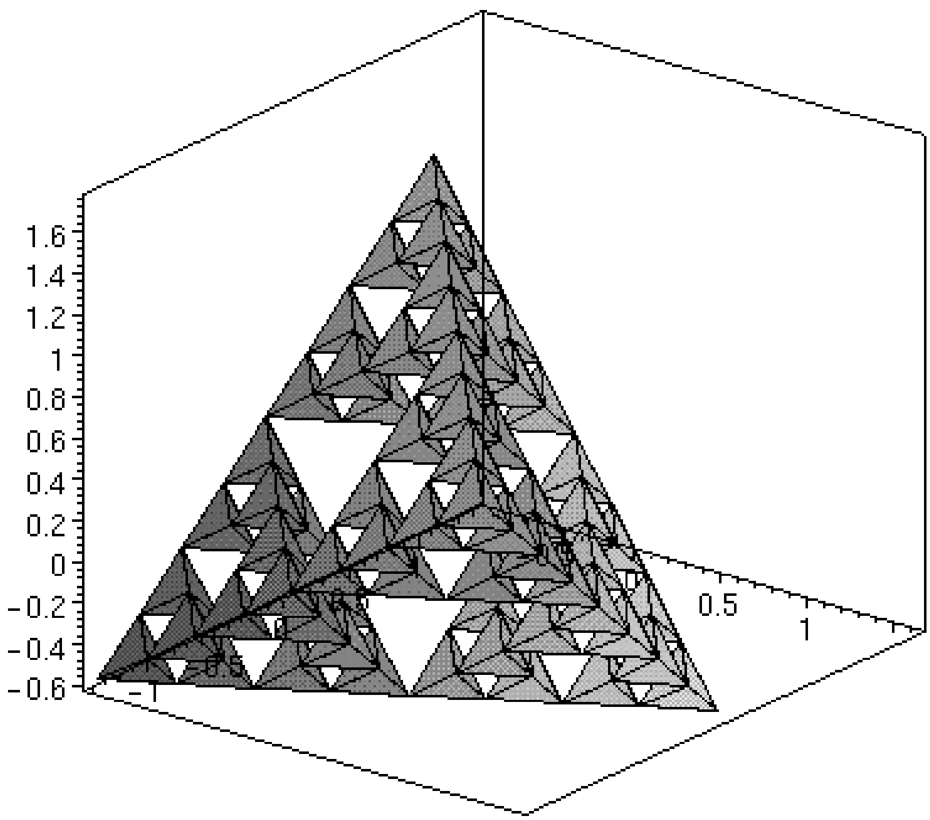
Solmu

Matematiikkalehti
2/1999–2000



http://www.math.helsinki.fi/Solmu/

Solmu 2/1999–2000

Matematiikan laitos
PL 4 (Yliopistonkatu 5)
00014 Helsingin yliopisto

http://www.math.helsinki.fi/Solmu/

Päätoimittaja *Pekka Alestalo*
Toimitussihteerit *Jouni Seppänen* ja *Mika Koskenoja*

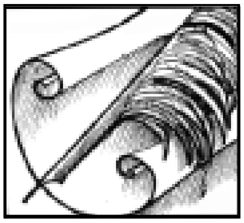
Sähköposti
pekka.alestalo@helsinki.fi
jouni.seppanen@iki.fi

Seuraavaan lehteen tarkoitetut kirjoitukset pyydämme lähettämään vuoden 1999 loppuun mennessä.

Lehden aloittamisen tekivät taloudellisella tuellaan mahdolliseksi Nokia (http://www.nokia.com/) ja Taloudellinen Tiedotustoimisto (http://www.tat.fi/). Opetusministeriö (http://www.minedu.fi/) on keväästä 1997 alkaen avustanut taloudellisesti Solmua.

Huom! Solmun paperiversio postitetaan nykyisin vain niihin kouluihin, jotka ovat sitä erikseen pyytäneet. Solmun Internet-sivuilta saatava paperiversio on mahdollista tulostaa omalla kirjoittimella. Toivomme, että lehti ei jää vain opettajien luettavaksi, vaan sitä kopioidaan kaikille halukkaille.

Sisällys



Pääkirjoitus

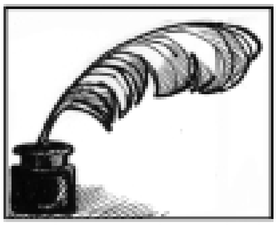
Käsillä – tai ruudulla – oleva vuoden viimeinen Solmu käsittelee pääasiassa tietokoneohjelmistoja ja tietokoneiden käyttöä matematiikassa muutaman eri kirjoittajan näkökulmasta.

Itse törmäsin näihin ohjelmiin ensimmäisen kerran opiskeluaikanani. Suoraan sanoen en ollut erityisen ihastunut niiden toimintaan, mikä ehkä osaltaan johtui kiinnostuksen ja siitä seuranneesta taitojen puutteesta. Muistaakseni tyypillinen istunto käsitti muutamia laskuja, joiden jälkeen koneen muisti täyttyi ja kaikki piti aloittaa uudelleen alusta. Ensimmäinen itse kirjoittamani hieman pidempi ohjelma kieltäytyi itsepintaisesti toimimasta, kunnes pitkällisen pohdinnan jälkeen tajusin, että kyseinen symbolisen laskennan ohjelmisto osaa kyllä laskea kompleksiluvun $\sqrt{2}+i$ pituuden eli modulin, muttei osaakaan tehdä samaa luvulle $1+i\sqrt{2}$.

Ohjelmien nykyisten versioiden kohdalla tällaiset ongelmat ovat historiaa. Käyttämisen alkuun pääsee ilman ohjelmointitaitoja, mutta tietenkin jonkinlainen kokemus tietokoneen käytöstä yleensä on eduksi. Uskaltaisin jopa väittää, että uusia ohjelmia on helpompi oppia käyttämään kuin tavallista graafista laskinta; ainakin pikainen käyttöohjeiden selaaminen viittaa vahvasti tähän suuntaan.

On selvää, että tietokoneiden käytön yleistyessä myös matemaattisten ohjelmistojen suosio kasvaa. Tässä yhteydessä täytyy kuitenkin jälleen kerran muistaa, etteivät ohjelmistot koskaan voi korvata matemaattisten käsitteiden ja ideoiden opiskelun merkitystä, sillä myös tietokonetta käytettäessä täytyy ymmärtää, mitä on tekemässä. Kynää ja paperia tarvitaan myös tietokoneluokissa.

Pekka Alestalo



Toimitussihteerin palsta

Solmun verkkosivuille on tänä syksynä tehty hakemisto[1], joka helpottaa Solmussa aikaisemmin ilmestyneiden artikkeleiden etsintää. Onhan Solmussa ilmestynyt jo noin sata artikkelia joulukuusta 1996 lähtien. Hakemistossa artikkelit on jaoteltu aihepiireittäin, joten erityisesti haku aiheen perusteella on aikaisempaa helpompaa. Sisällysluettelossa näkyvä otsikkokaan kun ei aina kerro paljoakaan artikkelin todellisesta aiheesta. Toivon, että tutustutte hakemiston toimintaan ja lähetätte palutetta Solmun toimitukselle.

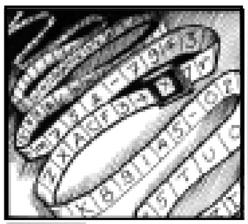
Solmun kaksi edellistä numeroa julkaistiin verkossa alunperin ainoastaan PDF-muodossa. Koska niiden lukemisessa ja tulostamisessa on ilmennyt erinäisiä ongelmia, olemme päättäneet, että Solmuista tehdään verkkoon edelleen myös HTML-versiot. Näin toteutetuissa verkkolehdissä matematiikan esittäminen ei vastaa LaTeX-ladontaohjelman tuottamaa tasoa, mutta sivujen lataaminen on PDF-tiedostoihin verrattuna nopeampaa ja tietokoneelta vähemmän resursseja vaativaa.

Mika Koskenoja
mika.koskenoja@helsinki.fi

Korjaus

Solmun numerossa 1/1999–2000 olleesta artikkelista *Numerosokeus – kansainvälinen vitsaus* puuttui maininta, että teksti oli aiemmin julkaistu Tietoyhteys-lehdessä (3/1999). Artikkeli julkaistiin Solmussa Tietoyhteys-lehden luvalla.

[1] http://www.math.helsinki.fi/Solmu/hakemisto.html



Tietokonepalsta: Matemaattisten symbolien käsittelyä

Jouni Seppänen aloitti Solmun numerossa 1/1998–1999 palstan, jossa käsitellään matematiikan ja tietotekniikan yhteyksiä. Pyrkimyksemme on saada tälle palstalle lisää aktiviteettia. Hyvin luonteva ja ajankohtainen aihepiiri tietotekniikan ja matematiikan yhteistyön alalla on *symbolilaskenta*, jota usein kutsutaan myös nimellä *tietokonealgebra*. Puhumme tässä kirjoituksessa lyhyyden vuoksi useimmiten *symboliohjelmista*. Kyseessä on niin laaja aihepiiri, että siitä riittää kerrottavaa montakin palstallista. Toivon, että tämä aihe kirvoittaa muitakin kirjoittajia näppäimistön äärelle.

Pyrin valottamaan joitakin symbolilaskennan perusajatuksia ja erityisesti lukion matematiikkaan liittyviä esimerkkejä. Joissakin kohdissa saattaa tulla vastaan jokunen koulukurssiin kuulumaton asia, mutta lukija voi huoletta sivuuttaa hänelle vieraat käsitteet kokonaisuuden siitä kärsimättä. Loppuhuipennuksena oleva esimerkiprojektikaan ei mene olennaisesti lukion suppean oppimäärän yli.

Keskityn tällä kertaa Maple-ohjelmaan.

Symbolien käsittelyä tietokonealgebrasysteemillä

Tietokoneen englanninkielinen nimi ”computer” viittaa laskemiseen. Siihen tehtäväänhän tietokone alunperin kehitettiin. Tällä on totuttu tarkoittamaan laskentaa numeroilla, tarkemmin sanoen liukuluvuilla, eli desimaaliluvuilla, joiden esitystarkkuus on tyypillisesti noin 16 numeroa.

Laskentaan tietokoneella on myös toinen näkökulma, josta käytetään nimityksiä *symbolilaskenta, symbolinen ja algebrallinen laskenta, tietokonealgebra* ym. Lyhyesti sanottuna tällä tarkoitetaan laskentaa symboleilla, jotka edustavat matemaattisia olioita.

Tällaisten ohjelmistojen lyhyt kehityshistoria on hyvä esimerkki eri tieteen ja tekniikan alojen yhteistyön hedelmällisyydestä. Pohjatyöhön tarvittiin matemaatikoiden, fyysikoiden ja tekoälytutkijoiden yhteisiä ponnisteluja, joita on tehty 1950-luvulta alkaen. Erityisesti *MIT*:n eli *Massachusetts Institute of Technology*:n symbolilaskentaryhmän työn tuloksena syntynyt, tekoälykielellä Lisp ohjelmoitu Macsyma oli eräs 1960-luvun lopun ohjelmistoihme. Systeemi on edelleenkin käytössä. Noihin aikoihin ja vielä 1980-luvun lopulle saakka vakava este symboliohjelmien laajamittaiselle käytölle oli erityisesti suuri muistin tarve. Viime vuosina tämä este on käytännöllisesti katsoen poistunut ja etevimmätkin tietokonealgebrasysteemit ovat tulleet tavallisen kotikoneen käyttäjän ulottuville.

Nykyiset symbolilaskentaohjelmat on yleensä kirjoitettu C-kielellä ja niissä on kehittynyt graafinen käyttöliittymä, jonka ansiosta voidaan yhdistää käyttäjäystävällisellä tavalla symbolinen, numeerinen ja grafiikkaa sisältävä matemaattinen toiminta, mistä lopputuloksena voidaan muokata tyylikäs matemaattinen dokumentti.

Yleiskäyttöisistä symboliohjelmista suosituimmat lienevät nykyisin Mathematica ja Maple. Lisäksi erityisesti koulumaailmaa kiinnostava ohjelma on Derive, joka alunperin suunniteltiin pieniin koneympäristöihin. Nykyisin suuri osa Derive-ohjelmistoa toimii jopa `TI-92`-laskimessa (Texas Instruments 1995).

Maple ja Mathematica ovat periaatteessa keskenään samankaltaisia ohjelmia. Niissä molemmissa on valtavan matemaattisen tietämyksen lisäksi edellä kuvatunlainen pitkälle kehittynyt graafinen käyttöliittymä. Lisäksi kumpaankin systeemiin kuuluu täydellinen ohjemointikieli, joten ne ovat käyttäjän laajennettavissa. Niissä on samankaltainen perusfilosofia, jonka oppiminen antaa käyttäjälle suuren vapauden toteuttaa mitä erilaisimpia matemaattisia ideoita.

Käyttöliittymät ja matematiikka verkossa

Maplen käyttöliittymänä on työarkki ("worksheet"), joka näyttää matemaattiset kaavat oikeassa asussaan ja jolle myös grafiikka tulostuu.

Työarkissa on hyvät editointiominaisuudet, sitä voidaan käyttää matemaattisen työn dokumentointiin ja se voidaan joko osina tai kokonaisuutena muuntaa erilaisiin muotoihin. Tässä kirjoituksessa hyödynnämme erityisesti muunnoksia LaTeX- ja HTML-muotoihin. Maplen uusimmassa versiossa V.5.1 voidaan työarkki ladata Maple-istuntoon suoraan verkosta antamalla kyseinen `www-URL` .

Molemmissa systeemeissä on myös kattava avustusjärjestelmä.

Sekä Maplen että Mathematican uusissa tai lähiaikoina julkistettavissa versioissa on MathML- tai OpenMath-muotoinen tulostus. Nämä tulevat olemaan WWW-selainten tukemia matematiikan esitysmuotoja verkossa.

Kerään kirjoitukseen liittyvät WWW-linkit sivulle `http://www.math.hut.fi/~apiola/solmu/`. Sieltä ovat myös saatavissa kaikki kirjoituksessa esiintyvät työarkit. Suunnittelen harjoittavani jonkinasteista ylläpitoa sivustolle ja erityisen mieluusti sijoitan sinne lukijoiden lähettämiä lisäyksiä, ehdotuksia, korjauksia, ratkaisutyöarkkeja, mielipiteitä ym.

