Simulacije putanja planeta pomoću programskog jezika Python

Mimica, Roko

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Roko Mimica

SIMULACIJE PUTANJA PLANETA POMOĆU PROGRAMSKOG JEZIKA PYTHON

Diplomski rad

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA I INFORMATIKA; SMJER NASTAVNIČKI

Roko Mimica

Diplomski rad

Simulacije putanja planeta pomoću programskog jezika Python

Voditelj diplomskog rada: prof. dr. sc. Tamara Nikšić

Ocjena diplomskog rada: ______________________

Povjerenstvo: 1. ______________________

2. ______________________

3. ______________________

Datum polaganja: ________________

Zagreb, 2019.

Zahvaljujem se svojoj obitelji na razumijevanju i podršci, svojim profesorima na prenesenom znanju i trudu te kolegama na ugodnom društvu i pomoći. Naposlijetku se zahvaljujem svojoj mentorici na razumijevanju i pomoći u izradi ovog diplomskog rada.

Sažetak

U ovom radu ćemo prikazati računalnu simulaciju Keplerovog problema i diskutirati njenu primjenu u istraživački usmjerenoj nastavi fizike. Keplerov problem je specijalan slučaj problema dvaju tijela gdje tijela međudjeluju centralnim silama koje opadaju s kvadratom udaljenosti. U tom slučaju problem dvaju tijela možemo svesti na problem jednog tijela u centralnom polju koji možemo riješiti analitički, a potom koristeći to rješenje možemo rekonstruirati orbite zasebnih tijela. Uz pomoć vis-viva jednadžbe i činjenice da je kutna količina gibanja konstantna u slučaju da na tijelo djeluje samo centralna sila, možemo izračunati iznose relativnih brzina tijela na bilo kojem položaju u orbiti. Da bi bila primjenjivija u obrazovne svrhe, simulacija prikazuje grafove ovisnosti iznosa relativnih brzina i udaljenosti između tijela o kutu kojeg vektor položaja planeta formira s pozitivnom x osi te po potrebi demonstrira drugi i treći Keplerov zakon. Da bi što manje ovisila o vrsti računala i instaliranim programskim paketima, za izradu simulacije koristili smo VPython (verzija 7) te Glowscript (verzija 2.8) u Glowscript IDE razvojnom okruženju.

Ključne riječi: Keplerov problem, Keplerovi zakoni, računalne simulacije, VPython, Glowscript

Simulating planetary orbits with programming language Python

Abstract

In this thesis we demonstrate a computer simulation of Kepler problem and discuss its application for educational purposes. Kepler problem is a special case of two body problem where bodies interact exclusively by a central force that declines in strength as the inverse square of the distance. In this case, the two body problem can be reduced to one body acted on by an external central gravitational force whose analytical solution can be used to return to the real two-body problem and solve for the actual motion of the two original bodies. Using the vis-viva equation and a fact that in the case when the only force on the body is a central force, angular momentum with respect to the origin is a constant of motion, we can calculate relative speed of the two bodies and angular speed of the individual bodies at any position in orbit. In order to make the simulation more useful for educational purposes, it shows diagrams of relative speeds and distances between bodies with respect to the angle which the position vector of the bodies makes with the positive x axis. Simulation, if necessary, can also demonstrate second and third Kepler law, and was programmed using VPython (version 7) and Glowscript (version 2.8) in Glowscript IDE integrated development enviroment to minimize the dependence on operating systems and installed packages.

Keywords: Kepler problem, Kepler's laws, computer simulations, VPython, Glowscript

Sadržaj

1 Uvod

1.1 Računalne simulacije

U najužem smislu, računalna simulacija je program kojeg pokrećemo na računalu i koji koristi postepene, korak po korak metode kako bi aproksimirao ponašanje matematičkog modela koji se odnosi na nekakav realan ili hipotetski sustav. Algoritam kao ulaz prima vrijednosti parametara sustava u nekom vremenu t te računa stanje sustava u vremenu $t+\Delta t$. Iz vrijednosti koje karakteriziraju drugo stanje, algoritam računa stanje sustava u trenutku $t+2\Delta t$ itd. U širem smislu, računalne simulacije možemo smatrati kao sveobuhvatnu metodu za proučavanje sustava koja se referira na čitav proces. Proces uključuje biranje i implementiranje modela u obliku koji se može izvesti na računalu, na način da računamo izlazne vrijednosti algoritma te vizualiziramo i analiziramo dobivene podatke [4].

Matematičkim modelom sustava pokušavamo pronaći analitička rješenja koja nam omogućuju predviđanje ponašanja sustava za zadani skup parametara i početnih uvjeta. No, računalne simulacije se često koriste i pri modeliranju sustava koji nemaju analitičkih rješenja. U tom slučaju sustav rješavamo numerički, metodama poput Runge-Kutta, Eulerove, implicitnom trapeznom itd.

U interaktivnoj istraživački usmjerenoj nastavi fizike, računalne simulacije fizikalnih sustava su od posebnog značaja za provođenja pokusa koji su preskupi, vremenski prezahtjevni ili preopasni da bi se izvodili u nastavnom okruženju, ili zahtjevaju visoke eksperimentalne vještine.

U ovom radu ćemo modelirati orbite dvaju tijela pod utjecajem centralnih (gravitacijskih) sila uz pomoć analitičkih rješenja Keplerovog problema.

1.2 Kratka povijest proučavanja planetarnog gibanja

Astronomija je najstarija od svih prirodnih znanosti. Rane civilizacije su povezivale nebeske objekte s bogovima, a gibanja istih objekata s prirodnim fenomenima poput kiše, suše, godišnjih doba itd. Kako su se civilizacije razvijale, potreba za efikasnim metodama čuvanja podataka socijalnih aktivnosti tokom dužih vremenskih perioda postala je društveni imperativ. Proučavanja gibanja planeta, primarno Sunca i Mjeseca u odnosu na Zemlju čovječanstvu su dala prve kalendare.

Od antičke Grčke do istraživanja astronoma Nikole Kopernika i Johannesa Keplera, vjerovalo se u geocentrični model svemira u kojem se planeti gibaju po kružnim orbitama konstantnim brzinama. Aristotel je vjerovao da se Sunce, Mjesec, planeti i zvijezde gibaju po kružnim orbitama oko stacionarne Zemlje [5]. Na temelju te ideje, Klaudije Ptolomej u 2 st.pr.Kr. stvara kompletan kozmološki model, gdje je Zemlja nepomična u središtu dok se svaki planet giba po manjoj kružnici zvanom epicikl, čiji se centar giba po većoj kružnici zvanoj deferent [3]. Ptolomejev model se, uz male preinake, održao sve do 16 st.

U 16 st. Nikola Kopernik otkriva da bi se Ptolomejev model mogao poboljšati ukoliko bi umjesto Zemlje u središte svemira postavili Sunce [5]. U Kopernikovom modelu, povremeno retrogardno gibanje planeta proizlazi iz kombiniranog gibanja Zemlje i ostalih planeta. Kako se Zemlja giba oko Sunca većom brzinom, periodički "prestiže" vanjske planete zbog čega opažamo kao da se gibaju unatrag. Kopernikov model, iako poboljšanje u odnosu na Ptolomejev, još uvijek smatra da se planeti gibaju u kružnim orbitama konstantnim brzinama.

Johannes Kepler je prihvatio Kopernikovu heliocentričnu ideju, i koristeći iznimno precizna mjerenja pozicija planeta danskog astronoma Tycha Brachea formulira tri zakona planetarnog gibanja. Prvi zakon nam govori da se planeti oko Sunca gibaju po eliptičnim putanjama sa Suncem u fokusu. Drugi zakon opisuje brže gibanje planeta duž svoje eliptične putanje bliže Suncu. Treći zakon daje precizan odnos veličina planetarnih orbita i vremena potrebnog da planet zatvori svoju putanju. Kepler je napustio ideju kružnih putanja tek nakon što je iscrpio sve mogućnosti da poveže orbitu Marsa sa nekom vrstom kružnog gibanja.

Iako je Kepler otkrio zakonitosti gibanja planeta u Sunčevom sustavu, uzrok planetarnog gibanja i njegov matematički opis formulirao je Isaac Newton, postuliravši da se sve mase privlače silom proporcionalnom njihovim masama, a obrnuto proporcionalnom kvadratima njihovih udaljenosti.

2 Opis gibanja

2.1 Centralne sile

Pretpostavimo da se čestica mase m giba u polju centralne sile. Centralnim silama zovemo sile čiji potencijal ovisi samo o udaljenosti od izvora. Ukoliko se izvor nalazi u ishodištu, za potencijalnu energiju vrijedi $U(\vec{r}) = U(r)$. Posljedica takvog potencijala je radijalna sila čiji iznos ovisi samo o r [1]

$$\vec{F}(\vec{r}) = -\nabla U(r) = -\frac{dU}{dr}\hat{r}. \tag{2.1}$$

Primjer centralnih sila su gravitacijska i elektrostatska sila, za koje vrijedi $U(r) \propto 1/r$. Promotrimo vremensku promjenu kutne količina gibanja čestice na koju djeluje isključivo centralna sila. Vremenska promjena kutne količine gibanja jednaka je zakretnom momentu $\vec{M}$

$$\frac{d\vec{L}}{dt} = \vec{M} = \frac{d}{dt}(\vec{r} \times \vec{p}) = \frac{d\vec{r}}{dt} \times \vec{p} + \vec{r} \times \frac{d\vec{p}}{dt} = \vec{v} \times (m\vec{v}) + \vec{r} \times \vec{F} = 0, \tag{2.2}$$

jer $\vec{F} \propto \vec{r}$ i vektorski produkt dva paralelna vektora iznosi 0, stoga nema promjene kutne količine gibanja u vremenu i konstantan je vektor

$$\frac{d\vec{L}}{dt} = 0 \Longrightarrow \vec{L} = konst. \tag{2.3}$$

Iz definicije kutne količine gibanja

$$\vec{L} = \vec{r} \times \vec{p} = m\vec{r} \times \dot{\vec{r}}, \tag{2.4}$$

sljedi da je vektor kutne količine gibanja okomit na vektore položaja i brzine, što znači da se putanja čestice nalazi u ravnini okomitoj na vektor $\vec{L}$. Koordinatni sustav orijentiramo tako da će vektor $\vec{L}$ biti usmjeren u smjeru z osi, a putanja čestice će se nalaziti u xy ravnini.

2.2 Problem dvaju tijela

Promatramo sustav dvije čestice koje međudjeluju centralnom silom. Položaje čestica označimo s $\vec{r}_1$ i $\vec{r}_2$, a potencijalnu funkciju s U($|\vec{r}_1 - \vec{r}_2|$). Jednadžbe gibanja su

$$m_1\ddot{\vec{r}}_1 = -\nabla_{\vec{r}_1} U(|\vec{r}_1 - \vec{r}_2|), \tag{2.5}$$

$$m_2\ddot{\vec{r}}_2 = -\nabla_{\vec{r}_2} U(|\vec{r}_1 - \vec{r}_2|). \tag{2.6}$$

Treći Newtonov zakon nalaže da su sile jednakog iznosa i suprotnog smjera, pa koristimo relaciju

$$\nabla_{\vec{r}_1} U(|\vec{r}_1 - \vec{r}_2|) = -\nabla_{\vec{r}_2} U(|\vec{r}_1 - \vec{r}_2|) = \nabla_{\vec{r}} U(|\vec{r}|), \tag{2.7}$$

gdje je $\vec{r} \equiv \vec{r}_1 - \vec{r}_2$. Zbrojimo jednadžbe (2.5) i (2.6)

$$m_1\ddot{\vec{r}}_1 + m_2\ddot{\vec{r}}_2 = 0 \Longrightarrow \frac{d^2}{dt^2}(m_1\vec{r}_1 + m_2\vec{r}_2) = 0. \tag{2.8}$$

Kombinaciju u zagradi prepoznajemo kao definiciju centra mase sustava

$$m_1\vec{r}_1 + m_2\vec{r}_2 = (m_1 + m_2)\,\vec{R}_{cm}, \tag{2.9}$$

što nam govori da se iz jednadžbe (2.8) može zaključiti da se centar mase može gibati samo jednoliko po pravcu. Pošto su svi inercijalni sustavi ravnopravni, izabiremo onaj kojem centar mase miruje u ishodištu

$$m_1\vec{r}_1 + m_2\vec{r}_2 = 0 \Longrightarrow \vec{r}_1 = -\frac{m_2}{m_1}\vec{r}_2. \tag{2.10}$$

Kao drugu kombinaciju vektora $\vec{r}_1$ i $\vec{r}_2$, linearno nezavisnu od položaja centra mase biramo relativnu koordinatu $\vec{r}$, do čije jednadžbe gibanja dolazimo oduzimanjem jednadžbi (2.5) i (2.6).

$$\ddot{\vec{r}} = -\frac{m_1 + m_2}{m_1 m_2}\nabla_{\vec{r}} U(|\vec{r}|). \tag{2.11}$$

Ovime smo početni problem, koji je uključivao dva tijela masa m_1 i m_2 i centralnu silu $U(|\vec{r}_1 - \vec{r}_2|)$, sveli na ekvivalentan problem jednog tijela mase $\mu = \frac{m_1 m_2}{m_1 + m_2}$ koje se giba u centralnom polju $U(|\vec{r}|)$

$$\frac{m_1 m_2}{m_1 + m_2}\ddot{\vec{r}} = -\nabla_{\vec{r}} U(|\vec{r}|) \Longrightarrow \mu\ddot{\vec{r}} = -\nabla_{\vec{r}} U(|\vec{r}|). \tag{2.12}$$

Masu $\mu = \frac{m_1 m_2}{m_1+m_2}$ zovemo reduciranom masom sustava.

