L 14–15, alk. pe 1.12.06

1 Differentiaaliyhtälöiden numeerisia menetelmiä

Pääasiassa keskitytään systeemeihin, ainakin periaatteessa yhden yhtälön tapaus on ollut esillä jo peruskursseissa 1 tai 2. Mutta aluksi kuitenkin:

Yksi skalaariyhtälö

Alkuarvotehtävä:

$$y' = f(t, y), \quad y(t_0) = y_0$$

Piirretään alkuarvopisteeseen lyhyt suuntakenttänuoli, jonka suunta on ratkaisukäyrän tangentin suunta: $y'(t_0) = f(t_0, y_0)$. Suuntanuolen loppupäässä tehdään sama uudestaan, aikapisteessä $t_1 = t_0 + h$, näin jatketaan.

Jos loppupisteessä t_1 olevaa tangenttisuoran $y-$arvoa merkitään y_1 :llä, on siis

$$y_1 = y_0 + hy'(t_0) = y_0 + h\, f(t_0, y_0).$$

Tässä meillä on *Eulerin menetelmän* askel.

Jotta saataisiin myös arvio virheelle, joka tehdään, kun ratkaisukäyrä korvataan tangentillaan, muodostetaan *Taylorin* kehitelmä pisteessä t: Jos $y(t)$ on ratkaisufunktio,

$$y(t + h) = y(t) + h\, y'(t) + O(h^2) = y(t) + h\, f(t, y(t)) + O(h^2).$$

Tässä jäännöstermi on $\frac{y''(\xi)}{2} h^2$, joka on muotoa $O(h^2)$.

Jos pudotetaan jäännöstermi, saadaan siten approksimaatio ratkaisufunktiolle pisteessä $t + h$, kun tunnetaan ratkaisu(approksimaatio) pisteessä t.

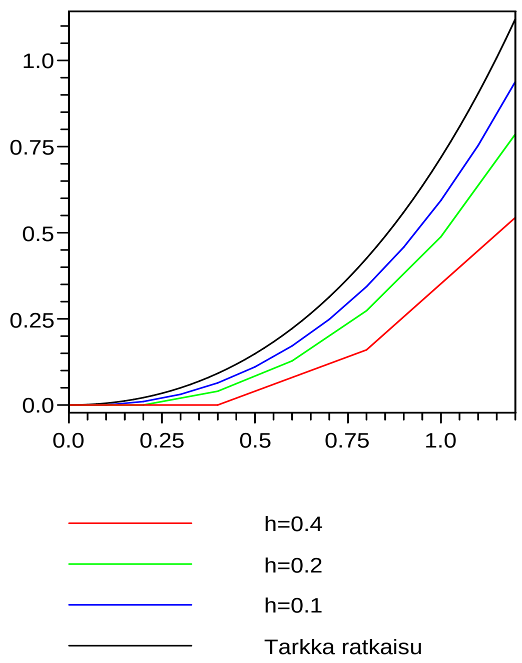
Eulerin menetelmä:

Annettu alkupiste: $(t_0, \mathbf{y}_0)$. Lasketaan

$$y_{k+1} = y_k + hf(t_k, y_k), k = 0, \ldots, n$$

Esim. $y' = t + y, \quad y(0) = 0.$

Tarkka ratkaisu: $y(t) = e^t - t - 1$.



1) $h = 0.4$

$$\begin{bmatrix} 0 & 0.4 & 0.8 & 1.2 \\ 0 & 0.0 & 0.16 & 0.544 \end{bmatrix}$$

2) $h = 0.2$

$$\begin{bmatrix} 0 & 0.20 & 0.40 & 0.60 & 0.80 & 1.0 & 1.20 \\ 0 & 0.0 & 0.040 & 0.1280 & 0.2736 & 0.48832 & 0.78598 \end{bmatrix}$$

