

# SPRAWOZDANIE Z TEORII OPTYMALIZACJI

Imię, Nazwisko, Numer	Michał Krzyszczyk N=14
Temat ćwiczenia	Metody gradientowe II
Data i godzina wykonania ćwiczenia	3 kwietnia 2019, godz: 14:30

## Zadanie 1.

Obserwacja zjawiska zygzakowania Należy przeanalizować wpływ wydłużenia zbiorów poziomicowych funkcjonału kwadratowego i punktu startowego na tempo zbieżności metody najszybszego spadku. Badania przeprowadzić dla funkcjonału:  
 $Q(x_1, x_2) = x_1^2 + a \cdot x_2^2$  Wybierając kolejno  $a = 1$ ,  $a = 0.5$ ,  $a = 0.2$  dla  $x_0 = [3 \cdot 14; 9 \cdot 14]$

## Metoda analityczna.

Funkcjonał celu jest sumą dwóch wyrażeń nieujemnych, zerujących się w zerze.

$$x_1 = 0 \wedge x_2 = 0 \quad Q(0,0) = 0$$

## Rozwiązanie numeryczne.

*koszt*

```
function [q,x]=koszt(x,z,d)

% KOSZT wylicza wskaźnik jakości dla wektora zmiennych
% decyzyjnych x+z*d.

if nargin==2, x=x+z;
elseif nargin==3, x=x+z*d;
end
q=x(1)^2+1*x(2)^2;
```

*gradie*

```
function g=gradie(x)
% GRADIE Wyznacza analitycznie gradient funkcji kosztu w
punkcie X.
g=[2*x(1);2*1*1*x(2)];
```

