# Betelgeuse's expanding Oort cloud: an interstellar threat?

Kilmetis Konstantinos, Jasper Mens,
Rahul Priyadarshan Ravichandran

December 23, 2022

**Abstract**

In this project, we performed simulations of the orbits of hypothetical Oort cloud analog objects belonging to the nearby red supergiant Betelgeuse. We found that the projected mass loss associated with the star going supernova results in significant expansion of the cloud in the galactic potential. Furthermore, we found that it may indeed be possible for some of these objects to reach the vicinity of the Sun, although this appears to be highly dependent on the specific orbits of both the Sun and Betelgeuse.

## 1 Introduction

In the wake of recent interstellar visitors such as 'Oumuamua, the question of where such objects come from, and how they were ejected has arisen. A common hypothesis is that such objects might have been ejected from their parent stars in a supernova explosion. One way to check this hypothesis could be to back-simulate the orbit of such objects through the galactic potential, and seeing if the path traces back to a supernova remnant. The problem with this method, however, is that close encounters to other stars can lead to highly chaotic scattering, which complicates the process significantly.

Another method is to infer the expected density of interstellar objects from simulations, and comparing this density to the observations. This approach is less burdened by chaotic scattering events, which comes at the cost of specificity to a single object. Nevertheless, it gives some measure of the chance of any given supernova launching an interstellar visitor towards the Sun, which would in turn allow one to deduce the most likely progenitor event for any given interstellar visitor. Furthermore, this method could allow us to predict the directions from which these objects would arrive at the Sun, which opens the door to targeted surveying of these regions. For this project, however, we decided to get ahead of the curve by employing this method to answer the question: *will interstellar asteroids launched by the imminent supernova explosion of Betelgeuse pose a threat to humanity?*

It is common knowledge that, this star, also known as *α-Orionis*[1] is projected to go supernova relatively soon (Guinan et al., 2019). This will undoubtedly be a spectacular sight to behold, but for all we know the fun may be short-lived. After all, a swarm of humanity-ending asteroids may have already started their journey toward the Earth. Given that the system is

[1] Or *Beet*, as we've endearingly taken to calling it.

only a few hundred light-years away, our hope is that getting on top of this question today buys us enough time to prepare for the worst.

### 1.1 Oort cloud analogs

We posit that, in this context, the most dangerous part of any star system must be its Oort cloud. Unfortunately, due to the inherent difficulty in observing a low density of small objects at wide distances from their host stars, we know very little about the properties (or even existence) of Oort cloud analogs around other stars. In order to formulate reasonable parameters for Betelgeuse's potential Oort cloud, we can only turn to simulations from previous studies on the subject, which are still, sadly, few and far between.

One such study (Portegies Zwart, 2021) found that the timescale for the formation of an Oort cloud should be on the order of 100 Myr (Portegies Zwart et al., 2021). As Betelgeuse is estimated to be $\lesssim 10$ Myr old, this would immediately disqualify Betelgeuse from having an Oort cloud. However, the study is only concerned with solar-mass stars, whose systems may evolve considerably differently from Betelgeuse's ($M \sim 20 M_\odot$).

Another potential issue is that this study, like most others, operates under the consensus hypothesis that Oort clouds are formed through the scattering of debris-disk objects by giant planets (Oort, 1950) . For as far as we know, the Betelgeuse system does not contain any giant planets in the first place, which means another process would have to be responsible. This may be problematic, as we do not know what kind of Oort cloud such a process would produce (assuming it exists in the first place).

All this is to say that the literature appears to point toward one conclusion, namely that it is unlikely that the Betelgeuse system contains an Oort cloud analog in the first place. Nevertheless, it remains possible. While further research may definitively put our minds to rest, we will operate under the assumption of the worst-case scenario for the remainder of this project.

We will begin by computing an order-of-magnitude estimate of the number of exo-Oort cloud objects we expect to come close to the Sun, after which we will go into detail on our choice of initial conditions and methods. Finally, we will show and discuss our results.

### 1.2 The Back of the Envelope

In this section, we will motivate the choice of parameters we require for our simulations using simple, order-of-magnitude estimates.

