The Disruption of Planetary Systems in the Galactic Centre

URN 6428183

Department of Physics, University of Surrey, Guildford GU2 7XH, United Kingdom

9th June, 2020

Abstract

Supermassive Black Holes (SMBH) are considered to be an important, and possibly vital, feature at the centre of most large galaxies, including the Milky Way [1]. Undoubtedly, the gravitational influence of these bodies on the evolution of their surrounding environment, and that of nearby planetary systems, can have an immense impact. Perhaps the most disruptive impact on a planetary system would be to lose a planet. Such an occurrence could lead to tidal disruption effects, observed as electromagnetic flares emanating from a black hole. We must then ask, under what initial conditions will a planet be captured by a SMBH?

This report aims to answer this question by investigating the pericentre, p, and planet distance, r_p, of a gravitational encounter comprising of a planetary system which is in a parabolic orbit around a SMBH. To aid this investigation, a large number of simulations are performed and their outcomes classified as a function of the initial parameters. From this, the favourable conditions for a SMBH to capture a planet are found to be approximately $300 \leq p \leq 400$ au, and $10 \leq r_p \leq 20$ au. A second planet is then added to the simulations revealing that the three and four-body gravitational encounter results are equivalent.

Supervisor: Dr Alessia Gualandris

Contents

Acknowledgements

I would like to thank my project supervisor, Dr Alessia Gualandris, for her continual assistance and scientific contributions relating to the outlined project. Additional thanks are given to Dr Robert Izzard for his noteworthy project feedback.

1 Introduction

Consider a simple planetary system containing a single planet in a stable orbit around a sun-like star. What is the likelihood that such a planet will vacate its star at some point in its future? Perhaps the most natural response to this question is to say it's extremely unlikely. Such a belief is established under the assumption of an isolated system which experiences no significant external gravitational forces. However, within the context of a planetary system in the neighbourhood of a massive body, the outcome becomes unclear. A precise response to the question is then predicated on the answer to a more interesting question. Under what conditions will a planet be removed from its star?

1.1 Spheres of Influence

A similar problem is presented in recent research investigating the chaos-assisted capture of irregular moons by large planets like Jupiter. Notably, this research explores an analogous gravitational encounter in an attempt to "shed new light on how planets capture other bodies" [2]. For this purpose, a detailed look at the sphere of influence of Jupiter proves useful in understanding the process by which this happens. This so-called sphere of influence refers to the region within which the gravitational influence of a body is dominant. In the context of our three-body gravitational encounter, the disruption of a planetary system is dependent on the sphere of influence exerted by the star with respect to the massive body. For example, a star with a large sphere of influence is more likely to retain its planet during one of these gravitational encounters. It is therefore important to examine the sphere of influence of the star when answering the latter question.

Indeed, several models of the sphere of influence exist, with the most prominent being the Hill sphere and the Laplace sphere, which specify fixed-size spheres based on constant parameters [3]. Other numerical models, which are more dynamical, describe the radius of the sphere of influence to be a function of the initial relative velocity of an encounter [3]. However, for this investigation, the Hill sphere is the most suitable model for consideration as it provides a maximum distance below which the planet should not be removed from its star. The Hill radius, R_h, is thusly defined as [4]

$$R_h = p\left(\frac{M_{\text{star}}}{3M_{\text{bh}}}\right)^{1/3}. \tag{1}$$

In the previously mentioned research investigating how a Jupiter-like planet can capture irregular moons, the Hill sphere proves to be a significant consideration. However, greater insight into this problem could be achieved by studying how the orbital parameters of an incoming body can affect the outcome of an encounter. This report therefore aims to investigate the effect of varying orbital parameters of an analogous gravitational system. To do this, we must first establish an unambiguous description of a simple three-body gravitational encounter.

1.2 A Simple Three-Body Gravitational Encounter

Here we clearly define a system of bodies to be used as a framework within which important orbital parameters can be explored. Such a system is defined to have three bodies: a SMBH at the origin of the system, a star in a parabolic orbit around the SMBH, and a planet in a circular orbit around the star. Naturally, a full description of this system cannot be achieved without specifying the relative positions and velocities of each body. In the simplest case, we only require three parameters to do this (assuming a randomized planet orbital position): the true anomaly θ, the pericentre p,

and the distance between the planet and star, r_p. Fig. 1 provides a visual representation of this arrangement.

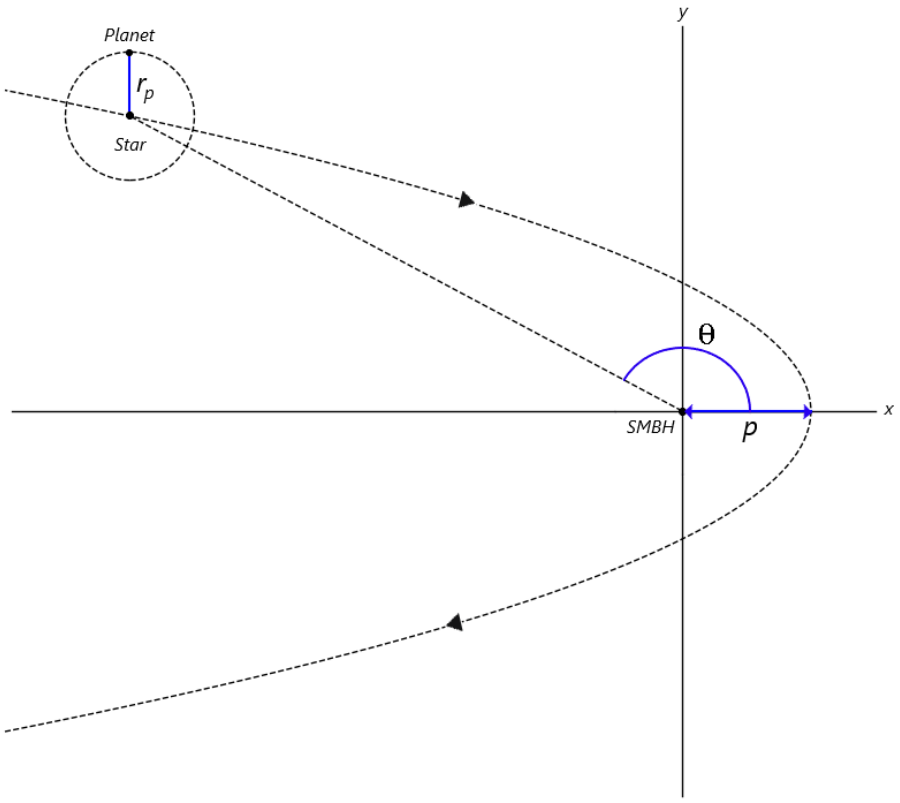

Figure 1: A simple arrangement for a three-body gravitational encounter. For simplicity, this example shows the planet's circular orbit in the same plane as the stars parabolic orbit.

Within this arrangement, an alteration of the true anomaly, θ, will change the starting position of the star and planet along a fixed parabolic path. This adjustment does not have a discernible effect on the removal of a planet from its system without an extensive, and time-consuming, exploration of the planets phase. For this reason, the effect of varying θ on the outcome of a gravitational encounter is not explored in this investigation. Instead, the distance of closest approach, also known as the pericentre p, and the planet distance, r_p, will be systematically varied as 'initial conditions' across numerous simulations. Following this, the outcomes of the simulated encounters are classified based on three possibilities:

- The planet is captured by the SMBH.
- The planet remains in a stable orbit around the star.
- The planet is unbound from both the star and the SMBH.

Multiple simulations for each set of p and r_p will be carried out whilst randomizing the planets orbital position. Consequently, the fraction of outcomes in each of the above categories can be calculated as a function of the simulated conditions. In addition to this, the outcomes of a four-body gravitational encounter involving two planets will be treated in the same way before the results for each planet are combined. This will allow us to determine the favourable conditions required for each of the above categories, thereby answering our original question.

2 Method

2.1 Generating Initial Conditions

To explore the outlined problem we must first determine the 'initial conditions' of a three-body system resembling the system in Fig. 1. These initial conditions include the mass of each body, and their initial x, y and z position and velocity components. For this investigation, the masses within the system will be identical to that of a Jupiter-Sun planetary system orbiting Sagittarius A*, the SMBH at the centre of the Milky Way. These masses are provided in Table 1 [5, 6].

Body	Mass ($M_\odot$)
SMBH	4.0×10^6
Star	1.0
Planet	9.543×10^{-4}

Table 1: The masses used for the SMBH, star and planet [5, 6].

It also proves pragmatic to place the SMBH at the origin of the system, and to assign its initial velocity a value of zero. On this basis, the remaining initial conditions were calculated within a C++ project by implementing Eqs. (2) to (12). The code for this is listed in Appendix A.1, and the full C++ project is accessible using the following github link: `https://github.com/ra00578/Disruption-Of-Planetary-Systems`.

