

Nelinearne enačbe

10. VAJA

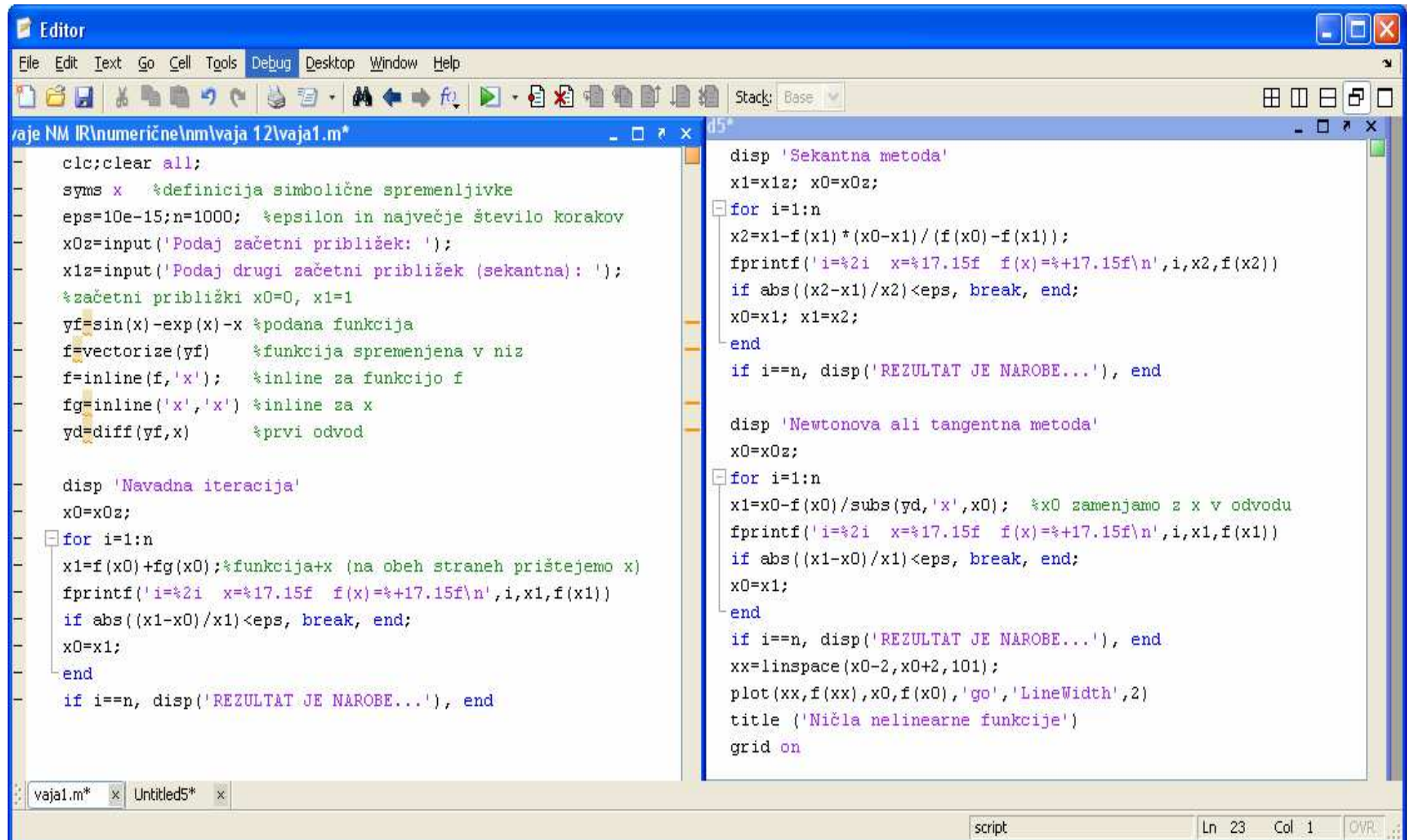
10.1 Odprte metode

Poišči ničlo nelinearne funkcije po sledečih metodah na petnajst mest natančno:

