Taivaanmekaniikka
Numeerinen integrointi Runge-Kutta -menetelmällä

Anni Järvenpää

17. joulukuuta 2015

1 Runge-Kutta -menetelmä

Runge-Kutta -menetelmällä voidaan ratkaista numeerisesti differentiaaliyhtälöitä. Sitä voidaan käyttää mihin tahansa muotoa

$$y' = f(t, y)$$
$$y(t_0) = y_o$$

olevaan alkuarvo-ongelmaan. Riippuen halutusta tarkkuudesta voidaan käyttää esimerkiksi toisen, neljännen tai kahdeksannen asteen Runge-Kutta -menetelmää. Näistä neljännen asteen menetelmä (RK4) on kuitenkin usein paras kompromissi suorituskyvyn ja tarkkuuden välillä. RK4-menetelmää käytettäessä määritellään

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{1}$$

missä

$$k_1 = f(t_n, y_n)$$
$$k_2 = f(t_n + \frac{h}{2}, y_n + k_1\frac{h}{2})$$
$$k_3 = f(t_n + \frac{h}{2}, y_n + k_2\frac{h}{2})$$
$$k_4 = f(t_n, y_n + k_3 h)$$

ja h on haluttu aika-askeleen pituus. Näin funktion derivaatan arvoa arvioidaan kolmessa pisteessä kutakin aika-askelta kohden, joista keskimmäistä kahdesti (k_1 ja k_2). Seuraava arvo voidaan aina laskea edellisen arvon ja aika-askeleen pituuden perusteella ja näin integroida halutun välin yli. [5]

2 RK4-menetelmän soveltaminen n kappaleen järjestelmään

Tutkittaessa n kappaleen järjestelmää, tiedetään kappaleiden paikoista ja nopeuksista

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
$$\vec{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$
$$\vec{a} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}.$$

Näistä nähdään kappaleiden liikkeen määräävät differentiaaliyhtälöt

$$\dot{\vec{r}} = \vec{v}$$
$$\dot{\vec{v}} = \vec{a}.$$

Järjestelmässä, jossa vaikuttaa ainoastaan painovoima, pätee

$$\dot{\vec{r}}_i = \vec{v}_i$$
$$\dot{\vec{v}}_i = \gamma \sum_{i=1}^{n} \sum_{j=0, j!=i}^{n} m_j \frac{\vec{r}_i - \vec{r}_j}{|r_{ij}^3|},$$

missä

$$r_{ij} = |\vec{r}_i - \vec{r}_j|.$$

r_0 (AU)			v_0 (AU/yr)			M ($M_\odot$)
0	0	0	0	0	0	1.56
-60.5884	30.8714	0	-0.43205	-0.847947	0	6.6845e-3
12.5497	-24.8536	0	1.39023	0.590117	0	9.5492e-3
5.66560	-13.3473	0	1.89707	0.80526	0	8.5943e-3

Taulukko 1: Tähden ja planeettojen paikat, nopeudet ja massat simulaation alussa.

Nyt sekä paikat että nopeudet yhden aika-askeleen kuluttua saadaan, kun merkitään $f = \dot{\vec{r}}$ ja $g = \dot{\vec{v}}$ ja näille lasketaan kertoimet k kummallekin funktiolle vuorotellen:

$$
\begin{aligned}
k_{1,r} &= f(r_0, y_0) \\
k_{1,v} &= g(r_0, y_0) \\
k_{2,r} &= f(r_0 + \frac{1}{2}\tau k_{1,r}, v_0 + \frac{1}{2}\tau k_{1,v}) \\
k_{2,v} &= f(r_0 + \frac{1}{2}\tau k_{1,r}, v_0 + \frac{1}{2}\tau k_{1,v}) \\
k_{3,r} &= f(r_0 + \frac{1}{2}\tau k_{2,r}, v_0 + \frac{1}{2}\tau k_{2,v}) \\
k_{3,v} &= f(r_0 + \frac{1}{2}\tau k_{2,r}, v_0 + \frac{1}{2}\tau k_{2,v}) \\
k_{4,r} &= f(r_0 + \tau, v_0 + \tau) \\
k_{4,v} &= g(r_0 + \tau, v_0 + \tau)
\end{aligned}
$$