2.1.1 Initial Conditions for the Star

The initial position and velocity of the star is dependent on two parameters only: the true anomaly θ, and the pericentre p of its parabolic orbit. Assuming this parabolic orbit is in the z-plane (i.e. $z_{\text{star}} = 0$), the stars initial x and y positions are given by Eqs. (2) and (3) [7].

$$x_{\text{star}} = p\left(\frac{\cos(\theta)}{1+\cos(\theta)}\right) \tag{2}$$

$$y_{\text{star}} = p\left(\frac{\sin(\theta)}{1+\cos(\theta)}\right) \tag{3}$$

Fundamentally, a parabolic orbit defines a trajectory on which a body will coast to infinity and never return. As a consequence, the velocity of the star must be equal to the escape velocity of the system as seen in Eq. (4)

$$v_{\text{par}} = v_{\text{esc}} = \sqrt{\frac{2GM}{r}} \tag{4}$$

where M is the total mass of the star and SMBH, and r is the distance between them [7]. Using this description, and the previously calculated positional coordinates, it is possible to calculate the total velocity of the parabolic orbit. Hence, if this orbit is parallel to the z-plane (i.e. $v_{z,\,\text{star}} = 0$), Eqs. (5) and (6) provide a means for calculating the initial x and y velocities of the star.

$$v_{x,\,\text{star}} = v_{\text{par}} \sin\left(\frac{\theta}{2}\right) \tag{5}$$

$$v_{y,\,\text{star}} = -v_{\text{par}} \cos\left(\frac{\theta}{2}\right) \tag{6}$$

2.1.2 Initial Conditions for the Planet

Within the setup outlined in Fig. 1, the initial position and velocity of the planet can be specified in relation to the star by randomizing its orbital position. This randomization is achieved using two angles, ϕ and i, seen in Fig. 2, which are randomly generated between 0 and 2π using a pseudo-random number generator.

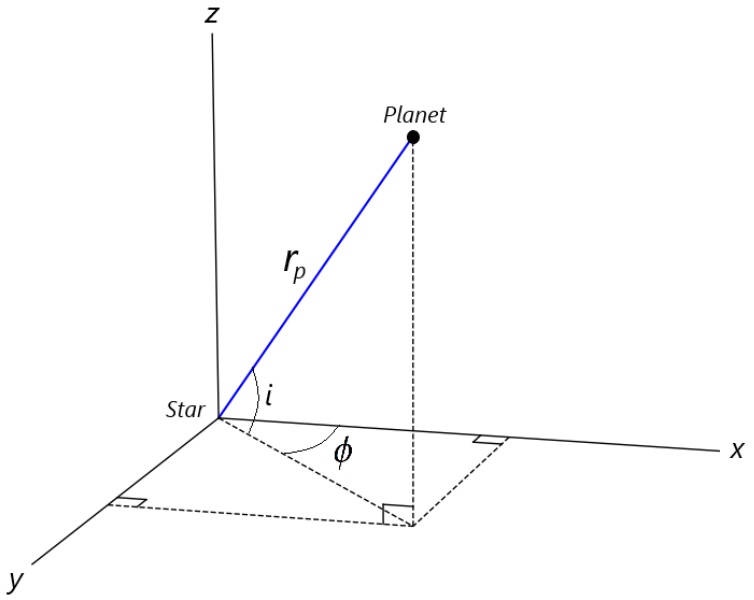

Figure 2: Randomizing the position of the planet with respect to the star.

This randomization is necessary for the calculation of likelihood for each possible simulation outcome. Accordingly, a large number of simulations with the same pericentre and planet distance, but different randomized planet orbital positions, are required for this calculation. For the purpose of the proposed problem, using a larger number of planet realizations for each pericentre and planet distance pair will provide better statistics, allowing a more accurate result. However, it is extremely time-consuming to simulate and process a large number of planet realizations. A suitable balance for these considerations must therefore be met to produce accurate results over a short timescale. Such a balance is achieved by simulating one-hundred planet realizations per parameter set.

To calculate the initial conditions for the planet, we must now consider the combined configuration of Figs. 1 and 2. For such a configuration, the planet's positional coordinates with respect to the SMBH can be calculated according to Eqs. (7), (8) and (9).

$$x_{\text{planet}} = x_{\text{star}} + r_p \cos(\text{i}) \cos(\phi) \tag{7}$$

$$y_{\text{planet}} = y_{\text{star}} + r_p \cos(\text{i}) \sin(\phi) \tag{8}$$

$$z_{\text{planet}} = z_{\text{star}} + r_p \sin(\text{i}) \tag{9}$$

Furthermore, the planet's circular orbit around the star has a velocity governed by Eq. (10)

$$v_{\text{circ}} = \sqrt{\frac{GM}{r}} \tag{10}$$

where M is the total mass of the planet and star, and r is the distance between them [7]. Assuming the planet is initially positioned at its highest (or lowest) orbital elevation from the z-plane (i.e.

$v_{z,\,\text{planet}} = 0$), the initial x and y velocity components of the planet can be calculated using Eqs. (11) and (12).

$$v_{x,\,\text{planet}} = v_{x,\,\text{star}} + v_{\text{circ}} \sin(\phi) \tag{11}$$

$$v_{y,\,\text{planet}} = v_{y,\,\text{star}} - v_{\text{circ}} \cos(\phi) \tag{12}$$

A full description for the generation of initial conditions for a three-body gravitational encounter is thereby reached through Eqs. (2) to (12). However, a systematic bias caused by the phase of the planets orbit still exists within the system. Such a bias must be addressed before simulations using these initial conditions can take place.

2.1.3 Removing a Systematic Bias

For each realization of a planet's orbit, the star and planet start at approximately the same position along their parabolic path. This position will determine the phase of a planets orbit and where the planet is located when the SMBH's gravitational influence is strongest. As a consequence, there is a bias for particular simulation outcomes depending on the size of the planet distance, r_p. To avoid such a bias, the true anomaly, θ, can be randomized such that the x-position of the planet and star varies within a range of $2r_p$. This will effectively randomize the phase of the planet, thereby removing the possibility of this systematic error. Fig. 3 provides a visual illustration for how the true anomaly is randomized assuming a parabolic orbit approaching from infinity.

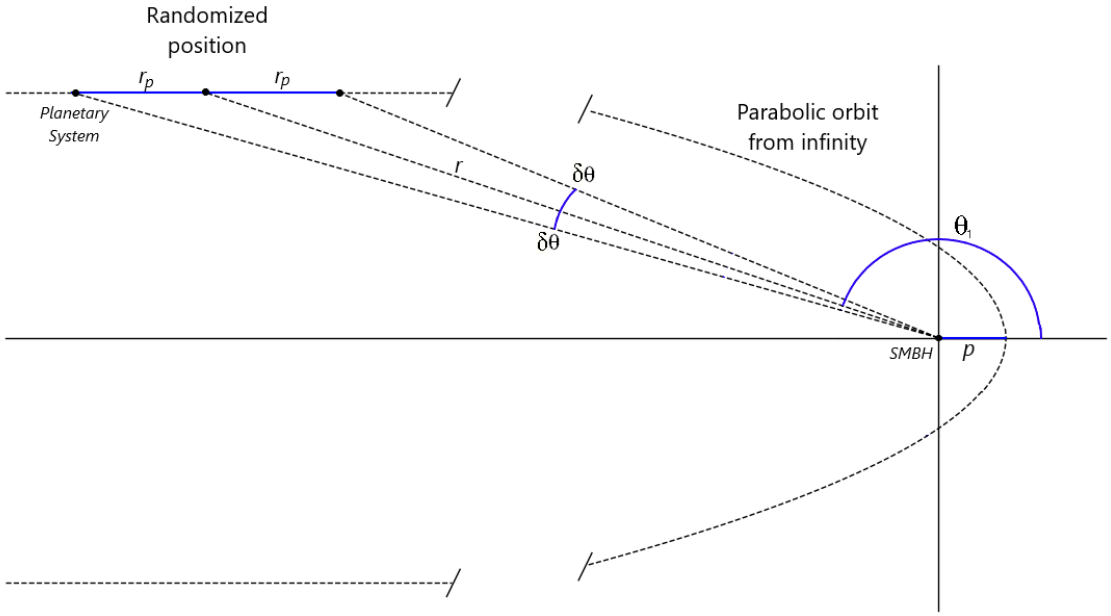

Figure 3: Randomizing the planets phase by randomizing the true anomaly, θ.

