

Zadatak 16.

```
type sistem.m
```

```
function x=sistem(A,b)

n=size(A,1); % Kada se length() primeni na matricu, vraća broj vrsti
A1=zeros(n);
b1=zeros(n,1);

% Potrebno je transformisati sistem Ax = b u oblik x = A1*x + b1
for i=1:n
    A1(i,:)=-A(i,+)/A(i,i);
    A1(i,i)=0;
    b1(i)=b(i)/A(i,i);
end
disp(A1)
disp(b1)
% Početna aproksimacija rešenja je vektor b1
x=b1;
% Za dobijanje nove aproksimacije koristi se rekurentna formula: x_n = A1*x_{n-1} + b1
% Za aproksimaciju ćemo uzimati xn dobijeno u pedesetoj iteraciji.
for i=1:50
    x=A1*x+b1;
end
```

```
A=[2 8 0;-2 4 -1; 3 -1 3];
b=[4 -2 4]';
B=A*diag([0.5 0.5 -0.5])*A^(-1); %uzimamo matricu cije su sopstvene
vrednosti po modulu manje od jedan
x=sistem(B,b)
```

```
      0      0      0
0.2041      0 -0.4898
0.6122 -1.5918      0

8.0000
-1.8776
-3.7551
x = 3×1
8.0000
-3.6487
6.9353
```

```
B\b
```

```
ans = 3×1
8.0000
-3.6522
6.9565
```

```
type matrica.m
```

```
function [A,b,x]=matrica(broj,d)
b=ones(d,1);
ad=broj*10*ones(d,1);
A=diag(ad);
for i=1:d-1
    ad=ones(d-i,1);
    A=A+diag(ad*(d-i),-i)+diag(ad,i);
end
```

```

disp('A=')
disp(A)
x=sistem(A,b);
disp('Resenje sistema je:')
disp(x);

```

```
[A,b,x]=matrica(3,3)
```

```

A=
    30     1     1
     2    30     1
     1     2    30

     0   -0.0333   -0.0333
   -0.0667     0   -0.0333
   -0.0333   -0.0667     0

    0.0333
    0.0333
    0.0333

```

```
Resenje sistema je:
```

```

    0.0313
    0.0302
    0.0303

```

```
A = 3×3
```

```

    30     1     1
     2    30     1
     1     2    30

```

```
b = 3×1
```

```

     1
     1
     1

```

```
x = 3×1
```

```

    0.0313
    0.0302
    0.0303

```

```
x=A\b % za tacno resenje sistema
```

```
x = 3×1
```

```

    0.0313
    0.0302
    0.0303

```

Zadatak 17.

```
type dominantna.m
```

```

function d=dominantna(A)
n=length(A);
d=zeros(n,1); %element d(i) nosice informaciju o ispunjenosti uslova u i-toj vrsti.

for i=1:n
%proveravamo da li je suma apsolutnih vrednosti vandijagonalnih elemenata
%i-te vrste manja od apsolutne vrednosti elementa na dijagonali.
if sum(abs(A(i,:)))-abs(A(i,i))<abs(A(i,i))
    d(i) = 1;
else d(i)=0;
end

```

```
end
d=all(d==1); %moze i d=all(d)
```

```
d=dominantna(B)
```

```
d =
logical
0
```

```
A = [10  2  1  1;
      1  8  2  1;
      2  1  9  1;
      1  1  1  7];
b=[1,1,1,1]';
type sistem2.m
```

```
function [iter,x]=sistem2(A,b,tol)
if dominantna(A)==0
    error('Matrica nije dijagonalno dominantna');
end
n=length(A);
A1=zeros(n);
b1=zeros(n,1);

for i=1:n
    A1(i,:)=-A(i,+)/A(i,i);
    A1(i,i)=0;
    b1(i)=b(i)/A(i,i);
end
q=norm(A1,inf);
tol=tol*(1-q)/q; %kriterijum zaustavljanja za Gaus zajedlovu metodu

x0=b1;
x=x0;
iter=1;

while(1)
% L+D+U=A, L je donjetrougaona, D je dijagonalna, U je gornje trougaona.
% Gaus-Zajdelova metoda se dobija iz relacije (L+D)x=-Ux=b,
% dakle iteracije mozemo sprovoditi koristeći istu matricu A1 i vektor b1
% kao kod metode proste iteracije,
% samo cemo u svakoj iteraciji koristiti umesto starih, nove, vec sracunate
% vrednosti koordinata vektora x.
    for i=1:n
        x(i)=A1(i,:)*x+b1(i); % nove koordinate racunamo preko starih i novih
                                % vec izracunatih u ovoj iteraciji
    end
    iter=iter+1;
    if all(abs(x-x0)<tol)
        break
    end
    x0=x;
end
```

```
[iter,x]=sistem2(A,b,1e-4)
```

```
iter =
6
x = 4×1
    0.0645
```

```
0.0843  
0.0751  
0.1109
```

```
A\b
```

```
ans = 4×1  
0.0645  
0.0843  
0.0751  
0.1109
```

```
[iter,x]=sistem2(B,b,1e-4)
```

```
Error using sistem2 (line 3)  
Matrica nije dijagonalno dominantna
```