# Simulation of gravitational scatterings in $N$-body Kepler systems

by

**Jort Koks**

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Wednesday June 28, 2023.


| | |
|---|---|
| Student number: | 5121396 |
| Project duration: | February, 2023 - June, 2023 |
| Thesis committee: | dr. P.M. Visser (supervisor) |
| | prof. dr. J.M. Thijssen (supervisor) |
| | prof. dr. ir. M.B. van Gijzen |
| | prof. dr. A. F. Otte |


An electronic version of this thesis is available at
`http://repository.tudelft.nl/`.

TU Delft

# Abstract

This research investigates the impact of gravitational scatterings caused by close encounters between particles in an $N$-body Kepler system, addressing three main questions: (1) the influence of scatterings on system evolution, (2) the correspondence between simulated and expected average times between scatterings, and (3) the effect of increasing different parameters individually on the average scattering time. Simulations demonstrate an average scattering angle of 15.2 ° for particles involved in the top 10 % of scatterings. This would indicate a non-negligible impact of gravitational scatterings, especially for systems with heavier bodies. The results indicate that the simulated average time between scatterings is higher than the expected average, necessitating further research for accurate estimation. Moreover, the time between scatterings $T_{scatt}$ decreases over time, before reaching a stationary state after roughly 300 scatterings. On this domain, the correlation coefficient between $T_{scatt}$ and the scattering counter $n_{scatt}$ was found to be $\rho = -0.08$. By varying the test domains for different parameters, a new expression for the expected time between two scatterings is proposed based on simulation data. A clear connection was found between the scattering time and the number of particles $N$, the maximum orbital radius $a_{max}$, and the maximum inclination angle $I_{max}$. The study acknowledges limitations, including the non-stationary initialization state and linear approximations to most computations, suggesting avenues for future improvement. Overall, this research aims to find the role of gravitational scatterings in Kepler systems and underscores the need to consider these interactions, which are now often considered to be negligible.

# Constants and symbols

In this report, a vector $\boldsymbol{v}$ will be indicated in bold letters. The length $\|\boldsymbol{v}\|$ of a vector will be denoted as $v$. A list of constants is given in Table 1. In all computations, the astronomical system is used to denote physical quantities. This way the accuracy of floating point numbers can be improved. A list of symbols is included in Table 2.

| Symbol | Definition | SI units | Astronomical system |
|---|---|---|---|
| AU | astronomical unit | $1.495978707 \cdot 10^{11}$ m | 1 |
| $M_\odot$ | Solar mass | $1.9891 \cdot 10^{30}$ kg | 1 |
| y | year | 31536000 s | 1 |
| G | Newton's constant | $6.6743 \cdot 10^{-11} m^3 kg^{-1} s^{-2}$ | 39.48 $AU^3 M_\odot^{-1} y^{-2}$ |
| S | Solar radius | $6.957 \cdot 10^8$ m | 0.00465047 AU |
| $m_J$ | Jupiter mass | $1.89813 \cdot 10^{27}$ kg | 0.0009543 $M_\odot$ |
| $s_J$ | Jupiter radius | $6.9911 \cdot 10^7$ m | 0.000467 AU |

**Table 1:** List of constants

| Symbol | Quantity |
|---|---|
| $t$ | time |
| $x$, $y$, $z$ | Cartesian coordinates |
| $V$ | volume |
| $\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ | position vector |
| $\boldsymbol{v} = \dot{\boldsymbol{r}}$ | velocity vector |
| $\boldsymbol{w} = \boldsymbol{v}_1 \times \boldsymbol{v}_2$ | |
| $\boldsymbol{d} = \boldsymbol{r}_2 - \boldsymbol{r}_1$ | difference position |
| $\boldsymbol{u} = \boldsymbol{v}_2 - \boldsymbol{v}_1$ | difference velocity |
| $\boldsymbol{R}$ | centre-of-mass position |
| $\boldsymbol{V}$ | centre-of-mass velocity |
| $m$ | particle mass |
| $s$ | particle radius |
| $s^{inf}$ | gravitational sphere of influence |
| $A$ | effective cross section |
| $t^0$ | particle creation time |
| $\omega$ | angular frequency |
| $T = 2\pi/\omega$ | orbit time |
| $T_{scatt}$ | time between consecutive scatterings |
| $a$, $b$ | semimajor-, semiminor axis |
| $c = a\epsilon$ | semi-focal separation |
| $\ell = b^2/a$ | semi-latus rectum |
| $\boldsymbol{r}^0$ | particle creation point |
| $\boldsymbol{L}$ | orbital angular momentum |
| $\boldsymbol{K} = \boldsymbol{L}_1 \times \boldsymbol{L}_2$ | direction of nodal line |
| $\boldsymbol{\epsilon}$ | eccentricity vector |
| $\varpi$ | argument of periapsis |
| $\Omega$ | ascending node |
| $I$ | inclination |
| $\nu$ | true anomaly |
| $E$ | eccentric anomaly |
| $M$ | mean anomaly |
| $\mathscr{R}$ | rotation matrix |
| $\delta$ | scattering angle in CM-frame |
| $N$ | number of particles |
| $i$, $j$ | particle indices |
| $n_{scatt}$ | scattering counter |

**Table 2:** List of symbols

# Contents

# 1. Introduction

A Kepler system of celestial bodies refers to a system of objects in space that follows the laws of planetary motion derived by the astronomer Johannes Kepler in the early 17th century. These laws describe the motion of objects under the influence of gravity. A Kepler system can include various celestial bodies such as planets, moons, asteroids, and comets, all of which interact with each other through the force of gravity. By understanding the Keplerian dynamics of a system, astronomers can predict the motion and behaviour of these celestial bodies within that system.

The conventional approach to studying planetary motion within a Kepler system typically assumes that the gravitational influence of one planet on another is negligible compared to the dominant central star. This simplification, while useful for certain scenarios, fails to capture the intricacies of the true interactions occurring between heavier celestial bodies. As a result, the dynamic interplay of gravitational forces, which can significantly affect the orbital dynamics and stability of planetary systems, is often disregarded or oversimplified.

In this research, we will try to expand an algorithm [1] that describes the evolution of a Kepler system consisting of $N$ celestial bodies, with these gravitational interactions which we will call scatterings. Furthermore, we will look at the average time between two consecutive scatterings, and determine whether our expected theoretical average gives an accurate approximation of this value. Also, we will determine what the influence of individual input parameters is on this average scattering time. This research thus aims to quantitatively answer the following questions:

1. *Do gravitational scatterings have a significant impact on the evolution of a Kepler system?*

2. *Does the simulated average time between two scatterings correspond with expected average?*

3. *What is the effect of increasing the number of particles, the maximum particle mass, the maximum semi-major axis, the maximum eccentricity and the maximum inclination on the average time between two scatterings?*

Chapter 2 of this report will explore the existing theory, providing an analysis of relevant topics used for answering these questions. Subsequently, in chapter 3, the experimental methodology employed in this research will be presented. Chapter 4 will show the results obtained from the simulations.

These results will then be discussed in chapter 5. Finally, chapter 6 will conclude this report by summarizing the key outcomes and their implications. This research report is part of the BSc programme Applied Mathematics and Applied Physics of the Delft University of Technology and is an attempt to obtain the degree of Bachelor of Science.

# 2. Theory

In this chapter, the theory required for the use of Visser's algorithm [1] with gravitational scatterings is described.

## 2.1 Kepler orbits

Orbital elements are used to describe the equations of motion for a particle in a Kepler system. This particle can for example be a planet in orbit around a large central mass such as the Sun, a satellite in orbit around the Earth or a ring particle in a planetary ring system. At large distance from other particles, we assume the particles move in elliptical orbits and only feel the gravitational force of the Sun. Each particle is in a Kepler orbit defined by its angular momentum vector $\boldsymbol{L}$ and eccentricity vector $\boldsymbol{\epsilon}$, where the latter has magnitude $\epsilon$ and points from the center of the orbit to the periapsis, which is the point of closest approach to the central body. We can write $\boldsymbol{L}$ and $\boldsymbol{\epsilon}$ in terms of the five independent conserved orbital elements: the semi-major axis $a$, the eccentricity $\epsilon$, the argument of periapsis, $\varpi$ (pronounced as 'varpi'), longitude of the ascending node $\Omega$ and the inclination $I$ (see Figure 2.1). This can be done by using the following equations from [1]:

$$\boldsymbol{L} = L\mathscr{R}\begin{pmatrix}0\\0\\1\end{pmatrix} = L\begin{pmatrix}\sin\Omega\sin I\\-\cos\Omega\sin I\\\cos I\end{pmatrix}, L = m\omega ab, \omega = \sqrt{\frac{GM}{a^3}}, \tag{2.1}$$

$$\boldsymbol{\epsilon} = \epsilon\mathscr{R}\begin{pmatrix}1\\0\\0\end{pmatrix} = \epsilon\begin{pmatrix}\cos\varpi\cos\Omega - \sin\varpi\sin\Omega\cos I\\\cos\varpi\sin\Omega + \sin\varpi\sin\Omega\cos I\\\sin\varpi\sin I\end{pmatrix}, \tag{2.2}$$

where $L = m\omega ab, \omega = \sqrt{\frac{GM}{a^3}}$ and $b = \sqrt{1-\epsilon^2}a$. The orbit is parametrized by the true anomaly $\nu$ or the eccentric anomaly $E$. The position vector and the velocity vector can now be expressed as:

$$\boldsymbol{r} = \begin{pmatrix}x\\y\\z\end{pmatrix} = r\mathscr{R}\begin{pmatrix}\cos\nu\\\sin\nu\\0\end{pmatrix} = \mathscr{R}\begin{pmatrix}a\cos E - c\\b\sin E\\0\end{pmatrix}, \tag{2.3}$$

$$\boldsymbol{v} = \dot{\boldsymbol{r}} = \frac{\omega a}{b}\mathscr{R}\begin{pmatrix}-a\sin\nu\\a\cos\nu + c\\0\end{pmatrix} = \frac{\omega a}{r}\mathscr{R}\begin{pmatrix}-a\sin E\\b\cos E\\0\end{pmatrix}, \tag{2.4}$$

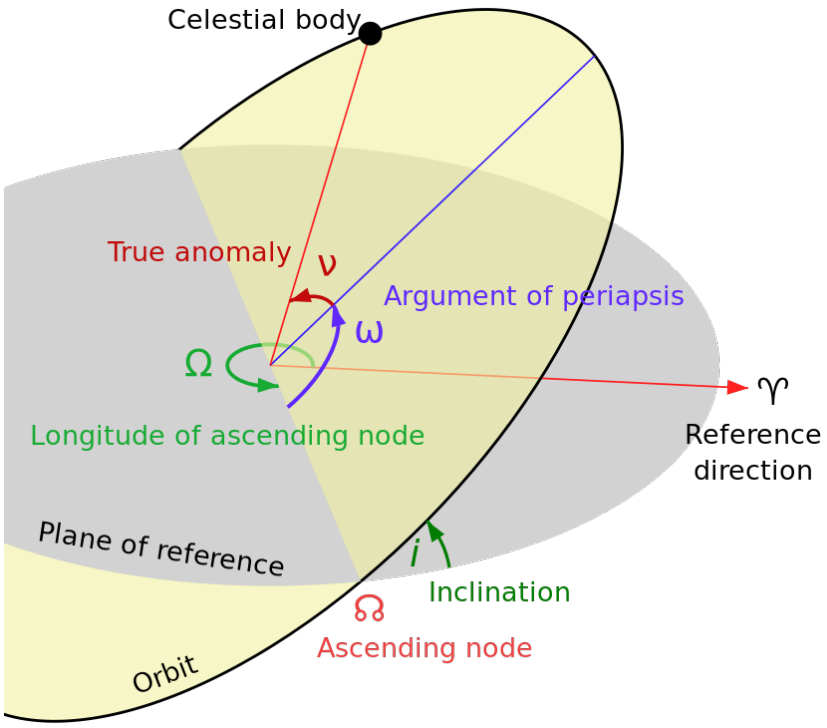


**Figure 2.1:** Diagram illustrating the orbital elements. [2]

where the equation for the orbit in the plane is:

$$r = \frac{b^2}{a + c\cos\nu} \tag{2.5}$$

and $\mathscr{R}$ is a constant rotation matrix:

$$\mathscr{R} = \begin{pmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos I & -\sin I \\ 0 & \sin I & \cos I \end{pmatrix} \begin{pmatrix} \cos\varpi & -\sin\varpi & 0 \\ \sin\varpi & \cos\varpi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.6}$$

The relation between the eccentric anomaly $E$, the mean anomaly $M$ and the time $t$ since periapsis is described by Kepler's equation:

$$M = \omega t = E - \sin E \Leftrightarrow t = \frac{E - \sin E}{\omega}. \tag{2.7}$$

## 2.2 Gravitational sphere of influence

Each particle in our Kepler system has mass and therefore a gravitational sphere of influence, within which the gravity of the particle becomes signif-

icant as opposed to that of the central body. When another particle enters this sphere of influence, we will perform a scattering event in CM-frame of the two particles, during which the gravitational influence of the Sun is neglected. There are several definitions available that each provide a useful estimate for this sphere of influence. We will use the Laplace sphere, which is defined as: *the radius of a spherical region where the perturbing effect of the Sun on the particle's planetocentric orbit is lower than the perturbing effect of the planet on the particle's heliocentric orbit* [3].

We will now derive Laplace's expression for the radius of this spherical region. Consider a three-body problem for a system with a Sun, a planet, and a satellite (see figure 2.2). The sphere of influence of the planet is now that region of space in which it is feasible to assume a planet as the central body and the Sun as the perturbing body when computing perturbations [4]. The equations of motion of the satellite can be written both in the reference frame of the Sun and in the reference frame of the planet.

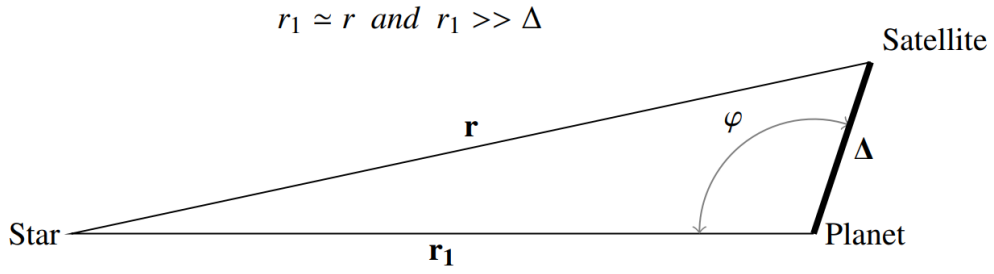


**Figure 2.2:** Geometry of the three-body problem (not to scale). $\boldsymbol{r}_1$ and $\boldsymbol{r}$ are the heliocentric position vectors of the planet and the satellite, respectively. In the reference frame attached to the planet, the position vector of the satellite is $\boldsymbol{\Delta} = \boldsymbol{r} - \boldsymbol{r}_1$[5].