- navadna iteracija
- sekantna metoda
- Newtonova ali tangentna metoda

$$\sin(x) - \exp(x) - x = 0$$

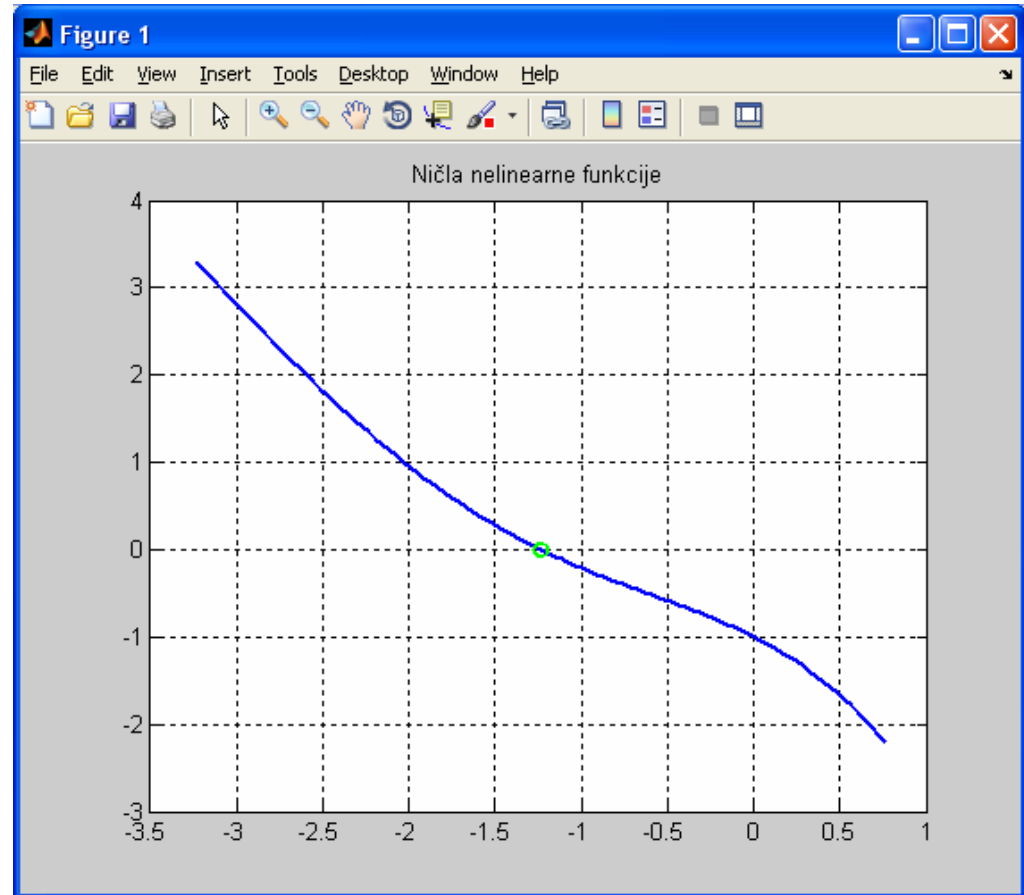
10.1 Odprte metode



```
Editor
File Edit Text Go Cell Tools Debug Desktop Window Help
vaje NM IR\numerične\m\ vaja 12\ vaja1.m*
- clc;clear all;
- syms x %definicija simbolične spremenljivke
- eps=10e-15;n=1000; %epsilon in največje število korakov
- x0z=input('Podaj začetni približek: ');
- x1z=input('Podaj drugi začetni približek (sekantna): ');
- %začetni približki x0=0, x1=1
- yf=sin(x)-exp(x)-x %podana funkcija
- f=vectorize(yf) %funkcija spremenjena v niz
- f=inline(f,'x'); %inline za funkcijo f
- fg=inline('x','x') %inline za x
- yd=diff(yf,x) %prvi odvod
-
- disp('Navadna iteracija')
- x0=x0z;
- for i=1:n
- x1=f(x0)+fg(x0);%funkcija+x (na obeh straneh prištejemo x)
- fprintf('i=%2i x=%17.15f f(x)=%+17.15f\n',i,x1,f(x1))
- if abs((x1-x0)/x1)<eps, break, end;
- x0=x1;
- end
- if i==n, disp('REZULTAT JE NAROBE...'), end
-
- disp('Sekantna metoda')
- x1=x1z; x0=x0z;
- for i=1:n
- x2=x1-f(x1)*(x0-x1)/(f(x0)-f(x1));
- fprintf('i=%2i x=%17.15f f(x)=%+17.15f\n',i,x2,f(x2))
- if abs((x2-x1)/x2)<eps, break, end;
- x0=x1; x1=x2;
- end
- if i==n, disp('REZULTAT JE NAROBE...'), end
-
- disp('Newtonova ali tangentna metoda')
- x0=x0z;
- for i=1:n
- x1=x0-f(x0)/subs(yd,'x',x0); %x0 zamenjamo z x v odvodu
- fprintf('i=%2i x=%17.15f f(x)=%+17.15f\n',i,x1,f(x1))
- if abs((x1-x0)/x1)<eps, break, end;
- x0=x1;
- end
- if i==n, disp('REZULTAT JE NAROBE...'), end
- xx=linspace(x0-2,x0+2,101);
- plot(xx,f(xx),x0,f(x0),'go','LineWidth',2)
- title('Ničla nelinearne funkcije')
- grid on
-
- vaja1.m* x Untitled5* x
script Ln 23 Col 1 OVR
```

