NUMERIČNE METODE

UNIVERZITETNI ŠTUDIJ 2006/07

ZAPISKI Z VAJ

asist. mag. Andrej Kotar

1. vaja

Prijava z uporabniškim imenom **Vaje.**

Kreiranje nove mape na **My Documents\vaje\ime_priimek.**

Navodilo za delo s programom MATLAB 7.0.4:

1. **Zagon programa:**

- dvoklik ikone MATLAB 7.0.4 na namizju ali

- Start
- All Programs
- MATLAB 7.0.4
- MATLAB 7.0.4

2. **Začetek pisanja M-datoteke v Editor:**

- klik ikone **`New M-File`** ali

- **`File`**
- **`New`**
- **`M-File`**

3. **Z urejevalcem teksta napišemo program.**

4. **Izvajanje napisanega programa**

- klik ikone **`Save and run`** ali

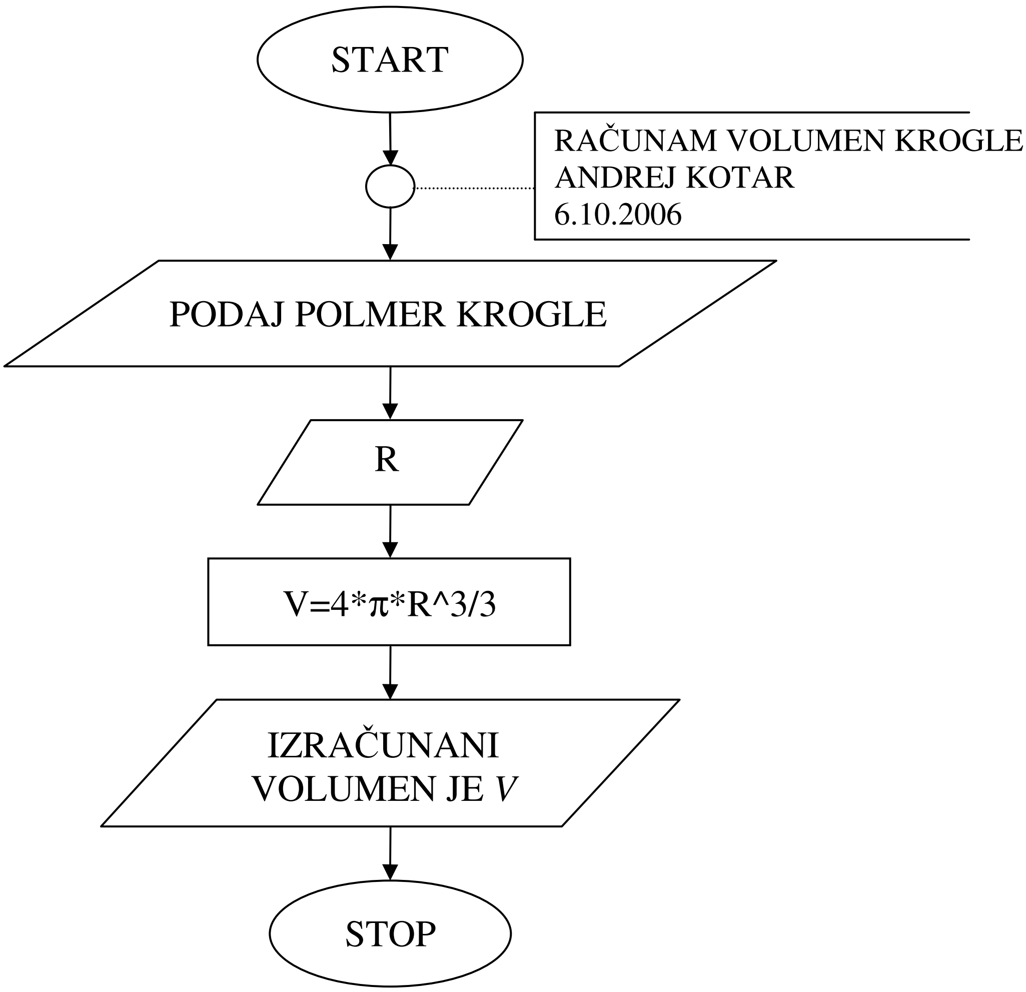
- **`Debug`**
- **`Save and Run`**

5. **V oknu Command Window se izpišejo rezultati.**

Po zaključku dela vse napisane M-datoteke shranite v svojo mapo!

1. naloga:

a) Napišite program za računanje volumna krogle:




```
% Program za racunanje volumna krogle
% Program sprejme polmer krogle
% Program vrne volumen krogle
% A. Kotar, 6. 10. 2006
clc;
close all;
R=input('Podaj polmer krogle: ');
pi=3.1415926535897;
V=4*pi*R^3/3;
fprintf(1,'Volumen = %12.8f \r',V);
```

b) Napišite program za računanje volumna stožca. Uporabite spodnjo formulo za izračun volumna

$$V = \frac{1}{3} r^2 \pi h$$

```
% Vaja 1.1a
% Program za racunanje volumna krogle
% Program sprejme polmer krogle
% Program vrne volumen krogle
% A. Kotar, 6. 10. 2006
clc; % ocisti komandno okno
close all; % zapre vsa odprta graficna okna
R=input('Podaj polmer krogle: ');
pi=3.1415926535897;
V=4*pi*R^3/3;
fprintf(1,'Volumen = %12.8f \r',V);
% test: R=3.5, V=179.59438003
```

```
% Vaja 1.1b
% Program za racunanje volumna stozca
% Program sprejme polmer in visino stozca
% Program vrne volumen stozca
% A. Kotar, 6. 10. 2006
clc;
close all;
r=input('Podaj polmer osnovne ploskve: ');
h=input('Podaj visino stozca: ');
V=r^2*pi*h/3;
fprintf(1,'Volumen = %12.8f \r',V);
% test: r=3.5, h=5, V=64.14085001
```

1. naloga:

a. Izračunajte naslednjo vsoto in produkt za poljuben x.

$$s = \sum_{i=1}^{10} \left(\frac{x}{i} \right)^i \quad p = \prod_{i=1}^{10} \left(\frac{x}{i} \right)^i$$

b. Izračunajte vsoto prvih stotih števil.

$$s = \sum_{i=1}^{100} i$$

2. naloga:

Napišite program za izračun faktorjele.

3. naloga:

V kartezijevem koordinatnem sistemu narišite srčnico, ki ima v polarnem koordinatnem sistemu enačbo $r = a(1+\cos\varphi)$, a=3.5 in $0 \le \varphi \le 2\pi$. Funkcijo tabelirajte s korakom $\Delta\varphi$=0.1.

4. naloga: Za spodaj napisani matriki A in B naredite:

a) Transponirajte matriki A in B

b) Združite transponirani matriki A^T in B^T v novo matriko

c) Narišite graf matrik A^T in B^T

$$A = \begin{bmatrix} 1 & 1 & 2 & 3 & 3 \\ 4 & 4 & 5 & 6 & 6 \\ 7 & 7 & 8 & 9 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 5 & 3 & 1 & 5 \\ 5 & 2 & 1 & 2 & 5 \\ 1 & 5 & 3 & 5 & 1 \end{bmatrix}$$

5. naloga:

Matlabova funkcija *sech* se izračuna po formuli

$$\mathrm{sech}(x) = s = 4\pi \sum_{i=1,3,5}^{\infty} \frac{(-1)^{(i-1)/2} i}{(i\pi)^2 + 4x^2}$$

Če seštejemo nekaj členov (do *n=305*), dobimo približek funkcije. Za izbrani x izračunajte vrsto s in z Matlabovim ukazom funkcijo *sech(x)*. Vse tri vrednosti izpišite na 5 decimalnih mest.

```
% Vaja2.1a
% Program izračuna vsoto in produkt vrste.
% A.K., 13. 10. 2006
clc;
x=input('Vpisite x: ');
i=[1:1:10];
a=(x./i).^i;
s=sum(a);
p=prod(a);
%izpis
fprintf(1,'Vsota elementov vrste: s = %10.6f \r',s);
fprintf(1,'Produkt elementov vrste: p = %10.6f \r',p);
%test: x=7, s=50.708492, p=140.081955
```

```
% Vaja2.1b
% Program izračuna vsoto prvih stotih stevil.
% A.K., 13. 10. 2006
clc;
i=[1:1:100];
s=sum(i);
fprintf(1,'Vsota prvih stotih stevil: s = %10i \r',s);
%test: s=5050
```

```
% Vaja2.2
% Program za izračun faktorjele.
% A.K., 13. 10. 2006
clc;
n=input('Vpisite n: ');
i=[1:1:n];
fakt=prod(i);
fprintf(1,'Faktorjela: n! = %8i \r',fakt);
%test: n=12, n!=479001600
```

```
% Vaja 2.3
% Program v kartezijevem koordinatnem sistemu nariše srčnico,
% ki ima v polarnem koordinatnem sistemu enačbo r=a(1+cos(fi)),
% a=3.5 in 0<=fi<=2pi. Funkcija je tabelirana s korakom fi=0.1.
clc;
a=3.5;
fi=[0:0.1:2*pi];  % boljše bi bilo - fi=linspace(0,2*pi,100)
r=a*(1+cos(fi));
x=r.*cos(fi);
y=r.*sin(fi);
X=[min(x),max(x)]
Y=[min(y),max(y)]
plot(x,y,X,[0,0],'r -',[0,0],Y,'g -');
```

```
% Vaja 2.4
% Program transponira matriki A in B in ju nato združi.
% Na osnovi transponiranih matrik A in B nariše graf.
% A. Kotar, 23. 2. 2006
clc;
close all;
fprintf(1,'Matrika A: \r');
A=[ 1 1 2 3 3
    4 4 5 6 6
    7 7 8 9 9]
fprintf(1,'Matrika B: \r');
B=[1 5 3 1 5; 5 2 1 2 5; 1 5 3 5 1]
fprintf(1,'Transponirani matriki AT in BT \r');
AT=A'
BT=B'
fprintf(1,'Zdruzeni matriki AT in BT \r');
C=[AT,BT]
% risanje
plot(AT,BT);
```

```
% Vaja 2.5
% Matlab funkcija sech se izračuna z vrsto.
% Če seštejemo člene do n=305, dobimo nek približek funkcije.
% Program izračuna funkcijo sech z vrsto in Matlabovim ukazom
% za vrednosti x med 0 in 2.
clc
x=input('podaj x: ');
n=305;
i=(1:2:n);
clen=(-1).^((i-1)/2).*i./((i*pi).^2+4*x^2);
s=4*pi*sum(clen);
fprintf(1,'x,s,sech(x) = %8.5f%8.5f%8.5f\r',x,s,sech(x))
```