We assume that Betelgeuse has an Oort cloud similar to that of the Sun, in which case we expect it to have around $10^{12}$ objects. Out of these objects, we are interested in asking how many can end up within an arbitrary, albeit plausibly interesting, radial distance from the Sun; say 1 parsec. We arrive at an estimate for this number as follows:

When Betelgeuse goes supernova, we expect the Oort cloud to get unbound from the gravitational influence of Betelgeuse and orbit the galactic center subject to the galactic potential. If we look at Betelgeuse at the instant of the supernova, we expect to see the objects move radially outward with respect to Betelgeuse. If we assume this expansion to be isotropic, we expect that the flux of Oort cloud objects hitting a larger sphere centered around Betelgeuse varies according to the inverse-square law. If we then assume that Betelgeuse's Oort cloud is on the order of 1 pc across, and Betelgeuse is located at a distance on the order of 100 pc from the Sun (in particular, 170 pc (Joyce et al., 2020b)), we should expect to see 1 in $10^4$ objects to end up within a radius of 1 pc around the Sun. If the cloud contains $10^{12}$ asteroids, this then means $10^8$ asteroidal interlopers. Of course, simulating a trillion asteroids is out of the question, but if we run a simulation containing $10^6$ objects, we expect to see on the order of 100 objects within 1 pc of the Sun. This seems like quite a lot. Within 1 AU, however, this reduces all the way down to 1 in $10^{15}$. Based on this, we do not expect to see any asteroids with 1 AU of the sun.

Another quantity of interest is the duration of the simulation. For this, we are interested in knowing the initial orbital velocities of the Oort cloud objects. With this and the distance in hand, we can approximate the time they may take to reach the Sun; to an order of magnitude, this will be the duration of our simulation run. To find the orbital velocity, we make use of the expression for the orbital velocity in a circular Keplerian orbit:

$$v = \sqrt{\frac{GM}{R}} \tag{1}$$

For comparison, the Earth has an orbital velocity of 30 km s$^{-1}$ while orbiting the Sun with a mass of 1 $M_\odot$ and at a distance of 1 AU. For a typical Oort cloud object orbiting around Betelgeuse - which has a mass of around 20 $M_\odot$ - at a distance of, say 100000 AU, the orbital velocity is just given by:

$$v_{\mathrm{oort}} = 30\sqrt{\frac{20}{100000}} \text{ km s}^{-1} \approx 1 \text{km s}^{-1} \tag{2}$$

Since 1 km s$^{-1} \approx 10^{-6}$pc yr$^{-1}$, the Oort cloud objects will take roughly 170 Myr to have a chance of coming close to the Sun. This is our estimate for the total duration of the simulation runs.

## 2 Methods

Given the multi-scale nature of our problem, we elected to use the the Astrophysical Multipurpose Software Environment, best known as `amuse`[2]. `amuse` allows us to easily use sophisticated codes through a `Python` interface. What is more, it permits us to combine codes through usage of `bridge`, which turned out to be vital for this project.

We can divide our problem into five parts:

1. Evolving Betelgeuse to its current age using the stellar evolution code `SeBa` (Portegies Zwart and Verbunt, 1996; Toonen et al., 2012).

2. Adding an Oort cloud, and evolving the system until Betelgeuse goes supernova. (Meanwhile also letting the Sun and Betelgeuse orbit the galaxy.)

3. Allow the Oort cloud objects and the Sun to drift in the galactic potential.

4. If an object gets close to the Sun, save its position and velocity and count it as a detection.

5. Run a lonely-planet simulation of the orbits of the detected objects around the Sun.

The simulation passes through these 5 stages in order. At any stage, different physics and scales are relevant. In the second phase, the orbits of the Oort cloud objects around Betelgeuse enforce a small timestep, while in the third phase we can afford using a larger timestep since we are only simulating galactic orbits. In the fourth phase, there could arise a need for a higher definition in order to not miss a detection; should the timestep be too high, an asteroid could conceivably cross the entire

[2] Bet you didn't expect that.

detection sphere in one timestep and thus not get registered as a detection. Finally the comparatively tiny number of particles to simulate during the fifth phase allows for the smallest timestep yet.

## 2.1 Initial Conditions

There are three objects in our simulation: Betelgeuse, its Oort Cloud, and the Sun. Each of which comes with its own set of initial conditions, which we will now discuss.

### Stellar Positions and Velocities

While the original plan was to take the true values for the Sun's position and velocity in the galactic frame, and combine these with Gaia measurements of Betelgeuse's relative position and velocity, this proved unfeasible. The first problem was that, since Betelgeuse is too bright for Gaia and it's centre of light is in constant flux[3], there were significant uncertainties on the measurements of Betelgeuse.

However, these 'physical' initial conditions could only ever be truly accurate in an equally physical galactic potential, which we did not use for this project. Thus, we elected to use an approximate value for the position of the Sun ($x = 8400$ pc(Binney and Tremaine, 2008), $y = 0$), and simply compute the circular velocity at that distance for our simplified potential.

