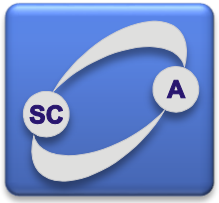


UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA AUTOMATIKU I UPRAVLJANJE SISTEMIMA

Matlab - Primeri

Modeliranje i simulacija sistema

Upravljanje, modelovanje i simulacija sistema

Primer selekcije

```
>> a=magic(6)
a =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> b=a(2:end, 3:end)
b =
     7    21    23    25
     2    22    27    20
    33    17    10    15
    34    12    14    16
    29    13    18    11

>> c=a(2:end, 3:end-1)
c =
     7    21    23
     2    22    27
    33    17    10
    34    12    14
    29    13    18

>> d=a([4 1 2], 3:end-1)
d =
    33    17    10
     6    26    19
     7    21    23
```

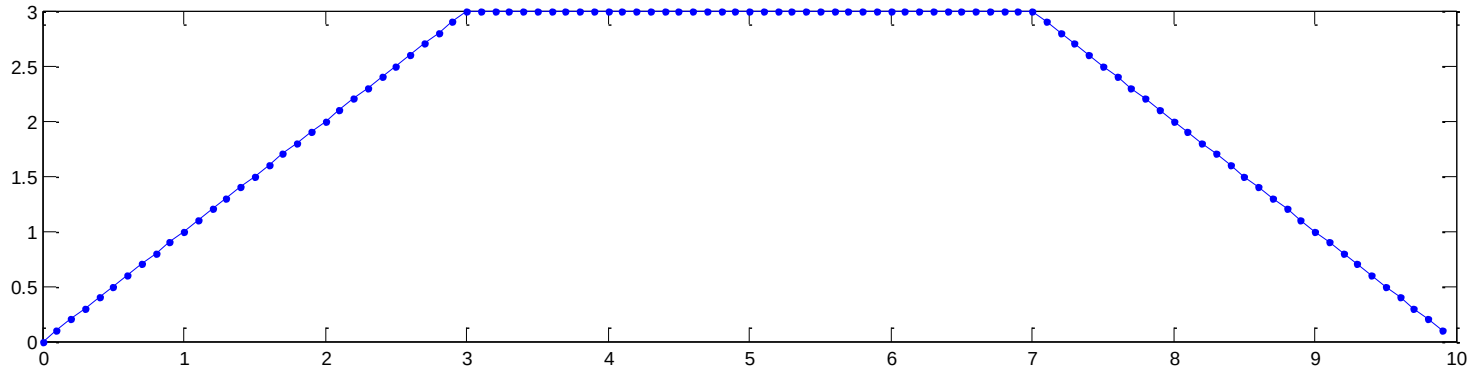
```
>> e=a([false true true false false false], :)
e =
     3    32     7    21    23    25
    31     9     2    22    27    20

>> e=a(boolean([0 1 1 0 0 0]), :)
e =

     3    32     7    21    23    25
    31     9     2    22    27    20
```

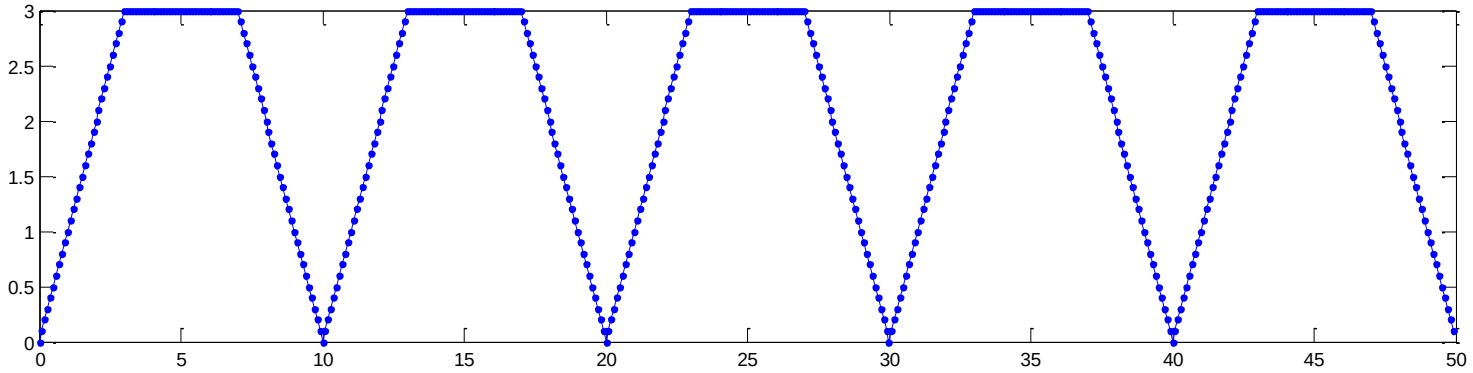
Primer brisanja kolona i vrsta

```
>> a=magic(6)
a =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> a([1 3 5],:)=[]
a =
     3    32     7    21    23    25
     8    28    33    17    10    15
     4    36    29    13    18    11

>> a(:,3:end)=[]
a =
     3    32
     8    28
     4    36
```