Otan mielelläni vastaan palautetta osoitteella `heikki.apiola@hut.fi` .

Käytön perusteita

Maple käynnistetään suoraan ikonista tai esim. komennolla `maple` tai `xmaple` järjestelmästä riippuen.

Komentoja voidaan ruveta kirjoittamaan riville, jonka alussa on Maple-kehote $\boxed{>}$. Jokainen komento vaatii loppumerkin, joko puolipisteen (;) tai kaksoispisteen (:).

Valaisen perusperiaatteita joukolla kommentoituja esimerkki-istuntoja. Pyrin esitykseen, joka antaisi mielikuvan joistakin ohjelman käyttömahdollisuuksista ja voisi olla kiinnostava sellaisillekin lukijoille, joilla ei ole ohjelmaa käytössään (ainakaan vielä).

Esimerkki-istunnot perustuvat Maplen versioon V.5.1 ja ne on tehty suurimmaksi osaksi Linux-järjestelmässä. Ne toimivat aivan samalla tavoin muissa Unix-järjestelmissä sekä Windows- ja Macintosh-ympäristöissä. Työarkit ovat järjestelmästä riippumattomia, ts. jossain ympäristössä aloitettua työtä voi ongelmitta jatkaa toisessa.

Ladontateknisiä huomioita

Näillä ei ole mitään tekemistä kirjoituksen sisällön kanssa, mutta kenties ne ovat jollekin lukijalle kiinnostavia. Sinänsä olisi varmaan paikallaan saada Solmuun kirjoitus matemaattisesta tekstinkäsittelystä LaTeX-ohjelmalla.

Esimerkit on tuotettu muuntamalla Maple-työskentelyn tuloksena syntyneet työarkit LaTeX-muotoon osittain suoraan työarkin FILE-valikon EXPORT → LATEX-valinnalla, osittain muuntamalla yksittäiset kaavat Maplen `latex`-komennolla ja tekemällä hiukan käsityötä editorilla.

Kuvat on siepattu ruudulta UNIX-työkalulla nimeltään *xv*. Tämän ohjelman tuottama Postscript ei vastaa työarkin tarkkuutta. Siksi lehden sivulla olevien kuvien laatu on todellisuutta huonompi.

Maple laskimena

Käytän sanontoja *komento* ja *funktio* kutakuinkin synonyymeinä. Funktio-nimitystä käytän erityisesti matemaattisten operaatioiden yhteydessä.

Aloitan aritmeettisella työarkilla, jossa silmiinpistävää on tarkka rationaalilaskenta täydennettynä irrationaalilukuja esittävillä symboleilla.

```
> 70!;
119785716699698917960727837216890987364589381425464258575553628646280095827\
    89845319680000000000000000
```

```
> 6^130;
144431708923714697282341921957287977256854489558705240134406338415596406037\
    318860947517804001179467776
```

Maple laskee tarkoilla kokonais- ja rationaalilukuarvoilla. Laskentatarkkuutta rajoittaa vain käytettävissä oleva muisti.

```
> 70!/6^130;
```

$$\frac{43804141699279503814533528960187340576703427139216850433349609375}{528168734793686199713952913070338335607038938965023384939698061312}$$

Osoittajan ja nimittäjän pituuksien perusteella näemme, että Maple supistaa automaattisesti yhteiset tekijät.

Komennolla `evalf` saadaan liukulukulikiarvo (**f**-kirjain sanan `evalf` lopussa viittaa termiim ”*floating point*”). Edellisen komennon tulokseen viitataan `%`-merkillä.

```
> evalf(%);
          .08293588547
```

Liukulukulikiarvon laskentatarkkuuden oletuksena on 10 numeroa. Tarkkuutta voidaan muuttaa antamalla se komennon toiseksi argumentiksi.

Maple tuntee matemaattisia vakioita, kuten π, jonka merkkinä on `Pi`.

```
> evalf(Pi,60);
         3.14159265358979323846264338327950288419716939937510582097494
```

Imaginaariyksikölle on varattu kirjain `I`, sensijaan *Neperin luvulle* ei ole varattu omaa merkkiä, vaan se lasketaan eksponenttifunktion avulla pisteessä $x = 1$.

```
> evalf(exp(1),50);
              2.7182818284590452353602874713526624977572470937000
```

Käytämme hyväksemme Maplen yhtälörakennetta tulostaaksemme matemaattisen identiteetin. Vasemmalla puolella käytämme sitaatteja, mikä estää arvon laskemisen.

```
> 'sin(Pi/2)'=sin(Pi/2);
```

$$\sin \pi/2 = 1$$

Funktiolla `seq` on kätevää muodostaa erilaisia jonoja.

```
> seq(2^i,i=0..15);
```

$$1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768$$

Funktio `ithprime` on esimerkki Maplen lukuisista voimakkaista matemaattisista funktioista. Sen nimi viittaa englanninkieliseen sanaan "i^{th} prime", eli i:s alkuluku. Niinpä lukija arvannee, mitä tässä tapahtuu:

```
> seq(ithprime(i),i=10..25);
```

$$29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97$$

Ei ole vaikeaa antaa ohjelmalle liian suurta tehtävää:

```
> seq(ithprime(i),i=1..10^6);
Error, object too large
```

Polynomeja ja muita lausekkeita

Aivan kuin missä tahansa ohjelmointikielessä, voimme ottaa käyttöön muuttujia, joille sijoitamme arvoja. Sijoittamismerkkinä on `:=` . Symboliohjelman tapauksessa arvot voivat olla myös symboleja tai symboleista koostuvia lausekkeita.

Merkittävä symboliohjelman ominaisuus on kyky sieventää lausekkeita. Tähän on monia komentoja. Yleiskomento on `simplify`. Tavallisesti käyttäjällä on mielessään jokin muoto, johon hän pyrkii. Silloin on syytä käyttää esim. auki kertovaa komentoa `expand`, tekijöihin jakoa yrittävää `factor`, yhteen keräävää `combine` jne. Komentoihin voidaan liittää monia lisätarkentimia.

Annamme muutaman esimerkin lukuisista sievennyskomennoista ja niiden lukuisista lisätarkentimista. Kannattaa suorittaa avustuskomento `?simplify` kattavan esittelyn saamiseksi. Aihetta on käsitelty kaikissa viitteissämme, mainittakoon [2][I.4, ss. 103–111], [6][Luku 10]. Perusteellisin esitys on *Heckin* kirjassa [4] .

```
> p:=(x-1)^4;
```

$$p := (x-1)^4$$

```
> expand(p);
```

$$x^4 - 4\,x^3 + 6\,x^2 - 4\,x + 1$$

```
> q:=x^4+2*x^3-12*x^2-40*x-32;
> factor(q);
```

$$q := x^4 + 2\,x^3 - 12\,x^2 - 40\,x - 32$$
$$(x-4)\,(x+2)^3$$

Sievennyskomennot säilyttävät lausekkeen matemaattisesti identtisenä, joten niiden avulla saadaan kätevästi yhtälökaavoja alla olevaan tyyliin.

```
> exp(x*ln(y))=simplify(exp(x*ln(y)));
```

$$e^{x \ln(y)} = y^x$$

```
> sin(x)^2+cos(x)^2=simplify(sin(x)^2+cos(x)^2);
```

$$(\sin(x))^2 + (\cos(x))^2 = 1$$

```
> 4*cos(x)^3=combine(4*cos(x)^3);
```

$$4\,(\cos(x))^3 = \cos(3\,x) + 3\,\cos(x)$$

Trigonometrisille lausekkeille `combine` lausuu sin:n ja cos:n potenssit moninkertaisten kulmien avulla, kun taas `expand` tekee päinvastoin.

Huomautamme, että samoja lausekkeita ei tarvitse kirjoittaa kahteen kertaan, vaan voimme käyttää työarkin leikkaus/liimaus-toimintoja samalla tavoin kuin WINDOWS-, MACINTOSH- tai XWINDOW-ympäristöissä yleensäkin tehdään. Tämä koskee myös tuloksena saatavia (sinisiä) matemaattisia kaavoja, joita voidaan osina tai kokonaisina siirtää suoraan (punaiselle) syöteriville.

Grafiikkaa

Maplessa on suuri joukko monipuolisia komentoja matemaattiseen visualisointiin.

Perustavin komento on `plot`, jonka käyttö on yksinkertaisimmillaan muotoa

```
> plot(lauseke,x=a..b);
```

Esimerkiksi:

```
> plot(sin(x)*exp(-x),x=0..2*Pi);
```

tuottaa oheisen kuvan (Kuva 1) työarkille.

Osa grafiikkafunktioista on erikseen ladattavassa pakkauksessa `plots`. Otamme tässä esimerkin funktion `animate` käytöstä. Näemme, että matemaattisten animaatioiden tekeminen käy hyvin vaivattomasti. Jäljempänä olevissa esimerkeissämme tulee käyttöön muutama muu `plots`-pakkauksen funktio.

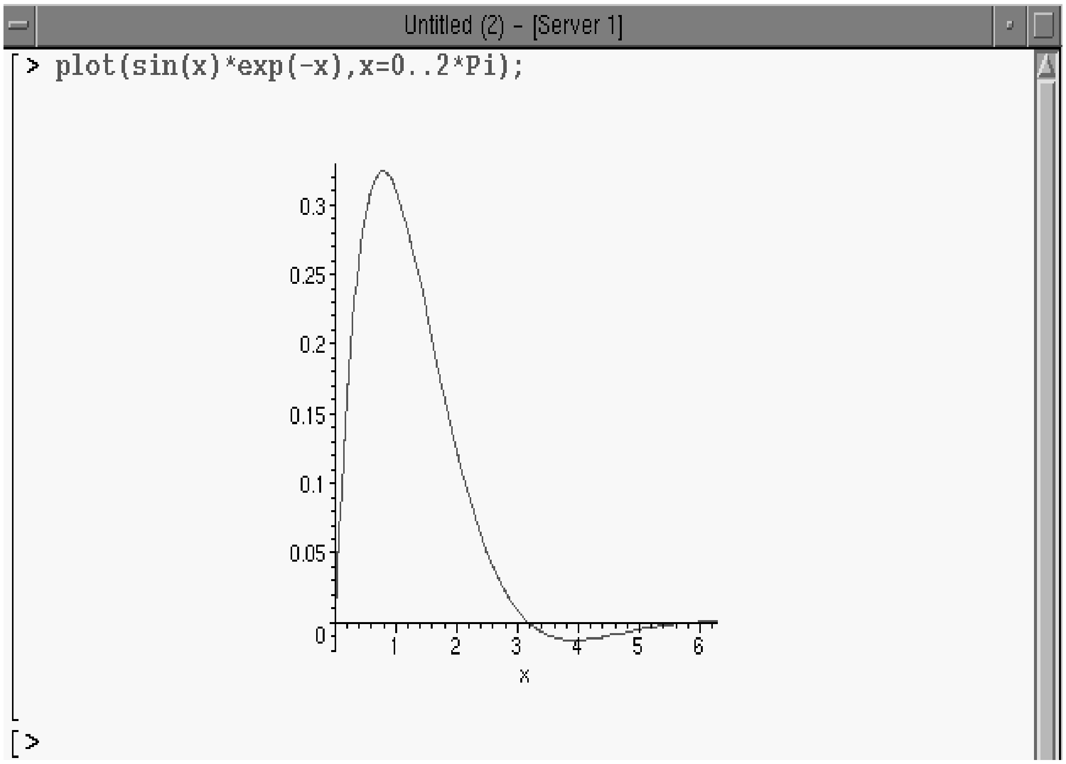
Haluamme tehdä animaation, joka havainnollistaa monomifunktioiden x^n käyttäytymistä välillä $[-1, 1]$, kun parametriä n vaihdellaan.

```
with(plots):   # Ladataan plots-pakkaus. ?plots antaa funktiolistauksen.
animate(x^n,x=-1..1,n=1..20,frames=20);
```

Tulokseksi saamme kuvan, joka hiirellä aktivoitaessa saa ympärilleen animaatiovalikon.

Työarkki voidaan tulostaa HTML-muodossa[2], jolloin animaatiot saadaan verkkoon eläviksi. Tässä ei saada kaikkia työarkin mahdollisuuksia, kuten animaation suoritusta askel kerrallaan, suunnan vaihtoa, hidastuksia/nopeutuksia ym. Ne lukijat, joilla on Maple käytössään, voivat ladata työarkin[3] Mapleen, kokeilla ja muunnella esimerkkiä.

[2] http://www.math.hut.fi/~apiola/solmu/elok99/ani.html
[3] http://www.math.hut.fi/~apiola/solmu/elok99/ani.mws



Kuva 1: Esimerkki `plot`-komennon käytöstä.

Grafiikkafunktioilla on suuri joukko kuvan asua sääteleviä valitsimia, jotka voidaan aktivoida suoraan hiirellä grafiikkavalikoista tai antaa piirtokomennon tarkentimina. Näitä voidaan kysellä esimerkiksi tyyliin `?plot` tai `?plot,options`. Jäljempänä esiintyvissä esimerkeissä varustan piirtokomennot asianmukaisilla tarkentimilla komentopohjaisesti. Tästä on se etu, että komennot voidaan ajaa uudelleen ja tuloksena on täsmälleen samanlaiset kuvat.

Pakkauksen `plots` lisäksi Maple sisältää muitakin grafiikkakirjastoja, kuten esimerkiksi `plottools`, `geometry` ja `geometry3d`. Lisäksi `stats`-pakkaus sisältää joukon tilastolliseen kuvaamiseen sopivia piirtofunktioita. Hyviä esimerkkejä grafiikkakirjastojen käytöstä on mm. kirjassa [6].

Yhtälöitä

Yhtälö ilmaistaan siis muodossa `vasen=oikea`. Yhtäsuuruusmerkkiä `=` ei pidä sekoittaa sijoitusmerkkiin `:=`. Toisen asteen yhtälö ei tuota vaikeuksia.

```
> solve(x^2-x=5,x);
```

$$\frac{1}{2}+\frac{1}{2}\sqrt{21}, \frac{1}{2}-\frac{1}{2}\sqrt{21}$$

Yhtälöryhmä ilmaistaan joukkona. Jos ratkaistavana olisi systeemi

$$\begin{cases} 2x+3y+z=1 \\ x-y-z=4 \\ 3x+7z=5, \end{cases}$$

kirjoittaisimme

```
> yhtsys:={2*x+3*y+z=1,x-y-z=4,3*x+7*z=5}:
> solve(yhtsys,{x,y,z});
```

$$\left\{x = \frac{101}{41}, z = -\frac{14}{41}, y = -\frac{49}{41}\right\}$$

Yllä ratkaisimme lineaarisen systeemin. Epälineaarisia yhtälöitä ja systeemejä voidaan vain harvoin ratkaista tarkassa symbolisessa muodossa.

