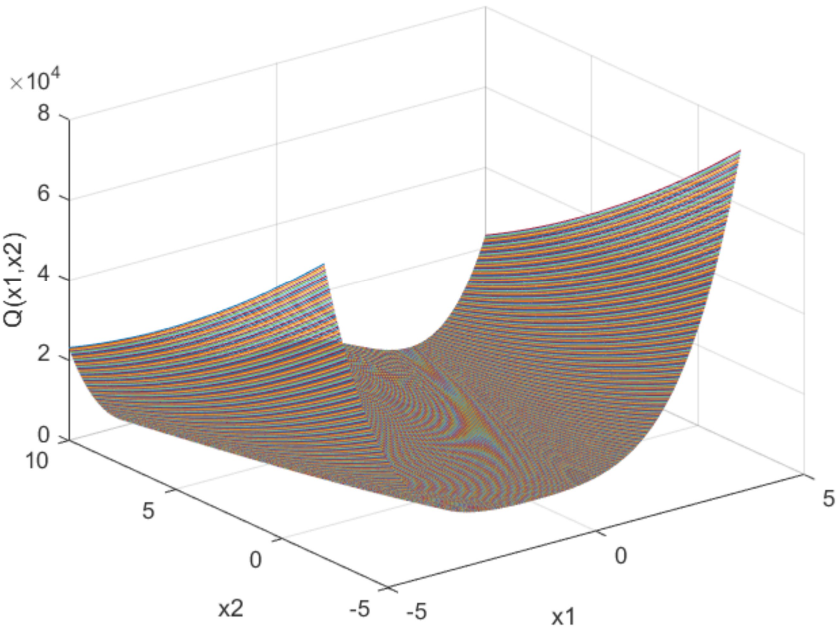
SPRAWOZDANIE Z TEORII OPTYMALIZACJI

Imię, Nazwisko, Numer	Michał Krzyszczuk N=14
Temat ćwiczenia	Metody Powela
Data i godzina wykonania ćwiczenia	13 marca 2019, godz: 14:30

Zadanie 1.

„Dolina bananowa" Rossenbrocka. Wyznaczyć minimum funkcji :

$Q(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ dla x0 = [7/2;7] *n



Metoda analityczna.

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

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

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

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=100*(x(2)-x(1)^2)^2+(1-x(1)^2)^2;
```

pownad.m

```
z=2;
xn= x0;
X=xn;
itp=1;
n=length(xn);
e0=0.001;
maxit=100;
dm=eye(2);
x_rozw = [ xn]
kier_baz = [dm]
metoda=1
   if metoda==1,powe_1
   else,powe_2
   end
```

powe_1.m

```
for iter=itp:maxit
   xn=prosta1(xn,dm(:,z));
   x0=xn;
   X=[X xn];
   for i=1:z
      [xn,qn]=prosta1(xn,dm(:,i));
   end
   delta=norm(xn-x0);
   if delta<e0, break, end
   dm(:,1)=[];
   dm(:,z)=(xn-x0)/delta;
   x_rozw = [x_rozw, xn];
   kier_baz = [kier_baz , dm(:, n)];
end
```

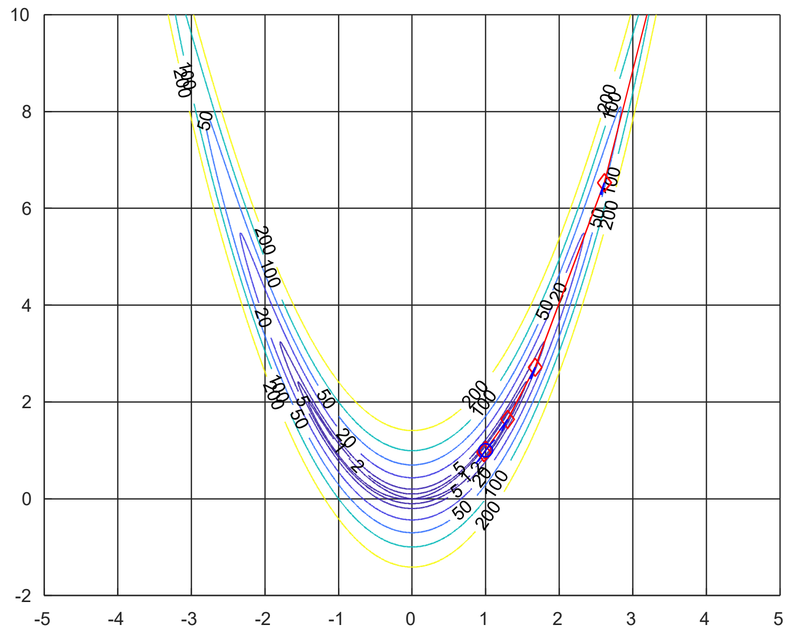
zad2.m

```
clear all;
close all;
clc
drawArrow = @(p1, p2, varargin) quiver(p1(1), p1(2), p2(1) -
p1(1), p2(2) - p1(2), 0, varargin{:});