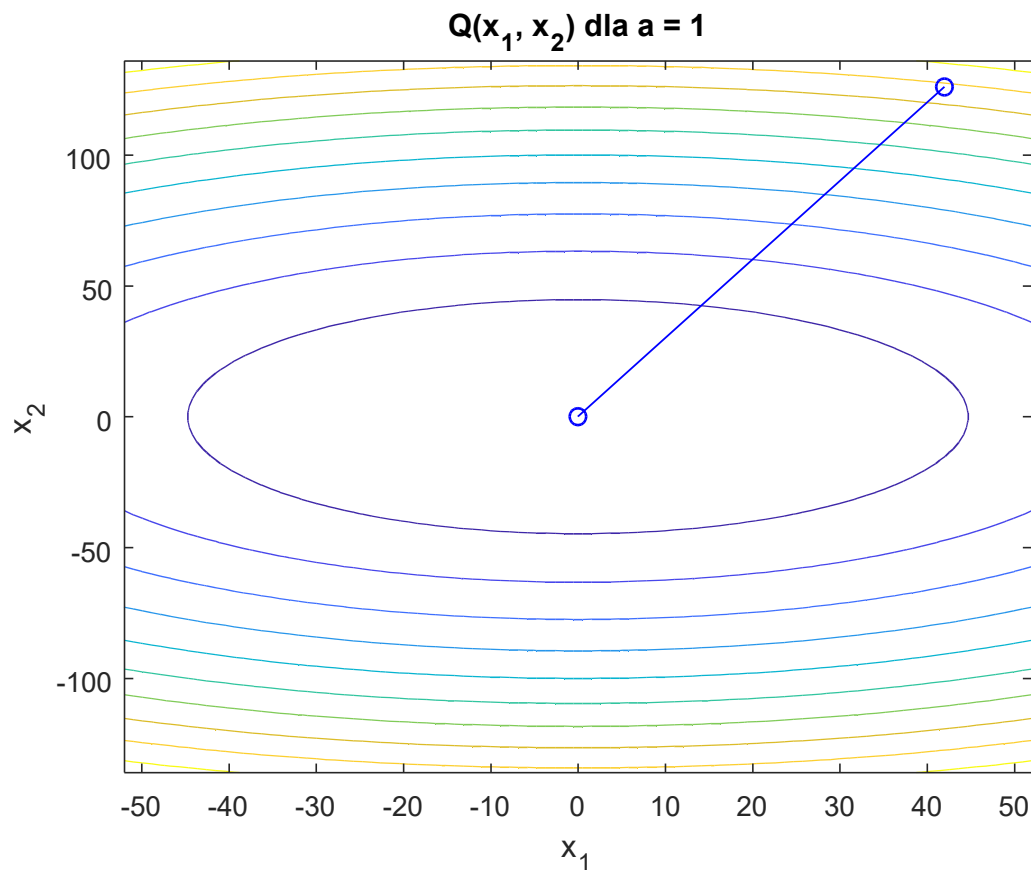
```

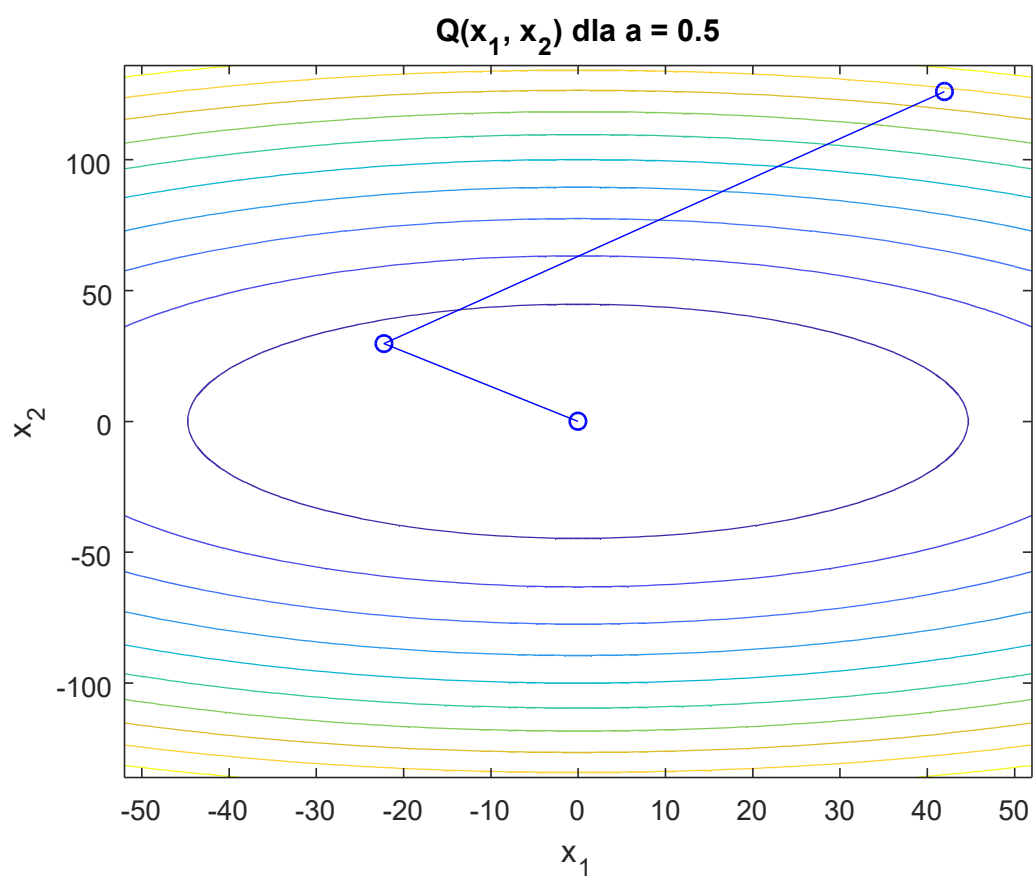
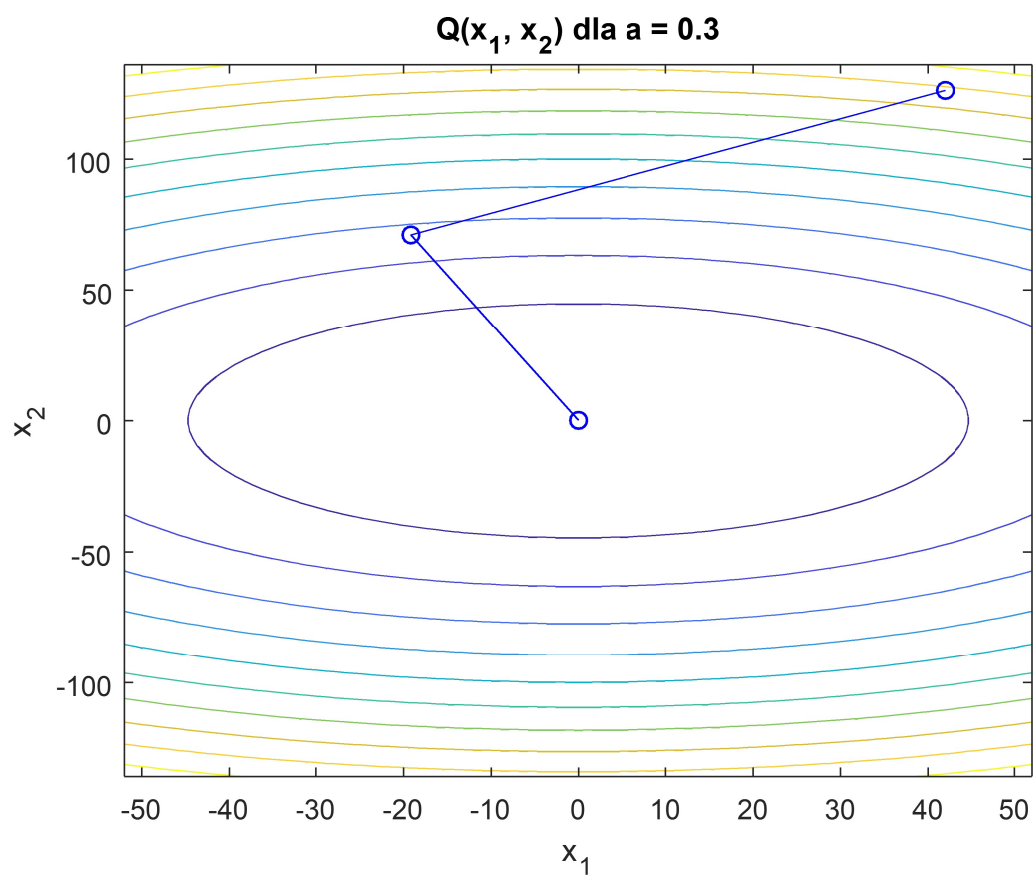
close all;
clear all;

n = 14;
x0 = [3*n; 9*n];
etap = [];
a = 1;
[x1, x2] = meshgrid(linspace(-3*n-10, 3*n+10), linspace(-9*n-10, 9*n+10));
q = @(x1, x2) x1.^2 + 1*x2.^2;
figure()