2.3 Jednadžba orbite jednog tijela

Ekvivalentan problem jednog tijela ima dvije konstante gibanja, ukupnu energiju i kutnu količinu gibanja oko ishodišta. Energija je konstanta jer je centralna sila u uvodnom problemu dva tijela konzervativne prirode. Kutna količina gibanja je konstantna jer nema zakretnog momenta u odnosu na ishodište (2.2) [2]. Vektor kutne količine gibanja smo usmjerili u smjeru pozitivne z osi, i razmatramo orbite u xy ravnini.

Razmotrimo tijelo mase μ, koje kruži oko neke točke pod utjecajem privlačne gravitacijske sile

$$\vec{F} = -G\frac{m_1 m_2}{r^2}\hat{r}. \tag{2.13}$$

Potencijalna energija, s izborom referentne točke u beskonačnosti U(∞) = 0 je dana s

$$U(r) = -\frac{Gm_1 m_2}{r}. \tag{2.14}$$

Ukupna energija E, iskazana kao zbroj kinetičke i potencijalne energije iznosi

$$E = \frac{1}{2}\mu v^2 - \frac{Gm_1 m_2}{r}. \tag{2.15}$$

Član kinetičke energije $\mu v^2/2$ je izražen kao umnožak reducirane mase i iznosa relativne brzine dvaju tijela. U daljnjim razmatranjima ćemo koristiti polarni koordinatni sustav gdje je

$$\vec{v} = v_r\hat{r} + v_\theta\hat{\theta} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta}, \tag{2.16}$$

$$v = |\vec{v}| = \left|\frac{d\vec{r}}{dt}\right|,$$

gdje je $v_r = dr/dt$ i $v_\theta = r(d\theta/dt)$. Jednadžba (2.15) sada postaje

$$E = \frac{1}{2}\mu\left[\left(\frac{dr}{dt}\right)^2 + \left(r\frac{d\theta}{dt}\right)^2\right] - \frac{Gm_1 m_2}{r}. \tag{2.17}$$

Kutna količina gibanja s obzirom na ishodište je dana s

$$\vec{L} = \vec{r} \times \mu \vec{v} = r\hat{r} \times \mu \left(v_r \hat{r} + v_\theta \hat{\theta} \right) = \mu r v_\theta \hat{z} = \mu r^2 \frac{d\theta}{dt} \hat{z} = L\hat{z}, \tag{2.18}$$

čiji je iznos

$$L = \mu r v_\theta = \mu r^2 \frac{d\theta}{dt} = \mu r^2 \dot{\theta}. \tag{2.19}$$

Koristeći izraz (2.19) želimo eliminirati ovisnost o θ u (2.17)

$$\frac{d\theta}{dt} = \frac{L}{\mu r^2}. \tag{2.20}$$

Mehanička energija izražena jednadžbom (2.17) postaje

$$E = \frac{1}{2} \mu \left(\frac{dr}{dt} \right)^2 + \frac{1}{2} \frac{L^2}{\mu r^2} - \frac{G m_1 m_2}{r}. \tag{2.21}$$

Jednadžba (2.21) je separabilna diferencijalna jednadžba koja sadrži varijablu r kao funkciju vremena

$$\frac{dr}{dt} = \sqrt{\frac{2}{\mu}} \left(E - \frac{1}{2} \frac{L^2}{\mu r^2} + \frac{G m_1 m_2}{r} \right)^{\frac{1}{2}}. \tag{2.22}$$

Da bismo dobili varijablu r kao funkciju θ, dijelimo jednadžbu (2.20) s (2.22)

$$\frac{d\theta}{dr} = \frac{\frac{d\theta}{dt}}{\frac{dr}{dt}} = \frac{L}{\sqrt{2\mu}} \frac{(1/r^2)}{\left(E - \frac{L^2}{2\mu r^2} + \frac{G m_1 m_2}{r} \right)^{\frac{1}{2}}}. \tag{2.23}$$

Separiramo varijable r i θ

$$d\theta = \frac{L}{\sqrt{2\mu}} \frac{(1/r^2)}{\left(E - \frac{L^2}{2\mu r^2} + \frac{G m_1 m_2}{r} \right)^{\frac{1}{2}}} dr. \tag{2.24}$$

Integriranjem jednadžbe (2.24) dolazimo do radijusa kao funkcije kuta θ

$$r(\theta) = \frac{p}{1 + e\cos(\theta - \theta_0)}, \tag{2.25}$$

gdje je

$$p = \frac{L^2}{\mu G m_1 m_2}, \tag{2.26}$$

konstanta koju zovemo parametrom orbite, dok veličinu

$$e = \left(1 + \frac{2EL^2}{\mu\left(Gm_1m_2\right)^2}\right)^{\frac{1}{2}}, \tag{2.27}$$

zovemo ekscentritetom orbite i određuje oblik orbite tijela. Za $e = 0$ tijela će se gibati u kružnim orbitama, za $0 < e < 1$ u eliptičnim orbitama, dok za $e = 1$ i $e > 1$ dobivamo parabolične i hiperbolične orbite. U ovom radu ćemo razmatrati samo kružne i eliptične orbite.

Jednadžbu (2.25) zovemo zovemo jednadžbom orbite tijela reducirane mase. Pericentar orbite, kojeg definiramo kao točku orbite u kojoj je udaljenost od centra polja najmanja, nalazi se u točki

$$\theta = \theta_0,\ r_{min} = \frac{p}{1+e}. \tag{2.28}$$

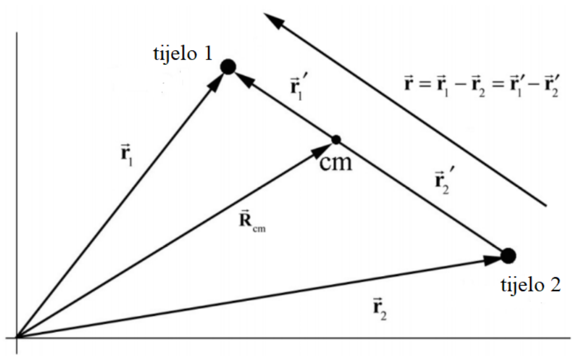
Koordinatni sustav ćemo orijentirati tako da se pericentar nalazi na osi x. Takav izbor odgovara konstanti $\theta_0 = 0$. U tom slučaju apocentar, najdalja točka orbite od centra polja se nalazi u točki

$$\theta = \pi,\ r = \frac{p}{1-e}. \tag{2.29}$$

2.4 Orbite dvaju tijela

Jednom kada nam je poznato rješenje problema jednog tijela $r(\theta)$, možemo pronaći orbite dvaju tijela. Koordinatni sustav smo postavili na način da se tijelo 1 nalazi na položaju $\vec{r}_1$, a tijelo 2 na položaju $\vec{r}_2$. Centar mase sustava je dan s

$$\vec{R}_{cm} = \frac{m_1\vec{r}_1 + m_2\vec{r}_2}{m_1 + m_2}. \tag{2.30}$$



Slika 2.1: Koordinatni sustav [2].

Neka $\vec{r_1}'$ bude vektor od centra mase do tijela 1, a $\vec{r_2}'$ vektor od centra mase do tijela 2 (Slika 2.1). U tom slučaju vrijedi

$$\vec{r_1}' = \vec{r_1} - \vec{R}_{cm} = \vec{r_1} - \frac{m_1\vec{r_1} + m_2\vec{r_2}}{m_1 + m_2} = \frac{m_2\left(\vec{r_1} - \vec{r_2}\right)}{m_1 + m_2} = \frac{\mu}{m_1}\vec{r}. \tag{2.31}$$

Na sličan način se može pokazati da je

$$\vec{r_2}' = -\frac{\mu}{m_2}\vec{r}, \tag{2.32}$$

gdje je

$$\vec{r} = \vec{r_1}' - \vec{r_2}'. \tag{2.33}$$

Oba tijela se gibaju oko centra mase na isti način kao što se tijelo reducirane mase giba oko centra polja (2.25), uz razliku da je udaljenost oba tijela do centra mase skraćena za faktor μ/m_i. Ukoliko je masa jednog tijela mnogo veća u odnosu na drugo, npr. $m_1 << m_2$, reducirana masa je približno jednaka manjoj masi [2]

$$\mu = \frac{m_1 m_2}{m_1 + m_2} \cong \frac{m_1 m_2}{m_2} = m_1, \tag{2.34}$$

te tijelo veće mase približno miruje u ishodištu, dok se lakše tijelo giba kao tijelo reducirane mase

$$\vec{r_1}' = \frac{\mu}{m_1}\vec{r} \approx \frac{m_1}{m_1}\vec{r} \approx \vec{r}, \quad \vec{r_2}' = -\frac{\mu}{m_2}\vec{r} \approx -\frac{m_1}{m_2}\vec{r} \approx 0, \tag{2.35}$$

što će biti slučaj za sustave planet - Sunce u Sunčevom sustavu, koji su nam u školskoj nastavi najzanimljiviji.

2.5 *Keplerovi zakoni i vis-viva jednadžba*

Da bi naša simulacija bila što informativnija i korisnija u školskoj primjeni, osim simulacije orbita želimo postići da prikazuje iznos relativne brzine tijela duž eliptične orbite i demonstrira Keplerove zakone.

Prvi Keplerov zakon - Planeti se gibaju po elipsama u čijem jednom žarištu se nalazi Sunce. [6]

Drugi Keplerov zakon - spojnica planet - Sunce opisuje u jednakim vremenskim intervalima jednake površine. [6]

Treći Keplerov zakon - Kvocijent kvadrata vremena ophoda planeta oko Sunca i treće potencije velike poluosi njegove orbite ima za sve planete jednaku vrijednost.

$$\frac{T^2}{a^3} = k, \tag{2.36}$$

gdje k konstanta.

Iznos relativne brzine dvaju tijela ćemo računati uz pomoć vis-viva jednadžbe, koju ovdje nećemo izvoditi

$$v^2 = \mu \left(\frac{2}{r} - \frac{1}{a} \right), \tag{2.37}$$

gdje je r je udaljenost između tijela, a označava duljinu velike poluosi elipse dok je μ umnožak zbroja masa planeta u orbiti i gravitacijske konstante kojeg zovemo standardnim gravitacijskim parametrom. [3]. Poslijedica je principa očuvanja mehaničke energije.

Pogledajmo omjere iznosa relativnih brzina tijela u dvjema točkama orbite

$$\frac{v_2}{v_1} = \frac{\mu \left(\frac{2}{r_2} - \frac{1}{a} \right)}{\mu \left(\frac{2}{r_1} - \frac{1}{a} \right)} \Longrightarrow v_2 = \frac{\left(\frac{2}{r_2} - \frac{1}{a} \right)}{\left(\frac{2}{r_1} - \frac{1}{a} \right)} v_1. \tag{2.38}$$

Ukoliko nam je na nekom položaju u orbiti iznos relativne brzine tijela poznat, možemo izračunati iznos relativne brzine tijela na svim ostalim položajima u orbiti. U simulaciji ćemo zadati iznos relativne brzine tijela u pericentru, na osnovu koje ćemo odrediti ostale iznose.

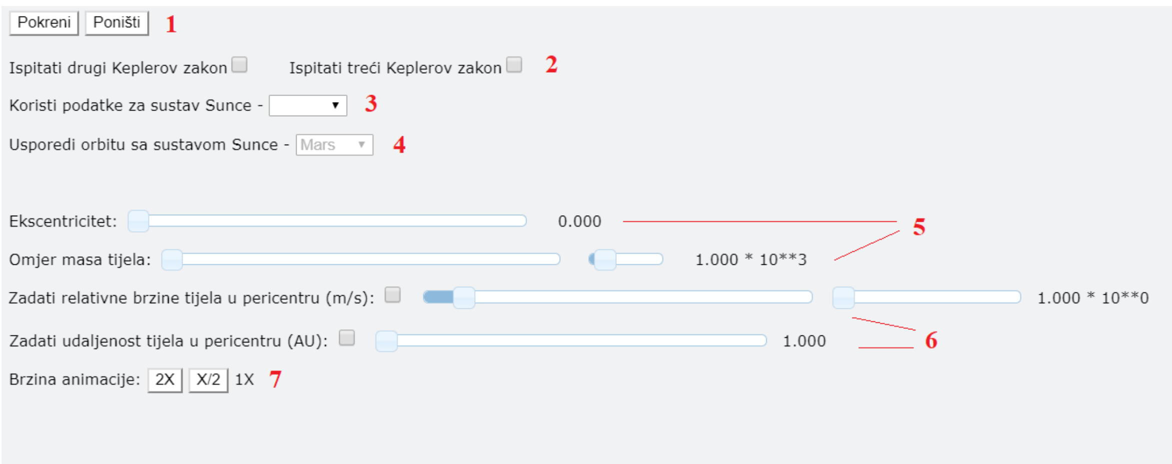
3 Simulacija orbita dvaju tijela

3.1 *Korišteni programski paketi*

Simulacija koju predstavljamo u ovom radu je napravljena u Glowscript IDE razvojnom okruženju, koristeći VPython (verzija 7) i Glowscript (verzija 2.8). VPython je alat koji se koristi za prikazivanje 3D objekata i grafova, čija je glavna primjena u edukativne svrhe. Kombinacija je programskog jezika Python i njegovog modula zvanog Visual. Glowscript omogućuje pisanje i izvođenje VPython koda na internet preglednicima kompilacijom VPython koda u Javascript kod uz pomoć RapydScript-NG kompajlera [9].

