Dynamiikan jatkokurssin lopputyö

Vili Oja 014339369

30. huhikuuta 2017

1 Ohjelmat

Koodasin sekä leapfrog-integraattorin että RK4-integraattorin. Ohjelman compiloiminen onnistuu helposti makefilen avulla. Turhat tiedostot saa siivottua `make clean` -komennon avulla. Ohjelman suoritus tapahtuu seuraavanlaisilla komentoriviargumenteilla:

./integraattori.exe [inpufile] [outputfile] [aikajakso] [tallennustiheys] [aika-askel] [leapfrog]

Inputfile ja outputfile kertovat mistä tiedostosta ohjelma lukee simuloitavien kappaleiden alkuperäiset ominaisuudet, ja mihin simuloitu data tallennetaan. Inputfile on input_precise.dat. Aikajakso on vuosissa se aika, joka halutaan simuloida. Tämä voi olla kokonaisluku tai desimaaliluku. Tallennustiheys kertoo ohjelmalle kuinka monen aika-askeleen välein uudet lasketut paikat tallennetaan. Täytyy olla kokonaisluku. Mikäli tämä on 1, tallennetaan jokainen simuloitu paikka. Aika-askel kertoo kuinka monen vuoden välein ohjelma laskee uudet paikat. Voi olla kokonaisluku tai desimaaliluku. Eli jos tämä on 0.1, ohjelma laskee uudet paikat kymmene kertaa vuodessa. Leafrog-parametri kertoo ohjelmalle käytetäänkö leapfrogia vai RK4:sta. Tämä on kokonaisluku: 1 jos halutaan käyttää leapfrogia, ja 0 jos halutaan käyttää RK4:sta.

Esimerkki tästä:

./integraattori.exe input_precise.txt tulokset.dat 100000 1000 0.1 1

Tein myös Pythonilla ohjelman joka laskee rataelementtejä. Otin mallia laskuihin tästä ohjelmasta: `https://github.com/RazerM/orbital`. Ohjelmaa täytyy itse hieman muokata ja poistaa kommentit lopusta tietyiltä riveiltä riippuen mitä haluaa piirtää. Ohjelmaa kutsutaan seuraavanlaisella komennolla:

python elementtilaskuri.py [inpufile] [aikaväli]

Inputfile kertoo mistä tiedostosta piirrettävä data katsotaan. Käytännössä sama kuin integraattorin outpufile. Aikaväli kertoo kuinka monen vuoden välein inputfilen paikat on tallennettu. Tätä käytetään ajanhetken kirjaamiseen kuviin. Voi olla kokonaisluku tai desimaaliluku. Käytännössä tämä on sama kuin integraattorin [tallennustiheys] * [aika-askel].

Esimerkki tästä:

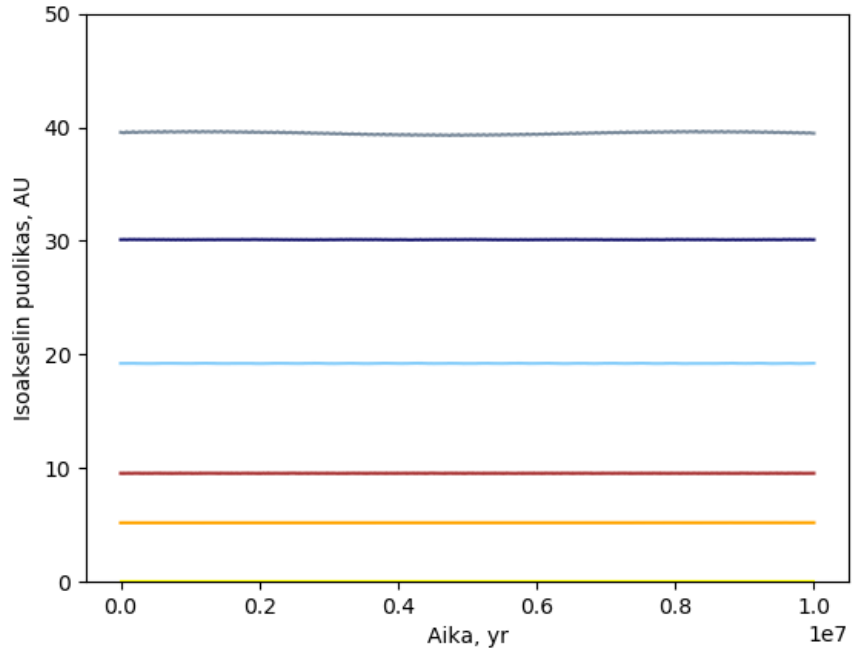
python elementtilaskuri.py tulokset.dat 100