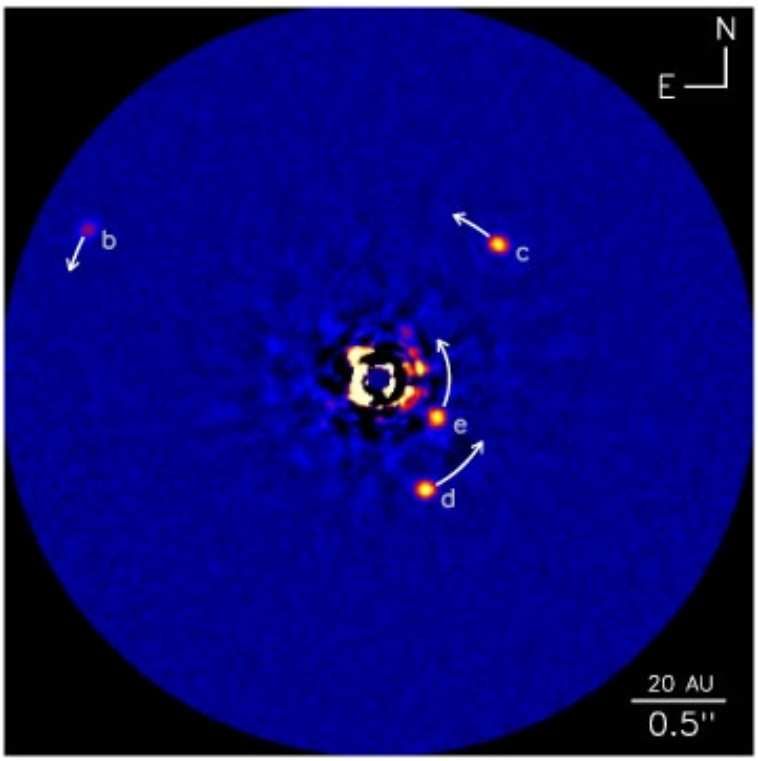
ja näistä

$$
\begin{aligned}
r_1 &= r_0 + \frac{1}{6}\tau(k_{1,r} + 2k_{2,r} + 2k_{3,r} + k_{4,r}) \\
v_1 &= v_0 + \frac{1}{6}\tau(k_{1,v} + 2k_{2,v} + 2k_{3,v} + k_{4,v})
\end{aligned}
$$

missä τ on käytetty aika-askeleen pituus.[5]

3 Planeettakunnan HR 8799 simuloiminen RK4-menetelmällä

Eksoplaneettakunnan HR 8799 dynaamisen evoluution tutkimiseksi toteutin oman RK4-implementaation Pythonilla. Kirjoittamani ohjelma on nähtävillä Githubissa `https://github.com/aajarven/runge-kutta`. Alkuarvot määritin planeettojen rataelementtien perusteella siten, että planeetat olivat simulaation alussa likimain samassa asennossa kuin kuvassa 1, jolloin alkuarvoiksi tuli taulukon 3 mukaiset arvot [1, 2, 3, 4].

