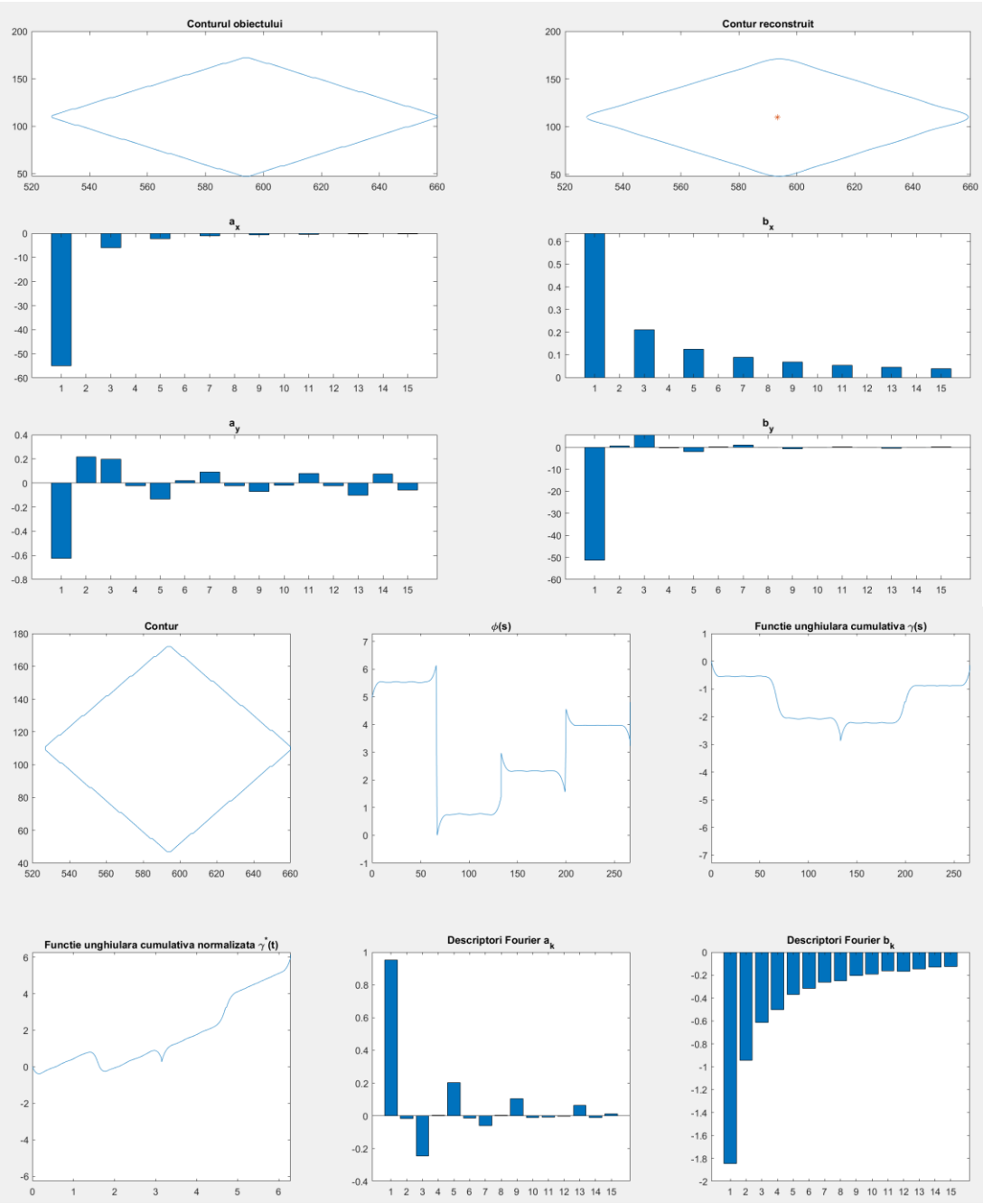
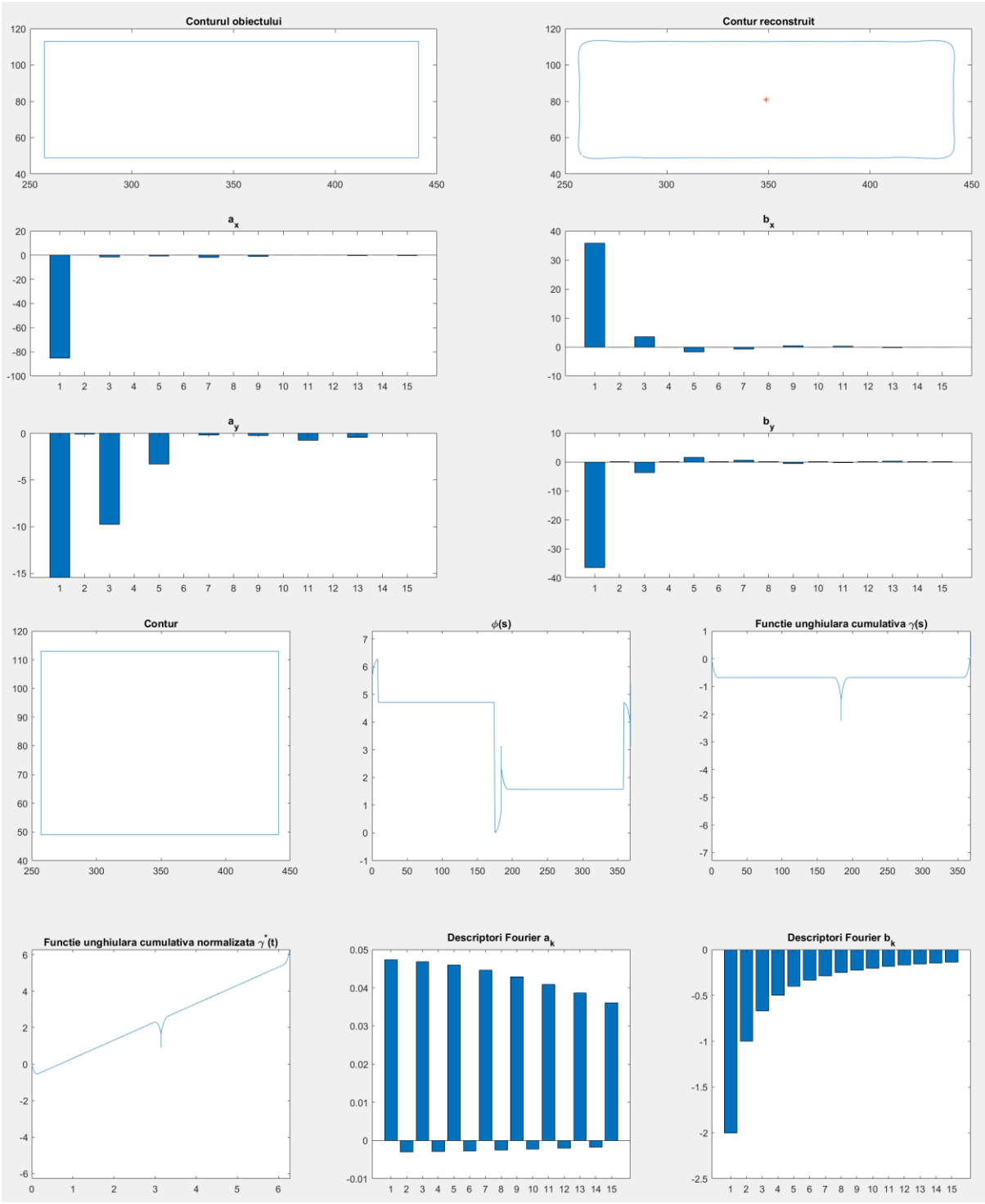
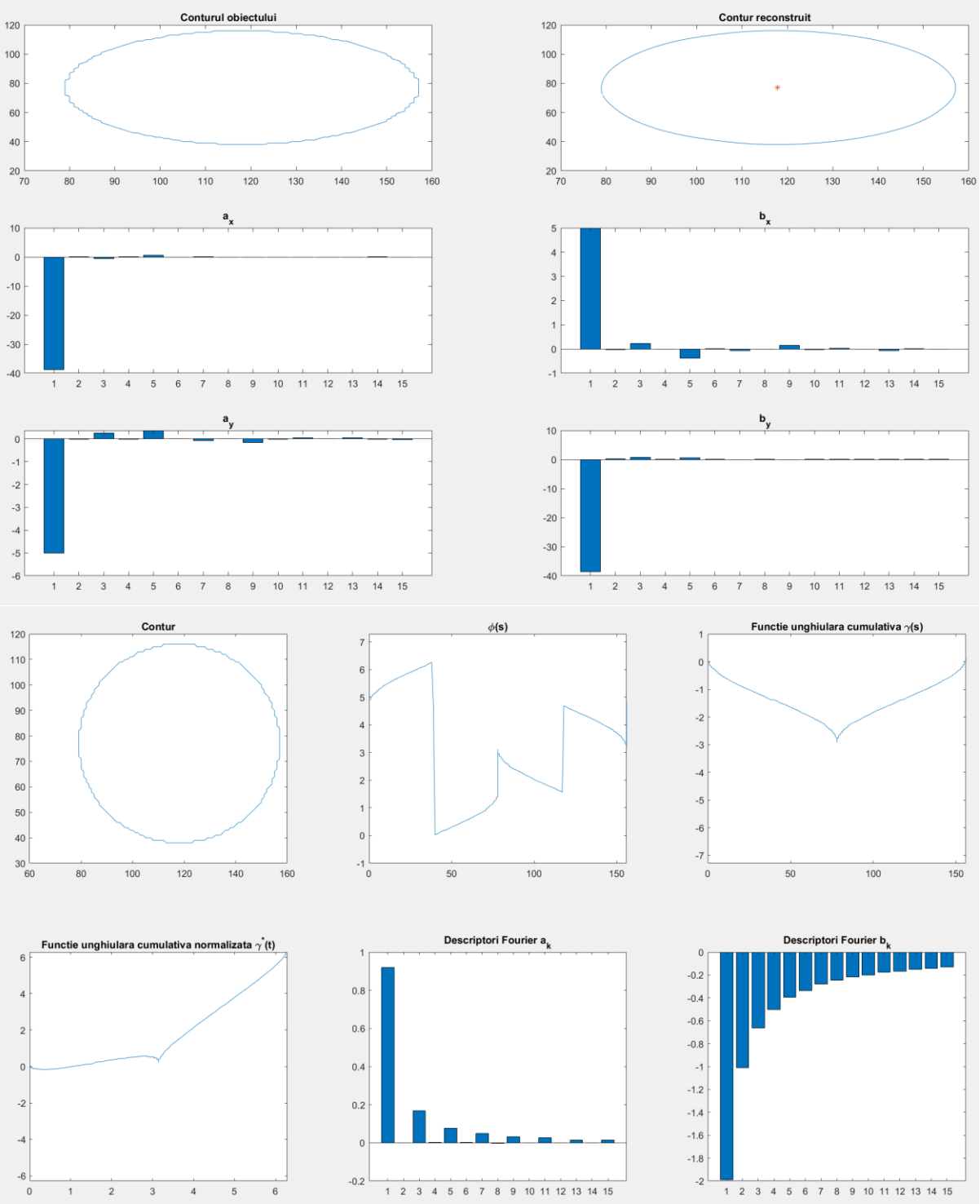
```
%% Exercitiul 1
clc
clear all
close all

% b si c
im = imread('imagine.jpg')
imshow(im)
```

```
if (~islogical(im))
    % transformare imagine RGB in imagine cu nuante de gri
    if (ndims(im)>2)
        im = rgb2gray(im);
    end
    % calcul prag in functie de care se va binariza imaginea
   level=graythresh(im);
    % binarizare imagine
    BW = im2bw(im,level);
end
figure, imshow(BW)
```

