Collision detection for N-body Kepler systems

P.M. Visser

Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
e-mail: p.m.visser@tudelft.nl

ABSTRACT

Context. In a Keplerian system, a large number of bodies orbit a central mass. Accretion disks, protoplanetary disks, asteroid belts, and planetary rings are examples. Simulations of these systems require algorithms that are computationally efficient. The inclusion of collisions in the simulations is challenging but important.

Aims. We intend to calculate the time of collision of two astronomical bodies in intersecting Kepler orbits as a function of the orbital elements. The aim is to use the solution in an analytic propagator (N-body simulation) that jumps from one collision event to the next.

Methods. We outline an algorithm that maintains a list of possible collision pairs ordered chronologically. At each step (the soonest event on the list), only the particles created in the collision can cause new collision possibilities. We estimate the collision rate, the length of the list, and the average change in this length at an event, and study the efficiency of the method used.

Results. We find that the collision-time problem is equivalent to finding the grid point between two parallel lines that is closest to the origin. The solution is based on the continued fraction of the ratio of orbital periods.

Conclusions. Due to the large jumps in time, the algorithm can beat tree codes (octree and k-d tree codes can efficiently detect collisions) for specific systems such as the Solar System with $N < 10^8$. However, the gravitational interactions between particles can only be treated as gravitational scattering or as a secular perturbation, at the cost of reducing the time-step or at the cost of accuracy. While simulations of this size with high-fidelity propagators can already span vast timescales, the high efficiency of the collision detection allows many runs from one initial state or a large sample set, so that one can study statistics.

Key words. gravitation – methods: analytical – methods: statistical – celestial mechanics – planets and satellites: formation – protoplanetary disks

1. Introduction

Simulations of the mechanical motion of many bodies are generally computationally expensive. Consider the N-body problem. Here, each of the N particles moves under the influence of the gravitation of all other particles and each particle can collide with any other particle. Therefore, there are N^2 interactions to account for. Various methods have been invented to speed up the simulation or increase the particle count: (i) direct N-body simulations with dynamic time-steps (see Dehnen & Read 2011, for an overview); (ii) the octree code for collision detection (Bentley 1975; Meagher 1980); (iii) the *Barnes-Hut algorithm* (Barnes & Hut 1986; Barnes 1990; Hamada et al. 2009; Burtscher & Pingali 2011) for mutual gravity, where nearby particles are grouped so that their effect on a distant particle can be combined, which requires $O(N \log N)$ computational steps; (iv) the fast multi-pole *Greengard and Rokhlin method* (FMM), where higher order moments of the particle groups are included (Rokhlin 1985; Greengard 1990); (v) parallelization of these methods (Warren & Salmon 1993); (vi) particle mesh methods, where the N force vectors are calculated using the Newton potential and the Poisson Equation for the potential is solved numerically with fast Fourier transforms (Bodenheimer et al. 2007); (vii) the finite-elements method (FEM); and finally, (viii) for a Keplerian system (with a large central mass), where the particles move in slowly precessing Kepler ellipses described by the *Laplace-Lagrange equations* for the orbital elements (Murray & Dermott 2009); because the Kepler ellipses change slowly over

time, the time-step in these numerical integration propagators can be many orbital periods.

In astronomy, collision detection is the problem of finding the precise moment at which asteroids, planets, or satellites collide. Here the difficulty in predicting collisions, or calculating the collision probability, stems from the fact that the objects are very small compared to the size of their orbits. In numerical simulations the number of nearby particles that need to be considered is a function of the step size. Because the particles travel during each time-step, the volume of space around the particle that needs to be probed for collision partners has a radius of the time-step times velocity. In order to limit the number of collision partners, this volume needs to remain small. Therefore, the step size decreases as $O(N^{-1/3})$ and the total number of steps for a fixed simulation time grows as $O(N^{1/3})$. Efficient codes, such as octree codes (Meagher 1982) or spatial hashing codes, have an algorithmic efficiency of $O(N \log N)$ per time-step. If these are used in collision detection, the number of steps grows as $O(N^{4/3} \log N)$. This makes the problem of collision detection in astronomy even more challenging than pure gravitational evolution without collisions (see Dehnen & Read 2011, for a comparison between codes with and without collision detection).

In this paper, we apply collision detection to Keplerian systems, such as astrophysical disks, where all particles feel a dominant gravity force from one heavy central mass. Each particle is in a Kepler orbit given by parameters a, ϵ, ϖ, I, Ω, and ν (see Table 1 for the symbols). However, the advantage of implementing collision detection in an analytic propagator is that the algorithm can be very efficient. In Sect. 2, we analyze the timescales, eval-
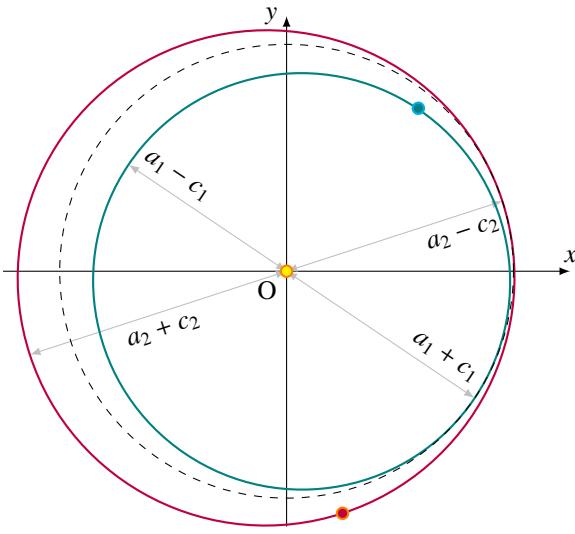
Fig. 1. Two orbits separated by a sphere (*dashed*). Orbit 1 (*blue*) has apoapsis $a_1 + c_1$ and orbit 2 (*purple*) has periapsis $a_2 - c_2$. Filtering out such collision-avoiding apoapsis–periapsis pairs is an efficient sweep and prune method.

uate the numerical efficiency, and compare it with algorithms based on tree code. As there is no numerical integration of the orbits, many physical effects are neglected (see Sect. 2.3).

We describe (in Sect. 3) the algorithm for an N-body code with collision detection in detail. Initially, it compares particles sorted by radial distance using equations from the seminal paper by Öpik (1951); see Fig. 1. This is effectively an implementation of the apoapsis/periapsis filter of Hoots et al. (1984) and an example of a sweep and prune method. The algorithm then uses analytic evaluation of the points of collision (near the nodal line in Fig. 2, following Hoots et al. 1984; Manley et al. 1998, as explained in Sect. 4), of the earliest crossing time (Sect. 5), and of the time of collision (derived in Sect. 6). The algorithm keeps track of pairs of particles that are on a collision course, from the earliest to the latest moment of collision. Each step of the simulation involves only the calculation of the next collision, and updating the list. In the method, time-steps increase with decreasing s, which allows long simulation times. Indeed, for the limiting case $s \longrightarrow 0$, the algorithm stops after initialization, as it finds that there are no collisions. In contrast, collision detection using a numerical integration propagator always requires a nearest-neighbor search for every particle. The time spent on this search is independent of s.

To our knowledge, the idea of bookkeeping a list of future possible collisions has not been studied elsewhere. The algorithm relies on a novel method to quickly find the exact collision times. We derived new expressions, Eqs. (11) and (12), for the difference in the eccentric anomaly between two given points on an orbit that are also accurate at small eccentricities, when the eccentric anomalies themselves are ill defined. These formulas were needed to calculate the time for a particle to get to the collision point.

2. Applications and estimates of timescales

The algorithm is intended for simulation of the dynamical evolution of a large planetary ring system or a debris disk around a star. In the latter, nearby passages also occur, where one planet is inside the sphere of gravitational influence of another. The re-

Table 1. List of symbols and notation.

symbol	quantity
t	time
$\mathrm{d}t$	time-step, small time interval
x, y, z	Cartesian coordinates
V	volume
$h = \Delta a$	thickness of spherical shell
$\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$	position vector
r	radial distance
$\boldsymbol{v} = \dot{\boldsymbol{r}}$	velocity vector
v	speed
$\boldsymbol{d} = \boldsymbol{r}_2 - \boldsymbol{r}_1$	difference position
d	distance
$\boldsymbol{u} = \boldsymbol{v}_2 - \boldsymbol{v}_1$	difference velocity
u	relative speed
$\boldsymbol{w} = \boldsymbol{v}_1 \times \boldsymbol{v}_2$	
G	Newton's constant
M	mass of central body
S	radius of central body
m	particle mass
s	particle radius
t^0	particle creation time
T	orbit time
$\omega = 2\pi/T$	mean motion
T_{coll}	time between collisions
T_{scat}	time between close encounters
T_{prec}	precession period/time scale
T_{sim}	total simulated time
a, b	semimajor-, semiminor axis
$c = a\epsilon$	semi-focal separation
ℓ	semi-latus rectum
$\boldsymbol{r}^0$	particle creation point
$\boldsymbol{S}$	spin angular momentum
$\boldsymbol{L}$	orbital angular momentum
$\boldsymbol{K} = \boldsymbol{L}_1 \times \boldsymbol{L}_2$	direction of nodal line
$\boldsymbol{\epsilon}$	eccentricity vector
ϖ	argument of periapsis
Ω	ascending node
I	inclination
ν	true anomaly
E	eccentric anomaly
$\mathscr{R}$	rotation matrix
N	number of particles
κ	number of fragments
i, j	particle indices
k, l	rounds to collision
n	counter

Notes. Symbol and significance of the physical quantities used. The symbol ϖ, which usually represents the longitude of the periapsis is here used for the argument of periapsis, so that we can reserve the symbol ω for the angular frequency or mean motion.

sulting gravitational scattering may be modeled with an elastic collision. These systems therefore have four different timescales: the orbit time T, the collision time T_{coll}, the scattering time T_{scat}, and the secular time T_{prec}.

In order to estimate how many collisions happen per unit of time, we first consider a thin disk with a completely random-ized (homogeneous) distribution of particles in near-circular or-
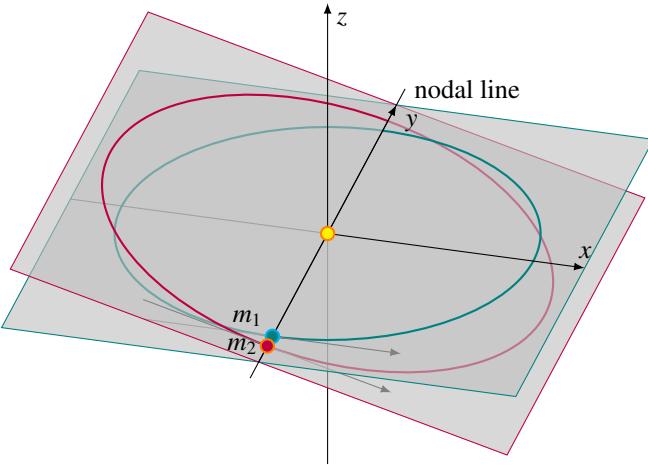
Fig. 2. Orbits of two planetoids m_1, m_2 in their orbital planes. Because the bodies are much smaller than the orbits, $s_j \ll a_j$, collisions happen near the mutual nodal line of intersection of the orbital planes, even for small inclinations. The tangent vectors (*gray*) indicate the linear approximation that may be used to find the collision point.

bits. A particle with radius s traces a cylindrical volume of size $(2\pi a/T)\pi s^2$ per unit of time. The disk is a cylinder of radius a and height $2Ia$, meaning that the particle density is $N/2\pi Ia^3$. Accounting for the $N^2/2$ pairs, the rate of collisions is estimated to be

$$\frac{1}{T_{\text{coll}}} = \underbrace{\frac{2\pi a}{T}}_{\substack{\text{velocity}}} \cdot \underbrace{\pi(2s)^2}_{\substack{\text{effective} \\ \text{cross section}}} \cdot \underbrace{\frac{N}{2\pi Ia^3}}_{\substack{\text{density}}} \cdot \underbrace{\frac{N}{2}}_{\substack{\text{pairs}}} = \frac{2\pi N^2 s^2}{Ia^2 T}.$$

If we want to include close encounters, we may substitute s into the formula for the radius of the sphere of influence $s = (m/M)^{2/5}a$. The timescales are therefore

$$T_{\text{scatt}} = \frac{I}{2\pi N^2}\left(\frac{M}{m}\right)^{4/5}T, \quad T_{\text{coll}} = \frac{Ia^2}{2\pi N^2 s^2}T, \quad T_{\text{prec}} = \frac{4M}{Nm}T.$$

