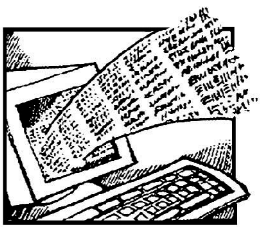


Tarvitseeko informaatioteknologia matematiikkaa?

Lasse Holmström
Matemaattisten tieteiden laitos, Oulun yliopisto
lasse.holmstrom@oulu.fi

Mitä informaatioteknologia on?

Ensimmäinen reaktio otsikon kysymykseen saattaa olla, että se on puhtaasti retorinen ja tarkoitettu ehkä vain provosoimaan lukijaa. Sillä, kyllähän informaatioteknologia kai nyt sentään matematiikkaa tarvitsee!

Onhan toki niin, että informaatioteknologia ja muu niin sanottu korkea teknologia on nimenomaan *matemaattista* teknologiaa, matemaattiseen osaamiseen perustuvaa toimintaa. Perusluonnontieteiden, kuten fysiikan ja kemian kieli on matematiikka ja tekniikka on sovellettua luonnontiedettä. Ja, fyysisten prototyyppien rakentamisen sijaan teollisuuden tuotekehittely perustuu yhä useammin simulointiin, joka puolestaan nojaa matemaattisiin malleihin. Lisäksi Mooren laki ennustaa mikrosirujen transtorien määrän per pinta-alayksikkö kaksinkertaistuvan n. 18 kk:n välein pitkälle tulevaisuuteen ja tästä seuraava laskentatehon kasvu mahdollistaa aina vain monimutkaisempien matemaattisten menetelmien soveltamisen.

Tämä kaikki on epäilemättä totta, mutta ongelma onkin enemmän siinä, että mitä informaatioteknologia oikeastaan on? Googlaamalla löytää informaatioteknologialle monia arvovaltaisia määritelmiä, joista tässä on yksi, sisällöltään melko tyypillinen:

Information Technology is the science and skills of all aspects of computing, data storage, and communications.

Informaatioteknologia on siis tiedettä ja taitoja, jotka liittyvät kaikenlaiseen laskentaan, tiedon tallentamiseen ja siirtoon.

On huomattava, että aikaisemmin, ehkä vielä n. 30 vuotta sitten, informaatioteknologia tarkoitti paljolti nimenomaan tietokoneiden rakentamiseen liittyvää tekniikkaa, numeerisia teollisuussovelluksia tai ehkä kaupallishallinnollista tietojenkäsittelyä. Nämä asiat eivät toki ole kadonneet minnekään, päinvastoin, niitä tehdään enemmän kuin koskaan, mutta niiden lisäksi on tullut hyvin paljon muuta ja tosiasia on, että pääosa tämän päivän tietotekniikasta ei olekaan esimerkiksi laitetekniikkaa vaan ennenkaikkea *ohjelmistokehitystä*, jolla ei sitten välttämättä olekaan enää ainakaan suoraa yhteyttä matematiikkaan. Esimerkkejä ovat vaikkapa internet ja siihen liittyvät monet sovellukset, erilaisten laitteiden mikroprosessoriohjaus, toimisto-ohjelmistot, elokuvat ja niiden digitaaliset tehosteet, pelit ja niin edelleen.

Ohjelmistotekniikka ja matematiikka

Suurin osa tämän päivän informaatioteknologiasta onkin siis nimenomaan ohjelmistotekniikkaa ja ohjelmistotekniikan ammattilaisista iso osa ei ehkä tarvitsekaan perinteisiä matematiikan tietoja, kuten differentiaali- ja integraalilaskentaa, lineaarialgebraa ja differentiaaliyhtälöitä. Tästä syystä artikkelini otsikon kysymys ei todellakaan ole vain retorinen ja kysymykseen vastaaminen on tärkeää, jotta paremmin ymmärettäisiin matematiikan ja tietotekniikan nykyinen suhde. Tarkastellaan asiaa siis hieman lähemmin.

