SPRAWOZDANIE Z TEORII OPTYMALIZACJI

Imię, Nazwisko, Numer	Michał Krzyszczuk N=14
Temat ćwiczenia	Metody gradientowe
Data i godzina wykonania ćwiczenia	27 marca 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(x1,x2) = x2 1 + a·x2 2 Wybierając kolejno a = 1, a = 0.5, a = 0.2 dla x0= [3*14;9*14]

Metoda analityczna.

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

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

Rozwiązanie numeryczne.

koszt.m

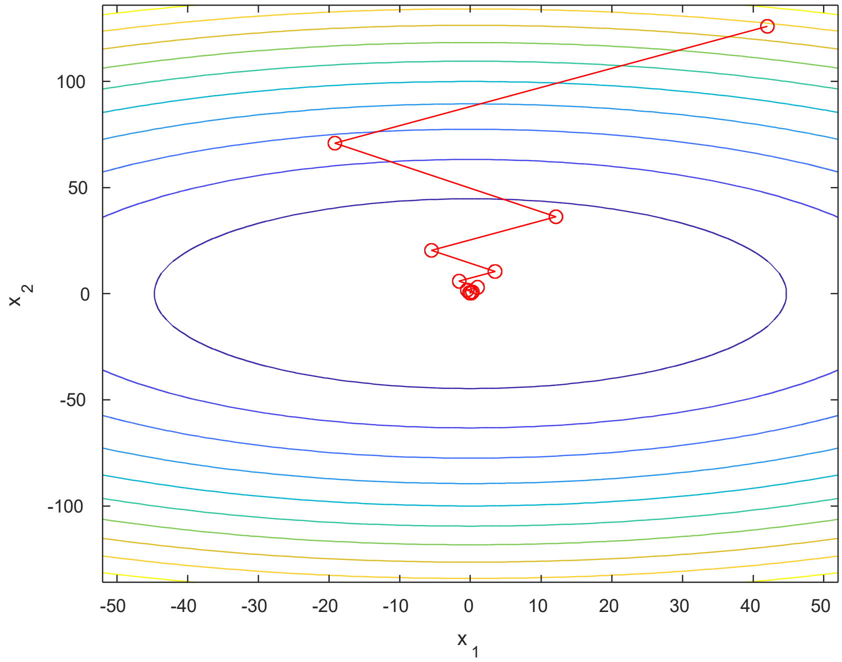
```
function [q,x]=koszt(x,z,d)

% 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.m

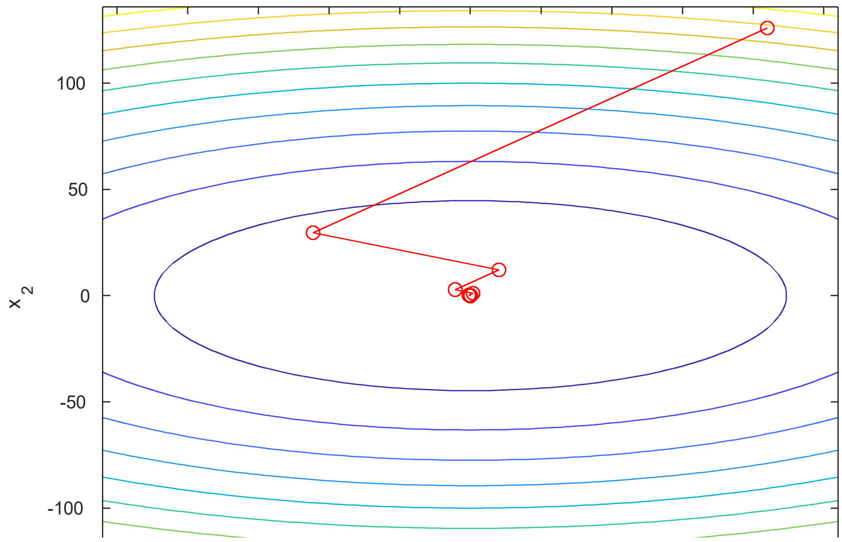
```
function g=gradie(x)

