Symbolinen laskenta (MAT180,1ov)

Kurssin tavoite ja sisältö

Symbolisen laskennan kurssilla opitaan tietokoneen käyttämistä apuvälineenä matemaattisessa ongelmanratkaisussa. Kurssin tavoitteena on antaa perusvalmiudet symboliseen laskentaan erikoistuneen *Mathematica*-ohjelmiston käyttöön. Kurssin suorittaneen tulisi pystyä käyttämään *Mathematicaa* (tai mikseі jotain muutakin vastaavaa ohjelmistoa) apuvälineenä vaikkapa opintoihin liittyvien harjoitustehtävien ratkaisemisen apuvälineenä.

Kurssin rakenne ja suoritustapa

Kurssi sisältää 6 (3x2) tuntia luentoja sekä 16 (8x2) tuntia ohjattuja mikroharjoituksia. Opetustapahtumiin osallistuminen on vapaaehtoista. Kurssi suoritetaan mikroluokassa tekemällä valvottu *näyttökoe*. Vaikkakaan harjoituksiin osallistuminen ei ole pakollista, on se ehdottoman suositeltavaa kokeen onnistumisen kannalta. Kurssi on poikkeustapauksissa mahdollista suorittaa myös *itsenäisellä harjoitustyöllä*. Tämä vaihtoehto on näyttökoetta työläämpi, varsinkin harjoituksiin osallistuneille.

Pääteharjoitusten tehtäviä ei jaeta etukäteen vaan ne saa harjoituskerran yhteydessä. Kurssin periaatteena on itse oppia kokeilemalla erilaisia tehtäviä harjoitustuntien aikana. Kurssista saa suoritusmerkinnän opintorekisteriin kun näyttökoe on suoritettu tai harjoitustyö on palautettu ja hyväksytty. Kurssista ei anneta arvosanaa.

Kurssin ohjelma on pääosin seuraava:

1. luento	Käytännön järjestelyt, johdantoa symboliseen laskentaan ja Mathematican käyttöön
1. harjoitus	Ohjelman käyttöön tutustuminen, aritmetiikkaa ja sieventämistä
2. harjoitus	Muuttujasymbolien käyttö, listat ja matriisit
2. luento	Tutustumista Mathematican Front Endiin
3. harjoitus	Lausekkeiden määrittely, derivointi ja integrointi
4. harjoitus	Funktioiden määrittely, summat, sarjat
3. luento	Kokeisiin ilmoittautimen, Mathematica-esimerkkejä
5. harjoitus	Yhtälöt ja yhtälöryhmät
6. harjoitus	Optimointitehtäviä
7. harjoitus	Grafiikkaa
8. harjoitus	Ohjelmointi Mathematicalla

Matemaattisista ohjelmistoista

Erilaisten matemaattisten ohjelmistojen kehittäminen alkoi jo 1970-luvulla ja on luonnollista, että kehitys on seurannut tietotekniikan kehitystä. Prosessoritehon kasvun ja laitteistojen nopeutumisen myötä on myös ohjelmistoihin voitu lisätä ominaisuuksia. Laskennallisen tehon lisäksi viime vuosina on voimakkaasti lisätty ohjelmistojen käyttömukavuutta ja graafisia ominaisuuksia. Matemaattisten tulosten mallintaminen ja visualisointi onkin muuttunut vuosi vuodelta helpommaksi.

Laskentatehon ollessa kovin rajallista ensimmäiset matemaattiset ohjelmistot suunniteltiin *numeeriseen* laskentaan. Näiden käyttäminen perustui vahvasti ohjelmointitaitoihin ja tämän vuoksi käyttö olikin hankalaa ja vaati asiantuntemusta. Tältä aikakaudelta on edelleen käytössä mittavia aliohjelmakirjastoja (esim. NAG, IMSL) eikä kehitys ole hidastunut vieläkään. Numeerisia välineitä hyödynnetään laajalti erilaisissa tekniikan ja luonnontieteiden ongelmissa. Puhtaiden ohjelmakirjastojen jatkeeksi on sittemmin kehitetty erilaisia käyttöliittymiä ja näin ohjelmistot ovat muuttuneet käytöltään helpommiksi ja tehokkaammiksi. Hyvänä esimerkkinä tällaisesta ohjelmistosta mainittakoon *Matlab*.