For Betelgeuse, however, things got slightly more complex. We know that radially, it lies approximately 168 parsec (van Leeuwen, 2007) away from the Sun. Furthermore, for the sake of consistency, we want to initialize its velocity as circular as well. Then, if we know its galactic longitude, we can do a bit of trigonometry to compute its position and velocity with respect to the galactic plane. A diagram of this setup can be seen in figure 1. We found the angle $\phi$ more intuitive to work with than the galactic longitude $l$, so that is the angle that we will be using from now on. Note that it is nothing more than the supplementary angle of $l$.

[3]We thank Prof. Dr. A. De Koter for this observation.

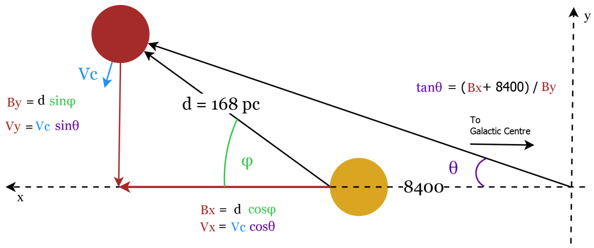


**Figure 1:** The geometry of the positions of the Sun and Betelgeuse in the (x,y) plane. The red and yellow circles represent Betelgeuse and the Sun, respectively. From any combination of $d$ and $\phi$, $B_x$ and $B_y$ can easily be derived. Then, the components of the galactocentric velocity are trivial to deduce.

Finally, since both stars reside in the thin disk of the galaxy, we assumed the $z$ coordinate of the Sun to be 0, and because Betelgeuse's galactic latitude is relatively small as well we extend this assumption to Betelgeuse as well. Thus, we initialise both stars in circular orbits with $z = 0$.

### Betelgeuse: Age, $M$, $M_0$, and $z$

In order to be able to model Betelgeuse's mass loss as accurately as possible, we needed a combination of current age, initial mass, current mass, and metallicity that agreed with the literature as closely as possible. Because the stellar evolution integrators that were used to obtain the literature values are far more nuanced than `SeBa`, no single combination of `SeBa` parameters can be perfect. As such, the time until Betelgeuse goes supernova deviates significantly from the literature value. The combination of parameters that we settled on can be seen in table 1

| Parameter | Literature | SeBa |
|---|---|---|
| Initial Mass | $18 - 21 M_{\odot}$ | 21 $M_{\odot}$ |
| Current Mass | $16.5 - 19 M_{\odot}$ | 18.2 $M_{\odot}$ |
| Metallicity | $0 - 0.2$ | 0.024 |
| Age | $8.0 - 8.5$ Myr | 8.4 Myr |
| $t_{SN}$ | $< 10^5$ yr | $5.3 \cdot 10^5$ yr |

**Table 1:** `SeBa` parameters compared to the literature (Joyce et al., 2020a; Dolan et al., 2016)

### The Oort Cloud

For the positions and velocities of the Oort cloud objects, we used the `amuse` built-in function `generate isotropic cloud`. This function allows the user to specify, among other things, a range of semi-major axes and a central mass, and uses these parameters to randomly generate a specified number of objects with a random combination of orbital elements within the set boundaries.

As we mentioned in the introduction, we know very little about the relation between Oort cloud characteristics and stellar properties. Thus, we are left to make educated guesses. Thus, we assume that the minimum semi-major axis should scale linearly with stellar mass, while the maximum semi-major axis should scale with the Hill radius of the star. Analytically, the Hill sphere scales with $M^{1/3}$, where $M$ is the stellar mass, thus we assume that the outer semi-major axis does the same.

Assuming that our own Oort cloud stretches from $\sim 3 \cdot 10^3$ AU to $\sim 2 \cdot 10^5$ AU (Morbidelli, 2005), we thus arrive at $a_{\text{min}} = 60\ 000$ AU and $a_{\text{max}} = 200\ 000$ AU.

For the central mass, we input the present-day mass of Betelgeuse, and we set the total cloud mass to 0, since we want the objects to behave as test-particles.

Finally, we set the minimum pericentre to $2 \cdot 10^4$ AU, as this confined the eccentricities of the orbits to reasonable values.

## 2.2 Custom Field Codes

One of the unique and powerful assumptions that we may make, is to ignore the gravitational forces between Oort objects, i.e. set their masses to zero. This reduces the time complexity of the problem to $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$. Of course, just setting the masses to zero and passing the objects to a N-body solver like `Hermite` does not reduce the complexity, it only makes the computations quicker. Therefore, we found ourselves in need of a custom solver that could take advantage of the nature of the problem.