The formula for the precession is taken from Murray & Dermott (2009). If we model the early inner Solar System by $N = 10^6$ planetesimals of characteristic size $s = 100$km in a disk with $a = 4$au and $I = .1$, we have

$$T_{\text{scatt}} \approx 45\text{min}, \quad T_{\text{coll}} \approx 35\text{yr}, \quad T_{\text{prec}} \approx 10^5\text{yr}, \quad T \approx 10\text{yr}.$$

Next, we consider a ring system around a planet. If we assume its radius is only a few times that of the planet and the ring particles have the same density as the planet, the collision time is comparable to the scattering time. For the Uranus ring system, we take $N = 10^{13}$, $s = 1$m, and $a = 10^5$km, which results in

$$T_{\text{scatt}} \approx T_{\text{coll}} \approx 10^{-9}\text{s}, \quad T_{\text{prec}} \approx 190\text{d}, \quad T \approx 1\text{d}.$$

The precession is now entirely due to planet oblateness ($J_2 = 3 \cdot 10^{-3}$). We now estimate the deflection angle due to scattering. When the scattering at impact parameter b is integrated over all values, for a path length of $2\pi a$ we find that

$$\frac{\text{deflection}}{\text{orbit}} = \underbrace{\frac{N}{2\pi Ia^3}}_{\substack{\text{density}}} \cdot \int_0^s \underbrace{\frac{2am}{bM\epsilon^2}}_{\substack{\text{deflection}}} \cdot \underbrace{2\pi a \, 2\pi b db}_{\substack{\text{volume shell}}} = \frac{4\pi N}{I\epsilon^2}\left(\frac{m}{M}\right)^{7/5}.$$

The orbital eccentricity ϵ accounts from the fact that the relative velocity is roughly $\sqrt{GM/a}\,\epsilon$, which becomes small for orbits

Table 2. Algorithmic efficiency

algorithmic step	runtime $\propto O(\cdot)$	memory $\propto O(\cdot)$
create particle list	N	N
sort particle list	$N \log N$	1
create collision list	$N^2\epsilon$	$\dfrac{N^2 s}{a}$
sort collision list	$\dfrac{N^2 s}{a} \log \dfrac{N^2 s}{a}$	1
reduce particle list	$\log N$	0
create κ fragments	κ	κ
sort κ fragments	$\kappa \log \kappa$	1
merge particle lists	$N + \kappa$	1
reduce collision list	$\dfrac{Ns}{a} \log \dfrac{N^2 s}{a}$	0
create new collision list	κN	$\dfrac{\kappa N s}{a}$
merge collision lists	$\dfrac{N^2 s}{a} + \dfrac{\kappa N s}{a}$	1
total simulation	$N^2\epsilon + \dfrac{N^4 s^3}{Ia^3}$	$\dfrac{N^2 s}{a} + \dfrac{N^4 s^3}{Ia^3}$

Notes. Order estimations, in big O, of the time and memory requirement in Keplerian collision detection. The top shows the initialisation, the middle rows show the managing of one collision, and the bottom is for the full simulation. This total is found because there are an expected $O(N^2 s^2/Ia^2)$ collisions in any fixed simulation time. For comparison, the time complexity for tree codes scales as $O(N^{4/3}I^{-1/3}\log N)$. The parameter κ is the number of fragments produced in a collision.

with the same sense of rotation. Although there are many close encounters where the mutual gravity takes over the central force, the (very crude) estimate of the deflection angle is about 10^{-5} and 10^{-7} per orbit for the inner Solar System and the Uranus ring system, respectively, mainly due to high relative velocities.

The algorithm maintains a list of all particle pairs that are on a collision course. As any pair can only collide near the line of intersection (in Fig. 2) of the two orbital planes, any random pair has the probability $\approx 2s/a$ of being on a collision trajectory. An estimate of the number of pairs on a collision course, or "collision pairs", is therefore $N^2 s/a$. The results of simulations shown in Figs. 3-4 validate these estimates.

As the algorithm steps from one collision to the next, the (average) time-step is equal to the collision time T_{coll}. Although exact precision is already lost in one orbit if corrections for secular motion are not included, we expect collision detection using the Kepler orbits to be able to give reliable statistical results for $T/N < T_{\text{coll}} < T_{\text{prec}}$.

2.1. Comparing time-steps

We are interested in comparing this approach with numerical integration propagators with collision detection. In order to detect a collision in numerical integration, one must find nearest neighbors. In the tree code, one uses boxes of volume dx^3 of a sufficiently small size to contain only one or a few particles. In one time-step, dt, the change in position should not move the particle too many boxes away from its original position; otherwise, it becomes impossible to select neighbors. Alternatively, in an algorithm that uses a sorted list of the coordinates (so-called spatial hashing codes), a particle coordinate, say x, can overtake the values of other particles when its position changes by dx. In this latter case, the number of particles that one particle overtakes in
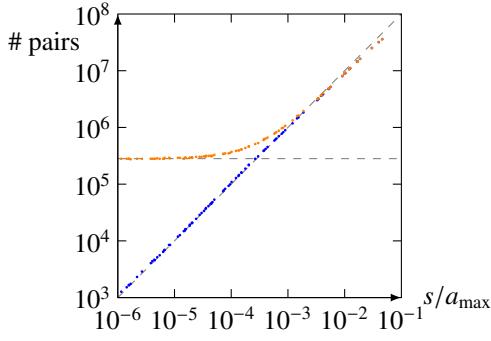
Fig. 3. Pairs vs. particle radius s for a homogeneous disk with $a \le a_{max} = 2$au, $I \le 10^{-3}$, $\epsilon \le 10^{-3}$, and with $N = 10^4$, from Aliberti (2022a,b). *Orange*: Twice the number of pairs that needed to be checked in the apoapsis/periapsis filter. *Blue*: Actual number of collision possibilities. The guideline with slope 1 (*dashed*) shows the approximate linear dependence of the collision pairs on s. The influence of the planet size can be seen in the apoapsis/periapsis filter for large s. Because there are two collision possibilities for each pair, the blue dots lie below or on the orange dots.
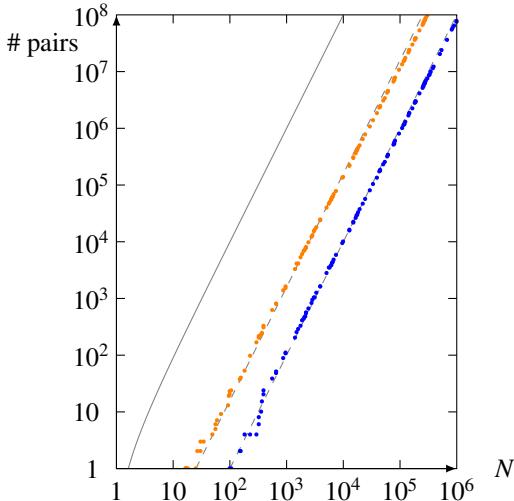
Fig. 4. Pairs vs. particle count N, as in Fig. 3, but with particle radius $s = 2 \cdot 10^{-3}$au. *Orange*: Number of pairs that needed to be checked in the apoapsis/periapsis filter. *Blue*: Actual number of collision pairs (that could ultimately collide). These determine the runtime and memory for the creation of the collision list; see Table 2. *Solid* lines: $N(N-1)/2$. *Dashed* line: guideline with a slope of 2.
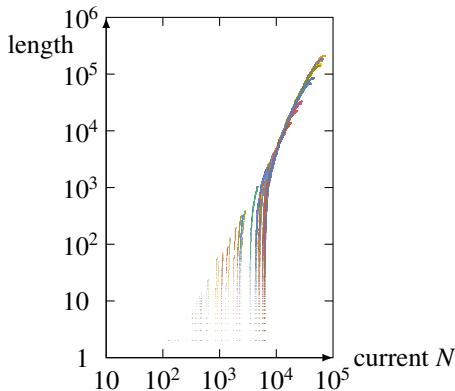

Fig. 5. Length of the list of collision pairs vs. particle count (same parameters as in Fig. 4). Only mergers are simulated, and consequently the N value decreases by one at every step. The graph shows 64 runs.
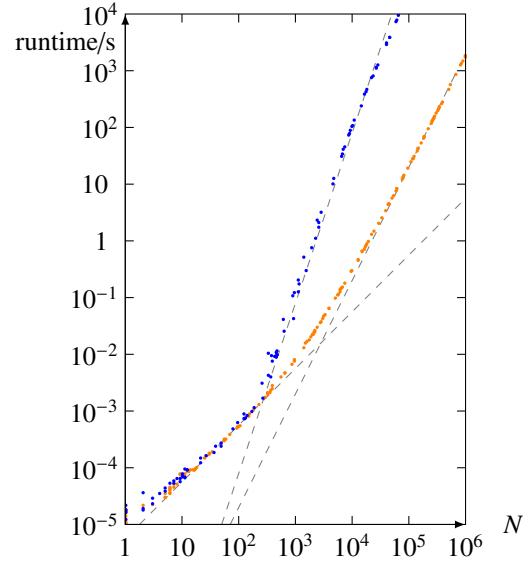
Fig. 6. Initialisation- (*orange*) and simulation runtime (*blue*) vs. particle count N, as in Fig. 4. Dashed lines are guidelines with slopes of 1, 2, and 3. The initialization time is respectively linear and quadratic in N for $N < 1/\epsilon$ and $N > 1/\epsilon$, in accordance with the estimates in Table 2. However, because only mergers were simulated, there were at most N collisions, which results in a runtime of $O(N^3)$ instead of $O(N^4)$.

one step should also remain small in order to limit the number of neighboring particles that need to be inspected. As a result, numerical integration with collision detection requires not only several time-steps for one orbital period ($dt \lessgtr T$) but also spatial steps of the order of the inter-particle distance ($dx \lessgtr \overline{r_{ij}}$). We may estimate the average distance by assuming a homogeneous distribution of the particles. We then find for a disk with $a = 4$au, $I = .1$:

$$\overline{r_{ij}} = a\Gamma(\tfrac{4}{3})\left(\frac{3I}{2N}\right)^{1/3} = \frac{2\text{au}}{N^{1/3}}, \quad dt \lessgtr \frac{T\Gamma(\tfrac{4}{3})}{2\pi}\left(\frac{3I}{2N}\right)^{1/3} = \frac{220\text{d}}{N^{1/3}}.$$

Clearly, the time-steps for numerical integration actually need to be quite small.

2.2. Efficiency of the algorithm

Apart from the advantage of having large time-steps, another benefit of our method is that only the collision products need to be tested for possible future collisions with the set of existing particles. This involves $\propto N$ computations per collision. However, a list of collision possibilities needs to be maintained. The length of this list is expected to be of the order $N^2 s/a$. This list takes up memory and therefore requires careful manipulation. At the creation or the removal of a particle, an average number of $2Ns/a$ new collision possibilities is added or removed, respectively. Hence, after $N/2$ collisions the list has mostly changed. This is understandable, because then most particles are replaced by new particles. It also means that for $Ns/a \gg 1$, most possibilities do not actually happen. Figure 4 shows the initial list length and Fig. 5 shows how this length changes during the simulation.