Toinen matemaattisten ohjelmistojen ryhmä on *symboliseen* laskentaan erikoistuneet ohjelmistot. Näiden kehittäminen on päässyt vauhtiin numeerisia työkaluja myöhemmin, koskapa symbolinen matematiikka vaatii usein mutkikkaita algoritmeja. Tietokoneet osaavat toki laskea numeroilla, mutta muuttujia sisältävien kaavojen käsittely ilman, että muuttujille annetaan arvoja, on huomattavasti vaikeampaa. Symboliseen laskentaan erikoistuneita ohjelmia on myös kehitetty paljon. Näistä vertaistaan hakeva esimerkki on *Mathematica*, jota käytetään laajalti matemaatikkojen keskuudessa ympäri maailmaa. Niin myös tällä kurssilla.

Nykyisin kahtiajako symbolisiin ja numerisiin ohjelmistoihin on alkanut menettää merkitystään. Symbolisista ohjelmistoista alkaa löytyä numeeriseen laskentaan tarvittavia ominaisuuksia ja jonkin verran myös päinvastoin. Peruskäyttäjä, joka ei tarvitse suurta laskennallista tehokkuutta ja huippuunsa viritettyjä algoritmeja, pystyy yleensä suoriutumaan laskutehtävistään käyttäen mitä tahansa nykyaikaista laskentaohjelmistoa.

Mathematicasta

Mathematica on matemaattinen ohjelmisto, joka soveltuu erityisesti symboliseen laskentaan mutta sisältää myös runsaasti numeerisia ominaisuuksia. Ohjelmiston laskentaydin sisältää myös pystyvän komentotulkin, mikä mahdollistaa ohjelman rajattoman laajennettavuuden. Perusaineksista löytyy lääkkeet keskivertokäyttäjän tarpeisiin, samalla kun vaativa ammattikäyttäjä voi ohjelmoida omiin tarpeisiinsa omia algoritmeja. Juuri tämä on *Mathematican* vahvuus ja juuri tämän vuoksi se on laajalti arvostettu käyttäjiensä keskuudessa.

Mathematica-ohjelmisto jakautuu ohjelmallisesti kahteen osaan:

- **Kernel** eli laskentaydin
- **Front End** eli käyttöliittymä

Laskentaydin on ohjelma (tai ohjelmakirjasto), jota voi yleensä käyttää myös suoraan käyttäen suoria komentoja. Pelkkä laskeminen ei sinällään vaadi mitään muuta kuin laskentaytimen, mutta käytännön työskentelyssä on usein helpompaa jos annetut syötteet ja laskentatoimenpiteiden tulokset ovat valmiiksi tulostettavassa muodossa. Tätä varten on suunniteltu käyttöliittymiä, joista suosituin on luonnollisesti ohjelmiston oma Front End. Ohjelmistoa käytettäessä on kuitenkin tärkeää pitää mielessä, että laskentaa EI suorita se ohjelma, joka ottaa käyttäjän näppäilyt vastaan. Käyttöliittymän tarkoituksena on tulkata ja välittää käyttäjän pyynnöt varsinaisen laskennan hoitavalle ohjelmalle, kernelille.

Erillisen laskentaytimen merkittävin etu on siinä, että laskentaydin voi sijaita vaikkapa tehokkaalla supertietokoneella. Laskennallisesti vaativissa tehtävissä voisi olla kannattavampaa lähettää työtehtävä verkkoa pitkin muualle laskettavaksi. Kun tallaisesta verkkoliikenteestä huolehtii käyttöliittymä, ei operaatio vaikuta käyttäjän toimiin mitenkään. Tällainen etäkernelin käyttö on kuitenkin menettänyt merkitystään sitä mukaa kuin yksittäisten työasemakoneiden tehot ovat kasvaneet. Kuitenkin tehokkaimmallekin laskukoneelle löytyy aina käyttäjä, joka keksii riittävän vaativan tehtävän, jota ei pystytä ratkaisemaan käytettävissä olevassa ajassa. Erillistä kerneliä voidaan toki ajaa vaikka 10000 konetta sisältävässä laskentaklusterissa. Kaikki on viime kädessä kiinni ohjelmiston soveltajista.