Using the arrangement presented in Fig. 3, θ is randomized within the range described by Eq. (13) starting from an arbitrary true anomaly value represented by θ_1.

$$\theta = \theta_1 \pm \delta\theta \tag{13}$$

Here, $\delta\theta$ is derived to be Eq. (14) using Fig. 3, where r represents the distance between the SMBH and the planetary system.

$$\delta\theta = \tan^{-1}\left(\frac{\mathrm{r_p}\sin(\theta_1)}{\mathrm{r} + \mathrm{r_p}\cos(\theta_1)}\right) \tag{14}$$

By using the randomized true anomaly represented in Eq. (13) within Eqs. (2), (3), (5) and (6), the systematic bias caused by the planet's orbital phase is successfully reduced. However, this bias is not completely eliminated due to the finite parabolic approach of the star and planet within the simulations. As a consequence, although the procedure above greatly reduces this systematic error, a slight preference for certain encounter outcomes might still be present.

2.2 Three-Body Gravitational Encounters

The proposed problem requires an important decision, namely what range of values should be used for p and r_p for the simulations in this investigation. Undoubtedly, the most distant planet in our own solar system, Neptune, provides a sensible suggestion of 30 astronomical units (au) for the maximum r_p value [8]. However, the assumption that our own solar system is representative of others is unreasonable. As a consequence, simulations up to a planet distance of 50 au were performed in an attempt to cover a wider range of possibilities. In addition to this, smaller increments in r_p were used at low values as shown in Table 2. This choice anticipates a strong dependence between r_p and the outcome of a simulation for small r_p values.

A less straightforward approach was required when deciding which pericentres to simulate. To facilitate this decision, several 'test' simulations were performed for pericentres ranging between 100 au and 1000 au. These tests revealed that the simulation results for $p > 600$ au remained relatively unchanged. Therefore, the pericentres used for the investigation were varied between 100 au and 600 au as seen in Table 2.

Pericentre (au)	Planet distances (au)	Number of realizations per r_p
100	1, 2, 3, 5, 10, 20, 30, 40, 50	100
200	1, 2, 3, 5, 10, 20, 30, 40, 50	100
300	1, 2, 3, 5, 10, 20, 30, 40, 50	100
400	1, 2, 3, 5, 10, 20, 30, 40, 50	100
500	1, 2, 3, 5, 10, 20, 30, 40, 50	100
600	1, 2, 3, 5, 10, 20, 30, 40, 50	100

Table 2: The parameters used for the three-body simulations.

Using the parameters provided in Table 2, the initial conditions were generated and then used to simulate three-body gravitational encounters using an integrator code provided by my supervisor. This integrator code is called AR-CHAIN and uses a chain regularization technique suitable for small N-body systems [9]. Few-body interactions are generally modelled in astrophysics with the direct summation technique, in which all pairwise gravitational forces are computed explicitly according to Newton's law and then added to obtain the total force acting on each body. For a system consisting of N bodies, N^2 forces need to be computed. This computational complexity is high, but its not a limiting factor in few-body simulations like the ones performed in this project. A potential problem is the divergence of the forces when two bodies come very close, as well as a loss of accuracy for unequal mass objects. The chain scheme employs a particular change of coordinates to avoid singularities in the gravitational forces thereby allowing the very accurate simulation of close encounters, even for extreme mass ratios as in the case of a planet or star and SMBH.

By considering each of the different p and r_p combinations, and the number of realizations for each combination, it can be seen that 5400 of these three-body simulations were performed. For this reason, the results achieved in this investigation should be statistically reliable.

2.3 Four-Body Gravitational Encounters

During the final stage of the investigation, an additional planet was added to the planetary system such that its orbital plane was identical to that of the first. Naturally, this is achieved by using Eqs. (7) to (12) with the same positional angles, ϕ and i, but a different planet distance, r_p, to the first planet. Such an endeavour enables the comparison of results for three and four-body gravitational encounters. For this purpose, the same range of pericentres provided in Table 2 were simulated. However, the planet distances used for these four-body gravitational encounters are provided in Table 3. A total of 3000 four-body gravitational encounters were therefore performed.

Planet A distance (au)	Planet B distance (au)	Number of realizations
1	15	100
2	20	100
3	30	100
5	40	100
10	50	100

Table 3: The planet distances used for the four-body simulations.

2.4 The Classification of Gravitational Encounters

To determine the outcome of a gravitational encounter, the total energy of the planet with respect to the star, and with respect to the SMBH, must be calculated. The total energy of the planet is given by

$$E_{\text{total}} = E_{\text{kin}} + E_{\text{grav}} = \frac{1}{2} m_p v_p^2 - \frac{GMm_p}{r} \tag{15}$$

where m_p is the mass of the planet, M is the mass of the larger body, v_p is the velocity of the planet with respect to the larger body, and r their relative distance magnitude [7]. Such a calculation results in a negative value for two gravitationally bound bodies, and a positive value if the bodies are unbound. Using this characterisation, the outcome of a gravitational encounter can be determined as follows:

- If $E_{\text{total, SMBH}} < 0$, the planet has been captured by the SMBH as seen in Fig. 4.
- If $E_{\text{total, star}} < 0$, the planet has remained in a stable orbit around its star as seen in Appendix A.2.
- If $E_{\text{total, SMBH}} > 0$ and $E_{\text{total, star}} > 0$, the planet has become unbound from both bodies as seen in Appendix A.3.

Within the context of multiple simulated planet realizations using the same pericentre and planet distance, the probability for each of the above outcomes can be calculated as a fraction of the total number of realizations. On this basis, the statistical error is straightforwardly calculated using Poisson statistics to be the square root of an outcomes count, divided by the total number of

realizations simulated. These fractions can then be plotted as a function of the input parameters, thereby revealing the favourable conditions for planet deposition around a SMBH.

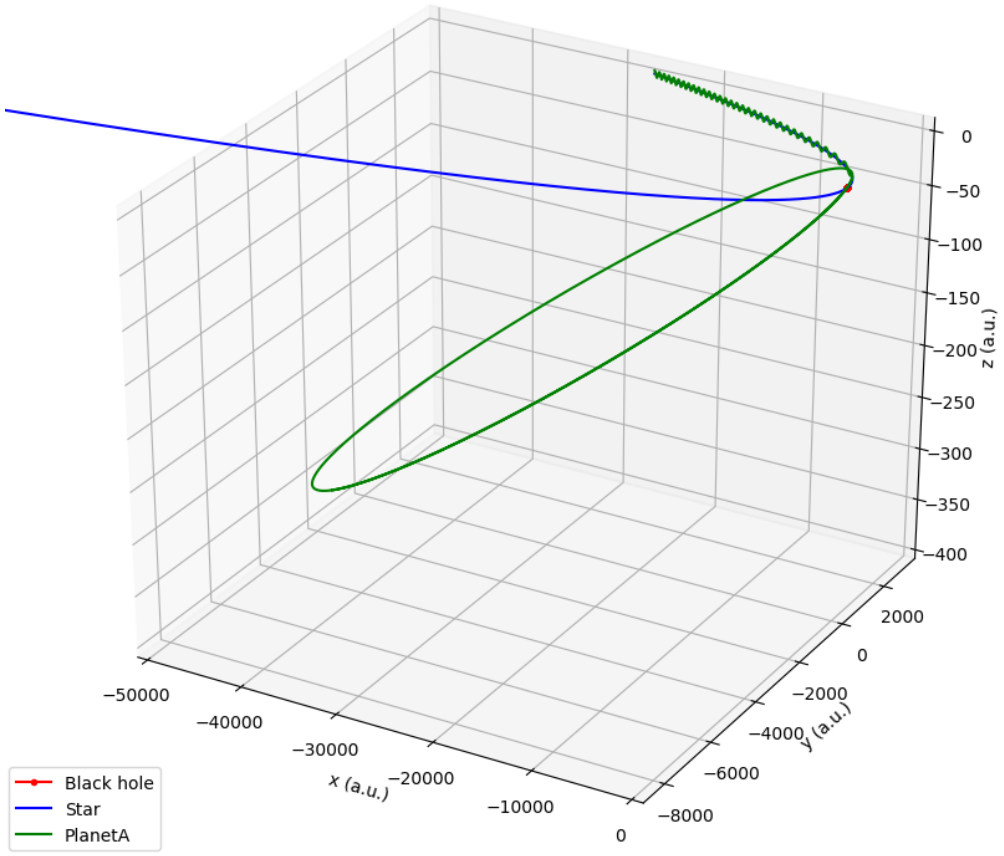

Figure 4: A simulated three-body gravitational encounter where the planet becomes bound to the SMBH. The planet is initially in a randomized circular orbit around the star, but then vacates its star when it reaches the SMBH.