contour(x1, x2, q(x1,x2));
hold on
par = 0; %
zmenad
plot(etap(1,:), etap(2,:), 'b');
plot(etap(1,:), etap(2,:), 'bo');
ylabel('x_2');
xlabel('x_1');
title('Q(x_1, x_2) dla a = 1');

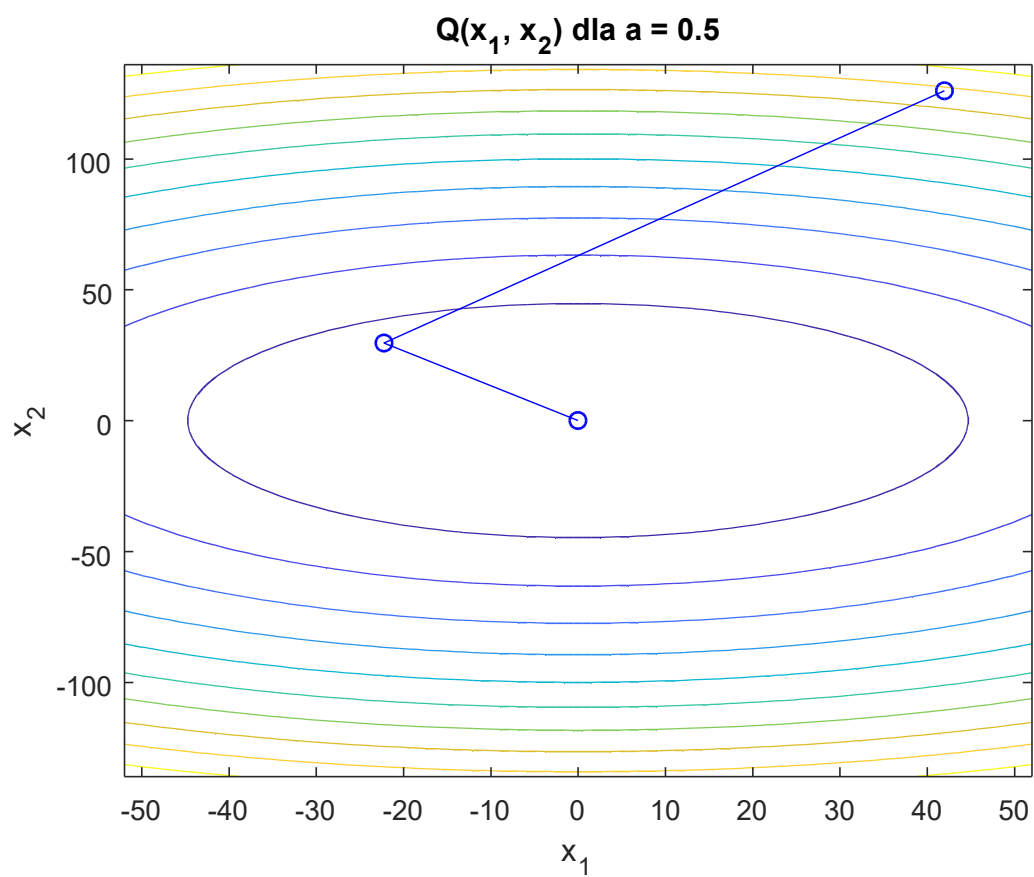
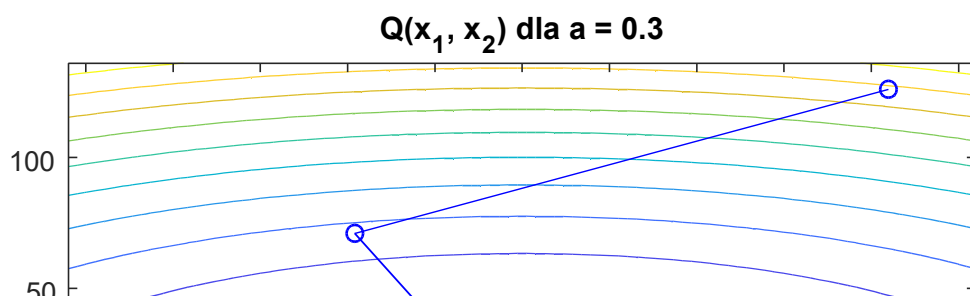
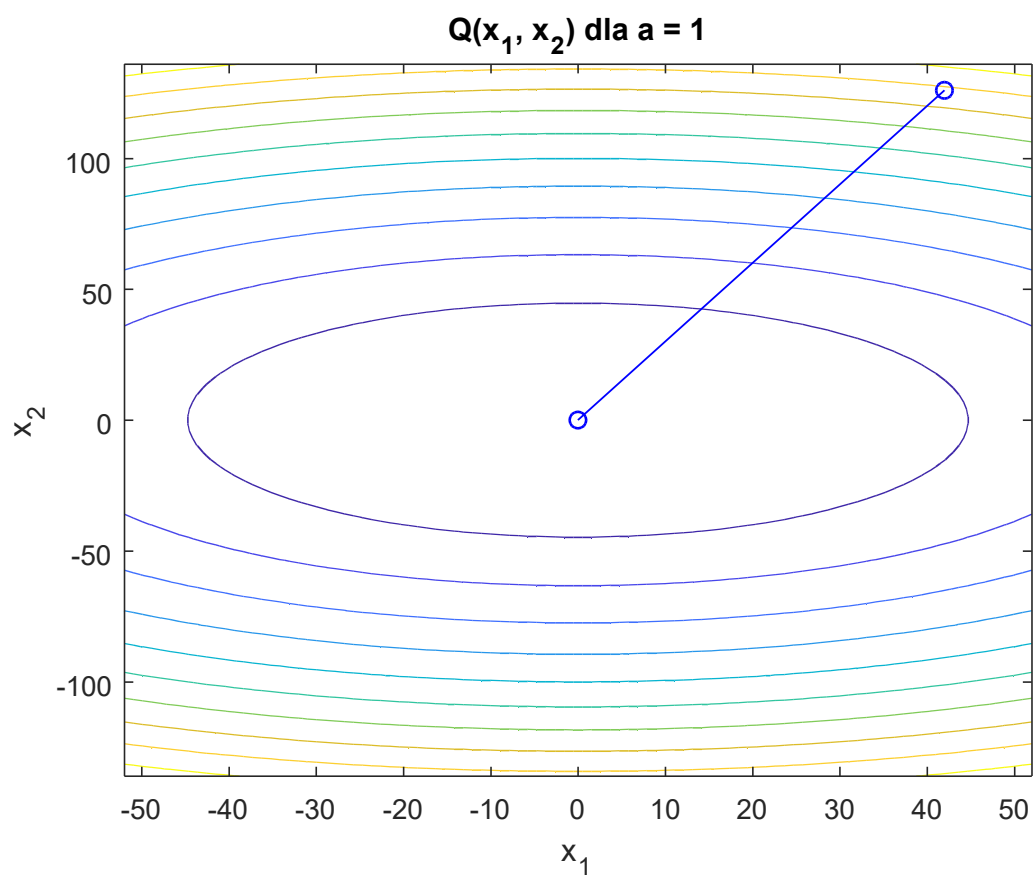
```

