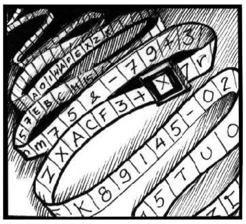


Polynomit, interpolaatio ja funktion approksimointi

Heikki Apiola
Lehtori
Matematiikan laitos, Teknillinen korkeakoulu

Johdanto, taustaa

Kirjoitus liittyy aihepiiriin *numeerinen analyysi, tieteellinen laskenta, tietokoneen käyttö matematiikassa*. Siihen liittyviä kirjoituksia on jonkinverran esiintynyt Solmun historian aikana, esimerkiksi

Jouni Seppänen: Fibonaccin lukujen laskennasta (`solmu.math.helsinki.fi/1998/2/seppanen/`),

Oma kirjoitukseni symbolilaskentaohjelmistosta Maple (`solmu.math.helsinki.fi/1999/5/apiola/`),

Antti Rasila: Numeerista matematiikkaa Python-kielellä (`solmu.math.helsinki.fi/2004/2/python2.pdf`).

Tarkoitukseni on aloittaa kirjoitussarja, jossa käsitellään numeerisen matematiikan eri teemoja siinä hengessä, että esitiedoiksi riittävät lukion matematiikan tiedot. Kirjoitukset voisivat parhaassa tapauksessa tarjota ideoita lukion erikoiskursseille, jotka ovat tyyppiä "numeerinen analyysi/tieteellinen laskenta/matemaattinen mallinnus".

Samalla esittelen alan tietokoneohjelmistojen käyttöä. Niitä on kahta päätyyppiä: numeeriset ja symboliset. Jälkimmäisistä kirjoitin yllä mainitussa viitteessä aika laajasti Tällä kertaa esittelen pientä nurkkaa suuresta ja kauniista ohjelmasta nimeltään *Matlab*. Kyseessä on hyvin tehokas ja suuren suosion maailmalla saanut tieteellisen laskennan työkalu. Kts. `www.mathworks.com`.

Lukija, joka haluaa etupäässä seurata aiheen matemaattista kehittelyä, voi jättää ohjelmakoodit ja selostukset lukematta ja suorittaa joitakin laskuja vaikkapa omalla laskimellaan. Toisaalta Matlab:sta kiinnostunut lukija voi opetella rinnakkain sekä Matlab-ajattelua että sen tukemaa matematiikkaa. Tässä on hyvä käyttää lisäapuna vaikkapa opasta: `www.math.hut.fi/~apiola/matlab/opas/lyhyt/`

Harvalla koululaisella on Matlab-ohjelma käytössään, siksi onkin suositeltavaa hakea verkosta julkisohjelma Octave, `www.octave.org/`, joka on "riisuttu versio" Matlab:sta. Sillä voidaan tehdä kaikki kirjoituksen esimerkit, ja sen avulla pääsee sisälle Matlab:n ajatusmaailmaan.

Luettavuuden parantamiseksi ja matemaattisen juonen seuraamisen helpottamiseksi sijoitan suurimman osan ohjelmakoodeista ja ohjeista tekstitiedostoihin, joiden sisältöä en ota varsinaiseen kirjoitukseen mukaan. Monet näistä ovat ajovalmiita Matlab-skriptejä, eli komentotiedostoja. Myös kaikki kirjoituksen kuvien tekemiseen käytetyt Matlab-skriptit ovat mukana. Koodit ja ohjeet ovat saatavilla sivulta `solmu.math.helsinki.fi/2004/3/apiola/`, josta ne voi haluttaessa suoraan leikata/liimata Matlab-istuntoon.

Symbolilaskennasta kiinnostunut lukija voi aivan hyvin (ja jopa helpomminkin) tehdä esimerkit Maplella tai Mathematicalla. Edellisen suhteen ohjeita on saatavissa edellä mainitusta kirjoituksesta `solmu.math.helsinki.fi/1999/5/apiola`. Jos sinulla sattuu olemaan Maple käytettävissäsi ja haluat ohjeita esimerkkeihin, niin lähetä mailia: `heikki.apiola@hut.fi`, saat paluupostissa asiaankuuluvan Maple-työarkin.

Esitiedot. Kirjoituksen lukemiseen tarvittavat matemaattiset esitiedot sisältävät vain perusalgebraa, hieman tottumusta polynomien käsittelyssä ja funktioopin perusteita.

Johdattelua

Ajatellaanpa, että veden viskositeetti on määrätty kokeellisesti eri lämpötiloissa, ja saatu seuraavanlainen taulukko, joka on annetulla tarkkuudella virheetön.

Lämpötila	2°	5°	7°	15°
Viskositeetti	1.670	1.519	1.430	1.140

Taulukko 1.

Voitaisiin kysyä vaikkapa viskositeetin arvoa lämpötilassa 10°.

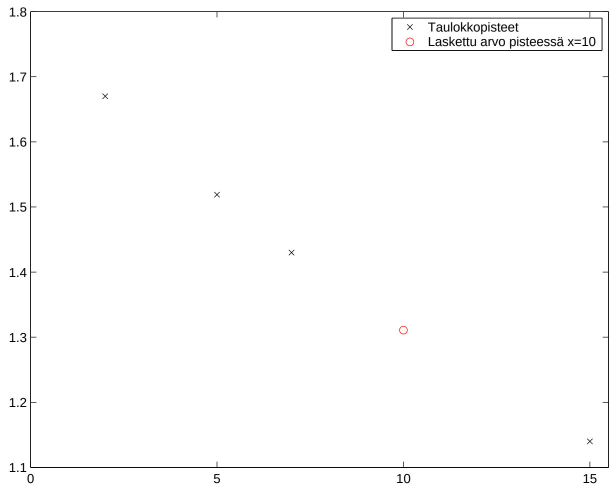
Eräs ratkaisuyritys olisi sovittaa aineistoon polynomi siten, että se kulkee kaikkien annettujen taulukkopisteiden kautta. Tällöin on luonnollista hakea kolmannen asteen polynomia, koska siinä on neljä määrättävää kerrointa, ja annettuna on sama määrä pisteitä. Saadaan neljän yhtälön ja neljän tuntemattoman yhtälöryhmä, joka voidaan ratkaista eliminaatiomenetelmällä. Näin saatavien kertoimien avulla voidaan kirjoittaa ratkaisupolynomi:

$$p(x) = -2.6282 \times 10^{-5} x^3 + 1.5346 \times 10^{-3} x^2 - 6.0051 \times 10^{-2} x + 1.7842$$

Kts. `viskositeetti.m`.

Tällaista polynomia, jonka kuvaaja kulkee kaikkien annettujen taulukkopisteiden kautta, sanotaan tähän taulukkoon (”dataan”) liittyväksi *interpolaatiopolynomiksi*.

Asettamallemme tehtävälle saadaan nyt ratkaisuapproksimaatio laskemalla $p(10) = 1.310$.



Kuva 1. Mittauspisteitä ja interpoloitu piste.

Tässä kirjoituksessa opitaan (vielä) yksinkertaisempi menetelmä vastaavanlaisten tehtävien ratkaisemiseksi. Menetelmän verraton arvo on siinä, että samalla, kun saadaan kauniin yksinkertainen ratkaisutapa, tarjoutuu myös helppo tapa yksikäsitteisen ratkaisun olemassaolon todistamiselle.

Miten voidaan tutkia edellä olevan ratkaisun mielekkyyttä ja virhekäytöstä? Näitä keskeisiä kysymyksiä kosketellaan (kevyesti) joidenkin dramaattistenkin esimerkkien valossa kirjoituksen loppupuolella. Samalla pohdiskellaan, milloin interpolaatio soveltuu ja milloin ei, ja esitellään myös muiden funktioiden kuin polynomien käyttöä tarkoitukseen.

Approksimointia interpoloiden ja interpoloimatta

Olkoon annettu taulukko

x_0	x_1	x_2	$\dots$	x_n
y_0	y_1	y_2	$\dots$	y_n

Taulukko 2.

Edellä olevaa johdantoesimerkkiä mukaillen ja yleistäen voidaan kysyä funktiota g, jonka kuvaaja kulkee annettujen taulukkopisteiden kautta.

Luonnollisin lähtökohta on etsiä polynomia. Tehtävänä on silloin määrittää korkeintaan astetta n oleva polynomi p, joka toteuttaa ehdot

$$p(x_0) = y_0, p(x_1) = y_1, \dots, p(x_n) = y_n.$$

Interpolaatio tarjoaa joissakin tapauksissa menetelmän annetun funktion approksimointiin annetulla välillä.

Tarkastellaan tässä lyhyesti myös muita approksimaatiotapoja myös muilla funktioilla kuin polynomeilla.

Katsotaan tilannetta kolmelta eri näkökannalta.

1. Voidaanko löytää yksinkertainen matemaattinen funktio g, jonka arvoja nämä annetut taulukkoarvot ovat, ts. $g(x_k) = y_k, k = 0, \ldots, n$?

2. Taulukko on peräisin mittauksista, joissa saattaa esiintyä virheitä. Tehtävänä olisi löytää matemaattinen lauseke, joka approksimoi aineistoa, mutta kuvaaja ei kulje tarkalleen annettujen taulukkopisteiden kautta.

3. On annettu funktio f, mahdollisesti tietokoneohjelman muodossa. Funktion arvojen $f(x)$ laskenta on "kallista", eli vaatii paljon laskentatehoa. Kysymys kuuluu: Voidaanko löytää yksinkertaisempi funktio g, joka approksimoi riittävän tarkasti funktiota f ja jonka arvojen laskenta on "halvempaa". Funktiolta g vaaditaan usein lisäksi yksinkertaisuutta esimerkiksi siten, että sitä on helppo derivoida ja integroida.

Kohdassa 1) on kysymys interpolaatiosta, funktio g voi olla polynomi, mutta se voi olla muutakin tyyppiä.

Kohdan 2) tapauksessa interpolaatio ei ole järkevä tapa, koska on turha yrittää pakottaa approksimaatiota kulkemaan virhettä sisältävien arvojen kautta tarkasti. Tähän sopii yleensä hyvin ns. pienimmän neliösumman approksimaatio, jolla annettujen arvojen pääsuunta, "trendi" saadaan esitetyksi.

Kohtaan 3) voi soveltua interpolaatio, mutta tilanteesta riippuen myös jokin muu approksimointitapa.

Polynomit ovat hyvä lähtökohta approksimoiviksi funktioiksi. Niillä on helppo suorittaa yhteen- vähennys- ja kertolaskuja, niitä voidaan derivoida ja integroida helposti, jne.

Muitakin funktioluokkia esiintyy sovellutuksissa. Jaksollisten ilmiöiden mallintamisessa on luonnollista käyttää "trigonometrisia polynomeja", eli muotoa

$$a_0 + a_1 \cos x + a_2 \cos 2x + \ldots + b_1 \sin x + b_2 \sin 2x + \ldots$$

olevia funktioita.