Timothy Lethbridge Ottawan yliopistosta teki jo noin 15 vuotta sitten tutkimuksen, jonka huomiota herättävät tulokset julkaistiin mm. arvovaltaisessa yhdysvaltalaisessa IEEE Computer lehdessä kansikuvajuttuna toukokuussa 2000 [1]. Lethbridge kysyi 200 useasta maasta kotoisin olevalta ohjelmistoammattilaiselta mm. mitä hyötyä heillä on ollut aikoinaan oppimistaan asioista. Tarkasteltavana oli 75 oppialaa tai osaamisaluetta, mm. perinteisesti yliopistoissa opetettavat matematiikan kurssit. Ohjelmistoammattilaisista kun oli kyse, ei liene yllätys, että kaikkein tärkeimpinä pidettiin välittömästi ohjelmistotyöhön liittyviä aloja kuten eri ohjelmointikielet, tietorakenteet, ohjelmistojen suunnittelumallit ja arkkitehtuurit. Korkealla kärjessä olivat myös etiikka, eikä hyvää neuvottelutaitoakaan väheksytty – se esimerkiksi tuli ennen ensimmäistäkään matematiikan kurssia. Nimittäin, matemaattisista oppialoista parhaiten sijoittui todennäköisyyslaskenta ja tilastotiede, mutta nekin löytyvät vasta läheltä listan puoliväliä, sijalta 31. Vasta listan loppupuolella alkaa näkyä jotain sellaista, minkä olisi ehkä luullut sijoittuvan korkeallekin: numeeriset menetelmät, jotka ovat aivan keskeisiä esimerkiksi erilaisissa simulaatioissa ja teollisuusprosessien optimoinnissa, löytyy uskomattoman alhaalta, sijalta 52. Ja ne tavallisimmat matematiikan pää- ja sivuaineopintoina suoritettavat kurssit ovat todellakin käytännössä hännänhuippuja: lineaarialgebra ja matriisit sijalla 56, kunnianarvoisa differentiaali- ja integraalilaskenta sijalla 67 ja differentiaaliyhtälöt sijalla 72.

Entä varsinaiset matemaatikot?

Joitakin vuosia sitten tein itsekin pienimuotoisen kyselyn IT-alalla tutkimustehtävissä toimivien matemaatikoiden ja matematiikkaa käyttävien tietojenkäsittelytieteilijöiden parissa tarkoituksena saada selville, mikä heille on ollut matematiikan merkitys ja ovatko kenties jotkin matematiikan alat heidän mielestään tärkeämpiä kuin toiset. Olihan kuitenkin ehkä niin, että kun Timothy Lethbridge kysyi asiaa nimenomaan ohjelmistoammattilaisilta, saattoivat tulokset olla perinteiselle matematiikalle turhan negatiiviset. Mitä siis sanoisivat matematiikan ammattilaiset?

He sanoivat esimerkiksi, että ”eri IT-aloilla tarvitaan hyvin erilaista matematiikkaa”, mikä jo näyttää vähän rohkaisevammalta. Usein myös katsottiin, että tärkeämpää kuin tiettyjen tietotekniikkaan liittyviksi miellettyjen matematiikan alojen opiskelu ja erikoistietojen hankinta on kuitenkin lujan yleismatemaattisen perustan valaminen. Päämääränä tulisi olla vankan mutta ei välttämättä ylettömän laajan matemaattisen perusosaamisen saavuttaminen, joka sitten mahdollistaisi eri sovelluksissa tarvittavan uuden käsitteistön nopean ja kunnollisen oppimisen.

Matematiikan osaamisella katsottiin olevan ajattelua jalostava ja itseluottamusta lisäävä vaikutus. Tätä varmaan tarkoitti eräskin, joka totesi, että hieno puoli abstraktin matematiikan opiskelussa oli tunne, että ”ainakin on joskus tultu tehtyä jotain kunnolla”. Kun on selvittänyt pohjiaan myöten matematiikan monimutkaisia teoreettisia konstruktioita, voi luottaa siihen, että pystyy kirjoista ja manuaaleista oppimaan mitä vain. Tämä usko näkyi myös toteamuksessa, että ”matemaatikkojen vahvuuksia ovat ajattelun kirkkaus ja selkeys”. Eräässä informaatioteknologian suuryrityksessä tutkimusjohtajana toiminut matemaatikko piti matemaatikkoja itse asiassa ylivertaisina ohjelmistokehittäjinä todeten heistä löytyvän alalla kipeäsi kaivattua ”särmää”.

Matematiikasta saatava hyöty tyyypilliselle tämän päivän ohjelmistoammattilaiselle on siis luultavimmin epäsuoraa. Matematiikka tarjoaa työkaluja abstraktioiden tunnistamiseen ja niiden muotoiluun. Matematiikka myös opettaa keskittymään olennaisuuksiin ja siten kirkastaa ajattelua. Nämähän ovat mitä ilmeisimmin tärkeitä kykyjä rakennettaessa suuria ja monimutkaisia ohjelmistoja ja niiden kautta voi siis ohjelmistoammattilainenkin aivan aidosti ammentaa matematiikasta ainakin epäsuoraa hyötyä – sitä kaivattua särmää.

