

2013

Physical modeling and high-performance GPU computing for characterization, interception, and disruption of hazardous near-Earth objects

Brian Douglas Kaplinger
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Aerospace Engineering Commons](#), and the [Applied Mathematics Commons](#)

Recommended Citation

Kaplinger, Brian Douglas, "Physical modeling and high-performance GPU computing for characterization, interception, and disruption of hazardous near-Earth objects" (2013). *Graduate Theses and Dissertations*. 13204.
<https://lib.dr.iastate.edu/etd/13204>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Physical modeling and high-performance GPU computing for characterization,
interception, and disruption of hazardous near-Earth objects**

by

Brian Douglas Kaplinger

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Program of Study Committee:

Bong Wie, Major Professor

Ping Lu

Zhi Wang

John Basart

Zhijun Wu

Ran Dai

Iowa State University

Ames, Iowa

2013

Copyright © Brian Douglas Kaplinger, 2013. All rights reserved.

DEDICATION

To my parents, Douglas and Dona,
and to my wife, Amber, who is my foundation and guiding light.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1. INTRODUCTION	1
1.1 Mitigation Options	2
1.1.1 Standoff Explosions	2
1.1.2 Subsurface and Surface Explosions	3
1.1.3 Verification and Extension of Previous Work	5
1.2 Impacting Trajectory Analysis	7
1.3 Computational Approach	9
CHAPTER 2. PHYSICAL MODELING	10
2.1 Semianalytical Standoff Model	11
2.1.1 Blast Model	13
2.2 HAIV System Targets	16
2.3 Hydrodynamic Equations	18
2.3.1 Adaptive Smoothed Particle Hydrodynamics	21
2.3.2 Tensor Damage Model	22
2.3.3 Neighbor Finding Implementation	23
2.4 Disruption Mission Profiles	24
2.4.1 Subsurface Explosion Setup	25
2.4.2 Surface Penetrator Model	26
2.4.3 Standoff Energy Deposition	26
2.4.4 HAIV Concept	27

CHAPTER 3. TRAJECTORY ANALYSIS	32
3.1 Impacting Orbit Solver	32
3.1.1 An Adaptive Shooting Method for Reference Impacting Trajectories	34
3.2 Fragmented System Estimation	36
3.2.1 Gravity Model	37
3.2.2 Collision Model	39
3.2.3 Planetary Orbit Models	41
3.3 Earth Rotation Model	42
3.3.1 A Direct Mapping to Earth-Fixed Coordinates	42
3.4 Fragment Reentry Model	44
3.5 Uncertainty Analysis	45
CHAPTER 4. CHARACTERIZATION AND INTERCEPTION	47
4.1 3D Target Polygon Model	47
4.1.1 Camera Pointing and Focus Plane	49
4.1.2 Lighting Conditions	50
4.1.3 Pixel Value Assignment	51
4.2 Line of Sight Vector	53
4.3 Estimation of Target Reference Frame	53
CHAPTER 5. COMPUTATIONAL STRUCTURE	56
5.1 CPU Parallelization	56
5.1.1 Description of Parallel Architectures	57
5.1.2 OpenMP and MPI	58
5.2 GPU Computing	59
5.2.1 Hardware and Implementation	60
5.2.2 Hydrodynamic Calculations	60
5.2.3 Orbital Calculations	62
5.2.4 Memory Model and Explicit Communication	63
5.2.5 Bandwidth Use and Serial Computation	64

5.2.6	Integration with SMP Computation	65
CHAPTER 6.	RESULTS	66
6.1	Semianalytical Standoff Model	66
6.2	Disruption Results	68
6.3	Nominal Fragmentation Behavior	70
6.3.1	Timing and Scaling Parameter Variation	75
6.3.2	Reentry Modeling	76
6.4	Computational Optimization	77
6.4.1	Performance	79
6.5	Optimal Mission Results	80
6.6	Characterization Results	82
6.7	Summary and Discussion	82
BIBLIOGRAPHY	85

LIST OF TABLES

Table 1.1	Orbital Parameters for AIAA Impact Deflection Problem	8
Table 2.1	Parameters for Tillotson Equation of State in Core Material	21
Table 3.1	Osculating Orbital Parameters for Fictitious Impact Trajectory	35
Table 3.2	Orbital Parameters for Planetary Ephemerides [1]	40
Table 3.3	Parameter Rates for Planetary Ephemerides [1]	40
Table 3.4	Coefficients for Static Atmosphere Model	44
Table 4.1	Simulated Camera Parameters	50
Table 5.1	Hardware for Benchmark Systems	61
Table 6.1	Number and Mass Ratio of Impacting Fragments	72
Table 6.2	Number and Mass Ratio of Impacting Fragments after 1 Orbit	72
Table 6.3	Impacting Fragments for 1 MT Disruption after 15 day Dispersion	72
Table 6.4	Reentry Modeling Results	77

LIST OF FIGURES

Figure 1.1	Structural Model of an Apophis-sized (270 m) Body.	6
Figure 1.2	Distribution of Fragments and Velocities for 300 kT Subsurface Explosion.	6
Figure 2.1	Basic geometry and area fraction dependence for standoff approach.	11
Figure 2.2	Irradiation shell of angle $d\phi$	13
Figure 2.3	Asymmetric Target for 2D Penetrated Explosive Modeling.	17
Figure 2.4	Description of Sorted Neighbor Kernel Process.	24
Figure 2.5	Neighbor Search Cost.	25
Figure 2.6	Subsurface Explosion and Resulting Fragment Velocities.	25
Figure 2.7	Radial Energy Deposition and Total Deposition Region.	26
Figure 2.8	SPH Nodes and Resulting Ablation for Standoff Model.	27
Figure 2.9	Radial Dispersion Velocity Histogram for HAIV Concept.	28
Figure 2.10	Asymmetric Shock Behavior.	29
Figure 2.11	Example Damage Localization for Tensor Fracture Model.	30
Figure 2.12	Final Disruption of NEO Target.	30
Figure 2.13	Location of Slowest Moving Debris.	31
Figure 2.14	Radial Dispersion Velocity Histogram for Contact Burst.	31
Figure 3.1	Histograms of Known NEO Population.	33
Figure 3.2	Convergence History of Error Function Norm for 11-body Model.	35
Figure 3.3	Cumulative Density Functions for Disrupted Asteroid.	36
Figure 3.4	Rotating Local-Vertical-Local-Horizontal (LVLH) Frame.	36
Figure 3.5	Temporary Grid Creation and Evaluation.	38
Figure 3.6	Block Diagram for Self-Gravity Process Logic (Contains Fragment i).	39

Figure 3.7	Collision Detection and Evaluation Process for Interacting Pair.	41
Figure 4.1	Surface Model for Eros.	47
Figure 4.2	Facet Geometry Definitions.	48
Figure 4.3	Parallel Trim Reduction Technique.	52
Figure 5.1	Comparison of SMP and DMP Architecture	57
Figure 5.2	Example of a Composite SMP/DMP Architecture	58
Figure 5.3	Visualization of Parameter Sweep Method.	62
Figure 5.4	Block Diagram of Simulation Procedure.	63
Figure 5.5	Visualization of Memory Model.	64
Figure 6.1	Comparison of velocity changes for momentum coupling and solid body model.	67
Figure 6.2	Relative Performance for Surface Impactor.	68
Figure 6.3	Impacting Mass for Subsurface Explosion on Orbits with Varying Inclination.	69
Figure 6.4	Impacting Mass Comparison for Subsurface and Dynamic Surface Cases.	70
Figure 6.5	Mean Ejecta Velocity for Single and Double Impactor Cases.	71
Figure 6.6	Miss Distance Histograms for 1 Hour Maximum Time Step Simulations.	73
Figure 6.7	Miss Distance Histograms for 1 Minute Maximum Time Step Simulations.	74
Figure 6.8	Comparison to Matlab Solver using Varied Time Steps.	74
Figure 6.9	Impact Locations on a Fixed Earth after Radial and Transverse Deflections.	75
Figure 6.10	Impacting Mass Fraction Contours for Low Lead Time Cases.	75
Figure 6.11	Analytical Approximation of Minimal Successful Mission Parameters.	76
Figure 6.12	Reentry Stress Distribution and Velocity Profile with Altitude for Example Disrupted and Impacting Fragments.	77
Figure 6.13	Comparison of Single-Node Performance on CPU and GPU.	80
Figure 6.14	Cost Function Contours for Sample Mission Approach Asymptotes.	81

Figure 6.15	Surface Estimation using limited LIDAR Measurements.	82
Figure 6.16	Convergence of Mass Estimate, Shape Parameters, and Orientation. . .	83

CHAPTER 1. INTRODUCTION

For the past few decades, both the scientific community and the general public have been becoming more aware that the Earth lives in a shooting gallery of small objects. We classify all of these asteroids and comets, known or unknown, that cross Earth's orbit as near-Earth objects (NEOs). A cursory look at our geologic history tells us that NEOs have collided with Earth in the past, and we expect that they will continue to do so. On average, our scientific community estimates that a near-Earth object with a diameter of up to 100 meters would impact the Earth once every one hundred to one thousand years [2], a notable example of which is the impact at Tunguska in 1908. Even an impact of this size would cause catastrophic damage if it occurred over an urban area. A similar fireball event occurring in the New York City metro area could result in up to 10 million casualties [3]. Fears of this possibility are heightened by the explosion of a small NEO (approximately 20 meters in diameter) over Chelyabinsk, Russia on February 15, 2013. This event released approximately 440 kilotons of energy in the upper atmosphere [4], injuring over 1000 people. Additionally, though the population of catastrophic impactors has been well surveyed, it is estimated that thousands of bodies over 140 meters in diameter remain undiscovered. It is estimated that the most likely NEO threat are airburst events due to asteroids on the lower end of this spectrum, formally considered too small to survive atmospheric entry [5]. Impacts of NEOs with diameters over 1 km are less probable, but even bodies at the lower end of this category would cause a global catastrophe resulting in near extinction of the human population. A 10-km asteroid is suspected to have resulted in the extinction of over half of the species on Earth late in the Cretaceous period 65 million years ago.

1.1 Mitigation Options

With thousands of known NEOs crossing the orbit of Earth, and almost 1400 of these considered potentially hazardous at this time [6], there has been significant scientific interest in developing the capability to deflect an NEO from an impacting trajectory. With a large number of such bodies crossing the orbit of Earth, the probability of a catastrophic impact is large enough to warrant careful monitoring of hazardous NEOs and research into the possible deflection of their orbit. For the last few decades, the scientific community has been actively engaged in identification and characterization of hazardous NEOs [2]. Conventional wisdom, and a substantial amount of planetary defense literature, has held that the best way to accomplish this goal is to slowly push the NEO onto another trajectory, or to use an impulsive force (nuclear standoff explosion or kinetic impactor) at least a decade in advance of impact [7]. In practice, deflection methods of sufficiently high energy density are preferred and need to be prepared in advance of an expected impact date with the Earth [7, 8, 9, 10, 11].

1.1.1 Standoff Explosions

Past results for low-energy methods indicate that compression waves created from kinetic impactors or incident radiation are not sufficient to fragment these NEOs, assuming they are solid bodies. Recent characterization research, however, suggests that many of these bodies have porous outer shells and that catastrophic fragmentation due to explosives is a reasonable concern [12]. In fact, calculations of specific energy imparted from nuclear explosives have been shown to exceed the gravitational binding energy of common NEOs, resulting in long-term dispersion of fragments along the orbital trajectory [13]. One of the proposed literature methods utilizes a nuclear explosion at a specified standoff distance from the target NEO to cause its velocity change by ablating and blowing off a thin layer of the surface [7, 8, 14, 15, 16]. This thesis investigates a simple model that can be used to assess the effectiveness of a nuclear standoff explosion approach. We use geometric principles and basic physics to construct a baseline model. This approach can be augmented to account for icy bodies, anisotropic ejecta distributions, and effects unique to the nuclear blast model. Possible fracturing of the

asteroid and other anticipated outcomes of a nuclear blast are analyzed. Use of this simple model has resulted in an estimation of NEO velocity change (ΔV) on the same order as other complex models, and has correlated data for optimal standoff distance of about 200 m for an ideal spherical model of a 1-km NEO. The standoff approach is further investigated from a hydrodynamics standpoint as a method for disruption of so-called “rubble piles” or small NEOs.

1.1.2 Subsurface and Surface Explosions

While these methods are intended to ablate a thin layer of material, the level of energy imparted to the NEO makes fragmentation of the target a plausible outcome. The specific energy added to the basaltic rock can approach or even exceed the energy needed for minimal break up of a small NEO [12]. In other cases, thermal ablation of surface material causes compression waves to propagate through the remainder of the NEO. The stress of these waves may be enough to continue fracturing the material [17]. Another proposed mitigation method is the use of nuclear explosive devices above, on, or beneath the surface of an NEO [18, 19] It has been suggested that fracturing a body may be a beneficial outcome, as smaller pieces may burn up in the atmosphere. However, even smaller pieces impacting the Earth can cause significant damage [5, 20]. The lower threshold for material ablation in the atmosphere is undetermined, and explosions in the atmosphere can be fatal and possibly catastrophic events.

A possible benefit of NEO fragmentation is to lower the number of small bodies impacting the Earth in cases where some level of impact is inevitable. This could be either the result of an unsuccessful deflection attempt or a backup measure when there is not enough time for another deflection mission. Previous research has shown a reduction in impacting mass of up to 80% through statistical methods [12]. Lead time, or warning time, has been suggested as the most important factor in the effectiveness of catastrophically fragmenting a NEO [12, 17]. However, Dave Dearborn proposed a radical idea that with sufficient energy an object typically considered catastrophic could be dispersed in as little as 10 days to impact [19, 13]. This proof of concept was initially tested through hydrodynamic methods at Lawrence Livermore National Laboratory.

Two main motives drive research into high-energy, last minute options for asteroid deflection: a) Many bodies on impacting trajectories may not be detected early enough to spend decades deflecting their orbits, and b) The expense and risk of an interplanetary mission to deflect an NEO will require that an object be demonstrated to pose a substantial and imminent threat before action is taken. While the late notice motive is becoming less relevant as more NEOs are catalogued, there are still small bodies that are detected within weeks of closest approach, or are even a complete surprise [4], and it would be hard to prove that no more such bodies exist.

The “late decision” motive is a political and fiscal reality that planetary defense researchers will have to consider. Even if a viable method of deflection is demonstrated and awaits deployment, a potential target must be shown to impact the Earth considering all relevant uncertainties. As the long-term ephemerides of a small body in the solar system contain many uncertainties that could account for several Earth radii of displacement [13, 21], we expect that a decision to deflect an NEO will come within the last few orbits before impact (possibly after a close approach with the Earth alters the body trajectory). The asteroid 99942 Apophis is one such example. Initially given a much higher probability of impact on April 13, 2036, the current estimated probability of impact is 1 out of 250,000 after many additional observations. There is still much uncertainty of what will happen after Apophis passes within 36,000 km of the Earth on April 13, 2029 [22].

If a body like Apophis was forced into a resonant orbit after a close approach with the Earth and was confirmed to be on an impacting trajectory, we would only have a short window in which to act (7 years in the Apophis scenario). This has been shown to be problematic for mission design, with reasonable launch windows only giving a few weeks of time for a deflection to take place [23]. Modeling of atmospheric reentry for a fragmented body has suggested that lowering the individual masses results in substantial reduction of ground impacts, with many fragments burning up or being partially ablated by the atmosphere [24]. Therefore, it is desired that any viable last-minute option lower the impacting mass below this threshold, allowing the atmosphere to have an increased effect. For this reason, we investigate the timing of NEO fragmentation and the scaling of initial explosive energy to model NEO disruption and

determine a desired course of action in a late notice or late decision scenario.

This thesis applies the ideas of Smoothed Particle Hydrodynamics (SPH) theory to the NEO disruption problem. A simulation package was designed that allows efficacy simulation to be integrated into the mission planning and design process. This is done by applying ideas in high-performance computing (HPC) on the computer graphics processing unit (GPU). Rather than prove a concept through large standalone simulations on a supercomputer, a highly parallel structure allows for flexible, target dependent questions to be resolved. Built around non-classified data and analysis, this computer package will allow academic institutions to better tackle the issue of NEO mitigation effectiveness.

1.1.3 Verification and Extension of Previous Work

Previous results in the targeted disruption of NEOs were conducted by David Dearborn of Lawrence Livermore National Laboratory. Work completed at the Asteroid Deflection Research Center at Iowa State University verified the orbital trajectory analysis of Dr. Dearborn. The target of the hypothetical deflection mission was an NEO approximately the size of the asteroid Apophis. It has a total mass of 2.058×10^{13} kg with a diameter of 270 meters, as shown in Figure 1.1. It has a two-component (inhomogeneous) spherical structure with a high density core consistent with granite (density = 2.63 g/cm^3), and a lower density (1.91 g/cm^3) mantle. The bulk density of the structure was 1.99 g/cm^3 , close to that measured for asteroid Itokawa (density = 1.95 g/cm^3) [25]. A nuclear explosion was simulated in a region below the surface of the body by sourcing in energy corresponding to 300 kT. The source region is cylindrical, and the dimensions are 1 m in diameter and 5 m long. The energy source region expands, creating a shock that propagates through the body resulting in fragmentation and dispersal. The structure of the asteroid was modeled with a linear strength model, a core yield strength of 14.6 MPa, and a shear modulus of 35 MPa. The mass-averaged speed of the fragments after 6 seconds was near 50 m/s with peak near 30 m/s, as shown in Figure 1.2. A three-dimensional fragment distribution was constructed from the hydrodynamics model by rotating the position, speed, and mass of each zone to a randomly assigned azimuth about the axis of symmetry. Figure 1.2 also shows the two-dimensional distribution of body fragments after the completion

of the subsurface explosion simulation.

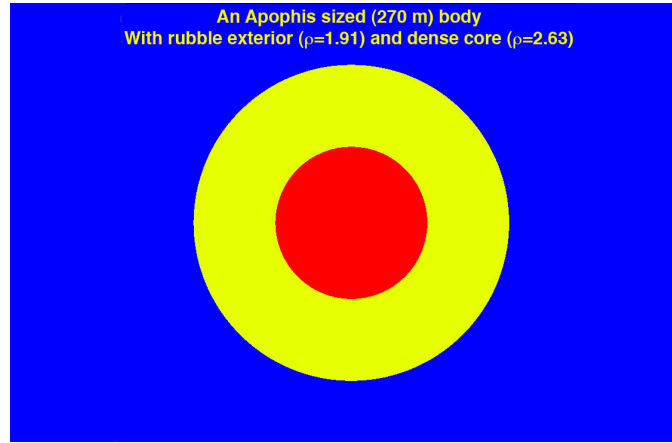


Figure 1.1 Structural Model of an Apophis-sized (270 m) Body.

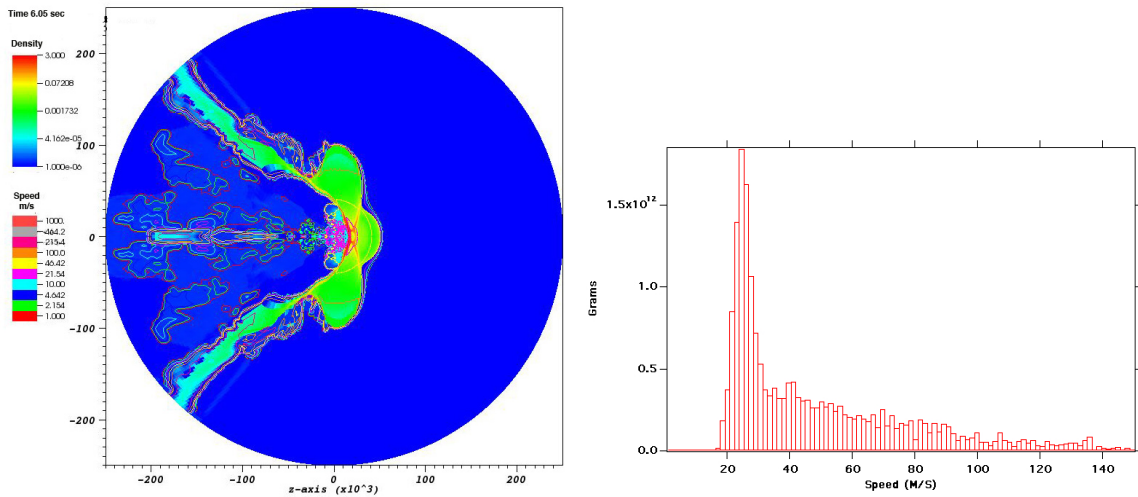


Figure 1.2 Distribution of Fragments and Velocities for 300 kT Subsurface Explosion.

We now have a three-dimensional distribution of relative position and velocity for a reasonable NEO fragmentation. To track the dispersion of these fragments along the orbital trajectory before impact, the relative position axes were aligned such that the highest momentum projectiles coincided with the desired deflection direction at the time of the explosion. A velocity scaling parameter allows the testing of distributions with fragment velocities different than the predicted 50 m/s maximum velocity from the explosion simulation.

1.2 Impacting Trajectory Analysis

In order to study a viable deflection mission, we must first have an orbit that can be assumed to impact the Earth if no action is taken. Newtonian n-body system simulation is simple in principle because it involves the integration of the ordinary differential equations describing the n-body motions in a Newtonian gravitational field. However, it becomes a non-trivial problem when a precise orbit prediction is required in the presence of various physical modeling uncertainties. An example case (the AIAA fictional asteroid impact problem) will be described in the next section to highlight this concern. A linearized least squares error correction algorithm was used to develop a fictitious orbital trajectory of the asteroid Apophis, precisely impacting the Earth on April 13, 2036. The estimated orbital parameters for the asteroid Apophis following its encounter with the Earth in 2029 were used as an initial guess. This numerical algorithm for the nonlinear shooting method proposed was shown to be superior in computational efficiency to gradient-based line search methods or random walk methods for this type of problem. The orbital analysis chapter will describe the search process for finding orbital parameters that intersect the position of the Earth at the appointed time, as well as the models used for successively higher fidelity simulation of fragment relative positions.

A fictional asteroid deflection problem was created by AIAA (American Institute of Aeronautics and Astronautics) in 2004 [21]. A similar fictional asteroid deflection problem, called the Defined Threat (DEFT) scenarios, has been also created for the 2004 Planetary Defense Conference. One of the four DEFT scenarios is about mitigating a fictional 200-m Athos asteroid with the predicted impact date of February 29, 2016. The fictional asteroid mitigation problem of AIAA is briefly described as follows. On July 4, 2004, NASA/JPL's Near Earth Asteroid Tracking (NEAT) camera at the Maui Space Surveillance Site discovered a 200-m diameter Apollo asteroid designated 2004WR. This asteroid has been assigned a Torino Impact Scale rating of 9.0 on the basis of subsequent observations that indicate there is a 95% probability that 2004WR will impact the Earth. The expected impact will occur in the Southern Hemisphere on January 14, 2015 causing catastrophic damage throughout the Pacific region. The mission is to design a space system that can rendezvous with 2004WR in a timely manner,

Table 1.1 Orbital Parameters for AIAA Impact Deflection Problem

Orbital Parameter	Value
Semimajor Axis	2.15374076 AU
Eccentricity	0.649820926
Inclination	11.6660258 deg
Longitude of Right Ascension	114.4749665 deg
Argument of Perihelion	66.2021796 deg
Mean Anomaly	229.8987151 deg
Epoch	53200 MJD

inspect it, and remove the hazard to Earth by changing its orbit and/or destroying it. The classical orbital elements of 2004WR are given in the J2000 heliocentric ecliptic reference frame by the values in Table 1.1.

The STK 5.0.4 software package, with a 9th-order Runge-Kutta integrator with variable step size and the planetary positions from JPL’s DE405, was known to have been used by AIAA to create this set of orbital parameters of 2004WR. It is further assumed that 2004WR is an S-class (stony-silicate) asteroid with a density of $2.720\text{E}3 \text{ kg/m}^3$ and that its estimated mass is $1.1\text{E}10 \text{ kg}$. If 2004WR is an M-class (nickel-iron) asteroid, then its estimated mass would be $2.2\text{E}10 \text{ kg}$.

An ideal Keplerian orbit simulation of 2004WR was performed first as a simple check of this fictional problem in —citewie4. The result indicated that its closest approach to Earth is about 0.035 AU, which is less than the Minimum Orbit Intersection Distance (MOID) of 0.05 AU of a Potentially Hazardous Object (PHO). It also had a close encounter with Mars by 0.1 AU. After checking the ideal orbital characteristics of 2004WR, three different N-body software packages were used to confirm 2004WR’s collision with Earth on January 14, 2015. These software packages were: JPL’s Horizons, CODES, SSCT, and STK all utilizing JPL’s DE405 ephemeris data for the planetary positions. Orbit simulation results of using these four N-body simulators indicate that 2004WR actually misses Earth by 1.6 Earth Radii contrary to an expected impact in the Southern Hemisphere on January 14, 2015. This example problem will be further examined using the reference trajectory method developed in this work.

Initial work in planetary deflection was focused mostly on prediction of relative impacting mass, but disruption at different times along a given orbit can have a large effect on the resulting shape of debris. The proposed approach looks at the fragmentation model to better address how uncertainty in the NEO breakup affects orbital prediction, particularly in the case of variable time-to-impact. This allows for a more clear set of objectives for mission design. Another new result is the availability of representative 3D fragment distributions for non-spherical bodies. This will improve the trajectory of the desired hypervelocity intercept mission by allowing full degrees of freedom in choosing the approach asymptote.