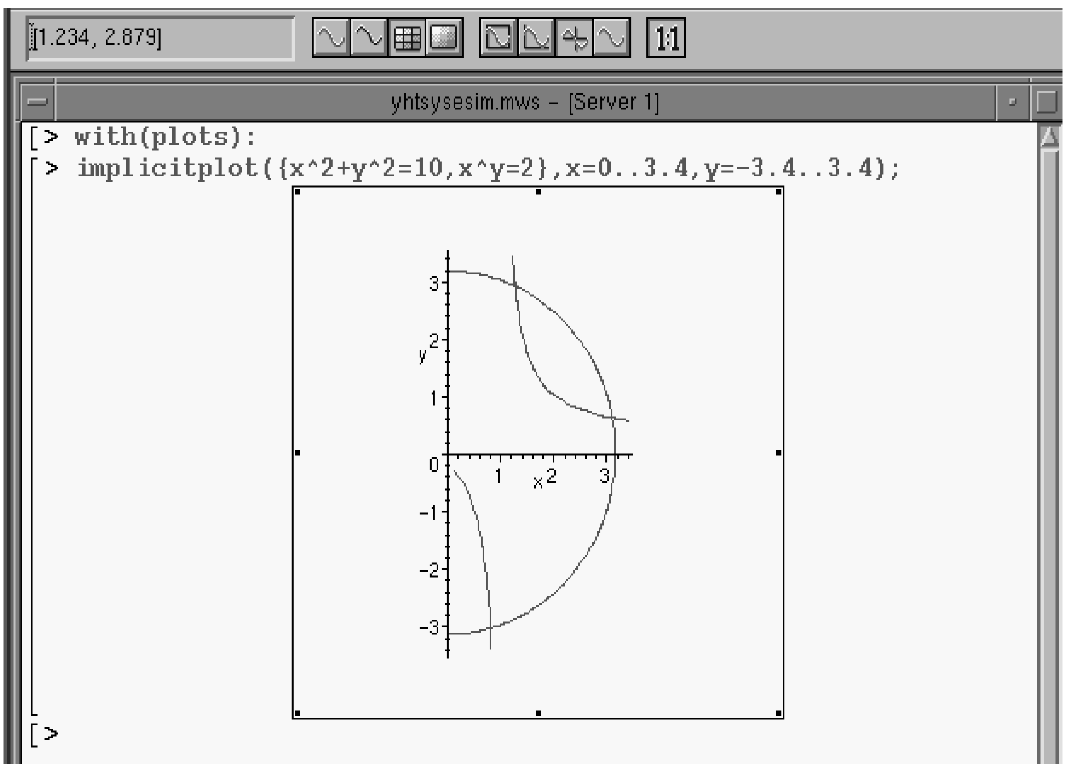
Kun halutaan numeerista ratkaisua, liitetään **f**-kirjain `solve`-sanan eteen (vrt. `evalf` edellä). Numeeriselle ratkaisijalle täytyy yleensä antaa alkuarvaus, jonka selvittämiseen voidaan käyttää grafiikkaa, numeerisia kokeiluja tai esim. fysikaalista perustetta.

Otetaan ratkaistavaksi yhtälöpari

$$\begin{cases} x^2 + y^2 = 10 \\ x^y = 2 \end{cases}$$

Lähtöarvon keksimiseksi piirrämme kuvan (Kuva 2) komennolla `implicitplot`, joka soveltuu ”implisiittisessä muodossa” annettuihin, tyyppiä $f(x, y) = c$ olevien yhtälöiden kuvaajien piirtoon. Lataamme ensin lisäpakkauksen `plots`.

```
> with(plots):
> implicitplot({x^2+y^2=10,x^y=2},x=0..3.4,y=-3.4..3.4);
```



Kuva 2: Esimerkki `implicitplot`-komennon käytöstä.

Kun osoitamme hiirellä kuvaan ja erityisesti asetamme osoittimen leikkauspisteen kohdalle (hiirikäden tarkkuudella), näemme vasemmassa yläkulmassa koordinaatit $\boxed{1.234, 2.879}$. Voimme antaa tämän alkupisteeksi numeeriselle ratkaisijalle

```
> fsolve({x^2+y^2=10,x^y=2},{x=1.23,y=2.88});
```

$$\{x = 1.270433346, y = 2.895858960\}$$

Havaitsemme, että hiirikäsi ei ollut kovin tarkka. Samalla tavoin voisimme määrittää kaksi muuta kuvassa näkyvää ratkaisupistettä.

Derivaatat

Derivointi on symboliohjelmien varhaisia riemuvoittoja. Itse asiassa lausekkeiden sievennys on vaikeampi tehtävä ja vasta kehittyneet, syvällistä algebrakoneistoa käyttävät sievennysalgoritmit ovat tehneet näistä käyttökelpoisia työvälineitä.

Symboliohjelmat osaavat suuren joukon derivoimissääntöjä, joiden turvin derivointi sujuu nopeasti ja luotettavasti. Maplessa lausekkeen derivointi tapahtuu komennolla

```
> diff(lauseke,muuttuja(jono));
```

Derivoinnin tulosta on tavallisesti syytä sieventää, usein päästään hyvään tulokseen yleissieventäjällä `simplify`, mutta tilanteesta riippuen kannattaa tietysti soveltaa erityisempiä, kuten tässä teemme. Saattaapa käydä niinkin, että `simplify` mutkistaa lauseketta, kuten kohta esitettävässä esimerkissä tapahtuu.

```
> f:=x^3*exp(x)*cos(x);df:=diff(f,x);
```

$$f := x^3 e^x \cos(x)$$
$$df := 3\,x^2 e^x \cos(x) + x^3 e^x \cos(x) - x^3 e^x \sin(x)$$

```
> factor(df);
```

$$-x^2 e^x \left(-3\, \cos(x) - x \cos(x) + x \sin(x)\right)$$

```
> collect(%,cos(x));
```

$$\left(3\,x^2 e^x + x^3 e^x\right) \cos(x) - x^3 e^x \sin(x)$$

Kokeillaan hieman mutkikkaampaa, muodostetaan x^{x^x}:n kolmas derivaatta. Korkeammat derivaatat saadaan toistamalla dervoimismuuttujaa.

```
> diff(x^(x^x),x,x,x);
```

$$x^{x^x} \left(x^x \left(\ln(x)+1\right)\ln(x) + \tfrac{x^x}{x}\right)^3 + 3\,x^{x^x} \left(x^x \left(\ln(x)+1\right)\ln(x) + \tfrac{x^x}{x}\right)$$
$$\left(x^x \left(\ln(x)+1\right)^2 \ln(x) + \tfrac{x^x \ln(x)}{x} + 2\,\tfrac{x^x (\ln(x)+1)}{x} - \tfrac{x^x}{x^2}\right) +$$
$$x^{x^x} \left(x^x \left(\ln(x)+1\right)^3 \ln(x) + 3\,\tfrac{x^x (\ln(x)+1)\ln(x)}{x} + 3\,\tfrac{x^x (\ln(x)+1)^2}{x} - \tfrac{x^x \ln(x)}{x^2} + 3\,\tfrac{x^x}{x^2} - 3\,\tfrac{x^x (\ln(x)+1)}{x^2} + 2\,\tfrac{x^x}{x^3}\right)$$

Huomaamme, että varsin yksinkertaiset operaatiot voivat johtaa massiivisiin lausekkeisiin, joita ei enää ole mahdollista hallita ilman tietokonealgebraohjelmaa. Tässä tapauksessa `simplify`-komento antaisi vielä pitemmän lausekkeen tulokseksi.

Symboliohjelma suorittaa derivoinnin tuntemiensa sääntöjen ja kaavojen nojalla. Tällaisia ovat esimerkiksi summan ja tulon derivoimissäännöt ja potenssin derivoimiskaava.

Komento `diff` ei kerro mitään derivaatasta sellaisissa pisteissä, joissa kaavoja tai sääntöjä ei voida soveltaa. Käyttäjän tulee itse olla selvillä kaavojen pätevyysalueesta. Ohjelmaa voidaan monin tavoin hyödyntää myös tilanteissa, joissa pelkät säännöt eivät toimi.

Otamme pienen esimerkki-istunnon, jossa määrittelemme yleisen erotusosamääräfunktion `eros`. Siinä jätetään f ja x vapaiksi symboleiksi (ohjelmointitermein sanottuna globaaleiksi muuttujiksi) ja otetaan vain x:n lisäys h argumentiksi. Esittelemme funktiomääritysasiat ”virallisesti” vasta ohjelmointia koskevassa kohdassa. Esimerkkifunktiona käytämme kaikkein tutuinta ei-derivoituvaa funktiota, itseisarvoa, jonka määrittelemme paloittain komennolla `piecewise`. (Emme käytä Maplen sisäänrakennettua `abs`-funktiota.)

```
> eros:=h->(f(x+h)-f(x))/h;
```

$$\mathrm{eros} := h \mapsto \frac{f(x+h) - f(x)}{h}$$

```
> f:=x->piecewise(x<0,-x,x>0,x);
```

$$f := x \mapsto \begin{cases} -x, x < 0 \\ x, x > 0 \end{cases}$$

Tutkitaan derivoituvuutta pisteessä $x = 0$.

```
> x:=0:
```

Tutkimme erotusosamäärän vasemman- ja oikeanpuoleisia raja-arvoja,

```
> limit(eros(h),h=0,left);
> limit(eros(h),h=0,right);
```

$$-1$$

$$1$$

Koska ne ovat erisuuret, ei funktiollamme ole derivaattaa 0:ssa.

Käsittelimme tahallamme triviaalin esimerkin. Usein on viisasta aloittaa tapauksella, jonka ratkaisu tunnetaan. Oikeasti kiinnostavan tapauksen suhteen tarvitsee tässä tapauksessa vain muuttaa funktion f määrittely ja tarkastelupisteestä riippuen sijoituslause `> x:=0;`

Numeeriset ja graafiset kokeilut erotusosamäärällä ja niiden opetukset niin derivaattakäsitteen kuin numeeristen ilmiöiden suhteen voisivat olla seuraavan kirjoituksen aihepiiriä.

Tässä on sopiva paikka korostaa sitä, että symboliohjelman käyttö on ihmisen ja ohjelman välistä vuorovaikutusta, eikä suinkaan mekaanista nappien painelemista. Olennaisena välineenä ratkaisuprosessissa on edelleenkin myös kynä ja paperi.

Integrointi

Integrointi sujuu käyttäjän kannalta yhtä helposti, mutta se on huomattavasti vaikeampi tehtävä ohjelmalle. Tavallisten integrointisääntöjen lisäksi Maple sisältää varsin syvällistä matemaattista koneistoa hyödyntävän ns. *Rischin algoritmin*. Maple osaa myös suuren joukon ns. erikoisfunktioita, mistä syystä osa matemaattisen tradition mahdottomiksi julistamista integraaleista on myös symbolisesti laskettavissa. Joka tapauksessa integroinnin suhteen tilanne on aivan vastaava kuin edellä esiteltyjen epälineaaristen yhtälöiden kohdalla. Valtaosa todellisissa sovellutuksissa vastaan tulevista integraaleista on laskettava numeerista approksimaatiota käyttäen.

Integraalifunktio saadaan komennolla `int(lauseke,muuttuja);` Määrätty integraali lasketaan liittämällä rajat mukaan tyyliin `int(lauseke,muuttuja=a..b);` Käytämme edellä esiteltyä yhtälötyyliä, missä estämme vasemman puolen laskennan jälleen sitaateilla.

```
> 'int(1+x+x^3+sin(x),x)'=int(1+x+x^3+sin(x),x);
```

$$\int 1 + x + x^3 + \sin(x)\, dx = x + 1/2\, x^2 + 1/4\, x^4 - \cos(x)$$

Maple ei liitä mukaan integroimisvakiota. Määrätty integraali saadaan tähän tapaan:

```
> 'int(1+x+x^3+sin(x),x=2..3)'=int(1+x+x^3+sin(x),x=2..3);
```

$$\int_2^3 1+x+x^3+\sin(x)\,dx = \frac{79}{4} - \cos(3) + \cos(2)$$

Seuraavaksi otetaan hieman vaikeampi.

```
> 'int(1/(1+x^3),x)' = int(1/(1+x^3),x);
```

$$\int \left(1+x^3\right)^{-1}\,dx = 1/3\,\ln(1+x) - 1/6\,\ln(x^2-x+1) + 1/3\,\sqrt{3}\arctan(1/3\,(2\,x-1)\,\sqrt{3})$$

Tutuin esimerkki yllä mainitusta "mahdottomaksi julistetusta" integraalista johtaa normaalijakautumaan liittyvään erf-funktioon.

```
> int(exp(-t^2),t=0..x)'=int(exp(-t^2),t=0..x);
```

$$\int_0^x e^{-t^2}\,dt = 1/2\,\mathrm{erf}(x)\sqrt{\pi}$$

Seuraavasta Maple ei ymmärrettävästikään selviä, missä tapauksessa se palauttaa annetun tehtävän sellaisenaan.

```
> int(sin(1/x^3),x=1..2);
```

$$\int_1^2 \sin(x^{-3})dx$$

Numeeriseen integrointiin siirrytään liittämällä eteen vanha tuttavamme `evalf`.

```
> evalf(Int(sin(1/x^3),x=1..2));
```

$$.3548334332$$

Jääköön Maple-kirjoista luettavaksi, miksi tässä `Int` on suositeltavampi kuin `int` .

Symbolien hallintaa ja periaatteita

Otetaan lähtökohdaksi jokin matemaattinen kaava, vaikkapa korkolaskukaava.

