

Mat-1.192 Numeerinen ja symbolinen laskenta kevät 2005

<http://www.math.hut.fi/teaching/numsym/04/H/>

Laskuharjoitus 5 (viikko 10–11 , 17.3.2004)

Versio 1 ti 8.3.

<http://www.math.hut.fi/teaching/numsym/05/maple/ns05.mpl>

<http://www.math.hut.fi/teaching/numsym/05/L/>

1. (Cooper Exe 3.2 ss. 80–81) (a) Osoita, että seuraavat funktiot toteuttavat lämpöyhtälön $u_t = ku_{xx}$.

(b) Määritä jomman kumman funktion (1) tai (2) maksimi ja minimi suorakulmiossa $[-a, a] \times [0, T]$. Totea, että maksimiperiaate on voimassa ja visualisoi.

(1) $v(x, t) = e^{-\gamma^2 kt} \sin \gamma x$, $\gamma \in \mathbb{R}$ mielivalt.

(2) $w(x, t) = e^{kt+x}$.

Voit suorittaa min/max-määrittelyn joko Maple-avusteisesti ”klassiseen tyyliin”tai numeerisesti (luontevimmin Matlabilla), jolloin täytyy valita ao. vakioille jotkin numeroarvot. (Toki Maplessakin jossain vaiheessa, ainakin plot:ssa)

2. Maple-Matlab-yhteiskäyttöä: Tutustu `maple/ns05.mpl`-funktioon `sinkert` ja vastaavaan Matlab-funktioon `matlab/Msinsarja.m`. Kehitä jokin funktio sinisarjaksi ja kokeile erilaisia käyttötapoja. Kokeile ainakin sitä, jossa kertoimet kirjoitetaan Maplella ja luetaan Matlabiin.
3. (T-W Exa 4.2 s. 121) Tutustu ns05-funktioon `lamposin` ja samannimiseen Matlab-funktioon `matlab/lamposin.m`. Ratkaise ja visualisoi: $u_t = u_{xx}$, $u(x, 0) = f(x)$, 0– RE:t.

$$f(x) = \begin{cases} 2x, & x \leq 1/2 \\ 2(1-x), & x > 1/2 \end{cases}$$

Kokeile taas Maple-Matlab-yhteispeliä, mutta myös puhdasta Maple-loppukäsittelyä. (Toki mielellään myös (lähes) puhdasta Matlab-työtä.)

4. Tutustu ns05-funktioon `lampodiffE`. Laske sen avulla differenssiapproksimaatioita edellisen tehtävän ratkaisulle. Suorita [T-W] Exa 4:n tyyliin. (Voit verrata kirjan kuviin ja tuloksiin.)
5. Kirjoita implisiittinen differenssimenetelmä Matlab-funktioksi. Ota Dirichlet’n reunaehdot, mutta anna niiden olla mielivaltaiset, annettujen t:n funktioiden määräämät. Testaa.
6. (a) Cooper-funktiot `heat3–heat5`, tutustu, parantele, ratkaise Cooper-tehtäviä mm. Selvitä ainakin koodin `heat3` toiminta itsellesi.
(b) Muodosta numeerinen ratkaisu tehtävälle, $u_t = u_{xx}$ (siis $k = 1$), $f(x) = \frac{x}{2} - 2 + \sin(\frac{4\pi x}{10})$, $u(0, t) = -2$, $u(10, t) = 3$
(c) Suorita vertailua puhtaan implisiittisen ja Crank-Nicholsonin välillä (teht. 4 ja 5).
(c) Aja ohjelmia niin suureen t:n arvoon saakka, että ratkaisu asettuu ajasta riippumattomaan (”steady-state”) tilaan funktioksi $U(x)$ (sopivalla tarkkuudella).

Neuvo: Kannattaa ehkä aloittaa tästä tehtävästä vaikka ihan alkuperäistä `heat3`-ohjelmaa käyttäen, tai sitten tehtävän 5 implisiittistä, etenkin jos tuntuu, että menetelmien ohjelmointia on jo tullut liikaa.

Ohjeita

Teht. 1

Maple-malli visualisointiin ym. : `L/L6.mws` (ei sisällä max/min-laskentaa). Matlab:ssa voit vaikka määrittellä inline-funktioksi, joiden oltava x:n ja t:n suhteen ”array smart”(vrt. aiemmin jaettu Cooperin kirjan Matlab-appendix.). Sitten `meshgrid`, ...

Kts. myös <http://www.math.hut.fi/teaching/numsym/> ”Luennot-kohdan ohjeita Luento 6:n kohdalta.