1.3 Computational Approach

A bottleneck in determining appropriate mitigation methods for NEOs has been a lack of experimental data on the efficacy of each approach, forcing reliance on simulations to determine effectiveness. As we move from the concept stage into true mission planning for effective NEO threat mitigation, we must depart from simulation of a few sample cases and instead use mission parameters to integrate modeling and simulation into the design cycle. This thesis presents the development of simulation tools designed to be implemented as part of the mission design procedure for nuclear fragmentation and dispersion of an NEO. A brief history of GPU computing will be given, followed by the particulars of high-level language access for this simulation. Motivation for the parallelization of the presented model lies in each particle relying only on information for its immediate neighbors. Improvements of the fragmentation model are shown to result in 60% cost savings for the simulation and a large speedup compared to serial CPU implementation. The adaptation of previously presented models to the memory and compute capability of the GPU architecture will be described, as well as steps taken to optimize performance in the presence of GPU limitations.

Past work showed that a large amount of data can be processed using GPU simulation. This has allowed for a revolution in computing on a budget, allowing hundreds of complex simulations to be tested. While new HPC technology is shown to solve old problems faster, this work also addresses the identification of new problems that were previously intractable without the use of a supercomputer or dedicated cluster.

CHAPTER 2. PHYSICAL MODELING

This chapter presents the equations of motion and target model used in the fragmentation simulations. For initial representation of the standoff model, a semi-analytical approach is taken. For the axisymmetric 2D simulation work, two primary reference targets are used, to emphasize the differences between material composition. Both were 100 meters in diameter, but had different bulk densities and material strength properties. The first target is a rubble-pile asteroid, with a bulk density of 1.91 g/cm^3 . This is a likely target for demonstrating the behavior of more porous material. The second target is a single granite boulder with a bulk density of 2.63 g/cm^3 . A linear model for material strength is used in this target with a yield strength of 14.6 MPa and a shear modulus of 35 MPa, resulting in a more granulated fragmentation and slower dispersion velocities. Real asteroid targets are expected to fall within these two extremes, with variances for composition, distribution of mass, and orientation. A Smoothed Particle Hydrodynamics (SPH) model is used for the asteroid fragmentation simulation under 3 initial conditions: a subsurface explosion of 100 kt buried at a 5 m depth, a surface blast of 100 kt surrounded by a 1 m thick aluminum impactor, and a standoff blast at 10 m above the surface. We assumed an isotropic Weibull distribution of implicit flaws in the NEO material and conducted Monte Carlo simulation to establish a mean response of the target NEO to the fragmentation process for the initial model.

An asymmetric target was created to demonstrate the effectiveness of a two-body approach. This Hypervelocity Asteroid Interception Vehicle (HAIV) concept by the Asteroid Deflection Research Center is intended to overcome some of the technological challenges introduced by attempting a subsurface explosion. Since rendezvous with a target is not generally an option for short lead time [23], a spacecraft designed to separate into two bodies is employed. The first acts as a kinetic impactor at the relative velocity of 5 - 30 km/s. The second contains the

explosive device, which detonates in the crater of the initial impact. This mimics the behavior of a subsurface explosion by increasing the coupled energy, without requiring a velocity change for the spacecraft.

2.1 Semianalytical Standoff Model

The precise outcome of a NEO deflection attempt using a nuclear standoff explosion is dependent on myriad variables. Shape and composition of the target NEO are critical factors. These critical properties, plus others, would need to be characterized, ideally by a separate mission, prior to a successful nuclear deflection attempt. A first-order model can be constructed by assuming a spherical NEO of constant density. A simple representation is shown in Figure 2.1, where R is the radius of the target NEO, h is the blast height above the surface, k is the ratio of blast height to radius, and r is the straight-line distance from blast to surface. The distance r can be expressed in terms of a function b , which is the ratio of the distance r to the radius R (i.e., $b = r/R$). As illustrated in Figure 2.1, α is half the apex angle of the cone intercepting the NEO surface, ψ is the complement of this angle, and is the incident angle of radiation from the blast, measured from the vertical. The energy intensity on the surface is dependent on the distance traveled. The distance r at any angle from the horizontal can be determined using the law of cosines, as follows:

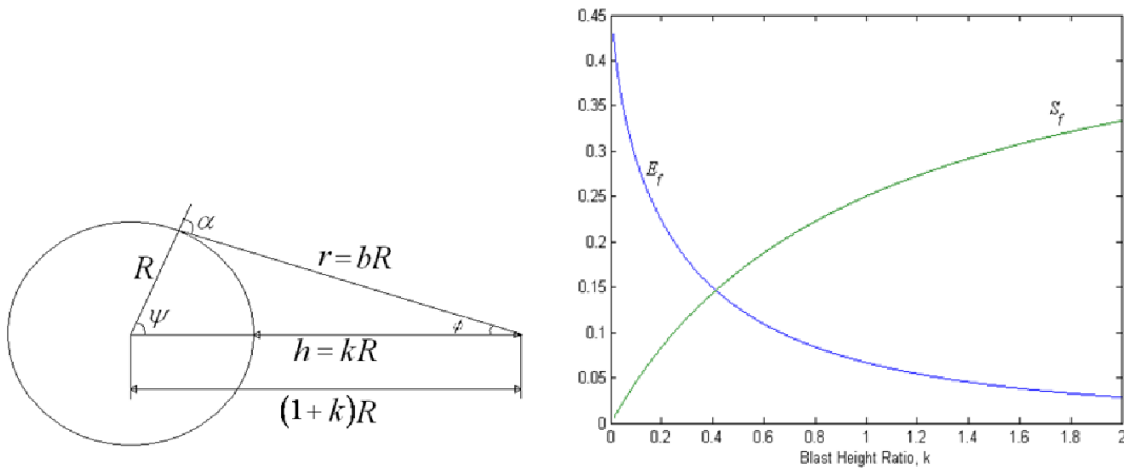


Figure 2.1 Basic geometry and area fraction dependence for standoff approach.

$$r^2 = b^2 R^2 = R^2 [k^2 + 2(k+1)(1 - \cos \psi)] \quad (2.1)$$

The solid angle subtended by any cone with apex angle 2ϕ is given by $2\pi(1 - \cos \phi)$ while the total solid angle of a sphere is 4π . Therefore the fraction of blast energy intercepting the surface, E_f , of the NEO and the fraction of the NEO surface irradiated, S_f , are determined by their respective maximum angles, as follows:

$$E_f = \frac{1}{2}(1 - \cos \phi_{\max}), \quad S_f = \frac{1}{2}(1 - \cos \psi_{\max}) \quad (2.2)$$

where ϕ_{\max} and ψ_{\max} are defined for r tangential to the spherical surface. These relations will be used to assess the importance of blast height and determine the energy coupling available for any deflection attempt. It is apparent from Figure 2.1 that the amount of available energy as well as the amount of surface irradiated is dependent on the precise blast height. The ratios of area subtended on a unit sphere for each blast height are also shown in Figure 2.1. These values are important in determining the energy density at any point on the NEO from the blast. The equation for r describes the distance dependence on blast height. The angles ϕ_{\max} and ψ_{\max} are also dependent upon the blast height and can be determined from the right triangle in Figure 2.1 as

$$\phi_{\max} = \sin^{-1} \left(\frac{1}{1+k} \right), \quad \psi_{\max} = \cos^{-1} \left(\frac{1}{1+k} \right) \quad (2.3)$$

We then obtain the following relationships for energy and surface fractions:

$$E_f = \frac{1}{2} \left(1 - \frac{\sqrt{2k+k^2}}{1+k} \right), \quad S_f = \frac{1}{2} \left(1 - \frac{1}{1+k} \right) \quad (2.4)$$

The energy densitys dependence on angle from the blast can be described using the previous equations. Figure 2.2 shows the relationships used to construct this dependence. Using Figure 2.2, we obtain the fraction of energy present in a thin conical shell with angular width, $d\phi$, at an arbitrary value of ϕ as follows:

$$dE = \frac{1}{2} \sin \phi d\phi \quad (2.5)$$

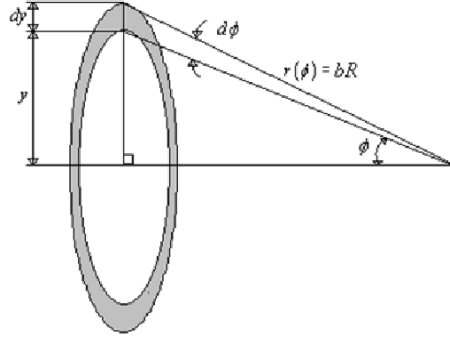


Figure 2.2 Irradiation shell of angle $d\phi$.

The impact area of this thin shell is also obtained as

$$dS = \frac{2\pi y dy}{4\pi R^2} = \frac{1}{4} b^2 \sin 2\phi d\phi \quad (2.6)$$

where $b = r/R$. As expressed earlier, b has a relationship to blast height as $b^2 = k^2 + 2(1 + k)(1 - \cos \psi)$. The area energy density fraction is then described as

$$\frac{dE}{dS} = \frac{1}{b^2 \cos \phi} \quad (2.7)$$

2.1.1 Blast Model

To assess the effects of different nuclear devices, we assume a modular approach in which the blast model can be interchanged to produce alternate results. As a first step in this process, a symmetric, booster-principle device is assumed. This nuclear device is modeled as a spherical mass of thermonuclear material surrounded by concentric spheres of fissile material and other components including shaped high explosives and detonation equipment. This model has the benefit of isotropic energy distribution, but is limited in yield due to a constrained volume of thermonuclear material [26].

Several basic fuels can be used in a thermonuclear device, among them combinations of deuterium (D) and tritium (T). Both of these are heavy isotopes of hydrogen, with deuterium having 2 nucleons and tritium having 3. The spherical configuration of our model makes it difficult to consistently obtain the ignition temperature for the pure D-D reaction involving

deuterium nuclei, while using a D-T reaction is problematic for systemic reasons [26]. A compromise can be made, however, using a compound of lithium including deuterium, ${}^6\text{LiD}$. This compound is solid at room temperature, and upon neutron bombardment creates tritium nuclei to sustain a D-T reaction [15, 26, 27]. A thermonuclear device based on ${}^6\text{LiD}$ requires a constant flux of neutrons beyond what is made available in the surrounding fission reactions. Therefore, the fuel is likely stored with a tamp material that breeds additional neutrons, such as ${}^{238}\text{U}$ (an isotope of uranium) or ${}^9\text{Be}$ (an isotope of beryllium) [26, 27].

A modern nuclear bomb known as the B83, which is listed as weighing 1100 kg, has a yield of around 1.2 MtTNT. A unit of tTNT is used to describe an energy yield equivalent to 1 ton of TNT explosive, which has a standard value of 4.184×10^9 J. Empirically, U.S. weapon masses have been proportional to the 0.85 power of the effective yield [15]. Determining a yield for a general weapon requires inverting this relationship. For the B83, an approximate formula for radiative yield, Y (in units of J), in terms of the weapon mass, M (in units of kg), is given by

$$Y = 1.1 \times 10^{12} M^{1.2} \quad (2.8)$$

The components and equipment needed to detonate the nuclear device reduce the effective yield to about 0.25 of the yield of an equivalent mass of fuel [28]. This difference in specific energy can be used to find the fuel mass fraction, m_f , as well as the resulting fractional mass of debris, m_d . The values of $m_f = 0.25$ and $m_d = 0.75$ derived in [7] are assumed here.

For the nuclear device described here, the fission reaction heats the thermonuclear material well above its ionization temperature, so the fuel becomes plasma [27]. This process can give an approximate value for the temperature, $T = 1.2 \times 10^8$, in degrees Kelvin based on a tenfold increase in material density during the fusion process [26]. The speed of sound in plasma dominated by blackbody radiation is the prevailing factor in a kinetic energy model of bomb debris and the rate at which freed neutrons excite reactions [26, 27], and it is described by

$$c_s = \sqrt{\frac{4}{3} \frac{aT^4}{\rho_2}} \quad (2.9)$$

where a is the radiation constant equal to $7.57 \times 10^{-16} \text{ Jm}^{-3} \text{ K}^{-4}$. Assuming a tenfold

increase in density, ρ_2 , to about 7800 kg/m^3 [26], the speed of sound can be estimated as $c_s \approx 5.2 \times 10^6 \text{ m/s}$. This calculated value is only slightly lower than the measured speed of a thermonuclear detonation wave in ${}^6\text{LiD}$, so it provides a good estimate of the kinetic energy delivered [26]. The model described in this paper assumes the thermonuclear detonation to be an impulsive event, taking place instantaneously. Future models, however, may make use of a significant quantity computed from the detonation wave speed known as the hydrodynamic disassembly time. This is the time it takes for the energy of the detonation wave to blow away any remaining fuel without ignition. For the value of c_s computed, this is approximately $5 \times 10^{-8} \text{ s}$ [26]. If the fraction m_f of the total mass is accelerated from rest to the speed c_s , the overall increase in kinetic energy K (in units of J) becomes

$$K = \frac{1}{2}m_f M c_s^2 \quad (2.10)$$

The neutron yield of a 1 MtTNT thermonuclear reaction is approximately 3×10^{26} neutrons [26]. At energies of 17.6 MeV each ($2.82 \times 10^{-12} \text{ J}$), this results in a total neutron energy release of $8.46 \times 10^{14} \text{ J}$, just over 20% of the overall yield. The yield is computed from a flux, and thus is proportional to the surface area of the fuel [26]. Changing the mass of the weapon results in a change of fuel radius. Assuming the relationship described, we obtain a corresponding value of neutron energy, N (in units of J), as a function of weapon mass, M (in units of kg), as follows:

$$N = 8.8 \times 10^{12} M^{0.67} \quad (2.11)$$

The overall energy distribution from the detonation can be approximated by a blackbody radiation curve [27]. We can assume bomb components and fissile material are opaque to almost all wavelengths above the X-ray part of the electromagnetic spectrum [27]. The resulting intensity of radiation in the X-ray part of the spectrum can be obtained by integrating Plancks equation [29] between the limits of 30 PHz and 30 EHz. This corresponds to 99.8% of the total radiative yield, while the remaining 0.2% is predominantly gamma rays [27]. In this paper, the

distribution of blast energy is assumed as

$$E_{\text{total}} = E_{\text{x-ray}} + E_{\gamma} + E_{\text{neutron}} = 0.998(Y - N) + 0.002(Y - N) + N \quad (2.12)$$

2.2 HAIV System Targets

Initial demonstration of the two body HAIV concept used spherical spacecraft dummy payloads to hit an inhomogeneous target with a diameter of 54 m. This method was directly compared to a single explosion on contact with the surface. The current asymmetric target consists of a contact binary system with a rubble pile exterior. With binary systems comprising about 16% of the known NEA population [30], an impactor mission faces an approximately 1 in 6 chance that the target it approaches will be a binary system. This is a characteristic that will be unable to be predicted ahead of time without radar observation, in the case of systems with close secondaries. It has been suggested that many irregularly shaped asteroids with unusual spin states could be contact binary (or multiple) systems. These types of systems would exhibit some of the same characteristics as monolithic rocks and as rubble piles [31]. Further, those asteroids identified as rubble piles could have large solid components beneath their regolith.

The two cores of the model system are elliptical, with major and minor axes of 50 and 30 meters, respectively. These cores are given material properties similar to granite using a linear elastic-plastic strength model, and are canted by 45 degrees relative to the horizontal. There is a vertical line of symmetry, so the cores are mirror images of one another. A rubble regolith extends 2 meters in depth vertically above each core, and is packed along lines of constant potential around the body, resulting in a maximum regolith depth of 14 meters. These properties result in exterior dimensions of the target being approximately 76 x 42 meters, as shown in Figure 2.3. The inner half of each core has an initial bulk density of 2630 kg/m³, while the outer portion of the core is more porous material with an average bulk density of 1910 kg/m³. Both sections use values for yield strength between 7-203 MPa and shear modulus between 8-22 MPa.

The initial impactor of the two-body spacecraft is an aluminum wedge 1 m in base diameter and 1.5 m in length. The nuclear payload follows, depositing 70 kilotons of energy upon reaching

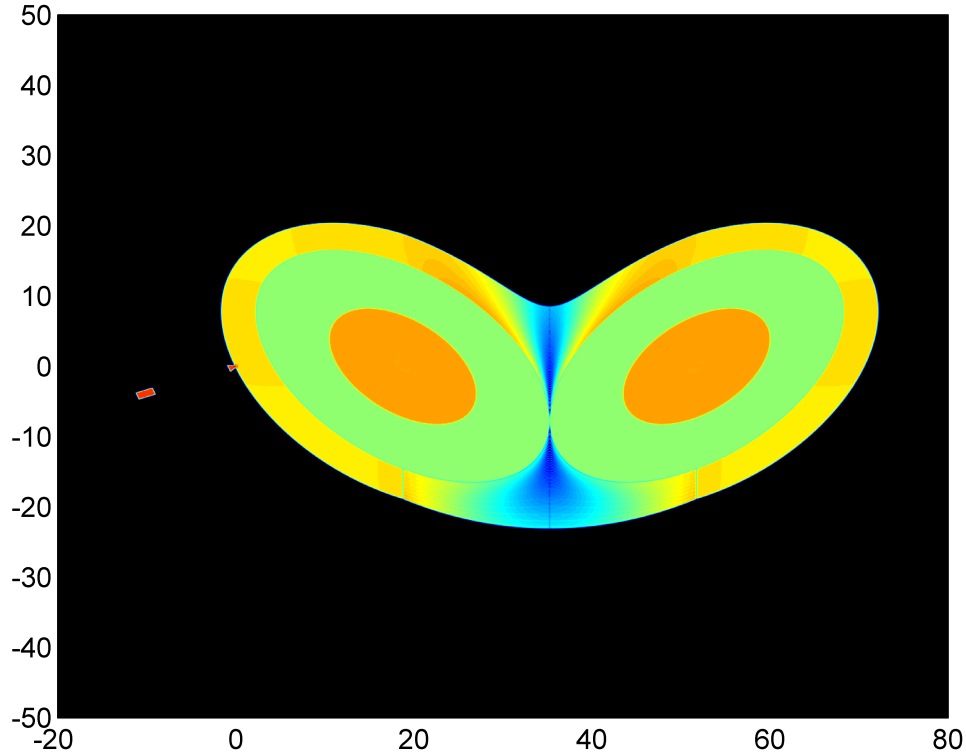


Figure 2.3 Asymmetric Target for 2D Penetrated Explosive Modeling.

the initial impact site of the lead body. Most of this energy is absorbed in the crater region formed by the initial impact, though deeper absorption is allowed due to the fact that much of the material in this region has already been vaporized and superheated into a plasma state. The resulting shock wave has a peak compression of more than 2 times the initial density, and quickly overtakes the initial shock of the lead body impact, which is much weaker. This shock compresses much of one core far beyond the fracture strength of even the worst case material, rebounding off of the nearer side. This asymmetric behavior dissipates some energy due to interactions with the rebounding shock front. In the center area of deeper regolith, the seeding process naturally results in a much more porous material, absorbing energy from the shock. Upon reaching the second core at the far side, some large chunks escape the disruption process in some cases (even with lower material strengths).

In addition to the equations of motion used for the previous simulation, an extra dimension was added. This was to investigate the potential for sources of errors in 2D cylindrical Smoothed Particle Hydrodynamics (SPH) codes, rather than an axisymmetric model. Other

than in increase to the complexity of neighbor-finding calculations (as discussed later), no significant increase in code complexity was required. This is due to the fact that the SPH model equations are originally a 3D component/tensor formulation. Resulting coherent masses from the fragmentation process were propagated through a model of solar system dynamics until the predetermined date of impact. Masses remaining on impact trajectories undergo a simulation of reentry into Earth’s atmosphere, resulting in final tallies of mass missing the Earth, fragments on capture trajectories, airburst events, and impacts of reduced-mass fragments.

2.3 Hydrodynamic Equations

For the purposes of the present simulation study, a meshless hydrodynamics model was desired. This approach would eliminate the need for storing and updating a grid, simplify calculations for large deformations, and allow for contiguous memory access to local field properties. The SPH formulation [32, 33] was chosen to satisfy the first two goals, while the latter will be discussed with regards to the GPU implementation. The core idea of SPH is to approximate a field property $f(x)$ by using a mollifier W (also known as an approximate identity) with compact support:

$$\langle f(x) \rangle = \int_{\Omega} f(s)W(x-s)ds, \quad W \in C_0^1(\mathbb{R}^n), \quad \Omega = \text{supp}(W) \quad (2.13)$$

where the brackets indicate the SPH approximation [33], allowing the field variables to be computed as a sum over the nearest neighbor particles representing the flow. In the present formulation, W is taken as the cubic spline kernel [32, 33], with a variable isotropic domain of support with radius h . Changing h in space and time allows for the simulation to respond to changes in flow conditions with a change in local resolution [32, 33]. A mass m is assigned to each particle representative in the model, as well as initial position and velocity components (x^β and v^β) in each β direction. Material properties such as density, ρ , and specific energy, e , complete the state description. Similar to the above integral relationship, derivatives and integrals of field functions can be approximated, resulting in the following set of equations [32, 33, 34] involving the kernel derivative (a scalar valued function of vector position \mathbf{x}):

$$\frac{Dx_i^\alpha}{Dt} = v_i^\alpha \quad (2.14)$$

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^N m_j (v_i^\beta - v_j^\beta) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\beta} \quad (2.15)$$

$$\frac{Dv_i^\alpha}{Dt} = - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_j^{\alpha\beta}}{\rho_j^2} + \Pi_{ij} \right) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\beta} + F_i^\alpha \quad (2.16)$$

$$\frac{De}{Dt} = \frac{1}{2} \sum_{j=1}^N m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) (v_i^\beta - v_j^\beta) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\beta} + \frac{1}{\rho_i} S_i^{\alpha\beta} \epsilon_i^{\alpha\beta} + H_i \quad (2.17)$$

where repeated indices in a product indicate implied summation over all possible values, $\sigma^{\alpha\beta}$ is the stress tensor, P is the pressure, $S^{\alpha\beta}$ is the deviatoric (traceless) stress tensor, $\epsilon^{\alpha\beta}$ is the local strain rate tensor, F represents external forces, and H represents energy sources. Π_{ij} represents the Monaghan numerical viscosity [33, 35] used to resolve shocks, accommodate heating along the shock, and resist unphysical material penetration. The material strength model for the solid target uses an elastic-perfectly plastic description of strength [32, 33, 34], where the hydrodynamic stress is determined as

$$\sigma_i^{\alpha\beta} = -P_i \delta^{\alpha\beta} + (1 - \eta) S_i^{\alpha\beta}, \quad \eta \in [0, 1] \quad (2.18)$$

where η is a material damage indicator, to be discussed later. It should be noted that fully damaged material ($\eta = 1$) is relieved of all stress due to deformation and behaves as a cohesionless fluid [34, 36]. The rubble-pile target is treated in this manner by default. In this elastic-plastic model, the components of the deviatoric stress tensor $S^{\alpha\beta}$ evolve using the following equation based on Hooke's law [32, 37]:

$$\frac{DS_i^{\alpha\beta}}{Dt} = 2G_s \left(\epsilon_i^{\alpha\beta} - 3\delta_i^{\alpha\beta} \epsilon_i^{\gamma\gamma} \right) + S_i^{\alpha\gamma} R_i^{\beta\gamma} + R_i^{\alpha\gamma} S_i^{\gamma\beta} \quad (2.19)$$

where $R^{\alpha\beta}$ is the local rotation rate tensor, G_s is the shear modulus, and the SPH approximation for these terms is given by

$$\epsilon_i^{\alpha\beta} = \frac{1}{2} \sum_{j=1}^N \frac{m_j}{\rho_j} \left[(v_j^\alpha - v_i^\alpha) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\beta} + (v_j^\beta - v_i^\beta) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\alpha} \right] \quad (2.20)$$

$$R_i^{\alpha\beta} = \frac{1}{2} \sum_{j=1}^N \frac{m_j}{\rho_j} \left[(v_j^\alpha - v_i^\alpha) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\beta} - (v_j^\beta - v_i^\beta) \frac{\partial W(\mathbf{x}_j - \mathbf{x}_i)}{\partial x^\alpha} \right] \quad (2.21)$$

To complete this system, we use the following equations governing the change of support radius h [32, 33], and the fracture damage ratio η [34]. The latter is limited in accordance with the number of material flaws activated in the structure.

$$\frac{Dh_i}{Dt} = -\frac{1}{n} \frac{h_i}{\rho_i} \frac{D\rho_i}{Dt}, \quad \frac{D}{Dt} \eta^{1/3} = \frac{c_g}{r_s} \quad (2.22)$$

where c_g is the crack growth rate, here assumed to be 0.4 times the local sound speed [34], and r_s is the radius of the subvolume subject to tensile strain. In the present model, the latter term is estimated by interpolation based on the strain rate tensor of neighbor particles. An equation of state remains to complete the mechanical system. We use the Tillotson equation of state [38] in the solid asteroid and in the aluminum penetrator used to deliver the surface explosive, with the parameters listed in Table 2.1. This is modified to include porosity, and an irreversible crush strength, for the ‘‘rubble pile’’ target [36, 39]. We assume a power law distribution for number of implicit flaws in a volume of material with respect to local tensile strain (a Weibull distribution), and assign flaws with specific activation thresholds to each SPH particle [34]. The maximum damage allowed to accumulate in a volume is described by

$$\eta_i^{\max} = \left(\frac{n_i}{n_i^{\text{tot}}} \right)^{1/3}, \quad \epsilon_i = \frac{\sigma_i^t}{(1 - \eta_i)E} \quad (2.23)$$

where n_i is the number of active flaws ($\epsilon > \epsilon^{\text{act}}$) and n^{tot} is the total number of flaws assigned to a particle, which can vary widely, but is always at least one. Equation (11) also gives the relationship for the local scalar strain, as a function of the maximum tensile stress σ^t , the local damage, and the Youngs modulus E .

