simulations performed in this study.

The "slow inspiral" group attempts to model the final stage of a close, tidally locked orbit decaying toward an impact. In runs Inspiral 1–12, we vary density and cohesion. Based on the measured bulk densities of cometary nuclei (Groussin et al., 2019) (for example, comets Tempel 1 and Churyumov-Gerasimenko (Samarasinha et al., 2009; Jorda et al., 2016)), we use 0.5 g/cm^3

as a nominal starting point. We also test values of 0.25 g/cm^3 and 0.16 g/cm^3. Four different cohesion values are also tested. To obtain a "base" cohesion value, we divide the gravitational force felt by a particle at the surface of the spherical LL progenitor by the cross-sectional area of a mean-radius particle. This is equivalent to approximately 275 Pa. With this calculation, we do not attempt to take into consideration any measured or theoretical value for the interparticle cohesion of a KBO, but simply to get an idea of the scale appropriate to the body. This is all that is really necessary here, since we test a broad range of cohesion values anyway. We simulate inspiral scenarios with cohesion values of 0, 1, 10, and 100 times this value. We repeat this set of tests for the three bulk densities listed above. It should be noted that the values quoted here are *interparticle* cohesion values, and unless explicitly stated otherwise, all mentions of cohesion in this paper refer to interparticle cohesion. The equivalent bulk cohesion is roughly 100 times smaller than these values for the type of material used in this study (Zhang et al., 2018). Also included in this group are three runs in which friction and body spin are varied for an inspiraling orbit (Inspiral 13–15).

The "oblate" group also consists of two tidally locked bodies spiraling inward, but uses oblate spheroids of varying axis ratios. Density is held constant at 0.5 g/cm^3 and all runs in this group are cohesionless. The minor axes of the spheroids are parallel during the approach. At the time of writing, Oblate 5 represents the best oblate spheroid approximation of the true shapes of LL and SL.

The "impacts" group does not assume that LL and SL are gravitationally bound before the merger, but tests direct impacts, varying collision angle and speed. We define the "impact angle" as the angle between the relative velocity vector and the line connecting the centers of the two bodies at contact. Impact 1–8 test impact speeds of 5 m/s and 10 m/s at angles of 45°, 65°, 75°,

and 85° and use a cohesive strength of 1 kPa. It also includes two frictionless, low-speed impacts that we use to evaluate the effects of material type (Impact 9–10).

To get an estimate of the impact speeds relevant to the problem, we estimate the mutual escape speed of LL and SL. If we make the assumption that LL and SL are spheres in contact at a single point, then their mutual escape speed can be derived by setting the two-point-mass total energy to zero:

$$\frac{v_{esc}^2}{2} - \frac{GM}{R} = 0, \tag{2.1}$$

where v_{esc} is the relative velocity between LL and SL, M is their combined total mass, and R is the distance between their centers. Their escape speed is then given by

$$v_{esc} = \sqrt{\frac{2GM}{R}}. \tag{2.2}$$

For a bulk density of 0.5 g/cm^3 and radii of R_{LL} = 8.97 km and R_{SL} = 6.82 km, this gives an escape speed of 4.3 m/s. The escape speed should be close to the minimum possible impact speed we might expect for progenitor bodies *not* in orbit about one another (i.e., zero relative speed at infinite distance). This motivates us to test direct impacts at 5 m/s and 10 m/s. If the lobes cannot survive collisions near the mutual escape speed, formation by direct impact becomes unlikely.

In cases where we wanted to reproduce a gradual spiral-in process with an impact angle near 90° (those in the "slow inspiral" and "oblate" groups), we initiated the simulations with the bodies in an approximately circular orbit. To calculate the appropriate speeds, we start with Kepler's third law

$$P^2 = \frac{4\pi^2}{GM} a^3, \tag{2.3}$$

where P is the orbital period of the system, a is the semimajor axis of the orbit, and M is the total mass. Let R_{LL}, R_{SL}, M_{LL}, and M_{SL} be the radii and masses of LL and SL. We give LL and SL an initial separation of $1.04(R_{LL} + R_{SL})$ (leaving the bodies slightly separated to avoid particle overlaps) and a total mass of $M = M_{LL} + M_{SL}$. We are interested in producing a grazing contact, but due to the irregularities of the surface, setting up scenarios precisely so that they result in a specific point of contact is difficult. To obtain a reasonable starting point, we calculate the orbital speed that the bodies would have if they were in contact at a single point. To find the period for our system, we solve for P and substitute:

$$P = \sqrt{\frac{4\pi^2}{G(M_{LL} + M_{SL})}} \, (R_{LL} + R_{SL})^{3/2} \tag{2.4}$$

The orbital speed for each body is then given by

$$v_{orb} = \frac{2\pi r_{orb}}{P}, \tag{2.5}$$

where r_{orb} is the orbital radius of each body (4.9 km for LL and 11.5 km for SL, relative to their mutual center of mass). This gives a v_{orb} of 0.9 m/s for LL and 2.0 m/s for SL in the case of Inspiral 1. In order to produce a collision in a reasonable time span, we impart both LL and SL with 90% of the v_{orb} values calculated in the manner described above. We emphasize that these initial speeds will not result in a circular orbit with the bodies in contact at precisely one point, but a slightly elliptical trajectory (representing the end-stage of an inspiral) that leads to a

collision at nearly 90°. In any case, any departures from sphericity in the progenitor objects will mean that these calculations overestimate their speeds. As we are interested in the collisions and not the orbits themselves, this difference is unimportant.

2.3.5 Neck measurements

In order to compare the results of our simulations to each other and the observed system, we devised a method for computing the neck widths of the contact binaries produced by our simulations. We divide the longest axis of the body into equal-length segments. All particles falling inside a segment are grouped together, creating cross-sectional "slices" of the object. For each slice, we calculate the maximum perpendicular distance between a particle in that slice and the central axis of the body. This allows us to get an estimate of the width of the object at a given point along the body's long axis, even when the object is not symmetrical. To calculate the width of the neck, we find the minimum width near the middle of the long axis. We report these measured values in Table 2.2. For severely misshapen bodies, this method does not give a meaningful result, but for contact binaries similar in shape to Arrokoth, it gives a good quantitative estimate of the neck width. We report neck widths as a fraction of Arrokoth's observed long axis length. We use the value 0.184 for the physical Arrokoth as a point of comparison, though the exact figure depends on how one measures since the neck is not symmetric.

2.4 Results

Results of the simulations are summarized in Table 2.2, showing the run label, final spin period, neck width, and a brief description of the final state of the lobes. Entries with "N/A" for

the period and neck width were cases that did not result in a contact binary. In the remainder of this section, we provide more detail on the results of the major parameters we tested, namely friction, bulk density, cohesion, initial spin, component shape, and impact speed and angle. A discussion of these results, which are summarized in Figure 2.9, is given in Section 2.5.

2.4.1 Effects of friction

Impact 9, Impact 10, and Inspiral 13 are frictionless, while all other runs use the "gravel" friction parameters described in Section 2.3. Impact 9 and Impact 10 both result in the total loss of the bilobate shape after contact merger—the progenitor bodies deform smoothly into one large mass (see Figure 2.2). Inspiral 13 preserves the shapes of LL and SL, but still produces a very wide neck. We conclude that material friction is important in maintaining the overall shape and neck width of the final shape of the contact binary.

2.4.2 Density, cohesion, and spin

Inspiral 1–12 vary bulk density between 0.16 g/cm^3 and 0.5 g/cm^3 and interparticle cohesion between 0 and 27.5 kPa. At 0.5 g/cm^3 (Inspiral 1–4), varying the cohesion does not have a strong effect on the outcome. Final spins range from 8.6 h to 9.0 h and neck widths vary between 0.179 and 0.225. We do notice, however, that at the highest cohesion value tested, we see somewhat more deformation, a slight increase in neck width, and a slightly shorter spin period (see Section 2.5 for further discussion of this trend). Acceleration maps show disruption confined to the neck area (see Figures 2.3 and 2.4).

At lower densities, this trend becomes more pronounced. For bulk densities of 0.25 g/cm^3

	Final spin period (h)	Neck width	Final state of lobes
Slow inspiral			
Inspiral 1	9.0	0.184	Intact
Inspiral 2	9.0	0.179	Intact
Inspiral 3	8.9	0.184	Intact
Inspiral 4	8.6	0.225	Slight deformation
Inspiral 5	12.9	0.179	Intact
Inspiral 6	12.9	0.179	Intact
Inspiral 7	12.7	0.184	Intact
Inspiral 8	12.3	0.276	Moderate deformation
Inspiral 9	16.1	0.189	Intact
Inspiral 10	16.1	0.189	Intact
Inspiral 11	15.8	0.194	Intact
Inspiral 12	15.9	0.292	Significant deformation
Inspiral 13	8.9	0.332	Significant deformation
Inspiral 14	11.6	0.317	Slight deformation
Inspiral 15	9.6	0.297	Slight deformation
Oblate			
Oblate 1	9.7	0.174	Intact
Oblate 2	10.7	0.169	Intact
Oblate 3	10.4	0.169	Intact
Oblate 4	11.5	0.179	Intact
Oblate 5	11.7	0.205	Intact
Impacts			
Impact 1	9.0	0.358	Significant deformation
Impact 2	N/A	N/A	Moderate deformation
Impact 3	N/A	N/A	Slight deformation
Impact 4	N/A	N/A	Intact
Impact 5	N/A	N/A	Significant deformation
Impact 6	N/A	N/A	Moderate deformation
Impact 7	N/A	N/A	Slight deformation
Impact 8	N/A	N/A	Intact
Impact 9	N/A	N/A	Completely merged
Impact 10	N/A	N/A	Completely merged

Table 2.2: A summary of the outcomes of the simulations. Final spin period and neck width are listed only if a recognizable contact binary results from the collision. Neck width is given as a fraction of Arrokoth's observed long axis length (35 km)

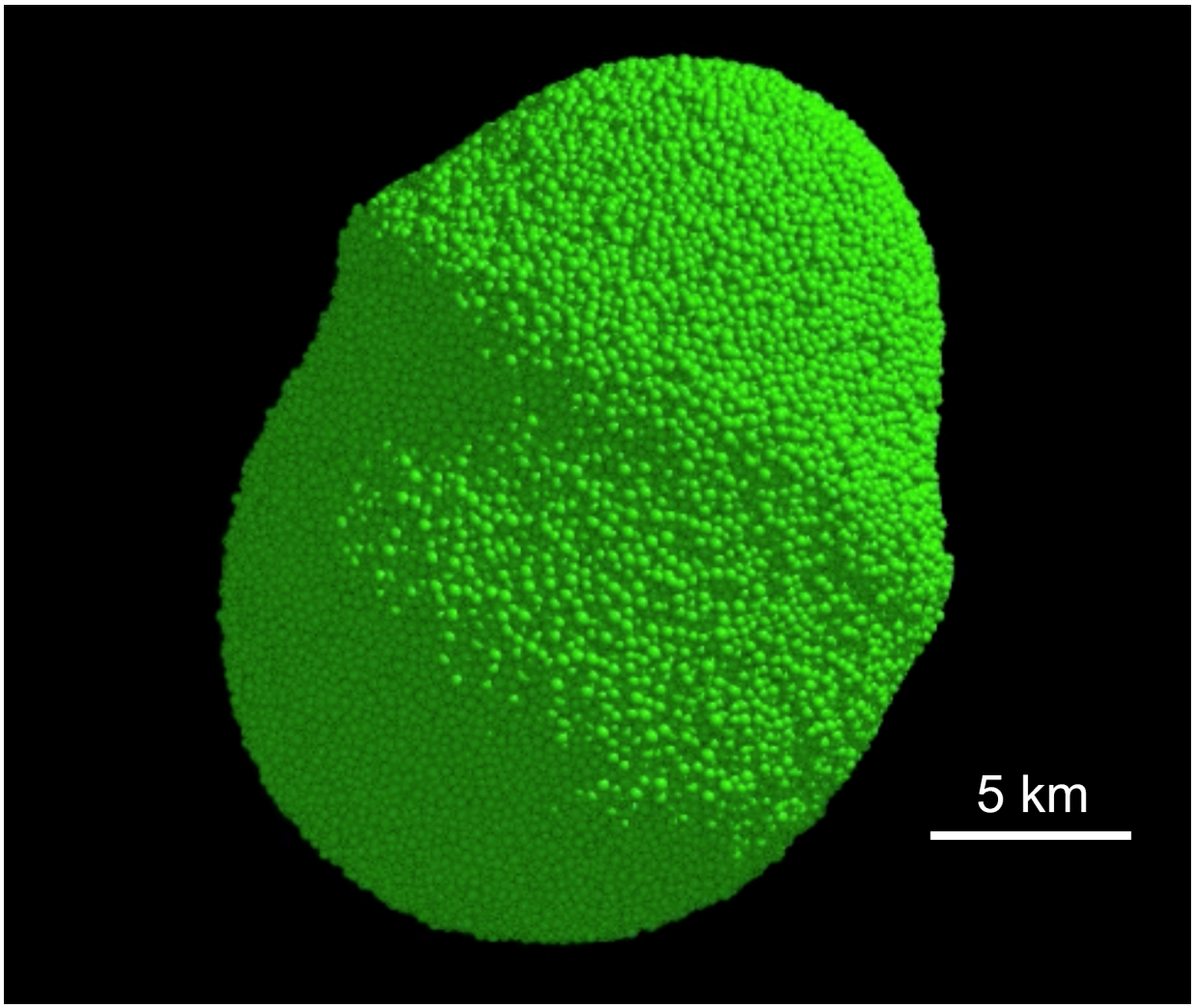


Figure 2.2: Impact 9. 45° impact angle, 1 m/s, frictionless. LL and SL merge almost completely, leaving little evidence that the object was once a binary.

(Inspiral 5–8), spin periods are 12.9 h and neck widths are 0.179 for the cohesionless and 275 Pa cases. At 27.5 kPa, the spin period falls to 12.3 h, the final neck width is 0.276, and the lobes deform noticeably. At 0.16 g/cm^3 (Inspiral 9–12), the cohesionless case results in a spin period of 16.1 h and a neck width of 0.189, while the highest cohesion tested produced a spin of 15.9 h, a neck width of 0.292, and a severely deformed contact binary (see Figures 2.5 and 2.6). Generally, at a given density, increasing cohesion decreases spin period and increases neck width. This relation is much stronger at lower densities. Fixing cohesion and decreasing density

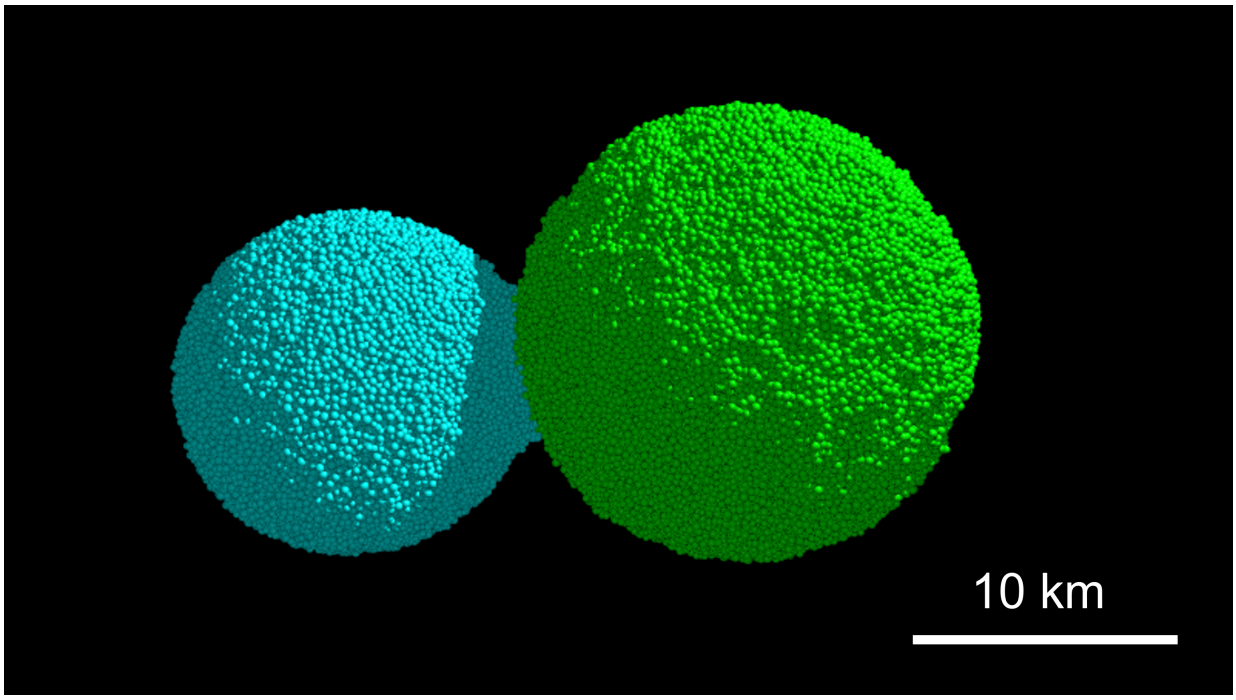


Figure 2.3: Inspiral 1. Cohesionless with gravel friction parameters. Both lobes are left intact and the contact between them forms a well-defined, narrow neck.

similarly decreases spin period and increases neck width.

Finally, Inspiral 14 and Inspiral 15 vary the spin of the progenitor bodies. LL and SL have no spin in Inspiral 14 and in Inspiral 15, LL spins at the synchronous rate, while SL spins at the same rate but in the opposite direction. We find that when neither LL nor SL have spin or when they have opposite spins, the contact binary that they form is somewhat lopsided and has a relatively thick neck, with widths of 0.317 and 0.297, respectively. The non-rotating case produces a spin period of 11.6 h, while the counter-rotating case produces a spin period of 9.6 h.

2.4.3 Progenitor body shape

In runs Oblate 1–5, the impacting components have oblate rather than spherical shapes. All simulations in this group model slowly inspiraling orbits with bulk densities of 0.5 g/cm^3,

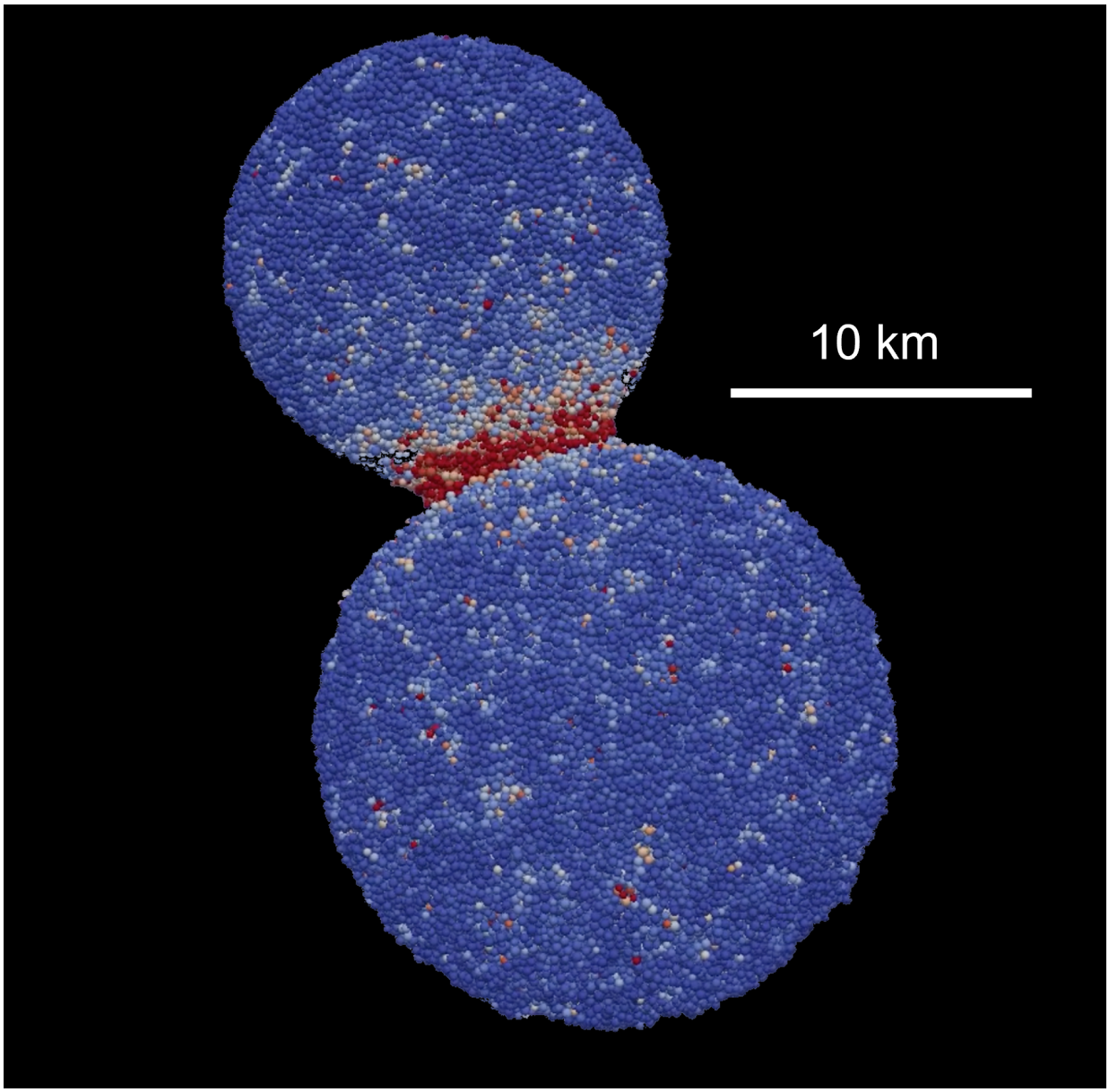


Figure 2.4: Map of maximum total accelerations experienced by particles in Inspiral 1. Darkest reds correspond to 8.8 x 10^{-1} m/s^2 and darkest blues correspond to 3.5 x 10^{-4} m/s^2 on a linear scale. For comparison, the surface gravity of a sphere with volume equal to that of Arrokoth is 1.7×10^{-3} m/s^2.

and synchronous rotations—characteristics we judged most likely to produce something similar to Arrokoth. Oblate 1–5 all produce relatively thin necks. Oblate 5 is the thickest at 0.205 and Oblate 2 and Oblate 3 are the thinnest at 0.169, pointing to no clear relation between body shape and neck width in these cases. Oblate 1 yields the shortest final period at 9.7 h, and Oblate 5 the

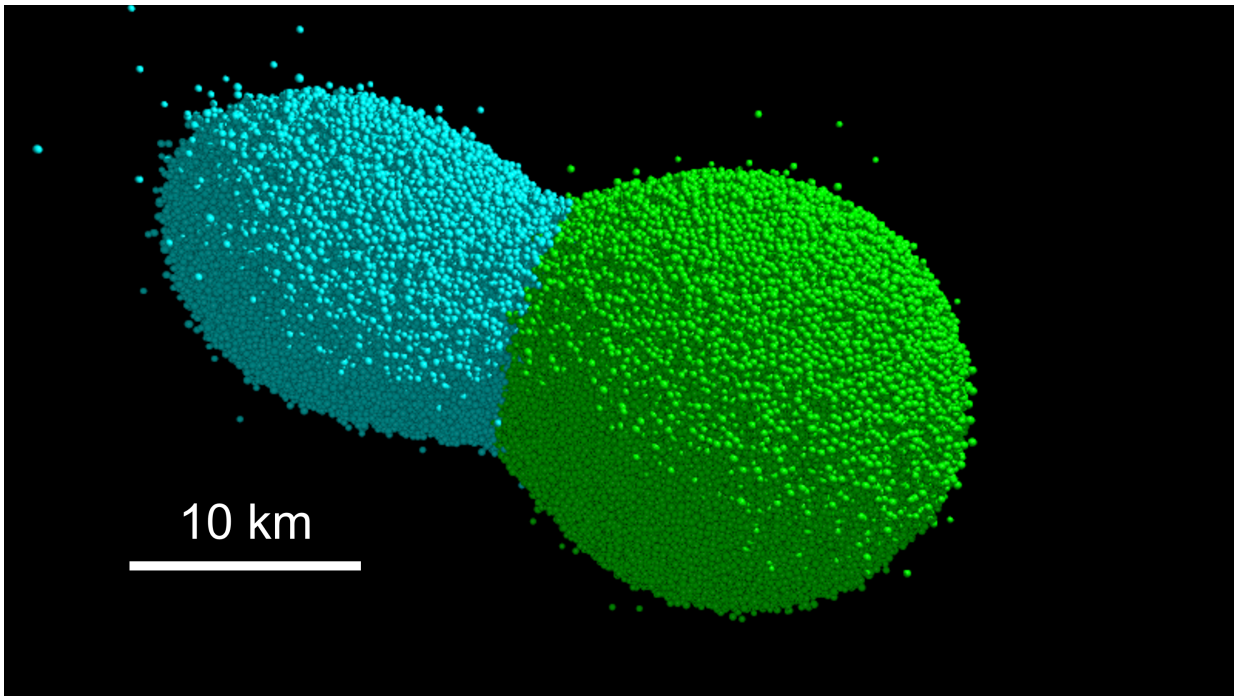


Figure 2.5:]
Inspiral 12. Interparticle cohesion of 27500 Pa and gravel friction parameters. Suprisingly, low-density lobes (here 0.16 g/cm^3) with high interparticle cohesion result in more deformation, as the avalanche toward the neck tends to pull more material with it. The low-density bodies don't have enough self-gravity to hold their shapes.

longest at 11.7 h. Final spin generally decreases with increasing oblateness, though this is more directly a function of total progenitor mass, as described in Section 2.5. All oblate cases tested produce well-formed contact binaries, with clearly intact LL and SL lobes.

2.4.4 Impact speed and angle

We test the effects of varying the speed and impact angle of direct collisions in Impact 1–8. Impact 1, at 5 m/s and 45°, is slow and direct enough to produce a recognizably bilobate object, though it appears quite lopsided and its neck (width 0.358) is considerably thicker than that of the real Arrokoth (see Figure 2.7). None of the remaining cases in this group produce a contact binary. Impacts at $\geqslant 75^\circ$ lead to glancing contacts after which the progenitors separate

mostly unscathed. Lower-angle impacts lead to moderate or even catastrophic disruption of the progenitor bodies (see Figure 2.8).

2.5 Discussion

Our aim in this study is to place constraints on the merger circumstances that led to the formation of the Kuiper belt contact binary Arrokoth. We want to determine the combination of parameters and impact circumstances that best reproduces the observed characteristics of Arrokoth, with intact lobes joined by a narrow neck and a spin period of 16 h. These are the criteria by which we judge the outcome of each simulation. However, we consider the spin period to be less important than the neck width and overall shape, as events after the merger may have modified the period. The Arrokoth contact binary may well have formed with a spin period much shorter than it has today and gradually spun down (through energy exchange with surrounding material, or even very slowly via the YORP effect—see McKinnon et al. (2020)). We also note that when we gradually increase the spin period of the contact binary produced by Inspiral 1 from 9 h to 16 h (using a spin-down procedure similar to that described by Zhang et al. (2017), the object retains its shape and neck width. This suggests that the period of the object immediately after the collision is somewhat incidental in judging whether a given merger scenario could have created Arrokoth. In Figure 2.9 we plot the final neck widths and spin periods of all simulations that produce a contact binary, giving a graphical representation of how successful each simulation is in replicating the present-day Arrokoth. We use 0.205 as an upper limit for a plausible target neck width (shown by the vertical dashed line in the figure).

2.5.1 Simulation outcomes

Among the slow inspiral group, Inspiral 1, 5, 6, and 9 come closest to matching the observed neck width, while Inspiral 9, 10, 11, 12 best match Arrokoth's spin. The observed decrease in final spin with bulk density is a function of the total mass of the bodies. Because volume is kept constant across runs in this group, the lower density progenitors are less massive, and the speeds needed for them to maintain a synchronous orbit are lower. These lower orbital speeds translate to lower rotation speeds after the merger. As discussed above however, final spin period is a less important diagnostic than final shape, because the spin could change after formation (without necessarily changing the body shape).

We also see in the case of Inspiral 1–12 that high interparticle cohesion actually produces a thicker neck and a more distorted final shape. In the model used in this study, cohesion is only applied to particles in physical contact, and when it is active it introduces a force that may enhance or oppose gravity, depending on where the particles lie in the effective potential of the system. It appears that, due to the unusual shape of Arrokoth, high cohesion increases the tendency of surface material to flow toward the neck. When cohesive forces dominate over gravity, particles farther from the neck can be accelerated more violently, and those that detach may be moving quickly enough to temporarily leave the surface. This effect is more pronounced at lower bulk densities. In Inspiral 8 and, to an even greater extent, Inspiral 12 it causes disruption across the entire surface of the object after the merger (see Figures 2.5 and 2.6). We emphasize here that while our cohesion model produces this outcome, the physical cohesive forces at play on Arrokoth may not. Since the true frictional and cohesive properties of Arrokoth's constituent particles are unknown, we can only address the behavior of our model and a future study would

be needed to examine other possible cohesion models.

Inspiral 13 demonstrates the importance of friction in maintaining the shapes of LL and SL, even when the centrifugal force from rotation aids in keeping the bodies separated. Finally, Inspiral 14 and 15 both test the effect of non-synchronous rotation, and both produce contact binaries that are misshapen with respect to Arrokoth. On the basis of final shape and neck width, we can likely rule out Inspiral 4, 8, 12, 13, 14, and 15.

All runs in the oblate group produce contact binaries with intact lobes and neck widths in a plausible range. Because the bulk density is fixed while volume varies, the total mass is different in each run. As in the case of the inspiral group, this leads to a variation in final spin period due to differences in initial orbital speed. From the point of view of neck width and shape, all of these scenarios are plausible, though Oblate 4 and 5 are of course preferable as they represents collisions between two objects that most closely match the present-day shapes of LL and SL.

The impacts group is the least successful overall in terms of reproducing Arrokoth's spin and neck. Only Impact 1 results in a contact binary, and even this scenario can be ruled out as it does not match Arrokoth's shape. A grazing collision could leave the progenitor objects gravitationally bound and lead to a future collision. However such a collision would likely be at or near the mututal escape speed, which should create a deformed contact binary, unless the orbit first circularized and then gradually shrank, leading to a situation more akin to that described in the inspiral scenarios. Although impacts at 5 m/s are slightly above the estimated mutual escape speed of the progenitor objects, simulated impacts at 3.95 m/s (not included here) have outcomes very similar to those of Impact 1–4 and can be ruled out as plausible formation scenarios for Arrokoth.

2.5.2 Favored scenario

It is clear that a scenario in which Arrokoth formed when two unbound KBOs collided is highly unlikely. Our simulations demonstrate that even with cohesive strength, such a collision would leave the progenitors either unbound in the case of a grazing collision, or deformed or destroyed in the case of a more direct collision. This holds true even with impact speeds that are small with respect to present-day Kuiper belt crossing speeds of $\sim$300 m/s (Greenstreet et al., 2019) Even when a recognizable contact binary is formed in this scenario, in the case of the 45° impact at 5 m/s, the neck that results is much less well-defined than what we see in images of Arrokoth. This leads us to believe that a history in which LL and SL were in a synchronous, decaying orbit is much more likely.

Most of our simulations had final spin periods shorter than that of Arrokoth. Only the 0.16 g/cm^3 inspiral scenarios had periods comparable to the observed 15.92 h, and these seem implausible due to the large deformation in the lobes after the merger. This may be an indication that Arrokoth's true density is greater than 0.16 g/cm^3 and that it initially had a shorter period but continued to lose angular momentum after the merger. The simulations with a bulk density of 0.5 and 0.25 g/cm^3 accurately reproduce the shape of Arrokoth, but have periods of ~ 9 h and ~ 12.5 h respectively—shorter than what is observed.

We can consider our results in the context of the KBO formation mechanisms discussed in Section 2.2. In the case of hierarchical coagulation (HC), LL and SL would have had to form separately via accretion in different parts of the Kuiper belt before finally coming together and becoming "stuck" in their current configuration. If they were brought together above the escape speed, the collision would have resulted in disruption of one or both of the lobes, a grazing

contact after which the bodies remained unbound, or a grazing contact that resulted in a highly eccentric binary. In the latter case, subsequent collisions between the bodies would probably occur at speeds high enough to damage one or both of the progenitors. Even if a contact binary formed in this way, it would likely not look much like Arrokoth.

To plausibly produce an Arrokoth-like object via HC, the progenitors would need to be brought together after forming separately at speeds $\lesssim 3$ m/s. This could be accomplished if LL and SL formed a temporary binary system before merging. Collisional binary formation mechanisms like that proposed by Weidenschilling (2002) would not be suitable here, since the speeds involved would be well above the thresholds for destruction of LL and SL that we have described. The exchange reaction scenario described by Funato et al. (2004) exclusively produces binaries with $e \gtrsim 0.8$, which leads to the same problem that the grazing collision case has. Goldreich et al. (2002) put forward two different collisionless capture pathways for forming binaries in the Kuiper belt. In the first (L^2s), two bodies become an unstable binary after entering each other's Hill spheres. A background of smaller particles then shrinks the binary and stabilizes it via dynamical friction. In the second (L^3), the binary is stabilized by an interaction with a third large body. Once a binary is formed, it must then collapse to the point that the two bodies come into contact. In the Goldreich models, further dynamical friction from surrounding smaller bodies causes the binaries to shrink. For the $\sim$100 km bodies with density 1 gm/cm^3 considered in the Goldreich paper, the authors estimate that the timescale for a large object to become bound in an equal-mass binary is 3×10^5 yr, and that the timescale for the binary to inspiral until the bodies come into contact is 10^6 yr (see Eqs. 13 and 14 in that work). In our case (roughly 10 km bodies with density 0.5 g/cm^3), the same equations give a binding timescale of $\sim$750 yr and merger timescale of $\sim 5 \times 10^4$ yr. Depending on the lifetime of the Sun's protoplanetary

disk, this could explain why larger, 100 km-sized binary systems survived in the Kuiper belt, while Arrokoth's progenitor bodies collapsed to a contact binary. On the other hand, a more recent work (Nesvorný et al., 2019) makes a strong case that the gravitational instability (GI) model produces binary inclinations that match observations much better than both L^2s and L^3 do. Furthermore, the HC model cannot explain the observed color correlation between LL and SL.

