

Numerička simulacija evolucije zvezdanog jata

Urađena je numerička simulacija evolucije zvezdanog jata u Moduli2. Program računa položaje i brzine 100 zvezda u vremenskom intervalu od milion godina sa korakom od 100 godina. Mase zvezda su slučajno raspoređene u intervalu od jedne do tri mase Sunca. Dobijeni rezultati potvrđuju hipotezu o grupisanju.

Da bi odredili jednačine kretanja N objekata u jednom gravitacionom sistemu, gde svi objekti učestvuju u gravitacionoj interakciji, moramo rešiti problem N -tela. Kako je ovaj problem analitički nerešiv (osim kada se svodi na problem 2 tela) mora se pristupiti numeričkom izračunavanju vektora položaja i brzina u bilo kojem vremenskom trenutku.

Na osnovu poznatih početnih uslova (početni vektori položaja i brzina objekata kao i njihove mase) možemo odrediti kasnije položaje i brzine svake od komponenti sistema i tako pratiti dinamičku evoluciju istog. Poznavanje dinamičkog evolutivnog toka nekog zvezdanog sistema, kao i poznavanje faktora stabilnosti, odnosno nestabilnosti, jednog gravitacionog sistema, može doprineti razvoju matematičkih modela zvezdanih sistema.

Zadatak je bio da se napravi računarska simulacija zvezdanog jata koje ima 100 objekata. Potom, na osnovu dobijenih rezultata, praćena je promena odnosa kinetičke i potencijalne energije u vremenu, kao i promena raspodele broja objekata na rasrojanju od centra mase sistema u vremenu i promena raspodele broja objekata od modula vektora brzine u vremenu. Očekivalo se da raspodela broja objekata od rastojanja centra mase menja oblik u smislu grupisanja objekata na nekom rasrojanju.

Kako se kretanje i -tog objekta u sistemu može predstaviti diferencijalnom jednačinom:

$$\frac{d^2 \vec{r}_i}{dt^2} = \gamma \sum_{k=1}^n \frac{m_k}{r_{ik}^3} \vec{r}_{ik}, \quad (1)$$

Zoran Todosijević
(1977), Kruševac, 22.
decembra 33, učenik 4.
razreda Gimnazije u
Kruševcu.

to je potrebno napraviti program koji numerički rešava sistem od N ovakvih jednačina. Korišćeni metod za rešavanje sistema diferencijalnih jednačina je četvorokoračni Runge-Kutta metod sa kontrolom greške i adaptivnim korakom. Program je implementiran u jeziku Modula2 u Top Speed varijanti.

Metod

Diferencijalnu jednačinu prvog reda možemo predstaviti kao:

$$\frac{d y}{d x} = f(x, y), \quad (2)$$

dok diferencijalnu jednačinu drugog reda možemo predstaviti kao sistem od dve diferencijalne jednačine prvog reda:

$$\begin{aligned} \frac{d y}{d x} &= g \\ \frac{d g}{d x} &= f(x, y) \end{aligned} \quad (3)$$

Zbog toga se sistem od N diferencijalnih vektorskih jednačina drugog reda (1) može predstaviti kao sistem od $6N$ običnih diferencijalnih jednačina prvog reda.

Kada su nam poznate vrednosti svih funkcija y_i u početnoj tački, konačni priraštaj (korak) nezavisno promenjive x dovodi do promene svih funkcija za odgovarajući priraštaj y_i :

$$\Delta y_i = f_i(x, y_1, y_2, \dots, y_n) \quad (4)$$

koji približno možemo da nađemo iz (2) zamenjujući $d x$ i $d y$ sa konačnim vrednostima x i y . Kod Runge-Kutta metoda četvrtog reda vrednosti svih desnih strana jednačina (2) računaju se četiri puta po jednom koraku, dva puta u sredini i jednom na krajevima intervala.

$$\begin{aligned} f_i^1 &= f_i(x, y_1, y_2, \dots, y_n) \\ f_i^2 &= f_i\left(x + \frac{\Delta x}{2}, y_1 + \frac{f_1^1 \Delta x}{2}, y_2 + \frac{f_2^1 \Delta x}{2}, \dots, y_n + \frac{f_n^1 \Delta x}{2}\right) \\ f_i^3 &= f_i\left(x + \frac{\Delta x}{2}, y_1 + \frac{f_1^2 \Delta x}{2}, y_2 + \frac{f_2^2 \Delta x}{2}, \dots, y_n + \frac{f_n^2 \Delta x}{2}\right) \\ f_i^4 &= f_i\left(x + \Delta x, y_1 + f_1^3 \Delta x, y_2 + f_2^3 \Delta x, \dots, y_n + f_n^3 \Delta x\right) \end{aligned} \quad (5)$$