Programi napisani u online Glowscript IDE razvojnom okruženju se mogu lako dijeliti i ugrađivati na web stranice što ga čini pogodnim za korištenje u edukativne svrhe, jer pokretanje takvih programa zahtijeva samo pristup internetu i moderniji internet preglednik (Chrome, Firefox, Safari; Internet Explorer nije podržan), bez potrebe za instalacijom bilo kakvih dodatnih programa ili programskih paketa.

3.2 *Korisničko sučelje*



Slika 3.2: Korisničko sučelje simulacije orbita dvaju tijela.

Korisničko sučelje simulacije smo podijelili na 7 zasebnih cjelina (Slika 3.2).

Cjelina 1 - dugme pokreni pokreće/zaustavlja simulaciju. Pokretanje simulacije zaključava cjelinu 2. Dugme poništi briše prethodnu simulaciju, vraća parametre na izvorne vrijednosti i otključava zaključane parametre.

Cjelina 2 - sadrži kućice koje označavamo ukoliko želimo demonstraciju jednog ili oba Keplerova zakona. Biranjem simulacije drugog Keplerovog zakona zaključavamo cjelinu 7. Biranjem trećeg Keplerovog zakona otključavamo cjelinu 4, a zaključavamo cjeline 5 i 6, čije parametre možemo mijenjati padajućim izbornicima cjelina 3 i 4. Prvi zakon je vidljiv iz same simulacije orbita tijela.

Cjelina 3 - padajući izbornik s unaprijed pripremljenim parametrima tijela Sunčevog sustava. Odabirom jednog od tijela zaključavamo parametre cjeline 5 i 6.

Cjelina 4 - inicijalno zaključana, biramo s kojim planetom Sunčevog sustava želimo usporediti orbitu tijela izabranog u cjelini 3.

Cjelina 5 - obavezni parametri: ekscentricitet orbite i omjer masa tijela.

Cjelina 6 - neobavezni parametri: iznosi relativnih brzina tijela i njihove udaljenosti u pericentru.

Cjelina 7 - brzina animacije, dugme 2X dvostruko povećava brzinu animacije, dugme X/2 je dvostruko smanjuje.

3.3 Određivanje položaja i brzine tijela

Za početni položaj tijela u simulaciji biramo onaj u pericentru, s koordinatnim sustavom orijentiranim na način da se pericentar orbite tijela reducirane mase nalazi na pozitivnoj x osi ($\theta_0 = 0$). Udaljenost dvaju tijela je početno zadana, za čije vektore položaja iz jednadžbi (2.31) i (2.32) zaključujemo da se također moraju nalaziti na x osi. Vektori položaja tijela su suprotnih orijentacija te udaljenost tijela predstavlja položaj tijela reducirane mase na x osi (2.33). Jednom kada odredimo položaj reduciranog tijela na x osi, iz jednažbe (2.25) možemo izračunati parametar orbite

$$p = r(0)(1 + e). \tag{3.39}$$

Poznavajući parametar orbite, možemo odrediti položaj tijela reducirane mase za bilo koju vrijednost θ, iz kojeg uz pomoć jednadžbi (2.31) i (2.32) računamo položaje pojedinih tijela.

Ukoliko je iznos početne relativne brzine dvaju tijela u pericentru zadan, iz jednadžbe (2.38) možemo izračunati iznose relativnih brzina na bilo kojem položaju u orbiti. Kutna količina gibanja je konstanta gibanja čiji iznos odgovara jednadžbi (2.19). Kako se tijela gibaju u xy ravnini, vektor kutne brzine je orijentiran duž z osi i njegova z komponenta dana s jednadžbom (2.20). U ostatku rada ćemo se pod kutnom brzinom referirati na komponentu vektora kutne brzine duž z osi.

Pošto nas za potrebe simulacije više zanima odnos kutnih brzina na raznim položajima u orbiti nego same brzine, promatramo omjere kutnih brzina tijela reducirane mase u dvjema točkama orbite

$$\frac{\dot{\theta}_2}{\dot{\theta}_1} = \frac{\frac{L}{\mu r_2^2}}{\frac{L}{\mu r_1^2}} = \frac{{r_1}^2}{{r_2}^2} \Longrightarrow \dot{\theta}_2 = \frac{{r_1}^2}{{r_2}^2}\dot{\theta}_1. \tag{3.40}$$

Kutne brzine pojedinih tijela su ekvivalentne onoj tijela reducirane mase. U našoj simulaciji $\dot{\theta}_1$ predstavlja početno zadanu kutnu brzinu u pericentru, a $\dot{\theta}_2$ kutnu brzinu na promatranom položaju u orbiti. Animacije se u VPythonu izvode pomoću programske petlje: pri svakom prolasku kroz petlju animacije crtamo jedan segment orbite na ekranu. Kut kojeg će vektor položaja tijela reducirane mase činiti s pozitivnom x osi pri sljedećem prolazu kroz programsku petlju je

$$\theta_{n+1} = \theta_n + \dot{\theta}_n \Delta t. \tag{3.41}$$

Vrijednost kuta θ_n pri prvom ulasku u petlju iznosi 0 rad. Kutnu brzinu u simulaciji zadajemo kao prebrisan kut po vremenskom intervalu izvođenja petlje animacije programa, dok će Δt predstavljati vremenski interval potreban za izvršenje petlje animacije i za čije trajanje smatramo da je kutna brzina konstantna. Postavljajući veličine na ovaj način, pri svakom prolazu kroz petlju vrijednost kuta povećavamo za iznos kutne brzine na promatranom položaju. Nakon što uz pomoć jednadžbe (3.41) izračunamo novu vrijednost kuta, iz jednadžbe (2.25) dobivamo novi položaj tijela reducirane mase te s jednadžbom (3.40) određujemo $\dot{\theta}_{n+1}$, kutnu brzinu tijela na novom položaju te ponavljamo postupak. Kako se zasebna tijela gibaju istom kutnom brzinom kao i tijelo reducirane mase, ovakvim pristupom možemo vjerno prikazati kutnu brzinu oba tijela naspram one u pericentru na svakom položaju u orbiti.

Svakim prolaskom kroz petlju animacije određeni segment orbite aproksimiramo ravnom linijom. Što su segmenti koje aproksimiramo ravnim linijama manji, aproksimacija će biti preciznija. Da bismo to postigli, trebamo postići što finiju podjelu intervala. Finoća podjele intervala ovisi o povećanju kuta pri svakom koraku, a povećanje ovisi o umnošku kutne brzine i vremenskog intervala (3.41). Za određeni vremenski korak možemo postići manja povećanja kuta pri prolascima kroz petlju biranjem manje kutne brzine u pericentru.

Na slici 3.3 prikazane su simulacije orbita ekscentriteta 0 za vrijednosti početne kutne brzine od približno 0.01745 i 0.349 rad po vremenskom intervalu izvođenja petlje animacije. U VPythonu, broj izvođenja programske petlje animacije u sekundi kontroliramo argumentom naredbe $rate()$, koji u simulaciji postavljamo na 15. Stoga, očekujemo maksimalno 15 izvođenja petlje po sekundi, tj. vremenski interval potreban za jedan prolaz kroz petlju je približno 0.067 s. Za početnu kutnu brzinu od približno 0.349 rad po vremenu izvođenja petlje animacije, početna kutna brzina izražena u rad/s će biti

$$\dot{\theta}_1 \approx \frac{0.349\,\mathrm{rad}}{\frac{1}{15}\,\mathrm{s}} \approx 5.24\,\mathrm{rad/s}. \tag{3.42}$$

Orbite ekscentriciteta 0 predstavljaju kružne orbite čiji je radijus konstantan te iz jednadžbe (3.40) zaključujemo da za svaku točku orbite vrijedi

$$\dot{\theta}_2 = \frac{{r_1}^2}{{r_2}^2}\dot{\theta}_1 = \frac{{r_1}^2}{{r_1}^2}\dot{\theta}_1 = \dot{\theta}_1. \tag{3.43}$$

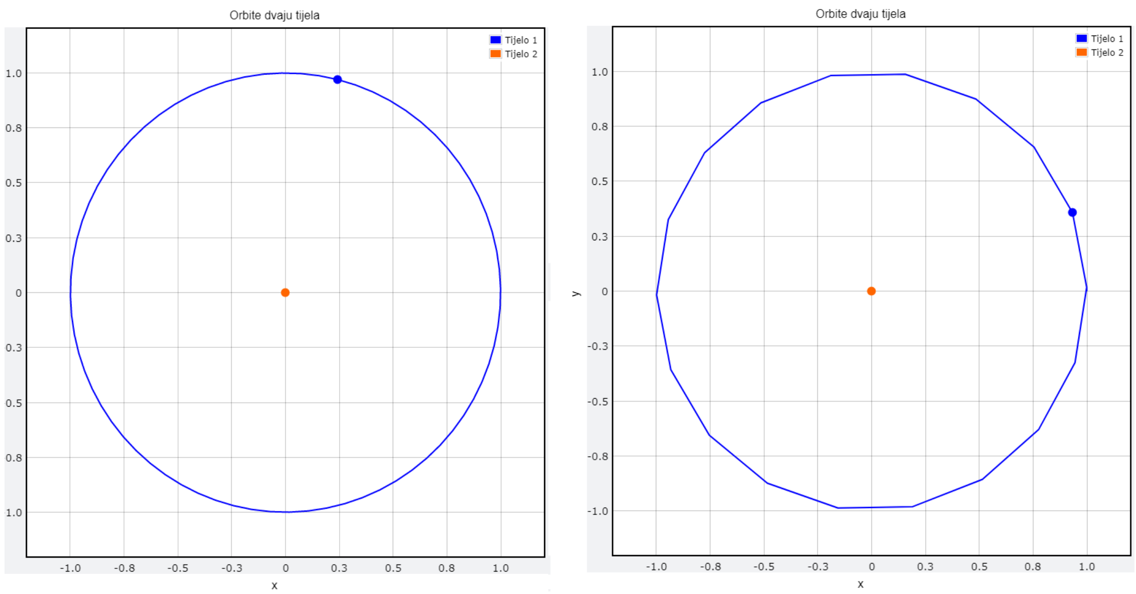
Nema promjene u kutnoj brzini, ista je stalna na svim položajima u orbiti. Vremenski korak jednadžbe (3.41) postavljamo kao vremenski interval izvođenja petlje animacije, te se pri svakom prolazu kut povećava za 0.349 rad. Da bi zatvorili putanju, potrebno je $n = \frac{2\pi}{0.349} = 18$ prolaza kroz petlju, tj. orbitu aproksimiramo s 18 ravnih linija i ta aproksimacija vidno odstupa od kružne orbite. Za početnu brzinu od približno 0.01745 rad po vremenskom intervalu izvođenja petlje animacije postižemo 20 puta finiju podjelu intervala, te takva aproksimacija daje mnogo bolje rezultate (Slika 3.3). U našoj simulaciji izabrali smo početnu kutnu brzinu od približno 0.0349 rad po vremenskom intervalu izvođenja petlje animacije. U tom slučaju, kutna brzina u pericentru iznosi

$$\dot{\theta}_1 \approx \frac{0.0349\,\mathrm{rad}}{\frac{1}{15}\,\mathrm{s}} \approx 0.524\,\mathrm{rad/s}. \tag{3.44}$$

Pri simulaciji orbita ekscentriciteta 0, očekujemo period od približno

$$T = \frac{2\pi}{0.524\,\frac{\mathrm{rad}}{\mathrm{s}}} \approx 12\,\mathrm{s}. \tag{3.45}$$

Period od 12 sekundi je teoretski najkraći mogući period, od kojeg pri izvođenju simulacije ne očekujemo velika odstupanja. Generalno, sve orbite istog ekscentrici-



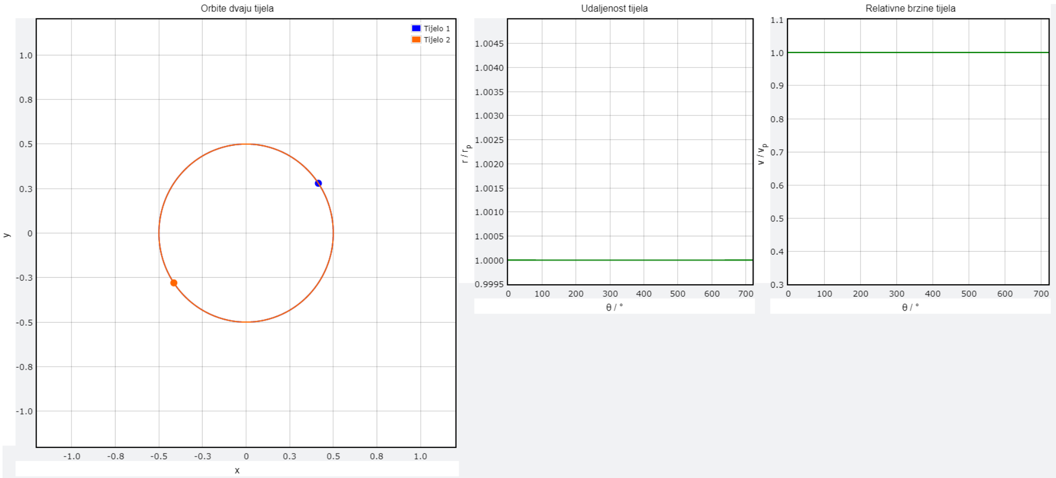
Slika 3.3: Simulacija orbite ekscentriciteta 0 s početnim kutnim brzinama od 0.01745 (lijevo) i 0.349 rad (desno) po vremenskom intervalu izvođenja petlje animacije.

teta će se zatvoriti u istom vremenu, neovisno o početnom radijusu. Orbite približno istih ekscentriciteta, što je slučaj za većinu planeta Sunčevog sustava, će se zatvoriti u približno jednakom vremenu. Što je promatrana orbita većeg ekscentriciteta, razlike iznosa vektora položaja su sve izraženije, kutne brzine duž orbite naspram one pericentru sve manje te se kut po jednadžbi (3.41) sporije povećava te posljedično aproksimiramo veći broj segmenata orbite. Da bismo postigli da se orbite velikih ekscentriciteta zatvore dovoljno brzo, u cjelini 7 korisničkog sučelja postavljena su dva dugmeta kojima reguliramo brzinu animacije.