Symbolinen ja numeerinen laskenta

Tutuin esimerkki numeerisen laskennan apuvälineestä on tavallinen taskulaskin, joka kykenee käsittelemään pelkkää numeroinformaatiota. Sisäisestä laskentatarkkuudesta riippumatta parhaimmankin laskimen antamat lukuarvot ovat kuitenkin aina likiarvoja todellisista luvuista. Symbolinen laskenta tarkoittaa sitä, ettei yhtäkään lukuarvoa pyöristetä vaan kaikki laskenta pyritään esittämään suljetussa muodossa käyttäen matemaattisia symboleja. Tällöin laskennassa muuttujien arvoilla ei ole minkäänlaista merkitystä.

Esimerkki:

Numeerisesti: $\sin\frac{\pi}{4} = 0.707106781$ Symbolisesti: $\sin\frac{\pi}{4} = \frac{1}{\sqrt{2}}$

Numeerisesti: $\int\limits_0^1 e^x dx = 1.718281828$ Symbolisesti: $\int\limits_0^1 e^x dx = e - 1$

Jotkut toimenpiteet ovat mahdollisia ainoastaan symbolisessa laskennassa. Vaikkapa edellisessä esimerkissä puhtaasti numeerinen ohjelmisto ei tiedä, että saatu luku 1.718281828 tarkoittaa nimenomaan tiettyä luonnovakiota vähennettynä yhdellä. Vastaavasti ainoastaan symbolinen ohjelmisto osaa tehdä sievennyksen $x + 3a - 2x - a = 2a - x$.

Symbolisen laskennan ohjelmistot pystyvät käsittelemään algebrallisia lausekkeita ja vaikkapa laskemaan integraalifunktioita tai derivaattoja. Mathematica onkin tehokas työkalu lausekkeitten sieventämisessä ja rutiinilaskujen laskemisessa/tarkastamisessa.

Usein tulee vastaan tehtäviä, joiden symbolinen ratkaiseminen on mahdotonta. Esimerkiksi monien yhtälöiden ratkaisuja ei voida esittää symbolisesti, joten numeerisia menetelmiä on pakko käyttää. Monet integraalit voivat myös olla vaikeita laskettavia symbolisesti, kun taas numeerinen menetelmä antaisi vastauksen huomattavasti nopeammin ja suoraviivaisemmin. Tämän vuoksi käytännön sovellukset (matemaattinen mallintaminen, simulointi, ...) käyttävätkin yleensä numeerista laskentaa, koskapa symbolisilla arvoilla ei ole sovelluksen kannalta merkitystä. Symbolisella laskennalla onkin suurin merkitys matemaattisten teorioiden kehittämisessä. Kun teoria on saatu valmiiksi, voidaan siirtyä käyttämään numeerisia menetelmiä. Käytännössä kuitenkin sekä symbolisia että numeerisia ohjelmistoja tarvitaan.

Numeerisen laskennan sudenkuopat piilevät pyöristysvirheissä. Liian aikaisessa vaiheessa tehty pyöristys saattaa viedä laskennan lopputulokset pahasti metsään. Katsotaan esimerkiksi lauseketta

$$uvwxy - (uvw)^{1/3}(vwx)^{1/3}(wxy)^{1/3}(uxy)^{1/3}(uvy)^{1/3}$$

Suoraan lauseketta katsomalla voidaan nähdä, että lauseke on aina identtisesti nolla riippumatta muuttujien arvoista. Entäpä jos lausekkeen arvo lasketaan numeerisesti arvoille, jotka ovat aidosti likiarvoja? Kokeilla voi vaikkapa peräkkäisillä alkuluvuilla 7919, 7927, 7933, 7937 ja 7949...

Mathematican käyttö

Kun Mathematica (itse asiassa sen Front End) käynnistetään, ohjelma avaa näytölle Notebook-ikkunan, johon käyttäjä kirjoittaa haluamiaan komentoja ja johon komentoja vastaavat tulosteet tulostetaan. Tässä työikkunassa olevaa teksti- ja kuvainformaatiota on mahdollista muotoilla monilla eri tavoilla. Huomattavaa kuitenkin on, ettei käyttöliittymäasioilla ole mitään tekemistä varsinaisen laskennan kannalta. Näytöllä näkyvä Front End on pelkkä välittäjä käyttäjän ja laskentaytimen välillä.