Table 2.1 Parameters for Tillotson Equation of State in Core Material

Parameter	Numerical Value	Units
a_t	0.5	
b_t	1.5	
A_t	7.1E10	Pa
B_t	7.5E10	Pa
α_t	5	
β_t	5	
E_0	4.87E8	J/kg
E_{iv}	4.72E6	J/kg
E_{cv}	1.82E7	J/kg

2.3.1 Adaptive Smoothed Particle Hydrodynamics

Since the hypervelocity impact and explosive simulations rely heavily on energy transmission through shocks, the current simulation framework uses Adaptive Smoothed Particle Hydrodynamics (ASPH) to mitigate some of the computational and fidelity issues that arise in more complex simulations. This approach is explained clearly by Owen et. al. [40], the implemented details of which are repeated here. A concise explanation of the differences from “standard” SPH can be found in [33]. SPH uses a kernel function that acts on a normalized position space. This kernel, $W(\nu)$ is a function of the normalized distance $\nu = r/h$, where r is the physical distance vector, and h is the isotropic smoothing length. To add a direction sense to ellipsoidal nodes, ASPH uses a symmetric, positive definite linear transformation G such that $\nu = Gr$. Then, the kernel derivative becomes:

$$\nabla W(\eta) = G \frac{\eta}{|\eta|} \frac{\partial W}{\partial \eta} \quad (2.24)$$

Given a matrix representation of the tensor G , the entries are evolved by:

$$\frac{DG}{Dt} = \frac{DR}{Dt}G - G\sigma \quad (2.25)$$

where $\sigma_{ij} = \partial v_i / \partial x_j$ is the deformation tensor. This can be easily obtained from the strain rate tensor and rotation rate tensor used for the general SPH implementation. R is an infinitesimal rotation of the smoothing kernel. It is well defined at the current time by noting that it must be antisymmetric and that the resulting offdiagonal elements of DG/Dt

must be symmetric. A description of a process to calculate these intermediate terms is given in Reference [40].

A smoothing process for the tensor G is implemented after many time steps, as suggested in [33]. This is done as a weighted average over the neighboring values to generate the new smoothing tensor G' by

$$G' = |G| |g_i| g_i \quad (2.26)$$

where

$$g_i = \frac{\sum_j G_j^{-1} W_{ij}}{\sum_j W_{ij}} \quad (2.27)$$

To mitigate spurious shear viscosity that is introduced by the artificial viscosity Π_{ij} , we modify this viscosity calculation using the Balsara correction [41]:

$$\tilde{\Pi}_{ij} = \frac{1}{2}(f_i + f_j)\Pi_{ij} \quad (2.28)$$

where

$$f_i = \frac{|\nabla \cdot v_i|}{|\nabla \cdot v_i| + |\nabla \times v_i| + \delta c_i/h_i} \quad (2.29)$$

where c is the local sound speed and δ is a small number chosen to prevent divergence.

2.3.2 Tensor Damage Model

In the initial SPH model for comparison, the behavior of the core material under high stress is governed by the activation of implicit flaws. These flaws are seeded in the representation particles using a Weibull distribution with a coefficient of around 4.2E23 and an exponent between 6.2-9.5. Using a range of distribution exponents and strength properties allows us to examine the behavior of the core material with varying brittleness and material cohesion. This turns out to be very important for this contact binary system, as strong core material absorbs energy from the disruption shock and can result in large remaining chunks of material. Smoothing lengths are chosen to allow for resolution of between 1 cm and 5 cm, which results

in a hydrodynamic system of between 800,000 and 6,000,000 nodes. This system is scaled to be an ideal size for the GPU simulation programs developed at the Iowa State Asteroid Deflection Research Center (ADRC), maximizing both computational efficiency and simulation turnaround time.

For this comparison, a damage model using a tensor variable was implemented. The details are the same as those used in the Spheral code, developed by Mike Owen at the Lawrence Livermore National Laboratory. We use a tensor damage variable defined per node $D^{\alpha\beta}$ in order to support directionality in the damage evolution. Cracks are allowed to open up in response to strain aligned perpendicularly to that direction, there is substantially reduced crack growth in orthogonal directions to the strain. The tensor strain, $\sigma^{\alpha\beta}$ used is the “pseudo plastic strain” of SolidSpheral, due to Mike Owen, which evolves in time as

$$\frac{D\sigma_i^{\alpha\beta}}{Dt} = \frac{1}{G_s} \frac{DS_i^{\alpha\beta}}{Dt} \quad (2.30)$$

This is decomposed into a set of eigenvalues, σ^ν , and eigenvectors, $\Lambda^{\alpha\nu}$, from which the directional scalar damage, Δ^ν is the magnitude of the ν -th column of $D^{\alpha\gamma}\Lambda^{\gamma\beta}$. The maximum damage allowed to accumulate in a volume allowing for directionality, is:

$$D^{\max} = \max\left(\frac{n_i}{n_i^{\text{tot}}}, \Delta_i^\nu\right) \quad (2.31)$$

where n_i is the number of active flaws ($\epsilon > \epsilon^{\text{act}}$) and n^{tot} is the total number of flaws assigned to a particle, which can vary widely, but is always at least one. These directional damages can then be time evolved using representative scalar evolution laws [34].

2.3.3 Neighbor Finding Implementation

One of the key limitations of the SPH simulation approach is that most proposed neighbor-finding methods for interpolation rely on complex logic and lists not suitable for efficient GPU implementation. Therefore, the addition of the third dimension makes this problem far more complex. A new approach for efficiently computing unions and intersections of integer sets on the GPU is proposed, allowing for neighbor-finding as an update process from previously

computed relative relationships. Based on a standard Sort-and-Sweep approach in computer graphics [42], the power of this approach lies in how it scales with increased number of SPH interpolants. In addition to scaling superlinearly (compared to quadratic brute force calculations), the present approach uses the Thrust library to sort the position components of the particles in parallel. This eliminates a series of memory transfers with the host and keeps all data on the GPUs.

A subsequent group of GPU kernels establish pointers to the limits on the sorted array for which candidate neighbor particles may belong. This reduces the neighbor finding to an integer union calculation, which can be conducted as a logical (true/false) operation. Comparing the position of the sorted particle IDs with the limits allows for a simple yes/no decision on whether a proposed neighbor could be within the support of the interpolation function. Figure 2.4 gives a depiction of this process for each computing thread. Figure 2.5 shows the improvements of the present model over in-place neighbor calculations (also on the GPU). While dimensionality affects the speed-up, there are still substantial gains made over past implementations.

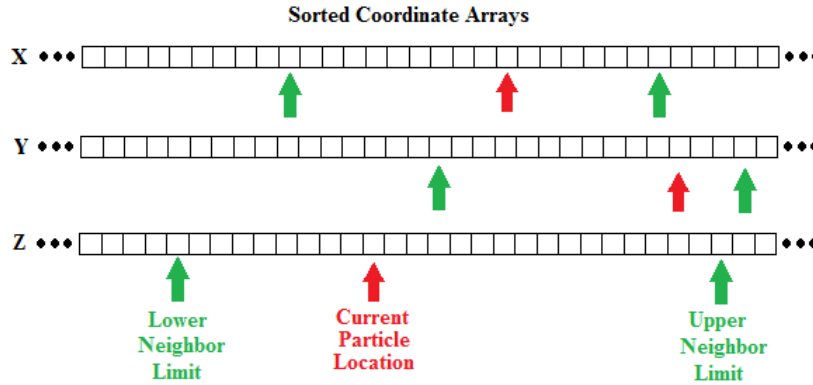


Figure 2.4 Description of Sorted Neighbor Kernel Process.

2.4 Disruption Mission Profiles

This section outlines the initial conditions for three methods of NEO deflection using nuclear explosive devices. For the initial demonstration cases, a 100 m diameter target asteroid is modeled with an energy source of 100 kt. The newer, asymmetric, model uses an energy

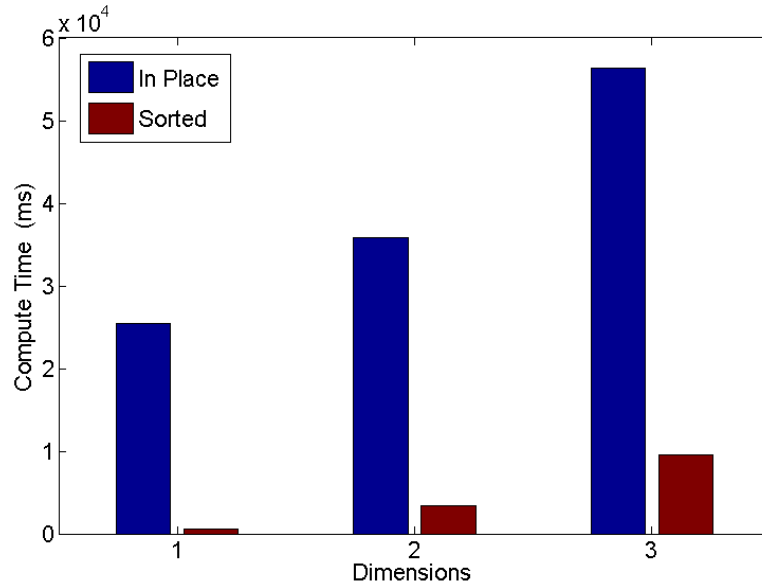


Figure 2.5 Neighbor Search Cost.

source equivalent to 70 kilotons. Thermal emission is omitted from the subsurface and surface explosions due to absorption by surrounding material in the time scale of interest.

2.4.1 Subsurface Explosion Setup

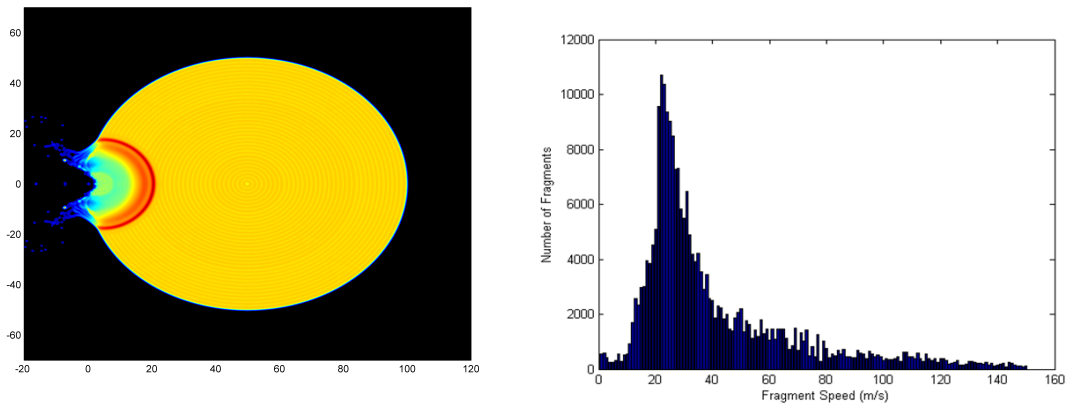


Figure 2.6 Subsurface Explosion and Resulting Fragment Velocities.

For this simulation, the explosive is modeled as a cylindrical energy source buried at a depth of 5 meters. As shown for the solid target in Figure 2.6, the blast wave compresses the NEO, reducing it to fragments, and disperses it primarily along the axis of the explosion. The

resulting fragment distribution for a case like this has a peak between 20-70 m/s, with a tail of high-speed ejecta like that shown in Figure 2.6.

2.4.2 Surface Penetrator Model

Two main models for an explosion at the surface are used. One is a static explosion, which results in vastly different systems depending on the composition of the body. For a solid target, cratering and pitting is expected rather than disruption. Even dispersed rubble-pile asteroids have a far lower mean fragment velocity than a similar subsurface system. The second model, shown here, includes an aluminum penetrator impacting the surface at 6.1 km/s. The explosion thermal energy turns the high-mass impactor into a plasma, which burrows into the surface as it releases its energy. Slower dispersion velocity is observed than the subsurface case, but this approach is extremely beneficial from an engineering standpoint, as there is strong coupling between time-to-impact and a reduction in mission fuel cost [23]. The benefit to this method relative to a subsurface explosion is that it does not require a rendezvous, and therefore there are available launch windows for this type of mission right up until immediately before the impact date.

2.4.3 Standoff Energy Deposition

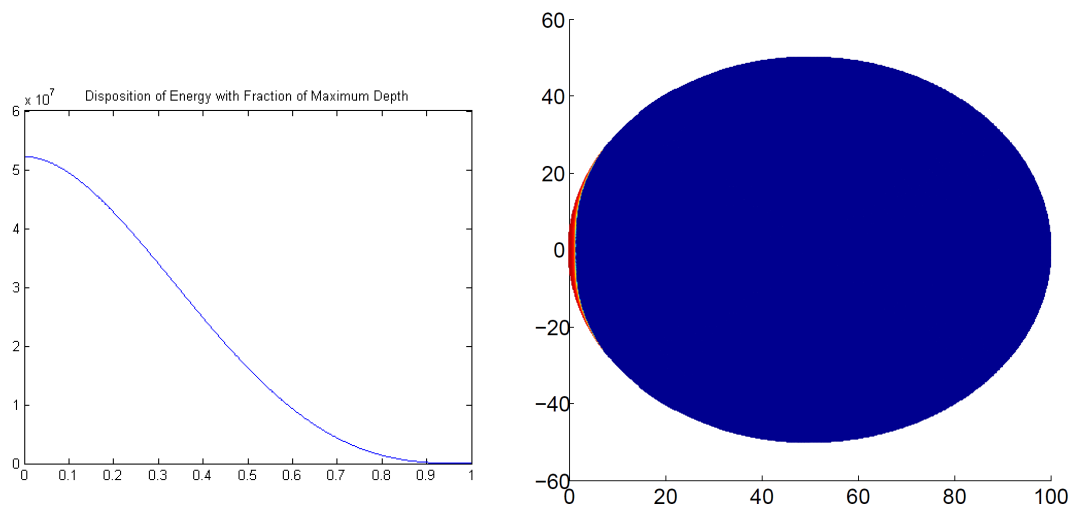


Figure 2.7 Radial Energy Deposition and Total Deposition Region.

For a standoff blast, additional physics must be considered. An energy deposition strategy is required that does not directly compute X-ray and neutron scattering in the target. For this, a ray-tracing algorithm is used with radial energy deposition at the surface as shown in Figure 2.7 for neutrons. This is derived from a Monte Carlo scattering result from TART, a DOE neutron deposition code, in NEO analog materials [16]. A 10% neutron yield is assumed for these simulations, and a maximum deposition depth of 1.5 m to compare to deposition predicted for chondritic materials [43]. The overall deposition region (shown as the logarithm of deposited energy) is also shown in Figure 2.7. A modified SPH node representation is created that resembles an ablative modeling grid used in high-energy deposition physics. This distribution is shown in Figure 2.8, and has a minimum smoothing scale of 0.1 cm with a maximum local change rate of 10% up to 0.2 m resolution. Also in Figure 2.8, the resulting ablation provides an effect similar to that of a rocket, but also disrupts the rubble-pile target completely.

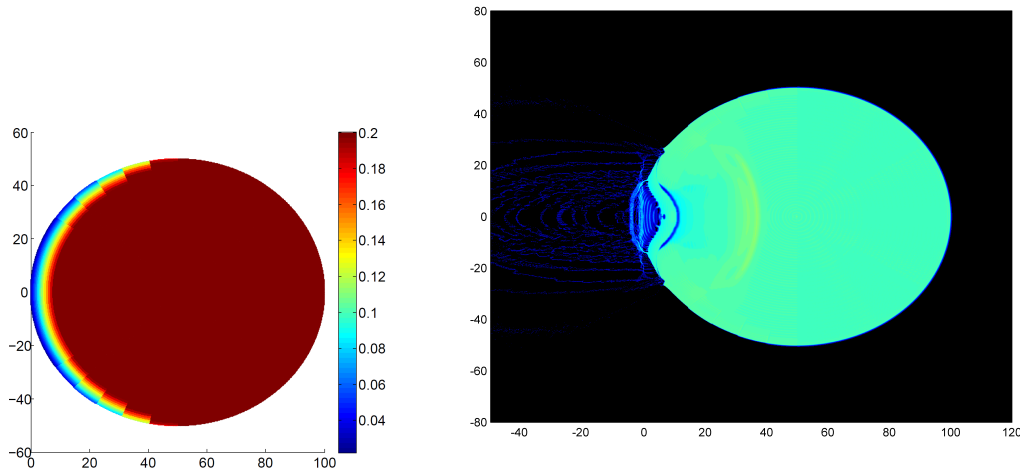


Figure 2.8 SPH Nodes and Resulting Ablation for Standoff Model.

2.4.4 HAIV Concept

A slice of the nominal three-dimensional target was shown in Figure 2.3. As an increase in computational burden, it performs moderately less efficiently than the two dimensional model. The overall velocity statistics, which are the governing variables behind successful disruption,

are similar to those for the cylindrical case. The histogram for radial dispersion velocities of the fractured particles can be seen in Figure 2.9. There is a mean dispersion velocity for the HAIV case of almost 350 m/s.

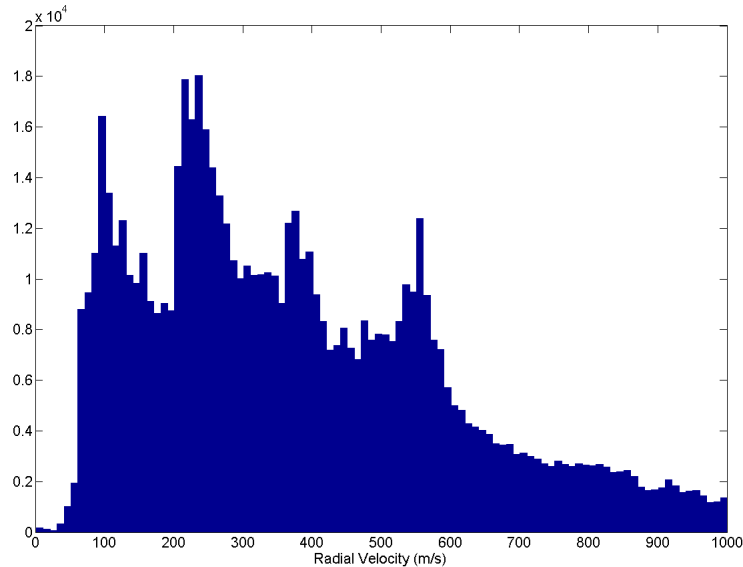


Figure 2.9 Radial Dispersion Velocity Histogram for HAIV Concept.

The travel of the explosive shock can be seen in Figure 2.10. This process dissipates some energy due to interactions with the rebounding shock front. In the center area of deeper regolith, the seeding process naturally results in a much more porous material, absorbing energy from the shock. The new damage model allows for better tracking of crack propagation, such as that shown in Figure 2.11. Upon reaching the second core at the far side, some large chunks escape the disruption process in some cases (even with lower material strengths). A final hydrodynamic state can be seen in Figure 2.12.

There remains a high risk for this target of single largest chunks on the order of tens of meters. However, this material is highly stressed due to velocity gradients, and may be ripped apart in further time. The large velocity gradients and the location of the slowest debris can be observed in Figure 2.13. Further, these large chunks are still imparted substantial velocities from the blast 10-20 m/s, and have sufficient energy to disperse from the nominal impacting trajectory over tens of days.

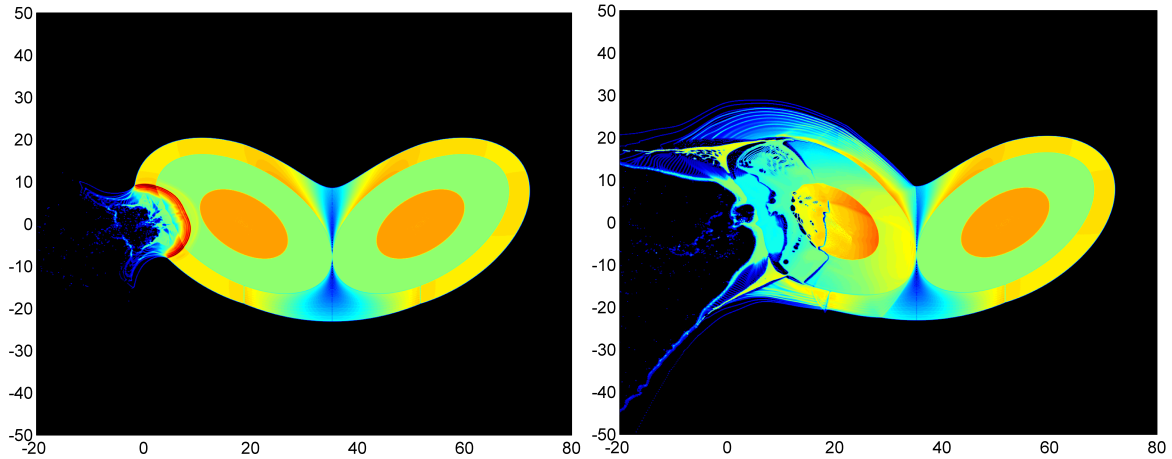


Figure 2.10 Asymmetric Shock Behavior.

Slower dispersion velocity is observed for the contact burst, as shown in Figure 2.14. The mean dispersion velocity is only 150 m/s, which is 2x less effective than the baseline HAIV. In terms of kinetic energy, the HAIV concept is superior by almost a factor of 10. It is clear that this HAIV approach is also extremely beneficial from an engineering standpoint, as there is strong coupling between time-to-impact and a reduction in mission fuel cost [23]. The benefit to this method relative to a subsurface explosion is that it does not require a rendezvous, and therefore there are available launch windows for this type of mission right up until immediately before the impact date.

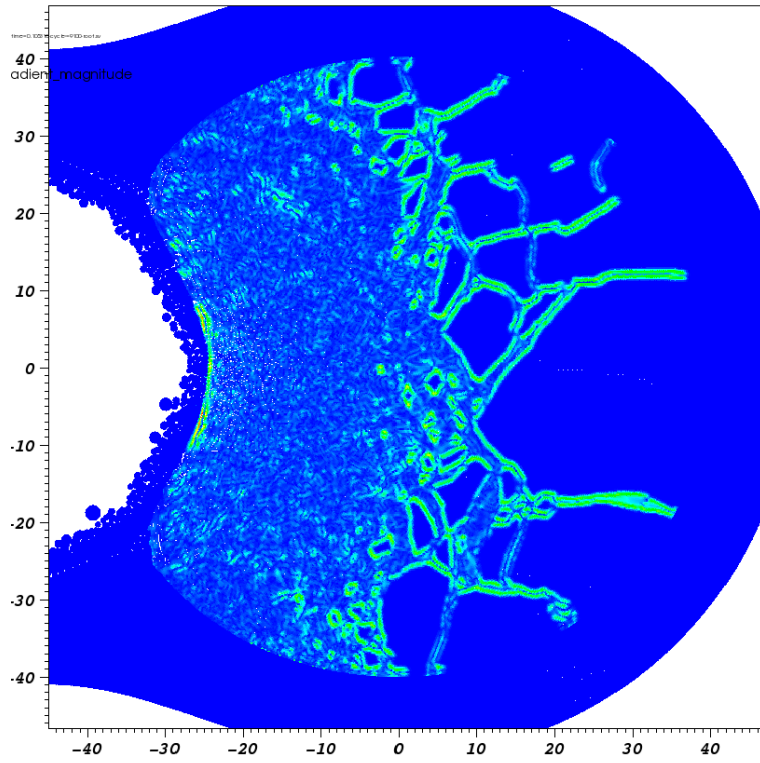


Figure 2.11 Example Damage Localization for Tensor Fracture Model.

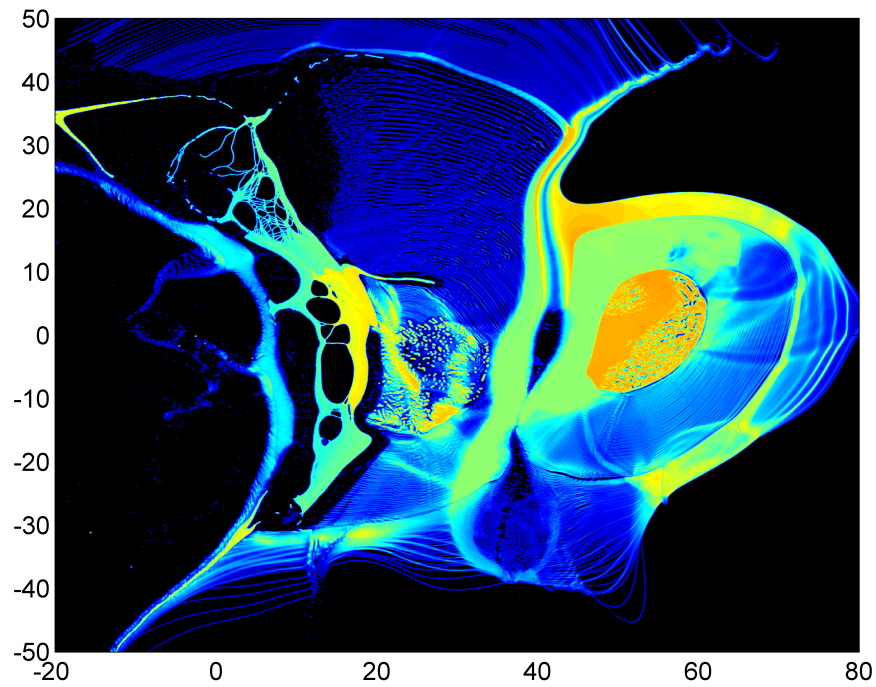


Figure 2.12 Final Disruption of NEO Target.