Matematiikka keskeisessä roolissa: teknis-luonnontieteellinen IT

Vaikka perinteisen matematiikan merkitys suurelle osalle ohjelmistotekniikkaa saattaakin siis olla melko vähäinen tai ainakin se usein on vain epäsuora, kokonaan oma lukunsa on kuitenkin teknisluonnontieteellinen tietotekniikka, jossa matematiikan välitön hyödyllisyys ja osaamistarpeet ovat kiistattomat.

Esimerkkinä olkoon vaikka GSM:n tulo Suomeen osana Nokian matkapuhelinbisneksen kehitystä 1980-luvulla. Siitähän Nokian voimakas nousu aikoinaan alkoi. Digitaalisen mobiilitekniikan läpimurtoon Suomessa keskeisen panoksen tuolloin antoi nimenomaan joukko nuoria matemaatikkoja ja teoreettisia tietojenkäsittelijöitä. GSM:n kehitystyössä tarvittuja matematiikan ja teoreettisen tietojenkäsittelytieteen menetelmiä olivat mm. stokastiikka, teleliikenne- ja jonoteoria, automaattien teoria ja kääntäjien teoria.

Matematiikka ja matkapuhelin

Nyt kun tarkastelumme on siirtynyt matkapuhelimeen, onkin ehkä paikallaan lähemmin tarkastella, minkälaista matematiikkaa tuohon Suomen IT-teollisuuden ikoniin oikeastaan sisältyy. Tämä katsaus ei suinkaan pyri olemaan tyhjentävä, mutta se toivon mukaan kuitenkin valaisee sitä, mitä kaikkea matematiikan saralta voidaan ja pitääkin käyttää, kun kännyköitä suunnitellaan ja rakennetaan.

Ensiksikin itse fyysinen laite. Pyrittäessä yhä monipuolisempia ominaisuuksia sisältäviin mutta kuitenkin pienikokoisiin matkapuhelimiin joudutaan käytettyjä elektronisia komponentteja jatkuvasti pienentämään. Skaalan pienuus yhdistettynä käytettyihin taajuuksiin johtaa mm. siihen, että elektroniikan suunnittelutyötä ei aina voida perustaa perinteisille virtapiirilaskuille, vaan sähkön kulkua komponenteissa on mallitettava sähkömagneettisena aaltoliikkeenä. Tämä tuo kuvaan mukaan esimerkiksi osittaisdifferentiaali- ja integraaliyhtälöt ja niihin liittyvät kehittyneet numeeriset menetelmät. Saman tyyppistä matematiikkaa tarvitaan myös ratkaistaessa äänen etenemiseen liittyviä akustiikan ongelmia, jotka tulevat vastaan matkapuhelimen pienien mutta tehokkaiden kovaäänisten ja mikrofonin suunnitelussa.

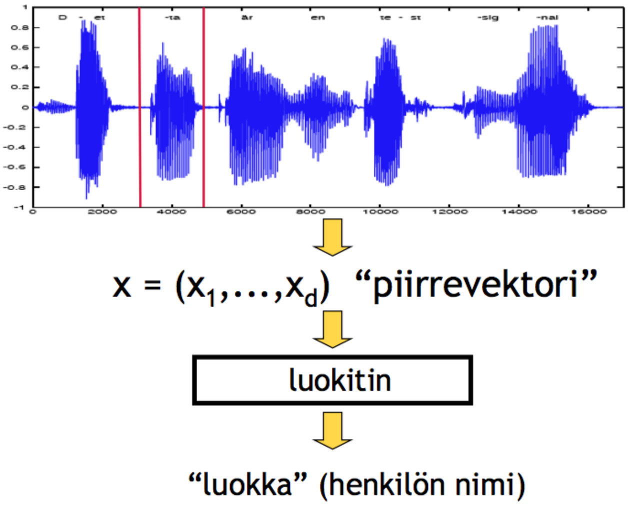
Toiseksi voidaan tarkastella puhelimen käyttöliittymää, eli sitä miten puhelinta nyt ja tulevaisuudessa käytetään. Modernia matkapuhelintahan pystyy näppäimistön lisäksi käyttämään myös puhutuilla kommennoilla ja haluttaessa voitaisiin puhelimessa olevaa kameraa käyttää myös vaikka puhujan tunnistamiseen. Tarvittava matemaattinen teknologia on niin sanottua *hahmontunnistusta* ja se hyödyntää mm. monenlaisia tilastomatemaattisia menetelmiä ja koneoppimista.

