Računalna simulacija vjerojatnosti udara asteroida u Zemlju

Stupalo, Petra

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:055113

Sveučilište u Splitu
Prirodoslovno – matematički fakultet

Računalna simulacija vjerojatnosti udara asteroida u Zemlju

Završni rad

Petra Stupalo

Split, rujan 2022.

Temeljna dokumentacijska kartica

Sveučilište u Splitu
Prirodoslovno – matematički fakultet
Odjel za fiziku
Ruđera Boškovića 33, 21000 Split, Hrvatska

Završni rad

Računalna simulacija vjerojatnosti udara asteroida u Zemlju

Petra Stupalo

Sveučilišni preddiplomski studij Fizika

Sažetak:
U ovom radu kreiramo računalnu simulaciju koja promatra gibanje planeta i asteroida oko Sunca. Za problem koji ne možemo analitički riješiti stvaramo matematičku/fizikalnu reprezentaciju kako bismo predvidili ponašanja nekog sustava. Problem kojim se rad bavi je računanje vjerojatnosti udara asteroida o Zemlju. Promatramo asteroide nasumično odabranih masa, brzina i početnih položaja i pratimo njihovo gibanje kroz Sunčev sustav. Ovaj kompleksan problem aproksimirali smo jednostavnim fizikalnim modelom. Iz simulacije računamo da oko 20% generiranih asteroida prođe putanju radijusa 1.2 AU, i time predstavljaju potencijalnu opasnost Zemlji.

Ključne riječi: Sunčev sustav, planeti, asteroidi, računalna simulacija, gravitacijska sila, Eulerova metoda

Rad sadrži: 22 stranice, 6 slika, 0 tablica, 11 literaturni navod. Izvornik je na hrvatskom jeziku.

Mentor: doc. dr. sc. Toni Šćulac

Ocjenjivači: doc. dr. sc. Toni Šćulac,
prof. dr. sc. Ante Bilušić,
dr. sc. Marin Vojković

Rad prihvaćen: 21. rujna. 2022.

Rad je pohranjen u Knjižnici Prirodoslovno – matematičkog fakulteta, Sveučilišta u Splitu.

Basic documentation card

University of Split
Faculty of Science
Department of Physics
Ruđera Boškovića 33, 21000 Split, Croatia

Bachelor thesis

A computer simulation of the probability for asteroid impact on Earth

Petra Stupalo

University undergraduate study programme Physics

Abstract:
In this paper, we create a computer simulation that observes the motion of planets and asteroids around the Sun. For a problem that we cannot solve analytically, we create a mathematical/physical representation in order to predict the behavior of a system. The problem that the paper deals with is the calculation of the probability of an asteroid hitting the Earth. We observe asteroids of randomly selected masses, velocities and initial positions and track their motion through the solar system. We approximated this complex problem with a simple physical model. From the simulation, we calculate that about 20% of the generated asteroids pass a path of radius 1.2 AU, and thus represent a potential danger to the Earth.

Keywords: Solar system, planets, asteroids, computer simulation, gravitational force, Euler's method

Thesis consists of: 22 pages, 6 figures, 0 tables, 11 references. Original language: Croatian.

Supervisor: Assist. Prof. Dr. Toni Šćulac

Reviewers: Assist. Prof. Dr. Toni Šćulac,
Prof. Dr. Ante Bilušić,
Dr. Marin Vojković

Thesis accepted: September 21, 2022.

Thesis is deposited in the library of the Faculty of Science, University of Split.

Sadržaj

1 Uvod

1.1 Asteroidi u Sunčevom sustavu

Sunčev sustav sastoji se od Sunca i manjih nebeskih tijela koje okuplja zajednička gravitacijska sila. Sustav sadrži 8 planeta, 5 patuljastih planeta, 185 njihovih prirodnih satelita, te mnoštvo sitnih objekata: kometa, planetoida (asteroida), tijela Kuiperova pojasa, meteoroida i međuplanetarne prašine [1]. Nas će u radu zanimati asteroidi. Asteroidi su mala čvrsta nebeska tijela koja kruže oko Sunca, ali ne zadovoljavaju određene kriterije koji definiraju planete. Astronomski objekt mora ispunjavati tri uvjeta koje je odredila Međunarodna astronomska unija (IAU) da bi se mogao nazvati planetom [2]. Planet treba izravno kružiti oko matične zvijezde, u našem slučaju oko Sunca. Mora biti dovoljno velik kako bi svojom gravitacijom mogao zadržavati sferni oblik. Konačno, objekt da bi bio planet mora očistiti svoje orbitalno susjedstvo od ostalih objekata slične veličine [3]. Iako kruže oko Sunca, asteroidi ne zadovoljavaju druga dva uvjeta pa se ne mogu nazivati planetima. Asteroidi nemaju atmosferu te se sastoje od metala i/ili stijena. Većina asteroida u Sunčevom sustavu se nalazi u glavnom asteroidnom pojasu između Marsa i Jupitera na udaljenosti 2 - 4 AU[1]od Sunca [5]. Dio asteroida prilazi Suncu bliže, prolazeći orbitu Marsa, ali ne i orbitu Zemlje (Amori), dio siječe Zemljinu orbitu s periodom većim od jedne godine (Apoloni), te dio s periodom manjim od jedne godine (Ateni) [5]. U ovom se radu fokusiramo na ovakve asteroide koje nazivamo Zemlji bliski asteroidi, jer predstavljaju potencijalnu opasnost Zemljinoj civilizaciji. U Zemljinu atmosferu svaki dan upada više od sto tona čestica prašine i pijeska, a otprilike jednom godišnje uđe asteroid veličine automobila koji izgori prije nego što stigne do Zemljine površine [6]. Asteroidi veličine nekoliko desetaka metara, sudare se sa Zemljom u prosjeku svakih 2000 godina te naprave značajnu štetu na području sudara [6]. Iako su asteroidi brojni u Sunčevom sustavu, samo jednom u nekoliko milijuna godina se pojavi tijelo dovoljno veliko da bi sudarom sa Zemljom bio prijetnja civilizaciji [6].