Primer generisanja signala - Trapez

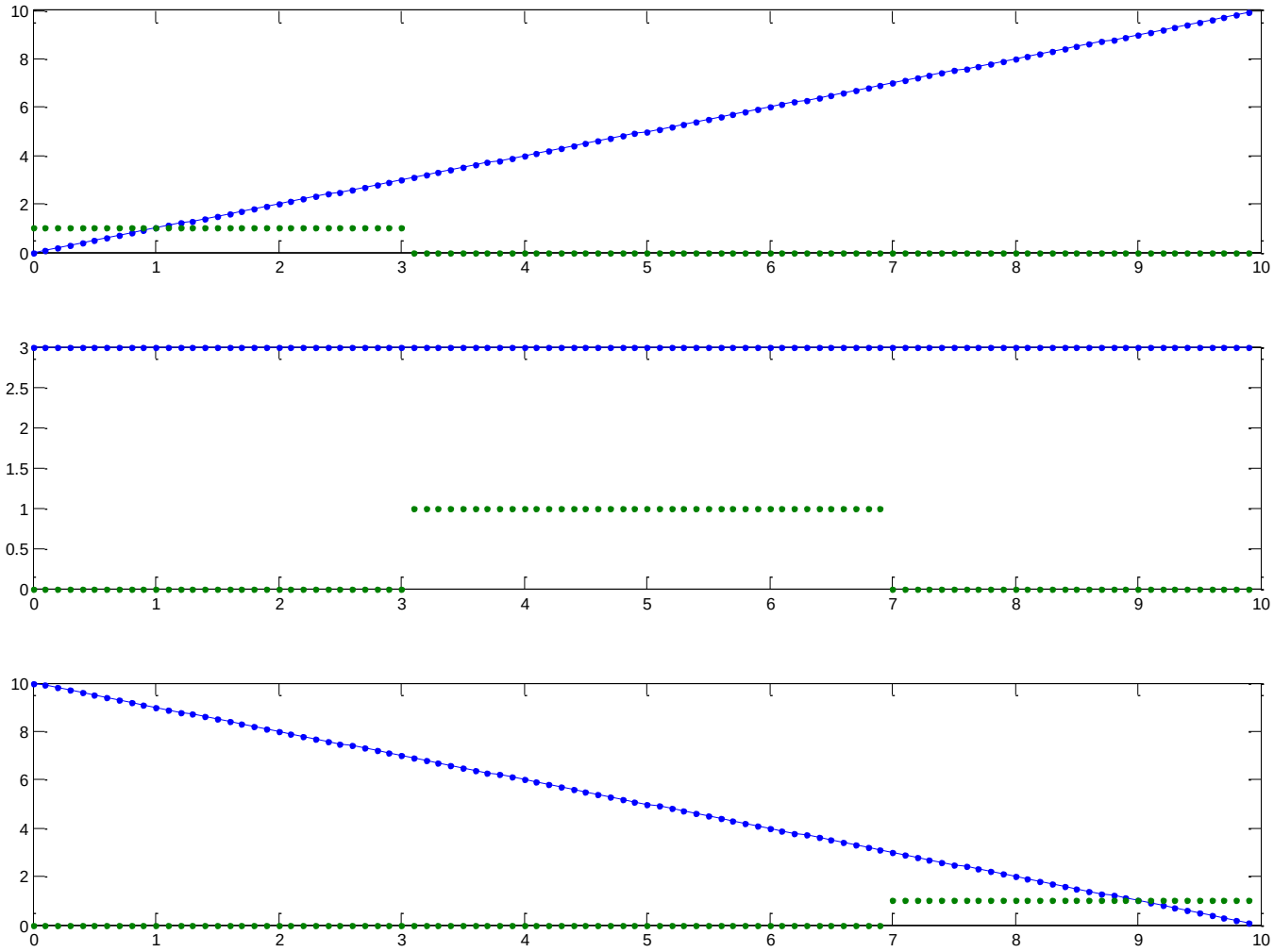
```
>> t=0:0.1:9.9;
>> x = t .* (t<=3) + 3 * (t>3 & t<7) + (10-t) .* (t>=7);
>> plot(t,x)
>> y = repmat(x,[1 5]);
>> plot(0:0.1:50-0.1, y, '.-')
```





Trapez (2)

```
>> subplot(3,1,1), plot(t,t,'.-', t,t<=3,'.')
>> subplot(3,1,2), plot(t,3*ones(1,length(t)),'.-', t,(t>3) & (t<7),'.')
>> subplot(3,1,3), plot(t,10-t,'.-', t,t>=7,'.')
```



Trapez (3)

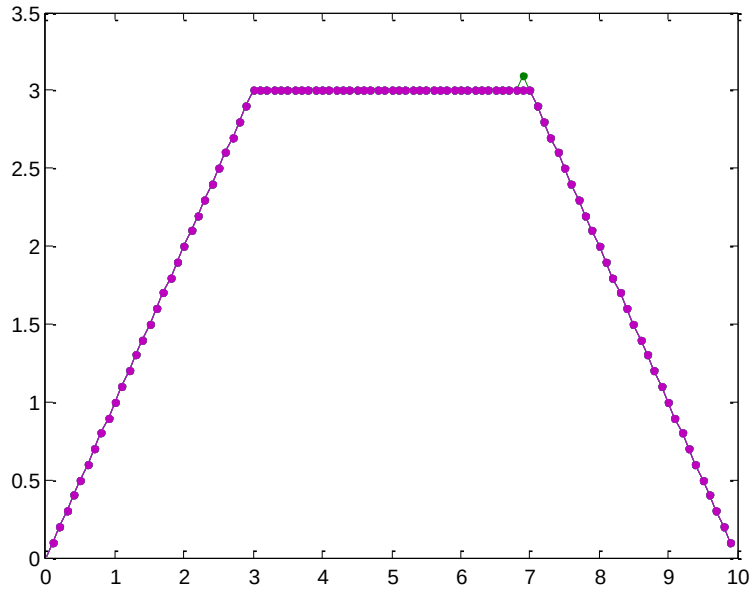
```
% Primer formiranja trapeznog signala
t = 0:0.1:9.9;
% nacin 0
x = t.*(t<=3) + 3*(t>3 & t<7) + (10-t).*(t>=7);

% nacin 1 - nategnut
for i=1:100
   if i <= 30
      x1(i) = (i-1)/10;
   elseif i < 70      % ima gresku <71!
      x1(i) = 3;
   else
      x1(i) = 3 - (i-71)/10;
   end
end

% nacin 2 - klasican
for i=1:100
   if t(i) <= 3
      x2(i) = t(i);
   elseif t(i) < 7
      x2(i) = 3;
   else
      x2(i) = 10-t(i);
   end
end
```

```
% nacin 3 – cudan ali korektan
x3 = [];
for i=1:100
   if t(i) <= 3
      x3 = [x3 t(i)];
   elseif t(i) < 7
      x3 = [x3  3];
   else
      x3 = [x3 10-t(i)];
   end
end

% nacin 4 – kao 3 ali lepsi
x4 = [];
for tau = t
   if tau <= 3
      x4 = [x4  tau];
   elseif tau < 7
      x4 = [x4  3];
   else
      x4 = [x4  10-tau];
   end
End
```