For the heliocentric reference frame, the main body is considered to be the Sun with mass $M$, and the perturbing acceleration is caused by the attraction of the planet with mass $m$. We thus find for the motion of the satellite (mass $m' \ll m$):

$$\ddot{\boldsymbol{r}} = -\frac{GM}{r^3}\boldsymbol{r} - Gm\left[\frac{\boldsymbol{\Delta}}{\Delta^3} + \frac{\boldsymbol{r}_1}{r_1^3}\right] = \boldsymbol{A} + \boldsymbol{F}, \tag{2.8}$$

where $\boldsymbol{\Delta} = \boldsymbol{r} - \boldsymbol{r}_1$, $\boldsymbol{A}$ is the central acceleration of the satellite caused by the Sun and $\boldsymbol{F}$ the perturbed acceleration caused by the planet. These accelerations and their magnitudes are thus given by:

$$\begin{cases} \boldsymbol{A} = -\frac{GM}{r^3}\boldsymbol{r}, & A = \frac{GM}{r^2} \simeq \frac{GM}{r_1^2}, \\ \boldsymbol{F} = -Gm\left[\frac{\boldsymbol{\Delta}}{\Delta^3} + \frac{\boldsymbol{r}_1}{r_1^3}\right], & F \simeq \frac{Gm}{\Delta^2}, \end{cases} \tag{2.9}$$

where the approximations hold since we assume that $r \simeq r_1$ and $r_1 \gg \Delta$. Similarly, for the planetary reference frame we find that:

$$\ddot{\boldsymbol{\Delta}} = -\frac{Gm}{\Delta^3}\boldsymbol{\Delta} - GM\left[\frac{\boldsymbol{r}}{r^3} - \frac{\boldsymbol{r}_1}{r_1^3}\right] = \boldsymbol{A}_1 + \boldsymbol{F}_1, \tag{2.10}$$

where $\boldsymbol{A}_1$ is now the central acceleration of the satellite caused by the planet and $\boldsymbol{F}_1$ is the perturbed acceleration caused by the Sun. These accelerations and their magnitudes are given by:

$$\begin{cases} \boldsymbol{A}_1 = -\frac{Gm}{\Delta^3}\boldsymbol{\Delta}, & A_1 = \frac{Gm}{\Delta^2}, \\ \boldsymbol{F}_1 = -GM\left[\frac{\boldsymbol{r}}{r^3} - \frac{\boldsymbol{r}_1}{r_1^3}\right], & F_1 \simeq \frac{GM\Delta}{r_1^3}\sqrt{1+3\cos^2\phi}, \end{cases} \tag{2.11}$$

where $\cos\phi = \frac{\boldsymbol{r}_1\cdot\boldsymbol{\Delta}}{r_1\Delta}$. $F$ and $F_1$ are derived by writing $\left\|\frac{\boldsymbol{\Delta}}{\Delta^3} + \frac{\boldsymbol{r}_1}{r_1^3}\right\|$ and $\left\|\frac{\boldsymbol{r}}{r^3} - \frac{\boldsymbol{r}_1}{r_1^3}\right\|$ as functions of $u = \frac{\Delta}{r_1}$ and $\cos\phi$, and then truncating the expressions to the $0^{th}$ order in $u$, since $u$ can be considered small. Step-by-step calculations to arrive at the aforementioned magnitudes of $\boldsymbol{F}$ and $\boldsymbol{F}_1$ are tedious and have therefore not been included in this chapter, but can be found in the paper by Chebotarev [4]. From the accelerations given above, equating the ratio of Solar accelerations (central/perturbing) to that of the planetary accelerations (central/perturbing) gives the natural bounding surface of the sphere of influence of the planet:

$$\frac{A}{F} = \frac{A_1}{F_1}, \tag{2.12}$$

where the sphere of influence itself is thus the region for which $\frac{A_1}{F_1} > \frac{A}{F}$ holds. Substituting Equations (2.10) and (2.11) into (2.12) gives:

$$s^{inf} = r_1\left(\frac{m^2}{M^2}\right)^{1/5}\frac{1}{(1+3\cos^2\phi)^{1/10}}. \tag{2.13}$$

Considering that $1 \leq (1+3\cos^2\phi)^{1/10} \leq 2^{2/5} \simeq 1.32$, we simply approximate this value with 1 and finally get the radius of Laplace's sphere of influence:

$$s^{inf} = r_1\left(\frac{m}{M}\right)^{2/5}. \tag{2.14}$$

## 2.3 Elastic collision using a hyperbolic orbit

When two particles travel inside one of the other's gravitational sphere of influence, the gravitational attraction between them has to be accounted for. During this encounter, we will ignore the gravitational attraction from the Sun, thus obtaining a classical two-body problem. If the two bodies do not hit each other, no energy is dissipated and the process can be modelled as an elastic collision. We consider two particles $m_1$ and $m_2$, moving on an elliptical orbits described by position vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ and velocity vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, respectively. The interaction between these particles can be described by the relative coordinate $\boldsymbol{d} = \boldsymbol{r}_2 - \boldsymbol{r}_1$. This relative coordinate moves in a hyperbolic orbit with velocity $\boldsymbol{u} = \boldsymbol{v}_2 - \boldsymbol{v}_1$ in the center-of-mass frame [1] (see Figure 2.3).

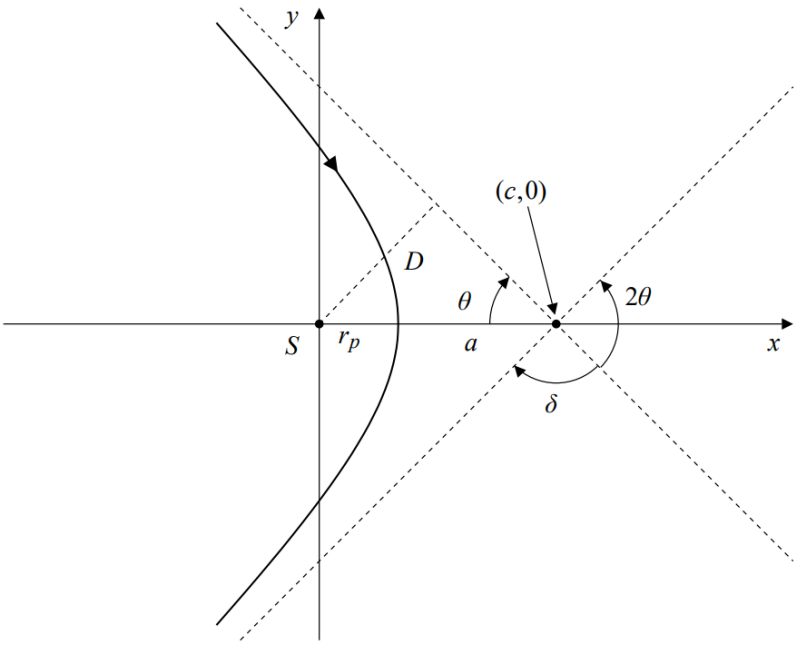


**Figure 2.3:** Hyperbolic trajectory and asymptotes of the relative coordinate. In this figure, $S$ is the centre-of-mass, $D$ is the shortest distance of the incoming asymptote to the origin, $a$ is the semi-transverse axis, $c$ is the semi-focal separation, $r_p$ is the distance between the trajectory and the centre-of-mass at periapsis ($y = 0$), $\theta$ is the angle between the $x$-axis and the incoming asymptote and $\delta$ is the angle through which the relative coordinate's path is deflected by the gravitational attraction of the centre-of-mass. [6]

We will approximate the trajectory of the relative coordinate by its asymptotes. Thus, before the scattering event, the position of the relative coordinate moves on a straight line $\boldsymbol{d}_i(t)$ with constant velocity $\boldsymbol{u}_i$. By conservation

of momentum, it immediately follows that the magnitude of the velocity vector is conserved, hence $u_f = u_i = u$. The shortest distance between the two masses is reached when $\boldsymbol{d}_i \perp \boldsymbol{u}_i$, and we define $\boldsymbol{d}_i^{\perp}$ for the position vector where this is the case (equal to the dashed line with length D in Figure 2.3). At this point, we perform the scattering event by rotating the direction of the position and velocity both by an angle $\delta = \pi - 2\theta$. The particle then continues its trajectory with velocity $\boldsymbol{u}_f$, starting from $\boldsymbol{d}_f^{\perp}$, which is similarly defined as the position vector where $\boldsymbol{d}_f \perp \boldsymbol{u}_f$. By conservation of angular momentum, $\boldsymbol{d}_i \times \boldsymbol{u}_i = \boldsymbol{d}_f \times \boldsymbol{u}_f$ and hence $d_f^{\perp} = d_i^{\perp} = d^{\perp}$.

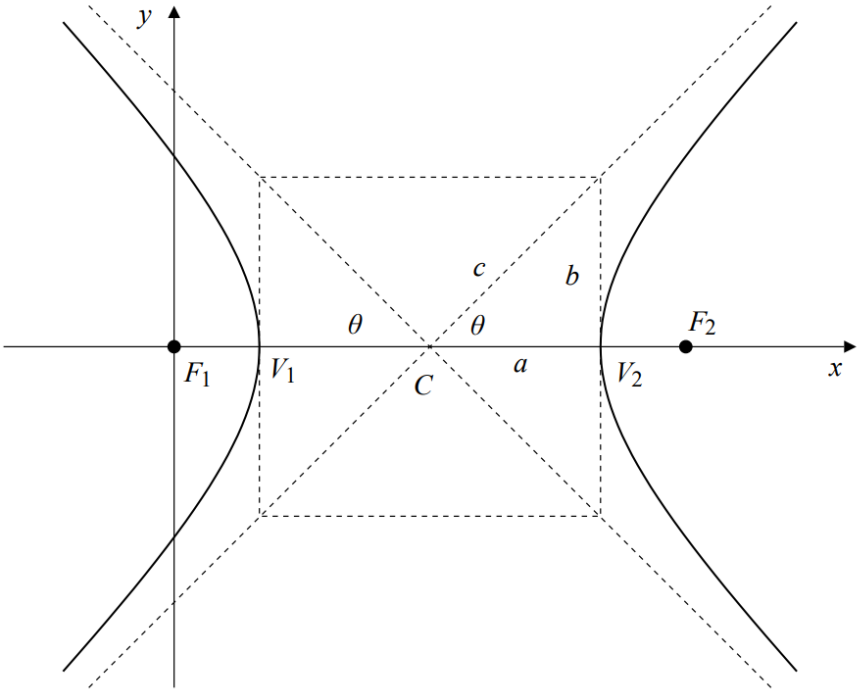


**Figure 2.4:** Geometry of a hyperbola. In this figure, $a$ is the semi-transverse axis, $b$ is the semi-conjugate axis, $c = \sqrt{a^2 + b^2}$ is the semi-focal separation, $\theta$ is the angle between the $x$-axis and the incoming asymptote. $F_1$ and $F_2$ are the focal points with vertices $V_1$ and $V_2$, respectively. $C = (c, 0)$ is the centre of the hyperbole. [6]

From the geometry of Figure 2.4, we see that $\tan\theta = \frac{b}{a}$. Since $\theta = \frac{\pi}{2} - \frac{\delta}{2}$ and $\tan\left(\frac{\pi}{2} - \frac{\delta}{2}\right) = (\tan\frac{\delta}{2})^{-1}$, it follows that $\delta = 2\arctan\frac{a}{b}$. The conservation of energy equation for a hyperbolic Keplerian orbit is given by the *vis-viva equation* (where $a$ is defined to be positive) [7]:

$$u^2 = G(m_1 + m_2)\left(\frac{2}{d} + \frac{1}{a}\right). \tag{2.15}$$

When $d \to \infty$, we find that $a = \frac{G(m_1+m_2)}{u^2}$. Since $c^2 = a^2 + b^2$, we find two angle-angle-side $(\frac{\pi}{2}, \theta, c)$ congruent triangles from which it follows that $b = D$. From the geometry of Figure 2.3 it follows that $D$ is equal to the shortest distance of the incoming asymptote to the origin, which is the length $d^\perp$ of the vector $\boldsymbol{d}_i^\perp$ that is perpendicular to $\boldsymbol{u}_i$. Hence $b = d^\perp$. We can now compute the rotation by using:

$$\begin{bmatrix} \boldsymbol{d}_f^\perp \\ \boldsymbol{u}_f \end{bmatrix} = \mathscr{R}_\delta \begin{bmatrix} \boldsymbol{d}_i^\perp \\ \boldsymbol{u}_i \end{bmatrix}, \tag{2.16}$$

where

$$\mathscr{R}_\delta = \begin{bmatrix} \cos\delta & \frac{b}{u}\sin\delta \\ -\frac{u}{b}\sin\delta & \cos\delta \end{bmatrix} = \begin{bmatrix} \frac{b^2-a^2}{c^2} & \frac{2ab^2}{uc^2} \\ -\frac{2au}{c^2} & \frac{b^2-a^2}{c^2} \end{bmatrix}. \tag{2.17}$$

The placement of the positive and negative signs on the off-diagonal elements follows from the fact that the force acting on the relative coordinate is attractive.

## 2.4 Estimated time between two scatterings

One of the objectives of this research is to test whether the measured average of the time between two scatterings $\bar{T}_{scatt}$ in our simulations corresponds with a theoretical estimate $\mathbb{E}[T_{scatt}]$. This theoretical estimate of the scattering time will be derived in this section.

In our simulation, every particle in the system will have mass $m_i \in [m_{min}, m_{max}]$, semi-major axis $a_i \in [a_{min}, a_{max}]$, an inclination $I \in [0, I_{max}]$ and eccentricity $\epsilon_i \in [0, \epsilon_{max}]$ Each particle thus exists inside a volume that represents a segment of a sphere. The volume of this segment is given by:

$$V = \int_0^{2\pi} \int_{\frac{\pi}{2}-I_{max}}^{\frac{\pi}{2}+I_{max}} \int_{a_{min}}^{a_{max}} r^2 \sin\theta dr d\theta d\phi = \frac{4\pi}{3}(a_{max}^3 - a_{min}^3)\sin I_{max}. \tag{2.18}$$

By taking $a_{min} = 0$ and $I_{max} = \frac{\pi}{2}$, we recover the volume of the entire sphere. For the distribution of the radii $s_i$ of the particles in the system we take $dN/ds \propto s^{-3}$, which is roughly the same as in the Asteroid belt [8]. From the assumption that each particle has the same density of mass, it follows that $m_i \propto s_i^3$ and thus $dN/dm \propto m^{-5/3}$. Since $\int_{m_{min}}^{m_{max}} p(m)dm = \int_{m_{min}}^{m_{max}} Am^{-5/3}dm = 1$, we find:

$$p(m) = \frac{2}{3}\frac{(m_{max}m_{min})^{2/3}}{m_{max}^{2/3} - m_{min}^{2/3}} m^{-5/3}. \tag{2.19}$$

To create a system of that has a uniform particle density for its entire volume, we require that $dN/da \propto a^2$ and $dN/dI \propto \sin I$. Since $\int_{a_{min}}^{a_{max}} p(a)da = \int_{a_{min}}^{a_{max}} Ba^2 da = 1$, we find:

$$p(a) = \frac{3}{a_{max}^3 - a_{min}^3} a^2. \tag{2.20}$$

For the inclination we then have $\int_0^{I_{max}} p(I)dI = \int_0^{I_{max}} C \sin I dI = 1$, from which it follows that:

$$p(I) = \frac{1}{1 - \cos I_{max}} \sin I. \tag{2.21}$$

The eccentricity is drawn from a uniform distribution $\mathcal{U}[0, \epsilon_{max}]$ and thus $p(\epsilon) = \frac{1}{\epsilon_{max}}$. Now, we assume that the scattering rate $\frac{1}{T_{scatt}}$, defined as the rate at which scatterings occur between two particles somewhere in the system, is a function of the time-averaged velocity $\langle v \rangle_i$ of a particle $i$, its effective cross section $A_i$, the total volume $V$ of the system and the number of possible scattering pairs $\binom{N}{2}$ in the following way [1]:

$$\frac{1}{T_{scatt}} = \frac{\langle v \rangle_i A_i}{V} \binom{N}{2}, \tag{2.22}$$

where $A_i = \pi (s_i^{inf})^2 = \pi \left(\frac{m_i}{M}\right)^{4/5} a_i^2$. For $\langle v \rangle_i$ we have to compute:

$$\langle v \rangle_i = \frac{1}{T_i} \int_0^{T_i} v_i dt = \frac{\omega}{2\pi} \int_0^{2\pi} v_i \frac{dt}{dE} dE. \tag{2.23}$$