1. naloga:

Izračunajte razliko dveh vektorjev nato pa dolžino izračunane razlike. Določite tudi število elementov vektorja **c**.

$$\vec{a} = (1.2, -1.8, 3.4)$$

$$\vec{b} = (-3.6, 1.9, 0.7)$$

$$\vec{c} = \vec{a} - \vec{b}$$

$$dc = \left|\vec{c}\right| = \sqrt{c_x^2 + c_y^2 + c_z^2}$$

2. naloga:

Izračunajte povprečje števil zapisanih v datoteki »*podatki.txt*«.

3. naloga:

Napišite program, ki ustvari matriko M kot magični kvadrat reda 3. Preverite njeno magičnost in izračunajte njeno determinanto. Izračunajte matriko B, ki je inverzna matrika od matrike M. Izračunajte produkta B·M = E1 in M·B = E2, nato pa preverite, da je E1=E2. Za konec določite še enotsko matriko E reda 3.

4. naloga:

Napišite program, ki množi matriki A in B na različne načine.

$$A = \begin{bmatrix} 3 & 5 & 2 \\ 2 & 7 & 8 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & 1 \\ 2 & -1 \\ 2 & 3 \end{bmatrix}$$

Nato dodajte matriki A vrstico [-2,3,-1] in matriki B stolpec [2, -3,1], ter tako dopolnjeno matriko A pomnožite z enotsko matriko.

```
% Vaja 3.1
% Program za racunanje razlike in dolzine vektorjev
% Program sprejme vektorja a in b
% Program vrne njuno razliko, dolzino ter st. elementov
% A. Kotar, 20. 10. 2006
clc;
% vektor a
fprintf(1,'Podaj komponente vektorja a: \r');
a(1)=input('a(1) = ');
a(2)=input('a(2) = ');
a(3)=input('a(3) = ');
% vektor b
fprintf(1,'Podaj komponente vektorja b: \r');
b(1)=input('b(1) = ');
b(2)=input('b(2) = ');
b(3)=input('b(3) = ');
% razlika dveh vektorjev
c=a-b;
fprintf(1,'Razlika vektorjev c = a - b: \r');
fprintf(1,'%12.8f  \r',c);
% dolzina vektorja c
dc=sqrt(c(1)^2+c(2)^2+c(3)^2);
fprintf(1,'Dolzina vektorja c: %12.8f \r',dc);
% stevilo elementov vektorja c
nc=length(c);
fprintf(1,'St. elementov vektorja c: %3i \r',nc);
% test: c = [4.80000000,-3.70000000,2.70000000]
%       dc = 6.63475697, nc = 3
```

```
%vaja3.2
%Povprečje stevil zapisanih v datoteki
%A.K., 20. 10. 2006
clc;
st=load('podatki.txt'); %bere vsa števila
n=length(st);
s=sum(st);
povp=s/n;   %povprecje
fprintf(1,'Povprecje stevil: %10.6f \r',povp);
% test:  4.000000
```

```
podatki.txt
7
9
2
1
1
```

```
%vaja3.3
%Matricne operacije z matriko M reda 3, ki
%je zapisana kot magični kvadrat tretjega reda.
% A.K., 20. 10. 2006
clc;
M=magic(3)       %magični kvadrat
V=sum(M)         %vsota po stolpcih
R=sum(M')        %vsota po vrsticah
d=sum(diag(M))   %vsota elementov diagonale
D=det(M)         %determinanta matrike M
B=inv(M)         %inverzna matrika
E1=B*M           %produkt inverzne in magične
E2=M*B           %produkt magične in inverzne
E=eye(3)         %enotska matrika
```

```
% Vaja 3.4
% Program množi matriki A in B na različne načine.
% A. Kotar, 20. 10. 2006
clc;
A=[ 3 5 2; 2 7 8]
B=[-1  1;
    2 -1
    2  3]
C=A*B
C1=A'*B'
C2=A.*B'
D=B*A
D1=B'*A'
D2=B.*A'
A=[A;-2,3,-1]    % dodatna vrstica matrike A
B=[B,[2;-3;1]]   % dodatni stolpec matrike B
% B=[B,[2,-3,1]']
E=eye(3)
F=E*A            % množenje z enotsko matriko
F1=A*E
```

1. naloga:

Napišite program za izračun korenov kvadratne enačbe.

$$ax^2 + bx + c = 0$$

Vrednosti koeficientov *a*, *b* in *c* preberite iz datoteke.

2. naloga:

Napišite program, ki bo za vpisane cifre med 1 in 5 izpisal njihova imena z besedo.

3. naloga:

Izračunajte funkcijo *sin(x)* s pomočjo neskončne vrste

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

na 6 decimalnih mest natančno. Vrednost funkcije *sin(x)*, ki ste jo izračunali s pomočjo vrste primerjajte z vrednostjo izračunano z Matlab-ovo funkcijo.

Na enak način s pomočjo vrste izračunajte tudi funkcijo *cos(x)*.

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$$

4. naloga:

Narišite funkcijo signum na intervalu od -2 do 2.

```
%vaja4.1
%Izračun korenov kvadratne enačbe
%A.K., 3. 11. 2006
clc;
st=load('kvadratna.txt');
[n,m]=size(st); %stevilo vrstic
for i=1:n
   a=st(i,1);
   b=st(i,2);
   c=st(i,3);
   if a==0
      if b==0
         if c==0
            fprintf(1,'Koeficienti so 0, neskoncno resitev!\r\r');
         else
            fprintf(1,'Samo c je razlicen od 0, protislovje!\r\r');
         end
      else
         x1=-c/b;
         fprintf(1,'Linearna enacba!\r');
         fprintf(1,'x = %10.6f \r\r',x1);
      end
   else
      d=b^2-4*a*c;
      if d>=0
         x1=(-b+sqrt(d))/(2*a);
          x2=(-b-sqrt(d))/(2*a);
         fprintf(1,'x1 = %10.6f \r',x1);
         fprintf(1,'x2 = %10.6f \r\r',x2);
      else
         fprintf(1,'Kompleksni resitvi!\r');
         x1=(-b+sqrt(d))/(2*a)
          x2=(-b-sqrt(d))/(2*a)
      end
   end
end
```

```
kvadratna.txt
3.4   5.1   0.3
0     1.6   2
4     1     7
0     0     0
0     0     7
```

```
x1 =  -0.061331
x2 =  -1.438669

Linearna enacba!
x =  -1.250000

Kompleksni resitvi!
x1 = -0.1250 + 1.3170i
x2 = -0.1250 - 1.3170i

Koeficienti so 0, neskoncno resitev!

Samo c je razlicen od 0, protislovje!
```

```
%vaja4.2
%Uporava switch stavka
%A.K., 3. 11. 2006
clc;
i=input('Vstavite stevilko od 1 do 5: ');
switch i
    case 1
        fprintf('Ena.\r');
    case 2
        fprintf('Dve.\r');
    case 3
        fprintf('Tri.\r');
    case 4
        fprintf('Stiri.\r');
    case 5
        fprintf('Pet.\r');
    otherwise
        fprintf('Ni stevilka med ena in pet.\r');
end

%vaja4.3
%Sinus izračunan z neskončno vrsto
%sin(x)=x/1!-x3/3!+x5/5!-x7/7!+...
%A.K., 3. 11. 2006
clc;
x=input('Vstavi x: ');
clen=x;
s=0;
i=0;
while(abs(clen)>1e-7)
    i=i+1;
    s=s+clen;
    clen=-clen*x^2/((i*2)*(i*2+1));
end
fprintf(1,'St. clenov za izracun: %3i \r',i);
fprintf(1,'Vrsta:    sin(x) = %12.8f \r',s);
fprintf(1,'Funkcija: sin(x) = %12.8f \r',sin(x));

%vaja4.4
%Nariše funkcijo signum na intervalu [-2,2]
% 3. 11. 2006
clc;
clear all;
X=-2:0.001:2;
Y=[];
for x=X
    if x<0
        Y=[Y,-1];
    elseif x==0
        Y=[Y,0];
    else
        Y=[Y,1];
    end
end
plot([-2,2],[0,0],'r-',[0,0],[-2,2],'r-',X,Y);
```

1. naloga:

Napišite program, ki vam iz podanih točk v ravnini izračuna dolžine stranic lika, nato pa vam določi najdaljšo izmed stranic. Koordinate točk preberite iz datoteke »*podatki.txt*«.