Ja kolmanneksi itse tietoliikenne kännyköiden ja tukiasemien välillä. Puhe-, teksti- tai kuvatiedon siirto, koodaus, tiivistäminen ja mahdollinen salaus ammentavat matemaattista perustaa stokastiikasta, signaalianalyysistä, algebrasta, informaatioteoriasta ja lukuteoriasta.

Fokusoimalla siis yhteen keskeiseen IT-tuotteeseen, matkapuhelimeen, nähdäänkin, että kunnon matematiikkaa tarvitaan itse asiassa hyvin monen ongelman ratkaisuun. Saadaksemme matematiikan roolista vielä yksityiskohtaisemman kuvan poraudutaan seuraavaksi tarkemmin vielä tuohon puhelimen käyttöliittymään ja sen puhetta tunnistavaan osaan.

Puheentunnistus

Miten siis kännykkä tunnistaa vaikkapa sille annetun käskyn soittaa tietylle henkilölle? Puhe on kännykälle tietysti vain signaali, aaltomuoto, jonka amplitudi muuttuu ajan funktiona. Tästä amplitudin vaihtelusta esimerkiksi soiton vastaanottajaksi halutun henkilön nimi pitää osata tunnistaa.



Kuva 1: Puheentunnistuksen eri vaiheet.

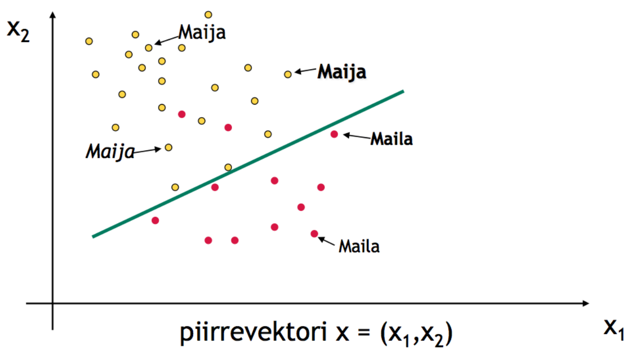
Ajatellaan vaikkapa, että kuvan 1 punaisten viivojen välissä on sen henkilön ääneen lausuttu nimi, jolle haluamme soittaa. Jo sekunninkin pätkä puhetta digitoituna tuottaa niin paljon numeerista dataa, että puhesignaali pitää yleensä puristaa tiiviimpään muotoon, jotta sen tehokas käsittely olisi mahdollista. Siksi tuosta puhutusta pätkästä voidaan ensin muodostaa niin sanottu *piirrevektori*, kuvassa 1 d:n pituinen jono reaalilukuja x_1:stä x_d:hen, joka sisältää kaiken oleellisen tiedon sen henkilön nimen tunnistamiseksi, jolle halusimme soittaa. Tässä d on jokin kokonaisluku, esimerkiksi 10, missä tapauksessa piirrevektorissamme on 10 komponenttia.

Tätä informaation tiivistämistä sanotaan piirteiden irrottamiseksi ja siinä nojataan usein erilaisiin tilastollisiin menetelmiin, joilla tiedetään olevan optimaalisia ominaisuuksia tiedon tiivistämisessä. Samalla voidaan suorittaa myös häiriönpoistoa sopivilla matemaattisilla signaalinkäsittelymenetelmillä. Tämä on tärkeätä, jos soittokomento annetaan vaikkapa hands-free -laitteiston välityksellä meluisassa autossa.

Seuraavassa vaiheessa piirrevektori sitten syötetään niin sanottuun *luokittimeen*, jonka tehtävänä on päättää, minkä henkilön nimestä olikaan kysymys. Luokitin on jossain mielessä se matkapuhelimessa, tai nykyisin usein myös niin sanotusti ”pilvessä” majailevan älyn ylin aste, joka tekee puolestamme valinnan siitä kenelle soitto menee. Siksi on kiinnostavaa vielä katsoa hieman tarkemmin, miten tuo päätös voidaan tehdä, eli katsotaan miten tyypillinen luokitin voisi toimia.

Luokiteltavissa piirrevektoreissa on tavallisesti useita komponentteja, mutta yksinkertaistamme tässä tilannetta ja tarkastelemme vain kuviteltua kaksiulotteista piirreavaruutta, jolloin piirrevektorin voi esittää tason pisteenä. Kuvan 2 pisteet esittävät matkapuhelimeen sanottuja nimiä, kun puhesignaali on puristettu kaksiulotteiseksi piirrevektoriksi eli x_1x_2-tason pisteeksi.

