

Sprawozdanie z laboratorium Teorii Optymalizacji

Imię i nazwisko	Jacek Gołda
Temat ćwiczenia	Druga metoda Powella
Data i godzina wykonania ćwiczenia	16 marca 2016, 14:00

1 Wstęp

Celem laboratorium było zbadanie własności drugiej metody Powella.

2 Ćwiczenie 2

Treść zadania:

„Dolina bananowa” Rosenbrocka. Wyznaczyć minimum funkcji:

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

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

Za parametr n przyjęto wartość 4.

$$Q(x_1, x_2) = 100[(x_2 + 48) - (x_1 + 48)^2]^2 + [1 - (x_1 + 48)]^2$$

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.

$$\begin{cases} (x_2 + 48) - (x_1 + 48)^2 = 0 \\ 1 - (x_1 + 48) = 0 \end{cases}$$

$$\begin{cases} x_2 = -47 \\ x_1 = -47 \end{cases}$$

Minimum będzie osiągane dla punktu $\hat{x} = [-47 \quad -47]^T$

2.2 Rozwiązanie numeryczne

Jako punkt początkowy przyjęto punkt $x_0 = [-49 \quad -49]^T$

W rozwiązaniu wykorzystano m-pliki realizujące drugą metodę Powella. Zmodyfikowano funkcję definiującą funkcję celu:

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

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

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

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

W pliku powe_2 dodano zapis wartości obecnego rozwiązania i nowego kierunku poszukiwań za pomocą instrukcji:

```
x_rozw = [x_rozw, xn];
kier_baz=[kier_baz, dm];
```

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

% funkcja rysujaca pojedyncza strzałke:
drawArrow = @(p1, p2) quiver(p1(1), p1(2), p2(1) - p1(1), p2(2) - p1(2), 0);

% rysowanie poziomic
[x1, x2] = meshgrid(-49.2:0.001:-46.5, -49.2:0.001:-45.8);
n = 4;
q = 100 * ( ( x2 + 3 * n ^ 2 ) - ( x1 + 3 * n ^ 2 ) .^ 2 ) .^ 2 + ...
    ( 1 - ( x1 + 3 * n .^ 2 ) ) .^ 2;
contour(x1, x2, q, [0.1, 2, 10, 40, 100, 200, 400], 'ShowText', 'on');

% druga metoda
x0 = [-49; -49];
metoda = 2;
pownad;
x_rozw

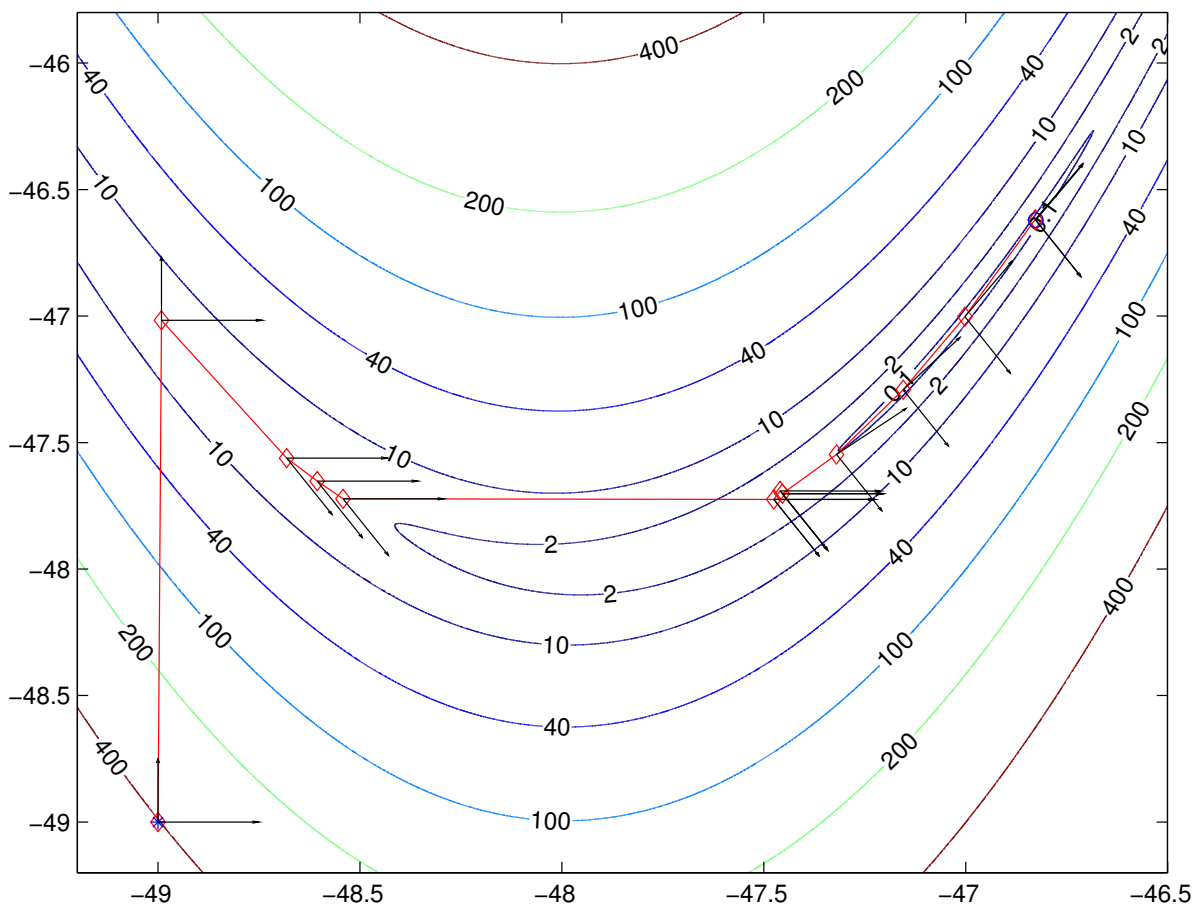
% rysuj punkty i linie
hold on;
plot(x_rozw(1, :), x_rozw(2, :), 'r');
plot(x_rozw(1, :), x_rozw(2, :), 'rd');