1.2 Računalna simulacija

Računalna simulacija je upotreba računala za imitaciju stvarnog sustava, tj. sustava promatranog u prirodi. Bazira se na kreiranju matematičkog ili/i fizikalnog modela problema, testiranju te, kada je moguće, usporedbi dobivenih rješenja sa stvarnim podacima. Računalnim simulacijama znanstvenici žele jednostavnije analizirati ponašanja određenih sustava. U ovom radu istim načinom želimo analizirati sudar asteroida s Zemljom.

U klasičnoj mehanici analitički možemo riješiti gibanje dvaju tijela, ali kada promatramo tri ili više tijela, problem više nije analitički rješiv i tada numerički pristupamo rješavanju problema.

1.3 Opis problema

Simuliramo jednostavni fizikalni sustav gibanja tijela u bližoj okolini Zemlje. Počinjemo simuliranjem pet planeta oko Sunca, a zatim dodajemo asteroide različitih brzina, masa i početnih položaja. Koristeći se Newtonovim zakonima te numeričkim rješavanjem jednadžbi želimo odrediti vjerojatnost sudara asteroida sa Zemljom te usporediti sa stvarnim podacima iz literature.

[1] Astronomska jedinica (hrvatska kratica AJ ili aj, međunarodna kratica od eng. AU, skraćeno od Astronomical Unit) jest mjerna jedinica za duljinu. Predstavlja prosječnu udaljenost Zemlje od Sunca, a definira se kao 149 597 870 700 metara. [4]

2 Metoda

2.1 Numeričko rješavanje običnih diferencijalnih jednadžbi

Za razliku od algebarskih jednadžbi, koje opisuju vezu među varijablama, diferencijalne jednadžbe opisuju vezu neke funkcije $x(t)$ i njenih derivacija $x^{(n)}(t)$

$$F\left(t, x, \dot{x}, ..., x^{(n-1)}\right) = 0, n\epsilon\mathbb{N}. \tag{2.1}$$

Rješenje takve jednadžbe na intervalu I je svaka realna (ili kompleksna) dovoljno glatka funkcija

$$x : I \rightarrow \mathbb{R}(\mathbb{C}) \tag{2.2}$$

čijim uvrštavanjem u jednadžbu (2.1) dobivamo jednakost za svaku vrijednost varijable $t\epsilon I$ [7]. Kod numeričkog rješavanja običnih diferencijalnih jednadžbi, cilj nam je odrediti rješenje $y(t)$ diferencijalne jednadžbe

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f\left(t, y\right) \tag{2.3}$$

na intervalu $t \; \epsilon \; [t_0, t_N]$ s početnim uvjetom

$$y(t_0) = y_0. \tag{2.4}$$

Za numeričko rješavanje potrebno je diskretizirati prostor. Prostor se diskretizira podjelom domene na N jednakih vremenskih intervala duljine

$$h = \frac{t_N - t_0}{N}. \tag{2.5}$$

Pri odabiru vremenskih intervala moramo pripaziti da su dovoljno mali da ne mijenjaju konačno rješenje. Svodimo problem na određivanje N vrijednosti

$$y_i \equiv y(t_i), \; i = 1, ..., N \tag{2.6}$$

gdje je vrijeme u nekom trenutku i

$$t_i \equiv t_0 + ih = t_{i-1} + h. \tag{2.7}$$

Rješenje u trenutku $t_{i+1} = t_i + h$ možemo zapisati u obliku [7]

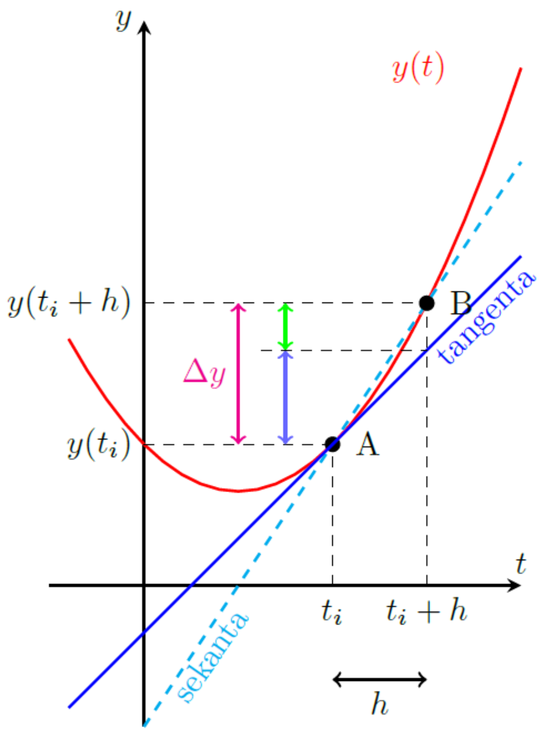
$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t, y)\mathrm{d}t. \tag{2.8}$$

Nakon *j* periodičnih koraka možemo dobiti y_i. Rješenje gore navedenog integrala nije trivijalno. Podintegralna funkcija može ovisiti o $y(t)$, čije nam ponašanje nije poznato u promatranom vremenskom intevalu, stoga želimo pronaći pravilo kojim ćemo moći aproksimirati vrijednost integrala.
Postoji nekoliko čestih metoda rješavanja ovog problema. U radu biramo Eulerovu metodu koja će biti objašnjena u sljedećem poglavlju radi jednostavnosti samog algoritma.

