OH till Föreläsning 6, Numme O1, 120208

GKN Kap 4.2. Ickelinjära ekvationssytem och Ickelinjära minstakvadratmetoden

Dagens termer

• Icke-linjära ekvationssystem • Newtons metod • Överbestämda icke-linjära ekvationssytem
• Gauss-Newtons metod • Picard-iteration • Jacobi-iteration • Gauss-Seidel-iteration

Linjära ekvationssystem

Exempel 1:

$$\begin{matrix} x+3y=0 \\ x-2y=1 \end{matrix} \quad \Longrightarrow \quad \begin{pmatrix} 1 & 3 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 1 & -2 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.6 \\ -0.2 \end{pmatrix}$$

Ickelinjära ekvationssystem

Exempel 2:

$$\begin{matrix} z+2zy+3y^2=0 \\ 2z^2y=1 \end{matrix} \quad \Longrightarrow \quad \begin{cases} f(z,y)=0 \\ g(z,y)=0 \end{cases} \quad \text{där} \quad \begin{cases} f(z,y)=z+2zy+3y^2 \\ g(z,y)=2z^2y-1 \end{cases}$$

Taylors formel för en funktion av två variabler

$$f(z+h,y+k)=f(z,y)+h\frac{\partial f}{\partial z}(z,y)+k\frac{\partial f}{\partial y}(z,y)+\frac{h^2}{2}\frac{\partial^2 f}{\partial z^2}(z,y)+\frac{2hk}{2}\frac{\partial^2 f}{\partial z\partial y}(z,y)+\frac{k^2}{2}\frac{\partial^2 f}{\partial y^2}(z,y)+\dots$$

$$\begin{matrix} 0=f(z,y)+h\dfrac{\partial f}{\partial z}(z,y)+k\dfrac{\partial f}{\partial y}(z,y) \\ 0=g(z,y)+h\dfrac{\partial g}{\partial z}(z,y)+k\dfrac{\partial g}{\partial y}(z,y) \end{matrix} \quad \Longrightarrow \quad \begin{pmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial z} & \frac{\partial g}{\partial y} \end{pmatrix} \begin{pmatrix} h \\ k \end{pmatrix} = \begin{pmatrix} -f(z,y) \\ -g(z,y) \end{pmatrix} \quad h \text{ och } k \text{ leder mot nollstället!}$$

Min startgissning till lösningen är $z_0=-1$ och $y_0=1$. Detta ger $f(z_0,y_0)=0$ och $g(z_0,y_0)=1$ och

$$J=\begin{pmatrix} 1+2y & 2z+6y \\ 4zy & 2z^2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1+2y_0 & 2z_0+6y_0 \\ 4z_0y_0 & 2z_0^2 \end{pmatrix} \begin{pmatrix} h_0 \\ k_0 \end{pmatrix} = \begin{pmatrix} -f_0 \\ -g_0 \end{pmatrix} \Rightarrow \begin{pmatrix} 3 & 4 \\ -4 & 2 \end{pmatrix} \begin{pmatrix} h_0 \\ k_0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$\Rightarrow \begin{cases} h_0=0.18181818\simeq 0.1818 \\ k_0=-0.13636364\simeq -0.1364 \end{cases} \Rightarrow \begin{cases} z_1=z_0+h_0=-1+0.1818=-0.8182 \\ y_1=y_0+k_0=1-0.1364=0.8636 \end{cases}$$

Nästa iterationssteg blir

$$\begin{pmatrix} \frac{\partial f}{\partial z}(z_1,y_1) & \frac{\partial f}{\partial y}(z_1,y_1) \\ \frac{\partial g}{\partial z}(z_1,y_1) & \frac{\partial g}{\partial y}(z_1,y_1) \end{pmatrix} \begin{pmatrix} h_1 \\ k_1 \end{pmatrix} = \begin{pmatrix} -f(z_1,y_1) \\ -g(z_1,y_1) \end{pmatrix} \quad \Longleftrightarrow \quad \begin{pmatrix} 2.7273 & 3.5455 \\ -2.8264 & 1.3388 \end{pmatrix} \begin{pmatrix} h_1 \\ k_1 \end{pmatrix} = \begin{pmatrix} -0.006198 \\ -0.1563 \end{pmatrix}$$

$$\Longrightarrow \quad \begin{cases} h_1=0.0399 \\ k_1=-0.0325 \end{cases} \quad \Longrightarrow \quad \begin{cases} z_2=z_1+h_1=-0.8182+0.0399=-0.7783 \\ y_2=y_1+k_1=0.8636+(-0.0325)=0.8312 \end{cases}$$

