

Symboliset laskentaohjelmat ja matematiikan opetus

Simo K. Kivelä

1.10.1999

Laskimet tulivat matematiikan opetuksen työvälineiksi 1970-luvulla, tietokoneet 1980-luvulla, 1990-luvulla ovat ohjelmistot kehittyneet valtavasti. Symbolinen laskenta, tietokonealgebra (CAS = Computer Algebra System) — mitä nimeä halutaankin käyttää — tuntuu olevan tulossa.

Kehitys näyttää tapahtuvan tekniikan ehdoilla. Symbolista laskentaakin on koulumatematiikkaan ajamassa markkinoiden paine. Vähemmän on pohdittu tarvetta sen käyttöön, saatavia hyötyjä, syntyviä haittoja, miten sitä oikeastaan pitäisi käyttää. Kiinnostus tutkimusmielessä paneutua syntyviin pedagogisiin ja didaktisiin ongelmiin tuntuu olevan hämmästyttävän vähäinen.

Dimension muutamassa viime numerossa on asiasta herätelty keskustelua ([1], [2]). Keskustelu jatkokoon.

Symbolinen ja numeerinen laskenta

Interaktiivisiksi matemaattisiksi ohjelmistoiksi — usein lyhyesti matemaattisiksi ohjelmiksi — on tapana kutsua ohjelmia, joissa käyttäjä voi kirjoittaa syötteensä, lähettää sen laskettavaksi ja saada välittömästi vastauksen. Seuraava syöte usein riippuu edellisen tuloksesta. Kyseessä on siten taskulaskimen tyyppinen käyttö, mutta ohjelmien tarjoamat mahdollisuudet luonnollisesti ovat laajemmat.

Matemaattiset ohjelmistot on perinteisesti jaettu numeerisiin ja symbolisiin.

Numeeriset ohjelmistot käsittelevät vain ns. liukulukuja, ts. desimaalilukuja, joiden esittämiseen käytetään rajallinen kiinteä määrä desimaaleja (oikeastaan bittejä) ja joihin liittyy tieto kertoimena olevasta kantaluvun potenssista. Jos desimaalimäärä eli mantissan pituus on esimerkiksi 4 ja kantaluku 10, ovat tämän järjestelmän liukulukuja $1234 = 0.1234 \cdot 10^4$, 0.1234 ja $0.00001234 = 0.1234 \cdot 10^{-4}$. Sen sijaan luku 0.10002 ei ole tämän järjestelmän liukuluku, vaan se esitetään pyöristettynä muodossa 0.1000 , jolloin se ei eroa luvusta 0.1 .

Taskulaskin on esimerkki numeerisesta ohjelmistosta. Kuitenkin jo mahdollisuus käsitellä murtolukuja on askel symbolisten ohjelmien suuntaan.

Numeeristen ohjelmien pohjana ovat usein olleet aluksi FORTRAN-, myöhemmin C-kieliset ohjelmakirjastot, joiden helpoksi käyttöliittymäksi numeerinen

ohjelma on luotu. Ohjelmakirjastojen kehitys alkoi 1950-luvun lopulla, mutta ensimmäisen varsinaisen kukoistuksensa se saavutti 1970-luvulla, minkä seurauksena syntyi vuonna 1980 ehkä tunnetuin numeerinen ohjelma Matlab [14]. Kuluneen parinkymmenen vuoden aikana tämä on kehittynyt matriisialgebrallisesta käyttöliittymästä erittäin monipuoliseksi numeerisen laskennan ohjelmistoksi.

Symbolisissa ohjelmistoissa voidaan numeeristen laskutoimitusten lisäksi käsitellä lausekkeita. Tyypillisimpiä toimintoja ovat lausekkeiden sieventäminen, yhtälöiden ratkaisemiset, derivoinnit, integroinnit. Numeeriset laskut voidaan suorittaa tarkasti mielivaltaisen monen numeron tarkkuudella, murtolukuja voidaan käsitellä, samoin juuria enemmän tai vähemmän menestyksellisesti, jakaa kokonaislukuja alkutekijöihin, jne. Koneen muisti ja tarvittava aika asettavat luonnollisesti rajoituksensa.