Otamme käyttöön muuttujan r, jolle sijoitamme arvoksi koron kaavan, missä symboleille K (pääoma), p (korkoprosentti) ja t (aika) ei ole annettu mitään arvoja.

```
> r:=K*p*t/100;
```

$$r := \frac{1}{100}\,Kpt$$

Jos annamme muuttujille arvoja, muuttuu edellä määritelty muuttuja r vastaavasti.

```
> K:=10225: p:=3: t:=1/12:
> r;
```

$$\frac{409}{16}$$

Annamme muuttujalle p uuden symbolisen arvon q ja poistamme t:ltä arvon. Arvon poistaminen, eli muuttujan vapauttaminen saadaan aikaan sijoittamalla muuttujalle arvoksi sen nimi varustettuna laskennan estävillä sitaateilla.

```
> p:=q: t:='t':
> r;
```

$$\frac{409}{4}\,qt$$

Lopuksi vapautamme kaikki muuttujat ja katsomme, mikä arvo muuttujallamme r on.

```
> K:='K': p:='p': t:='t':
> r;
```

$$\frac{1}{100}\,Kpt$$

Pääsimme näin takaisin peruskaavaan, josta lähdimme liikkeelle.

Tässä on tärkeää suorittaa juuri mainitussa järjestyksessä, siis kirjoitamme **yleisen kaavan ensin ja sijoittelemme vasta sen jälkeen** muuttujille arvoja. Jos tekisimme päinvastaisessa järjestyksessä, niin kyseisen muuttujan arvo kiinnittyisi kaavassa ja sitä ei voitaisi enää jälkeenpäin muuttaa. Otetaan pieni jatkoesimerkki tästä.

```
> restart:
> p:=5:
> r:=K*p*t/100;
                          r := 1/20 K t
> p:=6:
> r;

                             1/20 K t
```

Ilmiö johtuu ns. täysevaluaatiosta ”full evaluation”. Aina, kun muuttujalle sijoitetaan jokin arvo, joka sisältää symboleja, niin symbolien arvot lasketaan ensin, nämä arvot saattavat puolestaan sisältää symboleja, jotka puolestaan lasketaan jne. Lopuksi tuo ”täysin evaluoitu” arvo sijoitetaan ao. muuttujalle.

Otamme vielä toisen esimerkin, *napakoordinaattimuunnoksen*. Koordinaatiston piste (x, y) määräytyy antamalla pisteestä (x, y) origoon piirretyn ”napasäteen”ja positiivisen x-akselin välinen kulma φ ja pisteen etäisyys origosta, r.

Suorakulmaiset koordinaatit saadaan näiden avulla kaavoista

$$\begin{cases} x = r\cos\varphi \\ y = r\sin\varphi \end{cases}$$

Kirjoitamme ensin muunnoskaavat

```
> restart:
> x:=r*cos(phi):
> y:=r*sin(phi):
```

Olkoot pisteen napakoordinaatit $r = 1$, $\varphi = \pi/4$. Muunnoskaavoista näemme, että $x = y = 1/\sqrt{2}$.

Koska syötimme muunnoskaavat Maplelle, voimme kysyä sen mielipidettä yksinkertaisesti kirjoittamalla muuttujien nimet:

```
> r:=1: phi:=Pi/4:
> x, y;
```

$$\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}$$

Voimme näin antaa r:lle ja φ:lle mitä arvoja tahansa, muuttujiin x ja y ilmestyvät automaattisesti vastaavat xy-koordinaatit.

Tätä olisi houkuttelevaa kehitellä hiukan eteenpäin sopivaa ohjausrakennetta käyttäen ja kuvilla havainnollistaen, mutta jätetään se lukijalle ja ryhdytään sensijaan esittelemään ohjausrakenteita ja ohjelmointia.

Sitä ennen kertaamme vielä yhteenvetona, että etenimme **yleisestä erityiseen**. Kirjoitimme ensin yleisen kaavan ja vasta sen jälkeen annoimme kaavassa esiintyville muuttujille arvoja. Huomautamme, että numeerisen ohjelman tapauksessa näin ei voida menetellä, koska se ei suostu käsittelemään muuttujaa, jolla ei ole numeerista arvoa.

Ohjelmointi

Maple on myös täydellinen ohjelmointikieli, jolla voidaan määritellä omia funktioita eli proseduureja. Käyttäjä, joka ei halua varsinaisesti kirjoittaa ohjelmia, voi tehdä symboliohjelmalla hyvin paljon puhtaasti vuorovaikutteisesti. Omien matemaattisten funktioiden kirjoittamiseen kannattaa kuitenkin totuttautua, muuten käyttötapa muodostuu tarpeettoman köyhäksi. Tällä tarkoitan kohta esitettävällä "nuolityylillä" kirjoitettuja pikku funktioita. Niiden kirjoittamista en tässä pidä varsinaisena ohjelmointina.

Enemmän symboliohjelmaa käyttävä henkilö tuntee kyllä usein tarvetta laajentaa systeemiä jollakin omalla ohjelmallaan. Vaikka Maplessa on tuhansia funktioita, niin aina tulee vastaan tilanteita, joissa käyttäjä haluaisi jotain muuta, tai sitten hän ohjelmoi nopeammin kuin etsii erilaisista kirjastoista, WWW-sivuilta ym.

Annamme ohjelmointia välttelevälle lukijalle luvan lopettaa tämän kohdan lukemisen aloittaessamme *Fibonacci*-teemaa.

Edellä olemme käsitelleet matemaattisia funktioita yleensä lausekkeina. Derivaattakohdan lopulla käytimme esimerkissä myös oikeata funktiomääritystä, jonka esittelyn aika alkaa olla käsillä.

Jos kirjoitamme

```
> f:=x*sin(x);
```

voimme kohdistaa f:ään monenlaisia symbolisia operaatioita, kuten sievennys, derivointi, integrointi, kuvaajan piirtäminen ym. Sensijaan emme voi luontevalla tavalla laskea funktion arvoja eri pisteissä. Tämä johtuu siitä, että f ei ole Maplen kannalta funktio, vaan pelkästään symboleja sisältävä lauseke.

Maplessa on sinänsä hyvin hyödyllinen korvauskomento `subs`, jota voimme käyttää tyyliin

```
> fa:=subs(x=a,f);
```

Sen käyttö on kömpelöä, jos joudumme usein laskemaan arvoja eri pisteissä ja miltei toivotonta käsitellessämme yhdistettyjä funktioita.

Funktiomääritys saadaan aikaan normaalia matemaattista merkintätapaa muistuttavalla tyylillä:

```
> f:=x->x*sin(x);
```

Toisin sanottuna: *Funktio on olio f, joka liittää annettuun argumentin arvoon x laskukaavan (yleisemmin laskenta-algoritmin) antaman yksikäsitteisen arvon $f(x)$.* Onkohan tuttua matematiikan tunnilta?

Nyt voimme käytellä funktiotamme aivan normaalin matematiikan käytännön mukaisesti tähän tapaan:

```
> f(a);
```

$$a\sin(a)$$

```
> f(exp(z)*cos(z));
```

$$e^z\cos(z)\sin(e^z\cos(z))$$

```
> seq(f(x),x=0..5);
```

$$0, \sin(1), 2\sin(2), 3\sin(3), 4\sin(4), 5\sin(5)$$

Argumentteja voi olla useita, jolloin on kyse usean muuttujan funktiosta. Määrittelemme esimerkiksi kahden muuttujan funktion.

```
> g:=(x,y)->sqrt(x^2+y^2);
```

$$g := (x, y) \mapsto \sqrt{x^2 + y^2}$$

Laskemme funktion arvon pisteessä $(2, -1)$.

```
> g(2,-1);
```

$$\sqrt{5}$$

Maple tarjoaa myös vaihtoehtoisen syntaksin funktioiden eli proseduurien määrittelyyn. Jos haluaisimme kirjoittaa edellisen yksinkertaisen ja kauniin määritelmän mutkikkaasti, kömpelösti ja rumasti, voisimme kirjoittaa myös näin:

```
f:=proc(x)
x*sin(x);
end:
```

Proseduuri palauttaa viimeksi laskemansa arvon.

Tällainen perinteisen ohjelmointityylin näköinen syntaksi (jonka rumuutta hieman liioittelimme) on tarpeellinen monimutkaisempien ohjelmien tapauksessa. Palaamme asian havainnollistamiseksi Jouni Seppäsen kirjoituksen *Fibonacci*-teemaan. Näiden lukujen merkitys matematiikan ja sen sovellusten kannalta jää usein opiskelijoilta näkemättä. Kirjoituksemme yhteydessä ei ole mahdollisuutta korjata tätä asiantilaa. Todettakoon vain, että nuo luvut tarjoavat ainakin ohjelmoinnin opetukseen hyvin soveltuvan, helposti ohjelmoitavan algoritmin, jolla voidaan valaista monia näkökulmia niin algoritmien erilaisen toteutuksen, tehokkuuden, rekursion ym. suhteen. Näitä seikkojahan valotettiin monipuolisesti Jouni Seppäsen kirjoituksessa.

Kirjoitamme aluksi rekursiivisen funktion, jonka koodi samalla palauttaa lukijan mieleen Fibonaccin lukujen määritelmän.

```
Fib:=proc(n::nonnegint)
     option remember;
       if n=0 then 0
       elif n=1 then 1
       else Fib(n-1)+Fib(n-2)
       fi
     end:
```

Lause ”option remember” tallettaa lasketut arvot proseduurin ns. muistitauluun, jolloin laskettuja arvoja ei lasketa uudelleen.

Kyseessä on Jouni Seppäsen kirjoituksessa ”naiiviksi Fibonacciksi” kutsuttu toteutus, jota on parannettu tuolla muistamispiirteellä.

Kokeillaan hieman rajoja.

```
> restart:
> Fib(1000);
Error, (in type/nonnegint) too many levels of recursion
```

Lasketaan vaiheittain, jotta voimme hyödyntää muistamisominaisuutta.

```
> Fib(300);
   222232244629420445529739893461909967206666939096499764990979600
> Fib(500): # Laskemme, mutta emme tulosta.
> Fib(700): # Laskemme, mutta emme tulosta.
> Fib(1000);
434665576869374564356885276750406258025646605173717804024817290895\
    365554179490518904038798400792551692959225930803226347752096896\
    232398733224711616429964409065331879382989696499285160037044761\
    3779516684922887 5
```

Nytpä onnistui.

Fibonacci-ohjelma ei ole kuitenkaan tyypillistä symboliohjelmointia, siinä ei näy muita Maplen erityisominaisuuksia kuin rajoittamaton kokonaislukulaskentatarkkuus.

Teemme vielä version, jossa tulee ainakin hiukan symbolinkäsittelymahdollisuuttakin mukaan. Yleistämme tilannetta niin, että jätämme myös alkuarvot käyttäjän valittaviksi parametreiksi, jotka voivat olla myös symbolisia. Annamme ohjelmamme palauttaa koko jonon alusta alkaen. Kirjoitamme tällä kertaa ei-rekursiivisesti.

```
    yFib:=proc(a,b,n::nonnegint)
    local luvut,i;
    luvut[0]:=a;luvut[1]:=b:
    for i from 1 to n-1 do
        luvut[i+1]:=luvut[i]+luvut[i-1]
                        od;
    seq(luvut[i],i=0..n);
    end:
```

Kutsu jonon $f_0, \ldots f_{10}$ muodostamiseksi olisi

```
> yFib(0,1,10);
```

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55$$

Voimme kutsua funktiotamme myös symbolisilla lähtöarvoilla, jolloin Maple sieventää tuloksen suoraan muotoon, jossa jonon k:s alkio näkyy olevan $f_{k-1}a_0 + f_k a_1$, missä f_k tarkoittaa Fibonacci-lukua, jonka indeksi on k.

```
> yFib(a[0],a[1],9);
```

$$a_0, a_1, a_1 + a_0, 2\,a_1 + a_0, 3\,a_1 + 2\,a_0, 5\,a_1 + 3\,a_0, 8\,a_1 + 5\,a_0, 13\,a_1 + 8\,a_0, 21\,a_1 + 13\,a_0, 34\,a_1 + 21\,a_0$$

Tällainen käyttötapa voi olla hyödyllinen monissa yhteyksissä ohjelmakoodin toimintaa selviteltäessä.

Enemmän matematiikkaa ja symbolisia operaatioita sisältävien ohjelmien suhteen viittaamme kirjan [2] lukuun 5 ja muihin viitteinä oleviin Maple-kirjoihin. Erityisesti Maple-ohjelmoinnille omistettu kirja on [7] .

Huomautamme, että Maplen tulkkausluonteesta johtuen ohjauslauseita voidaan käyttää myös Maple-istunnossa suoraan. Jos todella haluaisimme vain saada käsiimme alkupään Fibonacci-lukuja vaikkapa 50 kpl, riittäisi kirjoittaa

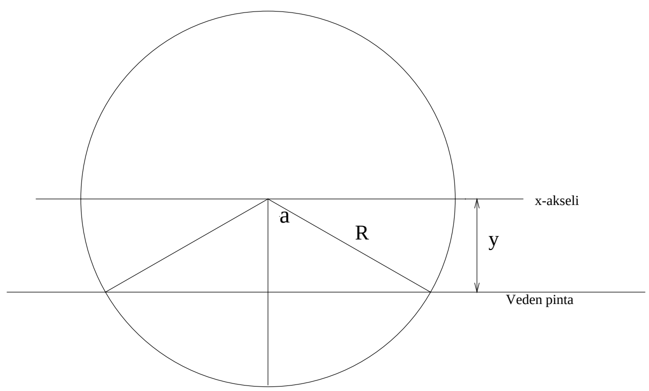
```
> f[0]:=0: f[1]:=1;
> for i from 2 to 50 do f[i]:=f[i-1]+f[i-2] od:
```

Esimerkkiprojektina kelluva lieriö

Lopuksi esitämme tehtävän, joka voisi olla vaikkapa pienen projektityön aiheena. Tehtävään sisältyvä matematiikka rajoittuu aika vähiin, lukion suppea oppimäärä on riittävä. Ainoa vakava lukion kurssin ylittävä kohta on epälineaarisen yhtälön ratkaiseminen, mutta se voidaan jättää Maple-ohjelman huoleksi.