The GI pathway for forming KBOs is well-supported by our work. Gravitationally bound concentrations of solid particles form in the midplane of the protosolar disk or as a result of the streaming instability. As they collapse, they could ultimately produce a nearly equal-mass, tightly spaced binary. This binary could then inspiral as it is drained of angular momentum. McKinnon et al. (2020) consider a number of possible mechanisms, but point to gas drag as the most convincing. They estimate that due to the pressure gradient in the nebular gas the orbital speed of the gas at Arrokoth's distance from the Sun would be about 1% of the Keplerian speed, so that a binary system feels a strong "headwind." This gives rise to torques on both bodies, which could cause the binary to collapse completely over as little as a few million years—within the lifetime of the gas nebula. Gas drag would likely also have a lesser effect on larger bodies, providing a potential explanation for why Arrokoth became a contact binary, but larger bound bodies remained separated. At least for bodies like Arrokoth, GI seems to be the more likely formation pathway. This could also explain why Arrokoth's current spin period is slower than that of the simulated Arrokoth in Inspiral 1–7—it could have continued to lose angular momentum after the merger. And as described in Section 2.4, a contact binary with the properties of Inspiral 1 can certainly maintain its shape even with the diminished centrifugal support provided by a 16 h spin period.

2.6 Conclusions

We used `pkdgrav`, an N-body tree code with soft-sphere particle interactions, to model the formation of the contact binary (486953) Arrokoth. We focused on modeling the merger itself, and not the origin of the objects that now make up Arrokoth. While images from the New Horizons spacecraft show that Arrokoth is composed of two distinct and apparently primordial lobes connected by a narrow neck, we showed that most direct impacts between the modeled progenitors lead to destruction or significant disruption of the lobes. A more likely formation scenario would involve a Kuiper belt binary that gradually lost angular momentum, spiraled toward a gentle contact to form a contact binary, and continued to spin down, leaving Arrokoth with the $\sim$16 h period that we observe today. While KBB formation mechanisms have received a good deal of attention, the literature on the dynamics of binaries in the outer solar system merging to form contact binaries is more sparse. Nesvorný and Vokrouhlický (2019) find that for cold classical objects of $\sim$100 km, the binary fraction could be as low as 10%. Thus, if Arrokoth is typical for objects of its size, the fraction of binaries that collapsed to contact must be much higher for smaller objects. This may lend further support to the idea that nebular gas was at least partially responsible for collapsing the Arrokoth binary. A clear next step would be to model an earlier period in Arrokoth's evolution. With an implementation of dynamical friction or gas drag in an N-body code, one could model the inspiral process to determine what initial conditions lead to impacts that could plausibly form an object like Arrokoth, which would give us further insight into planetesimal evolution and planet formation in the outer solar nebula.

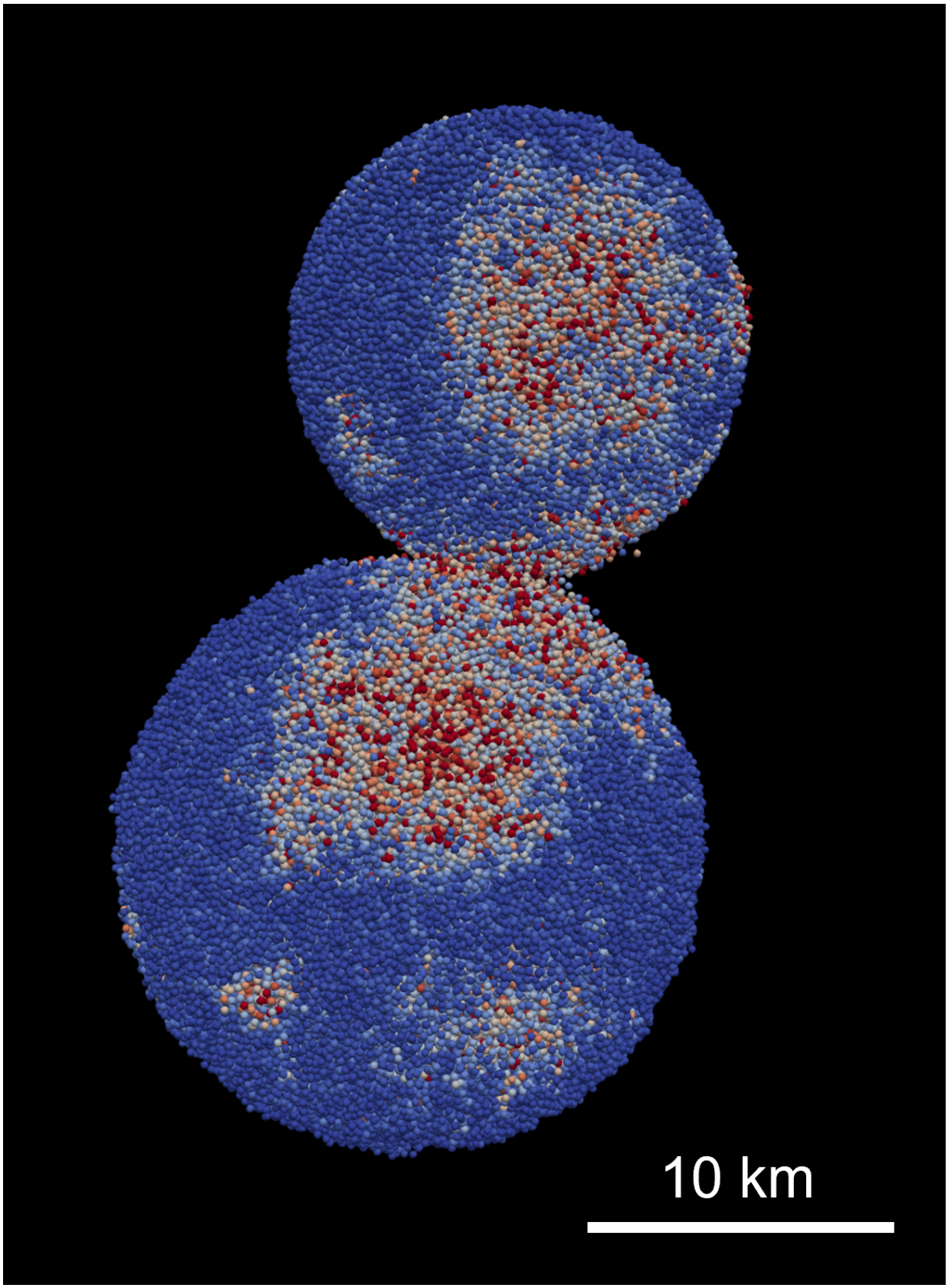


Figure 2.6: Inspiral 12. Map of maximum contact accelerations. Darkest reds correspond to 2.0 m/s^2 and darkest blues correspond to 0.0 m/s^2 on a linear scale. Stresses are not localized to the neck as they are in Inspiral 1.

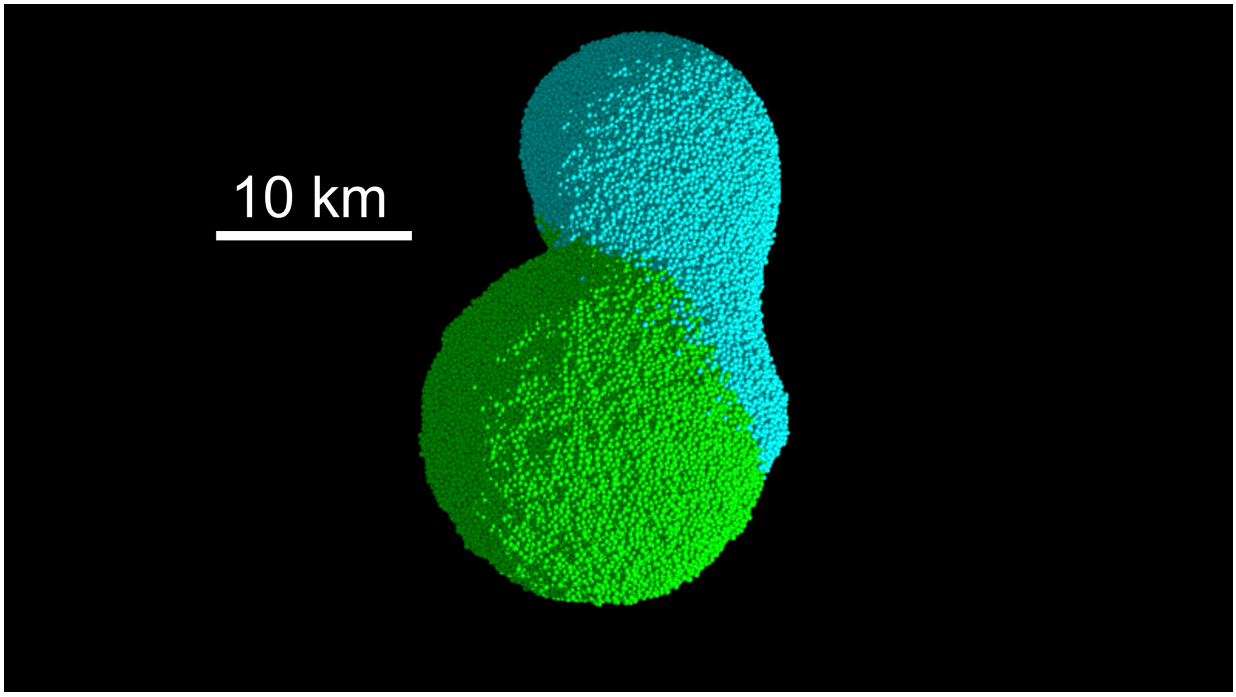


Figure 2.7: Impact 1. 45° impact angle, 5 m/s. The impact creates a contact binary, but with an asymmetric neck and a lopsided SL lobe. The green and blue particles come from LL and SL, respectively.

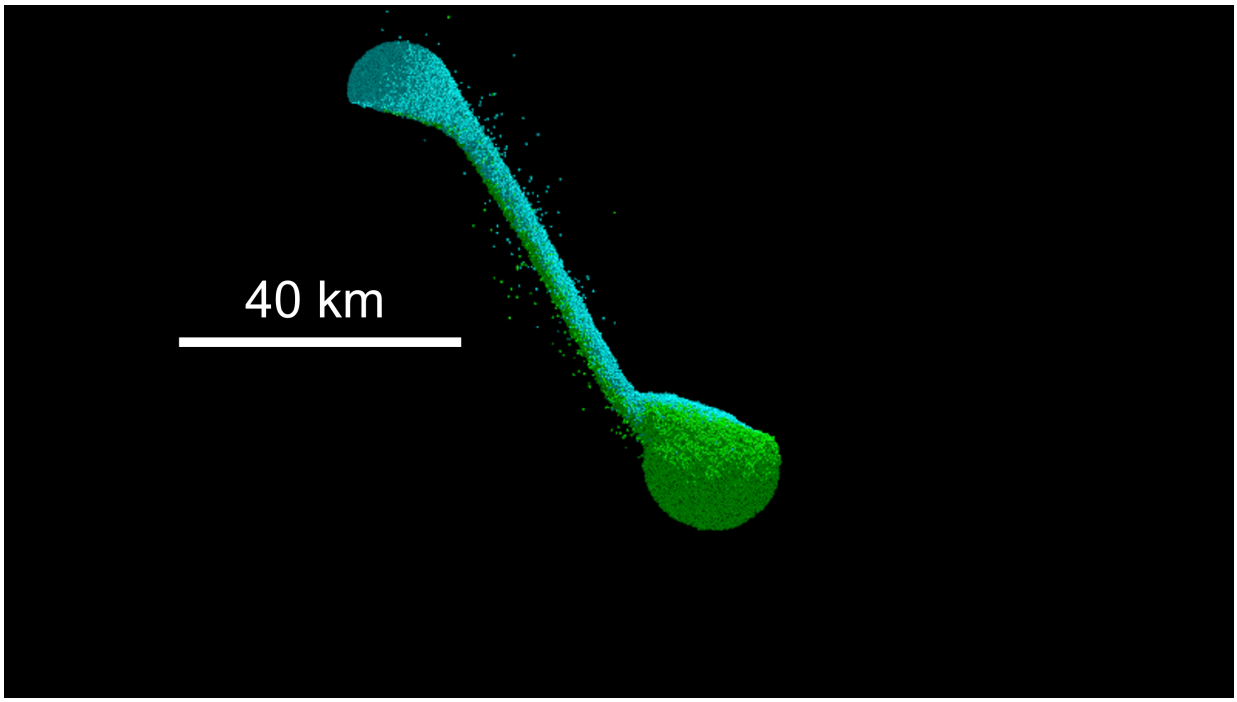


Figure 2.8: Impact 5. 45° impact angle, 10 m/s. The impact severely disrupts both bodies, leaving a long tongue of material stretched between them. As the simulation progresses, this connection breaks as SL moves farther from LL.

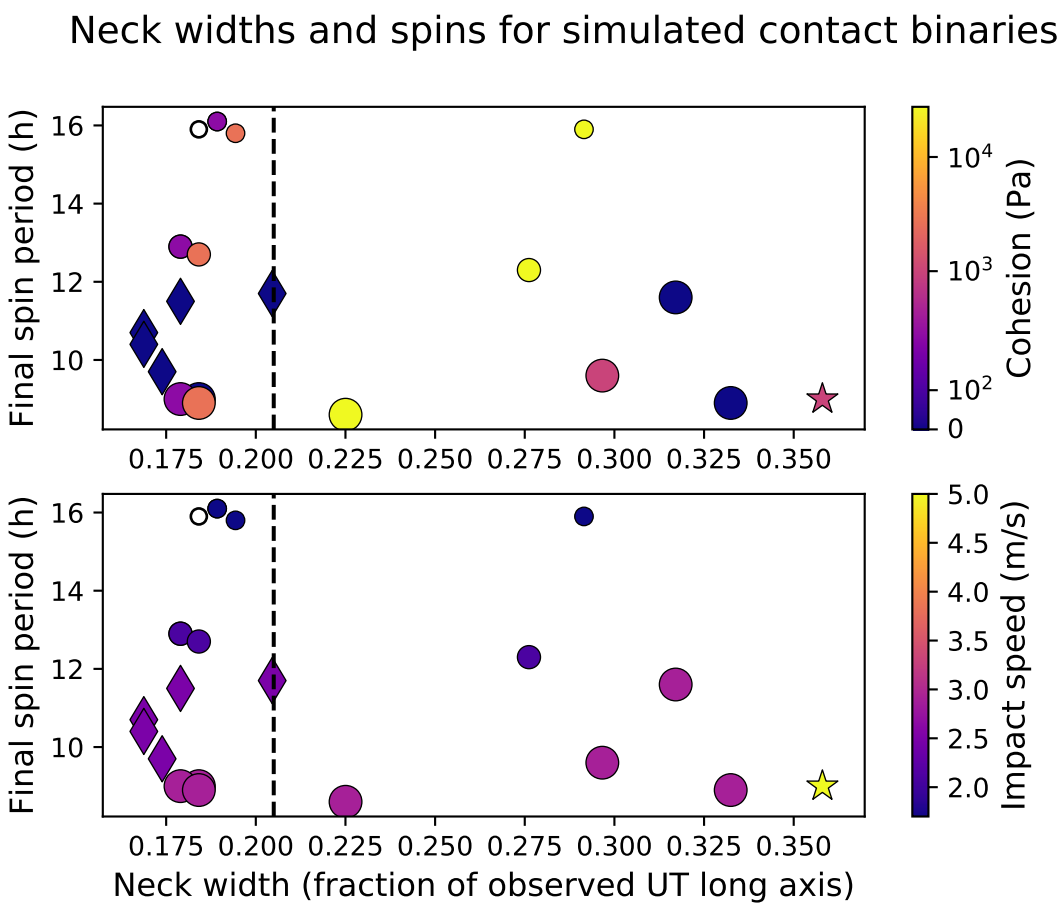


Figure 2.9: Neck widths versus final spin periods for those simulations that produced a contact binary shape. Filled circles represent spherical inspiral runs, diamonds represent oblate inspiral runs, and stars represent direct impacts. The single open circle represents the observed values for Arrokoth. The vertical dashed line at 0.205 is taken as the likely division between plausible and implausible scenarios. Simulations lying to the right of this line produce neck widths too large to be a good match for Arrokoth. We derive this cutoff by considering neck widths within 10% of Arrokoth's to be good matches. No runs produced neck widths that fell below the lower end of this range. Marker size is proportional to bulk density. Colors in panel 1 correspond to cohesion (logarithmic scale), while colors in panel 2 correspond to impact speed (linear scale).

Chapter 3: Modeling the effects of particle shape on rubble pile dynamics

3.1 Preface

This chapter describes the addition of non-spherical particles, or "bonded aggregates" with soft-sphere contact physics to `pkdgrav`, some important improvements in computational efficiency when integrating aggregates, and three trial applications of our new method. Bonded aggregates existed in `pkdgrav` well before we made these modifications (Richardson et al., 2009)—the idea to use a glued-sphere method to model particle shape in `pkdgrav` was not my own. However, that implementation was developed with the older hard-sphere model in mind. Before these modifications, using aggregates with the soft-sphere model would result in unphysical behavior in many circumstances and the underlying implementation was not compatible with SSDEM. In addition to the substantial changes detailed here, I spent quite a bit of time attempting to add rolling and twisting friction to the aggregate model as well. Unfortunately, there were certain issues with this that I was not able to solve. We ultimately decided that the situations in which aggregates would benefit from these additional frictional forces were limited enough that they were not worth the additional time that would be required to implement them properly. That work is not reflected in this chapter.

The sections on computational efficiency enhancements and the Brazil nut effect represent work carried out Joseph DeMartini and those sections were written jointly by the two of us. The

remainder of the work is my own, and I wrote the remainder of the manuscript myself. At the time of this writing, the work in this chapter has not appeared in publication, but has been accepted for publication in *The Planetary Science Journal* as "An efficient numerical approach to modeling the effects of particle shape on rubble-pile dynamics" (Marohnic et al., 2023). The body of that manuscript appears here with little modification. The abstract and acknowledgements included with the accepted manuscript have been removed and some formatting has been adjusted. The manuscript's appendix has been reproduced as Appendix A to this dissertation.

3.2 Introduction

Most small solar system objects are believed to be loose, unconsolidated masses of material, rather than monolithic bodies (Walsh, 2018). These "rubble piles" are largely held together by gravity and may be of various composition depending on their location in the solar system and specific history. Substantial work has been devoted to studying rubble-pile bodies, including via numerical techniques. Notable among these numerical methods are smoothed-particle hydrodynamics (SPH) (Benz and Asphaug, 1995; Jutzi, 2015) and discrete-element method (DEM) codes (Richardson et al., 1998; Sánchez and Scheeres, 2012; Schwartz et al., 2012; Sánchez and Scheeres, 2014; Zhang and Michel, 2020). In this study, we focus exclusively on DEM, which treats rubble-pile constituents as separate elements explicitly rather than as a continuum via approximate constitutive relations. In particular, we will consider the effects of the *shapes* of the granular elements and the effects of these shapes on the behavior of the body as a whole.

Solem (1994) and Asphaug and Benz (1996) used a frictionless, hard-sphere particle model to estimate the size and density of comet Shoemaker-Levy 9 after its tidally induced disaggrega-

tion. Movshovitz et al. (2012) approached the same problem using polyhedral particles, determining that disruption is more difficult than in the case of a similar sphere-based model. However, the study was restricted to relatively low-resolution simulations (about 4000 grains) due to computational limitations. Walsh and Richardson (2006, 2008) used a hard-sphere numerical model to study the formation of binary asteroid systems via tidal disruption of rubble-pile asteroids and found that this mechanism alone was not sufficient to explain the observed binary fraction in the near-Earth asteroid population. The authors later used the same model to show that YORP spin-up of rubble piles can create fast-rotating asteroids with close secondaries (Walsh et al., 2008). More recent work has applied the soft-sphere discrete element model (SSDEM; Cundall and Strack, 1979)—which allows for persistent particle contacts and friction—to the problem of disruption and shape change of rubble piles via spin-up and tidal forces, though these studies are still largely confined to spherical particles (Sánchez and Scheeres, 2012; Walsh et al., 2012; Yu et al., 2014; Zhang et al., 2018). Nevertheless, constituent shape is believed to be an important factor in the behavior of granular media generally and should be considered in the context of small rubble-pile bodies as well.

3.2.1 Importance of particle shape in granular media

Granular media made up of non-spherical components exhibit higher shear strength than those composed of spheres (see Wegner et al. (2014) and references therein). In a compacted state, non-spherical particles in a granular medium will interlock with each other. When subjected to shear, these interlocked constituents cannot easily slide past one another without the entire medium first expanding, or "dilating." The overall tendency of a granular body to dilate under

shear force is likewise known as "dilatancy." Though they are not monolithic objects, rubble-pile asteroids are subject to confining pressure due to their own self-gravity. As a result, when they experience shearing from a tidal encounter or spin-up that exceeds their shear strength, the particles that make up the body will tend to move relative to each other as in any other granular medium under compression.

While the spherical particles in traditional DEM models can easily slide or roll past one another, this is more difficult for particles with other shapes due to the increased dilatancy of the medium. Thus, we expect that the physical rubble piles in the solar system that are made up of irregular constituents will have a higher effective shear strength and will be more difficult to disrupt than in idealized DEM simulations with spherical particles. A related effect has been documented in the case of terrestrial granular flows in both experiments (Sarkar et al., 2019) and DEM simulations (Cleary, 2008; Mead and Cleary, 2015). When a granular solar system body is disrupted, the resulting fragments and reaccumulated successor bodies may have different properties from those in the spherical-grain case. They are also likely to be able to maintain shapes further from their fluid-equilibrium shapes. These differences have implications for the understanding of small-body internal structure. In reality, small solar system bodies are not made up of spheres, whether on the scale of small grains or large boulders, as imagery from the recent *Hayabusa2* and *OSIRIS-REx* missions, for example, have made clear (Michikami et al., 2019; Walsh et al., 2022b). Since many of the small bodies in our solar system are subjected to stresses from spin-up and tidal encounters at some point during their evolution, it is important to understand and quantify the effects of irregular particle shapes. For example, what role does particle shape play in setting the critical spin limit for rubble piles, and to what extent? How does particle shape affect binary formation under tidal and rotational stresses? Given the resolution

and efficiency we can now achieve with non-spherical particles in the code we describe here, these are questions we can begin to answer.

Most DEM codes used in the context of small solar system bodies have relied on spherical particles for the simplicity and computational efficiency they afford. That said, some N-body DEM codes have included non-spherical particles of various construction, with most examples either using poly-ellipsoids (Roig et al., 2003), or polyhedra (Ferrari and Tanga, 2020; Sánchez et al., 2021). Due to the complexity of applying contact physics and interparticle gravity to irregular shapes, these efforts have historically had to compromise on the fidelity of the physics model, the number of particles included in simulations, or both. Ferrari and Tanga (2020) include polyhedral particles with soft-sphere contacts in their model, but don't allow for gravitational torques as they treat these polyhedra as point particles when calculating gravitational interactions. And while this method uses a tree to reduce the cost of gravity calculations, it is confined to a single GPU, ultimately leading to memory limitations on overall resolution. Sánchez et al. (2021) also allow for non-spherical, self-gravitating particles, but use a non-smooth contact dynamics method rather than an SSDEM approach and likewise omit gravitational torques. Alternately, there are a number of existing DEM codes that are capable of modeling the interaction of hundreds of thousands or even millions of non-spherical particles with full soft-sphere DEM contacts (Zhao et al., 2006; Knuth et al., 2012; Longmore et al., 2013; Nguyen and Plimpton, 2019). However, none of these include the expensive calculations of interparticle gravitational forces that are critical for studying the dynamics of rubble-pile bodies. This work presents a scheme for assembling spheres into rigid, non-spherical grains. Although this implementation leverages our code's existing ability to quickly compute gravity and contact interactions between large numbers of particles, the techniques described here are broadly useful to the study of small, rubble-pile

bodies and could be applied in other N-body DEM codes. We use the "glued-sphere" method popular in granular dynamics (Nolan and Kavanagh, 1995; Song et al., 2006), attaching the existing spherical `pkdgrav` particles together in arbitrary shapes. This approach seamlessly joins the existing soft-sphere physics model and the highly efficient gravity tree from `pkdgrav`, allowing for N-body simulations with non-spherical particles, gravitational and collisional torques between grains, and resolutions up to hundreds of thousands or more particles. To the best of our knowledge, this represents the fastest implementation that combines all of these features. Section 3.3 describes the basic computational methods while Section 3.4 details how we have improved the efficiency of our particular implementation. We present some applications of the new code in Section 3.5 as proofs of concept and to lay the groundwork for future projects. Section 3.6 provides a brief summary of this work.

3.3 Computational methods

We begin with a brief review of our numerical approach to modeling particle contacts with SSDEM in our code `pkdgrav` prior to implementing non-spherical shapes. This is followed by details on the modifications needed for modeling non-spherical constituents.

A note on our terminology: instead of constructing polyhedral particles with flat faces and edges, the non-spherical particles or grains that we wish to model are made up of spheres locked together. The spheres themselves are the object traditionally considered "particles" in `pkdgrav` and are analogous to the particles used in most other DEM codes. In describing our implementation, we must frequently refer to the individual spherical particles that already exist in `pkdgrav`, as well as the rigid collections of these spheres that form the non-spherical grains.

To avoid confusion, we will refer to the non-spherical assemblies as "aggregates" or "bonded aggregates." The word "particle" left unmodified should be assumed hereafter to refer to the spherical elements that make up the aggregates. In addition, "constituents," "components," or "grains" will refer to either spherical particles or bonded aggregates in the context of the pieces that compose a rubble-pile body, regardless of their shape or size. Since most force calculations and position and velocity evolutions in our implementation are carried out at the particle level, this distinction is important.

3.3.1 Existing soft-sphere implementation for spherical particles

Our approach builds on the existing numerical gravity code `pkdgrav` (Richardson et al., 2000; Stadel, 2001). `pkdgrav` uses a hierarchical tree algorithm that reduces the cost of locating neighboring particles to an $\mathcal{O}(N \log N)$ operation, where N is the number of particles. Interparticle gravity is also computed using a tree to speed up the calculations, by replacing $\mathcal{O}(N^2)$ sums over individual particles with multipole expansions of the gravitational potential contributed by small or distant cells. The multipole expansions are taken to hexadecapole order as a middle ground between speed and accuracy. Both neighbor searching and gravity calculation are also parallelized, allowing `pkdgrav` to distribute the work across an arbitrary set of processors for further speed optimization.

`pkdgrav` also includes a soft-sphere discrete-element-method (SSDEM) scheme for treating particle interactions, which is described in much greater detail in previous works (Schwartz et al., 2012; Zhang et al., 2018). Unlike hard-sphere methods, SSDEM resolves collisions temporally, allowing particles to interpenetrate slightly as a proxy for surface deformation. `pkdgrav`'s

SSDEM implementation uses a spring-dashpot model, in which overlaps between neighboring particles are detected and normal and tangential restoring forces are then modeled as damped springs following Hooke's law. Spring and damping constants are user-adjustable, but in practice must be set carefully to capture the properties of the particular material being modeled and to maintain physically realistic behavior. To ensure that particle overlaps remain small, spring constant settings must account the masses, sizes, and speeds of all constituents. The spring forces capture the effects of deformation at particle contacts, while the damping forces capture the effects of kinetic friction. `pkdgrav` uses this approach to track forces and torques from twisting, rolling, and sliding friction. Particle overlaps are tracked for as long as particles remain in contact, and reaction forces depend not only on the degree of overlap at the current time step but also on the contact history. SSDEM is a substantially more realistic model of the granular physics we are interested in than hard-sphere DEM, able to capture multiple simultaneous and persistent contacts self-consistently. These are exactly the sort of interactions at work in small rubble-pile bodies; using an SSDEM model thus allows us to more accurately simulate the interaction of systems made up of hundreds of thousands of particles in contact.

`pkdgrav` uses a fixed-step, second-order leapfrog method to integrate gravity and contact interactions. The leapfrog integrator updates center of mass positions and velocities for both individual particles and aggregates, while the orientations and spins of bonded aggregates (described in greater detail below) are instead evolved with an adaptive Runge-Kutta method. While leapfrog integration is designed for equations of motion with no explicit velocity dependence, the friction model introduces velocity-dependent damping terms. To account for this, we also calculate "predicted" velocities and spin vectors for each particle. Using predicted velocities technically reduces the scheme to first order, but because the velocity-dependent terms are all

related to damping and because the time steps are very small in order to resolve the contacts, this is not an issue in practice (Schwartz et al., 2012).

3.3.2 Bonded aggregates

To address the question of grain shape in rubble-pile bodies, we need the ability to simulate non-spherical constituents. Our approach is to construct compound pseudo-particles, which we call "bonded aggregates," that consist of multiple spherical particles "glued" together with unbreakable bonds. This approach was used in the hard-sphere model of `pkdgrav` (Richardson et al., 2009) but requires updating for the soft-sphere model. We arrange arbitrary numbers of spherical particles in any desired shape and then fix their relative positions so that they behave as a unit, creating rigid, non-spherical aggregates. One of the primary advantages of constructing non-spherical constituents this way is that computing forces and torques is relatively straightforward. Since we already track the positions and velocities of the constituent spheres, we can easily calculate the total mass and COM positions and velocities of these bonded aggregates. They can then be treated as discrete objects that receive a unique identifying numerical tag for tracking. Much of the existing machinery for calculating contact and gravitational forces can also be reused in the context of bonded aggregates. In addition, we retain the static friction model used in the sphere-based version of `pkdgrav`, which allows us to capture the effects of grain roughness in addition to grain shape. In this section, we describe in more detail how this method is implemented. To model a full granular assembly with many non-spherical constituents using `pkdgrav` efficiently, further optimization is required to complement these modifications to our physics scheme (Section 3.4).

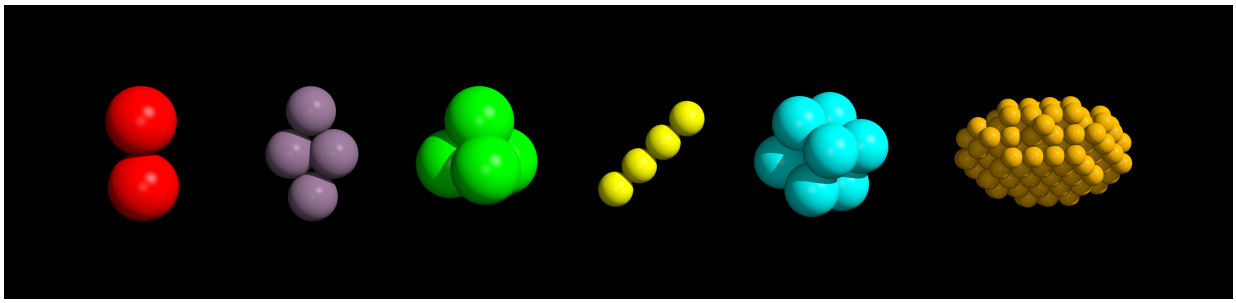


Figure 3.1: Non-spherical constituents or "bonded aggregates" as implemented in `pkdgrav`. From left to right, a two-particle "dumbbell" shape, a 4-particle planar diamond, a 4-particle tetrahedron, a 4-particle rod, an 8-particle cube, and a 155-particle irregular aggregate.

3.3.2.1 Preliminaries

We use the symbol A as a subscript to refer to a bonded aggregate and the same symbol in the context of a summation represents the set of indices of all of the constituent particles of aggregate A. We define the total aggregate mass M, which is simply the sum of the masses of the constituent particles,

$$M = \sum_{i \in A} m_i, \tag{3.1}$$

where m_i is the mass of a particle in aggregate A. We also define the COM position $\mathbf{r}_A$ of an aggregate, which can be considered the "position" of the aggregate,

$$\mathbf{r}_A = \frac{1}{M} \sum_{i \in A} m_i \mathbf{r}_i, \tag{3.2}$$

where $\mathbf{r}_i$ is the COM position of a particle in the aggregate. Other aggregate-specific quantities will be introduced as needed.

While the COM positions and velocities of aggregates follow our previously established equations of motion, the introduction of bonded aggregates means we must account for rigid-

body rotations as well. Aggregate rotations should obey the Euler rigid body equations, which we solve with a time-adaptive, fifth-order Runge-Kutta integrator. For a more detailed account of the integration scheme used for bonded aggregates, readers may consult Richardson et al. (2009), which describes an earlier hard-sphere implementation of bonded aggregates in `pkdgrav` using the same integration method.

3.3.2.2 Predicted velocities

We must calculate predicted velocities for bonded aggregates just as we did for spherical particles (see Sec. 3.3.1). Consider an individual particle that is a member of a bonded aggregate. Since forces are tabulated on a particle-by-particle basis, the predicted velocity is still important in the context of bonded aggregates. In this case, to predict the movement of the particle we must take into account the rotation of the aggregate itself in addition to the current COM velocity of the particle. The predicted velocity estimate for a particle in an aggregate is now given by the following equation:

$$\dot{\mathbf{r}}^{\mathrm{pred}}_{i,n+1} = \left[\dot{\mathbf{r}}_{A,n+\frac{1}{2}} + \frac{h}{2}\ddot{\mathbf{r}}_{A,n}\right] + \left[\boldsymbol{\omega}_{A,n+1} \times (\mathbf{r}_{i,n+1} - \mathbf{r}_{A,n+1})\right] \tag{3.3}$$

Calculating predicted velocities for particles belonging to aggregates necessarily works differently than in the case of stand-alone particles, since these particles are now part of rigid bodies. The first term in square brackets is the predicted COM velocity of the aggregate that particle i belongs to, which is in turn imparted to the particle itself. The second term in square brackets represents the contribution of aggregate rotation to predicted particle velocity, determined by the spin rate of the aggregate and the distance of the particle from the center of the aggregate. We note

that *aggregate* spins need not be extrapolated to whole-number time steps in the same way that they are for individual particles, since aggregate spins and orientations are integrated separately from the primary leapfrog scheme with a Runge-Kutta method as described previously.