Entäpä, jos käytössämme on suuri taulukko, sanokaamme 1000 arvoparia (x_k, y_k)? Tällöin interpolaatiopolynomin asteluku voisi olla 999. Tällainen polynomi heilahtelee varsin voimakkaasti, ja sen laskenta on muutenkin hyvin työlästä ja virhealtista. Tilanteeseen soveltuu luontevasti ns. splinifunktio, joka koostuu "polynomipaloista". Näitä esitellään lyhyesti tarinamme lopussa.

Kouluesimerkki, logaritmitaulukko

Lähdetään liikkeelle jostain taulukoidusta funktiosta. Kaikille vanhemman polven koulunkävijöille tuttuakin tutumpi funktiotaulukko on logaritmitaulu. Kun taulukon käytön tekniikan osasi, sai varmasti yhden laskun ylioppilaskirjoituksissa aikaan.

Tehtävänä oli yleensä laskea "vaikea tehtävä", kahden monella numerolla annetun luvun a ja b tulo hakemalla taulukosta $\log a$ ja $\log b$ ja laskemalla näiden summa ("helppo tehtävä"). Alkuperäisen tehtävän ratkaisu saatiin hakemalla taulukosta logaritmipuolelta summaa $\log a + \log b$ lähinnä oleva y-arvo ja katsomalla käänteiseen suuntaan vastaava x-arvo.

Jos haluat lisähavainnollistusta ja hieman logaritmiharjoittelua, avaapa sivu `www.eminent.demon.co.uk/sliderul.htm`. Kuvan ja perusteellisempaa opastusta aiheeseen löydät vaikkapa sivulta `www.sliderules.clara.net/`.

Tehtävätyyppi on mekaanisena suorituksena aika mielenkiinnoton, mutta sisältää koko joukon matemaattista viisautta. Kuten yllä olevissa viitteissä esitellään, periaate on implementoitu elegantiksi laskentavälineeksi, laskutikuksi, jota ilman ei vielä niinkin myöhään kuin 1960–1970-lukujen vaihteessa voitu kuvitella luonnontieteilijöiden ja insinöörien koskaan pärjäävän.

Lisäksi siinä näkyy pelkistetyssä muodossa eräs varsin paljon matematiikassa ja sen sovelluksissa käytettävä yleinen periaate: "Vaikea" tehtävä muunnetaan sopivalla muunnoksella "helpoksi", ratkaistaan "helppo" tehtävä ja käänteismuunnetaan "helppo" ratkaisu.

Tämänkertaisen teeman kannalta oleellista on, että tuossa koulutehtävässä oli mukana interpolaatio. Logaritmien summa ei yleensä osu tarkalleen taulukoituun arvoon, joten sitä joudutaan pyöristämään. Jos arvo on lähempänä puoliväliä kuin kumpaakaan päätepistettä, saadaan parempi tarkkuus laskemalla päätepistearvoja vastaavien x-arvojen keskiarvo. Hienommin sanottuna, suoritetaan *lineaarinen interpolaatio* taulukkopisteiden välillä.

Esimerkkinä lasketaan taulukko logaritmifunktion arvoista pisteissä $1.0, 1.2, 1.4, 1.6, 1.8, 2.0$.

Suoritamme laskut MATLAB:lla (tai OCTAVE:lla).

MATLAB-ohjelmalle annettavat komennot alkavat kehotemerkeillä `(>>)`. Ohjelman palauttamat tulokset ovat komentoa seuraavilla riveillä ilman kehotealkua. (Huomaa, että MATLAB:ssa `log` tarkoittaa luonnollista, e-kantaista logaritmia.)

```
>> X=1:0.2:2     % Luvut alkaen 1:stä, 0.2:n välein, 2:een saakka.
X =
    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
>> Y=log(X)      % Logaritmifunktion arvot X-pisteissä.
Y =
         0    0.1823    0.3365    0.4700    0.5878    0.6931
>> xytaulukko=[X;Y]   % 2-rivinen ‘‘matriisi’’,jossa X- ja Y-arvot
allekkain,
xytaulukko =
    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
         0    0.1823    0.3365    0.4700    0.5878    0.6931
```

Matlab-käsitteitä

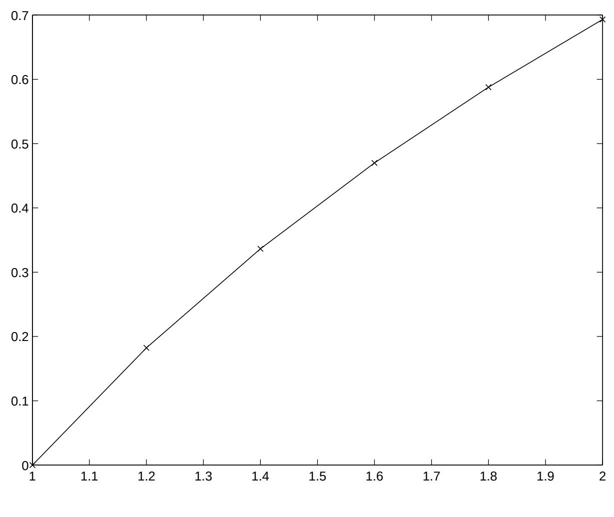
Jos haluat perehtyä aiheeseen tarkemmin, pääset alkuun edellä mainitulla lyhyellä www-oppaalla, kts. myös kirjallisuusviitettä 5.

Esittelemme nyt lyhyesti näihin riveihin liittyviä periaatteita.
Matlab operoi ”*matriiseilla*”, jolla tarkoitetaan suorakulmion muotoista lukutaulukkoa. Erikoistapaus matriisista on *vektori*, jossa on vain yksi rivi (vaakavektori) tai yksi sarake (pystyvektori).

