AVARUUSLUOTAIN

Rymdsonden Spaceprobe
3/2013, vol. 48

SUOMEN AVARUUSTUTKIMUSSEURA r.y.
SATS

Voyager 1 on poistunut aurinkokunnasta

Rakettitieteen jalanjäljillä

Osa 5: Osien integrointi

Sampo Niskanen

Artikkelisarjan aiemmissa osissa olen kertonut OpenRocket-ohjelmiston käyttämistä aerodynaamisista menetelmistä sekä kolmiulotteisen raketin asennon käsitteistä. Seuraavaksi on vuorossa itse lentoradan mallintaminen.

Lähes kaikkien fysikaalisten ilmiöiden mallintaminen tarkoittaa käytännössä differentiaaliyhtälöiden numeerista ratkomista. Differentiaaliyhtälö kuvaa sitä, miten jokin suure muuttuu ajan funktiona, kun suure itse vaikuttaa muutosnopeuteen. Kun esimerkiksi pallo heitetään ilmaan, ilma vastustaa sen liikettä voimalla, joka on suhteessa sen nopeuteen. Näin ollen pallon nopeuden muutos riippuu pallon sen hetkisestä nopeudesta.

Kaavana kirjoitettuna pallon nopeutta v voidaan kuvata

$$v' = -k\, v(t),$$

jossa v'(t) tarkoittaa nopeuden aikaderivaattaa eli kiihtyvyyttä. Hetkellinen nopeuden muutos on siis suoraan verrannollinen sen hetken nopeuteen. Yksinkertaisissa tapauksissa differentiaaliyhtälön voi ratkaista analyyttisesti, jolloin voi laskea pallon sijainnin minä hyvänsä ajanhetkenä tarkasti. Luontoa mallintaessa yhtälöt ovat kuitenkin lähes poikkeuksetta niin monimutkaisia, että joudutaan turvautumaan numeerisiin menetelmiin.

Aiemmissa osissa olen käynyt läpi menetelmät, joilla pystyy laskemaan rakettiin hetkellisesti vaikuttavat voimat ja momentit. Seuraavaksi onkin vuorossa lentoradan simuloiminen näitä hyödyntäen.

Raketin differentiaaliyhtälö

Kun rakettiin kohdistuvat voimat jollakin ajanhetkellä t tunnetaan, saadaan hetkellinen kiihtyvyys a(t) jakamalla voima raketin massalla. Kiihtyvyys on nopeuden aikaderivaatta, mikä on puolestaan paikan aikaderivaatta. Koska lentorataa mallintaessa halutaan selvittää raketin sijainti ajan funktiona, saadaan differentiaaliyhtälö

$$x'' = a(x', x, t)$$

Hetkellinen kiihtyvyys riippuu siis sekä raketin nopeudesta x' (esim. ilmanvastus), sijainnista x (painovoima), että ajasta t (työntövoima).

Sijainti x on kolmiulotteinen. Tällöin voidaankin muodostaa kolme erillistä yhtälöä paikan kolmelle komponentille xx, xy ja xz. Jokainen yhtälö voidaan laskea erikseen, joten vaikka työmäärä kasvaa, se ei teknisesti hankaloita ratkaisemista mitenkään. Useimmiten x ja a on helpointa kuvata vektoreina, jotka sisältävät kaikki kolme komponenttia. Näin yhtälö pysyy selkeänä ja helppona käsitellä, vaikka todellisuudessa yhtälöitä onkin kolme.

Yllä oleva on toisen asteen differentiaaliyhtälö, koska yhtälö sisältää paikan toisen derivaatan. Tällaisten ratkominen suoraan on hankalaa, ja numeerisissa menetelmissä ne lähes poikkeuksetta jaetaankin kahteen ensimmäisen asteen differentiaaliyhtälöön. Tämä onnistuu kätevästi ottamalla muuttujaksi myös nopeus v = x':

$$v' = a(v, x, t)$$

$$x' = v(x, t)$$

Nämä kaksi ensimmäisen asteen differentiaaliyhtälöä yhdistämällä saadaan alkuperäinen toisen asteen yhtälö. Nyt yhtälöitä on kolmen sijaan kuusi, mutta kaikki ovat ensimmäisen asteen yhtälöitä. Kaiken lisäksi oikealla puolella oleva raketin hetkellinen nopeus ja kiihtyvyys tiedetään tai osataan laskea.

Numeerinen integrointi

Numeerisessa ratkaisemisessa yleensä tiedetään fysikaalinen lähtötilanne, ja tiedetään miten suureet muuttuvat ajan ja paikan myötä. Matemaattisesti ensimmäisen asteen differentiaaliyhtälö voidaan esittää yleisesti muodossa

$$f'(t) = g(t, f(t))$$

Tässä siis muutosnopeus f'(t) voi riippua sekä f(t):n arvosta että t:n arvosta. Ongelmassa halutaan selvittää f(t) kun g(t, f) osataan laskea ja tiedetään alkuarvo f(0).