3.4 Simulacija orbite bez zadanih neobaveznih parametara

Simulacija je zamišljena sa samo dva obvezna parametra - ekscentricitetom i omjerom masa tijela u orbiti, grupiranih u cjelinu 5 korisničkog sučelja (Slika 3.2), gdje se omjer odnosi na tijelo veće mase naspram tijela manje mase. Neobavezni parametri, grupirani u cjelinu 6, označavaju iznos relativne brzine i udaljenost dvaju tijela u pericentru.

Ukoliko ne postavimo vrijednosti neobaveznih parametara, simulacija pokazuje kako se vrijednosti parametara u orbiti odnose naspram početnih vrijednosti zadanih u pericentru. Primjer simulacije sustava tijela jednakih masa koja se gibaju u orbiti ekscentriciteta 0 je prikazan na slici 3.4.

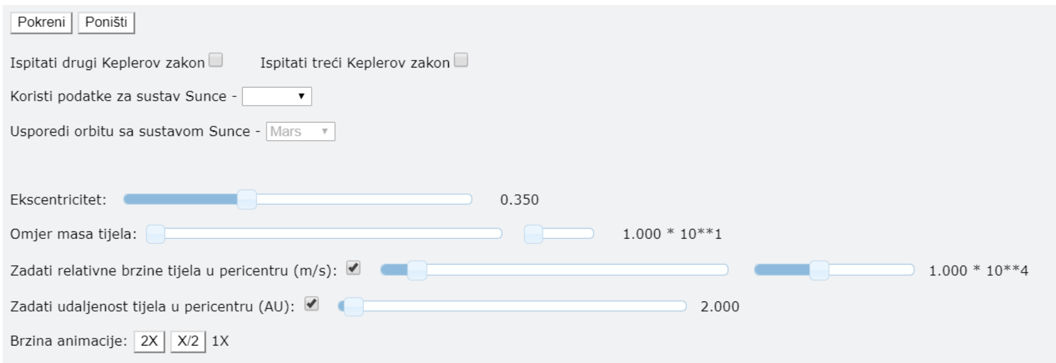


Slika 3.4: Simulacija bez zadanih neobaveznih parametara ($e = 0$, $\frac{m1}{m2} = 1$).

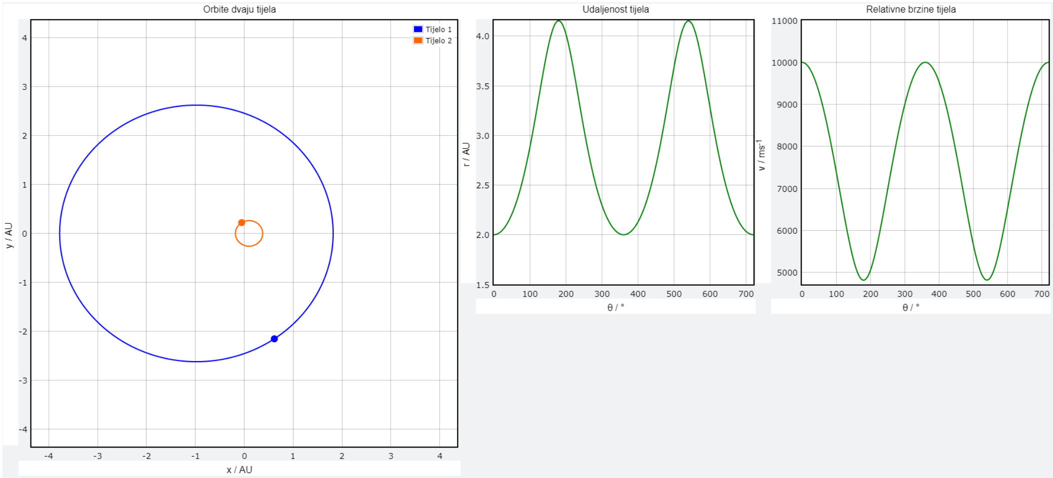
Primjećujemo da se tijela gibaju u kružnim orbitama jednakih radijusa. Iznos relativne brzine tijela i njihova udaljenost se ne mijenjaju naspram početno zadanih vrijednosti.

3.5 *Simulacija orbite sa zadanim neobaveznim parametrima*

Ukoliko zadamo neobavezne parametre, izražene u m/s i AU (astronomska jedinica), možemo odrediti apsolutni iznos udaljenosti i relativne brzine u svim točkama putanje umjesto iznosa u odnosu na početnu točku. Parametre možemo unijeti proizvoljno ili odabrati automatski unos vrijednosti za jedno od tijela u padajućem izborniku cjeline 3 korisničkog sučelja, koje će formirati sustav sa Suncem. Na slici 3.6 su prikazane izlazne vrijednosti simulacije sustava dvaju tijela gdje smo početne vrijednosti proizvoljno zadali (Slika 3.5).

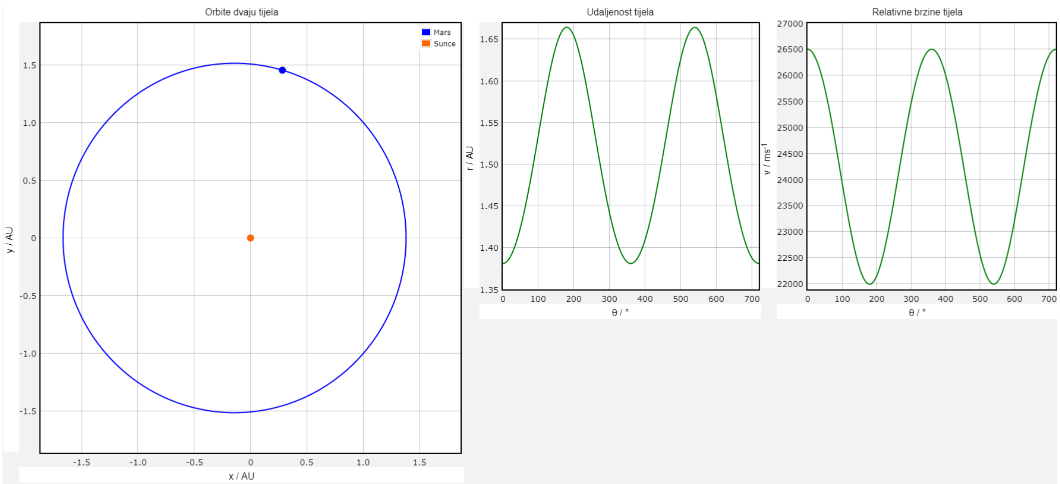


Slika 3.5: Ulazne vrijednosti simulacije s proizvoljno zadanim neobveznim parametrima.



Slika 3.6: Simulacija s proizvoljno zadanim neobaveznim parametrima (Slika 3.5).

Primjećujemo da orbite tijela, zbog većeg ekscentriciteta više nisu kružne već poprimaju eliptičan oblik. Iznos relativne brzine i udaljenost tijela se mijenjaju duž orbite, udaljenost je najveća a iznos relativne brzine tijela najmanji za vrijednost kuta π, gdje se nalazi apocentar. Postavljanjem pokazivača na bilo koju točku grafa možemo iščitati njene koordinate s preciznošću na najmanje drugu decimalu. Na slici 3.7 prikazane su izlazne vrijednosti simulacije sustava Sunce - Mars, odabranog u odgovarajućem padajućem izborniku.

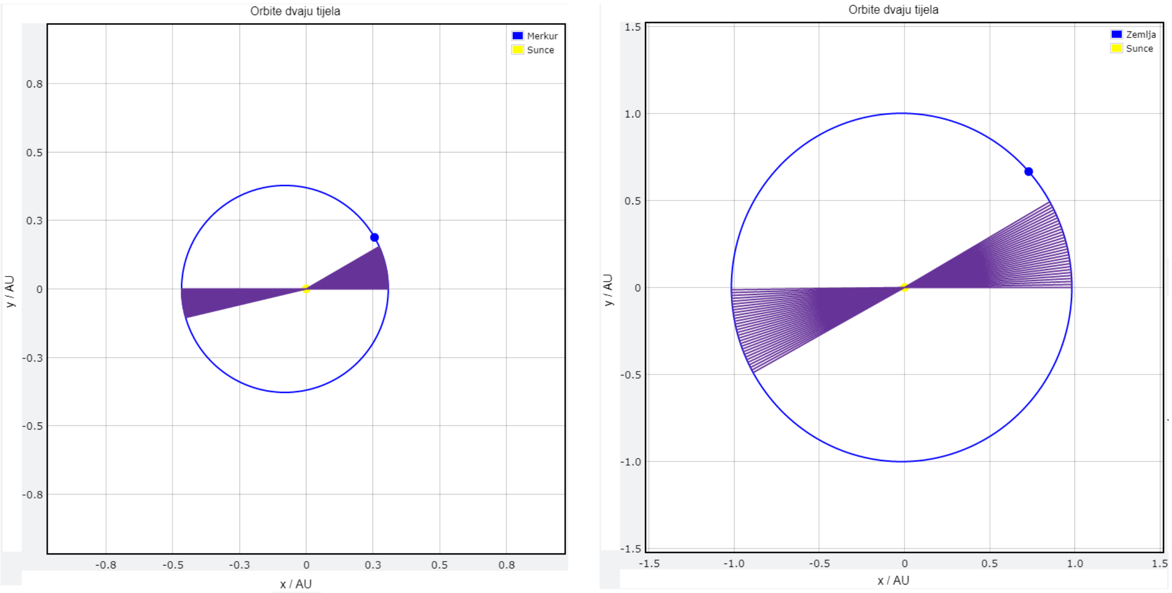


Slika 3.7: Simulacija sustava Sunce - Mars.

Mars se oko Sunca giba brzinom koja varira između 26.5 km/s i 22 km/s, jer Sunce približno miruje u ishodištu, stoga iznos relativne brzine između tijela odgovara iznosu brzine Marsa. Udaljenost između tijela varira između početno postavljene udaljenosti od 1.381 AU do 1.66 AU, što se slaže s opaženim vrijednostima.

3.6 *Simulacija Keplerovih zakona*

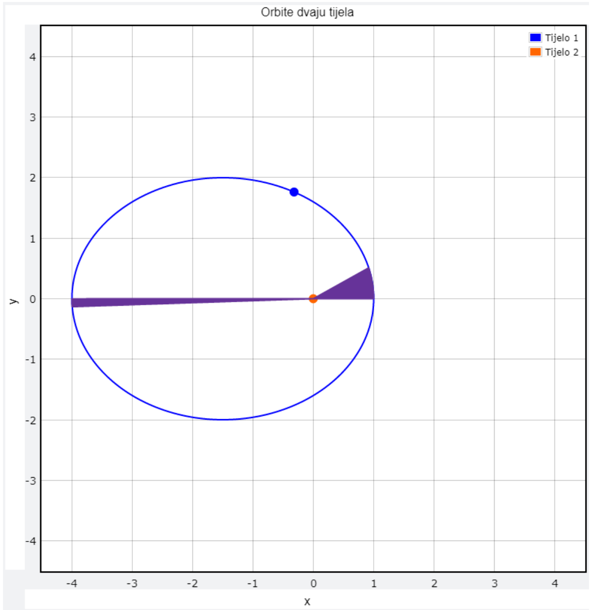
Ukoliko želimo demonstrirati drugi ili treći Keplerov zakon, prije pokretanja simulacije moramo označiti predviđenu kućicu u drugoj cjelini korisničkog sučelja. Na slici 3.8 je prikazana simulacija drugog Keplerovog zakona na primjeru Merkurove orbite (eliptična, značajnijeg ekscentriciteta) i Zemljine orbite (približno kružna) oko Sunca. Programsku izvedbu zakona temeljimo na odbrojavanju 30 ciklusa petlje animacije jednom kada vrijednost kuta kojeg vektor položaja tijela reducirane mase čini s pozitivnom x osi poprimi vrijednosti 0 i π rad. Da bi se simulacija ispravno izvodila, za njeno trajanje zaključavamo dugmad za mijenjanje brzine animacije. Drugi Keplerov zakon možemo demonstrirati i na proizvoljno zadanim orbitama (Slika 3.9).