Muuta Matlab-oppia emme varsinaiseen kirjoitukseen sisällytä. Oheismateriaalina olevat Matlab-komentotiedostot on varustettu runsailla selittävillä ja opettavaisilla kommenteilla. Ne ovat `.m`-loppuisia tekstitiedostoja ja saatavissa siis sivulta

`solmu.math.helsinki.fi/2004/3/apiola/`.

Matriisilaskentaa ei tarvita kirjoituksen ymmärtämiseksi, mutta olkoon se pikku maistiaisena siitä käsitteistöstä, jonka kaikki matematiikkaa ja sen sovelluksia koulun jälkeen opiskelevat hetimiten kohtaavat. Samalla saamme hyödyllisen puhetavan, jonka avulla Matlab-kielen operaatioita on helppo kuvata ja ymmärtää.



Kuva 2. Logaritmifunktion paloittain lineaarinen interpolaatio.

Yllä olevassa laskussa muodostamme vektorin `X`, johon sovellamme funktiota `log`. Tällöin Matlab soveltaa funktiota `log` argumenttivektorin jokaiseen komponenttiin, joten saamme yhdellä käskyllä kaikkien `log`-funktion arvojen vektorin `Y`.

Jos suoritamme Matlab-komennon `plot(X,Y)`, ohjelma piirtää `(X(k),Y(k))`-pisteiden väliset janat. Näin saamme kuvan, joka esittää logaritmifunktion *“paloittain lineaarista interpolaatiota”* annetuissa x-pisteissä.

Ennenkuin jatkamme approksimointiteemaa, kertaamme hiukan polynomien ominaisuuksia.

Polynomeista

Koulumatematiikassa polynomit tulevat varmasti jokapäiväisiksi tuttaviksi. Niillä opitaan suorittamaan peruslaskutoimituksia, derivointi ja integrointi on sujuvaa. Niiden kuvaajat tulevat tutuiksi, ainakin alhaisilla asteluvuilla, polynomiyhtälöitä opitaan ratkaisemaan, kun asteluku ≤ 2, jne.

Tässä kirjoituksessa valotan eräitä polynomien käyttöalueita, joita matematiikassa on miltei rajattomasti.

Aloitan kertaamalla koulusta tutun lauseen.

Lause 1. Jos polynomilla

$$p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$

on nollakohta x_0, niin $p(x)$ on jaollinen $(x - x_0)$:lla.

Todistus. Muodostetaan erotus

$$p(x)-p(x_0) = a_1(x-x_0)+a_2(x^2-x_0^2)+\ldots+a_n(x^n-x_0^n).$$

Jokaisessa termissä $(x^k - x_0^k)$ on $(x - x_0)$ tekijänä johtuen kaavasta

$$x^k - x_0^k = (x-x_0)(x^{k-1}+x^{k-2}x_0+\ldots+xx_0^{k-2}+x_0^{k-1}).$$

□

Tehtävä 1. Johda yllä oleva kaava. Voit laskea ensin vaikka tapauksen $k = 3$, jolloin varmasti näet koko juonen. Kerro vain auki kaavan oikea puoli sopivassa järjestyksessä.

Saamme heti yksinkertaisen, mutta tärkeän tosiasian.

Seuraus 2. [Yksikäsitteisyys] Jos kaksi korkeintaan astetta n olevaa polynomia yhtyy $(n+1)$:ssä eri pisteessä, niin ne yhtyvät kaikkialla (ts. ovat identtiset).

Todistus. Olkoot p ja q korkeintaan astetta n olevia polynomeja, jotka saavat samat arvot pisteissä $x_0, \ldots, x_n$. Tällöin erotuspolynomi $r(x) = p(x) - q(x)$ on niinikään korkeintaan astetta n. Olkoon tuo asteluku $d \le n$. Erotuspolynomilla r on oletuksen mukaan $d+1$ (jopa $n+1$) erillistä nollakohtaa $x_0, \ldots, x_d$. Kun lausetta sovelletaan peräkkäin d kertaa, seuraa erotuspolynomille esitys

$$r(x) = c(x - x_0) \cdot \ldots \cdot (x - x_{d-1}),$$

missä c on jokin vakio. Koska myös $r(x_d) = 0$ ja kaikki tekijät $x_d - x_i, i = 0 \ldots d-1$ ovat nollasta erillisiä, on tulon nollasäännön mukaan oltava $c = 0$, eli erotuspolynomi $r(x) = p(x) - q(x)$ on identtisesti 0. □

Huomautus 1. Yksinkertaisimmassa tapauksessa $n = 1$ on geometrisesti kyse siitä, että tason kaksi pistettä määrää yksikäsitteisesti suoran. Tapaus $n = 2$ tarkoittaa, että 3 pistettä määrää yksikäsitteisesti paraabelin.

Huomautus 2. Jos olemme tavalla tai toisella löytäneet interpolaatiotehtävälle ratkaisun, niin seuraus 2:n mukaan tämä on ainoa ratkaisu. (Toki polynomit saattavat esiintyä erilaisissa muodoissa, mutta yksikäsitteisyys (samuus) merkitsee, että ne ovat sievennettävissä toisikseen.)

Lagrangen interpolaatiomenetelmä

Kerrataan vielä:

Interpolaatiotehtävä Annettu x_k- ja y_k-pisteet, $k = 0, \ldots, n$ (taulukko 1). Määrättävä korkeintaan astetta n oleva polynomi p siten, että $p(x_k) = y_k, k = 0, \ldots, n$.

Voisimme muodostaa yhtälösysteemin polynomin kertoimien ratkaisemiseksi, kuten teimme johdantona olevassa viskositeettiesimerkissä. Jotta saisimme tätä kautta osoitetuksi ratkaisun olemassaolon yleisesti, joutuisimme kohtalaisen pitkiin matriisilaskennallisiin kehittelyihin. Lisäksi tämä ratkaisutapa on numeerisesti "häiriöaltis", pienillä lähtövirheillä on taipumus moninkertaistua laskun kuluessa.