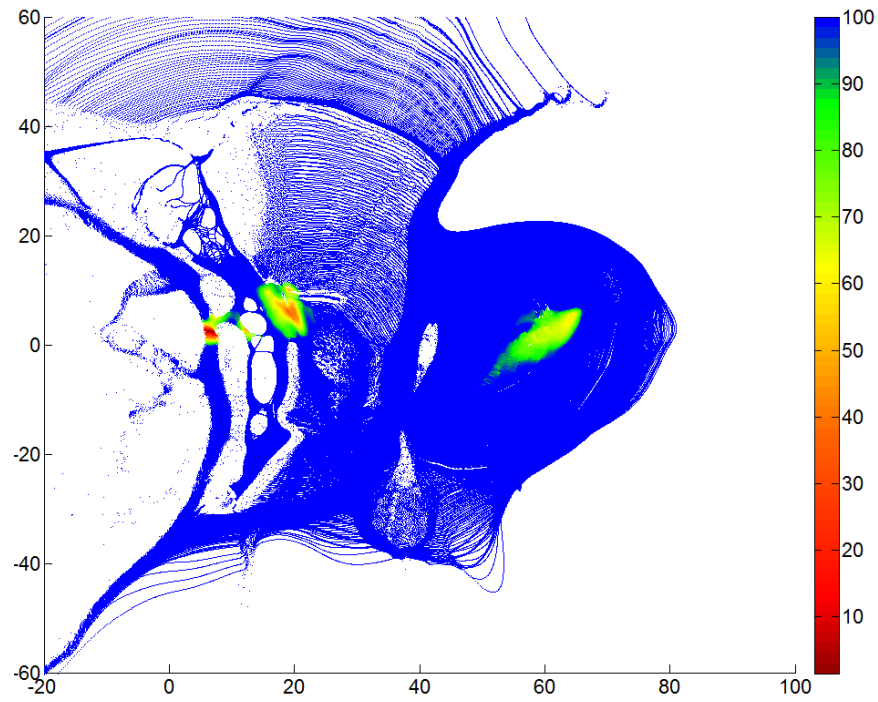


Figure 2.13 Location of Slowest Moving Debris.

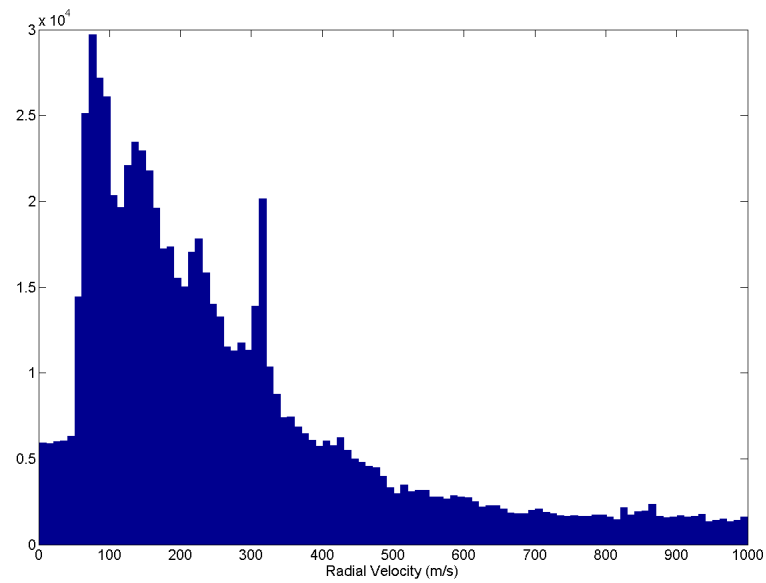


Figure 2.14 Radial Dispersion Velocity Histogram for Contact Burst.

CHAPTER 3. TRAJECTORY ANALYSIS

This section describes the identification of nominal orbits for a fixed impact time. Given a desired lead time, the optimal approach vector is computed using a differential step update, described in the following subsections. The cost parameter is the impacting percentage of the original target mass.

3.1 Impacting Orbit Solver

The orbital parameters for the nominal trajectory are sampled from a (a, e, i) space that represents the distribution of known NEOs, as shown in Figure 3.1. This is done using inverse transform sampling, in which a random number is mapped to the integral of the cumulative density function for each of these three parameters. Given a , e , i , and the impact date, we have all of the information needed to pin down an impacting orbit. If we assume that the orbit passes through the center of the Earth, then we have x_E , y_E , and z_E , which are the Cartesian coordinates for the Earth's center of mass at that epoch, which coincide with a point on the desired orbit.

Given a , e , and $r = \sqrt{x_E^2 + y_E^2 + z_E^2}$, the specific angular momentum is calculated as $h = \sqrt{\mu a(1 - e^2)}$ [52, 53]. Then, the true anomaly, θ and velocity magnitude, v , are calculated using the orbit equation and the vis-viva equation:

$$r = \frac{h^2}{\mu} \frac{1}{1 + e \cos \theta}; \quad \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \quad (3.1)$$

We can also calculate the radial velocity, v_r , as

$$v_r = \frac{\mu}{h} e \sin \theta \quad (3.2)$$

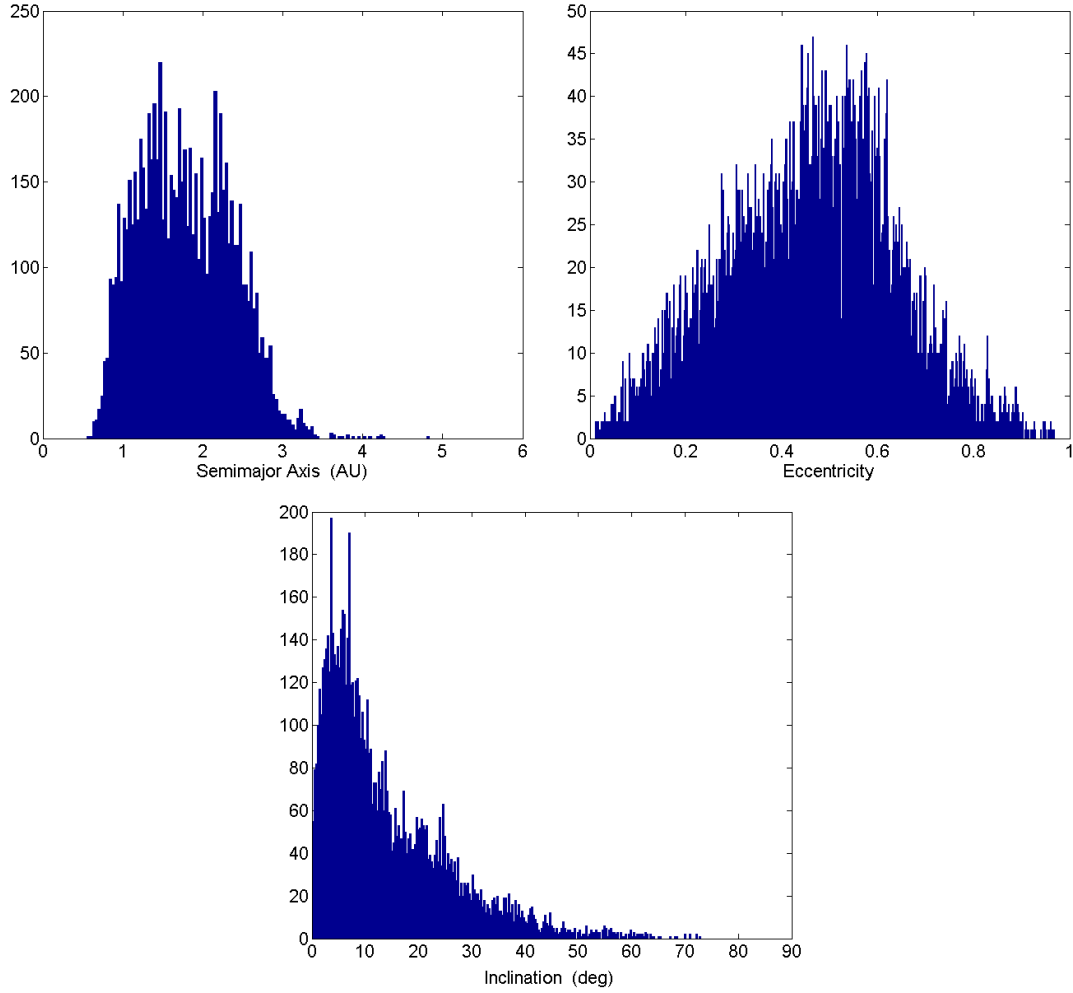


Figure 3.1 Histograms of Known NEO Population.

which gives us all the needed scalars to solve the following set of nonlinear equations for the velocity components v_x , v_y , and v_z , resulting in the desired state vector:

$$\begin{aligned}
 f_1(v_x, v_y, v_z) &= 0 = xv_y - yv_x - h \cos i \\
 f_2(v_x, v_y, v_z) &= 0 = xv_x + yv_y + zv_z - rv_r \\
 f_3(v_x, v_y, v_z) &= 0 = \sqrt{v_x^2 + v_y^2 + v_z^2} - v
 \end{aligned} \tag{3.3}$$

3.1.1 An Adaptive Shooting Method for Reference Impacting Trajectories

To generate a reference impacting trajectory, a directed search among initial orbital parameters is needed. While implicit methods [54, 55] for a nonlinear shooting problem are very robust, we desire the ability to dynamically generate reference trajectories independent of the integrator used and without the need for tuning update parameters. A fundamental resolution problem keeps many sets of orbital parameters from impacting the center of the Earth, even if previous (similar) parameters were shown to impact using a different integrator or different step sizes. This same resolution problem limits the efficacy of global searches based on finite differencing. For this problem, estimated post-2029 orbital parameters for the asteroid Apophis [22] were used as an initial guess for a directed search method.

Consider the vector function, \mathbf{f} , which gives the position error of a chosen set of orbital parameters for the asteroid Apophis on April 13, 2036 in a chosen coordinate system. Then, we note that \mathbf{f} has three components that can be related to Earth's position:

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} x_a - x_e \\ y_a - y_e \\ z_a - z_e \end{bmatrix} = \mathbf{r}_a - \mathbf{r}_e \quad (3.4)$$

If we consider a Taylor series expansion of \mathbf{f} limited to first order terms, we can evaluate the function at a new set of parameters, \mathbf{p}_n , which are close to the previous guess, \mathbf{p}_{n-1} , as measured in six dimensional configuration space. The function \mathbf{f} can then be viewed as a projection of these parameters on position coordinates at the predetermined final time:

$$\mathbf{f}(\mathbf{p}_n) = \mathbf{f}(\mathbf{p}_{n-1}) + \frac{1}{2} \mathbf{J} \mathbf{d}\mathbf{p} \quad (3.5)$$

where \mathbf{J} is the 3x6 matrix of partial derivatives (commonly called the Jacobian) and $\mathbf{d}\mathbf{p} = \mathbf{p}_n - \mathbf{p}_{n-1}$ is the vector difference of orbital parameters in the desired search direction. Assuming the desired search direction has an error of 0, we propose a correction algorithm for determining the search direction of the form:

Table 3.1 Osculating Orbital Parameters for Fictitious Impact Trajectory

Orbital Parameter	Value
Semimajor Axis	1.1082428 AU
Eccentricity	0.189928428
Inclination	2.18995362 deg
Longitude of Right Ascension	203.18642266 deg
Argument of Perihelion	69.929774 deg
Initial Mean Anomaly	296.74684241 deg
Epoch	64781 MJD
Miss Distance on Target Date	4.738466849E-011 Earth Radii

$$\mathbf{p}_n = \mathbf{p}_{n-1} + k\mathbf{dp} \quad (3.6)$$

$$\mathbf{dp} = -2\mathbf{J}^+\mathbf{f}(\mathbf{p}_{n-1})$$

where $\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ is the Moore-Penrose pseudoinverse of \mathbf{J} , resulting in a least squares solution to Equation (3.5) [56] when $\mathbf{f}(\mathbf{p}_n) = 0$ and k is a scalar to be properly chosen for robust convergence to an impacting trajectory. Exponential convergence of this method with respect to an 11-body dynamical model is shown in Figure 3.2.

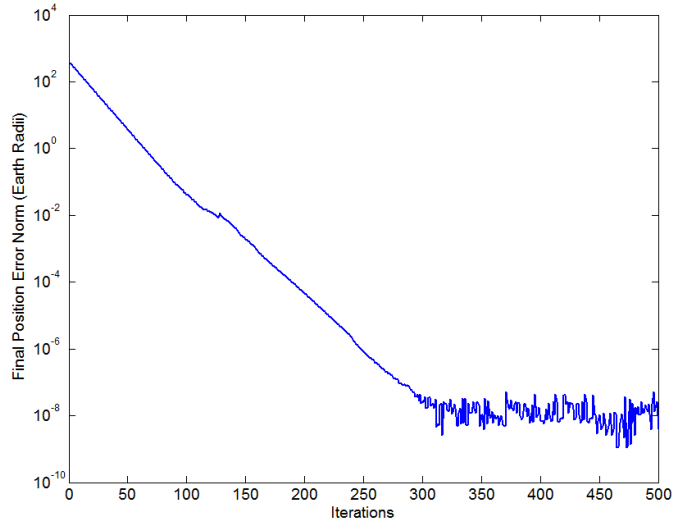


Figure 3.2 Convergence History of Error Function Norm for 11-body Model.

Also visible from the scale of Figure 3.2 is the fact that the final set of orbital parameters define a trajectory which impacts the Earth on the target date. The trajectory is designed to pass through the center of the Earth to within the precision available. This allows the offset to be eliminated as a cause for error when comparing the results of a fragmented body. Table 3.1 shows the initial osculating orbital parameters used for this simulation..

3.2 Fragmented System Estimation

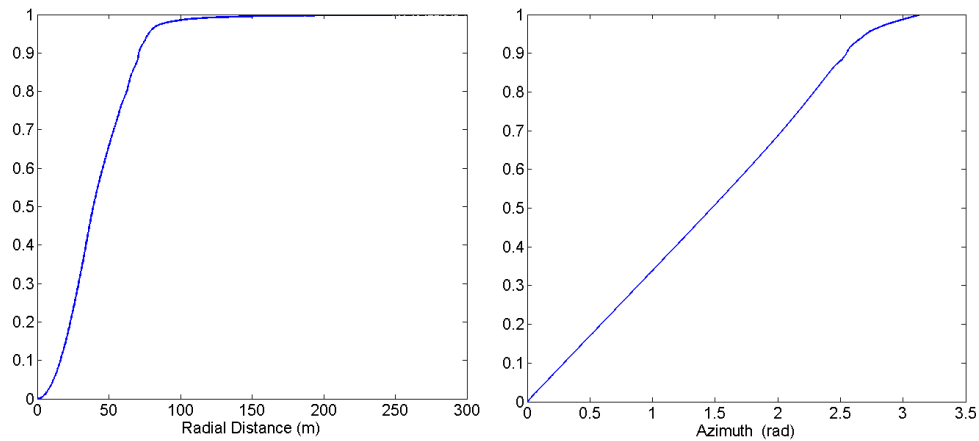


Figure 3.3 Cumulative Density Functions for Disrupted Asteroid.

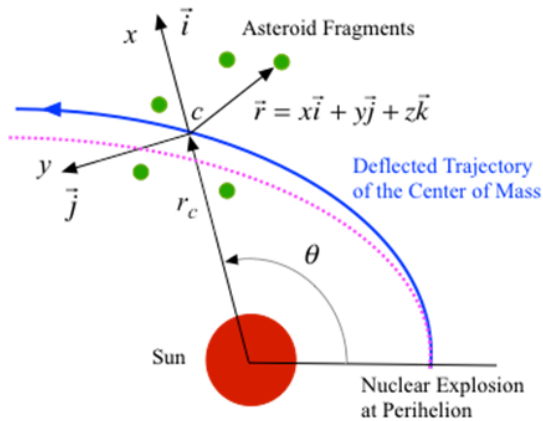


Figure 3.4 Rotating Local-Vertical-Local-Horizontal (LVLH) Frame.

Statistics representing the fragmented system are collected and stored as cumulative density functions for the needed variables, similar to those shown in Figure 3.3. A representative

fragment system of 10,000 to 100,000 fragments is created from these statistics using inverse transform sampling. The debris cloud is given global coordinates in a Local-Vertical-Local-Horizontal (LVLH) reference frame about the center of mass, as shown in Figure 3.4. Since the hydrodynamic model is axisymmetric, and has a definite direction of maximum momentum along the axis of symmetry, a desired deflection direction must be chosen. These are then integrated to predict an ephemeris for a 48 hour period surrounding the nominal time of impact. Since the LVLH reference frame is computationally beneficial for self-gravity and collision modeling among fragments, we use the nonlinear relative equations of motion for this frame to govern fragment trajectories [52, 53]:

$$\ddot{x}_i = 2\dot{\theta} \left(\dot{y}_i - \frac{\dot{r}_c}{r_c} y_i \right) + \dot{\theta}^2 x_i + \frac{\mu}{r_c^2} - \frac{\mu}{r_d^3} (r_c + x_i) + \frac{\mu_E}{r_{Ei}^3} (x_E - x_i) + F_i^x \quad (3.7)$$

$$\ddot{y}_i = -2\dot{\theta} \left(\dot{x}_i + \frac{\dot{r}_c}{r_c} x_i \right) + \dot{\theta}^2 y_i - \frac{\mu}{r_d^3} + \frac{\mu_E}{r_{Ei}^3} (y_E - y_i) + F_i^y \quad (3.8)$$

$$\ddot{z}_i = -\frac{\mu}{r_d^3} z_i + \frac{\mu_E}{r_{Ei}^3} (z_E - z_i) + F_i^z \quad (3.9)$$

where x , y , z , r_c , and θ are defined as shown in Figure 3.4, r_d is the length of the relative coordinate vector, μ and μ_E are gravitational parameters for the sun and the Earth, r_{Ei} is the distance from each fragment to Earth, and (F^x, F^y, F^z) are the combined acceleration components due to 3rd body gravitational terms (solar system major body model, self gravity, and collision corrections).

3.2.1 Gravity Model

Since the simulation is predominantly meshless, a temporary grid is created between the minimum and maximum coordinate values in the LVLH frame. In the current model, the fragments are regarded as spherical so the grid spacing is generally at least 2.0 times the maximum particle radius. In most cases, the grid was allowed to be up to 20 times the particle radius, as grid size determined the precision of self-gravity perturbations. Referring to the two-dimensional grid in Figure 3.5, the colored grid cells adjacent to the grid containing the current fragment of interest are the cells in which self-gravity has the form:

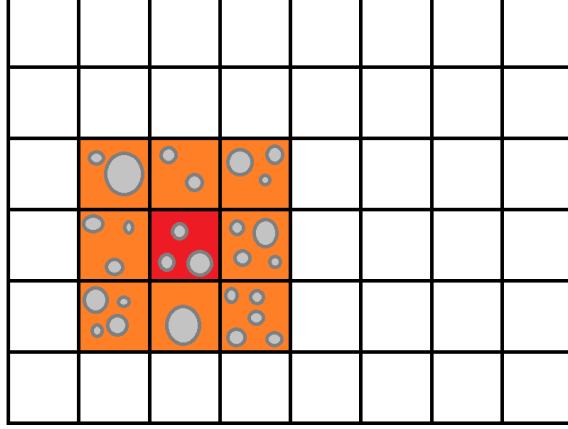


Figure 3.5 Temporary Grid Creation and Evaluation.

$$\begin{aligned}
 (F_x)_i &= \sum_j \frac{Gm_j(x_j - x_i)}{r_{ij}^3} + \sum_k A_k \\
 (F_y)_i &= \sum_j \frac{Gm_j(y_j - y_i)}{r_{ij}^3} + \sum_k A_k \\
 (F_z)_i &= \sum_j \frac{Gm_j(z_j - z_i)}{r_{ij}^3} + \sum_k A_k
 \end{aligned} \tag{3.10}$$

where G is the universal gravitational constant, m_j is the mass of the gravitating body, r_{ij} is the relative distance between each fragment and the gravitating body being considered, and A_k are the cell averages for all nonadjacent cells. Outside of this region, an averaged fragment containing the total mass contained within the cell is modeled at the centroid of the cell. Using this method, the expense of the self-gravity calculation is reduced while retaining an aggregate force for many small masses. Some methods for self-gravity neglect the force between bodies if it is sufficiently small. However, this cutoff is not well defined, and a large number of small fragments at a considerable distance produce an effect similar to a much larger body at the average distance of these fragments. Therefore, it would substantially alter the system dynamics to neglect this force rather than average it. The radius of cells in which self-gravity is considered is a user set parameter for the present model, usually set between 1 and 3. A block diagram of the process logic for a computing thread block containing fragment i is shown in Figure 3.6.



Figure 3.6 Block Diagram for Self-Gravity Process Logic (Contains Fragment i).

3.2.2 Collision Model

Collisions between bodies are a concern of any aggregation model. With a self-gravity model attracting each fragment to all others, there must be a physically realistic way of limiting the distance between two bodies that cannot occupy the same space. However, checking for collisions through brute force can be as expensive as the self-gravity model. A model excluding interacting pairs in adjacent cells is employed similar to that discussed in [42]. A Sort and Sweep algorithm checks for colliding pairs along each coordinate, excluding fragments from further consideration. When an overlapping pair of fragments are detected, they are backed up along the normal connecting their centroids. An inelastic collision process with a coefficient of restitution of 0.5 predicts the post-collision velocity, as shown in Figure 3.7.

Table 3.2 Orbital Parameters for Planetary Ephemerides [1]

Planet	a (au)	e	i (deg)	Ω (deg)	$\tilde{\omega}$ (deg)	L (deg)
Mercury	0.38709893	0.20563069	7.00487	48.33167	77.45645	252.25084
Venus	0.72333199	0.00677323	3.39471	76.68069	131.53298	181.97973
Earth	1.00000011	0.01671022	0.00005	-11.26064	102.94719	100.46435
Mars	1.52366231	0.09341233	1.85061	49.57854	336.04084	355.45332
Jupiter	5.20336301	0.04839266	1.30530	100.55615	14.75385	34.40438
Saturn	9.53707032	0.05415060	2.48446	113.71504	92.43194	49.94432
Uranus	19.19126393	0.04716771	0.76986	74.22988	170.96424	313.23218
Neptune	30.06896348	0.00858587	1.76917	131.72169	44.97135	304.88003
Pluto	39.48168677	0.24880766	17.14175	110.30347	224.06676	238.92881

Table 3.3 Parameter Rates for Planetary Ephemerides [1]

Planet	\dot{a} (au/Cy)	\dot{e} (1/Cy)	\dot{i} (deg/Cy)	$\dot{\Omega}$ (deg/Cy)	$\dot{\tilde{\omega}}$ (deg/Cy)	\dot{L} (deg/Cy)
Mercury	0.00000066	0.00002527	-23.51	-446.30	573.57	538101628.29
Venus	0.00000092	-0.00004938	-2.86	-996.89	-108.80	210664136.06
Earth	-0.00000005	-0.00003804	-46.94	-18228.25	1198.28	129597740.63
Mars	-0.00007221	0.00011902	-25.47	-1020.19	1560.78	68905103.78
Jupiter	0.00060737	-0.00012880	-4.15	1217.17	839.93	10925078.35
Saturn	-0.00301530	-0.00036762	6.11	-1591.05	-1948.89	4401052.95
Uranus	0.00152025	-0.00019150	-2.09	-1681.4	1312.56	1542547.79
Neptune	-0.00125196	0.00002514	-3.64	-151.25	-844.43	786449.21
Pluto	-0.00076912	0.00006465	11.07	-37.33	-132.25	522747.90

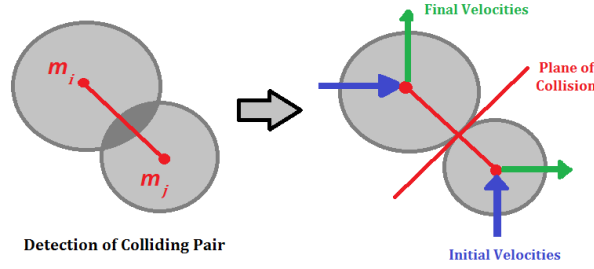


Figure 3.7 Collision Detection and Evaluation Process for Interacting Pair.

3.2.3 Planetary Orbit Models

In order to have a more accurate N-body integrator, we rely on planetary ephemeris data to prepare relative position vectors between a fragment and a gravitating body. At a simple level, this can be done by using classical orbital elements for each body at a given epoch, and interpolating drift from this value based on a linear approximation to orbital element rates of change. Table 3.2 shows the heliocentric orbital elements used to correspond to the J2000 epoch for the 8 major planets and the minor planet Pluto. These allow a linear approximation of the current orbital parameters using the rate data from Table 3.3:

$$Q(t) = Q_0 + \dot{Q}T \quad (3.11)$$

where Q is a particular orbital parameter, Q_0 is its J2000 epoch value in Table 3.2, \dot{Q} is the century rate of change found in Table 3.3, and T is the elapsed time since the J2000 epoch in Julian centuries (36525 Julian days). Using the current value for these parameters, they can be converted to include more common elements such as the argument of perihelion and mean anomaly using the following relationships [52]:

$$\omega = \tilde{\omega} - \Omega \quad (3.12)$$

$$M = L - \tilde{\omega}$$

The classical orbital elements are then translated into position and velocity state vectors for integration. Interpolation based on this scheme matched fragment position results using DE405

Planetary Ephemerides to within 1 meter for the 15 day mission profile described in [13], and therefore meet the goals of the present model with substantial computational reduction compared to more costly systems. This system for planetary ephemerides will therefore allow the present model to validate previous dispersion models including mutual gravitational acceleration terms among fragments. Application of a simulation system using DE405 ephemerides is currently underway for longer baseline scenarios such as the AIAA fictional impactor problem.

3.3 Earth Rotation Model

The present model of Earth's rotation is based on fixing a vector to the equator of a spherical Earth approximation at the Greenwich meridian. Orientation of this vector with respect to the Earth Centered Inertial (ECI) coordinate frame is obtained using Greenwich Sidereal Time (GST) [52]:

$$\begin{aligned}\theta_0 &= 100.4606184 + 3600.77004T + 0.000387933T^2 - 0.00000002583T^3 & (3.13) \\ \theta_{gst} &= \theta_0 + \frac{360.98564724}{24}t_{UT}\end{aligned}$$

where θ_{gst} represents the angle between the Greenwich Meridian and the line of vernal equinox (The X coordinate of the ECI system) in degrees, θ_0 is the local sidereal time at 0 hr UT of the current day, and t_{UT} is the current UT in hours.