Tehtävä. Miten syvällä vedessä kelluu alumiiniputki, jonka säde $R = 60$ cm ja vaipan paksuus $d = 1$ cm ? Putki tarkoittaa onttoa sylinteriä, joka on päistään suljettu ohuilla "painottomilla" levyillä. Alumiinin tiheys $\rho = 2.7$ g/cm^3.

Ratkaisu. Ideana on, että sylinterin veden alla olevan osan kokoisen vesimäärän paino antaa *Arkhimedeen lain* mukaisen nostevoiman, joka pitää tasapainossa sylinterirenkaan painon. Jälkimmäisen saamme kertomalla tiheydellä (ja g:llä) sisäkkäisten sylinterien tilavuuksien erotuksen.



Kuva 3: Sylinteri vedessä

Nostevoiman saamme kertomalla sylinterin pohjaympyrän vedenalaisen segmentin pinta-alan sylinterin pituudella (ja g:llä).

Kuvassa (Kuva 3) olemme valinneet koordinaatiston origoksi pohjaympyrän keskipisteen ja merkitsemme vedenpinnan ja x-akselin välistä pystysuoraa etäisyyttä y:llä siten, että pidämme y:tä positiivisena, jos veden pinta on x-akselin alapuolella, vastakkaisessa tapauksessa negatiivisena.

Keskuskulmaa 2α vastaavan kolmion pinta-ala $= y\sqrt{R^2 - y^2}$. Jos y otetaan positiivisena alaspäin ja negatiivisena ylöspäin, kuten sovimme, saadaan segmentin ala kaikissa tapauksissa kaavalla

$$\frac{2\alpha}{2\pi}\pi R^2 - y\sqrt{R^2 - y^2}.$$

Ratkaisutyöarkki on HTML:ksi muunnettuna kokonaisuudessaan luettavissa viitteestä
http://www.math.hut.fi/~apiola/solmu/sylinteri.html . Oikea Maple-työarkki on saatavissa
WWW-sivulta http://www.math.hut.fi/~apiola/solmu/sylinteri.mws .

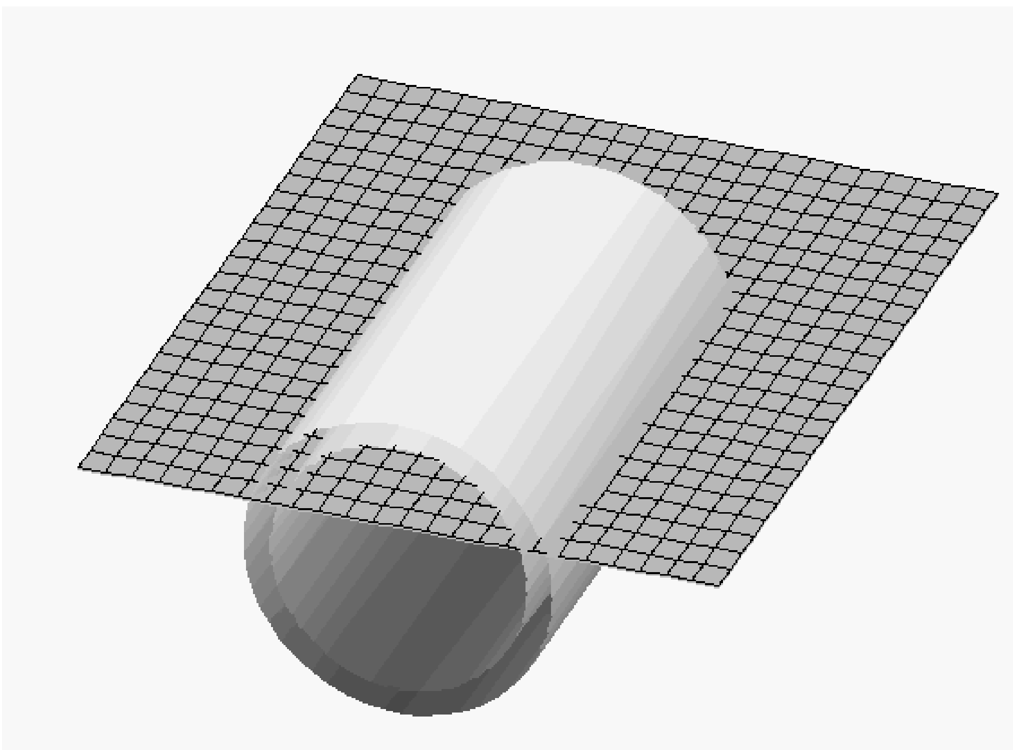
Ratkaisutyöarkin LaTeX-muunnos hieman lyhenneltynä

Piirrämme kuvan, jossa näkyy putki ja veden pinta. Piirretään ensin sylinteri. Tähän voidaan käyttää `plots`-pakkauksen `plot3d`-funkitota, jolla saadaan yleinen parametrimuotoinen pinta piirretyksi. (Tämä menee hieman yli koulukurssin, mutta nyt on pääasia, että saadaan kuva tavalla tai toisella aikaan.)

```
> with(plots):
> x:=t: y:=cos(theta): z:=sin(theta):
> X:=t: Y:=1.2*cos(theta): Z:=1.2*sin(theta):
```

Muodostamme 1-säteisen sylinterin $[x, y, z]$ ja 1.2-säteisen $[X, Y, Z]$, joiden akselina on x-akseli. Jos kiinnitetään t, saadaan ympyrä tasossa $x = t$, sylinteripinta muodostuu, kun annetaan t:n vaihdella. Piirrämme eri mitoilla, liioittelemme havainnollisuuden vuoksi putken paksuutta.

```
> sisasylinteri:=plot3d([x,y,z],t=-2..2,theta=0..2*Pi,axes=none,grid=[2,
> 50],scaling=constrained,orientation=[20,60],style=patchnogrid,...):
> ulkosylinteri:=plot3d([X,Y,Z],t=-2..2,theta=0..2*Pi,axes=none,grid=[2,
> 50],scaling=constrained,style=patchnogrid,shading=zgreyscale):
> vesi:=plot3d(1/3,xx=-4..2,yy=-2.5..2.5,shading=zgreyscale):
> display({sisasylinteri,ulkosylinteri,vesi});
```



Kuva 4: Pohjaympyrän ja vedenalaisen segmentin alan laskenta.

```
> x:='x': y:='y': z:='z':
```

Käytämme ainakin y:tä vapaana symbolina, joten se täytyy vapauttaa edellisestä arvostaan.

Nosteen laskemiseksi tarvitsemme vedenalaisen lieriösegmentin tilavuuden, jota varten laskemme ulkolieriön pohjasegmentin pinta-alan. Piirretään pohjaympyrä ja vedenpinta. Annamme komennot, mutta jätämme kuvan näyttämättä, koska samanlainen kuva oli jo edellä.

```
> restart: with(plots):
> R:=30: x:=R*cos(theta): y:=R*sin(theta):
> ympyra:=plot([x,y,theta=0..2*Pi]):
> t1:=-Pi/6: t2:=Pi+Pi/6: x1:=R*cos(t1): y1:=R*sin(t1): x2:=R*cos(t2):
> y2:=R*sin(t2): kolmio:=plot([[0,0],[x1,y1],[x2,y2],[0,0]]):
> vesi:=plot(y1,xaks=-40..40,color=blue):
> mittaviiva:=plot([[32,y1],[32,0]],color=black):
> y:='y': yteksti:=textplot([33.5,-7.5,"y"]):
> alteksti:=textplot([3,-5,"a"]):
> display({kolmio,ympyra,vesi,mittaviiva,yteksti,alteksti});
```

Jos ympyrä ei näytä ympyrältä, kannattaa näpäyttää kuvavalikon $\boxed{1:1}$ -painiketta.

Olemme käyttäneet monia muuttujia, nyt on syytä aloittaa puhtaalta pöydältä.

```
> restart:
> kolmion_ala:=y*sqrt(R^2-y^2);
```

$$\text{kolmion\_ala} := y\sqrt{900-y^2}$$

Huom! Tässä muodostettiin merkillä varustettu ala.

```
> sektorin_ala:=(2*alpha)/(2*Pi)*Pi*R^2;
```

$$\text{sektorin\_ala} := 900\,\alpha$$

```
> segmentin_ala:=sektorin_ala-kolmion_ala;
```

$$\text{segmentin\_ala} := 900\,\alpha - y\sqrt{900-y^2}$$

```
> noste:=g*segmentin_ala*syl_pituus;
```

$$\text{noste} := g\left(900\,\alpha - y\sqrt{900-y^2}\right)\text{syl\_pituus}$$

```
> putken_tilavuus:=(Pi*R^2-Pi*(R-d)^2)*syl_pituus;
```

$$\text{putken\_tilavuus} := \left(900\,\pi - \pi\,(30-d)^2\right)\text{syl\_pituus}$$

```
> putken_paino:=simplify(rho*g*putken_tilavuus);
```

$$\text{putken\_paino} := -\rho\,g\pi\,d\,(-60+d)\,\text{syl\_pituus}$$

Muodostamme tasapainoehtoyhtälön.

```
> ehto:=noste=putken_paino;
```

$$\text{ehto} := g\left(900\,\alpha - y\sqrt{900-y^2}\right)\text{syl\_pituus} = -\rho\,g\pi\,d\,(-60+d)\,\text{syl\_pituus}$$

Maple ei suorita puolittain jakamista, ennenkuin vaadimme sitä:

```
> ehto:=ehto/(g*syl_pituus);
```

$$\text{ehto} := 900\,\alpha - y\sqrt{900-y^2} = -\rho\,\pi\,d\,(-60+d)$$

Yhtälön sieventäminen käy parhaiten muodostamalla vasemman puolen (`lhs` - left hand side) ja oikean puolen (`rhs` - right hand side) erotus.

```
> ehto:=simplify(rhs(ehto)-lhs(ehto))=0;
```

$$\text{ehto} := 60\,\rho\,\pi\,d - \rho\,\pi\,d^2 - 900\,\alpha + y\sqrt{900-y^2} = 0$$

```
> alpha:=arccos(y/R);
```

$$\alpha := \arccos(1/30\,y)$$

Tarkistetaanpa, miten Maple ymmärtää arccos-funktion päähaaran. Kuva (jota emme tässä tulosta) näyttää, että se sopii juuri tarkoitukseemme.

```
> plot(arccos(x),x=-1..1);
> ehto;
```

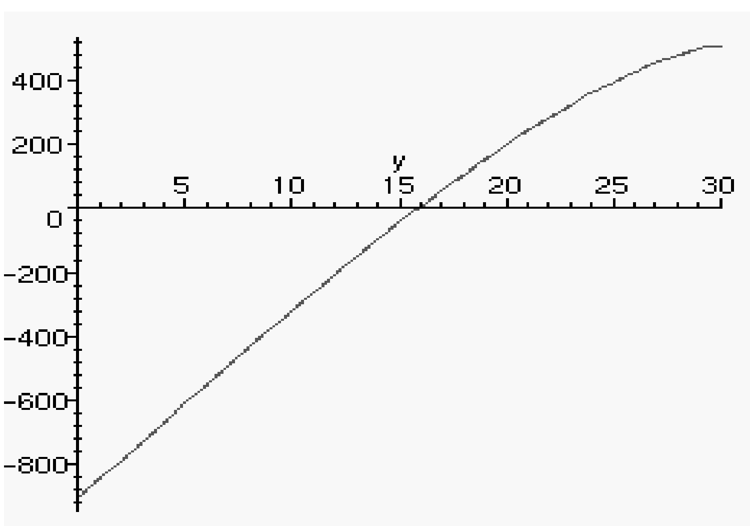
$$60\,\rho\,\pi\,d - \rho\,\pi\,d^2 - 900\,\arccos(1/30\,y) + y\sqrt{900-y^2} = 0$$

Tässä vaiheessa otamme numeroarvot mukaan. Muistamme, että kun lausekkeessa esiintyville muuttujille annetaan arvoja, ne sijoittuvat automaattisesti lausekkeeseen. (Muistele korkoa ja napakoordinaatteja!)

```
> R:=30: d:=1: rho:=2.7:
> ehto;
```

$$159.3\,\pi - 900\,\arccos(1/30\,y) + y\sqrt{900-y^2} = 0$$

Edelleenkin näyttää jokseenkin kaamealta yhtälöltä. Vaan onpa meillä välineet sen selvittelyyn. Katsotaan ensin, miltä yhtälön vasemman puolen kuvaaja näyttää (Kuva 5), jotta saamme käsityksen ratkaisujen olemassaolosta, lukumäärästä ja sijainnista.



Kuva 5: Yhtälön vasemman puolen kuvaaja.

```
> F:=lhs(ehto);
```

$$F := 159.3\,\pi - 900\ \arccos(1/30\,y) + y\sqrt{900 - y^2}$$

```
> plot(F,y=0..30);
```

Annamme kuvasta katsomamme karkean likiarvon $y = 16$ numeerisen ratkaisijan lähtöpisteeksi.

```
> yarvo:=fsolve(ehto,y=16);
```

$$\mathrm{yarvo} := 16.01860140$$

```
> syvyys:=R-yarvo;
```

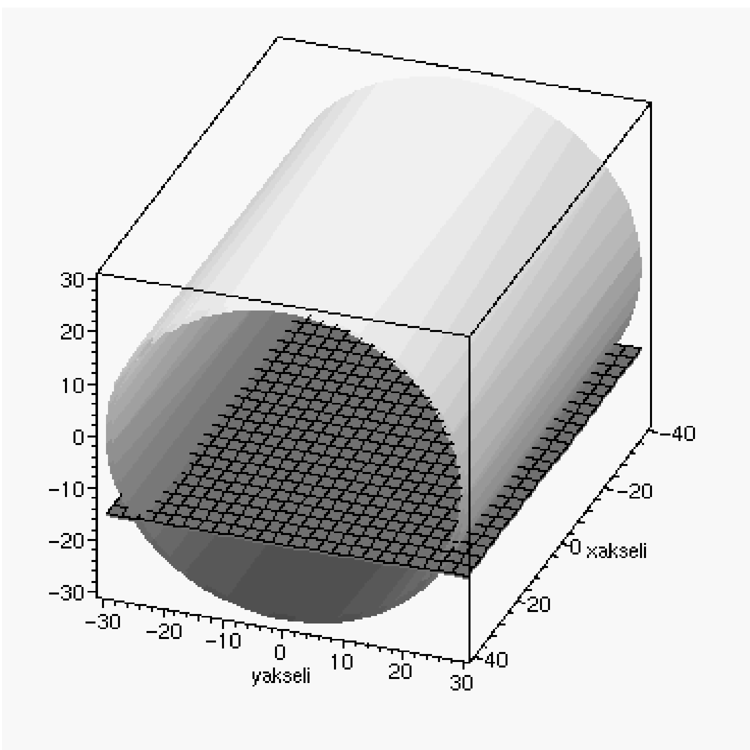
$$\mathrm{syvyys} := 13.98139860$$

Piirrämme lopuksi putken ja vedenpinnan oikeilla arvoilla.

```
> X:=t: Y:=R*cos(theta): Z:=R*sin(theta): x:=t: y:=(R-d)*cos(theta):
> z:=(R-d)*sin(theta):
```

Piirtokomennot ovat periaatteessa aivan samat kuin ensimmäisessä kuvassa, siksi emme niitä näytä tämän enempää.

```
> display({sisasylinteri,ulkosylinteri,vesi});
```



Kuva 6: Sylinteri vedessä.