Esitämme hämmästyttävän suoraviivaisen ratkaisun tehtävällemme. Se on täysin riippumaton yhtälöryhmien teoriasta. Emme tarvitse matriiseja emmekä determinantteja. Saamme olemassaolon todistetuksi konstruoimalla suoraan käyttökelpoisen muodon tehtävän ratkaisulle.

Itse asiassa tehtävälle on kaksi nerokkaan yksinkertaista ratkaisutapaa, joihin kumpaankin liittyy kuuluisan matemaatikon nimi: *Lagrange* ja *Newton*. Esitämme ratkaisun edellisen mukaan.

Lineaarinen interpolaatio

Jotta idea tulisi esiin mahdollisimman pelkistetysti, lähdetään yksinkertaisimmasta tilanteesta, jossa pisteitä on kaksi ja kyseessä on siten lineaarinen interpolaatio.

Analyyttisen geometrian tiedoilla osaamme muodostaa suoran kahden annetun pisteen (x_0, y_0) ja (x_1, y_1) kautta. Voimme kirjoittaa interpolaatiopolynomin muotoon:

$$p(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0).$$

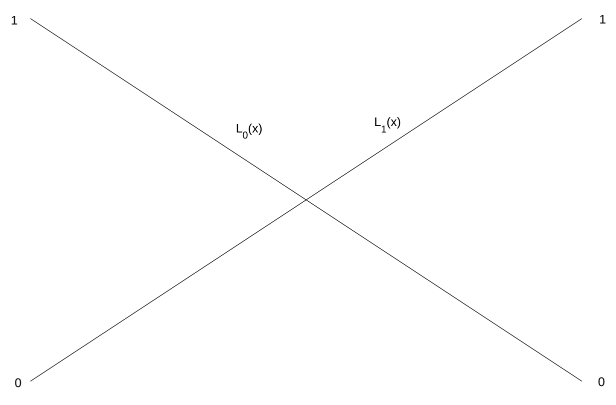
Jotta saataisiin yleistyskelpoinen muoto, kirjoitetaan kaava painotettuna keskiarvona y-arvoista keräämällä y-termien kertoimet tekijöiksi:

$$p(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}.$$

Merkitään kaavassa esiintyviä ensimmäisen asteen polynomeja $L_0(x)$ ja $L_1(x)$, jolloin siis

$$p(x) = y_0 L_0(x) + y_1 L_1(x).$$

Polynomeilla L_0 ja L_1 on ominaisuudet $L_0(x_0) = 1, L_0(x_1) = 0$ ja $L_1(x_0) = 0, L_1(x_1) = 1$. Niitä kutsutaan 1. asteen *Lagrangen kertojapolynomeiksi*.



KUVA 3. 1. asteen Lagrangen polynomit L_0 ja L_1.

Esimerkki 1. Olkoon annettu logaritmitaulukkoarvot $\ln 9.0 = 2.1972$ ja $\ln 9.5 = 2.2513$. Laske lineaarista interpolaatiota käyttäen likiarvo $\ln 9.2$:lle.

Ratkaisu. Lasketaan Lagrangen polynomit, kun $x_0 = 9.0, x_1 = 9.5$. (Huomaa, että Lagrangen polynomit määräytyvät pelkästään x_0- ja x_1-arvoista.)

$$L_0(x) = \frac{x - 9.5}{9.0 - 9.5},\ L_1(x) = \frac{x - 9.0}{9.5 - 9.0}.$$

Kun sijoitetaan $x = 9.2$, saadaan $L_0(9.2) = 0.6$, $L_1(9.2) = 0.4$. Siis $p(9.2) = y_0 \cdot 0.6 + y_1 \cdot 0.4 = 2.1972 \cdot 0.6 + 2.2513 \cdot 0.4 = 2.2188$.

Miten suuri virhe tehdään? Logaritmin arvo 5:n numeron tarkkuudella on 2.2192, joten virhe on 2.2192 - 2.2188 = 0.0004. Pyöristyksen jälkeen saamme 4 oikeaa numeroa.

Kvadraattinen (eli toisen asteen) interpolaatio

Miten voisimme yleistää edellä olevaa menettelyä? Kun siirrymme lineaarisesta kvadraattiseen tapaukseen, tulee samalla selvästi näkyviin, miten yleinen tilanne hoidetaan.

Nyt on siis annettu 3 pistettä x_0, x_1, x_2 ja vastaavat y_0, y_1, y_2. Jos osaisimme muodostaa toisen asteen polynomit L_0, L_1, L_2 siten, että

$$\begin{array}{lll} L_0(x_0) = 1, & L_0(x_1) = 0, & L_0(x_2) = 0 \\ L_1(x_0) = 0, & L_1(x_1) = 1, & L_1(x_2) = 0 \\ L_2(x_0) = 0, & L_2(x_1) = 0, & L_2(x_2) = 1, \end{array}$$

voisimme kirjoittaa interpolaatiopolynomin heti:

$$p(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x).$$

Perustelu:

1) Polynomi on toisen asteen polynomien summana korkeintaan astetta 2.

2) Lisäksi

$$p(x_0) = y_0 \underbrace{L_0(x_0)}_{1} + y_1 \underbrace{L_1(x_0)}_{0} + y_2 \underbrace{L_2(x_0)}_{0} = y_0.$$

Aivan samoin saadaan $p(x_1) = y_1$ ja $p(x_2) = y_2$.