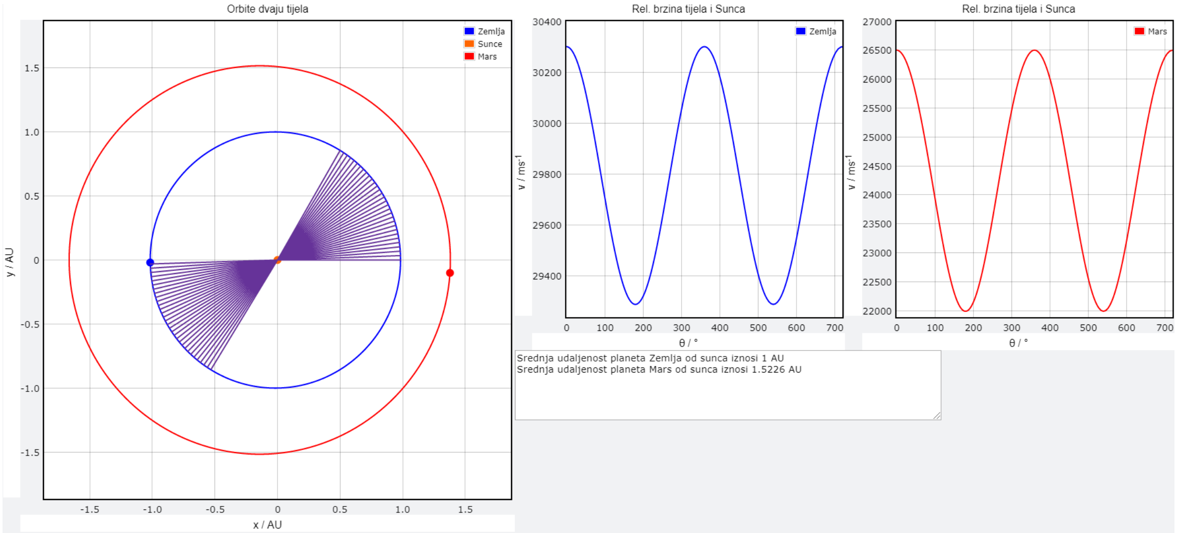
Slika 3.8: Demonstracija drugog Keplerovog zakona na primjeru orbita Merkura i Zemlje oko Sunca.

Treći Keplerov zakon demonstriramo na način da istodobno, na istom grafu, simuliramo gibanja dva tijela oko Sunca. Jednom kada izaberemo simulaciju zakona u korisničkom sučelju, padajućim izbornicima cjelina 3 i 4 biramo tijela Sunčevog sustava čije orbite želimo usporediti. Za razliku od prethodnog zakona, u ovom slučaju ne dozvoljavamo unos proizvoljnih orbita već zakon demonstriramo isključivo na tijelima Sunčevog sustava, koja su nam najzanimljivija. Ukoliko izaberemo demonstraciju oba zakona u trećoj cjelini korisničkog sučelja, drugi Keplerov zakon će se demonstrirati na orbiti manjeg radijusa (Slika 3.10).

Da bi simulacija trećeg Keplerovog zakona bila uspješna, ne možemo se više os-



Slika 3.9: Demonstracija drugog Keplerovog zakona ($e = 0.6$, $\frac{m1}{m2} = \frac{1}{1000}$).



Slika 3.10: Demonstracija drugog i trećeg Keplerovog zakona na orbitama Zemlje i Marsa oko Sunca.

lanjati na prethodno rješenje gdje je početna kutna brzina tijela bila fiksna neovisno o udaljenostima tijela od ishodišta. U perihelu je vektor brzine okomit na vektor položaja, tj. nema komponentu brzine u $\hat{r}$ smjeru te vrijedi

$$v = v_\theta = r\omega \Longrightarrow \omega = \frac{v}{r}. \tag{3.46}$$

Kutna brzina Zemlje u perihelu iznosi

$$\omega_{zem} = \frac{v_{rel}}{r} = \frac{30300\frac{\mathrm{m}}{\mathrm{s}}}{0.9833\mathrm{AU}} \approx 30814.6\frac{\mathrm{m}}{\mathrm{s}\cdot\mathrm{AU}}. \tag{3.47}$$

Sunce približno miruje u ishodištu te je iznos relativne brzine planeta i Sunca ujedno i iznos brzine planeta. Same dimenzije kutnih brzina nam nisu bitne, već omjer kutnih brzina planeta u perihelu. Ukoliko želimo skalirati vrijednost kutne brzine Zemlje u perihelu na bezdimenzionalnu veličinu približne vrijednosti 1, moramo izraz (3.47) podijeliti s $\approx 30814\frac{\mathrm{m}}{\mathrm{s{\cdot}AU}}$. Tim faktorom ćemo skalirati isti izraz i za sva ostala tijela, što će nam dati ispravne omjere kutnih brzina. Zahvaljujući tome, u simulaciji smo slobodni izabrati dimenzije i veličinu kutne brzine Zemlje u perihelu, na osnovu koje ćemo skalirati brzine svih drugih tijela.

4 Metodički dio

4.1 Simulacije u interaktivnoj istraživački usmjerenoj nastavi fizike

Klasična predavačka nastava, koja postavlja učitelja u središte, a učenika u pasivnog primaoca informacija se nije pokazala efikasnom za razvoj prirodoznanstvene pismenosti. Istraživanja su pokazala da su učinci predavačke nastave gotovo neovisni o predavaču. Iz tog razloga, u edukacijskoj fizici se okrećemo istraživački usmjerenoj nastavi fizike gdje je učenik aktivan sudionik.

Istraživački usmjerena nastava fizike bazira se na vođenom istraživanju, gdje učenik sam dolazi do spoznaja uz vođenje od strane nastavnika. Ovaj tip nastave nastoji postupno izgraditi razumijevanja fizikalnih sadržaja i istraživačke vještine te učenicima pruža direktno iskustvo s istraživačkim karakterom fizike kao znanstvene discipline uz dovoljno prostora za samostalno razmišljanje i zaključivanje, osmišljavanje pokusa, testiranje hipoteza itd. [8]

Sat istraživački usmjerene nastave fizike čine uvodni, središnji i završni dio sata. Uvodnim dijelom sata kroz uvodno pitanje ili problem uvodimo učenike u tematiku i nastojimo je povezati sa svakodnevnim životom [8]. Demonstracija pojave je naročito važan dio uvodnog sata, uz pomoć koje se učenici upoznaju s pojavom koju proučavaju prije njenog matematičkog opisa. Uvodni dio sata završavamo uvođenjem naziva nove pojave.

Središnji dio sata započinjemo postavljanjem istraživačkog pitanja, koje daje usmjerenje cijelom satu. Dobro postavljenim istraživačkim pitanjem učenici će imati jasnu smjernicu što istražuju i na koje pitanje svojim istraživanjem daju odgovor. Učenici iznose svoje hipoteze te predlažu pokuse kojima bi ispitali istraživačko pitanje. Iako mogu biti demonstracijski, poželjno je da učenici izvode pokuse u manjim grupama kada god to mogućnosti dozvoljavaju te uz crtanje skica samostalno iskažu i zapišu opažanja i zaključke. Na kraju središnjeg dijela sata konstruiramo model i matematički opis promatrane fizikalne pojave.

Završni dio sata omogućuje učenicima da se uvjere u značenje i primjenu konstruiranog modela, a nastavniku da procjeni razumijevanje novog gradiva. U ovom dijelu sata se možemo vratiti na uvodni problem, na koji učenici pokušavaju dati odgovor koristeći novi model.

Simulacije spadaju u interaktivne nastavne metode koje se zasnivaju na korištenju računala i pogodne su u slučajevima gdje su pokusi opasni, preskupi ili vremenski ili eksperimentalno prezatjevni. Iako korisne, učinkovitost takvih metoda ako se koriste same za sebe nije visoka, već ih je potrebno kombinirati sa drugim nastavnim metodama koje uključuju verbalizaciju ideja, poput konceptualnih pitanja, rada u skupinama, razrednom raspravom i sl.

4.2 Nastavna priprema

U ovom poglavlju ćemo prikazati pripremu za sat interaktivne istraživački usmjerene nastave fizike u prvom razredu gimnazije na temu "Keplerovi zakoni", koja spada u nastavnu cjelinu gravitacijske sile i predstavlja izborni sadržaj predviđen za nastavni sat broj 61.

NASTAVNA PRIPREMA IZ FIZIKE

ŠKOLA: Srednja škola, gimnazija
RAZRED: 1.
NASTAVNA CJELINA: Gravitacijska sila
NASTAVNA JEDINICA: Keplerovi zakoni
PREDVIĐENI BROJ SATI: 1

OBRAZOVNI ISHODI (OČEKIVANA UČENIČKA POSTIGNUĆA)

- iskazati Keplerove zakone
- objasniti gibanje planeta pomoću Keplerovih zakona
- primjeniti Keplerove zakone u konkretnim problemskim zadatcima

ODGOJNI ISHODI

- osposobiti učenike za logično mišljenje, zapažanje, grafičko izražavanje
- isticati važnosti fizike u svakidašnjem životu
- izražavanje vlastitog i uvažavanje tuđeg mišljenja

- poticati aktivnosti učenika

<u>FUNKCIONALNI ISHODI</u>

- poticanje sistematičnosti
- poticanje istraživačke radoznalosti

VRSTA NASTAVE: INTERAKTIVNA ISTRAŽIVAČKI USMJERENA NASTAVA

NASTAVNE METODE

1. Demonstracija pokusa
2. Metoda razgovora - usmjerena rasprava
3. Kooperativno rješavanje zadataka u skupinama
4. Metoda pisanja/crtanja

OBLICI RADA

1. Frontalni
2. Individualni
3. Rad u skupinama

KORELACIJA S DRUGIM PREDMETIMA
Povijest, matematika, geografija

NASTAVNA POMAGALA I SREDSTVA
Računalna simulacija, kreda, ploča, računalo i projektor

TIJEK NASTAVNOG SATA

1. **Uvodni dio**

   Nastavnik se kratko referira na gradivo iz prošlog sata. Spominje da se sve do 16 st. vjerovalo u Ptolomejev geocentrični planetarni model. U 16 st. Nikola Kopernik uvodi ideju heliocentričnog sustava, no nije mogao uskladiti svoju teoriju s rezultatima opažanja. Njemački astronom Johannes Kepler prihvatio je Kopernikov heliocentrični planetni model te uz pomoć vrlo preciznih astronomskih podataka o pozicijama planeta njegovog učitelja, danskog astronoma Tychea Brachea, formulirao tri zakona planetarnog gibanja.

Nastavnik uvodi naslov - **Keplerovi zakoni**.

2. **Središnji dio sata**

   ISTRAŽIVAČKO PITANJE: Gdje se nalazi Sunce u odnosu na orbitu tijela koje kruži oko njega?

   Učenici zapisuju pretpostavku te je dijele s razredom. Očekujemo pretpostavku da se Sunce nalazi u središtu kružne putanje planeta.

   Pokrenite simulaciju Zemljine orbite oko Sunca. Što opažate?

   Učenici u grupama po dvoje pokreću animaciju orbite Zemlje oko Sunca, zapisuju svoja opažanja i dijele ih s razredom. Ukoliko za to ne postoje uvjeti, nastavnik izvodi simulacije na projektoru. Učenici opažaju da se Sunce doista nalazi u središtu kružne putanje Zemlje.

   Pokrenite simulaciju Merkurove orbite oko Sunca. Što opažate?

   Učenici pokreću simulaciju, opažaju pomak Sunca od središta orbite te zapisuju zaključak istraživačkog pitanja i dijele ga s razredom - Sunce nije nužno u središtu orbite.

   ISTRAŽIVAČKO PITANJE: Kakvog su oblika putanje nebeskih tijela oko Sunca?

   Učenici zapisuju pretpostavke za istraživačko pitanje. S obzirom na prethodno naučeno gradivo i činjenicu da orbita Merkura, unatoč tome što Sunce nije u središtu, u simulaciji djeluje kružna, očekujemo pretpostavku kružnih putanja. Nakon izlaganja pretpostavki, nastavnik traži od učenika da pokrenu simulaciju orbite Halleyjevog kometa oko Sunca i zapišu svoja opažanja. Učenici primjećuju putanju koja je jako izdužena u odnosu na kružnicu. Očekujemo da će neki učenici prepoznati eliptične putanje s nastave geografije. Nastavnik naglašava da je kružnica samo specijalan slučaj elipse. Učenici zapisuju i izlažu zaključak istraživačkog pitanja - putanje nebeskih tijela su eliptične.

Nastavnik uvodi prvi Keplerov zakon koji kaže da se planeti gibaju po elipsama, izduženim kružnicama, sa Suncem u jednom od žarišta. Zapisuje zakon, crta skicu na ploču te definira žarišta kao točke čiji je zbroj udaljenosti od planeta u svakoj točki orbite konstantan. Što je elipsa manje izdužena, žarišta su bliže središtu, dok se za savršenu kružnicu žarišta poklapaju - njeno središte je jednako udaljeno od planeta za svaku točku orbite.

Pogledajte vrijednosti ekscentriciteta u simulaciji za orbite Zemlje, Merkura i Halleyjevog kometa. Što zaključujete?

Učenici promatraju vrijednosti ekscentriciteta i zaključuju da eliptičnost orbite ovisi o ekscentricitetu. Što je ekscentricitet orbite veći, orbita je eliptičnija i obratno. Nastavnik naglašava da je orbita savršeno kružna samo za vrijednost ekscentriciteta 0.

ISTRAŽIVAČKO PITANJE: Kako brzina tijela ovisi o udaljenosti od Sunca?

Učenici zapisuju pretpostavke i dijele ih sa razredom.

Pokrenite simulaciju Marsa oko Sunca i obratite pozornost na graf relativne brzine planeta i Sunca i udaljenosti planeta od Sunca. Što nam oni govore?

Učenici pokreću simulaciju te iz grafova iščitavaju da brzina varira - veća je na položajima u orbiti gdje je planet bliži Suncu te se smanjuje sa povećanjem udaljenosti. Zapisuju opažanja i zaključak istraživačkog pitanja koja iznose ostatku razreda.