2.1.1 Eulerova metoda

Diferencijalnu jednadžbu (2.3) najjednostavnije možemo prebaciti u diskretni oblik koristeći pravilo za numeričko deriviranje [7]. Numeričko deriviranje opisujemo prateći oznake na slici 1. Neka je $h \equiv \Delta t$ prirast argumenta u točki t_i. Tada je pripadni prirast funkcije

$$\Delta y \equiv \Delta y(t_i) = y\,(t_i + h) - y(t_i). \tag{2.9}$$



Slika 1: *Skica prirasta funkcije $y(t)$ u točki t_i (slika preuzeta iz [7])*

Derivacija funkcije y u točki $t = t_i$ je limes [7]:

$$f(t_i, y_i) = y'(t_i) \equiv \frac{\mathrm{d}y}{\mathrm{d}t}(t_i) = \lim_{h \to 0} \frac{\Delta y(t_i)}{h}. \tag{2.10}$$

Kada $B \to A\,(h \to 0)$ sekanta $\bar{AB}$ postaje tangenta na graf od y u točki A [7]

$$y = y(t_i) + y'(t_i)(t - t_i). \tag{2.11}$$

Ako je y diferencijabilna funkcija u t_i, tada je

$$\Delta y = y'(t_i)h + \epsilon h. \tag{2.12}$$

Kada $h \to 0$ tj. kada je dovoljno malen, vrijedi linearna aproksimacija

$$\Delta y \approx y'(t_i)h \tag{2.13}$$

te za dovoljno male h rekurzija (2.8) postaje

$$y_{i+1} = y_i + f(t_i, y_i) \cdot h \equiv y_i + k_1 h. \tag{2.14}$$

Prethodni izraz je aproksimacija izraza (2.12).

2.2 Numeričko rješavanje diferencijalnih jednadžbi višeg reda

Obzirom da promatramo gibanje planeta pod utjecajem gravitacije, problem možemo svesti na numeričko rješavanje Newtonovih jednadžbi gibanja.
Drugi Newtonov zakon glasi:

$$\vec{F} = \frac{d\vec{p}}{dt} = m\frac{d\vec{v}}{dt}. \tag{2.15}$$

Ukoliko promatramo tijelo konstantne mase, jednadžba postaje

$$\ddot{x} = \frac{F(t, x, v)}{m}. \tag{2.16}$$

Uvođenjem pokrate $\dot{x} = v$ problem se svodi na paralelno rješavanje sustava vezanih jednadžbi prvog reda [7]

$$\dot{v} = \frac{F(t, x, v)}{m}, \tag{2.17}$$

$$\dot{x} = v. \tag{2.18}$$

U promatranom slučaju jednadžba gibanja je

$$\ddot{x} = \frac{\mathrm{d}v}{\mathrm{d}t} = \frac{F_g}{m} \tag{2.19}$$

gdje je F_g sila gravitacije, tj. gravitacijsko međudjelovanje dvaju ili više tijela. Za gravitacijsku silu trebamo uzeti u obzir vektorske udaljenosti planeta pa za izraz sile gdje planet *i* djeluje na planet *j* vrijedi

$$\vec{F_{ji}} = -G\frac{m_i m_j}{r_{ij}^3}\vec{r_{ij}} \tag{2.20}$$

gdje su m_i i m_j mase promatranih planeta, r_{ij} njihova udaljenost, a G gravitacijska konstanta [2]. Koristeći prethodni izraz jednadžba gibanja (2.19) postaje

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \vec{a} = -G\frac{m_i}{r_{ij}^3}\vec{r_{ij}}. \tag{2.21}$$

[2]Gravitacijska konstanta, prirodna je konstanta koja se pojavljuje u općem zakonu gravitacije kao faktor razmjernosti privlačne sile između dvaju tijela na određenoj udaljenosti, i njihovih masa. Današnja izmjerena vrijednost gravitacijske konstante iznosi: $G = 6,67408 \cdot 10^{-11} Nm^2/kg^2$.

Pritom r_{ij} računamo formulom za udaljenost dvije točke

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{2.22}$$

gdje su x_i i y_i koordinate $i - tog$ tijela, a x_j i y_j koordinate $j - tog$.
Prema jednadžbi (2.21) promjenu brzine računamo kao

$$dv = a \cdot dt. \tag{2.23}$$

Analogno, za promjenu položaja možemo pisati

$$dx = v \cdot dt. \tag{2.24}$$

Diferencijalne jednadžbe (2.23) i (2.24) su baza prema kojoj gradimo algoritam same računalne simulacije opisan u sljedećem poglavlju.

3 Algoritam

Rješavamo problem računalnom simulacijom u C programskom jeziku. Koristimo prethodno navedenu metodu za simuliranje gibanja planeta oko Sunca. Ne simuliramo svih osam planeta jer se fokusiramo na unutarnji dio Sunčevog sustava, odnosno na Zemlju. Jupiter smo uključili u simulaciju zato što je poznat po tome da svojom gravitacijom privlači asteroide i "čuva" Zemlju od sudara [8]. Dakle, promatramo pet Suncu najbližih planeta: Merkur, Venera, Zemlja, Mars i Jupiter. Putanje planeta Sunčeva sustava su nagnute manje od 7° u odnosu na Zemljinu, pa smatramo da se gibanje odvija približno u ravnini i promatramo 2D problem (samo x i y komponente) [9]. Radi jednostavnosti, planete tretiramo kao točke u prostoru konstantne mase, pa njihova jednadžba gibanja (2.21) sadrži ukupnu gravitacijsku silu koja djeluje na promatrani planet. Znamo da sila gravitacije ovisi o gravitacijskoj konstanti, masi pojedinog planeta te njihovoj međusobnoj udaljenosti. Za izračunati akceleraciju, moramo za svaki planet definirati početne uvjete, tj. početne vrijednosti navedenih veličina. Poznajemo iznose masa planeta m_i te udaljenost d_i svakog planeta od Sunca [9]. Najlakše je na početku planete rasporediti po $x - osi$ jer je tada trivijalno odrediti komponente položaja i brzina. Vrijedi sljedeće:

$$y_i(0) = 0, \tag{3.1}$$

$$x_i(0) = d_i. \tag{3.2}$$

Brzina u početnom trenutku je usmjerena u y smjeru te je iznosom jednaka orbitalnoj brzini planeta, čije vrijednosti poznajemo iz literature [9].
Nakon što smo definirali početne uvjete Eulerovom metodom za svako nebesko tijelo računamo brzinu i položaj u svakom trenutku t. Za svaki od N planeta, x i y komponenta akceleracije $i - tog$ planeta u trenutku t iznosi

$$a_x(t) = \frac{x_0 - x_i}{(d_i - d_0)^3} \cdot G \cdot m_0 + \frac{x_1 - x_i}{(d_i - d_1)^3} \cdot G \cdot m_1 + \cdots + \frac{x_N - x_i}{(d_i - d_N)^3} \cdot G \cdot m_N, \tag{3.3}$$