A further consideration of the semi-major axis and eccentricity of successfully captured planets could prove useful for comparison with previous research. These are given by [10].

$$a = \left(\frac{2}{r} - \frac{v^2}{GM}\right)^{-1} \tag{16}$$

$$e = \sqrt{1 - \frac{h^2}{GMa}} \tag{17}$$

Here, r is the distance between the planet and the SMBH, v is the planets relative velocity, and M is the total mass of the two bodies. For Eq. (17), $\mathbf{h} = \mathbf{r} \times \mathbf{v}$ is the angular momentum per unit mass. These quantities are suitably plotted in normalized histogram distributions. The statistical error for a normalised bin of width w which contains N counts out of a total of N_{total} counts is then calculated according to Eq. (18).

$$\Delta_{\text{bin}} = \frac{\sqrt{N}}{wN_{\text{total}}} \tag{18}$$

3 Results

3.1 Three-Body Gravitational Encounter Results

This section summarizes the key results for the three-body gravitational encounter simulations. Fig. 5 shows the fraction of planets remaining bound to the star (left), and the corresponding fraction of planets becoming bound to the SMBH (right) after a three-body gravitational encounter. In the left-hand side contour plot, it can be seen that the planet has approximately a 0% to 15% chance of remaining bound to its star when $r_p > 10$ au. Under the same conditions, the planet has a 40% to 56% (approximate) chance of becoming bound to the SMBH according to the right-hand side contour plot. In addition to this, these plots show that a higher proportion of planets remain bound to the star when $r_p < 10$ au - as is anticipated. Furthermore, it can be seen that 100% of the gravitational encounters result in the star retaining its planet when $r_p < R_H$. This finding is in agreement with the theory behind the Hill radius, provided by Eq. (1), and consequently these results are consistent with expectations.

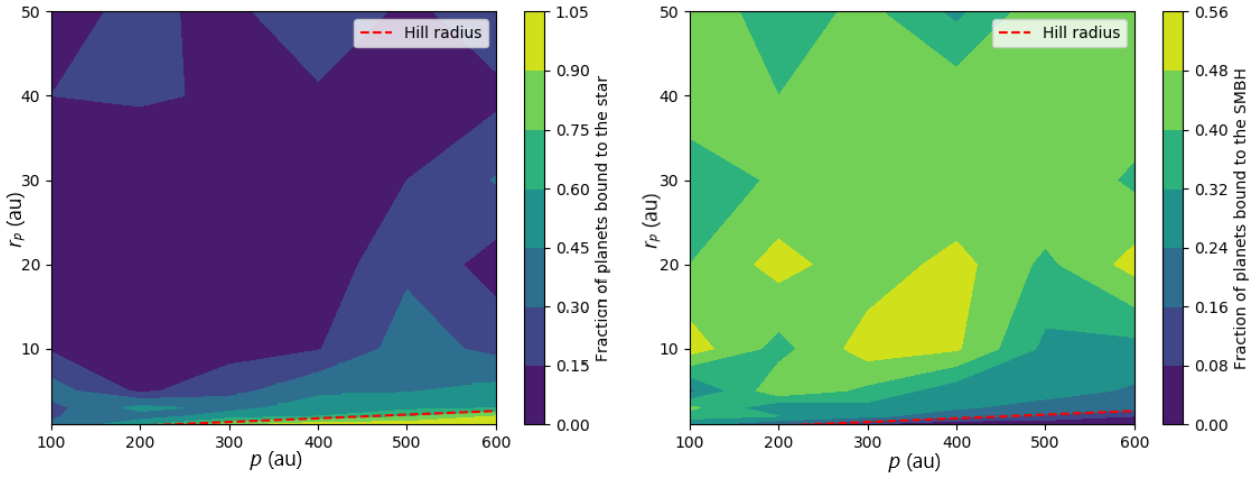

Figure 5: The left-hand side (lhs) contour plot shows the fraction of planets remaining bound to the star as a function of pericentre and planet distance. The right-hand side (rhs) contour plot shows the fraction of planets being bound to the SMBH as a function of pericentre and planet distance. The red dotted lines represent the Hill radius given by Eq. (1).

Fig. 6 displays two 'cross-sections' through these contour plots at a fixed pericentre of 300 au (left), and at a fixed planet distance of 2 au (right). The top left plot shows that the fraction of planets becoming bound to the SMBH increases rapidly between $r_p = 1$ au and $r_p = 10$ au, before remaining approximately constant at around 50% for $r_p > 10$ au. A similar relationship is seen in the bottom left plot. In contrast to this, the middle left plot shows that the fraction of planets remaining bound to the star decreases rapidly between $r_p = 1$ au and $r_p = 10$ au. This behaviour is expected because the planet should be more tightly bound to its star for smaller planet distances. However, the constant fraction for $r_p > 10$ au was an unanticipated result from these simulations. Such a result suggests that the planet will almost always become unbound from its star when r_p is greater than 10 au for the system outlined in Section 2.

The top right plot of Fig. 6 shows that the fraction of planets becoming bound to the SMBH decreases linearly with increasing pericentre. Again, this behaviour is almost identical for the fraction of planets being unbound from both the star and the SMBH. Interestingly, this suggests that these two outcomes are equally probable no matter which initial parameters are used for the simulation. A further look at the middle right plot shows that the fraction of planets remaining bound

to the star increases linearly with increasing pericentre. This relationship is expected because the gravitational influence of the SMBH on the planet is smaller for gravitational encounters involving larger pericentre.

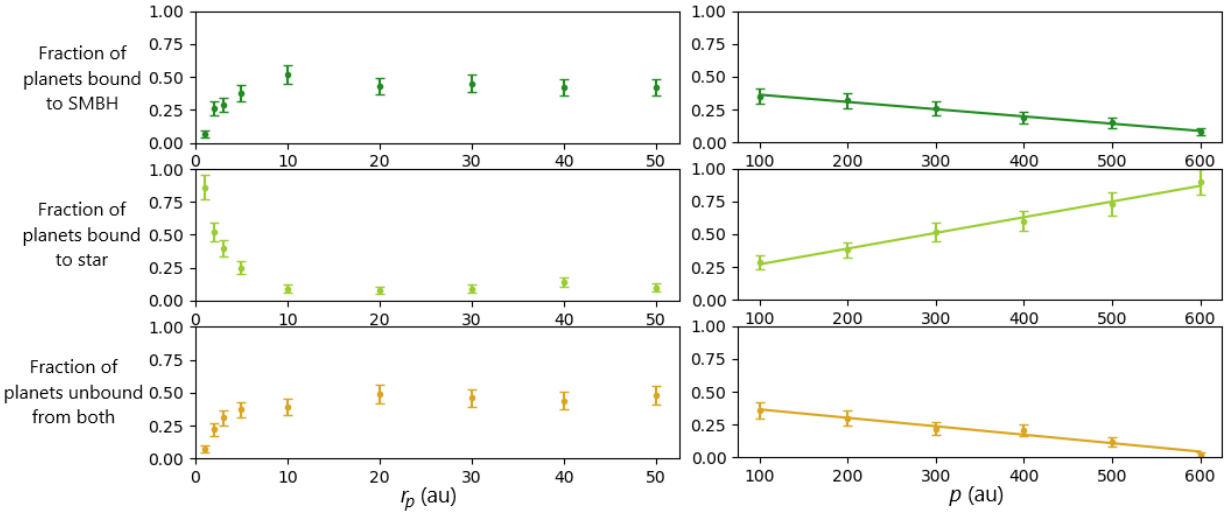

Figure 6: These plots show the fraction of planets being bound to the SMBH (top), the fraction of planets remaining bound to the star (middle), and the fraction of planets becoming unbound from both (bottom) as a function of the initial conditions. The lhs plots are varying in planet distance for a fixed pericentre of 300 au. The rhs plots are varying in pericentre for a fixed planet distance of 2 au.

Fig. 7 provides two normalised histogram plots for the semi-major axis (left) and eccentricity (right) of successfully captured planets around the SMBH in the case of a fixed pericentre of 300 au. The histogram on the left reveals that the semi-major axes are very large in comparison to the pericentre distance, with a strong peak at approximately 0.25×10^6 au. These large semi-major axes are set by the energetics of the system. Meanwhile, the histogram on the right shows a preference for very large eccentricities in excess of 0.99 with a peak somewhere between 0.9975 and 1.0 - a finding which is supported by previous research [11].