Table 2 sums up the efficiencies in the various steps of the algorithm. Figure 6 shows the measured runtime for a large set of simulations with different particle numbers. As we did not include defragmentation but only mergers, Figure 7 shows the values of particle radius s and particle count N for which the Kepler collision detection is more efficient than the algorithmic
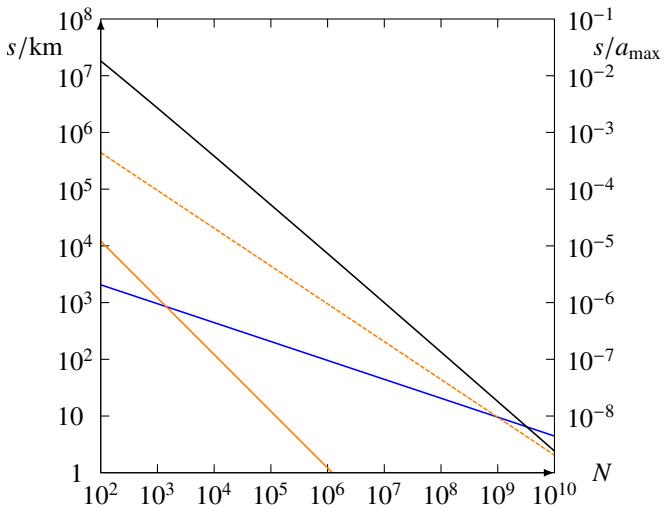
Fig. 7. Particle radius *s* vs. particle count *N* for a disk system with $I \leq .1$ and $a \leq 4$au. Above the *black* line, the numerical integration propagator with tree code for collision detection beats the analytic propagator using Kepler orbits (in terms of time efficiency). Precession due to a Jupiter has a small effect above the *orange* line, where $T_{coll} = T_{prec}$, and has a negligible effect above the *orange dashed* line, $aT_{coll} = sT_{prec}$. If the mass of three Earths is distributed over equal-sized planetoids (of Earth density), one is constrained to the *blue* line. On this line, precession is small for $N > 10^3$ but only attains the much smaller error of *s* per orbit for $N > 10^9$. Clearly, tree-code is better for these high number densities.

efficiency $(T_{sim}/dt)N \log N \propto N^{4/3} \log N$ of numerical integration with nearest neighbor search using a tree code.

In order to make another comparison between the methods, consider a Solar System with a fixed amount of material volume. We take a disk with $a = 4$au, $I = .1$, and a mass $Nm = 3M_{Earth}$ and with particles of the density of Earth. We thus have $Ns^3/a^3 = 10^{-15}$. The resulting average *s* is also plotted in Fig. 7. The efficiencies for collision detection for the cases without and with precession are then $(N^2 s/a)(T/T_{coll}) \propto N^3$ and $N^2 T/T_{prec} \propto N^2$, respectively. These are shown in Fig. 8, and are also compared with the algorithmic efficiency numerical integration with a tree code. For a Solar System model with these parameter values, the octree code is faster for $N > 10^8$.

2.3. Applications and neglected effects

We can think of the following applications: (i) Gravity assists at planetary flybys of a probe traversing the Solar System: the collision detection scheme calculates the times of passage using the sphere of influence. (ii) The tracking of a comet as its orbit is perturbed at close encounters by the planets. (iii) The rings of Saturn, with contact collisions and/or the gravitational influence of shepherd moons. (iv) Growth by merging planetesimals into protoplanets, in young planetary systems. (v) A simple model to study the Kessler syndrome, where artificial satellites break apart due to impacting space debris.

Because of the approximations, collision detection with Kepler orbits is often inaccurate. However, sometimes accuracy is not the aim, or not even possible due to the chaotic nature of the problem. Instead, we may simply want to find out what could happen, and calculate the probabilities of the various outcomes. The speedup allows sampling of initial states, either by adding many small perturbations to one initial state or by adding one perturbing body with many initial states from a large phase-space volume. Analytic propagation with collision detec-
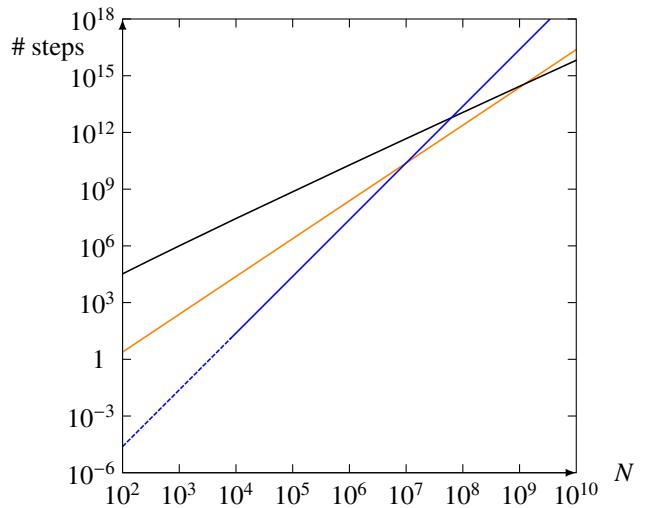
Fig. 8. Theoretical algorithmic efficiency: number of computational steps (big *O*) for one orbital period vs. particle count *N*. *Blue*: Intersecting Kepler orbits without secular dynamics. *Orange*: with secular dynamics. The *blue dotted* line indicates where precession due to secular dynamics cannot be neglected. The *black* line shows the estimated number of steps for a tree code and/or spatial hashing. The slopes of the lines are 3, 2, and 4/3, respectively.

tion based on the Kepler orbits neglects the following effects: (i) orbital precession due to mutual gravity or oblateness of the central body (as discussed in Sect. 6.3), (ii) three-body gravitational scattering, (iii) planetary migration due to interaction with the gas in the protoplanetary disk, (iv) capture of planets in mean-motion resonances, (v) the Kozai mechanism, (vi) moons and binaries, (vii) atmospheric and (viii) tidal drag, (ix) solar wind, and (x) the Poynting-Robertson effect.

3. The algorithm

3.1. Initialization

We have a system of planets, or particles orbiting a central mass. The particles are numbered $j = 1, \ldots, N$. For each particle, we store the following set of variables:

$$\{t_j^0, \quad a_j, \quad c_j, \quad s_j, \quad m_j, \quad \boldsymbol{r}_j^0, \quad \boldsymbol{L}_j, \quad \boldsymbol{\epsilon}_j, \quad \omega_j\}. \tag{1}$$

Here, t^0 is the time of creation, a, c are the orbital radius and 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 ω is the frequency.

An initial state would consist of many particles in nearly circular orbits and therefore with small ϵ. If one draws random numbers for the mean anomaly from the interval $[0, 2\pi]$, the resulting (smoothed) phase-space distribution will become stationary; if the values of ϖ, Ω are sampled from $[0, 2\pi]$, the distribution will become axisymmetric; if also $\cos I$ is drawn from $[0, 1]$, it will become spherically symmetric (see Savransky et al. 2011). To simulate a thin disk, one takes small values for I. Equations (B.6) and (B.7) give the vectors $\boldsymbol{L}$ and $\boldsymbol{\epsilon}$ in terms of a and ϵ and the angles I, ϖ, and Ω.

We then sort the particle list by increasing value of periapsis. This will allow the implementation of the apoapsis/periapsis filter (Baraff & Witkin 1992). The next step is to consider all particle pairs, and list the pairs that can collide. For these, we also store the calculated collision time $t^1_{(i,j)}$. If sufficient memory is available, it is possible to store the parameters in Eq. (1)

for the new particle that would be formed after the collision. The soonest collision is at the top of the list.

3.2. Main loop

1. If the collision list is empty, end the simulation.
2. Take the pair (i, j) with the soonest collision: the first in the list.
3. Update the time t to the time $t^1_{(i,j)}$ of the collision.
4. Remove any pair containing i and any pair containing j from the pairs list.
5. Remove the particles i and j from the particle list.
6. If the orbit of the new particle intersects the central mass or is unbound, go to the next collision on the list.
7. Create new particle(s) defined by $\{t^1_{(i,j)}, a, c, s, m, r^0, L, \epsilon, \omega\}$.
8. For any new particle, consider the other particles and decide if the pair is on a collision course. If this is the case, calculate the time of the earliest collision.
9. Make a sorted list of the new collision possibilities with a record of the collision time and the pair, soonest collision first.
10. Merge this sorted list with the existing sorted list of collision possibilities into a full list of pairs, sorted by time of the collision event, soonest collision first.

3.3. Determining if a pair is on a collision course

At the initialization, pairs of particles need to be considered for a possible future collision. Also, during the simulation, each time a new particle is created, all existing particles need to be paired with the new particle and considered for a possible future collision. However, as we implement the sweep and prune method, only pairs need to be considered with an overlap in the range of radial motion. The radial coordinate for each particle i ranges over the interval from the periapsis to the apoapsis:

$$[a_i - c_i - s_i, a_i + c_i + s_i],$$

including an extension s_i of the size of the particle, or with the substitution of its gravitational reach $s_i = (m_i/M)^{2/5} a_i$. It is sufficient to compare each particle i with the particles $j = i + 1, j = i+2, \ldots$. Because the list is sorted, we have $a_i - c_i - s_i \le a_j - c_j - s_j$. As long as $a_i + c_i + s_i \ge a_j - c_j - s_j$, the intervals for i and j overlap and the pair is a candidate for collision. Once we encounter the first j where $a_i + c_i + s_i < a_j - c_j - s_j$, there are no more particles that can interact with i and we can go to particle $i + 1$. If ϵ is the average eccentricity, only a fraction of 2ϵ of the total number of pairs $N^2/2$ need to be checked. The resulting reduced number of checks in our numerical simulations is shown in Figs. 4 and 6.

For a particle i created during the simulation, the selection of pairs is slightly different. Again, it is sufficient to consider only particles j with periapsis smaller than the apoapsis of i. However, this time we have to start at $j = 1$.

1. Consider a pair, say $(i, j) = (1, 2)$; we assume that $a_1 > a_2$. First we need to find the minimal orbit intersection distance to decide whether or not there can be a collision. For this, retrieve m_1, L_1, ϵ_1, and m_2, L_2, ϵ_2. Then calculate the direction of the nodal line (see Fig. 2), the semi-latus recti, the

intersection points, and the velocities at these points:

$$
\begin{aligned}
K &= L_1 \times L_2, \\
K &= \sqrt{K \bullet K}, \\
\ell_1 &= \frac{L_1 \bullet L_1}{GMm_1^2}, \qquad\qquad \ell_2 = \frac{L_2 \bullet L_2}{GMm_2^2}, \\
r_1 &= \frac{K\ell_1}{\pm K + \epsilon_1 \bullet K}, \qquad r_2 = \frac{K\ell_2}{\pm K + \epsilon_2 \bullet K}, \qquad (2) \\
r_1 &= \sqrt{r_1 \bullet r_1}, \qquad\qquad r_2 = \sqrt{r_2 \bullet r_2}, \\
v_1 &= \frac{L_1}{m_1 \ell_1} \times \left(\epsilon_1 + \frac{r_1}{r_1}\right), \qquad v_2 = \frac{L_2}{m_2 \ell_2} \times \left(\epsilon_2 + \frac{r_2}{r_2}\right). \quad (3)
\end{aligned}
$$

We write G for Newton's constant. Equation (2) for the intersection points is derived in Sect. 4.1. Equation (3) is Eq. (B.1) from Appendix B. The two pairs (r_1, r_2) at the opposite sides are indicated with the plus/minus symbol. The steps that now follow must be performed on both of the pairs.

2. Next, calculate for both particles the (approximate) points on the orbits where the distance is minimal:

$$
\begin{aligned}
d &= r_2 - r_1, \\
w &= v_1 \times v_2, \\
(w^2) &= w \bullet w, \\
r'_1 &= r_1 + \left(d \bullet \frac{v_2 \times w}{(w^2)}\right) v_1, \quad r'_2 = r_2 + \left(d \bullet \frac{v_1 \times w}{(w^2)}\right) v_2, \quad (4) \\
r_1 &= r'_1, \qquad\qquad r_2 = r'_2, \\
d &= r_2 - r_1, \\
d &= \sqrt{d \bullet d}.
\end{aligned}
$$

Equation (4) for the collision points r'_1 and r'_2 is derived in Sect. 4.2.

3. Retrieve a_1, s_1 and a_2, s_2. Decide whether or not an interaction can take place.

$$
\begin{aligned}
d &< s_1 + s_2 &&\implies \quad \text{1 and 2 collide.} \\
d &< (m_1/M)^{2/5} a_1 &&\implies \quad \text{1 can perturb orbit 2.} \\
d &< (m_2/M)^{2/5} a_2 &&\implies \quad \text{2 can perturb orbit 1.}
\end{aligned}
$$

If there is no contact or perturbation: go to the next pair.

3.4. Deterministic collision time

We now give the steps in the calculation of the exact collision moment.