Trapez (4)

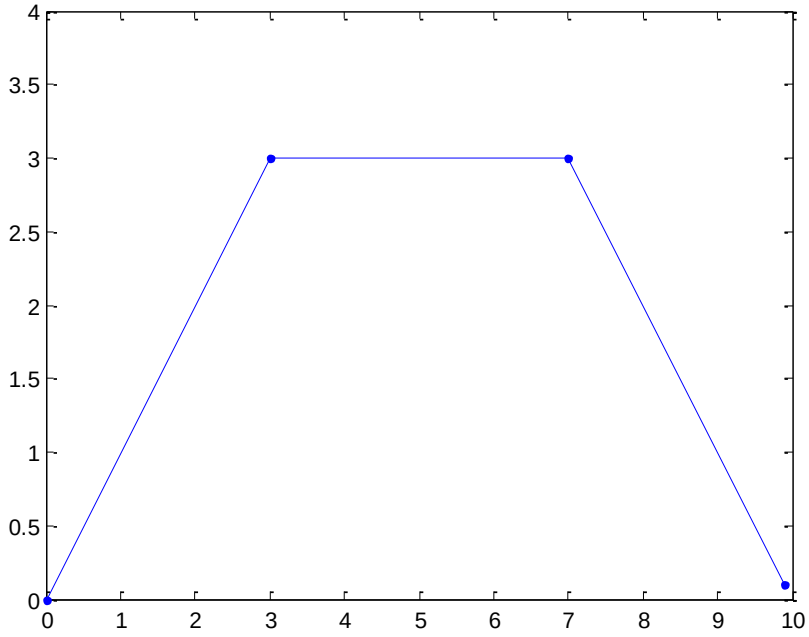
```
>> [(1:length(x))' x' x1' x2' x3' x4']
ans =
    1.0    0      0      0      0      0
    2.0    0.1    0.1    0.1    0.1    0.1
    3.0    0.2    0.2    0.2    0.2    0.2
    4.0    0.3    0.3    0.3    0.3    0.3
    5.0    0.4    0.4    0.4    0.4    0.4
           …
   27.0    2.6    2.6    2.6    2.6    2.6
   28.0    2.7    2.7    2.7    2.7    2.7
   29.0    2.8    2.8    2.8    2.8    2.8
   30.0    2.9    2.9    2.9    2.9    2.9
   31.0    3.0    3.0    3.0    3.0    3.0
   32.0    3.0    3.0    3.0    3.0    3.0
   33.0    3.0    3.0    3.0    3.0    3.0
           …
   67.0    3.0    3.0    3.0    3.0    3.0
   68.0    3.0    3.0    3.0    3.0    3.0
   69.0    3.0    3.0    3.0    3.0    3.0
   70.0    3.0    3.1    3.0    3.0    3.0
   71.0    3.0    3.0    3.0    3.0    3.0
   72.0    2.9    2.9    2.9    2.9    2.9
   73.0    2.8    2.8    2.8    2.8    2.8
   74.0    2.7    2.7    2.7    2.7    2.7
           …
   98.0    0.3    0.3    0.3    0.3    0.3
   99.0    0.2    0.2    0.2    0.2    0.2
  100.0    0.1    0.1    0.1    0.1    0.1

% radi jasnijeg prikaza ‘’ je obrisano iz brojeva
```

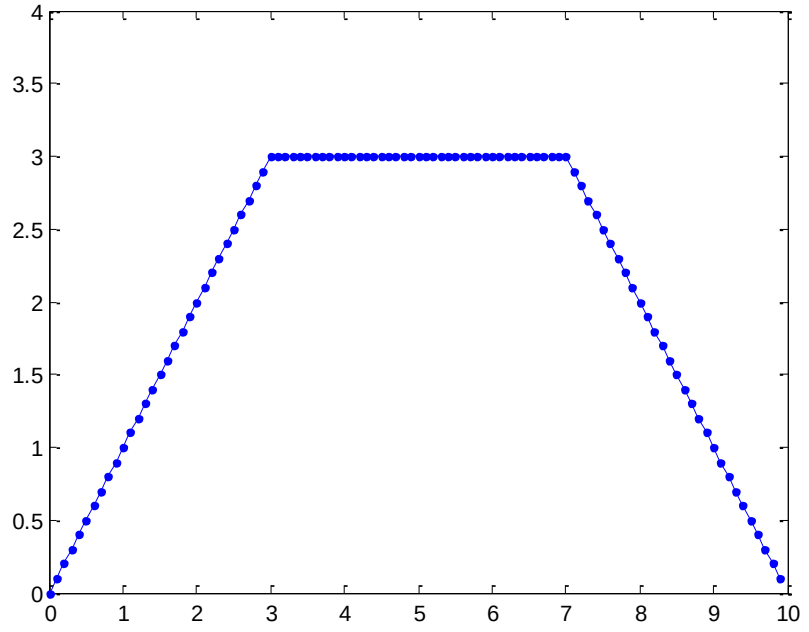
```
>> plot(t,[x' x1' x2' x3' x4'])

>> r = [x1' x2' x3' x4'] - x'*[1 1 1 1];
>> any(r ~= 0)
ans =
     1     0     0     0          % x1 nije Ok
```