Siispä asettamamme tehtävän ratkaisuna on tämä polynomi p. Kaiken lisäksi se on seuraus 2:n perusteella yksikäsitteinen.

Miten sitten tuollaiset L_k-polynomit löydetään? Katsotaan vaikka L_0 :aa. Polynomin pitää saada arvo 0 pisteissä x_1 ja x_2. Sellainen polynomi on $c(x - x_1)(x - x_2)$, missä c on mielivaltainen vakio. Määrätään vakio c siten, että ehto $L_0(x_0) = 1$ toteutuu, ts. $c(x_0 - x_1)(x_0 - x_2) = 1$, josta saadaan kerroin $c = 1/((x_0 - x_1)(x_0 - x_2))$, joten

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}.$$

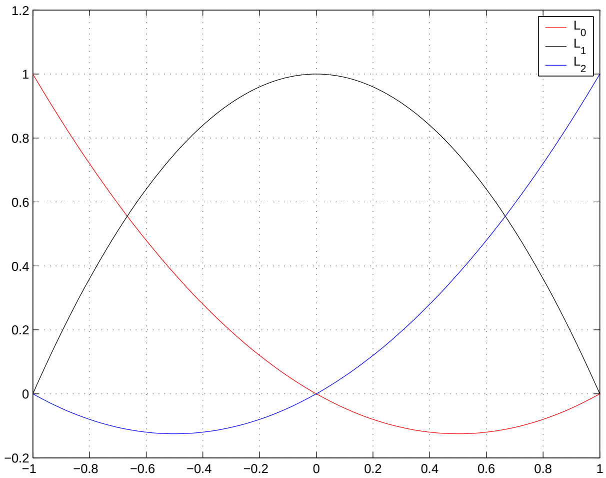
Aivan samoin saadaan

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)},$$
$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

Jos merkitään $l_0(x) = (x - x_1)(x - x_2), l_1(x) = (x - x_0)(x - x_2)$, $l_2(x) = (x - x_0)(x - x_1)$, niin

$$L_k(x) = \frac{l_k(x)}{l_k(x_k)}, k = 0, 1, 2.$$

Sanallisesti sanottuna: $L_k(x)$:n osoittajassa on tekijät $(x - x_j), j \neq k$, ja nimittäjä saadaan osoittajasta korvaamalla x arvolla x_k.



KUVA 4. 2. asteen Lagrangen polynomit L_0, L_1 ja L_2.

Tehtävä 2. Täydennä edellistä esimerkkiä siten, että otat lisäpisteen $\ln 11.0 = 2.3979$. Laske näin saatavan toisen asteen polynomin arvo samassa pisteessa $x = 9.2$ ja vertaa virheitä.

Huomautus 3. *Lagrangen* menetelmän varjopuoli on, että lisättäessä interpolaatiopisteitä, joudutaan kaikki laskut suorittamaan uudestaan. Tässä suhteessa edellä mainittu *Newtonin* menetelmä on etevämpi.

Kuten luonnollista on, tarkkuus paranee, kun annettuja pisteitä lisätään: approksimoitavan funktion kannalta ajatellen on luonnollista, että kun käytettävissämme on yksi lisäparametri, jolla suora voidaan "taivuttaa" paraabeliksi, niin tarkkuutta saadaan parannetuksi.

Yleinen tapaus

Annettu pisteet $x_0, \ldots, x_n$ ja vastinpisteet $y_0, \ldots, y_n$. Etsitään siis korkeintaan astetta n olevaa polynomia p, jolle $p(x_k) = y_k, k = 0, \ldots, n$.

Edellinen yleistyy nyt ilmeisellä tavalla.

1) Muodostetaan n-asteiset Lagrangen kantapolynomit, joilla on ominaisuus: $L_k(x_j) = 1$, kun $k = j$ ja 0, kun $k \neq j$.

Kirjoitetaan kaava L_0:lle:

$$L_0(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)}.$$

Vastaavasti saadaan muut kantapolynomit $L_k, k = 1, \ldots, n$.

Sääntö. Kenties on selvempää antaa sanallinen kuvaus L_k-polynomin muodostamissäännöstä kuin yleinen kaava:

$L_k(x)$ saadaan osamääränä, jonka osoittajassa on termien $(x - x_j)$ tulo, josta puuttuu tekijä $(x - x_k)$. Nimittäjä saadaan korvaamalla osoittajan x luvulla x_k. (Puuttuva termi on juuri se, jossa tapahtuisi tällä kohdalla 0:lla jako.)

2) Aivan samoin kuin edellä kvadraattisessa tapauksessa, nähdään heti, että ratkaisuna tehtävällemme on polynomi

$$p(x) = y_0 L_0(x) + y_1 L_1(x) + \ldots + y_n L_n(x).$$

Kootaan tulos vielä lauseeksi.

Lause 3. Polynomi-interpolaatiotehtävällä (PI) on yksikäsitteinen ratkaisu p, joka voidaan esittää muodossa.

$$p(x) = y_0 L_0(x) + y_1 L_1(x) + \ldots + y_n L_n(x),$$

missä L_k :t ovat edellä esitettyjä *Lagrangen polynomeja*

Todistus. Olemassaolopuolen perustelimme edellä, yksikäsitteisyys saadaan taas suoraan seuraus 2:sta. $\square$

Harjoitustehtäviä

Tehtävä 3. Ratkaise johdantoesimerkkinä oleva viskositeettitehtävä Lagrangen menetelmällä. (Huomaa, että interpolaatiopolynomia ei ole tarpeen ”kertoa auki”.)