$$a_y(t) = \frac{y_0 - y_i}{(d_i - d_0)^3} \cdot G \cdot m_0 + \frac{y_1 - y_i}{(d_i - d_1)^3} \cdot G \cdot m_1 + \cdots + \frac{y_N - y_i}{(d_i - d_N)^3} \cdot G \cdot m_N, \tag{3.4}$$

gdje uzimamo u obzir međudjelovanje svih planeta na $i - ti$ planet. Brzine, a zatim i položaje je jednostavno izračunati iz jednadžbi (2.23) i (2.24). Brzina u trenutku $t + dt$, gdje je dt vremenski korak, iznosi

$$v_x(t + dt) = v_x(t) + a_x(t)dt, \tag{3.5}$$

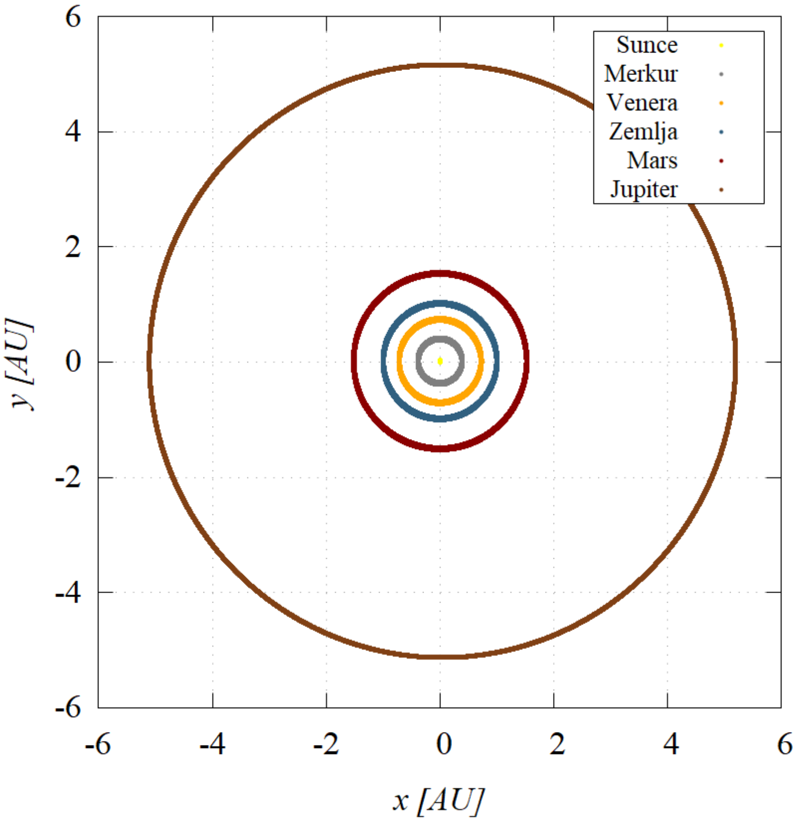
$$v_y(t + dt) = v_y(t) + a_y(t)dt. \tag{3.6}$$

Analogno, položaj je jednak

$$x_x(t+dt) = x_x(t) + v_x(t)dt, \tag{3.7}$$

$$x_y(t+dt) = x_y(t) + v_y(t)dt. \tag{3.8}$$

Spremamo dobivene iznose položaja da bismo kasnije mogli pratiti udaljenosti s asteroidima i grafički prikazati gibanje (slika 2).



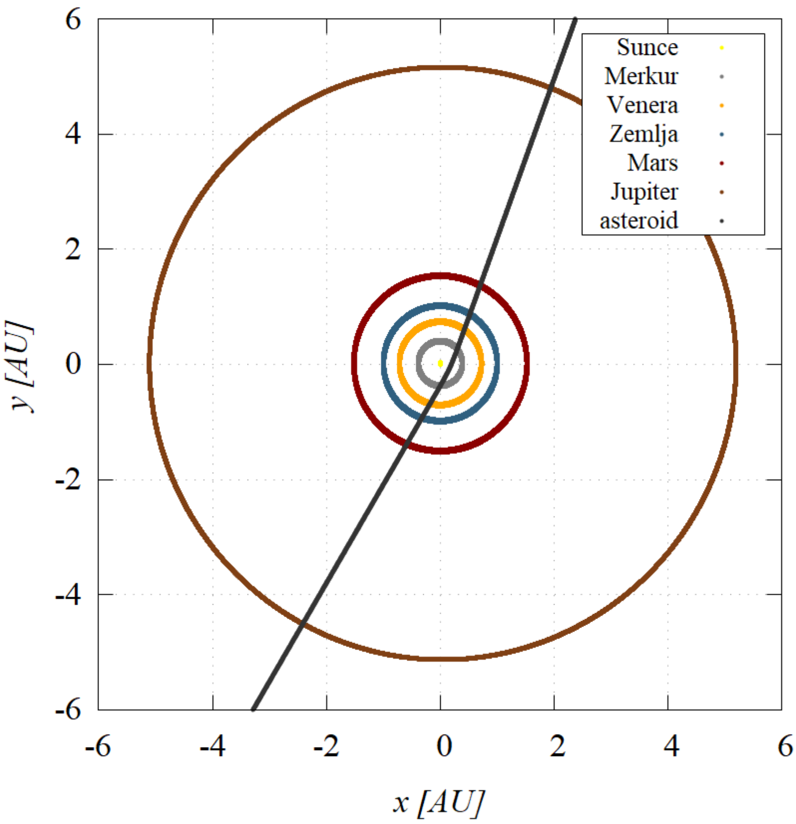
Slika 2: *Simulirana vizualizacija gibanja unutarnjih 5 planeta oko Sunca, prije dodavanja putanja asteroida.*