Uključite demonstraciju drugog Keplerovog zakona. Što opažate? Što mislite, kakav je odnos tih površina? Zašto?

Učenici opažaju dvije površine koju je prebrisala spojnica planet - Sunce u jednakim vremenskim intervalima, te procjenjuju da su površine jednake unatoč različitoj duljini spojnice, zbog nejednake brzine tijela duž orbite.

Nastavnik uvodi drugi Keplerov zakon, zapisuje ga na ploču i crta sliku. Definira ga kao činjenicu da spojnica Sunca i planeta u jednakim vremenskim intervalima prebriše jednake površine.

ISTRAŽIVAČKO PITANJE: Istražite odnos ophodnih vremena i udaljenosti planeta oko Sunca.

Učenici zapisuju pretpostavke i dijele ih s razredom. Nastavnik nalaže učenicima da u simulaciji označe ispitivanje trećeg Keplerovog zakona za orbite Zemlje i Venere oko Sunca. Učenici zapisuju opažanja i zaključak i dijele ih s ostatkom razreda - što je udaljenost planeta od Sunca veća, ophodno vrijeme je također veće.

Nastavnik uvodi treći Keplerov zakon - kvocijent kvadrata vremena ophoda planeta oko Sunca i treće potencije srednje udaljenosti planeta od Sunca ima za sve planete jednaku vrijednost. Zapisuje definiciju i matematičku formulaciju trećeg Keplerovog zakona na ploču. Naglašava da je Kepler opisao gibanje planeta, ali nije znao uzrok tom gibanju, kojeg je opisao Newton [7].

3. **Završni dio sata**

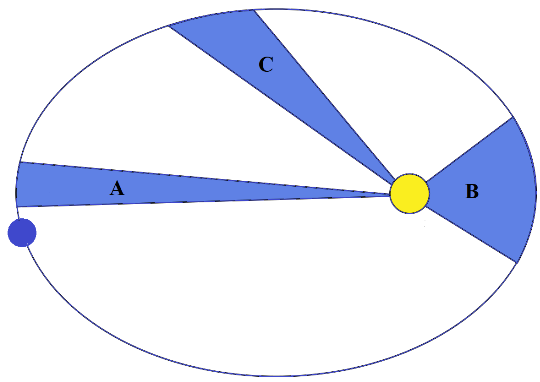
Zadatak 1: Koja od prebrisanih površina (u istom vremenskom intervalu) odgovara najvećoj brzini planeta?

(a) A

(b) B

(c) C

(d) Sve, brzina je svugdje ista



Slika 4.11: Zadatak 1

Točan odgovor je pod B: da bi spojnica planet - Sunce prebrisala jednake površine u jednakim vremenskim intervalima, brzina mora biti najveća u području najbližem Suncu.

Zadatak 2: Kada se udaljenost između fokusa poveća, što se dogodi sa oblikom elipse?

(a) postane zaokruženija

(b) postane izduženija

(c) ostane ista

(d) ništa od navedenog

Točan odgovor je pod B: Što je elipsa manje izdužena žarišta su bliže središtu i obratno.

Zadatak 3: Odredite konstante trećeg Keplerovog zakona za Zemlju i Veneru u simulaciji te ih usporedite.

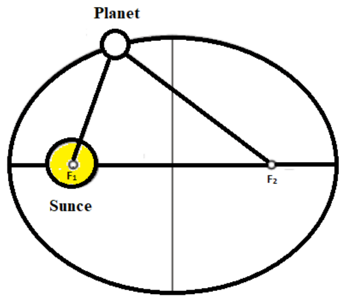
Kako bi precizno odredili vrijeme potrebno da planet jednom okruži Sunce?

Učenici na temelju prethodnih pokusa zaključuju da bi izmjerili vrijeme 10 perioda te izračunali srednju vrijednost. Simulacija računa i ispisuje srednju udaljenost planeta od Sunca, te računanjem omjera vremena ophoda i srednje udaljenosti te uspoređujući rezultate učenici zaključuju da su konstante približno jednake, zbog neminovne pogreške pri mjerenju.

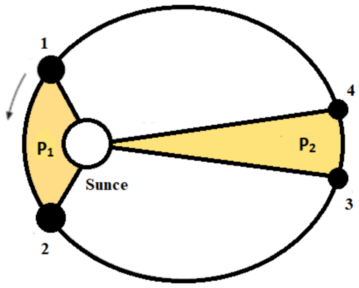
PLAN PLOČE

KEPLEROVI ZAKONI

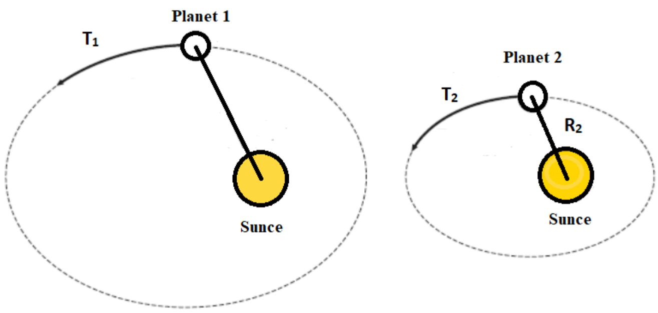
1. **ZAKON:**
   Planeti se oko Sunca gibaju po elipsama. Sunce je smješteno u jednom od žarišta elipse



2. **ZAKON:**
   Spojnica Sunca i planeta u jednakim vremenskim razmacima prebriše jednake površine



3. **ZAKON:**
   Kvocijent kvadrata vremena ophoda planeta oko Sunca i treće potencije srednje udaljenosti planeta od Sunca ima za sve planete jednaku vrijednost



$$\frac{T^2}{R^3} = \text{konst}$$

Slika 4.12: Plan ploče [7].

5 Zaključak

U nastavi fizike, umjesto klasične predavačke nastave, sve se više okrećemo interaktivnoj istraživački usmjerenoj nastavi fizike. U njene interaktivne metode spadaju simulacije koje primjenjujemo za izvođenje pokusa koji iz bilo kojeg razloga nisu izvedivi u učionici. Primjer takvog pokusa je promatranje gibanja planeta oko Sunca, koje bi iziskivalo preveliki financijski i vremenski trošak. Da bi simulacija bila dostupna što većem broju korisnika, neovisno o korištenoj platformi, za njenu izradu smo odabrali kombinaciju VPythona i Glowscripta što omogućava izvođenje simulacije u internet pregledniku bez potrebe za instalacijom dodatnih paketa ili programa. Kako problem dvaju tijela koja međudjeluju centralnim silama možemo svesti na problem jednog tijela koje se giba u centralnom polju, simulaciju orbita dvaju tijela temeljimo na analitičkom rješenju problema jednog tijela uz pomoć kojeg određujemo orbite polaznih tijela. Uz pomoć vis-viva jednadžbe grafički prikazujemo relativne brzine tijela, dok brzinu izvođenja animacije kontroliramo pravilnim omjerima kutnih brzina naspram početno zadane vrijednosti na svakom položaju u orbiti. Za problem više tijela ne postoje analitička rješenja te ukoliko želimo simulirati takav sustav moramo se okrenuti numeričkim metodama. Simulacija, iako važna alternativa pokusu, ne može ga u potpunosti nadomjestiti. Učinkovitost simulacija kada se koriste same za sebe nije visoka, već ih je potrebno kombinirati s drugim nastavnim metodama koje uključuju verbalizaciju ideja, poput konceptualnih pitanja, rada u skupinama, razrednom raspravom i sl.

Dodaci

Dodatak A Programska izvedba simulacije

```
1 ### Diplomski rad: Simulacije putanja planeta pomoću programskog jezika Python
2 ### Mentor: prof.dr.sc. Tamara Nikšić
3 ### Student: Roko Mimica
4 GlowScript 2.8 VPython
5 from vpython import *
6 running = False #Varijabla koja kontrolira da li se animacija izvodi ili je pauzirana
7 #Linije koda 8 - 53: inicijalizacija korištenih varijabli na početne vrijednosti
8 AU = 149597871000
9 brz_anim = 1
10 faktor_skaliranja = 30814
11 e_poc = 0 #Linije koda 11 - 16 označavaju početne vrijednosti slidera, na koje ih vraćamo nakon poništavanja animacije.
12 masa_baza_poc = 1
13 masa_eksp_poc = 3
14 v_baza_poc = 1
15 v_eksp_poc = 0
16 r_poc = 1
17
18 t1_ime = "Tijelo 1"
19 t2_ime = "Tijelo 2"
20 t3_ime = "Mars"
21
22 e2 = 0.093 #Linije koda 22 - 27: vrijednosti za inicijalni izbor padajućeg izbornika cjeline 4 (Mars)
23 masa_baza2 = 3.112
24 masa_eksp2 = 7
25 v_baza2 = 2.65
```

```
26 v_eksp2 = 4
27 r2 = 1.381
28 
29 r = 1 #Linija koda 29 - 34: Početne vrijednosti slidera. Vrijednosti ovih parametara se mijenjaju stoga nam trebaju vrijednosti na linijama 11 - 16
30 masa_baza1 = 1
31 masa_eksp1 = 3
32 v_baza1 = 1
33 v_eksp1 = 0
34 e = 0
35 
36 polumjer = 0
37 kut3 = 0
38 flag4 = False
39 
40 kut = 0
41 rat = brz_anim * 30
42 
43 #Zastavice za simulaciju drugog keplerovog zakona
44 flag1 = False
45 flag2 = True
46 flag3 = True
47 flag4 = False #Zastavica koja bilježi da li smo koristili predeterminirane vrijednosti za planete u izborniku cjeline 3
48 
49 y_os_v = "v / v<sub>p</sub>"
50 y_os_pol = "y"
51 x_os_pol = "x"
52 y_os_ud = "r / r<sub>p</sub>"
53 
54 #Računa položaj tijela uz pomoć parametra orbite, kuta i ekscentriciteta orbite)
55 def izr_polozaj(r0, kut, e):
56     r = r0/(1 + e * cos(kut))
57     #Konverzija iz polarnog u kartezijev sustav
```

```
    x = r * cos(kut)
    y = r * sin(kut)
    return [x,y]

#Određuje položaje zasebnih tijela iz položaja tijela reducirane mase
def pol_tijela(r,masa, red_masa):
    return red_masa/masa * r

#Programska implementacija jednadžbe (2.38)
def rel_lin_br(v_poc, r1,r2,a):
    brojnik = 2/r2 - 1/a
    nazivnik = 2/r1 - 1/a
    return sqrt(v_poc**2 * brojnik/nazivnik)

#Funkcija pridružena dugmetu pokreni/zaustavi
def go1(b):
    global running, kut
    running = not running
    if running:
        b.text = "Zaustavi"
        kep_3.disabled = True
        kep_2.disabled = True
    else:
        b.text = "Pokreni"

#Funkcija pridružena dugmetu poništi, briše grafove, otključava zaključane
    parametre te postavlja vrijednosti varijabli na početne
def go2(b):
    global running, kut,flag1,flag2, flag3, kut2, t1_ime, t2_ime, flag4,
        y_os_v, y_os_pol, y_os_ud, x_os_pol

    running = False
    kut = 0
    kut2 = 0

```

```
    flag1 = False
    flag2 = True
    flag3 = True
    flag4 = False

    y_os_v = "v / v<sub>p</sub>"
    y_os_pol = "y"
    x_os_pol = "x"
    y_os_ud = "r / r<sub>p</sub>"

    rel_ud.checked = False
    rel_br.checked = False
    meni2.disabled = True
    kep_3.checked= False
    kep_2.checked = False
    kep_3.disabled = False
    kep_2.disabled = False
    animacija_mi.disabled = False
    animacija_pl.disabled = False
    v_baza.disabled = True
    v_eksp.disabled = True
    udaljenost.disabled = True

    otkljucaj()
    meni1.selected = ""
    t1_ime = "Tijelo 1"
    t2_ime = "Tijelo 2"
    ecc.value = e_poc
    masa_baza.value = masa_baza_poc
    masa_eksp.value = masa_eksp_poc
    v_baza.value = v_baza_poc
    v_eksp.value = v_eksp_poc
    udaljenost.value = r_poc
    azuriraj_vrijednosti()
    print_options(clear=True)
```

```
    b1.text = "Pokreni"

    graf_putanje.delete()
    graf_udaljenosti1.delete()
    graf_rel_brzine1.delete()
    graf_rel_brzine2.delete()
    graph1.delete()
    graph2.delete()
    graph5.delete()
    graph7.delete()
    graph9.delete
    graph11.delete()
    graph12.delete()

#Funkcija pridružena kućici zadati relativne brzine
def check_brz(s):
    if s.checked:
        v_baza.disabled = False
        v_eksp.disabled = False
        vr_baza = v_baza.value
        vr_eksp = v_eksp.value
        v_baz(v_baza)
        v_exp(v_eksp)

    else:
        v_baza.disabled = True
        v_eksp.disabled = True
        v_baza.value = 1
        v_eksp.value = 1
        v_baz(v_baza)
        v_exp(v_eksp)

#Funkcija pridružena kućici zadati relativne udaljenosti
def check_ud(s):
    if s.checked:
```

```
        udaljenost.disabled = False
        vr_ud = udaljenost.value
        udaljenost_fun(udaljenost)
    else:
        udaljenost.disabled = False
        udaljenost.value = 1
        udaljenost_fun(udaljenost)

#Linije koda 170 - 200: instrukcije koje se često ponavljaju grupiramo u funkcije (otključavanje i zaključavanje parametara, ažuriranje vrijednosti)
def prom_ime():
    global t1_ime, t2_ime
    t1_ime = "Tijelo 1"
    t2_ime = "Tijelo 2"

def zakljucaj_sve():
    rel_br.checked = False
    rel_ud.checked = False
    rel_br.disabled = True
    rel_ud.disabled = True
    v_baza.disabled = True
    v_eksp.disabled = True
    masa_baza.disabled = True
    masa_eksp.disabled = True
    ecc.disabled = True
    udaljenost.disabled = True

def otkljucaj():
    rel_ud.disabled = False
    rel_br.disabled = False
    masa_baza.disabled = False
    masa_eksp.disabled = False
    ecc.disabled = False

