

Satelliittien rataliikkeen ennustamisesta

Markus Hiltunen

Perustieteiden korkeakoulu

Kandidaatintyö

Espoo 03.6.2023

Vastuupettaja

Prof. Pauliina Ilmonen

Työn ohjaaja

Prof. Pauliina Ilmonen

Copyright © 2023 Markus Hiltunen

The document can be stored and made available to the public on the open internet pages of Aalto University.
All other rights are reserved.



Tekijä Markus Hiltunen

Työn nimi Satelliittien rataliikkeen ennustamisesta

Koulutusohjelma Teknistieteellinen kandidaattiohjelma

Pääaine Matematiikka ja systeemitieteet

Pääaineen koodi SCI3029

Vastuuopettaja Prof. Pauliina Ilmonen

Työn ohjaaja Prof. Pauliina Ilmonen

Päivämäärä 03.6.2023

Sivumäärä 27+5

Kieli Suomi

Tiivistelmä

Satelliitit näyttävät yhä suurempaa roolia modernin yhteiskunnan toiminnoissa, ja satelliittien tekninen hyödyntäminen edellyttää kykyä ennustaa niiden ratoja, ja siten sijaintia. Painovoiman ohjaamaa kiertorataliikettä on mahdollista käsitellä analyttisesti staattisina elliptisinä ratoina, nk. Keplerin ratoina, tekemällä voimakaita yksinkertaistavia oletuksia painovoimakentästä ja satelliittiin vaikuttavista voimista. Hienostuneempia ja tarkempia ennusteita voidaan saavuttaa joko lisäämällä Keplerin mekaniikkaan matemaattisesti kömpelöitä perturbaatioita, tai hylkäämällä analyttinen paradigma, ja siirtymällä painovoiman numeeriseen integrointiin.

Luomme tässä työssä katsauksen kolmeen varsin yksinkertaiseen numeeriseen integroinnin menetelmään: Eulerin metodi, Verlet-integraatio, sekä 4. asteen Runge-Kutta -metodi. Implementoimme nämä algoritmit, vertailemme niiden soveltuvuutta rataliikkeen numeeriseen integrointiin, sekä tarkastelemme kattavasti eroja niiden tarkkuudessa. Käytämme tähän tarkoitukseen sekä ideaalista gravitatiivista mallia, että hyvin tarkkaa ja realistista referenssidataa aurinkokunnan kappaleiden liikkeistä.

Johtava havaintomme on toistuva hierarkia algoritmien välillä: RK4 erottuu säännönmukaisesti menetelmistä tarkimpana, ja Eulerin metodi osoittautuu odotetusti käyttökelvottoman kehnoksi ratapropagointiin. Erot tarkkuudessa Verletin ja RK4:n välillä osoittautuvat maltillisiksi, ja RK4:n toisaalta raskaampi laskentakuorma voisi antaa aihetta kattavampaan hyötysuhdetarkasteluun.

Avainsanat numeerinen integrointi, satelliitit, ratamekaniikka



Author Markus Hiltunen

Title Concerning satellite orbit prediction

Degree programme Bachelor's Programme in Science and Technology

Major Mathematics and Systems Sciences

Code of major SCI3029

Teacher in charge Prof. Pauliina Ilmonen

Advisor Prof. Pauliina Ilmonen

Date 03.6.2023

Number of pages 27+5

Language Finnish

Abstract

Satellites play an increasingly important role in modern societies and economy, and their utilization necessitates the capability of making accurate predictions of satellite orbits, and positions. Orbital trajectories can be described analytically as static elliptic Keplerian orbits. This approach imposes severe simplifications on the gravitational field, and forces acting on the satellite. More sophisticated and precise predictions can be achieved through mathematically unwieldy perturbations on Keplerian orbits, or by abandoning the paradigm of analytical orbits altogether, and using numerical integration instead.

In this work, we analyze three numerical integration algorithms: Euler method, Verlet integration, and 4. order Runge-Kutta method. We implement these algorithms, compare their suitability to numerical integration of orbital movement, and make a thorough investigation of their differences in precision. For this purpose we introduce an idealized gravitational model, followed by a very detailed and realistic reference data set on the movements of major bodies in our solar system.

Our primary finding is a repeating hierarchy between the algorithms: RK4 consistently stands out as the most precise method of those analyzed, and the Euler method is established as inadequate for orbital propagation. The differences in precision between Verlet and RK4 are found to be moderate, and the heavier computational load of RK4 could justify a more involved efficiency analysis between the two.

Keywords Numerical integration, satellites, orbits

Sisällys

Tiivistelmä	3
Tiivistelmä (englanniksi)	4
Sisällys	5
1 Johdanto	7
2 Taustaa	7
3 Menetelmät	9
3.1 Eulerin metodi	9
3.2 Verlet integrointi	10
3.3 Runge-Kutta -metodi (4. aste)	11
3.4 Koeasetelmat	13
3.4.1 Klassinen satelliittirata	13
3.4.2 Maa-Kuu -järjestelmän auringonpimennykset	14
3.5 Tekninen toteutus	15
4 Tulokset	16
4.1 Klassinen satelliittirata	16
4.2 Maa-Kuu -järjestelmän auringonpimennykset	19
5 Tarkastelu	23
5.1 Päätelmät tarkastelluista menetelmistä	23

5.2 Aallonharjan tekniikka	23
A Integraattorin lähdekoodi	28

1 Johdanto

Satelliitit ovat ihmisen avaruuteen laukaisemia, ja Maan kiertoradalle asettamia laitteita, jotka näyttävät yhä suurempaa roolia yhteiskunnan normaalissa toiminnassa (EU, 2023). Planeetan – kuten Maan – kiertoradalla olo on luontaisesti hyvin dynaaminen tila, jossa kappaleet liikkuvat suurilla nopeuksilla planeetan pinnan suhteen.

Jotta satelliitteja voidaan hyödyntää maasta käsin, on niiden sijaintia, ja siten liikettä kyettävä ennustamaan. Onneksemme vapaiden kappaleiden liike avaruuden tyhjiössä noudattaa hyvin puhtaasti mekaniikan lakeja.

Painovoiman kaltaisen konservatiivisen voiman hallitsemalle liikkeelle on olemassa analyttinen ratkaisu, kun kappaleita on kaksi, ja massat ovat pistemäisiä (Karttunen ja Valtonen, 2006). Keplerin parametrit kuvaavat tällaisia ideaalisia ratoja (Curtis, 2005). Käytännön tilanteissa esimerkiksi satelliitteihin vaikuttaa havaittavasti myös Kuun ja Auringon painovoima, Maan painovoimakenttä ei ole pinnan läheisyydessä isotrooppinen, ja satelliiteilla on käytössään oma propulsio. Mm. näistä syistä satelliittien ratojen ennustamisessa on käytettävä numeerisen integroinnin menetelmiä. Nämä etenevät lähtötilanteesta diskreetin aika-askelin, ja mallintavat satelliittien liikettä siihen kullakin hetkellä kohdistuvan kiihtyvyyden kautta.

Painovoiman kuvaamiseen kykeneviä numeerisen integroinnin menetelmiä on lukusia erilaisia, ja niiden välillä on suuria eroja numeerisessa tarkkuudessa, toteutuksen kompleksisuudessa, tai kyvyssä kuvata epäjatkuvia ei-konservatiivisia voimia. Joillekin integrointimenetelmille voidaan todistaa houkuttelevia matemaattisia ominaisuuksia, kuten kineettisen energian säilyminen.

Tämän työn tarkoituksena on esitellä pieni joukko tunnettuja integrointimenetelmiä, demonstroida niiden toteutus, sekä vertailla niiden soveltuvuutta satelliittien rataliikkeen ennustamiseen järkeviä vertailukriteereitä käyttäen.

Luomme myös silmäyksen edistyneempiin tekniikoihin, ja tarkastelemme kvalitatiivisesti miten avaruushallintojen aallonharjan menetelmät eroavat siitä mitä tässä demonstroimme.

2 Taustaa

Painovoiman ohjaaman taivaankappaleiden liikkeen – taivaanmekaniikan – ennustaminen on eräs vanhimmista tieteenaloista. Kuten fysiikassa on tyypillistä, tässäkin on ollut löydettävissä yksinkertaistavia malleja ja teorioita, joitka kuvaavat luontoa

onnistuneesti, mutta vain varsin karkealla tarkkuudella. Ymmärryksen karttuessa on kyetty kehittämään toinen toistaan tarkempia, mutta myös monimutkaisempia ja hienosyisempiä teorioita, jotka yksinkertaisissa tapauksissa toki toistavat vanhojen teorioiden ennusteita.

Ensimmäinen kuvaus kiertoratamekaniikasta, jota voidaan pitää onnistuneena, on epäilemättä Johannes Keplerin (1571-1630) 1600-luvun alussa muotoilema elliptisten ratojen mekaniikka. On merkillepantavaa että hän kykeni tähän tuntematta Newtonin painovoimateoriaa, tai liikkeen lakeja, jotka löydettiin vasta puoli vuosisataa myöhemmin. Keplerin parametrit ovat kuitenkin yhä tänään käyttökelpoinen (ja käytetty) tapa ilmaista satelliittien ja taivaankappaleiden ratoja tarkkojen numeeristen ennusteiden ulkopuolella. ([Curtis, 2005](#))

Newtonin mekaniikan avulla Keplerin lait voidaan johtaa helposti yhdeltä istumalta. Näin saadaan kuitenkin kuvatuksi analyttisesti vain kahden kappaleen muodostaman systeemin liike. Tämä rajoite tunnetaan yleisesti nimellä 'kolmen kappaleen ongelma' ja nykyisin on jo osoitettu ettei analyttistä ratkaisua suljetussa muodossa ole. ([Karttunen ja Valtonen, 2006](#), s. 1–7)

Monimutkaisempien painovoimaympäristöjen kuvaamiseksi on kehitetty myös matemaattisesti kompleksisia variaatiolaskennan menetelmiä. Nämä pohjautuvat yleensä Hamiltonin mekaniikkaan, ja ilmaisevat esimerkiksi differentiaalisia perturbaatioita Keplerin rataparametreihin ([ESA-Space-Debris-Office, 2018](#), s. 7). Varsinkin 1900-luvun alussa ennen tietokoneiden kehitystä tarvittiin näitä analyttisesti monimutkaisia keinoja minimoimaan käsin suoritettavan laskennan määrää.