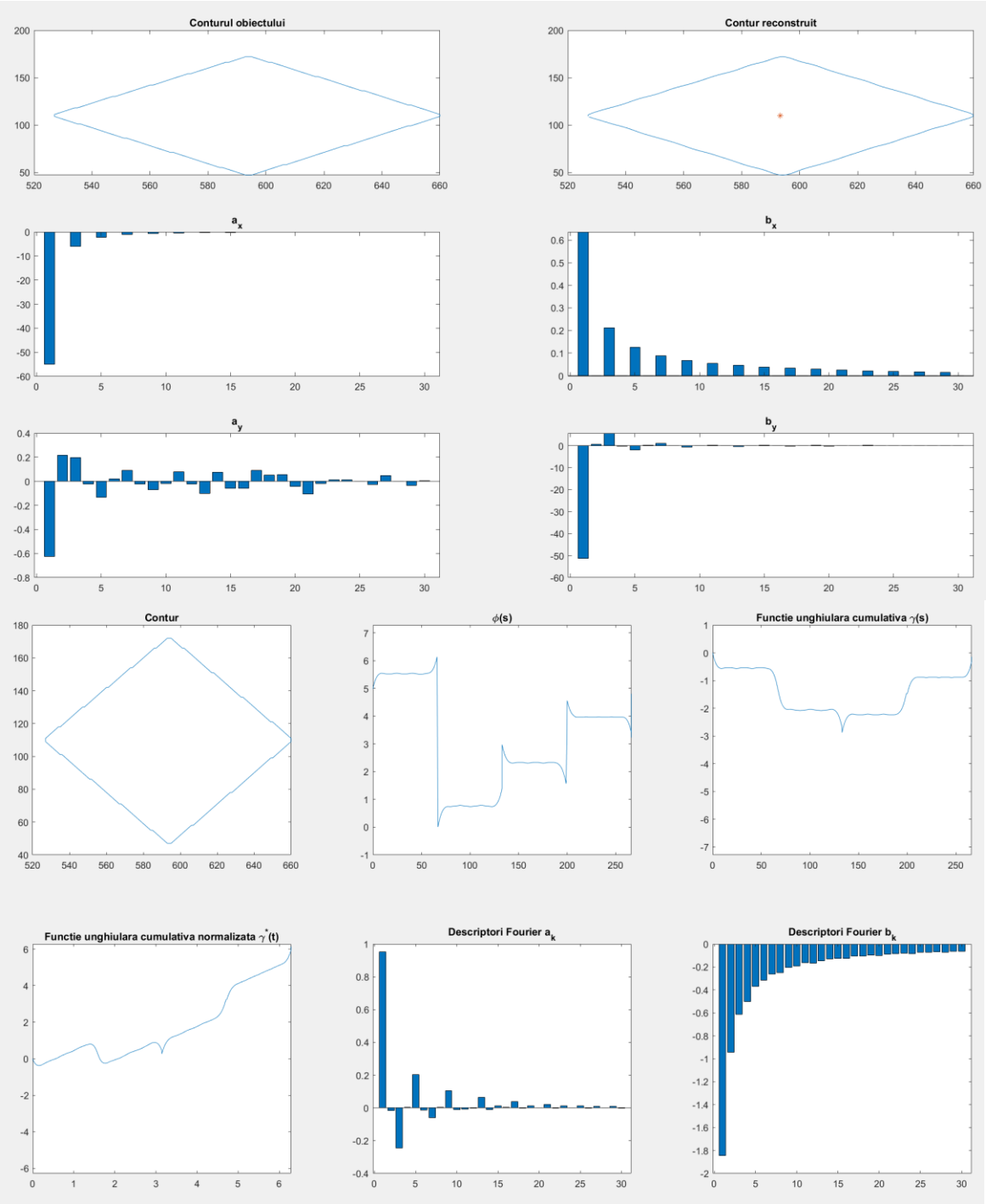
```
[B,L,N] = bwboundaries(BW,'noholes');

%% 15 coeficienti
nc = 15;
for (k=1:N)
    X = B{k}(:,2);
    Y = B{k}(:,1);
    contur = [X'; Y'];

    angularFunctionDescriptors(contur, nc);
   ellipticFourierDescriptors(contur, nc);

   %pause

end
```