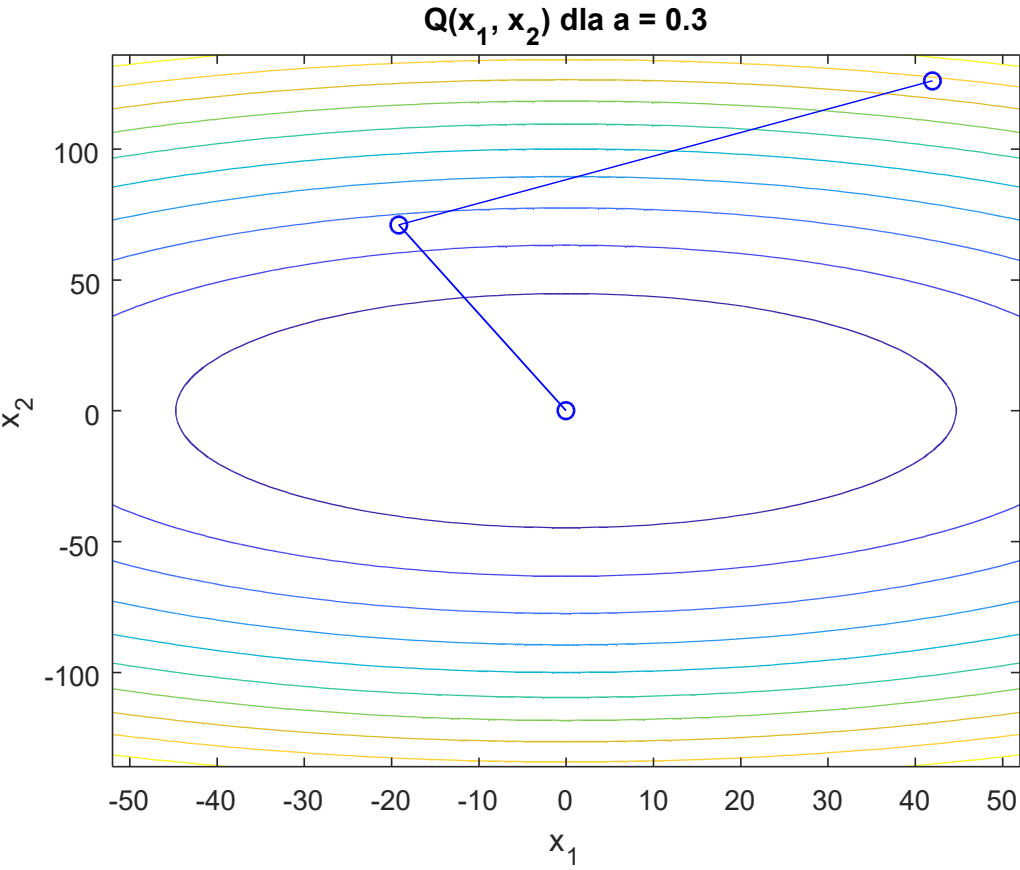
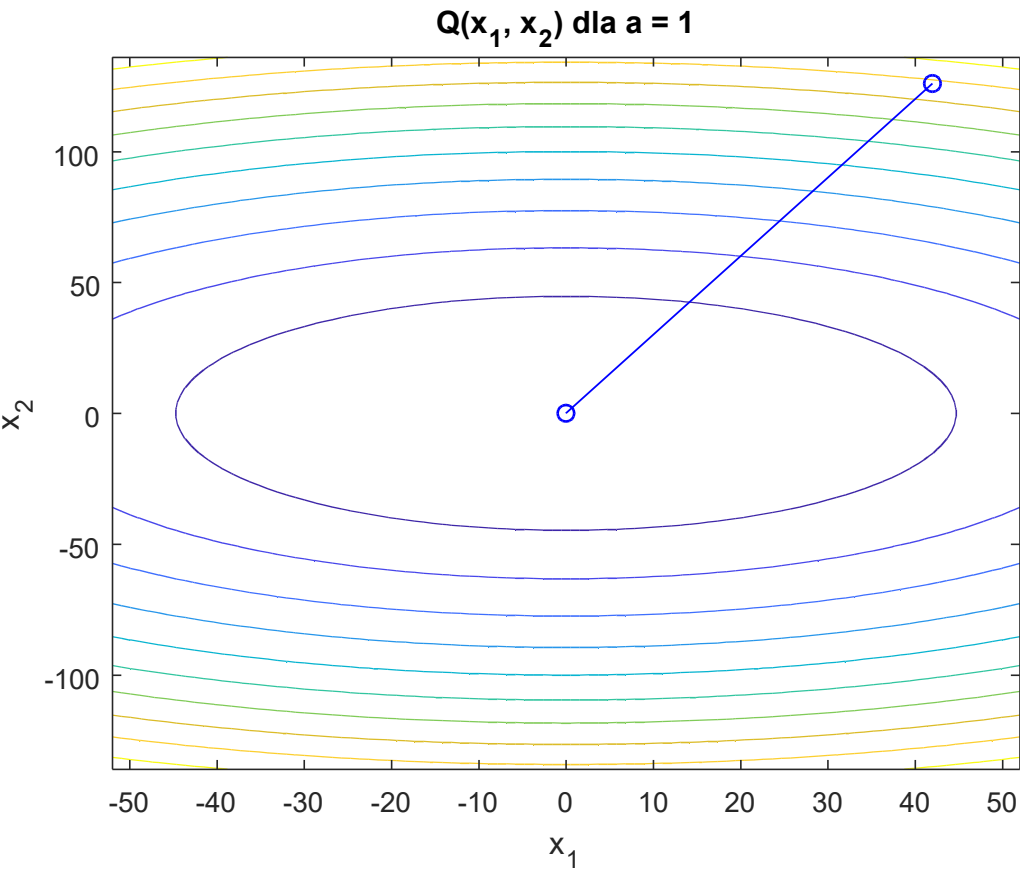
Davidona - Fletcher - Powell