It follows from Equations 2.4 and2.5 that:

$$v = \|\boldsymbol{v}\| = \frac{\omega \sqrt{a^2 \sin^2 E + b^2 \cos^2 E}}{1 - \epsilon \cos E} = \frac{\omega a \sqrt{1 - \epsilon^2 \cos^2 E}}{1 - \epsilon \cos E}, \tag{2.24}$$

and from Equation 2.7 that:

$$\frac{dt}{dE} = \frac{1 - \epsilon \cos E}{\omega} \tag{2.25}$$

We are thus able to write:

$$\langle v \rangle = \frac{\omega a}{2\pi} \int_0^{2\pi} \sqrt{1 - \epsilon^2 \cos^2 E} dE \simeq \frac{\omega a}{2\pi} \int_0^{2\pi} \left(1 - \frac{\epsilon^2}{2} \cos^2 E\right) dE, \tag{2.26}$$

where we have applied the Taylor series of $\sqrt{1 - x^2}$, since $\epsilon^2$ can be assumed small. Hence, for the time-averaged velocity of particle $i$, we find that:

$$\langle v \rangle_i \simeq \omega_i a_i (1 - \frac{1}{4}\epsilon_i^2) = \sqrt{\frac{GM}{a_i}} (1 - \frac{1}{4}\epsilon_i^2). \tag{2.27}$$

$\binom{N}{2} \simeq \frac{N^2}{2}$ for large $N$. Substituting all these expressions back into 2.22 gives us for the expected duration of time between two scatterings somewhere in the system:

$$T_{scatt} = \frac{2V}{N^2 v_i A_i} = \frac{8M^{3/10}(a_{max}^3 - a_{min}^3)\sin I_{max}}{3\sqrt{G}N^2}(1 - \frac{1}{4}\epsilon_i^2)^{-1} a_i^{-3/2} m_i^{-4/5}. \tag{2.28}$$

In order to find $\mathbb{E}[T_{scatt}]$, we compute:

$$\mathbb{E}[(1 - \frac{1}{4}\epsilon^2)^{-1}] = \int_0^{\epsilon_{max}} \frac{1}{1 - \frac{1}{4}\epsilon^2} p(\epsilon) d\epsilon = \frac{1}{\epsilon_{max}}[\ln(2 + \epsilon_{max}) - \ln(2 - \epsilon_{max})], \tag{2.29}$$

and we note that $\mathbb{E}[(1 - \frac{1}{4}\epsilon^2)^{-1}] \simeq 1 + \frac{\epsilon_{max}^2}{12}$ for small values of $\epsilon_{max}$. Furthermore, we compute:

$$\mathbb{E}[a^{-3/2}] = \int_{a_{min}}^{a_{max}} a^{-3/2} p(a) da = \frac{2}{a_{max}^{3/2} + a_{min}^{3/2}}, \tag{2.30}$$

and

$$\mathbb{E}[m^{-4/5}] = \int_{m_{min}}^{m_{max}} m^{-4/5} p(m) dm = \frac{5}{11} \frac{m_{max}^{22/15} - m_{min}^{22/15}}{m_{max}^{22/15} m_{min}^{4/5} - m_{max}^{4/5} m_{min}^{22/15}}, \tag{2.31}$$

which is a decreasing function for $m_{max}$ when fixing $m_{min}$. We thus obtain the theoretical estimate for the time between two scatterings:

$$\mathbb{E}[T_{scatt}] = \frac{4}{9} \frac{M^{3/10}(a_{max}^{3/2} - a_{min}^{3/2})(12 + \epsilon_{max}^2)\sin I_{max}}{\sqrt{G}N^2} \mathbb{E}[m^{-4/5}]. \tag{2.32}$$

Hence, we expect that, while keeping all other parameters fixed, to find an increase in the expected time between two scatterings $\mathbb{E}[T_{scatt}]$ (so less scatterings) when $a_{max}$, $\epsilon_{max}$ or $I_{max}$ is increased and when $N$ or $m_{max}$ is decreased.

# 3. Experimental Method

In this chapter, the specific ways the simulations will be run are elaborated upon. First, the initialisation of the system is discussed. Next, the main loop of Visser's algorithm [1] will be described with which the system will be evaluated through time. After this, the main focus of this research will be presented, which is the finding and performing of gravitational scattering events for close encounters between two particles.

## 3.1 Initialisation

We now initialise a system of particles orbiting a central mass. The particles are numbered $i = 1, ..., N$. For each particle $i$, we store the following set of variables:

$$\{t_i^0, a_i, c_i, s_i, m_i, \boldsymbol{r}_i^0, \boldsymbol{L}_i, \boldsymbol{\epsilon}_i, \omega_i\}. \tag{3.1}$$

Here, $t^0$ is the time of creation, $a$ is the semi-major axis, $c = \epsilon a$ the focal distance, $s$ is the particle radius, $m$ is the particle mass, $\boldsymbol{r}^0$ is its position, $\boldsymbol{L}$ its angular momentum vector, $\boldsymbol{\epsilon}$ is the eccentricity vector, and $\omega$ is the angular frequency of the orbit. For each particle that is created at the start of the simulation, we have that $t_i^0 = 0$. We choose each particle mass at random such that the mass density function is proportional to $m^{-5/3}$. In order to make a uniform particle disk we choose each of the six orbital $a_i, \epsilon_i, I_i, \varpi, \Omega_i$ and $M_i$ of the initial particles at random within a specified domain, see Table 3.1. Derivations of the expressions in Table 3.1 can be found in Appendix A. We give each of the particles the same mass density as Jupiter (1326 kg/m$^3$), hence $s_i = s_J(m_i/m_J)^{1/3}$. The eccentric anomaly is determined from the mean anomaly by performing three iterations of the recursive formula $E_{i,n+1} = E_{i,n} - (E_{i,n} - \epsilon_i \sin E_{i,n} - M_i)/(1 - \epsilon_i \cos E_{i,n})$ [9].

The boundaries of each parameter domain are specified in Table 3.2 and are based on those in the Asteroid belt [8]. We define a standard simulation as one where all domain boundaries are given by their standard value. When testing the influence of a specific parameter, we randomize the boundary within its test domain once for the entire simulation, while keeping all other boundaries fixed at their standard value. We then compare the effect on the scattering time of this parameter over multiple simulations.

| Parameter | Domain |
|---|---|
| $m_i$ | $m_{min}[1-\xi(1-(m_{min}/m_{max})^{2/3}]^{-3/2}$ |
| $a_i$ | $\sqrt[3]{a_{min}^3+\xi(a_{max}^3-a_{min}^3)}$ |
| $\epsilon_i$ | $\epsilon_{max}\xi$ |
| $I_i$ | $\arccos(\cos I_{max}+\xi(1-\cos I_{max}))$ |
| $\varpi_i$ | $2\pi\xi$ |
| $\Omega_i$ | $2\pi\xi$ |
| $M_i$ | $2\pi\xi$ |

**Table 3.1:** Initialization domains of the particles in the system. We choose $\xi \sim \mathcal{U}[0,1]$ randomly for the parameters of each particle. See Appendix A for derivations of the shown expressions.

| Parameter | Standard value | Test domain |
|---|---|---|
| $N$ | $10^3$ | $10^{[2,3]}$ |
| $m_{max}$ | $10^{-7}$ $M_\odot$ | $10^{[-8,-6]}$ $M_\odot$ |
| $m_{min}$ | $10^{-10}$ $M_\odot$ | |
| $a_{max}$ | 4 AU | [2,6] AU |
| $a_{min}$ | 1 AU | |
| $\epsilon_{max}$ | 0.1 | [0,0.5] |
| $I_{max}$ | 0.1 | $10^{[-4,-1]}$ |

**Table 3.2:** Boundaries of the initialization domains. The boundaries of each of the parameters are randomized individually within their test domains, while keeping all other parameters fixed at their standard value.

## 3.2 Main loop of the algorithm

The algorithm used in the simulations in this research is described in Visser's paper [1]. We have adjusted the algorithm to look for gravitational scatterings instead of collisions between two particles.

1. If the scattering list is empty, end the simulation.
2. Take the pair $(i,j)$ with the soonest scattering: the first in the list. Perform the scattering event (see section 3.4).
3. Update the time $t$ to the time $t^1_{(i,j)}$ of the scattering.
4. Remove any pair containing $i$ and any pair containing $j$ from the list of scattering pairs.
5. Remove the particles $i$ and $j$ from the particle list.

6. If the orbit of the new particles intersects the central mass or is unbound, go to the next scattering on the list. (see section 3.5)

7. Create new particles $i'$ and $j'$ defined by $\left\{t^1_{(i,j)}, a, c, s, m, \boldsymbol{r}^0, \boldsymbol{L}, \boldsymbol{\epsilon}, \omega\right\}$ (see section 3.5)

8. For any new particle, consider the other particles and decide if there are new scatterings possible. If this is the case, calculate the time of the earliest scattering.

9. Make a sorted list of the new scattering possibilities with a record of the scattering time and the pair, soonest encounter first.

10. Merge this sorted list with the existing sorted list of scattering possibilities into a full list of pairs, sorted by time of the scattering event, soonest scattering first.

## 3.3 Scattering detection

Before executing the main loop of the algorithm as described above, we have to create a list of all possible scattering pairs sorted by the time they occur. The method for detecting these scattering pairs and the moments they take place are described in this section and are based on derivations in [1].

### 3.3.1 Determining scattering pairs

The radial coordinate for a body with a gravitational sphere of influence $s_i^{inf} = a_i(\frac{m_i}{M})^{2/5}$ ranges over the interval from periapsis to apoapsis: $[a_i - c_i - s_i^{inf}, a_i + c_i + s_i^{inf}]$. As long as $a_i + c_i + s_i^{inf} \geq a_j - c_j - s_j^{inf}$, the intervals for $i$ and $j$ overlap and the pair is a candidate for scattering. Now, consider a pair $(i,j) = (1,2), a_1 > a_2$ for which the above criteria hold. We compute the direction of the nodal line $\boldsymbol{K}$ and its magnitude $K$:

$$\boldsymbol{K} = \boldsymbol{L}_1 \times \boldsymbol{L}_2, \qquad K = \sqrt{\boldsymbol{K} \cdot \boldsymbol{K}}. \tag{3.2}$$

For both particles (1,2) we now compute the semi-latus rectum $\ell$, the two intersection points with the nodal line $\boldsymbol{r}_\pm$ and the velocities $\boldsymbol{v}_\pm$ at these points:

$$\ell_1 = \frac{\boldsymbol{L}_1 \cdot \boldsymbol{L}_1}{GMm_1^2}, \qquad \ell_2 = \frac{\boldsymbol{L}_2 \cdot \boldsymbol{L}_2}{GMm_2^2}, \tag{3.3}$$

$$\boldsymbol{r}_{1,\pm} = \frac{\boldsymbol{K}\ell_1}{\pm K + \boldsymbol{\epsilon}_1 \cdot \boldsymbol{K}}, \qquad \boldsymbol{r}_{2,\pm} = \frac{\boldsymbol{K}\ell_2}{\pm K + \boldsymbol{\epsilon}_2 \cdot \boldsymbol{K}}, \tag{3.4}$$

$$r_{1,\pm} = \sqrt{\boldsymbol{r}_{1,\pm} \cdot \boldsymbol{r}_{1,\pm}}, \qquad r_{2,\pm} = \sqrt{\boldsymbol{r}_{2,\pm} \cdot \boldsymbol{r}_{2,\pm}}, \tag{3.5}$$

$$\boldsymbol{v}_{1,\pm} = \frac{\boldsymbol{L}_1}{m_1\ell_1} \times \left(\boldsymbol{\epsilon}_1 + \frac{\boldsymbol{r}_{1,\pm}}{r_{1,\pm}}\right), \qquad \boldsymbol{v}_{2,\pm} = \frac{\boldsymbol{L}_2}{m_2\ell_2} \times \left(\boldsymbol{\epsilon}_2 + \frac{\boldsymbol{r}_{2,\pm}}{r_{2,\pm}}\right). \tag{3.6}$$

It follows that $\boldsymbol{r}_{1+}$ and $\boldsymbol{r}_{2+}$ point in the direction of $\boldsymbol{K}$, while $\boldsymbol{r}_{1-}$ and $\boldsymbol{r}_{2-}$ point in the direction of $-\boldsymbol{K}$ (since $\boldsymbol{\epsilon}_{1,2} \cdot \boldsymbol{K} < K$ for $\epsilon < 1$).

Next, we approximate the points where the minimal orbit intersection distance (MOID) is found. This is done by considering the tangent lines of the orbits at the points $\boldsymbol{r}_{1,\pm}$ and $\boldsymbol{r}_{2,\pm}$, and then calculating the points $\boldsymbol{r}'_{1,\pm}$ and $\boldsymbol{r}'_{2,\pm}$ where the distance between these two lines is minimal. These positions can be written as:

$$\boldsymbol{r}^{min}_{1,\pm} = \boldsymbol{r}_{1,\pm} + \left[(\boldsymbol{r}_{2,\pm} - \boldsymbol{r}_{1,\pm}) \cdot \frac{\boldsymbol{v}_{2,\pm} \times (\boldsymbol{v}_{1,\pm} \times \boldsymbol{v}_{2,\pm})}{|\boldsymbol{v}_{1,\pm} \times \boldsymbol{v}_{2,\pm}|^2}\right] \boldsymbol{v}_{1,\pm}, \tag{3.7}$$

$$\boldsymbol{r}^{min}_{2,\pm} = \boldsymbol{r}_{2,\pm} + \left[(\boldsymbol{r}_{2,\pm} - \boldsymbol{r}_{1,\pm}) \cdot \frac{\boldsymbol{v}_{1,\pm} \times (\boldsymbol{v}_{1,\pm} \times \boldsymbol{v}_{2,\pm})}{|\boldsymbol{v}_{1,\pm} \times \boldsymbol{v}_{2,\pm}|^2}\right] \boldsymbol{v}_{2,\pm}. \tag{3.8}$$

For the shortest distance between the two lines we have:

$$d^{min}_{\pm} = |\boldsymbol{r}^{min}_{2,\pm} - \boldsymbol{r}^{min}_{1,\pm}| = \left|(\boldsymbol{r}_{2,\pm} - \boldsymbol{r}_{1,\pm}) \cdot \frac{(\boldsymbol{v}_{1,\pm} \times \boldsymbol{v}_{2,\pm})}{|\boldsymbol{v}_{1,\pm} \times \boldsymbol{v}_{2,\pm}|}\right|. \tag{3.9}$$

We will perform a gravitational scattering whenever $d^{min}_{\pm} \leq \max\{s^{inf}_1, s^{inf}_2\}$.