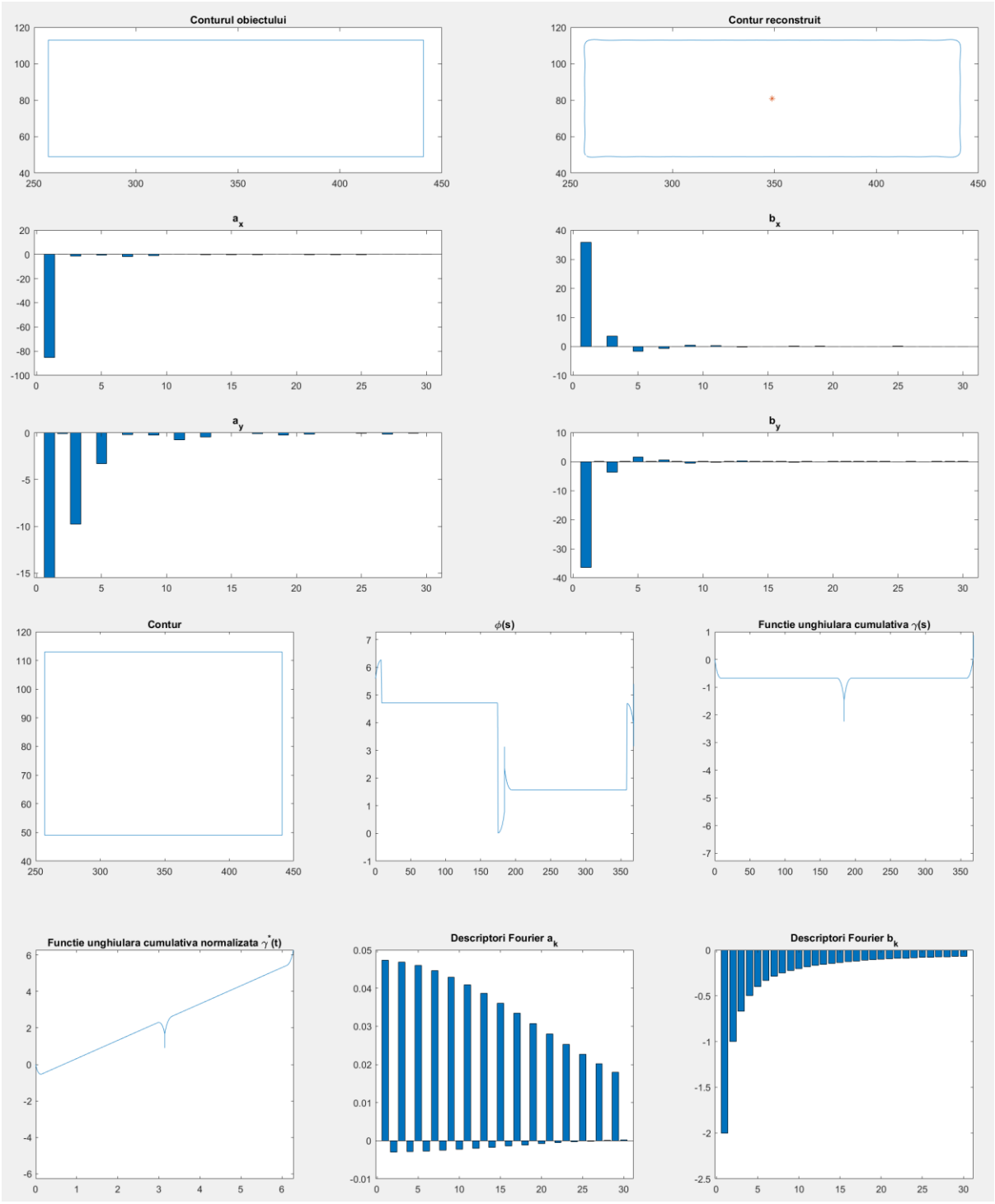


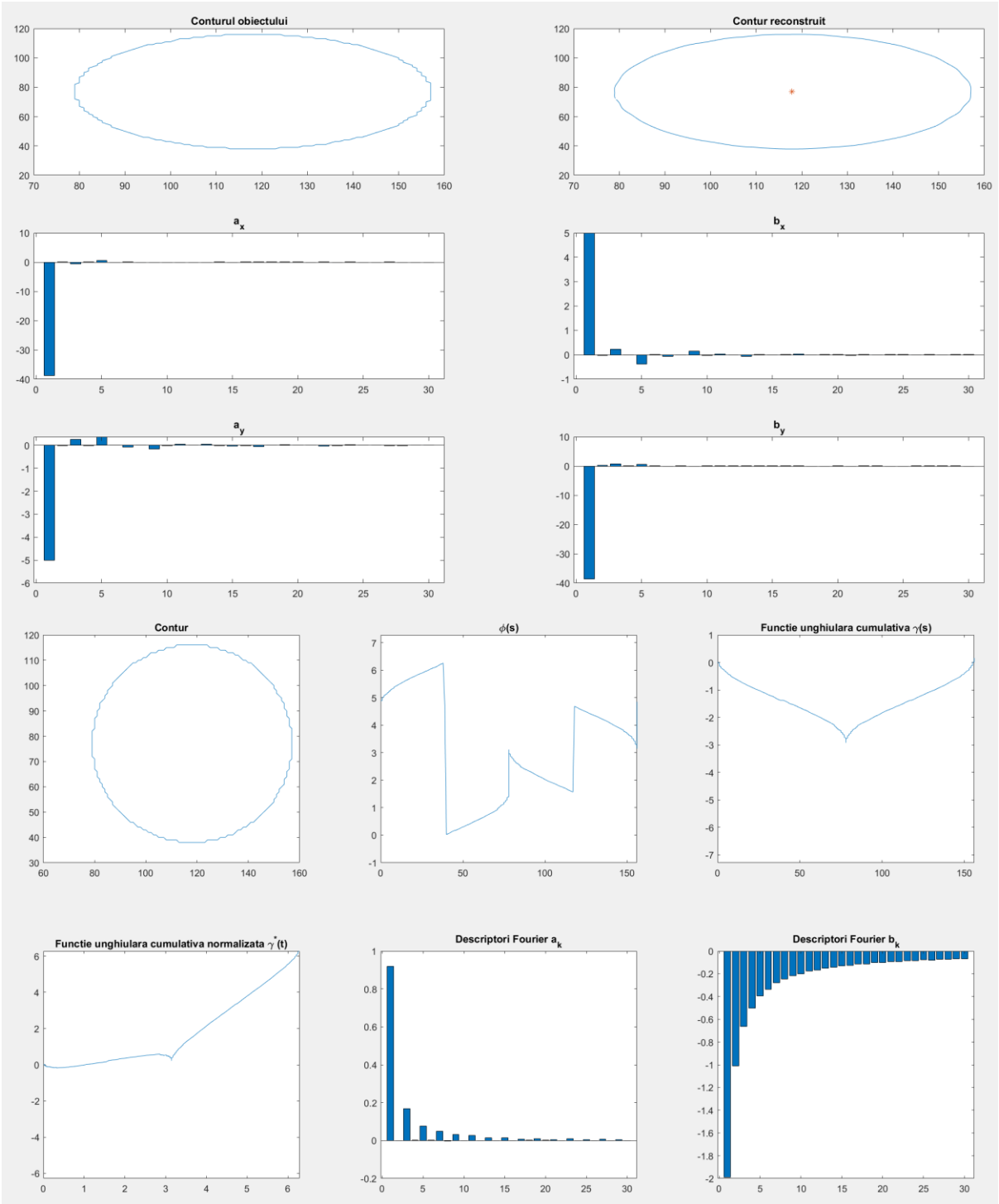

Conturul obiectului
Contur reconstruit
a_x
b_x
a_y
b_y
Contur
ϕ(s)
Functie unghiulara cumulativa γ(s)
Functie unghiulara cumulativa normalizata γ*(t)
Descriptori Fourier a_k
Descriptori Fourier b_k