Nykyisin halutaan esimerkiksi satelliittien ratoja ennustettaessa ottaa huomioon tekijöitä kuten erillisistä malleista johdettu paikkakohtainen ionosfääriplasman tiheys, tai maan painovoimakentän äärimmäisen tarkkaan kartoitetut epätasaisuudet ([ESA-Space-Debris-Office, 2018](#), s. 7). Tällaisten sisällyttäminen klassisen variaatiolaskennan piiriin on mahdotonta, tai ainakin epätoivoista. Toisaalta tietokoneiden laskentateho on kehittynyt valtavasti, ja analyttisten menetelmien käsittely on vaivalloista, niinpä numeeriset menetelmät ovat vakiintuneet ratamekaniikan ennustamisen.

Numeerisen integroinnin piiriin kuuluu valtavasti erilaisia algoritmeja, joiden kompleksisuus ja tulosten laatu poikkeavat paljon toisistaan. Tässä työssä paneudumme varsin yksinkertaisiin numeerisen integroinnin tapoihin: Eulerin metodi, Verlet integrointi, sekä 4. kertaluokan Runge-Kutta -metodi (RK4). Esittelemme realistisen satelliitin ratamekaanisen lähtötilanteen, ja vertailemme eri metodien tuloksia integroidun ajan- ja aika-askeleen pituuden funktiona. Vertailun metriikkana käytämme muutoksia rataenergiassa – jonka säilymislakien mukaan tulisi olla vakio – sekä variointia Keplerin rataparametreissa. Esittelemme myös tapauksen JPL:n (*Jet Propulsion Laboratory*) tarjoamalla tunnetulla ratamekaanisella lähtö- ja lopputilanteella, ja selvitämme kykenevätkö tarkasteltavat algoritmit toistamaan näitä tuloksia.

3 Menetelmät

Tulemme käyttämään seuraavia matemaattisia merkintöjä, symboleja ja konventioita:

$\mathbf{r}$	paikkavektori ($\mathbf{r} \in \mathbb{R}^3$)
$\mathbf{v}$	nopeusvektori ($\mathbf{v} \in \mathbb{R}^3$)
$\mathbf{a}$	kiihtyvyysvektori ($\mathbf{a} \in \mathbb{R}^3$)
m	massa, pieni, ei omaa gravitaatiota
M	massa, gravitoiva
γ	Newtonin gravitaatiovakio ($6.67430(15) \times 10^{-11} \text{Nm}^2/\text{kg}^2$)
μ	Standardi gravitaatioparametri ($\mu = \gamma M$)
h	numeerisen integroinnin askelpituus
t	aika
R	etäisyys

3.1 Eulerin metodi

Eulerin metodi on kukaties kaikkein tunnetuin, ja varmasti yksinkertaisin numeerisen integroinnin algoritmi. Se sisältää vain yhden lineaarisen kiihtyvyys-evaluaation aika-askelta kohti, ei säilytä tietoa aiemmista aika-askeleista, ja voidaan kuvata oleellisesti kolmella ohjelmakoodirivillä. Eulerin metodia voidaan pitää Hamiltonin mekaniikan numeerisen integraattorin äärimmilleen redusoituna versiona ([Ascher ja L.R., 1997](#)). Se onkin tunnettu jo vuosisatoja, ja vaikka kantaakin *Leonhard Eulerin* nimeä, hän ei luultavasti ollut ensimmäinen matemaatikko tai fyysikko, joka ideaa on hyödyntänyt.

Olettakaamme y :n alkuarvo-ongelma

$$y(0) = y_0 \quad \text{ja} \quad f(t, y) = y'(t), \quad (1)$$

missä y_0 on vakio, $t \in \mathbb{R}$, $y(t) \in \mathbb{R}^m$ ja missä $y'(t) = \frac{\partial y(t)}{\partial t}$.

Eulerin metodilla voidaan nyt estimoida $y(t)$:n arvoa iteratiivisesti seuraavalla tavalla

$$y_{n+1} = y_n + hf(t_n, y_n),$$

jossa h on integroinnin askelpituus.

Eulerin metodi on lähes aksiomaattisen intuitiivinen tekniikka tällaisen alkuarvo-ongelman ratkaisuun. Se voidaan myös rigööristi johtaa monin eri tavoin. Hyödyllisimpänä voidaan pitää lähestymistapaa Taylorin sarjan kautta:

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

Taylorin sarjasta saadaan

$$y(t_0 + h) = y(t_0) + y'(t_0)h + \frac{1}{2}y''(t_0)h^2 + \mathcal{O}(h^3),$$

joten

$$y(t_0 + h) = y(t_0) + f(t_0, y_0)h + \mathcal{O}(h^2).$$

Kuten näemme, johto Taylorin sarjan kautta antaa suoraan myös estimaatin $\mathcal{O}(h^2)$ numeerisen integraatiovirheen kokoluokasta.

Klassisen ratamekaniikan sovelluksessa kuvaamme jäljitettävän satelliitin sijaintia $\mathbf{r}$. Newtonin mekaniikka muotoutuu alkuarvo-ongelmaksi (1) seuraavasti:

$$\begin{aligned} \mathbf{r} &= \mathbf{r}(t), & \dot{\mathbf{r}}(t) &= \mathbf{v}(t), \\ \dot{\mathbf{v}}(t) &= \mathbf{a}(t), & \mathbf{a}(t) &= \sum_{i=1}^{N_M} \frac{\gamma M_i}{\|\mathbf{r}_{M_i} - \mathbf{r}\|^2}, \\ \mathbf{r}(0) &= \mathbf{r}_0, \quad \text{sekä} & \mathbf{v}(0) &= \mathbf{v}_0. \end{aligned}$$

Jossa termit M_i kuvaavat gravitoivia massoja. Pistenotaatio (kuten $\dot{\mathbf{v}}$) kuvaa tässä työssä aikaderivaattaa $\frac{\partial \mathbf{v}}{\partial t}$. Koska analyttinen kuvaus saavutetaan paikan toiselle derivaatalle eli painovoimakiihtyvyydelle, joudumme ylläpitämään kahta kytkeytyntä integraatioketjua, ja paikkavektori seuraa siis kaksoisintegraation tuotteena. Tämä on merkillepantavaa siksikin että numeeriset kaksoisintegraalit divergoivat helposti. Onneksi ratamekaniikka ei luonteeltaan ole kaoottisimpia systeemejä. Eulerin metodilla saamme siis seuraavanlaiset päivitysalgoritmit:

$$\begin{aligned} \mathbf{r}_{n+1} &= \mathbf{r}_n + h\mathbf{v}_n, \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + h\mathbf{a}_n, \text{ ja} \\ \mathbf{a}_{n+1} &= \sum_{i=1}^{N_M} \frac{\gamma M_i}{\|\mathbf{r}_{M_i, n} - \mathbf{r}_n\|^2} \end{aligned}$$

3.2 Verlet integrointi

Verlet-integrointi on hieman monimutkaisempi integraatiomenetelmä, jonka *Loup Verlet* esitteli 60-luvulla nykyisen numeerisen laskennan kontekstissa ([Verlet, 1967](#)). Menetelmää on kuitenkin ilmeisesti käytetty jo 1700-luvulta alkaen. *P.H. Cowell* ja *A.C.C. Crommelin* laskivat sillä Halley'n komeetan radan vuonna 1909, ja vuonna 1906 *Carl Størmer* käytti sitä mallintaessaan varattujen hiukkasten liikettä magneettikentässä. Toisinaan kirjallisuudessa käytetäänkin termiä Verlet-Störmer metodi.

Poiketen Eulerin metodista, Verlet integraatio on 'symplektinen' ([Dixon ja Reich, 2014](#)). Symplektisellä integraattorilla tarkoitetaan algoritmia, joka tietyillä ehdoilla

säilyttää numeerisesti mallinnettavan järjestelmän Hamiltonin energian. Tyypillisesti tämä tarkoittaa potentiaali – liikemäärä -analogista Hamiltonia, kuten:

$$\dot{p} = -\frac{\partial H}{\partial q} \quad \text{ja} \quad \dot{q} = \frac{\partial H}{\partial p}.$$

Symplektisyyden osoittaminen on varsin monimutkainen prosessi, emmekä tässä työssä paneudu siihen. Klassinen Verlet-integraation päivitysaskel Newtonin mekaniikalle on ensimmäiselle aikapisteelle $n = 1$:

$$\mathbf{r}_1 = \mathbf{r}_0 + h\mathbf{v}_0 + \frac{1}{2}\mathbf{a}_0h^2,$$

ja lopuille $n \in \{2, 3, 4, \dots\}$:

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \frac{1}{2}\mathbf{a}_nh^2.$$

Huomionarvoista on, ettei päivitysalgoritmi sinänsä tarvitse kappaleiden nopeuksia $\mathbf{v}$ mihinkään. Nopeuden ja paikan sijaan on pidettävä kirjaa kahden peräkkäisen aika-askeleen sijainneista, joten tietokonetoteutuksessa muistinkulutus on oleellisesti sama kuin Eulerin metodissa. On myös olemassa laajalti käytetty variantti: Nopeus-Verlet (*velocity Verlet*), jossa päivitysaskel kuuluu seuraavasti:

$$\begin{aligned} \mathbf{r}_{n+1} &= \mathbf{r}_n + h\mathbf{v}_n + \frac{1}{2}\mathbf{a}_nh^2, \quad \text{ja} \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{\mathbf{a}_n + \mathbf{a}_{n+1}}{2}h. \end{aligned}$$

Verlet-integraatio on suosittu menetelmä erityisesti molekulaarisissa simulaatioissa, plasmahiukkasten mallinnuksessa, ja vastaavissa tilanteissa. Se soveltuu näihin erinomaisesti, sillä symplektisyys palvelee hiukkassimulaatioille keskeistä vaatimusta pitää systeemin kokonaisenergia muuttumattomana. Verlet-integraatio on tarkkuutensa nähden myös hyvin yksinkertainen toteuttaa, nopea laskea, ja kuluttaa hyvin vähän muistia. Astronomisissa ratalaskuissa Verlet ei ole tyypillisesti käytetty, sillä symplektisyys ei – kukaties intuition vastaisesti – ole tae hyvästä tarkkuudesta, vaan Hamiltonin noudattamisesta. Yksittäisen integraatioaskeleen diskreetointivirheiden kontribuutiot kokonaisenergiaan eivät siis ole toisistaan riippumattomia, vaan sulkevat toisiaan pois huomattavasti tarkemmin kuin magnitudiltaan samankokoiset satunnaisjakautuneet virheet. (Quin ja Guan, 2008)

Toisaalta, ei-symplektiset integraattorit voivat saavuttaa samalla askelpituudella huomattavasti tarkempia tuloksia, mikäli arviointikriteerinä käytetään absoluuttista sijainnin virhettä $\|\mathbf{r}_n - \mathbf{r}(n \times h)\|$.