### 3.3.2 Calculating the time of a scattering

We suppose the expression above is satisfied for two particles $(i, j) = (1,2)$. To calculate the time a scattering occurs, we first have to know the time for which the particles first pass the nodal line $\boldsymbol{K}$. These times are denoted as $t^1_1$ and $t^1_2$, respectively. For each particle, we need to know the difference in eccentric anomaly $\Delta E = E^1 - E^0$ between position vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^0$. This angle $\Delta E$ can be written as the argument of a complex number as follows [1]:

$$\Delta E = \arg\left[\left(\frac{\boldsymbol{r}^1}{a} - \frac{\mathrm{i}r^1\boldsymbol{v}^1}{a^2\omega}\right) \cdot \left(\frac{\boldsymbol{r}^0 - \boldsymbol{\epsilon}\boldsymbol{\epsilon}\cdot\boldsymbol{r}^0}{a - \epsilon^2 a} + \boldsymbol{\epsilon}\right) + \frac{\boldsymbol{r}^0 \cdot \boldsymbol{\epsilon}}{a} + \epsilon^2\right] \tag{3.10}$$

The scattering time $t_{k,l}$, or the time when particles 1 and 2 have completed an integer $k$ and $l$ respective number of orbits after first reaching their intersection with nodal line, is computed by:

$$t_{k,l} = kT + t^0 + \frac{\Delta E}{\omega} - \frac{\boldsymbol{\epsilon} \times (\boldsymbol{r}^1 - \boldsymbol{r}^0)}{\omega b} \cdot \frac{\boldsymbol{L}}{L}, \tag{3.11}$$

where $0 \leq \Delta E < 2\pi$. The algorithm to find values of $k$ and $l$ can be found in Chapter 6 of [1]. We now perform the scattering at this time $t_{k,l}$, when the particles are both located adequately close to $\boldsymbol{r}_{1,\pm}^{min}$ and $\boldsymbol{r}_{2,\pm}^{min}$ to be inside one of other's gravitational sphere of influence.

## 3.4 Performing a scattering event

In this section, step 2 of the main loop of the algorithm is discussed in detail. When planets come very close to each other, their mutual gravitational interaction becomes significant and cannot be ignored. This event is treated as a scattering process between two particles, ignoring the gravity from the star and other particles in the system for a short period of time. We will divide this case into three phases: the period before gravitational scattering, the scattering event itself and the period after scattering.

### 3.4.1 Before scattering

Let $t_{k,l}$ be the time derived in Equation 3.11. We linearize the initial trajectory of the orbits of the two particles $m_1$ and $m_2$ around their respective positions $\boldsymbol{r}_1 = \boldsymbol{r}_1(t_{k,l})$ and $\boldsymbol{r}_2 = \boldsymbol{r}_2(t_{k,l})$ at the time $t_{k,l}$ (where we ignore the $\pm$ subscripts in this section):

$$\boldsymbol{r}_{1,i}(t) = \boldsymbol{r}_1 + (t - t_{k,l})\boldsymbol{v}_{1,i}, \qquad \boldsymbol{r}_{2,i}(t) = \boldsymbol{r}_2 + (t - t_{k,l})\boldsymbol{v}_{2,i}, \tag{3.12}$$

$$\boldsymbol{v}_{1,i} = \boldsymbol{v}(\boldsymbol{r}_1), \qquad \boldsymbol{v}_{2,i} = \boldsymbol{v}(\boldsymbol{r}_2), \tag{3.13}$$

where the $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ follow from equation 2.3, and $\boldsymbol{v}_{1,i}$ and $\boldsymbol{v}_{2,i}$ follow from equation 2.4. We will now transform these coordinates to the centre-of-mass frame. Before the scattering event, the relative mass $\mu = \frac{m_1 m_2}{m_1 + m_2}$ has initial position $\boldsymbol{d}_i$ and velocity $\boldsymbol{u}_i$:

$$\boldsymbol{d}_i(t) = \boldsymbol{r}_{2,i}(t) - \boldsymbol{r}_{1,i}(t), \qquad \boldsymbol{u}_i = \boldsymbol{v}_{2,i} - \boldsymbol{v}_{1,i}. \tag{3.14}$$

We will perform the scattering at the time $t^{\perp}$, when the distance from $\boldsymbol{d}_i(t)$ to the origin is minimal, which occurs when $\boldsymbol{d}_i(t) \cdot \boldsymbol{u}_i = 0$. Solving for $t$ gives:

$$t^{\perp} = t_{k,l} - \frac{(\boldsymbol{r}_2 - \boldsymbol{r}_1) \cdot \boldsymbol{u}_i}{u^2}, \tag{3.15}$$

and thus the position of the relative mass $\mu$ at this time $t^\perp$ is equal to:

$$\boldsymbol{d}_i^\perp = \boldsymbol{d}_i(t^\perp) = \boldsymbol{r}_2 - \boldsymbol{r}_1 - \frac{(\boldsymbol{r}_2 - \boldsymbol{r}_1)\cdot \boldsymbol{u}_i}{u^2}\boldsymbol{u}_i \tag{3.16}$$

where the superscript follows from the fact that $\boldsymbol{d}_i^\perp \perp \boldsymbol{u}_i$. Thus $\{\frac{\boldsymbol{d}_i^\perp}{d}, \frac{\boldsymbol{u}_i}{u}\}$ forms an orthonormal basis in $\mathbb{R}^2$. The centre-of-mass $m_1 + m_2$ moves with position $\boldsymbol{R}_i(t)$ and velocity $\boldsymbol{V}_i$:

$$\boldsymbol{R}_i(t) = \frac{m_1\boldsymbol{r}_{1,i}(t) + m_2\boldsymbol{r}_{2,i}(t)}{m_1 + m_2}, \qquad \boldsymbol{V}_i = \frac{m_1\boldsymbol{v}_{1,i} + m_2\boldsymbol{v}_{2,i}}{m_1 + m_2}. \tag{3.17}$$

Since $\boldsymbol{R}_i(t)$ moves with constant velocity, we can choose an inertial reference frame in which the CM is at rest in the origin and the total momentum is zero. Both particles $m_1$ and $m_2$ are moving, but with equal and opposite momenta. This situation gives precisely the same equation as the equations for a single particle of mass equal to the reduced mass $\mu$.

### 3.4.2 The scattering event

As there is no energy loss in the interaction, it can be considered an elastic collision, and the relative coordinate describes a hyperbolic orbit with parameters:

$$a = \frac{G(m_1 + m_2)}{u^2}, \qquad b = d = |\boldsymbol{d}_i^\perp|, \qquad c^2 = a^2 + b^2, \tag{3.18}$$

where $u = |\boldsymbol{u}_i| = |\boldsymbol{u}_f|$ because of conservation of momentum. The in-going asymptote has direction $\boldsymbol{u}_i$ and the out-going asymptote has direction $\boldsymbol{u}_f$. The gravitational interaction causes a rotation of the position and velocity vector of the relative mass $\mu$ over an angle $\delta = 2\arctan(a/b)$. From equations 2.16 and 2.17, we find that:

$$\boldsymbol{d}_f^\perp = \frac{b^2 - a^2}{c^2}\boldsymbol{d}_i^\perp + \frac{2ab^2}{uc^2}\boldsymbol{u}_i, \tag{3.19}$$

$$\boldsymbol{u}_f = -\frac{2au}{c^2}\boldsymbol{d}_i^\perp + \frac{b^2 - a^2}{c^2}\boldsymbol{u}_i. \tag{3.20}$$

### 3.4.3 After scattering

The motion of the relative mass $\mu$ after the scattering event ($t > t^\perp$) has velocity $\boldsymbol{u}_f$ and position $\boldsymbol{d}_f(t) = \boldsymbol{d}_f^\perp + (t - t^\perp)\boldsymbol{u}_f$. We now have to transform

back the particle coordinates $m_1 + m_2$ and $\mu = \frac{m_1 m_2}{m_1+m_2}$. For their respective velocities $\boldsymbol{v}_{1,f}$ and $\boldsymbol{v}_{2,f}$ after the scattering event we find:

$$\boldsymbol{v}_{1,f} = \boldsymbol{V}_i - \frac{m_2}{m_1 + m_2}\boldsymbol{u}_f, \qquad \boldsymbol{v}_{2,f} = \boldsymbol{V}_i + \frac{m_1}{m_1 + m_2}\boldsymbol{u}_f. \tag{3.21}$$

The particles now travel along two new lines given by:

$$\boldsymbol{r}_{1,f}(t) = \boldsymbol{R}_i(t) - \frac{m_2}{m_1 + m_2}\boldsymbol{d}_f(t), \tag{3.22}$$

$$\boldsymbol{r}_{2,f}(t) = \boldsymbol{R}_i(t) + \frac{m_1}{m_1 + m_2}\boldsymbol{d}_f(t). \tag{3.23}$$

We see that immediately after the scattering event, we have:

$$\boldsymbol{r}_{1,f}(t^\perp) = \boldsymbol{R}_i(t^\perp) - \frac{m_2}{m_1 + m_2}\boldsymbol{d}_f^\perp, \tag{3.24}$$

$$\boldsymbol{r}_{2,f}(t^\perp) = \boldsymbol{R}_i(t^\perp) + \frac{m_1}{m_1 + m_2}\boldsymbol{d}_f^\perp. \tag{3.25}$$

## 3.5 Finding new scattering possibilities

After removing the old particles (before scattering) from the scattering and particle lists, we find the new particles (after scattering) by adjusting the parameters of the particles as defined in Equation 3.1. For each of the two particles, we know its mass $m$, its position $\boldsymbol{r} = \boldsymbol{r}_f(t_{k,l})$ and velocity $\boldsymbol{v} = \boldsymbol{v}_f$. It follows that:

$$\boldsymbol{L} = m\boldsymbol{r} \times \boldsymbol{v}, \tag{3.26}$$

$$\ell = \frac{\boldsymbol{L} \cdot \boldsymbol{L}}{GMm^2}, \tag{3.27}$$

$$r = \sqrt{\boldsymbol{r} \cdot \boldsymbol{r}}, \tag{3.28}$$

$$\boldsymbol{\epsilon} = \frac{\boldsymbol{v} \times \boldsymbol{L}}{GMm} - \frac{\boldsymbol{r}}{r}, \tag{3.29}$$

$$\epsilon = \sqrt{\boldsymbol{\epsilon} \cdot \boldsymbol{\epsilon}}. \tag{3.30}$$

The particle stays in elliptical orbit and does not collide with the central mass if $\epsilon < 1$ and $\ell > (1 + \epsilon)S$, where $S$ is the radius of the central body. If that is indeed the case, we calculate the required orbital parameters of the new particle:

$$a = \frac{\ell}{1 - \epsilon^2}, \qquad c = a\epsilon, \qquad \omega = \sqrt{\frac{GM}{a^3}}. \tag{3.31}$$

In all other cases, we eject the particle from the system. Because of conservation of energy and angular momentum in elastic collisions, we note that ejections due to $\epsilon > 1$, which would imply an increase of energy, can only be the result of approximations or other inaccuracies in our model.

# 4. Results

## 4.1 Standard domain

In this section, the results are presented of five simulations of 500 scatterings each on the standard domain as described in Table 3.2. In Figure 4.1, the time a scattering occurred in each simulation is plotted. The scattering angle $\delta$ of the 2500 total scatterings will be shown in Figure 4.2 in order to give an indication of the impact of scatterings on the evolution of a Kepler system. In Figures 4.3 and 4.4, the time between two consecutive scatterings $T_{scatt}$ is analyzed, in order to verify the expected theoretical value in Equation 2.32.

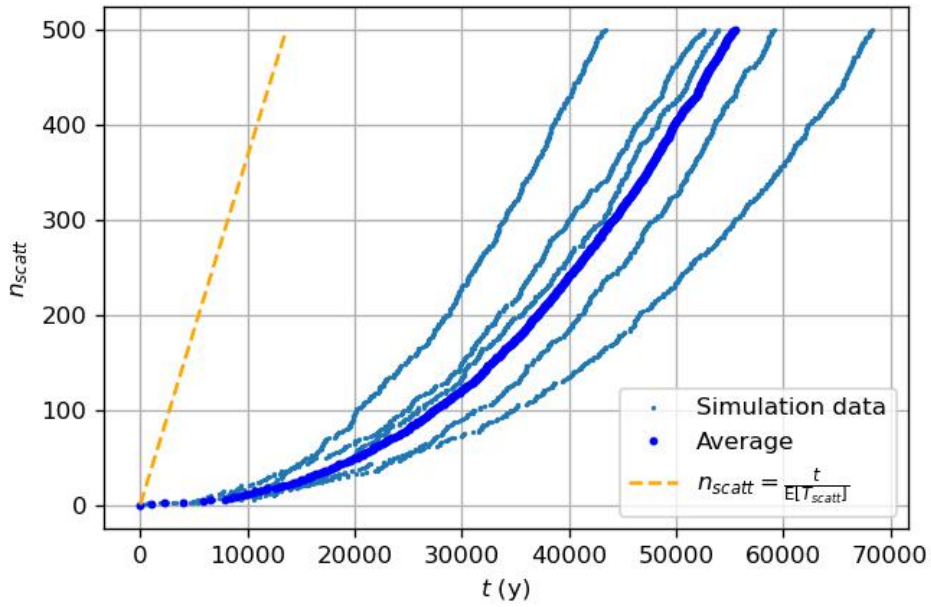


**Figure 4.1:** The number $n_{scatt}$ of each scattering is plotted for the time $t$ in years each scattering occurred. Five standard simulations of 500 scatterings (light blue dots) were computed. The thick blue line indicates the average of the five times each $n_{scatt}$ occurred. The dashed orange line shows the expected function $n_{scatt} = \frac{t}{\mathbb{E}[T_{scatt}]}$. The expected time between two scatterings $\mathbb{E}[T_{scatt}]$ is equal to 27.3 y.

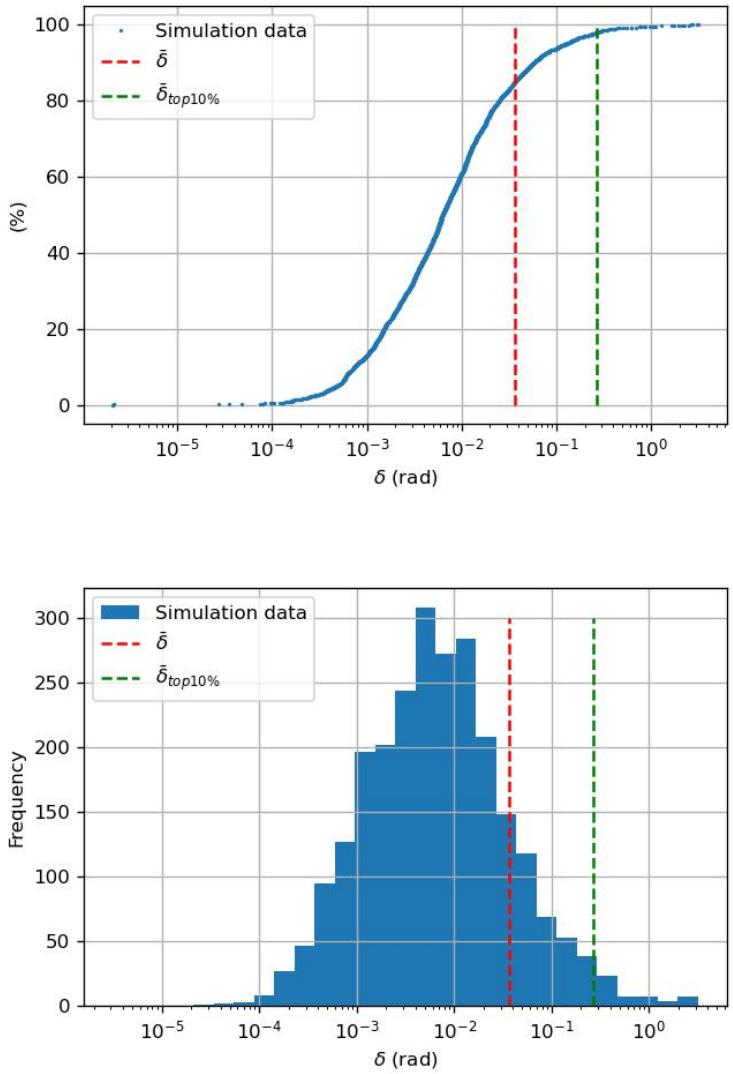


**Figure 4.2:** Above: cumulative distribution of the logarithm of the scattering angle $\delta$ for each of the 2500 scatterings that were performed. Below: Histogram of the logarithms of these 2500 scattering angles $\delta$. In these graphs, the average scattering angle of all scatterings $\bar{\delta} = 0.0366$ rad $= 2.10\ ^\circ$ is indicated with the red dashed line. The average of the 50 largest scattering angles (the top 10%) was 0.265 rad $= 15.2\ ^\circ$ and is indicated with the green dashed line.

