

Numeriska metoder grundkurs I

Övning 8 för Bio3

Övningsgrupp 2

Johannes Hjorth
hjorth@nada.kth.se
Rum 4538 på plan 5 i D-huset
08 - 790 69 02

Kurshemsida:
<http://www.nada.kth.se/kurser/kth/2D1210/05-06/BIO/>

Material utdelat på övningarna:
<http://www.nada.kth.se/~hjorth/teaching/numbio05>

Exempelsamlingen 3.10

Om vi linjäriserar $f(x)$ får vi

$$f(x) = f(x_n) + J(x_n)(x - x_n) + \dots = 0$$

Ifall vi ignorerar högre ordningens termer fås

$$J(x_n)(x - x_n) = -f(x_n)$$

vilket kan skrivas som

$$x = x_n - J^{-1}(x_n)f(x_n)$$

Om vi kallar $J^{-1}(x_n)f(x_n)$ för t_n kan vi få en bättre approximation till x från

$$x_{n+1} = x_n - t_n$$

där t_n är lösningen till

$$J(x_n)t_n = f(x_n)$$

f310.m

```
function [f] = f310(x)
f1 = 10*x(1) - 2*x(2) + power(x(1),3);
f2 = x(1) + 10*x(2) + 5*power(x(2),2) - 3;

f = [f1; f2];
```

J310.m

```
function [J] = J310(x)

J = [(10+3*power(x(1),2)) (-2); (1) (10+10*x(2))];
```

exs310.m

```
% Ta fram startvärdesgissning, antar x och y små

A = [10 -2; 1 10];
b = [0; 3];

x = A\b; % Detta är ett bra startvärde

t = [1 1];

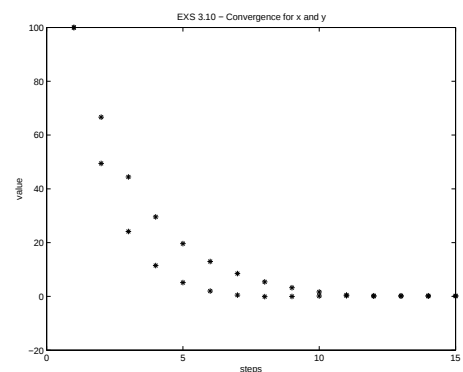
x = [100;100] % Test: Det funkar även med ett dåligt startvärde!
xSave = x;

while(norm(t)>1e-5*norm(x))
    t = -J310(x)\f310(x);
    x = x + t;
    xSave = [xSave x];
end

plot(1:length(xSave),xSave,'*')
xlabel('steps')
ylabel('value')
title('EXS 3.10 - Convergence for x and y')
xSave
```

Vi kör koden

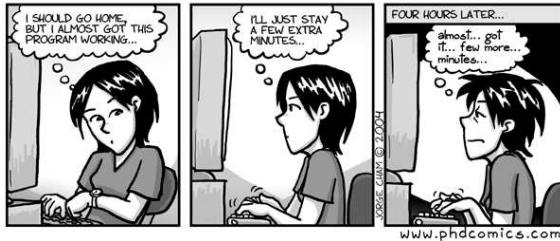
```
>> format compact
>> exs310
x =
    100
    100
xSave =
Columns 1 through 8
100.0000    66.6477    44.4021    29.5553    19.6326    12.9796     8.4870     5.4070
100.0000    49.4419    24.1488    11.4887     5.1512     1.9946     0.4811    -0.0844
Columns 9 through 15
     3.2353     1.6426     0.5149     0.0734     0.0522     0.0521     0.0521
    -0.0218     0.1390     0.2267     0.2595     0.2608     0.2608     0.2608
```



Här plottar vi det dåliga startvärdet, eftersom den plotten blev roligare.

Exempelsamlingen 4.25

Skillnaden i det här talet jämfört med förra talet är att vi måste approximera Jacobianen. Här gör vi det på enklaste sätt med en frammåt differenskvot.



exs425.m

```
clf, clear
global t y
format compact

y = [0.12 -0.09 0.06 -0.05 0.032]';
t = [0.8 1.7 2.5 3.3 4.1]';

s = 0.001;

wStart = 2*pi / (t(3)-t(1));
bStart = wStart/(2*pi)*(log(y(1))-log(y(3)));
x = [wStart; bStart];
xSave = x;

xCorr = inf;

while(norm(xCorr) > 1e-3*norm(x))
    f = f425(x(1), x(2));

    J = [(f425(x(1)+s, x(2))-f)/s (f425(x(1), x(2)+s)-f)/s];
    xCorr = -J\f;
    x = x + xCorr;

    xSave = [xSave x];
end

plot(t,y,'*'), hold on

w = x(1)
b = x(2)

T = 0:0.01:6;
F = -0.17*exp(-b*T).*(cos(w*T)+(b/w)*sin(w*T));
plot(T,F)

xlabel('time')
ylabel('amplitude')
title('EXS 4.25 - Gauss-Newton's metod')
```

f425.m

```
function f = f425(w,b)
global t y

f = -0.17*exp(-b*t).*(cos(w*t)+(b/w)*sin(w*t)) - y;
```

Vi kör koden...

```
>> exs425
w =
    3.7725
b =
    0.3904
```