1. Retrieve t^0_1, ω_1, r^0_1 and t^0_2, ω_2, r^0_2 of the particles involved (with $\omega_1 < \omega_2$). Calculate the first time a particle passes the

crossing point in Fig. 2. These times are denoted by t_1^1, t_2^1.

$$(\epsilon_1^2) = \boldsymbol{\epsilon}_1 \bullet \boldsymbol{\epsilon}_1,$$
$$(\epsilon_2^2) = \boldsymbol{\epsilon}_2 \bullet \boldsymbol{\epsilon}_2,$$
$$r_1 = \sqrt{\boldsymbol{r}_1 \bullet \boldsymbol{r}_1},$$
$$r_2 = \sqrt{\boldsymbol{r}_2 \bullet \boldsymbol{r}_2},$$
$$z = \left(\frac{\boldsymbol{r}_1}{a_1} - \frac{i r_1 \boldsymbol{v}_1}{a_1^2 \omega_1}\right) \bullet \left(\frac{\boldsymbol{r}_1^0 - \boldsymbol{\epsilon}_1 \, \boldsymbol{\epsilon}_1 \bullet \boldsymbol{r}_1^0}{a_1 - (\epsilon_1^2) a_1} + \boldsymbol{\epsilon}_1\right) + \frac{\boldsymbol{r}_1^0 \bullet \boldsymbol{\epsilon}_1}{a_1} + (\epsilon_1^2) \tag{5}$$

$$\Delta E_1 = \arg z, \quad 0 \le \Delta E_1 < 2\pi,$$
$$z = \left(\frac{\boldsymbol{r}_2}{a_2} - \frac{i r_2 \boldsymbol{v}_2}{a_2^2 \omega_2}\right) \bullet \left(\frac{\boldsymbol{r}_2^0 - \boldsymbol{\epsilon}_2 \, \boldsymbol{\epsilon}_2 \bullet \boldsymbol{r}_2^0}{a_2 - (\epsilon_2^2) a_2} + \boldsymbol{\epsilon}_2\right) + \frac{\boldsymbol{r}_2^0 \bullet \boldsymbol{\epsilon}_2}{a_2} + (\epsilon_2^2),$$

$$\Delta E_2 = \arg z, \quad 0 \le \Delta E_2 < 2\pi,$$
$$t_1^1 = t_1^0 + \frac{\Delta E_1}{\omega_1} - \frac{\boldsymbol{\epsilon}_1 \times (\boldsymbol{r}_1 - \boldsymbol{r}_1^0)}{1 - (\epsilon_1^2)} \bullet \frac{\boldsymbol{L}_1}{G M m_1}, \tag{6}$$
$$t_2^1 = t_2^0 + \frac{\Delta E_2}{\omega_2} - \frac{\boldsymbol{\epsilon}_2 \times (\boldsymbol{r}_2 - \boldsymbol{r}_2^0)}{1 - (\epsilon_2^2)} \bullet \frac{\boldsymbol{L}_2}{G M m_2}.$$

These equations are derived in Sect. 5. Equation (5) is obtained by combining Eqs. (11-12). The complex number has unit modulus. Equation (6) for the passage times follows from Eq. (10).

2. Evaluate the small dimensionless parameter.

$$\boldsymbol{u} = \boldsymbol{v}_2 - \boldsymbol{v}_1,$$
$$\delta = \frac{1}{|t_1^1 - t_2^1|} \sqrt{\boldsymbol{u} \bullet \boldsymbol{u} \frac{(s_1 + s_2)^2 - d^2}{(w^2)}}. \tag{7}$$

3. Next calculate the exact number of periods k that planet 1 makes before colliding with planet 2. The method uses the continued fraction of the ω_2/ω_1. Initialize the loop:

$$q_0 = \frac{2\pi}{\omega_1 |t_1^1 - t_2^1|}, \qquad\qquad k_0 = 1,$$
$$q_1 = \frac{2\pi}{\omega_2 |t_1^1 - t_2^1|}, \qquad\qquad k_1 = 0.$$

4. Start loop counter at $n = 0$. The loop creates integer sequences α_n, k_n and the positive sequence q_n. These are the *digits*, the denominators of the *convergents*, and the *remainders* of the continued fraction (we do not need the numerators, denoted l_n). The loop performs the iterations

$$\alpha_{2n} = \left\lfloor \frac{q_{2n}}{q_{2n+1}} \right\rfloor, \quad q_{2n+2} = q_{2n} - \alpha_{2n} q_{2n+1},$$
$$\text{if} \quad q_{2n+2} = 0 \quad \Longrightarrow \quad \text{next pair}$$
$$k_{2n+2} = k_{2n} - \alpha_{2n} k_{2n+1},$$
$$\alpha_{2n+1} = \left\lfloor \frac{q_{2n+1}}{q_{2n+2}} \right\rfloor, \quad q_{2n+3} = q_{2n+1} - \alpha_{2n+1} q_{2n+2},$$
$$\text{if} \quad q_{2n+3} = 0 \quad \Longrightarrow \quad \text{next pair}$$
$$k_{2n+3} = k_{2n+1} - \alpha_{2n+1} k_{2n+2}.$$

We use the notation $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ for the floor and the ceiling function. The time T_{sim} to be simulated sets an upper bound for the solution:

$$x_{\max} = \frac{(T_{\text{sim}} - t_1^1)\frac{\omega_1}{2\pi} q_{2n+1} - (1 + \delta) k_{2n+1}}{k_{2n} q_{2n+1} - q_{2n} k_{2n+1}}.$$

We then test the points with coordinates

$$x = \left\lceil \frac{1 - \delta}{q_{2n}} \right\rceil, \dots, \left\lfloor \frac{1 + \delta}{q_{2n+2}} \right\rfloor,$$
$$\text{if} \quad x > x_{\max} \quad \Longrightarrow \quad \text{next pair}$$
$$y = \max\left(0, \left\lceil \frac{q_{2n} x - 1 - \delta}{q_{2n+1}} \right\rceil\right).$$

If for any of these points $1 - \delta < x q_{2n} - y q_{2n+1}$, then we have found the solution. If not, we increase n by 1 and check this next.

5. The solution is

$$k = x k_{2n} - y k_{2n+1}, \quad t^0 = t_1^1 + \frac{2\pi k}{\omega_1}. \tag{8}$$

Equation (8) for the deterministic collision time is derived in Sect. 6.1.

The solution $k = 0$ means that the pair $(1, 2)$ is about to collide within the present period. If a gravitational scattering between the same pair $(1, 2)$ has happened at the previous time-step, the solution $k = 0$ corresponds to the scattering that was just simulated, and therefore is invalid. However, for a different pair where one particle participated in this last scattering, the solution $k = 0$ is valid, as it implies an immediate collision with a third particle. For three (or more) bodies inside each others sphere of influence, the multi-body gravitational scattering will therefore be treated as three (or more) successive two-body interactions.

3.5. Stochastic collision time

Although the time of collision between two bodies is uniquely determined by the initial conditions, it is highly sensitive to the precise values of the creation times and the orbital periods. Therefore, if the physical or numerical error in the (initial) values is bigger than s/a, the collision time becomes unpredictable.

If the collision process is assumed to be stochastic, one may adopt the following Monte Carlo method. First, a random number ξ is drawn from the interval $[0, 1]$. The moment of collision is calculated with

$$\boldsymbol{u} = \boldsymbol{v}_2 - \boldsymbol{v}_1,$$
$$t^0 = t_1^1 + \frac{(-\log \xi)(2\pi)^2}{2GM} \sqrt{\frac{(w^2)}{\boldsymbol{u} \bullet \boldsymbol{u}} \frac{a_1^3 a_2^3}{(s_1 + s_2)^2 - d^2}}. \tag{9}$$

Equation (9) is derived in Sect. 6.2.

3.6. The collision

In order to simulate the collision or scattering event, a physical model of the merger or break-up of the particles needs to be implemented. In the case of pure gravitational scattering (close passage), one can use the formulas from Appendix C for the momentum exchange. This elastic collision is depicted in Fig. 9. We now outline how to find the orbit of the new particle in case of a merger between particles 1 and 2.

1. For a simple merger, the new particle has radius, mass, position, velocity, and angular momentum[1] calculated from the

---

[1] As pointed out by the referee, orbital angular momentum is not strictly conserved, because it is transferred into spin for oblique collisions: this spin becomes $\boldsymbol{S} = \boldsymbol{S}_1 + \boldsymbol{S}_2 + (m_1 m_2 / m)\boldsymbol{d} \times \boldsymbol{u}$, provided $|\boldsymbol{S}| < (2/5)\sqrt{G m^3 s}$.
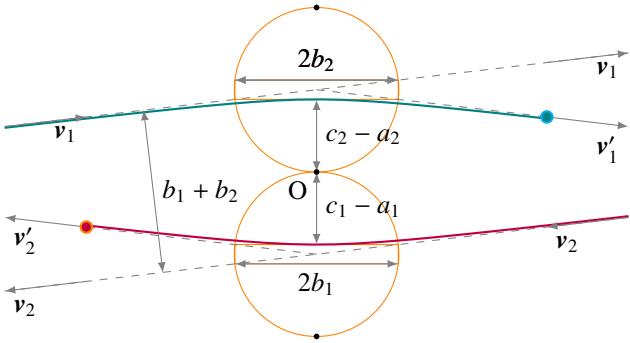
Fig. 9. Gravitational scattering between particles 1 and 2 in the center-of-mass frame. The initial velocities v_1, v_2 are scattered in the directions v'_1, v'_2 along the asymptotic lines (*dotted*). The actual orbits are hyperbolas (*blue, purple*). The focal points (*black dots*), with O being the common focal point of the orbits, lie on the *orange* circles. The four points of intersection of a circle with the asymptotes form rectangles with dimensions of the transverse axis $2a_j$ by the conjugate axis $2b_j$ (Adams & Essex 2021). The impact parameter is the sum $b = b_1 + b_2$.

basic conservation laws:

$$s = (s_1^3 + s_2^3)^{1/3} \qquad \text{(material volume)},$$

$$m = m_1 + m_2 \qquad \text{(mass)},$$

$$\boldsymbol{r} = \frac{m_1 \boldsymbol{r}_1 + m_2 \boldsymbol{r}_2}{m} \qquad \text{(center-of-mass motion)},$$

$$\boldsymbol{v} = \frac{m_1 \boldsymbol{v}_1 + m_2 \boldsymbol{v}_2}{m} \qquad \text{(momentum)},$$

$$\boldsymbol{L} = m \boldsymbol{r} \times \boldsymbol{v} \qquad \text{(angular momentum)}.$$

2. Decide whether or not the new particle collides with the central body. We suppose that it is a sphere of radius S.

$$\ell = \frac{\boldsymbol{L} \bullet \boldsymbol{L}}{GMm^2},$$

$$r = \sqrt{\boldsymbol{r} \bullet \boldsymbol{r}},$$

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

$$\epsilon = \sqrt{\boldsymbol{\epsilon} \bullet \boldsymbol{\epsilon}}.$$

In the general case that several collision fragments are created in the collision, there are five cases, with three outcomes (see Fig. 10) for a fragment. First, consider the cases where there is no crossing:

$$\ell > (1 + \epsilon)S \quad \text{and} \quad \epsilon < 1 \quad \implies \quad \text{new particle stays}.$$

$$\ell > (1 + \epsilon)S \quad \text{and} \quad \epsilon \geq 1 \quad \implies \quad \text{particle escapes}.$$

In the remaining cases, $\ell \leq (1 + \epsilon)S$ and the orbit crosses the central body.

$$\epsilon \geq 1 \quad \text{and} \quad \boldsymbol{r} \bullet \boldsymbol{v} > 0 \quad \implies \quad \text{particle escapes}.$$

$$\epsilon \geq 1 \quad \text{and} \quad \boldsymbol{r} \bullet \boldsymbol{v} \leq 0 \quad \implies \quad \text{collides with } M.$$

$$\epsilon < 1 \quad \implies \quad \text{collides with } M.$$

Only in the first case does the particle stay in a bound orbit, and we continue; otherwise, the particle is removed.
For a merger, there is only one collision product, and the newly formed particle will always have $\epsilon < 1$ because (total) energy can only decrease.