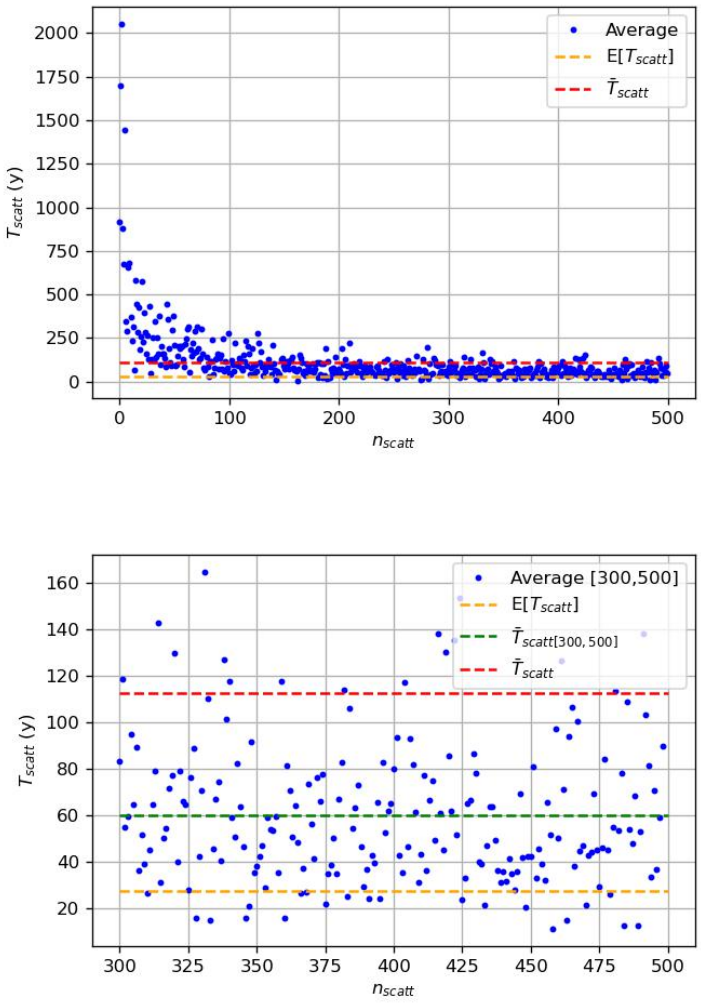


**Figure 4.3:** Above: the difference in time $T_{scatt}$ between each scattering and the scattering that occurred before it for the average of the five simulations (the thick blue line in Figure 4.1). Below: the same data points, but only on the domain $n_{scatt} \in [300, 500]$. We note that the time between two consecutive scatterings first seems to decrease as the simulation time is running, which is indicated by a negative correlation coefficient of $\rho = -0.44$ for the diagram above. In the diagram below, the time between two consecutive scatterings seems to be roughly stationary, which is indicated by the smaller correlation coefficient of $\rho = -0.08$. The average scattering time $\bar{T}_{scatt}$ of the 500 scatterings in the entire domain is equal to 107.1 y and is indicated by the dashed red line. The average scattering time $\bar{T}_{scatt}$ of the final 200 scatterings is equal to 60.0 y and is indicated by the dashed green line. The expected time between two scatterings $\mathbb{E}[T_{scatt}]$ is equal to 27.3 y and is indicated by the dashed orange line.

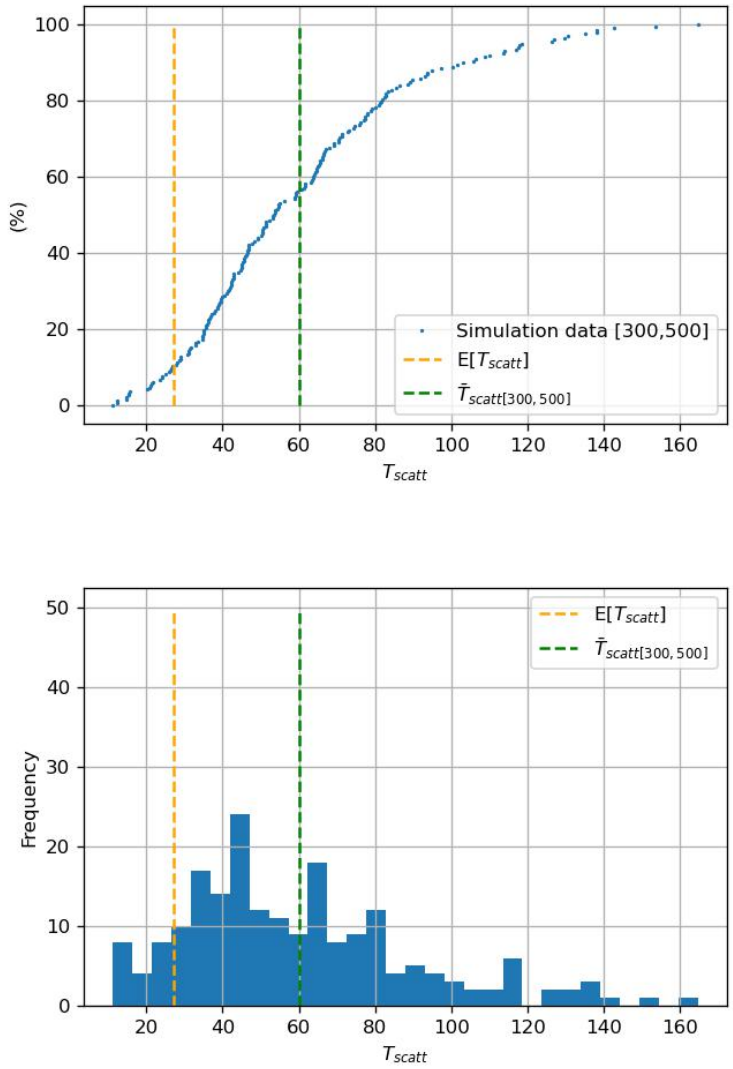


**Figure 4.4:** Cumulative distribution (above) and histogram (below) of the time between two consecutive scatterings on the domain $n_{scatt} \in [300, 500]$. It is now clear that only approximately 10% of scatterings are followed by another scattering within $\mathbb{E}[T_{scatt}] = 27.3$ years, whereas almost $T_{scatt} \leq \bar{T}_{scatt} = 60.0$ years for almost 60 % of scatterings.

## 4.2 Test domains

In this section, the results are presented of the effects of varying $N$, $m_{max}$, $a_{max}$, $\epsilon_{max}$ and $I_{max}$. These results are presented in Figures 4.5 through 4.9, respectively. In these figures, the $x$-axis is chosen is such a way that the expected linear correlation between each varied parameter and $T_{scatt}$ as derived in Equation 2.32 can be shown. Each parameter is varied individually within its respective test domain as described in Table 3.2 while keeping all other parameters fixed. This is done for 25 simulations with 100 scatterings each. In each figure, the dashed red line shows a linear fit through the simulation data and the dashed orange line shows the function of the expected value $\mathbb{E}[T_{scatt}]$ on the respective test domain.

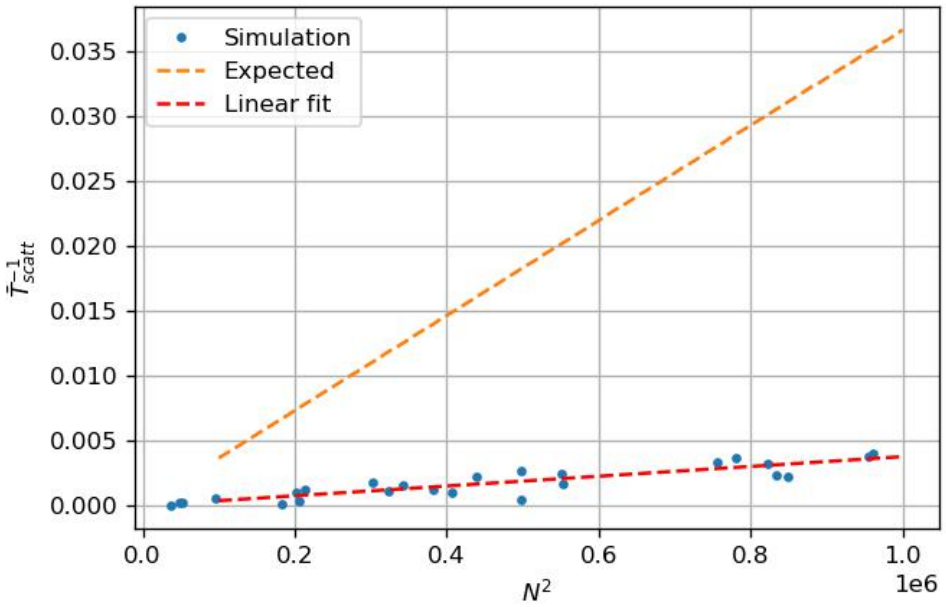


**Figure 4.5:** Inverse of the average time between two scatterings $(\bar{T}_{scatt})^{-1}$ plotted against the number of particles squared $N^2$ for 25 simulations, each with 100 performed scatterings. $N$ was varied within the interval $[10^2, 10^3]$ to obtain these results.The correlation coefficient of the simulation data (blue dots) is given by $\rho = 0.89$. The 95%-confidence interval for the slope of the dashed red line is equal to $[3.4, 4.2] \cdot 10^{-9}$.

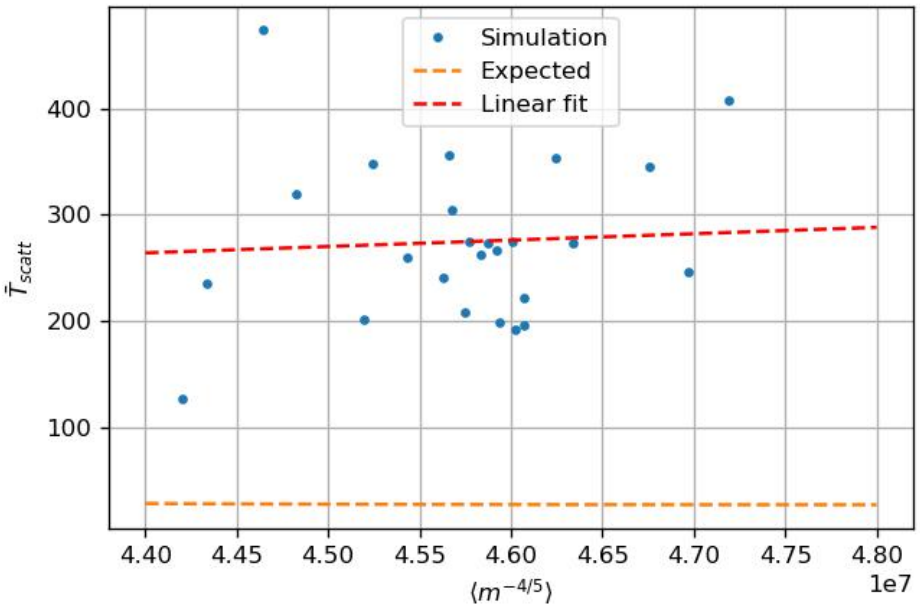


**Figure 4.6:** Average time between two scatterings $\bar{T}_{scatt}$ plotted against the average $\langle m^{4/5} \rangle$ value for 25 simulations, each with 100 performed scatterings. $m_{max}$ was varied within the interval $10^{[-8,-6]}\,\mathrm{M}_\odot$ to obtain these results. The correlation coefficient of the simulation data (blue dots) is given by $\rho = 0.16$. The 95%-confidence interval for the slope of the dashed red line is equal to $[5.3, 6.7] \cdot 10^{-6}\,\mathrm{yM}_\odot^{4/5}$.

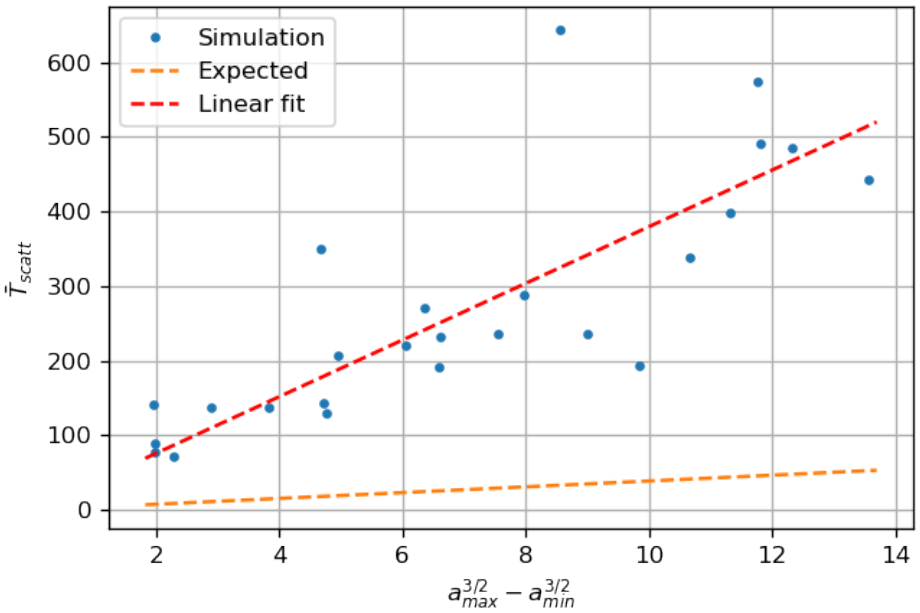


**Figure 4.7:** Average time between two scatterings $\bar{T}_{scatt}$ plotted against $a_{max}^{3/2} - a_{min}^{3/2}$ for 25 simulations, each with 100 performed scatterings. $a_{max}$ was varied within the interval [2,6] AU to obtain these results. The correlation coefficient of the simulation data (blue dots) is given by $\rho = 0.80$. The 95%-confidence interval for the slope of the dashed red line is equal to $[33, 43]\,\mathrm{yAU}_{\odot}^{-3/2}$.

