Sprawozdanie z laboratorium Teorii Optymalizacji

Imię i nazwisko	Jacek Gołda
Temat ćwiczenia	Pierwsza metoda Powella
Data i godzina wykonania ćwiczenia	9 marca 2016, 14:00

1 Wstęp

Celem laboratorium było zbadanie własności pierwszej metody Powella.

2 Ćwiczenie 2

Treść zadania:

„Dolina bananowa" Rosenbrocka. Wyznaczyć minimum funkcji:

$$Q(x_1, x_2) = 100\left[(x_2 - 3n) - (x_1 - 3n)^2\right]^2 + \left[1 - (x_1 - 3n)\right]^2$$

Na mapę poziomic doliny nanieść punkty pośrednie poszczególnych kroków oraz położenie baz.

Za parametr n przyjęto wartość 4.

$$Q(x_1, x_2) = 100\left[(x_2 - 12) - (x_1 - 12)^2\right]^2 + \left[1 - (x_1 - 12)\right]^2$$

Rozwiązanie:

2.1 Rozwiązanie analityczne

Zauważam, że funkcja jest sumą dwóch wyrażeń, z których każde jest nieujemne, ponieważ jest kwadratem pewnej liczby. Oznacza to, że minimalną wartością funkcji jest wartość 0.

Wyznaczam wartości x_1 i x_2 dla których obydwa wyrażenia się zerują.

$$\begin{cases} (x_2 - 12) - (x_1 - 12)^2 = 0 \\ 1 - (x_1 - 12) = 0 \end{cases}$$

$$\begin{cases} x_2 = 13 \\ x_1 = 13 \end{cases}$$

Minimum będzie osiągane dla punktu $\widehat{x} = \begin{bmatrix} 13 \\ 13 \end{bmatrix}$

2.2 Rozwiązanie numeryczne

Jako punkt początkowy przyjęto punkt $x_0 = \begin{bmatrix} 11 \\ 11.5 \end{bmatrix}$

W rozwiązaniu wykorzystano m-pliki realizujące pierwszą metodę Powella. Dokonano modyfikacji pliku definiującego koszt:

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

% KOSZT wylicza wskaznik jakosci dla wektora zmiennych
% decyzyjnych x+z*d.

if nargin==2, x=x+z;
elseif nargin==3, x=x+z*d;
end

n = 4;
x = x - 3 * n;
x1 = x(1);
x2 = x(2);

q = 100 * ( x2 - x1 .^ 2 ) .^ 2 + ( 1 - x1 ) .^ 2;
```

Zmodyfikowano również plik pownad. W wersji pobranej ze strony laboratorium znajdował się kod, który powodował powtórne szukanie minimum, z warunku początkowego przesuniętego o wektor jedynek w stosunku do ostatnio znalezionego punktu. Usunięto tę część kodu wraz z pętlą, gdyż eksperymentalnie sprawdzono, że minimum jest znajdowane od razu przy pierwszym wywołaniu skryptu powe_1, a powtórne wywołanie tylko zaciemnia uzyskany poniżej wykres.

```
maxit=100;
itp=1;
xn = x0;
n=length(xn);
dm=eye(n);
xa=xn;
e0=1e-7;
x_rozw = [x0];
kier_baz = dm;

if metoda==1
    powe_1
else
    powe_2
end
```

W pliku powe_1 dodano zapis wartości obecnego rozwiązania i nowego kierunku poszukiwań za pomocą instrukcji:

```
x_rozw = [x_rozw, xn];
kier_baz = [kier_baz, dm(:, n)];
```

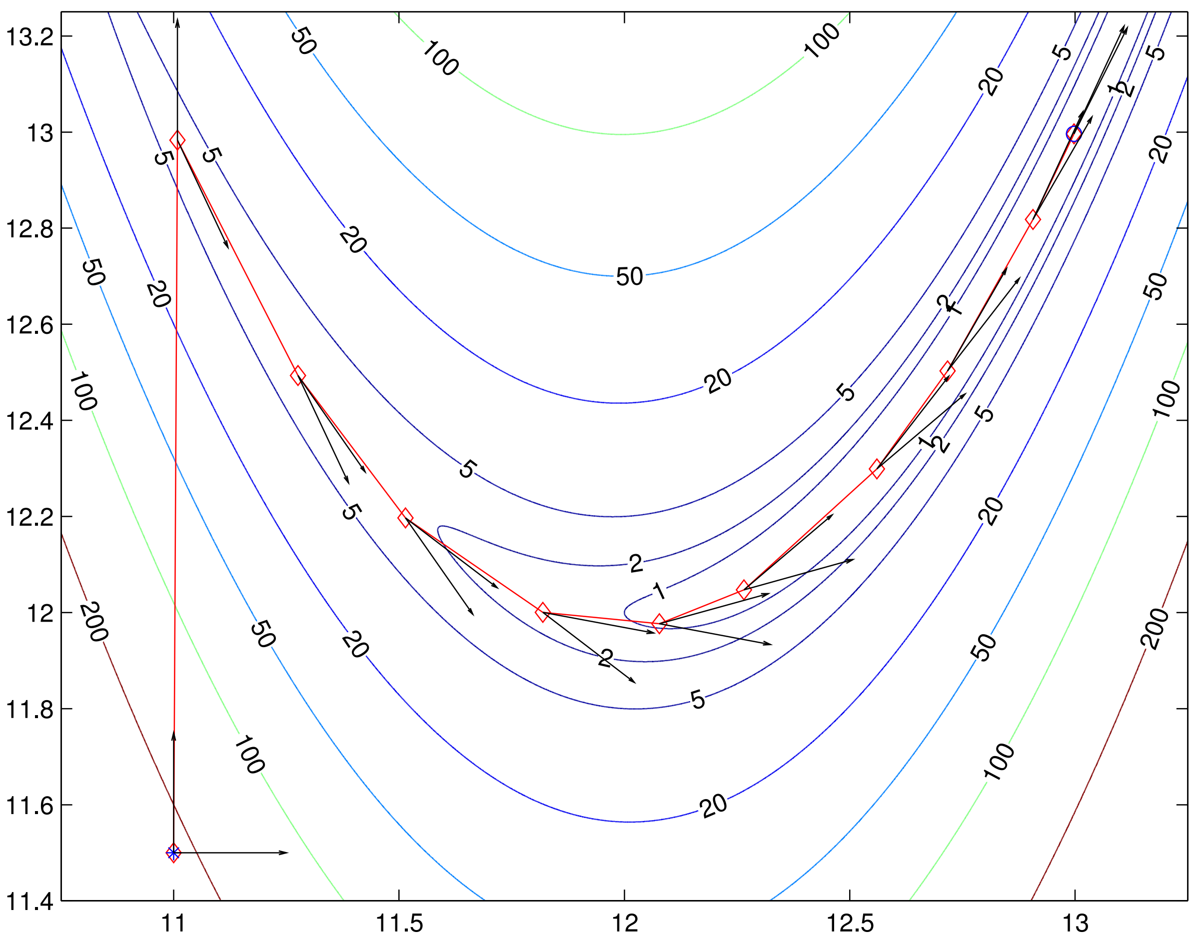
Nie załączono pełnego listingu, gdyż nie wnosiłby on dużo do sprawozdania.

Napisano poniższy m-plik w celu przedstawienia przykładowych poziomic funkcji celu i wyników pracy algorytmu: kolejno znalezionych punktów wraz z wektorami baz w których prowadzone były poszukiwania.