10.1 Odprte metode

```
MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
C:\vaje NM IR\nc
Shortcuts
Navadna iteracija
i= 1 x=-1.000000000000000 f(x)=-0.209350425979339
i= 2 x=-1.209350425979339 f(x)=-0.024427109529350
i= 3 x=-1.233777535508688 f(x)=-0.001157648700220
i= 4 x=-1.234935184208909 f(x)=-0.000045271818036
i= 5 x=-1.234980456026944 f(x)=-0.000001752649474
i= 6 x=-1.234982208676418 f(x)=-0.000000067825008
i= 7 x=-1.234982276501426 f(x)=-0.000000002624690
i= 8 x=-1.234982279126115 f(x)=-0.000000000101570
i= 9 x=-1.234982279227685 f(x)=-0.000000000003931
i=10 x=-1.234982279231616 f(x)=-0.000000000000152
i=11 x=-1.234982279231768 f(x)=-0.000000000000006
i=12 x=-1.234982279231774 f(x)=-0.000000000000000
Sekantna metoda
i= 1 x=-0.532818745896949 f(x)=-0.562092795633881
i= 2 x=-0.905039619544713 f(x)=-0.285935864867394
i= 3 x=-1.290441127059897 f(x)=+0.054334530349377
i= 4 x=-1.228900040320808 f(x)=-0.005834808117790
i= 5 x=-1.234867871140135 f(x)=-0.000109976458774
i= 6 x=-1.234982515716459 f(x)=+0.000000227333233
i= 7 x=-1.234982279222581 f(x)=-0.000000000008838
i= 8 x=-1.234982279231774 f(x)=+0.000000000000000
i= 9 x=-1.234982279231774 f(x)=+0.000000000000000
Newtonova ali tangentna metoda
i= 1 x=-1.000000000000000 f(x)=-0.209350425979339
i= 2 x=-1.252967870967838 f(x)=+0.017395848200716
i= 3 x=-1.235092079838279 f(x)=+0.000105555487574
i= 4 x=-1.234982283328451 f(x)=+0.000000003938143
i= 5 x=-1.234982279231774 f(x)=+0.000000000000000
i= 6 x=-1.234982279231774 f(x)=+0.000000000000000
>>
```



10.2 Zaprte metode

Poišči ničlo iste nelinearne funkcije kot v nalogi 10.1 po sledečih metodah na petnajst mest natančno:

- bisekcijska metoda
- metoda regula falsi

$$\sin(x) - \exp(x) - x = 0$$

Za začetni približek vzemi -2 in 0.