Na osnovu ovih vrednosti izvoda izračunavaju se vrednosti y_i na kraju intervala:

$$y_i(x + \Delta x) = y_i(x) + \frac{\Delta x}{6} (f_i^1 + 2f_i^2 + 2f_i^3 + f_i^4) \quad (6)$$

Primenjujući opisani metod na vektorsku diferencijalnu jednačinu (1), možemo je transformisati kao sistem od dve vektorske diferencijalne jednačine prvoga reda:

$$\begin{aligned} \frac{d \vec{r}_i}{d t} &= \vec{v}_i \\ \frac{d \vec{v}_i}{d t} &= \gamma \sum_{k=1}^n \frac{m_k}{3} \vec{r}_{ik} \end{aligned} \quad (7)$$

Na opisan način za zadati vremenski korak Δx dolazimo do podatka o vektorima položaja i brzina u vremenskom trenutku $t = t + \Delta t$. Ponavljajući opisani postupak dovoljan broj puta, možemo doći do podatka o vektorima položaja i brzina u zadatom vremenskom intervalu.

Jedan od faktora numeričke stabilnosti algoritma i njegove brzine je adaptivna promena integracionog koraka Δx . Da bi mogli da povećamo ili smanjimo korak Δx moramo imati kontrolni faktor, odnosno veličinu koja nam pokazuje da li možemo povećati integracioni korak, ili ga moramo smanjiti. Jedan od četiri opšta integrala problema N-tela je zbir kinetičke i potencijalne energije koji je uvek konstantan. Kako je ta vrednost konstantna u toku vremena, može se upotrebiti kao kontrolni faktor pri povećanju ili smanjenju integracionog koraka. Ako je relativna greška, merena između dva integraciona koraka, manja od neke unapred zadate vrednosti, iteracija je bila uspešna i korak se može povećati. U slučaju da je iteracija bila neuspešna, korak integracije se smanjuje i ta iteracija se ponavlja. Dati postupak ponavljamo sve dok vrednost nezavisno promenjive ne dostigne neku unapred zadanu vrednost.

U praksi, povećanje i smanjenje integracionog koraka je rešeno množenjem predhodnog koraka sa koeficijentom rasta ($k > 1.0$), odnosno koeficijentom pada ($k < 1.0$). Vrednost koeficijenta rasta iznosi 1.05, dok vrednost koeficijenta pada iznosi 0.2. To praktično znači da integracioni korak brže opada nego što raste, što rezultuje manjim brojem ponovljenih izračunavanja iste integracije.

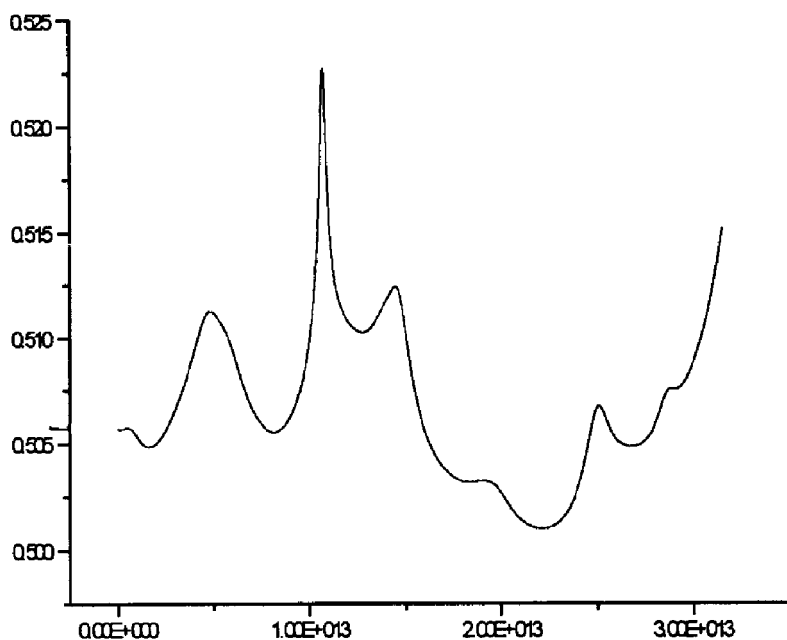
Posle određenog vremena program zapisuje vrednosti vektora položaja i brzina u datoteku koja se kasnije obrađuje.

