Tecnociencia, Vol. 27, N°1: 8-25
Enero-Junio 2025
ISSN L 2415-0940

An open Monte Carlo based implementation of Gauss's method for initial orbit determination

Una implementación abierta basada en Monte Carlo del método de Gauss para la determinación inicial de órbitas

José B. Batista-Mendoza
Universidad de Panamá, Facultad de Ciencias Naturales, Exactas y Tecnología, Escuela de Física, Panamá.
joseb.batistam19@gmail.com https://orcid.org/0000-0001-5485-0113

Eduardo Chung
Universidad de Panamá, Facultad de Ciencias Naturales, Exactas y Tecnología, Departamento de Física, Panamá. eduardo.chungng@up.ac.pa https://orcid.org/0000-0003-2834-9450

Adam's Martínez-Soto
Universidad de Panamá, Facultad de Ciencias Naturales, Exactas y Tecnología, Departamento de Física, Panamá. adams.martinez@gmail.com https://orcid.org/0000-0001-8060-686X

Joaquín Fábrega-Polleri
Universidad Católica Santa María la Antigua, Facultad de Ingeniería y Tecnología, Escuela de Ingeniería Civil, Panamá. fabrega@ae.com.pa https://orcid.org/0009-0007-7770-4996

Carlos A. Fernández-Valdés
Universidad Latina de Panamá, Facultad de Ingeniería, Escuela de Electrónica, Sistemas y Mecatrónica, Panamá. carlosfernandez@ulatina.edu.pa https://orcid.org/0000-0003-0983-8777

Fecha de recepción: 25 de abril de 2024
Fecha de aceptación: 9 de septiembre de 2024

DOI HTTPS://DOI.ORG/10.48204/J.TECNO.V27N1.A6633

ABSTRACT

Hundreds or thousands of Near-Earth Asteroids (NEAs) are discovered every year, so being able to determine their orbits to follow them successfully in the future is essential to warn of the danger they could present. Numerous methods have been developed to improve the precision and efficiency of calculations used in the Initial Orbit Determination (IOD), with Gauss's method being the benchmark due to its intuitive formulation, comparable precision, and historical importance. Herein, we present the results of the development of a new open access tool to simplify the process of IOD of celestial bodies, specifically, NEAs. This tool was based on a modern implementation, using code written in Python to calculate, propagate, and graph the orbits. The results obtained from the test data exhibited significant accuracy, with the maximum discrepancy not exceeding 1.2% compared to the Horizons System tool, and the average being 0.5%. Furthermore, we found that for the Monte Carlo simulations that the code uses, 5,000 iterations were more than enough to achieve the obtained accuracy.

KEYWORDS

Asteroids, Astronomy, Celestial Mechanics, Gauss' Method, Orbit Determination.

RESUMEN

Cada año se descubren cientos o miles de Asteroides Cercanos a la Tierra (NEAs, acrónimo en inglés de Near-Earth Asteroids), por lo cual, ser capaz de determinar sus órbitas para seguirlos con éxito en el futuro es indispensable para advertir del peligro que estos podrían presentar. Numerosos métodos se han desarrollado para mejorar la precisión y eficiencia de los cálculos en la determinación inicial de la órbita (IOD, acrónimo en inglés de Initial Orbit Determination), siendo el método de Gauss la referencia debido a su formulación intuitiva, precisión comparable e importancia histórica. Aquí se presentan los resultados del desarrollo de una nueva herramienta de acceso abierto para simplificar el proceso del IOD de cuerpos celestes, específicamente, los NEAs. Dicha herramienta estuvo fundamentada en una implementación moderna, empleando un código escrito en el lenguaje de Python para calcular, propagar y graficar órbitas. Los resultados obtenidos para los datos de prueba exhibieron una precisión significante, con la discrepancia máxima no superando el 1.2 % en comparación con la herramienta Horizons System. Además, se encontró que para las simulaciones de Monte Carlo que el código emplea, 5 000 iteraciones fueron más que suficientes para alcanzar la precisión obtenida.

PALABRAS CLAVES

Asteroides, Astronomía, Mecánica Celeste, Método de Gauss, Determinación de la Órbita.

INTRODUCTION

Every year, hundreds or even thousands of *Near-Earth Asteroids* (NEAs) are discovered, both by professional astronomers and enthusiasts (Minor Planet Center 2023). Therefore, being able to determine their orbits in order to successfully track them in the future is crucial. Determining the orbits of celestial bodies is a problem that has occupied astronomers since ancient times. However, it was not until the beginning of the 19th century that a revolution in celestial mechanics took place, with the work of the German scientist Johann Carl Friedrich Gauss, in which he presented a new orbit determination method from which he obtained an estimate for the orbit of the newly discovered Ceres (Gauss 1809). From his method, numerous algorithms have been developed to improve the accuracy and efficiency of the calculations (Gibbs 1889; Herrick 1971), in addition to the creation of alternative methods (Escobal 1965; Gooding 1997) that are currently considered more favorable for many situations. Nevertheless, Gauss's method is still used as a benchmark for evaluating other methods because of its intuitive formulation, comparable accuracy, and historical importance (Schwab 2022).