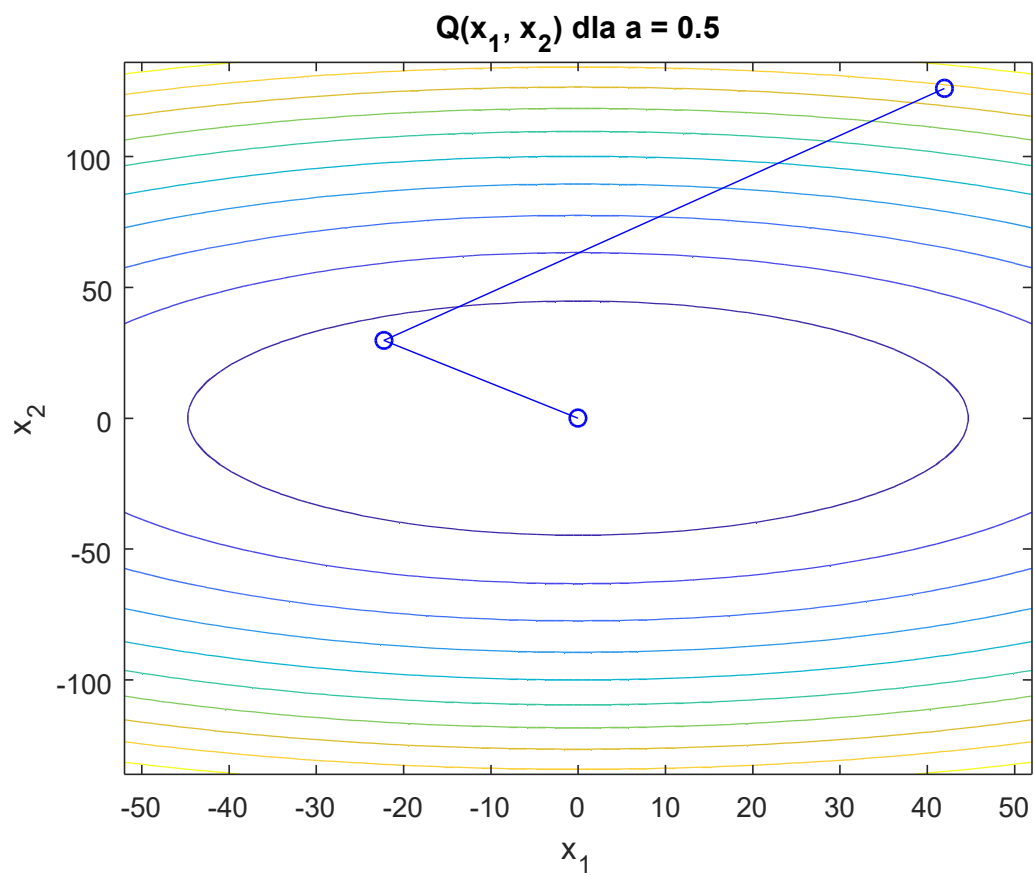




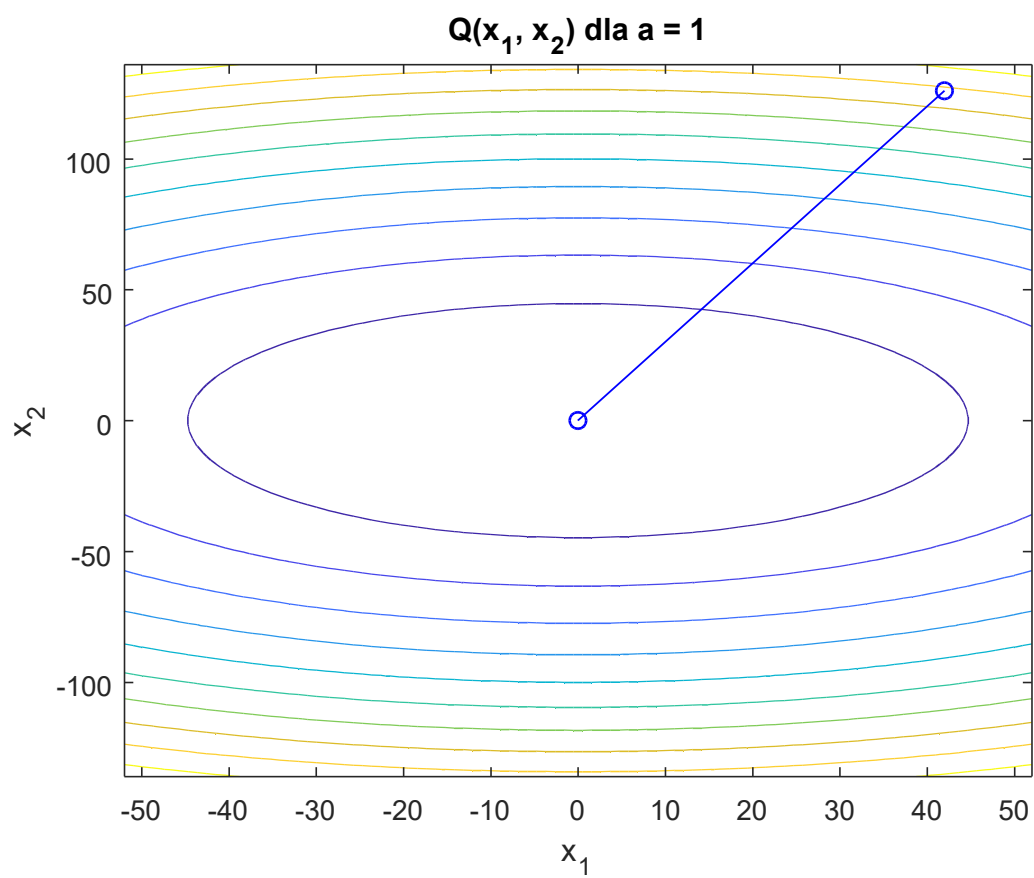
Wolfe'a - Broydena - Davidona

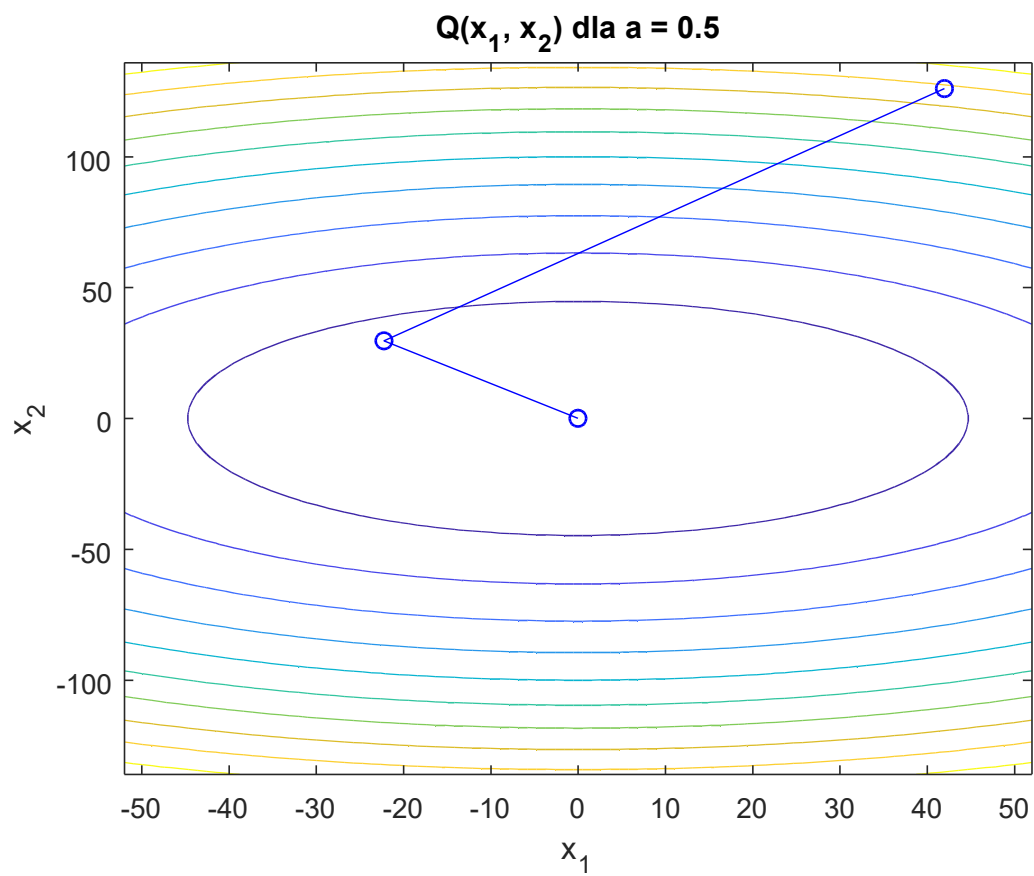
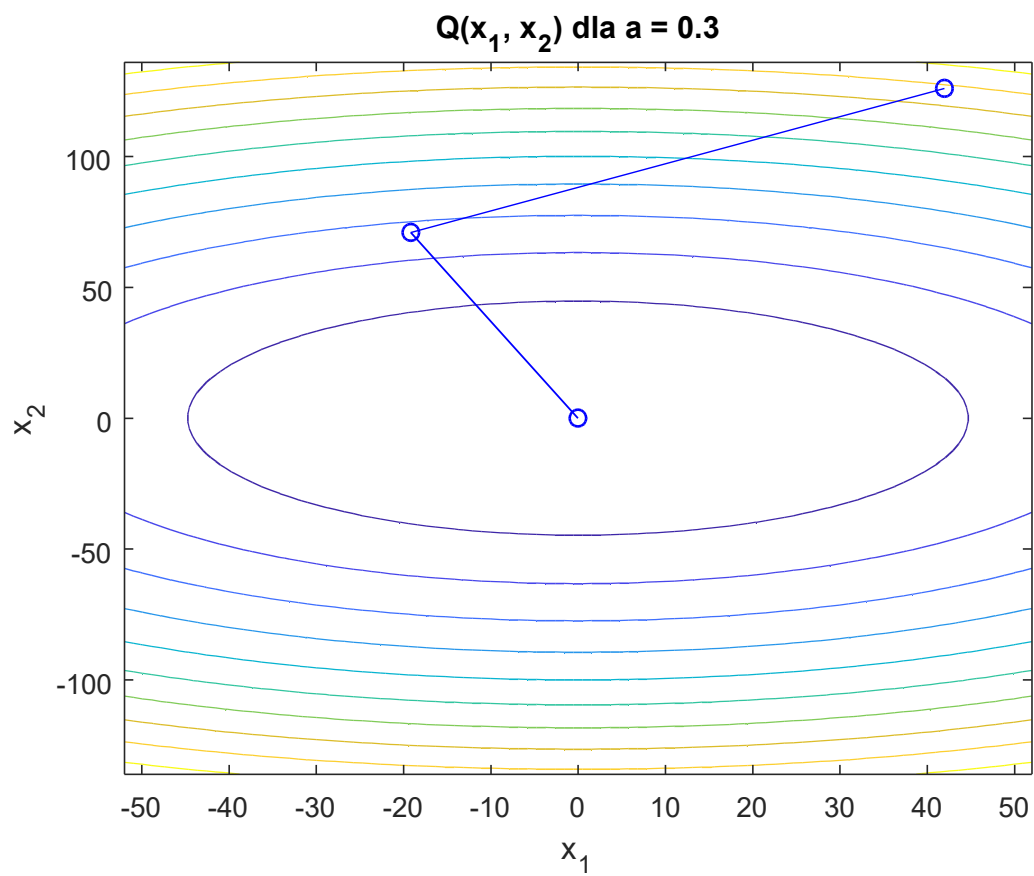