Da bi prethodno opisani metod bio primenjen na sistem gravitaciono povezanih objekata, moramo zadati neke inicijalne parametre, odnosno

početne vrednosti vektora položaja, brzine i mase za svaki od objekata sistema. Da bi se taj problem rešio, pristupilo se modeliranju zvezdanog sistema na sledeći način. U sferu radijusa R raspoređuje se na slučajan način N objekata. Potom se izračuna potencijalna energija sistema. Pod pretpostavkom da je sistem pseudostabilan i da u početku svi objekti imaju istu kinetičku energiju, izračunavamo module vektora brzina tako da je odnos ukupne kinetičke i potencijalne energije približno 0.5. Pod pretpostavkom da je vektor brzine svakog od objekata na početku normalan na ukupan vektor delovanja svih sila i da se dati objekat kreće suprotno od njemu najbližeg u sistemu, izračunavamo vektor brzine za svaki od objekata. Masa objekata se određuje na slučajan način tako da bude u nekom unapred definisanom intervalu. Raspodela broja objekata po masama je Gauss-ova. Simulacija je izvršena na računaru 406DX4 na 100 MHz

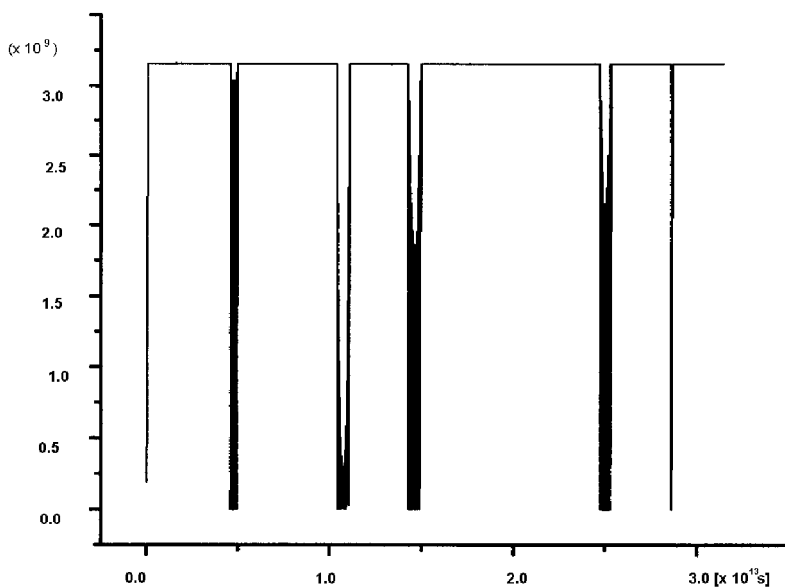
Rezultati i diskusija

Izvršena je simulacija sistema od 100 objekata koji se nalaze u sferi od 1.0×10^{17} m, u vremenu od 1.0×10^6 godina. Mase objekata variraju u intervalu od 1.0 do 3.0 Sunčevih masa. Podaci o vektoru položaja i brzina su snimani u datoteku na svakih 100 godina simulacionog vremena što znači da je dobijeno 1 000 000 podataka stanja sistema unutar vremenskog intervala od 1.0×10^6 godina.