One of the main arguments for studies in the area is to improve monitoring systems. Detecting asteroids and calculating their orbits makes it possible to warn of the danger they could present, in other words, to know if they are *Potentially Hazardous Asteroids* (PHAs), classification given to those asteroids suspected of a collision trajectory with our planet in the next centuries (CNEOS Editors 2023) and those responsible for impact events with our planet. Two of these events, and the most significant, are those of Tunguska (1908) and Chelyabinsk (2013), in Russia. In the former, more than 2 000 km^2 of the Siberian taiga were obliterated and caused up to 3 reported deaths; while the latter resulted in damage to more than 7 000 buildings in 6 cities in the region, injuring thousands of people (Jay 2008; David 2013). This highlighted the need for improved NEA detection and monitoring systems. Since then, efforts to identify and track NEAs and PHAs have intensified, as well as the development of strategies such as the *Double Asteroid Redirection Test* (DART) mission by NASA, which successfully altered the orbit of the asteroid Dimorphos (Bardan 2022).

The main focus of this work consists in the presentation and analysis of a new open access tool developed to simplify the *Initial Orbit Determination* (IOD) of celestial bodies, with a specific focus on NEAs. This tool is based on a modern implementation of Gauss's method, using a code written in Python to calculate, propagate and plot orbits. Furthermore, by being available to the public (at the GitHub repository github.com/joebro1907/NEIOD), it aims to encourage the participation of students, enthusiasts, and researchers in the celestial dynamics field, as well as computational physics.

MATERIALS AND METHODS

Gauss's Method

The method developed by Gauss is based entirely on the geometry of only three observations. His method provides an estimate of the three positions of the body from an eighth-order polynomial of the second position. Given the estimate of the three $\vec{r}$, a prediction of instantaneous $\vec{v}_2$ can be defined, thus completely defining the *state vectors*: $(\vec{r}_2, \vec{v}_2)$.

At any point in time, the heliocentric position $\vec{r}$ will be given by (Figure 1):

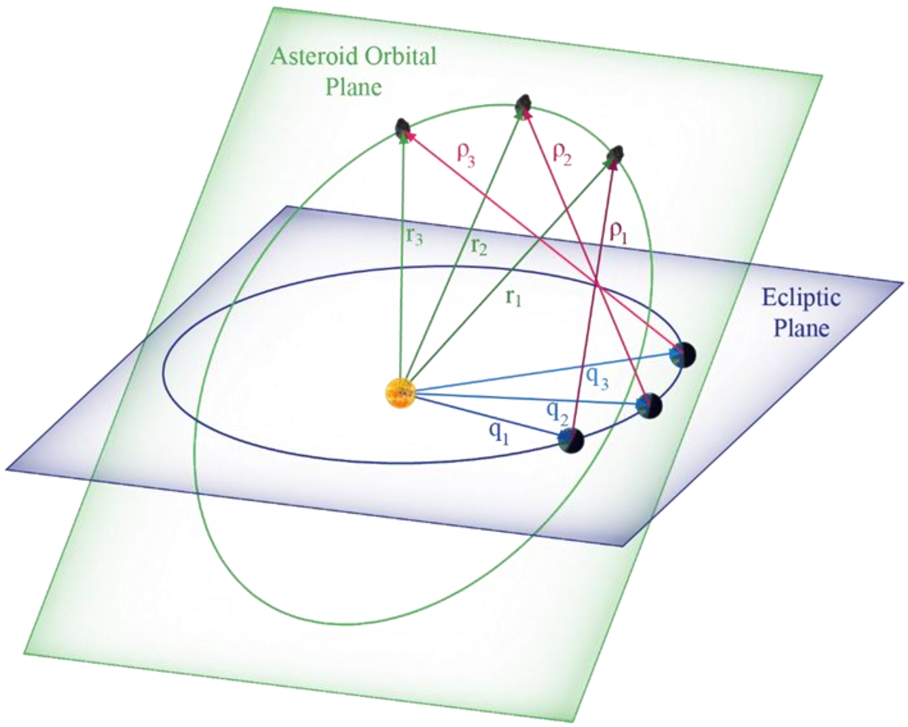
$$\vec{r} = \vec{q} + \rho\hat{\rho} \tag{1}$$

where $\vec{q}$ is the observer's heliocentric position, ρ the body's range (distance) to the observer, and $\hat{\rho}$ its unit vector. The vectors $\vec{q}$ and $\hat{\rho}$ are defined by

$$\vec{q} = \vec{q}_{\oplus} + \vec{p}_{obs} \quad (2)$$
$$\hat{\rho} = \cos(\delta)\cos(\alpha)\,\hat{i} + \cos(\delta)\sin(\alpha)\,\hat{j} + \sin(\delta)\,\hat{k} \quad (3)$$

where $\vec{q}_{\oplus}$ is Earth's heliocentric position, $\vec{p}_{obs}$ is the observer's geocentric position, and α and δ are the *right ascension* and *declination* (Curtis 2014).

Figure 1.
Geometry of the three observations [based on (Gronchi et al. 2021)].



One consequence of the two-body equation of motion is that, at any other time, the state vectors can be expressed in terms of the initial state vectors by means of Lagrange *f* and *g* coefficients. This means that it is possible to express $\vec{r}_1$ y $\vec{r}_3$ in terms of $\vec{r}_2$ y $\vec{v}_2$:

$$\vec{r}_1 = f_1\,\vec{r}_2 + g_1\,\vec{v}_2 \quad (4.\ a)$$
$$\vec{r}_3 = f_3\,\vec{r}_2 + g_3\,\vec{v}_2 \quad (4.\ b)$$

This is due to the fact that the Lagrange coefficients and their time derivatives in these expressions are functions of time and initial conditions themselves, thus allowing us to express how the state vectors change along the orbit (Danby 1988). If the intervals τ_1 and τ_3 between the three observations are small enough, f and g depend only on $\vec{r}_2$:

$$f_1 \approx 1 - \frac{\mu}{2r_2^3}\tau_1^2 \qquad f_3 \approx 1 - \frac{\mu}{2r_2^3}\tau_3^2 \tag{5. a}$$

$$g_1 \approx \tau_1 - \frac{\mu}{6r_2^3}\tau_1^3 \qquad g_3 \approx \tau_3 - \frac{\mu}{6r_2^3}\tau_3^3 \tag{5. b}$$

where $\tau_1 = t_2 - t_1$, $\tau_3 = t_3 - t_2$, and $\mu = 1.327\times10^{20}\ m^3\cdot s^{-2}$ (Sun's standard gravitational parameter, product of its mass and the universal gravitational constant).

