

Zadatak19

```
type Zadatak19/funkcija.m
```

```
%funkcija.m

f=@(x) cos(x)-x;

%Test: Njutn(0,0,1e-5)
%Tacna vrednost: fzero(f,0)
%Njutn(0,0,1e-500) bice dostignut maksimalan broj iteracija
```

```
type Zadatak19/Njutn.m
```

```
function x=Njutn(x0,y,tol)
%Funkcija x=Njutn(x0,y,tol) odredjuje resenje
%nelinearne jednacine f(x)=y koriscenjem Njutnove
%metode.
%
%Ulazni argumenti su:
%   x0 - pocetna vrednost iterativnog niza
%   y - desna strana jednacine f(x)=y
%   tol - tacnost
%
%Izlazni argument:
%   x - vrednost argumenta za koju je f(x)=y
%
%Pomocni fajl:
%   funkcija.m - sadrzi anonimnu funkciju f
%
funkcija

F=sym(f);%konvertujemo funkciju u simbolicku
F1=diff(F);%izvod simbolicke funkcije
f1=matlabFunction(F1);%konvertujemo simbolicku funkciju F1 u anonimnu funkciju

f=@(x) f(x)-y; % trazimo nulu ove funkcije

iter=0;
while iter<100
    x1=x0-f(x0)/f1(x0);
    iter=iter+1;
    if abs(x1-x0)<tol
        break;
    end
    x0=x1;
end
x=x1; % za aproksimaciju uzimamo najnoviju vrednost dobijenu iteracijama
%izasli smo iz petlje zato sto je zadovoljena tacnost
%ili je postignut maksimalan broj iteracija
if iter==100
    disp('Dostignut je maksimalan broj iteracija.')
end
```

```
x=Njutn(0,0,1e-5)
```

```
x =
0.7391
```

```
type Zadatak19/tablica.m
```

```
%tablica.m

%vrednosti su izabrane proizvoljno
Y=1:5;%desna strana jednacine f(x)=y
n=length(Y);
tol=10^(-5);% tolerancija
x0=zeros(1,n);%pocetne vrednosti

X=zeros(1,n);%koreni

for i=1:n
    X(i)=Njutn(x0(i),Y(i),tol);
end
disp(['X: ' num2str(X,5)];['Y: ' num2str(Y,5)])
```

```
tablica
```

```
X: 0      -2.9883      -3.7944      -4.3523      -4.8564
Y: 1              2              3              4              5
```

```
y=vredfunk(-3)
```

```
X: 0      -2.9883      -3.7944      -4.3523      -4.8564
Y: 1              2              3              4              5
y =
2.0100
```

zadatak21

```
type Zadatak21/tablica.m
```

```
X=[0 0.3 0.55 0.60 0.98 1.07 1.27 1.33 1.6 2];
F=[0.2500 0.5455 0.7727 0.8146 1.0805 1.1272 1.2051 1.2211 1.2496 1.1593];
```

```
%Drugi primer
%X=[1:10];
%F=log(X)+3;
%Test: nula(1,1e-4,20)
```

```
type Zadatak21/nula.m
```

```
function [x,briter]=nula(x0,tol,iterM)
%trazimo resenje funk(x)=x
briter=0;
while(briter<iterM)
    x1=funk(x0); %iterativni niz
    briter=briter+1;
    if abs(x0-x1)<=tol
        break
    end
    x0=x1;
end
x=x1; %izasli smo iz petlje jer je sdogvoljena tacnost
%ili je dostignut maksimalan broj iteracija
%u svakom slucaju procena resenja je poslednji clan iterativnog niza
% a to je x=x1
if briter==iterM
    disp('Dostignut je maksimalan broj iteracija!')
end
if abs(x0-x1)<=tol
```

```
disp('Tacnost je zadovoljena!')
end
```

```
[x,briter]=nula(0,1e-4,100)
```

```
Tacnost je zadovoljena!
x =
1.1712
briter =
14
```

```
type grafik.m
```

```
function grafik(x0,iterM)
tol=1e-4:1e-4:1e-3;
n=length(tol);
briter=zeros(1,n);
for i=1:n
    [x, briter(i)]=nula(x0,tol(i),iterM);
end
plot(tol,briter)
xlabel('Tolerancija');
ylabel('Broj iteracija');

title('Zavisnost brzine konvergencije od tacnosti tol');
```

```
grafik(0,100)
```

```
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
Tacnost je zadovoljena!
```