Symbolisten ohjelmien historia alkaa 1960-luvulta. Ensimmäinen laajemmin tunnettu PC-koneessa toimiva ohjelma muMATH ([15], [4]) on 1970-luvulta, mutta vasta 1990-luvulla on alkanut voimakas kehitys. Katsaus symbolisten ohjelmien — tietokonealgebran — historiaan on löydettävissä viitteen [3] johdannosta. Tällä hetkellä tarjolla olevista ohjelmista löytyy luettelo viitteestä [9].

Muutaman viime vuoden aikana kehitys on johtanut numeeristen ja symbolisten ohjelmien eron pienenemiseen. Numeerisiin ohjelmiin on usein lisätty mahdollisuus symboliseen laskentaan liittämällä mukaan jonkin tunnetun symbolisen ohjelman ydin. Esimerkiksi Matlabissa voidaan käyttää Maplen [11] tärkeimpiä rutiineja. Toisaalta symbolisten ohjelmien numeeristen algoritmien nopeus on tavattomasti noussut. Silti ne ovat edelleen numeerisessa laskennassa numeerisia ohjelmia hitaampia. Ongelma on usein pyritty ratkaisemaan mahdollisuudella käyttää ohjelman ulkopuolisia numeerisia ohjelmakirjastoja.

Numeeristen ja symbolisten ohjelmien nopeus- ja tehokkuuserot eivät yleensä tule pienissä tehtävissä esiin. Vaikkapa koulukäyttäjän kannalta on yleensä yhdenentekevää, kestääkö jokin lasku tuhannes- tai sadasosasekunnin. Jos kuitenkin sama lasku joudutaan tehtävän ratkaisemiseksi laskemaan miljoonia kertoja, eroista tulee merkittävä.

Ohjelmistojen pääominaisuudet

Sekä numeeriset että symboliset ohjelmistot ovat kehittyneet monikäyttöisiksi ja laajoiksi työskentely-ympäristöiksi. Usein näiden rakenne on modulaarinen: ohjelman ydin sisältää peruspiirteet ja lisää ominaisuuksia saadaan erikseen ladatavasta ja mahdollisesti myös erikseen hankittavista lisäpaketeista tai hankkimalla ohjelman laajempaan käyttöön oikeuttava lisenssi. Ohjelmaa voi siten kasvattaa tarpeen mukaan.

Modulaarisuuden puuttuessa ohjelmiston laajuus voi olla ongelma erityisesti opetuskäytössä: Joudutaan hankkimaan ja maksamaan laajoja ohjelmistoja, joiden ominaisuuksista kyetään ehkä muutaman prosentin verran hyödyntämään.

Seuraavassa on lueteltuna ohjelmistojen tärkeimpiä käyttöalueita. Kaikkia ominaisuuksia ei ole kaikissa ohjelmistoissa. Hankintapäätöksiä tehtäessä onkin syytä pohtia, miten tärkeänä mitäkin piirrettä pitää.

- *Numeerinen laskenta* tarkoittaa yleensä laskemista 12 – 16 numeron tarkkuudella. Symbolisissa ohjelmissa tarkkuus voidaan asettaa halutuksi. Kaikkien tavallisten alkeisfunktioiden arvot sekä reaali- että kompleksialueella pystytään laskemaan, sama koskee yleensä myös melkoista määrää ns. erikoisfunktioita.
- *Symbolinen laskenta* tarkoittaa lausekkeiden ja yhtälöiden käsittelyä, tarkkaa laskemista murtoluvuilla ja juurilla, yms. Alusta lähtien symbolisiksi suunnitelluissa ohjelmissa laskenta toimii johdonmukaisella tavalla. Numeerisissa ohjelmissa, joihin symbolien käsittely on erikseen liitetty, mahdollisuudet ovat rajoitetummat eivätkä aina kytkeydy luontevasti numeeriseen laskentaan.
- *Grafiikka* — sekä kaksi- että kolmiulotteinen — on yleensä kaikissa ohjelmissa. Kolmiulotteinen grafiikka on useimmissa ohjelmissa siinä mielessä aitoa, että koneen muistiin luodaan kolmiulotteinen geometria, joka sitten projisoidaan halutulla tavalla kaksiulotteiseen kuvaruudun tasoon.
- *Dokumentin* tuottaminen tarkoittaa mahdollisuutta varustaa suoritettut laskut kommentteilla ja selityksillä. Joissakin ohjelmissa tämä tarkoittaa myös mahdollisuutta matemaattista tekstiä sisältävien julkaisujen laatimiseen. Edellytyksenä on tällöin, että ohjelmassa on käytettävissä matemaattisten symbolien fontit.
- *Ohjelmoitavuus* on tärkeä ominaisuus käyttötärpeiden kasvaessa. Ohjelman alkeiskäyttö on yleensä interaktiivista syötteiden näppäilyä ja välittömästi saatujen tulosten tarkastelua. Työskentelyn vaativuuden kasvaessa on luonnollista ruveta kirjoittamaan aluksi lyhyitä, myöhemmin pitempiäkin ohjelmakoodia, joita ajetaan interaktiivisesti. Kyseessä on oikeastaan ohjelman laajentaminen omia tarpeita paremmin vastaavaksi ohjelmoimalla itse lisää. Moniin ohjelmistoihin liittyy erikseen hankittavia lisäpaketteja, jotka on tehty juuri tällä tavalla.
- *Liitännät muihin ohjelmiin* ovat usein tarpeen vaatimusten kasvaessa. Joustava työskentely edellyttää, että mitkä tahansa ohjelmalla lasketut tulokset voidaan siirtää jonkin muun ohjelman syötteeksi ja jatkaa työskentelyä sen avulla. Vastaavasti muualla saadut tulokset on voitava lukea sisään.