2 Tehtävä 15

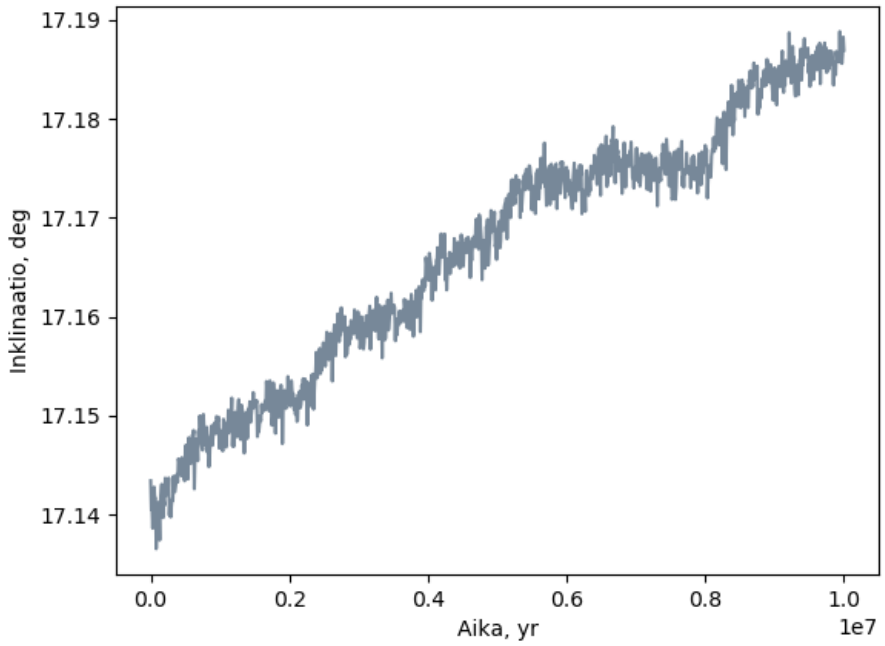
2.1 Oma data

Minulla oli integraattorini kanssa hieman outoja ongelmia. Ajamalla ohjelmaa 0,05 vuoden aika-askeleella ohjelma toimi hyvin, ja planeetat pysyivät kaikki paikallaan. Mutta jokin integroinnissa on pielessä, sillä rataelementit eivät muuttuneet odotetusti. Kuvassa 1 näkyy 10e7 vuoden ajan tehty integrointi leapfrogilla 0,05 vuoden välein ja sillä saadut isoakselin puolikkaat. Kuvassa 2 näkyy samasta integroinnista Pluton inklinaation kehitys, joka ei selvästikään ole oskilloivaa. RK4:llä kuvaajat olivat liki identtiset. Olen liittänyt saamani datan mukaan.

Koitin debugata koodiani, mutta en keskinyt syytä tälle käytökselle. Päädyin sellaiseen ratkaisuun, että koska Annilla integraattorin antama data toimi myös elementtien osalta, lainasin tuloksia häneltä jotta pystyin esittelemään elementtilaskurini toiminnallisuutta. Kaikki muut kuvat on siis tehty Annin datan perusteella.



Kuva 1: Leapfrogilla saadut planeettojen isoakselien puolikkaat. Planeetat näyttävät pysyvän siististi radoillaan.



