Łukasz Marczak

Sprawozdanie 4

Metody gradientowe zmiennej metryki

Zadanie 3

Zaobserwować działania różnych metod zmiennej metryki dla funkcjonału:

$$Q(x_1,x_2) = 7x_1^2\ ax_2^2$$

wybierając kolejno a=1, a=0.5, a= 0.3

Zmodyfikowany plik 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+d;
elseif nargin==3
    %x=x+z*d;
    x(1) = x(1) + z*d(1);
    x(2) = x(2) + z*d(2);
end
n=7; zadanie = 3;
if zadanie == 3
a=1;
%a=0.5;
%a= 0.3;
q = 7*x(1)^2 + a*x(2)^2;
elseif zadanie ==4
q = 6*x(1)^2 +6*x(1)*x(2) + 7*x(2)^2 + 4.5*(exp(x(1))-x(1)-1) + 1.5*(exp(x(2))-x(2)-1);
```

Zmodyfikowany plik gradie.m:

```
function [g]=gradie(x)
% GRADIE Wyznacza analitycznie gradient funkcji kosztu w punkcie X.
n=7; zadanie = 3;
if zadanie == 3
    a = 1;% a=0.5;a=0.3;
    g = [2*n*x(1); 2*a*x(2)];
elseif zadanie ==4
    qx1 = 12*x(1) +6*x(2) +4.5*exp(x(1))-4.5;
    qx2 = 6*x(1) + 2*n*x(2) + 1.5*exp(x(2)) - 1.5;
    g = [qx1; qx2];
end
```

Porównanie metod pod względem liczby iteracji oraz wartości parametru **a** prezentuje tabela 1:

	Metoda Fletchera – Powella	Metoda Wolfe'a – Broydena – Davidona	Metoda Broydena – Fletchera – Goldfarba - Shanno	Metoda Pearsona 1	Metoda Pearsona 2	Metoda McCormicka
a=1	2	2	2	2	4	8
a=0.5	2	2	2	2	3	4
a=0.3	2	2	2	2	4	4

Tabela 1

Jak widać, najmniej efektywna okazała się metoda McCormicka oraz druga metoda Pearsona. Mimo to udało się uzyskać rozwiązanie optymalne już po kilku iteracjach. Generalnie im mniejsza wartość parametru a, tym szybsza zbieżność metody.

Zadanie 4

Zmodyfikowany plik zmenad.m:

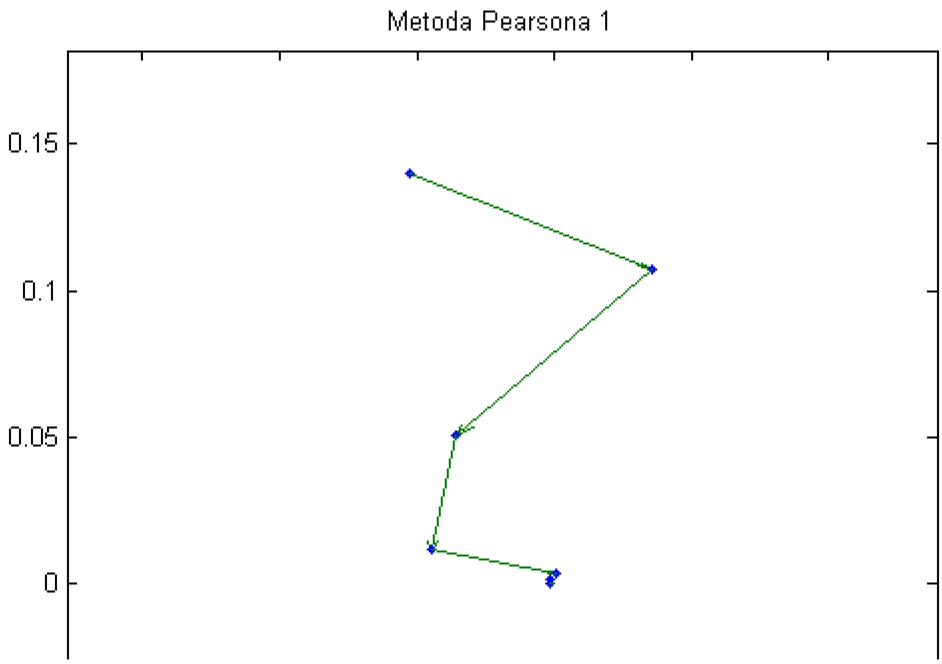
```
% ZMENAD Zawiera następujące metody zmiennej metryki:
%        - Davidona - Fletchera - Powella (par=0)
%        - Wolfe'a - Broydena - Davidona  (par=1)
%        - Broydena - Fletchera - Goldfarba - Shanno (par=2)
%        - Pearsona 1 (par=3)
%        - Pearsona 2 (par=4)
%        - McCormicka (par=5).
% Oznaczenia: maxit - maksymalna liczba iteracji górnego poziomu
%             x0 - aktualne przybliżenie rozwiązania optymalnego
%             e0 - metoda się zatrzymuje, gdy kwadrat normy gradientu
%                  n2 spadnie poniżej e0
%             g - gradient
%             d - kierunek poszukiwania
%             v - aproksymacja odwrotności hesjanu
%czod - częstość odnowy.
close all;
for jjj = 0:5