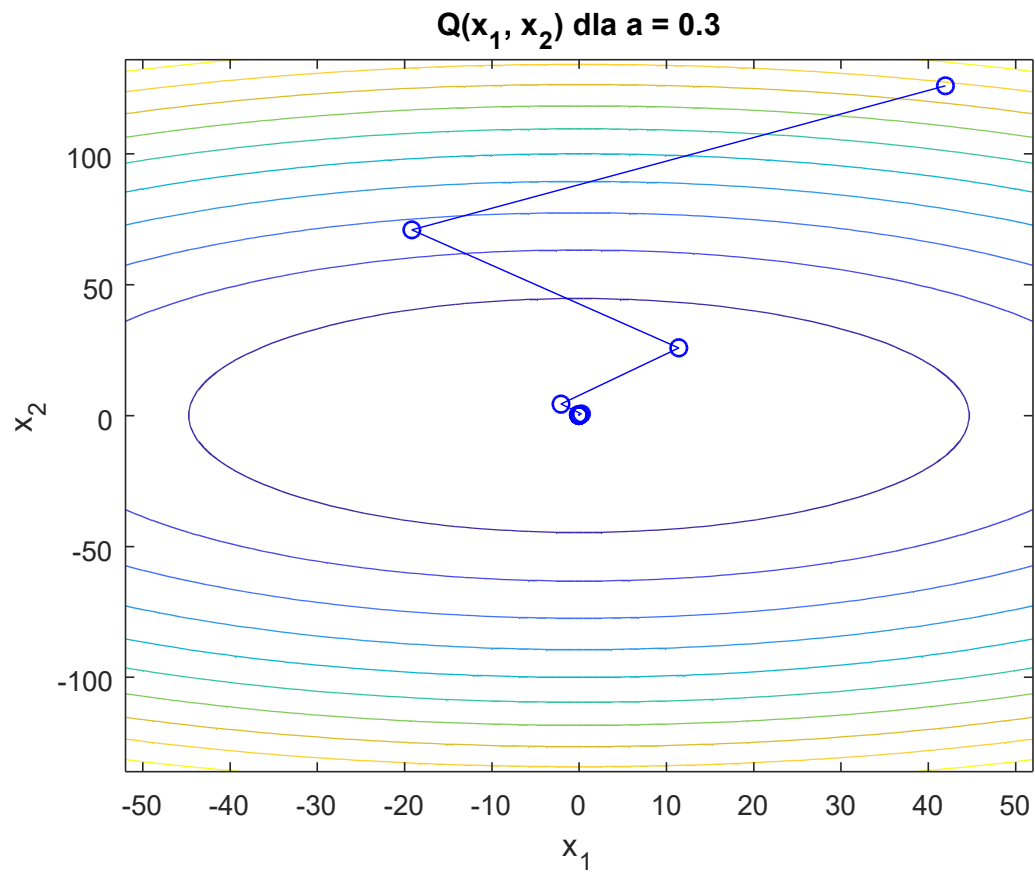
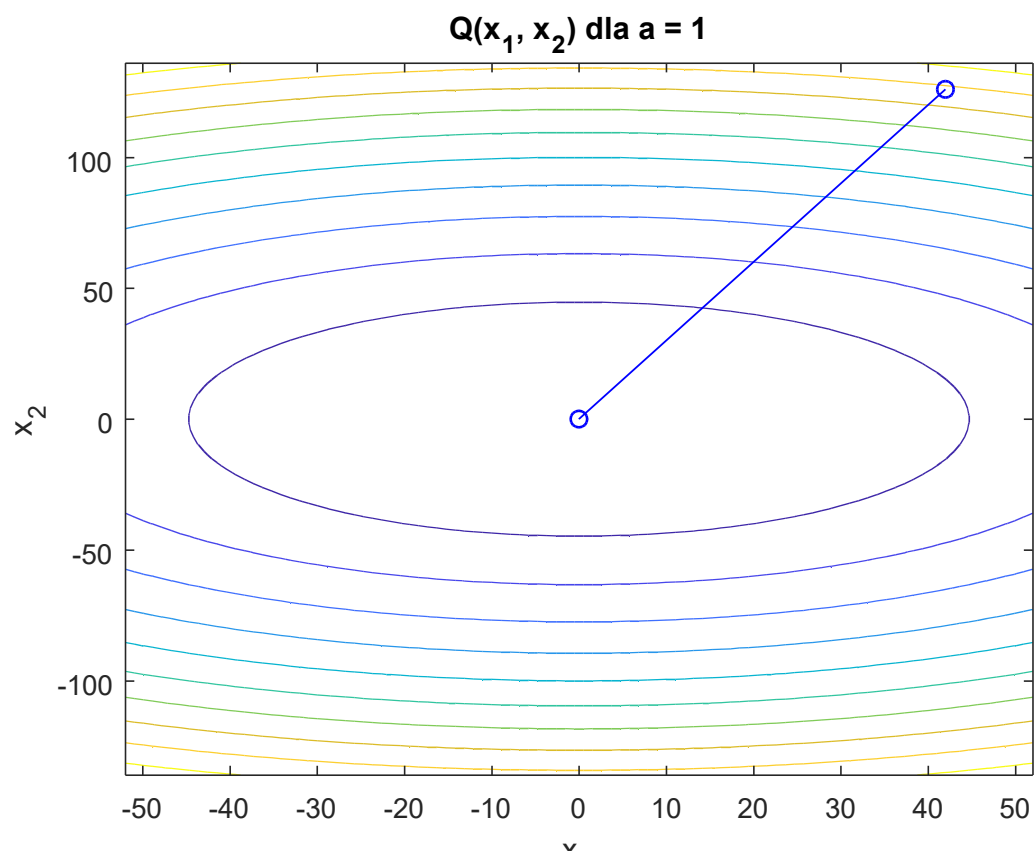