Yksinkertainen Matlab-ajo

```
clear; close all

% a) h=0.4
h=0.4; n=3;
ta=0:h:n*h;
ya(1)=0;   % Indeksointi alkaa 1:stä Matlabissa.
for k=1:n
    ya(k+1)=ya(k)+h*(ta(k)+ya(k));
end;
[ta;ya]
plot(ta,ya,'r');hold on;plot(ta,ya,'ro')
grid on;

%b)  h=0.2

h=0.2; n=6;
tb=0:h:n*h;
yb(1)=0;
for k=1:n
    yb(k+1)=yb(k)+h*(tb(k)+yb(k));
end;
[tb;yb]
plot(tb,yb,'g');plot(tb,yb,'go')

%c) h=0.1

h=0.1; n=12;
tc=0:h:n*h;
yc(1)=0;
for k=1:n
    yc(k+1)=yc(k)+h*(tc(k)+yc(k));
end;
[tc;yc]
plot(tc,yc,'b');plot(tc,yc,'bx')
```

Näytetään c)-kohdan tulos, edellä oli jo a)- ja b)-tulokset.

```
>> [tc;yc]
```

```
ans =

  Columns 1 through 7

         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
         0         0    0.0100    0.0310    0.0641    0.1105    0.1716

  Columns 8 through 13

    0.7000    0.8000    0.9000    1.0000    1.1000    1.2000
    0.2487    0.3436    0.4579    0.5937    0.7531    0.9384

loppuarvot=[ya(end),yb(end),yc(end)]
tL=ta(end)
yloppu=exp(tL)-tL-1
virheet=yloppu-[ya(end),yb(end),yc(end)]
%%%%%%%%%%% ajo ja tulokset %%%%%%%%%
loppuarvot =
    0.5440    0.7860    0.9384
>> tL=ta(end)
tL =
    1.2000
>> yloppu=exp(tL)-tL-1
yloppu =
    1.1201
>> virheet=yloppu-[ya(end),yb(end),yc(end)]
virheet =
    0.5761    0.3341    0.1817
```

Tämä on skriptinä: `L/Euleresim1.m`

Virheistä

Katkaisuvirheet (menetelmävirheet) ja pyöristysvirheet.

Käsittelemme tässä vain edellisiä.

Lokaali katkaisuvirhe: Yhdellä askeleella syntyvä virhe.

Globaali katkaisuvirhe: Koko tarkasteluvälillä syntyvä virhe, lokaalien virheiden yhteisvaikutus.

Eulerin menetelmässä Taylorin kaava antaa **lokaalin virheen**: $O(h^2)$.

Globaali virhe: Jos välin pituus on L ja askel h, niin askelten lukmäärä $n = L/h \sim 1/h$. Siten n :llä askeleella syntyvä virhe on muotoa $\frac{1}{h}\,O(h^2) = O(h)$. (Tämä on hiukan ylimalkainen argumentti, koska virheiden summa ei ilman muuta ole sama kuin kokonaisvirhe, joka tapauksessa johtopäätös on oikea.)

Kun askel puolittuu, niin kokonaisvirhe puolittuu

Tämänsuuntainen käytös on hyvin nähtävissä esimerkeissä, tätä kannattaa tarkkailla.

Esimerkki 1.1 *Oikein kunnolla epälineaarinen ja epäautonominen:*

$$y' = \sin(t\,y), \quad y(0) = 3.$$

Tässä siis $f(t, y) = \sin(t\,y)$. *Olkoon* $h = 0.1$. *Järjestetään laskut taulukon muotoon:*

t_n	y_n	$y_{n+1} = y_n + h\,f(t_n, y_n)$
$t_0 = 0.0$	$y_0 = 3.000$	$y_1 = 3 + 0.1\ \sin(0 \cdot 3) = 3.000$
$t_1 = 0.1$	$y_1 = 3.000$	$y_2 = 3 + 0.1\ \sin(0.1 \cdot 3) = 3.030$
$t_2 = 0.2$	$y_2 = 3.030$	$y_3 = 3.030 + 0.1\ \sin(0.2 \cdot 3.03) = 3.087$
$t_3 = 0.3$	$y_3 = 3.087$	$y_4 = 3.087 + 0.1\ \sin(0.3 \cdot 3.087) = 3.167$
$t_4 = 0.4$	$y_4 = 3.167$	$\ldots$