Ohjelmalle kirjoitetaan syöterivejä, jotka päätetään painamalla näppäinyhdistelmää `[SHIFT]+[ENTER]`. Tämä näppäinyhdistelmä kertoo käyttöliittymälle, että käyttäjän antama syöterivi on valmis laskettavaksi. Tämän jälkeen laskutoimitus viedään laskettavien töiden jonoon ja käyttöliittymä vapautetaan uusia syötteitä varten. Huomattakoon, että laskettavat työt todella jonottuvat. Kesken olevat laskennat tukkivat jonon eikä uusia suuria töitä yleensä kannata syöttää ennen jonon vapautumista.

Samalla syöterivillä voidaan suorittaa useampia komentoja. Jos komennon tulostusta ei haluta näytölle, voidaan komento päättää puolipisteeseen.

Aiemmin laskettuihin tuloksiin voidaan viitata tuloksen järjestysnumerolla. Jokainen syöte ja tulos numeroidaan ja yleensä Front End näyttää järjestysnumerot syötteiden kohdalla merkinnällä `In[n]:=` ja tuloksen kohdalla merkinnällä `Out[n]:=`, missä n on mainittu tuloksen järjestysnumero. Viittaaminen riviin n tapahtuu syöttämällä laskentaytimelle merkintä `%n`. Huomattakoon, että laskentaydin muistaa kaikki syötteensä ja tuloksensa niin (ja vain niin) pitkään, kuin se pidetään muistissa. Tallennuksen jälkeen seuraavalla ajokerralla rivien numeroinnit saattavat mennä eri tavoin, joten numerointia kannattaa tallennettavissa töissä käyttää todella harkiten. Viisaampaa onkin tallentaa tulokset, joihin aiotaan viitata myöhemmin, joidenkin kuvaavien symbolien arvoiksi. Esimerkiksi laskemansa integraalifunktiot voi tallentaa vaikkapa muuttujiin `integraali1,integraali2,integraali3,...`

Riviviittausta voi käyttää myös relatiivisesti. Pelkkä '`%`' viittaa edelliseen tulokseen, '`%%`' taas sitä edelliseen jne. Tätä viittaustapaa voi käyttää myös tallennettavissa töissä, mutta edelleen viisainta on kuvaavasti nimettyjen symbolien käyttö.

"Kielellisiä" asioita

Aritmeettiset peruslaskutoimitukset

+	Yhteenlasku	
-	Vähennyslasku	
*	Kertolasku	(Kertomerkin voi usein korvata välilyöntiä)
/	Jakolasku	
^	Potenssiin korotus	

Sulkujen käyttö

Mathematicassa käytetään kolmea eri sulkutyyppiä. Näistä jokaisella on oma erityinen käyttötarkoituksensa, eikä niiden käyttöä saa sekoittaa keskenään.

- Kaarisulut `( )` toimivat laskujärjestystä ohjaavina sulkuina aivan kuten on totuttu lausekkeita kirjoitettaessa.
- Hakasulut `[ ]` toimivat komennoissa ja funktioissa. Esimerkiksi $\sin x$ lasketaan komennolla `Sin[x]`.
- Aaltosuluilla `{ }` rakennetaan listoja, vektoreita ja matriiseja.

Funktioiden ja symbolien muodosta

Mathematicassa kaikkien sisäisten funktioiden ja sovittujen symbolien nimet alkavat isolla kirjaimella. Lisäksi nimet on valittu mahdollisimman hyvin toimintoja kuvaaviksi. Nimet saattavat olla hankalia muistaa, joten Mathematican help-järjestelmää kannattaa opetella käyttämään jo hyvissä ajoin.

Omien symbolien määrittely onnistuu sijoitusoperaattorilla (=). Huomattavaa on, että kaikki arvot säilyvät laskentaytimen muistissa kunnes arvoja muutetaan tai määrittelyt poistetaan. Omien symbolien nimissä saa käyttää kaikkia aakkosnumeerisia merkkejä, mutta nimi ei saa alkaa numerolla. Symbolien arvoiksi voi asettaa numeroita, lausekkeita, matriiseja, yhtälöitä, kuvia, ...