par=jjj;
maxit=10;
x0=[-3;-3];
x_prev = x0;
e0=1e-8;
vect = [];
n=length(x0);
czod=2*length(x0);
wskaz=0;
for i=1:maxit

   g=gradie(x0);
   n2=g'*g;
   if n2<e0,break,end
   if rem(i,czod)==1|~wskaz
      v=eye(n);
   else
      r=g-gs;
      vr=v*r;
      if par==0                          % DFP
         s=x0-xs;
         v=v+s*s'/(r'*s)-vr*vr'/(r'*vr);
      elseif par==1                      % WBD
         svr=x0-xs-vr;
         v=v+svr*svr'/(svr'*r);
      elseif par==2                      % BFGS
         s=x0-xs;
         sr=s'*r;
         vrs=vr*s';
         v=v+(1+r'*vr/sr)*s*s'/sr-(vrs+vrs')/sr;
      elseif par==3                      % P1
         v=v-vr*vr'/(vr'*r);
      elseif par==4                      % P2
         s=x0-xs;
         svr=s-vr;
```

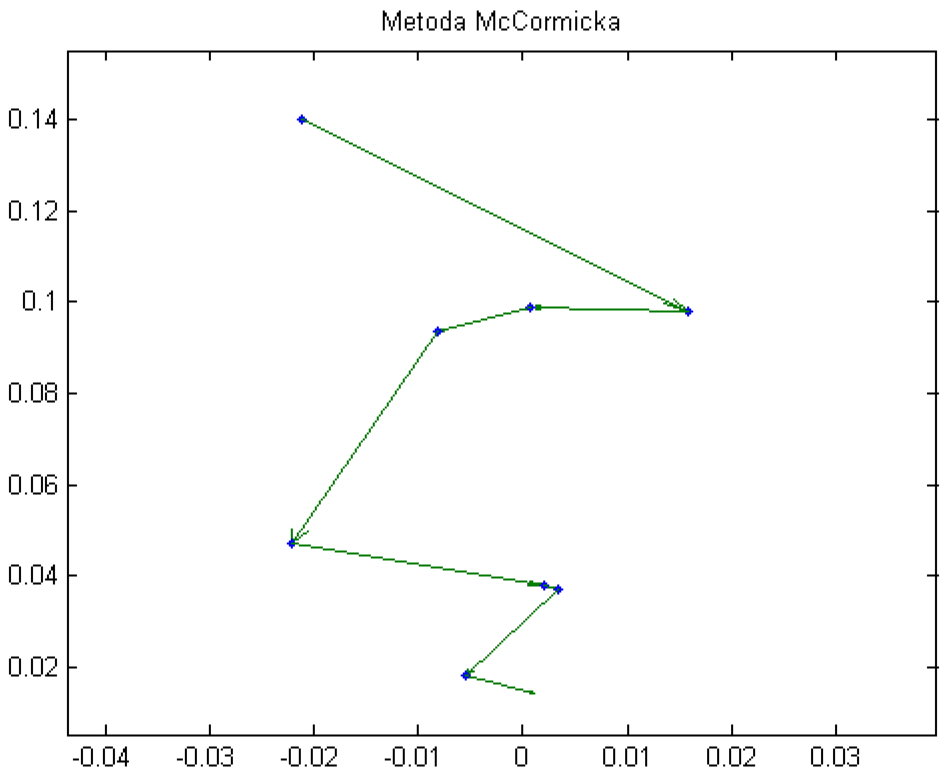
```
         v=v+svr*vr'/(vr'*r);
      elseif par==5                          % McC
         s=x0-xs;
         sr=s'*r;
         svr=s-vr;
         v=v+svr*s'/sr;
      end
   end
   d=-v*g;
   if d'*g<0
      gs=g;
      xs=x0;
      [x0,wskaz]=kierun(x0,d);
   else
      wskaz=0;
   end
   vect = [vect x0];
end