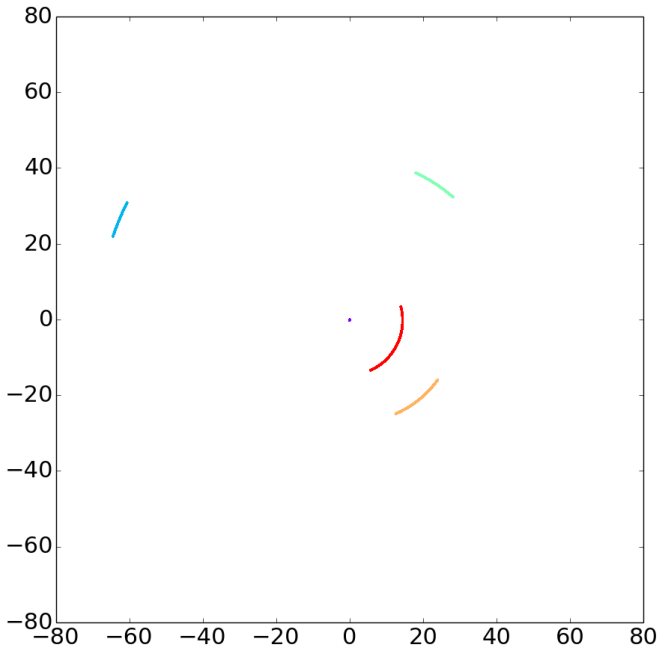


Kuva 1: W. M. Keck -teleskoopin kuva planeettakunnasta HR 8799 (Ben Zuckerman [CC BY 3.0])

Näistä loin liitteessä A esitetyn alkuarvotiedoston. Tiedostossa kukin kappale on omalla rivillään vektorien komponentit pilkulla eroteltuna ja vektorit puolipisteellä. #-merkillä alkavat rivit ovat kommentteja, eikä ohjelma huomioi niitä. Lisäksi kirjoitin yksinkertaisen skriptin, jolla voidaan helposti sekä ajaa simulaatio että plotata sen tulokset yhdellä komennolla. Tämä skripti on nähtävissä liitteessä B.

Liitteen A alkuarvoilla ajetun simulaation tuloksia on nähtävillä kuvissa 2–6. Kaikissa ploteissa yksikkönä on AU ja systeemin massakeskipiste on origossa. Violetilla tähti, punaisella b-, oranssilla c-, vihreällä d- ja sinisellä e-planeetta. Ploteista kahdessa ensimmäisessä ei ole nähtävissä mitään erikoista, mutta tuhannen vuoden simulaatioissa nähdään selvästi, etteivät planeettojen radat ole sulkeutuvia käyriä, vaan niissä esiintyy selvästi havaittavaa huojuntaa. Aika-askeleen lyhentäminen ei vähennä tätä efektiä, kuten kuvia 4 ja 5 vertaamalla voidaan huomata. 10 000 vuoden simulaatiossa (kuva 6) tämä efekti on vielä selvemmin nähtävissä.

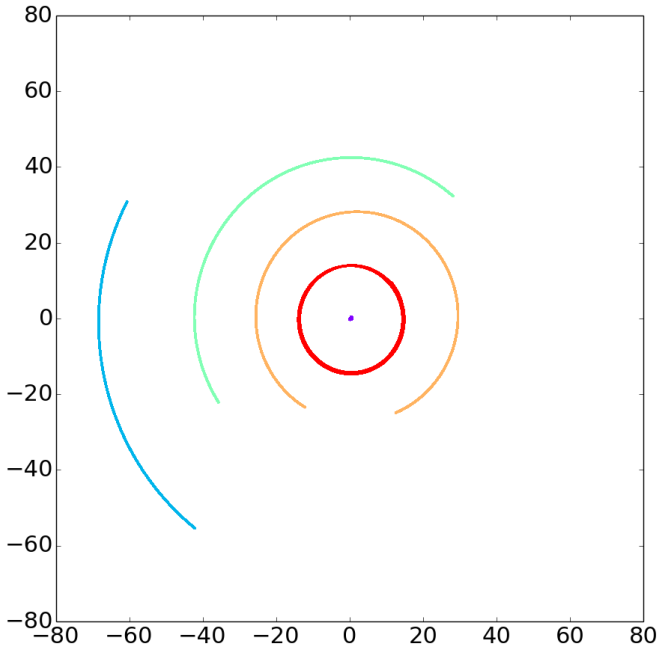
Todennäköisesti havaitut muutokset radoissa johtuvat epätarkkuuksista rataelementeissä. Kaikista kappaleista tunnettiin ainoastaan massa ja isoakselin puolikas ja näistäkin massoissa esiintyi jopa M_{ν_4} virheitä. Yhdelle planeetoista ilmoitettiin myös radan eksentrisyys, mutta ajamassani simulaatiossa kaikki planeetat lähtevät liikkeelle nopeuksilla, jotka pitäisivät ne ympyräradoilla, mikäli ne vuorovaikuttaisivat pelkästään tähden kanssa. Jonkin verran esimerkiksi Aurinkokuntaa kaoottisempi tutkittu järjestelmä saattaa