Tehtävä 4. Osoita, että pisteisiin x_0, x_1, x_2 liittyvät Lagrangen kantapolynomit L_0, L_1, L_2 toteuttavat ehdon $L_0(x) + L_1(x) + L_2(x) = 1$ kaikilla reaaliluvuilla x. Yleistä mieleivaltaiselle n:lle.

Vihje. Voit toki tehdä tapauksen $n = 3$ sieventämällä tai antamalla symboliohjelman (Maple, Mathematica) sieventää. Mutta varsinainen yleiseen tilanteeseen sopiva ahaa-elämys syntyy, kun sovellat vain interpolaatiolausetta (tai pelkästään seurauslausetta 2) sopivasti.

Interpolaatiovirhe

Kuten edellä oli puhe, interpolaatiota voidaan käyttää menetelmänä funktion approksimointiin. Olennainen kysymys on tällöin, miten suuri virhe tehdään.

Jos kyseessä on $n + 1$ kertaa jatkuvasti derivoituvan funktion interpolointi $n + 1$:ssä pisteessä (siis korkeintaan n-asteisella polynomilla), voidaan virheelle johtaa kaunis kaava, joka on muotoa

$$|f(x) - p(x)| \leq M|(x - x_0)(x - x_1)\ldots(x - x_n)|,$$

missä vakio M on $(n + 1)$:sen derivaatan itseisarvon $|f^{(n+1)}|$ maksimi ao. välillä jaettuna luvulla $(n + 1)! = 1 \cdot 2 \cdot \ldots \cdot (n + 1)$.

Emme ryhdy tämän kaavan perustelemiseen, emmekä myöskään esittele sen soveltamista tällä kertaa. Sensijaan otamme tuntumaa siihen, minkälaisia yllätyksiä voi virhekäytöksen suhteen tulla vastaan.

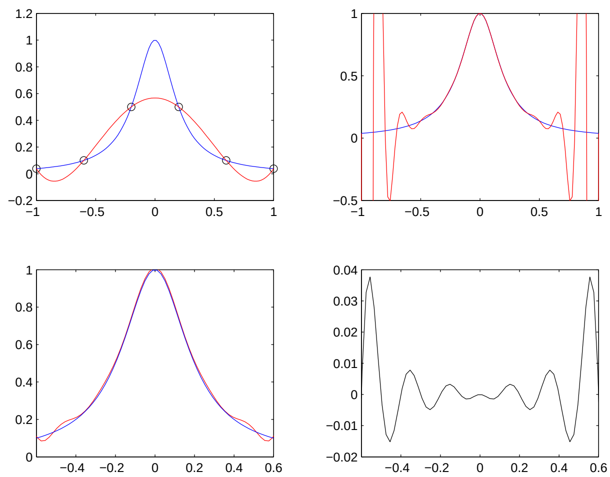
Edellä näimme esimerkkejä ensimmäisen ja toisen asteen interpolaatiopolynomista. Havaitsimme esimerkkiemme yhteydessä, että interpolaatiopisteitä lisäämällä ja siis polynomin astelukua kasvattamalla saimme virheen taulukkopisteiden välillä pienenemään. Niinpä herää luonnollinen kysymys.

Jos interpolaatiopisteitä lisätään rajatta, läheneekö interpolaatiovirhe nollaa?

Virhekaavasta ei ole mahdollista päätellä yleisesti, koska siinä esiintyy $\max|f^{(n+1)}|$. Sehän voi periaatteessa käyttäytyä aika mielivaltaisella tavalla n:n kasvaessa.

Niinpä tulee mieleen vastaesimerkin hakeminen. Sellaisen on ystävällisesti meille tarjoillut matemaatikko *C. Runge* jo v. 1901.

Esimerkki 2 (Rungen koe). Tarkastellaan funktiota $f(x) = \frac{1}{1+25x^2}$ välillä $[-1, 1]$. Katsotaan, mitä tapahtuu, kun suoritetaan tasavälinen polynomi-interpolaatio ja pisteitä (polynomin astelukua) kasvatetaan.



KUVA 5. Rungen funktion $\frac{1}{1+25x^2}$ interpolointia.

Vasemmassa yläkuvassa on $f(x) = \frac{1}{1+25x^2}$ (arvoon 1 saakka kurkottava) ja 5-asteinen interpolaatiopolynomi, oikeassa yläkuvassa on mukana heilumassa 20-asteinen interpolaatiopolynomi.

Vasemmassa alakuvassa on oikea yläkuva rajoitettuna keskemmälle väliä (välille $[-0.6, 0.6]$). Oikeassa alakuvassa on tämän virhekäyrä, eli $f(x) - p_{20}(x)$ samaisella keskiosalla.

Kuvista nähdään, että kun interpolaatiopolynomin aste kasvaa 5:stä 20:een, niin tarkkuus välin keskivaiheilla paranee huomattavasti. Toisaalta välin reunojen läheisyydessä korkempiasteinen polynomi heilahtelee aivan villisti, ja kuvan perusteella tuntuu varsin uskottavalta se *Rungen* todistama tosiasia, että koko välillä laskettu maksimivirhe lähenee jopa ääretöntä, kun n kasvaa rajatta.

Tämä käytös antaa aiheen arvella, että tasavälinen pisteistö ei olekaan hyvä, vaan kannattaisi valita pisteet niin, että ne ovat keskellä harvassa ja reunoille mentäessä tihenevät. Samainen *Runge* havaitsi, että valitsemalla pisteet ns. *Tsebyshev-polynomien* nollakohdiksi, saadaan virhe suppenemaan kohti nollaa. Kyseinen pisteistö tosiaankin kasaantuu kohti välin reunoja, ja se on tämän tehtävän kannalta optimaalinen.

Kuvat on tehty MATLAB-skriptillä `runge.m`. Sitä muokkamalla voit kehitellä omia kokeilujasi.