3.3.1 A Direct Mapping to Earth-Fixed Coordinates

In addition to the transformation of position coordinates provided by the previous Earth rotation model, we desire a coordinate relationship between the integrated fragment state vectors in heliocentric coordinates and their counterparts in the ECI system, which is not truly inertial but is in fact subject to acceleration in its orbit around the sun. The Earth-relative

position vectors for each fragment, \mathbf{r}_{rel} , are easily found as:

$$\mathbf{r}_{rel} = \mathbf{r} - \mathbf{r}_E = \begin{bmatrix} x_i - x_E \\ y_i - y_E \\ z_i - z_E \end{bmatrix} \quad (3.14)$$

where \mathbf{r} is a position vector describing each fragment, and the position vector for Earth, \mathbf{r}_E , is found using the interpolated ephemeris data for Earth orbit on the time step prior to impact. Similarly, the relative velocity to the ECI frame, \mathbf{v}_{rel} , is found using the relative velocity relationship to Earth's heliocentric velocity, \mathbf{v}_E . This can also be found by differentiating Equation 3.14 and substituting Earth's orbital angular momentum for the coordinate rotation [52]:

$$\mathbf{v}_{rel} = \mathbf{v} - \mathbf{v}_E - \frac{1}{r_E^2} (\mathbf{r}_E \times \mathbf{v}_E) \times \mathbf{r}_{rel} \quad (3.15)$$

In these relative motion equations, all components are still expressed in ecliptic coordinates. Since the ECI shares the vernal equinox as a common basis vector, we can transform components using a simple rotation around the X axis through an angle $-\epsilon$:

$$\mathbf{r}_{ECI} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\epsilon) & -\sin(\epsilon) \\ 0 & \sin(\epsilon) & \cos(\epsilon) \end{bmatrix} \mathbf{r}_{rel} \quad (3.16)$$

where ϵ is an angle measuring the mean obliquity of the ecliptic plane, computed in degrees using [53]:

$$\epsilon = 23.439291 - 0.130042T - 0.000000164T^2 + 0.000000504T^3 \quad (3.17)$$

Therefore, following a relative transformation of fragment position and velocity compared to the Earth and a rotation of the fundamental plane of our coordinate system, we arrive at components in the ECI system. The final integration of the simulation reentry model is conducted in this system, resolving impact locations using an embedded model of Earth rotation.

Table 3.4 Coefficients for Static Atmosphere Model

Layer	ρ_0 kg/m ³	a_1 km ⁻¹	a_2 km ⁻²	a_3 km
$0 \leq h_a < 20$	1.2280	0.090764	-0.20452E-2	0
$20 \leq h_a < 60$	0.09013	0.16739	0.62669E-3	20
$60 \leq h_a < 100$	0.3104E-3	0.137	-0.78653E-3	60
$100 \leq h_a < 120$	0.366E-6	0.18533	0.15397E-2	100
$120 \leq h_a$	9.80665	-18.65220 (km ^{-1/2})	0.6124000 (km ^{-1/2})	116.4154 km

3.4 Fragment Reentry Model

For an initial guess, fragment impact locations on the surface are computed using a ballistic case neglecting atmosphere. A cross-range and down-range error ellipse is appended to the hit location to provide a confidence area of probable impact. This allows for uncertainty in the reentry model due to drag forces. A secondary impact model was constructed using an exponential falloff of density with height, based on a static Russian GOST atmosphere [53] with coefficients listed in Table 3.4. A static night-time atmosphere neglecting solar effects is assumed for altitudes above 120 km. Exospheric coefficients are chosen for a solar flux value of $F_0 = 150$:

$$\begin{aligned} \rho &= \rho_0 \exp(-a_1(h_a - a_3) + a_2(h_a - a_3)^2), h_a < 120 \\ \rho &= \rho_0 \exp(a_1 - a_2\sqrt{h_a - a_3}), h_a \geq 120 \end{aligned} \quad (3.18)$$

where ρ_0 is the mean density at the previous interface, and h_a is the altitude above sea level in kilometers. This density is then used to add a drag term to the velocity equation of motion [57]:

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_{ECI} - \frac{\rho(\mathbf{v} \cdot \mathbf{v})SC_d}{2m_i} \hat{\mathbf{e}}_v \quad (3.19)$$

where C_d is the ballistic drag coefficient for the fragment, and S is the area subject to the leading bow shock, which is a function dependent on mass of the individual fragment, and $\hat{\mathbf{e}}_v$ is a unit vector in the direction of the fragment velocity. The structure model used for reentry of fragments is a cylinder inscribed in a cubic volume (diameter equal to height). The equivalent

drag coefficient used was 1.7, as provided in [57, 24]. Deceleration due to air density is only one component of reentry. Pressure stress and mass loss due to ablation are modeled using material parameters:

$$\dot{m} = -\frac{S}{Q} \min\left(\frac{1}{2}C_H\rho v^3, \sigma T^4\right) \quad (3.20)$$

where Q is the heat of ablation (assumed to be $1\text{E}7 \text{ J/m}^3$), C_H is the coefficient of heat transfer (assumed constant at 0.1), σ is the Stefan-Boltzmann constant, and T is the temperature of thermal ionization of the surrounding gas (25,000 K). This equation governs the ablative mass loss until the mean pressure in the cylinder $p = 0.25C_d\rho v^2$ exceeds the yield strength of the material, at which point the energy deposition implies burnup of the fragment.

3.5 Uncertainty Analysis

In order to test the response of orbital dispersion with respect to uncertain initial fragment positions and velocities, a Gaussian noise is added to the mapping around the nominal center of mass. A standard deviation of 10% is assumed, resulting in deviations from the hydrodynamic simulations up to $\pm 30\%$. For a given orbit, 1000 random perturbations are integrated to impact, resulting in an average system behavior and a standard deviation representative of the uncertainty due to the initial conditions.

This procedure is completed for a database of 906 orbits chosen to impact at a fixed date. The orbital parameters for the nominal trajectory are sampled from a (a, e, i) space that represents the distribution of known NEOs, as shown in Figure 3.1. For each chosen deflection direction, the Monte Carlo procedure described above results in a characteristic behavior of a disrupted NEO on the range of orbits tested.

Given a fixed lead time in which to allow the fragmented target to disperse along its orbit, or even a minimal desired lead time, we have a point (or set) at which a desired intercept is achieved. It is clear, however, that not all approach vectors are equal. From a mission design perspective, the approach asymptote affects the transfer orbit, and therefore the cost (or even feasibility) of the mission [23]. For the present study, bounds on the approach asymptote from

a mission perspective are not considered. Rather, the direction in which the approach occurs is a deciding factor in the behavior of the fragmenting body.

A simple differential optimization routine is applied to this vector for each of the sampled impacting orbits. There are two degrees of freedom for each of these problems. The optimal pointing direction will be something of interest in short warning scenarios, since a drastic difference in the dispersion patterns can occur. For some of the orbits, a grid search of the approach asymptotes was done to quantify the range of impacting mass ratios.

CHAPTER 4. CHARACTERIZATION AND INTERCEPTION

This chapter addresses work towards on-orbit parallel computing for characterization and interception of a hazardous near-Earth object. The preliminary steps employed include the implementation of a simulated image-processing environment to allow for vision-based guidance, navigation, and control (GNC) of the interceptor spacecraft. Even flying similar parallel algorithms is not that far outside of flight heritage; image processing has been included as dedicated chips on many space missions. The addition of general purpose algorithm computing would be primarily a software change. This thesis gives a brief discussion of the characterization and interception plugins for the simulation environment developed by the author for the ADRC, including image rendering of a wireframe asteroid model, simply GNC operations, and environment estimation from LIDAR observations.

4.1 3D Target Polygon Model

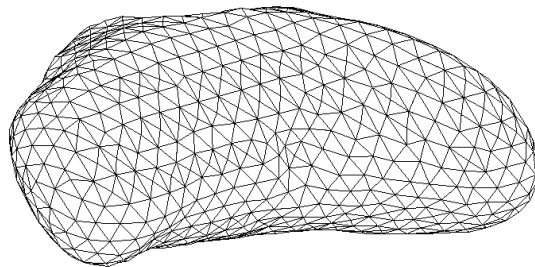


Figure 4.1 Surface Model for Eros.

This section addresses the rendering of the images in simulation memory that correspond to camera data to be processed by the spacecraft. We begin with a triangulation of surface points

represented as a 3D wireframe polygon model. The target model is derived from data for the asteroid 433 Eros [64], as shown in Figure 4.1. This data was collected by the NEAR mission, and has 200700 faces. This number of faces corresponds to the variable n_f of the computer model. Generally, the number of vertices needed to fill out the model, n_v , is strictly less than n_f , so it is more computationally efficient to do calculations on the vertices where possible. However, much of our information depends in some sense on an orientation of the body surface, for which the faces are necessary. A connectivity array stores the relationships between each set of vertex data and which face it applies to. If \mathbf{R}_1 , \mathbf{R}_2 , and \mathbf{R}_3 are the position vectors for the vertices of a face, as shown in Figure 4.2, then we can define for $1 \leq i \leq n_f$:

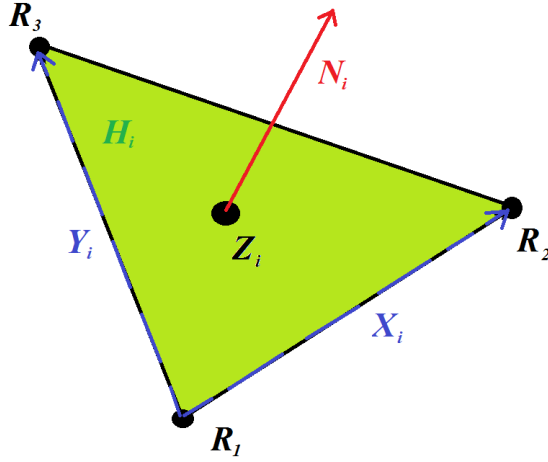


Figure 4.2 Facet Geometry Definitions.

$$\mathbf{X}_i = \mathbf{R}_2 - \mathbf{R}_1, \quad \mathbf{Y}_i = \mathbf{R}_3 - \mathbf{R}_1, \quad \mathbf{Z}_i = (\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3)/3 \quad (4.1)$$

The unit normal vectors of each face are calculated and stored using

$$\mathbf{N}_i = \mathbf{X}_i \times \mathbf{Y}_i, \quad \mathbf{N}_i \cdot \mathbf{Z}_i > 0 \quad (4.2)$$

The wireframe model uses a known rotation state of the target, though this information is not available to the spacecraft. A rotation matrix is applied to the initial conditions so that a simulated real state is known for the 3D model. To complete this system description, we

require a unit sun vector, $\hat{\mathbf{S}}$, in the direction of the sun from the target and a unit view vector, $\hat{\mathbf{V}}$, in the direction of the spacecraft from the target. We assume that the true center of mass of the target is at the origin on this coordinate system.

4.1.1 Camera Pointing and Focus Plane

In order to decouple the attitude and translation mechanics of the present simulation system, a perfect pointing was adopted for the camera. Thus, the choice of estimated center of mass or previously computed center of brightness is at the center of the camera focal plane, with the camera needing zero time to adjust to the new settings. Modifications to this scheme, including integration with spacecraft attitude are expected in the near future. The camera focal plane distance was chosen arbitrarily as the distance from the spacecraft to target. Since this plane is perpendicular to $\hat{\mathbf{V}}$ we can project the 3D model onto this plane by subtracting the component along this vector, thus the projected locations of the vertices are:

$$\mathbf{P}_j = \mathbf{R}_j - \hat{\mathbf{V}}, \quad 1 \leq j \leq 3 \quad (4.3)$$

To save computational time and storage space, a condensed array of these values are computed only for the faces satisfying $\mathbf{N}_i \cdot \hat{\mathbf{V}} > 0$. This results in only the faces visible to the spacecraft. Due to the decoupling of the camera dynamics, a direct orientation of this plane is not possible given the currently available information. Thus, a direction for camera up must be chosen. We choose unit vectors $\hat{\mathbf{U}}$ and $\hat{\mathbf{W}}$ such that

$$\hat{\mathbf{W}} = \hat{\mathbf{S}} \times \hat{\mathbf{V}} \quad \hat{\mathbf{U}} = -\hat{\mathbf{V}} \times \hat{\mathbf{W}} \quad (4.4)$$

Therefore, the sunlight will always come into frame from a horizontal direction, and up will always be orthogonal to the plane containing the target, the sun, and the spacecraft. The coordinates of each vertex in this plane can be computed as:

$$\tilde{\mathbf{P}}_j = \begin{bmatrix} u_j & w_j \end{bmatrix}^T = \begin{bmatrix} \mathbf{P}_j \cdot \hat{\mathbf{U}} & \mathbf{P}_j \cdot \hat{\mathbf{W}} \end{bmatrix}^T \quad (4.5)$$

Table 4.1 Simulated Camera Parameters

	High Resolution	Medium Resolution	Infrared
Resolution (x_r, y_r) (pixels)	1024×1024	1024×1024	512×512
Field of View (f_v) (radians)	2.05×10^{-3}	10.0×10^{-3}	10.0×10^{-3}
Pixel Size at 1,000 km (m)	2.0	9.8	19.5

At this point, what is visible to the spacecraft depends on camera parameters that interpret the real system. The resolution and field of view for the cameras simulated are listed in Table 4.1. At each time step, these are used to compute the half resolution of the image plane, R_h . If \mathbf{R}_T is the position vector representing the spacecraft in the target frame, then this can be computed as:

$$R_h = |\mathbf{R}_T| \tan\left(\frac{1}{2}f_v\right) \quad (4.6)$$

where f_v represents the field of view in radians (assumed isotropic). If x_r and y_r represent the x and y resolutions in the camera 2D pixel frame, then the information represented by each pixel corresponds to a size of $2R_h/x_r$ in the horizontal direction and $2R_h/y_r$ in the vertical direction.

4.1.2 Lighting Conditions

A simple flat shading model is used to calculate the brightness of the target surface. Given a diffuse lighting coefficient, k_d , and an ambient lighting coefficient, k_a , the corresponding brightness attributed to each face of the 3D model is

$$C_i = \begin{cases} k_d(\mathbf{N}_i \cdot \hat{\mathbf{S}}) + k_a, & \mathbf{N}_i \cdot \hat{\mathbf{S}} > 0 \\ 0, & \mathbf{N}_i \cdot \hat{\mathbf{S}} \leq 0 \end{cases} \quad (4.7)$$

where a value C_i of 1 corresponds to perfect reflectivity of the incident sunlight at that distance. A greyscale colormap is used for human visualization. The diffuse coefficient works closely in line with the albedo, and is chosen to be 0.25 for the simulations. Since the cameras we are using have very little time to resolve detail of the body, we assume that their lower

threshold for brightness is very low and therefore $k_a = 0$. This model is very fast, and can be computed completely in parallel for each face.

When additional lighting detail is desired, the following interpolation model due to Gouraud [65] is used. Let each vertex have a normal $\tilde{\mathbf{N}}_1$, $\tilde{\mathbf{N}}_2$, and $\tilde{\mathbf{N}}_3$ computed as an average of the adjacent faces, found from inverting the connectivity array. Then, a corresponding value for brightness at each vertex \tilde{C}_1 , \tilde{C}_2 , and \tilde{C}_3 is computed using the flat shading model equation. If a pixel representing the image plane at $[u \ w]^T$ is calculated to intercept this triangle, then an inverse squared weighted average is assigned to the pixel:

$$C(u, w) = \frac{\alpha_1}{\alpha_T} \tilde{C}_1 + \frac{\alpha_2}{\alpha_T} \tilde{C}_2 + \frac{\alpha_3}{\alpha_T} \tilde{C}_3 \quad (4.8)$$

where $\alpha_T = \alpha_1 + \alpha_2 + \alpha_3$ and

$$\alpha_j = \frac{1}{(u - u_j)^2 + (w - w_j)^2}, \quad 1 \leq j \leq 3 \quad (4.9)$$

4.1.3 Pixel Value Assignment

For either the flat or interpolated shading models, the facet at which a pixel intercepts the target projection must be determined. This is done on the GPU in parallel to reduce computational time, especially using a higher fidelity target model. Each pixel is assigned coordinates in the u, w plane representing the center of the box over which the pixel is integrated. This works well when the size of a pixel is on the scale of the surface facets, but an averaged value needs to be used at greater distances. To determine the intercepted triangle, we first compute the upper and lower bounds of each visible triangle in parallel. Then, a reduction is performed to get the outer limits of the target in the UW system. This process is represented in Figure 4.3. Pixels outside of this range can automatically be assigned a value of 0. In fact, only the location and value of nonzero image components are stored in the present computational model.

The arrays containing the limits of each facet are sorted, and then a scan of these arrays is completed. This lets us assign a vector of possible intersections for each pixel. This vector contains the IDs of all facets through which the pixel passes through the bounding box. This is generally a low number. For each candidate intersection, the convex hull of the triangle, H_i ,

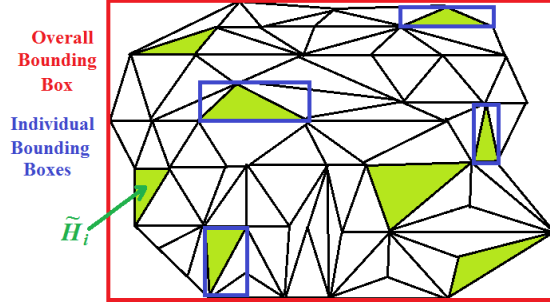


Figure 4.3 Parallel Trim Reduction Technique.

is identified, as shown in Figure 4.2. This is projected onto the UW plane, forming the set \tilde{H}_i shown in Figure 4.3. If $\tilde{\mathbf{X}}_i$ and $\tilde{\mathbf{Y}}_i$ are the projections of \mathbf{X}_i and \mathbf{Y}_i in the UW plane, then the coordinates of the pixel can be expressed as

$$\begin{bmatrix} u \\ w \end{bmatrix} = \tilde{\mathbf{P}}_1 + d_1 \tilde{\mathbf{X}}_i + d_2 \tilde{\mathbf{Y}}_i \quad (4.10)$$

which can be viewed as a transformation into the affine system defined by these two vectors. It is a well-known result from geometry that the point $[u \ w]^T$ lies within the triangle if $d_1 > 0$, $d_2 > 0$, and $d_1 + d_2 < 1$. This generally assigns a single facet as the possibility for intersection. However, at pixels near the boundary of the target, several potential intersection may occur. In these cases, the closest triangle ($\mathbf{Z}_i \cdot \hat{\mathbf{V}}$ largest) is chosen.

For the 3D LIDAR measurement, we interpolate based on the distances to the vertices of the intercepted triangle in the UW plane. Using the same weighting constants as the interpolated brightness model, we compute

$$\mathbf{L}_k = \frac{\alpha_1}{\alpha_T} \mathbf{R}_1 + \frac{\alpha_2}{\alpha_T} \mathbf{R}_2 + \frac{\alpha_3}{\alpha_T} \mathbf{R}_3 = \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}^T \quad (4.11)$$

where \mathbf{L} is a vector measurement of the reflection in Cartesian coordinates centered on the target. We also have $1 \leq k \leq n_p$, where the total number of measurements, n_p , grows with time.

4.2 Line of Sight Vector

For the present implementation, the estimated line of sight vector is chosen to be the center of brightness. If $I(u, v)$ is a matrix storing the values of brightness for the image, then the centroid of the image array is computed using an weighted average

$$\bar{u} = \frac{\sum I(u, v)u}{\sum I(u, v)}, \quad \bar{v} = \frac{\sum I(u, v)v}{\sum I(u, v)} \quad (4.12)$$

The line of sight vector in the spacecraft frame is therefore

$$\lambda = \begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix} - \mathbf{R}_T \quad (4.13)$$

4.3 Estimation of Target Reference Frame

Flash LIDAR measurements are desired as a way to reinforce optical measurements of target size and shape data. Unfortunately, we must be able to interpolate this from undersampled areas of the target. The present simulation model builds a least-squares energy ellipsoid representing the LIDAR measurements. Let A be a symmetric, positive definite matrix formed by the vector \mathbf{A} :

$$A = \begin{bmatrix} A_1 & A_2 & A_3 \\ A_2 & A_4 & A_5 \\ A_3 & A_5 & A_6 \end{bmatrix} \quad (4.14)$$

and E represent the quadratic form produced by \mathbf{A}

$$E_k = \mathbf{L}_k^T \mathbf{A} \mathbf{L}_k = A_1 x_k^2 + 2A_2 x_k y_k + 2A_3 x_k z_k + A_4 y_k^2 + 2A_5 y_k z_k + A_6 z_k^2 \quad (4.15)$$

Then, for nonzero integer values, m , the residual function

$$r_k = r(\mathbf{L}_k) = E^m - 1 \quad (4.16)$$

represents an energy measure of distance from the ellipsoid $E = 1$. Various values of m have interesting properties and are readily solved by nonlinear least squares methods. We chose the Levenberg-Marquardt method with line search, requiring QR factorization of the Jacobian at each linearization [66]. Results are shown for the case of $m = 1$. In this case, the residual is a linear function of the parameters \mathbf{A} , reducing the problem to a linear least squares parameter optimization. The rows of the constant Jacobian are readily computed as

$$J_k = \frac{\partial r_k}{\partial \mathbf{A}} = \begin{bmatrix} x_k^2 & 2x_k y_k & 2x_k z_k & y_k^2 & 2y_k z_k & z_k^2 \end{bmatrix} \quad (4.17)$$

This process works exceptionally for random input data. However, using simulated LIDAR measurements results in an overshoot by an order of magnitude followed by a slow correction process. Due to measurements being taken at close angular separation, heteroscedasticity in the input data is observed. This results from oversampling in some regions of the target and vastly undersampling other regions. Therefore, using a conditional variance, Ω , of the expected value $-r_k$ given J_k a weighted least squares problem is defined

$$\min_A \sum r_k^T \Omega^{-1} r_k \quad (4.18)$$

Assuming that Ω is diagonal, and that individual measurements are not dependent on one another, we can form the solution parameter vector

$$\mathbf{A} = -(J^T \Omega^{-1} J)^{-1} J^T \Omega^{-1} r_k \quad (4.19)$$

Choosing a weighting scheme such that

$$\omega_k = \Omega_{kk}^{-1/2} \quad (4.20)$$

Then, it can be shown that we have a linear least squares problem in the variables $\tilde{r}_k = \omega_k r_k$ and $\tilde{J}_k = \omega_k J_k$. While initial attempts to determine weights used a measure of data density to determine variance, it was quickly determined that this was not an ideal implementation. For large numbers of measurements, the density calculation is equivalent to a nearest neighbor search, which scales at best as $n_p \log n_p$ and at worst as n_p^2 . The formulation is particularly

vulnerable to the weighting problem due to E being positive definite. Therefore, under the original approach $|r_k| < 1$ for all $r_k < 0$, while $|r_k|$ is unbounded for $r_k > 0$. This tends to result in overshoot of the longest ellipsoid axis to minimize errors in oversampled regions, while allowing for large physical deviations for points with $E < 1$. Taking into account this issue, a weighting scheme to force inclusion of these points was used:

$$\omega_k = \frac{1}{1 + E_k} \quad (4.21)$$

Once suitable values for the matrix A are found, the representative ellipsoid can inform us of the target orientation vector frame. We have the eigenvector decomposition $A = VDV^T$, where the elements of the matrix D contain the eigenvalues of A , which correspond to the ellipsoid semiaxis parameters, a , b , and c :

$$D = \text{diag} \left(\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{c^2} \right) \quad (4.22)$$

and the matrix V contains the unit eigenvectors of A , which corresponds to the new target coordinate frame. This coordinate frame can then be used for orbit determination and operations, and the combined measurements can be used for a surface bound estimate to first order.

CHAPTER 5. COMPUTATIONAL STRUCTURE

This chapter addresses the parallelization and designed computational structure for the present simulation model. A brief introduction into the CPU parallelization ideas is presented, followed by a general description of GPU computing. The particulars of the high-level language access are discussed, and finally the hardware and software tested are given.

5.1 CPU Parallelization

inding a motivation for parallelization of a particular algorithm is usually not a difficult task. Even basic computations repeated for design cycle purposes, optimization, or other forms of iterative improvement, can explode the required computing time to unacceptable levels. The case of the application problem presented later is an example of where analysis of input conditions requires repeated runs of the model simulation. With millions of required calculations, short time steps or repeated runs become prohibitive in terms of computational cost. Many researchers believe that parallelizing their algorithms for compiled simulations is only necessary if they plan to run them on a supercomputer. However, modern multi-core desktop systems are readily available in most research environments, and proper use of parallelization techniques can reduce time-to-solution for simulation programs. Two main types of parallel platforms are available, and they have corresponding techniques and libraries for their use. Two open source libraries, OpenMP and Message Passing Interface (MPI), are discussed in this paper. The reader is provided with examples of relevant grammar and library references needed, and is referred to other sources for further study.