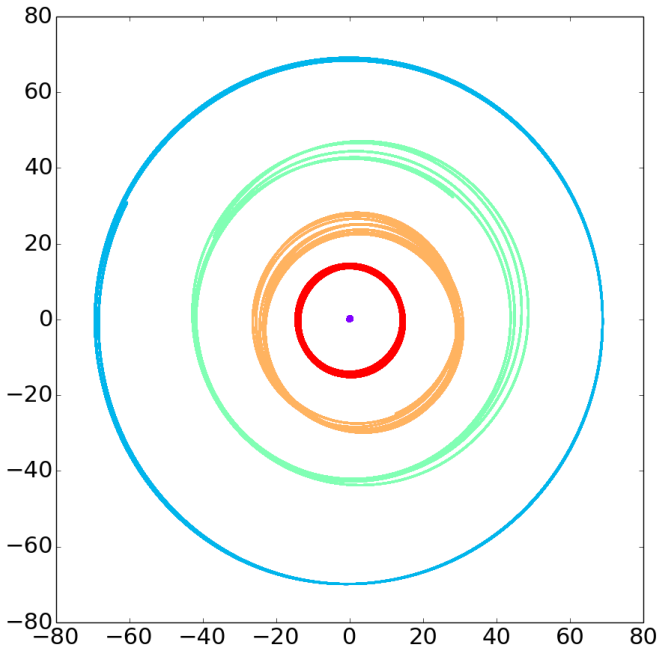
Kuva 2: 10 vuotta 0,001 vuoden (0,37 d) aika-askeleella

kuitenkin todellisuudessakin olla, sillä planeettakunnassa on Aurinkokuntaa enemmän massiivisia planeettoja, jotka häiritsevät toisiaan.

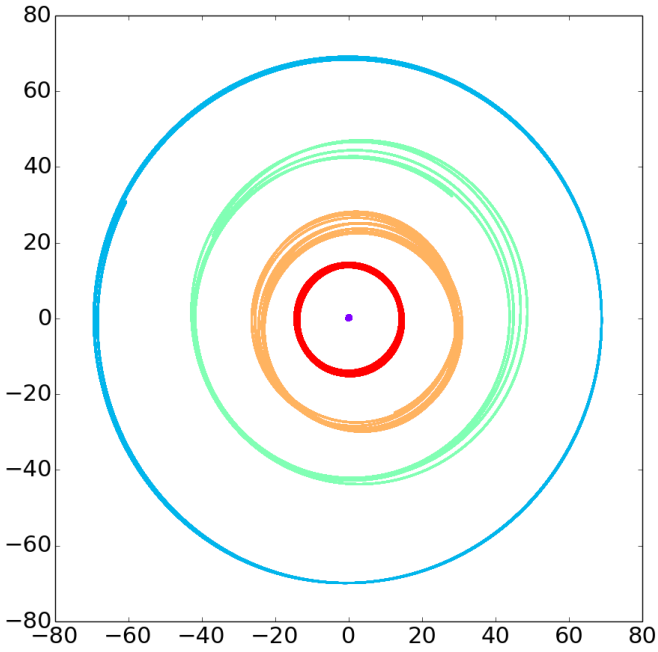
Järjestelmän kehitystä voidaan laskea RK4-menetelmällä niin pitkälle kuin käytössä oleva laskenta-aika ja muisti sallivat. Pitkissä ajoissa lähtöarvojen tarkkuuden merkitys kuitenkin korostuu, sillä pienet häiriöt saattavat kasaantua.