Polynomeista palapolynomeihin

Viittasimme kirjoituksen alkupuolella ongelmaan, joka liittyy laajan taulukon interpolointiin. Siinä esiintyvää problematiikkaa havainnollistimme *Rungen kokeilla*. Logaritmitaulukon yhteydessä emme suinkaan pyrkineet polynomiin, jonka kuvaaja kulkee kaikkien taulukkopisteiden kautta, vaan suoritimme *paloittain lineaarisen* interpolaation. Yleisemmin voisimme pyrkiä käsittelemään interpolaatiotehtävää sopivissa palasissa, jotka koostuisivat korkeamman kuin 1. asteen polynomeista ja jotka liitoskohdissa säädettäisiin mahdollisimman sileiksi, ts. järjestettäisiin kertoimet niin, että saadaan mahdollisimman monta jatkuvaa derivaattaa liitoskohdissa.

Tämän periaatteen mukaista palapolynomia on ruvettu kutsumaan englanniksi nimellä ”spline”, jonka suomenkielinen käännös on yksinkertaisesti ”splini”. Alunperin tällä tarkoitettiin kolmannen asteen polynomipalasista koostuvaa funktiota, jolla liitoskohdissa on jatkuvat derivaatat toiseen kertalukuun saakka. Tällaisen liitoksen silmä näkee täysin sileänä. Nykyisin spliniksi katsotaan edellä kuvattu yleinen tapaus, mutta kaikkein eniten käytetty lienee juuri tämä alkuperäinen kuutiollinen splini.

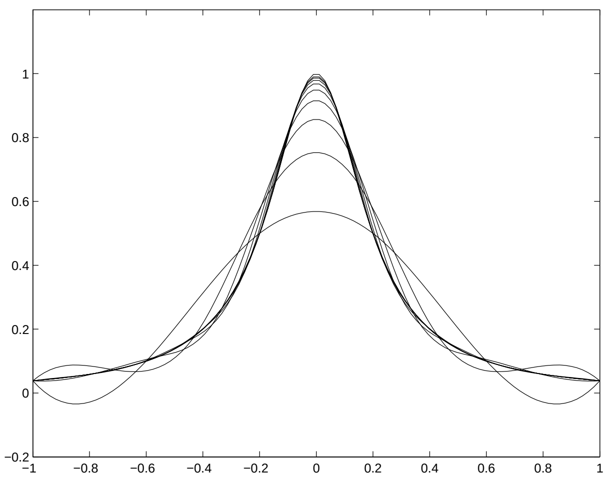
Samalla nähdään yleinen numeerisissa algoritmeissa esiintyvä tilanne. Menetelmän tarkkuuden parantamiseen on kaksi tapaa:

- Nostetaan menetelmän ”kertalukua”, jolla interpolaation tapauksessa tarkoitetaan polynomin astelukua.
- Jaetaan kyseessä oleva väli, alue tms. pienempiin osiin ja lasketaan approksimaatio kullakin osalla erikseen ja yhdistetään osat kokonaisuudeksi.

Yleensä numeerisissa algoritmeissa käytetään näiden menettelyjen sopivaa yhdistelmää.

Jos tehtävänä olisi muodostaa annettuun dataan liittyvä kuutiollinen splini, niin edellä sanotun, splinin määrittelevän periaatteen mukaan olisi suoraviivaista kirjoittaa yhtälöt. Nähtäisiin, että määrättäviä kertoimia on kaksi enemmän kuin ehtoja. Näin jäisi vapaasti valittavaksi kaksi ”reunaehtoa”, jonka jälkeen tehtävällä olisi yksikäsitteinen ratkaisu.

Koska en halua laajentaa kirjoitusta liikaa, en mene yksityiskohtiin lähemmin. Sensijaan turvaudun valmiiseen ”mustaan laatikkoon”, Matlab-funktioon `spline`. Sen käyttöesimerkki on MATLAB-skriptissämme `rungesplini.m`, jolla ao. kuva on piirretty.



KUVA 6. Funktion $f(x) = \frac{1}{1+25x^2}$ interpolointi kuutiollisilla splineillä, jakamalla väli n:ään osaan, $n = 5, 7, 9, \ldots, 21$.

Kuvasta nähdään, että heilahtelut pienenevät n:n kasvaessa ja paksu käyrä ilmentää funktiojonon ”tasaista” suppenemista koko välillä kohti alkuperäistä Rungen funktiota f.

Loppumietteitä, yhteenvetoa

Interpolaatio on hyvä lähtökohta funktion approksimointiin, sillä on keskeinen merkitys numeerisen analyysin useiden eri alueiden algoritmien ytimenä. Mainittakoon vaikka numeerinen derivointi ja integrointi, differentiaaliyhtälöiden ratkaisumenetelmät, funktion nollakohtien määrittäminen, jne.

Polynomi-interpolaatiosta johduimme splineihin, jotka ovat käyttökelpoisia monenlaisissa isojen taulukoiden interpoloinneissa. Niitä käytetään paljon myös mm. tietokonegrafiikan alalla.

Viitteet

1. Burden-Faires. *Numerical Analysis*, 7^{th} ed., Brooks/Cole, 2001.
2. Cheney-Kincaid. *Numerical Mathematics and Computing*, Brooks/Cole, 2004.
3. Forsythe-Malcolm-Moler. *Computer Methods, for Mathematical Computation*, Prentice Hall, 1977.
4. Kahaner-Moler-Nash. *Numerical Methods and Software*. Prentice Hall, 1989.
5. Moler. *Numerical Computing with Matlab*, 2004, saatavissa verkosta: `www.mathworks.com/moler/`