The easiest way to have our cake and eat it too - i.e. have a linear solver and stay within the `amuse` ecosystem - was to "hijack" `bridge`.

Most of the time, `bridge` is used to couple ("bridge") two or more existing codes from the `amuse` framework. However, it is entirely possible to instead couple two or more custom-made codes, which was exactly what we needed for this project. In essence, any `python` class that contains the method `get_gravity_at_point` can be used as a field code in `bridge`. Then, if the class also contains the attributes `particles` and `model_time`, and a method called `evolve_model`, it qualifies as a gravity code. The upshot is that all of these methods are black boxes for all `bridge` knows, which means you have full control over what the codes do. We will walk through how we leveraged this capability to create each of our custom codes to our requirements in order of complexity, starting with the galactic potential code.

Our model of the Milky Way is the simplest code that was used for this project. The model itself is axisymmetric, with separate contributions from the bulge, disk, and halo, and it is identical to the one described in Dehnen et al. (1998) (Dehnen and Binney, 1998) The class only contains the first of the aforementioned methods, as it is just a static potential. They return the potentials and accelerations at the given coordinates as dictated by an analytic function. (Dehnen and Binney, 1998)

Additionally, a function is included that computes the circular velocity at a certain distance from the center. This function does not interface with `bridge` in any way, though.

Next up: the test particle code. In a way, this code can be seen as the complement to the galactic potential, in that it does not generate a field, and instead only moves test particles around. Because `bridge` takes care of updating the velocities of the particles based on the other codes in the system, the test particle class only needs to move its particles according to their velocities. This was implemented by simply multiplying the velocities by some timestep, and adding the result to the existing positions. Importantly, though, the `get_gravity_at_point` method is still present, but it always returns 0.

Finally, the code representing Betelgeuse. This is a synthesis of the previous two, in that it contains both a particle whose position gets updated, as well as a non-trivial gravitational potential. This potential is a simply Newtonian model, with the notable caveat that, unlike in the galactic potential, which is always centered at the origin, it is centered around the position of the particle stored in the `particles` attribute.

On top of this, the star class also contains an instance of a stellar evolution code. (Because we are using `SeBa`, which is very lightweight, there is no reason to explicitly couple it to the star using `bridge`.) All together then, the `evolve_model` method evolves the star's mass using its stellar evolution code instance, and updates its position.

## 2.3 The Main Simulations

The simulations were split up into two different parts: pre-supernova, and post-supernova. The reasoning behind this approach is that the orbits of the Oort cloud objects are far tighter during this time than after the supernova (which is, of course, the underlying premise of this entire project). Thus, a smaller timestep was needed to accurately model the dynamics. After the supernova, all Oort cloud objects that remained bound could be removed from the simulation, which meant that we only had to concern ourselves with modelling the orbits around the Milky Way for the rest of the run. This was implemented with the help of the `SeBa.particles` property `stellar_type`. Once this parameter indicated that Betelgeuse had turned into a black hole, we moved on to the next leg of the run.

During the post-supernova regime, we implemented the capability for a (marginally) dynamic timestep. Once per Myr, the distance to the nearest Oort cloud object is computed. Once this distance gets small enough, the proper encounter detection is automatically enabled and the timestep for the simulation is changed to one that is limited by the detection radius and relative velocities. The upshot of this system is that the user can essentially 'set it and forget it', without needing any prior knowledge on when encounters might be likely.

## 2.4 Lonely Planet

During the main simulations, the Sun was implemented as a test-particle. The reasoning behind this was that adding an extra potential to the system would unnecessarily slow things down, especially considering the fact that the detection radius far surpassed the area of influence of the Sun's gravity anyway. Moreover, the timestep needed to accurately resolve the encounters would be significant. Therefore, we opted for a separate lonely-planet run for resolving the encounters.

As mentioned the positions and velocities of the detected asteroids were written out to a file during the simulations. Also included in this data was the model time at which the detection took place. This allowed us to simply read the positions and velocities and assign them to a set of test-particles. This time, however, the Sun was implemented as a star, and Betelgeuse and the galactic potential were left out of the equation under the assumption that their effects would be negligible. Due to the strongly decreased number of particles, we could afford to set a comparatively tiny timestep of 0.01 Myr. Furthermore, the duration of the run could be truncated by initialising all of the asteroids at time $t = 0$, and manually adding the original detection time after the fact if needed.