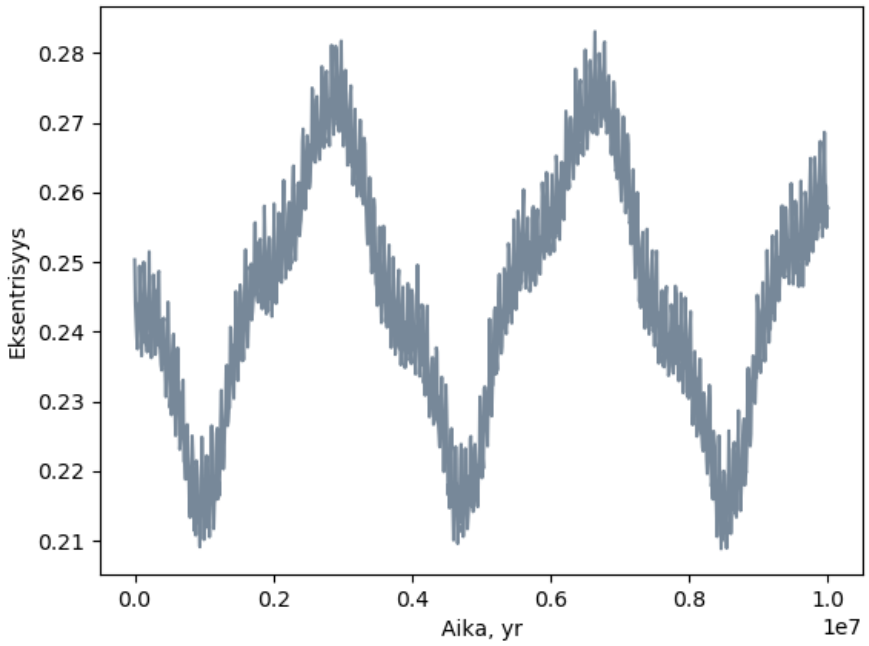
Kuva 2: Leapfrogilla saatu Pluton inklinaation kehitys.

2.2 Annin data

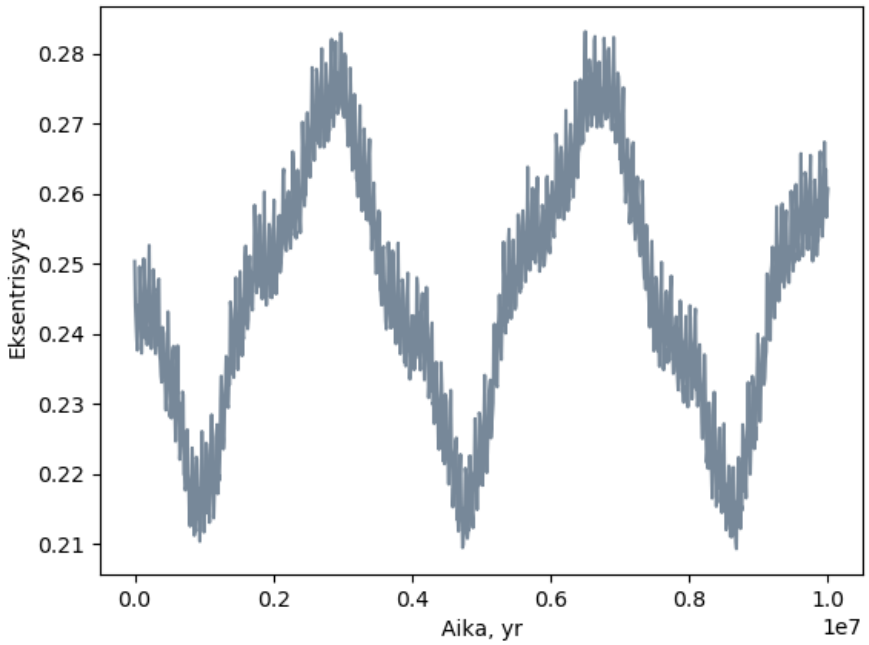
Dataa on neljässä tiedostossa: rk-pitka.dat sisältää 10e7 vuoden integroinnin 30 päivän aika-askeleella suoritettuna RK4:llä. lf-pitka.dat sisältää saman mutta suoritettuna leapfrogilla. rk-lyhyt.dat sisältää 100000 vuoden integroinnin 1 päivän aika-askeleela suoritettuna RK4:llä. lf-lyhyt.dat sisältää saman mutta suoritettuna leapfrogilla.

2.3 a)

