Numeriska metoder grundkurs I
Övning 7 för Bio3 och BM

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/2D1210/04-05/BIO/`

Material utdelat på övningarna:

`http://www.nada.kth.se/~hjorth/teaching/numbio04`

Exempelsamlingen 7.15

Vi skriver först funktionen $\frac{du}{dx} = f(x, u)$

```
function f=exs715fun(x, u)

u1prim = u(2);

g = u(1)*exp(4*(1-u(1))/(1+0.2*(1-u(1))));

if(x == 0)
    u2prim = g/3;
else
    u2prim = g - 2*u(2)/x;
end

f = [u1prim; u2prim];
```

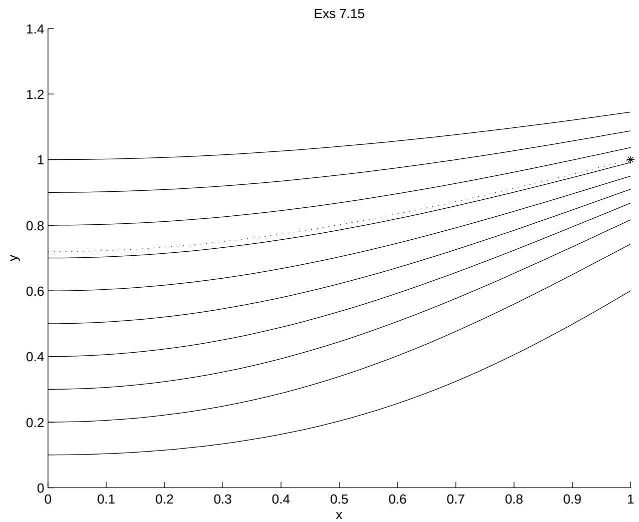
Vi använder oss av inskjutningsmetoden

```
clear, clf
hold on

% Plotta målpunkten
plot(1,1,'*r')

% Plotta olika lösningskurvor
for alpha=0:0.1:1
    [x y] = ode45('exs715fun', [0 1], [alpha 0]);
    plot(x,y(:,1))
end

% Målvärdet (viktigt!)
malvarde = 1;
```

```
% Startvärde
alpha1 = 0.7;
[x y] = ode45('exs715fun', [0 1], [alpha1 0]);
n1 = length(x);
y1 = y(n1,1) - malvarde;

alpha2 = 0.75;

i = 0; % Loopräknare
h = 1;

alphas = [alpha; alpha2];

while(abs(h/alpha2) > 0.5e-6 & i < 50)
    [x y] = ode45('exs715fun', [0 1], [alpha2 0]);
    n2 = length(x);
    y2 = y(n2,1) - malvarde;

    % Sekantmetoden
    h = (alpha2 - alpha1)/(y2 - y1) * y2;
    alpha1 = alpha2;
    alpha2 = alpha2 - h;

    alphas = [alphas; alpha2];
    i = i + 1;
    y1 = y2;
end

% Utskrift av alphas vandring...
alphas

% Sökt värde
eta = 3 * y(n2,2)

% Tjusig plot
plot(x,y(:,1),':r')
xlabel('x')
ylabel('y')
title('Exs 7.15')
```

Vi kör koden

```
>> exs715
alphas =
   1.00000000000000
   0.75000000000000
   0.71918474419493
   0.71947189796233
   0.71947633193996
   0.71947633131107
eta =
   1.35504006636729
```