2. naloga:

Izračunajte funkcijo *sinh(x)* s pomočjo neskončne vrste

$$\sinh(x) = \frac{e^x - e^{-x}}{2} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots + \frac{x^{2n-1}}{(2n-1)!} + \ldots$$

Za izračun faktorjele napišite samostojno funkcijo.

3. naloga:

Izračunajte funkcijo *exp(x)* s pomočjo neskončne vrste na 6 decimalnih mest natančno:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

Vrednost funkcije *exp(x)*, ki ste jo izračunali s pomočjo vrste primerjajte z vrednostjo izračunano z Matlab-ovo funkcijo.

4. naloga:

Z uporabo rekurzijske formule izračunajte integrale

$$I_n = \int_0^1 x^n e^{x-1} dx,$$

kjer je *n=0,1,2,…* Integriranje po delih nam pri *n > 0* da

$$I_n = x^n e^{x-1}\Big|_0^1 - n\int_0^1 x^{n-1} e^{x-1} dx = 1 - nI_{n-1} \quad I_0 = \int_0^1 e^{x-1} dx = e^{x-1}\Big|_0^1 = 1 - \frac{1}{e}$$

```
%vaja5.1
%Izračun najdaljše stranice lika
%A.K., 12. 11. 2006
clc;
clear all;
dolzina=inline('sqrt((x2-x1)^2+(y2-y1)^2)','x1','y1','x2','y2');
st=load('podatki.txt');
[n,m]=size(st);
for i=1:n
    x(i)=st(i,1);
    y(i)=st(i,2);
end
for i=1:n-1
    d(i)=dolzina(x(i),y(i),x(i+1),y(i+1));
end
d(n)=dolzina(x(n),y(n),x(1),y(1));
dmax=d(1);
m=1;
for i=2:n
    if(d(i)>dmax)
        dmax=d(i);
        m=i;
    end
end
fprintf('Najdaljsa stranica je med tockama:\r');
if(m==n)
  fprintf('x=%10.6f,y=%10.6f,x=%10.6f,y=%10.6f \r',x(n),y(n),x(1),y(1));
else
  fprintf('x=%10.6f,y=%10.6f,x=%10.6f,y=%10.6f\r',x(m),y(m),x(m+1),y(m+1));
end
fprintf('Dolžina = %10.6f \r',d(m));
plot([x,x(1)],[y,y(1)]);
```

```
podatki.txt
1 3
2 0
5 1
6 2
5 4
3 5
```

```
%vaja5.2
%glavni program za funkcijo sinh(x)
clc;
x=input('podaj argument za sinh ')
prim=1;
s=x;
i=3;
eps
while prim>eps
     cl=x^i/fakt(i);
     s=s+cl;
     prim=cl/s;
     i=i+2;
end
fprintf('vsota= %21.15f\n',s);
fprintf('sinh = %21.15f\n',sinh(x));
fprintf('upoštevano členov : %6.0f \n',(i+1)/2);


function f=fakt(n)
%racuna faktorielo pozitivnega celega števila
if n<0
    error(' n mora biti nenegativen')
elseif n==0
    f=1;
else
    if ceil(n)~=n;
    end
    f=1;
    for i=2:n
        f=f*i;
    end
end




%vaja5.3
%Eksponentna funkcija izračunana z neskončno vrsto
%A.K., 12. 11. 2006
clc;
x=input('Vstavi x: ');
clen=1;
s=1;
i=1;
while (abs(clen)>1e-6)
     clen=clen*x/i;
     s=s+clen;
     i=i+1;
end
fprintf(1,'Vrsta:    exp(x) = %12.8f \r',s);
fprintf(1,'Funkcija: exp(x) = %12.8f \r',exp(x));
```

```
%vaja5.4a
%Rekurzijska formula - nestabilen algoritem
clc;
n=0;
I=1-1/exp(1);
fprintf('  n        In \r');
fprintf('%3i %16.6f \r',n,I);
for n=1:25
     I=1-n*I;
     fprintf('%3i %16.6f \r',n,I);
end
```

```
n        In
  0         0.632121
  1         0.367879
  2         0.264241
  3         0.207277
  4         0.170893
  .         . . . .
 17         0.057192
 18        -0.029454
 19         1.559620
 20       -30.192395
 21       635.040293
 22    -13969.886437
 23    321308.388054
 24  -7711400.313301
 25 192785008.832528
```

```
%vaja5.4b
%Rekurzijska formula - stabilen algoritem
clc;
n=26;
I=0;
fprintf('  n        In \r');
fprintf('%3i %16.6f \r',n-1,I);
for n=25:-1:1
     I=(1-I)/n;
     fprintf('%3i %16.6f \r',n-1,I);
end
```

```
 25         0.000000
 24         0.040000
 23         0.040000
 22         0.041739
  .         . . . .
  4         0.170893
  3         0.207277
  2         0.264241
  1         0.367879
  0         0.632121
```

1. naloga:

Izračunajte funkcijo a^x s pomočjo neskončne vrste na 6 decimalnih mest natančno:

$$a^x = e^{x\ln a} = 1 + \frac{x\ln a}{1!} + \frac{(x\ln a)^2}{2!} + \frac{(x\ln a)^3}{3!} + \frac{(x\ln a)^4}{4!} + \dots$$

2. naloga:

V pravokotniku poznamo dolžino diagonale d in ploščino S. Dolžina diagonale je $d=2.0$ enoti, ploščina S pa je velikosti 1, 0.1, 0.01, $\dots, 10^{-8}$ ploščinskih enot. Za dane vrednosti diagonale d in ploščin S izračunajte stranici a in b po formulah

$$a = \frac{1}{2}\left(\sqrt{d^2 + 2S} + \sqrt{d^2 - 2S}\right),$$

$$b = \frac{1}{2}\left(\sqrt{d^2 + 2S} - \sqrt{d^2 - 2S}\right).$$

Nato izračunajte še manjšo stranico b po formuli $b=S/a$.

3. naloga:

Izračunajte število π po Arhimedu, iz obsegov pravilnih mnogokotnikov, ki so včrtani krogu s polmerom $r = 0.5$. Najprej računajte po formuli

$$S_{2n} = 2n\sqrt{\frac{1-\sqrt{1-(S_n/n)^2}}{2}},$$

nato pa še po formuli

$$S_{2n} = S_n\sqrt{\frac{2}{1+\sqrt{1-(S_n/n)^2}}}.$$

S črko S_n je označen obseg pravilnega n-kotnika. Iteracije začnemo s pravilnim šestkotnikom, kjer je obseg enak

$$S_n = 6r = 3.$$

```
%vaja6.1
%Potenčna funkcija izračunana z neskončno vrsto
%A.K., 19. 11. 2006
clc;
a=input('Vstavi a: ');
x=input('Vstavi x: ');
clen=1;
s=1;
i=1;
while (abs(clen)>1e-6)
     clen=clen*x*log(a)/i;
     s=s+clen;
     i=i+1;
end
fprintf(1,'Vrsta:    a^x = %12.8e \r',s);
fprintf(1,'Funkcija: a^x = %12.8e \r',a^x);
%a=5, x=10, x=-10


%vaja6.2
%Računanje stranic pravokotnika
%A.K., 19. 11. 2006
clc;
d=2;
S=1;
fprintf('   d      S      a        b1           b2 \r');
for i=1:20
    a=0.5*(sqrt(d^2+2*S)+sqrt(d^2-2*S));
    b1=0.5*(sqrt(d^2+2*S)-sqrt(d^2-2*S));
    b2=S/a;
    fprintf(' %4.1f %4.1e %5.3f %6.4e %6.4e \r',d,S,a,b1,b2);
    S=0.1*S;
end


%vaja6.3
%Število pi po Arhimedu
%A.K., 19. 11. 2006
clc;
r=0.5;
n=6;
S1=n*r;
S2=n*r;
fprintf('          n     S1n      S2n \r');
fprintf(' %10i %9.7f %9.7f \r',n,S1,S2);
for i=1:28
    S1=2*n*sqrt((1-sqrt(1-(S1/n)^2))/2);
    S2=S2*sqrt(2/(1+sqrt(1-(S2/n)^2)));
    fprintf(' %10i %9.7f %9.7f \r',2*n,S1,S2);
    n=2*n;
end
```

1. naloga:

Napišite program za reševanje spodnjega trikotnega sistema linearnih enačb $L\,c = b$, kjer sta

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -4.5 & 1 \end{bmatrix} \qquad b = \begin{Bmatrix} 7 \\ 13 \\ 5 \end{Bmatrix}.$$

2. naloga:

Napišite program za reševanje zgornjega trikotnega sistema linearnih enačb $U\,x = c$, kjer sta

$$U = \begin{bmatrix} 1 & 4 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & -9 \end{bmatrix} \qquad c = \begin{Bmatrix} 7 \\ 6 \\ 18 \end{Bmatrix}.$$

3. naloga:

Napišite program za reševanje spodnjega in zgornjega sistema linearnih enačb. Podatki so enaki kot pri prvih dveh nalogah. Za reševanje spodnjega in zgornjega sistema enačb zapišite samostojni funkciji.