The main function of the lonely planet simulations for our purposes was to find the aphelion distances and times, simply by keeping a running tally of the minimum distance for each particle, along with the corresponding model time. Although this could also be done analytically, we felt that this method was better suited to our application, and provide an easy path for future improvements.

## 2.5 Parameters

### Detection radius

There were several considerations involved in choosing an appropriate detection radius value. On the one hand, the radius must be small enough such that the detected asteroids would experience the gravitational influence of the Sun. On the other hand, however, it must catch enough objects to allow for some statistical analysis. Furthermore, the detection radius also constrains the timestep of the simulation (together with the incident velocities of the Oort cloud objects) through the crossing time of the detection sphere. After all, if the asteroids a velocity of $> 2$ detection radii per timestep, they can just 'hop over' the Sun without being detected. In this way, a smaller detection radius slows down the simulation.

In this project, our chief concern was to obtain a significant sample of asteroids. Therefore, informed by our back-of-the-envelope estimate of 1 in $10^4$ Oort cloud objects within 1 pc, we decided to simply set the detection radius to 1 pc.

### Timestep

As mentioned, there were three timesteps to consider. We will discuss these in chronological order.

Because the pre-supernova regime was both the shortest (in terms of model time), and the most demanding (many close encounters with Betelgeuse), we had both the liberty and the necessity for a stringent timestep. Informed by the period of the tightest possible orbit ($a = 60 \cdot 10^3$ au), we decided on a a timestep of 0.001 Myr (corresponding to 2000 points per orbit).

Post-supernova, any and all asteroids that were still orbiting Betelgeuse were deleted. As such, the focus of the simulation shifted from orbits around a star to orbits around the galaxy. Not only that, but these orbits were also far less eccentric, which also simplified things. Thus, the initial assumption was that a timestep of 1 Myr would be sufficient, sampling $> 200$ points per orbit around the Milky Way. However, because Betelgeuse remained present in the simulations, scattering through close encounters was still possible. Thus, we opted for a more strict timestep size of 0.1 Myr.

The timestep choice for the collision detection was informed by the detection radius, as well as the back-of-the-envelope estimate for the velocities of the objects. If we expect these objects to have a velocity on the order of parsecs per Myr, then we know that a timestep of 1 Myr would likely be too large for a detection radius of 1 pc. Thus, we set it at 0.1 Myr.

# 3 Results

Across 25 runs, comprising $10^6$ Oort cloud objects each, a total of $70.5 \cdot 10^3$ asteroids were detected within 1 pc of the Sun. (Note that this is using the adjusted initial position and velocity for Betelgeuse.) In terms of percentages, this comes down to 0.28%, or 1 in 354. This, however, is not the complete story. In this section, we will present a brief statistical analysis of these detected particles.

### Relative positions

Figure 2 shows a 2D histogram of the relative x,y coordinates of the detected particles with respect to the Sun. We can see that the majority of asteroids is detected in the lower left hand side, which is consistent with the overall motion of the Betelgeuse system relative to the Sun.

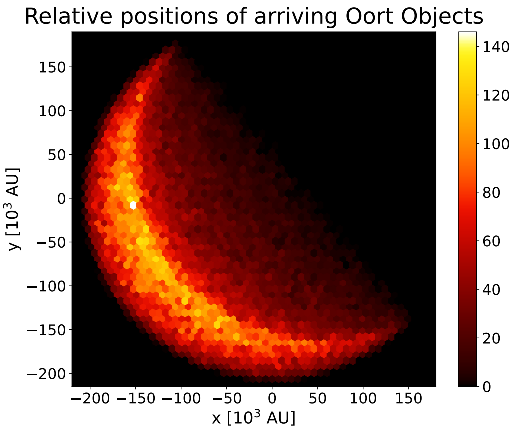


**Figure 2:** Histogram of the relative (x,y) positions of detected asteroids w.r.t. the Sun.

### Time of arrival

If we now turn our attention to the arrival times of these objects (figure 3), we can see a simple gaussian curve appear, centered around 55 Myr with a FWHM of approximately 5 Myr.

### Relative velocities

Likewise, the magnitudes of the relative velocities appear to neatly follow a gaussian curve, centered around 3.22 or so kilometers per second.