3.3.2.3 Gravity

Gravitational forces are calculated for each individual sphere in `pkdgrav`. When the tree code is being used, which is typical, these force calculations are approximated. In the case of a self-gravitating assembly of independent spherical particles, we calculate the acceleration $\ddot{\mathbf{r}}_i$ felt by each particle. In the case of an aggregate, we instead calculate the *force* of gravity on each constituent particle in the aggregate $\mathbf{F}_{i,g}$. We define this force as follows.

$$\mathbf{F}_{i,g} = \sum_{j \neq i} \frac{G m_i m_j \left(\mathbf{r}_j - \mathbf{r}_i\right)}{\left|\mathbf{r}_j - \mathbf{r}_i\right|^3} \tag{3.4}$$

To find the net gravitational acceleration of a bonded aggregate $\ddot{\mathbf{r}}_{A,g}$ we only need the vector sum of the gravitational forces on each particle in the aggregate, divided by the total aggregate mass.

$$\ddot{\mathbf{r}}_{A,g} = \frac{1}{M} \sum_{i \in A} \mathbf{F}_{i,g}. \tag{3.5}$$

Individual spheres feel no net torque from gravity while a non-spherical aggregate in general will experience some torque $\mathbf{N}_{A,g}$. The vector sum of the torque contributions from each constituent sphere in an aggregate is the net gravitational torque on the aggregate.

$$\mathbf{N}_{A,g} = \sum_{i \in A} \left[\left(\mathbf{r}_i - \mathbf{r}_A\right) \times \mathbf{F}_{i,g}\right] \tag{3.6}$$

3.3.2.4 Contacts

Particle contacts in general result in both a force and a torque on each particle. While gravitational forces act on particle centers, contact forces and torques act at the *contact point*. In the case of a small aggregate composed of only a few particles, the difference in the effective lever arm can be quite significant. In the event of a contact, each particle feels a normal restoring force that depends on the extent of the overlap. If the particles have non-zero friction, they may also experience a tangential surface force. Contact forces between particles are then applied to the aggregates that they belong to, and the net center-of-mass acceleration on the aggregate due to contact forces $\ddot{\mathbf{r}}_{A,c}$ is obtained according to

$$\ddot{\mathbf{r}}_{A,c} = \frac{1}{M} \sum_{i \in A} m_i \ddot{\mathbf{r}}_{i,c}, \tag{3.7}$$

where $\ddot{\mathbf{r}}_{i,c}$ is the acceleration due to contact forces on constituent particle i. In other words, $m_i \ddot{\mathbf{r}}_{i,c} = \mathbf{F}_{i,N} + \mathbf{F}_{i,T}$, where $\mathbf{F}_{i,N}$ and $\mathbf{F}_{i,T}$ are the normal and tangential contact forces (see Schwartz et al. (2012) for details).

The contact torque contribution to an aggregate from a constituent particle is calculated by considering the normal and tangential contact forces $\mathbf{F}_{i,N}$ and $\mathbf{F}_{i,T}$ felt by that particle. These forces act on a lever arm spanning from the aggregate center of mass to the contact point on the particle, rather than the particle center as in the case of gravity. To be precise, we say that the "contact point" itself lies at the center of the circle formed by the intersection of the surface of the particle and its overlapping neighbor. The distance between the center of the particle and the

contact point is equivalent to the lever arm l_p for a spherical particle, which is given by

$$l_p = \frac{s_p^2 - s_n^2 + |\mathbf{d}|^2}{2\,|\mathbf{d}|}. \tag{3.8}$$

In Eq. 3.8, s_p and s_n are the particle and neighbor radii, respectively, and $\mathbf{d}$ is the distance between their centers. The vector that points from the center of the particle to the contact point is then given by $l_p\hat{\mathbf{n}}$, where $\hat{\mathbf{n}} = \mathbf{d}/|\mathbf{d}|$ is the unit normal vector pointing from the center of the particle toward the center of its neighbor. The vector sum of these torque contributions from each particle in the aggregate gives the net torque due to contact forces on a bonded aggregate in `pkdgrav`:

$$\mathbf{N}_{A,c} = \sum_{i \in A} \left[(l_p\hat{\mathbf{n}} - \mathbf{r}_A) \times m_i \ddot{\mathbf{r}}_{i,c} \right]. \tag{3.9}$$

We note that the careful approach to applying gravity and contact forces described above is necessary for achieving proper physical behavior. Angular momentum conservation in particular degrades if torques are not applied at the contact points, especially in the case of large rubble piles.

3.3.2.5 Rolling and twisting friction

Rolling and twisting friction are currently included in `pkdgrav` for spherical particles only, though a prescription for applying rolling and twisting friction to bonded aggregates is currently being developed and may be added in the future. There are certain circumstances in which this capability would enhance the realism of the code (e.g., a rod-shaped aggregate rolling on a flat surface). However, in the great majority of systems that are of interest to the field of

small bodies, bonded aggregates would be used as constituents of a self-gravitating assembly rather than rolling or spinning freely on flat surfaces. Under these conditions, we expect that rolling and twisting friction will play a negligible role and, in fact, the non-spherical aggregate shapes should capture most of the physics that rolling and twisting friction schemes seek to parameterize in simulations with unbonded spheres.

3.4 Computational efficiency improvements

In this section, we detail the methods of locating and operating on bonded aggregates in `pkdgrav`, the improvements developed for these methods to complement the new physical model from Section 3.3.2, and the profiling analysis that we employ to measure the efficiency of the new methods. References to processes occurring "in serial" indicate that a single processor is handling the operations, while "in parallel" means that the load is split across multiple processors, generally with each processor performing the same operation on a different subset of particles in the simulation. We emphasize that while the computational efficiency improvements detailed in this section are described in the context of `pkdgrav`, these techniques are also applicable to any investigators intending to develop, modify, or improve other similar, parallel codes.

3.4.1 Aggregate Constituent Searching

The original hard-sphere aggregate routines for `pkdgrav` are optimized to handle a few large aggregates made up of many particles (Richardson et al., 2009). The routines keep track of aggregate properties (center-of-mass position/velocity, torques, etc.—see Table A.1) in serial on a single leader processor while some or most of the constituent particle data may be distributed

across multiple follower processors in groupings convenient for the interparticle gravity calculation. Load balancing performed each time step may cause these groupings to change. To carry out an operation on an aggregate, the leader processor requests data for each of the aggregate's constituent spheres, which may be on one or more follower processors. In the original routines, a "brute-force" search finds particles bound in aggregates; it is a serial method that examines the particles on each processor one-by-one, eventually finding all particles contained within the aggregate and returning the information needed to calculate an aggregate property only after reaching the end of the full list of particles. If the number of aggregate particles is much larger than the number of aggregates and similar to the total number of particles in the simulation, each processor will have many particles belonging to a given aggregate, so the brute force search "hits" frequently and works relatively well.

A noticeable inefficiency arises when the number of aggregates and the number of particles are both similar and large, as is the case when trying to simulate a granular system made up of many aggregates, each containing only a few constituent particles, like the aggregates in Fig. 3.1. In this scenario, there are perhaps only two particles belonging to a given aggregate, but thousands of aggregates in the simulation. Every time the properties of a single aggregate require updates, the brute-force search wastes time scanning the whole particle list on each processor despite needing to find only two particles. Since the properties of each aggregate are updated every time step and each aggregate contains only a few particles, the originally tolerable brute-force search approaches an $O(N^2)$ process per step (where N is the number of particles in the simulation); this is prohibitively expensive for large N processes, like the simulations we perform in Section 3.5. The brute-force aggregate particle search is thus the main bottleneck for simulations involving large numbers of aggregate operations in `pkdgrav`, and our aim in increasing the computational

efficiency of these simulations is to improve the search method for identifying particles belonging to a given aggregate.

First, we solve the issue of efficiently locating aggregate particles by reordering all particles before the search begins. On each processor, particles are reordered by their aggregate index numbers—an identifying integer unique to each aggregate that is stored in the data structure of each of its particle members. All particles in a given simulation thus "know" which aggregate they belong to, or whether they are simply unbound free particles. This reordering forces all particles already on a given processor into consecutive order by aggregate index, but allows for the possibility of aggregates that are split across processors. When data from aggregate member particles is required, we query each processor for its range of aggregate indices. If the index of the aggregate that we are operating on falls within the range bracketed by the end members on a processor, we search that processor for the relevant particles; if the desired aggregate index is outside a processor's range, it will not be queried. This prevents processors in parallel simulations from doing unnecessary work searching for particles that they will never find. However, this still leaves the issue of the $O(N^2)$ scaling of the brute-force search in serial simulations, as well as in parallel simulations when a processor does contain desired constituents in its range.

To further optimize the search method, we replace the original brute-force search with a binary-search algorithm (Bentley, 1975) that exploits the new particle order and scales as $O(N \log N)$ to reliably and efficiently find the first member particle on a processor, and add a "cache-line" method with best-case $O(N)$ scaling after the first member particle is found. The binary search is used whenever the particle information is not immediately known, i.e., not in the cache. The cache line stores identifying information of the previously found particle in order to easily step forward and find the next particle that needs to be acted on when the code returns

to the function. This is a linear operation in N, so it scales well, and the cost of reordering by aggregate index at each time step is relatively small.

3.4.2 Profiling Analysis

In order to measure the increase in efficiency gained from the improvements described in Section 3.4.1, we used code profiling to determine the length of time `pkdgrav` spends performing certain operations. In our analysis, we used the GPROF hybrid instrumentation and sampling profiler (Graham et al., 1982), one of the longest-standing profiling tools for compiled code. The output of our analysis contains a list of all the function calls in decreasing order of time spent in each function for a simulation that was run to completion. For each function, the number of calls to that function, the number of seconds spent in that function, and the corresponding percentage of the overall execution time devoted to that function are reported. The default resolution of the profiler is 0.01 s and 0.01%, below which it will not report the actual time or percentage of time spent in a given function. The table in Appendix A lists the `pkdgrav` functions modified to improve efficiency and gives brief descriptions of their purposes. Previously, these functions were the most time-consuming operations in simulations containing large numbers of aggregates relative to the total number of particles. In typical simulations of $N = 10,000$ equal-sized particles composing between 1,250 and 5,000 aggregates, these functions collectively account for $\geqslant 50\%$ of the total simulation time.

3.4.3 Performance Scaling

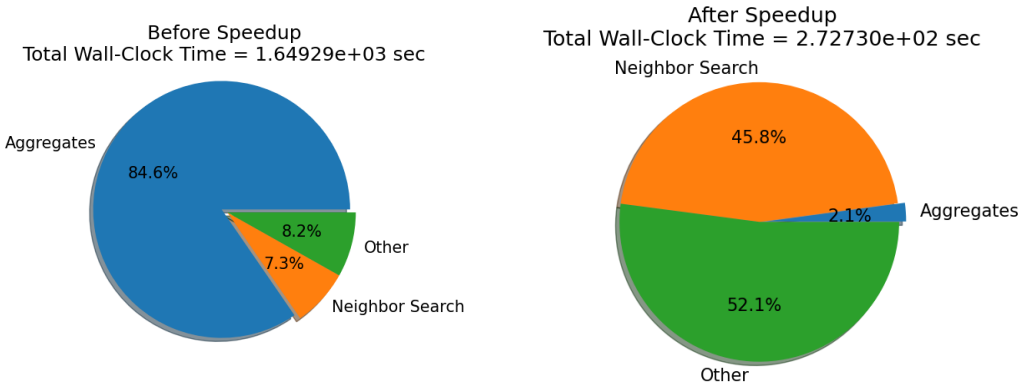
We compare granular dynamics simulations of particles settling for $10,000$ integration steps run on a single CPU core under both uniform gravity and mutual self gravity and for both the original aggregate routines and the new, more efficient routines for each of the aggregate constructions seen in Fig. 3.1. Representative figures (Figs. 3.2, 3.3) compare the profiling metrics for time spent in aggregate routines during the test simulations under the new and old aggregate handling schemes.

The profiling simulations of particles in uniform gravity capture a portion of the process of filling a rectangular, open-topped box with bonded aggregates. We use a box with base side lengths of 10 cm and a height of 60 cm. The aggregates are composed of equal-sized, 0.25 cm-radius spheres and we apply lunar-strength gravity (1.62 m s^{-2}). For these box-filling simulations, the total number of aggregate member spheres is held constant at N = 6,400 particles, composing between 800 and 3,200 aggregates depending on the aggregate geometry. We also model an analogous scenario with no aggregates but only free spherical particles, as well as a scenario where we mix the symmetric aggregate shapes from Fig. 3.1 that contain 8 or fewer member particles (but excluding free spheres). In this "mixed aggregates" scenario, we again keep fixed the particle radii and number of spheres at N = 6,400, consistent with the other box-filling models. Fig. 3.2 compares the profiling results of the mixed aggregates box-filling tests before and after the efficiency improvements are applied, as a representative for the speedup we see with the updated search and sort routines. The total wall-clock simulation time for 10^4 time steps in the simulation represented in Fig. 3.2 is reduced from 1.6×10^3 s to 2.7×10^2 s—more than five times faster in this case (see Fig. 3.4 for the direct comparison).

The trial scenarios with interparticle gravity model the gravitational collapse of a cloud of aggregates, which was constructed to ensure that both interparticle contact forces and self-gravity would be at work during the period that we profiled. The collapse simulations use aggregates composed of equal-sized 50 m-radius spheres. For each of the collapse simulations, the number of spherical member particles is held constant at $N = 2 \times 10^4$ particles, ranging from 2.5×10^3 to 1×10^4 aggregates, again depending on the number of particles in the aggregate shape. Here, as well, we have analogous models with free spheres and mixed aggregates for comparison. Fig. 3.3 shows the profiling results for the mixed aggregate collapse as a representative example of the speedup achieved with the updated search and sort methods in a simulation without a uniform gravity field. In this representative simulation, we see a reduction in total wall-clock simulation time over 10^4 simulation steps from 1.8×10^4 s to 6.4×10^3 s: almost a factor of 3 decrease in total runtime, as seen in Fig. 3.4.

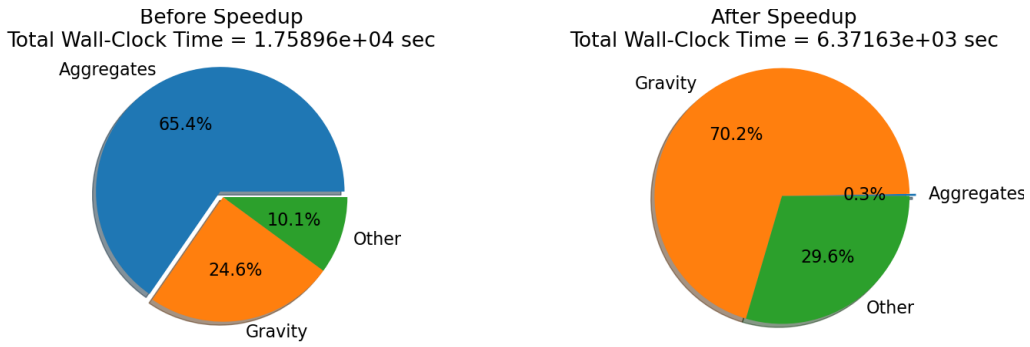
From our profiling, we find that the combined aggregate routines are by far the most expensive operations, easily accounting for over half of the time spent in simulations without our newly updated sorting and searching methods. In analogous tests with the improvements implemented, we find that the total simulation time greatly decreases (see Fig. 3.4). Operations on aggregates are now in the noise, representing only a few percent or less of the total simulation time.

To more clearly illustrate the value of the efficiency improvements, we divide the total pre-modification simulation time spent on each test into three segments: the fractions of time respectively devoted to the most costly operation, the second-most costly operation, and all other operations combined ("Other"). We then compare the relative computational costs of these segments before and after the efficiency modifications. We include representative examples for the tests of both uniform gravity (Fig. 3.2) and interparticle gravity (Fig. 3.3). Prior to the efficiency



(a) (b)

Figure 3.2: A representative example of profiling results from the uniform gravity trials described in Section 3.4.3. The charts show the fractional simulation time spent on each of the three most computationally expensive sets of calculations both before (a) and after (b) the sort and search method enhancements. For uniform gravity tests like the one shown here, those categories are aggregate operations (blue), neighbor search operations (orange), and all other operations (green). Chart sectors are labeled accordingly. This example was taken from the set of box-filling simulations with $N = 6{,}400$ member particles bound into a mix of different aggregate shapes and subjected to uniform lunar gravity. Notably, the percentage of simulation time spent in aggregates decreases from 84.4% to 2.1% with the new search and sort methods. The length of time spent on the neighbor search and "Other" functions is constant between (a) and (b), but the total simulation time decreases in (b) (see Fig. 3.4) due to the decrease in time spent on aggregate operations.



(a) (b)

Figure 3.3: A representative example of profiling results from the interparticle gravity trials described in Section 3.4.3, both before (a) and after (b) the efficiency improvements were applied. This test included 20,000 member particles bound into a mix of different aggregate shapes. The set of tests this example was drawn from are described in greater detail in Section 3.4.3. The two panels are analogous to the ones shown in Fig. 3.2. Note that the second most computationally expensive protocol in this instance before modification is particle self-gravity, in contrast with the example shown in Fig. 3.2.

improvements, aggregate operations accounted for a majority of computation time by a wide margin in both cases. The second-most costly operation was the neighbor search for the uniform gravity tests and gravity calculations in the case of the interparticle gravity tests. In both example trials, the relative cost of the second-most expensive operation compared to the cost of the "Other" category is constant, as these routines remain unchanged. However, the fraction of runtime spent on aggregate calculations shrinks dramatically, from 84.6% of total computation time to 2.1% for the uniform gravity test and from 65.4% to just 0.3% for the interparticle gravity test. This corresponds to a substantial decrease in overall runtime. Notably, in simulations using the new implementation, operations on aggregates collectively take more than an order of magnitude less time than the next most expensive operations. The savings in absolute simulation time are substantial as well. We see a typical decrease in total runtime by a factor of 2 to 3 in self-gravity simulations with $N = 20{,}000$ particles and by a factor of 4 or more in simulations with a uniform

gravity field and $N = 6{,}400$ particles (see Fig. 3.4). In both sets of models, simulations using the newly updated sort and search methods do not take significantly longer than simulations with the same number of free spheres. Interestingly, we see a dependence on the number of particles in each aggregate that is most pronounced in the box-filling simulations (Fig. 3.4A): when using the original routines, simulations with the largest number of aggregates (2-particle dumbbells, with 3,200 aggregates) took significantly longer than simulations with fewer aggregates (8-particle cubes, with 800 aggregates), despite having the same total number of spheres, due to the overhead of repeated brute-force searches. This trend is much less pronounced in the simulations with the new search and sort routines.

We caution that while we can compare models with small aggregates containing the same total number of particles as a simulation with only free spheres, the number of discrete bodies in the aggregate simulations must necessarily be several times smaller than in the spherical simulations and the size of the bodies several times larger. If we want to match the number and size of discrete objects in a simulation with aggregates to a simulation with only free spheres, we would have to replace each sphere with an aggregate of comparable size. Directly replacing spheres with aggregates requires a larger number of total member particles than there were free spheres, because each aggregate must contain several particles, and requires the particles to be smaller than the free spheres so that the total aggregate size matches that of the sphere it is replacing. Including more particles increases the computational load of the gravity calculations (which scale as $O(N \log N)$), and reducing particle size forces smaller time steps in order to resolve particle collisions, as described in Schwartz et al. (2012).

That said, while modeling tens or hundreds of thousands of self-gravitating, non-spherical constituents was previously untenable, these scales are now within reach. To demonstrate this,

we conducted additional trials at increasingly high particle resolution. A total of six tests is divided into pairs of spherical particle and mixed aggregate simulations at low, medium, and high resolutions (see Table 3.1 for labels and results). All six tests consisted of a stationary "cloud" of constituents collapsing to a settled, stable rubble pile. All initial clouds had nearly the same total mass with the only significant difference between each trial being the constituents. At ∼11,000 particles (trials C1 and C2), the wall-clock times approach parity with simulation time. We found that with 2,500 bonded aggregates composed of 10,912 particles (C1), it took 1.20 seconds of wall-clock time to integrate 1 second (or about 2.9×10^{-5} dynamical times) of simulation time on 1 CPU core. The spherical counterpart (C2) took 1.50 seconds to run, very similar to the result from C1. For serial runs, performance is typically comparable between aggregate and sphere-based simulations containing the same total number of spheres, as demonstrated in Fig. 3.4. The largest scale tested used over one million spherical particles, either as free grains (C6) or bound into 250,000 aggregates (C5). One second of integration time required 35.40 seconds of wall-clock time for the sphere-based model and 289.31 seconds for the aggregate model. We also found that tidal disruptions of 10,000 aggregates (∼40,000 particles) could be completed in less than a week on fewer than five CPU cores. These tests were conducted on AMD EPYC 7763 CPU cores using the C compiler included in version 9.4.0 of the GNU Compiler Collection (GCC).

Bonded aggregate performance is comparable to that of spheres in serial, allowing for near real-time simulations for ∼2,500 non-spherical constituents. Parallel scaling for aggregates is good enough to allow for high-resolution non-spherical simulations, but still lags behind performance with only spheres. Nominally, we should see $O(N \log N)$ scaling with total particle number N. However, limitations inherent in `pkdgrav`'s architecture prevent us from achieving

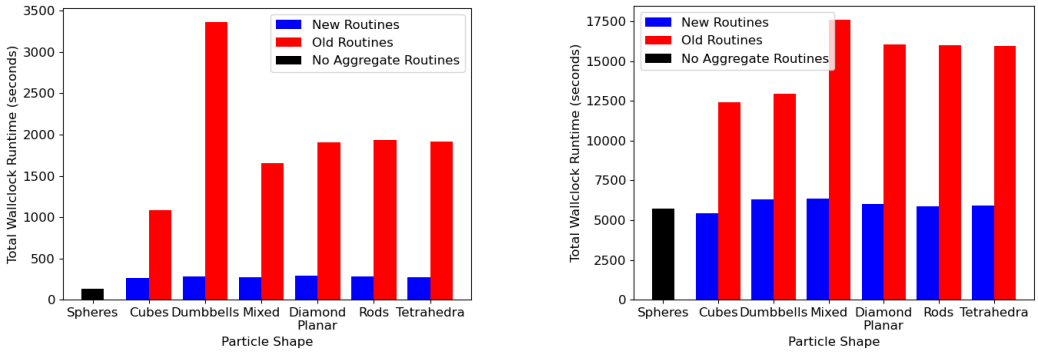
Trial	Composition	No. Constituents	No. Spheres	CPU Cores	Wall-Clock Time for 1s Simulation Time
C1	Mixed Shapes	2,500	10,912	1	1.20 s
C2	Spheres	10,912	10,912	1	1.50 s
C3	Mixed Shapes	25,000	110,150	11	17.19 s
C4	Spheres	110,150	110,150	11	7.51 s
C5	Mixed Shapes	250,000	1,101,336	50	289.31 s
C6	Spheres	1,101,336	1,101,336	110	35.40 s

Table 3.1: A set of six test simulations at increasingly high particle resolutions, with one spherical and one aggregate test at each resolution. Note that to match the constituent resolution of a sphere-based simulation rather than the particle resolution, the corresponding aggregate simulation would need to include *more* spheres than its counterpart, with the ratio depending on the composition of the aggregates. Each aggregate-based trial has the same number of spherical particles as its counterpart. In serial simulations, performance is comparable between sphere-based and aggregate simulations, while parallelized aggregate-based runs become more costly.

this ideal. While our optimization efforts have substantially improved these shortcomings, some clear obstacles remain. Currently, we are constrained to store all top-level aggregate information on a single processor when running in parallel, leading to a substantial bottleneck during multi-core simulations. In addition, aggregate operations require all particles to be reordered by aggregate index, which is not required for sphere-based simulations. Since addressing these design limitations would require fundamentally reworking the structure of `pkdgrav`, we leave this work for future `pkdgrav` development or for the developers of other codes attempting similar implementations.

3.5 Applications and future work

Here we provide three proof-of-concept applications that make use of our approach to soft-sphere contacts with bonded aggregates. The results described here are meant to illustrate the value of the method we describe in this article. These demonstrations are not intended to serve as comprehensive studies, but are part of ongoing work.



(a) (b)

Figure 3.4: Total wall-clock runtime as a function of aggregate shape for box-filling (a) and collapse (b) simulations run in serial. The blue bars (the left side of the histogram bar pairs) indicate total wall-clock simulation time with the improved search and sorting methods; the red bars (the right side of pairs) indicate total wall-clock simulation time with the original brute-force search. Black bars indicate time spent in analogous free-sphere simulations, with no aggregate options compiled at runtime.

3.5.1 YORP spin-up

Small solar system bodies particularly in the inner solar system are subject to the Yarkovsky–O'Keefe–Radzievskii–Paddack effect (YORP) in which asymmetries in the absorption and re-radiation of solar energy due to surface irregularities result in a small net torque, causing a change in the body's overall spin (Rubincam, 2000). Over long time spans these small net torques can cause significant spin-ups, potentially reshaping the object (Sánchez and Scheeres, 2012; Scheeres, 2015; Cotto-Figueroa et al., 2015), causing it to shed material (Sánchez and Scheeres, 2012; Walsh et al., 2012), or even form a binary or higher-multiple system (Ćuk, 2007; Walsh et al., 2008; Jacobson and Scheeres, 2011).

There is much interest in understanding the effects of irregular constituent shape on spin-up outcome, particularly given that some small bodies are such fast rotators that they may require either cohesion or enhanced shear strength (as provided by irregular grain shape, for example) to

remain presently stable (Rozitis et al., 2014; Hirabayashi and Scheeres, 2014; Zhang et al., 2017). We are modeling spin-up following the approach of Zhang et al. (2017) by applying an artificial rotational acceleration to a rubble pile, but, unlike in the earlier studies that used independent spheres, we use non-spherical bonded aggregates instead. In contrast with Zhang et al. (2017), we apply a constant angular momentum increment at each time step, as opposed to a constant spin period increment. Fig. 3.5 shows an example of a rubble pile composed of aggregates losing mass during a spin-up event in `pkdgrav`.

Our preliminary work indicates that grain shape does have a measurable effect on rubble-pile spin-up. For our trial runs, we selected a variety of simple shapes: 2-particle "dumbbells," 4-particle planar diamonds, 4-particle rod-shaped aggregates, 4-particle tetrahedra, and 8-particle cubes. These shapes are shown in Fig. 3.1. For each shape, we created a progenitor rubble pile composed solely of equal-sized aggregates of that shape. We also included both a rubble pile composed of a mix of all five shapes and four rubble piles composed only of the original, spherical `pkdgrav` particles. We refer to these runs using the labels S1–S10 (see Table 3.2). All progenitor bodies were given equal total mass and all constituent spheres had equal mass density. All progenitors contain 20,000 `pkdgrav` spheres, with the exception of S8 (mixed shapes), which contains 21,764 spheres. Using randomized mixed shapes doesn't allow for an easy division of particles, and we opted instead for the round number of 5,000 aggregates for simpler comparison with other trials. The bulk densities of the settled progenitor rubble piles vary due to inherent differences in packing efficiencies between the different aggregate shapes, so comparing the trials to each other requires some care.

For each settled rubble pile, we adopted a nominal predicted critical spin period P_{crit} =

$\sqrt{\frac{3\pi}{G\rho}}$, computed by equating surface gravity with centrifugal acceleration for a particle at the equator of a rigid, spinning sphere of the given bulk density. We then slowly increased the spin rate of each progenitor until it began to shed material, where we considered a "disruption" to have occurred when a rubble pile's radius has increased by 5%. Note that the specific stopping criterion of 5% used here is somewhat arbitrary; we are interested in the marginal differences in resistance to disruption for each composition, rather than determining a specific disruption threshold for each shape. The observed critical spin period for each composition is recorded as a percentage of the nominal critical spin period calculated for that rubble pile. This allows us to compare trials with different bulk densities to each other directly. We call this metric the "critical spin ratio." A number greater than 100% indicates a rubble pile that was easier to disrupt than the nominal case, while a number less than 100% means the rubble pile was more difficult to disrupt. The results are shown in Table 3.2. To determine the degree to which our results are dependent on the randomness inherent in the progenitor generation and collapse procedure, we repeated trial S1 once and trial S2 twice with new, randomly generated rubble piles. The two S1 spin-ups gave critical spin ratios of 102.5% and 102.6%, while the three S2 trials gave ratios of 102.6%, 101.7%, and 100.3%. In both cases, the variation is small enough that we feel confident in the conclusions drawn below. Given that we are only presenting a small number of tests for the purpose of a proof of concept, we do not include any further statistical analysis. For both S1 and S2, the average critical spin ratio of all realizations is the number quoted in Table 3.2.

We have attempted to select test cases that are as similar as possible to allow direct comparison. However, there are some difficulties in engineering a true like-against-like comparison in a test of this nature. Consider trials S1 and S9, for example. Both progenitor rubble piles

are composed of 20,000 spherical particles, but in the case of S9 these particles are bound into only 5,000 separate grains, as opposed to 20,000 grains in S1. S1 and S9 have different effective "resolutions." To account for this, we have included trials S2, S3, and S4 which use progenitor bodies made up of 20,000, 10,000, 5,000, and 2,500 spheres, respectively. We note that in S1–S4 there is an apparent trend of increasing resistance to disruption with increasing resolution, which saturates at around 10,000 particles, though the trials are too limited to draw any firm conclusions. It may be that increased resolution would increase resistance to disruption in the case of other shapes as well. In any case, it is evident from Table 3.2 that using non-spherical shapes increases resistance to spin-up disruption in all cases, regardless of the resolution. The more rounded, "sphere-like" dumbbells, cubes, and tetrahedra have higher critical spin ratios and so perform more like spheres, though they do confer some degree of additional strength. The case of cubes is more difficult, since they differ by over 10% in critical spin ratio from the most similar spherical trial (S4), but also score close to 100%, indicating a critical spin period fairly close to the nominal value. The more elongated diamonds and rods add substantial resistance, with a $\sim$10% reduction in critical spin ratio compared to S3, and the lowest critical spin ratios of all trials. Trial S8, with mixed shapes, falls in between the extremes.

These early results suggest resistance to breakup in fast-rotating rubble-pile bodies is affected by particle shape, as expected. Walsh et al. (2022a) showed that the asteroid Bennu has nearly zero interparticle cohesion, at least near its surface. It may be that particle shape alone could account for the stability of fast rotators without the need to invoke surface cohesion, though a more comprehensive study to establish the typical magnitude of the effect is needed.

Trial	Composition	Number of Grains	Critical Spin Ratio
S1	Spheres	20,000	$102.5 \pm 0.1\%$
S2	Spheres	10,000	$101.5 \pm 1.2\%$
S3	Spheres	5,000	104.5%
S4	Spheres	2,500	108.6%
S5	Dumbbells	10,000	98.4%
S6	Cubes	2,500	98.2%
S7	Tetrahedra	5,000	97.8%
S8	Mixed Shapes	5,000	96.8%
S9	Diamonds	5,000	96.3%
S10	Rods	5,000	94.0%

Table 3.2: Observed critical spin as a fraction of the nominal predicted value (see Section 3.5.1). A rubble pile composed of many, only-spherical particles (S1) nearly matches the predicted value, while a body made up of rod-shaped grains deviates most strongly. All progenitor bodies have approximately equal total mass and bulk radius.

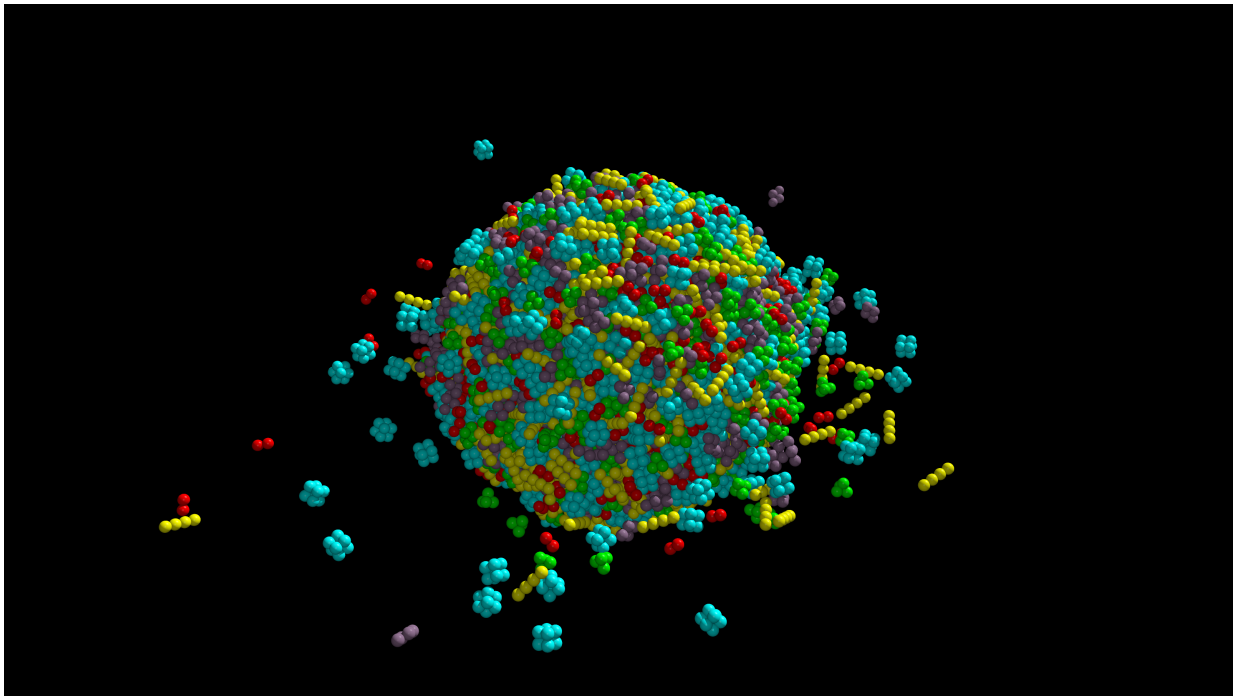


Figure 3.5: A rubble pile composed of 5,000 bonded aggregates (21,765 particles) shedding mass after being subjected to spin-up. Here we use a variety of regular shapes, including cubes, tetrahedra, and 4-particle rods, among others. Aggregates are color-coded by shape.