Huomaamme, että oikeilla mitoilla varustettuna sylinteri näyttää hyvin ohuelta, niinpä ei olekaan ihme, että se kelluu näin ylhäällä.

Loppumietteitä

Olemme nähneet joitakin symboliohjelman periaatteita ja mahdollisuuksia. Tämänkaltaisen ohjelman avulla rutiinioperaatioita voidaan siirtää koneelle ja antaa enemmän tilaa luovalle ajattelulle ja ideoinnille. Kehittynyt graafinen käyttöliittymä houkuttelee kirjoittamaan työstä viimeistellyn dokumentin ja antaa siten välineet projektiluonteisiin suorituksiin.

Koulumatematiikassakin voidaan ottaa nykyistä vaativampia ja siten kiinnostavampia tehtäviä eristämällä jokin ratkaisun osa ”mustaksi laatikoksi”, kuten edellä teimme epälineaarisen yhtälön ratkaisijan suhteen. Erillisenä projektina voitaisiin käsitellä käytetyn mustan laatikon toiminnan selvittelyä.

Toki on syytä muistaa, että myös käsin tapahtuva kaavojen sieventäminen ja niillä operoiminen on edelleenkin välttämätön taito, jota ohjelmalla ei voi korvata. Paras tulos ohjelman käytössä saavutetaan tietenkin siten, että ihminen panee myös koko luovan panoksensa peliin.

Rantasalmella elokuussa 1999

Heikki Apiola
Matematiikan laitos
Teknillinen korkeakoulu

Viitteet

[1] G. Andersson. *Applied Mathematics with Maple.* Studentlitteratur, 1997.

[2] H. Apiola. *Symbolista ja numeerista matematiikkaa Maple-ohjelmalla.* Otatieto, 1998.

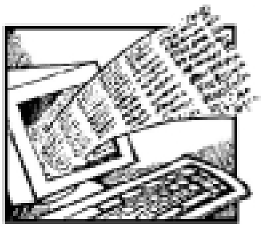
[3] K.M. Heal, M.L. Hensen, and K.M. Rickard. *Maple V Learning Guide.* Springer, 1996.

[4] A. Heck. *Introduction to Maple.* Springer, 1996. 2nd edition.

[5] R.B. Israel. *Calculus the Maple way.* Addison Wesley, 1996.

[6] M. Kofler. *Maple An Introduction and Reference.* Addison Wesley, 1997.

[7] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, and S.M. Vorkoetter. *Maple V Programming Guide.* Springer, 1996.



Symbolista merkistöä verkossa sekä WWW-pohjaisia opetusjärjestelmiä

Vierailin 5.–11.10.1999 Yhdysvalloissa Florida State Universityssä. Yliopisto sijaitsee Tallahasseessa, Floridan osavaltion pääkaupungissa Pohjois-Floridassa lähellä Meksikonlahden rannikkoa. Tallahasseen alueen sanotaankin muistuttavan kulttuurillisesti enemmän Etelä-Georgiaa kuin Floridan niemimaata. Tutustuin Florida State Universityssa työskentelevän matematiikan professori Mika Seppälän opastuksella kaupallisen BlackBoard-ohjelmiston käyttöön sikäläisessä matematiikan opetuksessa, ja seurasin perjantaina 8.10. järjestettyä workshopia *Symbolic Notations on the Web*.

Symbolic Notations on the Web

World Wide Webin dokumentit ovat valtaosin HTML-kielellä laadittuja. HTML:n yksinkertaisuudesta ja suppeudesta johtuen kaikenlaisen tavallisesta tekstistä poikkeavan ladonta-asun ja symbolisen merkistön kuvaaminen on ongelmallista. Matemaattinen dokumentti sisältää tunnetusti runsaasti kaavoja ja erikoismerkistöä. WWW:ssä kaavat ja erikoismerkit joudutaan esittämään kuvina, jolloin hyperdokumentin tiedostokoko kasvaa helposti hyvin suureksi ja ulkoasu ei välttämättä ole kovin hyvälaatuinen.

Monissa tavallisimmissa tekstinkäsittelyohjelmissa on valikoima erikoismerkistöä ja kaavaeditori, joiden avulla matemaattisen tekstin kirjoittaminen on mahdollista. Monipuolisin ja levinnein matemaattisen tekstin ladontaväline on kuitenkin TeX-ladontaohjelma ja siihen perustuva LaTeX-makropaketti. LaTeX:ista löytyy hyvin kattava valikoima erilaisia tyylimäärittelyjä, sekä monipuoliset mahdollisuudet erilaisten erikoismerkkien ja kaavojen ladonta-asun määrittelyyn. Vaikka TeX on ohjelmistona jo suhteellisen vanha, se on säilyttänyt asemansa hyvin. Käytännössä TeX:istä on muodostunut matemaattisen tekstinkäsittelyn standardi, joka on useimmissa matemaattisissa julkaisuissa suositeltu, ja joissain suorastaan vaadittu julkaisumuoto. Niinpä näillä näkymin tulevat symbolisen merkistön julkaisutekniikat eivät ole syrjäyttämässä TeX:ia, vaan integroituvat siihen.

TeX:in isänä voidaan pitää Stanfordin yliopiston emeritusprofessori Donald E. Knuthia. Alun perin Knuth ryhtyi kehittämään TeX:iä saadakseen haluamansa ladonta-asun klassikoksi muodostuneelle kirjasarjalleen *The Art of Computer Programming*. Ensimmäisenä TeX ja Metafont-julkaisujärjestelmillä tuotettuna suuremman luokan julkaisuna Knuth mainitsee *The Art of Computer Programming* -sarjan teoksen *Seminumerical Algorithms* toisen painoksen vuodelta 1980. *The Art of Computer Programming* -sarjan uusimmat painokset ovatkin hyviä esimerkkejä TeX:illä tuotetun dokumentin ulkoasusta ja TeX:in mahdollisuuksista.

TeX laadittiin alun perin hyvänlaatuisten lineaaristen paperidokumenttien tuottamista varten. Hyvien hyperdokumenttien tulisi kuitenkin olla epälineaarisia ja merkintöjen määrittelyn pitäisi sisältää ladonta-asun määritelmien lisäksi myös tietoa symbolisten merkintöjen merkityksistä. Jos esimerkiksi verrataan polynomia $ax^4 + bx^3 + cx^2 + dx + e$, eksponenttifunktiota $e^{i\beta}$ ja integraalilauseketta $\int 3x\,dx$, huomataan että merkinnät e ja dx esiintyvät kyseisissä lausekkeissa keskenään erilaisissa merkityksissä. Lausekkeiden semanttista rakennetta koskevaa informaatiota sisältävä määrittelytapa mahdollistaa myös niin sanotun alykkään tekstinkäsittelyn, jossa lausekkeita ja kaavoja voidaan siirtää ja välittää eri ohjelmien välillä siten, että merkintöjen ulkoasu ei säily muuttumattomana, vaan on kulloinkin käytettävän ohjelman kannalta tarkoituksenmukainen. Tällöin olisi esimerkiksi mahdollista leikata WWW-sivulta jokin ulkoasultaan TeX:iä muistuttava lauseke ja siirtää se symbolilaskentaohjelmaan, jolloin lauseke muuttuisi kyseisen ohjelman komennoksi, ja sillä voitaisiin suorittaa laskutoimituksia.

Symbolisten lausekkeiden ladonta-asun määrittelyä varten on kehitteillä kuvauskieli MathML. Sen ohella kehitteillä on OpenMath-kuvauskieli, joka on tavallaan MathML:n laajennus, jolla voidaan määritellä myös lausekkeiden merkityksiä. Jo nyt joidenkin ohjelmistojen uusimmat versiot, esimerkiksi Mathematica 4, tukevat MathML:aa. Sen sijaan ei ole tarkkaa tietoa siitä, koska yleisimmät WWW-selaimet Netscape ja Microsoft Internet Explorer alkavat tukea sitä. OpenMathin kehitystyön parissa työskenteleviä organisaatioita ovat muun muassa W3 Math Working Group, NAOMI (The North American OpenMath Initiative), OpenMath Society ja OpenMath Escript Project. Teollisuuden puolelta mukana ovat muun muassa IBM Techexplorer ja Waterloo Maple Inc. Esitetyn arvion mukaan Netscapen pitäisi tukea OpenMathia 1–2 vuoden kuluessa. Tarkempaa arviota ei OpenMathin osaltakaan esitetty. Sopii kuitenkin olettaa, että MathML tulee yleistymään jo jonkin verran ennen OpenMathia. Muita sellaisia ohjelmistoja, jotka alkavat tukea OpenMathia ja MathML:aa, ovat David Carlislen esittämän arvion mukaan mm. (lista ei ole täydellinen) Maple, WebEq, Techexplorer, Mozilla, Amaya, Mathematica, Reduce, MathType ja Microsoft Word. Tällä tavoin toteutuessaan OpenMath mahdollistaisi dokumenttien siirron laskentaohjelmien, ladontaohjelmien ja tekstinkäsittelyohjelmien välillä.

A. Diaz IBM:n tutkimuspuolelta esitteli IBM:n Techexplorer-ohjelmaa. Techexplorerin avulla TeX-muotoisia dokumentteja voidaan esittää WWW:ssä. Ohjelman perusversio on ainakin toistaiseksi saatavissa ilmaiseksi IBM:n WWW-sivuilta, ja professional-version hinta on Yhdysvaltain dollareissa $29.95.

Techexplorer asennetaan WWW-selaimen laajennukseksi (plug-in). Tällöin se näyttää TeX-muotoisen dokumentin selaimessa TeX:in ladonta-asun mukaisesti. Dokumenttiin voidaan määritellä hyperlinkkejä, joiden linkkiankkureina voivat olla joko kirjaimet, kuvat tai mitkä tahansa symboliset merkit. Niinpä esimerkiksi kaavan voi halutessaan määritellä hyperlinkin linkkiankkuriksi. Techexploreria käytettäessä on myös mahdollista yhdistää erilaisia protokollia. TeX-dokumenttiin voidaan liittää kuvia, animaatioita, ääntä ja liikkuvaa kuvaa. WWW-sivu voidaan jakaa kehyksiin (frame) sillä tavoin, että joissain lohkoissa on tavallinen HTML-osio, ja joissain Techexplorerin näyttämä TeX-muotoinen osio. Techexplorerin ja Java-aplettien yhteiskäyttö on myöskin mahdollista. Techexplorerin näyttämillä dokumenteilla ei ole yhtä oletuskokoa, joten ikkunan ja fonttien kokoa voi muuttaa.

Jos vertaillaan WWW-selaimen laajennukseksi asennetulla Acrobat Readerilla luettavaa PDF-muotoista dokumenttia ja Techexplorerilla luettavaa TeX-dokumenttia, on Techexplorerin käytön selvänä etuna se, että TeX-muotoisten dokumenttien tiedostokoko on huomattavasti ulkoasultaan vastaavia PDF-muotoisia dokumentteja pienempi. Tällöin niiden lukeminen verkossa on sujuvampaa. Lisäksi vältytään yhdeltä työvaiheelta, kun TeX-dokumenttia ei tarvitse muuntaa PDF-muotoon, vaan se voidaan esittää suoraan.

Tällä hetkellä Techexplorer toimii Unix-, Linux-, Windows- ja NT-käyttöjärjestelmissä. Se ei siis toistaiseksi tue Macintoshia, mutta Diazin mukaan myös Macintosh-tuki on tulossa.

Florida State Universityn tietojenkäsittelytieteen laitoksella työskentelevät apulaisprofessori R. van Engelen ja jatko-opiskelija Andreas Stortmann esittelivät kumpikin laatimaansa tietokoneavusteista matematiikan opiskelu- ja harjoitusmateriaalia. Molempien materiaalissa tekninen puoli oli korostetusti esillä.

van Engelen oli laatinut systeemin, jonka avulla voi suorittaa WWW:ssä yksinkertaisia laskutoimituksia antamalla komennnot samaan tapaan kuin numeerisen ja symbolisen laskennan ohjelmassa. Taustalla oleva ohjelma suorittaa laskutoimitukset ja palauttaa tuloksen WWW-sivulle. Vastaustulosteiden ulkoasu on hyvin siisti. van Engelenin harjoituksista löytyy lisätietoja ja esimerkkejä hänen kotisivunsa (ks. WWW-osoitteita) takaa. Hänen tuotoksiinsa kuuluu myös systeemi, joka hakee sattumanvaraisesti tietyt tehtävät tietokannasta aina kun kyseinen tehtäväsivu avataan. Näin ollen sivu on jokaisen avaamisen tai uudelleenlatauskomennon jälkeen erilainen kuin edellisellä kerralla.