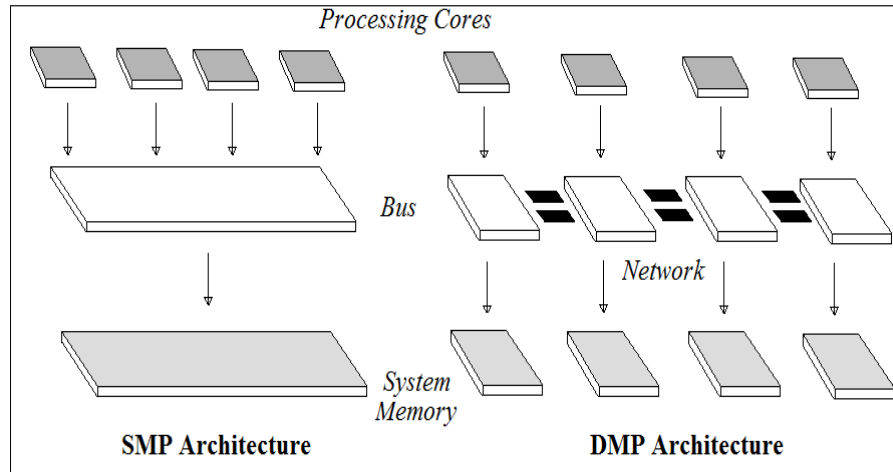


Figure 5.1 Comparison of SMP and DMP Architecture

5.1.1 Description of Parallel Architectures

As previously mentioned, there are two main types of parallel computer systems. The first is called Shared Memory Parallel (SMP) Architecture, while the second is called Distributed Memory Parallel (DMP) Architecture. Each computer system, often identified as its printed circuit board (also known as a system board, main board, or motherboard), will be referred to as a node [67]. Each Central Processing Unit (CPU) is represented virtually as the capability to run a single program (or thread), and will be referred to as a core. In the case where a system has one node, but many cores, the cores can all access the same memory bank. This type of computer, common of multi-core desktops, is a standalone SMP architecture. DMP systems will often have several nodes, each with its own memory bank that is unreadable by the other nodes. Communication that is required between nodes is handled using networking technology or other communication equipment [67, 68]. Figure 5.1 shows a visual comparison of SMP and DMP architectures. Traditional DMP parallelization can have more cores than physical nodes, but are still treated as if each core is a node. Multi-core systems that are networked together are handled by running threads on each core independently. Composite approaches utilize one library for communications between physical nodes while treating each node as a SMP system. These approaches can be more complicated and prone to programming errors, but offer benefits in memory usage and efficiency. Figure 5.2 shows an example composite

architecture, divided into SMP and DMP regions. More complicated architectures are also used, though combinations of these two are most likely to be encountered by the average researcher.

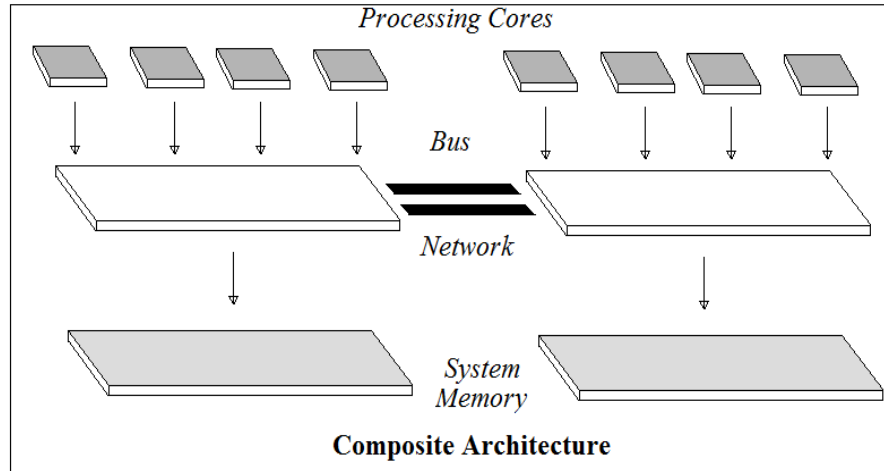


Figure 5.2 Example of a Composite SMP/DMP Architecture

5.1.2 OpenMP and MPI

OpenMP is an Application Program Interface (API) that facilitates parallelism in SMP environments. It consists of a library of architecture-dependent variables, functions, and sub-routines that are useful in speeding up the processing of algorithms. It also consists of compiler directives through which the programmer can direct the creation of threads, the use of memory, and other application-specific needs[69]. OpenMP specifications exist primarily for C, C++, and Fortran programs. They extend the basic language specifications to add loops, constructs, and communication necessary to fully utilize a SMP system, while maintaining the portability of using a compiled language for simulation. Programming directives are given as comments, so compiling them on a system without OpenMP results in the creation of a serial program without changes being made to the source code. Specific syntax was used in the Fortran language, though equivalent translations in C and C++ are available in the OpenMP specification guide[69].

Several of the routines available in the OpenMP library can be used within parallel regions

to set and determine the local working environment. OpenMP works through directing areas of a serial program to spawn teams of computing threads at specific points. These teams carry out the directed task, and then are reintegrated back into a serial section of the program. This allows for actions typically reserved for serial codes (such as input and output) to be taken, and for parallel execution of specific tasks. It often allows existing serial code to be parallelized in a few steps by identifying key loop and memory structures to be split up among threads. Parallel regions in a SMP program can be nested, though the exact application determines whether or not an error would occur. Further programming guidelines using OpenMP can be found in the API manual[69].

Contrasted to OpenMP, traditional implementations of Message Passing Interface take place using an external process scheduler. A package including MPI and MPI-compliant compilers must be installed on the DMP architecture. Specific implementation, including communication among nodes, must be handled by the programmer. For this reason, the algorithms presented in this paper will be restricted to OpenMP, though the relevant commands for MPI implementation can be found in References [68, 70]. Since the spawn of processes is handled outside of the source code, the simulation needs to be programmed with the knowledge that an individual executable will be run by each computing thread.

5.2 GPU Computing

As the Central Processing Unit (CPU) of the modern personal computers and workstations evolved, there was a substantial problem overcoming thermal issues that accompanied additional computing power. One solution to these thermal issues was to design multi-core CPUs capable of handling several computing threads in parallel [71]. Additionally, as graphics for games and commercial applications grew more demanding, a separate Graphics Processing Unit (GPU) was implemented to allow the screen rendering to be performed by an alternate processing chip. In most graphic applications, either pixels or sections of the display can be calculated separately from each other. Due to this fact, the GPU evolved into a many-core parallel structure, trading core clock speed for number of parallel computing threads. At the time of writing, retail GPU chips are available with up to 512 cores and over 1 trillion floating

point operations per second (FLOPs) of theoretical computing power [71, 72]. For comparison, a standard workstation processor averages 10-20 GFLOPs. Due to this leap in technology, the GPU has been considered by many the ideal commercially-available massively parallel architecture.

Early simulations and computing programs written for execution on the GPU were done using graphics processing languages. These approaches cleverly harnessed the computing power available by casting a problem as one involving textures and pixels. DirectCompute, OpenGL, and other languages were very powerful methods for moving computation from the CPU to the GPU, freeing resources and allowing problems to scale more easily [73]. However, for general scientific computing this method had a steep learning curve. In 2006, NVIDIA launched the Compute Unified Device Architecture (CUDA), which allowed compilation of GPU executables to be created through extensions to the C language. This approach is highly portable, and an example of the versatility of general scientific computing on the GPU [72]. The present simulation package uses a mixture of CUDA C and the Portland Group's CUDA FORTRAN [74].

5.2.1 Hardware and Implementation

A variety of hardware was available for this project, with a substantial difference in performance. This allowed us to get reasonable estimates on the computational cost of this simulation, in comparison to LINPACK performance numbers. Performance can vary based on the type of arrays used, and the number of threads dedicated to each GPU calculation. These factors are determined by the CUDA Compute Capability (CUDA CC), which is a property of the GPU [72]. These cost estimates are used to determine hardware performance on the various systems. A summary of the hardware used is shown in Table 5.1 (Note: all CPUs are Intel brand, and all GPUs are NVIDIA brand).

5.2.2 Hydrodynamic Calculations

Each thread on the GPU calculates the state variable change for one fragment, with the GPU kernel limited to one time step. This is necessary because the positions of the planets and other gravitating bodies must be calculated and transferred to the GPU at each time step.

Table 5.1 Hardware for Benchmark Systems

System	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
CPU	1x Core2 Q6600	1x Core2 Q6600	1x Xeon X5550	2x Xeon E5520	2x Xeon X5650
CPU Cores	4	4	4	8	12
CPU TPEAK	9.6 GFLOPs	9.6 GFLOPs	12.8 GFLOPs	21.36 GFLOPs	32.04 GFLOPs
GPU	1x 8800GTS	1x GTX470	1x GTX480	4x Tesla c1060	4x Tesla c2050
GPU Cores	112	448	480	960	1792
GPU TPEAK	84 GFLOPs	324 GFLOPs	385 GFLOPs	336 GFLOPs	2060 GFLOPs
CUDA CC	CC 1.0	CC 2.0	CC 2.0	CC 1.3	CC 2.0

Additionally, the positions of fragments at each integration substep are shared among multiple GPUs and CPU threads. For this reason, the present hydrodynamics model is predominantly bandwidth-limited for small data sets. While grid information is not retained, one of the disadvantages of the SPH hydrocode is that neighboring particles must be calculated at each time step. Our approach in this model is to create a bounding volume for each SPH particle and perform the same Sort and Sweep in parallel as used to detect collisions in the orbital model [42]. We retain the information for neighbors connected by material strength, as well as carrying neighbor information through the correction step of the integrator. This results in a 28% performance improvement over recalculating neighbors at both the prediction and correction steps, while allowing for a variable time step based on the Courant condition [32, 33]:

$$\delta t = \min_i \frac{h_i}{c_i} \quad (5.1)$$

where c is the local sound speed. While the reduction operation to determine the new time step can be done in parallel, all GPU threads must have position information for all particles to determine neighbors. This requirement could be eliminated through clever domain decomposition, but there is a tradeoff between associating a mesh to the model and taking advantage of contiguous memory sections of particles. Load balancing would also require additional communication between GPUs, which has an impact on performance, as PCI-E bandwidth is one of the limiting factors in GPU acceleration [71].

Our memory model for this simulation includes a shared host memory, distributed device memory for each GPU, and data transfers between them handled through explicit array trans-

fer. Each block of compute threads on the GPU takes the data it needs from the global device memory when the kernel reaches its block. This is an important factor, because the varying compute capabilities have different limitations on this block memory, changing the number of threads that may be used in the calculation. Constants are transferred to all GPU memories implicitly using a pointer to the host constant value.

5.2.3 Orbital Calculations

The implementation of the simulation is conducted in two ways. The first version uses CUDA extensions to the C language, and bindings for these kernels into existing Fortran 90 code. The second version uses CUDA Fortran, developed by the PGI group [74]. Since different sets of parameters are computationally independent of one another, one way to conduct parameter variation would be to have each computing thread handle a combined set of timing and scaling parameters, as shown in Figure 5.3. However, for memory management and to lower the amount of repeated calculations, we utilize the parallel nature of the fragments themselves as a basis for computation.

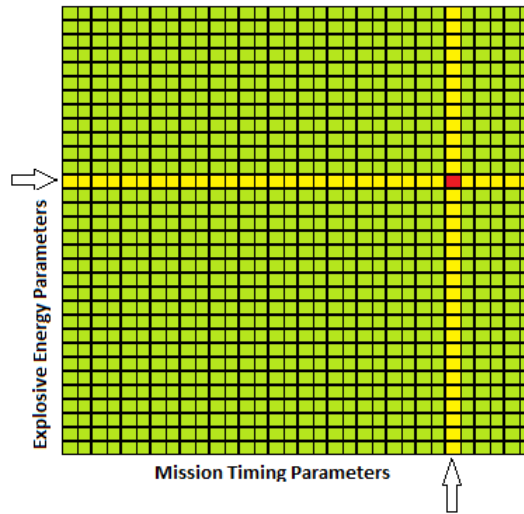


Figure 5.3 Visualization of Parameter Sweep Method.

We fix a value for the timing parameters, allowing the nominal orbit to be calculated and shared among all versions of memory (discussed below). Then, looping through the velocity scaling parameter values, each thread on the GPU calculates the state variable change for one

fragment, with the GPU kernel limited to one time step. This is necessary because the positions of the planets and other gravitating bodies must be calculated and transferred to the GPU at each time step. The final algorithm is shown in block diagram form in Figure 5.4.

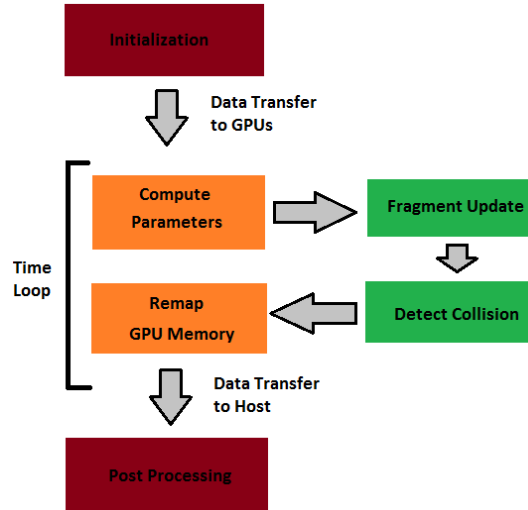


Figure 5.4 Block Diagram of Simulation Procedure.

5.2.4 Memory Model and Explicit Communication

Our memory model for this simulation includes a shared host memory, distributed device memory for each GPU, and data transfers between them handled through explicit array transfer. Each block of compute threads on the GPU takes the data it needs from the global device memory when the kernel reaches its block. This is an important factor, because hardware compute capabilities have different limitations on this block memory, changing the number of threads that may be used in the calculation. Constants are transferred to all GPU memories implicitly using a pointer to the host constant value. Figure 5.5 shows an overview of this computational memory model.

The explicit communication needed in the simulation for a single set of parameters is shown in the following pseudocode:

```

*Transfer state variable arrays Host to Device
Begin loop through time steps
    Calculate planetary positions at subintervals on host
  
```

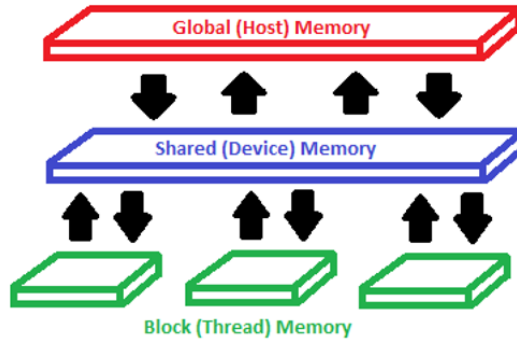


Figure 5.5 Visualization of Memory Model.

```

*Transfer planet position arrays Host to Device
Launch GPU kernels to update state variables
Calculate closest approach to Earth on GPU
*Overwrite state arrays with Device to Device memory transfer
End time step loop
*Transfer final state back to Host, including close approach
Postprocessing on Host

```

5.2.5 Bandwidth Use and Serial Computation

One of the primary limitations of GPU acceleration is the PCI bus connecting the Host to the Device. Communication between these two sets of memory, or between individual GPUs, is therefore very expensive [72, 74]. Thus, the use of communication bandwidth should be minimized to achieve optimal performance. For example, an early implementation of a numerical integrator may be a subroutine that reads state variables from Host memory, computes the updated state on the GPU, and returns the next state to the Host. Unless the computation of the updated state is extremely intensive, this approach will not yield a high speedup over a CPU implementation. For smaller problems, the approach discussed above is preferred. Though the device code may be more complicated, this method was found to be an order of magnitude faster for a fragmented NEO system with 18,220 fragments. With 511,744 fragments, the limitation from data transfer is less pronounced, though leaving arrays in device memory yielded a simulation that ran 5 times faster.

While modern dedicated compute GPUs have a high amount of onboard memory, it usually is far less than system memory. Though it may seem advantageous to calculate parameters for every time step before the start of the simulation, the resulting arrays can be quite large. Each model of GPU has a limited number of memory registers available to each computing block of threads [72]. Therefore, the use of several large arrays can actually slow down the simulation in some cases, by lowering the number of threads below the maximum allowed by the architecture. This trades off directly with the added expense of calculating parameters on the Host at each timestep. For the present work, calculating planetary positions at each step was found to be preferable to using a large pre-calculated array. For some hardware, sufficient GPU memory was not available.

5.2.6 Integration with SMP Computation

Some of the systems used to test this work had multiple GPUs. This was used to the advantage of the program by launching several Shared Memory Parallel (SMP) threads on the Host CPU. Each thread, or team of threads, was assigned a GPU on which to launch compute kernels. The calculations were conducted on portions of the state variable arrays. Since no interaction among fragments was assumed, the GPUs did not have to communicate the states of the fragments they were responsible for. This was found to be an extremely effective setup for large data sets, and scaled almost linearly to the number of GPUs used with some overhead for data transfer to partial arrays on the GPUs.

CHAPTER 6. RESULTS

This chapter details the results of the developed simulation package, and insights gained into the importance of several variables of the problem. A short description of the resulting trajectory analysis is given, followed by a discussion of the computational optimization steps employed. Finally, the author engages in a brief discussion of the conclusions drawn.

6.1 Semianalytical Standoff Model

To compare with the results of similar models, a NEO with a radius of 500 m was chosen, and the energy from neutrons was assumed to be the primary energy available of the overall stated yield of 1 MtTNT. The results for the momentum coupling approaches are shown in Figure 6.1, and provide an adequate range for velocity change due to momentum transfer. This approach lacks the benefit of an optimum value and predicts a drastic decrease in deflection effectiveness with increased standoff distance. Fig. 8 shows the results of the solid body pressure blow-off model with γ values of 1.1 and 1.3. Two basic scattering efficiency values are chosen to emphasize the change in results due to different assumptions of ejecta distribution. An optimal standoff distance of about 200 m for an ideal spherical model of this 1-km NEO example can be noticed in Figure 6.1. It is also interesting to notice that an optimal standoff distance of 20 m was obtained for the same 1-km NEO in [14]. The neutron assumption differs from the energy breakdown of the blast model presented, but the hydrodynamics simulations provide the changes necessary to confirm these results. Due to the reliance of the model on energy methods, our simple model predicts significant reduction in ΔV at small standoff distances, which may not be acceptable for a deflection mission. Accounting for changes such as melting or sublimation which might also make decreasing the optimal standoff distance should be a

topic of continuing research.

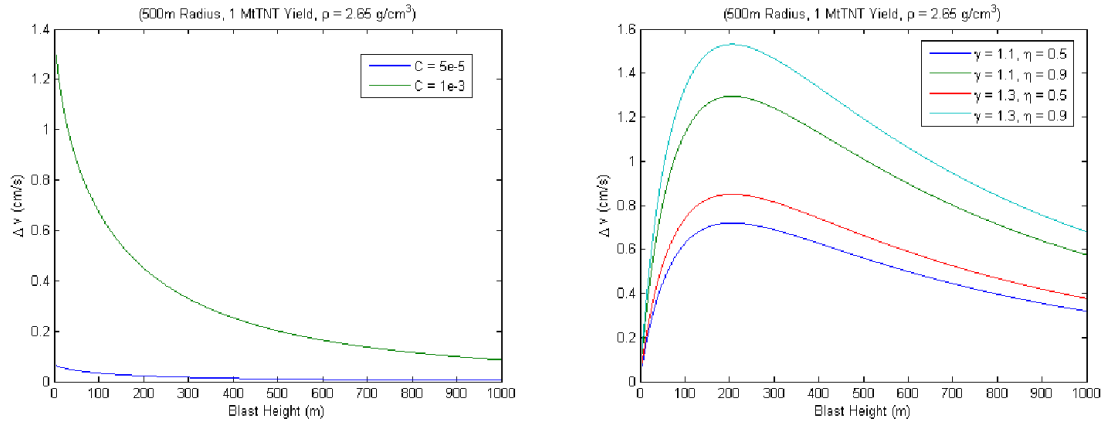


Figure 6.1 Comparison of velocity changes for momentum coupling and solid body model.

The resulting compression waves in the remaining material appear to be small enough that further fragmentation of the NEO is unlikely for large bodies [17], though many authors suggest that the initial blast may be enough to damage the material structure [10, 15]. Computer simulations will be valuable in determining the momentum of fragments and the results of the cratering process. Likelihood of the fragments impacting Earth would be a significant factor in choosing the method of deflection [50]. As an initial effort in establishing a model of nuclear standoff deflection, this comparison has shown that increased knowledge of object composition and structure is needed to accurately reflect the results of a nuclear standoff explosion. Corrections are likely needed for anisotropies of the nuclear blast and non-thermal interactions with material in the layer of deposition. A more rigorous model of the blast itself is therefore necessary to establish a general description of nuclear standoff explosions.

The generation of NEOs through collisions resulted in many fragmented asteroids with porous surfaces. Detailed analysis of a target asteroid would be needed to determine surface composition and distribution [11, 49]. While large asteroids are usually solid bodies, smaller asteroids can have porosities ranging up to 70 percent [49], and therefore present a considerable problem for deflection. Bodies with high porosity, or rubble piles, have significant changes to material strength characteristics. These changes are complicated, and analytical formulas to account for them have not been agreed upon [14]. It is possible that at very high energies

even porous NEOs behave similarly to solid bodies [15], and this is a future study topic worth investigating using high-fidelity simulation.

Deflection attempts involving nuclear explosions involve the interaction of high-energy neutrons. The primary mode of interaction with these neutrons is inelastic scattering within surface material. The energy transfer is therefore primarily thermal excitation [51]. The characteristic penetration depth for asteroid material is often assumed to be around 20 cm [17], though different models for the mechanism of energy transfer predict different depths. In general, the specific energy is weakly dependent on incident angle, but the model considered in this paper assumes a uniform distribution of energy over the irradiated surface. Absorption and scattering of photons are the simplest methods of energy transfer, though gamma radiation may be subject to more complicated mechanisms [51]. Most X-ray photons contain enough energy to ionize material, which makes crystal structure more likely to fragment and may lower the threshold necessary for the material to sublime. More advanced approaches should account for the absorption changes with incident angle.

6.2 Disruption Results

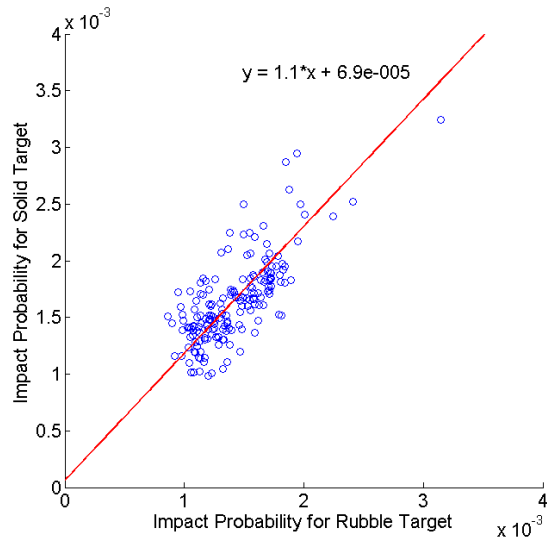


Figure 6.2 Relative Performance for Surface Impactor.

In order to address the effectiveness of different fragmentation methods, we compare the

mass remaining on impacting trajectories (including the uncertainty from the Monte Carlo process) against other methods for each orbit. For example, Figure 6.2 shows the relative impacting mass for the surface penetrator in both the solid and the rubble-pile targets. On average across the orbits tested, the impacting mass was 10% higher for the solid target compared to the rubble target for deflections in the radial direction. Estimates like this will eventually allow for tabular look-up of performance for various methods without direct computation. It was also found that impacting mass for the solid target was 20% higher than the rubble target in the transverse direction.

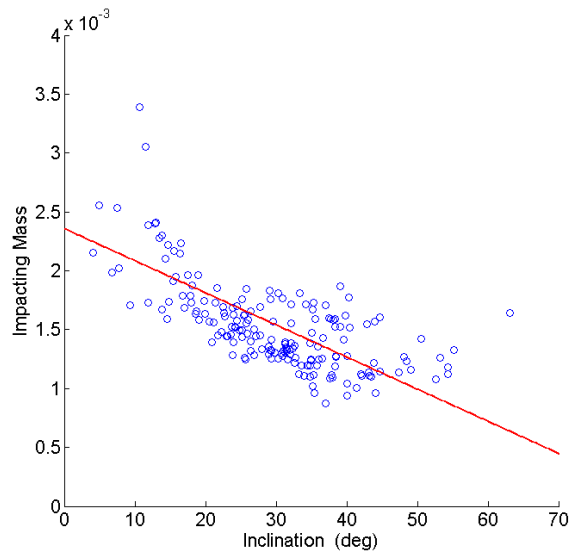


Figure 6.3 Impacting Mass for Subsurface Explosion on Orbits with Varying Inclination.

No strong correlation was found for the semimajor axis or eccentricity of the NEO orbit with only 15 days of lead time. However, deflections on orbits with high inclination were more effective, as shown in Figure 6.3 for the subsurface case. Ejecta velocities for the dynamic surface burst (at 6.1 km/s) were within the 10% assumed noise range compared to a static buried explosive, as shown in Figure 6.4. Thus, an emphasis might be placed on hypervelocity intercept and guidance technology rather than a rendezvous mission. One possible interceptor design includes an aluminum impactor followed by an explosive. With both interceptors impacting at 6.1 km/s, the resulting ejecta speed is on average 25% higher than the single surface blast, with a standard deviation of 5.3%. Figure 6.5 shows the relative velocities for these cases, which

results in 20% lower impacting mass on most orbits tested for the 54 m initial spherical target due to the proposed HAIV concept.

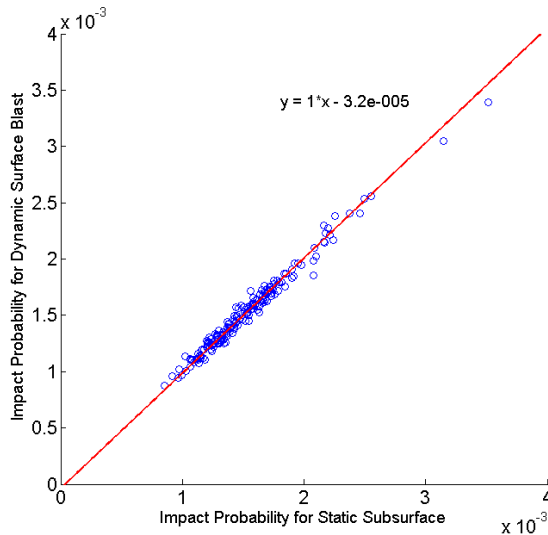


Figure 6.4 Impacting Mass Comparison for Subsurface and Dynamic Surface Cases.

6.3 Nominal Fragmentation Behavior

Orbital propagation for the Apophis-like orbit described earlier yields insight into the types of variables that effect the behavior of a disrupted NEO on a collision trajectory with the Earth. Using identical time step constraints we achieve results for impacting mass between 0.1% and 7.1%, consistent with previous simulations. These results, with dominant dispersion speeds along the coordinate axes of the LVLH system, are shown in Figure 6.6. However, a tightening of time step restrictions to a maximum step of 1 minute tells a much different story, with the impact values converging to 0.6-26.1% of the total asteroid mass after 15 days of dispersion along an Apophis-like trajectory, as shown in Figure 6.7. This brings up the point that, while longer time steps are advantageous in overall computational cost, accuracy is sacrificed, particularly within the sphere of influence of the Earth.