3.5.2 Tidal encounters

When small solar system bodies like asteroids and comets pass near large, dense bodies, they can be subject to significant tidal forces. Depending on the specific parameters of the encounter, these tidal forces can cause small disturbances (Richardson et al., 1998; Yu et al., 2014; DeMartini et al., 2019), resurfacing (Binzel et al., 2010; Yu et al., 2014), changes in spin state (Scheeres et al., 2004, 2005), and even reshaping or disruption of the entire body (Bottke Jr et al., 1999; Walsh and Richardson, 2006; Zhang and Michel, 2020). Some have speculated that the interstellar object 'Oumuamua's unusual elongated shape could be the result of the same tidal encounter that ejected it from its host system (Ćuk, 2018; Zhang and Lin, 2020). It is plausible that the additional internal friction conferred by non-spherical particles could play a significant role in determining its very elongated shape after a tidal encounter, although Zhang and Lin (2020) instead invoke sintering of bonds between surface particles as a possible explanation. While recent work by Zhang and Michel (2020) applied spherical-particle version of `pkdgrav` to the question of tidal distortion of rubble piles, the role of particle shape in this process has not been investigated. See Fig. 3.6 for an illustration of a tidal disruption simulation using non-spherical particles in `pkdgrav`.

We conducted a set of ten trials (T1–T10) to assess the effects of particle shape on tidal reshaping and disruption. The progenitor rubble piles are identical to those used in Section 3.5.1 and are named accordingly, so the rubble pile in trial T1 is the same as the rubble pile in trial S1, and so on. In each trial, a rubble pile passes by an Earth-mass particle with a close-approach distance of 1.2 Earth radii and is disrupted by tidal forces. For each trial, we set a nominal disruption threshold distance using the Roche limit $d \approx 1.26R\sqrt[3]{\rho_M/\rho_m}$, where R is the radius

of the primary (disrupting) body, ρ_M is the density of the primary, and ρ_m is the bulk density of the rubble pile. This value is not intended to serve as a "prediction" of the disruption threshold. In analogy with Section 3.5.1, we use it as a normalization factor to allow direct comparison of disruption thresholds for different trials with different bulk densities. For each trial, we record the distance of the rubble pile from the primary when the rubble pile's bulk radius has increased by 0.5%[1] of its initial value. We then quote this distance relative to the nominal disruption threshold calculated for that progenitor body as a percentage. We refer to this value as the "disruption ratio." Higher values indicate lower resistance to tidal disruption. Results for each trial are shown in Table 3.3. We again conducted two realizations of T1 and three realizations of T2 to determine how resistant our results were to random variation. The disruption ratios were 83.8% and 86.8% for the T1 trials and 77.0%, 78.4%, and 79.1% for the T2 trials. As in the case of Section 3.5.1, we feel that the natural variation here is small enough not to affect our conclusions. For T1 and T2, we quote the average disruption ratio across all realizations.

In contrast to Section 3.5.1, we see a trend toward decreasing resistance to disruption with increasing resolution when comparing runs T1–T4. This is reasonable since, near the disruption threshold, the smaller constituents in the higher-resolution simulations will be able slide past each other under shear forces more easily than larger constituents will. The difference in results between spheres (T1–T4) and bonded aggregates (T5–T10) is more varied in the case of tidal disruption as it was in the case of spin-up disruption. When comparing runs of the same resolution the spherical progenitors are always easier to disrupt, with the exception of dumbbells. Rods (T10) show the greatest change in disruption threshold when compared to their spherical analogs,

[1] As in Section 3.5.1, the 0.5% disruption threshold figure is arbitrary. In this case, it was chosen in order to produce a disruption ratio near 100% for trial T1, recognizing that these are not fluid bodies and therefore we would expect them to disrupt interior to the normalization distance. Threshold values of 1% and 5% produced comparable results.

Trial	Composition	Number of Grains	Disruption Ratio
T1	Spheres	20,000	$85.3 \pm 1.5\%$
T2	Spheres	10,000	$78.2 \pm 1.2\%$
T3	Spheres	5,000	74.7%
T4	Spheres	2,500	63.8%
T5	Dumbbells	10,000	78.7%
T6	Cubes	2,500	54.9%
T7	Tetrahedra	5,000	62.7%
T8	Mixed Shapes	5,000	70.1%
T9	Diamonds	5,000	60.8%
T10	Rods	5,000	58.6%

Table 3.3: Observed disruption threshold distance as a fraction of the Roche limit for each tidal disruption trial (see Section 3.5.2). With the exception of T5, trials with bonded aggregates show more resistance to disruption than the analogous trials using spherical particles.

followed by planar diamonds (T9). By this metric, rod-shaped and diamonds-shaped aggregates are again the most difficult to disrupt. This is reasonable, given their highly non-spherical shapes. The overall tendency again is toward greater resistance for progenitors composed of non-spherical shapes. As in the case of spin-up, a more in-depth study is needed to fully characterize and quantify the effect.

3.5.3 Brazil nut effect

The Brazil nut effect (BNE) (Rosato et al., 1987) is a suggested mechanism for the vertical migration of boulders in a granular medium, in which frictional interactions between particles permit larger blocks to rise to the surface when subjected to repeated seismic shaking. Granular convection, where constituents move down along confining walls and up through the central medium, is the dominant mechanism in confined experiments (Asphaug, 2007), but simulations using periodic boundary conditions find that small grains moving into empty areas below the large constituents ("void-filling") plays a larger role in low-gravity BNE models (Maurel et al.,

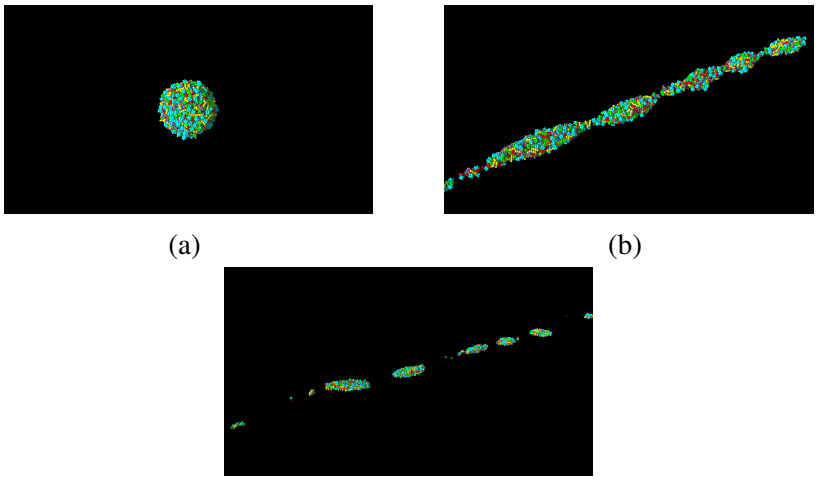


Figure 3.6: A rubble pile body composed of 5,000 bonded aggregates (21,765 particles) before and after a tidal encounter. As in Fig. 3.5, aggregates are color-coded by shape.

2016; Chujo et al., 2018). Both convection and void-filling are influenced by friction: when interactions disrupt the flow of grains past one another, their ability to convect or move into voids is disrupted, thus halting the rise of the "Brazil nut" (large intruder). Furthermore, we expect to find more voids in systems of irregularly shaped grains due to the increased shear strength of aggregates versus their spherical counterparts, leading to a more porous equilibrium state when packing under uniform gravity. The BNE has been investigated several times in the past with `pkdgrav` (Matsumura et al., 2014; Maurel et al., 2016; Chujo et al., 2018) and recent studies have shown how ellipsoidal shapes influence the process of boulder stranding (Zeng et al., 2022). However, no investigation has yet been published using the varied geometries that we can model with our approach. The non-spherical BNE could also explain the highly porous, boulder-dominated surfaces of rubble piles like Bennu, the target of the recent OSIRIS-REx sample return mission (Walsh et al., 2022a).

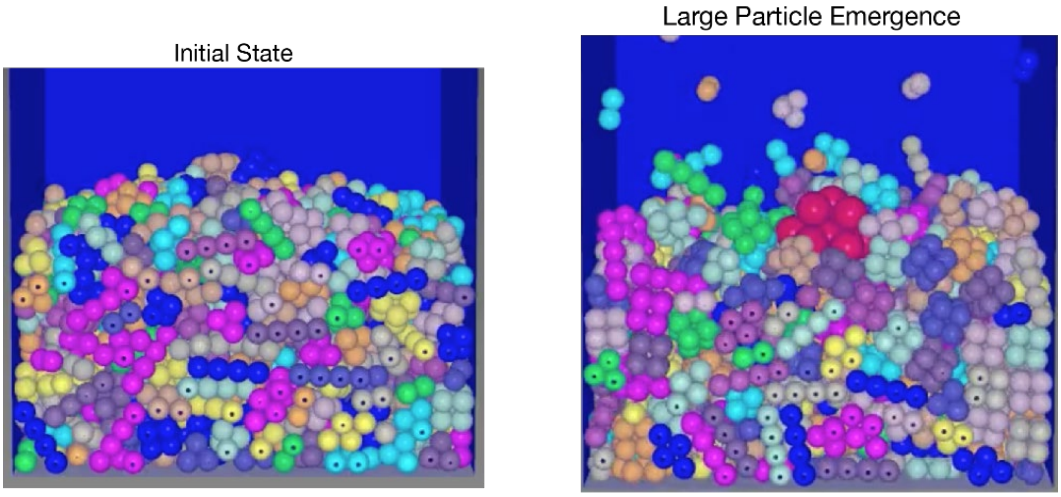
We conducted a set of 7 sample trials (B1–B7) to assess the effects of particle shape on the

Trial	Composition	Intruder Shape	Number of Grains	Rise Time (cycles)
B1	Dumbbells	Sphere	2,000	18
B2	Tetrahedra	Sphere	1,000	13
B3	Diamonds	Sphere	1,000	28
B4	Rods	Sphere	1,000	22
B5	Cubes	Sphere	500	–
B6	Spheres	Sphere	4,000	36
B7	Mixed Shapes	Axisymmetric Aggregate	900	32

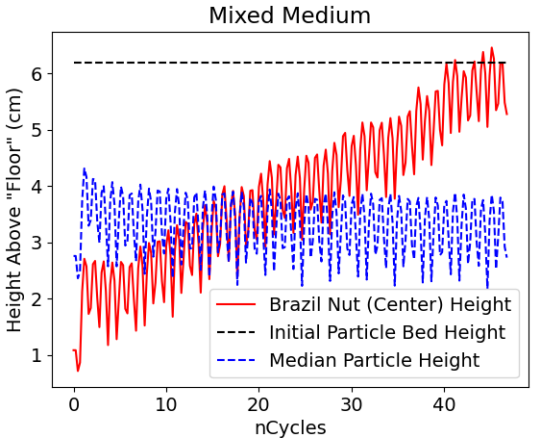
Table 3.4: Observed rise time of a large intruder grain in a medium of aggregates (B1–B5, B7) or spheres (B6). The intruder notably rises faster in media made of irregularly shaped grains in all cases except for B5, where it does not rise at all, likely due to high packing efficiency in the medium. All particles have equal density but the intruder is the largest constituent in radius.

rise speed of large intruder constituents in a medium of irregularly shaped grains. We initialize our system for these trials by placing a large spherical (for B1–B6) or aggregate (B7) intruder at the bottom of a rectangular chamber of base area 10×10 cm^2 under uniform Earth gravity ($a_g = -9.8$ m s^{-2}). We then fill the chamber with grains of the shape defined in Table 3.4 such that the total number of particles is about 4,000. The intruder has radius 1.5 cm and the radius of aggregate component particles (or free spheres in B6) is 0.25 cm but all particles in the simulation have an equivalent density of 2.7 g cm^{-3} and the same gravel-like friction parameters (Zhang et al., 2017). Once we have filled the chamber, we force a sinusoidal oscillation of the walls in the system, with oscillation amplitude 1 cm and frequency 54.2 rad sec^{-1}, giving the system a dimensionless acceleration (Froude Number; see Matsumura et al. (2014)) of about 3. We shake the system for 50 cycles and track the rise of the large intruder through the medium, as seen in Figs. 3.7 and 3.8, with results shown in Table 3.4.

Fig. 3.7 shows the initial frame and the frame of intruder emergence from trial B7, a sample model of the BNE with a mix of irregular aggregate shapes in the medium around a larger intruder aggregate. The third frame in Fig. 3.7 shows the height of the center of mass of the intruder



(a) (b)



(c)

Figure 3.7: Images of the initial packing of a system of mixed aggregates (color-coded by shape) with a large intruder buried in the medium (a) and the intruder (red) emerging after repeated sinusoidal shaking of the medium (b). Panel (c) is a plot demonstrating rise of the intruder particle and thus the non-spherical Brazil nut effect. In panel (c), the red, solid line demonstrates the height of the center of the intruder particle, the blue, dashed line indicates the median height of the non-intruder particles in the medium, and the black, dashed line indicates the initial height of the particle bed.

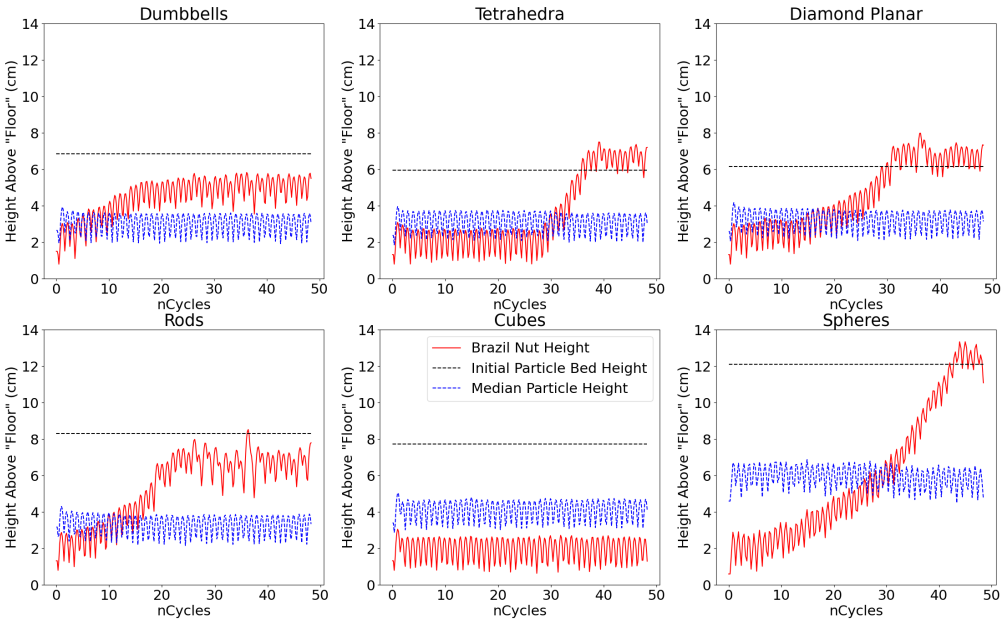


Figure 3.8: A grid of plots demonstrating the rise of the intruder particle surrounded by media of different symmetric aggregate shapes, each with the same curves and definitions as described in Fig. 3.7C. From left to right and top to bottom, the media around the intruders are composed of: 2-particle dumbbell shapes, 4-particle tetrahedra, 4-particle planar diamond shapes, 4-particle rods, 8-particle cubes, and free spheres.

(red), the median center of mass height for non-intruder aggregates (blue) and the initial bed height (black). Similar rise plots are shown for trials B1–B6 in Fig. 3.8. What we can see from these plots and Table 3.4 is that in almost every case tested, the speed of the intruder's rise, measured by inspection from when it moves above its average initial height (averaged over the oscillations) at the bottom of the container to when it reaches its final height (again, averaged over the oscillations) at the top, is faster with a medium made of irregularly-shaped aggregates than in one composed of only spheres. An interesting exception comes in the case of a medium constructed of cube shaped aggregates (B5), which suppresses the rise of the Brazil nut. This may be due to the relative sizes of the cubes to the intruder (an average ratio of $1 : \sqrt{3}$) reducing the efficiency of void filling. It could also be that a medium constructed of only cubes has a significantly higher effective shear strength than the other aggregate media we tested, and thus requires more vigorous shaking to move the intruder up through the medium. All of these models require further work, including testing with periodic lateral boundaries (Maurel et al., 2016) and investigations with different and potentially more realistic shaking waveforms. However, the initial results shown here indicate that modeling with irregularly-shaped grains can lead to a systematically faster rise of large subsurface blocks (DeMartini and Richardson, 2020).

3.6 Conclusion

Grain shape is known to play an important role in determining shear strength and resistance to deformation in the context of granular media (Wegner et al., 2014). However, the effects of non-spherical grains have not been considered in most numerical work on rubble-pile bodies to date or have at best been restricted to limited, low-resolution studies. We have presented an

implementation of non-spherical grains of arbitrary shape with soft-sphere contacts in our DEM N-body code `pkdgrav`. As far as we are aware, `pkdgrav` is the only code that combines an N-body gravity solver and a soft-sphere contact model with an efficient "glued-sphere" approach to constructing non-spherical constituents. `pkdgrav`'s existing optimizations and parallel implementation allow us to conduct simulations of self-gravitating, colliding, irregular grains complete with gravitational and contact torques, even when including hundreds of thousands of constituents. Preliminary tests of our method, applied to YORP spin-up, tidal disruption, and the Brazil nut effect, show the potential importance of simulations that include non-spherical grains. Grain shape may be a factor in determining the resistance to disruption of rubble-pile bodies subject to planetary tides or spin-up and could help explain efficient boulder stranding on bodies with regolith surfaces via the Brazil nut effect, and these effects are evident in our trial simulations. Our method improves realism while retaining high dynamic range and efficiency.

Chapter 4: The effect of particle shape on tidal disruption

4.1 Preface

This chapter is an application of the work from Chapter 3 to tidal disruptions. It has not been submitted for publication yet, but it is more or less complete. The prepared manuscript also includes an abstract and an acknowledegments section, neither of which are reproduced here. I performed all of the simulations included in this chapter, conducted the analyses, and wrote the manuscript with editing by Derek Richardson. I also wrote the great majority of a Python package, `pkdtools`, that I used to perform the bulk of the analyses presented here, although Joseph DeMartini also contributed to this project. Most of the features of `pkdtools` are documented in Appendix B, although some recently added capabilities are not. We are also expecting our submitted manuscript to include a contribution of some text from Kevin Walsh, a collaborator on this project, that puts the work in context with a related study of the NEA size distribution and tidal disruption. Otherwise, this chapter will be submitted largely as it appears here along with an abstract and an acknowledgements section. We had initially intended to include an additional chapter in this dissertation in a spirit similar to this one, but applying our particle shape model to YORP spin-up rather than tidal disrutpion. Unfortunately, time constraints did not allow for this.

4.2 Introduction

Close encounters between small bodies and planets are a natural result of the dynamical evolution of the solar system. Initially, it was believed that internal strength would be more than enough to prevent the tidal disruption of objects $\lesssim$ 200 km in diameter (Jeffreys, 1947; Walsh, 2018). However, as knowledge of the rubble-pile nature of asteroids developed, this notion changed. It is now understood that asteroids and comets less than about 10 km in diameter are nearly strengthless relative to rigid bodies (Richardson et al., 2002; Walsh, 2018). As a consequence, small bodies present far less resistance to tidal forces than initially thought. The effects of tidal forces on rubble-pile bodies depend on many factors, including the distance of closest approach between the objects, their relative speeds, bulk densities, rotation state, internal cohesion, and constituent particle shape.

4.2.1 Tidal processes

Tidal disturbance drives a variety of solar system processes involving small bodies. Relatively gentle encounters that leave a body intact can still prompt the motion of surface material, leading to the resurfacing of objects previously reddened by space weathering. Tidal-induced resurfacing is a likely cause of the difference in color observed between S-type and Q-type asteroids (Binzel et al., 2010). Rubble piles can also be wholly reshaped as a result of a close flyby. Depending on the parameters of the encounter, the potential deformation ranges from subtle to dramatic elongation. Possible examples include 1620 Geographos and the interstellar object ‘Oumuamua (Bottke Jr et al., 1999; Zhang and Lin, 2020). If the forces are strong enough relative to the body’s strength and gravitational binding energy, the rubble pile may lose mass or

even fragment completely (e.g., Richardson et al., 1998; Holsapple and Michel, 2006; Zhang and Michel, 2020).

Tidal disruption of small bodies is a possible mechanism for forming binaries. In the aftermath of an encounter violent enough to fracture a rubble pile, some fragments may remain gravitationally bound to form a binary or even a higher-multiple system (Walsh and Richardson, 2006; Ćuk, 2007), although Walsh and Richardson (2008) determine that this cannot be the dominant mechanism in forming NEA binaries. It has also been suggested that spin-up-induced fission of near-Earth asteroids (NEAs) could be triggered by tidal forces during close planetary encounters (Margot et al., 2002; Walsh and Richardson, 2006).

The influence of tidal forces on small rubble piles may be responsible for the crater chains or "catenae" observed on terrestrial bodies. Sharply-defined, linear collections of craters on Callisto and Ganymede have been ascribed to the successive impacts of fragments of cometary-density rubble piles broken up by tides prior to impact (Melosh and Schenk, 1993). It has been proposed that similar features on the Moon formed in a similar fashion (Bottke Jr et al., 1997; Wichman and Wood, 1995).

Among the most notable tidal encounters in recorded history was the breakup of the comet Shoemaker-Levy 9 (SL9) under the gravitational influence of Jupiter. During a close pass of Jupiter in 1992, SL9 was pulled apart into dozens of pieces. These eventually coalesced into $\sim$20 fragments before impacting Jupiter's southern hemisphere on a subsequent orbit. This event served as an important probe into the interior structure of comets, as the breakup implied that SL9's bulk density and internal strength were both quite low (Asphaug and Benz, 1996). More recently, there has been a great deal of interest in the near-Earth asteroid 99942 Apophis. When first discovered, Apophis was classed as a Potentially Hazardous Asteroid (PHA). Subsequent

observations have ruled out the possibility of an impact; instead, Apophis is now predicted to execute a close flyby (within $\sim$ 5-6 Earth radii) of the Earth in the year 2029 (Brozović et al., 2018). At this distance, the influence of tidal forces on Apophis may be measurable (DeMartini et al., 2019).

4.2.2 Numerical models of tidal interactions

Early work by Roche provided analytical disruption limits for strengthless, fluid bodies (Roche, 1847). However, the complexities evident in the breakup of SL9 spurred the development of an approach that could treat bodies as being made up of discrete fragments (Asphaug and Benz, 1994; Solem and Hills, 1996; Richardson et al., 1998; Walsh, 2018). In the intervening decades, the tidal disruption of rubble piles has been studied using a variety of numerical approaches. Most prominent among them are smoothed-particle hydrodynamics (Benz and Asphaug, 1995; Jutzi, 2015) and the discrete-element method (DEM) (Walsh and Richardson, 2006; Movshovitz et al., 2012; Veras et al., 2016; Zhang and Michel, 2020). DEM is the focus of this work. In the context of small solar system objects, DEM simulations represent rubble piles as collections of self-gravitating, discrete particles (usually spheres). The state of the art of the discrete element method has advanced considerably since its first applications to rubble pile tidal disruption, with a steady push towards increased particle resolution and increased realism. Many implementations now use the soft-sphere variant of DEM (SSDEM). Instead of instantaneous, sequentially computed collisions, SSDEM allows for simultaneous, extended contacts between particles (Wada et al., 2006; Sánchez and Scheeres, 2011; Schwartz et al., 2012). This approach is well-suited to self-gravitating granular systems and allows for increased realism when modeling rubble piles.

4.2.3 Particle shape

Particle shape is another area in which the realism of astronomical DEM simulations is improving. Naturally, real-world rubble piles are composed of grains of a variety shapes and sizes—the spheres typically used in DEM simulations were chosen to reduce the complexity of calculating interparticle gravity and contact forces as well as overall computational cost. In the broader field of granular dynamics, it has long been accepted that particle shape influences the behavior of granular media like rubble piles. Spheres can slide past one another and rearrange themselves with relative ease even with parameterized friction applied, while non-spherical grains tend to interlock with each other and generally inhibit bulk motion of the medium in a way that spheres cannot (Wegner et al., 2014). To account for this, many modern granular dynamics codes can model non-spherical particles (Zhao et al., 2006; Longmore et al., 2013; Nguyen and Plimpton, 2019). Even so, there are relatively few codes that have combined non-spherical shapes with N-body capabilities and most that have are limited in their ability to scale to large numbers of constituents (Roig et al., 2003; Movshovitz et al., 2012; Ferrari and Tanga, 2020; Sánchez et al., 2021).

The SSDEM-capable N-body code `pkdgrav` (Richardson et al., 2000; Stadel, 2001) was recently modified to accommodate non-spherical particles with soft-sphere contacts via the "glued sphere" method (Marohnic et al., 2023). Combining this new feature with `pkdgrav`'s existing efficiency advantages allows us to model rubble piles composed of tens or even hundreds of thousands of non-spherical, self-gravitating particles. In this work, we conducted 1,890 simulations using `pkdgrav` to study the influence of particle shape on the outcome of tidal encounters between a rubble pile and an Earth-mass object. Section 4.3 outlines the basic workings of our

code and our experimental approach. We present our results in Section 4.4, and offer some final thoughts in Section 4.5.

4.3 Methodology

This section briefly describes the modified version of `pkdgrav` used to conduct this work, as well as the experimental design and analysis methods. For this study, we carried out six trials consisting of 1,890 tidal encounter simulations. Each trial is based on single rubble-pile prototypes, which differ from each other in their constituent shapes and resolutions. In each trial, copies of the prototype are subjected to tidal encounters with an Earth-mass object under varying encounter geometries.

4.3.1 Simulation techniques

All simulations described in this work use the *N*-body code `pkdgrav`. `pkdgrav` uses a parallelized, k-D tree for finding neighbors and calculating interparticle gravitational forces and also includes a soft-sphere discrete element method (SSDEM) for mediating particle contacts. Under the soft-sphere model, particles in contact are allowed to interpenetrate slightly and in turn experience a restoring spring force. The tree allows for sufficient speed when computing interparticle gravitation at high particle resolutions, while the soft-sphere model is necessary for capturing realistic behavior during prolonged multi-particle contacts. This combination is ideal in the context of large, self-gravitating granular systems and `pkdgrav` is thus well-suited to modeling rubble-pile asteroids. Readers may refer to Schwartz et al. (2012) and Zhang et al. (2017) for a more detailed treatment of soft-sphere contacts in `pkdgrav`. More recently, `pkdgrav`

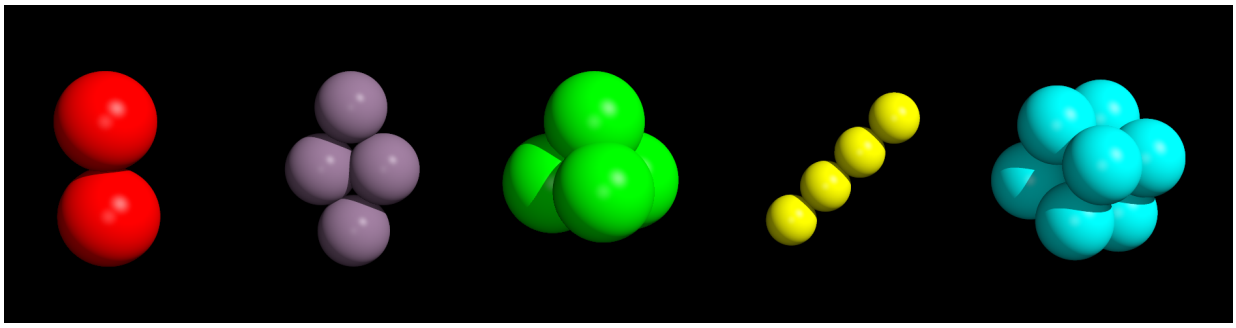


Figure 4.1: The five bonded aggregate shapes used in this study. From left to right: a two-particle dumbbell, a four-particle planar diamond, a four-particle tetrahedron, a four-particle rod, and an eight-particle cube.

has been adapted to support non-spherical constituents with soft-sphere contacts while retaining most of the efficiencies afforded by the existing tree code and parallelization scheme. We refer to non-spherical constituents in `pkdgrav` as "bonded aggregates" or just "aggregates." They are constructed from standard spherical particles using the "glued-sphere" method—`pkdgrav` spheres in a single aggregate have their positions fixed relative to one another and behave as a single rigid body. There are no limitations on the arrangement of spheres in aggregates, but in this study we confine ourselves to five different symmetric shapes for the sake of simplicity (see Fig. 4.1). Accelerations and torques are still computed at the level of individual spheres, which then relay these inputs to their respective aggregates. Thus, aggregates will generally feel not only forces but also torques from both contacts and interparticle gravity (see Marohnic et al. (2023) for details).

4.3.2 Rubble-pile prototypes

To study the effects of rubble pile constituent shape during tidal encounters, we conducted 1,890 encounter simulations in total in `pkdgrav` across six different trials. In all simulations the fundamental arrangement is the same: a close encounter between a spherical rubble-pile

body and a single spherical particle with mass equal to that of the Earth (see Fig. 4.2 for a simulation snapshot of a disrupted rubble pile). We use six different rubble-pile prototype objects that differ in the number and shapes of their constituents. Each of these is the basis for one of six separate trials. Two of the rubble piles are composed only of free spheres, two are composed only of four-particle rod-shape aggregates, and two are composed of a random mix of the regular aggregate shapes shown in Figure 4.1. Each of these three pairs includes two versions of the rubble pile, with one made up of approximately 2,500 constituents and the other approximately 10,000 constituents. Since the overall sizes of all six rubble piles are similar, each constituent shape pair effectively consists of a "low-resolution" and a "high-resolution" rubble pile. All six prototypes have zero initial spin and zero interparticle cohesive force applied.

Our primary goal is to study the relative effects of different constituent shapes on tidal reshaping and disruption, but we will also consider the effects of constituent resolution. We have tried to make these six rubble piles as similar to each other as possible beyond their constituent shapes and resolution to allow for convenient and meaningful comparison. However, we are faced with some inherent limitations. Since different constituent shapes have naturally different packing fractions, rubble pile bulk densities will differ substantially if all other attributes are held constant. Likewise, with a fixed bulk density and number of constituents, matching rubble pile bulk radii is difficult. We addressed this by comparing measured bulk densities in our rubble piles and fine tuning the mass densities and radii of the spherical particles until bulk densities were nearly equal. We have also attempted to keep all six prototypes as close to spherical as possible. This is more difficult for the four prototypes with bonded aggregate composition, largely due to their reduced ability to "slump" together when compared with rubble piles made up of spheres. This is reflected in slightly higher initial elongation values for all four bonded aggregate rubble

Label	Composition	N Constituents	N Spheres	Bulk Density (g/cm^3)	Bulk Radius (km)	Elongation
S1	Spheres	2,500	2,500	1.61	1.36	0.012
S2	Spheres	10,011	10,011	1.61	1.13	0.013
M1	Mix	2,280	10,020	1.61	1.23	0.039
M2	Mix	10,020	43,874	1.61	1.20	0.022
R1	Rods	2,505	10,020	1.62	1.26	0.032
R2	Rods	10,020	40,080	1.63	1.20	0.022

Table 4.1: A description of the six rubble pile prototypes featured in this work. The elongation metric quoted here is defined in Section 4.4.3.

pile prototypes than for the two spherical rubble piles (see Tab. 4.1). Given the focus of this work, we chose to prioritize keeping bulk density and constituent number as consistent as possible across all six initial rubble piles at the expense of bulk radius and total mass. The initial rubble piles were created by generating spherical clouds of free spheres or aggregates and allowing them to collapse under self gravity until fully settled. To make fine adjustments to a rubble pile's bulk density, we increased or decreased particle densities and allowed the rubble pile to resettle. In this way, we iteratively adjusted any discrepant rubble piles until no bulk densities differed by more than $\sim 1\%$. The six trials and their corresponding rubble pile prototypes are listed in Table 4.1.

