

**Mat-1.414 Matematiikan peruskurssi V2 kevät 2001**

<http://www.math.hut.fi/teaching/v/3/H/>

**Laskuharjoitus 3** (viikko 6 , 7 – 9.2.)

**Alkuviikko (AV), ke 7.2.**

1. Muodosta seuraaviin datoihin liittyvät interpolaatiopolynomit:

(a)	x	7	1	2	(b)	x	3	7	1	2
	y	146	2	1		y	10	146	2	1

  

(c)	x	3	7	1	2
	y	12	146	2	1

Laske (a)-kohta Lagrangen menetelmällä sekä (b) ja (c)-kohdat Newtonin menetelmällä. Huomaa, että aloitettuasi Lagrangella, voit aivan hyvin jatkaa Newtonilla (vaikka missään kirjassa ei koskaan näin menetellä, hm ...). Ovatko kaikkien polynomien asteluvut sellaisia, kuin odotit?

Huomaa myös: Newtonilla on menetelmiä vähän joka alueella, tällä ei ole mitään tekemistä yhtälön ratkaisumenetelmän kanssa.)

2. Olkoot  $L_0, L_1, \dots, L_n$  (erillisiin) pisteisiin  $x_0, x_1, \dots, x_n$  liittyvät *Lagrangen kertojapolynomit*. Osoita, että

$$\sum_{i=0}^n L_i(x) = 1 \quad \forall x.$$

Vihje: Mieti polynomien asteluvun ja nollakohtien lukumäärän välistä suhdetta.

3. Seuraavassa on taulukoitu funktion  $f(x) = \log_{10}(\tan x)$  arvoja:

x	1.00	1.05	1.10	1.15
f(x)	0.1924	0.2414	0.2933	0.3492

Laske tämän taulukon avulla likiarvo  $f(1.09)$ :lle ja esitä virheelle yläraja-arvio interpolaatiovirhekaavaa käyttäen. Vertaa todelliseen virheeseen.

4. Annettuna on seuraava data:

```
xd:= [seq(1.0+k*0.1, k=0..4)];
yd:= [1.684370, 1.949477, 2.199796, 2.439189, 2.670324]
```

Rakenna tähän liittyvä vapaa kuutiollinen splini.

Y-arvot ovat itse asiassa funktion

$f(x) = \ln(e^{2x} - 2)$  arvoja **xd**-pisteissä. Ne on muuten saatu aikaan näin: `yd:=map(x->ln(exp(2*x)-2), xd);`

Approksimoi arvoja  $f(4.1)$  ja  $f'(4.1)$  splinin ja sen derivaatan avulla ja vertaa.

Laske vielä ihan huviksesi “lukitun” (“clamped”) splinin tapaus reunaehdoilla  $f'(1.0) = 2.742245$  ja  $f'(1.4) = 2.276919$  ja samat asiat sen avulla.

Vihje: Splinin laskemiseksi kannattaa ottaa `../L/interpolaatio.mws`-työarkki ja modifioida siinä olevaa “nautiskeluesimerkkiä”. Tietysti voit laskea käsinkin, mutta nautintoelämys voi jäädä vaisummaksi.

Kaikissa splinitehtävissä voit myös kokeilla valmista **spline**-funktiota, mutta tässä ainakin vain tarkistukseksi.

5. Muodosta vapaa kuutiollinen splini, joka approksimoi funktiota  $f(x) = \cos \pi x$ , käyttäen solmuja `xd:= [0, 0.25, 0.5, 0.75, 1.0]`. Integroi splini ja vertaa arvoon  $\int_0^1 \cos \pi x (= 0)$ . Vertaa edelleen splinin derivaatta-arvoja vastaaviin oikeisiin:  $f'(0.5), f''(0.5)$ .

## Torstai 8.2

1. Muodosta interpolaatiopolynomi pisteistölle, joka saadaan laske-  
malla funktion  $\cos(1+x^2)$  arvot tasavälisessä x-pisteistössä, jossa  
on 7 pistettä välillä  $[0, 3]$ . Piirrä samaan kuvaan funktio, datapisteet  
ja interpolaatiopolynomi.

Arvioi Lagrangen interpolaatiokaavan virhetermin avulla interpolaatiovirheen yläraja yo. välillä ja vertaa todelliseen.

2. Annettu data on näyte lentävän ankkalinnun selkäprofiilista. Näyt-  
teitä on otettu tiheämmin alueella, jossa profiili muuttuu voimak-  
kaammin, kuten kuvasta näkyy.

```
xda:=0.9,1.3,1.9,2.1,2.6,3.0,3.9,4.4,4.7,5.0,6.0,7.0:  
xdb:=8.0,9.2,10.5,11.3,12.0,12.6,13.0,13.3;  
yda:=1.3,1.5,1.85,2.1,2.6,2.7,2.4,2.15,2.05,2.1,2.25:  
ydb:=2.3,2.25,1.95,1.4,0.9,0.7,0.6,0.5,0.4,0.25  
xd:=xda,xdb: yd:=yda,ydb:
```

Muodosta (a) interpolaatiopolynomi ja (b) (vapaa) kuutiollinen  
splini dataan liittyen. Jos et arvaa, kumpi antaa paremman kuvan,  
olet tosi huono arvaamaan. Laske ja piirrä joka tapauksessa. Mikäli  
Maple osoittautuu liian hitaaksi, annetaan ohjeet Matlab-funktioiden  
käyttöön, katsotaan!

KUVAA EI TÄSSÄ PDF-versiossa ole (se ei haittaa tehtävän te-  
koa).

