

```

% helpUNM_2
% nadam se da ce vam olaksati pripremu za drugi kolokvijum

% funkcija je data eksplicitno
f = 'sin(x)';

% vrednost funkcije f u tacki x
vf_x = feval(inline(f),x); % feval vraca vrednosti inline
funkcija, % kako f nije u startu definisana kao
% inline, dodefinisemo je...

% izvod funkcije f
fdf = diff(f);

% vrednost I izvoda funkcije f u tacki x
dvf_x = feval(inline(df),x);

% ZADATAK:
% Naci nulu funkcije f(x)=0 ! // odnosno naci x* za koje vazi f(x*)=0;

%-----
%-----

% NJUTNOVA METODA
% f(x)=c;
% x0 = a;
% eps = tacnost;
% kriterijum zaustavljanja abs(f(xi)-c)<eps
% vase svi uslovi metode

function x = Njutn(a,c,eps)

X = [];
X = [X a];
brIteracija=1;

while (abs(vredFunkcije(X(end))-c)>eps) && (brIteracija<50)
    X = [X X(end)-(vredFunkcije(X(end))-c)/diferencijal(X(end))];
    brIteracija = brIteracija+1;
end;

disp('Nula funkcije f(x)=c je ')
x = X(end);

% NAPOMENA: Ako je funkcija data eksplicitno: diferencijal i
vredFunkcije
% dobijate koristeći vf_x i dvf_x ..
% Ukoliko je funkcija data tablicno:
% vrednost funkcije nalazimo koristeći neki interpolacioni polinom
% vrednost prvog izvoda funkcije takodje dobijamo koristeći neki
interpolacioni polinom

```

```

% METODA POLOVLJENJA INTERVALA

% f(x)=0;
    % a,b = krajevi intervala u kome trazimo nulu funkcije;
    % eps = tacnost;
    % vase svi uslovi metode

function y=polov(a,b,eps)

while (b-a)>eps
    if funk(a)==0
        y = a;
        b = a; % ili break;
    end;
    if funk(b) == 0
        y = b;
        b = a; % ili break;
    end;

    c = (a+b)/2;
    if funk(c) == 0
        y = c;
        b = a; % ili break;
    end;
    if funk(a)*funk(c)<0
        b=c;
    else
        if funk(c)*funk(b)<0
            a=c;
        end;
    end;
end;

% NAPOMENA: Mozda bi mogli u okviru while petlje da dodamo i broj
iteracije (i naravno vodimo racuna da ih ima ne vise od 50)

```

```

% METODA PROSTE ITERACIJE

% x=f(x);
    % a,b = krajevi intervala u kome trazimo nulu funkcije;
    % eps = tacnost;
    % vase svi uslovi metode
    % kriterijum zaustavljanja abs(x(n-1)-x(n))<eps

function [nulaf brIteracija] = nula(eps)

tablica;
brIteracija = 1;
x = [];          % x konvergira ka x* takvo da je x*=f(x*);
x = [x X(1)];   % x0 biramo proizvoljno
xs = x(end);   % obzirom da uporedjujemo vr x u poslednje dve iteracije
                % zapamticemo ih kao xs (x staro) i xn (x novo)
x = [x funk(xs)];
xn = x(end);
while (abs(xn-xs)>eps) && (brIteracija <50)
    xs = xn;
    x = [x funk(xs)];
    xn = x(end);
    brIteracija = brIteracija +1;
end;

%disp('Nula funkcije x=f(x) je: ');
nulaf = xn;
disp(nulaf);

disp('Iterativni niz je :')
disp(x');

% NAPOMENA: u zadatku se trazi nula funkcije x = f(x). Na vezbama smo
prvo funkciju f(x)=0 preveli u funkciju x = g(x) pa onda koristili
metodu... Ovde smo u startu dobili pogodan oblik za metodu.
Kada budete testirali zadatak vodite racuna da vam funkcija f(x) bude
kontrakcija!!!

```

```

% METODA REGULA FALSI

% f(x)=0;
    % eps = tacnost;
    % vaze svi uslovi metode
    % kriterijum zaustavljanja abs(fx(n))<eps

function nulaf = nula(eps)

tablica;
brIteracija = 1;
x = [];
x = [x X(1)]; % x0 biramo proizvoljno
xF = X(end); % xF biramo proizvoljno
f_xF = funk(xF);
xn = x(end);

while (abs(funk(xn))>eps) && (brIteracija <50)
    x = [x xn-(funk(xn)/(f_xF-funk(xn)))*(xF-xn)];
    xn = x(end);
    brIteracija = brIteracija +1;
end;

disp('Nula funkcije x=f(x) je: ');
nulaf = xn;
disp(nulaf);

disp('Iterativni niz je :')
disp(x');

% METODA SECICE
% f(x)=0;

function nulaf = nula(eps)

tablica;
brIteracija = 1;
x = [];
x = [x X(1)]; % x0 biramo proizvoljno
x = [x X(end)]; % x1 biramo proizvoljno

xn = x(end); % ovaj korak je suvisan,
              % uvodimo ga radi lakse preglednosti
              % mozemo samo da pamtimo xs i xn i radimo sa njima
while (abs( funk(xn) )>eps) && (brIteracija <50)
    i = length(x);
    xn_1=x(i-1);
    x = [x xn-(funk(xn)/(funk(xn_1)-funk(xn) ) )*(xn_1-xn)];
    xn = x(end);
    brIteracija = brIteracija +1;
end;

disp('Nula funkcije x=f(x) je: ');
nulaf = xn;
disp('Iterativni niz je :')
disp(x');

```