Nakon što smo simulirali Sunčev sustav trebamo simulirati asteroide. Na asteroide djeluju iste sile kao i na planete te se način računanja položaja ne razlikuje od onoga ranije objašnjenog. Jedino što je drugačije kod simulacije asteroida su početni uvjeti. Kako bismo dobili raznolike slučajeve, generiramo asteroide tako da im nasumično biramo masu, početnu brzinu i početni položaj. Koristimo generator nasumičnih brojeva iz *Numerical Recipies [10]* i prilagođavamo ga za dolje navedene raspone veličina već poznatih asteroida [11] kako bismo se što bolje približili stvarnim uvjetima

$$m_a \epsilon \, [0.1, 9.4] \cdot 10^{20} kg, \tag{3.9}$$

$$v_a \epsilon \, [15, 51.7] \, km/s. \tag{3.10}$$

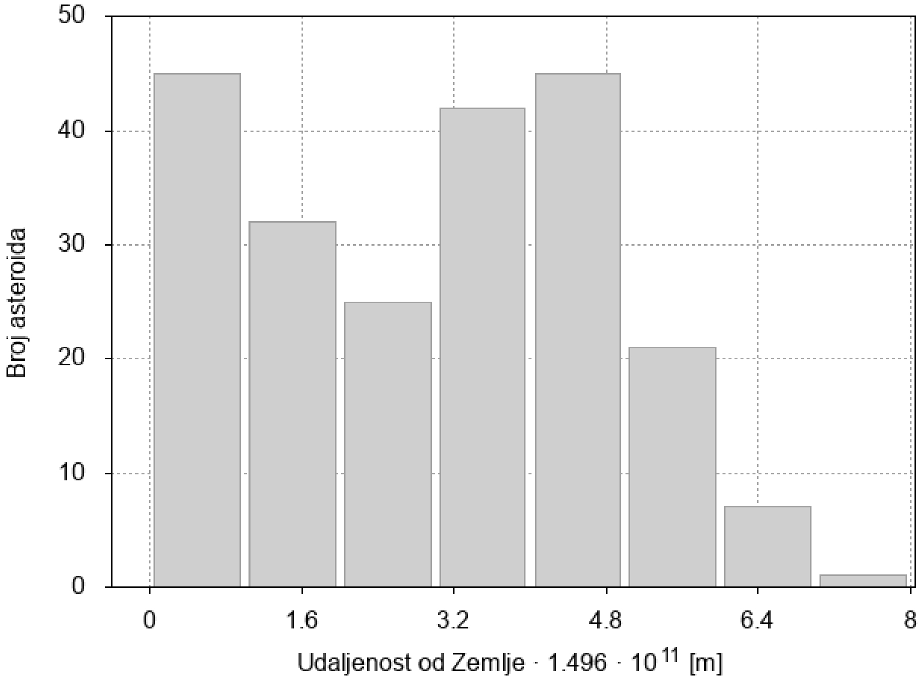
Želimo spriječiti stvaranje nepotrebno velikog broja datoteka za svaki asteroid. Iz tog razloga, ne razlikujemo n asteroida kao n novih tijela, već kada računanjem udaljenosti asteroida od Sunca zaključimo da je asteroid prošao orbitu Jupitera, isti se asteroid vraća u simulaciju generiranjem položaja, brzine i mase. Asteroide generiramo jedan za drugim zato što ne želimo simulirati više od jednog asteroida u isto vrijeme. Istovremeno s planetima ispisujemo i položaje asteroida u svakom trenutku te spremamo najmanju udajenost od Zemlje do koje je došao svaki asteroid. Osim toga, spremamo i broj asteroida koji uđu u Zemljinu orbitu, tj. prođu kroz krug polumjera 1.2 AU sa središtem u Suncu. Na slici 3 je prikazan jedan primjer prolaska asteroida kroz Sunčev sustav. Ovo je primjer i za asteroid koji ulazi u Zemljinu orbitu te predstavlja potencijalnu opasnost.



Slika 3: *Simulacija Sunčevog sustava s jednim asteroidom koji je prošao unutar orbite Zemlje.*

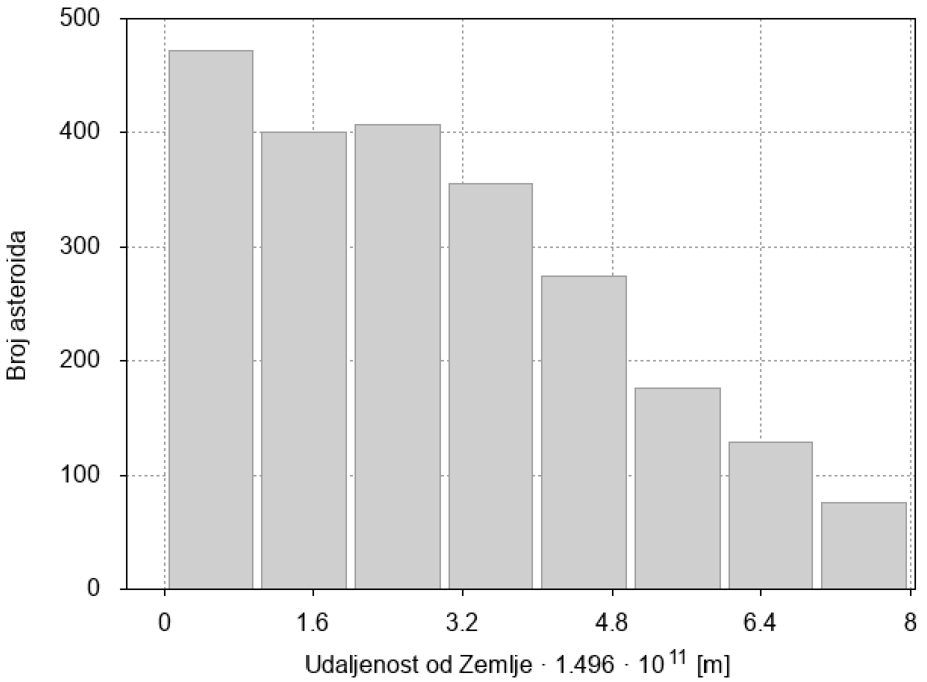
4 Rezultati

Izmjerene udaljenosti asteroida od Zemlje prikazujemo histogramima gdje $y - os$ predstavlja broj asteroida, a $x - os$ udaljenost od Zemlje u metrima. Želimo prikazati koliko asteroida dođe na određene udaljenosti od Zemlje te, ako je moguće, usporedbom s podacima iz literature zaključiti kolika je vjerojatnost da asteroid pogodi Zemlju.
U svrhu testiranja algoritma, izabrali smo za početak da vrijeme trajanja simulacije bude sto godina. Obzirom da smo zadali da se novi asteroid generira nakon što prethodni prođe orbitu Jupitera, asteroidi se generiraju u prosjeku dvaput godišnje. Dakle, ako je vrijeme trajanja simulacije T, broj asteroida će biti približbo $2T$.