4.3.3 Tidal encounter simulations

Each tidal encounter trial is based on just one of the six rubble pile prototypes. A single trial includes several hundred simulations of a close encounter between its associated rubble pile and a single, Earth-mass, spherical particle. Across a single trial, the only variables are the initial speed and the encounter distance. We consider relative speeds at infinity v_∞ from 0 to 20 km/s with an increment of 1 km/s and close encounter distances q between 1.1 and 2.5 Earth radii $R_\oplus$ with an increment of 0.1 $R_\oplus$. This leaves us with a total of 315 simulations for each trial, or

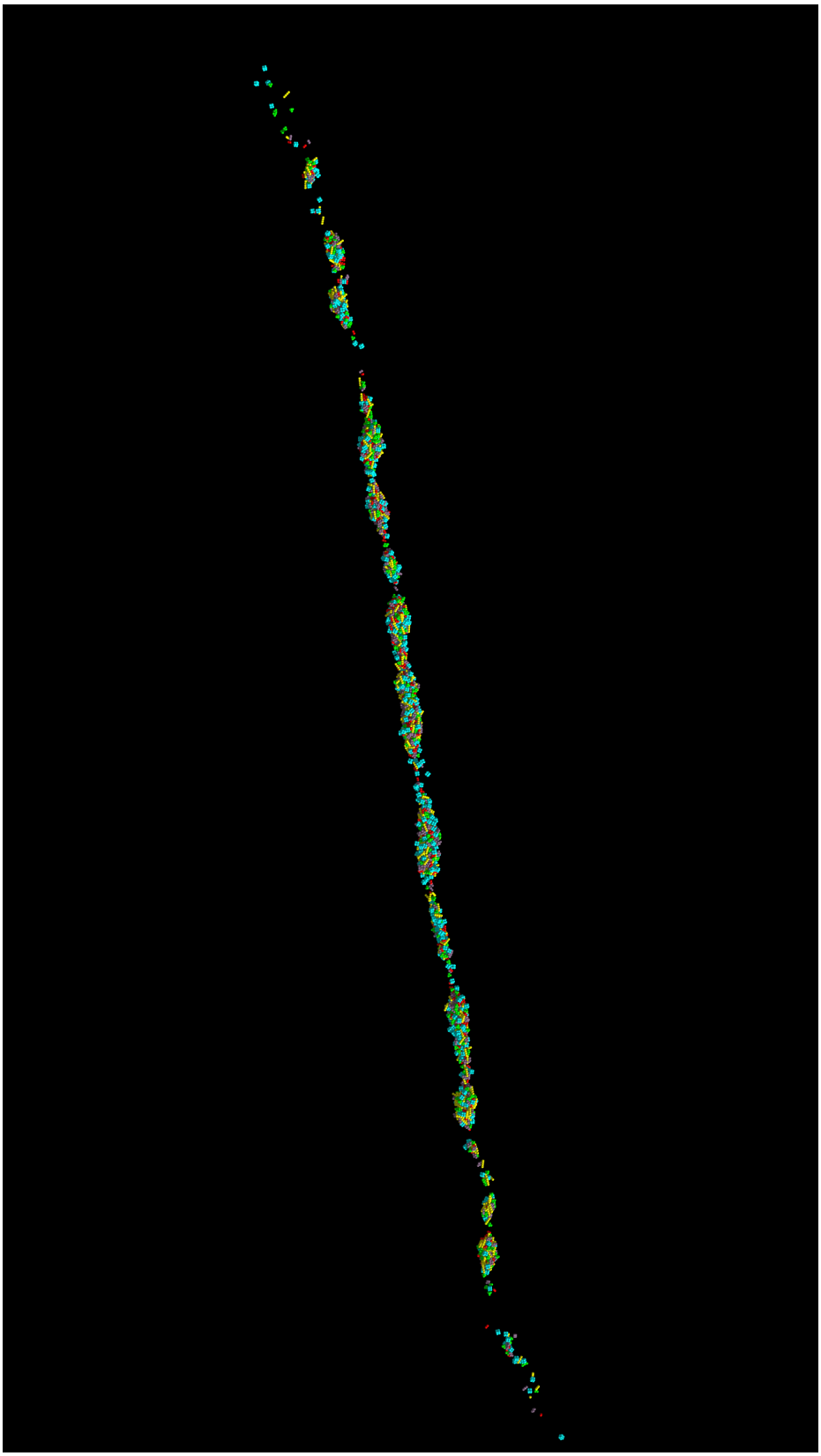


Figure 4.2: A still image from a tidal encounter simulation in trial M1 (see Tab. 4.1 for an inventory of disruption trials). The rubble pile is shown mid-disruption. Constituents are mixed aggregates shapes and are color-coded by shape according to the model of Fig. 4.1.

1,890 simulations across all six trials.

4.3.4 Simulation parameters and time step

Our SSDEM model requires users to set several parameters that determine the behavior of the particles in our simulations. We use separate frictional coefficients for sliding, rolling, and twisting motion: μ_S, μ_R, and μ_T along with the normal and tangential restitution coefficients ε_N and ε_T; these are all unitless. The compressive strength of the material is set by the normal and tangential stiffness coefficients k_N and k_T, in units of N/m. As k_T is always set to $\frac{2}{7}k_N$, we only quote values of k_N here. We list the values of each of these parameters for all six trials in Table 4.2. Note that these parameters are left unchanged across all trials, apart from the frictional coefficients. We set the rolling and twisting coefficients to zero for the bonded-aggregate trials since these settings are intended to parameterize particle shape effects in the case of spherical particles. Since the purpose of our aggregates is to model these effects explicitly, it is appropriate to to set these values to zero. Otherwise, the frictional and restitution coefficients are chosen to match standard values used in previous studies, representing properties of rough, gravel-like material like that we might expect to find on rubble-pile bodies in the solar system. Conducting a parameter sweep that varies these parameters is beyond the scope of this work.

In all simulations, we use a time step of 9.0×10^{-2} seconds. The simulation time step is chosen along with the stiffness parameters to ensure that collisions are resolved appropriately and no unphysical particle overlaps occur. We configure each simulation to run for a nominal 2,000,000 steps, which equates to 50 hours of in-simulation time. While the large majority of our simulations reached this endpoint, we ran additional checks on all simulations to ensure that

Label	μ_S	μ_R	μ_T	$k_N\ (N/m)$	ε_N	ε_T
S1	1.0	1.3	1.05	1.54×10^9	0.55	0.55
S2	1.0	1.3	1.05	1.54×10^9	0.55	0.55
M1	1.0	0	0	1.54×10^9	0.55	0.55
M2	1.0	0	0	1.54×10^9	0.55	0.55
R1	1.0	0	0	1.54×10^9	0.55	0.55
R2	1.0	0	0	1.54×10^9	0.55	0.55

Table 4.2: A list of the key simulation parameter values for each rubble pile prototype. Parameters are held constant for all simulations within a given trial.

sufficient time had elapsed to ensure that tidal encounters were complete and that fragment trains, if present, had finished evolving. Our completion check worked as follows: for each simulation, we calculated the time to perigee from each rubble pile's starting position. We first confirm that the system has been integrated for at least 3 times the initial time to perigee. Then, we check to see whether a disruption occurred. If the most massive fragment in the final time step computed retained more than 95% of its original mass, we concluded that no significant disruption occurred, the encounter was over, and the simulation complete. If a disruption did occur, we calculated the system-wide mass fraction of the most massive fragment in the time step 25,000 steps prior to the final output. If this value was within 10% of the final value, we considered the evolution to be complete. We imposed these requirements on all simulations in addition to the 2,000,000 time step nominal endpoint. Thus, all simulations are guaranteed to have reached a steady state with regard to the properties we are interested in here. For more details on how material parameters and time steps are determined in our SSDEM model, readers may consult Schwartz et al. (2012) and Zhang et al. (2017).

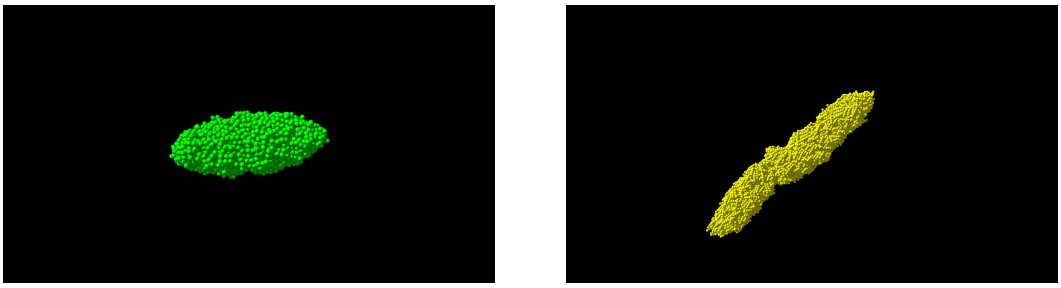


Figure 4.3: The most massive fragments remaining after disruption simulations from trials S1 (left) and R1 (right).

4.3.5 Fragment identification

In the context of tidal disruptions, we are usually interested in outcomes at the level of the post-encounter fragments. In our simulations, a fragment may be composed of hundreds, thousands, or even tens of thousands of constituents (see Fig. 4.3 for examples). It is relatively easy to assign particles or aggregates to distinct fragments by eye. However, given the large number of simulations involved, we took a more systematic approach. Our algorithm proceeds according to the following outline: begin by assigning all of the constituents to their own group. Then, we look for contacts between groups using best-fit ellipsoids. Any groups in contact will be combined. We then repeat the process, gradually merging groups until no more mergers are possible. At this point, all remaining groups containing two or more constituents are considered "fragments."

4.4 Results and discussion

Our goal is to understand how the shape composition of a rubble pile can affect disruption outcomes. We compare the mass and quantity of bound material about the most massive post-

encounter fragment (hereafter also referred to as the "largest fragment" or "primary fragment"), the elongation of the primary fragment, and the number of fragments produced by the encounter. We computed each metric for all ∼1,890 simulations. For each of the six trials, we then produced tiled heat maps of the outcome for each simulation as well as a second plot showing how the trial differs from the corresponding spherical particle trial. Finally, we investigated how material from different portions of the parent body was distributed throughout the fragment train and how particle shape affected this process. To keep the overall number of simulations tractable, rotation was not considered. As mentioned earlier, all rubble piles are given zero initial spin. However, we will note that in general prograde rotation has been shown to amplify tidal effects, while retrograde rotation impedes disruption (Richardson et al., 1998; Holsapple and Michel, 2006).

4.4.1 Disruption regions

The outcomes of the tidal encounters we simulated can be broadly categorized into three groups: no effect, reshaping, and disruption. The reshaping region generally spans a steady gradient from a nearly spherical primary fragment to one that is quite elongated. The disruption region is characterized by more chaotic outcomes that may vary widely between "neighboring" simulations in the q and v_∞ parameter space. The divisions between these categories are not rigidly defined, but we will use these terms to refer to different classes of outcomes when discussing our results. To more easily quantify our results, we define the "disruption region" for a trial using the following procedure. For each value of q, begin with the highest-speed encounter. This represents the gentlest possible encounter for a given q and in all of our simulations results in no mass loss. We then consider encounters with incrementally smaller values of v_∞. In a tile plot,

this is equivalent to beginning at the top of a column and moving downwards towards increasingly intense encounters. Once we find a simulation whose largest fragment experienced mass loss of 15% or more, we stop and use a colored circle to indicate that tile on plots. The colored circle then represents the beginning of the disruption region in that column. All points below are considered to lie within the disruption region for that trial. The 15% figure is somewhat arbitrary and was chosen to capture the first instances of significant mass loss in our parameter sweep. For an example, see Figure 4.4. For encounter geometries within the disruption region, we compare outcomes between corresponding sphere and aggregate compositions and report mean and median differences. See Table 4.3 for a representative example. Table 4.3 will be discussed in more detail in Section 4.4.2.

4.4.2 Mass retention and material captured in orbit

For each simulation, we identify the most massive post-encounter fragment and calculate its mass normalized by the mass of the original rubble pile. Results are shown in Figures 4.4 and 4.5. In the case of relatively mild encounters, this statistic is 1.0, since the original object retained all of its mass. Moving toward slower, closer encounters in the disruption zone, we see progressively smaller values, saturating in the 15–30% range for encounters with the closest and slowest approaches. To quantify the observed effects of particle shape, we consider encounters that fall within the disruption zone for each trial. For each simulation in this region, we compare the mass fraction of the largest fragment to the same figure for the corresponding sphere-based simulation. We report the mean and median fractional differences in Table 4.3. In all bonded aggregate trials, primary fragments tend to retain a larger fraction of the initial mass than com-

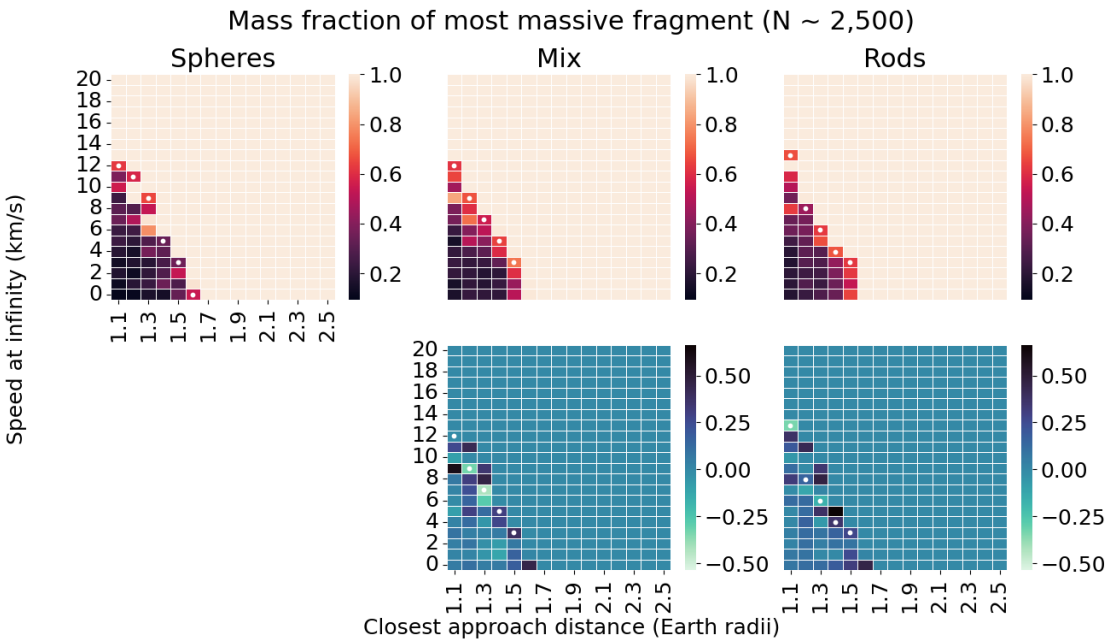


Figure 4.4: The top row shows heat maps of the primary fragment mass fraction for trials S1, M1, and R1 from left to right. The bottom two panels show the discrepancy in mass fraction between M1 and S1 and R1 and S1, respectively. Positive numbers indicate a larger primary fragment mass fraction than in the spherical case. The white circles represent the beginning of the "disruption region." See Sec. 4.4.1 for a more detailed explanation.

parable sphere-based trials when disrupted. This effect is most pronounced at the margins of the disruption region, but the discrepancy persists even for the slowest and closest encounters. The effect is substantially weaker in the higher-resolution trials, and much stronger in rods than in mixed shapes.

We also measure the quantity of material in orbit about the largest remaining fragment. For

Trial	Composition	N Constituents	Mean change in fragment mass	Median change in fragment mass
M1	Mix	2,280	+35.8%	+23.8%
M2	Mix	10,020	+8.0%	+1.3%
R1	Rods	2,505	+48.3%	+51.8%
R2	Rods	10,020	+28.7%	+23.5%

Table 4.3: Fractional changes in largest fragment mass fraction between spherical and aggregate compositions. This table only includes results from runs in the disruption region for each trial (see Sec. 4.4.1). Spherical trials S1 and S2 are omitted since they would be compared to themselves and thus show no change.

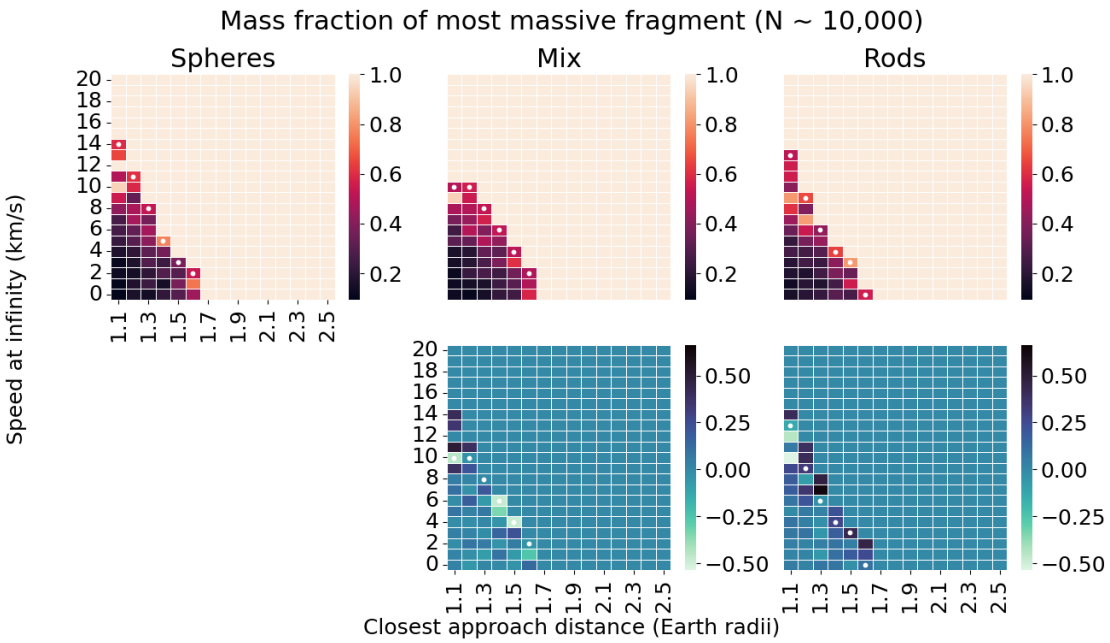


Figure 4.5: Heatmaps in the style of Fig. 4.4, but for trials S2, M2, and R2.

a given simulation, we identify all particles or aggregates in the final time step as belonging to fragments using the method described in Section 4.3.5, and identify the largest of these. We determine whether each fragment in turn is bound to the largest fragment using a simple two-body center-of-mass calculation. The result will not be perfectly accurate, but we find that given sufficient evolution time, large, unbound fragments will drift apart and the two-body calculations give reasonable results. The orbiting mass fraction is then the sum of the masses of those fragments that are gravitationally bound to the primary fragment, normalized by the total progenitor mass. The mass of the largest fragment is not included in this figure. Results are shown in Figures 4.6 and 4.7. The phenomenon of matter being lifted off of the primary object but remaining bound in orbit is mostly confined to the boundary between the reshaping and disruption regions. The most intense tidal interactions leave little or no orbiting material in all cases. Average differences in outcome when comparing spheres and aggregates are negligible—less than $\sim$2% in all cases.

The granular dynamics implementation in `pkdgrav` has been validated against various

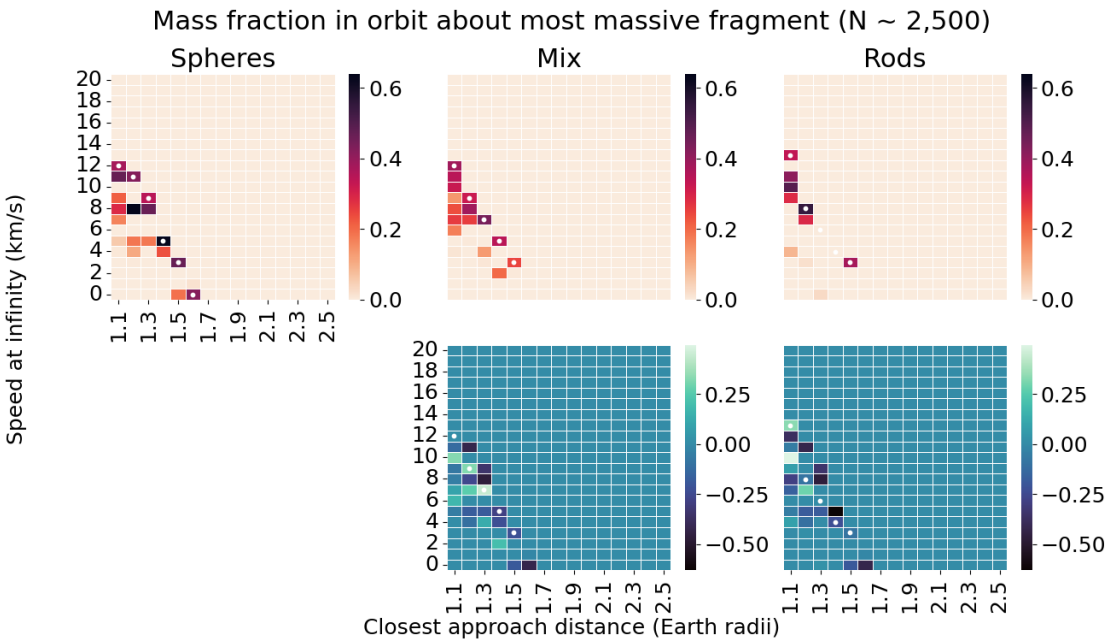


Figure 4.6: Heat maps of mass fraction in orbit about the primary fragment for the lower-resolution trials S1, M1, and R1.

laboratory experiments (Schwartz et al., 2013, 2014). As a further check, we compared one of our simulations to the results of Zhang and Michel (2020), who used `pkdgrav` without bonded aggregates to conduct a study similar in concept to our own. With $v_\infty = 0$ we find that mass loss begins between close approach distances of 1.6 and 1.7 $R_\oplus$ for spherical constituents. For the same material parameters and encounter speed, Zhang and Michel (2020) also find that disruption begins at a close approach distance of approximately 1.6 $R_\oplus$, although the magnitude of the mass loss at that point seems to be somewhat greater. This is perhaps not surprising, since that study uses progenitor bodies with a direct ("prograde") spin period of 4.3 hours that facilitates mass loss, while our progenitors have no initial spin. Unlike their study however, our primary interest is in comparing relative differences in outcomes between differently shaped constituents.

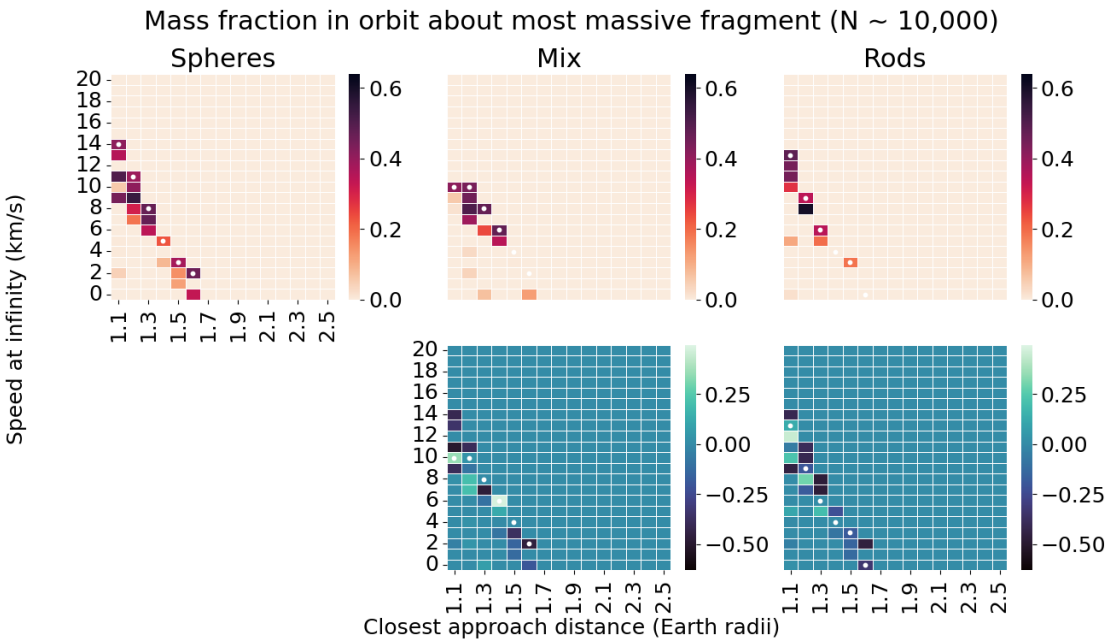


Figure 4.7: Heat maps of mass fraction in orbit about the primary fragment for the higher-resolution trials S2, M2, and R2.

4.4.3 Fragment elongation

It is plausible that the non-spherical shapes of the bonded aggregates in our simulations will influence the overall shape of the post-encounter rubble fragments. We consider the fragment elongation η, which we calculate using the following metric:

$$\eta = 1 - \frac{1}{2}\left(\frac{a_3}{a_1} + \frac{a_2}{a_1}\right),$$

where a_1, a_2, and a_3 are the fragment's long, middle, and short semi-major axes, respectively. For each simulation, we compute this statistic for the primary post-encounter fragment. Overall results are plotted in Figures 4.8 and 4.9. We find that primary fragment elongation varies only weakly across different rubble-pile compositions, at least within the disruption region. Refer to Table 4.4 for details. M1 shows moderately increased elongation compared to S1, but M2 average

elongations are slightly lower than S2. R1 elongations are nearly the same as those in S1 overall, while R2 elongations are somewhat lower than S2 elongations.

Notably, it seems that the most significant discrepancies between spheres and aggregates lie outside of the disruption region. For encounters with large q value (outside of the disruption zone), we find that bonded aggregate composition produces larger elongations than in the spherical case. This trend is actually stronger in the higher-resolution trials, and can be seen most clearly in Figure 4.9. In the cases of M2 and R2, we see a peak in the discrepancy around $q = 1.9$, where the elongation discrepancy relative to spherical composition approaches 0.2. Since we expect non-spherical shapes to inhibit bulk motion, this result runs contrary to our expectations. If we move toward the more extreme encounters just above the disruption zone cutoff, we find the opposite relation. In this region, stronger tidal forces result in much larger final elongations in all cases, but trials M1, M2, R1, and R2 all produce consistently lower elongation than their spherical counterparts. How should we interpret these results? The latter trend seems easier to explain. In the case of a relatively close, slow encounter, a body of non-spherical constituents is better able to resist large-scale reshaping as the particles interlock and limit bulk motion. Especially in the higher-resolution trials, we can see that the observed disruption limit threshold is pushed to somewhat lower v_∞ values. This is consistent with our results in Section 4.4.2—rubble piles composed of bonded aggregates are more "reluctant" to shed mass on the margin and therefore need a closer or slower encounter to meet the 15% mass loss criterion. Elongation seems to work similarly, at least during encounters near to the disruption region; the overall trend in responses is the same, but to see the same outcome the aggregate rubble piles require a slower, closer encounter.

The countervailing trend outside of the disruption region is more difficult to understand,

but we can speculate. The small impulse provided by the tidal forces in these cases is clearly enough to produce some response from the rubble piles, but not enough to completely reshape the objects. It may be that in this regime the spheres are better able to return to their initial positions after a slight disturbance, while the aggregates tend to "lock in" their new shape more effectively. If so, this could be a mechanism to achieve high elongation in small bodies with repeated planetary flybys. Even fairly gentle encounters if repeated enough times could "ratchet up" elongation and gradually change an object's shape.

Another possibility is that there is simply a large sensitivity to initial conditions in this regime—a more elongated rubble-pile prototype may be more susceptible to the tidal forces and in turn receives a greater increase in elongation from the encounter. The weak tidal forces here produce final elongations that are still close in magnitude to the initial, pre-encounter values. Perhaps differences in the initial shapes of the rubble pile prototypes are enough to explain this result, while the larger forces in and near the disruption region overwhelm discrepancies in the initial rubble-pile shapes. Conversely, this tendency is evident in all four aggregate-based trials. If this were the result of random variation in initial conditions, we would not expect to see such a consistent trend across different trials. Perhaps the higher initial elongations of all four aggregate trials when compared with S1 and S2 are producing such an effect, which could explain this consistency. However, initial elongations are actually larger in both M1 and R1 than they are in M2 and R2. If excessive initial elongation was responsible, we would consequently expect that M1 and R1 would show a significantly larger discrepancy from their spherical counterparts. In fact, we see the opposite. M2 and R2 are substantially more discrepant from the spherical cases around $q = 1.9$ than are M1 and R1. These caveats don't rule out other systematic factors, but do reduce the likelihood that this trend is solely the result of initial body shapes.

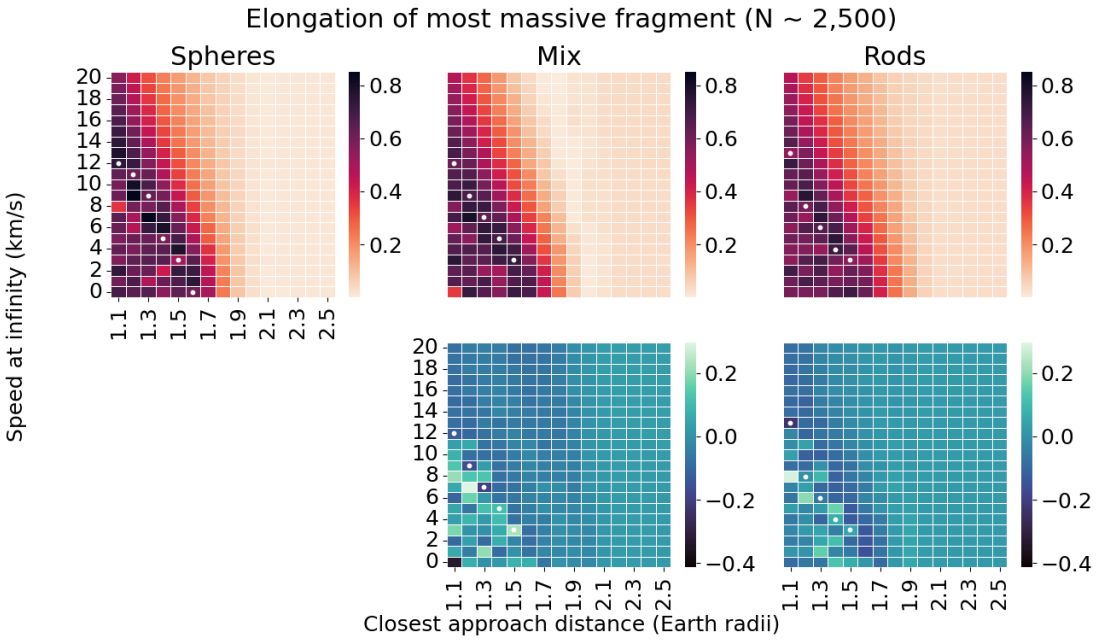


Figure 4.8: Heat maps of primary fragment elongation for trials S1, M1, and R1.

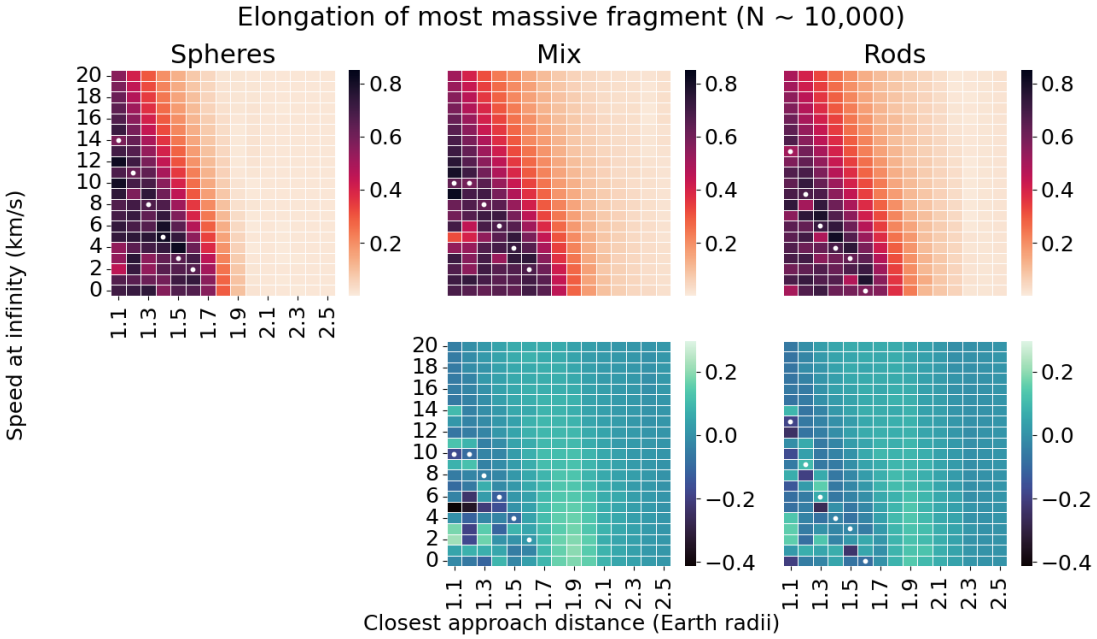


Figure 4.9: Heat maps of primary fragment elongation for trials S2, M2, and R2.

Trial	Composition	N Constituents	Mean change in fragment elongation	Median change in fragment elongation
M1	Mix	2,280	+7.4%	+5.4%
M2	Mix	10,020	-3.6%	-4.5%
R1	Rods	2,505	+2.1%	-2.0%
R2	Rods	10,020	-4.0%	-5.4%

Table 4.4: Fractional changes in elongation of the primary fragment between spherical and aggregate compositions. This table only includes results from runs in the disruption region for each trial.

4.4.4 Fragment quantity

We count post-encounter fragments in all trials using the same method described in Section 4.3.5. Our analysis here requires a "fragment" to contain two or more distinct constituents. That is, at least two spheres in trial S1 and S2 or at least two bonded aggregates in the other four trials. To account for the possibility of more subtle differences in fragmentation patterns, we compared differences in the *total* number of fragments as well as differences in the number of fragments holding 5% or more of the original rubble pile's mass. Results for the first analysis are shown in Figures 4.10 and 4.11 and Table 4.5. Data from the second analysis may be found in Figures 4.12 and 4.13 and Table 4.6.

Grain shape appears to have a significant effect on the number of fragments produced by tidal encounters. The disruption by its definition includes nearly all of the simulations that result in fragment production. Of the aggregate-based trials, M1 stands out as being the only one that tends to produce more fragments in the disruption region. M2, R1, and R2 all show a strong deficit in fragments. The trend is even more pronounced at higher particle resolutions and appears significantly stronger with rod-shaped constituents than with a mix of shapes. Even in the case of M1, if we restrict ourselves to the strongest encounters (corresponding to the extreme bottom left corner of the heatmaps in Figure 4.10) we find a trend of decreased fragment production when

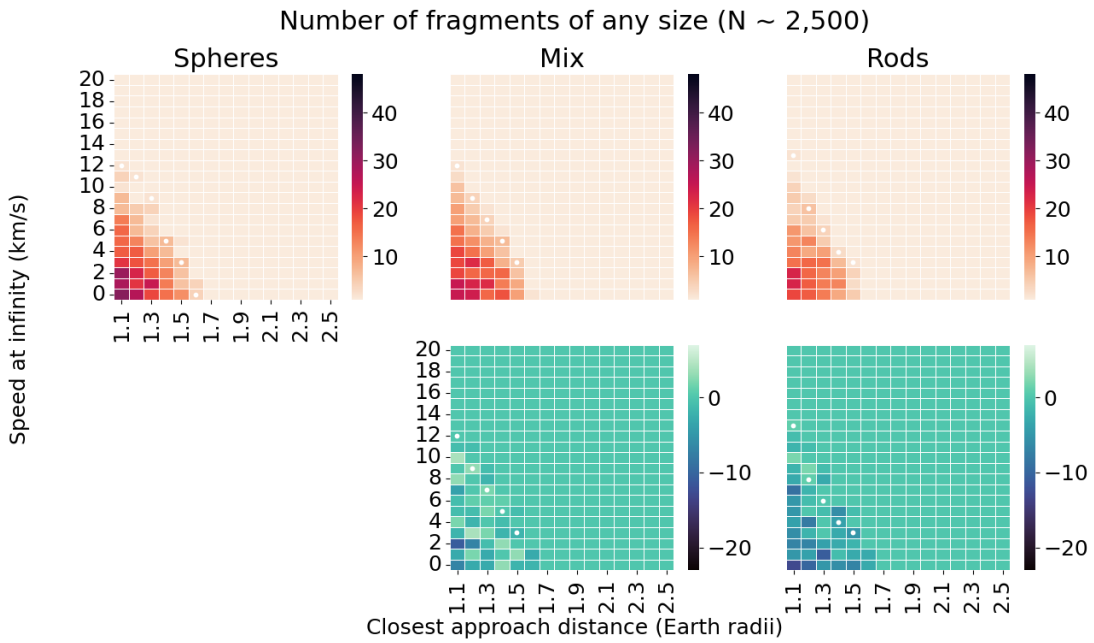


Figure 4.10: Heat maps of total fragment production for trials S1, M1, and R1. We include any agglomeration of two or more constituents in our count.