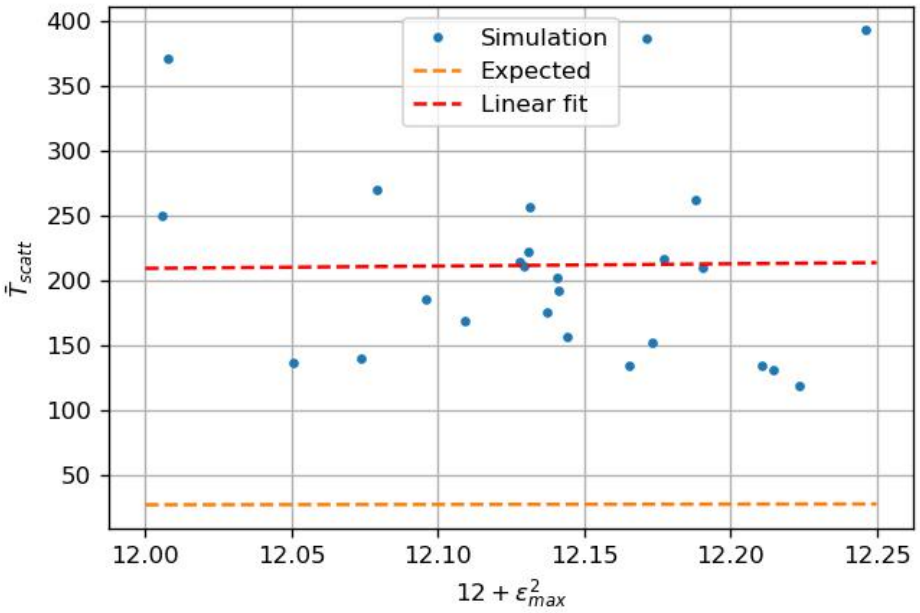


**Figure 4.8:** Average time between two scatterings $\bar{T}_{scatt}$ plotted against $12 + \epsilon^2_{max}$ for 25 simulations, each with 100 performed scatterings. $\epsilon_{max}$ was varied within the interval [0,0.5] to obtain these results. The correlation coefficient of the simulation data (blue dots) is given by $\rho = -0.11$. The 95%-confidence interval for the slope of the dashed red line is equal to [15,20].

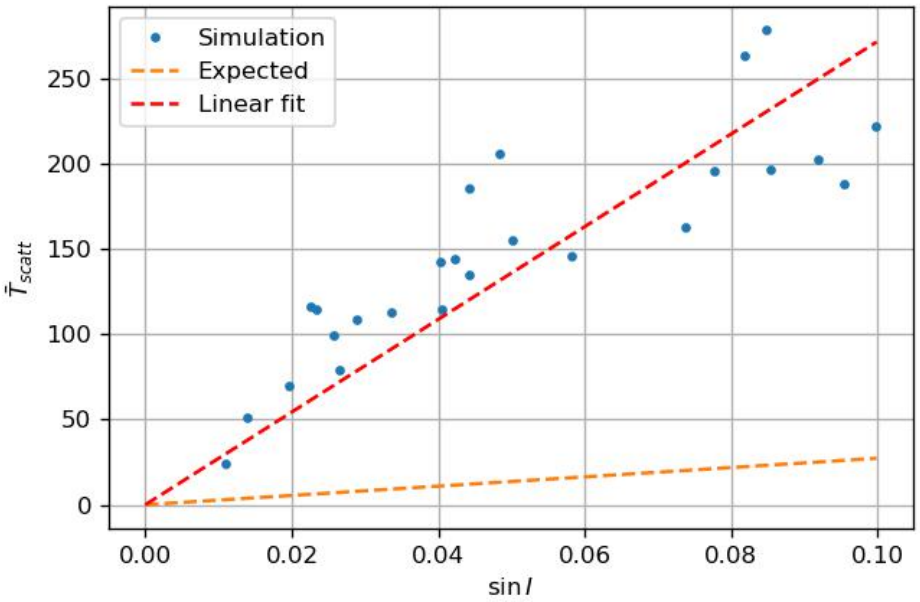


**Figure 4.9:** Average time between two scatterings $\bar{T}_{scatt}$ plotted against $\sin I_{max}$ for 25 simulations, each with 100 performed scatterings. $I_{max}$ was varied within the interval $[10^{-2}, 10^{-1}]$ to obtain these results. The correlation coefficient of the simulation data (blue dots) is given by $\rho = 0.87$. The 95%-confidence interval for the slope of the dashed red line is equal to $[2.4, 3.0] \cdot 10^3$.

## 4.3 Correcting the estimate

The correlation coefficients of the five parameters whose domains were varied are summarised in Table 4.1.

| Correlation coefficient | Value |
|---|---|
| $\rho(N^2, T_{scatt}^{-1})$ | 0.89 |
| $\rho(\langle m^{-4/5} \rangle, T_{scatt})$ | 0.16 |
| $\rho(a_{max}^{3/2} - a_{min}^{3/2}, T_{scatt})$ | 0.80 |
| $\rho(12 + \epsilon_{max}^2, T_{scatt})$ | -0.11 |
| $\rho(\sin I_{max}, T_{scatt})$ | 0.87 |

**Table 4.1:** Correlation coefficients of the five parameters whose domains were varied.

We note that only $N^2$, $a_{max}$ and $I_{max}$ give a relatively strong indication of the value of $T_{scatt}$ on their respective test domains, whereas $m_{max}$ and $\epsilon_{max}$ had almost no predictive value on their respective test domains. We will now improve our initial expected value $\mathbb{E}[T_{scatt}]$ by writing it as a function of $N^2$, $a_{max}$ and $I_{max}$ multiplied by a constant $K$:

$$\mathbb{E}[T_{scatt}] = K \frac{(a_{max}^{3/2} - a_{min}^{3/2}) \sin I_{max}}{N^2}, \tag{4.1}$$

where we note that this function would only be valid when $m_{max} = 10^{-7}\,\mathrm{M}_\odot$, $m_{min} = 10^{-10}\,\mathrm{M}_\odot$, $a_{min} = 1$ AU and $\epsilon_{max} = 0.1$, where each parameters is also distributed as in Table 3.1. Our goal now is to find a 95%-confidence interval for the value $K$. Using our results from Figures 4.5, 4.7 and 4.9, we can construct the following table:

| Parameter | Slope | 95%-confidence interval |
|---|---|---|
| $N^2$ | $[K(a_{max}^{3/2} - a_{min}^{3/2}) \sin I_{max}]^{-1}$ | $[2.4, 2.9] \cdot 10^8]$ |
| $a_{max}^{3/2} - a_{min}^{3/2}$ | $K \sin I_{max}/N^2$ | $[33, 43]$ |
| $\sin I_{max}$ | $K(a_{max}^{3/2} - a_{min}^{3/2})/N^2$ | $[2.4, 3.0] \cdot 10^3$ |

**Table 4.2:** 95%-confidence interval of the slope as expressed in the form of equation 4.1.

The resulting three 95%-confidence intervals for the value of $K$ are shown in Table 4.3

We are now able to conclude that the 95%-confidence interval of $K$ must be contained within $K = (3.8 \pm 0.5) \cdot 10^8$. Equation 4.1 can thus be written as:

$$\mathbb{E}[T_{scatt}] = (3.8 \pm 0.5) \cdot 10^8 \cdot \frac{(a_{max}^{3/2} - a_{min}^{3/2}) \sin I_{max}}{N^2}, \tag{4.2}$$

| Parameter | 95%-confidence interval for $K$ |
|---|---|
| $N^2$ | $[3.4, 4.2] \cdot 10^8$ |
| $a_{max}^{3/2} - a_{min}^{3/2}$ | $[3.3, 4.3] \cdot 10^8$ |
| $\sin I_{max}$ | $[3.4, 4.3] \cdot 10^8$ |

**Table 4.3:** 95%-confidence interval of the slope as expressed in the form of equation 4.1.

where we again note that this relation only holds on the domains as specified in Tables 3.1 and 3.2.

# 5. Discussion

Since the results of our simulations often did not correspond to the expected values, it is likely that our model, assumptions and calculations had shortcomings in them. Possible causes of these shortcomings will now be further elaborated upon. In further research, it would be interesting to try to adjust the model in order to overcome these shortcomings.

### Stationarity of the initialisation state

We see in Figure 4.1 that the time between the first scatterings is higher than the time between later scatterings, giving a root-like interaction between $n_{scatt}$ and $T_{scatt}$. However, it was tried to set up the system as a uniformly homogeneous disk, from which it would be expected that the evolution of the system was time-invariant and thus stationary. It is unclear why the system becomes more reactive (more scatterings per unit of time) over time. One cause could be that the average eccentricity increases as there are more scatterings, which would mean higher particle velocity and thus again more scatterings.

### Assumptions for the scattering time estimate

In Equation 2.22, it is implicitly assumed that all particles are allowed to move freely and independent of each other throughout the entire volume $V$, as if they are particles in an ideal gas. In a Kepler system, this is however not the case. Particles move in near-circular orbits around a heavy central mass, which means that particles that are close to each other will also have near-similar velocity vectors. These effects are not accounted for in the further deviation of our estimate for the scattering time in 2.32.

### Test domains for $m_{max}$ and $\epsilon_{max}$

In Figures 4.6 and 4.8 the expected $T_{scatt}$ is almost constant on the domain that was tested, since $\langle m^{-4/5} \rangle$ and $12 + \epsilon_{max}^2$ react relatively little to changes in $m_{max}$ and $\epsilon_{max}$, respectively. To observe a stronger connection between $\langle m^{-4/5} \rangle$ and $T_{scatt}$, it would likely have been more interesting to vary $m_{min}$ instead of $m_{max}$. However, since simulations of systems with a lot of very small masses sometimes did not reach 100 scatterings, it was chosen to keep $m_{min}$ constant instead of $m_{max}$ in order to maintain comparable data points. For the dependence of $T_{scatt}$ on $\epsilon_{max}$, it could still be interesting to look at what happens in the limit $\epsilon_{max} \downarrow 0$ or when $\epsilon_{max} \uparrow 1$, since this would greatly

alter the dynamics of the Kepler system. Verifying these effects was beyond the scope of this research.

### Validity of corrected scattering time estimate

For computational reasons, the simulation data with which the corrected estimate in Equation 4.2 was derived, was based on the average time between the first 100 scatterings of in a simulation for each of the 25 simulations. However, because the initialisation state is usually non-stationary, this means that we are in principle not able to extend this estimate to systems in which a different number of scatterings were executed, even if they where to have the exact same initialisation conditions.

### Linear approximations

In order to simplify the calculations that are needed to compute the evolution of the Kepler system, a lot of interactions are linearised, the most important being in the computation of the minimum orbit intersection distance in Equation 3.8 and the linearisation of the hyperbolic orbit in Equation 3.12. These approximations become worse for larger masses, since they cause greater scattering angles.

### Gravitational sphere of influence

In the derivation of the definition of the gravitational sphere of influence $s^{inf}$, it was assumed that a satellite enters a planet's sphere of influence. However, in our case, a planet with roughly the same mass enters its sphere of influence. Since we do not assume planets start orbiting each other (as would be the case for satellites), this should not have greatly affected our calculations.

### Collisions

In order to focus on the evolution of the system, we did not perform a proper collision when the distance between two planets was smaller than the sum of their radii. Instead, we acted as if the two planets collided elastically, which would mean they bounced off each other without losing energy. This is not physical, however it can be noted that these situations would imply scattering angles of more than 90 degrees, which almost never occurred.

### Long-range gravitational interactions

A next step in fully simulating the gravitational forces present in a Kepler system, would be to not only include the gravitational interactions of particles which are close to each other, but also the gravity between particles that are further away. Especially when the system contains one or multiple particles with a relatively large mass (such as Jupiter in our Solar System), such a mass would significantly affect the stationarity of orbits of smaller bodies on long timescales.

# 6. Conclusion

The aim of this research was to answer three questions about the simulation of gravitational scatterings in $N$-body Kepler systems. For clarity, the answers to these questions will be evaluated separately.

1. *Do gravitational scatterings have a significant impact on the evolution of a Kepler system?*

   In the five simulations of 500 scatterings on the standard domain it was found that the average scattering angle $\bar{\delta}$ was equal to 2.10 °. For the 50 largest scatterings, the average $\delta$ was equal to 15.2 °. This implies a significant change of orbit for the particles involved in the scattering. In order to make an accurate description of the evolution of a Kepler system, it is thus necessary to include scatterings. This is especially the case for systems with heavier particles such as planets in a Solar system.

2. *Does the simulated average time between two scatterings correspond with expected average?*

   While looking at the results in Figures 4.3 and 4.4, it becomes clear that the average scattering time from the simulations $\bar{T}_{scatt}$ is significantly higher than $\mathrm{E}[T_{scatt}]$ for both the total 500 scatterings (by a factor 3.9) and the last 200 scatterings (by a factor 2.2). While having the same order of magnitude, which indicates that the assumptions on which our derivations are likely to be at least partially correct, further research is needed to give a more accurate estimate of the time between two scatterings. Throughout all five standard simulations, the time between two scatterings first starts to decrease after initialisation ($\rho = -0.44$). The time between two scatterings did not significantly change after 300 scatterings and became stationary ($\rho = -0.08$).

3. *What is the effect of increasing the number of particles, the maximum particle mass, the maximum semi-major axis, the maximum eccentricity and the maximum inclination on the average time between two scatterings?* From Figures 4.6 and 4.8 it followed that we do not find a clear correlation between $\langle m^{-4/5} \rangle$ and $\bar{T}_{scatt}$, and between $12 + \epsilon_{max}^2$ and $\bar{T}_{scatt}$, which was also indicated by the relative stability of our estimate on the respective test domains. By omitting these factors in a new estimate based on our simulation data, we found the following expression for the expected time between two scatterings: $\mathrm{E}[T_{scatt}] = (3.8 \pm 0.5) \cdot 10^8 \cdot (a_{max}^{3/2} - a_{min}^{3/2}) N^{-2} \sin I_{max}$, where we again

note that this equation only holds on the domains as specified in Tables 3.1 and 3.2.

Shortcomings in the model and computations are likely attributed to possibly wrong assumptions for the scattering time estimate, the non-stationarity of the initialisation state, several minor other causes. In further research, it is recommended to try to find methods to decrease their impact on the results of the simulations.

# Bibliography


[1] P. M. Visser. Collision detection for $N$-body Kepler systems. *Astronomy & Astrophysics*, 669:A97, jan 2023. doi: 10.1051/0004-6361/202243754. URL `https://doi.org/10.10512F0004-6361F202243754`.

[2] Lasunncty. Diagram illustrating and explaining various terms in relation to Orbits of Celestial bodies. *Wikimedia Commons*, 2007.

[3] Irene Cavallari, Clara Grassi, Giovanni F. Gronchi, Giulio Baù, and Giovanni B. Valsecchi. A dynamical definition of the sphere of influence of the Earth. *Communications in Nonlinear Science and Numerical Simulation*, 119, 2023. doi: 10.1016/j.cnsns.2023.107091.

[4] G.A. Chebotarev. Gravitational spheres of the major planets, moon and sun. *Soviet Astronomy*, 7(5):618–622, 1964.

[5] D Souami, J Cresson, C Biernacki, and F Pierret. On the local and global properties of gravitational spheres of influence. *Monthly Notices of the Royal Astronomical Society*, 496(4):4287–4297, jun 2020. doi: 10.1093/mnras/staa1520.

[6] R. A. Adams and C. Essex. *Student Solutions Manual to Calculus: A Complete Course, Eighth Edition*, page 442. Pearson, 2014.

[7] T. Logsdon. *Orbital Mechanics: Theory and Applications*. Wiley, 1998.

[8] Don Davis, D. Durda, F. Marzari, Adriano Campo Bagatin, and Ricardo Gil-Hutton. Collisional Evolution of Small-Body Populations. *Asteroids III*, 01 2002.

[9] E. D. Charles and J. B. Tatum. The Convergence of Newton–Raphson Iteration with Kepler's Equation. *Celestial Mechanics and Dynamical Astronomy*, 69(4):357–372, 1997. doi: 10.1023/a:1008200607490.