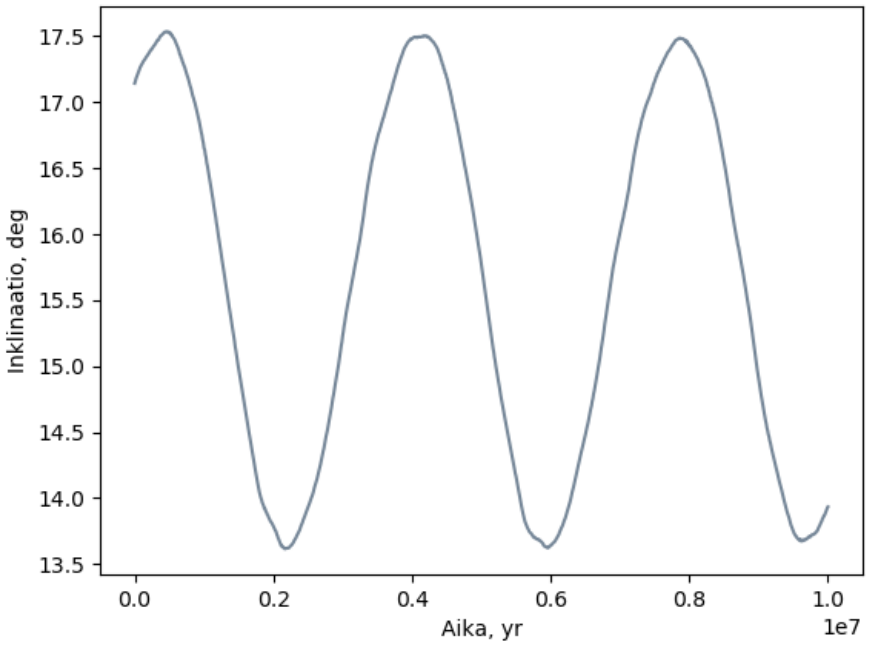
Alla olevista kuvista näkee Pluton eksentrisyyden ja inklinaation oskillaatiot. Ne näyttävät olevan tehävänannossa kerrotut 0,05 ja noin 4 astetta.



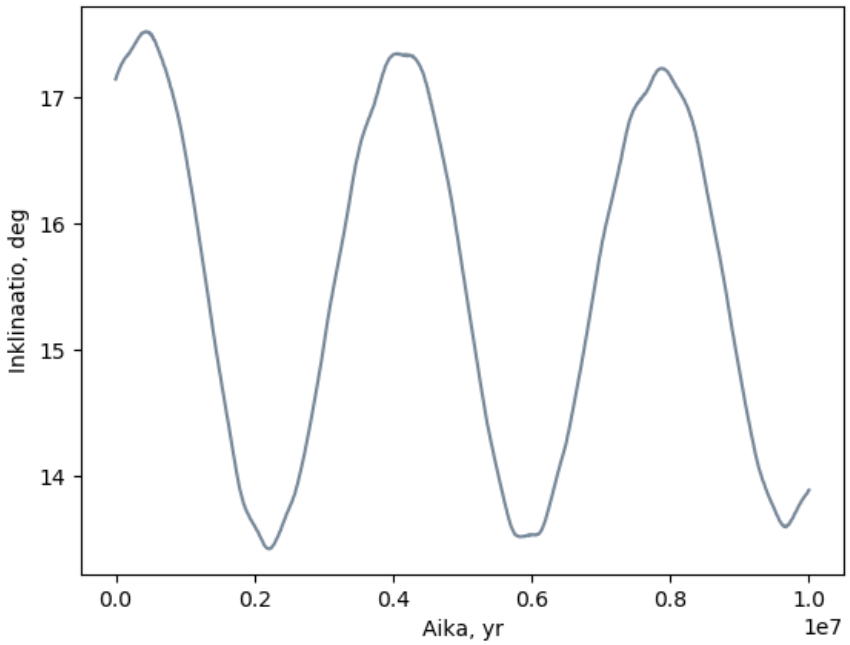
Kuva 3: RK4:llä saatu Pluton eksentrisyys.



Kuva 4: Leapfrogilla saatu Pluton eksentrisyys.