Esimerkki voitaisiin tehdä Matlab:lla/Octavella vaikka näin:

```
clear; close all
h=0.1; n=10;
t=0:h:1;
y(1)=3;   % Indeksointi alkaa 1:stä Matlabissa.
for k=1:n
    y(k+1)=y(k)+h*sin(t(k)*y(k));
end;
[t;y]
```

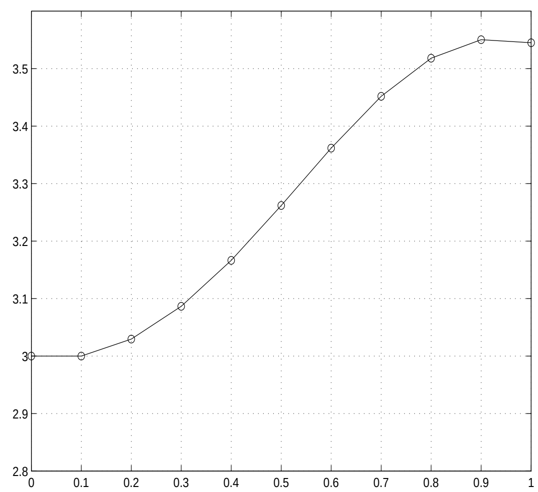
Näytetään tuloksesta 7 ensimmäistä arvoa

```
ans =
  Columns 1 through 7
        0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
   3.0000    3.0000    3.0296    3.0865    3.1664    3.2618    3.3616
```

Tulosten pienet eroavuudet johtuvat siitä, että "käsinlasku"on tehty 4 :n numeron tarkkuudella, kun taas Matlab, laskee n. 16 :lla numerolla, joista näytetään oletusarvona 5 ja systeemikomennon `>> format long` jälkeen kaikki 16.

Voidaan tietysti jatkaa piirtäen:

```
plot(t,y); hold on; plot(t,y,'o'); ylim([2.8 3.6])
```



1.1 Diffyhtälösysteemit

Kuten todettu, tulkitaan vain $\mathbf{y}$ ja $\mathbf{f}$ yllä vektoriarvoisiksi.

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\mathbf{f}(t_k, \mathbf{y}_k), k = 0, \ldots, n$$

Otetaan aluksi esimerkki lineaarisesta systeemistä.

Esimerkki 1.2 *Lineaarinen diffyhtälösysteemi.*

$$\mathbf{y}' = A\,y, \quad \mathbf{y}(0) = [1, 0].$$

Tässä siis $\mathbf{f}(t, \mathbf{y}) = A\,\mathbf{y}$ *(systeemi on autonominen, joten* t *ei oikeasti esiinny oikealla.) Siten iteraatioaskel on*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\,\mathbf{f}(t_n, \mathbf{y}_n) = \mathbf{y}_n + h\,A\,\mathbf{y}_n = (I + hA)\mathbf{y}_n.$$

Toisin sanoen iteraatio on yksinkertaisesti matriisilla $B = I + hA$ *kertomista. Matlab:lla vaikka tähän tapaan:*

```
clear
A=[1 -4;-1 1];
h=0.2; n=5;                % Askelpituus ja askelten lukumäärä.
B=eye(size(A))+h*A;        % Iteraatiomatriisi.

Y=zeros(2,n);    % Alustetaan 2-rivinen matriisi, jonka sarakkeiksi sijoitetaan
                 % Eulerin iteraation tuottamat pisteet. (Alustus ei ole välttämätön,
                 % mutta lienee selkeyttävä, lisäksi parantaa tehokkuutta.)
Y(:,1)=[1;0];    % Alkuarvo sijoitetaan 1. sarakkeeksi. (Matlabin indeksointi alkaa 1:stä

for k=1:n
    Y(:,k+1)=B*Y(:,k);   % Iteraatioaskel
end;
Y               % Katsotaan, mitä Y-matriisin sarakkeissa on.
Y =
    1.0000    1.2000    1.6000    2.3040    3.4816    5.4067
         0   -0.2000   -0.4800   -0.8960   -1.5360   -2.5395
```

Sarake 1: alkuarvo, sarake 2: ensimmäinen iteraatio, ...