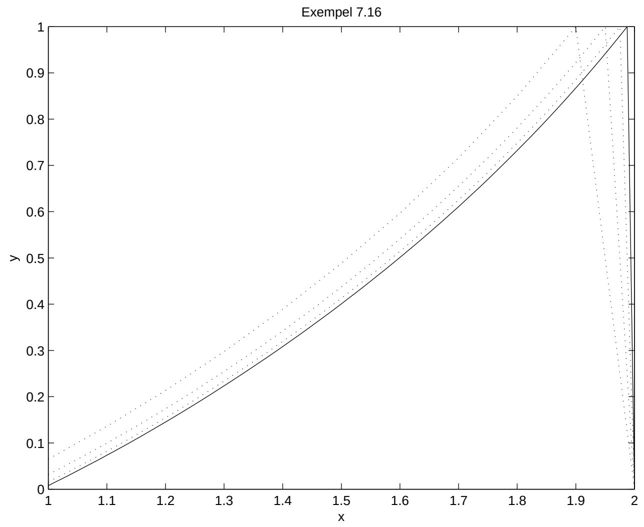


Exempelsamlingen 7.16

```
clear, clf

N = 10; res = [];
linjer = [':' ':' ':' '-'];

for studie=1:4

    n = N-1; % Antalet fria noder (y0 är fix pga RV)
    h = 1/N;

    x = 1+h*(1:n)';
    g = 2-2*x+x.^2;

    dia = (1-2/h^2)*ones(n,1);       % dia, sub och sup måste ha samma längd
    sub = [1/h^2+g(2:n)/(2*h); 0];   % Sista elementet i sub anv. ej
    sup = [0; 1/h^2-g(1:n-1)/(2*h)]; % Första elementet i sup anv. ej

    A = spdiags([sub dia sup], -1:1, n, n);
    b = sparse(n,1,-(1/h^2-g(n)/(2*h)));

    y = A\b;
    X = [1;x(1:n);2];
    Y = [0;y(1:n);1];

    plot(X,Y,linjer(studie))
    hold on

    yprim0 = (-Y(3)+4*Y(2)-3*Y(1))/(2*h);
    Th = h*(sum(Y)-(Y(1)+Y(n+2))/2);
    res = [res; [h yprim0 Th]];
    N = 2*N;
end

yprimbest = yprim0;
Iappbest = Th;
format short
disp('    h         yprim0    yprimdiff Th         trapdiff    ')
disp([res(:,1:2)  res(:,2)-yprimbest  res(:,3)  res(:,3)-Iappbest])
xlabel('x')
ylabel('y')
title('Exempel 7.16')
```

Vi kör koden...

```
>> exs716
    h         yprim0    yprimdiff Th         trapdiff
    0.1000    0.6176    0.0002    0.4252    0.0009
    0.0500    0.6174    0.0001    0.4245    0.0002
    0.0250    0.6174    0.0000    0.4243    0.0000
    0.0125    0.6173         0    0.4243         0
```



Facit har en annan lösning

```
clear, clf

N = 10; res = [];

for studie=1:4

    n = N-1;
    h = 1/N;
    x = 1+h*(1:n)';
    g = 2-2*x+x.^2;
    dia = (1-2/h^2)*ones(n,1);
    sub = 1/h^2+g(2:n)/(2*h);
    sup = 1/h^2-g(1:n-1)/(2*h);
    b = zeros(n,1);
    b(n)=-(1/h^2-g(n)/(2*h));
    y = tridia(dia,sup,sub,b);
    X = [1;x;2];
    Y = [0;y;1];

    plot(X,Y)
    hold on

    yprim0 = (-Y(3)+4*Y(2)-3*Y(1))/(2*h);
    Th = h*(sum(Y)-(Y(1)+Y(n+2))/2);
    res = [res; [h yprim0 Th]];
    N = 2*N;
end

yprimbest = yprim0;
Iappbest = Th;
format short
disp('     h     yprim0       yprimdiff       Th       trapdiff    ')
disp([res(:,1:2)  res(:,2)-yprimbest  res(:,3)  res(:,3)-Iappbest])
```

Deras lösning använder sig av `tridia.m`

```
function x = tridia(d,p,q,b)
% Beräknar lösningsvektorn x till tridiagonalt system
% med diagonal d, superdiagonal p, subdiagonal q och högerled b
    n=length(b);  r=d; g=b; x=b;
    for i=2:n
      i1=i-1;  mult=q(i1)/r(i1);           % Gausseliminera
      r(i)=d(i)-mult*p(i1); g(i)=b(i)-mult*g(i1);
    end
    x(n)=g(n)/r(n);                        % Bakåtsubstituera
    for i=n-1:-1:1, x(i)=(g(i)-p(i)*x(i+1))/r(i); end
```