Trial	Composition	*N* Constituents	Mean change in fragment number	Median difference in fragment number
M1	Mix	2,280	+13.3%	0.0%
M2	Mix	10,020	-15.8%	-23.7%
R1	Rods	2,505	-22.6%	-28.6%
R2	Rods	10,020	-49.5%	-53.6%

Table 4.5: Fractional changes in the number of post encounter fragments for encounters inside the disruption region.

compared with trial S1.

The overall picture is similar when we set a lower limit on the mass of the fragments we consider. Figures 4.12 and 4.13 and Table 4.6 show the results of the same analysis as in the preceding section, but now limited to fragments whose mass is at least 5% that of the original body. Of course, the overall number of fragments reported is smaller in all cases, but we again find a strong tendency for bonded aggregates to reduce the number of fragments relative to a comparable spherical composition. Fragment quantity is lower on average in all four aggregate trials, but especially in R1 and R2. However, the effect of resolution here appears to be opposite

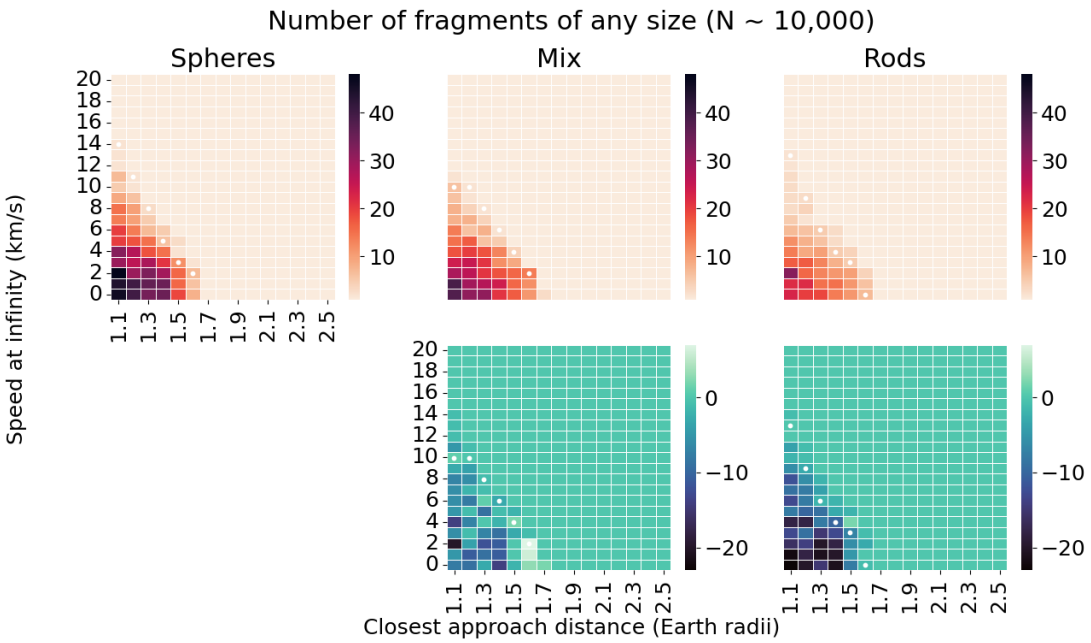


Figure 4.11: Heat maps of total fragment production for trials S2, M2, and R2. We include any agglomeration of two or more constituents in our count.

to that of the previous analysis: M2 and R2 show decreased shape effects when compared with M1 and R1. The difference between the R2 trial in Figures 4.11 and 4.13 is especially interesting. When we consider all fragments, R2 exhibits strong deficits in almost all disruption region simulations compared to S2. This trend is much weaker when we set the 5% mass threshold. This indicates that rubble piles composed of spheres are not just producing more fragments overall, but that they are disproportionately producing lots of very small fragments. We already saw in Section 4.4.2 that rod-shaped constituents are better than spherical constituents at preventing overall mass loss from the main body—these results suggest a possible shape-driven difference in the size distribution of fragments as well.

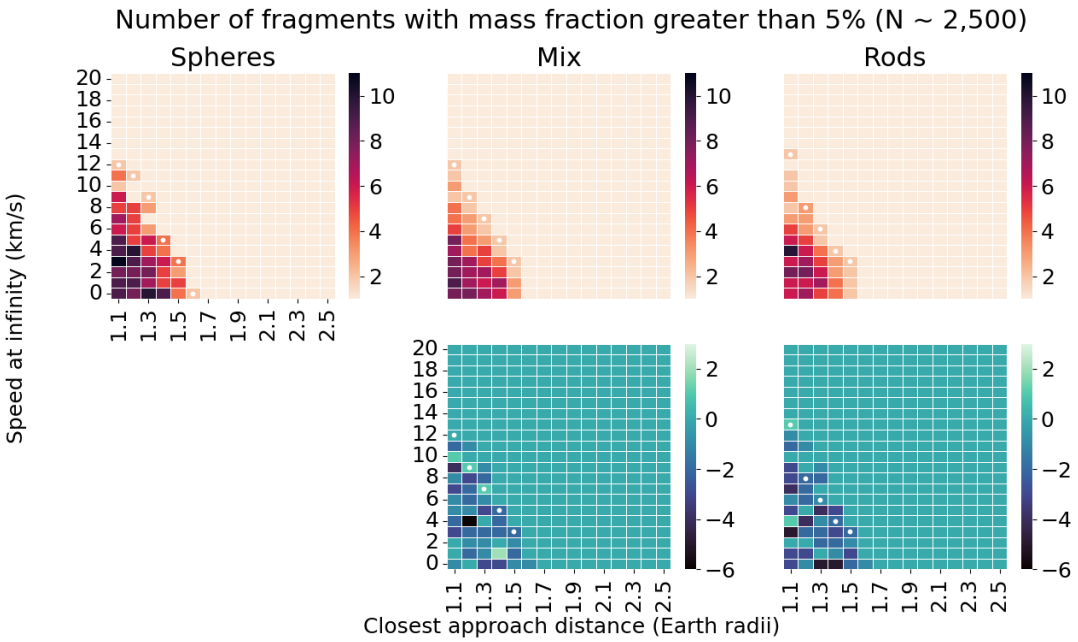


Figure 4.12: Heat maps of fragment production for trials S1, M1, and R1, including only fragments with a mass fraction of 5% or greater.

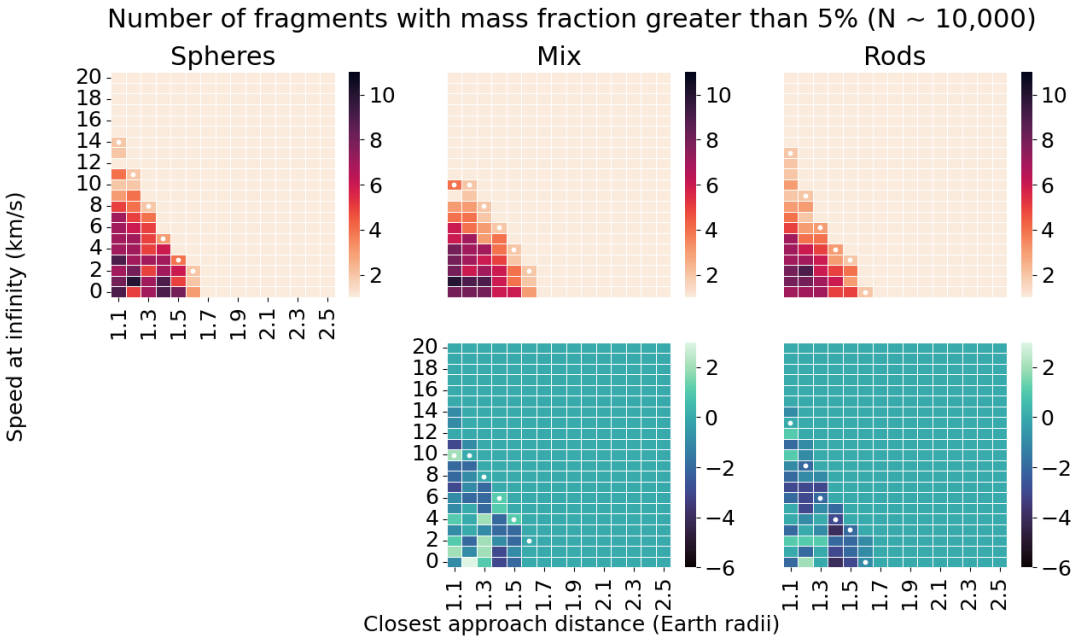


Figure 4.13: Heat maps of fragment production for trials S2, M2, and R2, including only fragments with a mass fraction of 5% or greater.

Trial	Composition	N Constituents	Mean change in fragment number	Median change in fragment number
M1	Mix	2,280	-18.0%	-22.2%
M2	Mix	10,020	-4.6%	-12.7%
R1	Rods	2,505	-30.2%	-33.3%
R2	Rods	10,020	-18.2%	-25%

Table 4.6: Fractional changes in the number of post encounter fragments for encounters inside the disruption region. We include only fragments with mass fractions of at least 5%.

4.4.5 Material excavation and partitioning

Finally, we sought to investigate the effects of particle shape on the distribution of interior and exterior material during tidal disruptions. It has been suggested that rubble-pile asteroids can undergo resurfacing due to tidal stresses during relatively gentle planetary encounters (Binzel et al., 2010; Kim et al., 2023). In this section, we consider the exposure and distribution of interior material during full-fledged disruption events in which the initial body is fragmented, and the degree to which constituent shape affects this process. We take a different approach here than in the rest of this section. This is largely due to the difficulty of tracking interior material from the initial object through such a large number of different disruption simulations. Instead of attempting to summarize each simulation with a single number, we looked at the outcome of one disruption from each trial in greater detail. Specifically, we picked the encounters with $q = 1.1$ and v_∞ = 0, as illustrative examples of significant disruption events. We assign every constituent in the simulation to one of three groups in accordance with its position in the progenitor rubble pile. Grains in the innermost third of the body by radius are called "interior" material. We define "middle" and "surface" material similarly. We then examine the full set of fragments after the disruption and determine the interior, middle, and surface material content of each fragment. Due to our definitions, each simulation will contain more surface material than middle material, and more middle material than interior material. The specific fractions vary somewhat across our

rubble-pile prototypes due to the inherently different packing fractions of difference constituents. To avoid this complication, we compare only the relative concentrations of the three material classes across fragments. In other words, we are not concerned with the absolute mass fraction of interior material in a fragment, only whether the fragment is enriched or depleted in interior material compared with the prototype rubble pile it came from. We then plot these relative mass fractions as a function of distance from the center of the fragment train.

We find no clear differences in outcome across constituent shapes. We include a plot of the results for the M1 trial (see Fig. 4.14), but the rest are omitted as they are generally similar and do not contribute to overall understanding of the outcomes. We emphasize that we have only considered one particular encounter geometry. It may be that other portions of the parameter space or more detailed analyses would reveal a meaningful difference in results. However, this goes beyond the scope of the present work.

With that said, we do find an interesting trend outside the question of particle shape that is of general interest. From Figure 4.14, we can see that there is a strong tendency for material from different depths in the progenitor body to segregate spatially along the fragment train. Many fragments close to the center of the string end up highly enriched in material from the interior of the initial rubble pile. Most of the six simulations we analyzed produced at least one smaller fragment with an even higher concentration of material originating in the inner third of the rubble pile. Figure 4.14 includes an example of this. Fragments near the tails, which are typically small compared to the central fragments, end up almost exclusively composed of surface material. Middle material follows an intermediate trend but is also almost totally absent from the tails. The image in Figure 4.15 is a snapshot of a disruption in progress, after the fragments have largely taken on their final structure but before they have fully drifted apart from one another. The central

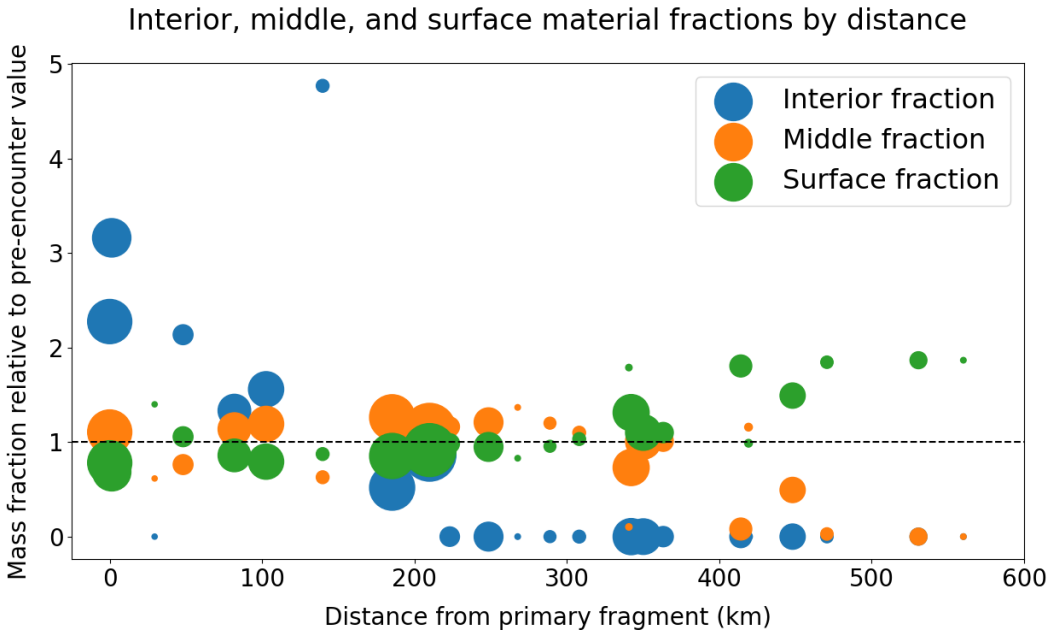


Figure 4.14: A plot of the interior, middle, and surface material fractions of disruption fragments as a function of distance from the center of the fragment train. We describe how these figures are calculated in Sec. 4.4.5. Points are scaled by the overall mass fraction of the fragment. Results are shown from the $q = 1.1$ $v_\infty = 0$ case in trial M1.

fragment seems to have taken on a form like that of a series of coaxial cylinders. The layered, cylindrical structure is then severed on both ends as the neighboring fragments on either side are pulled away, leaving a greater concentration of interior material exposed on each end of the fragment. This effect is clearly visible in Figure 4.16. Due to the intensity of the disruption, the middle and interior layers are frequently exposed, while the former surface material now comprises only a patchy outer coating. A good deal of material from the innermost third of the progenitor appears to be exposed to the surface. When we observe the products of disruption events in the solar system, we thus might expect to find that their surfaces have not been uniformly weathered. Instead, we may see a patchwork of more- and less-weathered sections, especially on the ends, corresponding to material that was already on the surface of the rubble pile and that which was exposed from below.

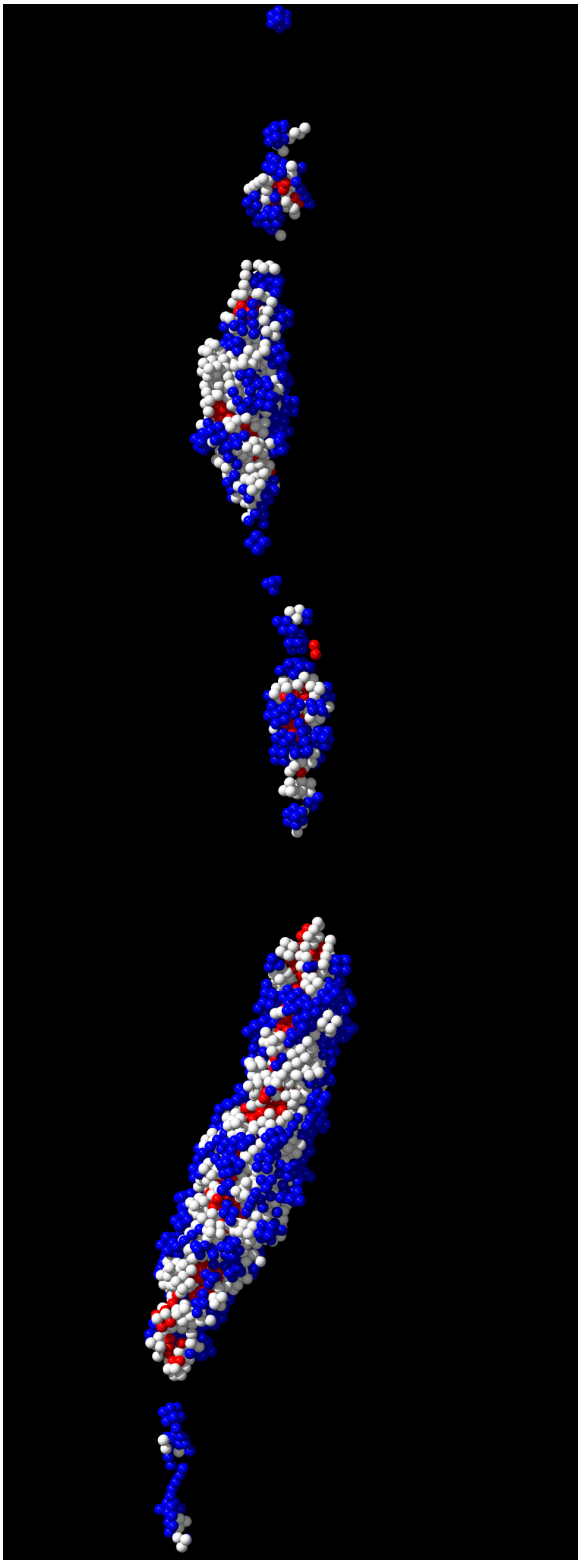


Figure 4.15: A snapshot of a tidal disruption from trial M1. Constituents are color-coded by their depth in the pre-disruption rubble pile. Red material comes from the innermost third by radius, white material from the middle third, and blue material from the outermost third.

Figure 4.16: A close-up snapshot of the primary fragment from the disruption shown in Fig. 4.15, seen from two different angles. Exposed middle and interior material is visible all along the sides of the fragment, and interior material is especially visible on the ends.

4.5 Further discussion and conclusion

Before concluding, we share some final thoughts on the question of particle shape in tidal encounters. Work in the field of granular dynamics has consistently shown that particle shape has an impact on the behavior of granular media. Generally, non-spherical particles impede the ability of a medium to flow, forcing the medium to reshape or dilate to allow the more angular particles to move past one another. In other words, non-spherical particles also tend to add shear strength to a medium. In this light, we expected to find that our bonded aggregates would inhibit disruption and shape change under tidal forces. Overall, it appears that this effect is real and in some cases quite pronounced. Particle shape has a significant impact on mass loss and the fragmentation process during tidal disruption.

Even so, the role of constituent shape turned out to be more complex and context-dependent than we had anticipated. The interplay between shape and fragment elongation in particular proved to be less than straightforward, with seemingly opposite effects in different regimes. And while we do find that particle shape influences fragment elongation, it is quite possible that our elongation metric cannot fully capture the ways in which particle shape bears on fragment shape.

Further, we find that the issue of particle resolution is sometimes difficult to disentangle from particle shape. For example, it has a clear impact on mass loss in the disruption region. Our lower-resolution trials show a much stronger shape effect than the higher-resolution trials, though both are significant. On the other hand, increased particle resolution seems only to increase the effect of particle shape on the quantity of fragments produced by a disruption. These two examples apply to both mixed aggregates and rod-shaped aggregates and so are not easily explained by systematic issues. The nature of resolution itself is somewhat nebulous. Clearly, the models presented here do not capture the number of constituent pieces present in a real-world rubble pile. On the other hand, resolution in DEM simulations is clearly not a knob that can simply be increased asymptotically toward a "correct" value. Size distribution likely interacts with particle shape in ways that are not fully understood and is unfortunately out of the scope of this work. Future work to determine where, if at all, increased resolution no longer has any effect in this context would be of great use.

We find that the shape of a rubble pile's constituents has a meaningful impact on tidal encounter simulation outcomes. This is especially true when considering rates of mass loss and fragment production, but fragment shape is also affected. While many interesting avenues remain open for future study, the importance of shape is clear. In the context of numerical approaches to tidal encounters, particle shape should be included when possible. At the very least, practitioners should be aware of its potential effects.

Chapter 5: Conclusions

5.1 Future work

5.1.1 Potential advances in simulation realism and resolution

Chapter 3 discusses our improvements to `pkdgrav`'s capabilities and efficiency. We see substantial room for further advances. One example is our implementation of particle shape using glued-sphere bonded aggregates. We chose this route because it provided the enhanced realism of particle shape while still leveraging `pkdgrav`'s efficiency in computing sphere-to-sphere contacts and interparticle gravity. While they provide significant advantages over spheres, bonded aggregates carry clear drawbacks. To maintain tractable simulation times, we require that the number of spheres per aggregate remain small. This in turn produces constituents with bulbous, rounded shapes and divots (see Fig. 3.1). We can avoid this by building bonded aggregates containing many smaller spheres, but this requires us to use extremely small time steps to resolve overlaps between massive aggregates of tens to hundreds of particles and their tiny constituent spheres. If we fail to properly account for this, we risk excessive particle overlaps and unphysical behavior. This is an unavoidable consequence of using glued-sphere bonded aggregates with soft-sphere contact physics. This could potentially be avoided if we used a different model entirely to implement non-spherical particles and calculate particle contacts and overlaps. We have

begun to investigate the merits of alternative approaches, including level sets, Clifford algebras, and others. These efforts are still in an early, exploratory stage. Finally, the fundamental code structure of `pkdgrav` limits the degree to which we can parallelize aggregate operations. This cannot be avoided without making deep changes to `pkdgrav`'s structure that go far beyond the substantial modifications that we have already made. All of these impediments point to the ultimate need to build a new code to serve as as successor to `pkdgrav`. Luckily, we need not start from the ground up. The fields of granular dynamics, galactic dyanmics, and cosmology offer alternatives that could be leveraged to accomplish this.

`pkdgrav` has been applied productively to a wide variety of planetary science problems in its 20-odd years and remains at the cutting edge of N-body DEM codes in many respects. However, there are now granular dynamics codes that can simulate very large numbers of non-spherical particles, some with soft-sphere contact physics (see Section 3.2), though not including self-gravity. At the same time, pure N-body codes used in galactic dynamics and cosmology simulations have continued to improve. More sophisticated gravity models, more efficient parallelization, and the adoption of GPUs have made N-body simulations on very large scales possible (Jetley et al., 2008; Potter and Stadel, 2016). As just one example, Potter et al. (2017) successfully performed a cosmological simulation with two *trillion* particles in just 80 hours using over 4,000 GPU nodes. Of course, there are limitations and difficulties inherent in bringing the two approaches together. Such a combination likely could not achieve the resolution of either, even if implemented well. Even so, a careful marriage of advances in both of these realms could yield a next-generation N-body DEM code with capabilities far exceeding those used in planetary science today. If realized, a code taking advantage of these features could potentially open the door to modeling entire rubble-pile bodies with constituents approaching realistic sizes. This would

clearly be of great benefit to the field.

5.1.2 Arrokoth

More examples of small KBOs are needed to understand the range of diversity in shapes and formation pathways in the Kuiper belt. Direct observations of more Arrokoth-like bodies would almost certainly yield interesting results, although this is not likely to happen at any point in the foreseeable future. Observations of the Kuiper belt have revealed a large population of widely separated binaries (Noll et al., 2020) and models of KBO formation (Nesvorný et al., 2010) via gravitational collapse similarly produce large quantities of binaries and higher-multiple systems. If Arrokokth is indeed a representative classical, cold KBO, we may be seeing two ends of a spectrum of possible outcomes. McKinnon et al. (2020) speculate on possible mechanisms for bringing these binaries together into close mutual orbits that could then produce something like Arrokoth. While their results are not definitive, they find that gas drag is the most likely culprit. Perhaps other, similar KBOs are trilobate or 4-lobate bodies, rather than being "snowmen." Recent work by Stern et al. (2023) has already pushed numerical study of Arrokoth further by modeling possible formation scenarios for the lobes themselves. As simulation codes become more capable and computational resources more abundant, it could be fruitful to attempt to bridge the gap between the large-scale gravitational collapse simulations of Nesvorný et al. (2010) and the small-scale simulations presented here and by Stern et al. (2023). If we could follow the process from the initial cloud all the way to resolved KBOs with reasonable resolution, we could more easily validate our models against Arrokoth itself as well as make more detailed predictions about the KBO population more generally.

5.1.3 Constituent shape effects

The work described in Chapter 4 points to a number of interesting new avenues of investigation. We have attempted to assemble a large-scale picture of overall trends regarding the influence of particle shape for this particular application, but more detailed work is required in many places to understand exactly how these effects are mediated within rubble piles. A more focused look at the boundary between the disruption and reshaping regions described in Chapter 4 is needed. Due to the initial range of q and v_∞ values we chose, a large portion of our parameter space is effectively wasted for many of the metrics we consider. This gives us relatively low "resolution" (in terms of our parameter space, rather than particle number) than is ideal in some interesting regions. An expanded look at edge-case outcomes around the transition between disruption and deformation could shed more light on exactly how particle shape is influencing these processes.

A more detailed look at the relationship between particle shape and post-encounter fragment shape seems necessary. We use elongation as a proxy for shape, but this is only one possible approach and was chosen in part for its relative simplicity. Elongation seems to be a reasonable metric for comparing broadly similar ellipsoidal fragments, but breaks down for more exotic shapes. We did not include a comprehensive image catalog of final fragment shapes because it would have been much too large. However, at a glance, we see hints that rubble piles with spherical composition may be more likely to produce fragments with contorted or "bent" shapes. If this effect is real, it could be that bonded aggregates are better than spheres at maintaining rigidity in rubble piles. Highly elongated fragments composed of spheres would then be disproportionately likely to bend and fold in on themselves. To our knowledge, this has not been studied directly,

but it would be in keeping with what we know about particle shape and granular media. This could have implications for the elongated bodies produced by strong planetary tidal encounters. The asteroid 1620 Geographos is a possible example.

Our method of simulating non-spherical particles has applications beyond tidal disruption. We had originally planned to include a study of particle shape effects in rubble pile spin-up processes as an additional chapter in this dissertation. We described preliminary results in Chapter 3, but a large-scale study is warranted and would be a natural next step. Many of the tools developed for the tidal disruption work (see Appendix B) could easily be leveraged to carry out and analyze such a study and in fact were developed with this application in mind.

5.1.4 Rubble pile interiors

Beyond extending the specific studies described in this dissertation, we can speculate on how the study of small rubble-pile bodies may be advanced in the future. One of the most significant missing pieces in our overall understanding of these objects is a more detailed model of interior structures. The current general picture of low-strength, gravitational aggregates seems to be correct, but the reality is probably more complicated. Rubble piles, even within a given class of bodies (e.g., asteroids), likely span a continuum. At one end, we may have large bodies that are shattered but have not been dispersed and thus have relatively low porosity. At the other, we may find highly porous objects that have been heavily processed by collisions. Even in the wake of a single catastrophic collision, some rubble piles may coalesce with large boulders at their centers or even at the surface, while others end up with only small fragments throughout. These different structures will probably behave differently under mechanical stress, despite all being

classified as "rubble piles." The *Rosetta* spacecraft was intended to use radar to study the interior of comet 67P/Churyumov-Gerasimenko in conjunction with the *Philae* lander. Unfortunately the lander was lost prematurely during deployment and the experiment was never carried out. To date, we still only have indirect knowledge of rubble-pile interiors via density, inferred porosity measurements, and surface impact experiments. This situation may improve in the near future. The *Hera* mission will be launched in late 2024 and will rendezvous with the Didymos binary asteroid system in 2026. *Hera* will carry with it the Juventus Radar (JuRa), a CubeSat that will perform the first direct observations of an asteroid interior. In addition to other measurements, JuRa will survey the interiors of Didymos and its satellite Dimorphos (Michel et al., 2022). Of course, *Hera* cannot answer all of our questions about asteroid interiors, to say nothing of comets and Kuiper belt objects. Even so, these potential radar observations represent an exciting first step into a new dimension of empirical constraints for our models. Assumptions about internal structure underlie all of the simulations presented in this dissertation, and new observations will help us to further refine our simulations and better ground them in empirical data.

5.2 This dissertation

In Chapter 2, we examined the origins of the Kuiper belt contact binary 486958 Arrokoth. Given the unique opportunity posed by the *New Horizons* flyby, this is likely the first numerical study of a directly observed cold classical Kuiper belt object. Its unusual shape may provide insight into how the primordial small bodies in the more distant reaches of the solar system were formed, possibly as the result of very gentle collisions between cometesimals. This is an origin story different than that of most previously observed small bodies. Within a reasonable range of

material parameters, we find that possible origin scenarios are constrained. Even relatively low-speed impacts can generally be ruled out, and a very gentle merger or collision following a low-separation, synchronous orbit is determined to be the most probable mechanism for producing an object like Arrokoth.

Chapter 3 details our work in expanding the capabilities of our code, `pkdgrav`. It has been established in terrestrial studies of granular dynamics that irregular particle shape can inhibit bulk motion of media and add shear strength. Motivated by this work, we add the capability to model non-spherical grains with soft-sphere contact physics and test their effect on simulation outcomes. Although our presentation describes our specific implementation, this method should be applicable to DEM gravity codes more generally. Particle shape appears to have a substantial impact. Proof-of-concept simulations of asteroid spin-up, tidal disruption, and the Brazil nut effect with irregular particles show an increase in effective strength when compared with spherical particles. We find that the scale of the change is meaningful, and that particle shape should be incorporated into numerical work on small bodies when possible.

Finally, in Chapter 4 we apply our work from Chapter 3 to tidal disruption in a dedicated study. We conduct a large suite of simulated encounters between the Earth and small rubble-pile bodies, varying both the geometry of the encounters and rubble-pile-constituent shapes. We find that particle shape does indeed matter in tidal disruptions, though its effects can be quite varied depending on the context. In some instances (e.g., the post-encounter mass of satellites of the largest remnant), particle shape has no measurable effect. On the other hand, we find that the mass fraction and number of fragments produced after violent tidal encounters can vary up to 50% between cases with spherical and non-spherical constituents, though the magnitude of this effect is dependent on the exact particle shape and resolution.

In this work, we have considered how small bodies in the solar system may evolve as a result of low-energy collisions, deformations, and disruptions. We have applied computational simulation techniques to these processes in a number of different contexts. We have discussed how our improved model can account for particle shape when making predictions of the dynamical evolution of small bodies, especially in the case of tidal disruption. Simulations are an indispensable tool and will only become more capable as available computing power continues to grow and modeling techniques become more sophisticated. At the same time, Earth-based and space-based observations remain central to the study of small bodies. Laboratory experiments and in-situ data from in-progress and planned spacecraft missions will be essential to developing better constraints for models like `pkdgrav`, which can then better support future investigations of small bodies.

Appendix A: Appendix to Chapter 3

Table A.1: Functions modified by updates in Section 3.4.1, each of which previously contained a brute-force search for aggregate members.

Function Basename	Definition
CountPart	Counts number of particles in an aggregate
GetAccelAndTorque	Returns center-of-mass (COM) acceleration of and torques on an aggregate
GetAxesAndSpin	Builds inertia tensor, returns angular momentum vector relative to COM
GetCOM	Returns COM position, velocity, and total mass of aggregate
SetBodyPos	Transforms positions of aggregate particles to body frame
SetMassDEM	Stores aggregate total mass in constituent particle structures, needed for SSDEM routines
SetSpacePos	Transforms positions of aggregate particles to space frame
SetSpaceSpin	Transforms spins of aggregate particles to space frame
SetSpaceVel	Transforms velocities of aggregate particles to space frame

Appendix B: Documentation of analysis tools

NOTE: This Appendix documents the `pkdgrav` analysis package `pkdtools`. It was originally written with `pkdgrav` users in mind and contains some code-specific references and terminology, as well as notes for the developers and suggestions for future improvement. We have reproduced it here as originally written. Readers who are not familiar with `pkdgrav` may simply ignore these references.

B.1 Preface

`pkdtools` is a Python package thats aids in creating and manipulating `pkdgrav` `ss` data. It does not rely on any existing `pkdgrav` utilities, with the exception of `ssio.py`. `pkdtools` is not interactive in the way that tools like `rpx` or `ssinfo` are. Rather, `pkdtools` is designed to be imported by user-created Python scripts as a general toolkit for working with `ss` files in `pkdgrav`. Ideally, interactive Python-based `pkdgrav` utilities developed in the future will use `pkdtools` as a "back end." This document contains an inventory of the classes, methods, and functions introduced in `pkdtools`, its organization into various subpackages, as well as miscellaneous information about best practices and usage. `pkdtools` is an expansion of the `pkdtools` module and was created by Julian Marohnic (JCM) with important contributions from Joseph DeMartini (JVD).

pkdtools is organized into several core modules that make up the base pkdtools package, as well as several subpackages that contain a variety of optional, specialized, or experimental features. As of this writing, the subpackages are pkdtools.aggs, pkdtools.rp, pkdtools.constants, and pkdtools.exp.