```

```
194 def azuriraj_vrijednosti():
195     ecc_fun(ecc)
196     baza(masa_baza)
197     eksp(masa_eksp)
198     v_baz(v_baza)
199     v_exp(v_eksp)
200     udaljenost_fun(udaljenost)
201
202 #Linije koda 203-231: funkcije pridružene pojedinačnim sliderima korisničkog sučelja. Očitavaju vrijednosti, prikazuju ih te spremaju u varijable
203 def ecc_fun(s):
204     global e
205     vr_ecc.text = '{:1.3f}'.format(s.value)
206     e = s.value
207
208 def baza(s):
209     global masa_baza1
210     vr_baza.text = '{:1.3f} * '.format(s.value)
211     masa_baza1 = s.value
212
213 def eksp(s):
214     global masa_eksp1
215     vr_eksp.text = '10**{:1.0f} '.format(s.value)
216     masa_eksp1 = s.value
217
218 def v_baz(s):
219     global v_baza1
220     vr_vbaza.text = '{:1.3f} * '.format(s.value)
221     v_baza1 = s.value
222
223 def v_exp(s):
224     global v_eksp1
225     vr_veksp.text = '10**{:1.0f}'.format(s.value)
226     v_eksp1 = s.value
227
```

```
228 def udaljenost_fun(s):
229     global r
230     vr_ud.text = '{:1.3f}'.format(s.value)
231     r = s.value
232 
233 #Linije koda 234-412: funkcije pridružene padajućim izbornicima cjelina 3 i 4
234 def izbor(m):
235     global t1_ime,t2_ime, flag4
236     #Za svaki izbor, varijablama pridružujemo predeterminirane vrijednosti te zaključavamo slidere da se unesene vrijednosti ne mogu mijenjati
237     if (m.selected == "Zemlja"):
238         ecc.value = 0.017
239         masa_baza.value = 3.33
240         masa_eksp.value = 6
241         v_baza.value = 3.03
242         v_eksp.value = 4
243         udaljenost.value = 0.983
244         t1_ime = "Zemlja"
245         azuriraj_vrijednosti() #Mijenja vrijednosti prikazane na sliderima na predeterminirane vrijednosti
246         zakljucaj_sve()
247 
248     if(m.selected == "Halleyjev komet"):
249         ecc.value = 0.967
250         masa_baza.value = 9.039
251         masa_eksp.value = 15
252         v_baza.value = 5.50
253         v_eksp.value = 4
254         udaljenost.value = 0.586
255         t1_ime = "Halleyjev komet"
256         azuriraj_vrijednosti()
257         zakljucaj_sve()
258 
259     if(m.selected == "Mars"):
260         ecc.value = 0.093
```

```
        masa_baza.value = 3.112
        masa_eksp.value = 7
        v_baza.value = 2.65
        v_eksp.value = 4
        udaljenost.value = 1.381
        t1_ime = "Mars"
        azuriraj_vrijednosti()
        zakljucaj_sve()

    if(m.selected == "Jupiter"):
        ecc.value = 0.049
        masa_baza.value = 1.047
        masa_eksp.value = 3
        v_baza.value = 1.372
        v_eksp.value = 4
        udaljenost.value = 4.95
        t1_ime = "Jupiter"
        azuriraj_vrijednosti()
        zakljucaj_sve()

    if(m.selected == "Merkur"):
        ecc.value = 0.206
        masa_baza.value = 6.024
        masa_eksp.value = 6
        v_baza.value = 5.898
        v_eksp.value = 4
        udaljenost.value = 0.307
        t1_ime = "Merkur"
        azuriraj_vrijednosti()
        zakljucaj_sve()

    if(m.selected == "Venera"):
        ecc.value = 0.007
        masa_baza.value = 4.085
        masa_eksp.value = 5
```

```
        v_baza.value = 3.526
        v_eksp.value = 4
        udaljenost.value = 0.718
        t1_ime = "Venera"
        azuriraj_vrijednosti()
        zakljucaj_sve()

    if(m.selected == "Saturn"):
        ecc.value = 0.056
        masa_baza.value = 3.499
        masa_eksp.value = 4
        v_baza.value = 1.018
        v_eksp.value = 4
        udaljenost.value = 9.041
        t1_ime = "Saturn"
        azuriraj_vrijednosti()
        zakljucaj_sve()

    if(m.selected == "Uran"):
        ecc.value = 0.0464
        masa_baza.value = 2.291
        masa_eksp.value = 4
        v_baza.value = 7.11
        v_eksp.value = 3
        udaljenost.value = 18.33
        t1_ime = "Uran"
        azuriraj_vrijednosti()
        zakljucaj_sve()

    if(m.selected == "Neptun"):
        ecc.value = 0.009
        masa_baza.value = 1.942
        masa_eksp.value = 4
        v_baza.value = 5.47
        v_eksp.value = 3
```

```
            udaljenost.value = 29.81
            t1_ime = "Neptun"
            azuriraj_vrijednosti()
            zakljucaj_sve()
        t2_ime = "Sunce"
        flag4 = True


def izbor2(m):
    global e2, masa_baza2, masa_eksp2, v_baza2, v_eksp2, r2,t3_ime, t2_ime

    if (m.selected == "Zemlja"):
        e2 = 0.017
        masa_baza2 = 3.33
        masa_eksp2 = 6
        v_baza2 = 3.03
        v_eksp2 = 4
        r2 = 0.983
        t3_ime = "Zemlja"

    if(m.selected == "Mars"):
        e2 = 0.093
        masa_baza2 = 3.112
        masa_eksp2 = 7
        v_baza2 = 2.65
        v_eksp2 = 4
        r2 = 1.381
        t3_ime = "Mars"

    if(m.selected == "Jupiter"):
        e2 = 0.049
        masa_baza2 = 1.047
        masa_eksp2 = 3
        v_baza2 = 1.372
        v_eksp2 = 4
```

```
        r2 = 4.95
        t3_ime = "Jupiter"

    if(m.selected == "Merkur"):
        e2 = 0.206
        masa_baza2 = 6.024
        masa_eksp2 = 6
        v_baza2 = 5.898
        v_eksp2 = 4
        r2 = 0.307
        t3_ime = "Merkur"

    if(m.selected == "Venera"):
        e2 = 0.007
        masa_baza2 = 4.085
        masa_eksp2 = 5
        v_baza2 = 3.526
        v_eksp2 = 4
        r2 = 0.718
        t3_ime = "Venera"

    if(m.selected == "Saturn"):
        e2 = 0.056
        masa_baza2 = 3.499
        masa_eksp2 = 4
        v_baza2 = 1.018
        v_eksp2 = 4
        r2 = 9.041
        t3_ime = "Saturn"

    if(m.selected == "Uran"):
        e2 = 0.046
        masa_baza2 = 2.291
        masa_eksp2 = 4
        v_baza2 = 7.11
```

```
        v_eksp2 = 3
        r2 = 18.33
        t3_ime = "Uran"

    if(m.selected == "Neptun"):
        e2 = 0.009
        masa_baza2 = 1.942
        masa_eksp2 = 4
        v_baza2 = 5.479
        v_eksp2 = 3
        r2 = 29.81
        t3_ime = "Neptun"

#Linije koda 415 - 434: funkcije pridružene dugmadi cjeline 7, ubrzavaju/usporavaju brzinu animacije te prikazuju koliko je trenutna brzina animacije veća/manja naspram početne
def plus(s):
    global rat, brz_anim
    rat = int(rat * 2)
    brz_anim = brz_anim * 2
    if (brz_anim < 1):
        brz_anim2.text = ’{:1.2f}X’.format(brz_anim)
    else:
        brz_anim2.text = ’{:1.0f}X’.format(brz_anim)

def minus(s):
    global rat,brz_anim, kut2, kut
    rat = int(rat * 1/2)
    brz_anim = brz_anim * 1/2
    if (brz_anim < 1):
        brz_anim2.text = ’{:1.2f}X’.format(brz_anim)
    else:
        brz_anim2.text = ’{:1.0f}X’.format(brz_anim)

def brz_anim2(s):
```

```python
    brz_anim2.text = '{:1.2f}'.format(s.value)

#Linije koda 437-464: funkcije pridružene kućicama cjeline 2
def kep3(s): #funkcija pridružena kućici koja označava treći Keplerov zakon
    global t1_ime, t2_ime, v_baza, v_eksp, udaljenost, flag4
    if s.checked:
        meni2.disabled = False #Otključavamo padajući izbornik cjeline 4
        zakljucaj_sve() #Zaključavamo slidere
        meni1.selected = "Zemlja" #Početno izabran planet u padajućem izborniku cjeline 3
        ecc.value = 0.017
        masa_baza.value = 3.33
        masa_eksp.value = 6
        v_baza.value = 3.03
        v_eksp.value = 4
        udaljenost.value = 0.983
        azuriraj_vrijednosti() #Mijenjamo vrijednosti koje se prikazuju na sliderima na vrijednosti Zemlje
        t1_ime = "Zemlja"
        t2_ime = "Sunce"
        flag4 = True

    else:
        meni2.disabled = True #Ukoliko ne simuliramo treći Keplerov zakon, padajući izbornik cjeline 4 se ponovo ključa

def kep2(s): #Funkcija pridružena kućici koja označava drugi Keplerov zakon
    if s.checked:
        animacija_pl.disabled = True #Zaključavamo dugmad za mijenanje brzine animacije
        animacija_mi.disabled = True #Zaključavamo dugmad za mijenanje brzine animacije

    else:
```

```
463         animacija_pl.disabled = False #Otključavamo dugmad za mijenanje brzine animacije
464         animacija_mi.disabled = False #Otključavamo dugmad za mijenjanje brzine animacije
465
466 #Ne radimo s canvasom pa ga minimiziramo
467 scene.width = 0
468 scene.height = 0
469 scene.visible = False
470 scene.resizable = False
471
472 #Linije koda 473 - 520: inicijalizacija elemenata korisničkog sučelja, postavljanje njihovih početnih vrijednosti i pridruženih funkcija
473 scene.caption = "Ispitati drugi Keplerov zakon"
474 kep_2 = checkbox(bind = kep2)
475 scene.append_to_caption("   Ispitati treći Keplerov zakon")
476 kep_3 = checkbox(bind = kep3)
477 scene.append_to_caption('\n\n')
478
479 scene.append_to_caption("Koristi podatke za sustav Sunce - ")
480 meni1 = menu(choices = ['','Merkur', 'Venera', 'Zemlja', 'Mars', 'Jupiter', 'Saturn', 'Uran', 'Neptun'], index = 0, bind = izbor)
481 scene.append_to_caption('\n\n')
482 scene.append_to_caption("Usporedi orbitu sa sustavom Sunce - ")
483 meni2 = menu(choices = ['Merkur', 'Venera', 'Zemlja', 'Mars', 'Jupiter', 'Saturn', 'Uran', 'Neptun'], index = 3, bind = izbor2, disabled = True)
484 scene.append_to_caption('\n\n')
485 scene.append_to_caption('\n\n')
486
487 scene.append_to_caption("Ekscentricitet: ")
488 ecc = slider(min=0, max=0.999, step = 0.001, value = 0, length=400, bind=ecc_fun, right=30, top = 5)
489 vr_ecc = wtext(text='{:1.3f}'.format(ecc.value))
490 scene.append_to_caption('\n\n')
491
```

```
scene.append_to_caption("Omjer masa tijela: ")
masa_baza = slider(min=1, max=10, step = 0.001, value = 1, length=400, bind=baza, right=15, top = 5)
masa_eksp = slider(min=0, max=15, step = 1, value = 3, length=100, bind=eksp, right=30, top = 5)
vr_baza = wtext(text='{:1.3f} * '.format(masa_baza.value))
vr_eksp = wtext(text='{:1.1f}'.format(masa_eksp.value))
scene.append_to_caption('\n\n')

scene.append_to_caption("Zadati relativne brzine tijela u pericentru (m/s): ")
rel_br = checkbox(bind = check_brz)
v_baza = slider(min=0, max=9.9, step = 0.001, value = 1, length=400, bind=v_baz, right=15, top = 5, disabled = True)
v_eksp = slider(min=0, max=10, step = 1, value=0, length=200, bind=v_exp, right=15, top = 5, disabled = True)
vr_vbaza = wtext(text='{:1.3f} * '.format(v_baza.value))
vr_veksp = wtext(text='10**{:1.0f}'.format(v_eksp.value))
scene.append_to_caption('\n\n')

scene.append_to_caption("Zadati udaljenost tijela u pericentru (AU): ")
rel_ud = checkbox(bind = check_ud)
udaljenost = slider(min=0, max=50, step = 0.001, value = 1, length=400, bind=udaljenost_fun, right=15, top = 5, disabled = True)
vr_ud = wtext(text='{:1.3f}'.format(udaljenost.value))
scene.append_to_caption('\n\n')

scene.append_to_caption("Brzina animacije: ")
animacija_pl = button(text = "2X", bind = plus)
animacija_mi = button(text = "X/2", bind = minus)
brz_anim2 = wtext(text='{:1.0f}X'.format(brz_anim))
scene.append_to_caption('\n\n')

b1 = button(text="Pokreni", pos=scene.title_anchor, bind=go1)
b2 = button(text = "Poništi", pos = scene.title_anchor, bind = go2)
```

```
521
522
523 #Početak petlje animacije
524 while(True):
525     rate(rat)
526
527     if(running):