3. Calculate the required orbital parameters of the new particle:

$$a = \frac{\ell}{1 - \epsilon^2}, \quad c = a\epsilon, \quad \omega = \sqrt{\frac{GM}{a^3}}.$$
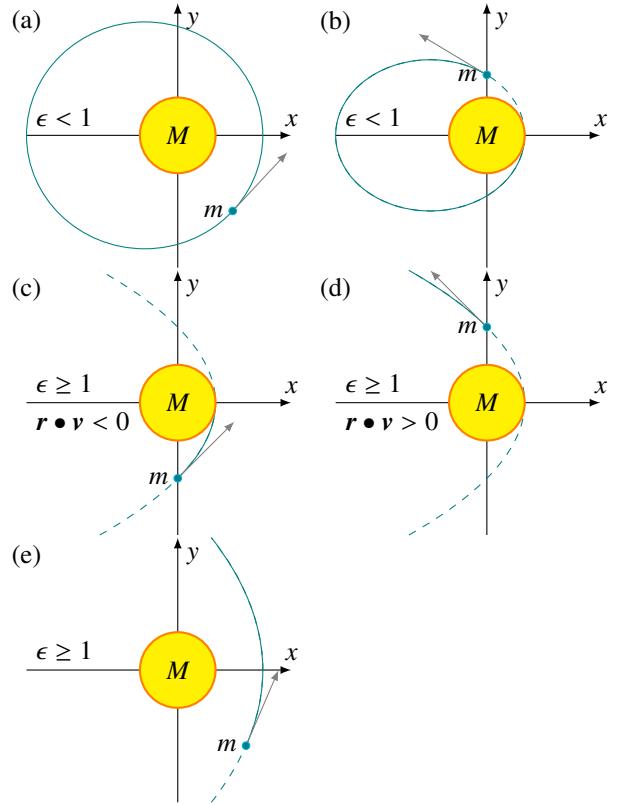
Fig. 10. Five cases for the obit of a collision fragment. (a) The particle is bound. (b), (c) The particle collides with the central mass. (d), (e) The particle leaves the system. For long intervals between collisions ($T_{\text{coll}} \gg T$), cases (b-e) can be dealt with by removing the particle. Otherwise, the fragments need to be stored and paired in a similar manner to the bound particles.

4. Register $\{t^0, a, c, s, m, \boldsymbol{r}, \boldsymbol{L}, \boldsymbol{\epsilon}, \omega\}$ of the new particle.
5. Sort the list of new particles by increasing $a - c - s$.
6. Merge these new-particle lists with the existing-particle list.

Go to the next time-step.

4. Calculating points of closest approach

In this section, we derive an approximation for the points on two Kepler orbits with minimal separation. This distance is called the minimum orbit intersection distance (MOID). As we are considering possible collisions between planets, we are interested in the case where the MOID for the orbits of planet 1 and planet 2 is less than $s_1 + s_2$. We assume that the planet radii s_j are many orders of magnitude smaller than the orbital radii a_j. Then, for an angle I between the orbital planes of larger than $(s_1 + s_2)/(a_1 + a_2)$, the MOID will be close to the line of intersection of the orbital planes (Hoots et al. 1984; Manley et al. 1998). This principle is illustrated in Fig. 2: Outside a cylinder with radius $(s_1 + s_2)/(2 \sin \frac{I}{2})$ about the "mutual nodal line", all points in orbit 1 are separated by more than $s_1 + s_2$ from points in orbit 2. Any collision must therefore happen inside the cylinder. The cylinder is only large for very small inclinations. If the system is a disk with an average inclination angle $\bar{I}$, these small inclinations are rare for $N < \bar{I}a^2/s^2$.

Because the range for gravitational scattering can be larger, our approach will only work for low-mass planets. The iterative scheme converging to the MOID that projects the points onto

the orbit followed by linearization is described in Hoots et al. (1984). Various other methods to obtain the MOID have been found (see e.g. Gronchi 2005; Milisavljević 2010; Segan et al. 2011; Wiźniowski & Rickman 2013; Hedo et al. 2018).

4.1. Intersecting orbit 1 with orbital plane 2

The Kepler orbit of a planet is entirely determined by its angular momentum L and eccentricity vector ϵ (Goldstein 1964). The angular momentum is normal to the place of the orbit and the vector $a\epsilon$ points from the center of the ellipse to the focal point where the central mass is located (see Fig. 11). Now let us consider two orbits, for planet 1 and planet 2, specified by L_1, ϵ_1 and L_2, ϵ_2, respectively. The line of intersection of the two orbital planes, or the nodal line, can be found as follows. Because the angular momenta L_1 and L_2 are both perpendicular to the nodal line, a direction vector of the nodal line is

$$K = K\hat{K} = \pm L_1 \times L_2.$$

The plus/minus symbol indicates the two opposite directions in which the intersection points with an orbit are found. Because the eccentricity vector ϵ of an orbit points from the central mass towards the periapsis, the true anomaly ν of the intersection point in the direction K is given by

$$\cos \nu = \frac{\epsilon \bullet \hat{K}}{\epsilon}.$$

The point of intersection can now be found from the formula for the orbit Eq. (B.3). We find

$$r = r\hat{K} = \frac{(1-\epsilon^2)a}{1+\epsilon\cos\nu}\hat{K} = \frac{(1-\epsilon^2)a}{1+\epsilon\bullet\hat{K}}\hat{K} = \frac{(1-\epsilon^2)a}{K+\epsilon\bullet K}K.$$

Therefore, for the two pairs of intersection points, we obtain

$$r_{1,\pm} = \frac{(1-\epsilon_1^2)a_1 L_1 \times L_2}{\pm|L_1 \times L_2| + \epsilon_1 \bullet (L_1 \times L_2)},$$

$$r_{2,\pm} = \frac{(1-\epsilon_2^2)a_2 L_1 \times L_2}{\pm|L_1 \times L_2| + \epsilon_2 \bullet (L_1 \times L_2)}.$$

4.2. Pair of closest points between two orbits

Next, we approximate the points where the MOID is found. In order to do so, we consider the tangent lines of the orbits at the points r_1 and r_2 of intersection with the nodal line. The tangent lines point in the direction of the velocities v_1 and v_2 at r_1 and r_2. These can be found using Eq. (B.1). The distance between the two lines is given by

$$d = \left| (r_2 - r_1) \bullet \frac{(v_1 \times v_2)}{|v_1 \times v_2|} \right|.$$

This is the projection of the difference vector onto the direction of shortest distance. We refer to the two points on the lines where the distance is minimal as r_1' and r_2'. These positions are given by

$$r_1' = r_1 + \left[(r_2 - r_1) \bullet \frac{|v_2|^2 v_1 - (v_1 \bullet v_2)v_2}{|v_1|^2|v_2|^2 - (v_1 \bullet v_2)^2} \right] v_1,$$

$$= r_1 + \left[(r_2 - r_1) \bullet \frac{v_2 \times (v_1 \times v_2)}{|v_1 \times v_2|^2} \right] v_1,$$

$$r_2' = r_2 + \left[(r_2 - r_1) \bullet \frac{v_1 \times (v_1 \times v_2)}{|v_1 \times v_2|^2} \right] v_2.$$

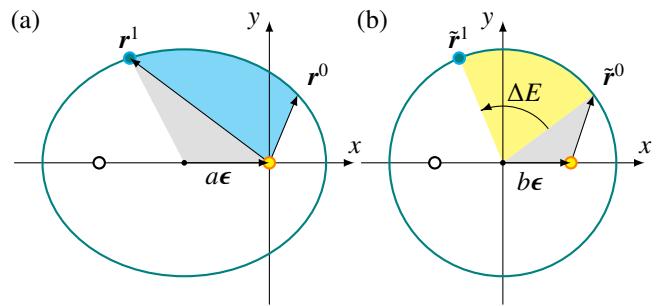**Fig. 11.** Elliptical Kepler orbit (a) and orbit squeezed into a circle (b) from scaling the *x*-axis by b/a. The travel time from r^0 to r^1 is equal to the ratio of the swept (*cyan*) area in (a) to πab times the period. In (b), this ratio is the (*yellow*) circle segment plus the (*gray*) triangle minus the image of the (*gray*) triangle in (a) over πb^2. The circle segment has area $\Delta E b^2/2$, with $\Delta E = E^1 - E^0$ being the difference in eccentric anomaly.

One may verify that $(r_2' - r_1') \bullet v_1 = (r_2' - r_1') \bullet v_2 = 0$. This proves that the minimal distance between the lines is realized at the points r_1' and r_2'. We also have that $(r_2' - r_1') \bullet (r_2 - r_1) = d^2$, implying that $|r_2' - r_1'| = d < |r_2 - r_1|$.

If the inclination between the orbital planes, which is given by $I \approx K/L_1 L_2$, is not much larger than $(s_1 + s_2)/(a_1 + a_2)$, the linear approximation is inaccurate. One can improve this approximation by reducing the lengths of r_i', so that the points lie on the respective orbits, and then finding the shortest distance between the tangent lines to the orbits at these new points. Here, one may iterate as in the method of Newton Raphson.

5. Calculating the earliest passage of the crossing

In this section, we derive expressions for the time it takes a particle on a Kepler orbit to go from r^0 to r^1. In the algorithm, r^0 is the particle's creation point and r^1 is the collision point. A standard approach is to use the eccentric anomaly E values at these points. However, E becomes ill defined for small eccentricities[2]. Here, we derive the exact expression, Eq. (6), that does not depend on the value of E; only the difference between the values of E enters the derivation.

Kepler's second law states that the position vector sweeps out equal areas in equal times. The swept area can be decomposed into a segment of the ellipse (with an angle determined by the eccentric anomaly) plus the triangle between $-a\epsilon$, 0, and r^0 minus the triangle between $-a\epsilon$, 0, and r^1 in Fig. 11(a). Therefore,

$$\text{area} = \underbrace{\frac{(E^1 - E^0)ab}{2}}_{\text{ellipse segment}} + \underbrace{\frac{a\epsilon \times r^0}{2} \bullet \hat{L}}_{\text{added triangle}} - \underbrace{\frac{a\epsilon \times r^1}{2} \bullet \hat{L}}_{\text{subtracted triangle}}.$$

Kepler's second law therefore implies that the time it takes a body to move from r^0 to r^1 is equal to

$$t^1 - t^0 = \frac{\text{area}}{\pi ab}T = \frac{2\,\text{area}}{\omega ab} = \frac{E^1 - E^0}{\omega} - \frac{\epsilon \times (r^1 - r^0)}{\omega b} \bullet \hat{L}. \quad (10)$$

As the ellipse has its major axis pointing in the direction of ϵ, we can transform the elliptic orbit into the circle in Fig. 11(b) with the linear transformation

$$r \xrightarrow{\text{squeeze}} \left(1 - \frac{\epsilon\,\epsilon^{\mathsf{T}}}{\epsilon^2} + \frac{b}{a}\frac{\epsilon\,\epsilon^{\mathsf{T}}}{\epsilon^2}\right)r = r - \frac{a\epsilon \bullet r}{a+b}\epsilon.$$

---

[2] The need for the following calculations was pointed out by Soliman (2022).

This squeezes the semimajor axes by a factor b/a, and leaves the semiminor axes intact. The eccentric anomaly is defined with respect to the center of the circle. We therefore require the vector pointing from the center to the point on the squeezed r. This vector is found by

$$r \xrightarrow{\text{translate}} r + (a\epsilon) \xrightarrow{\text{squeeze}} \left(r - \frac{a\epsilon \bullet r}{a+b}\epsilon\right) + (b\epsilon).$$

When r is on the orbit, the transformed vector has a length of b. The cosine of the difference in eccentric anomalies is the dot product between the directions of the squeezed vectors:

$$\cos(E^1 - E^0) = \frac{1}{b^2}\left(r^1 - a\epsilon\frac{\epsilon \bullet r^1}{a+b} + b\epsilon\right) \bullet \left(r^0 - a\epsilon\frac{\epsilon \bullet r^0}{a+b} + b\epsilon\right).$$

This simplifies to

$$\cos(E^1 - E^0) = \frac{r^1 \bullet r^0}{b^2} + \frac{(r^1 + r^0) \bullet \epsilon}{a} - \frac{r^1 \bullet \epsilon \, \epsilon \bullet r^0}{b^2} + \epsilon^2. \quad (11)$$

By noting that the cross product between the two direction vectors gives us the sine, we find in terms of the position vectors:

$$\sin(E^1 - E^0) = \frac{r^0 \times r^1}{ab} \bullet \hat{L} + \frac{\epsilon \times (r^1 - r^0)}{b} \bullet \hat{L}.$$

If we combine this with Eq. (10), we obtain Eq. (2.69) in Murray & Dermott (2009) for the so-called *g*-function:

$$t^1 - t^0 = \frac{E^1 - E^0 - \sin(E^1 - E^0)}{\omega} + \underbrace{\frac{m(r^0 \times r^1) \bullet L}{L^2}}_{g\text{-function}}.$$

Although this Equation is remarkable because it does not contain ϵ or the values a, b, we will not need it.