### Aphelion Distance

Figure 5 shows the cumulative distribution of closest separations for the asteroids, as dictated by an analytical computation of the Keplerian orbital elements. Although none of the asteroids are found to have aphelion distances $< 1000$ au, we find that the CDF is well-described by a quadratic relation with coefficient $2.38 \cdot 10^{-11}$, which allows us to extrapolate. According to the quadratic fit, then, 1 in $42 \cdot 10^3$ asteroids that reach 1 pc has an aphelion of $< 1000$. More trivially, we can extrapolate that 1 in $4.2 \cdot 10^{10}$ asteroids within a parsec gets to $< 1$ au. Combining this with the fraction of asteroids that gets within a parsec in the first place, we get a final ratio of 1 in $1.48 \cdot 10^{13}$ Oort Cloud objects possibly posing a threat to the Earth.

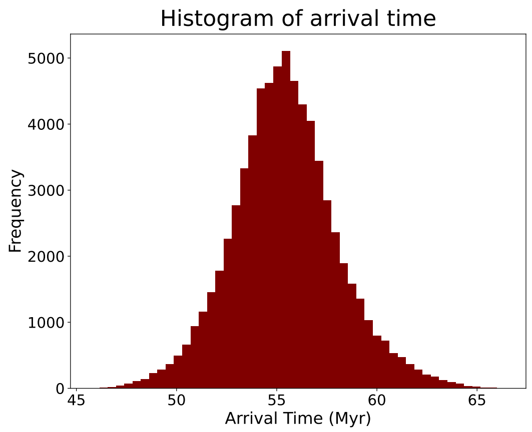


**Figure 3:** Histogram of the time of perihelion, computed numerically in a lonely-planet simulation

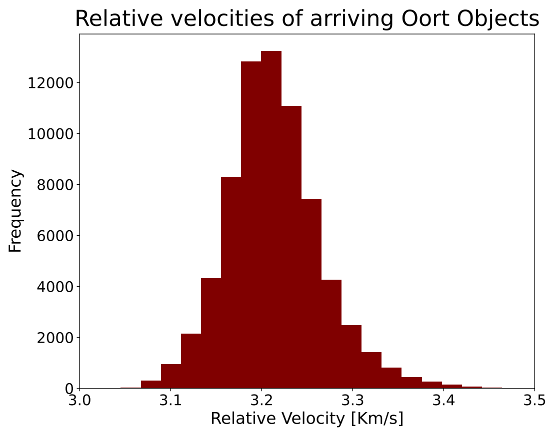


**Figure 4:** Relative velocities at time of detection

### CPU time

All told, each run, consisting of $10^6$ Oort cloud objects simulated for for 75Myr took well under an hour to complete on our (reasonably modern) computers. As for the lonely planet simulations, those only took on the order of seconds, using a timestep of 0.01Myr.

## 4 Discussion

We started by assuming that Betelgeuse has an Oort cloud, and that its imminent supernova may pose a threat to Earth by letting this cloud of asteroids become unbound and make its way to the Solar System. We found the number of these Oort cloud objects that gets close to the Solar System to be highly sensitive to the initial conditions. As a result, while the measured position and velocity of Betelgeuse did not result in any interstellar visitors, only a small deviation from the measurements was needed before the Sun crossed orbits with the expanding Oort cloud. In this scenario, we found that 1 in $1.48 \cdot 10^{13}$ Oort cloud objects would come close to the Earth. This is far greater than the initial estimate of 1 in $10^{15}$. In this section, we try to make sense of this number and interpret the statistics we obtained. Furthermore, we see that our code is quite reusable in helping us explore different astrophysical applications, which we shall also outline in this section.

### 4.1 Validity of assumptions

We started by assuming that around $10^{12}$ objects to be present in the Oort cloud of Betelgeuse, but that modelling approximately $10^6$ objects would be sufficient for our purposes of sampling the particles that get within 1 pc of the Sun. We found the chances of getting a detection to be highly sensitive to the angle $\phi$. $\phi = -34.5°$ yielded $70.5 \cdot 10^3$ detections within 1 pc of the sun, while only a slight offset was needed to reduce that number to zero. This suggests that, at least in this setup, whether or not an interstellar visitor reaches the sun is a simple matter of orbital alignment. Perhaps if the simulations were continued for several orbits around the galaxy, the cloud would have expanded enough to offset this problem. Scattering by nearby stars, which were not included in this project, could also contribute to the spreading of the cloud.