# Appendices

# A. Derivations for Table 3.1

For each of the parameters $x \in \{m, a, \epsilon, I, \varpi, \Omega, M\}$, we want to create the distributions as specified in Section 2.4 in order to create a uniformly homogeneous disk of particles, where the masses of the particles are distributed similarly as in the Asteroid belt. This is done by setting the cumulative distribution function of each parameter equal to a random value $\xi$ chosen uniformly between 0 and 1:

$$P_X(x) = \mathrm{P}(X \leq x) = \int_{-\infty}^{x} p_X(u)du = \xi, \qquad \xi \sim \mathcal{U}[0,1]. \tag{A.1}$$

Using the probability density functions from Equations 2.19 through 2.21 and $p(\epsilon) = \frac{1}{\epsilon_{max}}$, we find:

$$P(m) = \int_{m_{min}}^{m} p(m')dm' = \left(\frac{m_{max}}{m}\right)^{\frac{2}{3}} \frac{m^{\frac{2}{3}} - m_{min}^{\frac{2}{3}}}{m_{max}^{\frac{2}{3}} - m_{min}^{\frac{2}{3}}}, \tag{A.2}$$

$$P(a) = \int_{a_{min}}^{a} p(a')da' = \frac{a^3 - a_{min}^3}{a_{max}^3 - a_{min}^3}, \tag{A.3}$$

$$P(\epsilon) = \int_{0}^{\epsilon} p(\epsilon')d\epsilon' = \frac{\epsilon}{\epsilon_{max}}, \tag{A.4}$$

$$P(I) = \int_{I}^{I_{max}} p(I')dI' = \frac{\cos I - \cos I_{max}}{1 - \cos I_{max}}. \tag{A.5}$$

Setting Equations A.2 through A.5 equal to $\xi$ and solving for $m, a, \epsilon$ and $I$ respectively, gives the expressions in Table 3.1. For $x \in \{\varpi, \Omega, M\}$, $x$ is chosen randomly from a uniform distribution between 0 and $2\pi$. Hence $x = 2\pi\xi$ for these parameters.