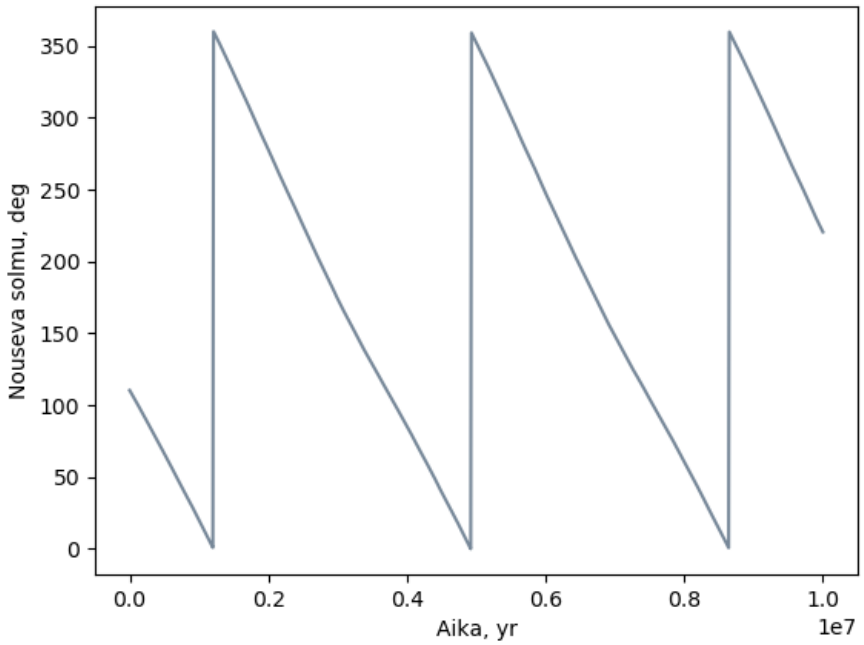


Kuva 5: RK4:llä saatu Pluton inklinaatio.

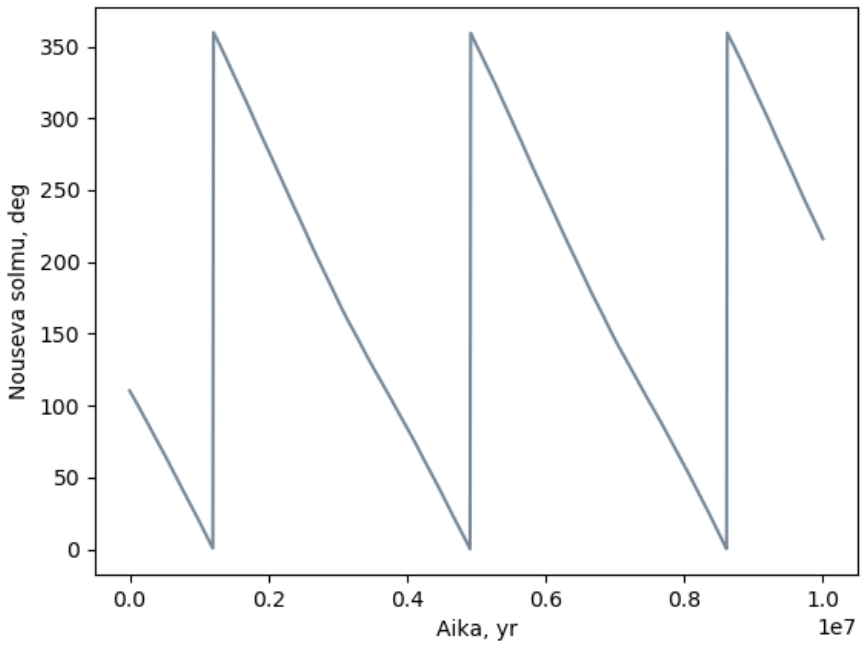


Kuva 6: Leapfrogilla saatu Pluton inklinaatio.

2.4 b)



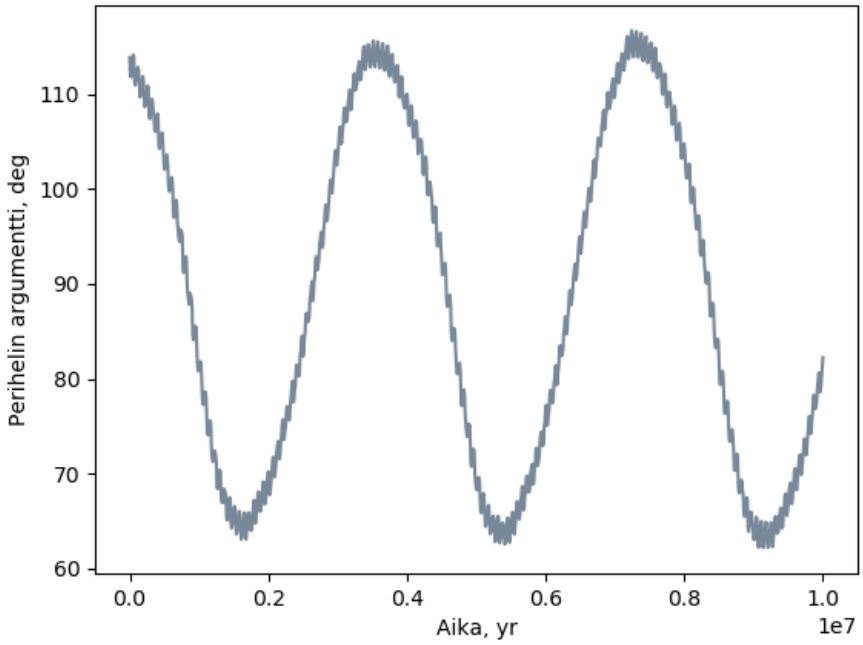
Kuva 7: RK4:llä saatu Pluton nouseva solmu.