The slant ranges ρ_1, ρ_2, and ρ_3 are given by

$$\rho_1 = \frac{1}{D_0}\left[\frac{6r_2^3\left(\frac{\tau_1}{\tau_3}D_{31} + \frac{\tau}{\tau_3}D_{21}\right) + \mu(\tau^2 - \tau_1^2)\frac{\tau_1}{\tau_3}D_{31}}{6r_2^3 + \mu(\tau^2 - \tau_3^2)} - D_{11}\right] \tag{6. a}$$

$$\rho_2 = A + \frac{\mu B}{r_2^3} \tag{6. b}$$

$$\rho_3 = \frac{1}{D_0}\left[\frac{6r_2^3\left(\frac{\tau_3}{\tau_1}D_{13} + \frac{\tau}{\tau_1}D_{23}\right) + \mu(\tau^2 - \tau_3^2)\frac{\tau_3}{\tau_1}D_{13}}{6r_2^3 + \mu(\tau^2 - \tau_1^2)} - D_{33}\right] \tag{6. c}$$

where A, B, D_0, and D_{ij} are

$$A = \frac{1}{D_0}\left(-\frac{\tau_3}{\tau}D_{12} + D_{22} - \frac{\tau_1}{\tau}D_{32}\right) \tag{7. a}$$

$$B = \frac{1}{6D_0}\left[(\tau_3^2 - \tau^2)\frac{\tau_3}{\tau}D_{12} + (\tau^2 - \tau_1^2)\frac{\tau_1}{\tau}D_{32}\right] \tag{7. b}$$

$$D_0 = \hat{\rho}_1 \cdot \left(\hat{\rho}_2 \times \hat{\rho}_3\right) \tag{7. c}$$

$$D_{ij} = \vec{q}_i \cdot \left(\hat{\rho}_k \times \hat{\rho}_l\right) \tag{7. d}$$

To calculate $\vec{r}_2$, one can use the square of equation 1 with the new expression for ρ_2:

$$r_2^2 = \left(A + \frac{\mu B}{r_2^3}\right)^2 + 2E\left(A + \frac{\mu B}{r_2^3}\right) + q_2^2$$

where $E = \vec{q}_2 \cdot \hat{\rho}_2$. Expanding and rearranging terms leads to the eighth-order polynomial $r_2^8 + k_2 r_2^6 + k_1 r_2^3 + k_0 = 0$, with its the coefficients being

$$k_2 = -\left(A^2 + 2AE + q_2{}^2\right), \qquad k_1 = -2\mu B(A+E) \qquad y \qquad k_0 = -\mu^2 B^2$$

And the speed $\vec{v}_2$ is given by

$$\vec{v}_2 = \left(\frac{f_1\vec{r}_3 - f_3\vec{r}_1}{g_3 f_1 - f_3 g_1}\right) \tag{8}$$

It is important to keep in mind that there is a limit to the angular (thus temporal) separation between observations. Separations that are too small lead to numerical instability, especially when measurement noise can be dominant; conversely, too large of a separation renders approximations useless (Tennenbaum & Director 1997). This results in an upper limit around 60°, as indicated by (Escobal 1965) and (Long et al. 1989), and a lower limit of approximately 1°, according to (Vallado 2013), who states that the method works especially well when the separation is about 10°.