4. naloga:

Izračunajte rešitev sistema linearnih enačb

$$\begin{aligned} 1x_1 + 1x_2 + 0x_3 + 3x_4 &= 4 \\ 2x_1 + 1x_2 - 1x_3 + 1x_4 &= 1 \\ 3x_1 - 1x_2 - 1x_3 + 2x_4 &= -3 \\ -1x_1 + 2x_2 + 3x_3 - 1x_4 &= 4 \end{aligned}$$

po postopku *Gaussove eliminacije*. Glavni program naj kliče podprogram za izračun po Gaussovi metodi.

```
%vaja7.1
%Reševanje spodnjega trikotnega sistema linearnih enačb
clc;
L=[ 1 0 0;
    1 1 0;
    2 -4.5 1]
b=[ 7
    13
    5 ]
c(1)=b(1)/L(1,1);
c(2)=(b(2)-L(2,1)*c(1))/L(2,2);
c(3)=(b(3)-L(3,1)*c(1)-L(3,2)*c(2))/L(3,3);
%izpis
fprintf('c(1) = %6.3f \r',c(1));
fprintf('c(2) = %6.3f \r',c(2));
fprintf('c(3) = %6.3f \r',c(3));
%test: c(1)=7.000, c(2)=6.000, c(3)=18.000


%vaja7.2
%Reševanje zgornjega trikotnega sistema linearnih enačb
clc;
U=[ 1  4  1;
    0  2 -2;
    0  0  -9]
c=[ 7
    6
    18 ]
x(3)=c(3)/U(3,3);
x(2)=(c(2)-U(2,3)*x(3))/U(2,2);
x(1)=(c(1)-U(1,2)*x(2)-U(1,3)*x(3))/U(1,1);
%izpis
fprintf('x(1) = %6.3f \r',x(1));
fprintf('x(2) = %6.3f \r',x(2));
fprintf('x(3) = %6.3f \r',x(3));
%test: x(1)=5.000, x(2)=1.000, x(3)=-2.000



%vaja7.3
%Reševanje sistema linearnih enačb
clc;
% podatki
L=[ 1  0  0; 1  1  0; 2 -4.5  1]
U=[ 1  4  1; 0  2 -2; 0    0 -9]
b=[ 7; 13; 5]
% funkcija - reševanje spodnjega trikotnega sistema
c=spodnja(L,b)
% funkcija - reševanje zgornjega trikotnega sistema
x=zgornja(U,c)
%test: x(1)=5.000, x(2)=1.000, x(3)=-2.000
```

```
% Reševanje spodnjega trikotnega sistema
function c=spodnja(L,b)
[m,n]=size(L);
r=length(b);
if m~=n | m~=r
    error('podatki niso pravi');
end
for i=1:n
    for j=i+1:n
        if L(i,j)~=0
            error('matrika ni trikotna');
        end
    end
end
d=det(L);
if d==0
    error('sistem nima enolične rešitve');
end
c(1:n)=0;
c(1)=b(1)/L(1,1);
for i=2:n
    c(i)=(b(i)-dot(L(i,1:i-1),c(1:i-1)))/L(i,i);
end
```

```
% Reševanje zgornjega trikotnega sistema
function x=zgornja(U,c)
[m,n]=size(U);
r=length(c);
if m~=n | m~=r
    error('podatki niso pravi');
end
for i=1:n
    for j=1:i-1
        if U(i,j)~=0
            error('matrika ni trikotna');
        end
    end
end
d=det(U);
if d==0
    error('sistem nima enolične rešitve');
end
x(1:n)=0;
x(n)=c(n)/U(n,n);
for i=n-1:-1:1
    x(i)=(c(i)-dot(U(i,i+1:n),x(i+1:n)))/U(i,i);
end
```

```
%vaja7.4
%Resevanje sistema linearnih enačb - Gauss
clc;
clear all;
A=[ 1  1  0  3; 2  1 -1  1; 3 -1 -1  2; -1  2  3 -1 ];
b=[4 1 -3 4]';
A=[A,b];
%podprogram - reševanje z Gaussovo metodo
x=gauss(A);
[n,m]=size(A);
for i=1:n
    fprintf('x(%1i) = %6.3f \r',i,x(i));
end
%test: x(1)=-1, x(2)=2, x(3)=0, x(4)=1

%podprogram - Gaussova metoda
function x=gauss(A)
[n,m]=size(A);
%postopek primitivne Gaussove eliminacije
for i=1:n-1
    for j=i+1:n
        f=A(j,i)/A(i,i);
        %tvorba zgornjega trikotnega sistema
        A(j,i:n+1)=A(j,i:n+1)-f*A(i,i:n+1);
    end
end
%resevanje zgornjega trikotnega sistema
x(n)=A(n,n+1)/A(n,n);
for i=n-1:-1:1
    x(i)=(A(i,n+1)-dot(A(i,i+1:n),x(i+1:n)))/A(i,i);
end
```

1. naloga:

Izračunajte rešitev sistema linearnih enačb

$$1x_1 - 1x_2 + 2x_3 - x_4 = -8 \quad (6)$$
$$2x_1 - 2x_2 + 3x_3 - 3x_4 = -20 \quad (10)$$
$$1x_1 + 1x_2 + 1x_3 \qquad = -2 \quad (-1)$$
$$1x_1 - x_2 + 4x_3 + 3x_4 = 4 \quad (8)$$

po *Gaussovi metodi z delnim pivotiranjem*. Program naj kliče podprogram »sistem_lin_enacb«, obravnavanim na predavanjih.

2. naloga:

Izračunajte rešitev sistema linearnih enačb po *Gauss-Jordanovi metodi*.

$$1x_1 + 1x_2 + 0x_3 + 3x_4 = 4$$
$$2x_1 + 1x_2 - 1x_3 + 1x_4 = 1$$
$$3x_1 - 1x_2 - 1x_3 + 2x_4 = -3$$
$$-1x_1 + 2x_2 + 3x_3 - 1x_4 = 4$$

3. naloga:

Izračunajte rešitev sistema linearnih enačb v prvi nalogi po *LU metodi*. Uporabite Matlabovo funkcijo *linsolve*. Zatem naredite še dekompozicijo matrike A z Matlabovo funkcijo *lu*.

4. naloga

Izračunajte rešitev sistema linearnih enačb po *metodi Cholesky*.

$$6x_1 + 15x_2 + 55x_3 = 82$$
$$15x_1 + 55x_2 + 225x_3 = 170$$
$$55x_1 + 225x_2 + 979x_3 = 502$$