Differentiating Eq. (11) with respect to time t^1 in the endpoint gives another equation:

$$\frac{-\omega a}{r^1}\sin(E^1 - E^0) = \frac{v^1 \bullet r^0}{b^2} + \frac{v^1 \bullet \epsilon}{a} - \frac{v^1 \bullet \epsilon \, \epsilon \bullet r^0}{b^2}. \quad (12)$$

These results can be verified by direct substitution of Eqs. (B.4), (B.5), and (B.7) into the right-hand side of Eq. (12). Equation (10) with the smallest non-negative value for $E^1 - E^0$ that satisfies Eqs. (11) and (12) gives the time to get from r^0 to r^1.

6. Calculating the time to collision

We consider two planets 1 and 2, with a MOID of less than $s_1 + s_2$, and we want to determine the time at which the planets collide. To this end, let r_1 be the point on orbit 1 with minimal distance to orbit 2, and r_2 the corresponding point on orbit 2 that is closest to r_1. Let v_1 and v_2 be the respective velocities of the planets if they pass these points. Now, assume that there is a possible collision:

$$|r_2 - r_1| < s_1 + s_2.$$

Let t_1^1 be the first time that planet 1 passes r_1, and t_2^1 be the first time planet 2 passes r_2. A collision occurs at time $t(k, l)$ when both planets are near the points where the distance to the other ellipse is minimal. At that time, planet 1 then passes the point for the k-th time, and planet 2 for the l-th time. The collision is therefore at

$$t(k, l) = t_1^1 + T_1 k + dt_1 = t_2^1 + T_2 l + dt_2.$$

Here k and l are integers and dt_1 and dt_2 are small shifts that allow for the fact that the planets only need to be close to the point where the distance between the orbits is minimal. Because the algorithm moves forward in time,

$$k \geq 0, \quad l \geq 0.$$

The shifts in time from the point of closest approach are therefore

$$dt_1 = t(k, l) - t_1^1 - T_1 k, \quad dt_2 = t(k, l) - t_2^1 - T_2 l, \quad (13)$$

and these need to be small. We linearize the motion about the collision time $t(k, l)$, as

$$r_1 + v_1 dt_1, \quad r_2 + v_2 dt_2.$$

By solving for the closest approach between the two particles (in contrast to the MOID, the smallness of the differentials will be a consequence of the fact that the minimal distance for colliding particles is smaller than $s_1 + s_2$. For this, we introduce the difference vectors

$$r_{12}(t) = r_2 + v_2 dt_2 - r_1 - v_1 dt_1, \quad v_{12} = v_2 - v_1.$$

We note that dt_1 and dt_2 need to be considered as functions of the collision time t. At this time t, the distance is minimal, which is at (see Eq. (A.1) in Appendix A)

$$t(k, l) = \frac{-v_{12} \bullet r_{12}(0)}{v_{12}^2},$$

with $v_{12} = |v_{12}|$. The value of the distance must be smaller than the sum of the planet radii (see JeongAhn & Malhotra 2017):

$$\frac{|v_{12} \times r_{12}(0)|}{v_{12}} < s_1 + s_2.$$

When we expand this equation, we find

$$\left|\left(r_2 - r_1 - v_2 t_2^1 - v_2 T_2 l + v_1 t_1^1 + v_1 T_1 k\right) \times v_{12}\right| < (s_1 + s_2)v_{12}.$$

Because $(r_2 - r_1) \bullet v_1 = (r_2 - r_1) \bullet v_2 = 0$, this is equivalent to

$$\left|\left(v_1 t_1^1 + v_1 T_1 k - v_2 t_2^1 - v_2 T_2 l\right) \times v_{12}\right| < \sqrt{(s_1 + s_2)^2 - |r_2 - r_1|^2} \, v_{12}.$$

Using $v_{12} = v_2 - v_1$, this can be further simplified to

$$\left|t_1^1 + T_1 k - t_2^1 - T_2 l\right| < \frac{\sqrt{(s_1 + s_2)^2 - |r_2 - r_1|^2} \, v_{12}}{|v_1 \times v_2|}.$$

We need to find the smallest non-negative integers k, l for which this inequality is satisfied. We can now recast the problem of the time to collision as finding the smallest integers k, l, so that

$$1 - \delta < kp - lq < 1 + \delta. \quad (14)$$

In this inequality, we use dimensionless parameters p, q, δ, which are defined as

$$p = \frac{T_1}{|t_1^1 - t_2^1|}, \quad q = \frac{T_2}{|t_1^1 - t_2^1|}, \quad \delta = \frac{\sqrt{(s_1 + s_2)^2 - |r_2 - r_1|^2} \, v_{12}}{|t_1^1 - t_2^1||v_1 \times v_2|}.$$

The linearization of the motion around the crossing points of the nodal line translates the collision problem into integer linear programming (in two dimensions). We can find the exact solution in a few steps, even if k and l turn out to be very large numbers.
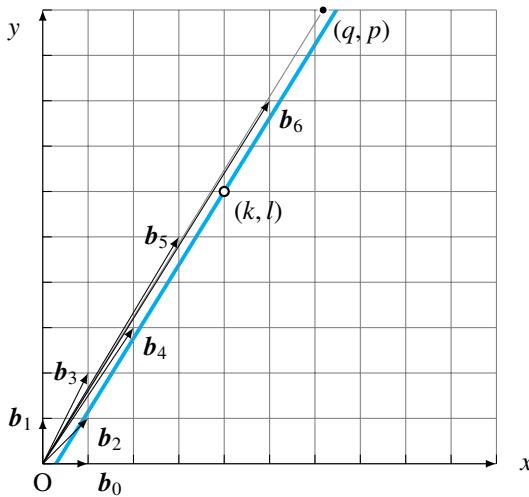
Fig. 12. Search space. The problem of finding the collision time for two planets is equivalent to finding the grid point (k, l) in the small *blue* band that is closest to the origin. The horizontal and vertical axes are given by $x = (t - t_1^1)/T_1$ and $y = (t - t_2^1)/T_2$ in Eq. (13), respectively. The slope is the ratio $p/q = T_1/T_2$ of orbital periods. The center of the intersection with the horizontal axis is $T_2/|t_1^1 - t_2^1|$, where $|t_1^1 - t_2^1|$ is the difference between the time-of-passage of the collision point for the two planets. The width of the band depends on the minimal distance of the orbits and the planet radii. The solution can be rapidly found using the bases $\{\boldsymbol{b}_{2n}, \boldsymbol{b}_{2n+1}\}$.
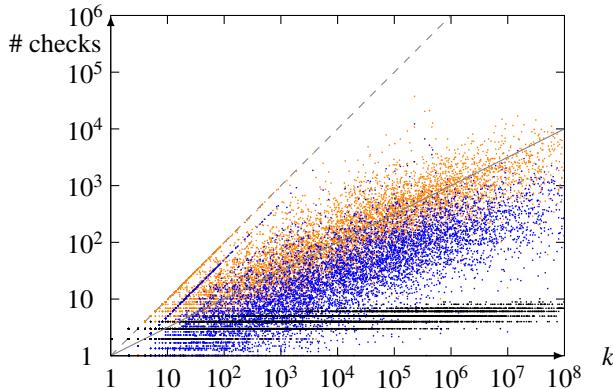

Fig. 13. Numerical checks needed to calculate the exact collision time for random collision pairs. Here we show the number of checks (*orange*), the number of iterations (*black*), and the average number of checks per iteration (*blue*) performed by the algorithm. On the horizontal axis is the number of orbits k before the collision, which is the solution calculated by the iteration scheme. The *gray* line is #checks $= k^{1/2}$ and shows the trend.

We assume, without loss of generality, that $T_1 > T_2$. Consequently, $p > q > 0$ and $p > 1$. Equation (14) says that we need an integer linear combination of the irrationals p and q that is within a distance δ from 1. We therefore need to find the point in the grid $\mathbb{N}^2$ closest to the origin that lies in between the lines

$$x = \left(\frac{q}{p}\right)y + \frac{1 - \delta}{p} \quad \text{and} \quad x = \left(\frac{q}{p}\right)y + \frac{1 + \delta}{p}$$

in the first quadrant of the *xy*-plane ($\mathbb{R}^2$). This is shown in Fig. 12. The (horizontal) width of the narrow band is δ/p, which will be of order of magnitude of s/a. For the following solution method to give the correct values of k and l, the numerical accuracy needs to be smaller than this number.

6.1. Deterministic collision time

Here we present an iteration scheme that finds (k, l) in $\log(q/\delta)$ steps. We will need the continued-fraction representation of p/q. This is found from the recursion relation that calculates the successive remainders:

$$q_0 = p, \quad q_1 = q, \quad q_{n+2} = (q_n \bmod q_{n+1}), \quad \text{for} \quad n = 0, 1, 2, \dots$$

This is similar to the Euclidian algorithm for finding the greatest common divisor. For rational p/q, the sequence becomes zero in finite steps, and for irrational p/q the sequence decreases to zero (Khinchin 1964; Rockett & Szüsz 1992):

$$0 < q_{n+1} < q_n \longrightarrow 0, \quad \text{as} \quad n \longrightarrow \infty.$$

The q_n are all integer linear combinations of p, q, and are therefore elements of $\mathbb{Z}$-span$\{p, q\}$. When we write $q_n = k_n p - l_n q$, then the rationals l_n/k_n are precisely the successive convergents of the continued-fraction representation.

Next, we define the bases $\{\boldsymbol{b}_n, \boldsymbol{b}_{n+1}\}$ for $\mathbb{Z}^2$, with slopes equal to the fractions:

$$\boldsymbol{b}_n = (-1)^n \binom{k_n}{l_n}, \quad \frac{l_{2n}}{k_{2n}} < \frac{p}{q} < \frac{l_{2n+1}}{k_{2n+1}}.$$

The slopes of the even base vectors increase and the slopes of the odd base vectors decrease to p/q. This is shown in Fig. 12. The sequence of remainders may be found from

$$\binom{q_{n+2}}{q_{n+1}} = \begin{pmatrix} -\lfloor q_n/q_{n+1} \rfloor & 1 \\ 1 & 0 \end{pmatrix} \binom{q_{n+1}}{q_n},$$

and base vectors can be found from the recursion relation:

$$\boldsymbol{b}_0 = \binom{1}{0}, \quad \boldsymbol{b}_1 = \binom{0}{1}, \quad \boldsymbol{b}_{n+2} = \boldsymbol{b}_n + \left\lfloor \frac{q_n}{q_{n+1}} \right\rfloor \boldsymbol{b}_{n+1}.$$

Therefore, each basis is related to the preceding basis by the transformation

$$\left(\boldsymbol{b}_{n+1} \middle| \boldsymbol{b}_{n+2}\right) = \left(\boldsymbol{b}_n \middle| \boldsymbol{b}_{n+1}\right) \begin{pmatrix} 0 & 1 \\ 1 & \lfloor q_n/q_{n+1} \rfloor \end{pmatrix}.$$

The transformation matrix is unimodular, which implies that the transformation is a bijection between the points of $\mathbb{Z}^2$. The integer coordinates (x_n, y_n) in each base are defined by

$$\binom{x}{y} = x_n \boldsymbol{b}_n + y_n \boldsymbol{b}_{n+1},$$

which means that $x_0 = x$ and $y_0 = y$. Substitution in Eq. (14) gives us the inequalities

$$1 - \delta < x_0 p - y_0 q = (-1)^n (x_n q_n - y_n q_{n+1}) < 1 + \delta.$$

Consider the following bases, which are composed of the successive even- and odd-numbered vectors:

$$\{\boldsymbol{b}_0, \boldsymbol{b}_2\}, \quad \{\boldsymbol{b}_1, \boldsymbol{b}_3\}, \quad \{\boldsymbol{b}_2, \boldsymbol{b}_4\}, \quad \{\boldsymbol{b}_3, \boldsymbol{b}_5\}, \dots$$