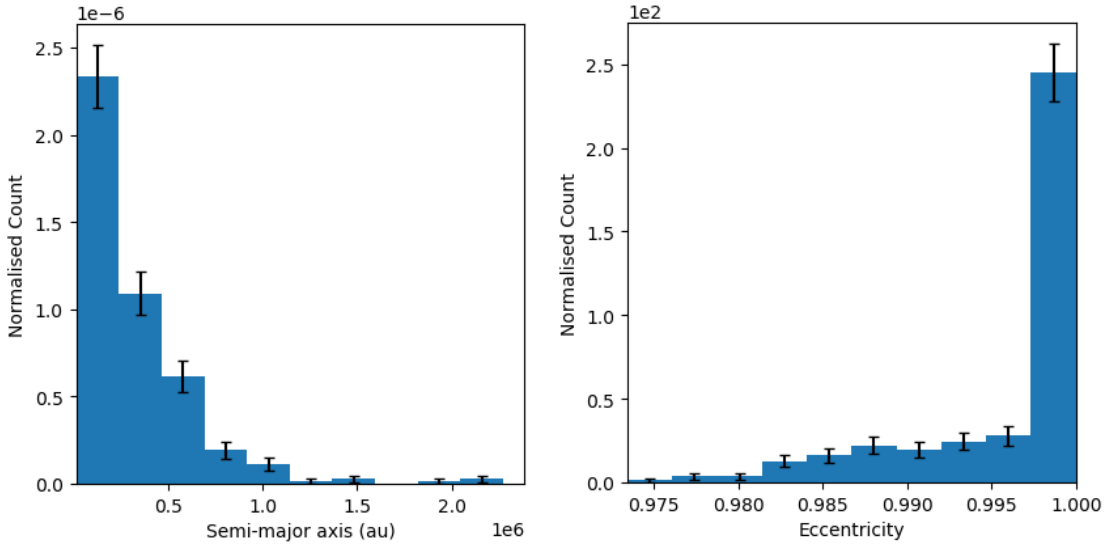

Figure 7: The normalised histogram distributions for the semi-major axis (left) and eccentricity (right) of successfully captured planets around the SMBH for a fixed pericentre of 300 au.

3.2 Four-Body Gravitational Encounter Results

Fig. 8 shows the achieved fractions for each possible outcome of the four-body gravitational encounters for a fixed pericentre of 300 au (left), and for a fixed planet distance of 2 au (right). By comparing these results to the three-body gravitational encounter results in Fig. 6, it can be seen that they are within 2σ of each other. This means we can consider the outcomes of the simulations for three- and four-body gravitational encounters to be identical. Such a finding is justifiable due to the small mass ratio between the planets mass and the SMBH mass. As a consequence, the addition of an extra planet into a planetary system should have little to no effect on whether the planets in the system become bound to the SMBH or remain bound to the star. This conclusion is supported by the histogram distribution in Fig. 9 which is almost identical to the corresponding three-body results in Fig. 7.

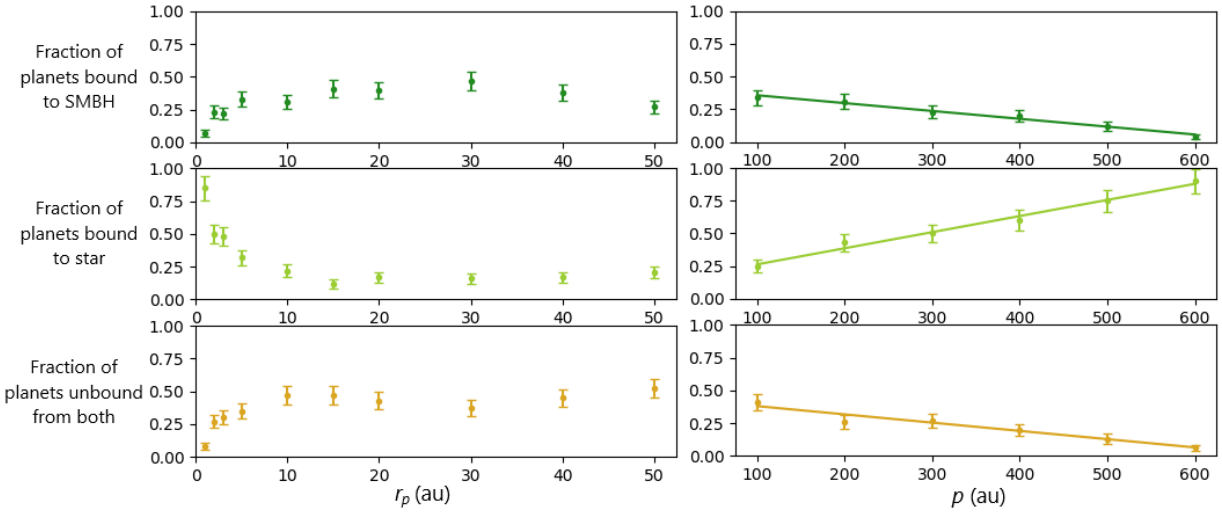

Figure 8: These plots show the fraction of planets being bound to the SMBH (top), the fraction of planets remaining bound to the star (middle), and the fraction of planets becoming unbound from both (bottom) as a function of the initial conditions for a four-body encounter. The lhs plots are varying in planet distance for a fixed pericentre of 300 au. The rhs plots are varying in pericentre for a fixed planet distance of 2 au.

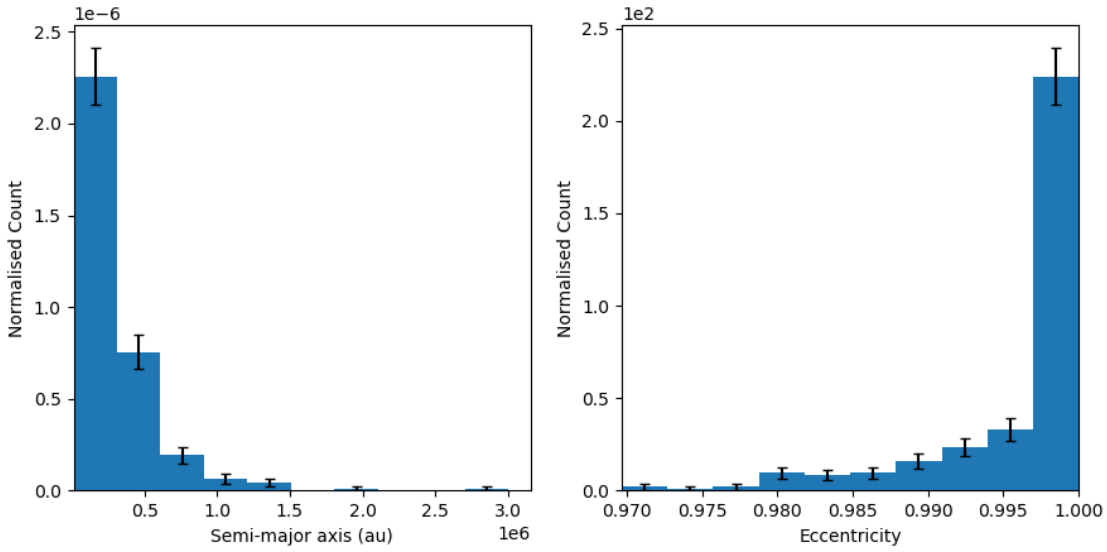

Figure 9: The normalised histogram distributions for the semi-major axis (left) and eccentricity (right) of successfully captured planets around the SMBH for four-body encounters using $p = 300$ au.

4 Discussion

Undoubtedly, several key results in Section 3 were anticipated before the simulations were performed. One such result is that an increase in pericentre for the parabolic orbit will lead to a smaller fraction of planets becoming bound to the SMBH as seen in Fig. 6. Indeed, the same fraction is also expected to be low for small distances between the planet and star. These findings are supported by various theoretical models of the dynamics of spheres of influence within multi-body systems [3, 12]. Furthermore, these results are potentially relevant to recent research which hypothesizes that irregular moons are not formed around their planets, but are instead captured by them, as indicated by their large orbital eccentricities [13]. Such a theory requires the investigation of a system of bodies which is comparable in some respects to our own.

Several unanticipated results were also achieved during this investigation. Notably, the fractions of planets becoming bound to the SMBH look almost identical to the fractions of planets becoming unbound from both bodies. This suggests that these two outcomes are equally probable for all simulation parameter combinations. Such a result could be coincidental, however it is undeniable that a further investigation could be worthwhile.

Assuredly, another unforeseen finding from this investigation is the relatively unchanged fractions when simulating across a fixed pericentre for $r_p > 10$ au as seen in Fig. 6. An explanation for such a result is unclear, however it might occur because the planet has a maximum distance from its star of 10 au at which it can no-longer remain bound. Therefore, there are only two possible outcomes for these gravitational encounters when $r_p > 10$ au. As a consequence, this finding is perhaps not completely inconceivable, however the similarity of the fractions for the two remaining simulation outcomes is again a remarkable feature.

A further key result of this investigation was the equivalence of the three- and four-body gravitational encounter outcomes when comparing Figs. 6 and 8. As previously mentioned, such a finding is anticipated due to the small mass ratio between the planets mass and the SMBH mass. On this basis, the addition of an extra planet into a planetary system should have little to no effect on whether the planets in the system become bound to the SMBH or remain bound to the star. Interestingly, this result implies that a planetary system can lose any number of its planets during a gravitational encounter based solely on the relative position and velocity of each of its planets with respect to the SMBH. In light of this, our own solar system would have a $93 \pm 3\%$ chance of losing at least one of its planets to a SMBH capture event for a pericentre distance of 300 au, as calculated by the python script listed in Appendix A.4.