% GRADIE Wyznacza analitycznie gradient funkcji kosztu w
punkcie X.

g=[2*x(1)
   2*x(2)];
```

```
close all;
clear all;

n = 14;
x0 = [3*n; 9*n];

[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; %
granad
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');
```

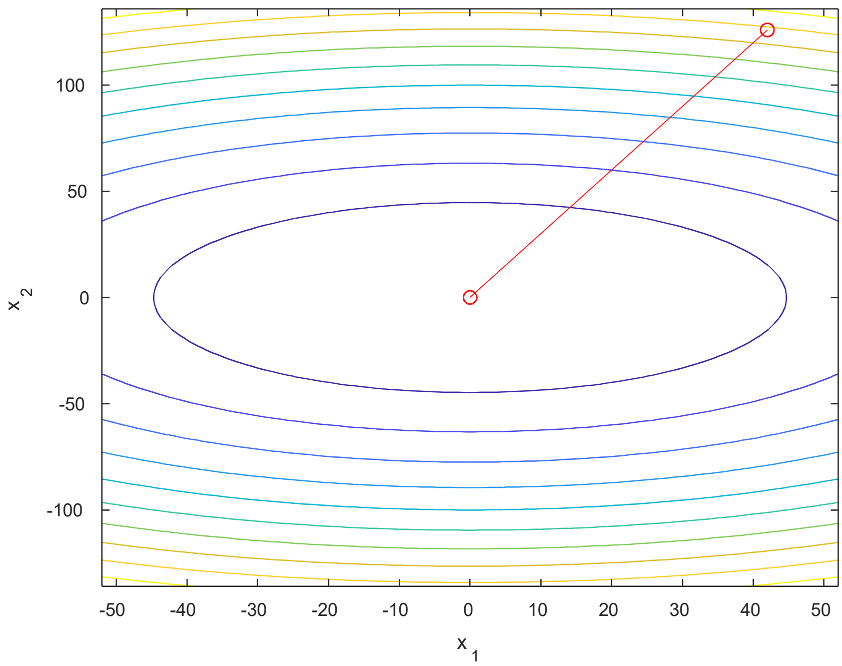
Q(x 1 , x 2) dla a = 0.3



Q(x 1 , x 2) dla a = 0.5



Q(x 1 , x 2) dla a = 1



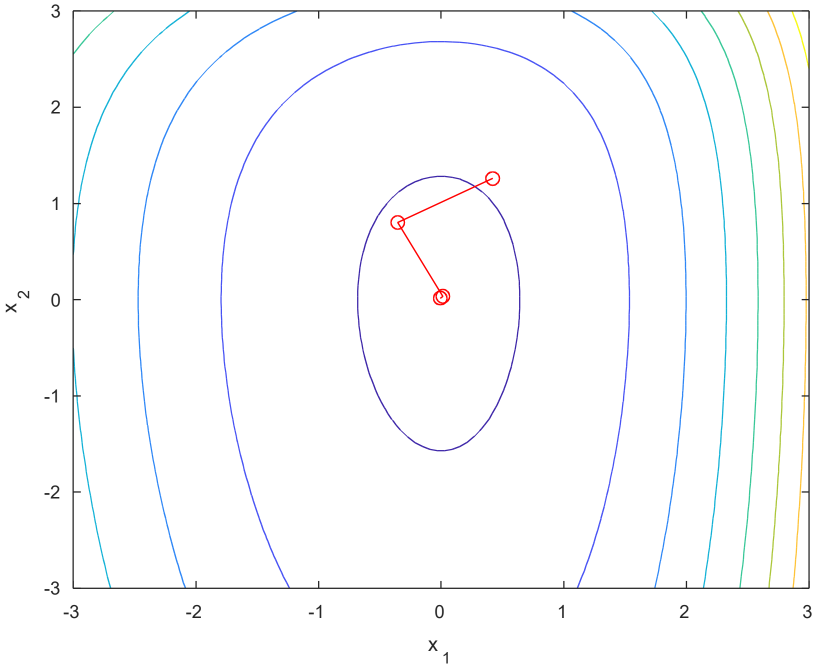
Zadanie 2.