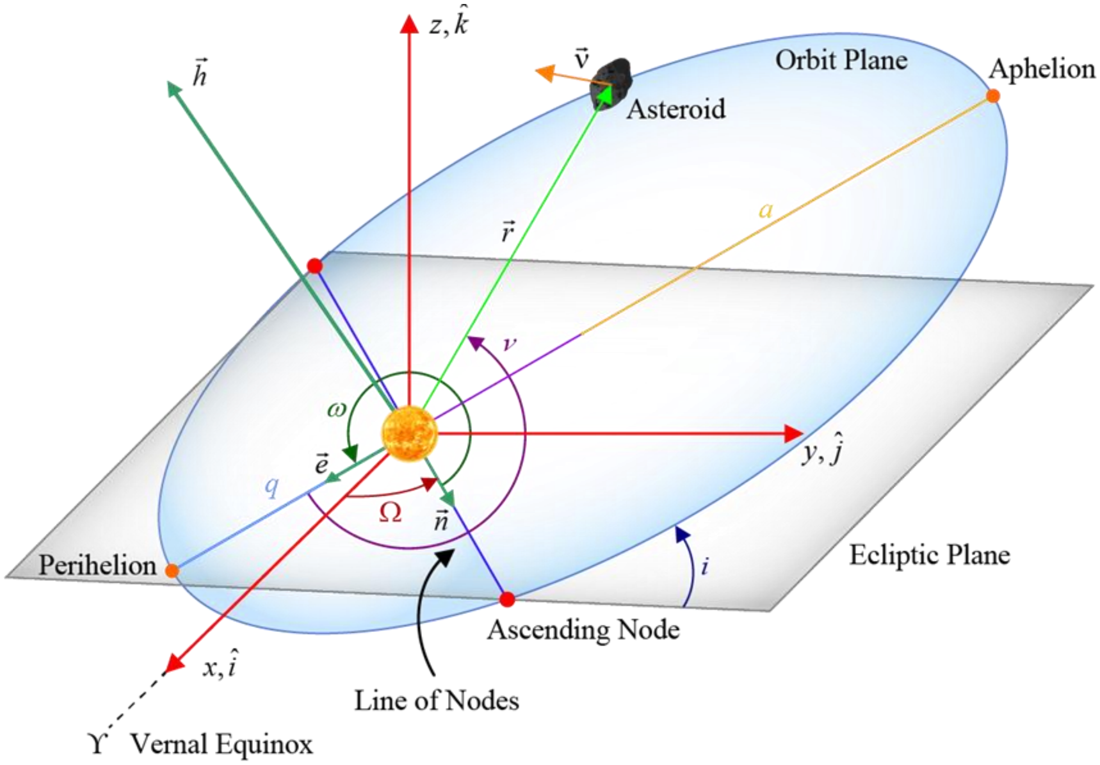
Classical Orbital Elements

Once the state vectors are calculated, the orbital elements can be obtained using equations which can be found in (Curtis 2014). The elements give the shape and orientation of the orbit in space (Figure 2). These elements are:

- Eccentricity, *e*: elongation of the orbit.
- Semi-major axis, *a*: half the length of the line joining the points of perihelion and aphelion (least and greatest distance from the Sun, respectively).
- Inclination, *i*: angle between the body's orbital plane and the ecliptic (Earth's orbital plane).
- Longitude of the Ascending Node, *Ω*: angle between the direction of the vernal equinox (point on the ecliptic at which the Sun passes from the southern celestial hemisphere to the northern) and the ascending node (intersection point of the orbital plane and the ecliptic, in the upward direction).
- Argument of Perihelion, *ω*: angle between the ascending node and the perihelion.
- True Anomaly, *v*: angle between the perihelion and the body's current position.

In the case of parabolic and hyperbolic orbits, *a* is not defined (since these are open conic sections), so the Perihelion Distance, *q*, is used. However, both parabolic and hyperbolic NEAs are rare.

Figure 2.
The classical orbital elements and state vectors [based on (Barbee 2005)].



Monte Carlo Method

It is not reasonable to propagate errors in a traditional way when using extensive and iterative algorithms such as this one. Therefore, the Monte Carlo method is usually chosen to calculate uncertainties since it has been consistently considered a reliable approach and has had wide application in the validation of uncertainty propagators, whether linear or nonlinear in nature. (Ding et al. 2014; Luo & Yang 2017). In the *sampling* use case, the Monte Carlo method simply involves random sampling from a certain probability distribution (Kroese et al. 2014). The idea is to repeat the algorithm or procedure numerous times in order to obtain the quantities of interest using the Law of Large Numbers, a mathematical theorem that states that the average of the results obtained from a large enough number of independent random samples converges to the true value, if it exists (Yao & Gao 2015). This way, we can get the mean value and standard deviation of the simulations, and therefore the uncertainties.

Python Code

The code was written with the popular code development software *Sypder IDE* and is based on Python version 3.11. This code was made on the basis of Gauss's method algorithm developed by (Curtis 2014), which has been translated into the Python language and modified to accommodate our implementation. In addition, we made use of popular astronomy libraries (Table 1).

Table 1.

Python libraries used in the code, essential for various functions and routines.

Library	Description	Creators
Astropy	Core Python package for astronomy, providing tools for celestial calculations.	(Price-Whelan et al. 2018)
Astroquery	Simplifies querying of astronomical databases and web services	(Ginsburg et al. 2019)
Matplotlib	Popular library for creating 2D graphs and charts	(Hunter 2007)
Numpy	Fundamental library for scientific computing with arrays and mathematical functions	(Harris et al. 2020)
Pandas	Data manipulation and analysis library with DataFrames and data arrays	(McKinney 2010)
Poliastro	Library specializing in astrodynamics and orbital mechanics	(Rodríguez et al. 2023)
SciPy	Extends Numpy with advanced scientific computing capabilities	(Virtanen et al. 2020)

The code was written to make its operation as intuitive as possible. Once executed, it runs on a terminal window. Firstly, the program asks for observational data: depending on the user, this can be collected either automatically by reading a text file formatted in the *Astrometry Data Exchange Standard* (ADES) format, or by entering it manually. This data includes: the α and δ values with their respective RMS errors for each of the three observations, the corresponding dates, the NEA identification number, if known, given by the *Minor Planet Center* (MPC, official body for observing and reporting on minor planets under the auspices of the International Astronomical Union) and the observatory code, also assigned by the Minor Planet Center, if it has one (otherwise the geographic coordinates must be entered). After entering all this data, the user will have specified the number N of *Monte Carlo* (MC) *simulations (a value between 5,000 and 10,000 is recommended)* to be done. Before any calculation, we made sure that the code applied *Light-Time Correction* (LTC) to the dates, due to the fact that in reality, the observed positions would correspond to when the light "left" the NEA, so there is a delay. It is calculated with ρ and the speed of light c (Gronchi 2013):