3.3 Runge-Kutta -metodi (4. aste)

Runge-Kutta -metodit (tästädes RK), muodostavat implisiittisten ja eksplisiittisten integraattorialgoritmien perheen, joita tyypillisesti käytetään nimenomaisesti

ajassa etenevien differentiaalisten alkuarvo-ongelmien ratkaisuun. RK-metodien perhe jakautuu astelukuihin algoritmisyklin sisältämien laskentapisteen määrän mukaan. Ensimmäisen asteen RK-metodi on ekvivalentti Eulerin metodin kanssa. Epäilemättä useimpia perheen jäseniä on käytetty eri yhteyksissä historian varrella, mutta yhtenäistävän teorian kehittivät *Carl Runge*, (Runge, 1894), ja *Wilhelm Kutta*, (Kutta, 1896), 1900-luvun alussa. Lukuisista asteluvuista ehdottomasti tunnetuin ja käytetyin on 4. asteen RK-metodi, eli ytimekkäästi RK4. Aiemmin esiteltyyn alkuarvo-ongelmaan (1) sovellettuna RK4 saa seuraavanlaisen muodon:

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4), \quad \text{sekä} \\ t_{n+1} &= t_n + h. \end{aligned}$$

Painokertoimet k_1, k_2, k_3, k_4 määritetään eksplisiittisesti näin:

$$\begin{aligned} k_1 &= f(t_n, y_n), & k_2 &= f(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}), \\ k_3 &= f(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}), & k_4 &= f(t_n + h, y_n + hk_3), \end{aligned}$$

jossa $f(t, y) = \frac{\partial y}{\partial t}(t)$.

Ratamekaniikan kontekstissa paikkavektorin $\mathbf{r}$ ja nopeusvektorin $\mathbf{v}$ päivityksille on muotoiltava omat rinnakkaiset RK4-syklinsä, jotka kytkeytyvät aikaderivaattojensa kautta, sillä

$$f_{\mathbf{r}}(t_n) = \frac{\partial \mathbf{r}}{\partial t}(t_n) = \mathbf{v}_n \quad \text{ja} \quad (2)$$

$$f_{\mathbf{v}}(t_n) = \frac{\partial \mathbf{v}}{\partial t}(t_n) = \mathbf{a}_n = \sum_{i=1}^{N_M} \frac{\gamma M_i}{\|\mathbf{r}_{M_i, n} - \mathbf{r}_n\|^2}. \quad (3)$$

On syytä huomata, että muista tarkastelluista menetelmistä poiketen RK4:n k_i -kertoimien evaluointi tapahtuu pisteissä, jotka eivät ole kuulu simulaation aikapisteisiin $n \in \{1, 2, 3, \dots\}$. Niinpä ratamekaaninen RK4-päivitys täytyy suorittaa vaiheittain, siten että kappaleiden sijainneille ja nopeuksille päivitetään väliaikaiset $\mathbf{r}_*$ ja $\mathbf{v}_*$ arvot. Väliaikainen $\mathbf{a}_*$ -kiihtyvyys evaluoidaan $\mathbf{r}_*$ -sijainteja käyttäen yllä kuvatuun summakaavan (3) mukaan. RK4-päivitys $n \rightarrow (n+1)$ muodostuu siis seuraavista

askelista:

$$\begin{aligned}
k_{\mathbf{r}_n,1} &= \mathbf{v}_n, & k_{\mathbf{v}_n,1} &= \mathbf{a}_n, \\
\mathbf{r}_* &\leftarrow \mathbf{r}_n + \frac{h}{2}k_{\mathbf{r}_n,1}, & \mathbf{v}_* &\leftarrow \mathbf{v}_n + \frac{h}{2}k_{\mathbf{v}_n,1}, \\
k_{\mathbf{r}_n,2} &= \mathbf{v}_*, & k_{\mathbf{v}_n,2} &= \mathbf{a}_*, \\
\mathbf{r}_* &\leftarrow \mathbf{r}_n + \frac{h}{2}k_{\mathbf{r}_n,2}, & \mathbf{v}_* &\leftarrow \mathbf{v}_n + \frac{h}{2}k_{\mathbf{v}_n,2}, \\
k_{\mathbf{r}_n,3} &= \mathbf{v}_*, & k_{\mathbf{v}_n,3} &= \mathbf{a}_*, \\
\mathbf{r}_* &\leftarrow \mathbf{r}_n + hk_{\mathbf{r}_n,3}, & \mathbf{v}_* &\leftarrow \mathbf{v}_n + hk_{\mathbf{v}_n,3}, \\
k_{\mathbf{r}_n,4} &= \mathbf{v}_*, & k_{\mathbf{v}_n,4} &= \mathbf{a}_*.
\end{aligned}$$

ja lopulta,

$$\begin{aligned}
\mathbf{r}_{n+1} &= \mathbf{r}_n + \frac{1}{6}h(k_{\mathbf{r}_n,1} + 2k_{\mathbf{r}_n,2} + 2k_{\mathbf{r}_n,3} + k_{\mathbf{r}_n,4}) \quad \text{ja} \\
\mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{1}{6}h(k_{\mathbf{v}_n,1} + 2k_{\mathbf{v}_n,2} + 2k_{\mathbf{v}_n,3} + k_{\mathbf{v}_n,4}).
\end{aligned}$$

3.4 Koeasetelmat

Arvioimme tarkasteltavia integrointimenetelmiä käyttäen kahta eri koeasetelmaa. Ensimmäisessä seuraamme satelliittia klassisella matalalla kiertoradalla Maan ympärillä. Käsitlemme maan painovoimakenttää pistemassan kenttänä, emmekä ota huomioon muita satelliitteihin vaikuttavia voimia. Toisessa koeasetelmassa simuloimme Maan ja Kuun liikettä aurinkokunnassa tämän vuoden alusta ja arvioimme, kuinka hyvin eri menetelmät onnistuvat ennustamaan auringonpimennyksiä seuraaville vuosille.

3.4.1 Klassinen satelliittirata

Tätä koeasetelmaa varten arvomme satunnaisen tilavektoriparin $(\mathbf{r}, \mathbf{v})$ Maata kevyen elliptisellä radalla kiertävälle satelliitille, siten että radan apoapsis on 700km ja periapsis 500km. Koska koeasetelmassamme ei muita voimia ole, satelliitin radan tulisi säilyä muuttumattomana. Voimme laskea alkutilan viitearvot rataenergialle E_o , sekä Keplerin rataelementeille $\mathcal{K} = e, a, i, \Omega, \omega, \theta$. Tämä on hyvin vanha, mutta yhä käytetty tapa ilmaista staattiseen keskuskappaleeseen sidottu elliptinen rata kuudella termillä ([Curtis, 2005](#)):

e :	eksentrisyys,	a :	isoakselin puolikas,
i :	inklinaatio,	Ω :	nousevan solmun pituus,
ω :	perihelin argumentti,	θ :	luonnollinen anomalia,

jossa ensimmäiset viisi kuvaavat radan muodon, ja θ määrittää kiertolaisen paikan radalla ajanhetkellä t (epookkissa t_0). Ulkoisten häiriöiden puuttuessa muiden kuin θ tulisi pysyä vakioina.

Vertaamme integrointimenetelmiä toisiinsa käyttämällä metriikkana näiden suureiden $(E_o, e, a, i, \Omega, \omega)$ variointia suhteessa viitearvoihin ajan t funktiona, sekä vakioisen integrointiajan T_I jälkeen aika-askeleen pituuden h funktiona. Määritämme seuraavat residuaalit:

1. *Energiavirhe* $\mathfrak{R}_{E_o}(t, h) = \frac{|E_{o,h,sim}(t) - E_{o,ref}(t)|}{|E_{o,ref}(t)|}$ seuraa satelliitin rataenergian $E_{o,h,sim}(t)$ ja viiterataenergian $E_{o,ref}(t)$ erotuksen magnitudia ajanhetkellä t , siten että aika-askeleen pituus h on kullekin integraatioajolle vakio. Tässä, ja muissa yhteyksissä, *sim*-indeksi viittaa integraattorilla simuloituun tulokseen, ja *ref*-indeksi joko analyttisesti laskettuun tai taulukoista saatuun viitearvoon, eli referenssiin.
2. *Elementtinvirhe* $\mathfrak{R}_{Kepler}(t, h) = \sum_{i=0}^5 \left| \frac{\mathcal{K}_{i,h,sim}(t) - \mathcal{K}_{i,ref}(t)}{\mathcal{K}_{i,ref}(t)} \right|$ seuraa satelliitin Keplerin statististen rataelementtien $\mathcal{K}_{i,h,sim}(t)$ normalisoidun virheen summaa suhteessa elementtien viitearvoihin $\mathcal{K}_{i,ref}(t)$ ajanhetkellä t . Jälleen pidämme aika-askeleen pituuden h kullekin integraatioajolle vakiona.

3.4.2 Maa-Kuu -järjestelmän auringonpimennykset

Tätä koeasetelmaa varten hankimme totuudenmukaiset lähtöasemat Maalle, Kuulle, Auringolle, sekä muutamalle läheiselle planeetalle. Onneksemme JPL julkaisee säännöllisesti hyvin tarkkoja taulukoita taivaankappaleiden sijainneista menneisyydessä, nyt, ja tulevaisuudessa. Käytämme uusinta DE441 -efemeridiä (Park et al., 2021). Valitsemme integraation alkupisteeksi vuoden 2023 alun, eli 2023.01.01 00:00 (UTC), ja integroimme kaksi vuotta ajanhetkeen 2025.01.01 00:00 (UTC). Tällä aikavälillä meidän tulisi kohdata neljä auringonpimennystä (Espanak, 2023). Kykenemme siis validoimaan tuloksemme hyvin robustisti: Toisaalta JPL:n efemerididatan tulee toisintaa täsmälleen NASA:n taulukoimat pimennykset. Näin validoimme taivaankappaleiden asemista, epookkiajasta ja maan kääntökulmasta maantieteelliset koordinaatit laskevan matematiikan. Toisaalta oman rataintegraatiomme tulisi toisintaa efemeridin mukaiset asemat. Näin validoimme itse integraation fysikaalisuuden. Mikäli kykenemme toistamaan usean pimennyksen maantieteellisen kulun yksin taivaankappaleiden tilavektoreista, on käsissämme erittäin voimakas tulos rataintegraatiomme oikeellisuuden puolesta.