5. naloga

Rešite četrto nalogo z uporabo Matlab-ovih funkcij.

```
%vaja8.1
%Reševanje sistema s funkcijo sistem_lin_en
clc;
A=[ 1 -1  2 -1
    2 -2  3 -3
    1  1  1  0
    1 -1  4  3 ];
b=[-8 -20 -2 4; 6 10 -1 8]' % dve desni strani
pivot=1;                    % pivot=0, pivotiranje izključeno
[X,D,L,U,pog_st,P]=sistem_lin_enacb(A,b,pivot)
%[X,D]=sistem_lin_enacb(A,b);
disp(' Resitev sistema enacb...')
disp(X)
% Resitev sistema enacb...
%  -7.0000   -2.5000
%   3.0000   -1.5000
%   2.0000    3.0000
%   2.0000   -1.0000
```

```
function [X,D,L,U,pog_st,P]=sistem_lin_enacb(A,b,pivotiranje)
%A matrika koeficientov
%b matrika desnih strani
%X rešitev sistema enačb
if (nargin==2)              % pivotiranje je izključeno
    pivotiranje=0;
end
if(nargin==3 & nargout==5)  % št. vhodnih in izhodnih argumentov
    warning(' L U razcep bo lahko neuporaben')
end
B=A;
A=[A,b];              % razširjena matrika A
[m,n]=size(A);        % m = št. vrstic, n = št. stolpcev
P=eye(m);
D=1;                  % začetna vrednost
```

```
for i=1:m-1          % indeks vrstice
    % delno pivotiranje
    if pivotiranje==1
        [e,k]=max(abs(A(i:m,i)));
        k=k+i-1;
        if k~=i
            % zamenjava vrstic
            pv=A(i,i:n);
            A(i,i:n)=A(k,i:n);
            A(k,i:n)=pv;
            D=D*(-1);   % sprememba predznaka
            P1=eye(m);
            pv=P1(:,k);
            P1(:,k)=P1(:,i);
            P1(:,i)=pv;
            P=P1*P;
        end
    end
    if(abs(A(i,i))<1.e-20) % pivotni element je enak 0
        error('matrika koeficientov je singularna')
    end
    % eliminacija - tvorba zgornjega trikotnega sistema
    for j=i+1:m
        L(j,i)=A(j,i)/A(i,i);
        A(j,i)=0;
        A(j,i+1:n)=A(j,i+1:n)-L(j,i)*A(i,i+1:n);
    end
end
D=D*prod(diag(A));  % determinanta
x(1:m)=0;
L=[L,x']+eye(m);    % sestavljanje matrike L
U=A(1:m,1:m);       % opustitev desne strani
% reševanje zgornjega trikotnega sistema z več desnimi stranmi
for k=m+1:n % k števec desnih strani
    for i=m:-1:1
        x(i)=(A(i,k)-dot(A(i,i+1:m),x(i+1:m)))/A(i,i);
    end
    if k==m+1
        X=x'; %postavim rešitev kot stolpec
    else
        X=[X,x']; %dodajam stolpce
    end
end
if nargout>=5
    pog_st=cond(B,2);
else
    pog_st=0;
end
```

```
%vaja8.2
%Reševanje sistema po Gauss-Jordanovi metodi
clc;
A=[  1  1  0  3
     2  1 -1  1
     3 -1 -1  2
    -1  2  3 -1 ];
b=[4 1 -3 4]';
A=[A,b];
[n,m]=size(A);
for i=1:n
    %normiranje
    A(i,i:m)=A(i,i:m)/A(i,i);
    for j=1:n
        if i==j
            continue
        end
        %eliminacija
        A(j,i:m)=A(j,i:m)-A(j,i)*A(i,i:m);
    end
    disp(A)
end
for i=1:n
    fprintf('x(%1i) = %6.3f \r',i,A(i,n+1))
end
%test: x(1)=-1, x(2)=2, x(3)=0, x(4)=1
```

```
%vaja8.3
%Resevanje sistema linearnih enačb - LU metoda
clc;
%podatki
A=[ 1 -1  2 -1
    2 -2  3 -3
    1  1  1  0
    1 -1  4  3 ];
b=[-8 -20 -2 4]';
%podprogram - reševanje z LU metodo
x=linsolve(A,b);
[n,m]=size(A);
%izpis
for i=1:n
    fprintf('x(%1i) = %6.3f \r',i,x(i))
end
%dekompozicija matrike A
[L,U,P]=lu(A)
%test: x(1)=-7, x(2)=3, x(3)=2, x(4)=2
```

```
%vaja8.4
%Reševanje po metodi Choleskega
clc;
clear all;
A=[ 6 15 55; 15 55 225; 55 225 979 ];
b=[82 170 502]';
[n,m]=size(A);
%razcep matrike v spodnjo in zgornjo trikotno matriko
U=zeros(n);
for i=1:n %določa stolpec
    for j=i:n %določa vrstico
        if i==j
            U(i,i)=sqrt(A(i,i)-dot(U(1:i-1,i),U(1:i-1,i)));
        else
            U(i,j)=(A(i,j)-dot(U(1:j-1,i),U(1:j-1,j)))/U(i,i);
        end
    end
end
U
U'*U
%rešujemo spodnji trikotni sistem Lc=b ->c=b/L
L=U'
c(1)=b(1)/L(1,1);
for i=2:n
    c(i)=(b(i)-dot(L(i,1:i-1),c(1:i-1)))/L(i,i);
end
c
%rešujemo zgornji trikotni sistem Ux=c -> x=c/U
x(1:n)=0;
for i=n:-1:1
    x(i)=(c(i)-dot(U(i,i+1:n),x(i+1:n)))/U(i,i);
end
x
%test: x(1)=12, x(2)=8, x(3)=-2
```

```
%vaja8.5
%Reševanje po metodi Choleskega
clc;
%podatki
A=[ 6 15 55; 15 55 225; 55 225 979 ];
b=[82 170 502]';
%razcep v spodnjo in zgornjo trikotno matriko
U=chol(A)
L=U'
L*U
%reševanje spodnjega in zgornjega sistema
c=L\b
x=U\c
%test: x(1)=12, x(2)=8, x(3)=-2
```

1. naloga:

Z metodo konjugiranih gradientov za reševanje pozitivno definitnih simetričnih sistemov rešite sistem enačb.

$$4x_1 - x_2 - x_4 = 0$$
$$-x_1 + 4x_2 - x_3 - x_5 = 5$$
$$-x_2 + 4x_3 - x_6 = 0$$
$$-x_1 + 4x_4 - x_5 = 6$$
$$-x_2 - x_4 + 4x_5 - x_6 = -2$$
$$-x_3 - x_5 + 4x_6 = 6$$

2. naloga

Izračunajte rešitev sistema linearnih enačb

$$6x_1 + 3x_2 - x_3 = -2$$
$$x_1 + 3x_2 - x_3 = -7$$
$$-x_1 - x_2 + 3x_3 = 7$$

z *Gauss-Seidlovo* iterativno metodo.

3. naloga:

Za spodnjo tabelo izračunajte Newtonov interpolacijski polinom. Nato izračunajte vrednost polinoma pri $x = 4.2$.

1.32	1.07
2.42	1.41
3.67	1.73
4.29	2.04
5.96	1.23
6.73	-0.24

4. naloga:

Z uporabo inverzne interpolacije po Newtonovi metodi izračunajte ničlo tabelirane funkcije.

-0.12	0.67
0.41	0.38
1.15	-0.96
1.48	-1.28

5. naloga:

Za tabelo v tretji nalogi izračunajte z Matlab-ovo funkcijo *spline* kubični zlepek brez dodatnih robnih pogojev in zlepek s predpisanimi robnimi odvodi.

```
%vaja9.2
%Reševanje diagonalno dominantnega sistema
%po Gauss-Seidelovi iterativni metodi
clc, clear all
A=[ 6  3 -1
    1  3 -1
   -1 -1  3 ];
b=[-2 -7  7]';
[n,m]=size(A);
x(1:n)=0;
xn=x;
napaka=1.;
while napaka>1.e-4
 for i=1:n
  xn(i)=(b(i)-dot(A(i,1:i-1),xn(1:i-1))-dot(A(i,i+1:n),x(i+1:n)))/A(i,i);
 end
 napaka=norm(x-xn);
 x=xn;
 fprintf('%18.5f%12.5f%12.5f%17.8f\n',x,napaka)
end
%test: x(1)=1, x(2)=-2, x(3)=2
```

```
%vaja9.3
%Newtonova interpolacijska metoda
clc; close all;
x=[ 1.32 2.42 3.67 4.29 5.96 6.73]
y=[ 1.07 1.41 1.73 2.04 1.23 -0.24]
c=polynewtkoef(x,y)
xt=x(1):0.01:x(end);
n=length(xt);
yt=[];
for i=1:n
    yt(i)=polynewtvred(c,x,xt(i));
end
plot(xt,yt,x,y,'b+');
%vrednost polinoma v točki x0
x0=input('Podajte x0: ');
y0=polynewtvred(c,x,x0)
%test: y(4.2)=2.0011
```

```
%Izračun koeficientov interpolacijskega polinoma
function k=polynewtkoef(x,y)
n=length(x);
f(:,1)=y';
for j=2:n
    for i=1:n-j+1
        f(i,j)=(f(i+1,j-1)-f(i,j-1))/(x(i+j-1)-x(i));
    end
end
k=f(1,:);

%Izračun vrednosti polinoma
function p=polynewtvred(k,x,X)
n=length(k);
p=k(n);
for i=n-1:-1:1
    p=k(i)+(X-x(i))*p;
end
```

```
%vaja9.4
%Inverzna interpolacija
clc; close all;
y=[-0.12  0.41  1.15  1.48];
x=[ 0.67  0.38 -0.96 -1.28];
c=polynewtkoef(x,y);
xt=x(1):-0.01:x(end);
n=length(xt);
yt=[];
for i=1:n
    yt(i)=polynewtvred(c,x,xt(i));
end
plot(xt,yt,x,y,'b+');
%vrednost polinoma v točki x0
x0=0;
y0=polynewtvred(c,x,x0)
%test: x0=0.7710
```

```
%vaja9.5
%interpolacija z zlepkom
hold on
x=[ 1.32 2.42 3.67 4.29 5.96 6.73]
y=[ 1.07 1.41 1.73 2.04 1.23 -0.24]
plot(x,y,'r+')
y=[3,y,1]  % predpisani robni odvodi
yp=[]; xp=x(1):0.01:x(end);
yp=[yp,spline(x,y,xp)];
plot(xp,yp)
```

1. naloga:

Za spodnjo tabelo izračunajte po Aitkenovi metodi vrednost polinoma pri $x = 4.2$.

1.32	1.07
2.42	1.41
3.67	1.73
4.29	2.04
5.96	1.23
6.73	-0.24

2. naloga:

Aproksimirajte funkcijsko tabelo

x	$y(x)$
-1.30	0.61
-1.26	0.59
-0.98	0.65
-0.12	0.71
0.81	0.88
1.15	0.96
1.34	1.08
1.36	1.10

s premico $f(x,a,b)=a + b\,x$ v smislu najmanjših kvadratov.

3. naloga:

Aproksimirajte funkcijsko tabelo iz druge naloge s premico. Uporabite Matlab-ovo lastnost, da reši predoločen sistem enačb po metodi najmanjših kvadratov.

4. naloga:

Izračunajte koeficiente aproksimacijske funkcije

$$f(x, r_1, r_2, r_3, r_4) = r_1 + r_2\,x + r_3 \cos(2x) + r_4\, e^x$$

na osnovi tabele iz druge naloge. Nato jo narišite po koraku 0.01, ter izračunajte tabelo odstopkov v tabeliranih točkah in vsoto kvadratov odstopkov v tabeliranih točkah.