3. Ohjetiedostossa ../H/ohjeita3.html on annettu USA:n väestölasken-  
tadataa vuosina 1940 – 1990. Muodosta interpolaatiopolynomi ja  
laske sen avulla arviot väkiluvulle vuosina 1930, 1965, 2000, 2010.  
Havainnollista kuvilla, joissa datapisteet ja interpolaatiopolynomi.  
Tiedetään, että vuoden 1930 lukema on 123203000. Mitä arvelet  
muiden arvojen tarkkuudesta? (Etsi jostain (Alta Vista) Census  
2000- arvo ja vertaa.)

Tee sama splini-sovituksella (ota vapaat reunaehdot).

4. Muistammehan *Rungen ilmiön*. Selvittelepä syitä katsomalla inter-  
polaatiovirheen kaavaa. Tarkoituksena tässä ei ole niinkään tutkia,  
miksi juuri Rungen funktio on ongelmallinen, vaan mitä tapahtuu  
sen virhekaavan tulotermin takia.

Valitse jokin väli  $[a, b]$ , jota voit vaihdellakin. (Esim  $[-1, 1]$  tai  
 $[-5, 5]$ .)

(a) Tutki tulotermiä  $x$ :n funktiona eri  $n$ :n arvoilla kun väli jaetaan  
tasavälisesti  $n$ :ään osaan. Missä välin osassa termi on suurimmillaan  
ja missä taas pienimmillään? Entä välin ulkopuolella (extrapolaa-  
tio)?

(b) Tee sama, kun käytetään *Tsebysev-pisteitä*

$$x_k = \frac{a+b}{2} + \frac{a-b}{2} \cos \frac{2k+1}{2n+2} \pi, \quad i = 0, \dots, n$$

Vertaa (a)- ja (b)-kohtaa keskenään.

5. **Käänteinen interpolaatio** (KRE s. 859, “Inverse interpolation”).  
Oletetaan, että derivoituvalla funktiolla  $f$  on yksi 0-kohta välillä  
 $[a, b]$  ja  $f'(x) \neq 0$  koko välillä. Funktioiden nollakohta  $s = f^{-1}(0)$ .  
Eräs idea on approksimoida käänteisfunktioita interpoloimalla. Tä-  
tä varten valitaan pisteet  $x_1, \dots, x_n$  väliltä  $[a, b]$ , lasketaan  $y_k =$   
 $f(x_k), k = 1 \dots n$  ja muodostetaan interpolaatiopolynomi  $p(y)$  niin,  
että vaihdetaan x- ja y-pisteiden roolit. Nollakohdan approksima-  
tio on nyt siis  $p(0)$ . Tietysti käänteisfunktion arvoja voidaan yhtä  
hyvin approksimoida muissakin pisteissä.

Laske tähän perustuen approksimaatio  $f(x) = e^{-x} - x$ :n 0-kohdalle  
välillä  $[0, 1]$  tähän tapaan: Laske taulukko  $[x_k, f(x_k)], k = 0 \dots n$   
(voit ottaa  $n$ :n vaihdeltavaksi parametriksi).

Muodosta interpolaatiopolynomi vaihtamalla  $x$ :n ja  $y$ :n roolit. Sit-  
tenpä vaan lasket sen arvon 0:ssa.

Tästä voisi kehitellä iteratiivisen algoritmin. Itse asiassa käänteistä  
interpolaatiota käytetään monissa yhtälönratkaisukoodeissa osana  
koko ratkaisualgoritmia.

6. Tasokäyrä, jolta tunnetaan vain diskreetti pistejoukko, voidaan yrit-  
tää rekonstruoida splinisovituksella. Tätä varten esitetään pisteet

parametrimuodossa. Yksinkertaisin tapa on parametrisoida ne kokonaisluvulla  $t = 1, 2, \dots, n$ , mikä on ihan toimiva, kunhan datapisteet sijoitetaan siten, että niiden välisten osakaarien pituudet ovat samaa suuruusluokkaa. (Jos halutaan muuta, niin voidaanhan se ilmaista epätasavälisillä parametrinumeroilla.)

Sitten vain konstruoidaan koordinaattifunktioita  $x(t)$  ja  $y(t)$  interpoloivat splinit.

Piirrä ruutupaperille jokin kirjain, numero tms. hauska häkkyrä ja valitse siitä n. 8 pistettä. Interpoloi splineillä kumpaakin koordinaattifunktiota ja piirrä.

Lähtökuviosi tulisi mieluummin olla jotensakin sileä, kirjaimia voit tyylitellä, saat panna kaiken taiteellisen luovuutesi peliin. Mieti myös reunaehdoja. (Tietysti voit kokeilla sitäkin, mitä sileä approksimaatio tekee kulmikkaalle)

Sopivia kirjaimia tai numeroita ovat esim.

$\mathcal{C}, \mathcal{E}, \mathcal{G}, \mathcal{S}, \mathcal{O}, \mathcal{P} \quad \alpha, \beta, \gamma, \varphi, \xi, \zeta, 2, 6, 8, 9$ .

Sellaista ei hyväksytä, joka voidaan esittää muodossa  $y = f(x)$ . Joissakin tapauksissa voi olla tarpeen tehdä parissa osassa, mutta osat tulee tehdä parametrikäyrinä.

Maple-ohje parametrimuotoiseen piirtoon:

```
plot([x(t),y(t),t=a..b]);
```

Siis ainoa ero kahden kuvaajan piirtoon verrattuna on piirtorajojen  $t=a..b$  tuominen listasulkujen (hakasulut) sisään. Kokeile ja vertaa varmuuden vuoksi:

```
plot([cos(t),sin(t),t=0..2*Pi]);  
plot([cos(t),sin(t)],t=0..2*Pi);  
complexplot(exp(I*t),t=0..2*Pi);    # Älä tätäkään unohda!
```