Trapez – još jednom

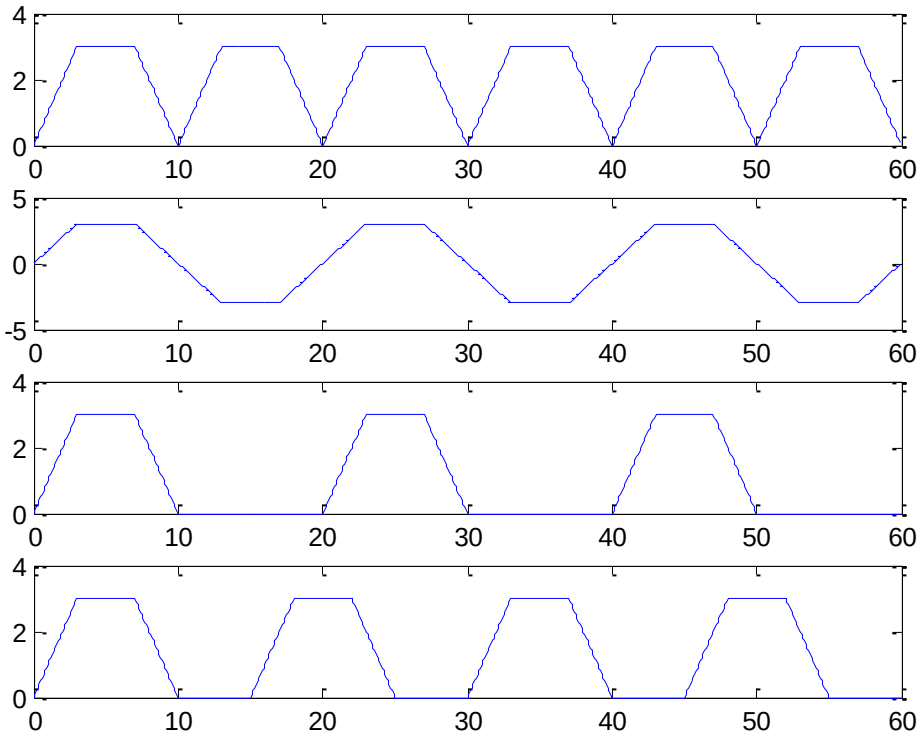
```
t = [0  3  7  9.9];
x = [0  3  3  0.1];
plot(t,x,'.-'); axis([0 10 0 4])

ti = 0:0.1:9.9;
xi = interp1(t,x,ti);
plot(ti,xi,'.-'), axis([0 10 0 4])

>> [ti' xi']
ans =
    0      0
    0.1    0.1
    0.2    0.2
    0.3    0.3
           …
    2.8    2.8
    2.9    2.9
    3.0    3.0
    3.1    3.0
    3.2    3.0
           …
    6.8    3.0
    6.9    3.0
    7.0    3.0
    7.1    2.9
    7.2    2.8
           …
    9.8    0.2
    9.9    0.1
```



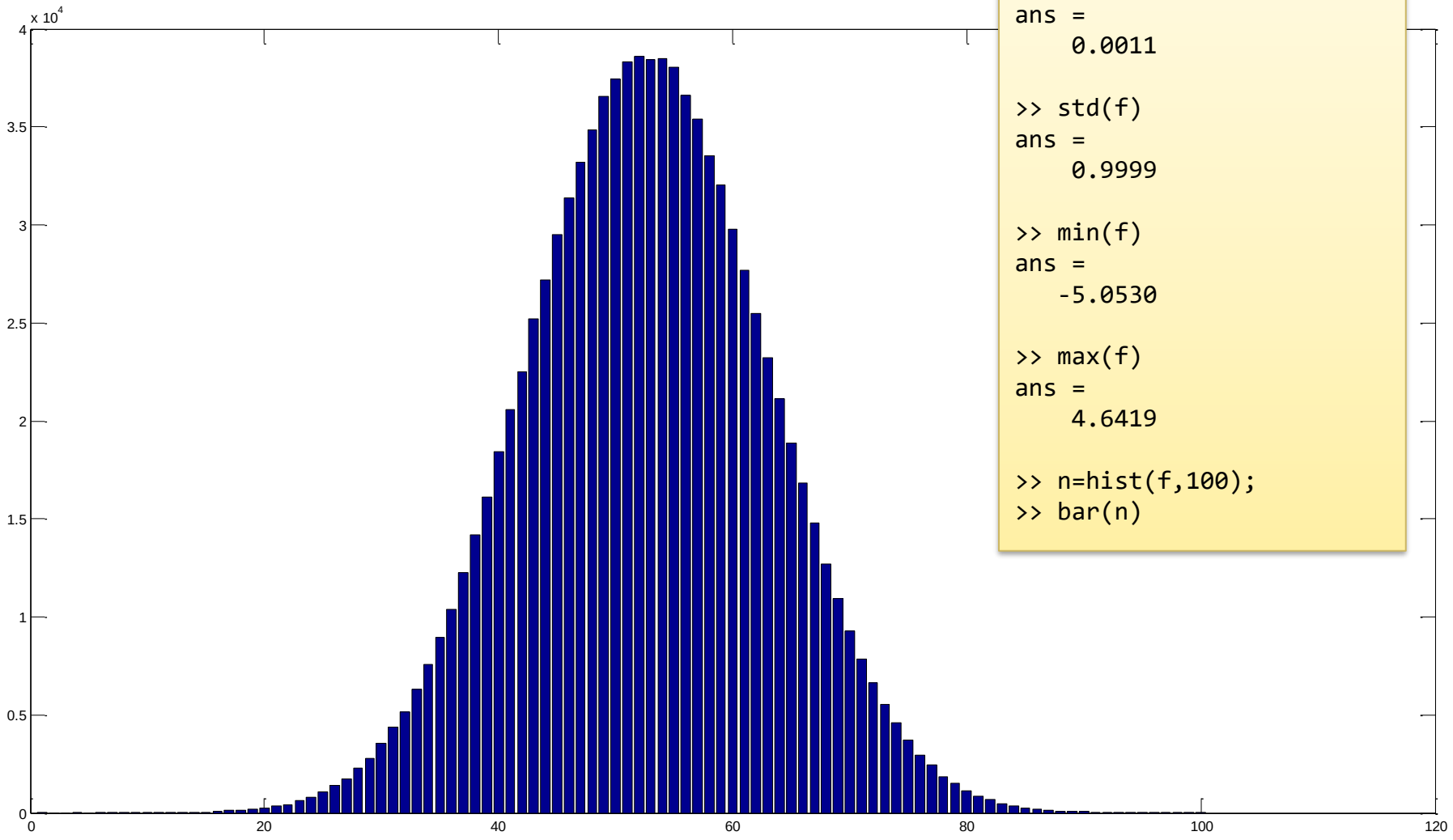


Ponavljanje “periode trapeza”

```
x = xi;
t = 0:0.1:60-0.1;

subplot(4,1,1), plot(t, [x  x  x  x  x  x]);
subplot(4,1,2), plot(t, [x -x  x -x  x -x]);
z = zeros(1,length(x));
subplot(4,1,3), plot(t, [x  z  x  z  x  z]);
z = zeros(1,length(x)/2);
subplot(4,1,4), plot(t, [x  z  x  z  x  z  x  z]);
```