10.2 Zaprte metode

```
C:\vaje NM IR\numerične\nm\vaja 12\vaja2.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 %Bisekcijska metoda in regula falsi metoda
2 function glavni
3 clc; clear all
4 eps=10e-15;n=1000; %epsilon in največje število korakov
5 f=inline('sin(x)-exp(x)-x','x');
6 g=inline('(x1+xd)/2','x1','xd','fx1','fxd'); %bisekcija
7 h=inline('(x1*fxd-xd*fx1)/(fxd-fx1)','x1','xd','fx1','fxd');%regula falsi
8 x1=input('Podaj levi približek: ');
9 xd=input('Podaj desni približek: ');
10 if f(x1)*f(xd)>0
11 error('f(x1)*f(xd) mora biti negativen')
12 end
13 disp(' ');disp('Bisekcija'); xn=koren(g,f,n,eps,x1,xd);
14 disp(' ');disp('Regula falsi');xn=koren(h,f,n,eps,x1,xd);
15 end
16 function xn=koren(fun,f,n,eps,x1,xd)
17 for i=1:n
18 xn=fun(x1,xd,f(xd),f(x1));
19 if f(xn)==0, break, end
20 if f(x1)*f(xn)<0
21 xd=xn;
22 else
23 x1=xn;
24 end
25 e=abs((xd-x1)/(xd+x1));
26 if e<eps, break, end
27 fprintf('i=%2i x=%17.15f f(x)=%+17.15f\n',i,xn,f(xn))
28 end
29 if i==n, disp('REZULTAT JE NAROBE'), end
30 fprintf('koren=%17.15f\n',xn)
31 fprintf('ocena relativne napake=%17.15f\n',e)
32 end
glavni Ln 1 Col 1 OVR
```

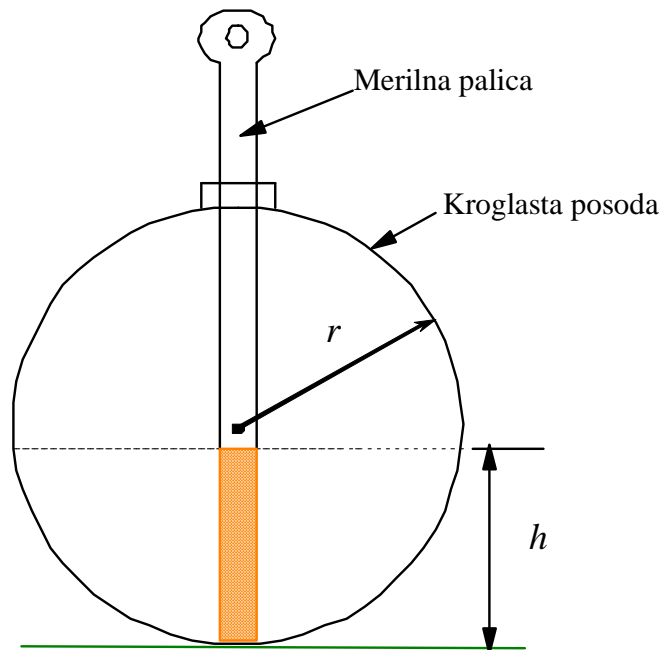
10.3 Matlab funkcije

V kroglasti posodi s polmerom 3m je 4m³ tekočine. Izračunaj višino gladine tekočine, če velja enačba:

$$f(h) = h^3 - 9h^2 + 3.8197 = 0$$

Uporabi matlabova orodja *fzero* in *roots*.

Za začetna približka privzemi 0 in 2r.



10.3 Matlab funkcije

The screenshot displays the MATLAB 7.6.0 (R2008a) environment. The main window is titled 'Editor - C:\vaje NM IR\numerične\nm\vaja 12\vaja3.m*'. The script content is as follows:

```
1 %uporaba funkcije fzero in roots
2 clc;
3 clear all
4 f=inline('x^3-9*x^2+3.8197','x');
5 xs=0; %'Spodnji približek ');
6 xz=6; %'Zgornji približek (2r) ');
7 [x,fv]=fzero(f,[xs,xz]);
8 p=roots([1,-9,0,3.8197]);
9 if x==nan
10     error('Ni rešitve')
11 end
12 fprintf('Višina=%15.12f \n',x)
13 fprintf('funkcijska vrednost=%12.8f\n',fv)
14 disp(' ')
15 disp('Rešitve polinoma so')
16 p
17 disp('Izberemo pozitivno rešitev, ki ni večja od 2r')
18
```

The Command Window on the left shows the execution results:

```
Višina= 0.677464917281
funkcijska vrednost= 0.00000000

Rešitve polinoma so

p =

    8.9523
    0.6775
   -0.6298

Izberemo pozitivno rešitev, ki ni večja od 2r
>>
```

The status bar at the bottom indicates 'script', 'Ln 1', 'Col 33', and 'OVR'.