*Slika 1.
Odnos kinetičke i
potencijalne energije.
Vreme je dato u
sekundama.*

*Figure 1.
Kinetic and potential
energy ratio. The
time is given in
seconds.*



*Slika 2.
Veličina
integracionog koraka
u funkciji vremena.*

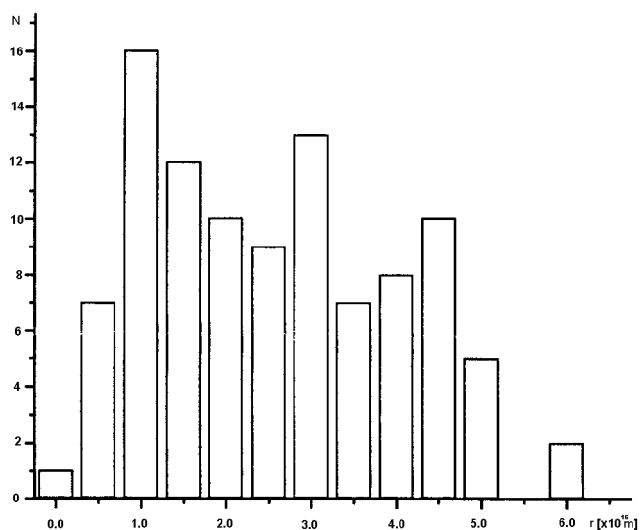
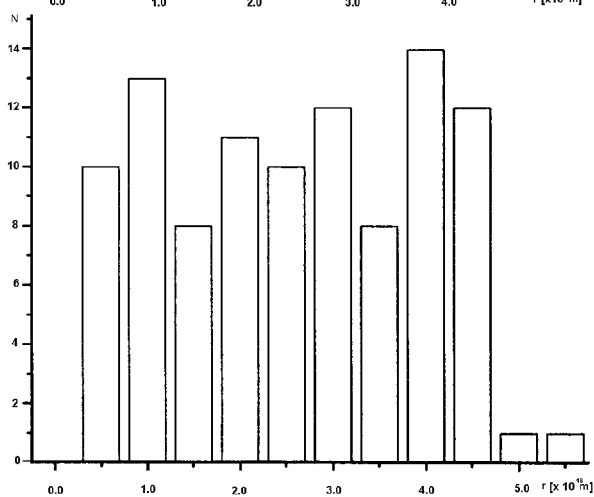
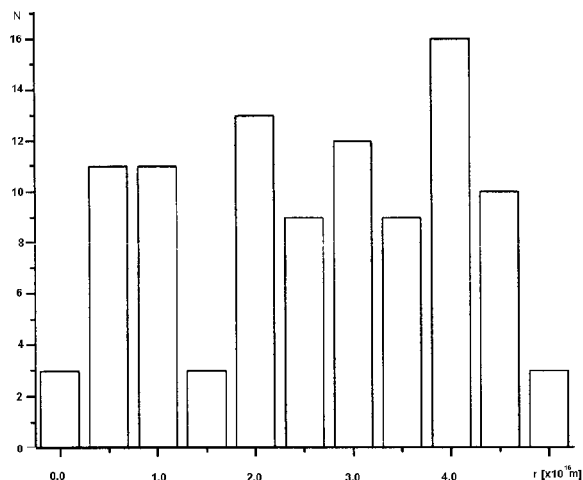
*Figure 2.
Step of integration as
a function of time.*

Grafik na slici 1 prikazuje odnos kinetičke i potencijalne energije u vremenu, dok je na slici 2 predstavljena vrednost integracionog koraka u vremenu. Primećuje se da sa porastom odnosa kinetičke i potencijalne energije opada vrednost integracionog koraka. S tim u vezi primećuje se da, kada vrednost odnosa kinetičke i potencijalne energije poraste, smanjenje integracionog koraka dovodi do njegovog ponovnog opadanja, gde je granična vrednost 0.5.

Grafici na slici 3 prikazuju raspodelu broja objekata pa rastojanju od centra mase. Početna slučajna raspodela (grafik a) se menja u toku vremena u smislu koncentracije objekata oko centra mase sistema što se vidi sa grafika b i c. Podaci za grafike su uzeti sa početka, sredine i kraja simulacije. Na graficima sa slike 4 su date raspodele broja objekata od modula vektora brzina. Primećuje se da se početna slučajna raspodela menja tokom vremena u smislu da se sa vremenom smanjuje broj objekata sa malim ili velikim brzinama. I ovde grafici ilustruju podatke uzete sa početka, sredine i kraja simulacije.

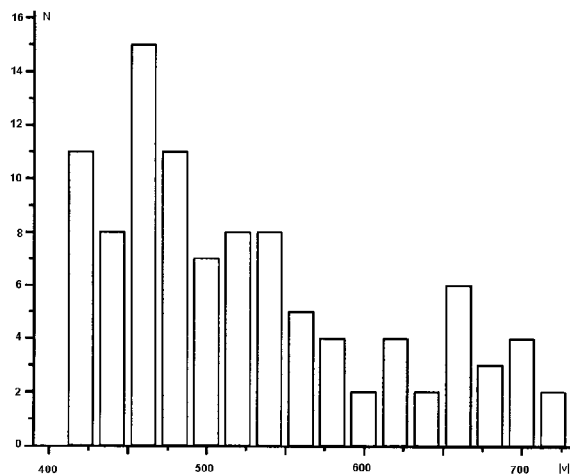
Zaključak

Dobijeni rezultati su potvrdili pretpostavke o grupisanju objekata, kao i o promeni odnosa kinetičke i potencija energije u vremenu. Runge-Kutta metod se pokazao kao pogodan za ovakve proračune, ali autor rada predlaže u istu svrhu korišćenje bržeg Adamsovog korektor prediktor metoda. Što se tiče numeričke stabilnosti algoritma, on se pokazao kao dosta pouzdan, ali ne i brz.