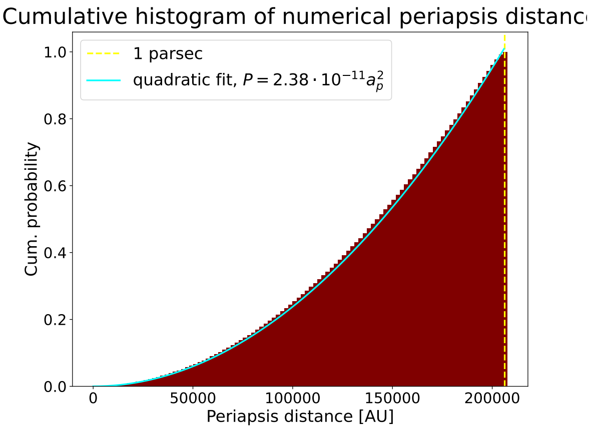


**Figure 5:** CDF of the analytical perihelion distribution. The vertical line denotes the detection radius of 1 parsec. Overplotted is a quadratic fit with coefficient $2.38 \cdot 10^{-11}$.

Our assumption that the Oort cloud objects should not interact gravitationally with each other allowed for significant improvements in computation time through our custom linear solver. With the N-body solver `Hermite`, a run with 1000 objects took $\approx 3$ seconds. Since `Hermite` is a $\mathcal{O}(N^2)$ solver, we expect that a run with $\approx 10^6$ objects should then take $(10^3)^2$ seconds, i.e. around 10 days. Instead, with our custom solver, a run with $10^6$ Oort cloud objects takes less than an hour to complete. It was therefore essential for us to have implemented a solver that scales linearly with $N$.

Figure 4 shows that the relative velocities of the detected Oort cloud objects with respect to the Sun are around 3 km/s. This is a bit higher than our initial estimate of around 3.3 km/s. We attribute this to the fact that the initial estimate did not include the relative velocities of Betelgeuse and the Sun. This is supported by the fact that the arrival times shown in figure 3 are centered around 55 Myr instead of the predicted 170 Myr, which is consistent with an approximate factor 3 increase in velocity. A side-effect of this earlier arrival time was that the total duration of the simulations was drastically reduced as well. Concretely, we were able to stop our simulations after 75 Myr, instead of having to go beyond 170 Myr.

Our understanding of the need for three different timesteps stemmed from the requirement to accurately sample orbits of the asteroids in three different regimes. Because the orbits were the tightest during the pre-supernova phase, this part required the smallest feasible timestep. Lucklily, we could afford this expense due to the phase also being the shortest by far. Once the asteroids started orbiting in the galactic potential, we were permitted to use a larger timestep. We inuited that a timestep of 1 Myr would be sufficiently small, as the orbits around the Galactic center would have periods on the order of $> 200$ Myr. Nevertheless, we found that smaller timesteps would result in significantly different numbers of detections, which indicated that a smaller timestep was needed. We attribute this mostly to close encounters with Betelgeuse.

Our choice of detection timestep is easiest to validate. Figure 2 shows that the relative positions of the detected asteroids are concentrated in the bottom left. This is consistent with the direction of the velocity of the Solar system relative to the cloud. The fact that there are very few detections on the trailing edge of the detection sphere suggests that the timestep that we used was sufficiently small, as it was impossible for the particles to travel through a significant portion of the sphere within a single timestep.

## 4.2 Future improvements

We do not believe our work has conclusively enabled us to ascertain whether Oort cloud objects orbiting Betelgeuse pose a threat to Earth up to a certain extent. For a more rigorous answer, we suggest a number of improvements to our setup.

First and foremost is the fact that we have used very rudimentary initial conditions for the positions and velocities of the Sun and Betelgeuse. This choice was motivated by the fact that the accuracy of the simulation is limited by the accuracy of the galactic potential code, which was very primitive in our case. Thus, both of these are intricately connected. In future work, a more accurate model of the galactic potential should be used, which would in turn allow for the positions and velocities of the stars to be based on precise measurements[4]. We suggest `galaxia` first and foremost, as it is already included in the `amuse` framework, and should therefore be easy to incorporate into our setup.

Another avenue of improvement would be simulating nearby field stars as well. We believe that close encounters with nearby field stars could have a significant effect on the spreading of the Oort cloud. In particular, we suggest that the cloud should expand faster, which decrease its density, but meanwhile increase its radius, thus theoretically reducing the sensitivity to the initial conditions. This approach would however come at a significant cost to the compute time, as resolving close encounters necessitates smaller timesteps.

### 4.3 Applications

Our setup can with ease be applied to any nearby star or supernova remnant, as long as reasonable initial conditions can be found. Thus, it could be used to investigate other nearby stars to see if their eventual mass-loss could pose a threat in the form of an expanding Oort cloud. Due to the speed of our code, it could even be modified to cycle through a number of nearby stars, and assess the danger of on the order of 168 stars per week (or significantly more if one uses a larger timestep and a smaller number of asteroids).

