



Funções de Minimização



FZERO

“Scalar nonlinear zero finding”

FMINSEARCH

“Multidimensional unconstrained nonlinear minimization”

FMINBND

“Scalar bounded nonlinear function minimization.”



fzero:

Encontra o ponto que a função é zero.

Atenção! Não é o ponto de menor valor!

```
%grafico da funcao
x=-10:0.01:15;
y=((x-3).^2)-6;
plot(x,y)

hold on

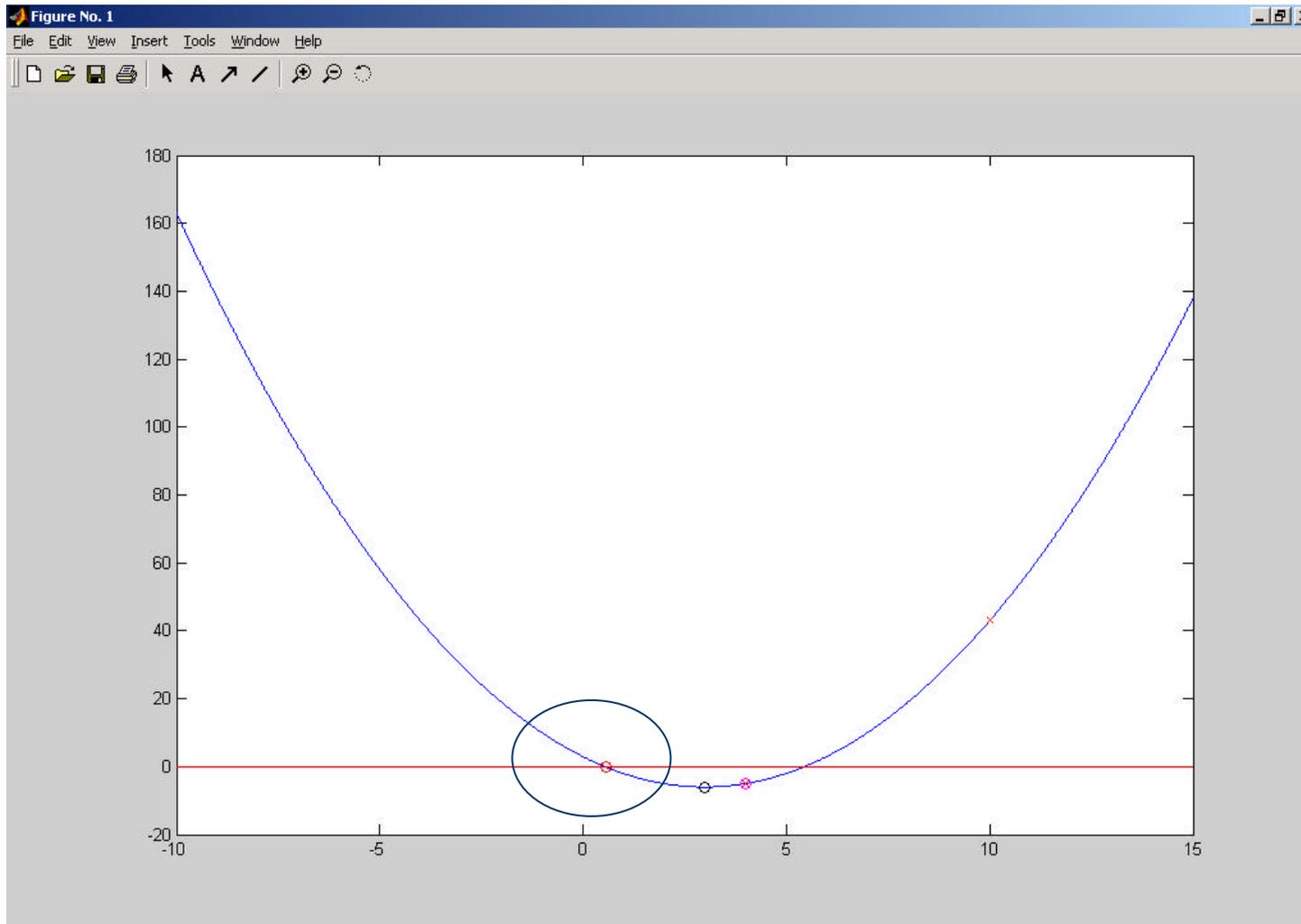
%linha base (linha zero)
xbase=x;
ybase=zeros(length(xbase),1);
plot(xbase,ybase,'-r');

%Achando o ponto zero da funcao

% Usando fzero:
%
options = optimset('display','iter')

% resultado = fzero( 'equacao' , chute inicial, opcoes avancadas)
Xzero = fzero('((x-3).^2)-6', -4, options)

%marca o ponto que a funcao eh zero
plot(Xzero,0,'or')
```