Slika 3.
 Raspodela broja objekata po rastojanju od centra mase. Podaci su za:
 a) početak simulacije (prvo snimanje podataka)
 b) $t=1.90598 \times 10^{13} s$
 c) kraj simulacije ($t=3.124359 \times 10^{13} s$)

Figure 3.
 Distribution of objects along the distance from the center of mass. Data are taken from:
 a) the beginning of the simulation (the first sampling).
 b) the second sampling
 c) the final sampling



Slika 4.
 Raspodela broja objekata po modulima vektora brzina. Vreme je uzeto za:
 a) početak simulacije (prvo snimanje podataka)
 b) $t=1.90598 \times 10^{13} s$
 c) kraj simulacije ($t=3.124358 \times 10^{13} s$)

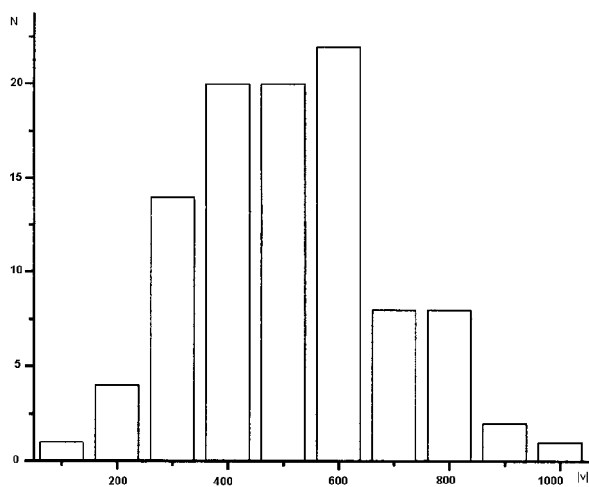
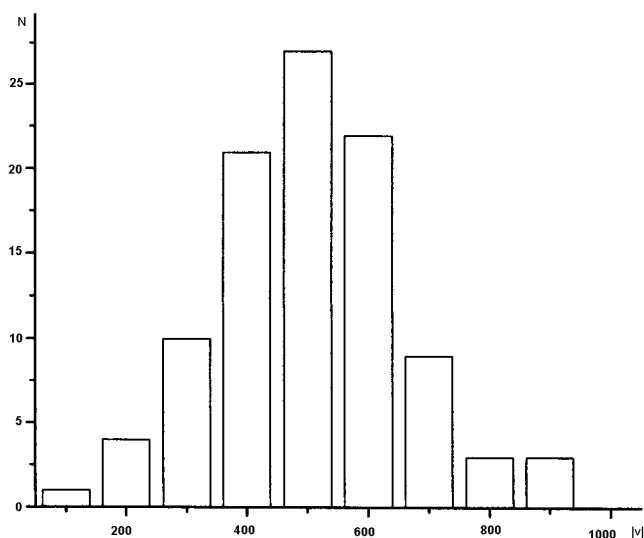


Figure 4.
 Distribution of velocities of objects along the distance from the center of mass. Data are taken from:
 a) the beginning of the simulation (the first sampling).
 b) the second sampling
 c) the final sampling



Literatura

- [1] Mendaš, Milutinović, Ignjatijević. 1991. *100 najkorisnijih FORTRAN-skih potprograma,*

Zoran Todosijević

Numerical Simulation of a Star Cluster Evolution

Numerical simulation of a star cluster evolution using Modula2 programming language was made. The program calculates motions (positions and velocities) of one hundred stellar objects of 1 to 3 solar masses during the interval of a million year, at every 100 years of simulated time. At first the objects are distributed randomly. The results support the hypothesis that the objects will become grouped after some time, and that their total kinetic and potential energy will also change in time. Author used Runge-Kutta method, and suggested the faster Adams method.

