

# Disentangling the Shadows of a Planetary Collision

MSc Thesis

Catherine Slaughter



Universiteit  
Leiden  
Sterrewacht

# Disentangling the Shadows of a Planetary Collision

A thesis submitted in partial fulfillment  
of the requirements for the degree of

Master of Science  
in  
Astronomy

Author:  
Student ID:  
Supervisor:  
Second corrector:  
Project duration:

Catherine Slaughter  
s3238865  
Prof. Matthew Kenworthy  
Prof. Michiel Hogerheijde  
September 2022 – June 2023

Cover image:  
Template style:  
Template licence:

*Star Wars Episode IV: A New Hope*  
Thesis style by Richelle F. van Capelleveen  
Licenced under CC BY-NC-SA 4.0



Universiteit  
Leiden  
Sterrewacht

P.O. Box 9513, 2300 RA Leiden, The Netherlands

# Abstract

Planetesimal collisions are significant to our understanding of planetary system formation histories. For example, in our own solar system, it is believed the formation of the Earth-Moon system relied on such an event. These collisions are believed to occur around 100-300 Myr in a stellar system's lifetime. Despite their importance, planetesimal collisions are extremely difficult to observe, and not very well understood. Observations of young sun-like star ASASSN-21qj back in December 2021 discovered a transit-like dimming event which is likely the result of a cloud of debris occulting the star. The dimming event—combined with related infrared observations—is an indication of a very recent rocky planetesimal collision. Similar anticipated observations to be made with the forthcoming Vera C. Rubin telescope (LSST) justify the need for tools to model and analyze such events. By assuming Keplerian motion from the point of collision, we create a simple model to simulate the ASASSN-21qj system, and calculate the resulting observed light curve. We fit the light curve model to the transit observations (with reduced chi-squared of 3.42), and determine an expected mass-velocity profile for the creation of the debris cloud. This profile is fitted to a power-law distribution, with best fit model having exponent  $-0.907 \pm 0.106$  and coefficient  $0.0095 \pm 0.0028$ , with an R-squared fit statistic of 0.568. Future research on the ASASSN-21qj system will be well supplemented by this code, as will forthcoming studies of similar system dynamics.

# Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor and mentor for this project, Professor Matthew Kenworthy. Whose guidance, knowledge, and seemingly boundless ingenuity with *NumPy* arrays has helped me become a better scientist and programmer.

A huge thank you is owed to my entire family, especially my parents, Sue-Sue and Steveo, who are finally realizing the benefits of answering far too many questions from a far-too-curious kid. Thank you also to my Uncle Matt, the first person to show me the stars through a telescope, and Aunt Lindsey. They are always there to help me up when I fall.

I would also like to thank my softball teammates at HSV Adegeest, for welcoming me with open arms, and for always providing the perfect way to work off some stress when my code had seemingly un-squash-able bugs.

Finally, special recognition goes to my partner, Thomas Culver Paul. The man who helps me hunt supernovae in the night sky, and whose support and love gave me the courage to come to Leiden in the first place. You hung the moon and the stars.

The code for this project was written in the *Python* programming language<sup>1</sup>. It draws heavily from the *NumPy* (Oliphant, 2006), *SciPy* (Virtanen et al., 2020), *Matplotlib* (Hunter, 2007), and *AstroPy* (Astropy Collaboration et al., 2013, 2018) libraries.

---

<sup>1</sup>Python Software Foundation, <https://www.python.org/>

# Contents

<b>Abstract .....</b>	<b>iii</b>
<b>Acknowledgements .....</b>	<b>iv</b>
<b>List of Tables .....</b>	<b>vii</b>
<b>List of Figures .....</b>	<b>viii</b>
<b>Nomenclature .....</b>	<b>x</b>
<b>1 Introduction .....</b>	<b>1</b>
1.1 Planet Formation and Collision Events	1
1.2 The ASASSN-21qj system	2
<b>2 Methods .....</b>	<b>4</b>
2.1 Assumptions and Computational Constraints	4
2.2 Explosion and Orbital Modeling	5
2.2.1 Kepler Orbital Elements .....	5
2.2.2 Initial Kick Velocity Vectors .....	7
2.2.3 Simulation with Keplerian Dynamics .....	8
2.3 Light Curve Modeling	11
2.4 Final Light Curve Fitting	12
2.4.1 General Fitting Routine .....	12
2.4.2 Binning the ASASSN-21qj Data .....	13
2.4.3 Transit Alignment .....	13
2.4.4 Testing Model Fit .....	14
2.4.5 Mass-Velocity Profile .....	15
2.5 Workflow	15
2.5.1 Particle Filtering .....	15
2.5.2 Some Notes on Computation Time and Storage Requirements .....	16

<b>3</b>	<b>Results</b>	<b>18</b>
3.1	Model Alignment	18
3.2	Final Model Fit	19
3.2.1	Comparison to Simulated Data	20
3.3	Mass-Velocity Profile	21
<b>4</b>	<b>Discussion</b>	<b>23</b>
4.1	Model Fit	23
4.2	Mass-Velocity Profile	23
4.3	Assumptions and Computational Impacts	24
4.3.1	Maximum Kick Velocity	24
4.3.2	Spherical Symmetry	24
4.3.3	Zero Inclination	25
<b>5</b>	<b>Conclusion</b>	<b>26</b>
5.1	Future work	27
	<b>Bibliography</b>	<b>28</b>

# List of Tables

<b>2.1</b>	The parameter inputs for the initial run of our final simulation. . . . .	<b>11</b>
<b>2.2</b>	The time breakdown of various steps for our final simulation and model. Note that times <i>will</i> vary from run to run. . . . .	<b>17</b>

# List of Figures

1.1	The photometric light curves of ASASSN-21qj in the optical (top) and infrared (middle). The optical data comes from the ASASSN survey, while the infrared comes from WISE. It is clear that a large spike in infrared radiation preceded the optical dimming event. The bottom panel shows an estimated color temperature from the different WISE bands. Figure taken from Kenworthy et al. (Submitted for Publication). . . . .	2
2.1	A diagram showing the major elements of an elliptical orbit. The foci are labeled $F_1$ and $F_2$ . The semi-major axis (a) is shown in red, the semi-minor axis (b) in blue, and the center-focus distance (c) in green. . . . .	6
2.2	A diagram description of the standard orbital elements, as used in this project. While the diagram uses the variable $v$ for the true anomaly, we use $f$ . From <a href="#">Wikimedia Commons</a> by Lucas Snyder, 10 October 2007. Public Domain. . . . .	6
2.3	Example projections of distributions of points for a Fibonacci Sphere Lattice (left), a Latitude-Longitude Lattice (center), and for randomly selected theta and phi (right). The Fibonacci lattice is visually the most uniform, while the Latitude-Longitude lattice shows higher densities at the poles, and the random selection has clumping in some regions. . . . .	8
2.4	Stills from an animation of the orbital simulation code. The black circle in the center represents ASASSN-21qj (not to scale), and the gray line represents the progenitor's orbit. Points are colored according to their initial velocity shell. A much smaller example is being shown than was used in obtaining our final results, for purpose of illustration. As the simulation progresses, the cloud of points gets sheared. . . . .	9
2.5	The geometric depiction of how a transit is determined. The conic section described is outlined with black dotted lines. The coordinate system defined for the simulation is overlaid in gray. Not to scale. . . . .	12

3.1	The best-fit model residuals vs. model-data offset for the horizontal alignment procedure. The minimum residual, corresponding to the location of our final model fit, is marked with a red star. ....	18
3.2	A comparison between a model with a vertical shift component (left) and one without (right). The general form of the two is very similar. On the plot with the vertical shift, the black dashed line indicates the coefficient value for the vertical shift component, as found by the NNLS. ....	19
3.3	A number of example light curves from our final simulation, from shells at varying kick velocities (lowest at the top, highest at the bottom). Each curve is individually normalized to one for fitting purposes. In general, the shells that are kicked hardest experience the most shear, leading to longer transits. ....	19
3.4	The final best-fit model to the ASASSN-21qj data. The full model is shown in orange, while the subset of the model used for aligning (see subsection 2.4.3) is highlighted with red. Error bars are shown on just the residual plot, for clarity. ....	20
3.5	The simulated data and related best-fit model, for comparison purposes. ....	21
3.6	The Mass-Velocity profile for the best-fit model, as well as the corresponding power law fit on linear (left) and logarithmic (right) axes. Residual plots are shown in the lower frames. ....	21

# Nomenclature

## Simulation Parameters

Equation Notation	Code Notation	Description
$e$	e	Eccentricity
$a$	a	Semi-major axis
$I$	inc	Inclination
$\omega$	omega	Argument of periapsis
$\Omega$	anode	Longitude of ascending node
$f$	f	True anomaly
$\Delta v$	delta_v	Kick velocity of current particle
$\phi_i$	phi	Phi component of kick
$\theta_i$	theta	Theta component of kick
$\Delta v_{max}$	kick_vel	Maximum kick velocity
$v_{orb}$	v_i	Progenitor velocity at explosion
$M$	M	Mass of primary
$m$	m	Mass of secondary (progenitor)
$n$	num_per_shell	Number of particles per shell
-	t_firstpass	Time of transit start
-	t_start	Simulation start
-	delta_t	Simulation length
-	n_steps	Number of time steps
-	n_shells	Number of shells
-	R_star	Primary radius