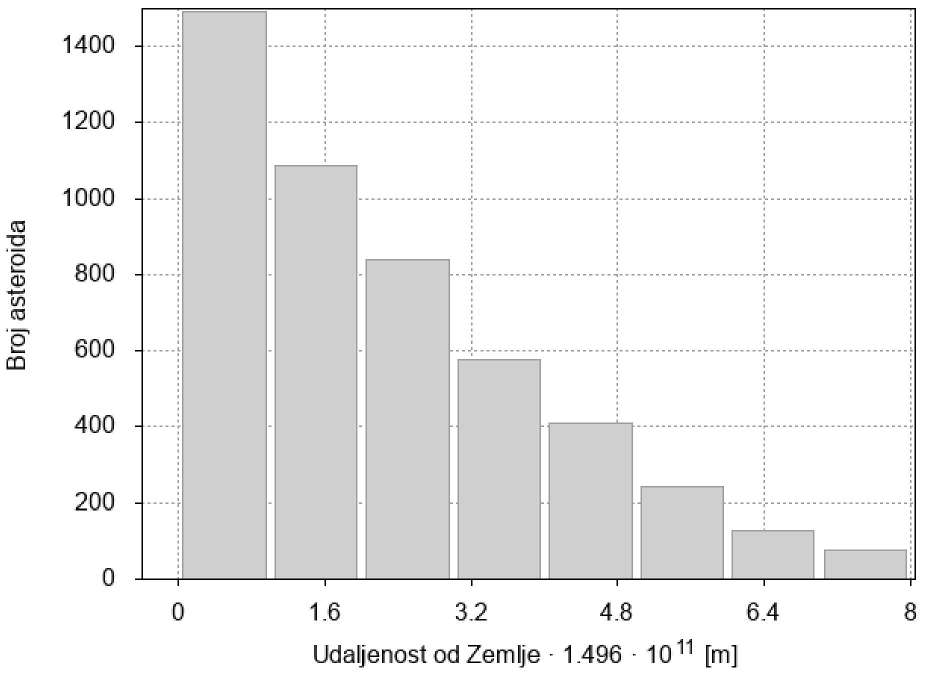


Slika 4: *Histogram koji prikazuje broj asteroida koji dođu na različite minimalne udaljenosti od Zemlje tokom simulacije u trajanju od sto godina*

Na slici 4 je prikazan broj asteroida te udaljenosti od Zemlje. Rezultati nam pokazuju da najviše asteroida dođe na udaljenost između 0 i 1 AU te 4 i 5 AU. Nakon što smo manjim vremenskim intervalom testirali radi li algoritam, prelazimo na veće vremenske intervale.
Povećavamo trajanje simulacije na tisuću godina i dvije tisuće godina, te promatramo iste intervale udaljenosti od Zemlje. Navedeni intervali su odabrani jer se pri većim intervalima simulacija sporije odvija i zahtjeva više memorije računala. U oba slučaja se najviše asteroida nalazilo na udaljenosti od 1 AU te je njihov broj opadao s udaljenosti kao što vidimo na slikama 5 i 6. Najbliže što je asteroid prišao Zemlji u tisuću godina trajanja simulacije je 0.003682 AU, što je više od 80 radijusa Zemlje.



Slika 5: *Histogram koji prikazuje broj asteroida koji dođu na različite minimalne udaljenosti od Zemlje tokom simulacije u trajanju od tisuću godina*



Slika 6: *Histogram koji prikazuje broj asteroida koji dođu na različite minimalne udaljenosti od Zemlje tokom simulacije u trajanju od dvije tisuće godina*

Promotrit ćemo i šire područje oko Zemlje, odnosno, želimo dio sustava oko Zemljine orbite označiti kao "opasno područje". Ukoliko postoje asteroidi koji su presjekli Zemljinu orbitu možda bi i došlo do sudara da se dogodio u nekom drugom trenutku. Točnije, mjerit ćemo koliko asteroida uđe u Zemljinu orbitu, tj. na udaljenost od 1.2 AU. Za ukupno 2892 asteroida, broj asteroida opasnih za Zemlju je zabilježeno 550 što čini približno 19%. Za ukupno 7917 asteroida, generiranih u najduljoj simulaciji, broj asteroida opasnih za Zemlju je 1741 što čini 22%.

5 Zaključak

U ovom smo radu promatrali asteroide u Sunčevom sustavu. Promatrani asteroidi su masom i brzinom slični već poznatim asteroidima iz literature kako bismo dobili što realnije rezultate. Cilj nam je bio razviti računalnu simulaciju gibanja asteroida u Sunčevom sustavu i vjerojatnost sudara asteroida sa Zemljom. Za razvoj algoritma korišten je Drugi Newtonov zakon, te metoda numeričkog rješavanja diferencijalnih jednadžbi višeg reda. Radi jednostavnosti, gibanje nebeskih tijela promatramo kao gibanje tijela konstantne mase pod utjecajem sile gravitacije. Sila na svako tijelo korištena je za izračun akceleracije tijela u nekom trenutku t. Nadalje smo Eulerovom metodom iz akceleracije računali brzinu i položaj tijela u svakom trenutku. Asteroide smo tretirali kao i planete s razlikom u početnim uvjetima. U konačnici smo dobili najmanje udaljenosti asteroida od Zemlje i broj asteroida koji je prošao kroz Zemlji "opasno područje". Opasno područje smo odredili kao područje oko Sunca malo šire od putanje Zemlje, konkretno 1.2 AU. Simulaciju smo provodili za različite vremenske intervale (sto, tisuću i dvije tisuće godina). Sa sva tri različita vremenska intervala smo dobili kvalitativno slične rezultate. U provedenoj simulaciji nijedan asteroid nije došao bliže od udaljenosti 80 radijusa Zemlje, tj. nijedan asteroid nije pogodio Zemljinu površinu. Usprkos tome, približno 20% generiranih asteroida je proslo kroz Zemljinu orbitu. U stvarnosti zadnji asteroid koji je sudarom sa Zemljom prouzrokovao veliku štetu se pojavio prije otprilike 65 milijuna godina pa rezultat gdje nam ni jedan asteroid ne pogađa Zemlju ima smisla.