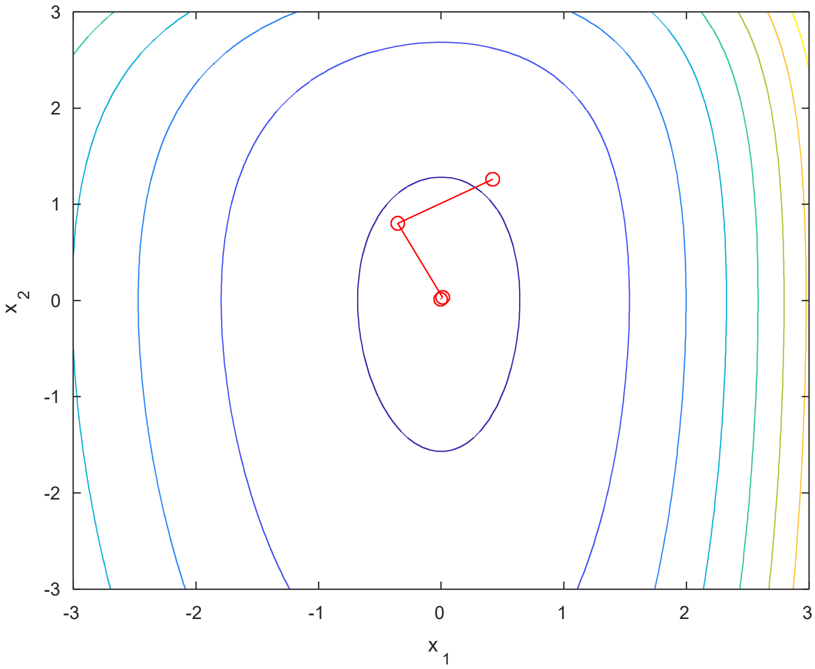
Badanie porównawcze gradientowych metod optymalizacji Badanie porównawcze gradientowych metod optymalizacji przeprowadzić dla funkcjonału:

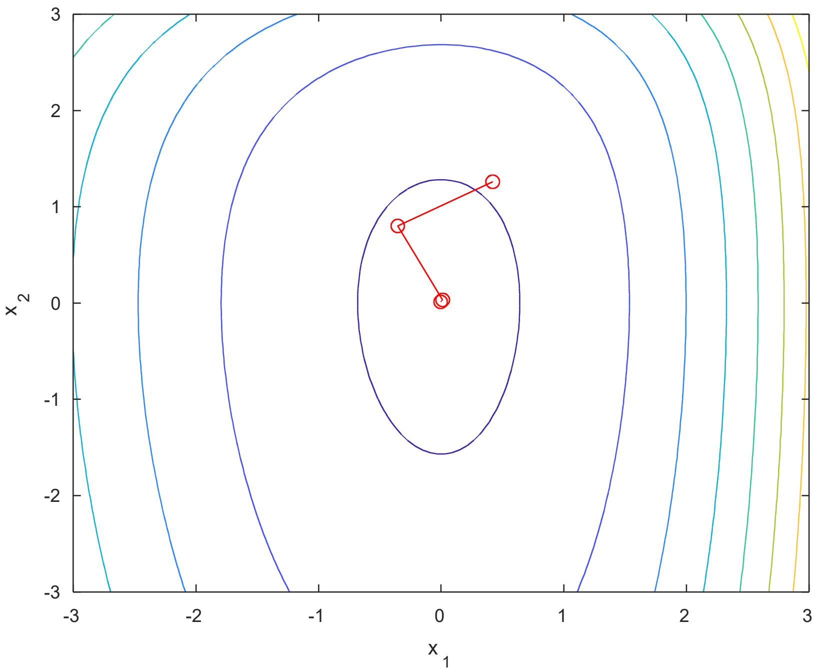
$$Q(x_1, x_2) = 6x_1^2 + 6x_1x_2 + x_2^2 + 4.5(e^{x_1} - x_1 - 1) + 1.5(e^{x_2} - x_2 - 1)$$

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

Rozwiązanie numeryczne.







Dla wysztkich metod zostały uzyskane wyniki po wykonaniu 4 iteracji.