Andreas Strotmann on kehitellyt hiukan samantyyppisiä harjoitus- ja koetehtäviä kuin van Engelenkin. Myös Strotmannin suunnitelmiin kuuluu systeemi, jossa kone generoisi eri kerroilla erilaisia kysymyksiä tietokannasta. Kysymykset olisivat tyypiltään sekä yksinkertaisempia täydennystehtäviä että monimutkaisempia, pedagogisesti hyviä monivalintakysymyksiä. Strotmannin tavoitteena on lisäksi kehittää järjestelmää siten, että oppija saisi palautetta vastauksistaan. Lisäksi hänellä oli ajatuksia sellaisista ominaisuuksista, että ohjelma mahdollistaisi virheanalyysin tekemisen tyypillisistä oppilaiden tekemistä virheistä, ja tyyppivirheet kirjautuisivat jonkinlaiseksi omaksi tietokannakseen. Strotmann pohti esityksessään myös sitä, millainen harjoituspaketin ja kysymysten rakenne olisi kognitiivisesti mielekäs. Lisäksi hän totesi usean muun tahon tapaan, että silloin kun numeerisen ja symbolisen laskennan ohjelmia hyödynnetään opetuksessa, niiden evaluaatioastetta tulee joskus opiskelutilanteessa ja -materiaalissa hillitä, jotta ohjelma ei heti suorittaisi laskutoimituksia oppijan ja oppimisen näkökulmasta katsottuna ”liian pitkälle” ilman välivaiheita.

WWW-pohjaiset opetusjärjestelmät

Kaupalliset opetusjärjestelmät, jotka tarjoavat valmiin käyttöliittymän ja yksinkertaisia työkaluja omien kurssisivujen tekemiseksi ovat nopeasti yleistymässä. Järjestelmien etuna on muun muassa se, että kurssin pitäjältä ei kulu aikaa itse käyttöliittymän laatimiseen ja ylläpitämiseen. Tunnetuimpia, ja keskenään samantyyppisiä, WWW-pohjaisia opetusjärjestelmiä ovat WebCT, Lotus LearningSpace ja BlackBoard.

BlackBoard Inc. tarjoaa palvelimellaan mahdollisuuden luoda ja kehittää ilmaiseksi WWW-muotoinen kurssi BlackBoardin käyttöliittymää hyödyntäen. Tarvittaessa opastus kurssisivujen luomiseen on hyvinkin kädestäpitävää, joten kurssisivujen laatijan ei tarvitse välttämättä osata edes HTML:ää. Toisaalta sivuille voidaan liittää normaaleja verkossa esitettäviä HTML-dokumentteja sekä esimerkiksi selaimen laajennukseksi (plug-in) asennetun Techexplorerin avulla luettavia TeX-dokumentteja tai selaimen laajennukseksi asennetun Acrobat Readerin avulla luettavia PDF-dokumentteja. Näin ollen pohjamateriaali voi siis olla BlackBoardista riippumatonta. Ne kurssisivut, joita voi laatia ilmaiseksi, tulevat sijaitsemaan BlackBoardin palvelimella.

BlackBoardin kaupallinen versio on ohjelmasovellus CourseInfo, joka mahdollistaa paikallisen opiskeluympäristön luomisen.

Florida State Universityssa BlackBoardia käytetään useamman tieteenalan opetuksessa. Tällöin etuna on muun muassa se, että oppilaan on myöhemmin helpompi käyttää BlackBoardia, kun ohjelman perusrakenne on jo tullut jonkin kurssin yhteydessä tutuksi. Oppilaiden suhtautuminen BlackBoardin käyttöön on varsin kaksijakoista; toiset kokevat tietokoneiden käytön selkeästi myönteisenä asiana, kun taas toiset pitävät sitä kaiken muun kurssiaineksen lisäksi tulevana ylimääräisenä rasitteena. Harva suhtautuu tietokoneiden käyttöön neutraalisti, ja suhtautumiseen vaikuttaa epäilemättä aikaisempi kokemus tietokoneiden käytöstä.

Professori Mika Seppälä esitteli parhaillaan käynnissä olevaa matematiikan peruskurssiaan, jonka opetuksessa hän hyödyntää BlackBoard-ohjelmistoa. Kurssiin kuuluvat WWW-materiaalin lisäksi myös normaaliin tapaan etenevät luennot. Seppälän ratkaisuissa opiskelijat tarvitsevat omaan koneeseensa WWW-selaimen, johon on asennettu laajennukseksi (plug-in) Techexplorer ja apusovellukseksi (helper) numeerisen ja symbolisen laskennan ohjelma Maple V.

Osa Seppälän materiaalista on HTML-muotoista. HTML-osioihin on linkitetty TeX-tiedostoja, joita luetaan Techexplorerin avulla. TeX-tiedostoissa on sopivissa paikoissa laskutoimituksissa linkkejä Maplen työarkkeihin (worksheet), jolloin kyseistä linkkiä klikkaamalla Maple käynnistyy omaan ikkunaansa. Samalla Mapleen avautuu kyseistä asiaa käsittelevä työarkki, jossa on mahdollista sekä ajaa valmiiksi kirjoitettuja laskutoimituksia että suorittaa muitakin laskutoimituksia. Maplen työarkkeja on mahdollista linkittää samaan tapaan myös suoraan HTML-dokumenttiin. Maplen ei välttämättä tarvitse olla asennettuna selaimen apusovellukseksi, sillä Maplen uusimmissa versioissa ohjelmalla itselläänkin voidaan avata jossain määrätyssä WWW-osoitteessa oleva työarkki.

Kuvatun kaltaisen vuorovaikutteisen opiskelumateriaalin lisäksi BlackBoardilla on mahdollista pitää esimerkiksi monivalintatyyppisiä pikku kokeita, joiden tulokset järjestelmä kirjaa suoraan. Lisäksi opiskelijoilta voidaan kerätä palautetta systeemin avulla, ja kurssin yhteyteen voidaan perustaa esimerkiksi sähköpostin kautta käytettävä ilmoitussivu tai keskusteluryhmä.

Reino Virrankoski
reino.virrankoski@hut.fi ⋄ http://www.math.hut.fi/~virranko

Artikkelin kirjoittaja työskentelee tutkimusapulaisena ja tuntiassistenttina Teknillisessä korkeakoulussa. Hän on juuri saanut pro gradunsa *Laskentaohjelmien hyödyntäminen digitaalisen opetusmateriaalin laatimisessa – optimointia Maplella* valmiiksi Simo K. Kivelän johtamassa MatTa-projektissa.

Aiheeseen liittyviä linkkejä

Henkilöitä ja materiaalia

Apulaisprofessori R. van Engelenin (Florida State University) kotisivu. Sivun takaa löytyy esimerkkejä van Engelenin laatimasta opiskelumateriaalista: http://www.cs.fsu.edu/~engelen/

Emeritusprofessori Donald E. Knuthin (Stanford University) kotisivu: http://www-cs-faculty.stanford.edu/~knuth/

Professori Mika Seppälän (Florida State University) kotisivu: http://www.math.fsu.edu/~seppala/

Andreas Strotmannin kotisivu. Tällä hetkellä Strotmann tekee jatko-opintojaan Florida State Universityssa: http://www.rrz.uni-koeln.de/~a0047/english.html

Ohjelmistoja

IBM:n sivu, jolta Techexplorer on saatavissa: http://www.software.ibm.com/techexplorer/

Adoben sivut, joilta PDF-tiedostojen lukemiseen tarvittava Adobe Acrobat Reader on saatavissa: http://www.adobe.com

WebCT:n kotisivu: http://www.webct.com

Lotuksen kotisivut, Lotus LearningSpace: http://www.lotus.com/home.nsf/welcome/learnspace

BlackBoardin kotisivut: http://www.blackboard.com

Organisaatioita

The OpenMath Society: http://www.nag.co.uk/projects/openmath/omsoc/

W3C's Math Home Page: http://www.w3.org/Math/

Northeast Parallel Architectures Center, koko joukko erilaisia tietotekniikan ja Web-teknologian sovellusprojekteja: http://www.npac.syr.edu/Projects/index.html

Suomenkielistä materiaalia

MatTa-projektin (matematiikkaa tietokoneavusteisesti) kotisivu. Lehtori Simo K. Kivelän johtama projekti Teknillisen korkeakoulun matematiikan laitoksella: http://www.math.hut.fi/matta/

Apiola, H., *Symbolista ja numeerista laskentaa Maple-ohjelmalla* -teoksen verkkoliite: http://www.math.hut.fi/~apiola/maple/opas/index.html.fi

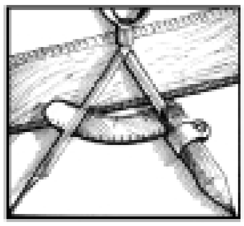
Astola, L., *Mathematica-alkeisopas*: http://www.math.hut.fi/matta/Iso_M/Oppaat/Math/html/paasivu.html

Virrankoski, R., *Maple V -perusteita*: http://www.math.hut.fi/matta/Iso_M/Oppaat/Maple/html/kansi.htm

Postituslistoja

IBM Techexplorer -postituslista: techexplorer@listserv.nodak.edu
(listalle ilmoittautuminen techexplorer-reques@listserv.nodak.edu).

Matematiikan julkaisemista verkossa sekä erilaisia WWW-sovelluksia käsittelevä WebMath-postituslista: webmath@camel.math.ca



Geometriakulma: Mikä on käyrä?

Käyräksi tuntuisi luontevalta kutsua viivaa, jonka voi piirtää kynää paperista nostamatta. Matemaattiseksi määritelmäksi tämä ei kumminkaan kelpaa: kynä, paperi, piirtäminen ovat konkreettisia asioita, joita ainakaan moderni matematiikka ei määrittelyn pohjaksi kelpuuta.

Kun tasoon sijoitetaan suorakulmainen xy-koordinaatisto, voisi käyrän määritelmäksi ottaa yhtälön: ne pisteet (x, y), jotka toteuttavat muotoa $F(x, y) = 0$ olevan yhtälön, muodostavat käyrän. Kahden muuttujan funktiolta F täytyy tällöin olettaa jonkinlaista säännöllisyyttä, ainakin jatkuvuutta. Esimerkiksi yhtälö $x^2 + y^2 - 1 = 0$ esittää origokeskistä ympyrää, jonka säde on $= 1$.

Tälläkin määritelmällä on heikkoutensa. Vaikka F olisikin säännöllinen funktio, ei yhtälö välttämättä esitä yhtään mitään: $x^2 + y^2 + 1 = 0$. Tai se voi esittää yhtä pistettä: $x^2 + y^2 = 0$. Lukija tutkikoon, mitä esittää yhtälö

$$25 + 10x + 2x^2 + 2x^3 + x^4 - 10y - 20xy - 2x^2y + 2y^2 + 2xy^2 + 2x^2y^2 - 2y^3 + y^4 = 0.$$

Vihjeeksi: Yhtälön vasemman puolen voi jakaa tekijöihin. Jos ei käsinlasku onnistu, niin matemaattisten ohjelmien – Mathematica, Maple, Derive, jne. – tekijöihinjako- eli faktorointialgoritmit ovat varsin hyviä.

Määritelmässä on toinenkin heikkous: Se antaa pikemminkin pistejoukon kuin käyrän. Kahdesta käyrän pisteestä tulisi nimittäin voida sanoa, kumpi on käyrää pitkin kuljettaessa lähempänä alkupistettä. Määrittely yhtälön avulla ei aseta pisteitä järjestykseen.

Parempi käyrän määritelmä saadaan vetoamalla ns. *parametriesitykseen*: Olkoon annettuna funktiot $x(t)$ ja $y(t)$. Kun parametri t käy läpi jonkin reaaliakselin välin $[a, b]$ (joka voi myös olla koko akseli, puoliääretön väli $[a, \infty[$, tms.), niin vastaavat funktioiden arvot antavat käyrän pisteet $(x(t), y(t))$. Pisteille saadaan tällöin järjestys: Jos $t_1 < t_2$, niin piste $(x(t_1), y(t_1))$ on ennen pistettä $(x(t_2), y(t_2))$.

Funktioilta $x(t)$ ja $y(t)$ täytyy edellyttää riittävästi säännöllisyyttä, vähintäänkin jatkuvuus. Tämä vastaa intuitiivista käsitystä, että käyrä täytyy voida piirtää kynää paperista nostamatta. Jatkuvuus ei kuitenkaan riitä. Esimerkiksi jos funktiot ovat vakiofunktioita, niin saadaan vain yksi piste, eikä tällaista ehkä ole luontevaa kutsua käyräksi. Itse asiassa matemaattisessa kirjallisuudessa käyrän määritelmät jossain määrin vaihtelevat: Funktioilta edellytetään ainakin jatkuvuus, mutta usein enemmän.

Esimerkiksi:

1. Jos $x(t) = R\cos t$, $y(t) = R\sin t$, $0 \le t \le 2\pi$, niin käyrä on origokeskinen R-säteinen ympyrä.

2. Funktiot $x(t) = a\cos t$, $y(t) = b\sin t$, $0 \le t \le 2\pi$, määrittävät tavallisen ellipsin $x^2/a^2 + y^2/b^2 = 1$.

3. Käyrä $x(t) = a\cos^3 t$, $y(t) = b\sin^3 t$, $0 \le t \le 2\pi$, on nimeltään *asteroidi*.

4. $x(t) = a(t - \sin t)$, $y(t) = a(1 - \cos t)$, $t \in \mathbb{R}$, on *sykloidi*.

5. Suora on käyrän erikoistapaus; vaikkapa $x(t) = 1 + 2t$, $y(t) = 3 + 4t$, $t \in \mathbb{R}$.

Käyrän muotoa $F(x, y) = 0$ olevaan yhtälöön päästään eliminoimalla parametri t yhtälöistä $x = x(t)$, $y = y(t)$ algebrallisella manipuloinnilla. Aina tämä ei tietenkään onnistu. Lukija johtakoon yhtälön edellä esitetyissä asteroidin ja suoran tapauksissa ja pohtikoon, miten käy sykloidin tapauksessa. Lukija myös piirtäköön ainakin asteroidin ja sykloidin.

Määritelmä parametriesityksenä samastaa käyrän ja funktiot $x(t)$, $y(t)$. Toisaalta sama pistejoukko voidaan saada erilaisilla funktioiden valinnoilla. Edellä mainitun suoran pisteet saadaan myös parametriesityksestä $x(t) = 1 + 2t^3$, $y(t) = 3 + 4t^3$ tai jopa $x(t) = 1 + 2(t^3 - t)$, $y(t) = 3 + 4(t^3 - t)$. Mikä oleellinen ero on viimeksi mainitun ja kahden muun välillä?