$$\delta t = \frac{\rho}{c} \tag{9}$$

Once this correction is applied, Monte Carlo simulations are done by randomly generating *N* (α, δ) coordinates for each observation, drawn from the normal distributions defined by their uncertainties. The simulations then proceed by using these coordinates in the Gauss's method algorithm for calculating the corresponding state vectors, and iteratively refining them. The refined state vectors are then used to determine the classical orbital elements as well as the mean anomaly *M*, orbital period *T*, mean motion *n*, perihelion distance *q*, aphelion distance *Q*, and perihelion epoch *tp*. Both the state vectors and the orbital elements are transformed to the ecliptic reference plane with the equations by (Vallado 2013; Sharaf et al. 2014). After this, using the *SciPy* library, the average from each state vector in the MC simulations is calculated with its respective standard deviation, and the same is done for the orbital elements. Once the orbit is calculated, if the NEA is known, the code validates the results by comparing the orbital elements with the ones given by *Horizons System*, an online service by the Solar System Dynamics Group of NASA's JPL for calculating ephemeris (positions) and other high-precision data for bodies in our solar system (Giorgini & SSD Group 2022); otherwise, it queries the JPL Small-body Database (SBDB) for possible matches. In either case, a text file is then generated with the results, including observational data, observation arc (time between the first and last observation), state vectors, and orbital elements with their comparison. In addition, the angular separation θ (angular distance between the first and last observation) is included:

$$\theta = \cos^{-1}[\sin(\delta_1)\sin(\delta_3) + \cos(\delta_1)\cos(\delta_3)\cos(\alpha_1 - \alpha_3)] \quad (10)$$

As a next step, from the orbital elements, the code creates graphics as a visual representation of the NEA orbit, from different views. Finally, the user is given the option to generate ephemerides for the asteroid. To do this, the method of propagating $\vec{r}_2$ with the *f* and *g* coefficients is used (Lee et al. 2019). Throughout the entire run, the code lets the user know what it is doing. It is important to note that the code requires an internet connection at all times to access most tools and services. The recommended value for *N* was determined with modified versions of the code to calculate the computation time and the consistency of the results for values of *N* that would be frequently used in MC simulations: 1,000, 5,000, 10,000, 50,000 and 100,000, with 10 iterations each to determine their average. We used observational data (Table 2) obtained from the March 15, 2024 *Minor Planet Circulars Supplement* (Minor Planet Center 2024) with uncertainties of $4.0\times10^{-6\circ}$ and $4.0\times10^{-5\circ}$ for α and δ, respectively, since these were not provided, and we wanted to adhere to MPC's astrometry reporting standards (Project Pluto 2023).

Table 2.
Observational data with the corresponding observation arc and angular separation (between each first and last observation).

NEA	Epoch (LTC UTC)	α [°]	δ [°]	θ [°]	Obs. Arc [d]
1995 FO	2024-02-27 01:13:56.874	141.607171	-24.26946	25.765	16.155
	2024-03-06 21:57:08.999	148.803975	-9.74072		
	2024-03-14 04:57:05.948	153.047900	-1.00581		
2024 ED4	2024-03-12 08:41:43.720	170.134933	-7.38124	2.434	1.822
	2024-03-13 22:18:11.238	172.010721	-8.33164		
	2024-03-14 04:25:06.806	172.334958	-8.46681		
85095 Hekla	2021-08-04 18:59:36.321	1.27685	-7.40092	12.471	40.966
	2021-08-29 18:23:33.218	355.39752	-5.24053		
	2021-09-14 18:10:57.900	349.22838	-3.96226		
200974	2024-03-01 11:56:51.783	158.471079	9.50644	2.986	12.703
	2024-03-06 08:25:22.276	157.295863	9.87134		
	2024-03-14 04:48:37.681	155.573779	10.38525		
526142	2024-03-02 08:53:24.303	178.233829	8.83903	2.785	11.970
	2024-03-06 09:05:28.550	177.390371	9.21129		
	2024-03-14 08:10:49.216	175.630033	9.91533		

RESULTS AND DISCUSSION

Computation Time

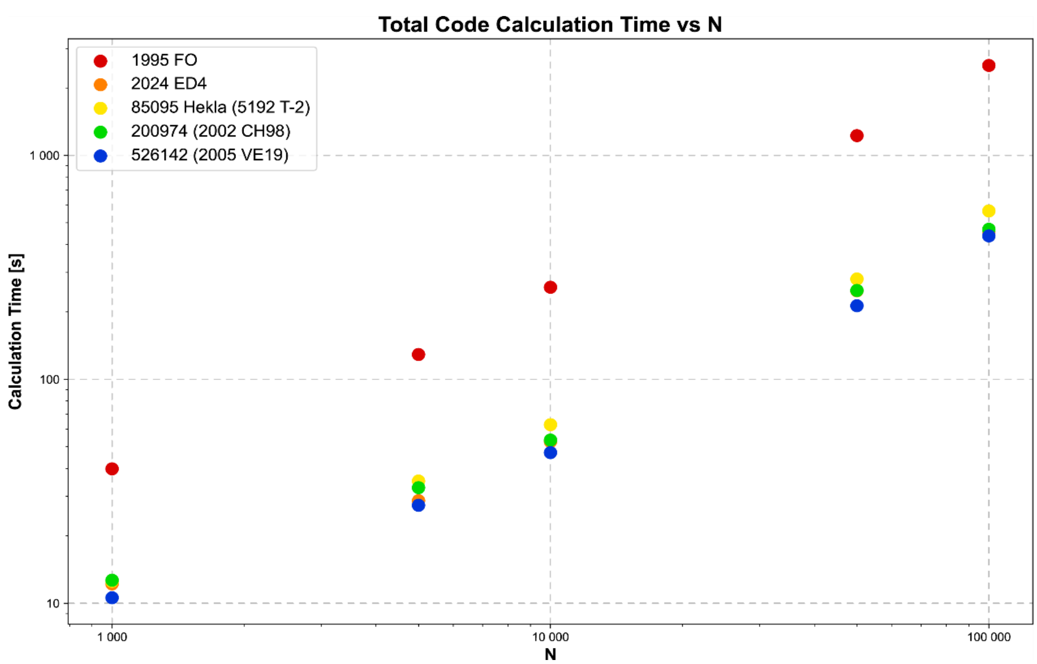
As expected, the code took longer when choosing larger *N* values (Table 3). In addition, it is seen that, with the exception of 1995 FO, the computation times were similar for each NEA at the respective *N* values (Figure 3).