```
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);
%f = @(x1, x2) 100*(x2-x1^2)^2 + (1-x1)^2;
q = f(x(1),x(2));
%q=x(1)^2+1*0.3*x(2)^2;
```

```
function g=gradie(x)

% GRADIE Wyznacza analitycznie gradient funkcji kosztu w
punkcie X.

%g=[2*x(1)
%   0.6*x(2)];
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)];
%g = [-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1)); 200*(x(2)-x(1)^2)];
```

Zadanie 5

„Dolina bananowa” Rossenbrocka. Wyznaczyć minimum funkcji :

$Q(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

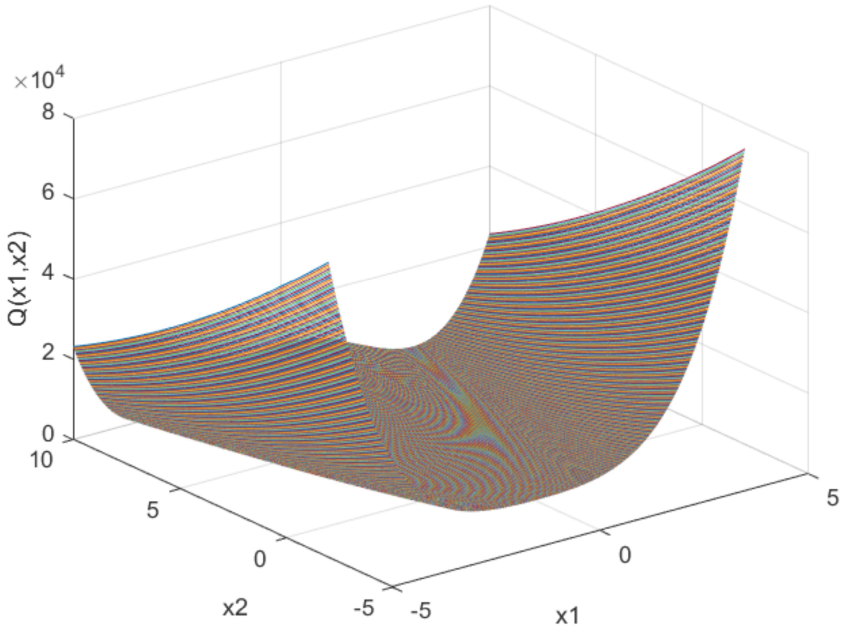
dla x0 = [9/2;9] *n

Metoda analityczna.

Obydwa składniki funkcjonału są liczbami nieujemnymi, co więcej mogą przyjąć 0, a więc minimum funkcjonału wynosi 0.

$100(x_2 - x_1^2)^2 + (1 - x_1)^2 = 0 \rightarrow 100(x_2 - x_1^2)^2 = (1 - x_1)^2$

$100(x_2 - x_1^2)^2 = 0 \wedge 0 = (1 - x_1)^2, więc\; x_1 = 1 \wedge x_2 = 1\; Q(1,1) = 0$



Metoda numeryczna

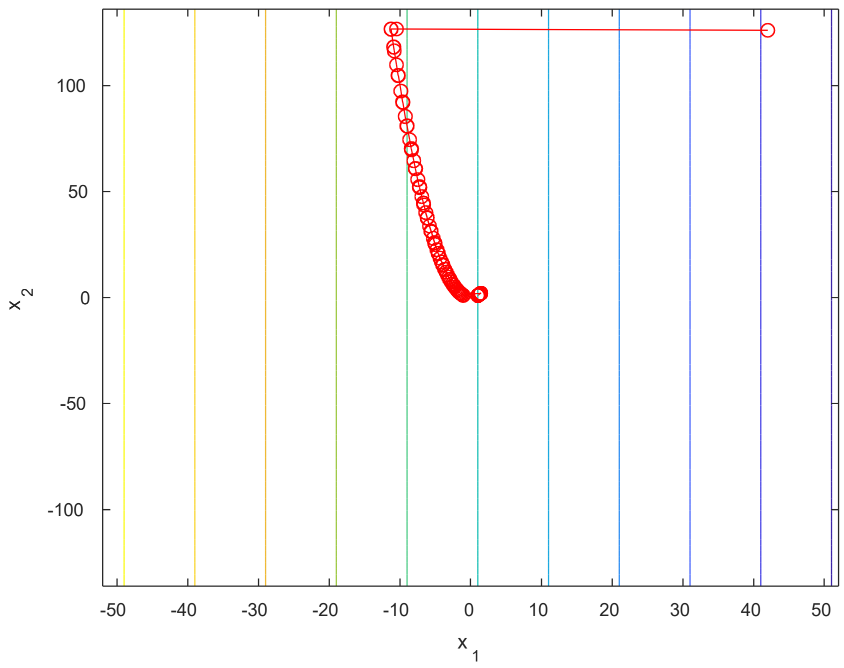
```
function [q,x]=koszt(x,z,d)