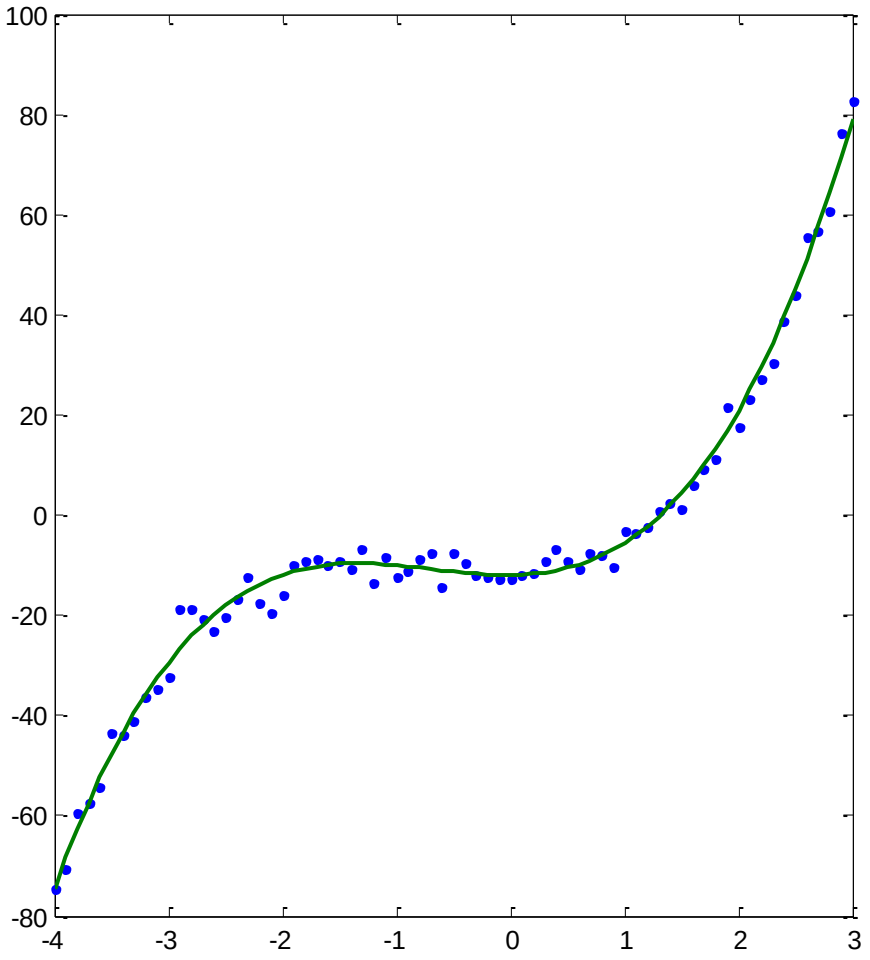
Brojevi generisani po normalnoj raspodeli




```
>> f = randn(1,1000000);

>> mean(f)
ans =
    0.0011

>> std(f)
ans =
    0.9999

>> min(f)
ans =
   -5.0530

>> max(f)
ans =
    4.6419

>> n=hist(f,100);
>> bar(n)
```

Fitovanje krive

```
>> P = [2 4.1 0 -12];
>> x = -4:0.1:3;
>> y = polyval(P,x);
>> sum = 3 * randn(1, length(x));
>> ysum = y + sum;
>> Q = polyfit(x,ysum,3)
Q =
    1.9867    4.0911    0.2211  -11.9147
>> y3 = polyval(Q,x);
>> plot(x, ysum,'.', x, y3)
```