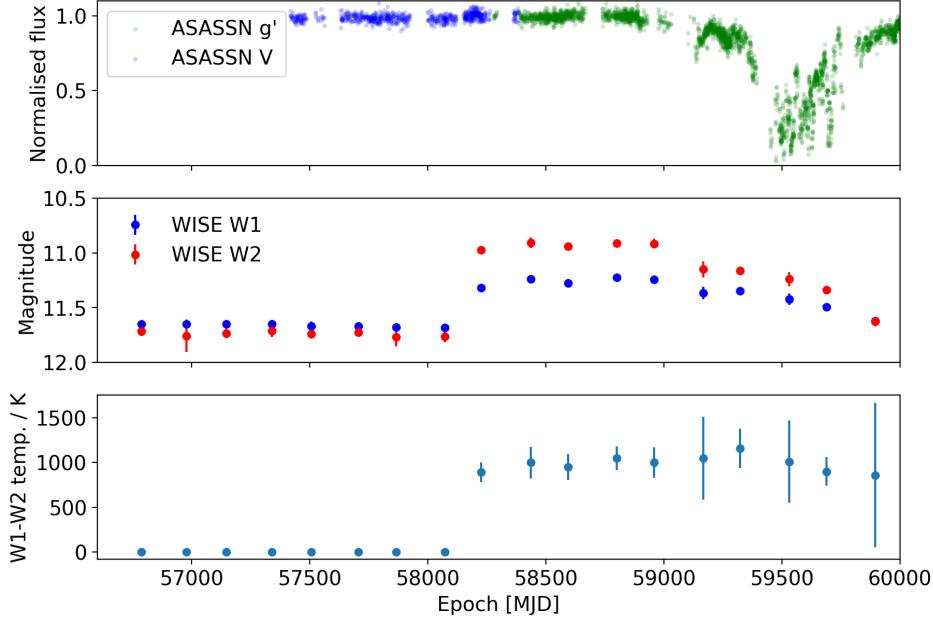
# 1. Introduction

## 1.1 Planet Formation and Collision Events

All planetary systems start out as large clouds of interstellar gas and dust (Hoyle, 1960). Over time, various processes (such as nearby supernovae) can cause these clouds to become unstable, eventually collapsing into new stars. Due to conservation of angular momentum, additional material flattens out into a circumstellar disk (Terebey et al., 1984). Given proper conditions, such a disk will eventually form individual planets (Cameron, 1962). In general, these circumstellar disks last anywhere from  $10^6 - 10^7$  years (Strom et al., 1989).

The processes occurring in the circumstellar disk between its creation and the end of the planet formation period are complicated and interdependent. In general, several significant processes occur semi-simultaneously. For one, lightweight gas is quickly removed from the disk via accretion onto growing planetesimals, photoevaporation, or stellar wind blowing (Williams & Cieza, 2011; Wyatt et al., 2015). Before its removal, this gas creates a dampening effect on movement within the disk. The loss of this dampening paves the way for collisional cascades to occur. These cascades create dust debris small enough that it, too, is pushed to the outer parts of the disk via radiation pressure (Wyatt, 2008). This process is observable due to the central star heating the dust debris, which radiates in the mid-infrared. Near the end of the planet-forming period, the remaining large planetesimals may become unstable due to gravitational effects. These can lead to planetesimals being thrown out of the system entirely, or slammed into one another. These collisions create bright outbursts in the infrared, as they form large debris clouds that are heated by the central star.

Extensive evidence for such collisions can be found within our own solar system. For many years, a "giant impact" has even been the prevailing theory for the formation of the Moon, given the unusually similar masses in the Earth-Moon system (Canup, 2004; Halliday & Canup, 2023). Additionally, this theory has given insight into the Earth's geologic history, including its enrichment and volatile depletion (Solomatova & Caracas, 2023) well beyond the time that the standard geologic record can cover. Elsewhere in our solar system, planetesimal collisions have been an explanatory mechanism for heating of icy planets, as well as the odd rotational direction of Uranus (Rufu & Canup, 2022). Additionally, similar to the Earth system, some theorize that a planetesimal collision has impacted



**Figure 1.1:** The photometric light curves of ASASSN-21qj in the optical (top) and infrared (middle). The optical data comes from the ASASSN survey, while the infrared comes from WISE. It is clear that a large spike in infrared radiation preceded the optical dimming event. The bottom panel shows an estimated color temperature from the different WISE bands. Figure taken from Kenworthy et al. (Submitted for Publication).

the formation of the Pluto-Charon System (Canup, 2005), though this remains a topic of debate.

Previous work has been done to model such collisions in the Earth-Moon system context (Canup, 2004). Additionally, more recently, Jackson et al. (2014) placed such systems in the context of Keplerian orbital mechanics. This paper lays out the mathematics for modeling debris cloud creation as if ejected isotropically from a point source. They additionally study the long-term observability of the resultant debris cloud, analyzing asymmetries in the resulting orbits and varying observational wavelengths.

## 1.2 The ASASSN-21qj system

ASASSN-21qj is a young, sun-like star. Despite being in the 2MASS catalog (Cutri et al., 2003), it was not of particular interest until December 2021, when it was observed by the ASASSN survey (Shappee et al., 2014; Kochanek et al., 2017) to be going through a deep dimming episode (Rizzo Smith et al., 2021).

Kenworthy et al. (Submitted for Publication) shows that this dimming episode was predicated by a significant increase in infrared emission from the system by about  $\sim 900$  days, as observed from the WISE mission (Wright et al., 2010). Figure 1.1 shows the photometric light curves from both these events. Kenworthy et al. (Submitted for Publication) outline several explanations for these events, including the possibilities that they may be unrelated, or that they may be a result

of an irregular debris disk. However, they ultimately determine that the most likely explanation is a sudden collision between two Super-Earth sized exoplanets, creating a cloud of dust debris. The dust, heated by the star and the energy of the collision, will have emitted in the infrared. About 900 days later, the debris would have begun transiting ASASSN-21qj , the cloud stretched and sheared by its own differential orbital motion.

Despite the plentiful science indicating that such events are significant to planetary system formation histories, they have thus far proven difficult to study. With the forthcoming first light on the Vera C. Rubin Telescope (LSST), we hope that many more such systems may be observed in the near future. Such observations would yield important insight into planetary formation histories. As such, it is useful to develop the tools to better analyze these systems now, in order to apply them quickly in the future. To this end, we make use of the mathematics outlined in Jackson et al. (2014) to simulate the ASASSN-21qj collision, modeling the resulting light curve, and fitting said light curve to the ASASSN data.

# 2. Methods

In general, the methods used for this work can be broken down into three main code families: the code(s) used to create the explosion and model the resulting orbits, the code(s) used to generate resulting light curves, and the code(s) used to fit the light curve models to the real ASASSN-21qj data. These code families are outlined in the following sections, followed by a general description of the workflow for generating results.

Given that our goal is to fit a modeled light curve to the real light curve data, we ultimately only need to recreate the resulting dust cloud from the collision, not the collision itself. As such, our simulation begins by "exploding" a simulated secondary (nicknamed "Alderaan", see title page).

## 2.1 Assumptions and Computational Constraints

As is often the case with simulation-based research (especially in the early stages) the model we present here comes with a number of simplifying assumptions and computational constraints. While most of these assumptions are either reasonable or have presumably negligible impact, those few that likely to be relevant for future research are addressed further in section 4.3.

### Cloud Assumptions

We begin by assuming the material in the debris cloud is kicked from the explosion isotropically. We model the cloud as a series of concentric spherical shells (see subsection 2.2.2), where each shell has been given a different kick velocity magnitude. Such an assumption serves only as a starting point for our initial modeling, and is not particularly physically founded. There is not, however a single preferable geometry to enforce at this point in time, as collisions can be exceptionally varied, so we remain with the symmetric case.

We also assume the dust experiences no radiation pressure from the star light. This comes with it an implied assumption that the dust grains in the debris cloud are larger than the wavelengths observed. This is fairly reasonable, as a single collision would be expected to primarily create larger debris.

As in Jackson et al. (2014), we also (more reasonably) assume that there is negligible intra-cloud gravitational interaction between dust grains. Instead, the gravitational potential from the central star is dominant. Along with being

physically well founded—given the relative masses of the star and the debris cloud—this assumption is also computationally convenient. Traditionally, gravitational N-body simulations with simple implementation require calculating the total force vector on every particle at every time step, an algorithm with  $O(N^2)$  time complexity (due to the pair-wise, particle-particle calculation of the force). By deciding the intra-cloud forces are negligible, however, the only force relevant on each particle is that from the star, reducing our complexity to  $O(N)$ . This, additionally, leaves us with a simple 2-body calculation (for each particle), the dynamics of which have been solved analytically via the Kepler equations. Using this analytical solution allows us to calculate the position and velocity of each particle at every time step all at once, using *NumPy* array functionality, instead of explicitly iterating over every individual time step, further speeding up our code.

### Progenitor Orbit

In addition to the assumptions made about the nature of the debris cloud, we make two significant assumptions about the orbit of the progenitor planetesimal (see subsection 2.2.1). We assume that the orbit is circular (zero eccentricity) and is at zero inclination with the Earth’s line of sight. The latter is believed, out of all the assumptions listed, to have the greatest potential impact on the final model fit. The implications of this and ideas for future improvements are outlined in section 4.3 and section 5.1.

It should be noted that these two assumptions create a degeneracy in the simultaneous values of the true anomaly, argument of periapsis, and longitude of the ascending node, which are now constrained only by the time of transit start (see Figure 2.2). As such, we set the latter two equal to zero.