Kuva 8: Leapfrogilla saatu Pluton nouseva solmu.

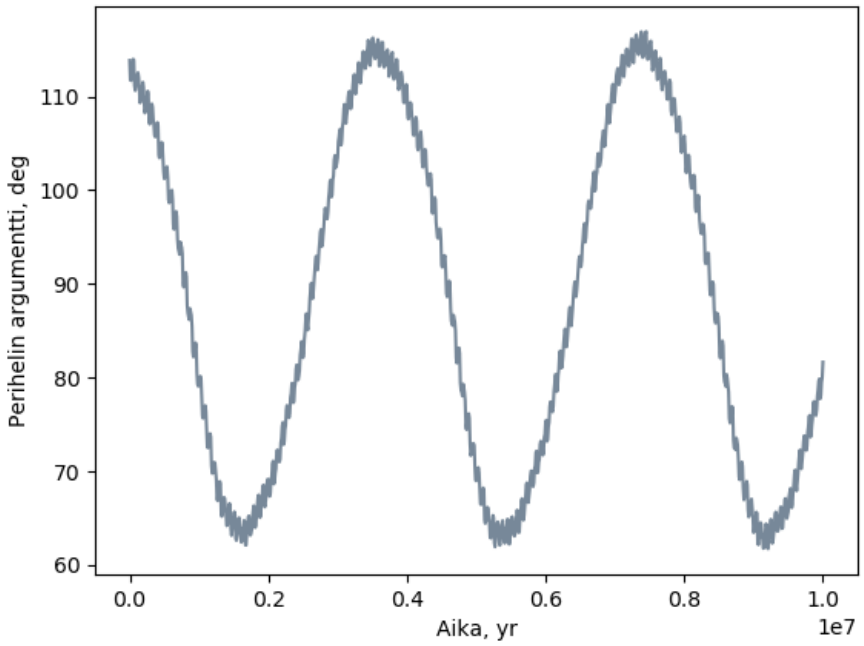
Kuvien sahalaitaisuus johtuu siitä että kulmat on rajattu välille 0-360 astetta. Laskemalla kulmakertoimen saamme prokessiolle RK4:llä arvoksi noin -1.67285702995e-06 deg/yr ja leapfrogilla -1.67679252461e-06 deg/yr.

2.5 c)



Kuva 9: RK4:llä saatu Pluton perihelin argumentti.

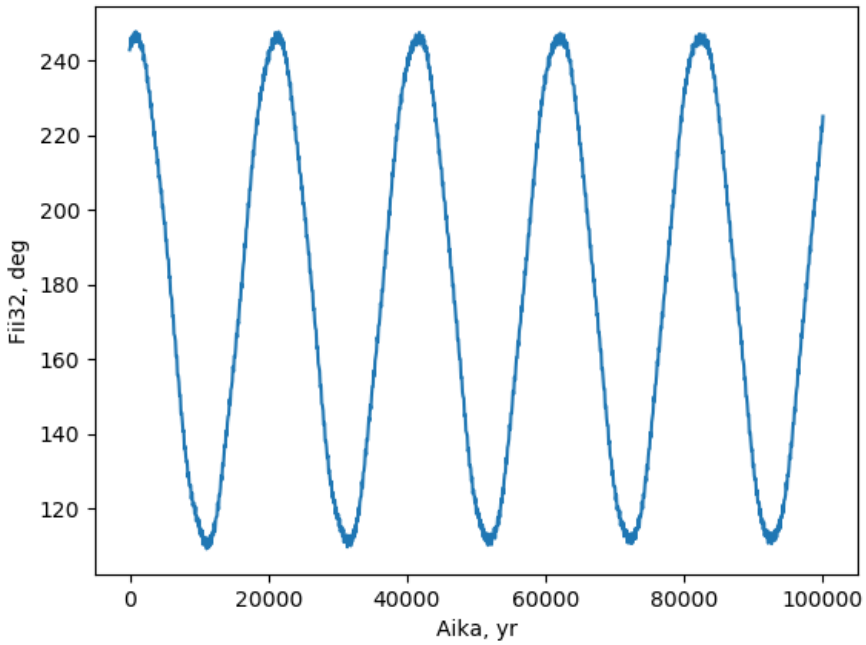
Approksimoimalla amplitudia ja periodia saamme niiden arvoksi noin 50 astetta ja noin 3,8e6 vuotta vastaavasti. Ne ovat sekä RK4:llä että leapfrogilla hyvin lähellä samat.