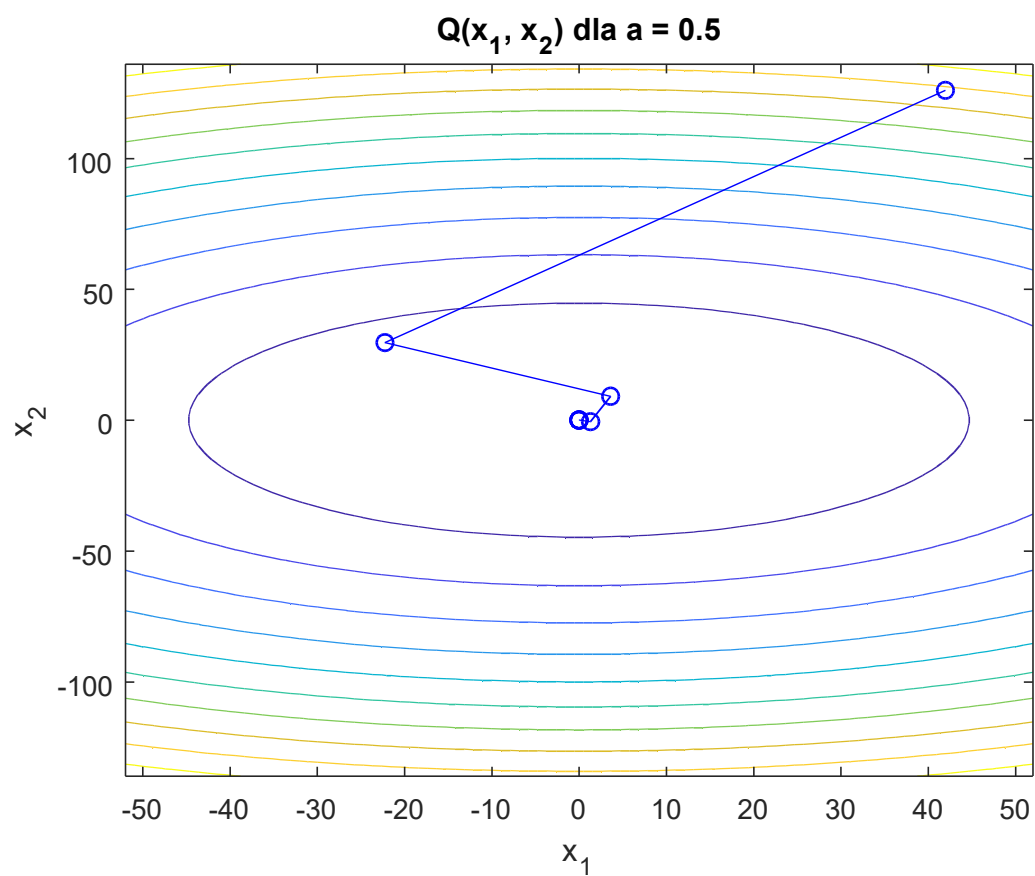
Pearsona 1



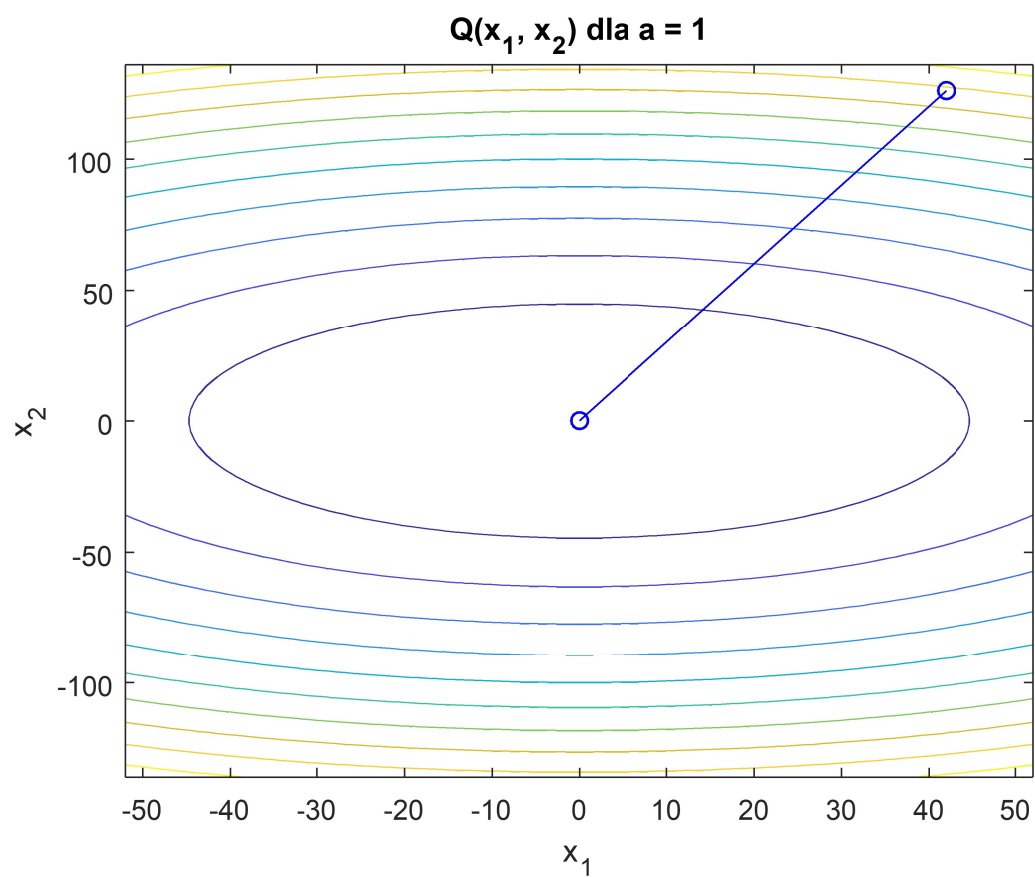


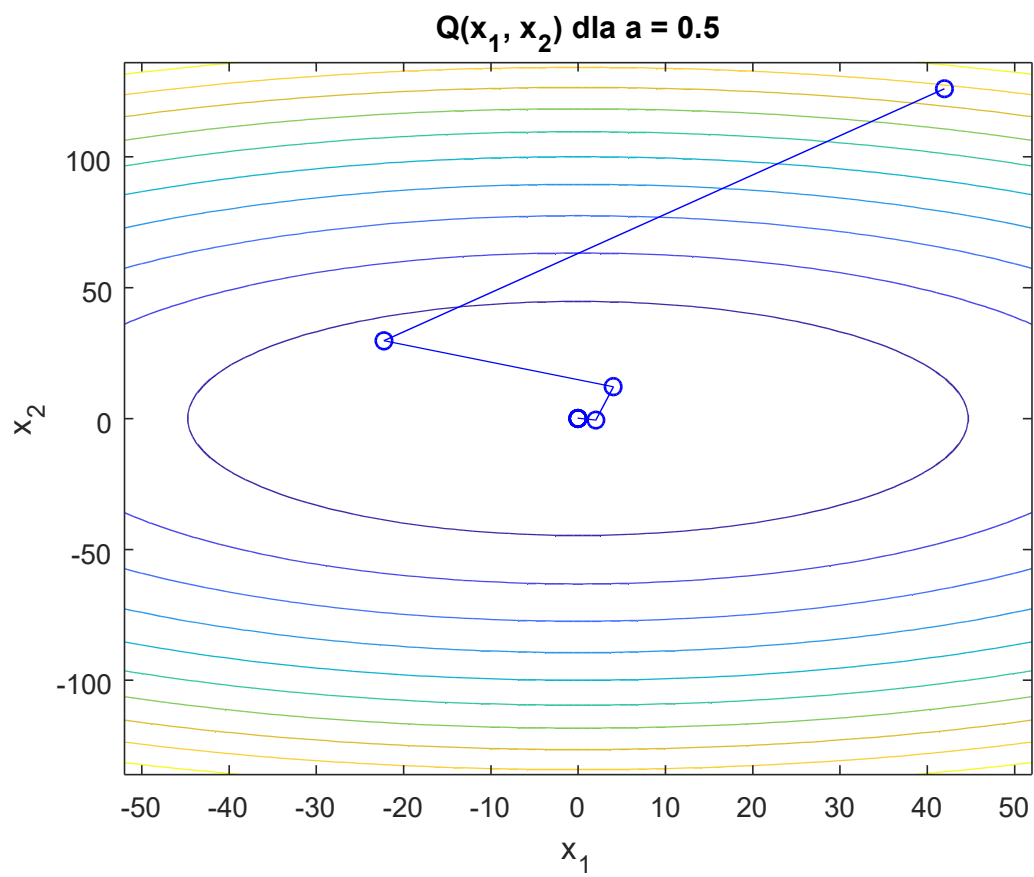
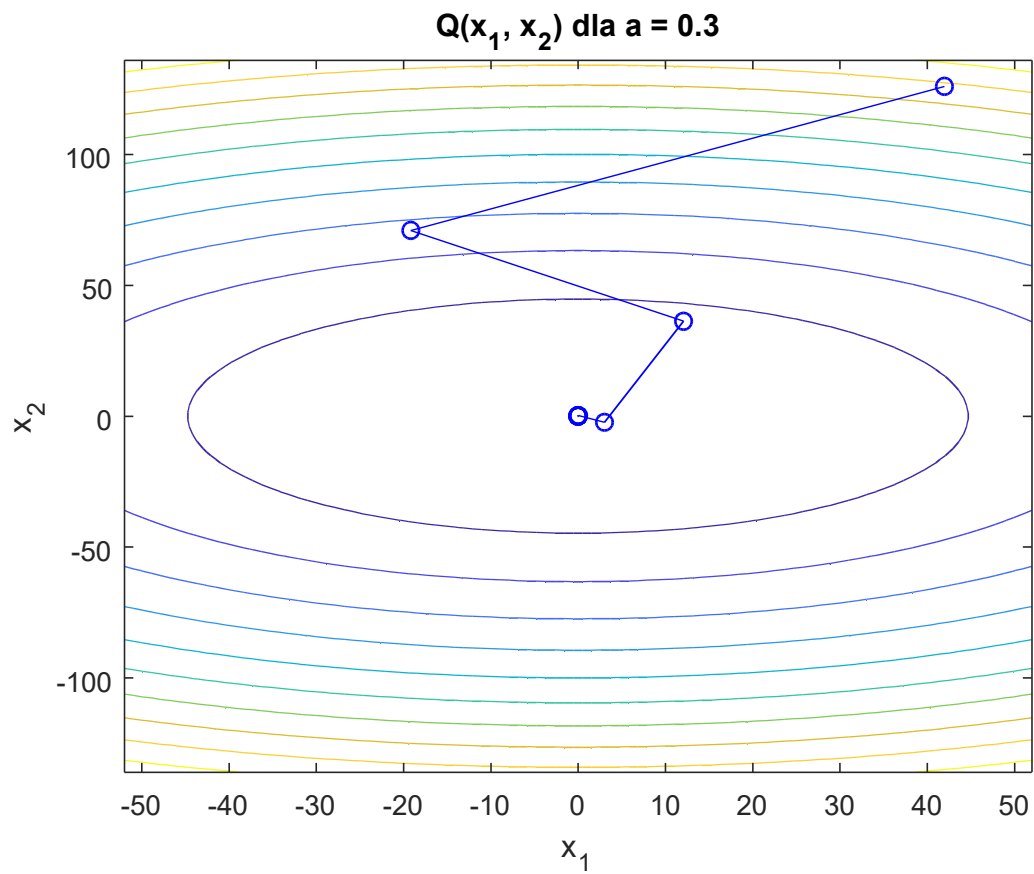






McCormicka





## Zadanie 2.

Badanie porównawcze gradientowych metod optymalizacji Badanie porównawcze gradientowych metod optymalizacji przeprowadzić dla funkcjonału:

$$Q(x_1, x_2) = 6x_1^2 + 6x_1x_2 + x_2^2 + 4.5(e^{x_1} - x_1 - 1) + 1.5(e^{x_2} - x_2 - 1)$$

$$x_0 = [3/100 \ 9/100] * N$$

## Rozwiązanie numeryczne.

```
function [q,x]=koszt(x,z,d)

% KOSZT wylicza wskaźnik jakości dla wektora zmiennych
% decyzyjnych x+z*d.

if nargin==2, x=x+z;
elseif nargin==3, x=x+z*d;
end
f = @(x1, x2) 6*x1.^2 + 6*x1*x2 + x2.^2 + 4.5*(exp(x1) - x1 - 1) + 1.5*(exp(x2) - x2 - 1);
q = f(x(1),x(2));
```

```
function g=gradie(x)

% GRADIE Wyznacza analitycznie gradient funkcji kosztu w punkcie X.

g=[12*x(1)+6*x(2)+4.5*(exp(x(1))-1);
6*x(1)+2*x(2)+1.5*(exp(x(2))-1)];
```