Table 6.1 shows the number of impact bodies for each scenario described previously, both for the large time step and the small time step models. Also shown is the total fraction of the initial mass estimated to hit the planet. Not shown are bodies that may be captured

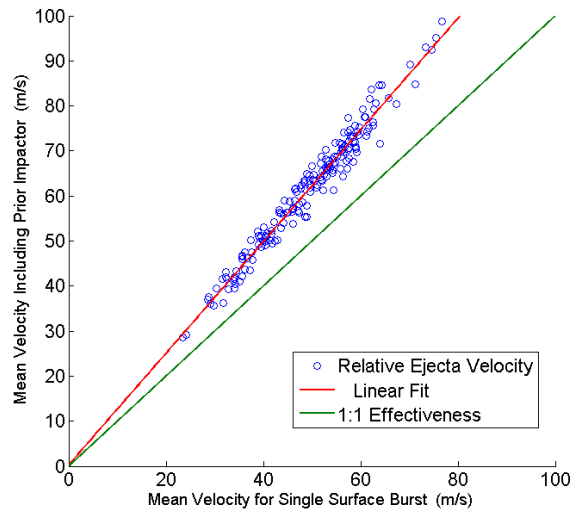


Figure 6.5 Mean Ejecta Velocity for Single and Double Impactor Cases.

or perturbed to future impacting orbits, which is a subject of current research. It is clear that a maximum time step of 1 hour is not sufficient to resolve the details we require. This is further emphasized by the results for a dispersion along a full orbit, which are shown in Table 6.2. These values clearly indicate that previous accuracy requirements underestimate impacting fragments, and do not yield statistically representative results. The results for a 300 kT disruption are contrasted to results from initial velocity conditions scaled up to simulate the increased kinetic energy provided by a 1 MT explosion in Table 6.3. The decreased time step is further justified by comparing results to the ODE solver package available in Matlab, shown in Figure 6.8.

It should be noted that, while a large fraction of the initial asteroid mass misses the Earth following the fragmentation attempt, it is spread over an area substantially larger than the initial impact plane. Fragments are ejected normal to the nominal orbit plane from part of the explosion-induced velocity, and therefore the Earth faces something far more like a cloud than an individual impacting body. While many of these fragments are small enough to burn up in the atmosphere, the Earth passes through a large cloud of them while rotating to expose new target areas over the course of several hours. A map of fragment impacts relative to a fixed rotating meridian is shown in Figure 6.9 on the left for a deflection along the radial direction

Table 6.1 Number and Mass Ratio of Impacting Fragments

Long Time Step	Number	Mass Ratio	Short Time Step	Number	Mass Ratio
+Radial (1 hour)	85	1.7114E-4	+Radial (1 min)	201	6.5036E-3
-Radial (1 hour)	87	1.4754E-3	-Radial (1 min)	117	5.8318E-3
+Transverse (1 hour)	2190	7.1259E-2	+Transverse (1 min)	6283	0.26061
-Transverse (1 hour)	2151	7.1007E-2	-Transverse (1 min)	5967	0.24096
+Normal (1 hour)	113	4.9174E-3	+Normal (1 min)	500	3.2357E-2
-Normal (1 hour)	124	5.5269E-3	-Normal (1 min)	523	3.4444E-2
			Perpendicular Velocity Direction	210 5652	8.2003E-3 0.18602

Table 6.2 Number and Mass Ratio of Impacting Fragments after 1 Orbit

Long Time Step	Number	Mass Ratio	Short Time Step	Number	Mass Ratio
+Radial (1 hour)	0	0.0000	+Radial (1 min)	70	9.8319E-5
-Radial (1 hour)	0	0.0000	-Radial (1 min)	75	4.2690E-4
+Transverse (1 hour)	0	0.0000	+Transverse (1 min)	124	5.5841E-3
-Transverse (1 hour)	0	0.0000	-Transverse (1 min)	114	3.8744E-3
+Normal (1 hour)	0	0.0000	+Normal (1 min)	73	1.5628E-4
-Normal (1 hour)	0	0.0000	-Normal (1 min)	78	3.0914E-3

Table 6.3 Impacting Fragments for 1 MT Disruption after 15 day Dispersion

Long Time Step	Number	Mass Ratio	Short Time Step	Number	Mass Ratio
+Radial (1 hour)	75	2.3150E-4	+Radial (1 min)	88	4.6531E-4
-Radial (1 hour)	75	8.9270E-5	-Radial (1 min)	83	5.4553E-4
+Transverse (1 hour)	1019	3.6597E-2	+Transverse (1 min)	3038	0.13655
-Transverse (1 hour)	954	3.3698E-2	-Transverse (1 min)	2637	0.12496
+Normal (1 hour)	81	4.7705E-3	+Normal (1 min)	148	1.3163E-2
-Normal (1 hour)	85	6.5133E-3	-Normal (1 min)	147	1.4271E-2

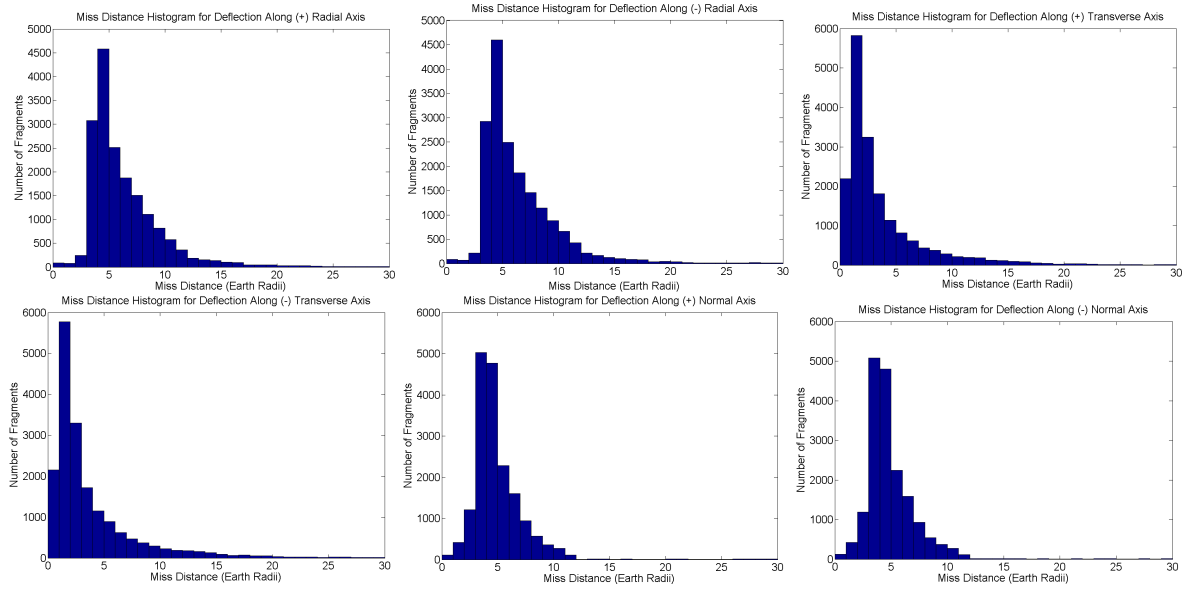


Figure 6.6 Miss Distance Histograms for 1 Hour Maximum Time Step Simulations.

resulting in 0.6% of the original asteroid mass on impact trajectories. A clustering of strikes on one side of the globe can be seen, as well as strikes from bodies that impact over 4 hours later than the first fragment, as the Earth rotates. Strikes predominantly in the Pacific Ocean and East Asia were observed for this scenario.

It is clear from these preliminary simulation results that the radial deflection case is optimal with such short lead time, Figure 6.7 shows that transverse deflections can still be catastrophic if the NEO is fragmented rather than deflected, with up to 26% of the original mass still on impacting trajectories. Furthermore, it can also be seen in Figure 6.9 on the right that this deflection attempt results in a concentrated band of material striking the Earth. It has been hypothesized that dense clusters of small bodies actually shield each other from some measure of reentry heating, and therefore have improved atmospheric penetration. In that scenario, fragmentation would not be beneficial and great care should be taken to deflect the orbit of the NEO rather than disrupt it along the flight path. At this time it is believed that deflection mission designs for radial deflection are preferable for short warning time mission planning utilizing nuclear explosions.

Also important to the viability of fragmentation is orbit geometry. Past analysis indicated that circular orbits (and orbits for which the interaction with the Earth occurred near aphelion)

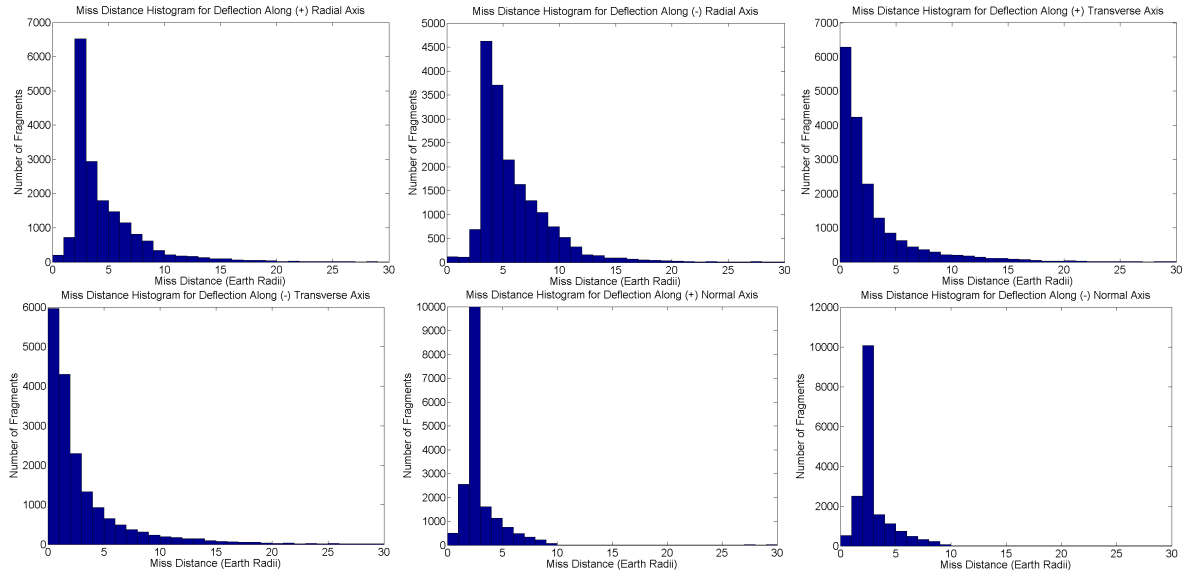


Figure 6.7 Miss Distance Histograms for 1 Minute Maximum Time Step Simulations.

resulted in particularly large fractions of impacting mass. This same phenomenon is observed in the present results for deflections along the transverse axis. It is thought that for short duration dispersion, such as the 15 day hypothetical mission presented here, deflection along the velocity direction is substantially less effective than perpendicular to it. The results of these two cases are also presented in Table 6.1, in which deflection along the velocity direction yields an impacting mass fraction similar to that for transverse deflection and deflection perpendicular to the velocity direction is comparable to radial deflection.

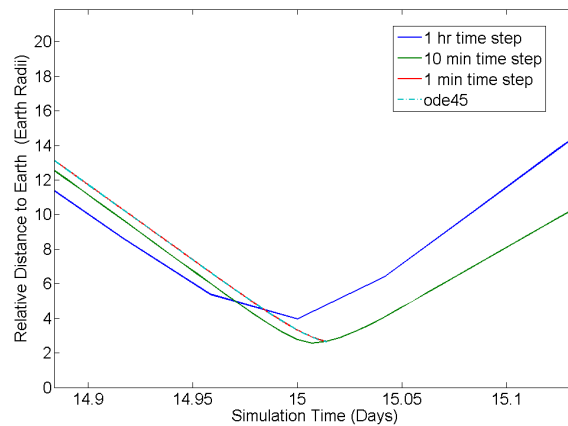


Figure 6.8 Comparison to Matlab Solver using Varied Time Steps.

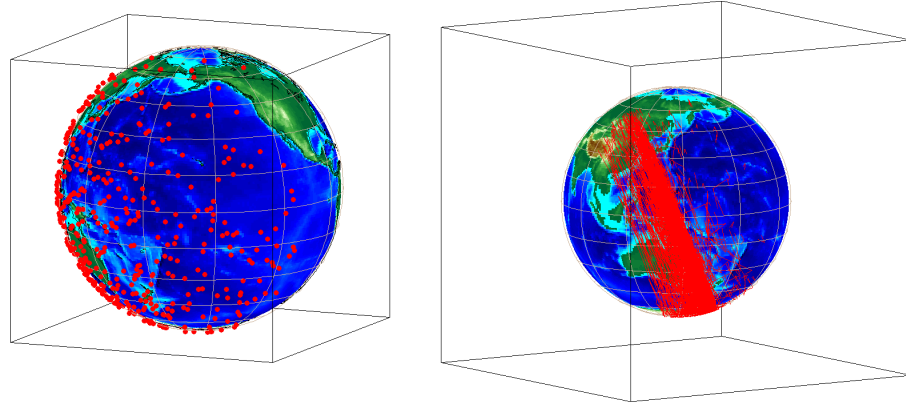


Figure 6.9 Impact Locations on a Fixed Earth after Radial and Transverse Deflections.

6.3.1 Timing and Scaling Parameter Variation

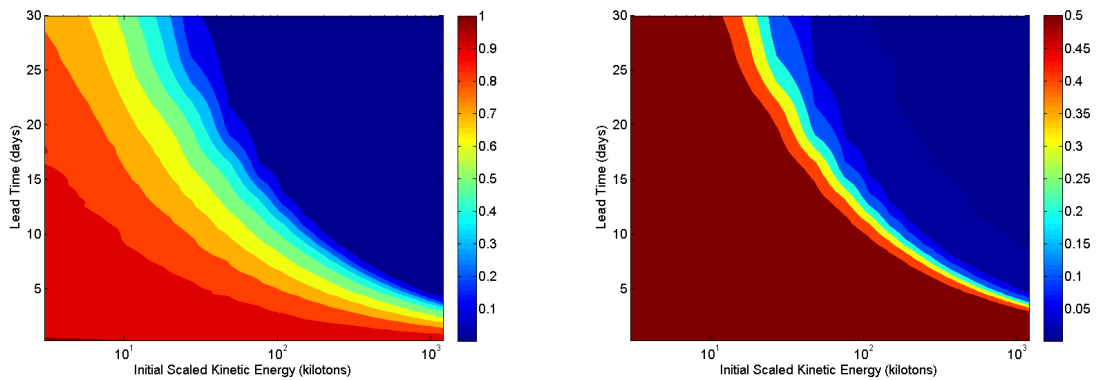


Figure 6.10 Impacting Mass Fraction Contours for Low Lead Time Cases.

A wide variety of timing and scaling parameters were investigated. Figure 6.10 shows the steep drop-off in impacting mass when both the available amount of time and the explosive power are increased for a range of up to 1.2 Mt and 30 days lead time. A different scale showing the contour for 0.5% of the initial mass is also given in Figure 6.10. The advantage to this approach over single parameter variation is that coupling between these two parameters can be observed. For example, If we take 1% of the initial impacting mass to be a measurement of success, then we can use the contour data to generate a relationship between lead time and the required explosive energy. A comparison of an analytical model and the data is shown in Figure 6.11. If L is the mission lead time and E is the required energy, this approximation is

given by:

$$E = \exp(-1.8401E-4L^3 + 1.4826E-2L^2 - 4.5491E-1L + 9.9829) \quad (6.1)$$

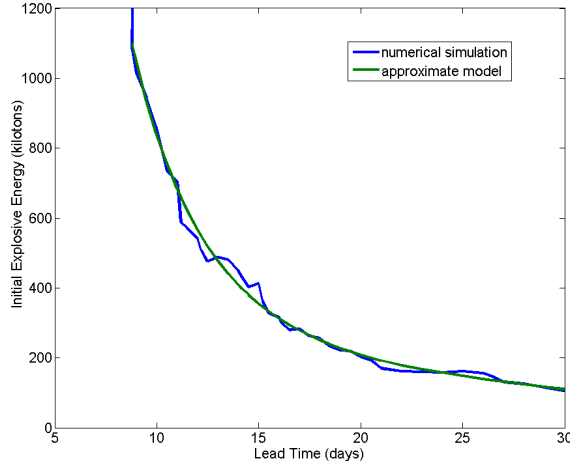


Figure 6.11 Analytical Approximation of Minimal Successful Mission Parameters.

The required explosive energy levels out below 200 kT as lead time increases. Therefore, even if a conventional explosive were to disrupt the target, it would not have enough initial kinetic energy to disperse the resulting fragments such that less than 1% of the initial mass remains on impacting trajectories.

6.3.2 Reentry Modeling

Applying a simple drag and ablation reentry modeling to the impacting fragments of previous radial deflection results, we find that over 85% of the mass on impacting trajectories is ablated from the fragments during the reentry process, and that only 10.7% of the mass listed in Table 6.1 reaches the surface. The results of these simulations can be seen in Table 6.4. A sample atmospheric stress distribution profile can be seen in Figure 6.12. The material yield stress is clearly exceeded at an altitude of 8.5 km, at which point the fragment is presumed to be catastrophically disrupted. Reentering fragments hit the atmosphere at relative speeds between 8 km/s and 15 km/s, depending on velocity orientation. A sample velocity profile in Figure 6.12 shows that velocity is reduced by drag and ablation to less than half its equivalent

Table 6.4 Reentry Modeling Results

Deflection Type	Bodies	Burnups	Impacts	Final Mass Ratio
+Radial (1 minute)	201	82	119	5.5259E-4
-Radial (1 minute)	117	34	83	6.2385E-4
+Transverse (1 minute)	6283	3211	3072	2.2025E-2
-Transverse (1 minute)	5967	2742	3225	3.0726E-2
+Normal (1 minute)	500	258	242	3.5593E-3
-Normal (1 minute)	523	278	245	4.0355E-3

ballistic entry velocity. A corresponding reduction in fragment mass results in substantially reduced impact energy.

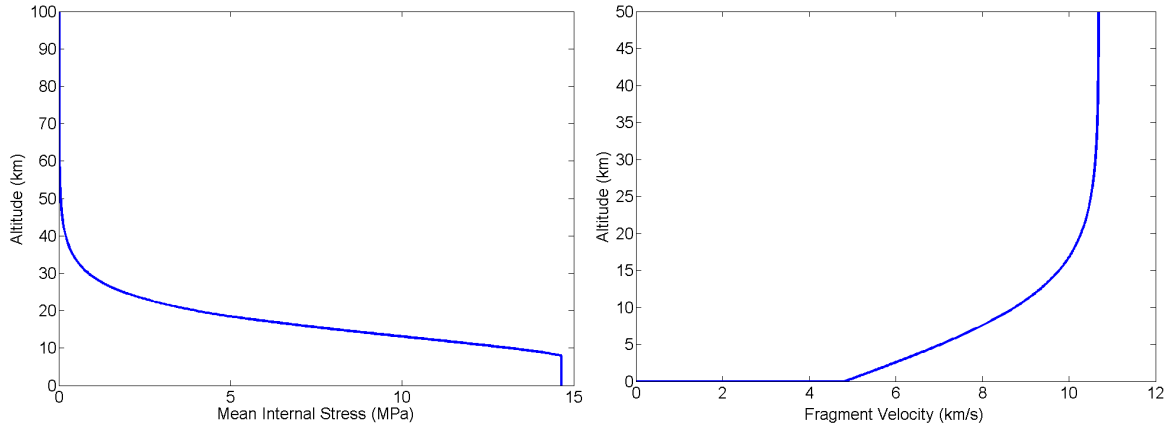


Figure 6.12 Reentry Stress Distribution and Velocity Profile with Altitude for Example Disrupted and Impacting Fragments.

6.4 Computational Optimization

A single computational node was used to determine optimal distribution of MPI and OpenMP processes across the current worker topology being considered. This system has 2 sockets populated with Intel Xeon X5650 six-core CPUs at 2.66 Ghz. Intel HyperThreading technology is enabled, resulting in 24 logical processors visible to the operating system. Additionally, the default level of OpenMP threading is 24. There are 4 NVIDIA Tesla C2050 GPU cards, each connected on a dedicated PCI-E x16 bus. System RAM is 32 GB, while each GPU has 3 GB GDDR5 for a total GPU work unit of 12 GB (11.2 GB with ECC enabled).

Fourteen multiprocessors on each card result in 448 shader cores each, limited to a maximum kernel launch of 1024 threads per thread block. This new “Fermi” GPU architecture has a theoretical peak performance of 515 Mflops in double precision, representing a game-changing leap forward in GPU double precision computing, as shown by real-world results.

While grid information does not need to be stored for this model, the drawback is that neighboring particles need to be determined at each time step. Since the integration scheme is a second order predictor-corrector scheme, particle information is needed at both steps. The first change made to the standard scheme was to retain the neighbor ID information for the corrector step. Only the kernel and kernel derivative values at the new neighbor predicted position need to be computed. This reduced time-to-solution by 30.2% compared to a two-stage neighbor finding algorithm. Results for both cases were compared, and while ending state values could be slightly different the distribution remained the same, and the method conserved energy slightly better through the end of the simulation. A possible beneficial side effect of this approach is the reduction of importance of neighbor changes in a prediction step, which might help damp out numerical instabilities and allow for larger time step changes. This is something to be tested in the future. Also, while brute force computation of neighbor particles was the original approach, a Sort-and-Sweep method reduced this time by 36% for the present target model. This method scales as $N \log N$ rather than N^2 [42].

Neighbor information arrays were stored in a column-major format by particle, allowing stride 1 access to the ID number, kernel value, and kernel derivative values for each neighbor of a particle. Additionally, loop unrolling and inlining for simple functions were implemented, and optimization flags were passed in the build step. For the GPU model, utilizing asynchronous kernel launches to continue computation without synchronization resulted in an 8% performance increase. The theoretical load on each process should be equal, since each has the same number of particles for which a state update needs to be computed. However, in areas of quickly changing density (for example the expanding shock wave), the number of average neighbors for a particle goes up dramatically. This is controlled in 2 ways to aid load balancing. First, the ID assignment scheme works outward in a radial manner, while making sure that mirroring particles on opposite sides of the primary axis are adjacent in memory. Second, the evolution

of h strives to keep the number of neighboring particles near the starting value, resulting in an equal computational burden. For the GPU model, a load factor was developed, dividing the minimum time to complete a section between synchronizations by the maximum time. Sampling this load factor allows one to better understand the efficiency of the code section. At a time of 1.2 ms, an example chosen because of the high energy of this point of the simulation, a vertical distribution of particle IDs resulted in a load factor efficiency of around 0.68. The present method has improved this portion to a median of 0.87.

6.4.1 Performance

Pure MPI scalability for up to 12 processes was tested on the present hardware, resulting in near linear scaling and a total parallel speedup of 8.9 for MPI. Including OpenMP in a Hybrid parallel scheme, a total parallel speedup of 11.9 is achieved, showing near perfect expected scalability across a single node as shown in Figure 6.13. Thus, each additional planned node might add almost 12x speedup for host computation, minus internode communication overhead. As shown in Figure 6.13, when the binding option is passed to the Hydra process manager to set 1-2 MPI processes per socket, and an OpenMP thread level of 6 is set, the best performing speedup for the system is obtained. This corresponds to a value of 11.2 for 12 computational threads and 11.9 for 24 computational threads. Performance improvement using ≥ 12 threads is predominantly dependent on the HyperThreading hardware implementation. This is shown to only have an improvement over 12 threads when the shared thread level is 4, 6, 8, or 12. However, good performance with 12 threads among these hybrid schemes was limited to an OpenMP level of 6 and 12. While the default OpenMP maximum thread level for this system is 24, benefits from this technology are implementation dependent, so the preferred setup for future system programming is 1 MPI process per socket with an OpenMP threading level of 6 unless improvement from additional MPI processes can be demonstrated.

GPU acceleration performance for this method is a substantial improvement over a larger CPU-only cluster. Since the threading structure of the GPU is limited to SIMD kernel launches of multiple threads on a multiprocessor, serial performance for comparison is measured on the host CPU. Figure 6.13 shows the relationship between the number of GPUs used in the state

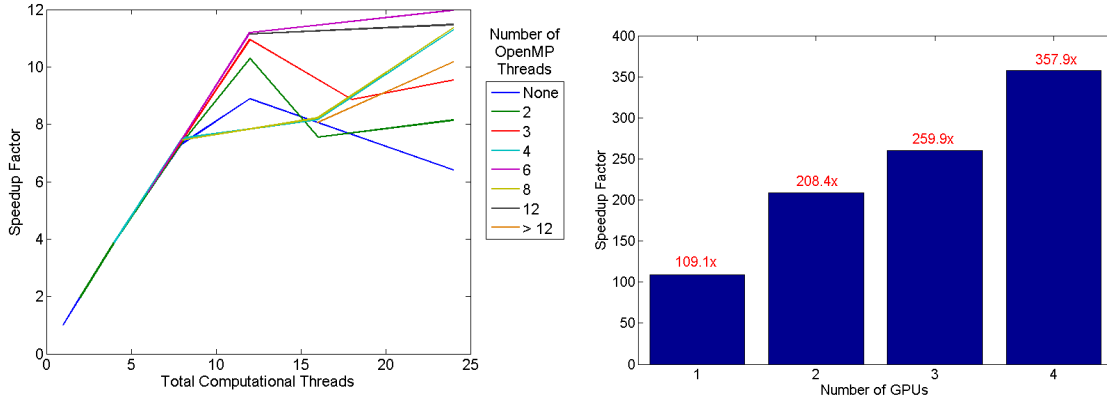


Figure 6.13 Comparison of Single-Node Performance on CPU and GPU.

update process and the parallel speedup. At least 1 MPI thread is needed per GPU. In fact, using the currently supported CUDA Fortran toolkit (version 4.0), binding between CPU thread and GPU control requires that additional threading use a shared memory approach such as OpenMP. In a previous test, GPU speedup for this architecture ranges from 50x to 120x for a 50 m diameter target problem. Since the GPU approach works well for data-parallel problems, one would expect that increasing the scale of the problem would yield better performance. In fact, using the current solid target standoff model (3.1M particles) maximum speedup on a single node is increased to 357.9x, as shown in Figure 6.13. Since the neighbor search problem is substantially increased, the parallel structure of the GPU is far preferred to the hybrid CPU programming model.