Arguably, a more significant result is seen in the eccentricity histogram distributions of Figs. 7 and 9. These distributions are in good agreement with previous research as they show that large eccentricities are achieved for the orbits of captured planets [11]. Moreover, these high eccentricites provide an explanation for the large orbital eccentricites of irregular moons currently under investigation [2, 13]. Naturally, these large eccentricities can be reduced over time via gravitational relaxation. On this basis, gravitationally bound bodies with less-eccentric orbits may have also been captured by their host before undergoing relaxation over a long time period. This suggests that gravitational capture in N-body systems is an extremely important research area as it could play a significant role in the formation of planetary systems.

Within the context of a gravitational encounter involving a SMBH, the relevance of this work is again highlighted. Notably in 2015, a powerful X-ray flare emanating from the centre of the Milky Way was detected by the Chandra X-ray observatory [14]. This flare of electromagnetic radiation is thought to emerge from a SMBH after a tidal disruption event. Such an event involves the pulling-apart of a large body, such as a planet which is sufficiently close to the SMBH, followed by the capture of some of its mass into an accretion disk. Indeed, this event results in a temporary

flare of electromagnetic radiation which can be used to calculate the mass of the SMBH [15]. It is therefore important to understand the conditions under which a planet might be captured by a SMBH, and thereby be used to 'fuel' these powerful flares. With this in mind, the right panel of Fig. 5 is used to approximate the most favourable conditions for planetary capture by a SMBH to be within $300 \leq p \leq 400$ au, and $10 \leq r_p \leq 20$ au.

A discussion of the perceived shortcomings of the investigation now becomes necessary in order to ascertain the accuracy of the results in this report. The main concern during this investigation was a systematic bias towards specific gravitational encounter outcomes caused by the phase of a planets orbit. Such a bias was reduced by randomizing the true anomaly so that the x-position of the planet and star varied within a range of $2r_p$. This procedure assumed a parabolic orbit which approached from infinity, something which cannot be implemented within a simulation. Therefore, a slight systematic error caused by the planets orbital phase is still present within the simulated gravitational encounters.

A lesser consideration was given to the initial direction of the planets orbital velocity in the simulations. This velocity was consistently calculated such that a given ϕ would provide a circular velocity for the planet which is always in the same direction. Such a procedure neglects any thought into how reversing its direction could affect the outcome of an encounter. Although the impact of this assumption on the simulation results is unknown, a further investigation might benefit from randomizing the direction of this velocity.

A final thought is given to the circular orbits of the planets within the simulations. Assuredly, these circular orbits are unrealistic within observed planetary systems such as our own solar system. Perhaps a more suitable representation of a planetary system would therefore use elliptical orbits instead. This improvement would be implemented such that the ellipticity of a planets orbit is sampled from a distribution of observed elliptical parameters for planetary systems. The outcome of each gravitational encounter could then be classified as a function of the semi-major axis, or eccentricity, of the planets initial orbit. Such an endeavour avoids an over-simplification of this SMBH gravitational encounter scenario.

5 Summary and Conclusions

In summary, the initial conditions for a large number of three and four-body gravitational encounters were generated and then simulated using an AR-CHAIN integrator code. The gravitational outcomes were then classified as a function of the initial parameters. From the results it is seen that the three and four-body gravitational encounters are analogous. The favourable conditions for the successful capture of a planet by a SMBH are then found to be approximately $300 \leq p \leq 400$ au, and $10 \leq r_p \leq 20$ au. For this scenario, very large eccentricities with a peak between 0.9975 and 1 were achieved for the captured planets - which is supported by previous research [11]. In addition to this, the probability of a planet becoming unbound from both bodies is approximately equal to the SMBH capture probability for all initial parameter combinations.

Although the equivalence of these outcome probabilities could be coincidental, a further investigation is desirable. Such an investigation would benefit from randomizing the initial planet velocity direction. Furthermore, a more realistic set of simulations could be performed by using elliptical orbits instead of circular orbits for the planets. This improvement would mean the gravitational encounters are classified as a function of the semi-major axis, or eccentricity, of the planet's initial orbit instead of the planet distance, r_p, used in this investigation. A similar planetary disruption study could also investigate the frequency of planet deposition around a SMBH given a rate of stellar passages. Such a study is encouraged as it could provide interesting results which relate to the rate of flare events observed at the centre of the Milky Way.

References

[1] Kormendy, J. & Richstone, D. (1995) *Inward Bound - The Search for Supermassive Black Holes in Galactic Nuclei*. Annual Review of Astronomy and Astrophysics. 33(581).

[2] Astakhov, S., Burbanks, A., Wiggins, S. & Farrelly, D. (2003) *Chaos-assisted capture of irregular moons*. Nature. 423(6937), pp. 264-267.

[3] Araujo, R., Winter, O., Prado, A. & Martins, R. (2008) *Sphere of influence and gravitational capture radius: a dynamical approach*. Monthly Notices of the Royal Astronomical Society. 391(2), pp. 675-684.

[4] Hamilton, D. P. & Burns, J. A. (1992) *Orbital stability zones about asteroids. II - The destabilizing effects of eccentric orbits and of solar radiation*. Icarus. 96(1), pp. 43-64.

[5] Abuter, R. et al. (2019) *A geometric distance measurement of the Galactic centre black hole with 0.3% uncertainty*. Astronomy & Astrophysics. 10(625), p. 3.

[6] Jacobson, R. A., Haw, R. J., McElrath, T. P. & Antreasian, P. G. (2000) *A Comprehensive Orbit Reconstruction for the Galileo Prime Mission in the J2000 System*. Journal of the Astronautical Sciences. 48(4), pp. 495–516.

[7] Curtis, H. D. (2014) *Orbital Mechanics for Engineering Students (Third Edition)*. Butterworth-Heinemann. pp. 59-144.

[8] Stansberry, J. et al. (2007) *Physical Properties of Kuiper Belt and Centaur Objects: Constraints from Spitzer Space Telescope*. Astronomy & Astrophysics. p. 165.

[9] Mikkola, S. & Aarseth, S. J. (1989) *A chain regularization method for the few-body problem*. Celestial Mechanics and Dynamical Astronomy. 47, pp. 375-390.

[10] Kemble, S. (2006) *Interplanetary Mission Analysis and Design*. Springer-Verlag Berlin Heidelberg. pp. 135-329.

[11] Perets, H. B., Gualandris, A., Kupi, G., Merritt, D. & Alexander, T. (2009) *Dynamical Evolution of the Young Stars in the Galactic Center: N-body Simulations of the S-Stars*. The Astrophysical Journal. 702(2), pp. 884-889.

[12] Bate, R., Donald, D. & Mueller, J. E. (1971) *Fundamentals of Astrodynamics*. New York: Dover Publications. pp. 333–334.

[13] Nesvorny, D., Vokrouhlicky, D. & Morbidelli, A. (2007) *Capture of Irregular Satellites During Planetary Encounters*. The Astronomical Journal. pp. 1962-1975.

[14] Grant, C. E. (2019) *Twenty years of the Advanced CCD Imaging Spectrometer on the Chandra X-ray Observatory*. American Astronomical Society. 153(13).

[15] Velsen, S. V., Mendez, A. J., Krolik, J. H. & Gorjian, V. (2016) *Discovery of transient infrared emission from dust heated by stellar tidal disruption flares*. The Astrophysical Journal. 829(1).