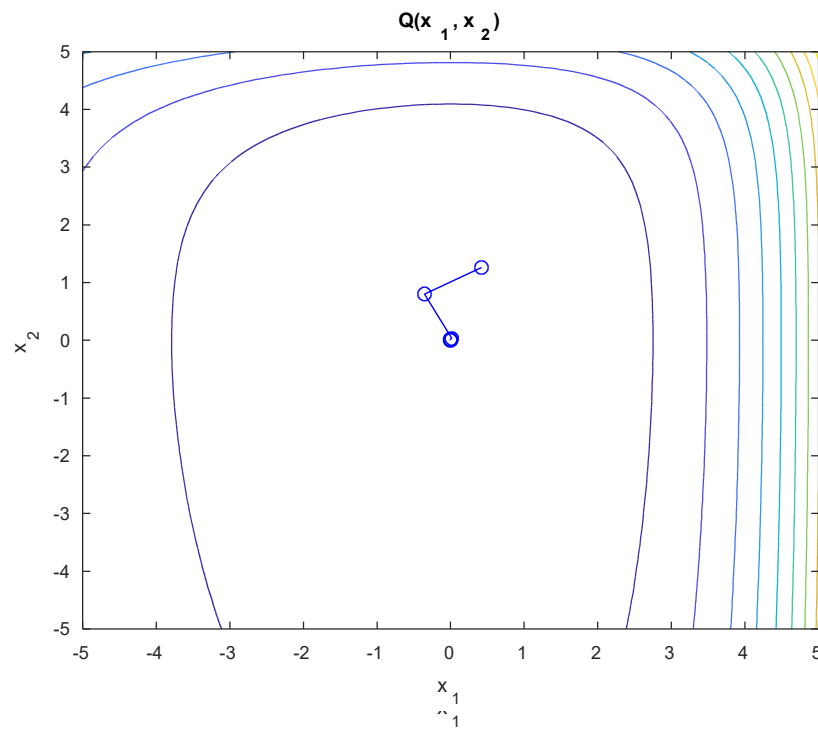
```

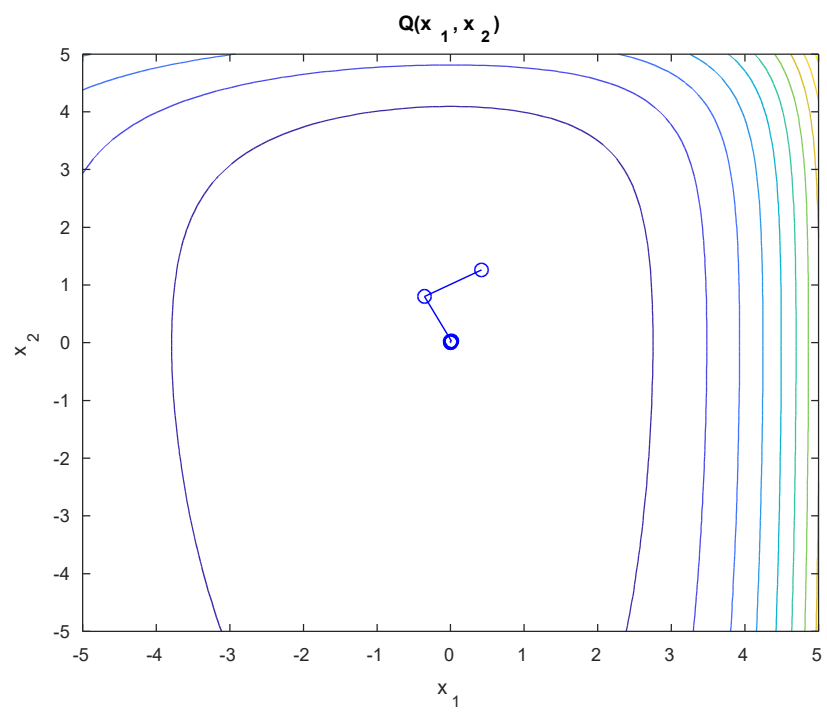
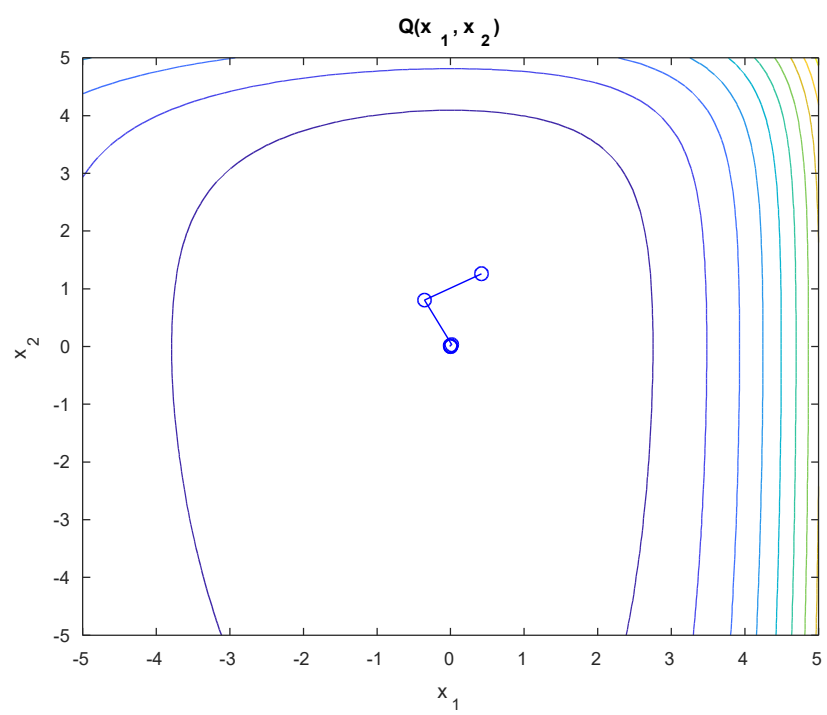
close all;
clear all;
for iii=0:5

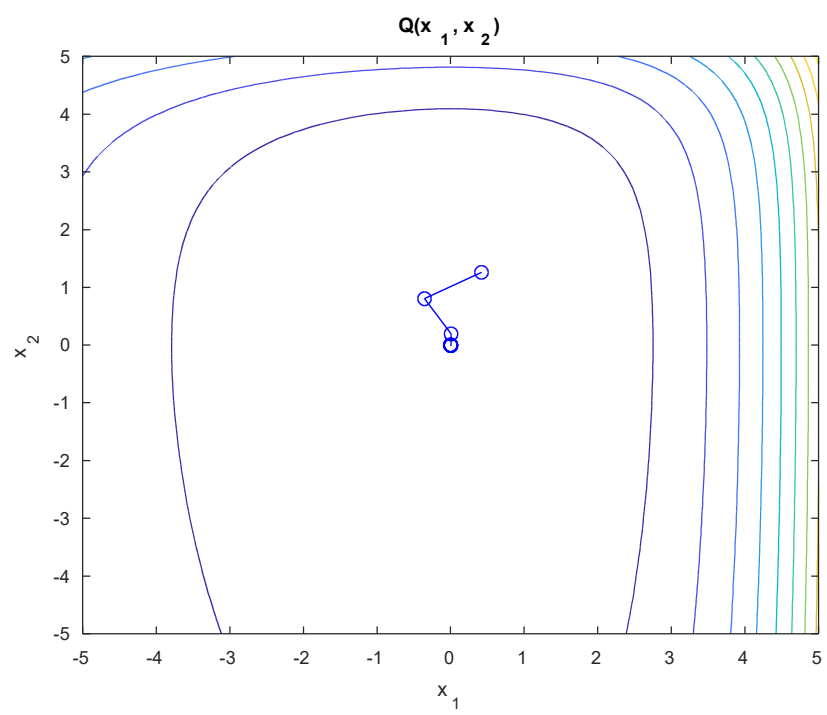
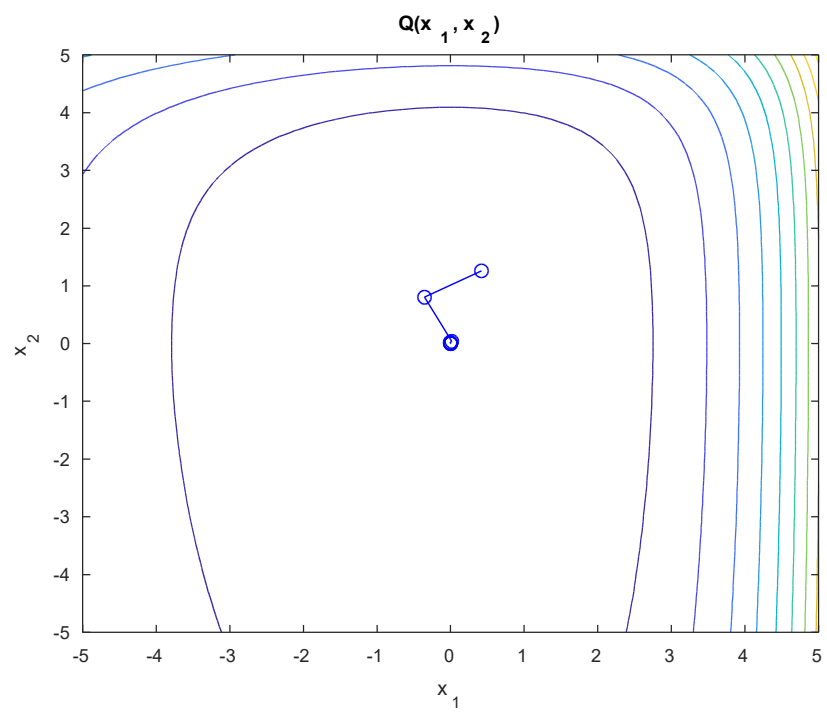
n = 14;

x0 = [3/100; 9/100]*n;
etap = [];
a = 1;
[x1, x2] = meshgrid(linspace(-5, 5), linspace(-5, 5));
q = @(x1, x2) 6*x1.^2 + 6*x1*x2 + x2.^2 + 4.5*(exp(x1) - x1 - 1) + 1.5*(exp(x2) - x2 - 1);
figure(1)
contour(x1, x2, q(x1,x2));
hold on
par = 0; %
zmenad
plot(etap(1,:), etap(2,:), 'b');
plot(etap(1,:), etap(2,:), 'bo');
ylabel('x_2');
xlabel('x_1');
title('Q(x_1, x_2)');
s = strcat(num2str(iii), 'plot', '.emf');
saveas(1,s);
close all;
end

```







Znalezione rozwiązanie:

```
x0 =  
  
1.0e-03 *  
  
-0.1589  
0.2702
```

Kolejne wykresy, obrazują przybliżenia dla różnych wartości zmiennej par=0:5

## Funkcja fminsearch

Dla sprawdzenia, wywołano funkcję MATLAB:

```
>> fun= @(x) 6*x(1).^2 + 6*x(1)*x(2) + x(2).^2 + 4.5*(exp(x(1)) - x(1) - 1) + 1.5*(exp(x(2)) -  
x(2) - 1);  
>> rozwiazanie_fmin = fminsearch(fun,x0)  
  
rozwiazanie_fmin =  
  
1.0e-03 *  
  
-0.1589  
0.2702
```