Viime vuosien aikana on lisääntyvässä määrin sekä ohjelmien syötteissä että tulostuksissa ryhdytty käyttämään tavanomaista matemaattista notaatiota ohjelmakoodin tyyppisen kirjoitustavan sijasta. Tämä helpottaa ainakin aloittelijan työskentelyä ja on luonnollista pyrittäessä tekemään laskennasta julkaisutasoinen dokumentti.

Matemaattisella notaatiolla on kuitenkin myös haittapuolensa: Se ei ole yksikäsittelistä sillä tarkkuudella kuin tietokonejärjestelmässä tarvitaan. Tyypillinen

esimerkki on yhtäläisyysmerkki, joka voi tarkoittaa loogista lausekkeiden yhtäsuuruutta, olla määritelmä tai ilmoitus arvon sijoittamisesta muuttujalle. Ohjelmakoodin tyyppisen notaation käyttö syötteissä takaakin yksikäsitteisyyden ja lisäksi pakottaa käyttäjän tarkemmin ajattelemaan, mitä oikeastaan haluaa.

Eräitä matemaattisia ohjelmia

Tutkimus- ja opetustarkoituksiin kehitettyjä matemaattisia ohjelmistoja on paljon. Seuraava luettelo kattaa ainoastaan tavallisimmat kaupalliset ohjelmistot, joiden käyttö matematiikan opetuksessa lukio-, ammattikorkeakoulu- ja yliopistojen peruskurssitasolla on luontevaa.

Ei-kaupallisten ohjelmien joukosta saattaa löytyä varsin käyttökelpoisiakin, mutta ongelmana on usein viimeistelemättömyys ja tuen puute. Käyttäjä on niiden kanssa suuremmassa määrin yksin kuin kaupallisia ohjelmia käyttäessään.

Kaupallisissa ohjelmistoissakin on toki puutteensa eikä valmistaja luonnollisesti vastaa toimivuudesta. Virheellisiä toimintoja on yllättävänkin helppoa löytää. Kyse voi tosin olla myös eroista käyttäjän ajattelussa ja ohjelman toimintaperiaatteissa.

Matlab [14] on hyvin laaja ja tehokas numeerinen ohjelmisto, johon liittyy melkoinen määrä eri alojen — mm. signaalinkäsittelyn ja säätöteorian — lisäpaketteja. Rajoitetut symboliset ominaisuudet saadaan käyttöön Maplen ytimeen perustuvan lisäpaketin avulla. Hyvät graafiset ominaisuudet, ohjelmoitava. Saatavana useisiin käyttöympäristöihin, mm. Windows ja Linux, Unix yleisemminkin. Tuskin kovin hyvin lukioon soveltuva, pikemminkin tiettyjen tekniikan alojen koulutukseen ja tutkimustyöhön.

Mathematica [13] on erittäin laaja ja monipuolinen symbolinen ohjelmisto, joka sisältää monipuoliset mahdollisuudet ohjelmointiin. Hyvät graafiset ominaisuudet, varsin hyvät mahdollisuudet dokumentin tuottamiseen. Saatavana useisiin käyttöympäristöihin, mm. Windows ja Linux (Unix). Tutkijan työkalu, ainakin lukiokäyttöön paljon laajempi kuin mitä tarvitaan.