Conturul obiectului
Contur reconstruit
a_x
b_x
a_y
b_y
Contur
ϕ(s)
Functie unghiulara cumulativa γ(s)
Functie unghiulara cumulativa normalizata γ*(t)
Descriptori Fourier a_k
Descriptori Fourier b_k

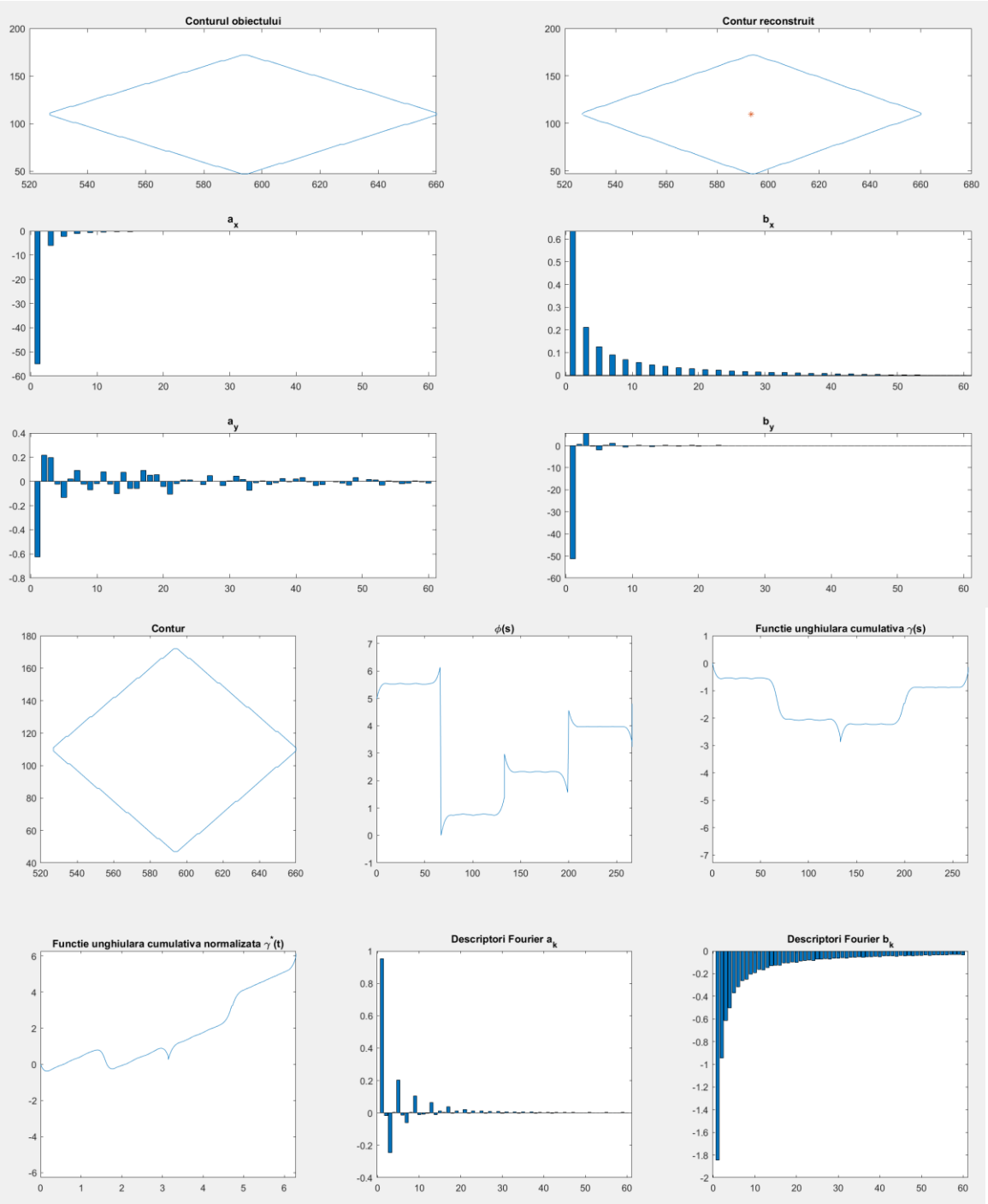
```
%% 30 coeficienti
nc = 30;
for (k=1:N)
    X = B{k}(:,2);
    Y = B{k}(:,1);
    contur = [X'; Y'];

    angularFunctionDescriptors(contur, nc);
   ellipticFourierDescriptors(contur, nc);

   %pause

end
```