The positive spans (linear combinations) of these pairs partition the first quadrant of $\mathbb{R}^2$ into segments (see again Fig. 12). As we may assume that $(k, l) = (0, 0)$ is not a solution, the band intersects the *y*-axis at negative *y* and therefore lies below the segments spanned by the odd bases. However, in any even basis, the intersection of the band with a segment is a trapezoid. Because the union of all segments is the entire first quadrant, the solution of Eq. (14) must lie in one of the trapezoids. However, the pairs of even and odd base vectors do not form a unimodular

matrix, and therefore they do not necessarily span $\mathbb{Z}^2$. For this reason, we must instead search in the original bases $\{\boldsymbol{b}_n, \boldsymbol{b}_{n+1}\}$ inside the parallelogram of the strip between $y' = 0$ and the y'-value where the bottom of the strip intersects span$\{\boldsymbol{b}_{n+2}\}$. This is the top-right point and the bottom-right point of the subsequent parallelogram. The parallelogram has corner points:

$$(x', y') = \left(\tfrac{1-\delta}{q_n}, 0\right), \quad \left(\tfrac{1-\delta}{q_n} + \tfrac{(1+\delta)(q_n - q_{n+2})}{q_n q_{n+2}}, \tfrac{(1+\delta)(q_n - q_{n+2})}{q_{n+1}q_{n+2}}\right),$$
$$\left(\tfrac{1+\delta}{q_n}, 0\right), \quad \left(\tfrac{1+\delta}{q_{n+2}}, \tfrac{(1+\delta)(q_n - q_{n+2})}{q_{n+1}q_{n+2}}\right),$$

where n is even and the coordinates are defined by

$$\begin{pmatrix} x \\ y \end{pmatrix} = x'\boldsymbol{b}_n + y'\boldsymbol{b}_{n+1} = \begin{pmatrix} x'k_n - y'k_{n+1} \\ x'l_n - y'l_{n+1} \end{pmatrix}.$$

Successive parallelograms overlap. One needs to search one entire parallelogram first before going to the next, because the integer point closest to the origin will be found at the earliest occasion. Also, for each y'-value, only the lowest integer x'-value to the right of the left line segment needs to be checked. [3]

When the calculated orbital periods T_1 and T_2 have a ratio close to that of two small integers, the planets could be in mean-motion resonance and may never collide (as is the case with Neptune and Pluto). The method neglects these cases, and will erroneously find a collision at a high k number. If the continued fraction is actually finite (because p/q is rational), the strip has an integer slope at the final step. Figure 13 shows the results of a numerical test with random collision pairs. We find that the total number of checks grows as $k^{1/2}$, while the number of iterations grows as $\log k$ with the solution k.

6.2. Stochastic collision times

For small δ and irrational p/q, a generic solution (k, l) will form a pair of large integers, with

$$\frac{l}{k} \approx \frac{p}{q}.$$

The precise value depends very sensitively on q and p. If we assume that we cannot obtain the required numerical accuracy to find the exact solution, we may use a statistical approach. The integer points (k, l) are uniformly distributed over the plane. If we assume that the points are statistically independent (this it clearly an approximation valid for $\delta \ll 1$), the distribution is that of a Poisson point process. The probability that there is a grid point (k, l) between the lines with k in the interval $[x, x + \mathrm{d}x]$ is equal to the area of the small parallelogram. This area is $(2\delta/q)\mathrm{d}x$. Now, the area between the lines below $k = x$ is given by

$$\frac{2\delta}{q}x - \frac{2\delta - 4\delta^2}{pq} \approx \frac{2\delta}{q}x, \quad \text{for} \quad x \gg 1.$$

Therefore, the respective probabilities for the solution k to be found above x and below x are

$$\text{Prob}(k > x) = \lim_{\mathrm{d}x \to 0} \left(1 - \frac{2\delta}{q}\mathrm{d}x\right)^{x/\mathrm{d}x} = \mathrm{e}^{-2x\delta/q},$$
$$\text{Prob}(k \le x) = 1 - \mathrm{e}^{-2x\delta/q}.$$

---

[3] It was pointed out by Schouten (2022) that by looping through the integer x values instead of the integer y values the number of checks is significantly lower.

The latter formula is the cumulative distribution function for k. We obtain a realization by generating a random real number ξ inside the interval $[0, 1]$ and use

$$k = \frac{(-\log \xi)q}{2\delta}, \quad l = \frac{(-\log \xi)p}{2\delta} - \frac{1}{q}.$$

Because these values are large, rounding off to the nearest integer is not important. The time of the collision is then given by

$$t = t_1^1 + kT^1 = t_1^1 + \frac{(-\log \xi)T_1 T_2 |\boldsymbol{v}_1 \times \boldsymbol{v}_2|}{2\sqrt{(s_1 + s_2)^2 - |\boldsymbol{r}_2 - \boldsymbol{r}_1|^2}\, v_{12}}.$$

The formula for the average waiting time $\bar{k}T^1$ is precisely the reciprocal of the collision probability for one orbit. The fact that this reproduces the formula Eq. (23) in Öpik (1951), Eq. (29) in JeongAhn & Malhotra (2017), and Eqs. (2-3) in Diserens et al. (2020) for this probability P validates our Eq. (14).

The Monte Carlo method for finding the collision time also follows from assuming homogeneous distributions of the mean anomalies of two fixed Kepler orbits, as in Öpik's scheme. This method assumes large k, implying that the initial crossing time t_1 cannot be precisely known (due to numerical inaccuracy or neglected physics effects). However, when the system contains one or more large planets, the collision or nearby passage could happen after a few revolutions, that is, for small k.

6.3. Including orbital precession

Our method for calculating the collision time outlined in the previous subsection assumes perfect Kepler orbits. However, if one intends to make accurate predictions over long timescales, the slow precession of the periapsis and of the orbital plane cannot be neglected. For example, the perihelion shift for the planet Jupiter in one revolution is about twice the planet's diameter (see Fig. 14 for the precession rates for the Solar System planets). For a planet ring system, precession is mainly due to the oblateness of the planet. Hence, if one is not just interested in statistical averages, then the secular dynamics must be included on the orbital timescale.

A method to remedy this problem is to numerically propagate the Laplace-Lagrange equations. The time-steps are now set by the secular timescale $\mathrm{d}t \lessapprox T_{\text{prec}}$. There are N^2 terms in the system of differential equations and there are $N^2\epsilon/2$ collision possibilities, which need to be evaluated. The theoretical algorithmic efficiency is shown by the orange curve in Fig. 8.

It may be possible to include the collision detection in the following way. At each time-step (now shorter than the collision time), one calculates the instantaneous change is the orbital elements ϵ, ϖ, I, Ω. One expresses the resulting linear change in the points $\boldsymbol{r}_j$ and velocities $\boldsymbol{v}_j$ near the points of closest approach, as linear functions in the passage numbers k and l. The step where the minimal distance d is compared with the sum $s_i + s_j$ is skipped. Instead one directly uses the modified inequality Eq. (14). This would then lead to a problem from integer linear programming, as before. For this modified case, we would expect the two lines bounding our search domain to be nonparallel. The solution for the collision problem has a spatial accuracy of less than a planet radius on the longer timescale where the perihelion shift can be approximated as linear motion. A thorough development of this idea is a possible direction for follow-up research.
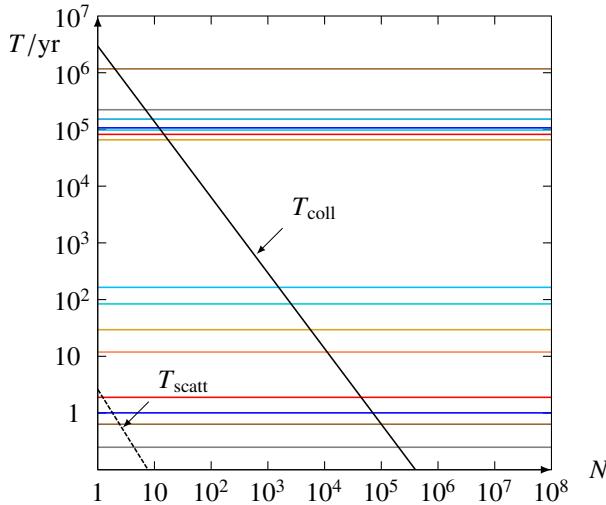
Fig. 14. Timescale vs. particle count N. The horizontal lines are the orbital periods T_j and the precession periods $T_{\mathrm{prec}\,j}$ of the Solar System planets (*gray*, *brown*, *blue*, *red*, *coral*, *golden*, *turquoise*, *azure* are Mercury through Neptune) from Murray & Dermott (2009). The *black* curve shows the collision time T_{coll} estimated for a disk with $a = 4\mathrm{au}$, $I = .1$, and a mass of three Earths containing N particles of Earth density. For $N < 10$, the collision time is comparable to the precession time ($10^5\mathrm{yr}$). The radius of gravitational influence is roughly 10^3 times the planet radius s; this determines T_{scatt}, the *dashed* curve.

7. Conclusions

We describe an algorithm that simulates collisional Keplerian systems: N bodies in the Coulomb potential of a central mass. The method uses the orbital elements and has three basis ingredients, of which the third is novel: (i) For a new particle i, a small set of possible collision candidates j is selected using the apoapsis/periapsis filter. (ii) The MOIDs between the particle pairs (i, j) can be approximated numerically. (iii) For the pairs (i, j) on a collision trajectory, one can obtain the collision time $t_{(i,j)}$ with integer linear programming. During the simulation, sorted lists of the particles and the collision pairs are maintained. The algorithm steps from one collision to the next as it updates the particle orbits and collision possibilities.

We show that the problem of finding the collision time is mathematically equivalent to the problem in integer linear programming of finding the grid point (k, l) in $\mathbb{N}^2$ between two parallel lines that is closest to the origin. The exact solution uses the continued-fraction representation of the ratio T_i/T_j of the orbital periods.

Because at most N new collision possibilities have to be added to the list, less than N interactions need to be considered at each step. The length of the collision list is $O(N^2 s/a)$ and the total number of collisions is $O(N^2 s^2/a^2)$, resulting in an algorithmic efficiency of $O(N^4 s^3/a^3)$. This may be compared to the efficiency $O(N^{4/3} \log N)$ of a numerical integration propagator with collision detection (tree code or spatial hashing), which is independent of the particle radius s. In the astronomical applications, the radii are usually small compared to the orbits. The collisions are therefore rare, and the proposed collision-detection method can be fast. However, the perturbations we neglect become increasingly important, and, at the same time, the result becomes progressively sensitive to the initial state. Needless to say, including collisions is important, even if they are rare. Studying statistics of outcomes of the dynamics requires many simulations with near-identical initial states. For relatively small particle numbers, say for $N < 10^6$, the individual realizations can be fast.

References

Adams, R. A. & Essex, C. 2021, Calculus A Complete Course, 10th edn. (North York, Ontario: Pearson Education)
Aliberti, G. D. 2022a, Analytical propagator with collision detection for Keplerian systems, Astrophysics Source Code Library, record ascl:2211.002
Aliberti, G. D. 2022b, Bachelor thesis, TU Delft, Netherlands
Baraff, D. & Witkin, A. 1992, SIGGRAPH Comput. Graph., 26, 303–308
Barnes, J. & Hut, P. 1986, Nature, 324
Barnes, J. E. 1990, Journal of Computational Physics, 87, 161
Bentley, J. L. 1975, Commun. ACM, 18, 509–517
Bodenheimer, P. H., Laughlin, G., Różyczka, M., & Yorke, H. W. 2007, Numerical methods in astrophysics: an introduction, Series in astronomy and Astrophysics (New York London: Taylor & Francis)
Burtscher, M. & Pingali, K. 2011, GPU Computing Gems Emerald Edition
Dehnen, W. & Read, J. 2011, European Physical Journal Plus, 126, 1
Diserens, S., Lewis, H. G., & Fliege, J. 2020, Journal of Space Safety Engineering, 7, 274
Goldstein, H. 1964, Classical Mechanics, ninth dover printing, tenth gpo printing edn. (New York: Dover)
Greengard, L. 1990, Computers in Physics, 4, 142
Gronchi, G. F. 2005, Celestial Mechanics and Dynamical Astronomy, 93, 295
Hamada, T., Nitadori, K., Benkrid, K., et al. 2009, Computer Science - R&D, 24, 21
Hedo, J., Ruiz, M., & Pelaez, J. 2018, Monthly Notices of the Royal Astronomical Society, 479
Hoots, F. R., Crawford, L. L., & Roehrich, R. L. 1984, Celestial Mechanics, 33, 143
JeongAhn, Y. & Malhotra, R. 2017, The Astronomical Journal, 153, 235
Khinchin, A. Y. 1964, Continued Fractions (University of Chicago Press)
Manley, S. P., Migliorini, F., & Bailey, M. E. 1998, A&AS, 133, 437
Meagher, D. 1980
Meagher, D. 1982, Computer Graphics and Image Processing, 19, 129
Milisavljević, S. 2010, Serbian Astronomical Journal, 180, 91
Murray, C. & Dermott, S. 2009, Solar System Dynamics (New York: Cambridge University Press)
Öpik, E. J. 1951, in Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences, Vol. 54, JSTOR, 165–199
Rockett, A. & Szüsz, P. 1992, Continued Fractions (Singapore: World Scientific)
Rokhlin, V. 1985, Journal of Computational Physics, 60, 187
Savransky, D., Cady, E., & Kasdin, N. J. 2011, The Astrophysical Journal, 728, 66
Schouten, A. 2022, Bachelor thesis, TU Delft, Netherlands
Segan, S., Milisavljević, S., & Marceta, D. 2011, Acta Astronomica, 61, 275
Soliman, P. 2022, Bachelor thesis, TU Delft, Netherlands
Warren, M. & Salmon, J. 1993, in A parallel hashed Oct-Tree N-Body algorithm, 12– 21
Wiźniowski, T. & Rickman, H. 2013, Acta Astron., 63, 293