Primer upotrebe strukture

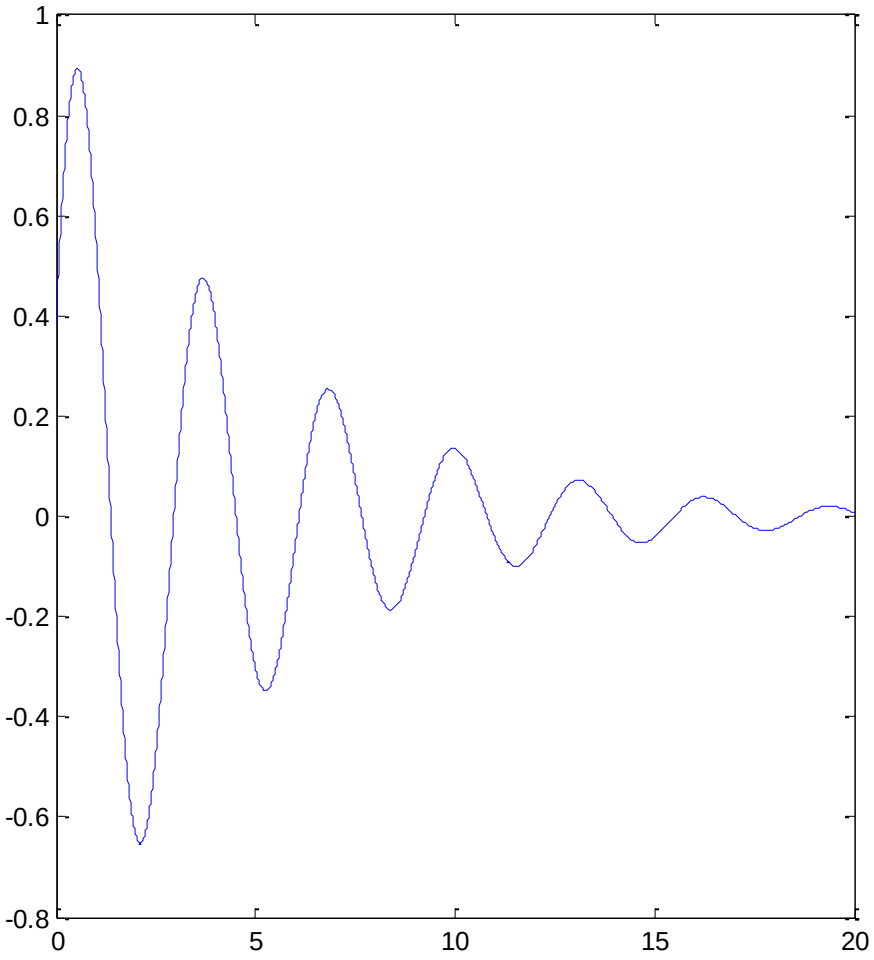
```
>> s(1)  = struct('index', 12017, 'ime', 'Nenad Nikolic',     'test1', 0,   'test2', NaN  );
>> s(2)  = struct('index', 12383, 'ime', 'Sandor Jozef',      'test1', 0,   'test2', 0    );
>> s(3)  = struct('index', 12570, 'ime', 'Sandra Kukolj',     'test1', 0,   'test2', NaN  );
>> s(4)  = struct('index', 12643, 'ime', 'Igor Cverdelj',     'test1', 6,   'test2', 3    );
>> s(5)  = struct('index', 12644, 'ime', 'Renata Vaderna',    'test1', 1.5, 'test2', 4    );
>> s(6)  = struct('index', 12647, 'ime', 'Milena Milosevic',  'test1', 2.5, 'test2', 0.5  );
>> s(7)  = struct('index', 12654, 'ime', 'Milos Jokic',       'test1', 2,   'test2', 4    );
>> s(8)  = struct('index', 12655, 'ime', 'Igor Trifunovic',   'test1', 0.5, 'test2', 1    );
>> s(9)  = struct('index', 12658, 'ime', 'Krsto Lazic',       'test1', 5.5, 'test2', NaN  );
>> s(10) = struct('index', 12659, 'ime', 'Vanja Knezevic',    'test1', 6,   'test2', NaN  );
>> s(11) = struct('index', 12664, 'ime', 'Zarko Milovanovic', 'test1', 2,   'test2', NaN  );
>> s(12) = struct('index', 12669, 'ime', 'Marko Kovacevic',   'test1', 6,   'test2', 1    );
>> s(13) = struct('index', 12670, 'ime', 'Danijel Blagojevic','test1', 1,   'test2', NaN  );
>> s(14) = struct('index', 12676, 'ime', 'Milan Knezevic',    'test1', 0,   'test2', 0.5  );
>> s(15) = struct('index', 12678, 'ime', 'Arpad Sagi',        'test1', 0.5, 'test2', NaN  );

>> s
s =
1x15 struct array with fields:
    index
    ime
    test1
    test2
```

```
>> t1 = [s.test1]                % rezultrati prvog testa
t1 =
     0     0     0     6.0   1.5   2.5   2.0   0.5   5.5   6.0   2.0   6.0   1.0   0     0.5
>> t2 = [s.test2]                % rezultrati drugog testa
t2 =
     NaN   0     NaN   3.0   4.0   0.5   4.0   1.0   NaN   NaN   NaN   1.0   NaN   0.5   NaN
>> isnan(t2)                     % kako isnan(…) radi
ans =
     1     0     1     0     0     0     0     0     1     1     1     0     1     0     1
>> t10 = t1; t10(isnan(t10))=0   % t10 su t1 rezultati sa zamenjim NaN sa 0
t10 =
     0     0     0     6.0   1.5   2.5   2.0   0.5   5.5   6.0   2.0   6.0   1.0   0     0.5
>> t20 = t2; t20(isnan(t20))=0   % t20 su t2 rezultati sa zamenjim NaN sa 0
t20 =
     0     0     0     3.0   4.0   0.5   4.0   1.0   0     0     0     1.0   0     0.5   0
>> tu = t10+t20                  % ukupni rezultati
tu =
     0     0     0     9.0   5.5   3.0   6.0   1.5   5.5   6.0   2.0   7.0   1.0   0.5   0.5
>> [m,index] = max(tu)           % najbolji rezultat
m =
     9
index =
     4
>> najbolji = s(index)           % student sa najboljim rezultatom
najbolji =
    index: 12643
      ime: 'Igor Cverdelj'
    test1: 6
    test2: 3
>> p1 = mean(t1(~isnan(t1)))     % prosek prvog testa
p1 =
    2.2333
>> p2 = mean(t2(~isnan(t2)))     % prosek drugog testa
p2 =
    1.7500
```

Primer funkcije

```
function y=priosc(t)
y = exp(-0.2*t).*sin(2*t + pi/8);
```

```
>> vreme = 0:0.01:20;
>> izlaz = priosc(vreme);
>> plot(vreme,izlaz)
```



Minimum funkcije jedne promenljive u zadatom intervalu

Primer 1: Traženje minimuma f-je
f(x)=sin(x) na intervalu [0, 2π].

```
>> x = fminbnd(@sin, 0, 2*pi)
x =
    4.7124
>> sin(x)
ans =
   -1.0000
>> provera = 3*pi/2
provera =
    4.7124
```

Primer 2: Traženje minimuma f-je
$f(x)=(x-3)^2-1$ na intervalu [0, 5].

```
>> f = @(x) (x-3)^2-1;
>> xmin = fminbnd(f, 0, 5)
xmin =
     3
>> fmin = f(xmin)
fmin =
    -1
```

Minimum funkcije više promenljivih bez ograničenja

Primer 1: naći minimum funkcije
f(x)=sin(x)+3 u okolini tačke 2.

```
>> f = @(x) sin(x)+3;
>> xmin = fminsearch(f, 2)
xmin =
    4.7124
```

Primer 2: naći minimum funkcije
$f(\mathbf{x})=3x_1^2+2x_1x_2+x_2^2$ u okolini tačke (1,1).

```
>> f = @(x) 3*x(1)^2+2*x(1)*x(2)+x(2)^2;
>> [xmin,fmin] = fminunc(f,[1;1])
xmin =
  1.0e-006 *
    0.2541
   -0.2029
fmin =
  1.3173e-013
>> [xmin,fmin] = fminsearch(f,[1;1])
xmin =
  1.0e-004 *
   -0.0675
    0.1715
fmin =
  1.9920e-010
```