528         if(kut == 0):#Pokreće se pri inicijalnom prolasku kroz petlju
529
530             rat = brz_anim * 15 #Određuje brzinu animacije, ovisno o parametru brz_anim kojeg mijenjamo pomoću dugmadi cjeline 7, inicijalno postavljen na 15
531
532             #Linije koda 533 - 538: određujemo naslove osi
533             if(rel_br.checked == True or flag4 == True):
534                 y_os_v = "v / ms<sup>-1</sup>"
535             if(rel_ud.checked == True or flag4 == True):
536                 y_os_ud = "r / AU"
537                 x_os_pol = "x / AU"
538                 y_os_pol = "y / AU"
539
540             #Linije koda 541 - 545: određujemo naslove grafova
541             naslov_rel_ud = "Udaljenost tijela"
542             naslov_rel_br = "Relativne brzine tijela"
543             if(kep_3.checked == True):
544                 naslov_rel_ud = "Udaljenost tijela od Sunca"
545                 naslov_rel_br = "Rel. brzina tijela i Sunca"
546
547             #Linije koda 548-561: inicijaliziramo potrebne grafove
548             graf_putanje = graph(title = "Orbite dvaju tijela", xtitle=x_os_pol, ytitle=y_os_pol,align="left", width = 775, height = 775)
549             graph1 = gcurve(color=color.blue, dot = True)
550             graph2 = gcurve(color = vec(1,0.4,0), dot = True)
551             graph5 = gcurve(color=color.red, dot = True)
```

```
552         graph7 = gcurve(color = color.purple)
553         graf_udaljenosti1 = graph(title = naslov_rel_ud, ytitle=y_os_ud, xtitle="\u03B8 / \u00b0",align = "left", width = 500, height = 500, xmax=720)
554         graph9 = gcurve(color=vec(0,0.5,0))
555         graf_rel_brzine1 = graph(title = naslov_rel_br, ytitle=y_os_v, xtitle="\u03B8 / \u00b0",align = "left", width = 500, height = 500, xmax=720)
556         graph11 = gcurve(color = vec(0,0.5,0))
557         graf_rel_brzine2 = graph(title = naslov_rel_br, ytitle=y_os_v, xtitle="\u03B8 / \u00b0",align = "left", width = 500, height = 500, xmax=720)
558         graph12 = gcurve(color = color.red)
559         graf_brzine = graph(title = "Orbitalna brzina 1", xtitle="\u03B8 / \u00b0", ytitle= y_os_v,align="left", width = 500, height = 500, xmax=720)
560         graph1.label = t1_ime
561         graph2.label = t2_ime
562
563         #Linije koda 564-566: Računamo masu drugog tijela, parametar orbite i relativnu brzinu uz pomoć vrijednosti zadanih sliderima
564         m2 = masa_baza1 * 10**masa_eksp1 #Masa tijela veće mase
565         r0 = r * (1 + e)#Parametar orbite
566         v_relA = v_baza1 * 10**v_eksp1 #Iznos relativne brzine tijela u pericentru
567         m1 = 1
568
569         poc_polozaj = izr_polozaj(r0, radians(0),e) #Početni položaj tijela reducirane mase (u pericentru)
570         kon_polozaj = izr_polozaj(r0, radians(180),e)#Položaj tijela reducirane mase u apocentru
571         red_masa = (m2*m1)/(m1+m2)
572
```

```
573         if(kep_3.checked == True): #Ukoliko simuliramo treći Keplerov zakon, ponavljamo postupak za još jedno tijelo
574             m4 = masa_baza2 * 10**masa_eksp2
575             v_relA2 = v_baza2 * 10**v_eksp2
576             m3 = 1
577             red_masa2 = (m3*m4)/(m3+m4)
578             r0_2 = r2 * (1 + e2)
579             poc_polozaj2 = izr_polozaj(r0_2, radians(0),e2)
580             kon_polozaj2 = izr_polozaj(r0_2, radians(180),e2)
581             brz2 = (poc_polozaj2[0]**2 + poc_polozaj2[1]**2)/(kon_polozaj2[0]**2 + kon_polozaj2[1]**2)
582 
583             #Linije koda 584-589: postavljamo oznake i boje grafova
584             graph5.label = t3_ime
585             graph9.label = t1_ime
586             graph11.label = t1_ime
587             graph12.label = t3_ime
588             graph9.color = color.blue
589             graph11.color = color.blue
590 
591         #Linije koda 592-617: postavljamo granice grafova
592         if (kep_3.checked == False or kon_polozaj[0] < kon_polozaj2[0]):
593             graf_putanje.xmax = r0/(1-e)+ 0.2
594             graf_putanje.xmin = -r0/(1-e)- 0.2
595             graf_putanje.ymax = r0/(1-e)+ 0.2
596             graf_putanje.ymin = -r0/(1-e)- 0.2
597 
598         else:
599             graf_putanje.xmax = r0_2/(1-e2)+ 0.2
600             graf_putanje.xmin = -r0_2/(1-e2)- 0.2
601             graf_putanje.ymax = r0_2/(1-e2)+ 0.2
602             graf_putanje.ymin = -r0_2/(1-e2)- 0.2
603 
```

```
604 brz = (poc_polozaj[0]**2 + poc_polozaj[1]**2)/(kon_polozaj[0]**2 + kon_polozaj[1]**2) #odnos brzine tijela reducirane mase u apocentru prema onomu pericentru
605
606
607 if(v_relA == 1):
608     ymin1 = brz * vector(pol_tijela(kon_polozaj[0], m1, red_masa), pol_tijela(kon_polozaj[1], m1, red_masa), 0).mag/r * v_relA - 0.1
609 else:
610     ymin1 = brz * vector(pol_tijela(kon_polozaj[0], m1, red_masa), pol_tijela(kon_polozaj[1], m1, red_masa), 0).mag/r * v_relA - 50
611
612 graf_udaljenosti1.ymax = -kon_polozaj[0] + 0.005 #najudaljeniji su u apocentru
613 graf_rel_brzine1.ymin = ymin1 - 0.1
614
615 if(kep_3.checked == True):
616     ymin2 = brz2 * vector(pol_tijela(kon_polozaj2[0], m3, red_masa2), pol_tijela(kon_polozaj2[1], m3, red_masa2), 0).mag/r2 * v_relA2 - 1
617     graf_rel_brzine2.ymin = ymin2 - 100
618
619     #Linije koda 620-623: skaliranje kutnih brzina ukoliko je treći Keplerov zakon označen
620     faktor_brzine = 2 * v_relA/r/faktor_skaliranja #označava početnu kutnu brzinu tijela
621     faktor_brzine2 = 2 * v_relA2/r2/faktor_skaliranja
622     brzina2 = faktor_brzine2
623     brzina = faktor_brzine
624 else:
625     faktor_brzine = 2
626     brzina = faktor_brzine
627
```

```
            a1 = r*(1+e)/(1-e**2) #Velika poluos elipse
            if(kep_3.checked == True):
                kut2 = 0
                a2 = r2*(1+e2)/(1-e2**2)
                r_sred = (sqrt(poc_polozaj[0]**2 + poc_polozaj[1]**2) + sqrt(kon_polozaj[0]**2 + kon_polozaj[1]**2))/2 #Računamo srednju udaljenost od Sunca
                r_sred2 = (sqrt(poc_polozaj2[0]**2 + poc_polozaj2[1]**2) + sqrt(kon_polozaj2[0]**2 + kon_polozaj2[1]**2))/2
                print("Srednja udaljenost nebeskog tijela", t1_ime, "od Sunca iznosi", r_sred, "AU")
                print("Srednja udaljenost nebeskog tijela", t3_ime, "od Sunca iznosi", r_sred2, "AU")

            br_koraka = 30 #Koristimo za simulaciju drugog Keplerovog zakona
            counter = 0 #Koristimo za simulaciju drugog Keplerovog zakona

        polozaj = izr_polozaj(r0, radians(kut),e) #Računamo položaj tijela reducirane mase pomoću parametra orbite, kuta i ekscentriciteta

        if(kep_3.checked == True):
            polozaj2 = izr_polozaj(r0_2, radians(kut2),e2) #Računamo položaj drugog tijela reducirane mase pomoću parametra orbite, kuta i ekscentriciteta

        #Računamo položaje zasebnih tijela, jednom kada je položaj tijela reducirane mase poznat
        tijelo_1 = vector(pol_tijela(polozaj[0], m1, red_masa), pol_tijela(polozaj[1], m1, red_masa), 0)
        tijelo_2 = vector(pol_tijela(-polozaj[0], m2, red_masa), pol_tijela(-polozaj[1], m2, red_masa), 0)

        r_rel = tijelo_1.sub(tijelo_2) #Udaljenost između tijela
        v_rel_tren = rel_lin_br(v_relA, r, r_rel.mag, a1)#računa iznos relativne brzine između tijela
```

```

        #Za simulaciju trećeg Keplerovg zakona, ponavljamo postupak za drugo tijelo
        if(kep_3.checked == True):
            tijelo_3 = vector(pol_tijela(polozaj2[0], m3, red_masa2), pol_tijela(polozaj2[1], m3, red_masa2), 0)
            r_rel2 = tijelo_3.sub(tijelo_2)
            v_rel_tren2 = rel_lin_br(v_relA2, r2,r_rel2.mag, a2)

        if(kep_2.checked == True):
            #Linije koda 660-691: programska izvedba drugog Keplerov zakona sa trima zastavicama
            if (kep_3.checked == True and flag1 == True):
                if(kut >= 180 or kut2 >=180):
                    br_koraka = 30
                    flag1 = False
                    flag3 = True
            else:
                if(kut >= 180 and flag1 == True):
                    br_koraka = 30
                    flag1 = False
                    flag3 = True

            if(flag2 == True):
                if(br_koraka) >= 0:
                    br_koraka = br_koraka - 1
                    if(kep_3.checked == False):
                        polumjer = tijelo_1.mag
                        kut3 = kut
                    elif(r > r2):
                        polumjer = tijelo_3.mag
                        kut3 = kut2
                    else:
                        polumjer = tijelo_1.mag
                        kut3 = kut
```

```
                graph7.plot([0,0])
                graph7.plot([polumjer*cos(radians(kut3)),
                    polumjer*sin(radians(kut3))])
            else:
                flag1 = True
                if flag3 == True:
                    counter = counter + 1
                    flag3 = False
                if(counter == 2):
                    flag2 = False

    #Crtanje putanje tijela na grafu
    graph1.plot(tijelo_1.x, tijelo_1.y)
    graph2.plot(tijelo_2.x, tijelo_2.y)
    if(kep_3.checked == True):
        graph5.plot(tijelo_3.x, tijelo_3.y)

    #Crtanje grafa udaljenosti planeta u ovisnosti o kutu
    if(kep_3.checked == False):
        graph9.plot(kut, r_rel.mag)

    #Crtanje grafova relativnih brzina u ovisnosti o kutu
    graph11.plot(kut, v_rel_tren)
    if(kep_3.checked == True):
        graph12.plot(kut2, v_rel_tren2)

    #Povećavamo vrijednost kuta za slijedeći prolaz kroz petlju
    kut = kut + brzina
    if(kep_3.checked == True):
        kut2 = kut2 + brzina2

    #Određujemo položaj na novoj vrijednosti kuta, te pripadnu kutnu
        brzinu na novom položaju
    polozaj2 = izr_polozaj(r0, radians(kut), e)
    if(kep_3.checked == True):
```

```
        polozaj3 = izr_polozaj(r0_2, radians(kut2), e2)
    brzina = (poc_polozaj[0]**2 + poc_polozaj[1]**2)/(polozaj2[0]**2 +
        polozaj2[1]**2) * faktor_brzine
    if(kep_3.checked == True):
        brzina2 = (poc_polozaj2[0]**2 +
            poc_polozaj2[1]**2)/(polozaj3[0]**2 + polozaj3[1]**2) *
            faktor_brzine2
```

Literatura

[1] Morin, D. Introduction to Classical Mechanics: With Problems and Solutions. 1st ed. : Cambridge University press, 2008.

[2] Dourmashkin, P. Classical Mechanics: MIT 8.01 Course Notes. 1st ed. : Wiley, 2013.

[3] Logsdon, T. Orbital mechanics: Theory and application. 1st ed. : Wiley, 1997.

[4] Winsberg, E. Computer Simulations in Science, The Stanford Encyclopedia of Philosophy, (Spring 2019 ed.), https://plato.stanford.edu/archives/spr2019/entries/simulations-science/, 23.06.2019

[5] Hawking, S. A Brief History of Time. 10th ed. : Bantam, 2008.

[6] Labor, J. Fizika 1: udžbenik za 1. razred gimnazije. Zagreb : Alfa, 2008.

[7] Paar, V. ; Hrlec, A. ; Sambolek, M. ; Rešetar K.V. Fizika oko nas 1. Zagreb : Školska knjiga, 2018.

[8] Skup autora: XIV. hrvatski simpozij o nastavi fizike : Hrvatsko fizikalno društvo, 2019.

[9] VPython, (08.03.2019.), https://en.wikipedia.org/wiki/VPython, 10.06.2019.