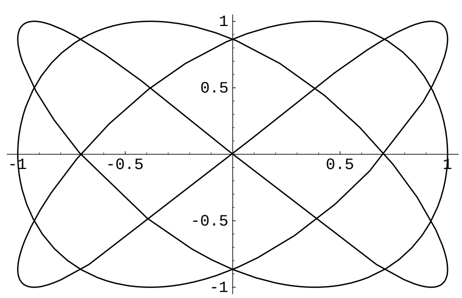
Parametriesityksen käyttö antaa helpon tavan yleistää käyrän määritelmä kolmiulotteiseen avaruuteen: *Avaruuskäyrä* määräytyy parametriesityksestä $x(t)$, $y(t)$, $z(t)$, missä $t \in [a, b]$. Yksinkertainen esimerkki on

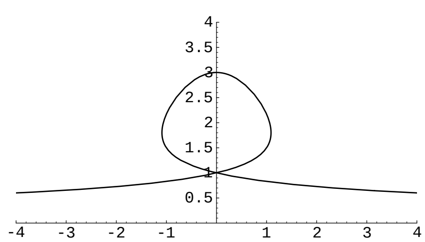
$$x(t) = a\cos t, \quad y(t) = a\sin t, \quad z(t) = \frac{t}{2\pi}, \quad t \in \mathbb{R}.$$

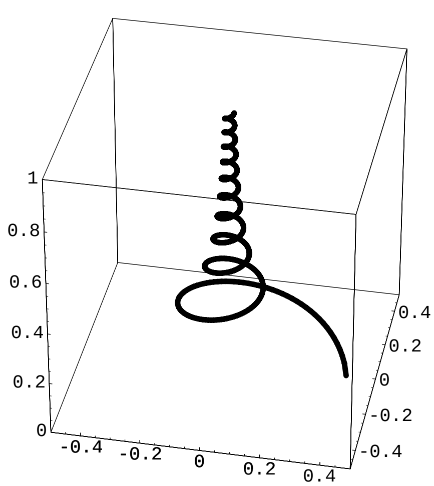
Ympyrän parametriesityksen perusteella on pääteltävissä, että parametrin t kasvaessa tämän käyrän pisteet kiertävät xy-tason origokeskisen a-säteisen ympyrän ala- tai yläpuolella ja ovat sitä korkeammalla, mitä suurempi arvo parametrilla t on. Käyrän nimi onkin kuvaava: *ruuviviiva*.

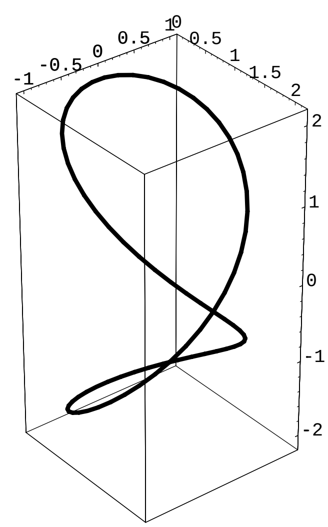
Esimerkkejä erilaisista käyristä lukija löytää osoitteesta
http://www-groups.dcs.st-and.ac.uk:80/~history/Curves/Curves.html .

Lopuksi käänteinen harjoitustehtävä: Seuraavat kuvat esittävät kahta tasokäyrää ja kahta avaruuskäyrää. Tehtävänä on kehitellä niille mahdollisimman hyvät parametriesitykset (jotka eivät suinkaan ole yksikäsitteisiä).









Vihjeeksi seuraavaa: 1) Käyrässä on jaksollisuutta, joten parametriesityksessä kannattaa käyttää sinejä ja kosineja. 2) Polynomeilla päästään pitkälle: osoittajaan tai nimittäjään. 3) Kyseessä on eräänlainen ruuviviiva, on vain löydettävä sopivasti muuttuva säde ja sopivasti muuttuva nousu. 4) Käyrä syntyy pallon ja sopivan lieriön leikkauksena.

Simo K. Kivelä

Tietokoneavusteisen opiskelupaketin julkistus:

$\mathfrak{M}$ niinkuin matematiikka

Teknillisen korkeakoulun matematiikan laitoksella käynnissä olevan MatTa-projektin (**Mat**ematiikkaa **T**ietokone**a**vusteisesti) tähänastinen päätuotos, tietokoneavusteinen lukiomatematiikan kertauspaketti $\mathfrak{M}$ *niinkuin matematiikka* tai lyhyemmin *Iso-M* on valmistunut. Kyseessä on ensimmäinen versio kokeiluluontoisesta verkko-opetusmateriaalista.

Projektia ovat rahoittaneet Suomen Akatemia, opetusministeriö ja Teknillinen korkeakoulu.

Iso-M -paketti julkistetaan täten ja asetetaan suomalaisten koulujen ja eri asteisten oppilaitosten vapaaseen opetuskäyttöön. Myös yksityiseen käyttöön materiaalia voidaan vapaasti kopioida. Sisällön muuttaminen sekä muunlainen, erityisesti kaupallinen käyttö on ilman eri sopimusta kielletty.

Paketti on käytettävissä verkossa WWW-selaimen kautta. Se voidaan myös asentaa oppilaitoksen tai yksityisen käyttäjän Unix- tai Windows-koneeseen.

Paketin yhteydessä suositellaan käytettäväksi jotakin laskentaohjelmaa. Tällä hetkellä tuettuja ovat Mathematica ja Maple. Koska nämä ovat kaupallisia ohjelmia, ne eivät sisälly pakettiin, vaan käyttäjä joutuu hankkimaan ohjelman itse.

Lisätietoja saa seuraavista WWW-osoitteista:

Paketin käyttö ja ohjeet omalle koneelle kopioimisesta http://www.math.hut.fi/matta/material.html .

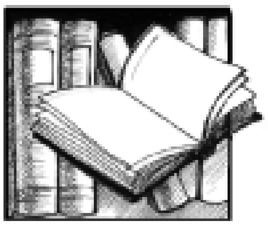
Verkkoversion käynnistys http://www.math.hut.fi/matta/Iso_M/Kansi.htm .

MatTa-projektin WWW-sivut http://www.math.hut.fi/matta/ .

Koska kyseessä on kokeiluluontoinen materiaali, käyttäjiä pyydetään raportoimaan käyttöön otosta, käytön luonteesta, esiintyneistä ongelmista, paketin hyödyllisyydestä, jne. alla olevaan osoitteeseen, mieluiten sähköpostilla.

Tarkemmat tiedot:

Simo K. Kivelä
Matematiikan laitos
Teknillinen korkeakoulu
Puh. (09) 451 3032
S-posti Simo.Kivela@hut.fi



Kirja-arvostelu

Heikki Ruskeepää: Mathematica-opas, Versio 4, Otatieto 1999, 133 s.

Otatieto jatkaa matematiikkaohjelmistojen suomenkielisten opaskirjojen julkaisemista. Viime vuonna ilmestyneen H. Apiolan oppaan rinnalle on nyt saatu myös johdatus Maplen pahimman kilpailijan Mathematican uuden version käyttöön. Sen on kirjoittanut Heikki Ruskeepää Turun yliopiston sovelletun matematiikan laitokselta.

Matemaattisista ohjelmistoista löytyy nykyisin oppaita myös verkosta, viittaan vain edellä olevan R. Virrankosken kirjoituksen linkkeihin. Verkko-oppaat ja ohjelmien Help-valikot ovat tietysti omiaan, kun täytyy saada ensiapua koneen äärellä työskennellessä, mutta itse ainakin tarvitsen asioiden parempaan hahmottamiseen kunnollisen kirjan.

Ruskeepään opas jakaantuu neljään jaksoon, joista ensimmäinen käsittelee luonnollisesti perusteita. Jokaisen, joka joskus aikoo käyttää Mathematicaa, soisi lukevan edes nämä ensimmäiset 19 sivua. Liian useinhan aloittelija panee kaikki vastaan tulevat ongelmat tietokoneen tai ohjelman syyksi, ja pelkäänpä syyllistyneeni tähän itsekin. Syynä lienee luonnollinen into aloittaa esimerkiksi tämän oppaan lukeminen vasta seuraavasta jaksosta, jossa perehdytään Mathematican monipuolisiin graafisiin ominaisuuksiin.

Kirjan kolmannessa ja varsinkin neljännessä jaksossa matematiikka on sitten pääosassa. Aluksi tarkastellaan lausekkeiden, funktioiden ja erilaisten tietorakenteiden kuten listojen ja taulukoiden käsittelyä. Intoa puhkuva matemaatikko saattaa jälleen helposti sivuuttaa tämän jakson, kun viimeisessä ja sivumäärältään laajimmassa osassa odottavat mm. derivaatat, integraalit, matriisilaskenta ja differentiaaliyhtälöt. Näin ei kuitenkaan kannata tehdä, sillä ainakin itse olen huomannut, että varsinaiset matemaattiset käskyt ovat suhteellisen helppokäyttöisiä, kun taas suurimmat ongelmat piilevät tulosten tehokkaan jatkokäsittelyn kohdalla.

Oppaan liitteissä käsitellään myös Mathematican ja muiden ohjelmien yhteispeliä sekä kirjoitelmien laatimista. Itse asiassa koko Ruskeepään opas on kirjoitettu Mathematica-dokumenttina, mikä parhaiten todistaa menetelmän käyttökelpoisuudesta.

Ruskeepään kirjoitus on sujuvaa ja sivumäärän rajoittamana melko lyhytsanaista. Kirjan koon huomioon ottaen olisi perusteetonta väittää, että jotakin olennaista jää sanomatta. Mikäli haluaa tehdä vertailuja käsillä olevan Mathematica- ja Apiolan Mapleoppaan välillä, voi todeta, että Ruskeepään oppaassa käsitellään useampia matemaattisia aihepiirejä ja se sisältää varmasti enemmän erilaisia käskyjä, mutta Apiolan kirjasta löytyy puolestaan eräitä laajempia sovellusesimerkkejä. En osaa tehdä eroa näiden valintojen paremmuuden välillä, mutta jos mukaan halutaan molemmat osat, on tuloksena Wolframin tai Heckin kirjojen kaltaisia tiiliskiviä. Sen sijaan vertailu Maplen tai Mathematican paremmuuden välillä on mielestäni yhtä turhaa, ja samalla mielenkiintoista, kuin väittely jumalan olemassaolosta tai optioiden oikeutuksesta: ne ovat kaikki maailmankatsomuksellisia asioita. Ainoa varma seikka on se, ettei kannata yrittää opiskella molempia yhtäaikaa.

Kaikenkaikkiaan Ruskeepään opas on erinomainen lisä suomenkieliseen ohjelmistokirjallisuuteen. Kuten jo pääkirjoituksessa tuli todettua, tällaisia kirjoja lukee paljon mieluummin kuin graafisten laskinten käyttöohjeita.

Pekka Alestalo

Kirjallisuutta:

H. Apiola: *Numeerista ja symbolista matematiikkaa Maple-ohjelmalla*, Otatieto 1998, 214 s.

A. Heck: *Introduction to Maple*, 2nd ed., Springer-Verlag, 1997, 699 s.

S. Wolfram: *The Mathematica Book*, 4th ed., Wolfram Media/Cambridge University Press, 1999, 1470 s.

Kannen kuva

Tämän numeron kansikuva esittää *Sierpinskyn pyramidia*. Lisätietoja kuvasta ja sen piirtämisestä löytyy osoitteesta

`www.math.hut.fi/~apiola/solmu/`

jossa on kuvan piirtämisessä käytetty Maple-työarkki `sierpinsky.mws` ja sen HTML-versio `sierpinsky.html` .

Unkarilaisia matematiikan koulutehtäviä Solmuun

Solmun yhteyteen liitetään vuoden sisällä kokoelma unkarinkielestä käännettyjä matematiikan koulutehtäviä ratkaisuineen. Käännöstyötä vie eteenpäin fil. tri Taneli Huuskonen, matemaatikko, joka on opiskellut asian harrastuksesta unkarin kieltä Helsingin yliopistossa. Osa käännöstyöstä tehdään AKO:n rahoittamalla käännöskurssilla, jolla on kymmenkunta suomalaista unkarinopiskelijaa ja kaksi unkarilaista suomenopiskelijaa.

Yhteistyön taustalla on huoli jo vuosia sitten havaitusta riittämättömästä matematiikan osaamisen tasosta, joka vaikeuttaa jo teollisuuden työvoimansaantiakin. Valitettavan yleinen väärinkäsitys Suomessa on, ettei matematiikkaa enää tarvita tietokoneiden yleistymisen takia; on jäänyt huomaamatta, ettei matematiikka ole koneella korvattavaa mekaanista laskemista. Matematiikka on moderni, nopeasti kehittyvä tiede, korkean teknologian perusta ja tarpeellinen yhä useammilla, myös ns. "pehmeillä" aloilla. Tilanteen korjaamiseksi on mietitty, mistä suunnasta kannattaisi ottaa oppia. Unkari on pieni maa, jolla on kansainvälisissä kouluvertailuissa poikkeuksellisen hyvä menestys, lukuisia ensi luokan matemaatikkoja ja yli kymmenen Nobelin palkintoa aloilla, jotka vaativat hyvää matemaattista pohjaa. Kielisukulaisuuden takia yhteistyövalmius on erityisen sydämellistä.

Unkarin matematiikan menestyksen takana on pitkät perinteet, työ ja matemaattis-luonnontieteellisten alojen arvostus. Sata vuotta sitten aloitettiin matematiikkalehti Kömal ja jokavuotiset matematiikkakilpailut, jotka mahdollistivat matemaattisten kykyjen löytämisen ja kehittämisen tasapuolisesti koko maassa. Opetusmenetelmänä käytetään paljon matemaattisten tehtävien ratkaisua. Menetelmä vaatii paljon opettajalta, koska tehtävät eivät ole erillisiä "knoppeja", vaan ongelmia, joiden pohjalta taitava opettaja pystyy ohjaamaan matematiikan rakenteen omaksumista. Pystyäkseen tähän opettaja tarvitsee vahvan pohjakoulutuksen.

Opetushallitus ja Wihurin rahasto tukevat käännöshanketta. Lopputuloksen toivotaan innostavan koululaisiamme matematiikan opiskeluun. Tämä olisi erityisesti tytöille tarpeen, jotta he eivät löisi laimin koulumatematiikan opiskelua ja näin ymmärtämättä asian vakavuutta karsisi myöhempiä opiskelumahdollisuuksiaan ja uravalintojaan.

Marjatta Näätänen
dos.
Matematiikan laitos
Helsingin yliopisto