```
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(10.75:0.001:13.25, 11.4:0.001:13.25);
n = 4;
q = 100 * ( ( x2 - 3 * n ) - ( x1 - 3 * n ) .^ 2 ) .^ 2 + ( 1 - ( x1 - 3 * n ) ) .^ 2;
figure
contour(x1, x2, q, [1, 2, 5, 20, 50, 100, 200], 'ShowText', 'on');
```

```
% pierwsza metoda
x0 = [11; 11.5];
metoda = 1;
pownad;

hold on;
plot(x_rozw(1, :), x_rozw(2, :), 'r');
plot(x_rozw(1, :), x_rozw(2, :), 'rd');

% rysuj pierwszy i ostatni punkt
plot(x_rozw(1, 1), x_rozw(2, 1), 'b*');
plot(x_rozw(1, size(x_rozw, 2)), x_rozw(2, size(x_rozw, 2)), 'bo');

for i = 1:size(x_rozw, 2)
    drawArrow(x_rozw(:, i), x_rozw(:, i) + kier_baz(:, i)/4, 'color', 'black');
    drawArrow(x_rozw(:, i), x_rozw(:, i) + kier_baz(:, i + 1)/4, 'color', 'black');
end;

print -depsc2 'wykres.eps'
```

Uzyskano następujący wykres:



Czerwonymi rombami są zaznaczone kolejne wyznaczone punkty, są one połączone czerwonymi liniami. Na czarno zaznaczone są wektory bazy, dla której było poszukiwane minimum.

Tabela prezentuje kolejne przybliżenia rozwiązania optymalnego uzyskane w kolejnych iteracjach algorytmu.

Numer iteracji	Współrzędne x_1, x_2		Wartość funkcji
1	11	11,5000000000000	229
2	11,0083492816806	12,9833711471434	3,96667258378229
3	11,2756387349858	12,4934416487712	3,07112548730848
4	11,5143140110961	12,1970213939505	2,35834655078255
5	11,8190130839544	12,0002956410268	1,50009929882121
6	12,0777064235548	11,9765479363083	0,93759352698522
7	12,2653362504702	12,0473117105480	0,59305309443070
8	12,5603174351663	12,2990611591440	0,21550527851975
9	12,7171323701030	12,5028255026689	0,09313198105451
10	12,9064978793585	12,8182126772852	0,00998572235486
11	12,9978706073696	12,9965156505864	0,00006380915022

Kolejne rozwiązania zbiegają do wyznaczonego teoretycznie rozwiązania $\widehat{x}$, a kolejne wartości funkcji zbiegają do zera.

3 Ćwiczenie 3

Treść zadania:

Zbadać działanie metod Powella dla funkcji:

$$Q(x) = 100\big[(x_1-3n)^2-(x_2-3n)\big]^2+\big[1-(x_1-3n)\big]^2+90\big[(x_3-3n)^3-x_4\big]+\big[1-(x_3-3n)\big]^3+$$

$$+10.1\Big[\big[(x_2-3n)-1\big]^2+big[(x_4-3n)-1\big]^2\Big]+19.8\big[(x_2-3n)-1\big]\big[(x_4-3n)-1\big]$$

3.1 Oszacowanie analityczne

Ze względu na fakt, że analityczne wyznaczenie minimów tej funkcji byłoby bardzo złożone, analityczne oszacowanie minimum ograniczono do zauważenia, że po wstawieniu $z_i = x_i + 3n$ uzyskuje się funkcję zbliżoną do tej, którą badano na poprzednim laboratorium. Wyznaczono wtedy minimum na kierunku $d = [1,1,1,1]^T$ — było ono osiągane dla $\widehat{x} = [1,1,1,1]^T$, a wartość funkcji w minimum wynosiła 0. W analogiczny sposób sprawdzono, czy dla tej funkcji również istnieje to minimum.

Dokonano podstawienia opisanego powyżej, a następnie wstawiono parametryzację prostej o kierunku d przechodzącej przez punkt $(0,0)$:

$$z = z_0 + td = t$$

Po wstawieniu i uporządkowaniu wyrazów uzyskano:

$$q(t) = Q(t,t,t,t) = (t-1)^2(90t^4+180t^3+190t^2+42)$$

Sprawdzono, czy wyrażenie w drugim nawiasie nie ma pierwiastków. Po zróżniczkowaniu tego wyrażenia uzyskuje się:

$$360t^3+540t^2+380t = t\cdot(360t^2+540t^1+380)$$

Wyróżnik trójmianu kwadratowego w nawiasie jest mniejszy od zera, więc jedynym punktem, w którym zeruje się pochodna jest punkt $t=0$. Będzie to minimum globalne tego wyrażenia. Wartość wyrażenia dla $t=0$ wynosi 42, jest to więcej niż

0. Wynika stąd, że punkt $t = 1$ jest minimum globalnym na rozważanym kierunku. Wartość funkcji w tym punkcie jest równa 0. Po cofnięciu się do zmiennych x_i uzyska się następujący punkt jako oszacowanie minimum:

$$x = \begin{bmatrix} 13 \\ 13 \\ 13 \\ 13 \end{bmatrix}$$

Ta wartość będzie oszacowaniem minimum funkcji dla rozwiązania numerycznego.

3.2 Rozwiązanie numeryczne

W celu znalezienia rozwiązania numerycznego ponownie zmodyfikowano funkcję obliczającą wartość funkcji celu.

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

% KOSZT wylicza wskaznik jakosci dla wektora zmiennych
% decyzyjnych x+z*d.

if nargin==2, x=x+z;
elseif nargin==3, x=x+z*d;
end

n = 4;
x = x - 3 * n;
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 );
```