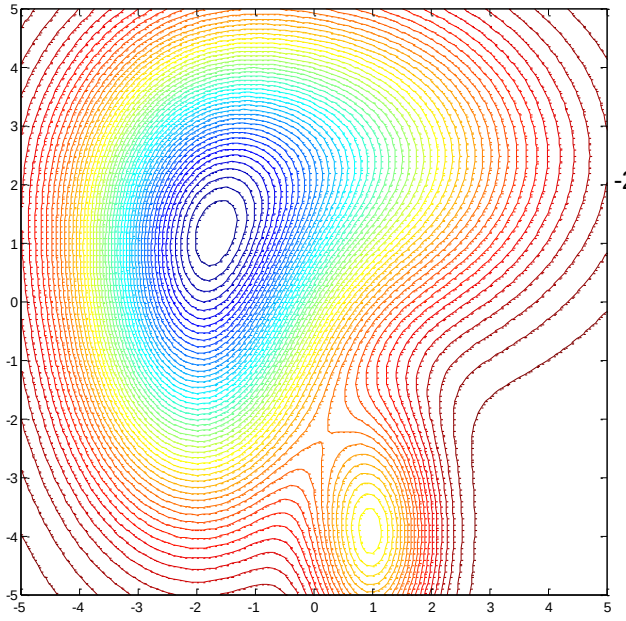
‘Min 2’ – traženje minimuma

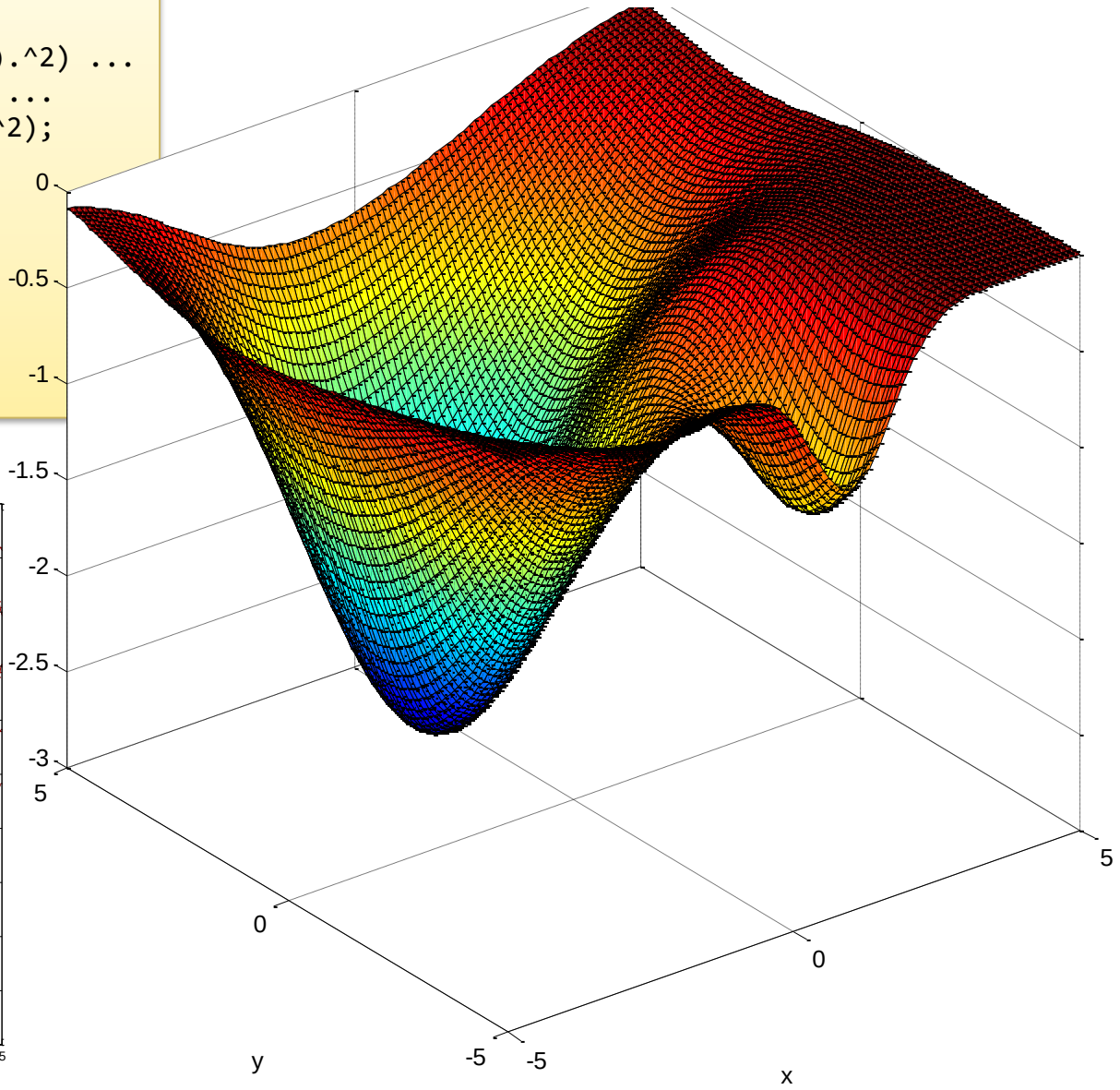
- Naći minimum funkcije 2 promenljive (bez ograničenja)

$$f(x,y) = -e^{-(x-1)^2-(0.5y+2)^2} - 2e^{-(0.5x+1)^2-(0.3y)^2} - 1.5e^{-(0.3x)^2-(0.4y-1)^2}$$

‘Min 2’ - funkcija

```
a = -5:0.1:5;
[x,y] = meshgrid(a,a);
g = @(x,y) -exp(-(x-1).^2-(.5*y+2).^2) ...
  - 2*exp(-(.5*x+1).^2-(.3*y).^2) ...
  - 1.5*exp(-(.3*x).^2-(0.4*y-1).^2);

f = @(x) g(x(1),x(2));

z = g(x,y);
surf(x,y,z)
contour(x,y,z,50)
axis('square')
```