Jos vaikka pyysin soittamaan Maijalle, niin piste esittää ”Maija”-sanan synnyttämää puhesignaalia. Jos piste olisi eri soittokerroilla aina samassa paikassa, olisi tilanne tunnistamisen kannalta hyvin yksinkertainen, mutta näin ei todellisuudessa ole. Kuten kuvassa 2 esitetään, eri kerroilla ”Maija” tulee sanottua aina hieman eri tavalla. Tähän vaikuttavat puhenopeus, tilanne jossa nimi sanotaan, ulkoiset olosuhteet ja niin edelleen. Eri Maijan versiot siis osuvat eri osiin piirreavaruutta, eikä tässä kaikki. Puhelinluettelossamme on toki muitakin nimiä, joista jotkut saattavat läheisesti muistuttaa Maijaa. Miten erottaisimme Maijan ja Mailan, kun meluisassa ympäristössä huolimattomasti lausuttuna ne voivat muistuttaa toisiaan paljonkin ja siten täyttävät osittain saman osan piirreavaruuttamme?



Kuva 2: Kaksiulotteinen piirreavaruus, jossa matkapuhelimelle sanotut nimet on kuvattu pisteillä. Suora määrää luokittelusäännön, joka pyrkii erottamaan ”Maija” ja ”Maila” -luokat toisistaan.

Puheentunnistus saattaakin toimia mukautuvasti siten, että järjestelmän suorituskyky paranee käytön myötä. Tämä perustuu siihen, että kun matkapuhelin kuulee useita versioita samoista nimistä, se ikäänkuin oppii esimerkkien avulla, missä päin piirreavaruutta on vaikkapa Maija-alue ja missä on Maila-alue. Esimerkkien kuulemisen jälkeen puhelin pyrkii sitten erottamaan nämä alueet toisistaan jollain tavalla. Se mitä edellä sanoin luokittimeksi on täsmälleen tämä erottelusääntö. Niin sanottu *lineaarinen luokitin* esimerkiksi erottaisi Maija ja Maila -luokan sopivalla suoralla viivalla (Kuva 2). Kun suora on kerran sitten asetettu, luokitellaan sen vastakkaisille puolille osuvat puhesignaalit eri luokkiin, toiset Maijaksi ja toiset Mailaksi.

Suoraa käyttävä lineaarinen luokitin on ehkä yksinkertaisin mahdollinen menetelmä ja parempi päätössääntö saadaankin tavallisesti käyttämällä jotain yleisempää luokkia erottavaa käyrää. Kun piirreavaruuden dimensio on korkeampi kuin 2, kuten se yleensä on, erotetaan luokat hypertasolla tai jollain yleisemmällä korkeaulotteisella pinnalla. Optimaalinen luokkien välinen rajapinta on sellainen, että siihen perustuva luokitin tekee mahdollisimman harvoin virheen. Käyttäen todennäköisyyslaskentaa voidaan osoittaa, että optimaalinen luokitin riippuu luokkien todennäköisyysjakaumista piirreavaruudessa ja paras luokkia erottava rajapinta määräytyy tietyllä tavalla jakaumien todennäköisyystiheysfunktioiden korkeuseroista. Optimaalinen luokitin ei tavallisesti ole lineaarinen, vaan luokkia erottava pinta saattaa olla periaatteessa minkä muotoinen tahansa.

Yksinkertaisen kaksiulotteisen esimerkkimme luokittelusäännön määrittely käy vielä melko helpon tuntuisesti, mutta todellisuudessa piirrevektori harvoin on 2-ulotteinen vaan ehkä 10-, 100- tai vaikkapa 1000-ulotteinen ja jakaumia erottava optimaalinen rajapinta saattaa olla hyvin monimutkainen. Lisäksi teoreettisia jakaumia ei käytännössä tunneta, vaan ne tulee jotenkin estimoida puhe-esimerkkien avulla. Selvää on, että korkeaulotteisissa piirreavaruuksissa tehtävä on tällöin sekä matemaattisesti että myös käytännön toteutuksen kannalta haastava.