Wykorzystano zmodyfikowane w poprzednim ćwiczeniu pliki pownad i powe_1

Wyznaczano minimum funkcji dla punktu startowego $x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Uzyskano następujące wyniki:

Numer iteracji	Współrzędne x_1, x_2, x_3, x_4				Wartość funkcji
1	0	0	0	0	267459738
2	11,9971162516832	151,758149502015	0,421696047728145	-1397,16559042962	20198192,8178327
3	36,4762887690099	579,020262702104	1,62014788159304	-1041,48258423002	3130649,28947490
4	41,4012682651431	874,577054085991	2,47111674992001	-853,081105479597	299947,374310426
5	41,3683386086852	874,677005068520	2,47209016612369	-850,809602713037	299108,426152982
6	41,3682223466364	874,670462772799	2,47211822815103	-850,788301784999	299103,481673144
7	39,4628952262741	762,879305386457	3,03240970326098	-703,637625584493	230764,195358163
8	36,7217135256719	615,917675061527	3,56907566837157	-576,924658458411	159516,931356873
9	33,1944814357768	448,231726445676	4,40575599723750	-423,947338807499	93785,6518707890
10	32,8156800798772	432,239517454662	4,50097664117885	-410,004256785505	88589,9986808328
11	30,2784984567285	337,870223113102	5,16287678240868	-303,391001142585	50581,1472976224
12	27,2775945271226	241,030774602880	6,05035770197600	-195,969562168258	25521,6606189173
13	25,2339718428315	181,758119169330	6,69297218659321	-136,987642427368	16787,9079059575
14	24,3309348602804	161,283171324532	7,00020276776944	-111,880752155778	13979,2903967697
15	22,1324024896152	110,551957122609	7,77079921289386	-64,2913316528182	9002,58659124203
16	20,6791156503973	83,6128346868430	8,31089873699871	-38,5041549148167	6611,13005775743
17	19,1442099620320	61,1086904828676	8,91322048764375	-16,9859889441049	4337,84132992468
18	17,5516442705116	40,7544192582619	9,56193452082609	-2,19659509847426	2228,84772860074
19	16,0297363556495	26,5189258015428	10,2097442025562	6,17822717708862	803,280628688845
20	15,0640996453851	20,0055204779791	10,6354241807837	8,27633636781139	392,895110437795
21	15,0369461269039	19,9743861506800	10,6481330200560	8,20325989613152	385,150916691244
22	14,1593841346066	16,4780364769228	11,0396749402924	11,1466899330501	37,9396972659049
23	13,4047164257992	14,0435523851946	11,3822972032309	11,7991321328045	4,13565408180657
24	13,4187336423480	14,0644230114266	11,3759045156895	11,7684703197111	3,89786207293810
25	13,4708067418839	14,1702468794475	11,3520448182415	11,7404403035416	3,62646452112417
26	13,4714521860462	14,1722801140344	11,3517519305415	11,7397538899957	3,62645791065936
27	13,4714539709471	14,1722884263528	11,3517511035296	11,7397630434865	3,62645787091147
28	13,4714540309972	14,1722887679603	11,3517510751633	11,7397636724007	3,62645786529552