Table 3.
Calculation time [s] per N value for each NEA.

N	1995 FO	2024 ED4	85095 Hekla	200974	526142
1 000	30.353	10.212	12.079	10.899	10.459
5 000	129.17	28.729	35.147	32.823	27.397
10 000	257.84	52.713	62.732	53.539	47.078
50 000	1225.514	248.483	280.655	249.538	213.167
100 000	2521.983	457.661	564.805	468.083	437.314

Performing an analysis of the code we found that this discrepancy between the computation times for 1995 FO is due to the iterative improvement routine in Gauss's method. It turns out that 1995 FO required 5 to 7 times more iterations to achieve state vectors convergence: 270 versus 34, 54, 43 and 37, respectively. This is because the (fixed) tolerance is set to 1.0×10^{-10} to ensure accuracy, therefore, lowering it would reduce the required iterations and computational time, in spite of accuracy, of course.

Figure 3.
Calculation time for each N. The discrepancy in 1995 FO times (in red) can be seen.



Comparing the times shows that 10,000 iterations of the MC simulation took about one minute to complete, so we consider them to be convenient values. Clearly, if time were the only priority, 1,000 would be the best choice. Additionally, computation times depend on the processing power of the user's computer, logically; therefore, the code was run on a personal desktop computer of modest specifications (Inter Core i7-11700F CPU and 16 GB of RAM), to simulate an average modern user.

Dispersion and Discrepancy of Calculation

Comparing the computational discrepancies for each *N* (Table 4) we appreciate that the discrepancies of the results with the *Horizons System* reference values did not vary significantly, so increasing *N* does not necessarily lead to more accurate results; in fact, it only increases the computational demand and, therefore, the computation time. This is what is known as *diminishing returns*, when after a certain threshold, increasing *N* will not result in noticeable improvements in accuracy (Figure 4).

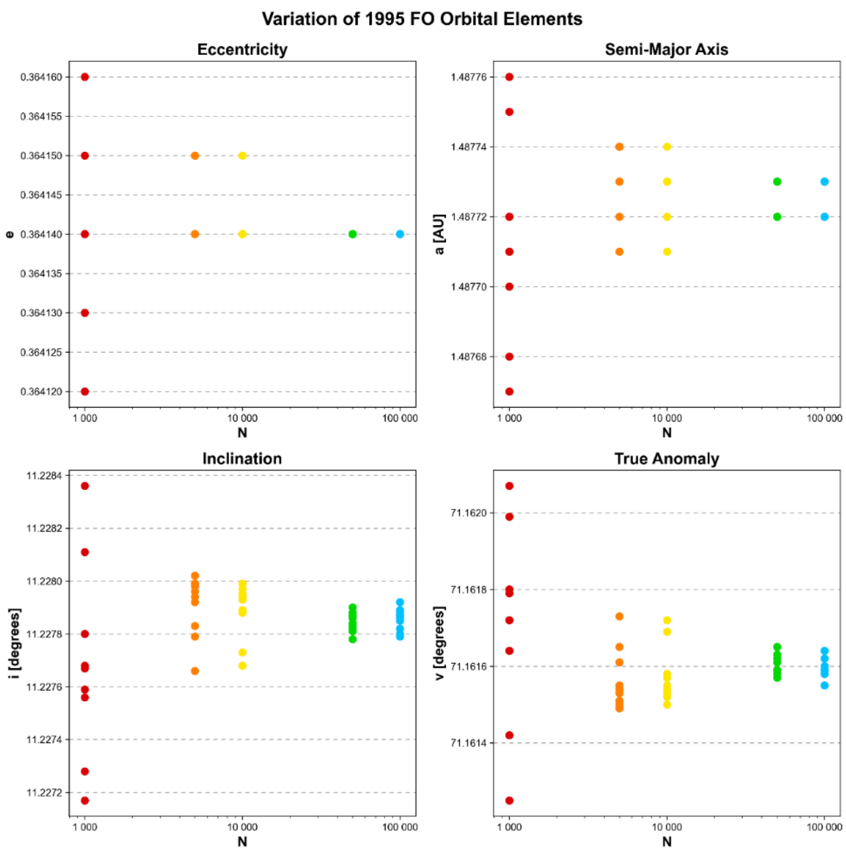
Table 4.

Discrepancy of classical orbital elements with those of the Horizons System for each NEA varying N. The values correspond to the average of 10 iterations.