% 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);
f = @(x1, x2) 100*(x2-x1^2)^2 + (1-x1)^2;
q = f(x(1),x(2));
%q=x(1)^2+1*0.3*x(2)^2;
```

```
function g=gradie(x)

% GRADIE Wyznacza analitycznie gradient funkcji kosztu w
punkcie X.

%g=[2*x(1)
%    0.6*x(2)];
%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)];
g = [-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1)); 200*(x(2)-x(1)^2)];
```

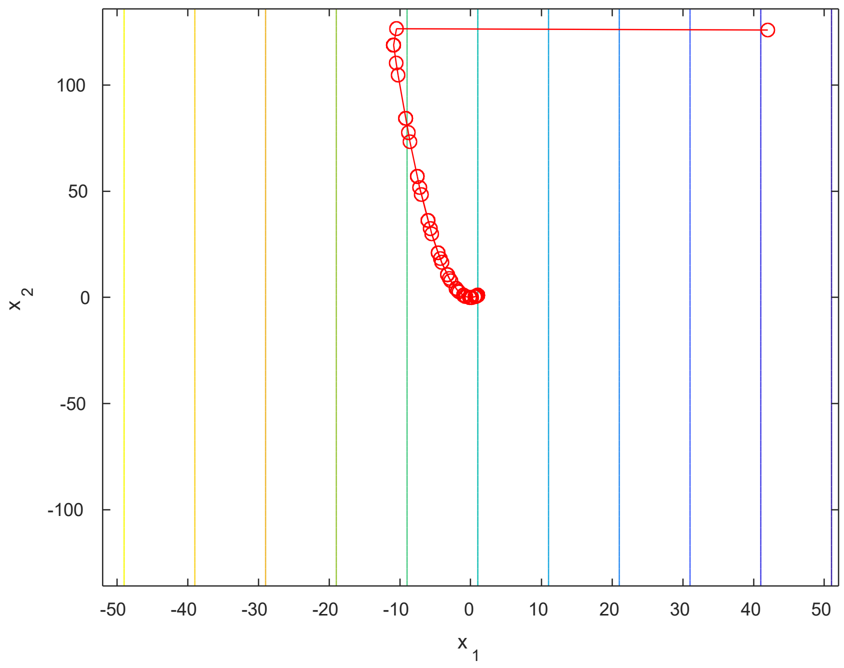
```
n = 14;
x0 = [3*n; 9*n];
[x1, x2] = meshgrid(linspace(-3*n-10, 3*n+10), linspace(-9*n-
10, 9*n+10));
q = @(x1, x2) 100*(x2-x1^2)^2 + (1-x1)^2;
figure()contour(x1, x2, q(x1,x2));
hold on
par = 3; % metoda najszybszego spadku
granad
plot(etap(1,:), etap(2,:), 'b');
plot(etap(1,:), etap(2,:), 'bo');
xlabel('x_1');ylabel('x_2');
```

Metoda najszybszego spadku



i=89 ; x = [1.00397543967224 1.00824269836930]

Metoda Fletchera – Reevesa



i=50 ; x = [1.0039754654654 1.000145683644560]