Mana izrade ovakvog algoritma je što ne možemo promatrati veći broj godina zbog nastajanja jako velikih datoteka za koje su potrebna moćnija računala ili drugačiji način praćenja podataka. Kad bismo mogli promatrati puno veći vremenski interval, sigurno bi dobili nekolicinu asteroida koji bi se sudarili sa Zemljom. Također bi bilo zanimljivo promatrati što bi se dogodilo ako asteroid udari o neki planet bliže Zemlji, i kako bi to utjecalo na Zemlju i sami Sunčev sustav. Kao daljni rad, mogao bi se promatrati utjecaj svih osam planeta Sunčevog sustava, te detaljnije ispitati utjecaj Jupitera na putanje asteroida. Također, za daljnje simuliranje bi mogli proučiti preciznost Eulerove metode, te testirati različite metode. Obzirom da problem simuilramo diskretno, postoji mogućnost da je asteroid pogodio neki planet, ali se u točno tom trenutku nisu mjerili položaji ili planeta ili asteroida.

U našoj simulaciji dobili smo da oko 20% asteroida prođe kroz područje udaljeno manje od 1.2 AU od Sunca, te time predstavljaju potencijalnu opasnost Zemlji. Obzirom da smo promatrali pojednostavljeni fizikalni model gibanja planeta i asteroida u Sunčevom sustavu, rezultati koje smo dobili su zadovoljavajući.

6 Literatura

[1] Sunčev sustav, Wikipedia, URL: https://hr.wikipedia.org/wiki/Sun%C4%8Dev_sustav

[2] Planets, Dwarf Planets and Small Solar System Bodies, International Astronomical Union, URL: https://www.iau.org/public/themes/pluto/

[3] Planets, NASA, URL: https://solarsystem.nasa.gov/planets/in-depth/

[4] Astronomska jedinica, Wikipedia, URL: https://hr.wikipedia.org/wiki/Astronomska_jedinica#cite_note-The_International_System_of_Units-1

[5] General Information on Asteroids, NASA, URL: https://nssdc.gsfc.nasa.gov/planetary/text/asteroids.txt

[6] Asteroid Fast Facts, NASA, URL: https://www.nasa.gov/mission_pages/asteroids/overview/fastfacts.html

[7] Bilješke s predavanja, Obične diferencijalne jednadžbe (ODJ), Petar Stipanović, 2020/21..

[8] Saving the World One Asteroid at a Time, The Planetary Society, URL: https://www.planetary.org/the-downlink/saving-the-world-one-asteroid-at-a-time

[9] Simuliranje gibanja, bilješke s predavanja, Petar Stipanović, 2021/2022.

[10] Numerical Recipes in C, Cambridge University Press, 2ed, 2002.

[11] Asteroids, Wikipedia, URL: https://en.wikipedia.org/wiki/Asteroid

A Priloženi kod

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

#include "ran1.c"

#define G 1.9739E-29 // m^3/s^2kg => Au^3/(year^2kg)
#define tmax 15.0 //year
#define dt 0.0001 //year

#define err 1.0E-96

#define n 7 //broj planeta (i sunce) + broj asteroida
#define p 6
#define k 1 //broj asteroida

//range, ASTEROIDI
#define mmin 0.1E20
#define mmax 9.4E20
#define rmin 15.0 //m
#define rmax 30000.0 //m
#define vmin 15.0 //km/s
#define vmax 51.7 //km/s
#define RZ 4.26E-5

double m[n], T[n], x[n], y[n], vx[n], vy[n], ax[n], ay[n], R[n];
int brojac[8] = {0,0,0,0,0,0,0,0}, brojacR[8] = {0,0,0,0,0,0,0,0}; // svi ↩
    podatci



double Random(float min,float max)
{
    // funkcija koja generira random broj u granicama [min,max]
    float broj;
    static long idum = (-2578); // RAN: generatoru u pocetku ↩
        prosljedjujemo negativni cijeli broj

    broj = ran1(&idum)*(max-min)+min;

    return broj;

}
```

```
void postavi_pocetne_vrijednosti_za_planete(){

    //planeti : Sunce, Merkur, Venera, Zemlja, Mars i Jupiter

    m[0]=2.0E30; //sunce
    m[1]=6.0E24; //zemlje
    m[2]=1.9E27; //jupiter
    m[3]=4.87E24; //venera
    m[4]=0.33E24; //merkur
    m[5]=0.642E24; //mars


    x[0]=0.0;
    x[1]=1.0; //Au
    x[2]=5.2;
    x[3]=0.723; //v
    x[4]=0.387; //m
    x[5]=1.5234;

    T[0]=1.0; // NIJE 1, VEC JE NE NULA, TAKO DONJA PETLJA NE DAJE NaN ZA ↩
        vy[0]
    T[1]=1.0; // 1 godina
    T[2]=11.9;
    T[3]=0.6153;
    T[4]=0.241;
    T[5]=1.8811;

    //radijusi -- dodaj


    for(int i=0; i<p; i++) {
      y[i]=vx[i]=0.0;
      vy[i]=2*M_PI*x[i]/T[i];
      //printf("y[%d] je %lf \n vx[%d] je %lf \n vy[%d] je %lf \n ", i, y[↩
          i], i, vx[i], i, vy[i]);
      //provjera
    }
}
    //asteroid
    //petlja u kojoj cemo generirati razlicite mase, pocetne brzine, ↩
        radijuse te udaljenost asteroida

void postavi_pocetne_vrijednosti_za_asteroide(){


        m[6] = Random(mmin, mmax);

```

```
        //koordinate
        x[6] = Random(-6.0, 6.0); //grid [-6, 6]

        if(x[6]==6.0 || x[6]==-6.0)
        {
            y[6] = Random(-6.0, 6.0);