Dwie rzeczy zwracają uwagę:

1. Algorytm rozpoczął od bardzo odległego skoku z punktu początkowego, a następnie inną („okrężną") drogą zaczął zbiegać najpewniej do punktu $\widehat{x}$.

2. Algorytm nie odnalazł minimum, ponieważ wcześniej zadziałało kryterium stopu badające zmianę rozwiązania w kolejnych iteracjach.

Powyższe eksperymenty wykonywano z domyślną dokładnością 10^{-7}. Ponowiono eksperyment żądając większej dokładności 10^{-10}. Obliczenia zakończono w 29 iteracji, czyli o jedną iterację później. Uzyskano następujący wynik końcowy:

$$x_s = \begin{bmatrix} 13,4714538670667 \\ 14,1722882038308 \\ 11,3517511493721 \\ 11,7397638642203 \end{bmatrix} \qquad Q_s = 3,62645785234968$$

Takie zachowanie się algorytmu może być wynikiem degeneracji bazy objawiającej się zerowaniem się wyznacznika z macierzy kierunków. Sprawdzono, czy w tym wypadku zaszło takie zjawisko. Obliczono wyznacznik kolejnych macierzy kierunków wykorzystywanych do wyznaczania kolejnych rozwiązań. Przedstawia je poniższa tabela:

Numer iteracji	Wartość wyznacznika macierzy
1	1
2	-0,0574224682668565
3	0,0374233294919469
4	-4,12363549973580e-05
5	4,47499993678181e-05
6	-0,000263575448586089
7	-0,000107694696554208
8	4,81807632599087e-06
9	1,33485423982365e-06
10	4,25302530876073e-07
11	3,78096537878616e-07
12	2,63232784423189e-07
13	2,06720301501312e-07
14	1,95611636841665e-07
15	1,58265151408770e-07
16	1,46658514739426e-07
17	1,40475685978987e-07
18	1,52491439634781e-07
19	2,09434753827643e-07
20	-2,21975846243208e-07
21	1,45167023316031e-06
22	9,60400598696611e-07
23	-1,47491883519371e-06
24	-9,23401181932753e-07
25	2,28661104889235e-07
26	2,58699049803582e-07
27	2,58698760597505e-07
28	1,09230200516028e-10
29	1,22960375324095e-09

Wyznaczniki macierzy dla kolejnych iteracji zbliżają się do zera. Oznacza to, że dalsze poszukiwanie nie odniesie oczekiwanego efektu. Aby uniknąć tego zjawiska, należałoby zastosować zmodyfikowaną metodę Powella, w której nie podmienia się wektorów bazy, jeżeli spowodowałoby to jej zbyt dużą degenerację.

4 Podsumowanie

W trakcie laboratorium zbadano działanie pierwszej metody Powella na przykładzie dwóch funkcji.

Dla doliny bananowej dała ona zadowalający wynik, dość bliski rzeczywistemu minimum. Obliczenia zakończyły się w 11 iteracji.

Dla drugiej funkcji zaobserwowano efekty degeneracji bazy. Uzyskany wynik był znacznie gorszy. W pobliżu punktu w którym oszacowano minimum funkcja bardzo wolno zbiegała do wartości oszacowania wyznaczonej analitycznie. Nie pomogło zwiększenie żądanej dokładności obliczeń o trzy rzędy wielkości. Algorytm wykonał w wyniku tego tylko jedną iterację więcej, uzyskując nieznacznie lepsze rozwiązanie.

Rozwiązaniem tego problemu jest zmodyfikowana metoda Powella, która zapobiega degeneracji bazy.