# B. Source code

```
import numpy as np
from numpy import linalg as LA
import math
from scipy.spatial.transform import Rotation
import random
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

M = 1 #solar mass
G = 39.48 #Newton's constant in AU^3*yr^-2*M^-1
S = 0.00465047 #solar radius in AU
MJ = 0.0009543 #Jupiter's mass in solar mass M
SJ = 0.00046732617 #Jupiter's radius in AU
m_min = 10**(-10) #minimial mass of particle in solar M, constant in all simulations
a_min = 1 #minimal semi-major axis of particle in AU ,constant in all simulations

class Particle(object):
    """Represents an asteroid or other astronomical particle in 3D space."""
    def __init__(self, p):
        self.t0 = p[0]
        self.a = p[1]
        self.c = p[2]
        self.s = p[3]
        self.m = p[4]
        self.r0vec = p[5]
        self.Lvec = p[6]
        self.eccvec = p[7]
        self.omega = p[8]
        self.v0vec = p[9]
    @property
    def Inclination(self):
        return np.arccos(self.Lvec[2]/LA.norm(self.Lvec))
    @property
    def ColRange(self): #gravitational sphere of influence
        return (self.m/M)**(2/5)*self.a
    @property
    def TimePeriod(self):
        return 2*np.pi/self.omega
    @property
    def LeftBound(self):
        return self.a-self.c-self.ColRange

class system:
    def __init__(self,N, m_max, a_max, ecc_max, inc_max): #N is number of particles in the system
        sys=Generate_sys(N, m_max, a_max, ecc_max, inc_max)
        self.particles=sys
        self.cl=pairlist(sys)
        self.tl=t_collision(self.cl)

def KeplToProps(t0,s,m,a,ecc,argper,ascnode,inc,E): #use after initialization
    omega = np.sqrt((G*M)/(a**3))
    b = np.sqrt(1-ecc**2)*a
    c = ecc*a
    L = m*omega*a*b
    r0vec = np.matmul(Rotation_orb(ascnode,inc,argper),np.array((a*np.cos(E)-c,b*np.sin(E),0)))
    v0vec = (omega*a/LA.norm(r0vec))*np.matmul(Rotation_orb(ascnode,inc,argper),np.array((-a*np.sin(E
     ),b*np.cos(E),0)))
    Lvec = L*np.matmul(Rotation_orb(ascnode,inc,argper),np.array((0,0,1)))
    eccvec = ecc*np.matmul(Rotation_orb(ascnode,inc,argper),np.array((1,0,0)))
    props = [t0,a,c,s,m,r0vec,Lvec,eccvec,omega,v0vec]
    return props

def CartToProps(t,s,m, rvec, vvec): #use after collision
    Lvec = m*np.cross(rvec,vvec)
    l = np.dot(Lvec,Lvec)/(G*M*m**2)
    r = LA.norm(rvec)
    eccvec = np.cross(vvec,Lvec)/(G*M*m)-rvec/r
    ecc = LA.norm(eccvec)
    a = l/(1-ecc**2)
    c = a*ecc
    omega = np.sqrt((G*M)/(a**3))
    if (ecc >= 1 or l <= (1+ecc)*S):
        print("Ejection needed")
    props = [t,a,c,s,m,rvec,Lvec,eccvec,omega,vvec]
    return props

def MOID(p1,p2): #determine minimal orbit intersection distance between two particles, returns [p1,p2
    ,[d_MOID,r1,r2,v1,v2]]
    Kvec = np.cross(p1.Lvec,p2.Lvec)
    K = LA.norm(Kvec)
```

```
    l1 = np.dot(p1.Lvec,p1.Lvec)/(G*M*p1.m**2)
    r1vec_plus = (Kvec*l1)/(K+np.dot(p1.eccvec,Kvec))
    r1vec_minus = (Kvec*l1)/(-K+np.dot(p1.eccvec,Kvec))
    r1_plus = LA.norm(r1vec_plus)
    r1_minus = LA.norm(r1vec_minus)
    v1vec_plus = np.cross((p1.Lvec/(p1.m*l1)),(p1.eccvec+r1vec_plus/r1_plus))
    v1vec_minus = np.cross((p1.Lvec/(p1.m*l1)),(p1.eccvec+r1vec_minus/r1_minus))
    l2 = np.dot(p2.Lvec,p2.Lvec)/(G*M*p2.m**2)
    r2vec_plus = (Kvec*l2)/(K+np.dot(p2.eccvec,Kvec))
    r2vec_minus = (Kvec*l2)/(-K+np.dot(p2.eccvec,Kvec))
    r2_plus = LA.norm(r2vec_plus)
    r2_minus = LA.norm(r2vec_minus)
    v2vec_plus = np.cross((p2.Lvec/(p2.m*l2)),(p2.eccvec+r2vec_plus/r2_plus))
    v2vec_minus = np.cross((p2.Lvec/(p2.m*l2)),(p2.eccvec+r2vec_minus/r2_minus))
    dvec_plus = r2vec_plus - r1vec_plus
    wvec_plus = np.cross(v1vec_plus,v2vec_plus)
    w_plus = LA.norm(wvec_plus)
    r1vec_MOID_plus = r1vec_plus + (np.dot(dvec_plus,(np.cross(v2vec_plus,wvec_plus)/(w_plus**2))))*
     v1vec_plus
    r2vec_MOID_plus = r2vec_plus + (np.dot(dvec_plus,(np.cross(v1vec_plus,wvec_plus)/(w_plus**2))))*
     v2vec_plus
    dvec_MOID_plus = r2vec_MOID_plus - r1vec_MOID_plus
    d_MOID_plus = LA.norm(dvec_MOID_plus)
    dvec_minus = r2vec_minus - r1vec_minus
    wvec_minus = np.cross(v1vec_minus,v2vec_minus)
    w_minus = LA.norm(wvec_minus)
    r1vec_MOID_minus = r1vec_minus + (np.dot(dvec_minus,(np.cross(v2vec_minus,wvec_minus)/(w_minus
     **2))))*v1vec_minus
    r2vec_MOID_minus = r2vec_minus + (np.dot(dvec_minus,(np.cross(v1vec_minus,wvec_minus)/(w_minus
     **2))))*v2vec_minus
    dvec_MOID_minus = r2vec_MOID_minus - r1vec_MOID_minus
    d_MOID_minus = LA.norm(dvec_MOID_minus)
    arr = np.array([], dtype = object)
    if d_MOID_plus < max(p1.ColRange,p2.ColRange):
        arr = np.append(arr,np.array([p1,p2,np.array([d_MOID_plus,r1vec_plus,r2vec_plus,v1vec_plus,
     v2vec_plus],dtype = object)], dtype=object))
        arr = arr.reshape(int(len(arr)/3),3)
    if d_MOID_minus < max(p1.ColRange,p2.ColRange):
        arr = np.append(arr,np.array([p1,p2,np.array([d_MOID_minus,r1vec_minus,r2vec_minus,
     v1vec_minus,v2vec_minus],dtype = object)], dtype=object))
        arr = arr.reshape(int(len(arr)/3),3)
    if min(d_MOID_plus,d_MOID_minus) < max(p1.ColRange,p2.ColRange):
        return (arr)

def Rx(theta):
    return np.array(([1,0,0],[0,np.cos(theta),-np.sin(theta)],[0,np.sin(theta),np.cos(theta)]))
def Rz(phi):
    return np.array(([np.cos(phi),-np.sin(phi),0],[np.sin(phi),np.cos(phi),0],[0,0,1]))
def Rotation_orb(ascnode,inc,argper): #rotate orbital parameters
    RxRz = np.matmul(Rx(inc),Rz(argper))
    RzRxRz = np.matmul(Rz(ascnode),RxRz)
    return(RzRxRz)

def Rotate_rel(m1,m2,dvec_i,uvec_i): #rotate relative coordinate over angle delta in CM frame
    b = LA.norm(dvec_i)
    u = LA.norm(uvec_i)
    a = (G*(m1+m2))/(u**2)
    c = np.sqrt(a**2 + b**2)
    delta = 2*np.arctan(a/b)
    dvec_f = ((2*a*b**2)/(u*c**2))*uvec_i + ((b**2-a**2)/(c**2))*dvec_i
    uvec_f = ((b**2-a**2)/(c**2))*uvec_i - ((2*a*u)/(c**2))*dvec_i
    return (dvec_f,uvec_f,delta)

def ElCollision(m1,m2,r1vec_i,r2vec_i,v1vec_i,v2vec_i): #perform a scattering between two particles
    uvec_i = v2vec_i - v1vec_i
    dvec_i = r2vec_i - r1vec_i - ((np.dot((r2vec_i - r1vec_i),uvec_i))/(LA.norm(uvec_i)**2))*uvec_i
    Rvec_i = (m1*r1vec_i+m2*r2vec_i)/(m1+m2)
    Vvec_i = (m1*v1vec_i + m2*v2vec_i)/(m1+m2)
    dvec_f,uvec_f,delta = Rotate_rel(m1,m2,dvec_i,uvec_i)
    r1vec_f = Rvec_i - (m2/(m1+m2))*dvec_f
    r2vec_f = Rvec_i + (m1/(m1+m2))*dvec_f
    v1vec_f = Vvec_i - (m2/(m1+m2))*uvec_f
    v2vec_f = Vvec_i + (m1/(m1+m2))*uvec_f
    return(r1vec_f, r2vec_f, v1vec_f,v2vec_f,delta)

#some mathematical functions
def calculate_eccentric_anomaly(mean_anomaly, eccentricity):
    E = mean_anomaly
    for i in range(3):  # Perform three iterations
        E = E - (E - eccentricity * math.sin(E) - mean_anomaly) / (1 - eccentricity * math.cos(E))
    return E
def E_M_45_func(m_max):
```

```
    return (5/11)*((m_max**(22/15)-m_min**(22/15))/(m_max**(22/15)*m_min**(4/5)-m_max**(4/5)*m_min**(22/15)))
def Exp_value_45(mass_array):
    mass45_array = np.zeros(len(mass_array))
    for i in range(len(mass_array)):
        mass45_array[i] = mass_array[i]**(-4/5)
    exp_value_45 = np.mean(mass45_array)
    return exp_value_45
def E_Tscatt(N, m_max, a_max, ecc_max, inc_max):
    return ((4/9)*M**(3/10)*(a_max**(3/2)-a_min**(3/2))*(12+ecc_max**2)*np.sin(inc_max)*E_M_45_func(m_max))/(np.sqrt(G)*N**2)
def Line(x_data, a):
    return x_data*a

def Generate_sys(N, m_max, a_max, ecc_max, inc_max): #generate random system of N particles
    arr = np.array([],dtype = object)
    leftbound = np.array([])
    timeperiod = np.array([])
    for n in range(N):
        t0 = 0
        m = m_min/((1-random.uniform(0,1)*(1-(m_min/m_max)**(2/3)))**(3/2))
        s = SJ*(m/MJ)**(1/3)
        a = np.cbrt(random.uniform(0,1)*(a_max**3-a_min**3)+a_min**3)
        ecc = random.uniform(0,ecc_max)
        inc = np.arccos(random.uniform(0,1)*(1-np.cos(inc_max))+np.cos(inc_max))
        argper,ascnode,mean_anomaly = random.uniform(0,2*np.pi),random.uniform(0,2*np.pi),random.uniform(0,2*np.pi)
        E = calculate_eccentric_anomaly(mean_anomaly, ecc)
        props = KeplToProps(t0,s,m,a,ecc,argper,ascnode,inc,E)
        particle = Particle(props)
        arr = np.append(arr,particle)
        leftbound = np.append(leftbound, particle.LeftBound)
        timeperiod = np.append(timeperiod, particle.TimePeriod)
    print("Min time period is ", min(timeperiod))
    print("Max time period is ", max(timeperiod))
    arr = arr[leftbound.argsort()]
    return arr

def pairlist(sys): #create list of scattering pairs
    arr = np.array([],dtype = object)
    for i in range(len(sys)):
        for j in range(i+1,len(sys)):
            if sys[i].a+sys[i].c+sys[i].ColRange >= sys[j].a-sys[j].c-sys[j].ColRange:
                k = MOID(sys[i],sys[j])
                if type(k)!=type(None):
                    arr = np.append(arr,k)
                else:
                    break
    return arr.reshape(int(len(arr)/3),3)

def newpairs(sys,p1new,p2new): #compute new scatterings
    arr = np.array([],dtype = object)
    for i in range(len(sys)):
        if sys[i].a+sys[i].c+sys[i].ColRange >= p1new.a-p1new.c-p1new.ColRange:
            k = MOID(sys[i],p1new)
            if type(k)!=type(None):
                arr = np.append(arr,k)
        if sys[i].a+sys[i].c+sys[i].ColRange >= p2new.a-p2new.c-p2new.ColRange:
            k = MOID(sys[i],p2new)
            if type(k)!=type(None):
                arr = np.append(arr,k)
    return arr.reshape(int(len(arr)/3),3)

def t_collision(cl): #Calculate list of scatttering times
    arr = np.array([], dtype=object)
    for i in range(len(cl)):
        p1 = cl[i][0]
        p2 = cl[i][1]
        k = cl[i][2]
        e1sq, e2sq = np.dot(p1.eccvec,p1.eccvec), np.dot(p2.eccvec,p2.eccvec)
        r1,r2 = LA.norm(k[1]),LA.norm(k[2])
        z1 = np.dot(k[1]/p1.a-1j*(r1*k[3]/(p1.a**2*p1.omega)),(p1.r0vec-p1.eccvec*np.dot(p1.eccvec,p1.r0vec))/(p1.a-e1sq*p1.a)+p1.eccvec)+(np.dot(p1.r0vec,p1.eccvec))/p1.a+e1sq
        z2 = np.dot(k[2]/p2.a-1j*(r2*k[4]/(p2.a**2*p2.omega)),(p2.r0vec-p2.eccvec*np.dot(p2.eccvec,p2.r0vec))/(p2.a-e2sq*p2.a)+p2.eccvec)+(np.dot(p2.r0vec,p2.eccvec))/p2.a+e2sq
        DE1, DE2 = np.angle(z1)%(2*np.pi), np.angle(z2)%(2*np.pi)
        t11 = p1.t0+DE1/p1.omega-np.dot(np.cross(p1.eccvec,k[1]-p1.r0vec)/(1-e1sq),p1.Lvec/(G*M*p1.m))
        t21 = p2.t0+DE2/p2.omega-np.dot(np.cross(p2.eccvec,k[2]-p2.r0vec)/(1-e2sq),p2.Lvec/(G*M*p2.m))
        uvec, wsq = k[4]-k[3], np.dot(np.cross(k[3],k[4]),np.cross(k[3],k[4]))
        delta = (1/np.abs(t11-t21))*np.sqrt(np.dot(uvec,uvec)*(((p1.ColRange+p2.ColRange)**2-k[0]**2)/wsq))
```

```
        q0,q1,k0,k1 = 2*np.pi/(p1.omega*np.abs(t11-t21)),2*np.pi/(p2.omega*np.abs(t11-t21)),1,0
        q,alpha,ks,n= np.array([q0,q1]),np.array([]),np.array([k0,k1]),0
        while n <=8:
                q, alpha, ks = np.append(q,[0,0]), np.append(alpha, [0,0]), np.append(ks, [0,0])
                alpha[2*n] = np.floor(q[2*n]/q[2*n+1])
                q[2*n+2] = q[2*n]-alpha[2*n]*q[2*n+1]
                if q[2*n+2] == 0:
                    break
                ks[2*n+2] = ks[2*n] -alpha[2*n]*ks[2*n+1]
                alpha[2*n+1] = np.floor(q[2*n+1]/q[2*n+2])
                q[2*n+3] = q[2*n+1]-alpha[2*n+1]*q[2*n+2]
                if q[2*n+3] == 0:
                    break
                ks[2*n+3] = ks[2*n+1] -alpha[2*n+1]*ks[2*n+2]
                xmax = np.floor((1+delta)/q[2*n+2])+1
                if xmax > 1000000:
                    break
                x = np.arange(np.ceil((1-delta)/q[2*n]),xmax,1)
                y = np.maximum(np.zeros(len(x)), np.ceil(q[2*n]*x/q[2*n+1]-(1+delta)/q[2*n+1]))
                for xi in range(len(x)):
                    if x[xi]*q[2*n]-y[xi]*q[2*n+1] > 1- delta:
                        sol = x[xi]*ks[2*n]-y[xi]*ks[2*n+1]
                        tnew = t11+2*np.pi*sol/p1.omega
                        arr = np.append(arr,np.array([p1,p2,tnew,k[1:]], dtype=object))
                        n = 10
                        break
                n=n+1
    arr = arr.reshape(int(len(arr)/4),4)
    return arr[(arr[:,2]).argsort()]

def main(N,m_max, a_max, ecc_max, inc_max, simlen):
    sys = system(N, m_max, a_max, ecc_max, inc_max)
    particles = sys.particles
    mass_array = np.zeros(N)
    for i in range(N):
        mass_array[i]=particles[i].m
    cl = sys.cl
    tl = sys.tl
    tarr = np.array([0])
    NofCol = 0
    delta_arr = np.array([])
    leftbound_arr = np.zeros(len(particles))
    for i in range(len(particles)):
        leftbound_arr[i] = particles[i].LeftBound
    print("Initial shape of time list is ", np.shape(tl))
    while tarr[-1] <1e6 and not len(tl) == 0 and NofCol < simlen:
        p1 = tl[0][0]
        p2 = tl[0][1]
        tkl = tl[0][2]
        r12v12_i = tl[0][3]
        dvec_kl = r12v12_i[1]-r12v12_i[0]
        uvec_i = r12v12_i[3]-r12v12_i[2]
        t_scatt = tkl - (np.dot(dvec_kl,uvec_i))/(LA.norm(uvec_i)**2)
        tarr = np.append(tarr, t_scatt)
        r12v12_f = ElCollision(p1.m,p2.m,r12v12_i[0],r12v12_i[1],r12v12_i[2],r12v12_i[3])
        p1new = Particle(CartToProps(t_scatt,p1.s,p1.m,r12v12_f[0],r12v12_f[2]))
        p2new = Particle(CartToProps(t_scatt,p2.s,p2.m,r12v12_f[1],r12v12_f[3]))
        delta_arr = np.append(delta_arr, r12v12_f[4])
        boolean_particles = [False if (par == p1 or par == p2) else True for par in particles]
        boolean_particles_np = np.array(boolean_particles)
        index_scatt_par = np.where(boolean_particles_np == False)
        particles = particles[boolean_particles]
        leftbound_arr = leftbound_arr[boolean_particles]
        boolean_tl = [False if (p1 == col[0] or p2 == col[0] or p1 == col[1] or p2 == col[1]) else
 True for col in tl]
        tl = tl[boolean_tl]
        ncl = newpairs(particles,p1new,p2new)
        ntl = t_collision(ncl)
        merged_tl=np.append(tl,ntl)
        merged_tl = merged_tl.reshape(int(len(merged_tl)/4),4)
        tl = merged_tl[(merged_tl[:,2]).argsort()]
        particles = np.append(particles, (p1new,p2new))
        leftbound_arr = np.append(leftbound_arr, (p1new.LeftBound, p2new.LeftBound))
        particles = particles[leftbound_arr.argsort()]
        NofCol += 1
        print(NofCol)
    print("Number of collisions is ", NofCol)
    print("End shape of time list is ", np.shape(tl))
    sys.particles = particles
    scatt_arr = np.zeros(len(tarr)-1)
    for i in range(len(tarr)-1):
        scatt_arr[i] = (tarr[i+1]-tarr[i])
    return (tarr, delta_arr, scatt_arr,mass_array)
```

```
N = 1000
m_max = 10**(-7)
a_max = 4
ecc_max = 0.1
inc_max = 0.1
simlen = 500
tarr, delta_arr, scatt_arr, mass_array = main(N, m_max, a_max, ecc_max, inc_max, simlen)
Nsim=5 #execute 5 simulations of 500 scatterings
tarr_avg = np.zeros(simlen+1)
for i in range(simlen+1):
    tarr_avg[i] = (tarr1[i]+tarr2[i]+tarr3[i]+tarr4[i]+tarr5[i])/Nsim
delta_sum = np.concatenate((delta1_arr,delta2_arr,delta3_arr,delta4_arr, delta5_arr))
scatt_avg = np.zeros(len(tarr_avg)-1)
for i in range(len(tarr_avg)-1):
    scatt_avg[i] = (tarr_avg[i+1]-tarr_avg[i])
mean_scatt = np.round(np.mean(scatt_avg),2)
mean_scatt_300_500 = np.round(np.mean(scatt_avg[300:499]))
ETscatt = E_Tscatt(N, m_max, a_max, ecc_max, inc_max)

def N_func(Nsim,m_max, a_max, ecc_max, inc_max, simlen): #used for Figure 4.5
    delta_avg_arr = np.zeros(Nsim)
    scatt_avg_arr = np.zeros(Nsim)
    N_arr = np.zeros(Nsim)
    sim = 0
    while sim < Nsim:
        N = int(np.sqrt(10**4 + (10**6-10**4)*random.uniform(0,1)))
        N_arr[sim] = N
        delta_arr, scatt_arr = main(N, m_max, a_max, ecc_max, inc_max, simlen)[1:3]
        delta_avg_arr[sim] = np.mean(delta_arr)
        scatt_avg_arr[sim] = np.mean(scatt_arr)
        sim += 1
        print(sim)
    return N_arr, delta_avg_arr, scatt_avg_arr
N_arr, delta_avg_N_arr, scatt_avg_N_arr = N_func(Nsim,m_max, a_max, ecc_max, inc_max, simlen)
N_bar_arr = np.zeros(len(N_arr))
for i in range(len(N_arr)):
    N_bar_arr[i] = N_arr[i]**(2)
E_N2 = np.linspace(10**5,10**6,100)
E_N2inv = np.zeros(100)
E_N = np.zeros(100)
E_Tscatt_N = np.zeros(100)
for i in range(len(E_N)):
    E_N2inv[i] = 1/E_N2[i]
    E_N[i] = int(np.sqrt(E_N2[i]))
    E_Tscatt_N[i] = E_Tscatt(E_N[i], m_max, a_max, ecc_max, inc_max)
values, covariance = curve_fit(Line, N_bar_arr, 1/scatt_avg_N_arr)
a_fit = values[0]
y = Line(E_N2,a_fit)

def M_max(Nsim,N,a_max, ecc_max, inc_max, simlen): #used for Figure 4.6
    delta_avg_arr = np.zeros(Nsim)
    scatt_avg_arr = np.zeros(Nsim)
    m_max_arr = np.zeros(Nsim)
    exp_value_45_arr = np.zeros(Nsim)
    sim = 0
    while sim < Nsim:
        xi = random.uniform(0,1)
        m_max_arr[sim] = 10**(-8+2*random.uniform(0,1))
        delta_arr, scatt_arr, mass_array = main(N, m_max, a_max, ecc_max, inc_max, simlen)[1:4]
        delta_avg_arr[sim] = np.mean(delta_arr)
        scatt_avg_arr[sim] = np.mean(scatt_arr)
        exp_value_45_arr[sim] = Exp_value_45(mass_array)
        sim += 1
        print(sim)
    return m_max_arr, delta_avg_arr, scatt_avg_arr, exp_value_45_arr
m_max_arr, delta_avg_m_max_arr, scatt_avg_m_max_arr, exp_value_45_arr = M_max(Nsim,N,a_max, ecc_max,
     inc_max, simlen)
E_m_max = 10**(np.linspace(-8,-6,100))
E_M_45 = np.zeros(100)
E_M_45_2 = np.linspace(4.4*10**7, 4.8*10**7,100)
E_Tscatt_m = np.zeros(100)
for i in range(len(E_m_max)):
    E_M_45[i] = E_M_45_func(E_m_max[i])
    E_Tscatt_m[i] = E_Tscatt(N, E_m_max[i], a_max, ecc_max, inc_max)
values, covariance = curve_fit(Line, exp_value_45_arr, scatt_avg_m_max_arr)
a_fit = values[0]
y = Line(E_M_45_2,a_fit)

def A_max(Nsim,N,m_max, ecc_max, inc_max, simlen): #used for Figure 4.7
    delta_avg_arr = np.zeros(Nsim)
    scatt_avg_arr = np.zeros(Nsim)
    a_max_arr = np.zeros(Nsim)
    sim = 0
```

```
    while sim < Nsim:
        a_max = 2 + 4*random.uniform(0,1)
        a_max_arr[sim] = a_max
        delta_arr, scatt_arr = main(N, m_max, a_max, ecc_max, inc_max, simlen)[1:3]
        delta_avg_arr[sim] = np.mean(delta_arr)
        scatt_avg_arr[sim] = np.mean(scatt_arr)
        sim += 1
        print(sim)
    return a_max_arr, delta_avg_arr, scatt_avg_arr
a_max_arr, delta_avg_a_max_arr, scatt_avg_a_max_arr = A_max(Nsim,N,m_max, ecc_max, inc_max, simlen)
a_bar_arr = np.zeros(len(a_max_arr))
for i in range(len(a_max_arr)):
    a_bar_arr[i] = a_max_arr[i]**(3/2)-a_min**(3/2)
E_a_max = np.linspace(2,6,100)
E_a_bar = np.zeros(100)
E_Tscatt_a = np.zeros(100)
for i in range(len(E_m_max)):
    E_a_bar[i] = E_a_max[i]**(3/2)-a_min**(3/2)
    E_Tscatt_a[i] = E_Tscatt(N, m_max, E_a_max[i], ecc_max, inc_max)
values, covariance = curve_fit(Line, a_bar_arr, scatt_avg_a_max_arr)
a_fit = values[0]
y = Line(E_a_bar,a_fit)

def Ecc_max(Nsim,N,m_max,a_max, inc_max, simlen): #used for Figure 4.8
    delta_avg_arr = np.zeros(Nsim)
    scatt_avg_arr = np.zeros(Nsim)
    ecc_max_arr = np.zeros(Nsim)
    sim = 0
    while sim < Nsim:
        ecc_max = 0.5*np.sqrt(random.uniform(0,1))
        ecc_max_arr[sim] = ecc_max
        delta_arr, scatt_arr = main(N, m_max, a_max, ecc_max, inc_max, simlen)[1:3]
        delta_avg_arr[sim] = np.mean(delta_arr)
        scatt_avg_arr[sim] = np.mean(scatt_arr)
        sim += 1
        print(sim)
    return ecc_max_arr, delta_avg_arr, scatt_avg_arr
ecc_max_arr, delta_avg_ecc_max_arr, scatt_avg_ecc_max_arr = Ecc_max(Nsim,N,m_max, a_max, inc_max,
     simlen)
ecc_max_bar_arr = np.zeros(len(ecc_max_arr))
for i in range(len(ecc_max_arr)):
    ecc_max_bar_arr[i] = 12+ecc_max_arr[i]**(2)
E_ecc_max = np.linspace(0,0.5,100)
E_ecc_bar = np.zeros(100)
E_Tscatt_ecc = np.zeros(100)
for i in range(len(E_ecc_max)):
    E_ecc_bar[i] = 12+E_ecc_max[i]**2
    E_Tscatt_ecc[i] = E_Tscatt(N, m_max, a_max, E_ecc_max[i], inc_max)
values, covariance = curve_fit(Line, ecc_max_bar_arr, scatt_avg_ecc_max_arr)
a_fit = values[0]
y = Line(E_ecc_bar,a_fit)

def Inc_max(Nsim,N,m_max,a_max, ecc_max, simlen): #used for Figure 4.9
    delta_avg_arr = np.zeros(Nsim)
    scatt_avg_arr = np.zeros(Nsim)
    inc_max_arr = np.zeros(Nsim)
    sim = 0
    while sim < Nsim:
        inc_max = 10**(-2) + (10**(-1)-10**(-2))*random.uniform(0,1)
        print("Inc is ", inc_max)
        inc_max_arr[sim] = inc_max
        delta_arr, scatt_arr = main(N, m_max, a_max, ecc_max, inc_max, simlen)[1:3]
        delta_avg_arr[sim] = np.mean(delta_arr)
        scatt_avg_arr[sim] = np.mean(scatt_arr)
        sim += 1
        print(sim)
    return inc_max_arr, delta_avg_arr, scatt_avg_arr
inc_max_arr, delta_avg_inc_max_arr, scatt_avg_inc_max_arr = Inc_max(Nsim,N,m_max, a_max, ecc_max,
     simlen)
inc_max_bar_arr = np.zeros(len(inc_max_arr))
for i in range(len(inc_max_arr)):
    inc_max_bar_arr[i] = np.sin(inc_max_arr[i])
E_inc_max = np.linspace(10**(-2),10**(-1),100)
E_inc_bar = np.zeros(100)
E_Tscatt_inc = np.zeros(100)
for i in range(len(E_inc_max)):
    E_inc_bar[i] = np.sin(E_inc_max[i])
    E_Tscatt_inc[i] = E_Tscatt(N, m_max, a_max, ecc_max, E_inc_max[i])
values, covariance = curve_fit(Line, inc_max_bar_arr, scatt_avg_inc_max_arr)
a_fit = values[0]
y = Line(E_inc_bar,a_fit)
```