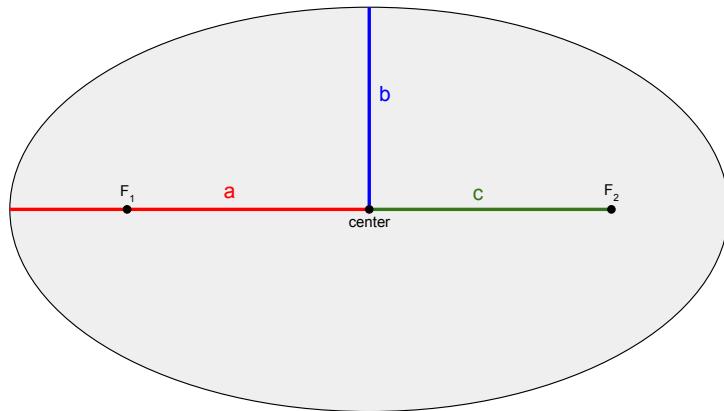
### Maximum Kick Velocity

While less of an initial assumption and more a computational constraint, the mathematics of Jackson et al. (2014) limit the maximum kick velocity we can give to the cloud. This is because the paper looks at long-term system evolution, and (rightfully) does not account for debris that may reach escape velocity after being kicked, as it quickly becomes irrelevant. For our purposes, however, we are only focused on the first pass of the system. If the real initial collision is powerful enough, it is possible that some of the ASASSN light curve is caused by material on an open orbit. This possible material is ignored in our work.

## 2.2 Explosion and Orbital Modeling

### 2.2.1 Kepler Orbital Elements

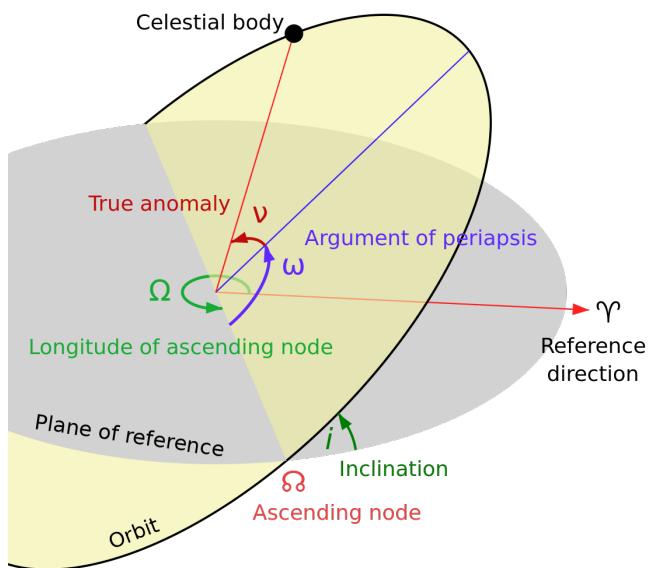
Two of the six kepler orbital elements define the size and shape of the orbit, and are shown in Figure 2.1. The semi-major axis ( $a$ ), represents the longest possible distance from the center of the ellipse to its edge. The eccentricity ( $e$ ) describes how circular an orbit is, and is defined as  $e = c/a$ , where  $a$ , again, is the semi-major axis, and  $c$  is distance between the ellipse’s center and foci. A perfect circle has an eccentricity of 0, as the center-focus distance is 0. For



**Figure 2.1:** A diagram showing the major elements of an elliptical orbit. The foci are labeled  $F_1$  and  $F_2$ . The semi-major axis ( $a$ ) is shown in red, the semi-minor axis ( $b$ ) in blue, and the center-focus distance ( $c$ ) in green.

a closed orbit, the eccentricity ranges on the semi-closed interval from  $[0,1]$ , 1 represents a parabolic orbit, and  $>1$  a hyperbolic orbit.

The four other angular Kepler orbital elements taken in by the routine from Jackson et al. (2014) (and therefore used by our code) are visualized in Figure 2.2. We define the orbital plane (yellow in the image) and the plane of reference, seen edge-on from our line of sight (gray in the image). In this case, "up" is defined as the direction of the angular velocity vector of the orbiting body, projected onto the axis perpendicular to the reference plane.



**Figure 2.2:** A diagram description of the standard orbital elements, as used in this project. While the diagram uses the variable  $v$  for the true anomaly, we use  $f$ . From Wikimedia Commons by Lucas Snyder, 10 October 2007. Public Domain.

The inclination ( $I$  in our work,  $i$  in the image) is the angular separation between

the orbital and reference planes. The ascending node refers to the location the celestial body crosses the plane of reference when ascending (the descending node sitting on the opposite side of the line of intersection). The longitude of the ascending node ( $\Omega$ ) is the angular distance between the reference direction (our line of sight) and the ascending node. The argument of periapsis ( $\omega$ ) represents the angle along the orbital plane between the intersection line (on the ascending node side) and the projection of the reference direction onto the orbital plane. The true anomaly ( $f$  in our work,  $v$  in the image) is the angular distance along the orbital plane between the same projection of the reference direction and the line connecting the celestial body to the center of the orbit. The true anomaly changes with time.

### 2.2.2 Initial Kick Velocity Vectors

Each spherical shell is created as a set of particles with varying initial kick directions on the phi-theta sphere. The distribution of these values is defined by the Fibonacci sphere lattice (González, 2010). Normally, the only way to truly evenly distribute points on a sphere is to use the vertices of a circumscribed regular solid. This is computationally expensive and significantly limits the number of possible particle counts. In contrast, the Fibonacci lattice makes use of the interesting geometry associated with the golden ratio in order to *mostly* evenly space points across the surface of a sphere. This is preferable to a latitude-longitude lattice, which exhibits higher densities at the poles, or a randomly-selected distribution, which is prone to clumping (see Figure 2.3).

Based on González (2010), the algorithm to distribute particles within the lattice are as follows. Given the golden ratio

$$R_{golden} = \frac{1 + \sqrt{5}}{2},$$

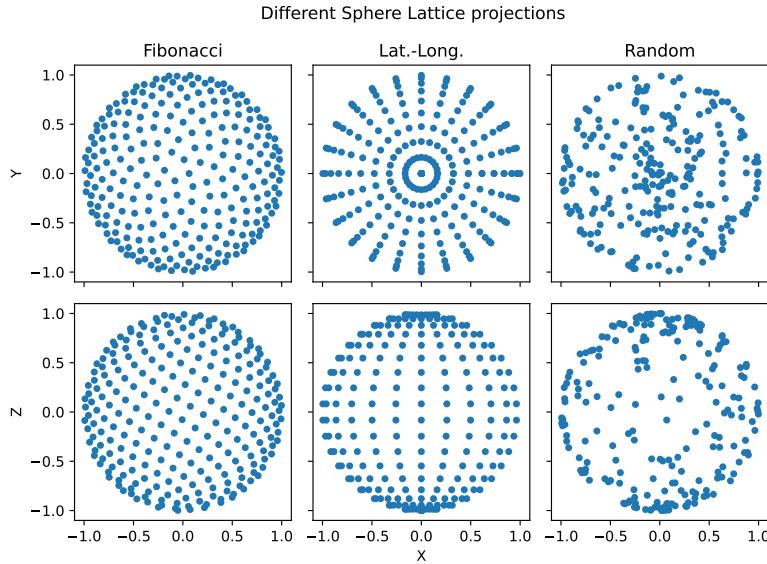
theta and phi are calculated with:

$$\theta = \arccos\left(\frac{1 - 2(p + .5)}{n}\right)$$

and

$$\phi = \frac{2\pi p}{R_{golden}}.$$

Where n represents the total number of particles in the shell, and p represents an individual particle's index, ranging on the interval [0,n). Every shell is given an identical distribution of particles. For each particle, this theta and phi represent the direction of the "kick" velocity vector. They are also assigned a magnitude based on their shell number, where each shell represents a given kick velocity. Velocities are evenly spaced on the semi-closed interval, (0, max\_vel], where the maximum velocity can be input by the user. For our purposes, we adopt a maximum kick velocity of 2.8 km/s (see section 2.1).



**Figure 2.3:** Example projections of distributions of points for a Fibonacci Sphere Lattice (left), a Latitude-Longitude Lattice (center), and for randomly selected theta and phi (right). The Fibonacci lattice is visually the most uniform, while the Latitude-Longitude lattice shows higher densities at the poles, and the random selection has clumping in some regions.