B.2 Core classes, functions, and methods

The basic functionality of pkdtools is split across the following modules: particle, assembly, tools, util, pkdio, and ssio. ssio predates pkdtools and is documented elsewhere. When importing pkdtools, all of the base modules will be imported automatically—the division into various modules is purely for organization and may be ignored by the user. This section describes these core classes, functions, and methods. Additional, more specialized tools are included in various subpackages (e.g., pkdtools.aggs) and are described in later sections of this document.

pkdtools introduces two new classes: Particle and Assembly, hereafter "particles" and "assemblies." Particles in pkdtools correspond to particles in pkdgrav, while assemblies correspond to any collection of zero or more particles. These two classes are the primary data structures used in pkdtools. Particles and assemblies both have "units" attributes. Legal unit settings are 'pkd' (pkdgrav customary units), 'cgs', and 'mks'. Units may be changed at will by the user, and the values contained in the relevant particle or assembly should be updated seamlessly. Changing the units of a particle will only update the data values for that particle, while changing the units of an assembly will update the units attribute of the assembly, as well as

the units and data values for all of its constituent particles. `pkdtools` makes use of a number of internal functions (described below) to effect unit changes, but users should always change particle units by simply setting the units attribute to the desired value.

B.2.1 Particles

Particles in `pkdtools` have attributes corresponding to the standard `pkdgrav` particle fields, and an additional units attribute. The names of the attributes are as follows:

```
iOrder
iOrgIdx
m
R
x
y
z
vx
vy
vz
wx
wy
wz
color
units
```

The `Particle` keyword will create a new particle object. For example:

```
Particle(0,0,10,100,1,0,0,5,0,0,0,0,0,5,'cgs')
```

will create a new particle with iOrder 0, iOrgIdx 0, mass 10 g, radius 100 cm, x position 1 cm, y and z positions 0, x speed 5 cm/s, y and z speeds of 0, no spin, color yellow, and a 'cgs' units tag. The 'cgs' argument is the only way to specify the units of the input arguments, so it is crucial to specify this value. When initializing a new particle, iOrder, iOrgIdx, mass, and radius must be specified. Position, velocity, and spin all default to zero if not specified. Particle color will default to green, and units default to pkdgrav. Particle attributes can be read or written using the `<class instance>.<attribute>` syntax, e.g.:

```
>>> p1 = Particle(0,0,15,50,units='cgs')
>>> p1.m
15.0
>>> p1.m = 20
>>> p1.m
20.0
>>> p1.units = 'mks'
>>> p1.m
0.02
```

Below is a list of the methods available to the `Particle` class in the base `pkdtools` package. In addition, particles have a `print` function implemented, which may be used to display the entire contents of the object.

`pos()`

Return particle position as a `pkdgrav` array.

`vel()`

Return particle velocity as a `numpy` array.

`spin()`

Return particle spin as a `numpy` array.

`set_pos(pos, units=None)`

Set particle position with a 3-element vector-like input. `pos` may be a `numpy` array, a tuple, or a list. Units default to the particle's current units.

`set_vel(vel, units=None)`

Set particle velocity with a 3-element vector-like input. `vel` may be a `numpy` array, a tuple, or a list. Units default to the particle's current units.

`set_spin(w, units=None)`

Set particle velocity with a 3-element vector-like input. `w` may be a `numpy` array, a tuple, or a list. Units default to the particle's current units.

`copy()`

Returns a copy of the particle. Setting a new variable equal to an existing particle will not create a new, independent particle object.

B.2.2 Assemblies

The `Assembly` class is derived from the Python `list` type, and shares many of its features. An assembly has two attributes of its own, which are `units` and `time`. Both `time` and `units` may be specified when initializing an assembly, or omitted in favor of the default values (`pkdgrav` units and a timestamp of 0.0). An assembly is a container for zero or more particles, though a list of particles on its own does *not* constitute an assembly. An assembly can be created with the `Assembly` keyword. `Assembly()` expects to be called on an arbitrary number of particles, along with optional units and time arguments. Note that calling `Assembly` on a list of particles will not work, but this can be circumvented by prepending the * operator to your list when creating the assembly. E.g.:

```
Assembly(particle1, particle2, ..., units='cgs')
```

OR

```
Assembly(*[particle1, particle2, ...], units='cgs')
```

Assemblies support list-style slicing, though there are some subtleties to be noted here. Consider an assembly `a1` containing 10 particles. `a1[3]` will return the 4th particle in the assembly. The

particle will not be removed from a1, but any manipulations made to this particle will be reflected in the 4th particle of a1. Contrast this with the get_particle() assembly method, which will instead return a copy of the particle requested that exists independently of the original. Functions and methods in pkdtools that deal with extracting particles from assemblies typically behave this way. Also, get_particle() searches by iOrder value, while list slicing does not "know" about iOrder values and will simply return the particle in the position requested. Both approaches may be useful in different circumstances. Assemblies are also iterable, so pkdtools allows for constructions like the following:

```
to_be_deleted =
[particle.iOrder for particle in a1 if particle.x > 1000]


OR


for particle in a1:
    if particle.iOrgIdx < 0:
        particle.set_vel((0,0,0))
```

Below is a list of the core Assembly methods that are included in the base pkdtools package. The print function can also be called on assemblies.

N()

Returns the number of particles in the assembly.

`M()`

Returns the total mass of the assembly.

`xbounds()`

Returns a `numpy` array containing the minimum and maximum x values across all particles in the assembly.

`ybounds()`

Returns a `numpy` array containing the minimum and maximum y values across all particles in the assembly.

`zbounds()`

Returns a `numpy` array containing the minimum and maximum z values across all particles in the assembly.

`com()`

Returns a `numpy` array containing the center of mass position of the assembly.

`comv()`

Returns a `numpy` array containing the center of mass velocity of the assembly.

`center()`

Returns a `numpy` array containing the mid-point of the assembly. In other words, the point located halfway between the extreme x, y, and z values. Note that this is not equivalent to the center of mass of the assembly.

`ang_freq()`

Return the angular frequency vector of an assembly.

`freq()`

Return the rotation frequency of an assembly.

`period()`

Return the spin period of an assembly.

`set_com(com, units=None)`

Translate all particles so that the assembly has the specified center of mass position. `com` must be a 3-element `numpy` array, tuple, or list. Units will default to the current assembly setting.

`set_comv(comv, units=None)`

Edit all particles so that the assembly has the new center of mass velocity. `comv` must be a 3-element `numpy` array, tuple, or list. Units will default to the current assembly setting.

`set_center(center=(0,0,0), units=None)`

Translate all particles to match the desired center location (center is described above under `center()`). `comv` must be a 3-element `numpy` array, tuple, or list. Units will default to the current assembly setting.

`R()`

Return the assembly "radius." This method will return a value regardless of whether the assembly is a single rubble pile or not. The radius is defined here to be the greatest possible distance between the center of mass of the assembly and any single particle plus that particle's radius.

`vol()`

Calculate the volume occupied by particles in the assembly. This method will return a value regardless of whether the assembly is a single rubble pile or not. `vol()` uses a simplistic convex hull method and could probably be improved. Clearly, usefulness and accuracy will depend on the inputs.

`avg_dens()`

Return the average particle density over all particles in the assembly.

`bulk_dens()`

Return the bulk density of the assembly. Relies on `vol()`, and so suffers from the same pitfalls.

`I()`

Return the inertia tensor of the assembly as a `numpy` array.

`axes()`

Return the principal axes of the assembly as a `numpy` array.

`L()`

Return the angular momentum vector of the assembly. (Untested)

`deeve()`

Return the DEEVE semi-axes, equivalent radius, and bulk density of the assembly.

`semi_axes()`

Calculate the lengths of the three semi-axes of an assembly.

`show_particles()`

Prints all particles in the assembly. Largely redundant with `print(<assembly>)`.

`add_particles(*particles)`

Add copies of an arbitrary number of particles to the assembly. Future manipulations of the assembly will not affect the original particles that were added. As in the case of assembly, this method takes each particle to be added as an argument, rather than a list of particles. Use the * operator to add a list of particles.

`get_particle(iOrder)`

Returns a copy of the first particle in the assembly with an iOrder matching the value passed in. To edit the actual particle in the assembly, use list slicing. E.g. `<assembly>[0].m = 100`. Currently, this method can only accept a single iOrder value at a time.

`del_particles(*iOrders)`

Deletes all particles with iOrder values matching any in `iOrders` from the assembly. Any list arguments containing the iOrder values must be unpacked with the * operator. **Use caution when combining `del_particles()` with loops. Deleting particles from an assembly while iterating over its particles is equivalent to removing elements of a list while looping through it and can lead to unexpected behavior.**

`copy()`

Returns an independent copy of the assembly.

`sort_iOrder(direction='a')`

Sorts the assembly by iOrder. Optional `direction` argument allows sorting in ascending `'a'` or descending `'d'` order. The default is ascending.

`sort_iOrgIdx(direction='d')`

Sorts the assembly by iOrgIdx. Optional `direction` argument allows sorting in ascending `'a'` or descending `'d'` order. The default is descending.

`condense_iOrder(direction='a')`

Renumber iOrder values consecutively. Optional `direction` argument allows sorting in ascending `'a'` or descending `'d'` order. The default is ascending. There are no guarantees on which particles will get which iOrder value, just that the particles will be unchanged and the iOrder values will be consecutive beginning with zero (or the largest iOrder value if the descending option is chosen).

`condense(direction='a')`

Reassign iOrder and iOrgIdx values so that particles are sequentially ordered in both iOrgIdx and iOrder. The default order is ascending is the default, and `direction` may be set to `'d'` for descending order.

`rotate(axis, angle)`

Rotate the entire assembly by `angle` about `axis`. Both arguments must be non-zero. **Note: currently, the reference point for the rotation is about the origin.** To rotate the assembly in place, relocate the center or center of mass to zero, rotate, and move the assembly back. An option to specify the center of rotation should be added in the future.

B.2.3 Reading and writing `ss` files

`pkdtools` includes dedicated reading and writing functions in the `pkdio` module, which is imported automatically with `pkdtools`. Both functions rely on the existing `pkdgrav` module `ssio` for interfacing between Python and the binary `ss` format. An unmodified copy of `ssio` is currently included in `pkdtools`. Once `pkdtools` is formally included in the `pkdgrav` repository, the `pkdgrav` version of `ssio` may be referenced instead.

`ss_in(filename, units='pkd')`

Read in a `pkdgrav` `ss` file. This function makes use of the `read_SS()` function from `ssio`, but returns a typical `pkdtools` assembly structure. Units may be specified, with the default being `pkdgrav` units. The `filename` argument must be passed in as a string.

`ss_out(assembly, filename)`

Write an assembly to an `ss` file. `ss_out()` uses the `write_SS()` function from `ssio`. When writing an assembly, units will be automatically converted to `pkdgrav` units by `ss_out` before writing. `ss_out()` will warn the user when attempting to write assemblies contain-

ing duplicate or non-sequential iOrder values. Since `ssio` will (reasonably) not respect this input, `ss_out()` will in all cases make a copy of the assembly to be written and call the `sort_iOrder()` method on it before passing it to `write_SS()`. Calling `condense()` on the original assembly should result in a renumbering equivalent to what will eventually be written to an `ss` file by `ssio`, though this has not been exhaustively tested. Apart from the iOrder values and sequence, all particles from the assembly will be reflected faithfully in the written `ss` file.

B.2.4 Miscellaneous

This subsection describes several extra functions that didn't fit neatly elsewhere. All of these functions are included in the `tools` module and are imported automatically with `pkdtools`.

`join(*assemblies, units='pkd')`

Combine an arbitrary number of existing assemblies into one new assembly. Particles in the new assembly are copies of those in the input assemblies. Any manipulations of the new assembly will not affect the originals. `pkdgrav` units are the default.

`subasbly(assembly, *iOrders, units=None)`

Returns a new assembly with only the input iOrders. List arguments must be unpacked using the * operator. By default, units will be preserved from the original assembly.

`viz(assembly, resolution=10)`

Visualize an assembly of particles using `matplotlib`. The `resolution` argument determines how round or "blocky" the particles appear, with higher values taking more time to render. Unfortunately, for $\gtrsim 1000$ particles `viz()` becomes prohibitively slow. This is because `viz()` is essentially making a 3D scatter plot using `matplotlib`, which is not designed for so many inputs and cannot easily handle this number of points. Using a standard scatter plot instead of plotting spheres does not fix this issue. Ultimately, the solution would be to use something like `mayavi` or `plotly` to render the particles, but I compromised here in the interest of accessibility since `matplotlib` is so much more widely available. An additional problem is that `matplotlib` cannot currently set an equal or "square" aspect ratio for 3D plots. As a workaround, `viz` determines the most extreme particle locations in x, y, and z and sets an equal range for all 3 axes to accommodate the worst case scenario. This results in a tolerable aspect ratio for the visualization.

B.2.5 Functions and methods for internal use

`pkdtools` includes a number of functions and methods that are not intended to be called by the end user. In the future, all of these should probably be renamed to match the initial underscore convention used in `pkdgrav`. They are cataloged here for the benefit of anyone making changes to `pkdtools` in the future. Most of these functions are defined in the `util` module.

iOrder_key(particle)

A key function used by the sort_iOrder() method. Returns the particle's iOrder value.

iOrgIdx_key(particle)

A key function used by the sort_iOrgIdx() method. Returns the particle's iOrgIdx value.

color_translate(pkd_color)

A single-use utility for converting pkdgrav numeric color codes to matplotlib colors in viz().

vector_rotate(vector, axis, angle)

A function for to perform a general rotation on a vector. Returns the rotated vector and leaves the original vector unaltered. This implementation was largely copied from ssgen2Agg. This function is used by the rotate() assembly method.

angle_between(v1, v2)

Return the angle between two vectors. Used when setting the orientation of generated regular aggs. There could be some funny issues here relating to the domain of np.arccos which JCM has not looked into very carefully. The current implementation was copied from a Stack Overflow post, which is linked in the pkdtools.py comments.

`makesphere(x, y, z, radius, resolution=10)`

Return the coordinates for plotting a "sphere" centered at `(x, y, z)`, though really it produces a sphere-like polyhedron. Increasing `resolution` increases the number of faces, giving a rounder look but increases time the time to render. Copied from a Stack Overflow post, which is linked in the `pkdtools.py` comments. This function is used by `viz()` for plotting spheres instead of scatter plot markers.

`convert(value='pkd')`

A particle method that changes particle units to `value`. Intended for internal `pkdtools` use. This function gets called when the particle `units` attribute is updated. Users should always change particle units by setting the units attribute to the desired units. `convert` is defined in the `particle` module.

The following six functions are called by the `convert()` particle method to handle unit conversions. Each function scales the input particle's attributes appropriately and returns nothing. These functions in turn make use of a series of constants defined in `pkdtools.py` (sourced from various locations around the internet) that encode the actual conversion factors. **NOTE:** DCR has pointed out that these constants may not match exactly with the constants used in `pkdgrav` or other existing utilies, which will cause small discrepancies to creep in. This should be investigated and reconciled.

pkd2cgs(particle)

cgs2pkd(particle)

pkd2mks(particle)

mks2pkd(particle)

mks2cgs(particle)

cgs2mks(particle)

B.3 pkdtools.aggs

aggs is a subpackage of pkdtools. There is no formal distinction between aggregates and assemblies in pkdtools—aggregates are simply assemblies whose particles all have the same iOrgIdx value. However, the aggs subpackage includes a number of useful assembly methods that are intended for use with aggregates as well as some functions for generating regular shapes like dumbbells, tetrahedra, etc. Given the increasing interest in aggregates and irregular shapes in pkdgrav, this package should grow significantly in the future. In particular, JVD has a wide array of improvements in the works for this package as of Fall '23 which should be fully incorporated soon.

B.3.1 Aggregate methods

`agg_max()`

Return the largest (negative) iOrgIdx value in the assembly.

`agg_min()`

Return the smallest (negative) iOrgIdx value in the assembly.

`agg_range()`

Returns a tuple with the minimum and maximum iOrgIdx values in the assembly.

`agg_list()`

Return a list of all iOrgIdx values in the assembly.

`N_aggs()`

Return the number of aggs in the assembly.

`get_agg(iOrgIdx)`

Return a new assembly consisting only of particles in the desired aggregate. Any particles in the new assembly are copies of the originals, and any manipulations should not affect original assembly. Currently, this method can only accept a single iOrgIdx value at once.

`del_aggs(*iOrgIdxs)`

Delete any particles with matching iOrgIdx values from the assembly. Use the * operator when passing a list of iOrgIdx values.

`pop_agg(iOrgIdx)`

Delete the agg from the assembly and return a copy of it. Currently, this method can only accept a single iOrgIdx value at once.

`fix_orphans()`

Find any single particles with a negative iOrgIdx value ("orphans") and set their iOrgIdx value equal to their iOrder value.

`all_aggs(assembly, units=None)`

Return a list of assemblies, each containing all of the particles from one agg in the assembly passed in. By default, units will match those of the assembly passed in.

B.3.2 Functions for generating regular aggregates

`pkdtools` includes a set of functions for generating the 5 standard `pkdgrav` aggregate shapes included in `ssgen2Agg`. These functions generally work, but are a work in progress and may have some bugs or be updated in the future. Each function has a large number of possible arguments. While all arguments are optional individually, at a minimum the user must specify a mass and radius, or a particle mass and particle radius. The `mass` and `radius` arguments set a

total mass and overall radius for the entire aggregate, while the `pmass` and `pradius` arguments define the mass and size of each constituent particle. An orientation may be specified, but the angle about the orientation axis cannot currently be set. For example, calling `make_diamond()` with a `orientation` set to (0,1,0) will align the long axis of the diamond with the y-axis, but no guarantees are made as to the orientation of the short axes. In the future, these functions could be split off into their own utility since they aren't really core functions. Some things here may need to be reconciled with `ssgen2Agg`.

```
make_db(iOrder=0, iOrgIdx=-1, mass=0, radius=0,
center=(0,0,0), orientation=(0,0,1), color=2,
pmass=0, pradius=0, sep_coeff=np.sqrt(3), units='pkd')
```

Generate an assembly consisting of a single 2-particle, dumbbell-shaped aggregate. The user may specify a mass and radius for the whole agg, or alternately for the particles in the agg by using the `pmass` and `pradius` arguments in lieu of the `mass` and `radius` arguments.

```
make_diamond(iOrder=0, iOrgIdx=-1, mass=0, radius=0,
center=(0,0,0), orientation=(0,0,1), color=12,
pmass=0, pradius=0, sep_coeff=np.sqrt(3), units='pkd')
```

Generate an assembly consisting of a single 4-particle, planar diamond-shaped aggregate.

```
make_tetra(iOrder=0, iOrgIdx=-1, mass=0, radius=0,
center=(0,0,0), orientation=(0,0,1), color=3,
pmass=0, pradius=0, sep_coeff=np.sqrt(3), units='pkd')
```

Generate an assembly consisting of a single 4-particle, tetrahedron-shaped aggregate.

```
make_rod(iOrder=0, iOrgIdx=-1, mass=0, radius=0,
center=(0,0,0), orientation=(0,0,1), color=5,
pmass=0, pradius=0, sep_coeff=np.sqrt(3), units='pkd')
```

Generate an assembly consisting of a single 4-particle, rod-shaped aggregate.

```
make_cube(iOrder=0, iOrgIdx=-1, mass=0, radius=0,
center=(0,0,0), orientation=(0,0,1), color=7,
pmass=0, pradius=0, sep_coeff=np.sqrt(3), units='pkd')
```

Generate an assembly consisting of a single 8-particle, cube-shaped aggregate.

B.4 `pkdtools.rp`

The `rp` package includes set of functions for managing assemblies or datasets that contain multiple distinct rubble piles. The core capability is `find_rp()`, which returns a list of assemblies, each of which represents a single rubble pile. Neighbor searches are conducted using a k-D tree by default, although the user may substitute their own tree function. Large portions of `rp` are

modeled on and draw very heavily from the existing C-based `rpa` utility, especially `find_rp()` and its core supporting functions. When time allows, `rp` would benefit greatly from efficiency improvements, in particular for `find_rp()`. `find_rp()` is quite handy, but can be very slow for large N.

B.4.1 User functions

`find_rp(L=1.1, tree_func=scipy.spatial.KDTree, units=None)`

An assembly method that that returns a list of assemblies, each corresponding to a coherent fragment in the initial assembly. `find_rp()` is modeled on `rpa` and should give similar results. `L` is the linking scale and `tree_func` is used to reduce the time spent finding neighbors.

`rm_single(rp_list, particles=True, aggs=True, mass_frac=False, min_num=False)`

Returns a new list of rubble piles assemblies with any "singlets" (lone particles or lone aggregates) removed. Changing the `particles` or `aggs` inputs to `False` will toggle OFF this behavior. It can also remove all fragments/rubble piles below a given mass fraction (off by default).

`elong()`

An assembly method that returns the elongation of a rubble pile assembly.

`tot_M(rp_list)`

Returns the total mass of a list of rubble piles.

`max_M(rp_list)`

Returns the mass and mass fraction of the most massive fragment/rubble pile in `rp_list`.

B.4.2 Important utilities

`ok_to_merge(rp1, rp2, L)`

Determine whether two rubble piles meet the merger condition. From `rpa.c`: The following merger strategy is based entirely on geometry and does not take the gravitational potential into account. It is suitable for searching from the bottom up, that is, for starting with individual particles and linking them together into larger groups. The search takes the ellipsoidal shapes of the current groups into account. In order for `rp1` to be merged with `rp2`, spheres drawn with radii equal to the major axes of the bodies (times linking-scale) and centered on the bodies must overlap. If the scaled minor spheres also overlap, the bodies are merged. Failing that, if either body has its center of mass in the other's scaled ellipsoid, the bodies are merged. Otherwise, no merge occurs. In contrast with `rpa`, linking-scale is passed in here as a parameter `L`, defaulting to 1.1. `rp1` and `rp2` must both be assemblies, containing one or more particles. Currently `ok_to_merge()` does not consider units. This shouldn't be a problem when used as a utility here, but use caution...

`find_closest(rp_com, tree, N=10)`

Identifies closest `N` rubble piles to `rp_com`.

`merge_rp(rp_list, locs, tree, L, units=None)`

Executes one merging pass over the rubble piles in `rp_list`. There is no guarantee that this will work nicely for anything other than a "good faith" rubble pile. This operation is currently very slow, and probably the source of the `fine_rp()` bottleneck.

`in_ellipsoid(r0, R, a)`

Returns True if a ball with radius `R` at `r0` lies entirely within the ellipsoid defined by semi-axes `a` (measured along the Cartesian axes and centered at the origin). To get this right, we need to compute direction from `r0` to nearest point on ellipsoid surface. This is too hard, so settle for more conservative boundary.

`overlap(pos1, pos2, R1, R2)`

Determine whether two spheres are overlapping, given their positions and radii.

`split_particles(units=None)`

Take a standard assembly and return a list of assemblies, each a singlet containing one of the original particles. `split_particles()` may also be called as an assembly method.

B.5 pkdtools.constants

The constants subpackage has three modules: pkd, mks, and cgs. Each module defines five useful constants in terms of its units: the gravitational constant G, the masses of the Sun and the Earth, and the radii of the Sun and the Earth. This package could easily grow or expand in scope.

B.6 pkdtools.exp

exp is essentially a "scratch" space for ideas that have not been fully implemented. There is currently only one orphaned function here. This package may go away in the future.

embed_boulder(assembly, center, radius, units='pkd')

Embed a spherical boulder in a rubble pile. The center argument is defined relative to the agg center of mass. This function is experimental and likely has some bugs to work out. It's intended for a niche use case for JCM, but is included here for the sake of completeness.

B.7 pkdtools.tidal

The tidal package contains tools that were developed to support the analysis of large suites of tidal disruption or spinup simulations (or any other processes that produce lots of rubble pile fragments). This was intended to support JCM's thesis work, but may end up being useful for other projects, especially the spinup work that JCM never completed. tidal is composed of three modules: dgrid, pgrid, and hist.

B.7.1 dgrid

The dgrid module handles the collection and storage of data from large suites of tidal runs. It defines the DataGrid class, which is used for the collection and storage of data from tidal runs. It also includes functions for traversing a large suite of runs and calling analysis functions, as well as a number of analysis metrics.

DataGrid

A new class intended to accumulate data as walk_grid() progresses through a given tidal suite of runs. DataGrid should be provided with a list of speed at infinity v_∞ values, a list of close approach distance q values, a filename, and an analysis function. In most cases, this function will ingest a list of rubble piles (i.e., output from find_rp()) and return a single value for a given run. DataGrid includes the following attributes:

- q_list: A list of the q values included in the suite of simulations
- vinf_list: A list of the v_∞ values included in the suite of simulations
- filename: The filename that the collected data will ultimately be written to.
- func: The analysis function that will be called for each simulation. This is where the main body of the grid analysis scheme happens.
- grid: A pandas dataframe size to match the q and v_∞ values supplied by the user. This is where the data will be stored.

Users must provide `q_list`, `vinf_list`, `filename`, and `func` upon initialization. `DataGrid` also has four associated methods:

- `read(q, vinf)` Returns the current value of `grid` at the given coordinates.
- `write(q, vinf, val)` Writes `val` to `grid` at the given coordinates.
- `save_csv()` Writes `grid` to `filename` in csv format. Intended for use with single value analysis functions.
- `save_pickle()` Writes `grid` to `filename` in pickle format.

```
walk_grid(q_list, vinf_list, filename=None,
fraggrids=[], stepgrids=[], othergrids=[], L=1.2,
fragunits='pkd', stepunits='pkd', otherunits='pkd')
```

Step through each run directory in the provided `DataGrid` objects and call the analysis functions on `filename`. `fragrids` should be a list of `DataGrid` objects for analyses that take a list of all fragments in `filename` as input. `stepgrids` should have analysis functions assigned that take q, v_∞, and a list of all steps in the run directory. `othergrids` should be a list of `DataGrid` objects for all analyses that do not meet this format, and require a more general initialization.

`analyze_frags(q, vinf, *fraggrids, filename=None, L=1.2, del_earth=True, units='pkd')`

Load data from `filename`, remove any Earth particle, and locate fragments using `find_rp()`. Any further fragment cleaning should happen in the analysis function, since this may be treated differently for different metrics. Grouping this way avoids calling `find_rp()` for each individual analysis, which can save quite a bit of time. Any `DataGrid` objects passed in will be updated. Nothing should be returned. If `filename` is not specified, `analyze_frags()` should automatically find the last valid output file written to the current run directory. Analysis functions called in this way must return a single numeric value and associated `DataGrid` objects will be saved in csv format

`analyze_steps(q, vinf, *stepgrids, del_earth=True, units='pkd')`

Similar to `analyze_frags()` in concept, but used for analyses that require stepping through all (or a subset of) output files, rather than considering only the final state. E.g., `tidal_threshold()`. Any function called in this way should accept q, v_∞, and a list of all steps in the run directory as arguments. This list is computed automatically in `analyze_steps()` by `steplist()`.Analysis functions called in this way must return a single numeric value and associated `DataGrid` objects will be saved in csv format

`analyze_other(q, vinf, *othergrids, units='pkd')`

We set aside calls to any functions that *do not* require fragment analysis, since `find_rp()` is expensive and we benefit substantially from only performing this calculation once per run. Also, we may want to use analysis functions that don't return numerical values, but instead return plots, or histograms, etc. Any analysis functions that don't fit the final state fragment or steplist analysis models well can be called here. No assumptions about inputs are made beyond q and v_∞.

`gen_walk(q_list, vinf_list, func)`

A much more general approach to traversing a suite of tidal encounter-style runs, in contrast to `walk_grid()`. Accepts q and v_∞ lists to be treated as a grid and a function to be called with no arguments. No assumptions are made about this function. Allows for maximum flexibility at the expense of efficiency.

`get_stepsize()`

Extract the `iOutInterval` values from the `ss.par` file in the current directory.

`get_zeropad()`

Extract the `nDigits` value from `ss.par` in the current directory.

`final_step(stepsize, zeropad)`

Find the final sequential `ss` output file written to the current directory.

`get_steps(stepsize, zeropad, init="initcond.ss")`

Return a list of all sequential `ss` output files in the current directory, beginning with `init`.

`pop_earth()`

Pop the Earth particle from an assembly and return, if one is present. May be called as a function or an assembly method.

`rm_earth()`

Remove the Earth particle from an assembly, if one is present. May be called as a function or an assembly method.

The remaining entries in this section are all analysis functions.

`largest_fragment_fraction(frags)`

Returns the fraction of total system mass of the most massive fragment in the list.

`largest_fragment_elongation(frags)`

Return the elongation of the most massive fragment in the list.

`largest_fragment_bound(frags)`

Returns the fraction of total system mass *bound* to the largest fragment, determined by a simple two-body calculation. In other words, if a a fragment in isolation is determined to be

gravitationally bound to the largest fragment, its mass is added to the total. The total also includes the mass of the largest fragment itself.

`largest_fragment_orbiting(frags)`

Returns the fraction of total system mass in orbit about the largest fragment. This *excludes* the mass of the largest fragment itself.

`largest_fragment_period(frags)`

Returns the spin period of the largest post-encounter fragment in units of hours.

`N_fragments(frags)`

Returns the total number of fragments, after removing anything under 10 particles.

`largest_fragment_volcore(q, vinf, steplist, units='pkd')`

Returns the fraction of material in the largest fragment that came from the inner 50% by *volume* of the initial body.

`largest_fragment_radcore(q, vinf, steplist, units='pkd')`

Returns the fraction of material in the largest post-encounter fragment that came from the inner 50% by *radius* of the initial body. This is almost identical to `largest_fragment_volcore()`, but with a different criterion for what constitutes the "core" of the initial rubble pile. Here, any particle closer to the center than half of the radius is considered

the core. This is a more restrictive condition, so we should expect lower values.

`tidal_threshold(q, vinf, steplist, units='pkd')`

Returns the distance from the primary body at which the rubble pile is disrupted or reshaped. We consider the threshold for a reshaping in this context to be an increase of 0.5% in bulk radius.

`tidal_run_complete(q, vinf, steplist, units='pkd')`

Determine if a tidal run is done evolving. We first check whether the run has proceeded for more than three times the expected time to periapse. If it hasn't, we consider the run incomplete. If it has, we consider the mass fraction of the largest fragment. If it is greater than 95% at this point, the is likely a case of minimal tidal disturbance and we consider the run to be complete. If it isn't, we know that a major disruption did occur. In this case, we compare the mass fraction of the largest fragment in the final recorded output file to the largest mass fraction in the 10th to last output file. If these values are within 10% of each other, we consider the run to be finished evolving. This usually works well, but does give false negatives in the case where a messy fragment train causes the fragment finder to give slightly different results for different time steps.

B.7.2 `pgrid`

`pgrid` works in tandem with `dgrid` to allow for easy plotting of the `DataGrid` objects created by `dgrid`. The data is saved as a plain `pandas` dataframe and may be loaded and plotted with the functions below.

`load_csv_grid(filename)`

Read in a dataframe written from a `DataGrid` in csv format and process in preparation for plotting.

`load_pickle_grid(filename)`

Read in a dataframe written from a `DataGrid` in pickle format and process in preparation for plotting.

`plot_grid(title, rows, columns, datagrids=[], labels=[], cmaps=[], fig=None, axes=[], center=None, big_size=32, med_size=28, xtick_freq=2, show=True)`

Intended for plotting multiple data grids. User may specify the number and arrangement of plots, font sizes, labels, color maps (from the Seaborn plotting package), and more. The `fig` and `axes` arguments may be used if the user wants to call `plot_grid()` multiple times on different sets of data grids but show all of the resulting plots on the same figure. If an existing figure and axes are specified, the plots will be placed there. Otherwise, a new figure and axes will be created.

B.7.3 `hist`

An unfinished module designed to create histograms from tidal (or spinup) grid-based data.

`get_mass_hist(q, vinf, units='pkd')`

An analysis function designed to be called with `analyze_other().get_mass_hist()` finds the final output file and calculates histogram data of fragment mass as a function of total system mass.

`get_frag_masses(q, vinf, units='pkd')`

An analysis function designed to be called with `analyze_other()`. Finds the final output file and returns a sorted list of all fragment masses as a function of total system mass.

Appendix C: Facilities and software

The great majority of the simulations described in this dissertation were carried out using the University of Maryland's Deepthought2 and Zaratan computing clusters. Zaratan replaced Deepthought2 as the University of Maryland's flagship computing cluster in 2022 and is maintained by the Division of Information Technology. Some work was also carried out on the Yorp cluster, which is administered by the Center for Theory and Computation at the University of Maryland Department of Astronomy. Our analyses also relied on the following open-source Python packages: NumPy (Harris et al., 2020), Matplotlib (Hunter, 2007), and SciPy (Virtanen et al., 2020). Finally, we used ParaView (Ahrens et al., 2005) and the Persistence of Vision Raytracer[1] (POV-Ray) in creating many of the visualizations presented in this dissertation.

[1] https://www.povray.org/

Bibliography

Abe, S., Mukai, T., Hirata, N., Barnouin-Jha, O. S., Cheng, A. F., Demura, H., Gaskell, R. W., Hashimoto, T., Hiraoka, K., Honda, T., et al. (2006). Mass and local topography measurements of itokawa by hayabusa. *Science*, 312(5778):1344–1347.

Aggarwal, H. and Oberbeck, V. (1974). Roche limit of a solid body. *The Astrophysical Journal*, 191:577–588.

Ahrens, J., Geveci, B., Law, C., Hansen, C., and Johnson, C. (2005). 36-paraview: An end-user tool for large-data visualization. *The visualization handbook*, 717:50038–1.

Alvarez, W., Smit, J., Lowrie, W., Asaro, F., Margolis, S. V., Claeys, P., Kastner, M., and Hildebrand, A. R. (1992). Proximal impact deposits at the cretaceous-tertiary boundary in the gulf of mexico: A restudy of dsdp leg 77 sites 536 and 540. *Geology*, 20(8):697–700.

Asphaug, E. (2007). Asteroid surfaces as expressions of seismic interiors. In *38th Annual Lunar and Planetary Science Conference*, page 2432.

Asphaug, E. and Benz, W. (1994). Density of comet shoemaker–levy 9 deduced by modelling breakup of the parent'rubble pile'. *Nature*, 370(6485):120–124.

Asphaug, E. and Benz, W. (1996). Size, density, and structure of comet shoemaker–levy 9 inferred from the physics of tidal breakup. *Icarus*, 121(2):225–248.