A Appendices

A.1 The C++ Script used for Generating the Initial Conditions of an Encounter

```
#include "GenerateInitFiles.h"

#include "Body.h"
#include "BodyCreator.h"
#include "InitSimulationParams.h"
#include "XYZComponents.h"
#include "FileManager.h"
#include "TaskRunner.h"

#define _USE_MATH_DEFINES
#include <math.h>

namespace {
using namespace BodyCreator;

std::size_t randomNumber(std::size_t lower, std::size_t higher) {
  return static_cast<std::size_t>(rand() % (higher - lower) + lower);
}

double randomDouble(double lower, double higher) {
  return (double)rand() / RAND_MAX * (higher - lower) + lower;
}

std::string formatSimParameter(double value) {
  return std::to_string(static_cast<int>(value)) + ".0";
}

void generateFileLine(std::string &fileText, double mass,
                      XYZComponents const &position,
                      XYZComponents const &velocity) {
  fileText += "\n " + std::to_string(mass) + " " +
              std::to_string(position.compX()) + " " +
              std::to_string(position.compY()) + " " +
              std::to_string(position.compZ()) + " " +
              std::to_string(velocity.compX()) + " " +
              std::to_string(velocity.compY()) + " " +
              std::to_string(velocity.compZ());
}

void generateFileText(std::string &fileText) { (void)(fileText); }

template <typename Body, typename... Bodies>
void generateFileText(std::string &fileText, Body const &body,
                      Bodies const &... bodies) {
  generateFileLine(fileText, body.mass(), body.position(0), body.velocity(0));
  generateFileText(fileText, bodies...);
}

} // namespace
```

```
InitFileGenerator::InitFileGenerator(std::string const &directory)
    : m_fileManager(std::make_unique<FileManager>()),
      m_directory(directory),
      m_taskRunner(TaskRunner::getInstance()) {}

InitFileGenerator::~InitFileGenerator() {}

void InitFileGenerator::resetGenerator(std::size_t numberOfInitFiles) {
  resetInitSimulationParams(numberOfInitFiles);
  m_taskRunner.setTask("Generating init files...", 0.0, 10.0);
  m_taskRunner.setNumberOfSteps(numberOfInitFiles);
}

void InitFileGenerator::resetInitSimulationParams(std::size_t numberOfInitFiles) {
  m_simulationParams.clear();
  m_simulationParams.reserve(numberOfInitFiles);
}

void InitFileGenerator::addInitSimulationParams(
     InitSimulationParams const &simParameters) {
  m_simulationParams.emplace_back(simParameters);
}

std::vector<InitSimulationParams> const &
InitFileGenerator::simulationParameters() const {
  return m_simulationParams;
}

void InitFileGenerator::createFile(std::string const &filename,
                                   std::string const &fileText) const {
  m_fileManager->setFilename(m_directory + filename);
  m_fileManager->createNewFile(fileText);
}

bool InitFileGenerator::generate(
    std::vector<std::string> const &pericentres,
    std::vector<std::string> const &planetDistancesA,
    std::vector<std::string> const &planetDistancesB,
    std::size_t numberOfOrientations) {
  resetGenerator(pericentres.size() * planetDistancesA.size() * numberOfOrientations);
  return generateInitFiles(pericentres, planetDistancesA, planetDistancesB,
                           numberOfOrientations);
}

void InitFileGenerator::generate3BodyInitFiles(
    std::string const &pericentre, std::string const &planetDistance,
    std::size_t numberOfOrientations) {
  for (auto index = 1u; index < numberOfOrientations + 1; ++index) {
    generate3BodyInitFile(pericentre, planetDistance, index);
    m_taskRunner.reportProgress();
  }
}
```

```
bool InitFileGenerator::generateInitFiles(
    std::vector<std::string> const &pericentres,
    std::vector<std::string> const &planetDistancesA,
    std::vector<std::string> const &planetDistancesB,
    std::size_t numberOfOrientations) {
  for (auto const &pericentre : pericentres)
    for (auto i = 0u; i < planetDistancesA.size(); ++i) {
      if (m_taskRunner.isRunning())
        generateInitFiles(pericentre, planetDistancesA, planetDistancesB, i,
                          numberOfOrientations);
      else
        return false;
    }

  saveSimulationParameters(m_simulationParams);
  return true;
}

void InitFileGenerator::generateInitFiles(
    std::string const &pericentre,
    std::vector<std::string> const &planetDistancesA,
    std::vector<std::string> const &planetDistancesB,
    std::size_t planetDistanceIndex,
    std::size_t numberOfOrientations) {
  if (OtherSimulationSettings::m_hasSinglePlanet) {
    generate3BodyInitFiles(pericentre, planetDistancesA[planetDistanceIndex],
                           numberOfOrientations);
  } else {
    generate4BodyInitFiles(pericentre, planetDistancesA[planetDistanceIndex],
                           planetDistancesB[planetDistanceIndex],
                           numberOfOrientations);
  }
}

void InitFileGenerator::generate3BodyInitFile(std::string const &pericentre,
                                              std::string const &planetDistance,
                                              std::size_t orientationIndex) {
  auto const phi = randomNumber(0, 360);
  auto const inclination = randomNumber(0, 360);
  generate3BodyInitFile(pericentre, planetDistance, orientationIndex, phi,
                        inclination);
}

void InitFileGenerator::generate3BodyInitFile(std::string const &pericentre,
                                              std::string const &planetDistance,
                                              std::size_t orientationIndex,
                                              std::size_t phi,
                                              std::size_t inclination) {
  generate3BodyInitFile(
      generate3BodyInitFilename(pericentre, planetDistance, orientationIndex),
      std::stod(pericentre), std::stod(planetDistance), orientationIndex, phi,
      inclination);
}
```

```
void InitFileGenerator::generate3BodyInitFile(
    std::string const &filename, double pericentre, double planetDistance,
    std::size_t orientationIndex, std::size_t phi, std::size_t inclination) {
  auto const star =
      createStar(pericentre, randomizeTrueAnomaly(pericentre, planetDistance));
  auto const planet =
      createPlanet(*star, planetDistance, orientationIndex, phi, inclination);

  auto fileText = generateInitHeader(filename, pericentre, planetDistance);
  generateFileText(fileText, *blackHole(), *star, *planet);

  createFile(filename + ".init", fileText);

  addInitSimulationParams(InitSimulationParams(filename, pericentre,
                                               planetDistance, orientationIndex,
                                               phi, inclination));
}

void InitFileGenerator::generate4BodyInitFiles(
    std::string const &pericentre, std::string const &planetDistanceA,
    std::string const &planetDistanceB, std::size_t numberOfOrientations) {
  for (auto index = 1u; index < numberOfOrientations + 1; ++index) {
    generate4BodyInitFile(pericentre, planetDistanceA, planetDistanceB, index);
    m_taskRunner.reportProgress();
  }
}

void InitFileGenerator::generate4BodyInitFile(
    std::string const &pericentre, std::string const &planetDistanceA,
    std::string const &planetDistanceB, std::size_t orientationIndex) {
  auto const phi = randomNumber(0, 360);
  auto const inclination = randomNumber(0, 360);
  generate4BodyInitFile(pericentre, planetDistanceA, planetDistanceB,
                        orientationIndex, phi, inclination);
}

void InitFileGenerator::generate4BodyInitFile(
    std::string const &pericentre, std::string const &planetDistanceA,
    std::string const &planetDistanceB, std::size_t orientationIndex,
    std::size_t phi, std::size_t inclination) {
  generate4BodyInitFile(
      generate4BodyInitFilename(pericentre, planetDistanceA, planetDistanceB,
                                orientationIndex),
      std::stod(pericentre), std::stod(planetDistanceA),
      std::stod(planetDistanceB), orientationIndex, phi, inclination);
}

std::string InitFileGenerator::generate3BodyInitFilename(
    std::string const &pericentre, std::string const &planetDistance,
    std::size_t const &orientationIndex) const {
  return "p" + pericentre + "_r" + planetDistance + "_o" +
         std::to_string(orientationIndex);
}
```

```
void InitFileGenerator::generate4BodyInitFile(
    std::string const &filename, double pericentre, double planetDistanceA,
    double planetDistanceB, std::size_t orientationIndex, std::size_t phi,
    std::size_t inclination) {
  auto const largestPlanetDistance =
      planetDistanceA > planetDistanceB ? planetDistanceA : planetDistanceB;

  auto const star = createStar(
      pericentre, randomizeTrueAnomaly(pericentre, largestPlanetDistance));
  auto const planetA =
      createPlanet(*star, planetDistanceA, orientationIndex, phi, inclination);
  auto const planetB =
      createPlanet(*star, planetDistanceB, orientationIndex, phi, inclination);

  auto fileText =
      generateInitHeader(filename, pericentre, largestPlanetDistance);
  generateFileText(fileText, *blackHole(), *star, *planetA, *planetB);

  createFile(filename + ".init", fileText);

  addInitSimulationParams(InitSimulationParams(
      filename, pericentre, planetDistanceA, planetDistanceB, orientationIndex,
      phi, inclination));
}

std::string InitFileGenerator::generate4BodyInitFilename(
    std::string const &pericentre, std::string const &planetDistanceA,
    std::string const &planetDistanceB,
    std::size_t const &orientationIndex) const {
  return "p" + pericentre + "_r" + planetDistanceA + "+" + planetDistanceB +
         "_o" + std::to_string(orientationIndex);
}

std::string InitFileGenerator::generateInitHeader(std::string const &filename,
                                                  double pericentre,
                                                  double planetDistance) const {
  auto const numBodies = std::to_string(InitHeaderData::numberOfBodies());
  auto const step =
      std::to_string(InitHeaderData::timeStep(pericentre, planetDistance));

  auto const numTimeSteps = std::to_string(
      InitHeaderData::numberOfTimeStep(pericentre, planetDistance));

  return "-1 " + numBodies + " " + step + " " + numTimeSteps +
         " 0.000000 0.000000 1.d0 1.d-3 0.d0 0 " + filename + ".out 1 1";
}

void InitFileGenerator::saveSimulationParameters(
    std::vector<InitSimulationParams> const &parameters) const {
  createFile("simulation_parameters.txt",
             generateSimulationParametersText(parameters));
}
```

```
std::string InitFileGenerator::generateSimulationParametersText(
    std::vector<InitSimulationParams> const &parameters) const {
  std::string fileText = generateSimulationParametersHeader(parameters);
  for (auto const &parameterSet : parameters)
    fileText += generateSimulationParametersLine(parameterSet);
  return std::move(fileText);
}

std::string InitFileGenerator::generateSimulationParametersHeader(
    std::vector<InitSimulationParams> const &parameters) const {
  if (OtherSimulationSettings::m_hasSinglePlanet)
    return "Index Pericentre PlanetDistance Phi Inclination";
  return "Index Pericentre PlanetDistanceA PlanetDistanceB Phi "
         "Inclination";
}

std::string InitFileGenerator::generateSimulationParametersLine(
    InitSimulationParams const &parameters) const {
  return "\n" + std::to_string(parameters.m_orientationIndex) + " " +
         formatSimParameter(parameters.m_pericentre) + " " +
         generateSimulationPlanetDistancesSubLine(parameters) + " " +
         std::to_string(parameters.m_phi) + " " +
         std::to_string(parameters.m_inclination);
}

std::string InitFileGenerator::generateSimulationPlanetDistancesSubLine(
    InitSimulationParams const &parameters) const {
  auto const subLine = formatSimParameter(parameters.m_planetDistances[0]);
  if (OtherSimulationSettings::m_hasSinglePlanet)
    return std::move(subLine);
  return subLine + " " + formatSimParameter(parameters.m_planetDistances[1]);
}

double InitFileGenerator::randomizeTrueAnomaly(double pericentre,
                                               double planetDistance) const {
  auto const trueAnomaly =
      InitHeaderData::trueAnomaly(pericentre, planetDistance);

  auto const x = 2.0 * pericentre * cos(trueAnomaly) / (1 + cos(trueAnomaly));
  auto const y = 2.0 * pericentre * sin(trueAnomaly) / (1 + cos(trueAnomaly));
  auto const xyz = XYZComponents(std::move(x), std::move(y), 0.0);

  auto const deltaAnomaly =
      atan(planetDistance * sin(M_PI - trueAnomaly) /
           (xyz.magnitude() - planetDistance * cos(M_PI - trueAnomaly)));

  return randomDouble(trueAnomaly - deltaAnomaly, trueAnomaly + deltaAnomaly);
}
```