Ensimmäisestä koeasetelmasta poiketen tässä voimme verrata saamiamme tuloksia oikeiksi tietämiimme verrokkituloksiin. Sen sijaan kappaleiden rataenergian ja Keplerin elementtien varioinnit tulevat olemaan huomattavasti vähäisempiä, sillä

etenemme vain joitain ratakiirroksia ja käytämme lyhyitä aika-askelia. Määritämme nyt seuraavat residuaalit:

1. *Sijaintivirhe* $\mathfrak{R}_r(t, h) = \frac{\|\mathbf{r}_{h, \text{sim}}^{\oplus \rightarrow \mathbb{C}}(t) - \mathbf{r}_{\text{ref}}^{\oplus \rightarrow \mathbb{C}}(t)\|}{\|\mathbf{r}_{\text{ref}}^{\oplus \rightarrow \mathbb{C}}(t)\|}$ seuraa maa-kuu -vektorien $\mathbf{r}^{\oplus \rightarrow \mathbb{C}}$ erotuksen L_2 -normia integroidun tilan ja efemeridin mukaisen tilan välillä ajanhetkellä t . Aika-askeleen pituus h on kullekin integraatioajolle vakioinen parametri. Tämä residuaali soveltuu paitsi kvalitatiiviseen, myös kvantitatiiviseen vertailuun.
2. *Pimennyskäyrät* $\{\mathcal{P}_i^{\odot \rightarrow \oplus}(t)\}$ ovat tarkasteltujen pimennysten joukkoa vastaava käyräjoukko. Pimennyskäyrä seuraa Auringosta $\odot$ kuun $\mathbb{C}$ massakeskipisteen kautta piirretyn janan osoittamaa pistettä $\mathcal{P}$ Maan $\oplus$ pinnalla ajan t funktiona pimennyksen i aikana. Nämä soveltuvat kvalitatiiviseen graafiseen vertailuun.
3. *Pimennysvirheet* $\{\mathfrak{R}_{\Omega, i}(t)\} = \{\|\Omega_{\text{sim}, i}(\alpha t + \beta) - \Omega_{\text{ref}, i}(t)\|\}$, $\Omega \in \mathbb{R}^2$, ovat tarkasteltujen pimennysten i joukkoa vastaava käyräjoukko. Niissä $\Omega_i(t)$ vastaa pimennyskäyrän $\mathcal{P}_i^{\odot \rightarrow \oplus}(t)$ geograafisten korkeus- ja pituuspiirien 2-vektoria ajanhetkellä t . Residuaali muodostetaan integroidun simulaation mukaisen- ja efemeridin mukaisen tuloksen vektorierotuksena. Pimennysvirhe on määritelty sillä aikavälillä $t \in [t_{\text{alku}, i, \text{ref}}; t_{\text{loppu}, i, \text{ref}}]$ jonka ajan kuun massakeskipisteen varjo leikkaa Maan kanssa efemeridien sijaintidatan mukaan. α ja β skaalaavat aikaparametrin täsmäämään integraation kanssa, siten että $t_{\text{alku}, i, \text{sim}} = \alpha t_{\text{alku}, i, \text{ref}} + \beta$ ja $t_{\text{loppu}, i, \text{sim}} = \alpha t_{\text{loppu}, i, \text{ref}} + \beta$. Nämä residuaalit soveltuvat paitsi kvalitatiiviseen, myös kvantitatiiviseen vertailuun.

3.5 Tekninen toteutus

Käytämme työn toteutukseen pääasiallisesti *Python3*-ohjelmointikieltä ([Python Software Foundation, 2023](#)). Pythonin ympärille on viime vuosikymmeninä muodostunut huomattavan kattava matemaattisten luonnontieteiden ja tietojenkäsittelyn työkalujen ekosysteemi. Puhtaan numeriiikan kirjastot, kuten *numpy* ja *numba*, ovat muodostuneet standardinkaltaisiksi välineiksi, ja niitä käytetään mitä vaativimmissa tieteellisissä prosesseissa. ([NumFOCUS-Inc, 2023](#))

Jotta efemeridien mukaisilla rataparametreilla suoritettut laskelmat ovat uudelleen verrattavissa efemeridin dataan muilla ajanhetkillä, on prosessissa kiinnitettävä erityistä huomiota oikein aikastandardien käyttöön. DE-441 ilmaisee sijainnit nk. TDB-ajassa, eli dynaamisessa barysentrisessä ajassa, joka käy maanpäällistä kelloa. Tämä on kompromissistandardi puhtaasti maanpäällisten TT ja UTC aikastandardien ja ratamekaniikassa käytettävän TCB-ajan välillä. TCB-aika (barysentrinen

koordinaattiaika) on määritelty sellaisen havaitsijan ajaksi, joka sijaitsee aurinkokunnan painovoimakentän ulkopuolella, ja liikkuu yhdensuuntaisesti aurinkokunnan massakeskipisteen mukana, eli on levossa sen suhteen. TCB-ajan sekunti on hieman maanpäällistä sekuntia nopeampi, ja TCB-kello edistääkin n. puoli sekuntia vuodessa maanpäälliseen kelloon verrattuna. Tämä johtuu yleisen suhteellisuusteorian mukaisesta ajan hidastumisesta maan painovoimakentässä. Lisäksi TDB- TCB- (sekä TT- ja TAI-) ajat ovat lineaarisia, eli niihin ei lisätä karkaussekunteja, toisin kuin siviilikäytössä yleiseen UTC-aikaan. UTC-aika sisältää nykyisin jo 37 karkaussekuntia. Tästä ja relativistisista efekteistä johtuen UTC- ja TCB-ajat ovat nykyisin erkaantuneet toisistaan jo yli 90 sekuntia. (Kaplan, 2005)

Aikastandardien käsittelyyn käytämme tunnettua tähtitieteeseen suunnattua *astropy*-kirjastoa, joka kykenee suorittamaan muunnokset yllä mainittujen standardien välillä vaivatta. (Astropy-Project, 2023)

Liitteeseen A on koottu oleelliset osat laskelmien lähdekoodia: itse integraattori, sekä geometrisia työkaluja auringonpimennysten määrittämiseen.

4 Tulokset

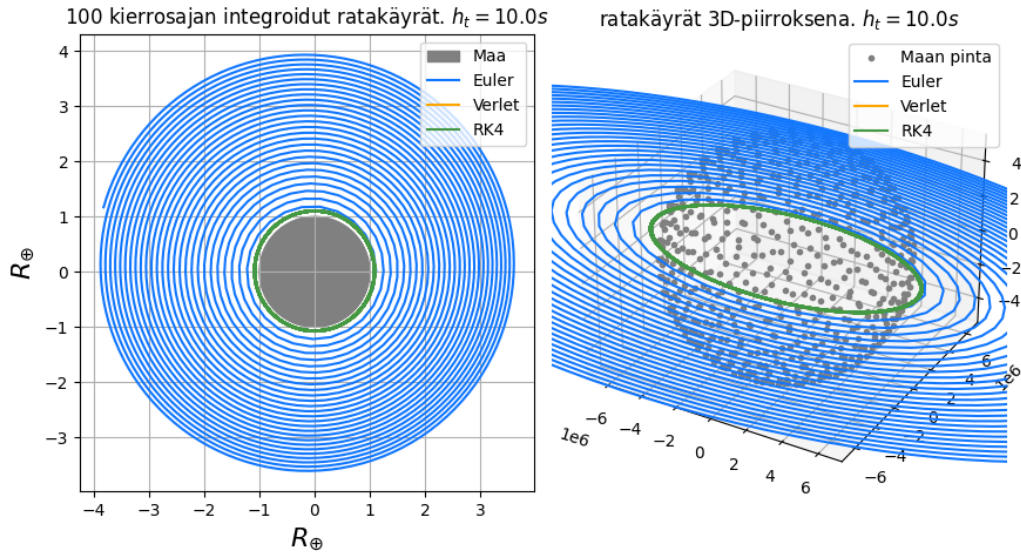
4.1 Klassinen satelliittirata

Valitsimme satunnaisgeneroidut alkuhetken tilavektorit seuraavien rataparametrien ja keskuskappaleen gravitaatiopotentiaalin mukaan:

perigee	500km
apogee	700km
$\mathbf{r}_0$	[100347.421809, -6601334.8286, 1903404.8112]m
$\mathbf{v}_0$	[7631.12335654, -107.09643757, -773.74134445] $\frac{m}{s}$
$\mu_\oplus$	$3.986004418 * 10^{14} \frac{m^3}{s^2}$

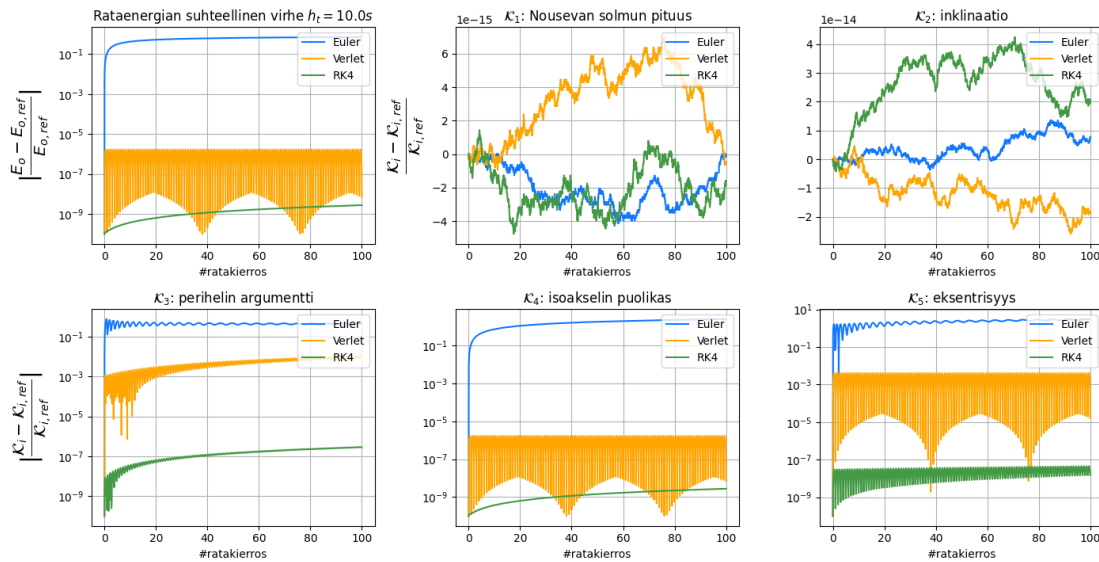
Koska tässä koeasetelmassa ei ole keskuskappaleen painovoiman lisäksi läsnä muita voimia, tulisi kaikkien rataparametrien, sekä satelliitin energian säilyä muuttumattomina. Saadaksemme yleiskuvan eri algoritmien kyvystä pysyä radallaan, piirrämmme ensin 100 ratakierron jälkeisen tilanteen:

Piirroksessa 1 näemme graafisesti kuvattuna satelliitin sijainnin ensimmäisen sadan ratakierron aikana, kullakin integrointimetodilla. Piirrosta hallitsee Eulerin metodin käyrä, jolla jokainen ratakierron loittonee keskuskappaleesta. Tämä kasvava virhe on seurausta Eulerin metodin äärimmäisestä yksinkertaisuudesta. Verlet ja RK4



Kuva 1: 100 ensimmäistä ratakierrosta 2D ja 3D tasossa, aika-askeleen arvolla $h_t = 10s$.

erottuvat heikosti piirroksessa: niiden radat piirtyvät muuttumattomina päällekkäin maan pinnan läheisyyteen, mikä onkin toivottava lopputulos.