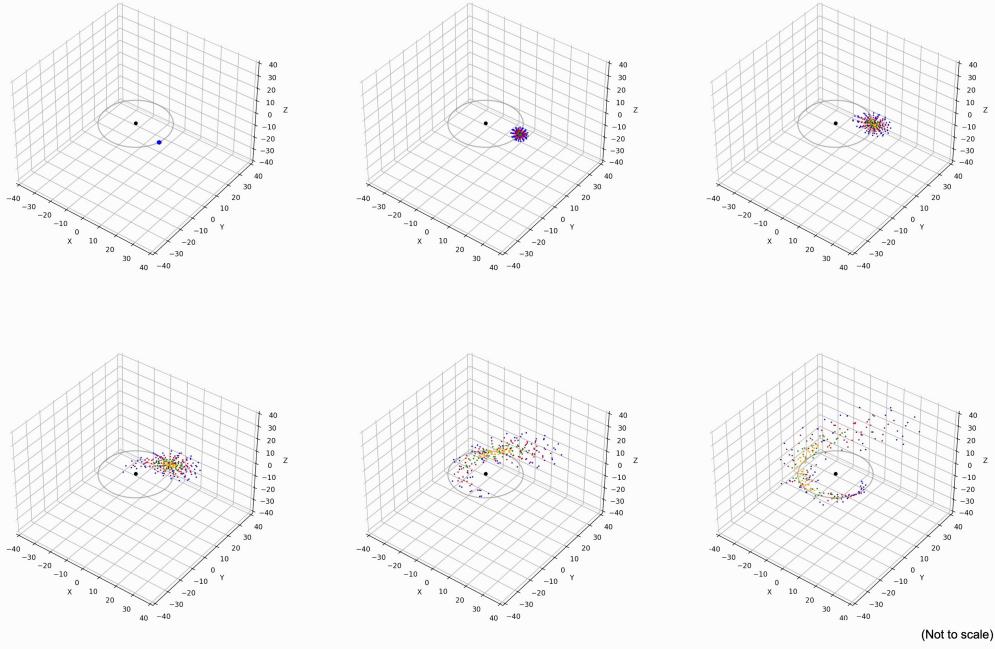
### 2.2.3 Simulation with Keplerian Dynamics

Following the mathematics of Jackson et al. (2014) (Section 2.1 and Appendix A), we are able to calculate the Keplerian orbital parameters of each individual particle, based on its kick velocity vector (as described above) and the initial orbital parameters of the progenitor. Using Keplerian dynamics is preferable in this case for the reasons outlined in section 2.1.

While a full treatment of the Keplerian kick mechanics can be found in Jackson et al. (2014), the equations used to convert the initial orbital parameters (see subsection 2.2.1) and kick velocity vectors to final orbital parameters for a given particle are below, where values with the subscript  $i$  indicate the orbital parameters of the progenitor. General variable definitions are outlined in Nomenclature under "Simulation Parameters."

For notation simplicity, sine and cosine functions will be shown with a capital letter S or C (respectively) with subscript denoting the input to the function. So, for example,  $S_\phi$  is the sine of angle phi.

To generalize to the case where any of  $I_i$ ,  $\omega_i$ , and  $\Omega_i$  is nonzero, we can apply a transformation to our kick direction values ( $\theta_i$  and  $\phi_i$ ) to rotate into the  $I_i, \omega_i, \Omega_i = 0$  frame, simplifying the math substantially. While, as described in section 2.1, our simulation is already in this frame, our code implements the more generalized case. We begin by defining two new angular values for convenience:  $\alpha = \omega_i + f_i$  and  $\beta = \phi_i - \Omega_i$ . From here, our new, transformed  $\theta$  and  $\phi$  can be calculated with



**Figure 2.4:** Stills from an animation of the orbital simulation code. The black circle in the center represents ASASSN-21qj (not to scale), and the gray line represents the progenitor’s orbit. Points are colored according to their initial velocity shell. A much smaller example is being shown than was used in obtaining our final results, for purpose of illustration. As the simulation progresses, the cloud of points gets sheared.

$$\cos(\theta) = C_{\theta_i} C_{I_i} - S_{\theta_i} S_{I_i} S_{\beta},$$

and

$$\tan(\phi) = \frac{S_{\theta_i} (S_{\beta} C_{I_i} C_{\omega_i} - C_{\beta} S_{\omega_i}) + C_{\theta_i} S_{I_i} C_{\omega_i}}{S_{\theta_i} (S_{\beta} C_{I_i} S_{\omega_i} + C_{\beta} C_{\omega_i}) + C_{\theta_i} S_{I_i} S_{\omega_i}}.$$

From here, we also define a dimensionless kick velocity:

$$v_{dim} = \Delta v / v_{orb}$$

As well as a dimensionless position parameter:

$$R_{dim} = 1 + 2 \frac{(1 - e_i^2)^{1/2}}{1 + e_i C_{f_i}} (v_{dim}) S_{\theta} S_{(\phi - f_i)} + \frac{1 - e_i^2}{(1 + e C_{f_i})^2} (v_{dim})^2 \left( C_{\theta}^2 + S_{\theta}^2 S_{(\phi - f_i)}^2 \right)$$

Using this framework, our final orbital parameters are calculated as follows.

### Semi-Major Axis ( $a$ )

$$\frac{a_i}{a} = 1 - (v_{dim})^2 - \frac{2}{\sqrt{1 - e^2}} (v_{dim}) S_{\theta} (S_{(\phi - f_i)} + e S_{\phi})$$

Eccentricity ( $e$ )

$$e^2 = 1 - \left(1 - e_i^2\right) \left(R_{dim} \frac{a}{a'}\right)$$

True Anomaly ( $f$ )

$$\sin(f) = \frac{1}{e} (R_{dim})^{1/2} \left( e_i S_{f_i} + \left(1 - e_i^2\right)^{1/2} (v_{dim}) S_\theta C_{(\phi-f_i)} \right)$$

and

$$\cos(f) = \frac{1}{e} (R_{dim}) (1 + e_i C_{f_i}) - 1.$$

Inclination ( $I$ )

$$\cos(I) = \left[ C_{I_i} + \frac{\sqrt{1-e_i^2}}{1+e_i C_{f_i}} (v_{dim}) S_{\theta_i} (C_\alpha S_\beta - S_\alpha C_\beta C_{I_i}) \right] (R_{dim})^{-1/2}$$

Longitude of Ascending Node ( $\Omega$ )

$$\tan(\Omega) = \frac{S_{\Omega_i} S_{I_i} + \frac{\sqrt{1-e_i^2}}{1+e_i C_{f_i}} (v_{dim}) [C_{\theta_i} (S_{\Omega_i} C_\alpha + C_{\Omega_i} S_\alpha C_{I_i}) - S_{\theta_i} C_{\phi_i} S_{I_i} S_\alpha]}{C_{\Omega_i} S_{I_i} + \frac{\sqrt{1-e_i^2}}{1+e_i C_{f_i}} (v_{dim}) [C_{\theta_i} (C_{\Omega_i} C_\alpha - S_{\Omega_i} S_\alpha C_{I_i}) - S_{\theta_i} C_{\phi_i} S_{I_i} S_\alpha]},$$

Argument of Periapsis ( $\omega$ )

$$\sin(\omega + f) = \frac{S_\alpha S_{I_i}}{S_I}$$

$$\cos(\omega + f) = \frac{1}{C_\Omega} \left( C_{\Omega_i} C_\alpha - S_{\Omega_i} S_\alpha C_{I_i} + S_\Omega S_\alpha \frac{S_{I_i} C_I}{S_I} \right)$$

We make sure to use NumPy's `atan2` function<sup>1</sup> in cases where we have an inverse sine and cosine, or inverse tangent to solve for, so as to ensure the resulting angle remains in the correct quadrant. These equations only work, as referenced in section 2.1, if we assume that the resulting orbit remains closed.

It should be highlighted that orbit calculation based off of Jackson et al. (2014) makes use of the six initial Kepler orbital elements of the progenitor. In order to best suit the needs of the ASASSN-21qj system, however, we have the user input the time difference between the collision and the beginning of the transit instead of the true anomaly, and the code calculates the true anomaly based on the other orbit position values (in a coordinate system where the line of sight comes from the positive x direction).

A full list of the initial code parameters used for our simulation can be found in Table 2.1. The simulation code reads in these initial parameters from an easily editable .csv file (a function to create said .csv file is also available), and returns the time step size for the simulation as well as the total number of particles

---

<sup>1</sup><https://numpy.org/doc/stable/reference/generated/numpy.arctan2.html>

Description	Value
Maximum kick velocity	2.8 km/s
Progenitor velocity at explosion	7 km/s
Time of transit start	1000 days
Simulation start	2000 years
Simulation length	10 years
Eccentricity	0
Inclination	0°
Argument of periapsis	0°
Long. of ascending node	0°
Mass of primary	$1 M_{\odot}$
Mass of secondary (progenitor)	$10 M_{\oplus}$
Number of particles per Shell	5000
Number of time steps	1000
Number of shells	150
Primary radius	$1 R_{\odot}$

**Table 2.1:** The parameter inputs for the initial run of our final simulation.

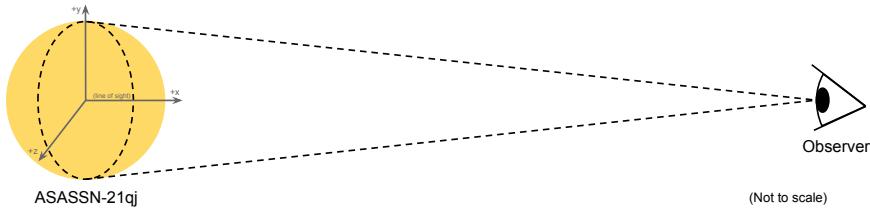
simulated. Additionally, as it simulates, the code creates an archive for each particle, containing a number of important values (see subsection 2.5.2).

## 2.3 Light Curve Modeling

Once a simulation of particles has been run and the position and shell number data have been saved, creating model light curves becomes fairly simple in methodology, though time consuming (see subsection 2.5.2). In general, to understand the conditions for a particle to be considered "transiting", one can imagine a conical volume in space, the base of which represents the image of the star from the point of view of the observer, and the vertex at the location of the observer (see Figure 2.5). It is reasoned that a particle is seen to be transiting the star if it exists within this volume.