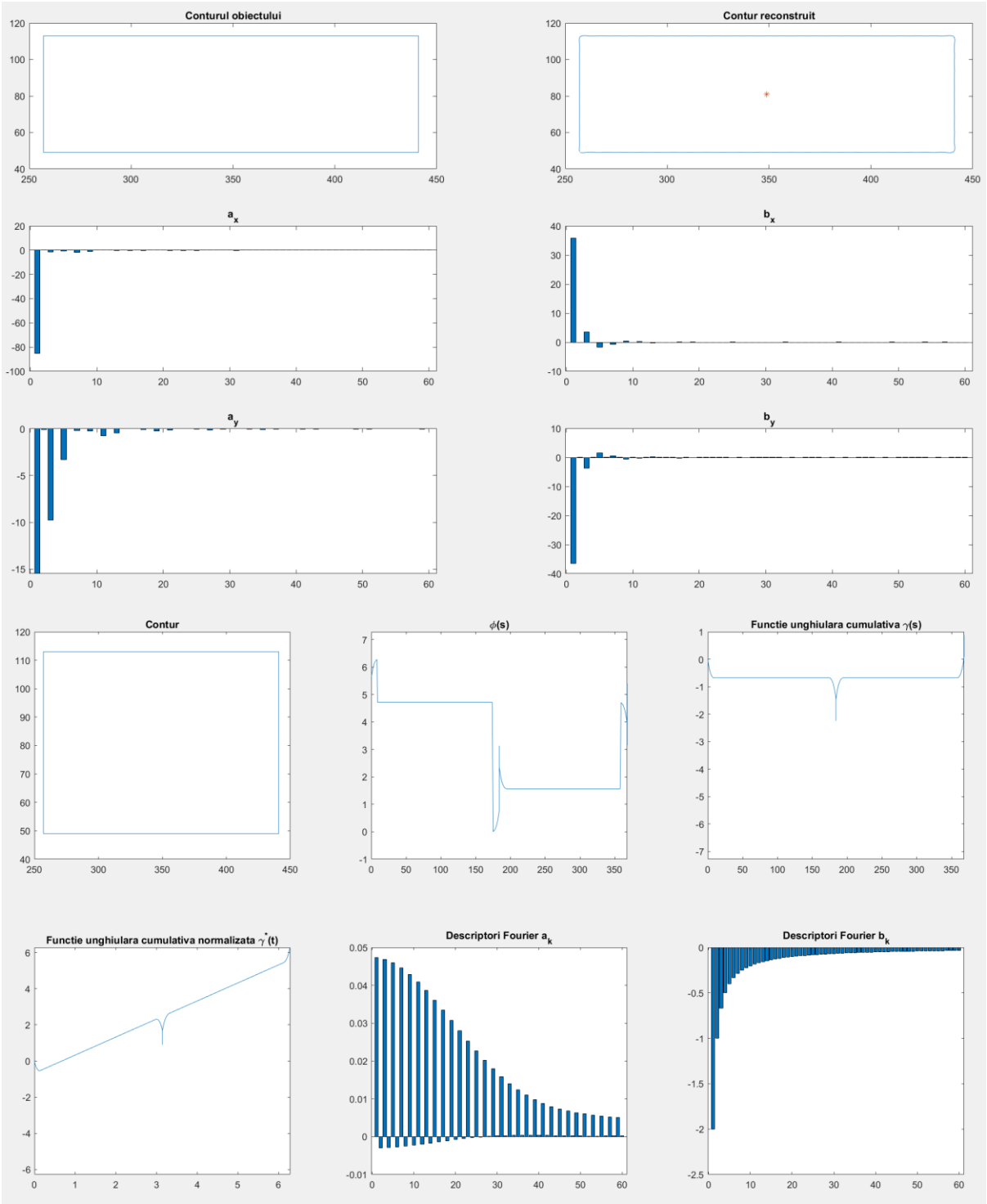


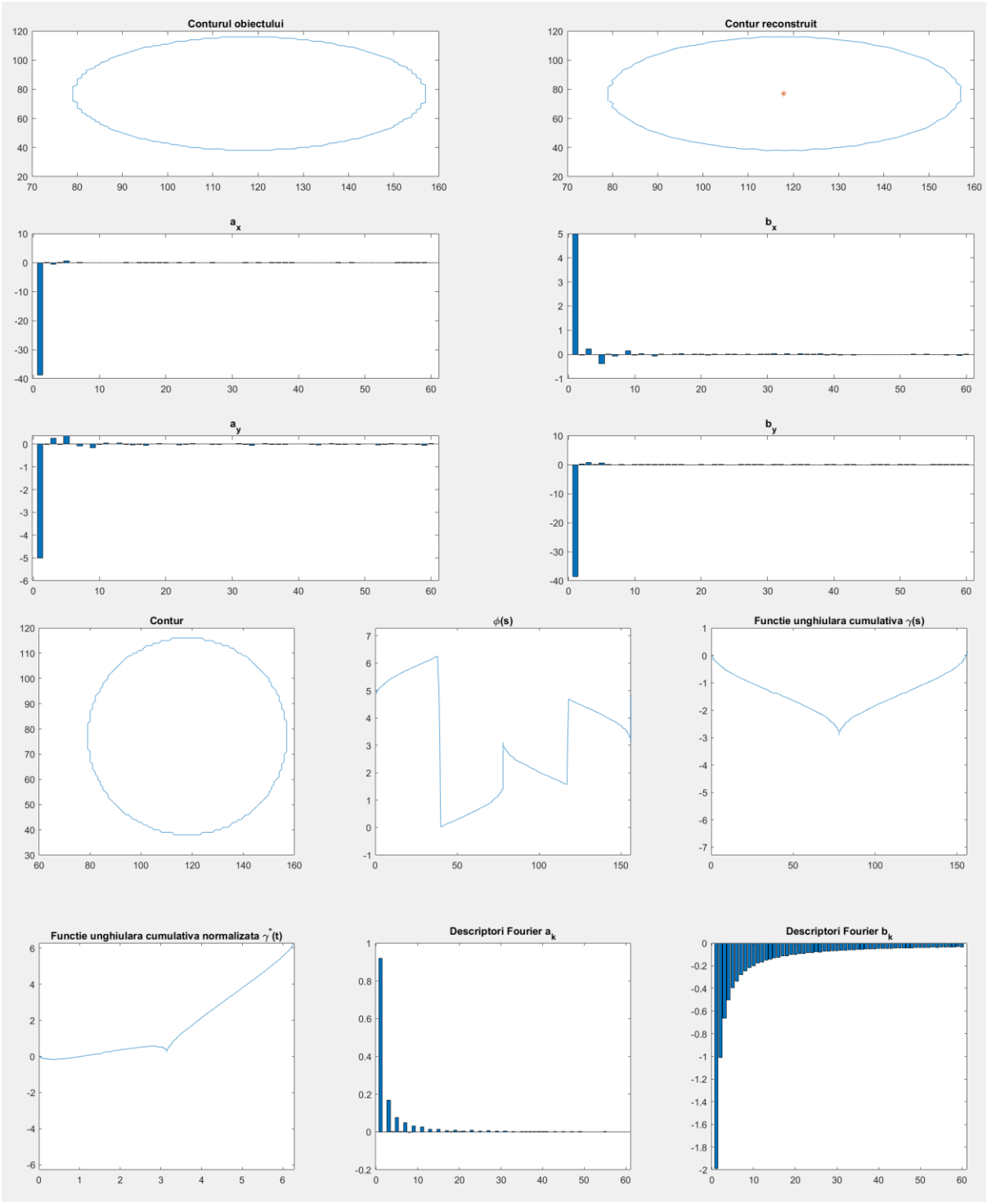

Conturul obiectului
Contur reconstruit
a_x
b_x
a_y
b_y
Contur
$\phi(s)$
Functie unghiulara cumulativa $\gamma(s)$
Functie unghiulara cumulativa normalizata $\gamma^*(t)$
Descriptori Fourier a_k
Descriptori Fourier b_k