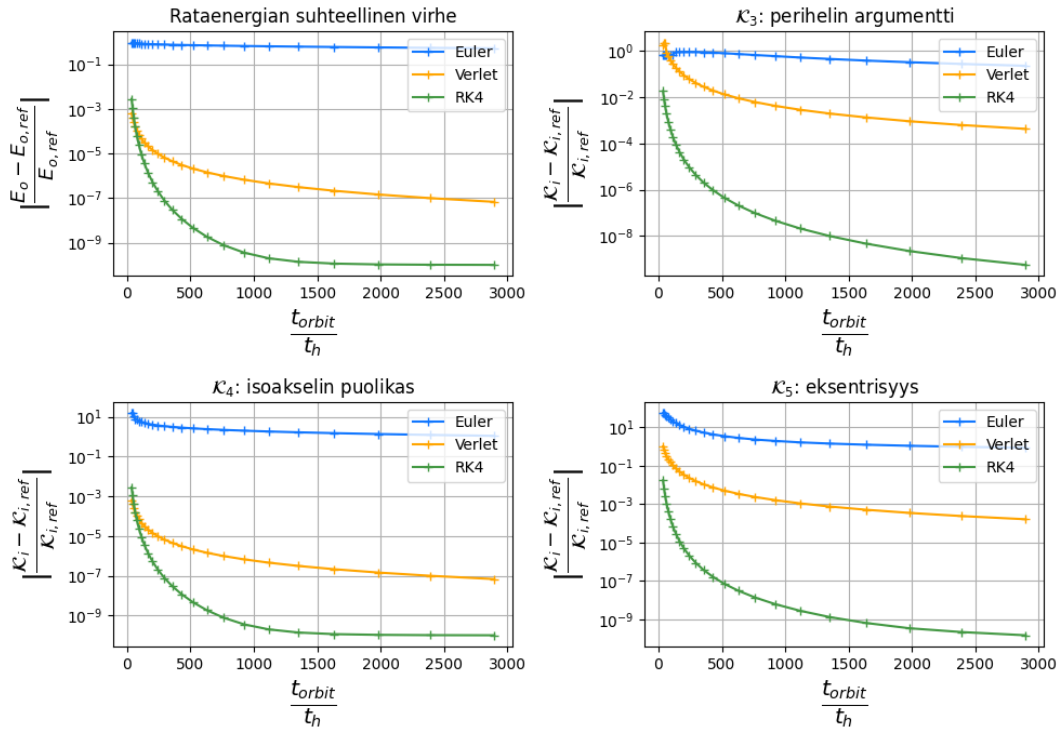


Kuva 2: Rataenergian ja Keplerin parametrien kehitys ajan suhteen, aika-askeleen arvolla $h_t = 10s$.

Kuvaajiin piirroksessa 2 on koottu normalisoidut virheet rataenergiassa, sekä kaikissa viidessä Keplerin rataparametrissa. Huom: Käytämme logaritmistista asteikkoa, jotta hyvin eri kokoluokissa olevat virheet ovat vertailtavissa. Jo rataenergiassa havaitsemme selkeän hierarkian integrointialgoritmien välillä: Eulerin metodin virhe kasvaa

hyvin pian yli viisi kertaluokkaa muita suuremmaksi. Verlet ilmentää kirjallisuuden mukaan symplektisille algoritmeille tyypillistä käytöstä: Energiavirhe oskilloi voimakkaasti, muttei kasva 10^{-6} tasoa suuremmaksi. Algoritmi kykenee pysyttelemään suhteellisen lähellä Hamiltonia, vaikka spatiaalisesti satelliitin sijainti ei välttämättä ole yhtä tarkka. Verletin ja RK4:n välillä näyttäytyy myös hierarkia, jossa RK4 saavuttaa konsistentisti suuremman tarkkuuden kuin Verlet, ja siis parhaan tarkkuuden tarkastelluista algoritmeista.

Keplerin parametreissa $\mathcal{K}_1$, *Nousevan solmun pituus*, sekä $\mathcal{K}_2$, *inklinaatio*, kaikkien algoritmien virheet koostuvat skaalan $\mathcal{O}(10^{-15})$ kohinasta, mikä vastaa käytettyjen 64-bittisten liukulukujen maksimitarkkuutta. Nämä parametrit määrittävät ratatason asennon avaruudessa. Piirroksen 1 3D-käyrä tukee tätä havaintoa: Eulerin metodin suurikokoinen virhekin karkaa ulospäin säilyttäen täsmälleen alkuperäisen radan tason.



Kuva 3: Rataenergian ja Keplerin virhe aika-askeleen t_h suhteen, 100 ratakierron jälkeen

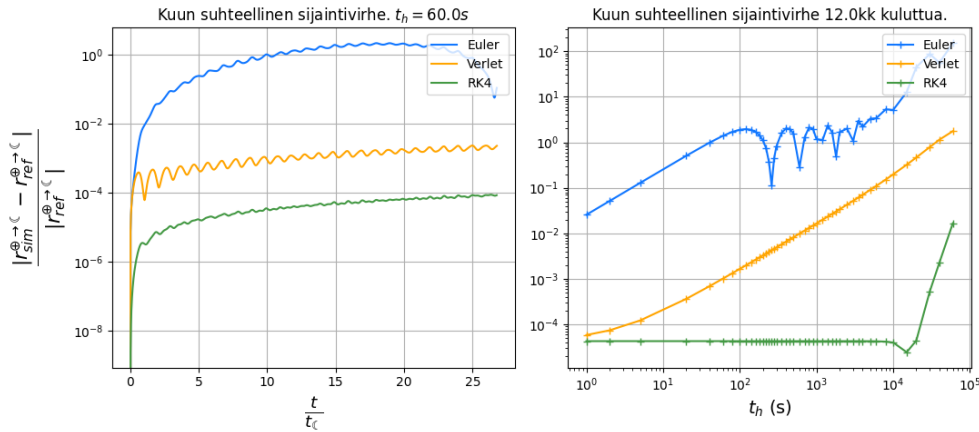
Piirroksen 3 kuvaajiin on koottu yllä mielenkiintoiseksi havaittujen muuttujien E_o , $\mathcal{K}_3$, $\mathcal{K}_4$, $\mathcal{K}_5$ normalisoidut virheet aika-askeltasossa integrointiajalla 100 ratakierrasta. Aika-askeltaso on normalisoitu koeasetelman karakteristisen aikaskaalan – kierrosajan – suhteen. Akselin numeerinen arvo kuvaa kuinka moneen osaan aika-askel jakaa karakteristisen aikaskaalan. Tässä tapauksessa kierrosaika $t_{orbit} = 96\text{min } 32\text{s}$.

Piirroksen 3 kuvaajat vahvistavat odotetun: Pienempi aika-askel johtaa suurempaan

tarkkuuteen kaikilla algoritmeilla, ja aikatasokuvaajissa käytetyn arvon $t_h = 10s$ tuottama hierarkia toistuu käytännössä kaikkialla aika-askeltasossa ($10s \approx 600 \frac{t_{orbit}}{t_h}$ aika-askeltasossa). Tämän koeasetelman tulosten pohjalta tässä tarkastelluista menetelmistä RK4-algoritmia voidaan pitää yksiselitteisesti tarkimpana rataintegraation metodina.

4.2 Maa-Kuu -järjestelmän auringonpimennykset

Valitsemme integraation alkupisteeksi vuoden 2023 alun, eli 2023.01.01 00:00:00 (UTC), ja integroimme kaksi vuotta ajanhetkeen 2025.01.01 00:00:00 (UTC). Julkisesti lähteistä tiedämme, että tänä aikana tulee tapahtumaan neljä auringonpimennystä, päivämäärinä 2023.04.20, 2023.10.14, 2024.04.08 ja 2024.10.02. ([Espenak, 2023](#))

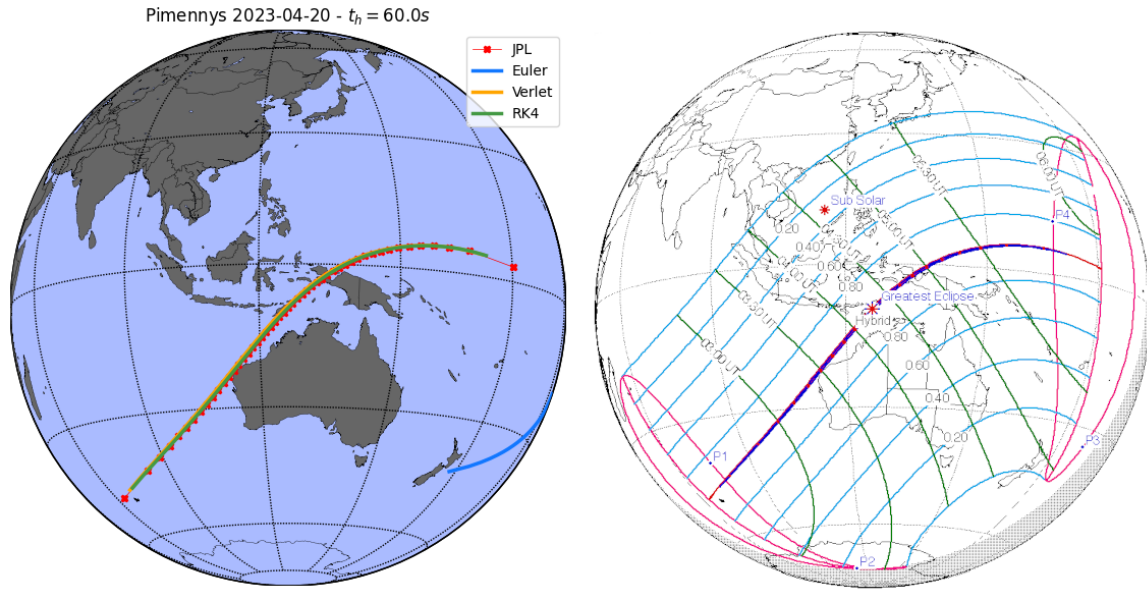


Kuva 4: Kuun sijaintivirhe aikatasossa, ja aika-askeltasossa

Piirroksen 4 on koottu sijaintivirheen $\mathfrak{R}_r(t, h)$ kuvaajat aikatasossa, sekä aika-askeltasossa. Koeasetelmassa 1 havaittu hierarkia algoritmien välillä toistuu: Eulerin metodi erottuu selvästi keinoimpana, siinä missä Verlet ja RK4 saavuttavat edes jollain määritelmällä tarkkoja tuloksia. RK4 näyttäytyy jälleen varsin yksiselitteisesti tarkimpana. On huomionarvoista, että Eulerin metodin virhe saavuttaa skaalan $\mathcal{O}(1)$, eli Kuun sijainti integraatioissa voi olla täysin toisella suunnalla suhteessa referenssiin. Tämä selittääkin virheessä tapahtuvan laskun, ja uudelleen nousun: Eulerin metodilla Kuun rata kiertyy auki kuten piirroksessa 1, ja sen sijainti radallaan siis jätättää. Lopulta Kuu sijaitsee Maan vastakkaisella puolella, jolloin sijaintivirhe saavuttaa karkeasti sanoen arvon $2|\mathbf{r}_{ref}^{\oplus}|$, minkä jälkeen virhe jälleen pienenee, kunnes Kuu on Eulerin integraatiossa kokonaisen ratakierron jäljessä referenssitulosta, mutta geometrisesti lähempänä kuin aiemmin.