However, given the semi-major axis of the progenitor's orbit ( $\sim 18$  AU) is only 0.000015% of the distance to the ASASSN-21qj system ( $\sim 567$  pc) (Kenworthy et al., Submitted for Publication), and the diameter of ASASSN-21qj even smaller than that, the section of this conical volume that is relevant can safely be approximated as a cylinder. Given that the coordinate system of the simulation is set up such that the line of sight comes from the positive x direction, creating an absorption light curve simply becomes a matter of projecting the particle positions onto the y-z plane (see Figure 2.5), and counting—at each time step—the number of particles with distance to the y-z origin less than the radius of ASASSN-21qj .

Using this method, a unique model absorption curve is created for each of the



**Figure 2.5:** The geometric depiction of how a transit is determined. The conic section described is outlined with black dotted lines. The coordinate system defined for the simulation is overlaid in gray. Not to scale.

velocity shells in the simulation (see Figure 3.3). If needed, typical transmission light curves can be found by subtracting the absorption curves from 1 in the usual fashion. In general, at zero inclination, the shells with lower kick velocities—those with particle orbits most similar to the initial progenitor—have thinner and deeper (before normalization) transits, while shells with higher initial kick velocities get sheared further, leading to wider, shallower transits.

## 2.4 Final Light Curve Fitting

Once the modeled light curves are created for each of the velocity shells, they need to be fit to the real ASASSN-21qj light curve data from Kenworthy et al. (Submitted for Publication). We have thus far set up our simulation such that each velocity shell contains the same number of particles. In reality, however, the amount of material ejected from the explosion is not going to be constant at varying kick velocities. This is why we create separate model light curves for each velocity shell.

### 2.4.1 General Fitting Routine

Our final total model (see Figure 3.4) for the light curve will be a linear combination of the individual shell models (created as described above):

$$\text{absorption}(t) = x_1 * \text{shell}_1(t) + x_2 * \text{shell}_2(t) + \dots + x_n * \text{shell}_n(t)$$

For computational reasons, each individual curve is normalized to one, and the normalization factors are saved for later use. The proper linear combination is determined using *SciPy*'s non-negative least squares (NNLS) function<sup>2</sup>. The NNLS works by minimizing the residual of the matrix equation:

$$\|Ax - b\|^2$$

---

<sup>2</sup><https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.nnls.html>

Where A represents the set of individual shell models, b is the (binned, see subsection 2.4.2) real data, and x is the solution vector of model weights ( $x_1 \dots x_n$ ) for the best fit. Because our set of models is not expected to be perfectly orthogonal, there is possible degeneracy in the fit, and some models will non-physically take a weight of 0. However, if there is a high enough resolution in our kick velocities, a general trend for the mass-velocity distribution can still be ascertained. It is crucial that we make use of a non-negative least square fit (which requires all the values in x to be non-negative) because the model weights are a proxy for the relative amount of mass ejected, and mass must be non-negative (see subsection 2.4.5).

It should be noted that, in order for the NNLS to fit properly, we choose to use the absorption light curves instead of the somewhat more usual transmission light curves. This is because, in the transmission case, we would expect a normalized model curve with zero mass to sit at a value of 1. The NNLS, however, expects the zero coefficient case to sit at 0, making the absorption curve the easier option.

### 2.4.2 Binning the ASASSN-21qj Data

In order for the NNLS function to be able to fit properly, we need the size of the time steps in each model to be the same as in the real data. This requires us to re-bin the real ASASSN-21qj data.

This is done by creating a number of bins corresponding to the number of time steps in the simulated data, where each bin is the same length of time, ensuring that the maximum time simulated is large enough to contain the entirety of the real data. From here, for each bin, the average of the real data values contained within that bin is calculated. In cases where a bin contains no data, the value is set to NAN, and a mask showing the locations of the empty bins, as well as the binned data, and corresponding time values, are passed through to the fitting function.

### 2.4.3 Transit Alignment

#### Horizontal Alignment

While the increase in IR emission before the transit gives us a strong sense of when the collision occurred, there is still an uncertainty of  $\sim 6$  months in the exact explosion time (Kenworthy et al., Submitted for Publication). Therefore, in order to obtain the best possible fit to the model, we want to ensure that our transits (modeled and real) are aligned correctly in time. We do this by taking advantage of the fact that the real ASASSN-21qj data begins well before the actual start of the transit. By selecting out only the part of the light curve models where the transit is actually occurring, we are able to systematically adjust the model position over the real data when fitting.

This is done by trimming the modeled data to just those time steps where some shell has at least some material transiting. Starting from some point well before the transit in the binned real data, we take a subset of the real data of equal length to that of the trimmed model data, and perform the fitting procedure,

saving the residual of the best fit model, as output by the NNLS. We can then "shift" the subset of real data selected over by one time step, and repeat until the last possible subset of the real data (or the user-input ending position) is reached. From here, the subset with the minimum best residual is selected for fitting. This process is automated in our code, and a plot of starting offset vs. best-fit model residuals is created (see Figure 3.1).

When fitting these subsets, the list of empty bins is used to remove values from both the real and modeled data to avoid accidentally fitting to non-physical zero values.

It should be noted that an erroneously small residual may be achieved when fitting the modeled transit to the real data at the continuum, wherein many more models are given zero weights. It is therefore necessary to pick the starting position and ending positions of the alignment process with care.

### Vertical Alignment

While both our modeled absorption data and the binned ASASSN-21qj data are normalized, noise in the real data means the pre-transit continuum level sits just above 0. The models, naturally, lack this noise and the regions that are not part of the transit are all exactly 0. This means there exists a small vertical shift between the two, which we do not want skewing our fit results. Given that the form of our final model is a linear combination of all our individual shell models, we can have the NNLS determine our vertical shift by simply adding in an additional dummy "model" array filled with ones, then fit as usual (making sure to include some of the model continuum in the fitting subset). This will create a standard linear increase in the continuum level of our models, so as to better fit the data. This step is conducted only after determining the correct horizontal alignment, so our final model retains the same general form of this initial fit, while being more correctly constrained along the vertical axis.

#### 2.4.4 Testing Model Fit

Due to the large number of individual shell models, our final fit to the data has a large number of degrees of freedom. As such, it is difficult to assess the fit of the model by eye. Instead, we choose to test our model fit using a reduced chi-squared, defined as the chi-squared per degree of freedom:

$$\chi^2_v = \frac{\chi^2}{v}.$$

Where the number of degrees of freedom is equal to the number of data points being fit minus the number of fitting parameters. The chi-squared is calculated in the usual way:

$$\chi^2 = \sum_i \frac{(O_i - C_i)^2}{\sigma_i^2}$$

In order to best contextualize the value of the reduced chi-squared, we create a secondary model, fit to simulated data. We simulate the data by creating an

artificial set of coefficients for our individual light curve models, calculate the linear combination, add a continuum shift, then introduce Gaussian noise. We then try to fit our models to this simulated noisy data in the usual way, and can calculate a reduced chi-squared of that fit for comparison.

#### 2.4.5 Mass-Velocity Profile

Once we have obtained a best-fit model at the correct horizontal and vertical positions, we can obtain a distribution of relative masses at varying velocities by multiplying—for each shell—the normalization coefficient from before by the shell's best-fit weight, as output by the NNLS. These relative masses are then normalized to accurately represent the fraction of the total cloud mass contained within the shell. Plotting these values versus the corresponding shell kick velocities yields a mass-velocity profile (see Figure 3.6). The trend(s) present in this profile may be used to better analyze the initial conditions of the cloud's formation.

### 2.5 Workflow

In order to minimize computation time whilst maximizing the resolution of the simulation, the actual workflow implemented does not simply go from orbit simulation to light curve creation to light curve fitting. Instead, the orbital simulation is completed in a two-step process.

In recognizing that, for most kick velocities, the majority of the resulting cloud will not end up transiting the star, a smaller run can be used to filter out those particles which will certainly not be relevant to the final results. For this work, the initial simulation creates 150 spherical shells of 5000 particles each, ranging in initial kick velocity from 0 to 2.8 km/s. Making use of the light curve creation code, the particles which pass the origin of the Y-Z plane within 3 times the radius of ASASSN-21qj are identified as "filtering particles".

For each velocity shell, the filtering particles identified by this process are then used to select a subset of particles from a higher-resolution shell of 1 million particles. By selecting (and modeling) only those high-resolution particle(s) which are closest to the filtered subset of original particles, the total number of particles actually simulated can be cut by several orders of magnitude without a loss in effective resolution.

#### 2.5.1 Particle Filtering

The process for identifying the high-resolution subset to model is as follows. First, an angular "search radius" ( $\alpha_{\text{search}}$ ) around each selected low-resolution particle is determined geometrically:

$$\alpha_{\text{search}} = \sqrt{4\pi/n}$$

Where  $n$  is the number of particles per shell from the initial run (5000). In order to avoid issues with the convergence of longitudinal coordinates at the poles, we make use of the haversine function.

$$\text{haversin}(\alpha_{\text{search}}) = \sin^2\left(\frac{\alpha_{\text{search}}}{2}\right) = \frac{1 - \cos(\alpha_{\text{search}})}{2}$$

We then compute the haversine distance between each low-resolution particle and the particles on the high-resolution spheres.

$$\text{haversin}(\alpha_{\text{dist}}) = \text{haversin}(\theta_h - \theta_l) + \cos(\theta_h) \cos(\theta_l) \text{haversin}(\phi_h - \phi_l),$$

Where subscript *l* represents the initial simulation low-resolution point we are searching from, and *h* represents the point on the high-resolution sphere being tested.