och nästa iterationssteg blir

$$\begin{pmatrix} \frac{\partial f}{\partial z}(z_2,y_2) & \frac{\partial f}{\partial y}(z_2,y_2) \\ \frac{\partial g}{\partial z}(z_2,y_2) & \frac{\partial g}{\partial y}(z_2,y_2) \end{pmatrix} \begin{pmatrix} h_2 \\ k_2 \end{pmatrix} = \begin{pmatrix} -f(z_2,y_2) \\ -g(z_2,y_2) \end{pmatrix} \quad \Longleftrightarrow \quad \begin{pmatrix} 2.6624 & 3.4306 \\ -2.5875 & 1.2114 \end{pmatrix} \begin{pmatrix} h_2 \\ k_2 \end{pmatrix} = \begin{pmatrix} -0.0005688 \\ -0.006888 \end{pmatrix}$$

$$\Rightarrow \begin{cases} h_2=0.001896 \\ k_2=-0.001637 \end{cases} \Rightarrow \begin{cases} z_3=z_2+h_2=-0.7783+0.001896=-0.7764 \\ y_3=y_2+k_2=0.8312+(-0.001637)=0.8295 \end{cases} \Longrightarrow \begin{cases} z=-0.7764\pm 0.0020 \\ y=\ \ 0.8295\pm 0.0017 \end{cases}$$

Uttryckt på ett mer kompakt sätt skulle Ex2 beskrivas $z = x_1$, $y = x_2$, $f = f_1$, $g = f_2$ $h = t_1$, $k = t_2$ och

$$\bar{f}(\bar{x}) = \begin{pmatrix} f_1(\bar{x}) \\ f_2(\bar{x}) \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1 + 2x_1x_2 + 3x_2^2 \\ 2x_1^2x_2 - 1 \end{pmatrix} \qquad J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 + 2x_2 & 2x_1 + 6x_2 \\ 4x_1x_2 & 2x_1^2 \end{pmatrix}$$

Newtons metod för icke-linjära system	**Newtons metod för en icke-linjär ekvation**
Gissa startvektorn $\bar{x}_0$.	Gissa startvärdet x_0.
För $n = 0, 1, 2, \ldots$ lös det linjära ekvationssystemet	För $n = 0, 1, 2, \ldots$ beräkna korrektionen
$J(\bar{x}_n)\bar{t}_n = \bar{f}(\bar{x}_n)$	$t_n = f(x_n)/f'(x_n)$
Sätt så $\bar{x}_{n+1} = \bar{x}_n - \bar{t}_n$	Sätt så $x_{n+1} = x_n - t_n$
Avbryt då $\|\bar{t}_n\| < \varepsilon$	Avbryt då $\|t_n\| < \varepsilon$

```
% Newton for system
x=[-1 1]';
t=1; it=0; maxit=10;
disp('  x     f     J     t')
while norm(t)>1e-9 & it<maxit;
  f=[x(1)+2*x(1)*x(2)+3*x(2)^2
     2*x(1)^2*x(2)-1];
  J=[1+2*x(2)      2*x(1)+6*x(2)
     4*x(1)*x(2)   2*x(1)^2];
  t=J\f;
  disp([x f J t]), disp(' ')
  x=x-t; it=it+1;
end;
if it<maxit;
  losn=x
else
  disp('Ingen konvergens!')
  losn=[];
end;
```

```
% Newton for system
x=[-1 1]';
t=1; it=0; maxit=10;
disp('  x     f     J     t')
while norm(t)>1e-9 & it<maxit;
  f1=x(1)+2*x(1)*x(2)+3*x(2)^2;
  f2=2*x(1)^2*x(2)-1;
  f=[f1
     f2];

  df1dx1=1+2*x(2);
  df1dx2=2*x(1)+6*x(2);
  df2dx1=4*x(1)*x(2);
  df2dx2=2*x(1)^2;
  J=[df1dx1 df1dx2
     df2dx1 df2dx2];

  t=J\f;
  disp([norm(x) norm(f) norm(t)])
  x=x-t; it=it+1;
end;
if it<maxit
   ....
end;
```

```
% Vanliga Newton-Raphson.
x=...;
t=1; it=0; maxit=10;
disp('  x     f     d     t')
while abs(t)>1e-9 & it<maxit;
  f= ...;
  d= ...;
  t=f/d;
  disp([x f d t])
  x=x-t; it=it+1;
end;
if it<maxit;
  rot=x
else
  disp('Ingen konv')
  rot=[];
 end;
```