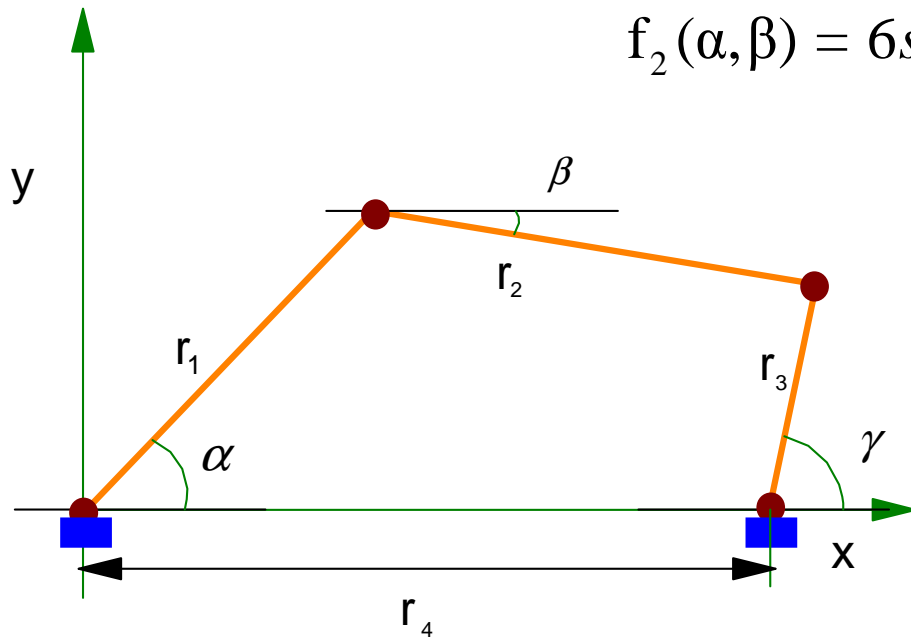
10.4 Sistem nelinearnih enačb

Lega palic v mehanizmu je popolnoma popisana s kotom γ . Izračunaj kota α in β , če je