6.5 Optimal Mission Results

The present simulation package has the advantage of being able to handle millions of decoupled optimization problems in parallel to one another. Thus, the generation of data outpaces the capability for displaying it in the present work. However, sample results are shown for a nominal impacting trajectory with a lead time of 15 days. Figure 6.14 shows the cost function contours for approach asymptotes of a sample mission. This impacting trajectory has a semimajor axis of 0.968, an eccentricity of 0.0242, and an inclination of 7.309 degrees.

It is clear for this case that, not only do local optimal solutions exist, but that there are

specific conditions which should be avoided. However, this was not the case for all of the virtual impacting trajectories. This fact was especially true for orbits of high eccentricity (> 20 degrees), which had many local minimums, and a wider range of effective dispersion options. Deeper cost function wells existed for these cases, though the geometry was more complicated than the lower inclination case, as shown. The contours are colored according to the base 10 logarithm of the resulting impact probability, showing a range of orders of magnitude. No clear result for the optimal direction for all cases was established. In the sample case, the conditions to be avoided were a perturbation normal to the plane of the orbit. The optimal directions in this case are near parallel to the velocity direction. The vectors forming the solutions of the tested orbits were uniformly distributed, which may be indicative of the lower lead time mission.

As discussed in [23], some approach asymptotes are critical for interception with a single launch. Therefore, future work should address the coupled problem of mission feasibility and mission effectiveness. This will likely place stricter limits on the available lead times and the payload mass deliverable to the target.

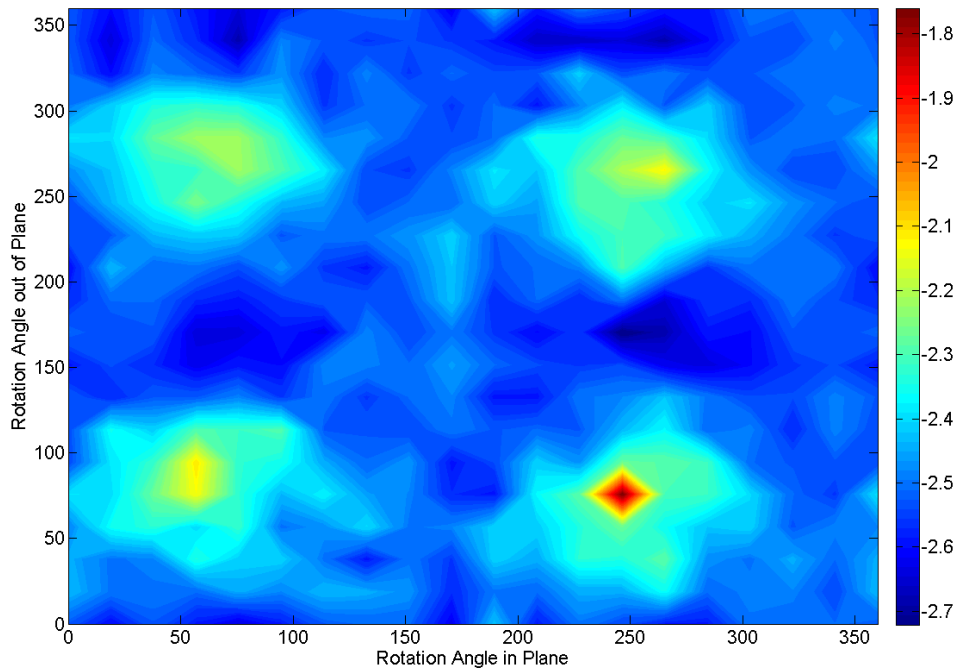


Figure 6.14 Cost Function Contours for Sample Mission Approach Asymptotes.

6.6 Characterization Results

An example of the mass and enclosing volume estimation with limited flash LIDAR observations is shown in Figure 6.15. Convergence of the mass, size parameters, and body frame orientation can be seen in Figure 6.16. The first order elliptical model gives us a scaled estimate that is accurate to approximately 10 %. This would be a good starting point for a more accurate local gravity model. It also gives us a collision avoidance ellipsoid, and well defines the body frame for subsequent maneuvers.

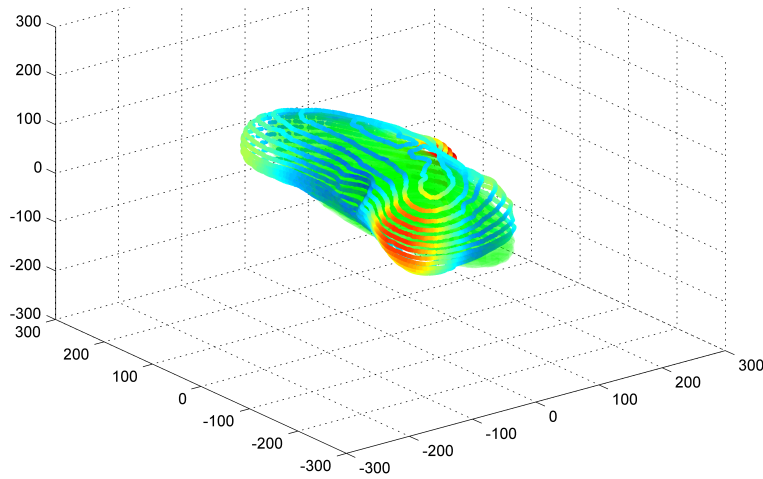


Figure 6.15 Surface Estimation using limited LIDAR Measurements.

6.7 Summary and Discussion

The present SPH hydrocode suggests that a dynamic model of a hypervelocity surface burst yields results similar in spatial and temporal distribution at Earth impact to a static subsurface explosion. This gives additional launch windows for mission design, limits the fuel needed for a rendezvous burn, and avoids the need to bury the explosive payload. Additionally, the dynamic model should better predict system behavior when addressing high velocity penetrator architectures. The primary mechanism for this improvement is to use impactor momentum to couple energy into the surface material. Since impact at over 6 km/s is not survivable by current explosive system technology, this naive approach is not a viable option. However, the proposed HAIV concept offers similar promise. This might give an option for realistically determining

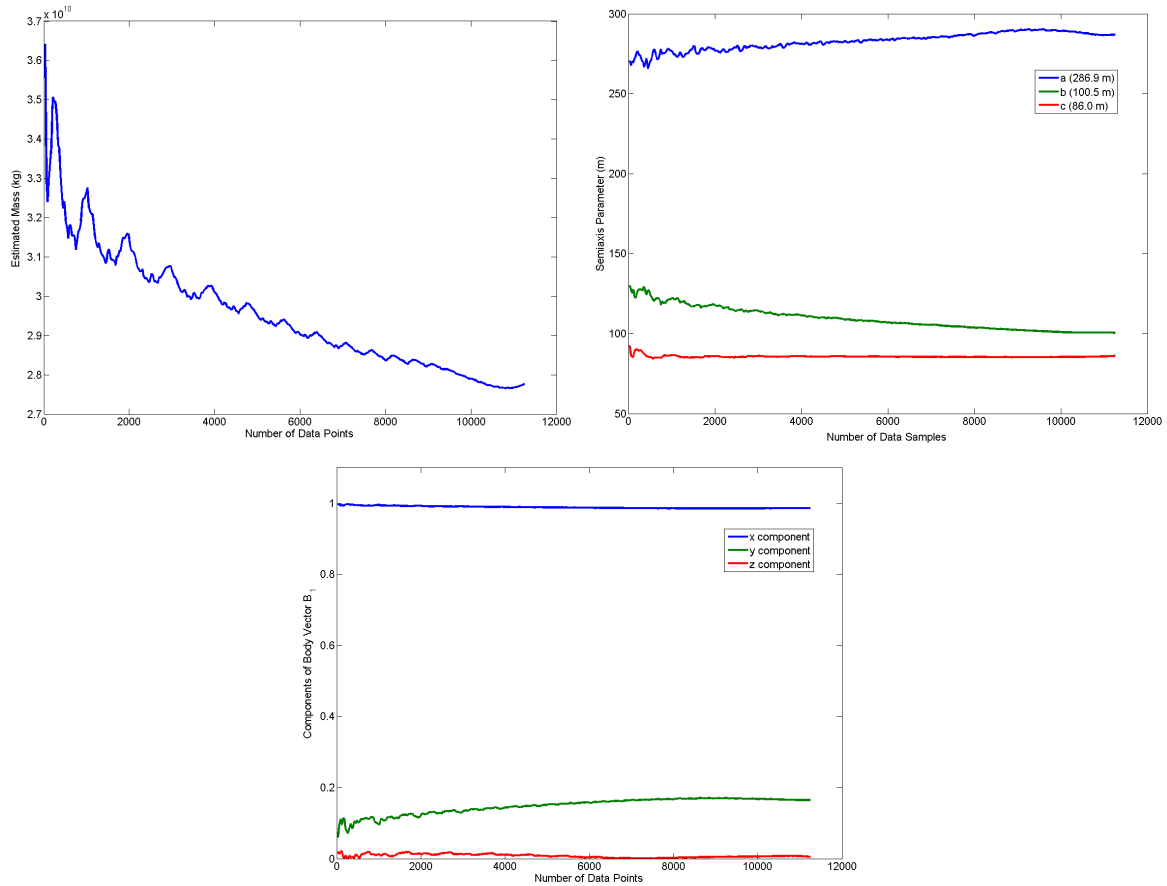


Figure 6.16 Convergence of Mass Estimate, Shape Parameters, and Orientation.

the limits of such a system for asteroid deflection missions. NEO orbital parameters such as semimajor axis and eccentricity were not found to be important for these time scale, but it was found that inclination was important in determining effectiveness of any given method.

All methods of disruption using a 100 kt nuclear energy source were quite effective for 100 m diameter targets for 15 days lead time, regardless of the orbit considered. The existing ADRC models of a hypervelocity impact fragmentation of an NEO were extended and applied to a 3D inhomogeneous asteroid model with randomly generated sections and generic material parameters. It is clear from the discrepancy in dispersion speed for the 76 m asymmetric target that the proposed HAIV concept is successful in reducing the mass remaining on impacting trajectories over a simple contact burst. Future work should consider larger bodies, a range of source energies, and lead times specific to the available mission time for a given orbit. At

this time we can say conclusively that we believe a 100 m target can be safely disrupted on all representative orbits with 15 days of dispersion time.

New HPC technology utilizing GPU acceleration has resulted in orders of magnitude improvement in computational ability. Speedup of the GPU accelerated model compared to serial execution for the both target models has been demonstrated. While the 330,000 particles of the penetrator target are limited mostly by communication bandwidth, the 3.1 million particles in the standoff model are limited by computational speed and memory bandwidth for the threads on the GPU. A substantial speedup improvement, from 53x to 358x, is observed. New high-throughput neighbor-finding methods were suggested, using the GPU acceleration technology of the current simulation toolkit. The current simulation set develops a tensor relationship for material characteristics and orientation. This allows for more realistic size and shape generation for NEO fragments by treating damage as a local quantity (cracks) rather than a distributed state variable. GPU acceleration of the 3D model is up to 200x on a single workstation, continuing a trend of increasing computational complexity while also increasing efficiency. This approach allows us to compute a range of values rather than monolithic single simulations, and is incredibly important for the orbital analysis.

This shows single node computational performance on the same order as a moderate cluster. The ability to run multiple cases to address statistical system behavior results in simulation being integrated into overall mission design. Mission effectiveness can be estimated in advance of a need for mission design, allowing new architectures and interchangeable components for a universal deflection plan. This technology provides a useful reduction in time-to-solution comparable to 30 similar CPU-only nodes (which would cost \$4,000 each) in a \$14,000 form factor, showing a 8.6x improvement in cost-adjusted performance. Since a large amount of data can be processed using GPU simulation, this work confirms that disruption at different times along a given orbit can have a large effect on the resulting shape of debris. This allows for a more clear set of objectives for mission design. Another new result is the availability of representative 3D fragment distributions. This will improve the trajectory of the desired hypervelocity intercept mission by allowing full degrees of freedom in choosing the approach asymptote.

BIBLIOGRAPHY

- [1] E.M. Standish, X.X. Newhall, J.G. Williams, and D.K. Yeomans. Orbital Ephemerides of the Sun, Moon, and Planets. In P.K. Seidelmann, editor, *Explanatory Supplement to the Astronomical Almanac*. University Science Books, 1992.
- [2] Committee to Review Near-Earth Object Surveys and Hazard Mitigation Strategies. *Defending Planet Earth: Near-Earth Object Surveys and Hazard Mitigation Strategies*. National Research Council, 2010.
- [3] Risk Management Solutions Inc. *Comet and Asteroid Risk: An Analysis of the 1908 Tunguska Event*. RMS Special Report, 2009.
- [4] D. Yeomans and P. Chodas. Additional Details on the Large Fireball Event over Russia on Feb. 15, 2013, March 2013. Available online at http://neo.jpl.nasa.gov/news/fireball_130301.html. Date accessed 3/14/2013.
- [5] M. Boslough. Airburst Warning and Response. In *2nd IAA Planetary Defense Conference*, number IAA-PDC-2166721, May 2011.
- [6] NASA Jet Propulsion Laboratory. Near Earth Object Program Discovery Statistics, March 2013. Available online at <http://neo.jpl.nasa.gov/stats>. Date accessed 3/14/2013.
- [7] R.B. Adams et al. Survey of Technologies Relevant to Defense from Near-Earth Objects. Technical Report NASA-TP-2004-213089, NASA Marshall Space Flight Center, July 2004.
- [8] M.J. Barrera. Conceptual Design of an Asteroid Interception for a Nuclear Deflection Mission. In *2004 Planetary Defense Conference: Protecting Earth from Asteroids*, number AIAA-2004-1481, February 2004.

- [9] B.W. Barbee and W.T. Fowler. Spacecraft Mission Design for the Optimal Impulsive Deflection of Hazardous Near-Earth Objects (NEOs) using Nuclear Explosive Technology. In *2007 Planetary Defense Conference*, March 2007.
- [10] V.A. Simonenko, V.N. Nogin, D.V. Petrov, O.N. Shubin, and J.C. Solem. Defending the Earth Against Impacts From Large Comets and Asteroids. In T. Gehrels, editor, *Hazards Due to Comets and Asteroids*, Tucson, AZ, 1994. University of Arizona Press.
- [11] W.J. Tedeschi. Systems-Level Considerations for Mitigating the NEO Impact Hazard. In *Planetary Defense Workshop, Lawrence Livermore National Laboratory*, May 1995.
- [12] J. Sanchez, M. Vasile, and G. Radice. On the Consequences of a Fragmentation Due to a NEO Mitigation Strategy. In *59th International Astronautical Congress*, number IAC-08-C1.3.10, September 2008.
- [13] B. Wie and D. Dearborn. Earth-Impact Modeling and Analysis of a Near-Earth Object Fragmented and Dispersed by Nuclear Subsurface Explosions. In *20th AAS/AIAA Space Flight Mechanics Meeting*, 2010.
- [14] K.A. Holsapple. About Deflecting Asteroids and Comets. In M.J.S. Belton, T.H. Morgan, N. Samarasinha, and D.K. Yeomans, editors, *Mitigation of Hazardous Comets and Asteroids*, Cambridge, U.K., 2004. Cambridge University Press.
- [15] D.B. Gennery. Deflections of Asteroids by Means of Standoff Nuclear Explosions. In *2004 Planetary Defense Conference: Protecting Earth from Asteroids*, number AIAA-2004-1439, February 2004.
- [16] A.R. Miles. Asteroid Deflection via Standoff Nuclear Explosions. In *Asteroid Deflection Research Symposium*, October 2008.
- [17] T.J. Ahrens and A.W. Harris. Deflection and Fragmentation of Near-Earth Asteroids. *Nature*, 360:429–433, December 1992.

- [18] C. Colombo, J. Cuartielles, M. Vasile, and G. Radice. A Comparative Assessment of Different Deviation Strategies for Dangerous NEO. In *57th International Astronautical Congress*, number IAC-06-A3.P.05, October 2006.
- [19] D. Dearborn. The Use of Nuclear Explosive Devices to Disrupt or Divert Asteroids. In *2007 Planetary Defense Conference*, March 2007.
- [20] J. Lewis. *Comet and Asteroid Impact Hazards on a Populated Earth*. Academic Press, 2000.
- [21] B. Wie. Solar Sailing Kinetic Energy Impactor Mission Design for Impacting and Deecting Near-Earth Asteroids. In *NASA Workshop on NEO Detection, Characterization, and Threat Mitigation*, June 2006.
- [22] B. Dachwald, R. Kahle, and B. Wie. Solar Sailing KEI Mission Design Tradeoffs for Impacting and Detecting Asteroid 99942 Apophis. In *AIAA/AAS Astrodynamics Specialist Conference*, number AIAA-2006-6178, August 2006.
- [23] S. Wagner and B. Wie. Analysis and Design of Fictive Post-2029 Apophis Intercept Mission for Nuclear Disruption. In *AIAA/AAS Astrodynamics Specialists Conference*, number AIAA-2010-8375, August 2010.
- [24] C.F. Chyba, P.J. Thomas, and K.J. Zahnle. The 1908 Tunguska Explosion: atmospheric disruption of a stony asteroid. *Nature*, 361:40–44, 1993.
- [25] S. Abe et al. Mass and Local Topography Measurements of Itokawa. *Science*, 312:1344–1347, 2006.
- [26] F. Winterberg. *The Physical Principles of Thermonuclear Explosive Devices*. Fusion Energy Foundation, New York, 1981.
- [27] H.R. Hulme. *Nuclear Fusion*. Springer-Verlag, London, 1969.
- [28] J.L. Remo. Energy Requirements and Payload Masses for Near-Earth Objects Hazard Mitigation. *Acta Astronautica*, 47(1):35–50, 2000.

- [29] R. Eisberg and R. Resnick. *Quantum Physics of Atoms, Molecules, Solids, Nuclei, and Particles*. Wiley and Sons, Louisville, KY, 1985.
- [30] J.L. Margot, M.C. Nolan, L.A.M. Benner, S.J. Ostro, R.F. Jurgens, J.D. Giorgini, M.A. Slade, and D.B. Campbell. Binary Asteroids in the Near-Earth Object Population. *Science*, 296, 2002.
- [31] P. Michel, W. Benz, and D.C. Richardson. Disruption of Fragmented Parent Bodies as the Origin of Asteroid Families. *Nature*, 421:608–611, February 2003.
- [32] J.J. Monaghan. Smoothed Particle Hydrodynamics. *Reports on Progress in Physics*, 68:1703 – 1759, July 2005.
- [33] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing, Singapore, 2003.
- [34] W. Benz and E. Asphaug. Simulations of Brittle Solids using Smooth Particle Hydrodynamics. *Computer Physics Communications*, 87:253–265, 1995.
- [35] S. Hiermaier, D. Konke, A.J. Stilp, and K. Thoma. Computational Simulation of the Hypervelocity Impact of Al-Spheres on Thin Plates of Different Materials. *International Journal of Impact Engineering*, 20:363–374, 1997.
- [36] M. Jutzi, W. Benz, and P. Michel. Numerical Simulations of Impacts Involving Porous Bodies I. Implementing Sub-Resolution Porosity in a 3D SPH Hydrocode. *Icarus*, 198:242–255, 2008.
- [37] P.W. Randles and L.D. Libersky. Smoothed Particle Hydrodynamics: Some Recent Improvements and Applications. *Computer Methods in Applied Mechanics and Engineering*, 139:375–408, 1996.
- [38] J.H. Tillotson. Metallic Equations of State for Hypervelocity Impact. Technical Report GA-3216, General Atomic, 1962.
- [39] S.H. Schuster and J. Isenberg. Equations of State for Geologic Materials. Technical Report DNA-2925Z, Defense Nuclear Agency, 1972.

- [40] J.M. Owen, J.V. Villumsen, P.R. Shapiro, and H. Martel. Adaptive Smoothed Particle Hydrodynamics: Methodology II. *The Astrophysical Journal Supplement Series*, 116:155209, June 1998.
- [41] D.S. Balsara. von Neumann stability analysis of smoothed particle hydrodynamics- suggestions for optimal algorithms. *Journal of Computational Physics*, 121:357–372, 1995.
- [42] S. Le Grand. Broad-Phase Collision Detection with CUDA. In Hubert Nguyen, editor, *GPU Gems 3*. Addison-Wesley Professional, August 2007.
- [43] C.S. Plesko, R.P. Weaver, and W.F. Huebner. Energy Deposition in Hazard Mitigation by Nuclear Burst: Sensitivity to Energy Source Characteristics, Geometry, and Target Composition. In *42nd Lunar and Planetary Science Conference*, March 2011.
- [44] C. Gritzner and Kahle R. Mitigation Technologies and Their Requirements. In M.J.S. Belton, T.H. Morgan, N. Samarasinha, and D.K. Yeomans, editors, *Mitigation of Hazardous Comets and Asteroids*, Cambridge, U.K., 2004. Cambridge University Press.
- [45] B.P. Shafer et al. The Coupling of Energy to Asteroids and Comets. In T. Gehrels, editor, *Hazards Due to Comets and Asteroids*, Tucson, AZ, 1994. University of Arizona Press.
- [46] W.J. Tedeschi. Mitigation of the NEO Impact Hazard Using Kinetic Energy. In *Planetary Defense Workshop, Lawrence Livermore National Laboratory*, May 1995.
- [47] S. Yamamoto, T. Kadono, S. Sugita, and T. Matsui. Velocity Distributions of High-Velocity Ejecta from Regolith Targets. *Icarus*, 178(1):264–273, 2005.
- [48] P.H. Schultz et al. The Deep Impact Oblique Impact Cratering Experiment. *Icarus*, 190(2):295–333, 2007.
- [49] T. Michikami, K. Moriguchi, S. Hasegawa, and A. Fujiwara. Ejecta Velocity Distribution for Impact Cratering Experiments on Porous and Low Strength Targets. *Planetary and Space Science*, 55:70–88, 2007.
- [50] K. Wada, H. Senshu, and T. Matsui. Numerical Simulation of Impact Cratering on Granular Material. *Icarus*, 180(2):528–545, 2006.

- [51] V.V. Balashov. *Interaction of Particles and Radiation with Matter*. Springer-Verlag, Berlin, 1993.
- [52] H. D. Curtis. *Orbital Mechanics for Engineering Students*. Elsevier, Oxford, UK, 2006.
- [53] D. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, 3 edition, 2007.
- [54] H. Schaub and J. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston, VA, 2003.
- [55] J.D. Faires and R. Burden. *Numerical Methods*. Brooks/Cole Thomson, Pacific Grove, CA, 3 edition, 2003.
- [56] D.C. Lay. *Linear Algebra and Its Applications*. Pearson, New York, 3 edition, 2006.
- [57] J.D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, New York, 4 edition, 2007.
- [58] B. Wie. Dynamics and Control of Gravity Tractor Spacecraft for Asteroid Deflection. *Journal of Guidance, Control, and Dynamics*, 31(5), 2008.
- [59] V. Chobotov, editor. *Orbital Mechanics*. AIAA Education Series, Reston, VA, 3 edition, 2002.
- [60] B. Wie. Astrodynamical Fundamentals for Deflecting Near-Earth Objects. In *60th International Astronautical Congress*, number IAC-09-C1.3.1, October 2009.
- [61] D.G. Korycansky and C.S. Plesko. Reaggregation Times of Potentially Hazardous Object Fragments After a Hazard Mitigation Impulse. In *41st Lunar and Planetary Science Conference*, March 2010.
- [62] S.G. Love and T.J. Ahrens. Catastrophic Impacts on Gravity Dominated Asteroids. *Icarus*, 124:141–155, 1996.
- [63] E. Asphaug, S.J. Ostro, R.S. Hudson, D.J. Scheeres, , and W. Benz. Disruption of Kilo-metre Sized Asteroids by Energetic Collisions. *Nature*, 393:437–440, 1998.

- [64] P.C. Thomas et al. Eros: Shape, Topography and Slope Processes. *Icarus*, 155:18–37, 2002.
- [65] H. Gouraud. Continuous Shading of Curved Surfaces. *IEEE Transactions on Computers*, 20(6):623629, 1971.
- [66] J. Nocedal and S.J. Wright. Least Squares Problems. In *Numerical Optimization*. Springer Series in Operations Research, 2006.
- [67] J. Hennessy and D. Patterson. *Computer Organization and Design*. Morgan Kaufmann Publishers, San Francisco, 1998.
- [68] M.P.I. Forum. *MPI-2: Extensions to the Message-Passing Interface*. 2003.
- [69] O.A.R. Board. *OpenMP Application Program Interface*, volume 3. 2008.
- [70] P. S. Pacheco and W. C. Ming. *MPI Users Guide in FORTRAN*. 1997.
- [71] D.B. Kirk and W.W. Hwu. *Programming Massively Parallel Processors: A Hands-On Approach*. Morgan Kaufmann, Burlington, MA, 2010.
- [72] NVIDIA Corporation. *NVIDIA CUDA C Programming Guide v3.1*. May 2010.
- [73] NVIDIA Corporation. *OpenCL Programming Guide for the CUDA Architecture v3.1*. May 2010.
- [74] The Portland Group. *CUDA Fortran Programming Guide and Reference*. 2010.