A.2 A Simulation where the Planet Remains Bound to its Star

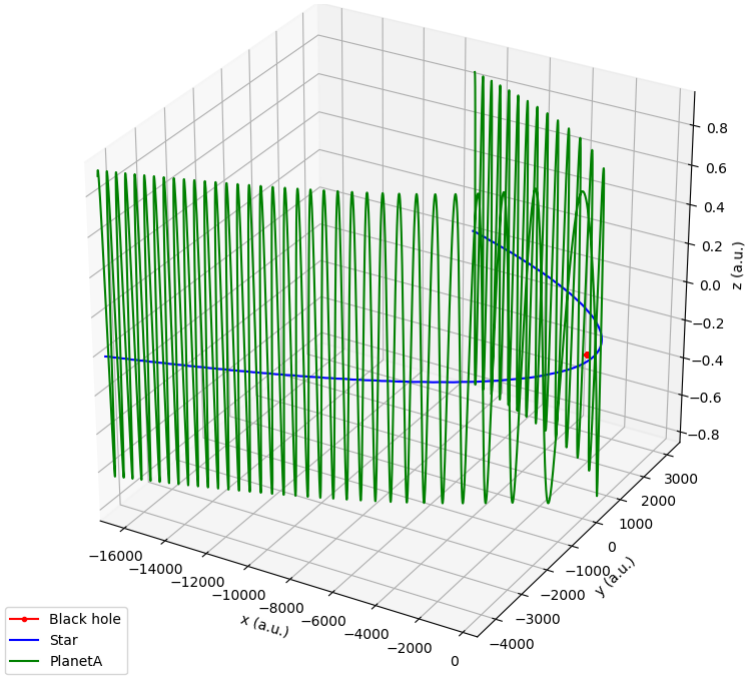

Figure 10: A simulated three-body gravitational encounter where the planet remains bound to its star.

A.3 A Simulation where the Planet Becomes Completely Unbound

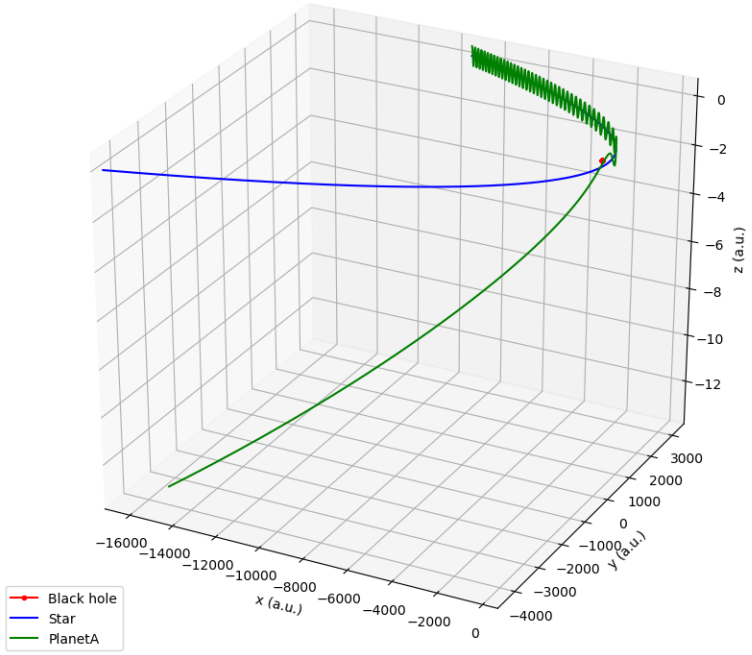

Figure 11: A simulated three-body gravitational encounter where the planet becomes completely unbound.

A.4 The Python Script used for a Probability Calculation

The following python script calculates the probability that at least one planet in our solar system becomes bound to the SMBH when using a pericentre of 300 au.

```
from itertools import combinations
import numpy as np
import dataReader as dr

star_bound_fractions, star_bound_uncertainties = dr.read_fractions("star-bound", p=300)
unbound_fractions, unbound_uncertainties = dr.read_fractions("unbound", p=300)


# Collects the probabilities for a potential gravitational outcome
def determine_an_outcome(planets, planet_combination, possibility1, possibility2):
    potential_outcome = {}

    for planet in planet_combination:
        potential_outcome[planet] = possibility1[planet]

    for planet in planets:
        if planet not in potential_outcome:
            potential_outcome[planet] = possibility2[planet]

    return potential_outcome


# Find the probabilities for a specific gravitational outcome
def determine_potential_outcomes(planets, planet_combinations):
    return [determine_an_outcome(planets, combination, star_bound_fractions,
            unbound_fractions) for combination in planet_combinations]


# Find the uncertainties for a specific gravitational outcome
def determine_potential_uncertainties(planets, planet_combinations):
    return [determine_an_outcome(planets, combination, star_bound_uncertainties,
            unbound_uncertainties) for combination in planet_combinations]


# Determine the different gravitational outcome possibilities
def determine_planet_combinations(planets):
    planet_combinations = []
    for n in range(len(planets) + 1):
        planet_combinations.extend(list(combinations(planets, n)))
    return planet_combinations


# Calculate the probability that a specific outcome will occur
def calculate_outcome_probability(planets, outcome):
    outcome_probability = 1
    for planet in planets:
        outcome_probability *= outcome[planet]
    return outcome_probability
```

```
# Calculate the uncertainty that a specific outcome will occur
def calculate_outcome_uncertainties(planets, outcome, uncertainties):
    uncertainty = 0
    for i in planets:
        res = uncertainties[i]**2
        for j in planets:
            if j != i:
                res *= outcome[j]**2
        uncertainty += res
    return np.sqrt(uncertainty)


# Calculate the probability that at least one planet will be captured by the SMBH
def calculate_probability_for_at_least_one_capture(planets):
    planet_combinations = determine_planet_combinations(planets)

    potential_outcomes = determine_potential_outcomes(planets, planet_combinations)
    uncertainties = determine_potential_uncertainties(planets, planet_combinations)

    outcome_probabilities = [calculate_outcome_probability(planets, outcome)
                             for outcome in potential_outcomes]
    outcome_uncertainties = [calculate_outcome_uncertainties(planets, outcome,
                              uncertainty) for outcome, uncertainty in
                              zip(potential_outcomes, uncertainties)]

    return (1.0 - sum(outcome_probabilities)), sum(outcome_uncertainties)


planet_keys = star_bound_fractions.keys()
probability, uncertainty = calculate_probability_for_at_least_one_capture(planet_keys)

print("{0:.2f} +/- {1:.2f}%".format(probability, uncertainty))
```