Siirryttäessä yksittäisten nimien tunnistamisesta pidempien, vapaamuotoisesti annettujen komentojen ja kysymysten käsittelyyn tarvitaankin sitten jo hyvin kehittyneitä teoreettisia menetelmiä ja todella runsaasti laskentaresursseja. Siksi uusimmissa matkapuhelimissa toimivissa puheentunnistusjärjestelmissä laskenta itse asiassa tehdäänkin osittain tai pääasiassa pilvessä, eli sovellusta ylläpitävän yrityksen palvelimilla. Se, että kännyköiden puheentunnistus onnistuu, osoittaa, että toimivia ratkaisuja on löydetty. Tässä vaiheessa lukijalle ei lienekään enää yllätys, kun totean, että monet tehokkaat tekniikat perustuvat nimenomaan kehittyneisiin matemaattisiin ja tilastollisiin menetelmiin.

Puheentunnistus on itseasiassa vain yksi esimerkki hahmontunnistuksesta. Kun puhesignaali korvataan vaikka aivosähkökäyrällä tai kudoksesta otetulla magneettikuvalla, päästään matemaattisia hahmontunnistusmenetelmiä soveltamaan lääketieteeseen. Voimakkaasti kasvavan kiinnostuksen kohteena on myös bioinformaatioteknologia, jossa hahmontunnistusta voidaan soveltaa mm. DNA-sirujen analyysiin. Huomattava informaatioteknologiaa hyödyntävä ala on myös robotiikka, jossa hahmontunnistusta on pitkään käytetty antamaan roboteille alkeellinen näköaisti.

Ja kyllä sitä niin sanottua perinteistäkin matematiikkaa informaatioteknologiassa ja ohjelmistotekniikassa tarvitaan. Esimerkiksi lineaarialgebraa, geometriaa ja pintojen teoriaa hyödynnetään tietokoneavusteisessa suunnittelussa, kolmiulotteisten kappaleiden tietokonemallintamisessa ja tietokonegrafiikassa – ja esimerkiksi elokuvien ja pelien digitaalisissa tehosteissa. Esimerkiksi kuvan 3 hahmot ja maailma, jossa ne seikkailevat, ovat lopultakin nekin matemaattisia konstruktioita.

Kuva 3: Ruutukaappaus mm. Applen älypuhelimille saatavasta Infinity Blade II -pelistä. Copyright Chair Entertainment and Epic Games.

Johtopäätöksiä

On totta, että informaatioteknologia nykyään on paljolti puhdasta ohjelmistotekniikkaa, jolle hyöty matematiikasta saattaa usein olla vain epäsuoraa. Kuitenkin, matematiikkaa on tarvittu ja tullaan jatkossakin tarvitsemaan paljon nimenomaan monien todella haastavien ongelmien ratkaisuun. Tämä pätee erityisesti teknis-luonnontieteelliseen informaatioteknologiaan. Tarvittavien ja potentiaalisesti hyödynnettävien matemaattisten menetelmien kirjo on todella laaja ja kasvaa jatkuvasti. Tämän takaa jo pelkästään alati kasvava laskentateho, joka tuo entistä eksoottisemmat ja vaativammat menetelmät käytännön sovellusten ulottuville.

Sen sijaan, että kysytään ”Tarvitseeko informaatioteknologia matematiikkaa?”, pitäisikin siksi mielestäni kysyä ”Onko sellaista matematiikkaa, jota informaatioteknologia EI tarvitse?”.

Viitteet

[1] Timothy C. Lethbridge. What knowledge is important to a software professional? *Computer*, 33:44–50, 2000.

Verkko-Solmun oppimateriaalit

Osoitteesta `http://solmu.math.helsinki.fi/oppimateriaalit.html` löytyvät oppimateriaalit:

Kilpailumatematiikan opas (Matti Lehtinen)

Geometrian perusteita (Matti Lehtinen)

Geometria (K. Väisälä)

Lukualueiden laajentamisesta (Tuomas Korppi)

Jaksolliset desimaaliesitykset algebrallisesta näkökulmasta (Jaska Poranen ja Pentti Haukkanen)

Algebra (Tauno Metsänkylä ja Marjatta Näätänen)

Algebra (K. Väisälä)

Matemaattista fysiikkaa lukiolaiselle 1: Mekaniikkaa (Markku Halmetoja ja Jorma Merikoski)

Matemaattista fysiikkaa lukiolaiselle 2: Sähköoppia (Markku Halmetoja ja Jorma Merikoski)

Lukuteorian helmiä lukiolaisille (Jukka Pihko)

Matematiikan peruskäsitteiden historia (Erkki Luoma-aho)

Matematiikan historia (Matti Lehtinen)

Reaalianalyysiä englanniksi (William Trench)