Alternatively, our test-particle and star codes could also be used to back-track the movement of interlopers such as 'Oumuamua. For instance, a sample of $10^5$ test particles could be launched with positions and velocities according to the uncertainties on the positions and velocities of the interloper. Then, nearby stars could be added. Perhaps this way a list of possible origin points could be constructed.

## 5 Conclusion

In this project, we have developed a purpose-built gravity integrator for assessing the likelihood of a nearby supernova to result in interstellar asteroids reaching the Solar System. We applied this tool to the case of Betelgeuse, and found that we are most likely safe, at least for the duration of this lap around the galaxy. However, we also found the number of interstellar visitor to be highly sensitive to the initial conditions of the simulations. As such, after shifting the position of Betelgeuse by a small amount, we found many more interstellar visitors than initially thought, most likely owing to the relative velocities of Betelgeuse and the Solar system. We found the chance of an Oort cloud object belonging to Betelgeuse coming within 1 AU of the Sun to be 1 in $1.48 \cdot 10^{13}$ in this case. Assuming that the Oort cloud around Betelgeuse (if it exists in the first place) contains a number of asteroids that is proportional to its mass when compared to the Sun, this means we could expect a small handful of exo-Oort cloud objects to come close to the Earth.

[4] For Betelgeuse in particular, however, the large uncertainties on the observations might warrant a sampling of the parameter space.

# References


James Binney and Scott Tremaine. *Galactic Dynamics: Second Edition*. 2008.

Walter Dehnen and James Binney. Mass models of the Milky Way. MNRAS, 294(3):429–438, March 1998. doi: 10.1046/j.1365-8711.1998.01282.x.

Michelle M. Dolan, Grant J. Mathews, Doan Duc Lam, Nguyen Quynh Lan, Gregory J. Herczeg, and David S. P. Dearborn. EVOLUTIONARY TRACKS FOR BETELGEUSE. *The Astrophysical Journal*, 819(1):7, February 2016.

E. F. Guinan, R. J. Wasatonic, and T. J. Calderwood. The Fainting of the Nearby Red Supergiant Betelgeuse. *The Astronomer's Telegram*, 13341:1, December 2019.

Meridith Joyce, Shing-Chi Leung, László Molnár, Michael Ireland, Chiaki Kobayashi, and Ken'ichi Nomoto. Standing on the Shoulders of Giants: New Mass and Distance Estimates for Betelgeuse through Combined Evolutionary, Asteroseismic, and Hydrodynamic Simulations with MESA. ApJ, 902(1):63, October 2020a. doi: 10.3847/1538-4357/abb8db.

Meridith Joyce, Shing-Chi Leung, László Molnár, Michael Ireland, Chiaki Kobayashi, and Ken'ichi Nomoto. Standing on the Shoulders of Giants: New Mass and Distance Estimates for Betelgeuse through Combined Evolutionary, Asteroseismic, and Hydrodynamic Simulations with MESA. ApJ, 902(1):63, October 2020b. doi: 10.3847/1538-4357/abb8db.

Alessandro Morbidelli. Origin and Dynamical Evolution of Comets and their Reservoirs. *arXiv e-prints*, art. astro-ph/0512256, December 2005.

J. H. Oort. The structure of the cloud of comets surrounding the Solar System and a hypothesis concerning its origin. *Bulletin Astronomical Institute of the Netherlands*, 11:91–110, January 1950.

S. Portegies Zwart. Oort cloud Ecology. I. Extra-solar Oort clouds and the origin of asteroidal interlopers. A&A, 647:A136, March 2021. doi: 10.1051/0004-6361/202038888.

S. F. Portegies Zwart and F. Verbunt. Population synthesis of high-mass binaries. A&A, 309:179–196, May 1996.

Simon Portegies Zwart, Santiago Torres, Maxwell X. Cai, and Anthony G. A. Brown. Oort cloud Ecology. II. the chronology of the formation of the Oort cloud. A&A, 652:A144, August 2021. doi: 10.1051/0004-6361/202040096.

S. Toonen, G. Nelemans, and S. Portegies Zwart. Supernova Type Ia progenitors from merging double white dwarfs. Using a new population synthesis model. A&A, 546:A70, October 2012. doi: 10.1051/0004-6361/201218966.

F. van Leeuwen. Validation of the new Hipparcos reduction. A&A, 474(2):653–664, November 2007. doi: 10.1051/0004-6361:20078357.