Teht. 2

Tässä on funktion `sinkert` kutsu ja alkukommentit. Koodia täytyy varautua muuttamaan ainakin tiedostopolun suhteen (Unix:ssa tuo koodissa oleva lie-nee aina hyvä.) Voidaan käyttää pelkkänä Maple-funktiona tai Maple-Matlab-yhdistelmänä.

```

siniSkertoimet:=proc(f,L,N,tiedosto)
# b:=siniSkertoimet(f,L,N,tiedosto); Funktion f sinisarjan kertoimet.
# b:=siniSkertoimet(f,L,N); palauttaa b-kertoimet, mutta ei kirjoita
# tiedostoon.
# f -- funktio (esim. x->x^2) Oltava oikea Maple-funktio, ei lauseke.
# L -- jakson puolikas
# N -- Mihin saakka kertoimia lasketaan
# tiedoston nimi merkkijononona (ilman polkua), esim. "bkert.txt"
# Kertoimet palautetaan liukulukutaulukkona (array) ja kirjoitetaan tiedostoon
# josta ne voi suoraan lukea Matlabiin load-komennolla:
# Lb=load('/tmp/bkert.txt')
# Kirjoitetaan vektori [L,b1,b2,...,bN].
# Työtä voi jatkaa joko Maplessa tai Matlabissa.

local B,n,t,omega,polkujatiedosto;
B:=array(1..N);
omega:=Pi/L;
for n from 1 to N do
    B[n]:=evalf(2/L*int(f(t)*sin(n*omega*t),t=0..L));
end do;
if nargs=3 then RETURN(B) end if;

#polkujatiedosto:=cat("C:/MATLAB6p1/work/",tiedosto); # HA:n kannettava
# polkujatiedosto:=cat("/tmp/",tiedosto); # Unix

writedata(polkujatiedosto,[L,seq(B[j],j=1..N)]); # Kirjoitetan myös L.
print("Minä, siniSkertoimet kirjoitin [L,b]:n tiedostoon ",polkujatiedosto);
#print("Palautan myös b-kertoimien taulukon (array) tähän sinulle");
RETURN(B)
end proc:

% Tiedostosta luetaan L ja b
% 2) [summa,osasummat] = Msinsarja(x,L,b)
% Kerroinvektori annetaan Matlabissa.
% Koko suoritus:
% 1) Maplessa: ns05.mpl-funktio siniSkertoimet(f,L,N,tiedosto);
% Esim: B:=siniSkertoimet(x->x^2,1,10,"bkert");
% 2) Matlab:ssa:
% x=linspace(-2,2);
% [summa,osasummat]=Msinsarja(x,'bkert');
% plot(x,summa); figure; plot(x,osasummat)
polku='/tmp/'; % UNIX % Muuta tarpeen mukaan.
%polku='C:\MATLAB6p1\work\'; % WIN % Muuta tarpeen mukaan.
if nargs==2
    tiedosto=[polku LtaiFile];
    Lb=load(tiedosto);
    L=Lb(1); % ensimmäisenä on jakso/2.
    b=Lb(2:end); % Loput muodostavat b-vektorin.
else
    L=LtaiFile;
end;
N=length(b);
kertoimet=b*ones(size(x)); % Eräs monista tavoista tehdä matriisi, jonka
% 1. rivi on vakio a0
% 2. rivi on vakio a1,
% ...
termit=kertoimet.*sin(pi/L*(1:N)'*x);
summa=sum(termit);
osasummat=cumsum(termit);

```

Vastaava Matlab-funktio on
www.math.hut.fi/numsym/05/matlab/Msinsarja. Voidaan siis käyttää pelkänä Matlab-funktiona tai Maple-Matlab-yhdistelmänä.

```

function [summa,osasummat] = Msinsarja(x,LtaiFile,b)
% matlab/Msinsarja.m
% Kutsut:
% 1) [summa,osasummat] = Msinsarja(x,tiedosto)

```

Teht. 3

```

lamposin:=proc(f,L,c,N,tiedosto)
description "Lämpöyhtälö u_t = c^2 u_xx, 0-RE't, u(x,0)=f(x)";
local b,lambda,x,t,n;
# Palautetaan osasummafunktio, jossa termit 1..N.
# Esim:
# S1:=lamposin(x->x,1,1,10):
# S2:=lamposin(x->x,1,1,20,"b"):
# plot3d(S1(x,t),x=0..L,t=0..0.1) # sama S2:lle.

```

```

if nargin = 4 then
    b:=siniSkertoimet(f,L,N);
else
    b:=siniSkertoimet(f,L,N,tiedosto);
end if;
lambda:=n->c*n*Pi/L;
RETURN(evalf@unapply(sum(b[n]*sin(n*Pi*x/L)*exp(-lambda(n)^2*t),n=1..N),x,t));
end proc:

```

Teht. 4

```

lampodiffE:=proc(f,nx,nt,dt)
    description"Lämpöyht, väli: [0,1], lämpövakio =1 (vrt. [T-W])";
    # f -- alkuarvofkt.
    # nx,nt,dt ilmeisiä
    # Tulos: Listojen lista, m:s lista edustaa ao. aikatasoa.
    # Tulostetaan lisäksi r=dt/dx^2. sen pitää olla alle 1/2, jotta stabiili.

```

Teht. 6

Koodissa heat3.m on ehkä selkeämpää ja tehokkaampaa korvata A:n rakentaminen ennestään käytetyllä tyyllillä:

```

e=ones(J-1,1);
A=spdiags([-r*e (1+2*r)*e -r*e],-1:1,J-1,J-1);

```

Sinänsä on opettavaista tutustua sparse komennon moninaiisiin muotoihin.