Yksinkertaisin numeerinen menetelmä tähän on nk. Eulerin menetelmä. Siinä askel askeleelta katsotaan, mihin päin funktio on menossa, ja otetaan määrätyn pituinen askel kyseiseen suuntaan.

Menetelmässä oletetaan, että f:n derivaatta säilyisi vakiona aikavälillä t = 0...h, jossa h on pieni aika-askel. Kun f(0) tunnetaan ja derivaatta g(0, f(0)) on laskettu, saadaan

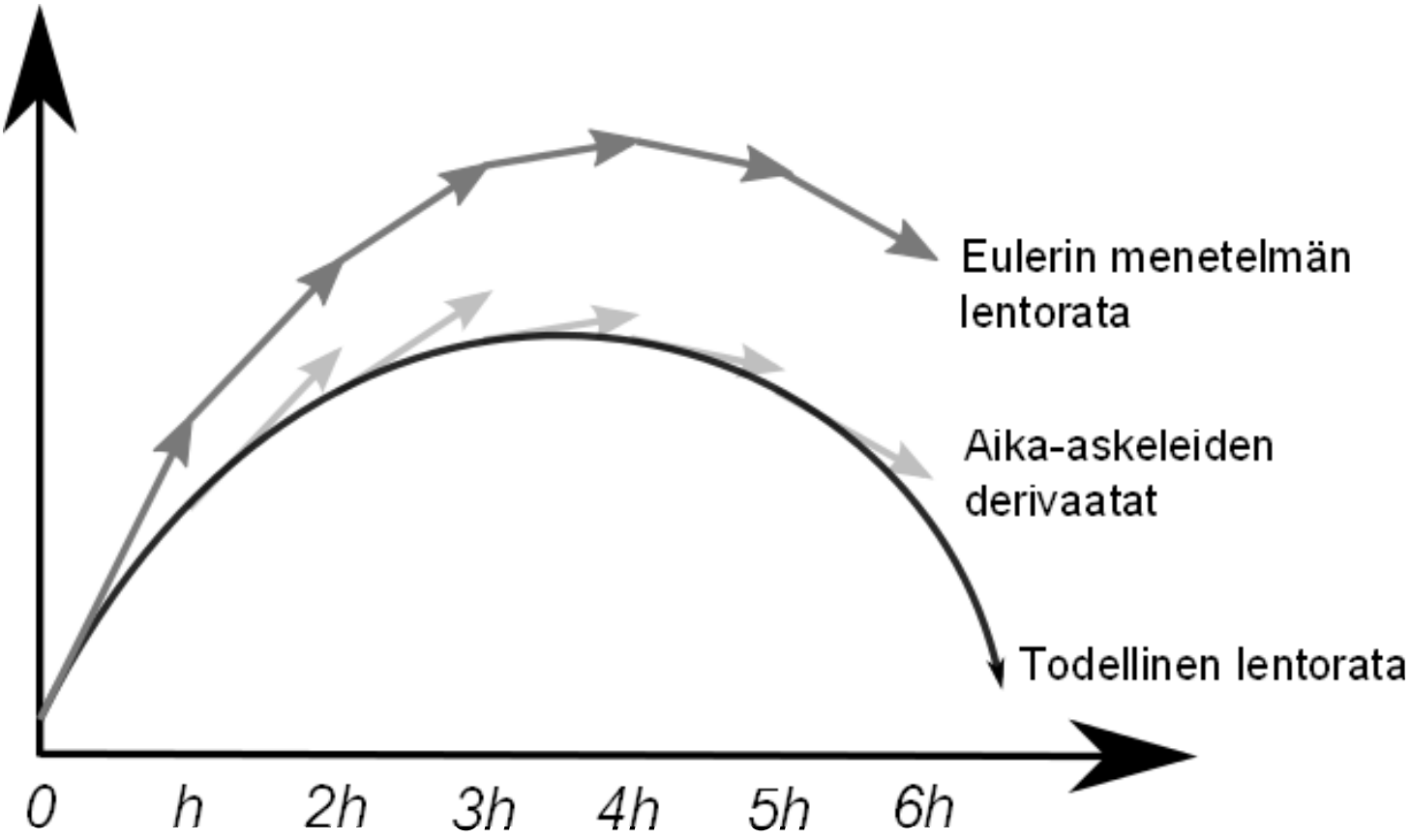
$$f(h) = f(0) + h\, g(0, f(0))$$

Tässä pisteessä operaatio voidaan toistaa, jolloin saadaan f(2h), siitä f(3h) jne. Yleinen sääntö on

$$f(t + h) = f(t) + h\, g(t, f(t))$$

Kun aika-askel pidetään tarpeeksi pienenä, oletus f:n derivaatan pysymisestä vakiona on suhteellisen hyvä. Eulerin menetelmä onkin hyvin helppo toteuttaa, ja hyvä menetelmä aloittaa.