Conturul obiectului
Contur reconstruit
a_x
b_x
a_y
b_y
Contur
ϕ(s)
Functie unghiulara cumulativa γ(s)
Functie unghiulara cumulativa normalizata γ*(t)
Descriptori Fourier a_k
Descriptori Fourier b_k

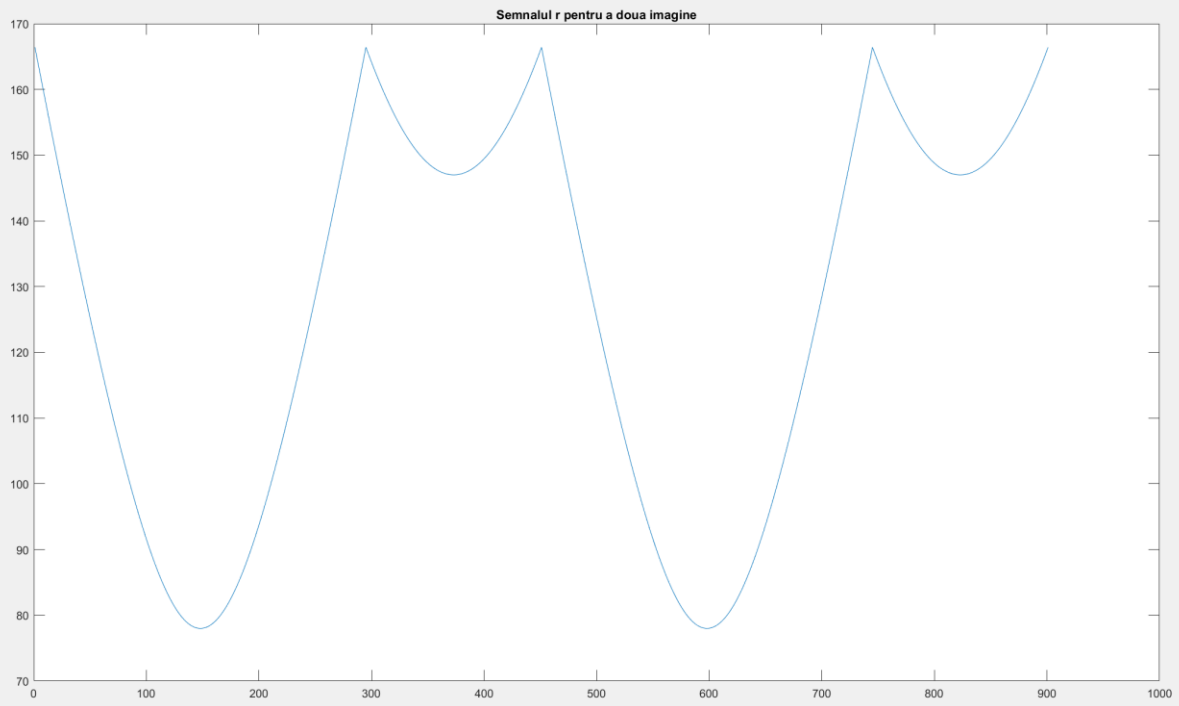
```
%% 60 coeficienti
nc = 60;
for (k=1:N)
    X = B{k}(:,2);
    Y = B{k}(:,1);
    contur = [X'; Y'];

    angularFunctionDescriptors(contur, nc);
   ellipticFourierDescriptors(contur, nc);

   %pause

end
```

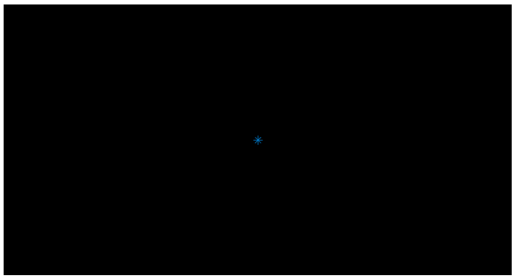


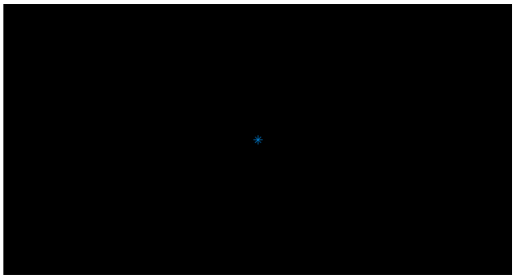

Conturul obiectului
Contur reconstruit
a_x
b_x
a_y
b_y
Contur
$\phi(s)$
Functie unghiulara cumulativa $\gamma(s)$
Functie unghiulara cumulativa normalizata $\gamma^*(t)$
Descriptori Fourier a_k
Descriptori Fourier b_k


Conturul obiectului
Contur reconstruit
a_x
b_x
a_y
b_y
Contur
$\phi(s)$
Functie unghiulara cumulativa $\gamma(s)$
Functie unghiulara cumulativa normalizata $\gamma^*(t)$
Descriptori Fourier a_k
Descriptori Fourier b_k