[x1, x2] = meshgrid(-5:0.01:5, -2:0.01:10);
x0 = [7/2; 7] * 14;%N=14
q=100*(x2-x1.^2).^2+(1-x1.^2).^2;
figure
hold on;
contour(x1, x2, q, [1, 2, 5, 20, 50, 100, 200], 'ShowText',
'on');
x0 = [48; 49];
metoda = 1;
pownad
grid on;

%ograniczenie wykresu ze względu na odległość punktu
początkowego od
%końcowego
axis([-5 5  -2 10])
hold on;

%kolejne wyznaczone punkty
plot(x_iter(1, :), x_iter(2, :), 'r');
plot(x_iter(1, :), x_iter(2, :), 'rd');
```

```
%punkt początkowy (49,98)
plot(x_iter(1, 1), x_iter(2, 1), 'b*');
%znaleziony punkt końcowy
plot(x_iter(1, size(x_iter, 2)), x_iter(2, size(x_iter, 2)),
'bo');

%rysuj wektory bazowe
for i = 1:size(x_iter, 2)
    drawArrow(x_iter(:, i), x_iter(:, i) + kier_baz(:, i)/4,
'color', 'blue ');
    drawArrow(x_iter(:, i), x_iter(:, i) + kier_baz(:, i +
1)/4, 'color', 'blue ');
end;
```

plot_funkcji.m

```
clear all;
close all;
[x1, x2] = meshgrid(-5:0.01:5, -2:0.01:10);
q=100*(x2-x1.^2).^2+(1-x1.^2).^2;


figure(1)
plot3(x1,x2,q)
grid on;

xlabel('x1')
ylabel('x2')
zlabel('Q(x1,x2)')
```



Otrzymane optymalne wartości:
$x_1 = 0.99994 \wedge x_2 = 0.99999 \rightarrow Q(x_1, x_2 = 0.0001)$

Zadanie 2.

Zbadać działanie metod Powella dla funkcji :

$$Q(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_3^3 - x_4)^2 + (1 - x_3)^2 + 10.1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19.8(x_2 - 1)(x_4 - 1)$$

gdzie $[x_{01}, x_{02}, x_{03}, x_{04}]^T = [7/2, 7, 7, 7/2]^T * N, N = 14$

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
x1 = x(1); x2 = x(2); x3 = x(3); x4 = x(4);
q = 100*(x1.^2-x2).^ 2 + (1-x1).^2+90*(x3.^3-x4).^2+ ( 1 - x3
) .^ 2 + 10.1 * ( ( x2 - 1 ) .^ 2 + ( x4 - 1 ) .^ 2 ) +  19.8
* ( x2 - 1 ) * ( x4 - 1 );
```

zad3.m

```
close all;
clear all;
metoda = 1;
x0 = [7/2;7;7;7/2];
pownad;
```

pownad.m

```
im=inf;%aby zmazac wartość przy poprzednich wywołaniach
programu

z=4;
xn= x0;
X=xn;
itp=1;
n=length(xn);
e0=0.000001;
maxit=100000;
dm=eye(n);
x_iter = [ xn]
kier_baz = [dm]
metoda=1
   if metoda==1,powe_1
   else,powe_2
   end
```

powe_1.m

```
y = []
for iter=1:maxit
   xn=prosta1(xn,dm(:,z));
   x0=xn;
   X=[X xn];
   for i=1:z
      [xn,qn]=prosta1(xn,dm(:,i));
   end
   delta=norm(xn-x0);
   if delta<e0, break, end
   dm(:,1)=[];
   y = [y qn];
   dm(:,z)=(xn-x0)/delta;
   x_iter = [x_iter, xn];
   kier_baz = [kier_baz , dm(:, n)];
end
```

Wyniki:

x1	x2	x3	x4	Q
3.50000	7,00	7,00	3.50000	1121354.4654654654
2.64589	-21.1455	6.75048	278.867	816567.033252650
1.37517	-36.2530	6.17471	212.133	502859.738783640
-0.2069	-8.34696	4.57791	78.3018	81984.0800122689
-0.7929	-1.90213	3.06543	22.2468	7942.97385362438
-1.2210	2.45353	2.16554	9.32675	1122.03771218435
-1.3549	2.28620	1.90596	5.40771	558.696825051697
-1.4976	1.62151	1.53077	3.12810	139.941923814570
-1.5928	2.29300	1.26556	1.50354	69.7461469399132
-1.7652	3.03686	0.70054	0.19747	26.3338286184074
-1.7728	3.02474	0.67541	0.16222	26.0123932577086
-1.7861	**3.07006**	**0.62836**	**0.12662**	**25.8582615870847**

Wnioski

I Metoda Powella jest stosowana dla funkcji, których poziomice mają kształt wąskich dolin. Jest metodą szybszą od używanych w ramach poprzednich ćwiczeń. Istnieje jednak niebezpieczeństwo

degeneracji bazy, czyli wolne zbieganie do wartości optymalnej w okolicach minimum. Rozwiązaniem tego problemu jest zastosowanie zmodyfikowanej metody Powella, która będzie omówiona na następnych laboratoriach.