Maple [11] on symbolinen ohjelmisto, Mathematican tasaveroinen kilpailija. Mahdollisuudet dokumentin tuottamiseen hieman Mathematicaa heikommat. Mm. Windows ja Linux (Unix). Käytöltään Mathematicaan verrattava.

Mathcad [12] on pohjaltaan numeerinen ohjelmisto, johon on liitetty symbolisia ominaisuuksia Maplen rutiinien avulla. Hyvät graafiset ominaisuudet, ohjelmoitavuus edellisiä heikompi, laskennasta syntyy dokumentti varsin luontevasti. Tarkoitettu edellä mainittuja pienempimuotoiseen työskentelyyn. Käyttöympäristönä vain Windows.

Derive [10] on Mathematicaa ja Maplea selvästi suppeampi, lähinnä opetustarkoituksiin tehty symbolinen ohjelmisto. Tyydyttävä grafiikka, ohjelmoitavuutta ei oikeastaan ole, julkaisutasoista dokumenttia ei voida tehdä. Selvästi edellä käsiteltyjä ohjelmistoja vaatimattomampi. Käyttöympäristönä Windows.

Derive on pienen kokonsa takia sijoitettavissa myös taskulaskimeen: Texas Instruments -laskimet TI-89 ja TI-92 [16] sisältävät muiden toimintojen ohella

Deriven pääpiirteet. Laskimien näyttöruudun pieni koko tosin rajoittaa käyttöä.

Mihin ja miten matemaattisia ohjelmistoja tulisi opetuksessa käyttää?

Ensimmäinen kysymys ehkä olisi, miksi matemaattisia ohjelmistoja tulisi opetuksessa käyttää ja pistäisikö niitä lainkaan käyttää. Varmaankin pitäisi, koska tekniikka on tuonut tällaiset työkalut saataville. Jokaisella aikakaudella on omat leikki- ja työkalunsa; eriaisteiset oppilaitokset opettakoot käyttämään niitä.

Ohjelmistoilla on toki myös annettavaa matematiikan oppimiseen ja opettajan päivittäiseen työhön. Asioiden havainnollistaminen tulee ennen näkemättömällä tavalla mahdolliseksi; oppilas voi tutkia asioita eri näkökulmista, tehdä 'entä-pä jos' -tyyppisiä kokeiluja. Kaikesta voidaan kirjoittaa dokumentti ja hioa se helposti huolelliseen asuun.

Opettaja voi kirjoittaa kalvot ja harjoitustehtävät siistiin asuun, liittää helposti mukaan graafisia esityksiä. Ohjelmien varaan voidaan rakentaa interaktiivisia opiskelupaketteja, jotka antavat oppilaalle mahdollisuuden edetä omaan tahtiinsa ([5], [6]).

Ensimmäisenä vastaan tuleva käyttötapa lienee kuitenkin graafisten esitysten laatimisen ohella taskulaskimenomainen käyttö: koneelle mennään sieventämään jokin lauseke, laskemaan jokin integraali, ratkaisemaan jokin yhtälö, piirtämään jokin kuvaaja. Taskulaskimissa olevat symboliset piirteet eivät paljoon muuhun anna mahdollisuuksiakaan.

Pelättävissä on, että tämäntyyppinen käyttö johtaa itse asiassa osaamisen tason laskuun. Ei enää kyetä hahmottamaan kuvaajia pelkästään lauseketta katsomalla, yksinkertaisiakaan sievennyksiä ei osata nähdä, yhtälöitä ei osata ratkaista ilman konetta. Integraalitkin voisi olla opettavaisempaa hakea taulukosta.

Välttämättä näin ei tietysti tarvitse käydä. Jos koneen käyttäjällä on periaatteellinen taito hallinnassaan — hän osaisi laskea laskun käsinkin, vaikka siihen ehkä menisi aikaa — ohjelman käyttö voi toki auttaa opiskelussa. Tulee mahdolliseksi tutkia ja analysoida monimutkaisiakin asioita sotkeutumatta laskuteknisiin yksityiskohtiin. Konetta ei siten pitäisi käyttää liian yksinkertaisiin tehtäviin: jos näitä ei osata laskea kyllin nopeasti ilman konetta, periaatteellinen taito ei ole hallinnassa.