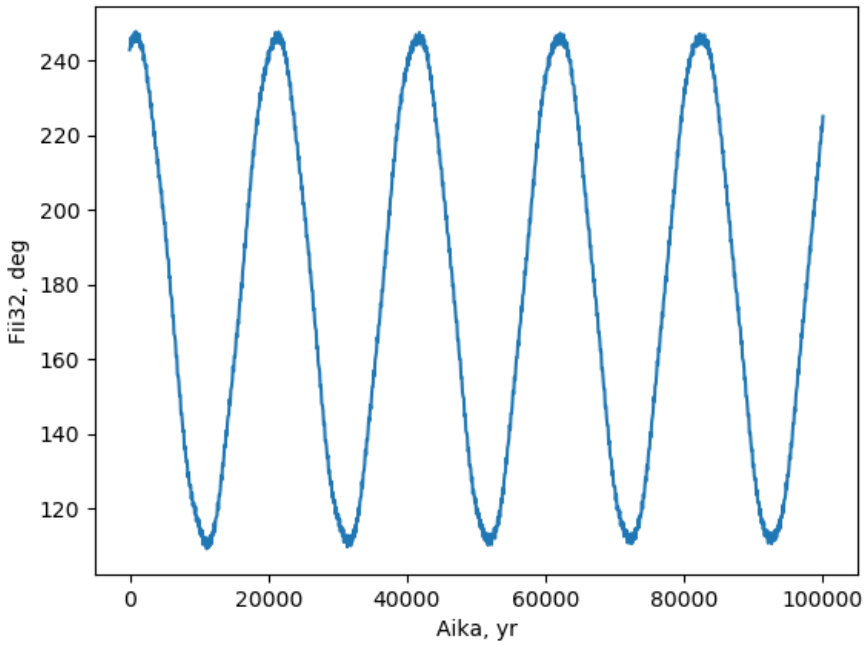
Kuva 10: Leapfrogilla saatu Pluton perihelin argumentti.

2.6 d)

RK4:llä ϕ_{32}:n keskiarvo on 177.86652703 astetta ja leapfrogilla 177.865867982 astetta. Molemmilla integraattoreilla huippuarvot ovat noin 246 asteen tienoilla, ja matalimmat arvot 110 asteen tienoilla. Eli amplitudi keskiarvon suhteen on molempiin suuntiin noin 68 astetta.



Kuva 11: RK4:llä saatu ϕ_{32}.



Kuva 12: Leapfrogilla saatu ϕ_{32}.

3 Johtopäätökset

Molemmat integraattorit suoriutuivat tehtävästä aika lailla täsmälleen samoin. Isoin ero integraattoreiden välillä koko projektissa oli varmaankin niiden ajamisessa kuluva aika. Leapfrog on huomattavasti nopeampi kuin RK4, eikä ainakaan tämän projektin mittakaavassa RK4:ssä ollut huomattavia etuja. Eli kaiken kaikkiaan ainakin tämän projektin osalta leapfrog on integraattoreista parempi.

On harmillista etten saanut omaa integraattoriani antamaan kunnollisia tuloksia. Vertasin koodiani sekä Annin että Markuksen kanssa, mutta en keksinyt mikä kriittinen ero omassa ohjelmassani on joka saa sen toimimaan väärin. Mutta onneksi Anni suostui jakamaan datansa jotta sain tehtävän kuitenkin loppujen lopuksi tehtyä. Ja vaikka en saanut integraattoriani toimimaan kunnola, projekti oli kaiken kaikkiaan mielenkiintoinen ja mukava.