NEA	N	Δe	Δa [AU]	Δi [°]	ΔΩ [°]	Δω [°]	Δν [°]
1995 FO	1 000	0.00032	0.00079	0.01035	0.00478	0.01116	0.00810
	5 000	0.00031	0.00077	0.01014	0.00469	0.01095	0.00796
	10 000	0.00032	0.00077	0.01016	0.00470	0.01097	0.00797
	50 000	0.00032	0.00077	0.01020	0.00472	0.01102	0.00800
	100 000	0.00032	0.00077	0.01019	0.00471	0.01101	0.00800
2024 ED4	1 000	0.00074	0.00320	0.00156	0.07789	0.15942	0.08199
	5 000	0.00075	0.00323	0.00153	0.07809	0.15970	0.08207
	10 000	0.00075	0.00323	0.00153	0.07813	0.15978	0.08211
	50 000	0.00075	0.00323	0.00154	0.07810	0.15974	0.08210
	100 000	0.00075	0.00323	0.00153	0.07814	0.15979	0.08211
85095 Hekla	1 000	0.00010	0.00011	0.00450	0.00005	0.00395	0.00456
	5 000	0.00010	0.00011	0.00457	0.00005	0.00420	0.00481
	10 000	0.00010	0.00011	0.00458	0.00005	0.00424	0.00486
	50 000	0.00010	0.00011	0.00457	0.00005	0.00419	0.00481
	100 000	0.00010	0.00011	0.00457	0.00005	0.00419	0.00481
200974	1 000	0.00176	0.00223	0.00875	0.02862	0.79071	0.81049
	5 000	0.00165	0.00210	0.00824	0.02700	0.74989	0.76853

	10 000	0.00159	0.00204	0.00799	0.02624	0.73326	0.75137
	50 000	0.00163	0.00208	0.00817	0.02678	0.74539	0.76388
	100 000	0.00165	0.00210	0.00824	0.02699	0.74982	0.76846
526142	1 000	0.00131	0.00021	0.00510	0.02326	0.65304	0.68059
	5 000	0.00130	0.00018	0.00522	0.02478	0.67272	0.70186
	10 000	0.00130	0.00019	0.00521	0.02468	0.67089	0.69993
	50 000	0.00130	0.00019	0.00519	0.02440	0.66751	0.69626
	100 000	0.00130	0.00019	0.00519	0.02447	0.66875	0.69756

Figure 4.
Non-linear variation reduction in some 1995 FO elements for each N.



Therefore, the most important thing in this regard is to choose a value of *N* that results in the least dispersion, i.e., the greatest consistency in the results. We determined then that if the only priority were consistency, 100,000 would be the best choice.

Determination of N

A value for *N* should be chosen with these two considerations (low variability and short calculation time). To determine such value, we used a weighted sum model the following way: a weight of 7.5 was given to accuracy, given as the 1-to-5 scoring of the average discrepancy across the elements for each *N* (smaller discrepancy, higher score); and a weight of 2.5 was given to the computing time, given as the 1-to-5 scoring of the average time for each *N* (smaller time, higher score). This means a maximum score of (7.5)(5) + (2.5)(5) = 50 points. All of this for each NEA, with their score sum determining the final ranking (Table 5).

Table 5.

Weighted scoring of N. The points are the sum of the score for each NEA.

N	Points per NEA (time and accuracy)					Total	Rank
	1995 FO	2024 ED4	85095 Hekla	200974	526142		
1 000	20	50	30	20	35	155	**2**
5 000	48	40	25	35	25	173	**1**
10 000	20	28	35	35	35	153	**3**
50 000	20	30	30	35	28	143	**4**
100 000	25	10	25	25	25	110	**5**

We thus determine that the value of *N* that presents a good balance between the variation of results and the calculation time is 5,000.

CONCLUSIONS

Our code developed in Python allows us to successfully compute the initial orbit of NEAs from three observations of right ascension α and declination δ, using Monte Carlo simulations to obtain the osculating ecliptic orbital elements. The results are sufficiently accurate compared to the reference values by the Horizons System, so the discrepancies are more than acceptable for this type of study. Finally, the developed code has the flexibility and scalability in mind to adapt to different data sets (and therefore, types of orbits) in addition to possible expansions, one of the main reasons for using the Python language since the scope of this work is the creation of an open access tool for the scientific community that facilitates and simplifies the process of studying these bodies.

ACKNOWLEDGEMENTS

This work is part of the research project *"Estudio de la Forma y Cálculo de las Velocidades Axiales de Asteroides Cercanos a la Tierra" (Study of the Shape and Calculation of Axial Velocity of Near-Earth Asteroids)*, with identification CUFI-2021-P-CNET-003, which was funded by the 2021 *Convocatoria Universitaria a Fondos de Investigación (University Call for Research Funds)* of the Vice Rectorship for Research and Postgraduate Studies of the University of Panama.

This includes data from the Horizons System tool of the Solar System Dynamics Group, belonging to NASA's Jet Propulsion Laboratory.

This work includes data from the MPC's database, which is made freely available to the public. Funding for this data and the MPC's operations comes from a NASA PDCO grant (80NSSC22M0024), administered via a University of Maryland - SAO subaward (106075-Z6415201). The MPC's computing equipment is funded in part by the above award, and in part by funding from the Tamkin Foundation.

BIBLIOGRAPHIC REFERENCES

Barbee, B. W. (2005). Mission Planning for the Mitigation of Hazardous Near-Earth Objects [Master's Thesis, The University of Texas at Austin]. https://www.researchgate.net/publication/36174303_Mission_planning_for_the_mitigation_of_hazardous_Near_Earth_Objects_electronic_resource

Bardan, R. (2022). NASA Confirms DART Mission Impact Changed Asteroid's Motion in Space. NASA. http://www.nasa.gov/press-release/nasa-confirms-dart-mission-impact-changed-asteroid-s-motion-in-space

CNEOS Editors. (2023). *NEO Basics*. Center for Near-Earth Objects Studies. https://www.nasa.gov/directorates/smd/planetary-science-division/planetary-defense-coordination-office/nasa-releases-agency-strategy-for-planetary-defense-to-safeguard-earth/

Curtis, H. D. (2014). Orbital Mechanics for Engineering Students (3rd ed.). Elsevier, BH, Butterworth-Heinemann is an imprint of Elsevier.