Appendix A: Closest approach

Consider two non-interacting particles in linear motion:

$$\boldsymbol{r}_1(t) = \boldsymbol{r}_1 + \boldsymbol{v}_1 t, \quad \boldsymbol{r}_2(t) = \boldsymbol{r}_2 + \boldsymbol{v}_2 t.$$

We call the intitial distance vector and the relative velocity:

$$\boldsymbol{d} = \boldsymbol{r}_2 - \boldsymbol{r}_1, \quad \boldsymbol{u} = \boldsymbol{v}_2 - \boldsymbol{v}_1.$$

We want to decide if there is a collision in the interval $[0, \mathrm{d}t]$. Therefore, we calculate the distance between the particles at $t = 0$ and at $t = \mathrm{d}t$ to see if there is a collision at the endpoints:

$$|\boldsymbol{d}| < s_1 + s_2, \quad |\boldsymbol{d} + \boldsymbol{u}\,\mathrm{d}t| < s_1 + s_2.$$

If not, the only possibility for a collision on the interval is that the distance obtains a minimum on the (interior) of the interval. This means that the relative velocity in the direction between the particles is first decreasing and then increasing:

$$\boldsymbol{u} \bullet \boldsymbol{d} < 0, \quad \boldsymbol{u} \bullet (\boldsymbol{d} + \boldsymbol{u}\,\mathrm{d}t) > 0.$$

The time of minimal distance is at

$$t = \frac{-\boldsymbol{u} \bullet \boldsymbol{d}}{u^2}, \tag{A.1}$$

which is then indeed between 0 and $\mathrm{d}t$. We then decide if the distance at this time t is smaller than the sum of the radii. When we substitute t back into the Equation for the distance, we find that it is equal to the component of $\boldsymbol{d}$ perpendicular to the direction of $\boldsymbol{u}$. Hence, the condition for a collision is equivalent to:

$$\frac{|\boldsymbol{u} \times \boldsymbol{d}|}{|\boldsymbol{u}|} < s_1 + s_2.$$

Appendix B: Orbital elements

Consider a single particle of mass m in a Kepler orbit about the central mass M. The orbit is an ellipse in a fixed plane. The angular momentum vector is defined by

$$\boldsymbol{L} = \boldsymbol{r} \times m\boldsymbol{v}.$$

The *Laplace-Runge-Lenz vector* is proportional to the dimensionless eccentricity vector:

$$\boldsymbol{\epsilon} = \frac{\mathbf{LRL}}{GMm^2} = \frac{\boldsymbol{v} \times \boldsymbol{L}}{GMm} - \frac{\boldsymbol{r}}{r}.$$

The orbit is fixed by the orthogonal pair of vectors $\boldsymbol{L}$ and $\boldsymbol{\epsilon}$.

For the problem of solving the MOID, we need a formula that expresses the velocity $\boldsymbol{v}$ along the orbit as a function of the position $\boldsymbol{r}$ and the orbital elements. Given $\boldsymbol{L}, \boldsymbol{\epsilon}, \boldsymbol{r}$, we equate

$$\boldsymbol{L} \times \left(\boldsymbol{\epsilon} + \frac{\boldsymbol{r}}{r}\right) = \frac{\boldsymbol{L} \times (\boldsymbol{v} \times \boldsymbol{L})}{GMm} = \frac{L^2 \boldsymbol{v}}{GMm},$$

and therefore the velocity can be expressed as:

$$\boldsymbol{v} = \frac{GMm}{L^2} \boldsymbol{L} \times \left(\boldsymbol{\epsilon} + \frac{\boldsymbol{r}}{r}\right). \tag{B.1}$$

The Equation for the energy is called the *vis-viva equation*

$$\frac{\text{Energy}}{m} = \frac{v^2}{2} - \frac{GM}{r} = -\frac{GM}{2a}.$$

With this Equation and *Kepler's third law*,

$$\omega^2 a^3 = GM,$$

the orbital elements a, ϵ and mean motion ω can be calculated from position and velocity:

$$a = \frac{1}{1 - \epsilon^2} \frac{L^2}{GMm^2} = \frac{1}{\dfrac{2}{r} - \dfrac{v^2}{GM}}, \quad \omega = \sqrt{\frac{GM}{a^3}}.$$

The orbit is parametrized by the true anomaly ν or the eccentric anomaly E. If we know the eccentric anomaly, we can calculate the time since periapsis from the *Kepler equation*

$$t = \frac{E - \epsilon \sin E}{\omega}. \tag{B.2}$$

The formula for the radial distance in terms of the parameters is

$$r = \frac{b^2}{a + c \cos \nu} = a - c \cos E. \tag{B.3}$$

The semimajor axes, a and b, the distance from the center to a focus, c, and eccentricity ϵ are related by

$$b = \sqrt{1 - \epsilon^2}\, a, \quad c = \epsilon a, \quad a^2 = b^2 + c^2.$$

The position vector and the velocity vector can now be expressed as:

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = r\mathscr{R} \begin{pmatrix} \cos \nu \\ \sin \nu \\ 0 \end{pmatrix} = \mathscr{R} \begin{pmatrix} a \cos E - c \\ b \sin E \\ 0 \end{pmatrix}, \tag{B.4}$$

and, using Eqs. (B.2) and (B.3),

$$\boldsymbol{v} = \dot{\boldsymbol{r}} = \frac{\omega a}{r} \mathscr{R} \begin{pmatrix} -a \sin E \\ b \cos E \\ 0 \end{pmatrix} = \frac{\omega a}{b} \mathscr{R} \begin{pmatrix} -a \sin \nu \\ a \cos \nu + c \\ 0 \end{pmatrix}. \tag{B.5}$$

In the expressions Eqs. (B.4) and (B.5), $\mathscr{R}$ is a constant rotation matrix, which can be expressed as a product of three elementary rotations

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

Here, ϖ is the argument of perihelion, I is inclination, and Ω is the ascending node. We have for the angular momentum and the eccenticity vectors,

$$\boldsymbol{L} = L\mathscr{R} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = L \begin{pmatrix} \sin \Omega \sin I \\ -\cos \Omega \sin I \\ \cos I \end{pmatrix}, \quad L = m\omega ab, \tag{B.6}$$

and

$$\boldsymbol{\epsilon} = \epsilon \mathscr{R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \epsilon \begin{pmatrix} \cos \varpi \cos \Omega - \sin \varpi \sin \Omega \cos I \\ \cos \varpi \sin \Omega + \sin \varpi \cos \Omega \cos I \\ \sin \varpi \sin I \end{pmatrix}, \quad \epsilon = \frac{c}{a}. \tag{B.7}$$

These five orbital elements a, ϵ, ϖ, Ω, I represent five independent conserved quantities. They can be found from the position and velocity vectors as follows. The equations at the beginning of this section give us $\boldsymbol{L}$ and $\boldsymbol{\epsilon}$ in terms of $\boldsymbol{r}$, $\boldsymbol{v}$. Then, the component $l = L_3$ can give us the angle I. The component L_1 can give us Ω, and ϵ_3 can give us the angle ϖ.

Appendix C: Elastic collisions

When planets approach each other so closely that the gravity they exert on each other is of the same strength as the gravity from the star, the mutual gravitational interaction cannot be neglected. The close encounter can be approximated as a scattering event. Here, for the short duration of the encounter, we neglect the gravity from the central star (and of all other particles in the system). As no energy is dissipated, the process is an elastic collision. It is described by a hyperbolic orbit for the relative coordinate $r_2 - r_1$. The hyperbola has parameters a, b, c, and is given by the following equations in Cartesian and in polar coordinates

$$\frac{(x-c)^2}{a^2} - \frac{y^2}{b^2} = 1, \quad x \le c - a, \quad r = \frac{b^2}{a + c \cos v}.$$

The parameter a is the semi-transverse axis of the hyperbola. It is a characteristic distance where the gravity between the two bodies becomes noticeable given the relative velocity. The semi-conjugate axis b of the hyperbolic orbit is equal to the impact parameter. The semi-focal separation is c (see Fig. 9). We have

$$a = \frac{G(m_1 + m_2)}{u^2}, \quad b = d = |d|, \quad c^2 = a^2 + b^2.$$

When the scattering between two planets is significant, the velocities of the two planets will change. The center-of-mass moves with velocity

$$v = \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2}.$$

In the center of mass frame, both velocities rotate over an angle $\pi - 2 \arctan(b/a)$. This is described by:

$$u' = \frac{b^2 - a^2}{c^2} u - \frac{2au}{c^2} d,$$

$$v'_1 = v - \frac{m_2}{m_1 + m_2} u',$$

$$v'_2 = v + \frac{m_1}{m_1 + m_2} u'.$$

Appendix D: Mathematica code

The following Mathematica code tests the deterministic collision-time algorithm. It first finds the solution by considering all integers k, and then uses the continued-fraction representation. Both procedures give the same result for random orbital periods.

```
tmax = 100;
T1 = RandomReal[tmax];
T2 = RandomReal[tmax];
If[T1 < T2, T1 = tmax - T1; T2 = tmax - T2];
t1 = RandomReal[T1];
t2 = RandomReal[T2];
deltat = Abs[t1 - t2];
p = T1/deltat
q = T2/deltat
del = deltat/10^5

qlist = {p, q};
n = 1;
While[qlist[[n + 1]] > del,
 qnm = qlist[[n]];
 n++;
 qn = qlist[[n]];
```

```
 AppendTo[qlist, qnm - Floor[qnm/qn] qn]
 ]
qlist

For[k = 1, k < 10^6, k++,
 l = Ceiling[p k/q - (1 + del)/q];
 If[l < p k/q - (1 - del)/q, Break[]]
 ]
Plot[
 {p x/q - (1 - del)/q, p x/q - (1 + del)/q},
 {x, k - del, k + del},
 PlotRange -> {1 - del, 1 + del},
 GridLines -> {{k}, {1}},
 AspectRatio -> 1, Axes -> None,
 Frame -> None
 ]
k
l

q0 = p; q1 = q;
k0 = 1; k1 = 0;
l0 = 0; l1 = -1;
plot = {};
found = False;
For[n = 0, Not[found], n++,
 a0 = Floor[q0/q1];
 q2 = q0 - a0 q1;
 If[q2 == 0, Break[]];
 k2 = k0 - a0 k1;
 l2 = l0 - a0 l1;
 a1 = Floor[q1/q2];
 For[x = Ceiling[(1 - del)/q0],
  x < (1 + del)/q2, x++,
  y = Max[0, Ceiling[(q0 x - 1 - del)/q1]];
  If[1 - del < x q0 - y q1,
   k = x k0 - y k1;
   l = x l0 - y l1;
   found = True;
   Break[]
   ]
  ];
 q3 = q1 - a1 q2;
 If[q3 == 0, Break[]];
 k3 = k1 - a1 k2;
 l3 = l1 - a1 l2;
 q0 = q2; q1 = q3;
 k0 = k2; k1 = k3;
 l0 = l2; l1 = l3;
 ]
k
l

Plot[
 {p x/q - (1 - del)/q, p x/q - (1 + del)/q},
 {x, k - del, k + del},
 PlotRange -> {1 - del, 1 + del},
 GridLines -> {{k}, {1}},
 AspectRatio -> 1, Axes -> None,
 Frame -> None
 ]
```