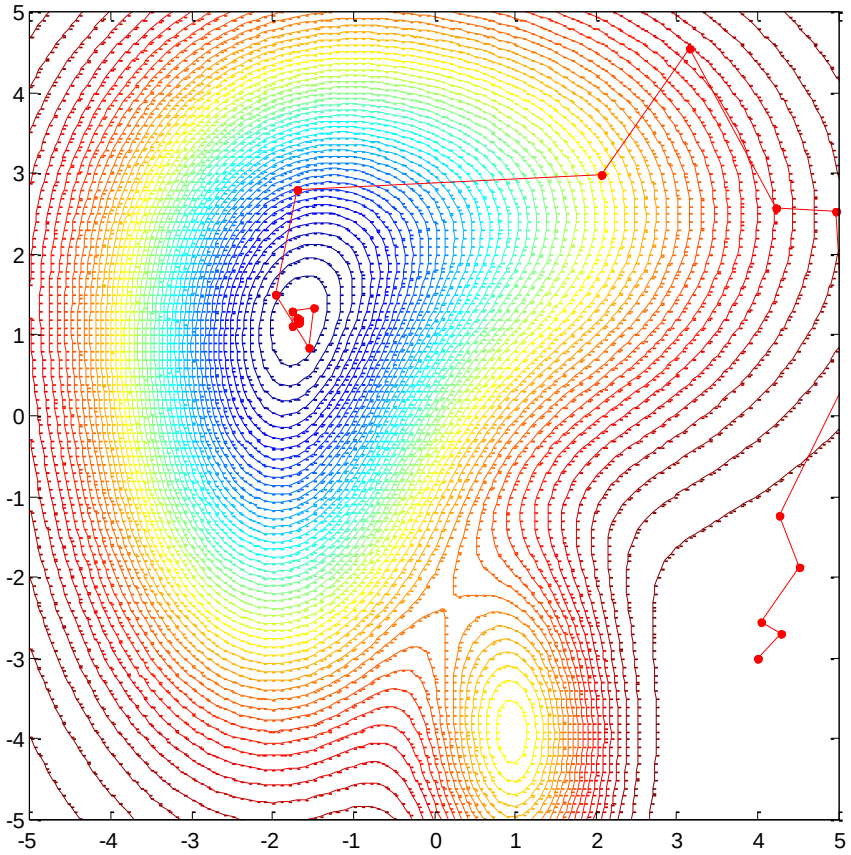




‘Min 2’ – traženje minimuma (1)

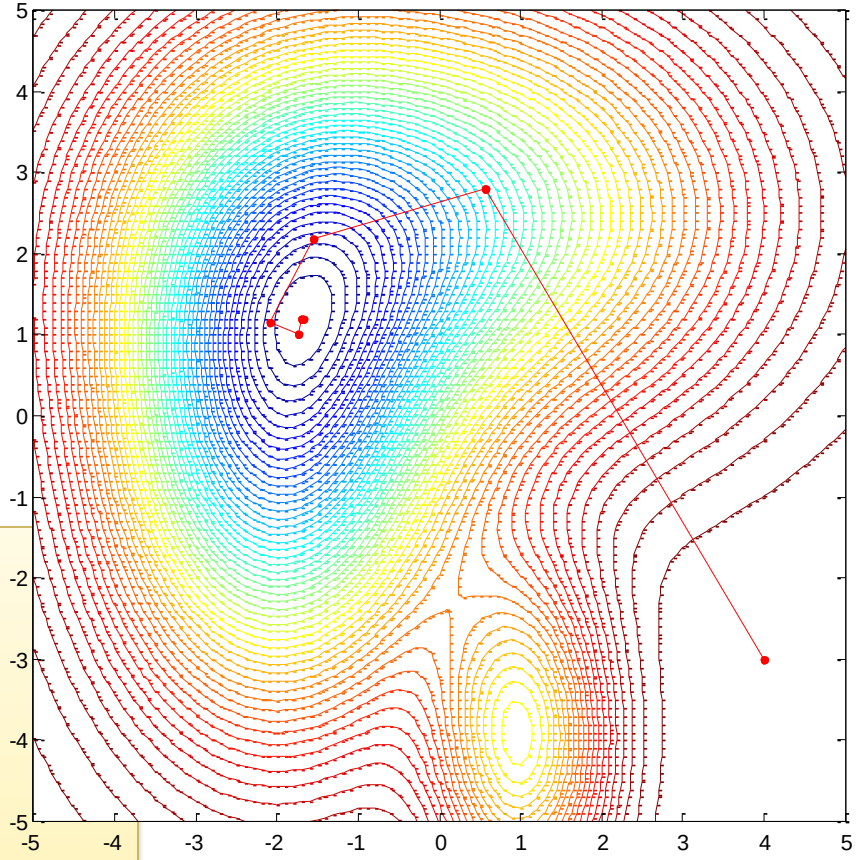
- Nelder-Mead algoritam direktne pretrage
  - bez upotrebe izvoda funkcije cilja
- Početna tačka (4, -3)
- Pronađen globalni optimum

```
options = optimset('GradObj','off', ...
                   'Display','iter');
[xmin,fmin]=fminsearch(f,[4;-3],options);
```



‘Min 2’ – traženje minimuma (2)

- Upotrebe izvoda funkcije cilja; Početna tačka (4, -3); Pronađen globalni optimum u 8 iteracija




```
options = optimset('GradObj','on','Display','iter');
[xmin,fmin]=fminunc(@fjagrad,[4;-3],options);

function [fja,grad] = fjagrad(xvec)
x = xvec(1); y = xvec(2);
fja = -exp(-(x-1).^2-(.5*y+2).^2) ...
    - 2*exp(-(.5*x+1).^2-(.3*y).^2) ...
    - 1.5*exp(-(.3*x).^2-(0.4*y-1).^2);
grad = [-exp(-(x-1).^2-(.5*y+2).^2)*(-2)*(x-1) ...
    - 2*exp(-(.5*x+1).^2-(.3*y).^2)*(-2)*(.5*x+1)*0.5 ...
    - 1.5*exp(-(.3*x).^2-(0.4*y-1).^2)*(-2)*(.3*x)*0.3
    -exp(-(x-1).^2-(.5*y+2).^2)*(-2)*(.5*y+2)*0.5 ...
    - 2*exp(-(.5*x+1).^2-(.3*y).^2)*(-2)*(.3*y)*0.3 ...
    - 1.5*exp(-(.3*x).^2-(0.4*y-1).^2)*(-2)*(0.4*y-1)*0.4];
```