5. naloga:

Aproksimirajte funkcijsko tabelo iz druge naloge z nelinearno funkcijo

$$f(x, r_1, r_2, r_3, r_4, r_5) = r_1 + r_2\, x\, r_3\, x^2 + r_4\, e^{r5x2}$$

s pomočjo Matlab-ove funkcije *lsqcurvefit(f,r_0,x,y)*. Začetne vrednosti koeficientov so $r_1=0.7$, $r_2=0.16$, $r_3=-0.03$, $r_4=0.08$ in $r_5=0.54$.

6. naloga:

Aproksimirajte funkcijsko tabelo v drugi nalogi z nelinearno funkcijo $f(x,a,b)=a\, e^{bx}$.

7. naloga:

Uporabite Matlab-ovo orodje na plot-u (Tools -> Basic Fitting).

8. naloga:

Aproksimirajte funkcijsko tabelo v drugi nalogi s polinomom. Uporabite Matlab-ovo funkcijo *polyfit*. Aproksimirajte s polinomi prve, druge, tretje, četrte in sedme stopnje.

```
%vaja10.1
%Aitkenova interpolacijska metoda
clc;
x=[ 1.32 2.42 3.67 4.29 5.96 6.73]
y=[ 1.07 1.41 1.73 2.04 1.23 -0.24]
x0=input('Podajte x0: ');
[p,dp]=aitken(x,y,x0)
%test: y(4.2)=2.0011 napaka=0.0056

%Funkcija za Aitkenovo interpolacijsko metodo
function [p,dp]=aitken(x,y,X)
n=length(x);
a(:,1)=y;
a(:,n+1)=x-X;
for j=2:n
  for i=j:n
    a(i,j)=(a(j-1,j-1)*a(i,n+1)-a(i,j-1)*a(j-1,n+1))/(x(i)-x(j-1));
  end
end
a
p=a(n,n); %vrednost polinoma
if nargout==2
     dp=a(n,n)-a(n-1,n-1); %ocena napake
end;
```

```
%vaja10.2
%primer aproksimacije s premico
clc;
clear all;
x=[-1.3 -1.26 -0.98 -0.12  0.81  1.15  1.34  1.36]
y=[ 0.61 0.59  0.65  0.71  0.88  0.96  1.08  1.10]
n=length(x);
h1=inline('x.^0','x');
h2=inline('x','x');
v1=h1(x);
v2=h2(x);
a(1,1)=dot(v1,v1);
a(1,2)=dot(v1,v2);
b(1)=dot(y,v1);
a(2,1)=a(1,2);
a(2,2)=dot(v2,v2);
b(2)=dot(y,v2);
a
b
r=linsolve(a,b')
xv=x(1):0.01:x(end);
yv=r(1)*h1(xv)+r(2)*h2(xv);
plot(xv,yv,x,y,'r+')
%test: r1=0.8010 r2=0.1722
```

```
%vaja10.3
%primer aproksimacije s premico
%sistem veliko enačb za malo neznank reši Matlab
%v smislu metode najmanjših kvadratov
clc;
clear all;
x=[-1.3 -1.26 -0.98 -0.12  0.81  1.15  1.34  1.36]
y=[ 0.61 0.59  0.65  0.71  0.88  0.96  1.08  1.10]
h1=inline('x.^0','x')
h2=inline('x','x')
A=[h1(x') h2(x')]
b=y'
r=A\y'
xv=x(1):0.01:x(end);
yv=r(1)*h1(xv)+r(2)*h2(xv);
plot(xv,yv,x,y,'r+')
```

```
%vaja10.4
%primer aproksimacije z linearno kombinacijo funkcij
%sistem veliko enačb za malo neznank reši Matlab
%v smislu metode najmanjših kvadratov
clc;
clear all;
x=[-1.3 -1.26 -0.98 -0.12 0.81 1.15 1.34 1.36];
y=[ 0.61 0.59 0.65 0.71 0.88 0.96 1.08 1.10];
h1=inline('x.^0','x');
h2=inline('x','x');
h3=inline('cos(2.*x)','x');
h4=inline('exp(x)','x');
A=[h1(x'),h2(x'),h3(x'),h4(x')];
r=A\y'
xv=x(1):0.01:x(end);
yv=r(1)*h1(xv)+r(2)*h2(xv)+r(3)*h3(xv)+r(4)*h4(xv);
plot(xv,yv,x,y,'r+')
%tabela odstopkov
t(:,1)=x;
t(:,2)=y;
t(:,3)=r(1)*h1(x)+r(2)*h2(x)+r(3)*h3(x)+r(4)*h4(x);
t(:,4)=t(:,2)-t(:,3);
t
%vsota kvadratov odstopkov
s=sum(t(:,4).^2)
% test: r1=0.5236, r2=-0.0530, r3=0.0370, r4=0.1725
```

```
%vaja10.5
% primer nelinearne aproksimacije
clc; clear all;
x=[-1.3 -1.26 -0.98 -0.12  0.81  1.15  1.34  1.36]
y=[ 0.61 0.59  0.65  0.71  0.88  0.96  1.08  1.10]
xv=x(1):0.01:x(end);
r0=[0.7,0.16,-0.03,0.08,0.54]
f=inline('r(1)+r(2).*x+r(3).*x.*x+r(4)*exp(r(5).*x.*x)','r','x')
[r,raz]=lsqcurvefit(f,r0,x,y)
yv=f(r,xv);
plot(xv,yv,x,y,'ro')
```

```
%vaja10.6
%primer aproksimacije z eksponentno funkcijo
clc; clear all;
x=[-1.3 -1.26 -0.98 -0.12  0.81  1.15  1.34  1.36]
y=[ 0.61 0.59  0.65  0.71  0.88  0.96  1.08  1.10]
lny=log(y)
n=length(x);
h1=inline('x.^0','x');
h2=inline('x','x');
v1=h1(x);
v2=h2(x);
a(1,1)=dot(v1,v1);
a(1,2)=dot(v1,v2);
b(1)=dot(lny,v1);
a(2,1)=a(1,2);
a(2,2)=dot(v2,v2);
b(2)=dot(lny,v2);
a
b
r=linsolve(a,b')
xv=x(1):0.01:x(end);
yv=exp(r(1))*exp(r(2)*h2(xv));
plot(xv,yv,x,y,'r+')
```

```
%vaja10.7
clc;
clear all;
x=[-1.3 -1.26 -0.98 -0.12  0.81  1.15  1.34  1.36]
y=[ 0.61 0.59  0.65  0.71  0.88  0.96  1.08  1.10]
plot(x,y,'ro');
```

```
%vaja10.8
%primer aproksimacije s polinomom
clc; clear all;
x=[-1.3 -1.26 -0.98 -0.12  0.81  1.15  1.34  1.36]
y=[ 0.61 0.59  0.65  0.71  0.88  0.96  1.08  1.10]
xv=x(1):0.01:x(end);
m=1; % linearna funkcija
%m=2; % kvadratna parabola
%m=3; % kubična parabola
%m=4; % polinom četrtega reda
%m=7; % interpolacija s polinomom sedmega reda
p=polyfit(x,y,m)
pv=polyval(p,xv);
plot(x,y,'b+',xv,pv,'g-');
```

<u>1. naloga:</u>
Izračunajte koren nelinearne enačbe

$$x + \sin x = e^{-x}$$

na tri mesta natančno z *izboljšano iterativno metodo*.

<u>2. naloga:</u>
Izračunajte koren nelinearne enačbe v prvi nalogi na tri mesta natančno z *Newtonovo (tangentno) metodo*.

<u>3. naloga:</u>
Izračunajte koren nelinearne enačbe v prvi nalogi na tri mesta natančno s *Sekantno metodo*.

<u>4. naloga:</u>
Izračunajte koren nelinearne enačbe v prvi nalogi na tri mesta natančno z *Bisekcijsko metodo*.

<u>5. naloga:</u>
Izračunajte koren nelinearne enačbe v prvi nalogi s pomočjo Matlab-ove funkcije *fzero*.

<u>6. naloga:</u>
Poiščite ničle polinoma

$$2x^4 + 7x^3 - 4x^2 + 29x + 14 = 0$$

z uporabo Matlab-ove funkcije *roots*. Zatem določite še polinom v standardni obliki z uporabo funkcije *poly*.

7. naloga:

Rešite sistem dveh nelinearnih enačb

$$x^2 + y^2 = 16$$
$$x^2 + 16\,y^2 = 64$$

z *Newtonovo metodo* na pet točnih cifer.