Toinen mainitsemisen arvoinen huomio liittyy RK4-algoritmin virheeseen aika-askeltasossa: Normalisoitu virhe 12kk kohdalla ei laske $\mathcal{O}(10^{-4.5})$ -tason alle aika-askelta lyhentämällä, (oletuksella että JPL:n tulokset ovat tässä skaalassa luotetta-

via). Todennäköisesti RK4:n tarkkuus noudattaa määritettyä fysiikkaa kyllä paranee, mutta määritetty fysiikka ei ole täsmälleen sama kuin efemeridin. Olemme mitä luultavimmin joko saavuttaneet relativististen korjausten tai vuoromaan skaalan Maa-Kuu -järjestelmässä (Eshagh ja Alamdari, 2006). Tätä hypoteesia tukee havainto, että myös Verlet-integrointi näyttää lähestyvän asympotoottisesti samaa virherajaa aika-askeleen kutistuessa.



Kuva 5: Pimennyksen 2023-04-20 geograafinen käyrä, vasemmalla sekä integroituna että efemeridistä piirrettynä, ja oikealla *Goddard Spaceflight Center*:n mukaan (Espanak, 2009)

Piirroksessa 5 näemme integrointivälin ensimmäisen auringonpimennyksen. Punaisella merkitty käyrä on piirretty suoraan efemerididatan mukaan. Se täsmää erittäin tarkasti refrenssipiirrosta, ja numeerisesti integroimamme tulokset asettuvat piirtotarkkuuden rajoissa täsmälleen oikealle paikalleen. Newtonin mekaniikan skaalassa integraattorin fysiikan oikeellisuudelle ei käytännössä jää epäilyksen sijaa. Aiempi havaintomme Eulerin metodin kehnoudesta suhteessa kahteen muuhun algoritmiin toistuu jälleen, mikä ei tässä vaiheessa enää yllätä lainkaan.

Pimennyksen pituuspiirien laskentaan on jo mainitun matematiikan lisäksi tunnettava Maan kääntökulmaparametrit, nk. *ERA* -vakiot. Näille pätee:

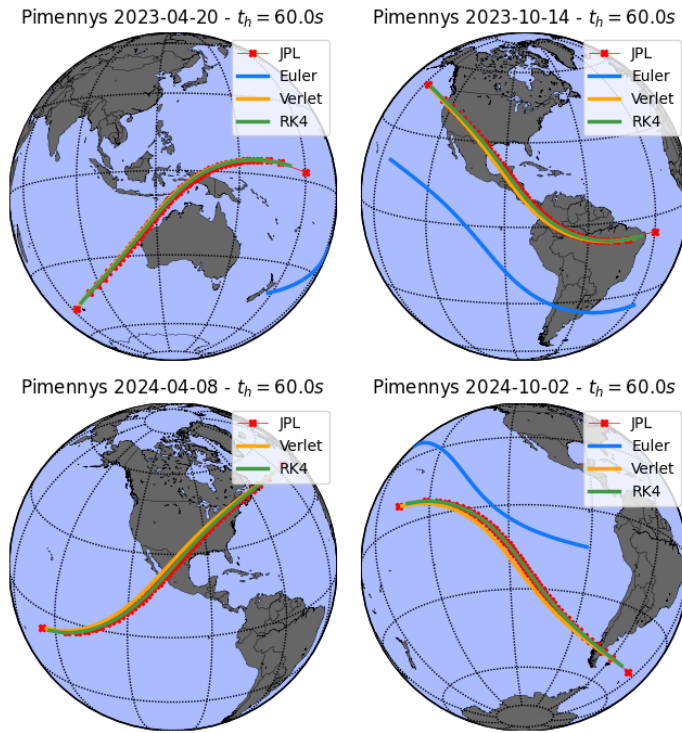
$$\begin{aligned} ERA_1 &= 0.7790572732640 \text{ rad}, & ERA_2 &= 1.00273781191135448 \frac{\text{rad}}{\text{jd}}, \\ \theta(\text{jd}_{2000}) &= 2\pi(ERA_1 + ERA_1 \text{jd}_{2000}), & \text{jd}_{2000} &= \text{jd} - 2451545.0. \end{aligned}$$

Tässä jd_{2000} on Julian-2000 päivännumero, eli Juliaaninen UT1-päivännumero -2451545.0: aikaparametri, joka saa arvon 0.0 meridiaanin 0° keskipäivän hetkellä 2000.01.01

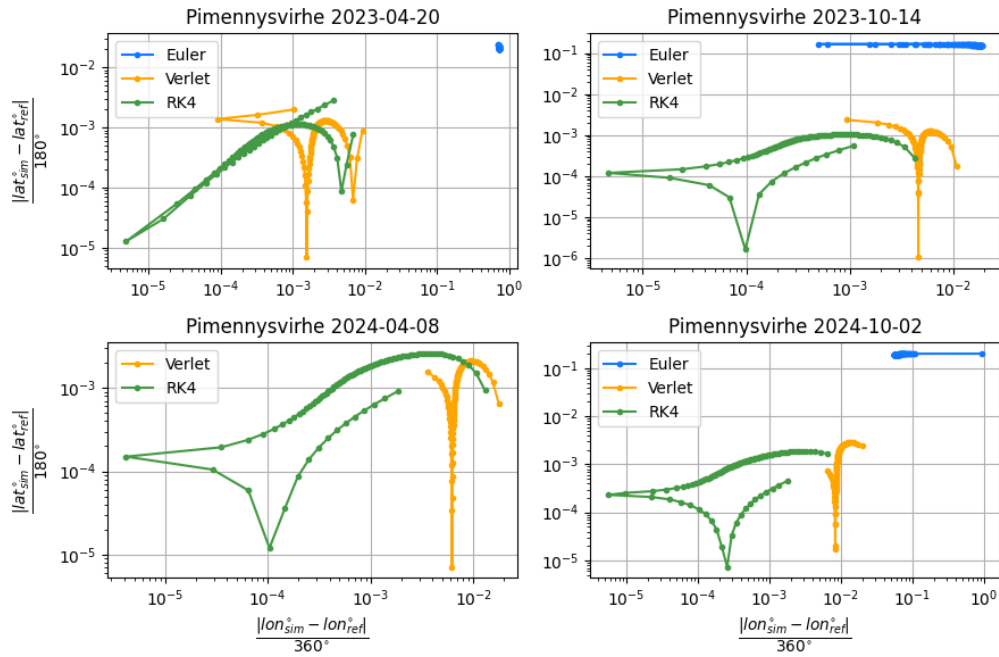
12:00. $\theta(\text{jd}_{2000})$ on meridiaanin 0° kääntökulma ICRS-koordinaattijärjestelmässä, jota mm. JPL:n efemeridit noudattavat ([Kaplan, 2005](#)). Maan kääntökulma θ siis ilmaisee 0-meridiaanin kulmaa x-akseliin koordinaatistossa, jonka kulma on vakio suhteessa kaukasiin tähtitaivaan kohteisiin.

Piirrokseen [6](#) on koottu kaikki tässä käsitellyllä integrointivälillä tapahtuvat pimennykset. Havaitsemme, että RK4 toistaa referenssidatan piirtotarkkuuden rajoissa täysin. Verlet alkaa viimeisissä kuvissa hieman viettää maantieteellisesti länteen päin. Verlet ilmeisesti jättää hivenen, jolloin Maan pyörimisliike siirtää pimennyskäyriä. Eulerin metodi ei edes tuota 2024.04.08 vastaavaa pimennystä. Tämä on analyysitekkinen ongelma, sillä piirtorutiini lukee pimennykset samoiksi vain mikäli hetki jolloin kuun varjo koskettaa maata ei eroa efemeridien mukaisesta pimennyksestä 24 tuntia enempää.

Pimennyskäyrien maantieteellisten koordinaattien normalisoidut virheet on koottu piirroksen [7](#) kuvaajiin.



Kuva 6: Integrintivälin 2023.01.01 - 2025.01.01 kaikki auringonpimennykset.



Kuva 7: Integrintivälin 2023.01.01 - 2025.01.01 pimennysten maantieteellisten koordinaattien virheet.

5 Tarkastelu

5.1 Päätelmät tarkastelluista menetelmistä

4. Asteen Runge-Kutta metodi osoittautui kaikissa koeasetelmissa tarkimmaksi algoritmiksi kolmesta tarkastellusta. Kuitenkin Verlet-integrointi näyttäisi pystyvän useimmissa tilanteissa samaan tarkkuuteen lyhyempää aika-askelta käyttäen. Algoritmien laskentakuorman arviointi rajattiin tietoisesti tämän työn ulkopuolelle, mutta karkeat mittaukset nyt ajettujen simulaatioiden aikana osoittivat seuraavaa:

Taulukko 4:

algoritmi	2 kappaletta: askelta/s	8 kappaletta: askelta/s
Euler	700 000	8460
Verlet	670 000	8220
RK4	175 000	6120

Laskentakuorma kasvaa gravitatiivisten kappaleiden määrän mukaan $\mathcal{O}(n^2)$, mutta kuten taulukon 4 arvoista näemme, muutkin tekijät vaikuttavat algoritmien nopeuteen. On täysin mahdollista, että Verlet voi oikein valitulla aika-askelpituudella yltää RK4:n tarkkuuteen lyhyemmässä laskenta-ajassa. Nopeusvertailu on kuitenkin aivan oma tietojenkäsittelytekniinen aihepiirinsä, sillä kompleksisempien algoritmien kohdalla edes optimaalisen implementaation määrittäminen ei ole lainkaan yksinkertaista. Tässä työssä käytimme kiihdytykseen *Python3*-ohjelmointikielen *numba*-kirjastoa, jolla voimme päästä suhteellisen pienellä vaivalla lähelle matalan tason kielten suorituskykyä. Kaikki laskelmat tätä työtä varten suoritettiin *AMD Ryzen 7 5800H*-ytimellä. (AMD, 2023)

On merkillepantavaa, että monasti vasta-alkajien valintana pidetty RK4 saavuttaa vaivatta tarkkuuden, jolla kuvaan astuu yksinkertaisen Newtonin mekaniikan ulkopuolisia ilmiöitä, ilmeisesti joko relativistisia efektejä, tai vuoromaan kaltaiset painovoimapotentiaalin anisotropiat Maa-Kuu -järjestelmässä.