Seuraavaksi katsotaan tuttua, esimerkkiä reippaasti epälineaarisesta tapauksesta, nimittäin heiluria.

Esimerkki 1.3 *Heiluriyhtälö* $y'' = -k\sin(y)$ *muunnetaan 1. kl:n systeemiksi:*

$$\begin{cases} y_1' = y_2 \\ y_2' = -k\,\sin y_1 \end{cases}$$

Siis $\mathbf{y}' = \mathbf{f}(\mathbf{y})$, *missä* $\mathbf{f}(\mathbf{y}) = \begin{bmatrix} y_2 \\ -k\,\sin y_1 \end{bmatrix}$. *Lasketaanpa vaikkapa alkupisteestä* $\mathbf{y}^0 = (0, 1)$ *lähtien muutama askel ja olkoon* $h = 0.1$. *Valitaan vielä vaikkapa* $k = 1$.

$$\mathbf{y}^0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{f}(y^0) = \begin{bmatrix} y_2^0 \\ -\sin\left(y_1^0\right) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

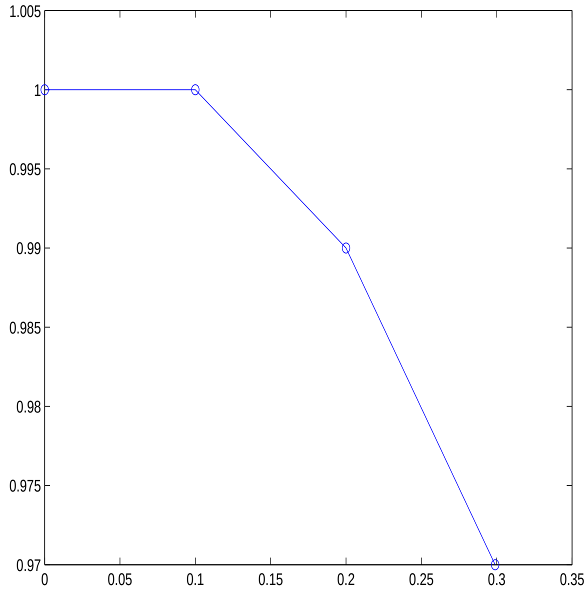
$$\mathbf{y}^1 = \mathbf{y}^0 + 0.1\,\mathbf{f}\left(\mathbf{y}^0\right) = \begin{bmatrix} 0.100 \\ 1.0 \end{bmatrix}$$

$$\mathbf{f}(y^1) = \begin{bmatrix} y_2^1 \\ -\sin\left(y_1^1\right) \end{bmatrix} = \begin{bmatrix} 1.0 \\ -0.09983341665 \end{bmatrix}$$

$$\mathbf{y}^2 = \mathbf{y}^1 + 0.1\,\mathbf{f}\left(\mathbf{y}^1\right) = \begin{bmatrix} 0.200 \\ 0.990 \end{bmatrix}$$

Samalla tavoin:

$$\mathbf{y}^3 = \mathbf{y}^2 + 0.1\,\mathbf{f}\left(\mathbf{y}^2\right) = \begin{bmatrix} 0.2990 \\ 0.97015 \end{bmatrix}$$



Stabiilisuus

Systeemi on **stabiili**, jos lähellä toisiaan olevat alkuarvot tuottavat lähellä tosiaan olevia ratkaisuja.

Tyypillisesti yhtälö $y' = f(t, y)$ on stabiili, jos $\frac{\partial}{\partial y} f(t, y) < 0$, systeemin tapauksessa $y-$ muuttujien suhteen lasketun Jacobin matriisin ominaisarvojen reaaliosat < 0.

Havainnollisesti tämä tarkoittaa, että trajektorien etäisyys toisistaan pienenee, kun t kasvaa.

Numeerinen menetelmä on stabiili, jos kokonaisvirhe ei kasva, kun t kasvaa.

Jos systeemi ei ole stabiili, on turha toivoa, että numeerinen approksimaatio tekisi prosessista stabiilin. Päinvastoin, numeerisilla menetelmillä on yleensä askelpituusrajoitus, jota suurempi askel tekee siitä epästabiilin.