Sieventäminen ja lausekkeen muokkaaminen ei symbolisessa ohjelmassa edes ole helppoa. Tavoitehan ei ole yksikäsitteinen, vaan se riippuu siitä, mihin tulosta on tarkoitus seuraavaksi käyttää. Ohjelman yleinen sievennyskäsky tekee, mitä se on ohjelmoitu tekemään. Lausekkeen muokkaaminen johonkin toiseen suuntaan saattaisi viedä laskua paljon paremmin eteenpäin, mutta tarvittavan käskyn löytäminen ei yleensä ole mahdollista, ellei käyttäjä näe, mihin suuntaan tulisi pyrkiä.

Uudella välineellä on aina hyötynsä ja haittansa. Sitä on opittava käyttämään oikealla tavalla. Taskulaskimet aikanaan siirsivät logaritmitaulut ja laskutikut

historiaan, ja oli saatu kiistatta uusia hyödyllisiä välineitä. Oikeaa käyttöä ei kuitenkaan ole vielä täysin opittu: siirrytään numeerisiin laskuihin silloinkin, kun yksinkertainen algebra auttaisi tarkan vastauksen saannissa ja tehtävän rakenteen hahmottamisessa; laskimesta luetaan likiarvoja muutaman numeron tarkkuudella ja hetken päästä nämä syötetään laskimeen uudelleen, vaikka muistirekistereitäkin voisi käyttää.

Symboliset laskentaohjelmat ovat uusi työkalu, jonka käyttö on opittava, ei yksinomaan teknisesti, vaan myös pedagogisesti. Opiskelu ja uuden välineen mahdollisuuksien kartoitus vaatii kuitenkin melkoisen työpanoksen.

Saattaisi myös olla tarpeen tuottaa valmista digitaalista materiaalia helpottamaan alkuunpääsyä ohjelmistojen käytössä. Luontevaa olisi, että tämä seisoisi omilla jaloillaan liittymättä tiettyyn oppikirjaan laajemman käyttökelpoisuuden saavuttamiseksi. Koska ohjelmistoilla yleensä alkaa olla myös julkaisuominaisuudet, tällaisen materiaalin tuottaminen on mahdollista.

Millaisia periaatteita sitten tulisi soveltaa laskentaohjelmistojä käytettäessä? Ainakin seuraavia näkökohtia on syytä pohtia:

- Taskulaskimenomainen erillisten laskentatehtävien suorittaminen ei ole riittävää, vaan ohjelmistoa on käytettävä kokonaisvaltaisemmin: Lähtötiedot ja välitulokset talletetaan ohjelman muuttujiin käyttäen kuvaavia nimiä ja laskuoperaatiot kohdistetaan näihin. Päähuomio ei kiinny yksittäisiin operaatioihin vaan kokonaisuuden hahmottamiseen ja ratkaisuprosessin hallinnassa pitämiseen. Kone tekee laskentatyön.
- Tehtävien ei tarvitse olla laskuteknisesti yksinkertaisia, vaan voidaan käsitellä aiempaa paljon monimutkaisempia tehtäviä. Jossain määrin tämä muuttaa matematiikan opetuksen luonnetta: laskuteknikka ei olekaan enää niin tärkeää, sen sijaan rakenteiden ymmärtäminen ja isompien kokonaisuuksien hallinnassa pitäminen painottuvat.
- Algebrallisten ja analyttisten laskentatekniikkojen rinnalla käytetään numeerisia menetelmiä, kun se on mielekästä. Esimerkiksi: Jos yhtälö ei ole muuten ratkaistavissa (tai symbolinen ratkaisu on liian monimutkainen), ratkaistaan se numeerisesti.
- Grafiikkaa hyödynnetään tilanteen hahmottamisessa. Piirretään sopivasti skaalattuja käyriä ja pintoja, funktioiden kuvaajia, etsitään graafisesti sopivia lähtöarvoja numeeristen menetelmien käytölle, etc.
- Tehtävän ratkaisun valmistuttua se dokumentoidaan kirjoittamalla tarvittavat selitykset ja kommentit kauniilla suomen kielellä, tavalla, joka kelpaa myös äidinkielen opettajalle.

Lukija varmaankin huokaa kauhusta ja toteaa, ettei koulussa ole moiseen aikaa. Näin varmasti onkin, ellei matematiikan opetusta paljolti ajatella uudestaan. Ja jos näin ei tehdä, tuskin tarvitsemme laskentaohjelmiakaan.