for iteracja=1:length(vect)
    integ = iteracja;
    fprintf('\nmetoda: %d, iteracja: %d, wartosc normy: %.12f ', par, integ,
abs(vect(iteracja)-x0));
end

fprintf('\n\nWektor vect: \n[ ');
for kuj=1:length(vect)
    fprintf('%.3f, ',vect(kuj));
end
fprintf(' ]');


points= -10:0.01:10;
[X Y] = meshgrid(points,points);
a=0.3;
Z =7*X.^2 +a*Y.^2;

figure
contour(X,Y,Z,30);
if par==0
    title('Metoda Davidona - Fletchera - Powella');
elseif par == 1
    title('MetodaWolfe''a - Broydena - Davidona');
elseif par == 2
    title('Metoda Broydena - Fletchera - Goldfarba - Shanno');
elseif par == 3
    title('Metoda Pearsona 1');
elseif par == 4
    title('Metoda Pearsona 2');
elseif par == 5
    title('Metoda McCormicka');
end
    hold on;
for inex=1:length(vect)-1
    plot(vect(1,inex),vect(2,inex),'.');hold on;
    p1 = vect(:,inex);
    p2 = vect(:,inex+1);
    dp = p2-p1;
    quiver(p1(1), p1(2), dp(1),dp(2), 0);
end
fprintf('\n\n');
end
```

W przypadku metod ze zmienną metryką, porównano ilości iteracji, po których metody uzyskały optymalne rozwiązanie:

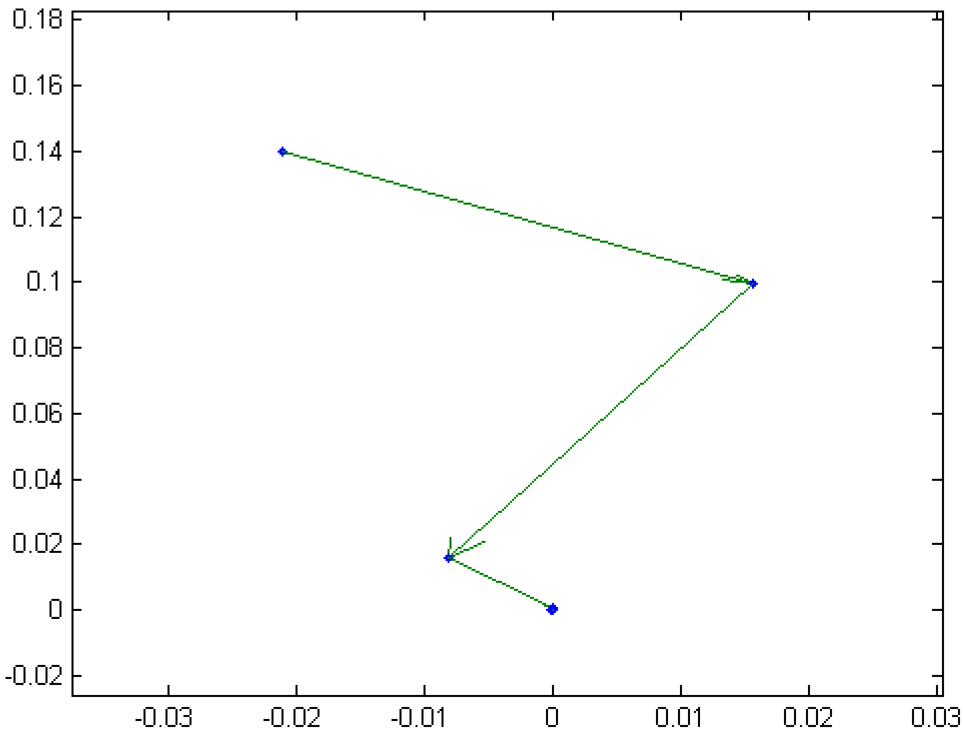
	Metoda Fletchera – Powella	Metoda Wolfe'a – Broydena – Davidona	Metoda Broydena – Fletchera – Goldfarba - Shanno	Metoda Pearsona 1	Metoda Pearsona 2	Metoda McCormicka
Ilość iteracji	4	4	4	6	4	5

Podobnie jak w poprzednim zadaniu, metody dały wyniki już po kilku iteracjach. Najgorzej spsiały się pierwsza metoda Pearsona oraz metoda McCormicka. Podobnie ja w przypadku gradientowych metod optymalizacji, w metodach ze zmienną metryką norma $| X^{apr} - X^* |$ zbiega do zera (choć nie jest to spadek monotoniczny).
Wszystkie powyższe metody uzyskały podobne wyniki. Poniżej zamieszczam kolejne kroki wybranych metod:

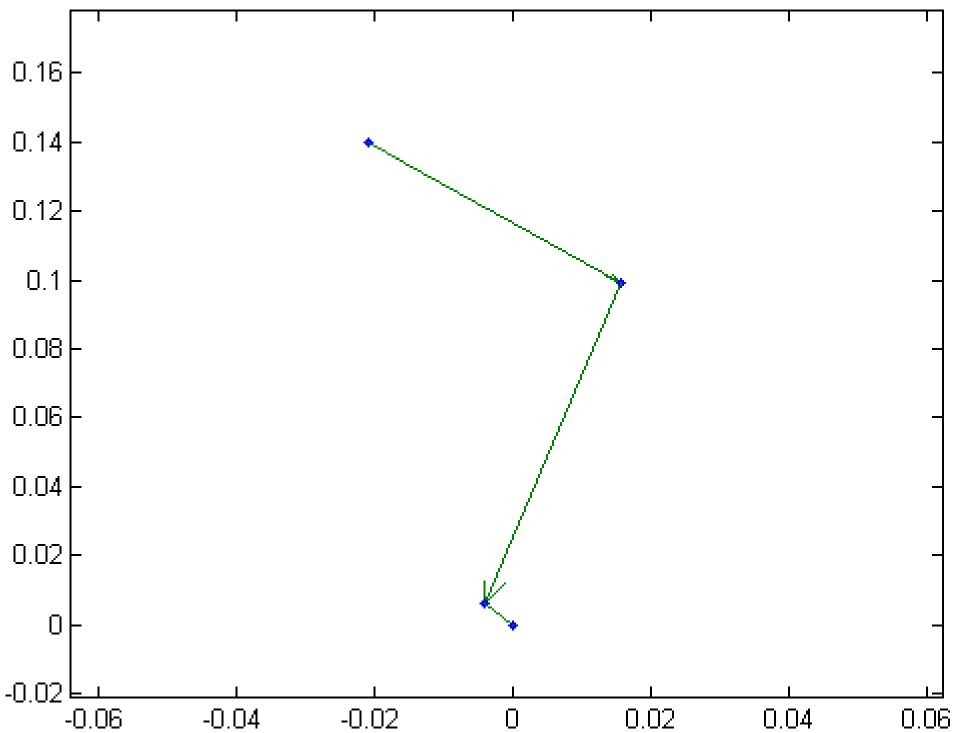




Metoda Broydena - Fletchera - Goldfarba - Shanno



Metoda Wolfe'a - Broydena - Davidona



<u>Wnioski</u>

Celem ćwiczenia było zapoznanie się z gradientowymi metodami optymalizacji zmiennej metryki. Metody zmiennej metryki są znacznie szybsze od „czystych” metod gradientowych, a przez to że osiągają porównywalne wyniki, są przez to efektywniejsze.