Jouluna 2008 toteutin Eulerin menetelmällä lentoradan laskennan mummolassa ollessani. Hämmästykseni oli suuri, kun heti ensimmäinen lentosimulaatio tuotti lähes metrin tarkkuudella edellispäivänä raketista mittaamani lentokorkeuden. Tällainen tarkkuus oli toki sattumaa, mutta varsin kannustavaa silti.



Virheen kasautuminen Eulerin algoritmissa. Mustalla ”oikea” f(x):n arvo, harmaalla numeerisesti laskettu käyrä. Vaalean harmaalla on kuvattu kunkin aika-askeleen derivaatta g(x). Kuva: Sampo Niskanen.

Euler, Runge vai Kutta?

Vaikka Eulerin menetelmä on selkeä ymmärtää, siinä on ongelmansa. Jokaisella aika-askeleella syntyy jonkin verran virhettä, joka kertyy pikkuhiljaa. Oheinen kuva esittää, kuinka lentoradan kaltaisilla kaarevilla funktioilla jokaisen askeleen aiheuttama virhe on samaan suuntaan, ja virhe kasautuu pikkuhiljaa. Tämän torjumiseksi on kehitetty parempia numeerisia menetelmiä. Yleisin näistä on nk. Runge-Kutta 4 -menetelmä.

RK4-menetelmässäkin lasketaan aina jokin aika-askel h eteenpäin nykyhetkestä. Kaava on kuitenkin mutkikkaampi, ja yhden aika-askeleen laskemiseksi täytyy funktio g(t, f) laskea neljä kertaa. Vaikka yhden aika-askeleen laskeminen vie karkeasti nelinkertaisen määrän aikaa, tulos on huomattavasti Eulerin menetelmää tarkempi.

Numeeristen menetelmien tarkkuutta arvioidaan usein sen suhteen, mitä kokonaisvirheelle tapahtuu, kun aika-askelta muutetaan. Eulerin menetelmässä kokonaisvirhe on suhteessa aika-askeleeseen h, eli jos aika-askeleen puolittaa, puolittuu myös kokonaisvirhe. RK4-menetelmän virhe on puolestaan suhteessa h^4:ään, eli kun aika-askeleen puolittaa, virhe tippuu 1/16 osaan. Vaikka aika-askeleen laskenta on hitaampaa, maksaa kulutettu aika itsensä nopeasti takaisin.

Runge-Kutta -menetelmistä voi lukea lisää Wikipediasta.

Osien integrointi

Kaikki tämä kuulostaa yhdessä varsin monimutkaiselta, lähtien Barrowmanin yhtälöistä ja ilmanvastuksen laskemisesta ja päätyen kvaternioiden kautta Runge-Kuttaan. Yhdessä kasassa nämä olisivatkin toivoton sotku, mutta abstraktiotasot pelastavat tilanteen.

Numeerisen integroinnin kannalta ei ole mitään väliä, mitä kaavaa integroidaan tai miten vaikeaa yhden aika-askeleen parametrien laskeminen on. Sen näkökulmasta tarvitsee vain laskea funktion g(t, f) arvo.

Voimien ja momenttien näkökulmasta taas ei ole väliä, miten aerodynaamiset kertoimet on laskettu. Kvaterniot kuulostavat hankalilta, mutta ovat loppujen lopuksi vain esitysmenetelmä. Barrowmanin menetelmä pilkkoutuu osiksi raketin komponenttien mukaan, samoin ilmanvastus. Jokaisen näistä osista voi rakentaa ja testata yksitellen, ja yhdessä niistä rakentuu raketin lentoradan simulaatio.

Artikkelisarjan seuraavassa osassa kerron ohjelmiston hyödyntämisestä SATS:n Haisunäätä-raketeissa.

Artikkelisarjan aiemmissa osissa kerroin OpenRocket-ohjelmiston kehitysvaiheista ja raketin aerodynaamisten ominaisuuksien laskennasta. Myöhemmissä osissa on luvassa tietoa moniulotteisesta optimoinnista, ohjelmiston käytöstä Haisunäätä-projekteissa sekä tulevaisuuden suunnitelmista.

OpenRocket-ohjelmiston sekä diplomityön saa ladattua osoitteesta http://openrocket.sourceforge.net/ Artikkelisarjan aiemmat osat ovat luettavissa SATS:n sivuilta osoitteesta http://www.sats-saff.fi/ □