5.2 Aallonharjan tekniikka

Kuten aiemmissa kappaleissa totesimme, tässä työssä käsitellyt tekniikat eivät edusta alan viimeisimpiä tai parhaimpia metodeja, vaan hyvin klassisia lähestymistapoja; jokaista tiedetään käytetyn yli vuosisata sitten. Tarkastelkaamme hieman miten alan eturintaman toimijat käsittelevät ratamekaanisia laskuja, ja mitä tekijöitä nykyisin otetaan huomioon.

Yleisesti voidaan todeta, että tähtitieteilijöiden lisäksi satelliittien ja taivaankappaleiden ratojen ennustaminen on tarpeellista kaikille tahoille, jotka operoivat laitteita maan kiertoradalla, syvemmällä avaruudessa, tai korkeilla ballistisilla kaarilla. Tämä kattaa käytännössä valtiolliset avaruusinstituutiot, satelliitteja omistavat ja ohjaavat yritykset ja valtiot, sekä tietyt asevoimat. (EU, 2023)

Karkeasti ratalaskennassa on erotettavissa kaksi toisistaan poikkeavaa toimintaympäristöä: Maan lähellä sijaitsevat- ja ulkoavaruuden kohteet.

Maan läheisyydessä kappaleiden sijaintia voidaan mitata suhteellisen usein, niitä voidaan usein ohjata, ja niihin kohdistuvat voimat ja vuorovaikutukset ovat potentiaalisesti erittäin kompleksisia ja joskus hankalasti ennustettavia. Tässä työssä oletimme Maan painovoimakentän yksinkertaisesti Newtonisen pistemassan isotrooppiseksi kentäksi. Todellisuudessa ensinnäkin maan pyörimisliikkeestä seuraava litistyminen ellipsoidiksi aiheuttaa matalilla radoilla merkittävän poikkeaman ideaalikenttään (Pavlis et al., 2012). Tämän lisäksi kallioperän paikallinen koostumus, laattatektoniikka, Kuun indusoima vuoromaa (maankuoren itsensä muovautuminen vuoroveden tapaan), vuorovesi, ja korkeampien kertaluokkien hyrräefektit aikaansaavat kompleksisia muutoksia maan potentiaalikenttään, jotka ovat riittävän suuria vaikuttamaan kiertoratoihin geostationaarisen radan alapuolella. Tätä hienosyistä kokonaiskenttää ei voida johtaa analyttisesti, vaan se on kartoitettu paitsi maan pinnalla tehtävin mittauksin, nykyään erittäin tarkkoilla interferometrisatelliittimissioilla. Lopputuotteena tämä geodesiaksi kutsuttava tieteenala kykenee tuottamaan potentiaalikarttoja, joissa kokonaiskenttä kuvataan palloharmonisia Laplacen kantafunktioita käyttäen alle kilometrin tarkkuudella. Tällainen kartta vaatii joitakin miljoonia kantafunktioiden painokertoimia (Pavlis et al., 2012). Ratamekaniikan kontekstissa näistä ensimmäiset kymmenisen tuhatta ovat tarpeellisia.

Painovoimakentän epätasaisuudet tasoittuvat nopeasti etäisyyden kasvaessa, mikä rajaa potentiaalikartan tarkkuusvaatimuksia korkeammilla radoilla. Matalilla radoilla puolestaan ilmakehän kaasusta avaruuteen kurottava ionosfääriplasma tihenee, ja hienosyisimmät painovoimakentän epätasaisuudet hukkuvat plasman aerodynaamiseen vastukseen. Satelliittien matalalla kokema ionosfäärivastus on vain rajallisesti ennustettavissa, sillä auringon aktiivisuus vaikuttaa voimakkaasti ionosfäärin kokoon ja plasman tiheyteen. Tunnetaan tapauksia, joissa rakettilaukaisun aikaan tapahtunut aurinkopurkaus on arvaamatta puhaltanut ionosfäärin monta kertaa odotettua suuremmaksi, ja laukaisuvälineen matalalle radalle jättämät satelliitit eivät kyenneet omin moottoreineen vastustamaan kasvanutta plasmakitkaa, vaan putosivat takaisin ilmakehään (Fang et al., 2022). Näiden kaoottisten tekijöiden takia erittäin matalalla lentävien satelliittien ratoja ei voida edes teoriassa ennustaa tarkasti kovin pitkälle tulevaisuuteen.

Maan lähiavaruudessa rataliikkeen ennustamisen vaikeudet liittyvät ylläkuvattuihin tekijöihin, ja erittäin hienostuneista moderneista integraatiomenetelmistä ei juuri ole lisähyötyä. Niinpä vakavaotteisimmatkin toimijat tyytyvät mitä ilmeisimmin

esimerkiksi 8. kertaluokan Runge-Kutta -metodeihin, jotka muistuttavat pitkälti tässä työssä käsiteltyjä algoritmeja. (ESA, 2023)

Ulkoavaruuden kohteet eli muut suuret taivaankappaleet, asteroidit, ja kaukaiset avaruusluotaimet kiinnostavat yksinomaan valtiollisia avaruusinstituutioita. Näiden ratoja pyritään – mutta myös voidaan – ennustaa hyvin pitkälle tulevaisuuteen, sillä merkityksellisiä kitkavoimia ei ole, ja valtavat etäisyydet mahdollistavat kappaleiden käsittelyn pistemassoina.

Tällä alalla NASA:n JPL on ehdottomasti merkittävin, ja monessa mielessä ainoa relevantti toimija. JPL julkaiseekin säännöllisesti tähtitieteessä standardiksi muodostunutta efemeridiä merkittävimpien taivaankappaleiden sijainneista nyt, menneisyydessä ja tulevaisuudessa (Park et al., 2021). Modernit efemeridit kattavat Maan ja sisempien planeettojen liikkeitä kymmenien metrien tarkkuudella nyt ja kilometrien tarkkuudella sadoiksi vuosiksi eteenpäin. Lisäksi JPL:n julkisesti saatavilla oleva HO-RIZONS -tietokanta kattaa satojen tuhansien asteroidien, komeettojen, luotaimien, ja muiden kappaleiden ratatietoja.

Siinä missä efemerididata on hyvin kattavasti julkista tietoa, näiden tuotteiden laskentametodit eivät ole. JPL:n tieteelliset artikkelit modernien taulukoiden teko-prosessista kuvailevat yksityiskohtaisesti laskuissa käytettävää lähdeaineistoa, ja sen epätarkkuuksia, mutta jättävät ilmeisen tietoisesti mainitsematta keskeiset integrointimenetelmät (Park et al., 2021). Prosessi nojaa tutkilla ja luotaimien radiolähettilä hankittuihin etäisyysmittauksiin, ja näiden kokeiden aikaleimoihin. Tämä aineisto yhdistetään taivaankappaleiden liikkeitä alkutilanteesta mallintamaan integraattoriin. Etäisyysmittauksien erot koetulosten ja mallin välillä kullakin ajanhetkellä tuottavat joukon residuaaleja, joiden painotettua summaa minimoimalla voidaan päästä todellisuutta parhaiten vastaaviin taivaankappaleiden sijainteihin. Luultavasti prosessissa käytetään jotakin hyvin monen askeleen integraattoria, kuten Gauss-Jackson -ennustus-korjaus -algoritmia, tai jotakin laajan Adams -integraattoriperheen menetelmää.

Viitteet

AMD. Ryzen 7 5800h, 2023. URL <https://www.amd.com/en/product/10821>.

U.M. Ascher ja Petzold L.R. Computer methods for ordinary differential equations and differential-algebraic equations, 1997.

Astropy-Project. Astropy library, 2023. URL <https://www.astropy.org/>.

H. Curtis. *Orbital Mechanics for Engineering Students*. Elsevier Butterworth-Heinemann, Norfolk, King's Lynn, United Kingdom, 2 edition, 2005.

- M. Dixon ja S Reich. Symplectic time-stepping for particle methods. *GAMM-Mitt*, 27(1):9–24, 2014.
- ESA. Godot tool, 2023. URL <https://godot.io.esa.int/docs/1.2.0/>.
- ESA-Space-Debris-Office. Esa re-entry prediction front-end user guide, 2018.
- M. Eshagh ja N.M. Alamdari. The effect of solid tide in geopotential field of an elastic and inelastic earth. *Journal of the Earth Space Physics*, 32:1–9, 2006.
- F. Espenak. Hybrid solar eclipse of 2023 apr 20, 2009. URL <https://eclipse.gsfc.nasa.gov/SEplot/SEplot2001/SE2023Apr20H.GIF>. [Verkkodokumentti, vierailtu 14.02.2023. Lisenssi: PD-USGOV-NASA, <https://www.nasa.gov/multimedia/guidelines/index.html>].
- F. Espenak. Solar eclipses: 2021 - 2030, 2023. URL <https://eclipse.gsfc.nasa.gov/SEdecade/SEdecade2021.html>.
- EU. European union space strategy for security and defence, 2023. URL https://defence-industry-space.ec.europa.eu/system/files/2023-03/EU%20SSSD%20factsheet_1.pdf.
- T. Fang, A. Kubaryk, D. Goldstein, Z. Li, T. Fuller-Rowell, G. Millward, H.J. Singer, R. Steenburgh, S. Westerman, ja E. Babcock. Space weather environment during the spacex starlink satellite loss in february 2022. *Space weather*, 20:1–14, 2022.
- H.G. Kaplan. The iau resolutions on astronomical reference systems, time scales, and earth rotation models, 2005. URL https://spsweb.fltops.jpl.nasa.gov/portaldataops/mpg/MPG_Docs/Source%20Docs/USNO_Circular_5.1.pdf.
- H. Karttunen ja M. Valtonen. *The Three-Body Problem*. Cambridge university press, Cambridge, United Kingdom, 1 edition, 2006.
- W. Kutta. Beitrag zur näherungsweise integration totaler differentialgleichungen., 1896.
- NumFOCUS-Inc. Numpy in eht, 2023. URL <https://numpy.org/case-studies/blackhole-image/>.
- R.S. Park, W.M. Folkner, Williams J.G., ja Boggs D.H. The jpl planetary and lunar ephemerides de440 and de441. *The Astronomical Journal*, 161(105):1–15, 2021.
- N.K. Pavlis, S.A. Holmes, S.C. Kenyon, ja J.K. Factor. The development and evaluation of the earth gravitational model 2008 (egm2008). *Journal of Geophysical Research*, 117(B04406):1–38, 2012.
- Python-Software-Foundation. Python3, 2023. URL <https://www.python.org/>.
- H. Quin ja X. Guan. Variational symplectic integrator for long-time simulations of the guiding-center motion of charged particles in general magnetic fields, 2008.