När systemet är stort är det ibland jobbigt att beräkna alla derivatorna. (Fem obekanta och fem ekvationer ger 25 derivator!). Då kan man använda Newtons modifierade metod där man, precis som i det endimensionella fallet, skattar derivatan med en differens:

Newtons modifierade metod

Approximera $\dfrac{\partial f_i}{\partial x_j} \approx \dfrac{f_i(\bar{z}) - f_i(\bar{x})}{s}$ där $z_k = \begin{cases} x_k & k \neq j \\ x_k + s & k = j \end{cases}$ och s är ett litet tal

```
% Modif Newton for system
x=[-1 1]';
t=1; it=0;
disp('  x    f    J    t')
while norm(t)>1e-10 & it<maxit;
  f=funk(x);
  .
  s=1e-6;    % Valj lagom litet!!!
  xs=x; xs(1)=xs(1)+s; fs=funk(xs);
  dfdx1=(fs-f)/s;
  xs=x; xs(2)=xs(2)+s; fs=funk(xs);
  dfdx2=(fs-f)/s;
  J=[dfdx1 dfdx2];
  .
  t=J\f;
  disp([x f J t]), disp(' ')
  x=x-t; it=it+1;
end;
if it<maxit;
  ...
end;
```

```
function f=funk(x);
f=[x(1)+2*x(1)*x(2)+3*x(2)^2
   2*x(1)^2*x(2)-1];
```

```
J=[]; s=1e-6;
for i=1:length(x);
  xs=x; xs(i)=xs(i)+s; fs=funk(xs);
  dfdxi=(fs-f)/s;
  J=[J dfdxi];
end;
```

```
function f=funk(x);
z=x(1); y=x(2);
f=[z+2*z*y+3*y^2
   2*z^2*y-1];
```

Exempel 3: Givet tabellen

x	1	2	4
y	3	5	13

. Bestäm a och b så att $y = a\,\mathrm{e}^{bx}$

Lösningsmetod från föreläsning 5: Linearisera problemet $\ln y = \ln a + b\,x$. Sätt $c_1 = \ln a$ och $c_2 = b$. Det överbestämda linjära ekvationssystemet blir

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \ln y_1 \\ \ln y_2 \\ \ln y_3 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \ln 3 \\ \ln 5 \\ \ln 13 \end{pmatrix} \Rightarrow c = \begin{pmatrix} 0.6209 \\ 0.4872 \end{pmatrix} \Rightarrow \begin{array}{l} a = \mathrm{e}^{c_1} = 1.8605 \\ b = c_2 = 0.4872 \end{array}$$

Dagens metod: Nu har vi lärt oss lösa ickelinjära system:

$$\begin{cases} a\,\mathrm{e}^{b\,x_1} - y_1 = 0 \\ a\,\mathrm{e}^{b\,x_2} - y_2 = 0 \\ a\,\mathrm{e}^{b\,x_3} - y_3 = 0 \end{cases}$$

Vi skriver detta system $\bar{f}(a, b, \bar{x}, \bar{y}) = \bar{0}$

Kom ihåg att de obekanta är a och b och inte x!

Vi löser nu systemet med avseende på a och b.

Det som skiljer Gauss-Newtons metod från Newtons metod för system är att ekvationssytemet $J\,\bar{t} = \bar{f}$ är överbestämt och alltså måste lösas med minstakvadratmetoden (dvs man löser $J^T J\,\bar{t} = J^T \bar{f}$).

Gauss-Newtons metod för icke-linjära överbestämda system

Lägg sökta parametrar i vektorn $\bar{c}$.

Givna mätdata är (x_i, y_i).

Definiera $f_i = f(\bar{c}, x_i, y_i)$.

Definiera $J_{ij} = \frac{\partial f_i}{\partial c_j}$

Då blir algoritmen:

Gissa startvektorn $\bar{c}_0$.