For each velocity shell, all the particles on the high-resolution sphere which are within the haversine search radius of the close-passing particles from the initial simulation are selected to be run in the final simulation. In doing so, we are able to effectively model 150 velocity shells of 1 million particles each, by only actually calculating the new orbital parameters of ~11.5 million particles (see subsection 2.5.2).

It should be noted that, for initial particle counts significantly less than 1 million/shell, there is a nonlinear inverse relationship between the number of particles in the initial simulation and the number of particles selected by these filtering steps. This is a result of the geometric way we define the search radius. In general, the more particles are in the initial simulation, the smaller the search radius is per particle, and the fewer particles end up being selected by the filter. This allows us to somewhat reduce the total simulation time by investing more time on the front end.

From here, the final simulation is run, the resulting model light curves are created, and these are fit to the real data as described in sections 2.2, 2.3, and 2.4, respectively.

### 2.5.2 Some Notes on Computation Time and Storage Requirements

For this project, work was conducted on a 2020 Macbook Pro, running on an Apple M1 chip, with 16GB RAM and a 1TB solid-state drive. While computing time may vary, the code is written such that it should run to completion on any modern machine without overflowing the memory, given that said machine has enough storage for the individual particle simulations.

While the filtering steps described in the above subsection can decrease the number of particles simulated by at least an order of magnitude, there are still some considerations to be made regarding computational requirements. Saving the 3-dimensional locations for several hundred time steps of ~11.5 million particles creates new computational problems. In order to save on storage, each particle's information is saved to a numbered .npz archive, which contains its x, y, and z positions, as well as its velocity shell number and initial kick direction (in angular coordinates). By using NumPy's native binary format (compared to, say, a .csv), file i/o is simplified, and storage space is lessened—each archive taking up just 13KB.

Step	Time
Initial Simulation	50m 49s
POI Identification	19m 30s
Filtering	22m 10s
Full Simulation	14h 31m 10s
Light Curve Creation	5h 24m 47s
Model Fitting	0.55s
Total	21h 28m 12.55s

**Table 2.2:** The time breakdown of various steps for our final simulation and model. Note that times will vary from run to run.

It should be noted, however, that for  $\sim$ 11.5 million particles, this still ends up at over 150 GB of data. Each particle is saved to its own .npz archive, because it is simply not possible for python to retain this much data at once without overflowing the machine's RAM and, presently, NumPy offers no native way of appending to these archives. As such, while using .npz saves on storage space and (as a generality) file i/o time, the sheer number of files being read ultimately increases the amount of time it takes for the light curve-creation code to run.

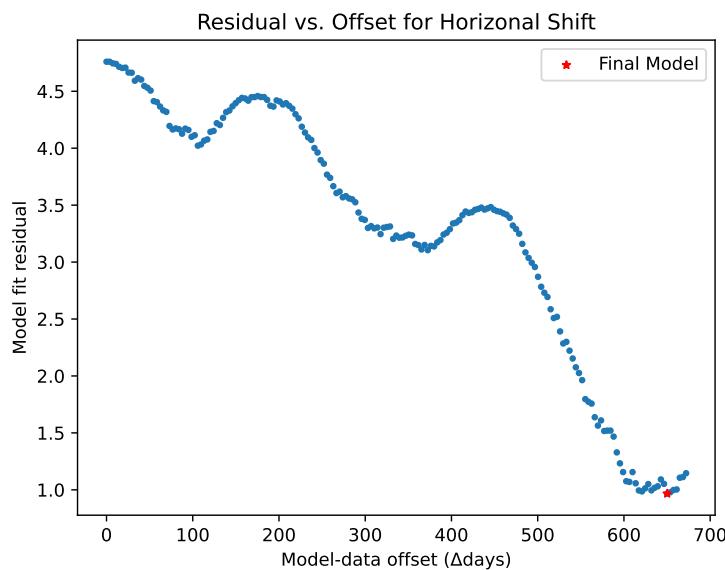
In the end, our initial simulation run of 750,000 particles (5000 per shell times 150 shells) yielded 20984 particles of interest (POI) which transited within  $3R_\odot$  of the primary. From these, the filtering steps selected out 11,559,349 particles of a possible 150 million from the high-resolution spheres. A breakdown of the times taken for our final simulation can be found in Table 2.2, though it should be emphasized that these values are expected to vary.

# 3. Results

## 3.1 Model Alignment

### Horizontal Alignment

The residual plot of the horizontal alignment process can be found in Figure 3.1. As the model and real data are shifted over one another, there is a general downward trend in residual with several local minima representing locations at which the center of the model light curves aligns momentarily with some significant feature in the real data. However, this eventually leads to a clear, sharp absolute minimum, with a fit residual just under 1.

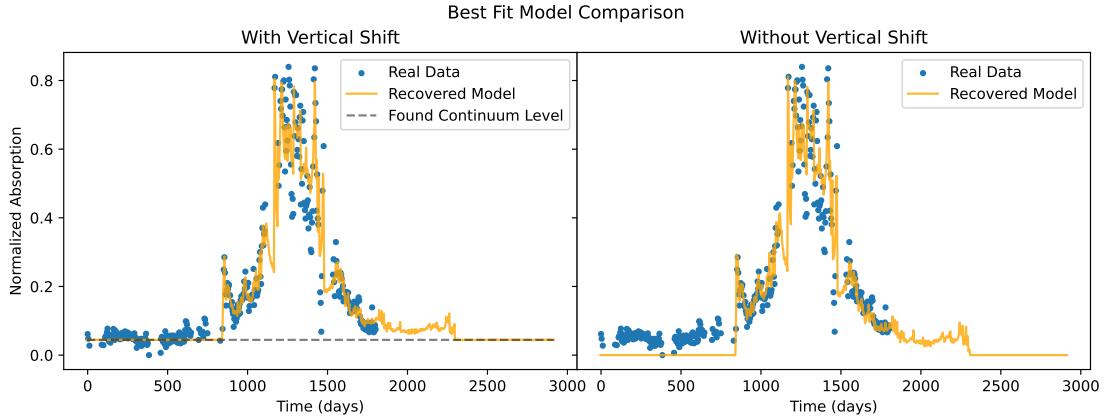


**Figure 3.1:** The best-fit model residuals vs. model-data offset for the horizontal alignment procedure. The minimum residual, corresponding to the location of our final model fit, is marked with a red star.

### Vertical Alignment

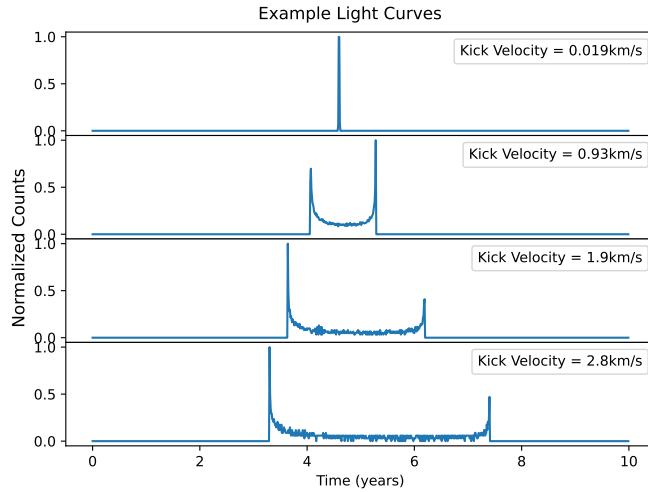
A figure comparing two model fits—with and without the extra linear vertical shift—can be found in Figure 3.2. The general form the fit takes (especially in the

transit itself) is the same between the two, with the same features being fit at comparable values. On the plot with vertical shift, however, the continuum level of the model is much more closely fit to the data.



**Figure 3.2:** A comparison between a model with a vertical shift component (left) and one without (right). The general form of the two is very similar. On the plot with the vertical shift, the black dashed line indicates the coefficient value for the vertical shift component, as found by the NNLS.

## 3.2 Final Model Fit

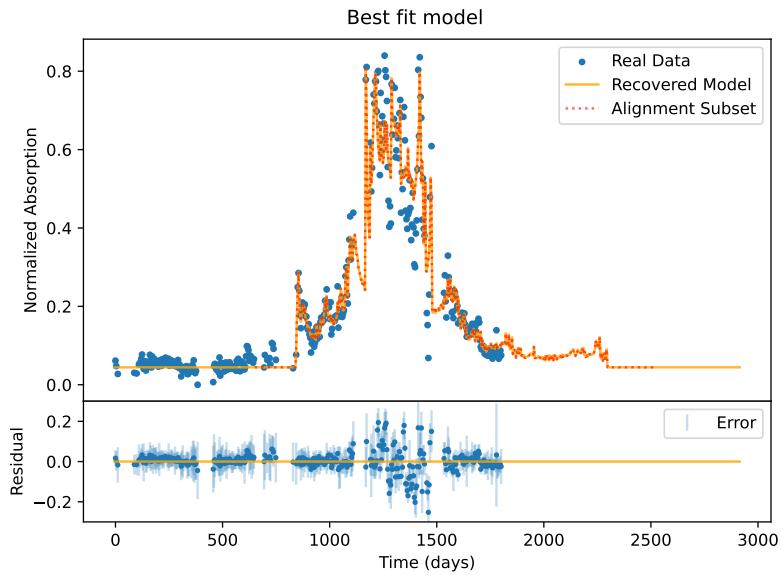


**Figure 3.3:** A number of example light curves from our final simulation, from shells at varying kick velocities (lowest at the top, highest at the bottom). Each curve is individually normalized to one for fitting purposes. In general, the shells that are kicked hardest experience the most shear, leading to longer transits.