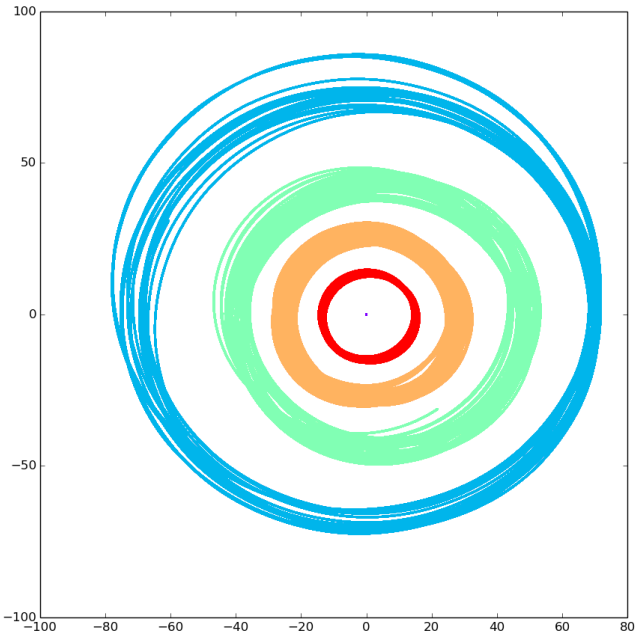
Kuva 3: 100 vuotta 0,001 vuoden (0,37 d) aika-askeleella



Kuva 4: 1000 vuotta 0,01 vuoden (3,7 d) aika-askeleella



Kuva 5: 1000 vuotta 0,001 vuoden (0,37 d) aika-askeleella



Kuva 6: 10000 vuotta 0,01 vuoden (3,7 d) aika-askeleella

Viitteet

[1] exoplanet.eu: Planet hr 8799 b. http://exoplanet.eu/catalog/hr_8799_b/#9048. Luettu 17.12.2015.

[2] exoplanet.eu: Planet hr 8799 c. http://exoplanet.eu/catalog/hr_8799_c/#9048. Luettu 17.12.2015.

[3] exoplanet.eu: Planet hr 8799 d. http://exoplanet.eu/catalog/hr_8799_d/. Luettu 17.12.2015.

[4] exoplanet.eu: Planet hr 8799 e. http://exoplanet.eu/catalog/hr_8799_e/. Luettu 17.12.2015.

[5] C. J. Voesnek. Implementing a fourth order runge-kutta method for orbit simulation. http://spiff.rit.edu/richmond/nbody/OrbitRungeKutta4.pdf. Luettu 17.12.2015.

A Käytetty alkuarvotiedosto

```
#X (AU)                  V (AU/yr)                M (M_sun)
#
# tähti
0,         0,         0;  0,         0,          0; 1.56
#
# b
-60.5884, 30.8714,  0; -0.43205, -0.847947, 0; 6.6845e-3
#
# c
28.1449,  32.3770,  0; -0.90426, 0.786061,  0; 9.5492e-3
#
# d
12.5497,  -24.8536, 0; 1.39023,  0.590117,  0; 9.5492e-3
#
# e
5.66560,  -13.3473, 0; 1.89707,  0.80526,   0; 8.5943e-3
```

B Simulointi- ja plottausskripti

```
#!/bin/bash
if [ "$#" -ne 5 ]; then
        echo "anna argumentteina input-tiedosto, simuloitava
           aika, aika-askelen pituus, output-tiedostonimen
           juuri ja plottaustapa (s, simple, t tai tahti)
           joista ensimmäiset kaksi plottaavat kappaleet siin
           ä koordinaatistossa jossa ne on annettu ja toinen
           keskittää plotin jokaisessa framessa listan ensimm
           äiseen kappaleeseen"
else
        python rk.py $1 $2 $3 $4
        echo "laskettu, aloitetaan plottaus"
        python plottool.py $4-X.txt 0 1 $5
fi
```