För $n = 0, 1, 2, \ldots$
lös det överbestämda linjära ekv-systemet

$J(\bar{c}_n)\bar{t}_n = \bar{f}(\bar{c}_n)$

Sätt så $\bar{c}_{n+1} = \bar{c}_n - \bar{t}_n$

Avbryt då $||\,\bar{t}_n|| < \varepsilon$

I vårt exempel 3 får vi

$$\bar{c}=\begin{pmatrix} a \\ b \end{pmatrix} \quad \bar{f}=\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}=\begin{pmatrix} a\,\mathrm{e}^{b\,x_1}-y_1 \\ a\,\mathrm{e}^{b\,x_2}-y_2 \\ a\,\mathrm{e}^{b\,x_3}-y_3 \end{pmatrix} \quad J=\begin{pmatrix} \frac{\partial f_1}{\partial a} & \frac{\partial f_1}{\partial b} \\ \frac{\partial f_2}{\partial a} & \frac{\partial f_2}{\partial b} \\ \frac{\partial f_3}{\partial a} & \frac{\partial f_3}{\partial b} \end{pmatrix}=\begin{pmatrix} \mathrm{e}^{b\,x_1} & a\,x_1\,\mathrm{e}^{b\,x_1} \\ \mathrm{e}^{b\,x_2} & a\,x_2\,\mathrm{e}^{b\,x_2} \\ \mathrm{e}^{b\,x_3} & a\,x_3\,\mathrm{e}^{b\,x_3} \end{pmatrix} \implies \begin{pmatrix} \times & \times \\ \times & \times \\ \times & \times \end{pmatrix}\begin{pmatrix} t_1 \\ t_2 \end{pmatrix}=\begin{pmatrix} \times \\ \times \\ \times \end{pmatrix}$$

Gauss-Newtons metod ger oss $a = 1.8840$ och $b = 0.4830$. Resultatet är inte exakt detsamma som för det lineariserade problemet. Det beror på att vi egentligen hittar MKV-lösningen för olika problem, det omskrivna lineariserade respektive det ursprungliga.

```
% Gauss-Newton
x=[1 2 4]'; y=[3 5 13]'; c=[1 1]';
t=1; it=0;
disp('  c       t')
while norm(t)>1e-10 & it<maxit;
  a=c(1); b=c(2);
  f=a*exp(b*x)-y;
  J=[exp(b*x)  a*x.*exp(b*x)];
  t=J\f;
  disp([c t]), disp(' ')
  c=c-t; it=it+1;
end;
if it<maxit;
  a=c(1), b=c(2)
else
  disp('Ingen konvergens!')
  a=[]; b=[];
end;



function f=funk2(c);
x=[1 2 4]'; y=[3 5 13]';
a=c(1); b=c(2);
f=a*exp(b*x)-y;
```

```
% Modif Gauss-Newton
c=[1 1]';
t=1; it=0;
disp('Normer:  c   f    J   t')
while norm(t)>1e-10 & it<maxit;
  f=funk2(c);

  s=1e-6;   % Valj s lagom litet!!!
  cs=c; cs(1)=cs(1)+s; fs=funk2(cs);
  dfdc1=(fs-f)/s;
  cs=c; cs(2)=cs(2)+s; fs=funk2(cs);
  dfdc2=(fs-f)/s;
  J=[dfdc1 dfdc2];

  t=J\f;
  disp([norm(c) norm(f) norm(J) norm(t)])
  c=c-t; it=it+1;
end;
if it<maxit
   a=c(1), b=c(2)
else
  disp('Ingen konvergens!')
  a=[]; b=[];
end;
```

Notera att Jacobian-matrisen (derivata-matrisen) har fler rader än kolumner. Antalet kolumner bestäms av antalet sökta parametrar. Antalet rader bestäms av antalet givna mätdata. Detta gör också att vi bara tycks behöva så många derivator som vi har kolumner. Givetvis kan man även här använda sig av skattningar av derivatan, se programmet till höger ovan. Det utnyttjar funktionen nere till vänster. Matlab-program för vanliga Newtons metod för system och Gauss-Newtons MKV-metod för överbestämda ickelinjära system ser likadana ut! Hur kommer det sig?

Metoderna för icke-linjära system är iterativa. De behöver en startgissning som metoden sedan förbättrar iterativt. Ligger startvärdena alltför långt ifrån lösningen kanske iterationerna inte konvergerar. Det är alltid klurigt att hitta bra startvärden till ekvationssytem. Oftast försöker man förenkla ekvationerna, kanske ända till ett linjärt ekvationssytem. Lösningen till det förenklade systemet tar man sedan som startvärde. När det gäller Gauss-Newtons metod är ett klassiskt sätt att hitta startvärden att försöka hitta ett lineariserat MKV-problem och lösa det.

© 2012 Ninni Carlsund Levin