A subset of the individual shell light curves can be seen in Figure 3.3 (normalized for computational convenience). Each light curve shows a sharp spike at the beginning and end of the transit (though, for the lowest-kick curves, these blend

into one large spike), representing the passing of the front and back edges of the shells. These sections have lower optical depth, as there are more particles in the line of sight due to the shell geometry. As is expected, the light curves at lower initial kick velocities are much thinner, with most the particles remaining on orbits similar to each other and to the original progenitor. In contrast, the shells with higher kick velocities experience a greater shear as they move along, leading to longer-lasting transits. Further discussion of how inclination may impact this trend can be found in section 4.3.

Of those light curves which show two distinct start and end peaks, the region in the middle is relatively rather noisy. This is most likely due to quantization noise, a result of the fact that relatively few particles are transiting at any given time. However, given that the NNLS is fitting the standard residual for the matrix equation, the larger features (the spikes) are going to be the most relevant for the fit. The central noise is not large enough to be a cause of concern.

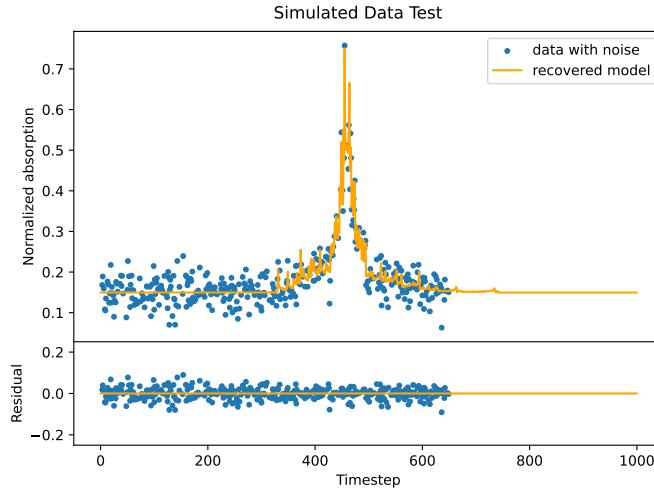


**Figure 3.4:** The final best-fit model to the ASASSN-21qj data. The full model is shown in orange, while the subset of the model used for aligning (see subsection 2.4.3) is highlighted with red. Error bars are shown on just the residual plot, for clarity.

Our final model fit with the real ASASSN-21qj data can be found in Figure 3.4, along with a residual plot (with added error bars). As expected, the beginning of the transit is clearly steeper than the end. The continuum data sits closely to the model line, instead of just above. Additionally, the spike-like features seen in the real data appear to be fit well by the model. In the residual plot, we can see that the errors on the data generally agree with the model, though there is some disagreement for a handful of points during the transit.

### 3.2.1 Comparison to Simulated Data

The simulated data test output can be found in Figure 3.5. In order to simulate this data, we calculate the linear combination of the model light curves with

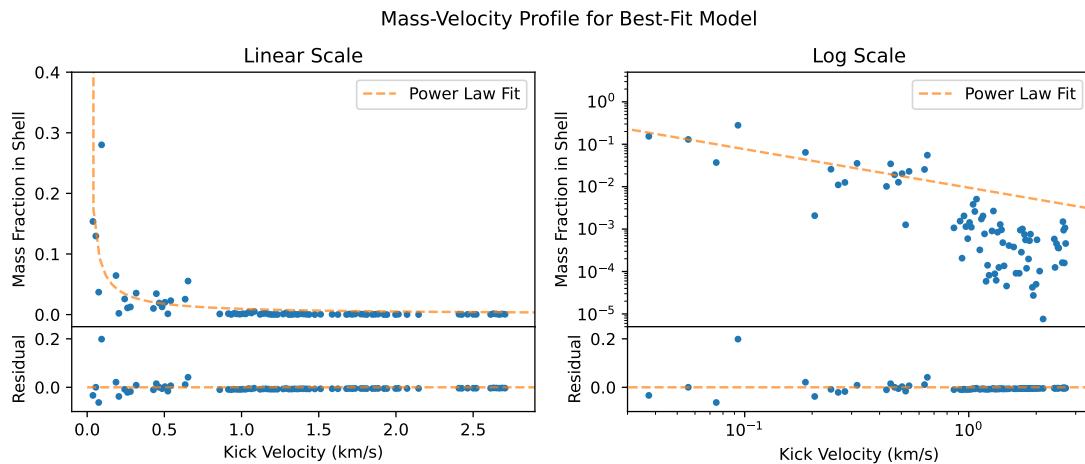


**Figure 3.5:** The simulated data and related best-fit model, for comparison purposes.

coefficients equal to  $.01n^{-1}$ , where  $n$  is the model number. We add Gaussian noise with a standard deviation of 0.03. We then fit our models to this noisy data.

### 3.3 Mass-Velocity Profile

The final Mass-Velocity profile for the model in Figure 3.4, on both linear and log scaling, can be found in Figure 3.6. Of the 150 shells modeled, 78 had nonzero coefficients to determine relative mass fraction. Only those nonzero coefficients are shown here. The mass data is normalized such that the value represents the fraction of the total mass ejected at the given kick velocity.



**Figure 3.6:** The Mass-Velocity profile for the best-fit model, as well as the corresponding power law fit on linear (left) and logarithmic (right) axes. Residual plots are shown in the lower frames.

As expected, the plot shows that the majority of the matter from the explosion is kicked at relatively lower speeds, staying somewhat nearer the progenitor orbit.

The mass fraction drops off very quickly, with most of the values above  $\sim 1$  km/s sitting very near zero.

# 4. Discussion

## 4.1 Model Fit

Ultimately, we were able to very closely fit our simulated transit models to the ASASSN-21qj data. In general, the fit appears good, with reasonably random scatter on the residual plot, indicating that over-fitting is not an issue, especially in the region of the transit. The model transit clearly extends beyond the end of the real data. The residual for this fit, as output by the NNLS function, is 0.97.

We calculated a reduced chi-squared for this fit of 3.42 (199 degrees of freedom). While not a perfect reduced chi-squared, the value is low enough to indicate this is a very promising initial simulation. For comparison, our simulated data set (see Figure 3.5) has a reduced chi-squared of 0.21 (215 degrees of freedom), about a one order-of-magnitude difference. This indicates that, while our model is working fairly well for an initial simulation, there is still a fair amount of room for improvement. It should be noted that, compared to our real model fit, the model fit to the simulated data experiences significantly less scatter around the transit region.

## 4.2 Mass-Velocity Profile

Upon seeing the general shape of the Mass-Velocity profile, we decided to fit the data to a power-law. Using the *SciPy curve\_fit* function, we calculated a least-squares fit of the form

$$y = ax^b,$$

and calculated its R-squared statistic. We find a best-fit power law with coefficient  $a = 0.0095 \pm 0.0028$  and exponent  $b = -0.907 \pm 0.106$ . The R-squared statistic on the fit is 0.568. While this fit visually appears fairly good in the linear scale plot of Figure 3.6, the log scale shows it systematically over-estimates the mass fraction value at high kick velocities. One possible explanation for this can be found in section 2.1.

It should be noted that, for the purposes of this project, a power-law function becomes nonphysical for the Mass-Velocity profile at very low velocities, where it tends toward infinity. An infinite mass at zero kick velocity is obviously nonphysical

for a planetesimal-sized object. This power-law fit only constitutes an initial attempt at describing the behavior of the ejecta from the collision, and the tentative fit is reasonable.

A recent paper, Raducan et al. (2020), studied ejecta from collisions with asteroids in the context of the <https://dart.jhuapl.edu/>. Despite analyzing a very different geometry, the general shape of their mass-velocity distribution (Fig. 7a in Raducan et al. (2020)) is quite similar in form to ours, with a strong linear section at low velocity and a quick drop-off, implying some underlying form for mass-velocity distributions of collision ejecta.

## 4.3 Assumptions and Computational Impacts

While most of the assumptions outlined in section 2.1 can be shown to have negligible impact on our final model, a few remain relevant. The most significant are discussed below.

### 4.3.1 Maximum Kick Velocity

Because of the limitations of the mathematics used from Jackson et al. (2014), we were limited in the maximum possible kick velocity in our simulations. If we kick a given shell too hard, some of its particles reach escape velocity, creating divide-by-zero errors in the math. For a more long-term project (say, disk formation from collisions), this would not be an issue, as particles at escape velocity quickly become unimportant to the long-term evolution of such a system. In our case, however, because we are studying only the first pass of this debris cloud, it is possible that escape-velocity particles would be more relevant.

If the ASASSN-21qj data includes transiting material ejected at velocities much greater than 2.8 km/s, that would most likely translate to our code via a poorer fit in the modeling step. This would be particularly true for our high-velocity shells, whose mass values may be artificially altered when attempting to fit to these higher-velocity data. However, as seen in Figure 3.6, these high-velocity shells contribute very little to the overall mass of the cloud, such that this impact would be barely perceptible, if not negligible.

That being said, in a future iteration of this code, this could be alleviated altogether by diverting to alternative simulation calculations for just those particles which cause divide-by-zero errors. This could be done via a full n-body treatment (deemed too intensive for this project in general), and could probably be implemented with one of many existing astrophysical code libraries. However, doing so would ultimately increase computation time for a given number of input particles.