        }
        else
        {
            if(Random(0,1)<0.5)
                y[6]=-6;
            else
                y[6]=6;
        }

        //brzina

        double v = (Random(vmin, vmax));

        double theta = Random(0, 0.5*M_PI);

        vx[6] = v*cos(theta);
        vy[6] = v*sin(theta);

        if(x[6]>0)
        {
            vx[6]=-abs(vx[6]);
        }
        else
        {
            vx[6]=abs(vx[6]);
        }

        if(y[6]>0)
        {
            vy[6]=-abs(vy[6]);
        }
        else
        {
            vy[6]=abs(vy[6]);
        }


    }


```

```c
void stvori_datoteke(FILE **dat, char *baza){ //stvaranje datoteke
    char ime[100], dodatak[100]; // stvaramo stringove
    for(int i=0;i<n;i++){
        strcpy(ime, baza); // u ime upisujemo bazu (onaj dio koji je isti svima)
        sprintf(dodatak, "_tijelo_%d.txt", i); // stvorimo dio karakteristican za jedan planet
        strcat(ime, dodatak); //zaljepimo taj karakteristicni dio na bazu
        dat[i]=fopen(ime,"w");//otvaramo file

        if(dat==NULL){
            printf("Greska s datotekom %s!!!!!!!!!!\n", ime);
            exit(EXIT_FAILURE);
        }

        fprintf(dat[i], "##t\t\t x\t\t y\t\t\n"); //upisuje zaglavlje
    }
}

void zatvori_datoteke(FILE **dat)
{
    for(int i=0;i<n;i++)
    {
        fclose(dat[i]);

    }
}

double d(int i, int j)
{
    if(i==j)
    {
        return 1.0;
    }
    else
    {
       return sqrt(pow((x[i]-x[j]),2)+pow((y[i]-y[j]),2)); // udaljenost i-tog i j-tog tijela
    }

}

double euler(int i)
{
    ay[i]=0.0;
```

```
    ax[i]=0.0;

    for(int j=0; j<n; j++)
    {
        ay[i]+=G*m[j]*((y[j]-y[i])/(pow((d(i,j)),3)));
        ax[i]+=G*m[j]*((x[j]-x[i])/(pow((d(i,j)),3)));
    }

    vx[i]+=ax[i]*dt;
    vy[i]+=ay[i]*dt;

    x[i]+=vx[i]*dt;
    y[i]+=vy[i]*dt;
}

void brojaci_2(double broj)
{

        if(broj <= 1.0)
        {
            //printf(" manje od 1\n");
            brojacR[0]++;
            //printf("brojac %d\n\n",brojac[0]);
        }
        else if( broj <= 2.0 && broj>1.0)
        {
            //printf(" manje od 2\n");
            brojacR[1]++;
        }
        else if( broj <= 3.0 && broj>2.0)
        {
            //printf(" manje od 3\n");
            brojacR[2]++;
        }
        else if( broj <= 4.0 && broj> 3.0)
        {
            //printf(" manje od 4\n");
            brojacR[3]++;
        }
        else if( broj <= 5.0 && broj>4.0)
        {
            //printf(" manje od 5\n");
            brojacR[4]++;
        }
        else if( broj <= 6.0 && broj>5.0)
        {
            //printf(" manje od 6\n");
```

```
            brojacR[5]++;
        }
        else if( broj <= 7.0 && broj>6.0)
        {
            //printf(" manje od 7\n");
            brojacR[6]++;
        }
        else if( broj <= 8.0 && broj>7.0)
        {
            //printf(" manje od 8\n");
            brojacR[7]++;
        }
}
int main (void){
    double t=0.0;

    FILE *dat[n];
    FILE *datoteka2;
    FILE *DAT;
    datoteka2 = fopen("BrojaciAU_2000_.txt","w");
    DAT = fopen("Udaljenosti_2000_.txt","w");
    char ime[] = "Sunce_Zemlja_pokusaj_1";
    stvori_datoteke(dat, ime);
    postavi_pocetne_vrijednosti_za_planete();
    postavi_pocetne_vrijednosti_za_asteroide();
    fprintf(datoteka2, " d<R\t d<2R\t d<3R\t d<4R\t d<5R\t d<6R\t d<7R\t d↩
       <8R\n",brojacR[0],brojacR[1],brojacR[2],brojacR[3],brojacR[4],↩
       brojacR[5],brojacR[6],brojacR[7]);


    int N=(int) tmax/dt;
    double najmanja_udaljenost=8.0,posto;
    int br=0,br_ast=0;

    //printf("%d", N);

    for(int j=1; j<=N;j++) //petlja po vremenu
    {
        t=dt*j;
        // planeti
        for(int i=0; i<n; i++)
        {
            fprintf(dat[i], " %lf \t%lf \t%lf \n\n\n", t, x[i], y[i]);
            euler(i);


        }
```

```
        if(d(0,6)<najmanja_udaljenost)
        {
            najmanja_udaljenost=d(0,6);
        }

        //gledamo je li asteroid prosao jupiter

        if(d(0,6) > 8.5)
        {

            brojaci_2(najmanja_udaljenost);
            //provjeravamo dode li asteroid u neposrednu blizinu Zemlje, tj. blizu Zemljine orbite
            if(najmanja_udaljenost<=1.2)
            {
                br++;

            }


            fprintf(DAT, "%lf\n",najmanja_udaljenost);
            postavi_pocetne_vrijednosti_za_asteroide();
            br_ast++;
            najmanja_udaljenost = 12.0;
            //printf("vrijeme %f  polozaj sunca %lf polozaj asteroida %lf\n udaljenost %f\n\n",t, x[0], x[6], broj);
        }






    }

    printf("Ukupan broj asteroida:%d, broj asteroida opasnih za Zemlju: %d\n",br_ast, br);
    fprintf(datoteka2, " %d\t %d\t %d\t %d\t %d\t %d\t %d\t %d\n",brojacR[0],brojacR[1],brojacR[2],brojacR[3],brojacR[4],brojacR[5],brojacR[6],brojacR[7]);


    zatvori_datoteke(dat);
    fclose(datoteka2);
    fclose(DAT);
    return 0;
}
```