Danby, J. M. A. (1988). Fundamentals of Celestial Mechanics (2nd ed.). Willmann-Bell.

David, L. (2013). Russian Fireball Explosion Shows Meteor Risk Greater Than Thought. Space.com. https://www.space.com/23423-russian-fireball-meteor-airburst-risk.html

Ding, C. et al. (2014). Orbit Determination for Asteroid 214088 (2004 JN13) Using Gauss' Method. Summer Science Program 2014. https://bpb-us-w2.wpmucdn.com/campuspress.yale.edu/dist/e/472/files/2015/06/DingChunyang_SSP_Paper-2jvdzig.pdf

Escobal, P. R. (1965). Methods of Orbit Determination (1st ed.). J. Wiley.

Gauss, C. F. (1809). Theoria Motus Corporum Coelestium In Sectionibus Conicis Solem Ambientium. sumtibus Frid. Perthes et I. H. Besser. https://preserver.beic.it/delivery/DeliveryManagerServlet?dps_pid=IE1661019

Gibbs, J. W. (1889). On the Determination of Elliptic Orbits from Three Complete Observations. National Academy of Sciences.

Ginsburg, A. et al. (2019). Astroquery: An Astronomical Web-Querying Package in Python. The Astronomical Journal, 157(3), 98. https://doi.org/10.3847/1538-3881/aafc33

Giorgini, J., & SSD Group, J. (2022). Horizons System. Solar System Dynamics. https://ssd.jpl.nasa.gov/horizons/app.html#/

Gooding, R. H. (1997). A New Procedure for Orbit Determination Based on Three Lines of Sight. CELESTIAL MECHANICS AND DYNAMICAL ASTRONOMY, 66(4), 387–423. https://doi.org/10.1007/BF00049379

Gronchi, G. F. (2013). Classical Methods of Orbit Determination. https://www.stardust2013.eu/Portals/63/Images/Training/OTS%20Repository/gronchi_OTS2013.pdf

Gronchi, G. F. et al. (2021). Generalization of a Method by Mossotti for Initial Orbit Determination. Celestial Mechanics and Dynamical Astronomy, 133(9), 41. https://doi.org/10.1007/s10569-021-10038-4

Harris, C. R. et al. (2020). Array Programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Herrick, S. (1971). Astrodynamics: Orbit Determination, Space Navigation, Celestial Mechanics (1st ed.). Van Nostrand Reinhold Company.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in science & engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jay, P. (2008). The Tunguska Event. CBC News. https://www.cbc.ca/news/science/the-tunguska-event-1.742329

Kroese, D. P. et al. (2014). Why the Monte Carlo method is so important today. Wiley Interdisciplinary Reviews: Computational Statistics, 6(6), 386–392. doi:10.1002/wics.1314

Lee, H. et al. (2019). Orbit Determination of Asteroid 2002 KM6. OSF Preprints. https://doi.org/10.31219/osf.io/36hkw

Long, A. C. et al. (1989). Goddard Trajectory Determination System (GTDS): Mathematical Theory. Goddard Space Flight Center. https://books.google.com.pa/books?id=X_WdtgAACAAJ

Luo, Y., & Yang, Z. (2017). A Review of Uncertainty Propagation in Orbital Mechanics. Progress in Aerospace Sciences, 89, 23–39. https://doi.org/10.1016/j.paerosci.2016.12.002

McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 445, 51–56. https://doi.org/10.25080/Majora-92bf1922-00a

Minor Planet Center. (2023). Orbits for Near Earth Asteroids (NEAs). Minor Planet Center website. https://www.minorplanetcenter.net/iau/MPCORB/NEA.txt

Minor Planet Center. (2024). *M.P.S. 2024 MAR. 15*. The Minor Planet Circulars/Minor Planets and Comets Supplement, 2145921–2152456. https://www.minorplanetcenter.net/iau/ECS/MPCArchive/2024/MPS_20240315.pdf

Price-Whelan, A. M. et al. (2018). The Astropy Project: Building an Open-Science Project and Status of the v2.0 Core Package. The Astronomical Journal, 156(3), 123. https://doi.org/10.3847/1538-3881/aabc4f

Project Pluto. (2023). Assorted thoughts on astrometric error handling. Project Pluto website. https://www.projectpluto.com/errors.htm

Rodríguez, J. L. C. et al. (2023). Poliastro 0.17.0 (SciPy US ’22 edition) (0.17.0). https://zenodo.org/record/6817189

Schwab, D. (2022). Efficacy of Gaussian Process Regression for Angles-Only Initial Orbit Determination [Master’s Thesis, The Pennsylvania State University]. https://etda.libraries.psu.edu/files/final_submissions/22263

Sharaf, M. A. et al. (2014). Two Unified Algorithms for Fundamental Planetary Ephemeris. International Journal of Astronomy and Astrophysics, 04(04), 598–606. https://doi.org/10.4236/ijaa.2014.44054

Tennenbaum, J., & Director, B. (1997). How Gauss Determined The Orbit of Ceres. https://archive.schillerinstitute.com/fid_97-01/982_orbit_ceres.pdf

Vallado, D. A. (2013). Fundamentals of Astrodynamics and applications (4th ed.). Microcosm Press.

Virtanen, P. et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Yao, K. & Gao, J. (2015). Law of Large Numbers for Uncertain Random Variables. Fuzzy Systems, 24(3), 615–621 https://doi.org/10.1109/TFUZZ.2015.2466080