Astakhov, S. A., Lee, E. A., and Farrelly, D. (2005). Formation of kuiper-belt binaries through multiple chaotic scattering encounters with low-mass intruders. *Monthly Notices of the Royal Astronomical Society*, 360(2):401–415.

A'Hearn, M. F., Belton, M. J., Delamere, A., and Blume, W. H. (2005). Deep impact: A large-scale active experiment on a cometary nucleus. *Space Science Reviews*, 117:1–21.

Barnouin, O., Daly, M., Palmer, E., Gaskell, R., Weirich, J., Johnson, C., Al Asad, M., Roberts, J., Perry, M., Susorney, H., et al. (2019). Shape of (101955) bennu indicative of a rubble pile with internal stiffness. *Nature geoscience*, 12(4):247–252.

Batygin, K., Brown, M. E., and Fraser, W. C. (2011). Retention of a primordial cold classical kuiper belt in an instability-driven model of solar system formation. *The Astrophysical Journal*, 738(1):13.

Benecchi, S., Borncamp, D., Parker, A., Buie, M., Noll, K., Binzel, R., Stern, S. A., Verbiscer, A., Kavelaars, J., Zangari, A., et al. (2019a). The color and binarity of (486958) 2014 mu69 and other long-range new horizons kuiper belt targets. *Icarus*, 334:22–29.

Benecchi, S., Noll, K., Grundy, W., Buie, M., Stephens, D., and Levison, H. (2009). The correlated colors of transneptunian binaries. *Icarus*, 200(1):292–303.

Benecchi, S. D., Porter, S., Buie, M., Zangari, A., Verbiscer, A. J., Noll, K., Stern, S., Spencer, J., and Parker, A. (2019b). The hst lightcurve of (486958) 2014 mu69. *Icarus*, 334:11–21.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

Benz, W. and Asphaug, E. (1995). Simulations of brittle solids using smooth particle hydrodynamics. *Computer physics communications*, 87(1-2):253–265.

Binzel, R. P., Morbidelli, A., Merouane, S., DeMeo, F. E., Birlan, M., Vernazza, P., Thomas, C. A., Rivkin, A. S., Bus, S. J., and Tokunaga, A. T. (2010). Earth encounters as the origin of fresh surfaces on near-earth asteroids. *Nature*, 463(7279):331–334.

Boss, A., Cameron, A., and Benz, W. (1991). Tidal disruption of inviscid planetesimals. *Icarus*, 92(1):165–178.

Bottke Jr, W. F. and Melosh, H. J. (1996). Binary asteroids and the formation of doublet craters. *Icarus*, 124(2):372–391.

Bottke Jr, W. F., Richardson, D. C., and Love, S. G. (1997). Can tidal disruption of asteroids make crater chains on the earth and moon? *Icarus*, 126(2):470–474.

Bottke Jr, W. F., Richardson, D. C., Michel, P., and Love, S. G. (1999). 1620 geographos and 433 eros: Shaped by planetary tides? *The Astronomical Journal*, 117(4):1921.

Brozović, M., Benner, L. A., McMichael, J. G., Giorgini, J. D., Pravec, P., Scheirich, P., Magri, C., Busch, M. W., Jao, J. S., Lee, C. G., et al. (2018). Goldstone and arecibo radar observations of (99942) apophis in 2012–2013. *Icarus*, 300:115–128.

Cannon, K. M., Gialich, M., and Acain, J. (2023). Precious and structural metals on asteroids. *Planetary and Space Science*, 225:105608.

Chapman, C. R. (1978). Asteroid collisions, craters, regoliths, and lifetimes. *NASA, Washington Asteroids*.

Cheng, A., Chabot, N., Rivkin, A., Stickle, A., Fahnestock, E., Della Corte, V., and Dotto, E. (2021). Double asteroid redirection test: Mission overview. *43rd COSPAR Scientific Assembly. Held 28 January-4 February*, 43:274.

Chodas, P. W. and Yeomans, D. K. (1996). The orbital motion and impact circumstances of comet shoemaker-levy 9. In *International Astronomical Union Colloquium*, volume 156, pages 1–30. Cambridge University Press.

Chujo, T., Mori, O., Kawaguchi, J., and Yano, H. (2018). Categorization of brazil nut effect and its reverse under less-convective conditions for microgravity geology. *Monthly Notices of the Royal Astronomical Society*, 474(4):4447–4459.

Cleary, P. W. (2008). The effect of particle shape on simple shear flows. *Powder Technology*, 179(3):144–163.

Cotto-Figueroa, D., Statler, T. S., Richardson, D. C., and Tanga, P. (2015). Coupled spin and shape evolution of small rubble-pile asteroids: Self-limitation of the yorp effect. *The Astrophysical Journal*, 803(1):25.

Ćuk, M. (2007). Formation and destruction of small binary asteroids. *The Astrophysical Journal*, 659(1):L57.

Ćuk, M. (2018). 1i/'oumuamua as a tidal disruption fragment from a binary star system. *The Astrophysical Journal Letters*, 852(1):L15.

Cundall, P. A. (1971). A computer model for simulating progressive, large-scale movement in blocky rock system. In *Proceedings of the international symposium on rock mechanics*, volume 8, pages 129–136.

Cundall, P. A. and Strack, O. D. (1979). A discrete numerical model for granular assemblies. *geotechnique*, 29(1):47–65.

Daly, R. T., Ernst, C. M., Barnouin, O. S., Chabot, N. L., Rivkin, A. S., Cheng, A. F., Adams, E. Y., Agrusa, H. F., Abel, E. D., Alford, A. L., et al. (2023). Successful kinetic impact into an asteroid for planetary defence. *Nature*, 616(7957):443–447.

Davidsson, B., Sierks, H., Güttler, C., Marzari, F., Pajola, M., Rickman, H., A'Hearn, M. F., Auger, A.-T., El-Maarry, M. R., Fornasier, S., et al. (2016). The primordial nucleus of comet 67p/churyumov-gerasimenko. *Astronomy & Astrophysics*, 592:A63.

Davidsson, B. J. (2001). Tidal splitting and rotational breakup of solid biaxial ellipsoids. *Icarus*, 149(2):375–383.

DeMartini, J. and Richardson, D. (2020). An efficient algorithm for operating on large numbers of aggregate particles with applications to simulating the dynamics of irregularly shaped grains. In *Europlanet Science Congress 2020*, page 383.

DeMartini, J. V., Richardson, D. C., Barnouin, O. S., Schmerr, N. C., Plescia, J. B., Scheirich, P., and Pravec, P. (2019). Using a discrete element method to investigate seismic response and spin change of 99942 apophis during its 2029 tidal encounter with earth. *Icarus*, 328:93–103.

Edgeworth, K. E. (1943). The evolution of our planetary system. *Journal of the British Astronomical Association, vol. 53, p. 181-188 (1943).*, 53:181–188.

Ferrari, F. and Tanga, P. (2020). The role of fragment shapes in the simulations of asteroids as gravitational aggregates. *Icarus*, 350:113871.

Fujiwara, A., Kawaguchi, J., Yeomans, D., Abe, M., Mukai, T., Okada, T., Saito, J., Yano, H., Yoshikawa, M., Scheeres, D., et al. (2006). The rubble-pile asteroid itokawa as observed by hayabusa. *Science*, 312(5778):1330–1334.

Funato, Y., Makino, J., Hut, P., Kokubo, E., and Kinoshita, D. (2004). The formation of kuiper-belt binaries through exchange reactions. *Nature*, 427(6974):518–520.

Gingold, R. A. and Monaghan, J. J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389.

Ginsburg, I., Lingam, M., and Loeb, A. (2018). Galactic panspermia. *The Astrophysical Journal Letters*, 868(1):L12.

Goldreich, P., Lithwick, Y., and Sari, R. (2002). Formation of kuiper-belt binaries by dynamical friction and three-body encounters. *Nature*, 420(6916):643–646.

Graham, S. L., Kessler, P. B., and McKusick, M. K. (1982). Gprof: A call graph execution profiler. *ACM Sigplan Notices*, 17(6):120–126.

Greenstreet, S., Gladman, B., McKinnon, W. B., Kavelaars, J., and Singer, K. N. (2019). Crater density predictions for new horizons flyby target 2014 mu69. *The Astrophysical Journal Letters*, 872(1):L5.

Groussin, O., Attree, N., Brouet, Y., Ciarletti, V., Davidsson, B., Filacchione, G., Fischer, H.-H., Gundlach, B., Knapmeyer, M., Knollenberg, J., et al. (2019). The thermal, mechanical, structural, and dielectric properties of cometary nuclei after rosetta. *Space Science Reviews*, 215:1–51.

Gulbis, A. A., Elliot, J., and Kane, J. F. (2006). The color of the kuiper belt core. *Icarus*, 183(1):168–178.

Güttler, C., Blum, J., Zsom, A., Ormel, C. W., and Dullemond, C. P. (2010). The outcome of protoplanetary dust growth: pebbles, boulders, or planetesimals?-i. mapping the zoo of laboratory collision experiments. *Astronomy & Astrophysics*, 513:A56.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Heinisch, P., Auster, H.-U., Gundlach, B., Blum, J., Güttler, C., Tubiana, C., Sierks, H., Hilchenbach, M., Biele, J., Richter, I., et al. (2019). Compressive strength of comet 67p/churyumov-gerasimenko derived from philae surface contacts. *Astronomy & Astrophysics*, 630:A2.

Hirabayashi, M. and Scheeres, D. J. (2014). Stress and failure analysis of rapidly rotating asteroid (29075) 1950 da. *The Astrophysical Journal Letters*, 798(1):L8.

Hirayama, T. and Moriyama, F. (1965). Observations of comet ikeya-seki (1965f). *Publications of the Astronomical Society of Japan, vol. 17, p. 433 (1965).*, 17:433.

Holsapple, K. A. and Michel, P. (2006). Tidal disruptions: A continuum theory for solid bodies. *Icarus*, 183(2):331–348.

Holsapple, K. A. and Michel, P. (2008). Tidal disruptions: Ii. a continuum theory for solid bodies with strength, with applications to the solar system. *Icarus*, 193(1):283–301.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Jacobson, S. A. and Scheeres, D. J. (2011). Dynamics of rotationally fissioned asteroids: Source of observed small asteroid systems. *Icarus*, 214(1):161–178.

Jeffreys, H. (1947). The relation of cohesion to roche's limit. *Monthly Notices of the Royal Astronomical Society*, 107(3):260–262.

Jetley, P., Gioachin, F., Mendes, C., Kale, L. V., and Quinn, T. (2008). Massively parallel cosmological simulations with changa. In *2008 IEEE International Symposium on Parallel and Distributed Processing*, pages 1–12. IEEE.

Jewitt, D. (1999). Kuiper belt objects. *Annual Review of Earth and Planetary Sciences*, 27(1):287–312.

Jewitt, D. and Luu, J. (1993). Discovery of the candidate kuiper belt object 1992 qb1. *Nature*, 362(6422):730–732.

Johansen, A., Oishi, J. S., Low, M.-M. M., Klahr, H., Henning, T., and Youdin, A. (2007). Rapid planetesimal formation in turbulent circumstellar disks. *Nature*, 448(7157):1022–1025.

Jorda, L., Gaskell, R., Capanna, C., Hviid, S., Lamy, P., Ďurech, J., Faury, G., Groussin, O., Gutiérrez, P., Jackman, C., et al. (2016). The global shape, density and rotation of comet 67p/churyumov-gerasimenko from preperihelion rosetta/osiris observations. *Icarus*, 277:257–278.

Jutzi, M. (2015). Sph calculations of asteroid disruptions: The role of pressure dependent failure models. *Planetary and space science*, 107:3–9.

Jutzi, M. (2019). The shape and structure of small asteroids as a result of sub-catastrophic collisions. *Planetary and space science*, 177:104695.

Jutzi, M. and Benz, W. (2017). Formation of bi-lobed shapes by sub-catastrophic collisions-a late origin of comet 67p's structure. *Astronomy & Astrophysics*, 597:A62.

Jutzi, M., Michel, P., and Richardson, D. C. (2019). Fragment properties from large-scale asteroid collisions: I: Results from sph/n-body simulations using porous parent bodies and improved material models. *Icarus*, 317:215–228.

Kenyon, S. J. (2002). Planet formation in the outer solar system. *Publications of the Astronomical Society of the Pacific*, 114(793):265.

Kim, Y., DeMartini, J. V., Richardson, D. C., and Hirabayashi, M. (2023). Tidal resurfacing model for (99942) apophis during the 2029 close approach with earth. *Monthly Notices of the Royal Astronomical Society*, 520(3):3405–3415.

Knuth, M. A., Johnson, J., Hopkins, M., Sullivan, R., and Moore, J. (2012). Discrete element modeling of a mars exploration rover wheel in granular material. *Journal of Terramechanics*, 49(1):27–36.

Kuiper, G. P. (1951). On the origin of the solar system. *Proceedings of the National Academy of Sciences*, 37(1):1–14.

Lee, E. A., Astakhov, S. A., and Farrelly, D. (2007). Production of trans-neptunian binaries through chaos-assisted capture. *Monthly Notices of the Royal Astronomical Society*, 379(1):229–246.

Leinhardt, Z. M., Richardson, D. C., and Quinn, T. (2000). Direct n-body simulations of rubble pile collisions. *Icarus*, 146(1):133–151.

Leonard, F. C. (1930). The new planet pluto. *Astronomical Society of the Pacific Leaflets, Vol. 1, No. 30, p. 121*, 1:121.

Longmore, J.-P., Marais, P., and Kuttel, M. M. (2013). Towards realistic and interactive sand simulation: A gpu-based framework. *Powder Technology*, 235:983–1000.

Margot, J.-L., Nolan, M., Benner, L., Ostro, S., Jurgens, R., Giorgini, J., Slade, M., and Campbell, D. (2002). Binary asteroids in the near-earth object population. *Science*, 296(5572):1445–1448.

Marohnic, J. C., DeMartini, J. V., Richardson, D. C., Zhang, Y., and Walsh, K. J. (2023). An efficient numerical approach to modeling the effects of particle shape on rubble-pile dynamics. *The Planetary Science Journal*, Acccepted for publication.

Marohnic, J. C., Richardson, D. C., McKinnon, W. B., Agrusa, H. F., DeMartini, J. V., Cheng, A. F., Stern, S. A., Olkin, C. B., Weaver, H. A., Spencer, J. R., et al. (2021). Constraining the final merger of contact binary (486958) arrokoth with soft-sphere discrete element simulations. *Icarus*, 356:113824.

Matsumura, S., Richardson, D. C., Michel, P., Schwartz, S. R., and Ballouz, R.-L. (2014). The brazil nut effect and its application to asteroids. *Monthly Notices of the Royal Astronomical Society*, 443(4):3368–3380.

Maurel, C., Ballouz, R.-L., Richardson, D. C., Michel, P., and Schwartz, S. R. (2016). Numerical simulations of oscillation-driven regolith motion: Brazil-nut effect. *Monthly Notices of the Royal Astronomical Society*, page stw2641.

McKinnon, W., Richardson, D., Marohnic, J., Keane, J., Grundy, W., Hamilton, D., Nesvorný, D., Umurhan, O., Lauer, T., Singer, K., et al. (2020). The solar nebula origin of (486958) arrokoth, a primordial contact binary in the kuiper belt. *Science*, 367(6481):eaay6620.

Mead, S. R. and Cleary, P. W. (2015). Validation of dem prediction for granular avalanches on irregular terrain. *Journal of Geophysical Research: Earth Surface*, 120(9):1724–1742.

Melosh, H. J. and Schenk, P. (1993). Split comets and the origin of crater chains on ganymede and callisto. *Nature*, 365(6448):731–733.

Michel, P., DeMeo, F. E., and Bottke, W. F. (2015). *Asteroids iv*. University of Arizona Press.

Michel, P., Küppers, M., Bagatin, A. C., Carry, B., Charnoz, S., De Leon, J., Fitzsimmons, A., Gordo, P., Green, S. F., Hérique, A., et al. (2022). The esa hera mission: detailed characterization of the dart impact outcome and of the binary asteroid (65803) didymos. *The planetary science journal*, 3(7):160.

Michikami, T., Honda, C., Miyamoto, H., Hirabayashi, M., Hagermann, A., Irie, T., Nomura, K., Ernst, C. M., Kawamura, M., Sugimoto, K., et al. (2019). Boulder size and shape distributions on asteroid ryugu. *Icarus*, 331:179–191.

Morbidelli, A., Bottke, W. F., Nesvorný, D., and Levison, H. F. (2009). Asteroids were born big. *Icarus*, 204(2):558–573.

Morbidelli, A., Chambers, J., Lunine, J. I., Petit, J.-M., Robert, F., Valsecchi, G. B., and Cyr, K. (2000). Source regions and timescales for the delivery of water to the earth. *Meteoritics & Planetary Science*, 35(6):1309–1320.

Morbidelli, A., Levison, H. F., and Gomes, R. (2008). The dynamical structure of the kuiper belt and its primordial origin. *The solar system beyond Neptune*, pages 275–292.

Morbidelli, A. and Nesvorný, D. (2020). Kuiper belt: formation and evolution. *The transneptunian solar system*, pages 25–59.

Movshovitz, N., Asphaug, E., and Korycansky, D. (2012). Numerical modeling of the disruption of comet d/1993 f2 shoemaker–levy 9 representing the progenitor by a gravitationally bound assemblage of randomly shaped polyhedra. *The Astrophysical Journal*, 759(2):93.

Mullet, B., Segall, P., and Fávero Neto, A. (2023). Numerical modeling of caldera formation using smoothed particle hydrodynamics (sph). *Geophysical Journal International*, 234(2):887–902.

Nesvorný, D., Li, R., Youdin, A. N., Simon, J. B., and Grundy, W. M. (2019). Trans-neptunian binaries as evidence for planetesimal formation by the streaming instability. *Nature Astronomy*, 3(9):808–812.

Nesvorný, D. and Vokrouhlický, D. (2019). Binary survival in the outer solar system. *Icarus*, 331:49–61.

Nesvorný, D., Youdin, A. N., and Richardson, D. C. (2010). Formation of kuiper belt binaries by gravitational collapse. *The Astronomical Journal*, 140(3):785.

Nguyen, T. D. and Plimpton, S. J. (2019). Aspherical particle models for molecular dynamics simulation. *Computer Physics Communications*, 243:12–24.

Nolan, G. and Kavanagh, P. (1995). Random packing of nonspherical particles. *Powder technology*, 84(3):199–205.

Noll, K. S., Grundy, W. M., Chiang, E. I., Margot, J.-L., and Kern, S. D. (2007). Binaries in the kuiper belt. *arXiv preprint astro-ph/0703134*.

Noll, K. S., Grundy, W. M., Nesvorný, D., and Thirouin, A. (2020). Trans-neptunian binaries (2018). In *The trans-Neptunian solar system*, pages 205–224. Elsevier.

Noll, K. S., Grundy, W. M., Stephens, D. C., Levison, H. F., and Kern, S. D. (2008). Evidence for two populations of classical transneptunian objects: The strong inclination dependence of classical binaries. *Icarus*, 194(2):758–768.

Öpik, E. (1966). Sun-grazing comets and tidal disruption. *Irish Astronomical Journal, vol. 7 (5), p. 141*, 7:141.

Öpik, E. J. (1950). Stellar models with variable composition. i. models with a discontinuity of molecular weight. In *Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*, volume 53, pages 1–16. JSTOR.

O'Brien, D. P., Walsh, K. J., Morbidelli, A., Raymond, S. N., and Mandell, A. M. (2014). Water delivery and giant impacts in the 'grand tack'scenario. *Icarus*, 239:74–84.

Pan, W., Tartakovsky, A. M., and Monaghan, J. J. (2013). Smoothed particle hydrodynamics non-newtonian model for ice-sheet and ice-shelf dynamics. *Journal of Computational Physics*, 242:828–842.

Piani, L., Marrocchi, Y., Rigaudier, T., Vacher, L. G., Thomassin, D., and Marty, B. (2020). Earth's water may have been inherited from material similar to enstatite chondrite meteorites. *Science*, 369(6507):1110–1113.

Potter, D. and Stadel, J. (2016). Pkdgrav3: Parallel gravity code. *Astrophysics Source Code Library*, pages ascl–1609.

Potter, D., Stadel, J., and Teyssier, R. (2017). Pkdgrav3: beyond trillion particle cosmological simulations for the next era of galaxy surveys. *Computational Astrophysics and Cosmology*, 4(1):2.

Pravec, P. and Harris, A. W. (2000). Fast and slow rotation of asteroids. *Icarus*, 148(1):12–20.

Pravec, P., Harris, A. W., and Michalowski, T. (2002). Asteroid rotations. *Asteroids III*, 113:35.

Prialnik, D., Sarid, G., Rosenberg, E. D., and Merk, R. (2008). Thermal and chemical evolution of comet nuclei and kuiper belt objects. In *Origin and Early Evolution of Comet Nuclei: Workshop honouring Johannes Geiss on the occasion of his 80th birthday*, pages 147–164. Springer.

Quillen, A. C., Kueter-Young, A., Frouard, J., and Ragozzine, D. (2016). Tidal spin-down rates of homogeneous triaxial viscoelastic bodies. *Monthly Notices of the Royal Astronomical Society*, 463(2):1543–1553.

Quinn, T., Perrine, R. P., Richardson, D. C., and Barnes, R. (2010). A symplectic integrator for hill's equations. *The Astronomical Journal*, 139(2):803.

Richardson, D., Michel, P., Walsh, K., and Flynn, K. (2009). Numerical simulations of asteroids modelled as gravitational aggregates with cohesion. *Planetary and Space Science*, 57(2):183–192.

Richardson, D. C., Bottke Jr, W. F., and Love, S. G. (1998). Tidal distortion and disruption of earth-crossing asteroids. *Icarus*, 134(1):47–76.

Richardson, D. C., Leinhardt, Z. M., Melosh, H., Bottke Jr, W., Asphaug, E., et al. (2002). Gravitational aggregates: Evidence and evolution. *Asteroids iii*, 1:501–515.

Richardson, D. C., Quinn, T., Stadel, J., and Lake, G. (2000). Direct large-scale n-body simulations of planetesimal dynamics. *Icarus*, 143(1):45–59.

Roche, E. A. (1847). La figure d'une masse fluide soumise á l'attraction d'un point éloigné. *Acad. Sci. Montpellier*, 1:243–262.

Roig, F., Duffard, R., Penteado, P., Lazzaro, D., and Kodama, T. (2003). Interacting ellipsoids: a minimal model for the dynamics of rubble-pile bodies. *Icarus*, 165(2):355–370.

Rosato, A., Strandburg, K. J., Prinz, F., and Swendsen, R. H. (1987). Why the brazil nuts are on top: Size segregation of particulate matter by shaking. *Physical review letters*, 58(10):1038.

Rozitis, B., MacLennan, E., and Emery, J. P. (2014). Cohesive forces prevent the rotational breakup of rubble-pile asteroid (29075) 1950 da. *Nature*, 512(7513):174–176.

Rubincam, D. P. (2000). Radiative spin-up and spin-down of small asteroids. *Icarus*, 148(1):2–11.

Saha, P. and Tremaine, S. (1992). Symplectic integrators for solar system dynamics. *Astronomical Journal (ISSN 0004-6256), vol. 104, no. 4, p. 1633-1640.*, 104:1633–1640.

Samarasinha, N. H., Belton, M., Farnham, T., Gutierrez, P., Mueller, B., and Chesley, S. (2009). Bulk density of comet 9p/tempel 1 based on orbital and rotational nongravitational effects. In *AAS/Division for Planetary Sciences Meeting Abstracts# 41*, volume 41, pages 37–05.

Sánchez, D. P. and Scheeres, D. J. (2012). Dem simulation of rotation-induced reshaping and disruption of rubble-pile asteroids. *Icarus*, 218(2):876–894.

Sánchez, P., Renouf, M., Azéma, E., Mozul, R., and Dubois, F. (2021). A contact dynamics code implementation for the simulation of asteroid evolution and regolith in the asteroid environment. *Icarus*, 363:114441.

Sánchez, P. and Scheeres, D. J. (2011). Simulating asteroid rubble piles with a self-gravitating soft-sphere distinct element method model. *The Astrophysical Journal*, 727(2):120.

Sánchez, P. and Scheeres, D. J. (2014). The strength of regolith and rubble pile asteroids. *Meteoritics & Planetary Science*, 49(5):788–811.

Sarkar, D., Goudarzy, M., and König, D. (2019). An interpretation of the influence of particle shape on the mechanical behavior of granular material. *Granular Matter*, 21(3):1–24.

Scheeres, D., Benner, L., Ostro, S., Rossi, A., Marzari, F., and Washabaugh, P. (2005). Abrupt alteration of asteroid 2004 mn4's spin state during its 2029 earth flyby. *Icarus*, 178(1):281–283.

Scheeres, D. J. (2015). Landslides and mass shedding on spinning spheroidal asteroids. *Icarus*, 247:1–17.

Scheeres, D. J., Marzari, F., and Rossi, A. (2004). Evolution of neo rotation rates due to close encounters with earth and venus. *Icarus*, 170(2):312–323.

Scheeres, D. J. and Sánchez, P. (2018). Implications of cohesive strength in asteroid interiors and surfaces and its measurement. *Progress in Earth and Planetary Science*, 5(1):1–10.

Schlichting, H. E. and Sari, R. (2008). Formation of kuiper belt binaries. *The Astrophysical Journal*, 673(2):1218.

Schwartz, S. R., Michel, P., Jutzi, M., Marchi, S., Zhang, Y., and Richardson, D. C. (2018). Catastrophic disruptions as the origin of bilobate comets. *Nature astronomy*, 2(5):379–382.

Schwartz, S. R., Michel, P., and Richardson, D. C. (2013). Numerically simulating impact disruptions of cohesive glass bead agglomerates using the soft-sphere discrete element method. *Icarus*, 226(1):67–76.

Schwartz, S. R., Michel, P., Richardson, D. C., and Yano, H. (2014). Low-speed impact simulations into regolith in support of asteroid sampling mechanism design i: Comparison with 1-g experiments. *Planetary and Space Science*, 103:174–183.

Schwartz, S. R., Richardson, D. C., and Michel, P. (2012). An implementation of the soft-sphere discrete element method in a high-performance parallel gravity tree-code. *Granular Matter*, 14:363–380.

Scotti, J. and Melosh, H. (1993). Estimate of the size of comet shoemaker-levy 9 from a tidal breakup model. *Nature*, 365(6448):733–735.

Sekiguchi, N. (1970). On the fissions of a solid body under influence of tidal force; with application to the problem of twin craters on the moon. *The moon*, 1(4):429–439.

Solem, J. C. (1994). Density and size of comet shoemaker–levy 9 deduced from a tidal breakup model. *Nature*, 370(6488):349–351.

Solem, J. C. and Hills, J. G. (1996). Shaping of earth-crossing asteroids by tidal forces. *Astronomical Journal v. 111, p. 1382*, 111:1382.

Song, Y., Turton, R., and Kayihan, F. (2006). Contact detection algorithms for dem simulations of tablet-shaped particles. *Powder Technology*, 161(1):32–40.

Spencer, J., Stern, S., Moore, J., Weaver, H., Singer, K., Olkin, C., Verbiscer, A., McKinnon, W., Parker, J. W., Beyer, R., et al. (2020). The geology and geophysics of kuiper belt object (486958) arrokoth. *Science*, 367(6481):eaay3999.

Stadel, J. G. (2001). *Cosmological N-body simulations and their analysis*. University of Washington.

Stern, S., Bagenal, F., Ennico, K., Gladstone, G., Grundy, W., McKinnon, W., Moore, J., Olkin, C., Spencer, J., Weaver, H., et al. (2015). The pluto system: Initial results from its exploration by new horizons. *Science*, 350(6258):aad1815.

Stern, S., Weaver, H., Spencer, J., Elliott, H., and Team, N. H. (2018). The new horizons kuiper belt extended mission. *Space Science Reviews*, 214:1–23.

Stern, S., Weaver, H., Spencer, J., Olkin, C., Gladstone, G., Grundy, W., Moore, J., Cruikshank, D., Elliott, H., McKinnon, W., et al. (2019). Initial results from the new horizons exploration of 2014 mu69, a small kuiper belt object. *Science*, 364(6441):eaaw9771.

Stern, S., White, O., Grundy, W., Keeney, B., Hofgartner, J., Nesvorný, D., McKinnon, W., Richardson, D., Marohnic, J., Verbiscer, A., et al. (2023). The properties and origin of kuiper belt object arrokoth's large mounds. *The Planetary Science Journal*, 4(9):176.

Stern, S. A. (1996). On the collisional environment, accretion time scales, and architecture of the massive, primordial kuiper belt. *Astronomical Journal v. 112, p. 1203*, 112:1203.

Sugiura, K., Kobayashi, H., and Inutsuka, S.-i. (2020). High-resolution simulations of catastrophic disruptions: Resultant shape distributions. *Planetary and Space Science*, 181:104807.

Tombaugh, C. W. and Moore, P. (2017). *Out of the darkness: The planet pluto*. Stackpole Books.

Tran-Duc, T., Phan-Thien, N., and Khoo, B. C. (2018). A smoothed particle hydrodynamics (sph) study on polydisperse sediment from technical activities on seabed. *Physics of Fluids*, 30(2).

Veras, D., Carter, P. J., Leinhardt, Z. M., and Gänsicke, B. T. (2016). Explaining the variability of wd 1145+ 017 with simulations of asteroid tidal disruption. *Monthly Notices of the Royal Astronomical Society*, page stw2748.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wada, K., Senshu, H., and Matsui, T. (2006). Numerical simulation of impact cratering on granular material. *Icarus*, 180(2):528–545.

Walsh, K. J. (2018). Rubble pile asteroids. *Annual Review of Astronomy and Astrophysics*, 56:593–624.

Walsh, K. J., Ballouz, R.-L., Jawin, E. R., Avdellidou, C., Barnouin, O. S., Bennett, C. A., Bierhaus, E. B., Bos, B. J., Cambioni, S., Connolly Jr, H. C., et al. (2022a). Near-zero cohesion and loose packing of bennu's near subsurface revealed by spacecraft contact. *Science Advances*, 8(27):eabm6229.

Walsh, K. J., Bierhaus, E. B., Lauretta, D. S., Nolan, M. C., Ballouz, R.-L., Bennett, C. A., Jawin, E. R., Barnouin, O. S., Berry, K., Burke, K. N., et al. (2022b). Assessing the sampleability of bennu's surface for the osiris-rex asteroid sample return mission. *Space Science Reviews*, 218(4):1–34.

Walsh, K. J. and Richardson, D. C. (2006). Binary near-earth asteroid formation: Rubble pile model of tidal disruptions. *Icarus*, 180(1):201–216.

Walsh, K. J. and Richardson, D. C. (2008). A steady-state model of nea binaries formed by tidal disruption of gravitational aggregates. *Icarus*, 193(2):553–566.

Walsh, K. J., Richardson, D. C., and Michel, P. (2008). Rotational breakup as the origin of small binary asteroids. *Nature*, 454(7201):188–191.

Walsh, K. J., Richardson, D. C., and Michel, P. (2012). Spin-up of rubble-pile asteroids: disruption, satellite formation, and equilibrium shapes. *Icarus*, 220(2):514–529.

Wandel, O., Kley, W., Schafer, C., Malamud, U., Grishin, E., and Perets, H. (2019). Numerical simulations on the formation of ultima thule. In *EPSC-DPS Joint Meeting 2019*, volume 2019, pages EPSC–DPS2019.

Watanabe, S., Hirabayashi, M., Hirata, N., Hirata, N., Noguchi, R., Shimaki, Y., Ikeda, H., Tatsumi, E., Yoshikawa, M., Kikuchi, S., et al. (2019). Hayabusa2 arrives at the carbonaceous asteroid 162173 ryugu—a spinning top–shaped rubble pile. *Science*, 364(6437):268–272.

Wegner, S., Stannarius, R., Boese, A., Rose, G., Szabo, B., Somfai, E., and Börzsönyi, T. (2014). Effects of grain shape on packing and dilatancy of sheared granular materials. *Soft Matter*, 10(28):5157–5167.

Weidenschilling, S. (2002). On the origin of binary transneptunian objects. *Icarus*, 160(1):212–215.

Weissman, P. R. (1986). Are cometary nuclei primordial rubble piles? *Nature*, 320(6059):242–244.

Wichman, R. and Wood, C. (1995). The davy crater chain: Implications for tidal disruption in the earth-moon system and elsewhere. *Geophysical research letters*, 22(5):583–586.

Yu, Y., Richardson, D. C., Michel, P., Schwartz, S. R., and Ballouz, R.-L. (2014). Numerical predictions of surface effects during the 2029 close approach of asteroid 99942 apophis. *Icarus*, 242:82–96.

Zeng, X., Wen, T., Yu, Y., Cheng, B., and Qiao, D. (2022). New practical discrete non-spherical n-body method: Validation with the brazil nut effect. *Icarus*, 387:115201.

Zhang, Y. and Lin, D. N. (2020). Tidal fragmentation as the origin of 1i/2017 u1 (‘oumuamua). *Nature astronomy*, 4(9):852–860.

Zhang, Y. and Michel, P. (2020). Tidal distortion and disruption of rubble-pile bodies revisited-soft-sphere discrete element analyses. *Astronomy & Astrophysics*, 640:A102.

Zhang, Y., Richardson, D. C., Barnouin, O. S., Maurel, C., Michel, P., Schwartz, S. R., Ballouz, R.-L., Benner, L. A., Naidu, S. P., and Li, J. (2017). Creep stability of the proposed aida mission target 65803 didymos: I. discrete cohesionless granular physics model. *Icarus*, 294:98–123.

Zhang, Y., Richardson, D. C., Barnouin, O. S., Michel, P., Schwartz, S. R., and Ballouz, R.-L. (2018). Rotational failure of rubble-pile bodies: influences of shear and cohesive strengths. *The Astrophysical Journal*, 857(1):15.

Zhao, D., Nezami, E. G., Hashash, Y. M., and Ghaboussi, J. (2006). Three-dimensional discrete element simulation for granular materials. *Engineering Computations*, 23(7):749–770.