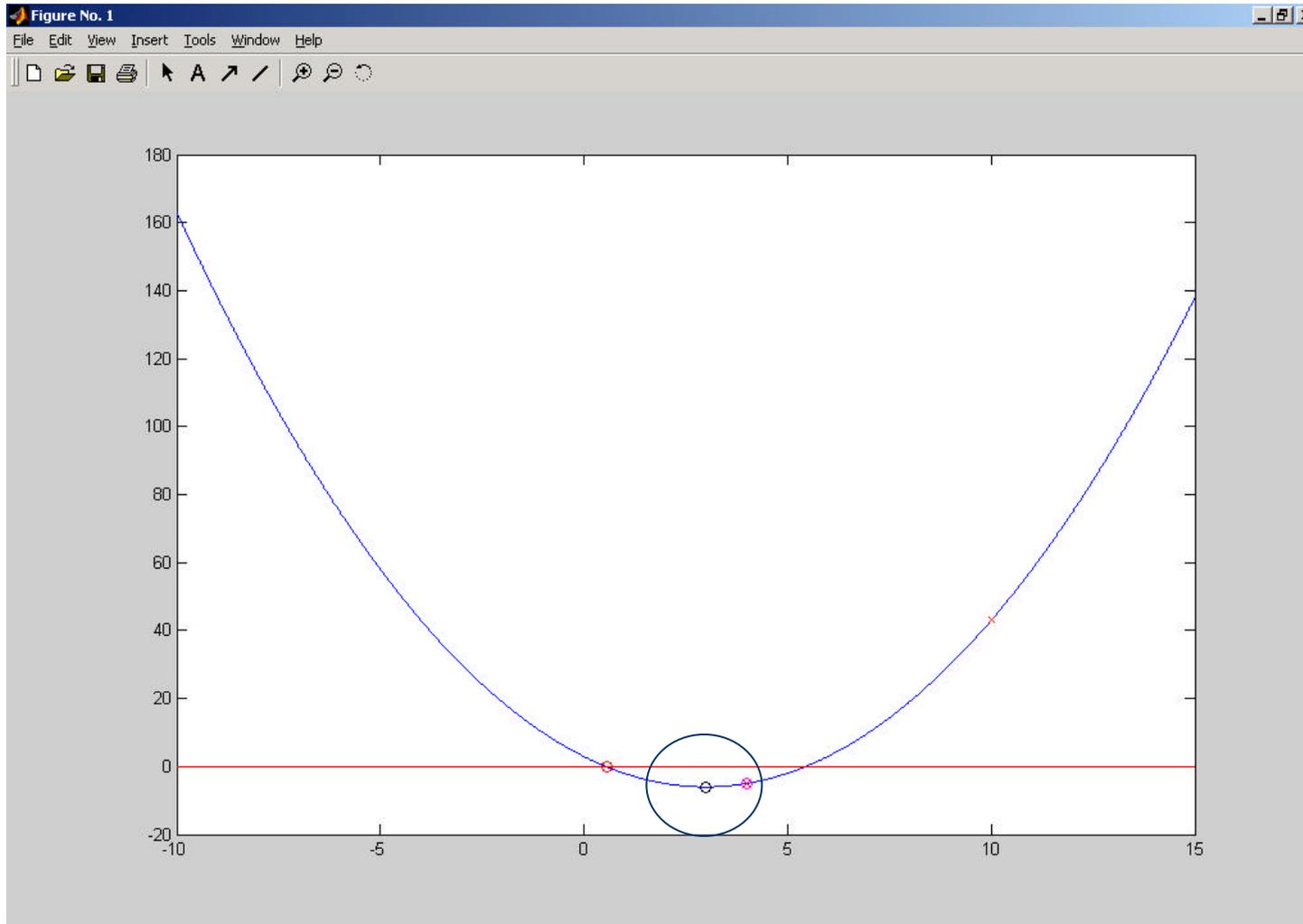


fminsearch: Minimiza a função sem restrições.

```
% Achando o minimo de uma funcao

% Usando fminsearch:
%
% resultado = fminsearch('equacao', chute inicial, opcoes avancadas)
Xmin = fminsearch('((x-3).^2)-6', 4, options)

%marca o ponto de minimo da funcao
Ymin=(Xmin-3).^2-6;
plot(Xmin,Ymin,'ok')
```





fminbnd:

Minimiza a função com restrições.

```
% Achando o minimo de uma funcao com restricoes
% (a busca eh realizada apenas dentro de um intervalo de x)

% Usando fminbnd

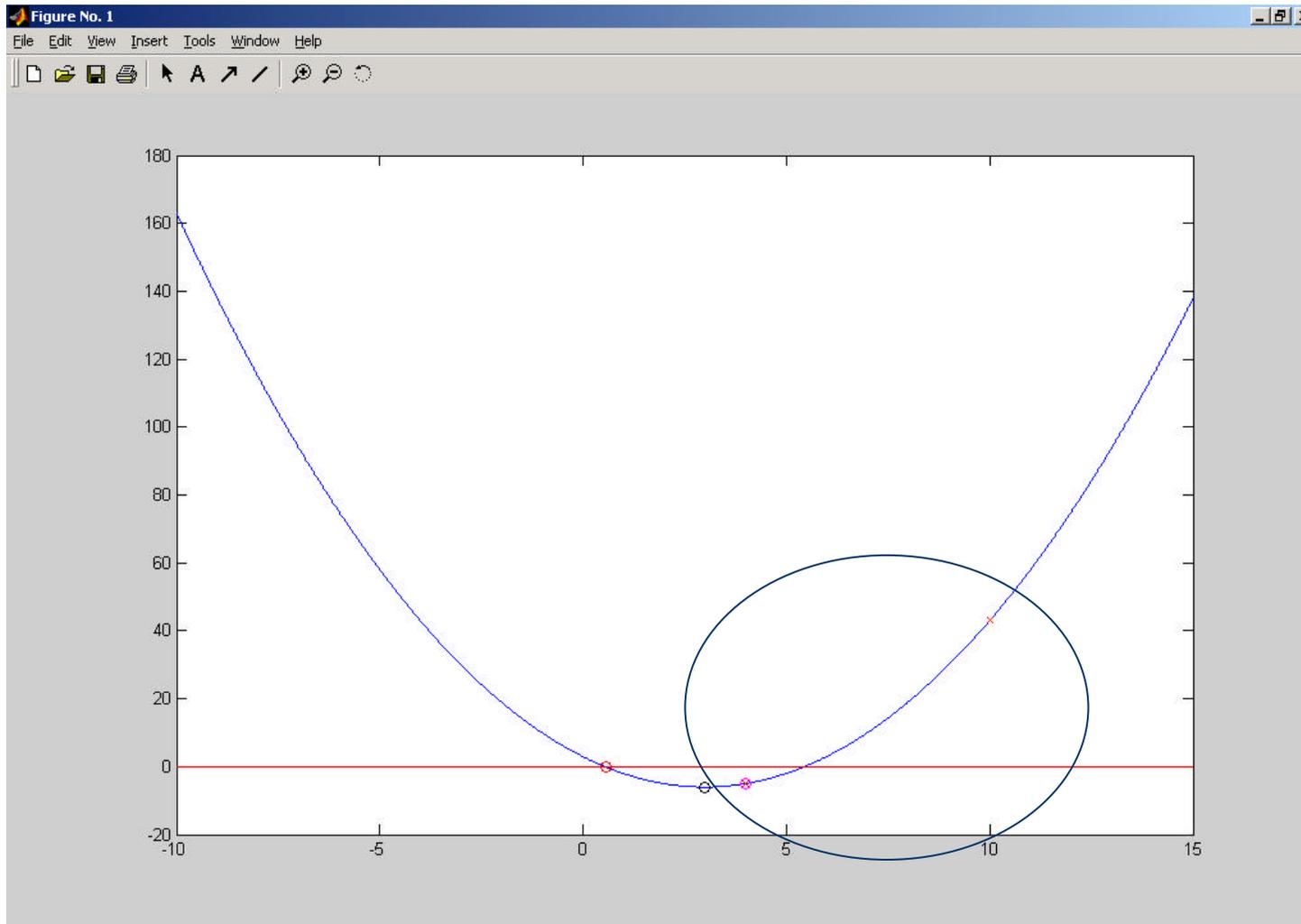
%restricoes:
ri=4;
rs=10;

% resultado = fminbnd('equacao', restricao inferior, restricao superior, opcoes avancadas)
Xmin = fminbnd('((x-3).^2)-6', ri, rs, options)

%marca o ponto de minimo da funcao
Ymin=((Xmin-3).^2)-6;
plot(Xmin,Ymin,'om')

yi=((ri-3).^2)-6;
plot(ri,yi,'xr')

ys=((rs-3).^2)-6;
plot(10,ys,'xr')
```





Observação:

O “fminsearch” permite minimizar funções multidimensionais

```
Xminduplo=fminsearch('exp(x(1)) * ( 4*x(1)^2 + 2*(x(2)^2 + 4*x(1)*x(2) + 2*x(2) + 1) ) ',  
[1 2] , options)
```



Otimização de parâmetros:

O objetivo é minimizar a função erro:

Programa principal:

```
global yexp xexp

%pontos experimentais:
yexp=[1.1 2.12 2.85 4.4 5.0 6.5];
xexp=[1 2 3 4 5 6];

Parametros = fminsearch('equacao', [1 2], options)
```



Função a ser minimizada:

```
function [saida] = equacao(par)
global yexp xexp

a=par(1);
b=par(2);

yteo=a.*xexp+b; %ajuste de primeira ordem

yerro=abs(yexp-yteo); %calculo do erro
saida=sum(yerro);

plot(xexp,yexp,'r*',xexp,yteo,'b-') %resultado na forma grafica
drawnow
```