### 4.3.2 Spherical Symmetry

Our results here begin with the assumption that the initial explosion from the collision is spherically symmetric. In general, we would expect this to change the general shape of our final model (especially at the wings of the transit), though

predicting precisely how alternative geometries would do so is complicated and very much degenerate.

By design, our code has been written to be extremely modular. The process to create the initial kick velocity vectors (as described in subsection 2.2.2) is relegated to its own function. In principle, any function which creates these vectors and can pass them back in separate arrays (one for the kick magnitude, one for theta, and one for phi) can just as well be slotted into the code. Similarly, the workflow described in section 2.5 is modular and can be adjusted to best fit the needs of the geometry being studied.

#### 4.3.3 Zero Inclination

Perhaps our most tenuous assumption, our modeled debris cloud is created from an initial progenitor with zero inclination. As a result, the transits in the models for the shells with lesser kick velocities are going to be deeper than expected. In terms of actually fitting the ASASSN-21qj data, these individual models are all normalized to one anyway. Meaning, unless the real transit is just scraping the outer edge of the debris cloud (unlikely, given the large normalized absorption), the quality of the model fit is unlikely to be impacted by the change in depth.

Possible issues occur when it comes to the analysis of the Mass-Velocity profile. In an edge-on case, a disproportionate number particles from the low-velocity shells will be seen transiting the star. This effect would be expected to cause a general over-estimate in the relative mass of these shells. Additionally, the edge-on case would create somewhat wider transits than one would expect to see in the inclined case at all velocities. This would be expected to create a general under-estimate of mass fraction, but especially impacting the highest velocity shells. Based on our results in Figure 3.6, either of these effects may improve the fit of our power law.

It would be fruitful to adjust the light curve-creation module such that the user can rotate the line of sight relative to the system. Doing so would allow for adjustment on the inclination, without having to re-run the entire 20+ hour simulation from square one. Such an improvement may open up the opportunity to use machine learning techniques to try to determine the inclination of the system without further observation, though this would be limited by the degeneracy inherent in the fitting process and data noise.

# 5. Conclusion

We have created a system for simulating debris clouds resulting from planetary collisions. By breaking down the simulation into spherical shells, we are able to obtain light curve models at varying velocities. These light curve models can be fit to real observational data using a non-negative least squares method. The reduced chi-squared of our model is **3.42**, as compared to **0.21** for a simulated test data set—imperfect, but very promising.

The resulting model fit is used to create a Mass-Velocity profile for the initial explosion. In doing so, we have obtained a best-fit power-law function for the Mass-Velocity profile of this collision event, with exponent **-0.907** and coefficient **0.0095**. The R-squared metric for this fit is **0.568**, though the power law function is an imperfect choice of model, as it is nonphysical at small kick velocities. Again, while this is imperfect, it serves as a great, and promising, starting point.

In general, we know that we ought not expect a planetesimal collision like this to eject material in a spherically symmetric way. In fact, much of the current work studying collisional events in space focus on glancing collisions (Canup, 2004; Raducan et al., 2020). With this understanding, the fact that we are achieving such a reasonable model fit is rather remarkable. We propose two possible explanations for the fit. The first is that such a model is simply so degenerate that a decent fit can be achieved regardless. This, however, does not explain the similarity of the shape of our mass-velocity distribution to that of Raducan et al. (2020). The second is that, in these cloud-creating collisions, asymmetries from the initial collision are very quickly smoothed over due to movement within the cloud, making the spherical symmetric case a reasonable fit. The latter of these two is the more exciting, astrophysically speaking.

If the spherical symmetry case is, in fact, a reasonable fit, then the results related to the long-term evolution of the cloud from Jackson et al. (2014) should hold. In that study, the authors find that, in the spherically symmetric cloud case, the resulting disk from the shearing of the collision remnant remains strongly asymmetric for several orbits, as the majority of the material continues to pass through the point of collision, before eventually smoothing out. Such a process could be well observed with future observations of new systems on the forthcoming Vera C. Rubin Telescope (LSST) and similar projects.

## 5.1 Future work

Beyond the possible improvements to the code described in section 4.3, there is much room for future work to be conducted both specifically with the ASASSN-21qj system, and more generally for our light curve simulation.

On the side of the ASASSN-21qj system, one of the most significant ways to improve upon the work presented here would be to try and ascertain a better grasp of the system inclination. Additionally, in a few years' time, it will be important to return to ASASSN-21qj , in order to observe how the dust cloud will have changed from present. It will be important to see if it follows the results for a planetesimal collision found in Jackson et al. (2014).

On the code side, the modular nature of the program means that, in principle, work can already begin assessing different (asymmetrical) cloud morphologies, as well as initial transverse velocities and non-circular orbits. Future improvements ought to be made, too, to allow the user more flexibility in the maximum kick velocity and inclination inputs (as described in section 4.3). Additionally, later down the line, it would be useful to adjust the code to account for radiation pressure in the star. In doing so, we could aim to allow users to input different parameters related to dust opacities, and dust grain size distributions.

Planetesimal collisions are an essential phenomenon in the late stages of planetary formation. While observations of such events are currently few and far between, we are certain to see more in the near future, as telescopes become more powerful and astronomy, as a whole, continues to take enormous amounts of data. Taking the time to develop tools like this code now will help us to better study and understand such phenomena in the future.

# Bibliography

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A , 558, A33, doi: [10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068)
- Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ , 156, 123, doi: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f)
- Cameron, A. G. W. 1962, Icarus , 1, 13, doi: [10.1016/0019-1035\(62\)90005-2](https://doi.org/10.1016/0019-1035(62)90005-2)
- Canup, R. M. 2004, Icarus , 168, 433, doi: [10.1016/j.icarus.2003.09.028](https://doi.org/10.1016/j.icarus.2003.09.028)
- . 2005, Science, 307, 546, doi: [10.1126/science.1106818](https://doi.org/10.1126/science.1106818)
- Cutri, R. M., Skrutskie, M. F., van Dyk, S., et al. 2003, 2MASS All Sky Catalog of point sources.
- González, Á. 2010, Mathematical Geosciences, 42, 49, doi: [10.1007/s11004-009-9257-x](https://doi.org/10.1007/s11004-009-9257-x)
- Halliday, A. N., & Canup, R. M. 2023, Nature Reviews Earth and Environment, 4, 19, doi: [10.1038/s43017-022-00370-0](https://doi.org/10.1038/s43017-022-00370-0)
- Hoyle, F. 1960, QJRAS , 1, 28
- Hunter, J. D. 2007, Computing in Science and Engineering, 9, 90, doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Jackson, A. P., Wyatt, M. C., Bonsor, A., & Veras, D. 2014, MNRAS , 440, 3757, doi: [10.1093/mnras/stu476](https://doi.org/10.1093/mnras/stu476)
- Kenworthy, M., Lock, S., Kennedy, G., et al. Submitted for Publication, Nature
- Kochanek, C. S., Shappee, B. J., Stanek, K. Z., et al. 2017, PASP , 129, 104502, doi: [10.1088/1538-3873/aa80d9](https://doi.org/10.1088/1538-3873/aa80d9)
- Oliphant, T. E. 2006, A guide to NumPy, Vol. 1 (Trelgol Publishing USA)
- Raducan, S. D., Davison, T. M., & Collins, G. S. 2020, Planet. Space Sci. , 180, 104756, doi: [10.1016/j.pss.2019.104756](https://doi.org/10.1016/j.pss.2019.104756)

- Rizzo Smith, M., Jayasinghe, T., Stanek, K. Z., et al. 2021, The Astronomer's Telegram, 14879, 1
- Rufu, R., & Canup, R. M. 2022, ApJ , 928, 123, doi: [10.3847/1538-4357/ac525a](https://doi.org/10.3847/1538-4357/ac525a)
- Shappee, B. J., Prieto, J. L., Grupe, D., et al. 2014, ApJ , 788, 48, doi: [10.1088/0004-637x/788/1/48](https://doi.org/10.1088/0004-637x/788/1/48)
- Solomatova, N. V., & Caracas, R. 2023, Communications Earth and Environment, 4, 38, doi: [10.1038/s43247-023-00694-9](https://doi.org/10.1038/s43247-023-00694-9)
- Strom, K. M., Strom, S. E., Edwards, S., Cabrit, S., & Skrutskie, M. F. 1989, AJ , 97, 1451, doi: [10.1086/115085](https://doi.org/10.1086/115085)
- Terebey, S., Shu, F. H., & Cassen, P. 1984, ApJ , 286, 529, doi: [10.1086/162628](https://doi.org/10.1086/162628)
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261, doi: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2)
- Williams, J. P., & Cieza, L. A. 2011, ARA&A , 49, 67, doi: [10.1146/annurev-astro-081710-102548](https://doi.org/10.1146/annurev-astro-081710-102548)
- Wright, E. L., Eisenhardt, P. R. M., Mainzer, A. K., et al. 2010, AJ , 140, 1868, doi: [10.1088/0004-6256/140/6/1868](https://doi.org/10.1088/0004-6256/140/6/1868)
- Wyatt, M. C. 2008, ARA&A , 46, 339, doi: [10.1146/annurev.astro.45.051806.110525](https://doi.org/10.1146/annurev.astro.45.051806.110525)
- Wyatt, M. C., Panić, O., Kennedy, G. M., & Matrà, L. 2015, Ap&SS , 357, 103, doi: [10.1007/s10509-015-2315-6](https://doi.org/10.1007/s10509-015-2315-6)