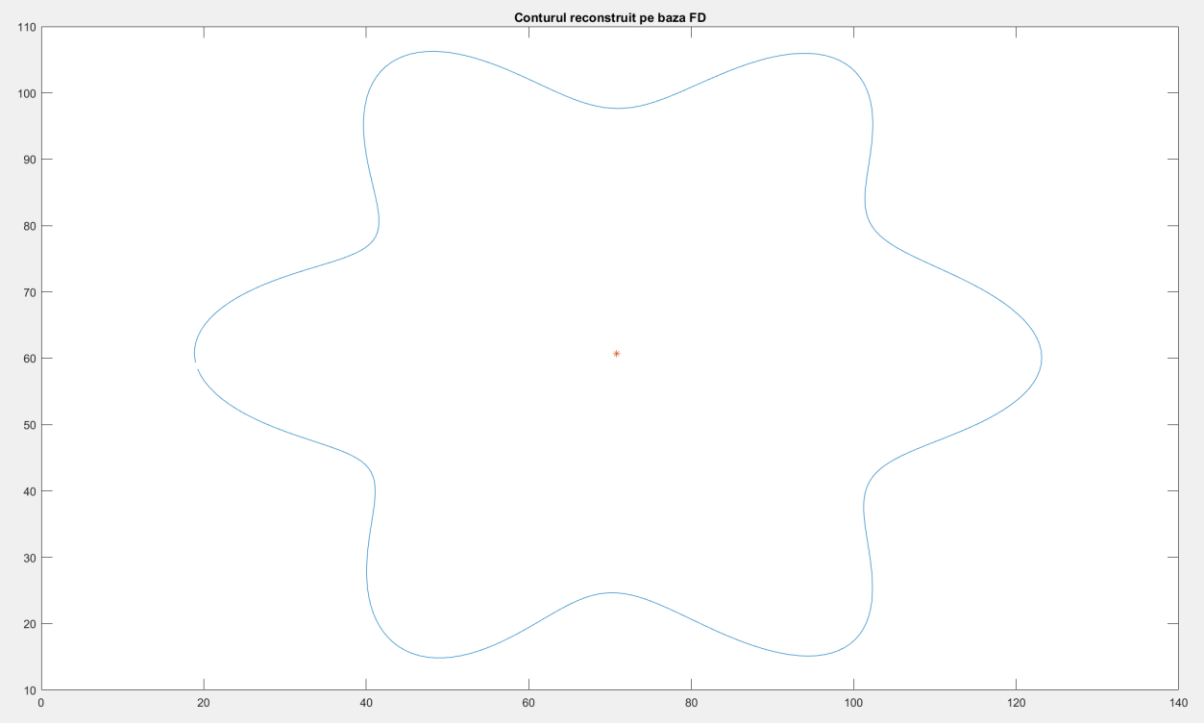
```
%% Exercitiul 2
clc
clear all
close all

% b
I1 = imread('drept.bmp');
BW1 = im2bw(I1);
BW1 = ~BW1;

[B1, L1, N1] = bwboundaries(BW1, 'noholes');

% c
X1 = B1{1}(:,2);
Y1 = B1{1}(:,1);
s1 = regionprops(L1, 'Centroid');
xc1 = s1.Centroid(1);
yc1 = s1.Centroid(2);

rho1 = sqrt((X1 - xc1).^2 + (Y1 - yc1).^2);

imshow(I1); hold on; plot(xc1, yc1, '*');
figure; plot(rho1);
title('Semnalul r pentru prima imagine');

% Prelucrarea celei de-a doua imagini
I2 = imread('drept.bmp'); % Incarcam a doua imagine
BW2 = im2bw(I2);
BW2 = ~BW2;

[B2, L2, N2] = bwboundaries(BW2, 'noholes');

X2 = B2{1}(:,2);
Y2 = B2{1}(:,1);
s2 = regionprops(L2, 'Centroid');
xc2 = s2.Centroid(1);
yc2 = s2.Centroid(2);

rho2 = sqrt((X2 - xc2).^2 + (Y2 - yc2).^2);

imshow(I2); hold on; plot(xc2, yc2, '*');
figure; plot(rho2);
title('Semnalul r pentru a doua imagine');

% f
R = corrcoef(rho1, rho2);
correlation = R(1, 2);

fprintf('Coeficientul de corelatie intre cele doua semnale r este: %.2f\n',
correlation);

if correlation > 0.9
    fprintf('Formele sunt similare.\n');
else
    fprintf('Formele nu sunt similare.\n');
end
```


Semnalul r pentru a doua imagine
170
160
150
140
130
120
110
100
90
80
70
0
100
200
300
400
500
600
700
800
900
1000





```
%% Exercitiul 3
clc
clear all
close all

ax(1) = -44.2,      ax(2) = 0.6,        ax(3) = 0.7,        ax(4) = -0.4,
      ax(5) = -4.8,       ax(6) = -0.01,      ax(7) = -3.7,
bx(1) = 1.2,        bx(2) = 0.1,        bx(3) = -0.03,      bx(4) = -0.2,
      bx(5) = 0.5,        bx(6) = 0.1,        bx(7) = 0.9,
ay(1) = -1.2,       ay(2) = -0.1,       ay(3) = 0.1,        ay(4) = 0.2,
      ay(5) = 0.6,        ay(6) = -0.3,       ay(7) = -0.6,
by(1) = -43.8,      by(2) = 0.8,        by(3) = 0.1,        by(4) = 0.4,
      by(5) = 3.7,        by(6) = -0.6,       by(7) = -3.6,
ax0 =  141.6, ay0 = 121.3,

m = 327;
nc = 7;
T = m; w = 2*pi/T;

for n = 1:m
    X2(n) = ax0/2;
    Y2(n) = ay0/2;
    for k = 1:nc
        X2(n) = X2(n) + ax(k)*cos(k*w*n)+bx(k)*sin(k*w*n);
        Y2(n) = Y2(n) + ay(k)*cos(k*w*n)+by(k)*sin(k*w*n);
    end
end

c2 = [X2; Y2];

figure, plot(c2(1,:),c2(2,:)), hold on, plot(ax0/2,ay0/2,'*'), title('Conturul
reconstruit pe baza FD')
```


Conturul reconstruit pe baza FD
110
100
90
80
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140