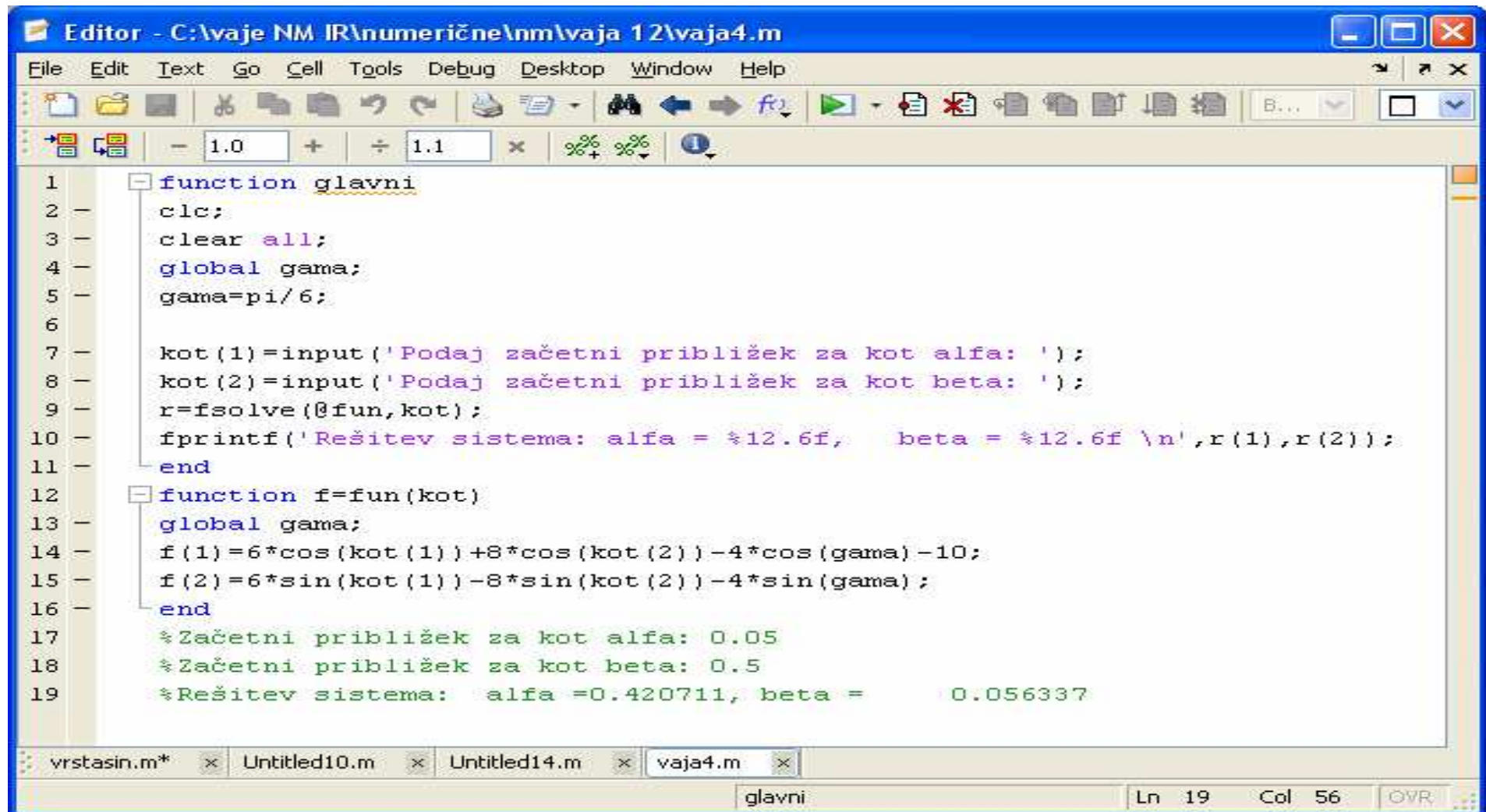
$$r_1=6, r_2=8, r_3=4, r_4=10 \text{ in } \gamma = \pi / 6.$$

$$f_1(\alpha, \beta) = 6\cos(\alpha) + 8\cos(\beta) - 4\cos(\gamma) - 10$$

$$f_2(\alpha, \beta) = 6\sin(\alpha) - 8\sin(\beta) - 4\sin(\gamma)$$



10.4 Sistem nelinearnih enačb



The screenshot shows a MATLAB editor window titled "Editor - C:\vaje NM IR\numerične\nm\vaja 1 2\vaja4.m". The window contains a MATLAB script with the following code:

```
1 function glavni
2 -   clc;
3 -   clear all;
4 -   global gama;
5 -   gama=pi/6;
6
7 -   kot(1)=input('Podaj začetni približek za kot alfa: ');
8 -   kot(2)=input('Podaj začetni približek za kot beta: ');
9 -   r=fsolve(@fun,kot);
10 -   fprintf('Rešitev sistema: alfa = %12.6f,   beta = %12.6f \n',r(1),r(2));
11 -   end
12 function f=fun(kot)
13 -   global gama;
14 -   f(1)=6*cos(kot(1))+8*cos(kot(2))-4*cos(gama)-10;
15 -   f(2)=6*sin(kot(1))-8*sin(kot(2))-4*sin(gama);
16 -   end
17   %Začetni približek za kot alfa: 0.05
18   %Začetni približek za kot beta: 0.5
19   %Rešitev sistema:  alfa =0.420711, beta =      0.056337
```

The script defines a function `glavni` that prompts the user for initial values of `alfa` and `beta`, then uses `fsolve` to solve the system of nonlinear equations. The function `fun` defines the equations. The output of the script is shown at the bottom of the window: `Rešitev sistema: alfa =0.420711, beta = 0.056337`. The status bar at the bottom indicates the current line is 19, column 56, and the file is named `glavni`.