8. naloga:

Rešite sistem dveh nelinearnih enačb v sedmi nalogi z Matlab-ovo funkcijo *fsolve*.

```
%vaja11.1
%Izboljšana iterativna metoda
clc;
f=inline('exp(-x)-x-sin(x)','x');
fc=inline('-exp(-x)-1-cos(x)','x');
x0=input('Podaj začetni približek: ');
n=input('Podaj število korakov: ');
eps=input('Podaj dopustno relativno napako: ');
k=fc(x0) %vrednost odvoda v začetnem približku
for i=1:n
    x1=x0-f(x0)/k
    e=abs((x1-x0)/x1);
    if e<eps
        break
    end
    x0=x1;
end
if i==n
    fprintf('REZULTAT JE NAROBE...')
end
x2=x1-f(x1)/k;
L=(x2-x1)/(x1-x0)
ocena_nap=L/(1-L)*abs(x2-x1);
fprintf('koren=%15.8f \n',x2);
fprintf('ocena napake=%15.10f \n',ocena_nap);
fprintf('število korakov=%10.0f \n',i);
fprintf('funkcijska vrednost=%12.4f\n',f(x2));
%test: x0=0.4, n=100, eps=0.0001
%koren = 0.35446311
```

```
%vaja11.2
%Newtonova (tangentna) metoda
clc;
f=inline('exp(-x)-x-sin(x)','x');
fc=inline('-exp(-x)-1-cos(x)','x');
x0=input('Podaj začetni približek: ');
n=input('Podaj število korakov: ');
eps=input('Podaj dopustno relativno napako: ');
for i=1:n
    x1=x0-f(x0)/fc(x0)
    e=abs((x1-x0)/x1);
    if e<eps
        break
    end
    x0=x1;
end
if i==n
    fprintf('REZULTAT JE NAROBE...')
end
x2=x1-f(x1)/fc(x1);
L=(x2-x1)/(x1-x0)
ocena_nap=L/(1-L)*abs(x2-x1);
fprintf('koren=%15.12f \n',x2);
fprintf('ocena napake=%15.12f \n',ocena_nap);
fprintf('število korakov=%10.0f \n',i);
fprintf('funkcijska vrednost=%12.4f\n',f(x2));
%test: x0=0.4, n=100, eps=0.0001
%koren = 0.354463104375
```

```
%vaja11.3
%Sekantna metoda
clc;
f=inline('exp(-x)-x-sin(x)','x');
x0=input('Podaj prvi začetni približek: ');
x1=input('Podaj drugi začetni približek: ');
n=input('Podaj število korakov: ');
eps=input('Podaj dopustno relativno napako: ');
for i=1:n
    x2=x1-f(x1)*(x0-x1)/(f(x0)-f(x1))
    e=abs((x2-x1)/x2);
    if e<eps
        break
    end
    x0=x1;
    x1=x2;
end
if i==n
    fprintf('REZULTAT JE NAROBE...')
end
L=(x2-x1)/(x1-x0)
ocena_nap=L/(1-L)*abs(x2-x1);
fprintf('koren=%15.12f \n',x2);
fprintf('ocena napake=%15.12f \n',ocena_nap);
fprintf('število korakov=%10.0f \n',i);
fprintf('funkcijska vrednost=%12.4f\n',f(x2));
%test: x0=0.3, x1=0.4, n=100, eps=0.0001
%koren = 0.354463104815
```

```
%vaja11.4
%Bisekcijska metoda
clc; clear all
f=inline('exp(-x)-x-sin(x)','x');
xl=input('Podaj levi približek: ');
xd=input('Podaj desni približek: ');
if f(xl)*f(xd)>0
    error ' f(xl)*f(xd) mora biti negativen'
end
n=input('Podaj število korakov: ');
eps=input('Podaj dopustno relativno napako: ');
for i=1:n
    xn=(xl+xd)/2
    if f(xn)==0
        break
    end
    if f(xl)*f(xn)<0
        xd=xn
    else
        xl=xn
    end
    e=abs((xd-xl)/(xd+xl));
    if e<eps
        break
    end
end
if i==n
    fprintf('REZULTAT JE NAROBE...')
end
fprintf('koren=%15.12f \n',xn)
fprintf('ocena rel. napake=%15.12f \n',e)
fprintf('število korakov=%10.0f \n',i)
fprintf('funkcijska vrednost=%12.4f\n',f(xn))
%test: xl=0.3, xd=0.4, n=100, eps=0.0001
%koren = 0.354443359375
```

```
%vaja11.5
%uporaba funkcije fzero
clc;
f=inline('exp(-x)-x-sin(x)','x');
xl=input('Podaj levi približek: ');
xd=input('Podaj desni približek: ');
[x,fv]=fzero(f,[xl,xd]);
if x==nan
    error 'Korena ne znam izračunati...'
end
fprintf('koren=%15.12f \n',x)
fprintf('funkcijska vrednost=%12.8f\n',fv)
%test: xl=0.3, xd=0.4
%koren = 0.354463104375
```

```
%vaja11.6
%Računanje korena polinoma
clc;
%polinom 2x^4+7x^3-4x^2+29x+14=0
c=[2 7 -4 29 14];
r=roots(c)
k=poly(r)
```

```
%vaja11.7
%Newtonova metoda za iskanje ničel dveh nelinearnih enačb
clc; clear all
f1=inline('x^2+y^2-16','x','y');
f2=inline('x^2+16*y^2-64','x','y');
f1x=inline('2*x','x','y');
f1y=inline('2*y','x','y');
f2x=inline('2*x','x','y');
f2y=inline('32*y','x','y');
x0(1)=input('Podaj začetni približek za x: ');
x0(2)=input('Podaj začetni približek za y: ');
eps=input('Podaj natančnost: ');
n=input('Podaj max. število iteracij: ');
for i=1:n
    fprintf('%4i %12.6f %12.6f \n',i,x0(1),x0(2));
    a(1,1)=f1x(x0(1),x0(2));
    a(1,2)=f1y(x0(1),x0(2));
    b(1)  =-f1(x0(1),x0(2));
    a(2,1)=f2x(x0(1),x0(2));
    a(2,2)=f2y(x0(1),x0(2));
    b(2)  =-f2(x0(1),x0(2));
    dx=linsolve(a,b');
    x1=x0+dx';
    e=abs(dx(1)/x1(1))+abs(dx(2)/x1(2));
    if e<eps
        break
    end
    x0=x1;
end
if i==n
    fprintf('REZULTAT JE NAROBE... \n');
end
fprintf('Rešitev sistema: x = %12.6f y = %12.6f \n',x1(1),x1(2));
fprintf('Dosežena natančnost: %12.6e\n',e);
% Test: x0=1.5, y0=3.5, eps=0.000005, n=10
% Rešitev sistema:    x = 3.577709   y = 1.788854
% Dosežena natančnost:   1.263924e-006
```

```
%vaja11.8
%Uporaba funkcije »fsolve« za iskanje ničel dveh nelinearnih enačb
clc;
clear all;
x0(1)=input('Podaj začetni približek za x: ');
x0(2)=input('Podaj začetni približek za y: ');
x=fsolve(@fun,x0);
fprintf('Rešitev sistema: x = %12.6f y = %12.6f \n',x(1),x(2));
% Test: x0=1.5, y0=3.5
% Rešitev sistema:    x = 3.577709   y = 1.788854


%funkcija vektor
function f=fun(x)
f(1)=x(1)^2+x(2)^2-16;
f(2)=x(1)^2+16*x(2)^2-64;
```

<u>1. naloga:</u>

Dana je tabela

x	$y(x)$
0.0	0.0
0.5	0.479
1.0	0.841
1.5	0.997
2.0	0.909

Izračunajte odvode $f'(0)$, $f''(0)$, $f'(1.0)$ in $f''(1.0)$.

<u>2. naloga:</u>

Tabelirajte funkcijo

$$f(x) = e^{-x} + x + sin(x)$$

po koraku $h = 0.1$ od 0 do 2. Nato izračunajte numerični prvi odvod po centralno diferenčni metodi. Izračunan odvod primerjajte z analitičnim.

<u>3. naloga:</u>

Izračunajte določeni integral

$$I = \int_1^4 x^2 \cdot \ln(x)dx$$

po *trapecni metodi* pri delitvi n.

<u>4. naloga:</u>

Izračunajte določeni integral v tretji nalogi po *trapecni metodi* pri predpisani dopustni napaki ε.

5. naloga:

Izračunajte določeni integral v tretji nalogi po *Simpsonovi metodi* pri delitvi n.

6. naloga:

Izračunajte določeni integral v tretji nalogi z Matlabovo funkcijo »*quad*« po adaptivni Simpsonovi metodi pri predpisani dopustni napaki ε.

7. naloga:

Izračunajte določeni integral v tretji nalogi po *Rombergovi metodi* pri predpisani dopustni napaki ε.

```
%vaja12.1
%Računanje odvoda tabelirane funkcije
clc;
clear all;
st=load('tabela.txt');
x=st(:,1);
y=st(:,2);
h=x(2)-x(1);
i=1; % tocka x=0
%prvi odvod - diferenca naprej
dy1=(y(i+1)-y(i))/h
%drugi odvod - diferenca naprej
dy2=(2*y(i)-5*y(i+1)+4*y(i+2)-y(i+3))/(h^2)
i=3; % tocka x=1.0
%prvi odvod - centralna diferencna metoda
dy1=(y(i+1)-y(i-1))/(2*h)
%drugi odvod - centralna diferencna metoda
dy2=(y(i-1)-2*y(i)+y(i+1))/(h^2)
%test: dy1(0)=0.9580, dy2(0)=-0.1120
%      dy1(1)=0.5180, dy2(1)=-0.8240
```

```
%vaja12.2
%Računanje odvoda funkcije
clc; clear all
f=inline('exp(-x)+x+sin(x)','x');
df=inline('-exp(-x)+1+cos(x)','x');
% tabeliranje funkcije
h=0.1;
x=[0:h:2];
y=f(x);
n=length(y)
%numerični odvod
odv=diff(y)/h;
n=length(odv) %krajša za en element
odv=[odv,odv(end)];
%analitični odvod
dy=df(x);
%primerjava v grafu
plot(x,dy,x,odv,'r--');
```

```
%vaja12.3
%Trapecna metoda
clc; clear all;
f=inline('x.^2.*log(x)','x');
a=input('Podaj spodnjo mejo integrala: ');
b=input('Podaj zgornjo mejo integrala: ');
n=input('Podaj število delitvenih točk: ');
x=linspace(a,b,n);
y=f(x);
%sestavljena trapecna formula
h=(b-a)/(n-1);
ti=y(1)+y(n);
for i=2:n-1
    ti=ti+2*y(i);
end
ti=ti*h/2;
mi=trapz(x,y); %Matlabova funkcija
%izpis
fprintf('Vrednost integrala %15.10f\n',ti);
fprintf('Vrednost integrala (matlab) %15.10f\n',mi);
% točna vrednost: s=22.5742797039, a=1, b=4

%vaja12.4
%Integral funkcije po trapecni metodi z dopustno napako
clc; clear all;
f=inline('x.^2.*log(x)','x')
a=input('Podaj spodnjo mejo integrala: ');
b=input('Podaj zgornjo mejo integrala: ');
eps=input('Podaj dopustno napako: ');
n=2;
x=linspace(a,b,n);
y=f(x);
TS=trapz(x,y);
for i=1:100
    n=2*n-1;
    x=linspace(a,b,n);
    y=f(x);
    TN=trapz(x,y);
    on=(TN-TS)/3;
    fprintf('%5.0f%5.0f%15.8f%15.8f\n',i,n,TN,on);
    if abs(on)<eps
        break;
    end
    TS=TN;
end
TN=TN+on;
fprintf('Izboljšana trapecna metoda %15.10f\n',TN)
int=quad(f,a,b,eps);
fprintf('Približek integrala je %15.10f\n',int);
fprintf('z natančnostjo %15.2e\n',eps);
% točna vrednost: s=22.5742797039, a=1, b=4
```

```
%vaja12.5
%Simpsonova metoda
clc; clear all;
f=inline('x.^2.*log(x)','x');
a=input('Podaj spodnjo mejo integrala: ');
b=input('Podaj zgornjo mejo integrala: ');
n=input('Podaj število delitvenih točk: ');
if mod(n,2)==0
    error ('Število delitvenih točk mora biti liho');
end
x=linspace(a,b,n);
y=f(x);
%sestavljena simpsonova formula
h=(b-a)/(n-1);
si=y(1)+y(n);
for i=2:2:n-1
    si=si+4*y(i);
end
for i=3:2:n-2
    si=si+2*y(i);
end
si=si*h/3;
%izpis
fprintf('Vrednost integrala %15.10f\n',si);
% točna vrednost: s=22.5742797039, a=1, b=4
```

```
%vaja12.6
%Integriranje z matlabovo
%funkcijo quad po adaptivni Simpsonovi metodi
clc; clear all;
f=inline('x.^2.*log(x)','x');
a=input('Podaj spodnjo mejo integrala: ');
b=input('Podaj zgornjo mejo integrala: ');
eps=input('Podaj dopustno napako: ');
int=quad(f,a,b,eps);
fprintf('Približek integrala %15.10f\n',int);
% točna vrednost: s=22.5742797039, a=1, b=4
```

```
%vaja12.7
%Numerično rač. integral funkcije po
%Rombergovi metodi z dopustno napako
clc; clear all;
f=inline('x.^2.*log(x)','x');
a=input('Podaj spodnjo mejo integrala: ');
b=input('Podaj zgornjo mejo integrala: ');
eps=input('Podaj dopustno napako ');
R=zeros(7,7);
n=2;
x=linspace(a,b,n);
y=f(x);
R(1,1)=trapz(x,y);
for i=2:100
    n=2*n-1;
    x=linspace(a,b,n);
    y=f(x);
    R(i,1)=trapz(x,y);
    for j=2:i
        R(i,j)=(4^(j-1)*R(i,j-1)-R(i-1,j-1))/(4^(j-1)-1);
    end
    if abs(R(i,i)-R(i-1,i-1))<eps
        break
    end
end
R
fprintf('Integral po Rombergovi metodi = %15.8f\n',R(i,i))
%Kontrola z Matlab funkcijo
int=quad(f,a,b,eps);
fprintf('Približek integrala je %15.10f\n',int);
fprintf('z natančnostjo %15.2e\n',eps);
% točna vrednost: s=22.5742797039, a=1, b=4
```

1. naloga:

Izračunajte vrednosti rešitve diferencialne enačbe

$$y' = y - x^2 + 1$$
$$y(0) = 0.5$$
$$0 \le x \le 2$$

po *Eulerjevi metodi* s korakom $h = 0.2$.

2. naloga:

Izračunajte vrednosti rešitve diferencialne enačbe

$$y' = x - xy$$
$$y(1) = 2$$
$$1 \le x \le 3$$

po metodi *Runge-Kutta 4. reda*, s korakom $h = 0.2$.

3. naloga:

Izračunajte vrednosti rešitve diferencialne enačbe iz prve naloge z Matlabovo funkcijo *ode45*. Funkcija uporablja kombinacijo Runge-Kutta 4. in 5. reda.

4. naloga:

Tabelirajte rešitev robnega problema

$$x^2 y'' + xy' + y = 0$$
$$y(1) = 0$$
$$y(2) = 0.638961$$

s korakom h=0.2 po diferenčni metodi.

5. naloga:

Matriki A izračunajte lastne vektorje in lastne vrednosti.

$$A = \begin{bmatrix} -2 & 2 & -3 \\ 2 & 1 & -6 \\ -1 & -2 & 0 \end{bmatrix}$$

```
%vaja13.1
%Eulerjeva metoda
clc; clear all
x0=input('Podaj x0: ');
y0=input('Podaj y0: ');
h=input('Podaj korak h: ');
b=input('Podaj konec intervala b: ');
f=inline('y-x^2+1','x','y')
x(1)=x0;
y(1)=y0;
n=ceil((b-x0)/h)+1
for i=2:n
    y(i)=y(i-1)+f(x(i-1),y(i-1))*h;
    x(i)=x(i-1)+h;
end
%analitična rešitev
g=inline('(1+x).^2-0.5*exp(x)','x')
X=linspace(x0,b);
Y=g(X);
plot(x,y,'r--',X,Y)
%kontrola
yt=g(x);
for i=1:n
    fprintf('%15.8f%15.8f%15.8f\n',x(i),y(i),yt(i));
end
```

```
%vaja13.2
%Metoda Runge - Kutta 4. reda
clc; clear all
x0=input('Podaj x0: ');
y0=input('Podaj y0: ');
h=input('Podaj korak h: ');
b=input('Podaj konec intervala b: ');
f=inline('x-x*y','x','y')
x(1)=x0;
y(1)=y0;
n=ceil((b-x0)/h)+1
for i=2:n
    k1=h*f(x(i-1),y(i-1));
    k2=h*f(x(i-1)+h/2,y(i-1)+k1/2);
    k3=h*f(x(i-1)+h/2,y(i-1)+k2/2);
    k4=h*f(x(i-1)+h,y(i-1)+k3);
    y(i)=y(i-1)+(k1+2*(k2+k3)+k4)/6;
    x(i)=x(i-1)+h;
end
%analitična rešitev
g=inline('exp(1/2-x.^2./2)+1','x')
X=linspace(x0,b);
Y=g(X);
plot(x,y,'r--',X,Y)
%kontrola
yt=g(x);
for i=1:n
    fprintf('%15.8f%15.8f%15.8f\n',x(i),y(i),yt(i));
end
```

```
%vaja13.3
%Reševanje z Matlabovo funkcijo ode45
clc; clear all
x0=input('Podaj x0: ');
y0=input('Podaj y0: ');
b=input('Podaj konec intervala b: ');
f=inline('y-x^2+1','x','y')
[x,y]=ode45(f,[x0,b],y0);
%analitična rešitev
g=inline('(1+x).^2-0.5*exp(x)','x')
X=linspace(x0,b);
Y=g(X);
plot(x,y,'r--',X,Y)
%kontrola
yt=g(x);
n=length(x)
for i=1:n
    fprintf('%15.8f%15.8f%15.8f\n',x(i),y(i),yt(i));
end
```

```
%vaja13.4
%Reševanje diferencialne enačbe x^2y''+xy'+y=0
%pri robnem pogoju y(1)=0 in y(2)=0.638961
%s korakom h po diferenčni metodi
clc; clear all
%izpeljemo sistem linearnih enačb v tabeliranih točkah
a=1;
b=2;
h=0.2
n=ceil((b-a)/h)+1
x=linspace(a,b,n);
A(1,1)=1;
for i=2:n-1
    A(i,i-1)=2*x(i)^2-h*x(i);
    A(i,i)=-4*x(i)^2+2*h^2;
    A(i,i+1)=2*x(i)^2+h*x(i);
end
A(n,n)=1
B(1:n-1)=0;
B(n)=0.638961
y=A\B';
%analitična rešitev
g=inline('sin(log(x))','x');
X=linspace(a,b,n);
Y=g(X);
plot(X,Y,x,y,'r+')
for i=1:n
    fprintf('%2i%15.8f%15.8f%15.8f\n',i,x(i),y(i),Y(i));
end
```

```
%vaja13.5
%Računanje lastne vrednosti in
%lastnih vektorjev matrike A
%z [X,D]=eig(A)
clc; clear all
A=[ -2  2 -3
     2  1 -6
    -1 -2  0]
[x,d]=eig(A);
x %matrika stolpcev lastnih vektorjev
diag(d) %na diagonali so lastne vrednosti
```