Symbolista laskentaa tukevien ohjelmien potentiaali on sen verran suuri, että vakava kehitystyö tulisi käynnistää. Tämä merkitsee monien matematiikan

opetukseen liittyvien asioiden uudelleen ajattelua, kokeilumateriaalin tuottamista, sen testaamista kouluympäristössä, opettajan omaa opiskelua. Unohtaa ei myöskään pidä sen tutkimista, millaista matematiikan oppimista ja millaista opiskeluprosessia laskentaohjelmien käyttö parhaimmillaan voi tukea.

Viitteissä [7] ja [8] on vaatimattomina esimerkkeinä Mathematican muistikirja-muotoiset (Notebook-muotoiset) dokumentit hieman yksinkertaisemmasta ja hieman monimutkaisemmasta ongelmasta.

Viitteet

- [1] Timo Suvanto, Matematiikan opettajan uudet työkalut, Dimensio 1/1999.
- [2] Teuvo Laurinolli, Miten teknologia muuttaa matematiikan opetusta, Dimensio 2/1999.
- [3] K. O. Geddes, S. R. Czapor, G. Labahn, Algorithms of Computer Algebra, Kluwer Academic Publishers, Boston, 1992.
- [4] C. Wooff, D. Hodgkinson, muMATH: A microcomputer algebra system, Academic Press, London, 1987.
- [5] Jerry Uhl, Bill Davis, Horacio Porta, Calculus & Mathematica, amerikkalainen Mathematicaa hyödyntävä matematiikan opiskelupaketti; 1991, kehitetään edelleen; <http://www-cm.math.uiuc.edu/> (14.9.1999).
- [6] Simo K. Kivelä, Riikka Nurmiainen, Jorma Joutsenlahti, \mathfrak{M} niinkuin matematiikka, tietokoneavusteinen matematiikan opiskelupaketti; Teknillinen korkeakoulu, matematiikan laitos, 1999; <http://www.math.hut.fi/matta/material.html> (15.10.1999).
- [7] Mathematica-dokumentti tienmutkan kaarevuussäteen laskemisesta, Simo K. Kivelä, 1999; <http://www.math.hut.fi/~kivela/mmadoc/tie.nb> (15.10.1999).
- [8] Mathematica-dokumentti pallopinnalle kitkatta asetettujen partikkeleiden sijoittumisesta, Simo K. Kivelä, 1997; <http://www.math.hut.fi/~kivela/mmadoc/partikkelit.nb> (15.10.1999).
- [9] Computer Algebra Information Network, Systems and Packages, <http://www.riaca.win.tue.nl/CAN/SystemsOverview/> (14.9.1999).
- [10] Derive, symbolinen ohjelma; The Soft Warehouse, Honolulu, Hawaii, USA; 1988, kehitettäneen edelleen; <http://www.derive.com/>.
- [11] Maple, symbolinen ohjelma; Waterloo Maple, Inc., Waterloo, Ontario, Canada; 1980, kehitetään edelleen; <http://www.maplesoft.com/>, Suomessa <http://www.comsol.fi/>.
- [12] Mathcad, numeerinen ohjelma; MathSoft, Inc., Cambridge, Massachusetts, USA; 1986, kehitetään edelleen; <http://www.mathsoft.com/>, Suomessa <http://www.zenex.fi/>.

- [13] Mathematica, symbolinen ohjelma, Stephen Wolfram; Wolfram Research, Inc., Champaign, Illinois, USA; 1988, kehitetään edelleen; <http://www.wri.com/>, Suomessa <http://www.clinet.fi/naykki/>.
- [14] Matlab, numeerinen ohjelma, Cleve Moler, ym.; TheMathWorks, Inc., Natick, Massachusetts, USA; 1980, kehitetään edelleen; <http://www.mathworks.com/>, Suomessa <http://www.comsol.fi/>.
- [15] muMATH, symbolinen ohjelma, David Stoutemyer, Albert Rich; The Soft Warehouse, Honolulu, Hawaii, USA; 1976, ei saatavissa enää.
- [16] Texas Instruments -laskimet; <http://www.ti.com/calc/docs/graph.htm>, <http://www.ti.com/calc/suomi/tuotetietoa.htm> (15.9.1999).