- C. Runge. Ueber die numerische auflösung von differentialgleichungen., 1894.
- L. Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159:98–103, 1967.

A Integraattorin lähdekoodi

Työ kokonaisuudessaan käsittää useita eri lähdekooditiedostoja ja muutamia tuhansia koodirivejä, joista valtaosa liittyy graafien piirtoon, laskentatulosten tallentamiseen ja valmiiden tulosten käsittelyyn piirtoa varten. Alle on liitetty lähdekoodit integraattorille, sekä muutamille olennaisimmille matemaattiselle työkalulle.

```
import numpy as np
from numba import float64, int64
from numba import types, typed, njit
from numba.experimental import jitclass

G_GRAVITY = 6.6743015e-11

float_array2D = float64[:, :]
float_array1D = float64[:]

@njit(cache=True)
def npv3(a,b,c):
    return np.array((a,b,c), dtype=np.float64)

@njit(cache=True)
def unitize(v):
    return v / np.linalg.norm(v, 2)

o_object_spec = [
    ('ID', int64),
    ('m', float64),
    ('mu', float64),
    ('I', float_array2D),

    ('r', float_array2D),
    ('k_r', float_array2D),
    ('v', float_array2D),
    ('k_v', float_array2D),
    ('acc', float_array2D),
    ('k_acc', float_array2D),

    ('r_arr', types.ListType(float_array1D)),
    ('v_arr', types.ListType(float_array1D)),
    ('acc_arr', types.ListType(float_array1D)),
    ('T_arr', types.ListType(types.float64))
]
@jitclass(o_object_spec)
class O_object:
    def __init__(self, m, r, v, I=None, mu=None):
        self.ID = np.random.randint(0, 10000000000)
        self.m = m
        self.mu = m * G_GRAVITY if (mu is None) else mu
        self.I = np.identity(3) if (I is None) else I
```

```

self.r = np.zeros((2,3), dtype=np.float64)
self.r[0] = r.copy()
self.k_r = np.zeros((5,3), dtype=np.float64)
self.v = np.zeros((2,3), dtype=np.float64)
self.v[0] = v.copy()
self.k_v = np.zeros((5,3), dtype=np.float64)
self.acc = np.zeros((2,3), dtype=np.float64)
self.k_acc = np.zeros((5,3), dtype=np.float64)

self.r_arr = typed.List.empty_list(float_array1D)
self.v_arr = typed.List.empty_list(float_array1D)
self.acc_arr = typed.List.empty_list(float_array1D)
self.T_arr = typed.List.empty_list(types.float64)

def records_as_numpy_arrays(self):
    r_arr = np.array(self.r_arr)
    v_arr = np.array(self.v_arr)
    acc_arr = np.array(self.acc_arr)
    T_arr = np.array(self.T_arr)
    return r_arr, v_arr, acc_arr, T_arr

o_object_instance = O_object.class_type.instance_type
sim_specc = [
    ('objects', types.ListType(o_object_instance)),
    ('T', float64)
]
@jitclass(sim_specc)
class SIM:
    def __init__(self):
        self.T = 0.0
        self.objects = typed.List.empty_list(o_object_instance)

    def add_obj(self, O):
        self.objects.append(O)

    def record_obj_states(self):
        for obj in self.objects:
            assert obj.r[0].shape == (3,)
            obj.r_arr.append(obj.r[0].copy())
            obj.v_arr.append(obj.v[0].copy())
            obj.acc_arr.append(obj.acc[0].copy())
            obj.T_arr.append(self.T)

    def fetch_state(self, i_from, i_to):
        for obj in self.objects:
            obj.r[i_to] = obj.r[i_from]
            obj.v[i_to] = obj.v[i_from]
            obj.acc[i_to] = obj.acc[i_from]

    def apply_k_to_state(self, i_k, i_state, coeff):
        for obj in self.objects:
            obj.r[i_state] += obj.k_r[i_k] * coeff

```

```

        obj.v[i_state] += obj.k_v[i_k] * coeff

def RK4_acquire_k(self, i_k, i_state_using, dt):
    for obj in self.objects:
        obj.k_r[i_k] = obj.v[i_state_using] * dt
        obj.k_v[i_k] = dt * self.get_grav_acc(obj.ID,
                                                obj.r[i_state_using],
                                                i_state_using)

def get_grav_acc(self, ID, r, state_idx):
    a = npv3(0,0,0)
    for obj in self.objects:
        if obj.ID == ID:
            continue
        o_r = obj.r[state_idx]
        rdelta = o_r - r
        dist = np.linalg.norm(rdelta, 2)
        if dist < 5:
            continue
        ai = obj.mu / (dist*dist)
        u = unitize(rdelta)
        a = a + ai*u
    return a

def compute_initial_accelerations(self):
    for obj in self.objects:
        obj.acc[0] = self.get_grav_acc(obj.ID, obj.r[0], 0)

def euler_integration(self, dt): #diverges
    self.T += dt
    for obj in self.objects:
        obj.acc[0] = self.get_grav_acc(obj.ID, obj.r[0], 0)
    for obj in self.objects:
        obj.r[0] = obj.r[0] + dt * obj.v[0]
    for obj in self.objects:
        obj.v[0] = obj.v[0] + dt * obj.acc[0]

def v_verlet_integration(self, dt):
    self.T += dt
    for obj in self.objects:
        obj.r[1] = obj.r[0] + obj.v[0]*dt + 0.5*obj.acc[0]*dt*dt
    for obj in self.objects:
        obj.acc[1] = self.get_grav_acc(obj.ID, obj.r[1], 1)
        obj.v[1] = obj.v[0] + 0.5*obj.acc[0]*dt + 0.5*obj.acc[1]*dt
    for obj in self.objects:
        obj.r[0] = obj.r[1]
        obj.v[0] = obj.v[1]
        obj.acc[0] = obj.acc[1]

def RK4_integration(self, dt):
    self.T += dt
    self.RK4_acquire_k(1, 0, dt)
    self.fetch_state(0, 1)
    self.apply_k_to_state(1, 1, 0.5)

```

```

self.RK4_acquire_k(2, 1, dt)
self.fetch_state(0, 1)
self.apply_k_to_state(2, 1, 0.5)
self.RK4_acquire_k(3, 1, dt)
self.fetch_state(0, 1)
self.apply_k_to_state(3, 1, 1.0)
self.RK4_acquire_k(4, 1, dt)
self.apply_k_to_state(1, 0, 1/6)
self.apply_k_to_state(2, 0, 1/3)
self.apply_k_to_state(3, 0, 1/3)
self.apply_k_to_state(4, 0, 1/6)

```

Alle liitettyyn lähdekoodinäytteeseen on koottu määritykset seuraaville funktioille:

Taulukko A1:

funktio	toiminto
<code>angle_between_vectors_3D(v1,v2)</code>	Laskee kahden 3-vektorin välisen terävän kulman.
<code>shadowpoint(p_sun,p_moon,p_earth,Re)</code>	Laskee Kuun massakeskipisteen heittämän varjon sijainnin Maan pinnassa.
<code>measure_eclipse(p_sun,p_moon,p_earth,Re)</code>	Määrittää varjostaako Kuu Maata, ja laskee varjopisteen.
<code>get_earth_orientation_rad(t_,sym_conv)</code>	Määrittää Maan absoluuttisen kääntökulman.

```

@njit(cache=True)
def angle_between_vectors_3D(v1,v2):
    dot = np.dot(v1,v2)
    cross = np.cross(v1,v2)
    det = np.linalg.norm(cross, 2)
    angle = np.arctan2(det, dot)
    return angle

@njit(cache=True)
def shadowpoint(p_sun, p_moon, p_earth, Re):
    u_sun_moon = unitize(p_moon - p_sun)
    threshold = 1.0
    alpha = angle_between_vectors_3D(p_moon - p_sun, p_earth - p_moon)
    d_moon_earth = np.linalg.norm(p_earth - p_moon, 2)
    beta = np.pi - np.arcsin(d_moon_earth * np.sin(alpha) / Re)
    gamma = np.pi - (alpha + beta)
    assert gamma < (np.pi / 2)
    r = np.sin(gamma) * Re / np.sin(alpha)
    d_to_ebc = np.linalg.norm(p_moon + r * u_sun_moon - p_earth, 2)
    d_to_surf = abs(d_to_ebc - Re)
    assert d_to_surf < threshold
    p_strike = p_moon + r * u_sun_moon
    assert d_moon_earth > np.linalg.norm(p_moon - p_strike)
    return p_strike, r

@njit(cache=True)
def measure_eclipse(p_sun, p_moon, p_earth, R_earth):

```

```

v_sun_moon = p_moon - p_sun
v_moon_earth = p_earth - p_moon
d_moon_earth = np.linalg.norm(v_moon_earth, 2)
alpha = angle_between_vectors_3D(v_sun_moon, v_moon_earth)
alpha_crit = np.arcsin(R_earth / d_moon_earth)
if alpha > alpha_crit:
    eclipsing = 0
    lat = 0
    lon = 0
else:
    p_strike, r_ = shadowpoint(p_sun, p_moon, p_earth, R_earth)
    _asr1 = angle_between_vectors_3D(p_strike-p_earth, p_moon-p_earth)
    assert _asr1 < (np.pi / 2)
    v_strike = p_strike - p_earth
    lat = rad2deg*np.arctan2(v_strike[2], np.linalg.norm(v_strike[0:2]))
    lon = rad2deg*np.arctan2(v_strike[1], v_strike[0])
    eclipsing = 1
return eclipsing, lat, lon, alpha, alpha_crit

def get_earth_orientation_rad(t_, sym_conv):
    ERA1 = 0.7790572732640
    ERA2 = 1.00273781191135448
    ret = (-2*np.pi * (ERA1 + ERA2 * (t_.ut1.jd - 2451545.0))) % (2*np.pi)
    if sym_conv and (ret > np.pi):
        ret = ret - (2 * np.pi)
    return ret

```