% 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');

% rysuj bazy w ktorych prowadzone bylo poszukiwanie
for i = 1:( size(x_rozw, 2) - 1 )
    drawArrow(x_rozw(:, i), x_rozw(:, i) + 0.25 * kier_baz(:, 2 * i - 1));
    drawArrow(x_rozw(:, i), x_rozw(:, i) + 0.25 * kier_baz(:, 2 * i));
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	-49	-49	404
2	-48,9916507183194	-47,0166288528566	3,96667258378229
3	-48,6815371740332	-47,5611673764769	2,89341234707553
4	-48,6060003111421	-47,6526211995466	2,61866933443331
5	-48,5416492850593	-47,7224995299857	2,40191100543253
6	-47,4752205331572	-47,7246330329337	0,225834625474513
7	-47,4752205331572	-47,7246330329337	0,225834625474513
8	-47,4594426746882	-47,6912358220117	0,238517410111691
9	-47,4539388926621	-47,7021974334374	0,206074970919278
10	-47,4539388926621	-47,7021974334374	0,206074970919278
11	-47,3197862763061	-47,5467384539587	0,111154175624051
12	-47,1547876588750	-47,2908963154774	0,0267472885678670
13	-47,0017328356156	-47,0027897339176	4,82868161012769e-05
14	-46,8278053320731	-46,6189616066034	0,0345482794738033
15	-46,8266369215112	-46,6212886723383	0,0304274066887947
16	-46,8266369215112	-46,6212886723383	0,0304274066887947

Rozwiązania nie zbiegają do wartości wyznaczonej analitycznie. Sprawdzono, czy metoda działa poprawnie i czy zapobiega

degeneracji bazy. Obliczono wyznaczniki macierzy stanowiących kierunki poszukiwań dla kolejnych punktów:

Numer iteracji	Wartość wyznacznika macierzy
1	1
2	1
3	-0,893676651811354
4	-0,893676651811354
5	-0,893676651811354
6	-0,893676651811354
7	0,892777170529466
8	0,892777170529466
9	0,892777170529466
10	-0,892777170529489
11	0,940579002676816
12	0,873657649841699
13	0,815534464993371
14	0,815534464993371
15	0,815534464993371
16	-0,815534464992863

Widać, że zgodnie z założeniem zaszytym w algorytmie wyznacznik macierzy kierunków bazowych jest co do modułu większy od wartości 0.8. Algorytm pracuje więc poprawnie, jednak nadal uzyskuje się niedokładne rozwiązanie.

Jest to dodatkowo dziwne, że algorytm w 13 iteracji trafił bardzo dokładnie w rozwiązanie. Położenie minimum zgadza się z tym wyznaczonym analitycznie co do trzeciego miejsca po przecinku, a wartość funkcji osiąga wartość rzędu 10^{-5} , czyli znacznie bliższą wartości wyznaczonej analitycznie.

Może być to efektem konstrukcji pliku `pownad` — badana metoda optymalizacji może zostać wywołana w nim ponownie, dla punktu początkowego przesuniętego o wektor jedynek w stosunku do ostatniego uzyskanego rozwiązania. Niewykluczone, że to „idealne trafienie” jest efektem normalnej pracy algorytmu, tylko następnie zostało sperturbowane przez wektor jedynek i w efekcie uzyskano gorsze rozwiązanie.

3 Ćwiczenie 3

Treść zadania:

Zbadać działanie metod Powella dla funkcji:

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

Rozwiązanie:

3.1 Oszacowanie analityczne

Funkcja ta posiada przesunięty argument w stosunku do funkcji badanej na poprzednim laboratorium. Wtedy oszacowano najmniejszą wartość funkcji jako 0. Wartość ta osiągana była dla punktu $x = [13 \ 13 \ 13 \ 13]^T$. Po przesunięciu argumentu tak, aby uzyskać funkcję badaną obecnie, uzyskano, że powyższe oszacowanie będzie odnosiło się do punktu $x = [-47 \ -47 \ -47 \ -47]^T$.

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)

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

n = 4;
x = x + 3 * n ^ 2;
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 );
```

Wyznaczano minimum funkcji dla punktu startowego $x_0 = [-50 \ -50 \ -50 \ -50]^T$

Napisano następujący m-plik służący do uruchamiania metody.

```
clear all;
close all;
clc

% druga metoda
x0 = -50 * ones(4, 1);
metoda = 2;
pownad;
```

Uzyskano następujące wyniki (ze względu na dość dużą ilość iteracji, zamieszczono dane o co siódmej iteracji):

Numer iteracji	Współrzędne x_1, x_2, x_3, x_4				Wartość funkcji
1	-50	-50	-50	-50	7218
7	-46,3538386062960	-45,0090227719492	-49,5667705425153	-51,5540078353008	92,5607453061778
14	-46,0496422217512	-44,1347828188393	-49,3612278632486	-50,4502540354841	14,7326470130358
21	-46,5227539818545	-45,8271132926138	-48,4037283507357	-48,0495282591478	2,87692876158391
28	-46,7331483858073	-46,3929396528482	-47,2644324877221	-47,6161730648562	0,310098551980737
35	-46,7437254874160	-46,4287030266371	-47,2240077565601	-47,5401427266651	0,258843907356583
42	-46,6097517485153	-46,0678887556174	-48,0025638202846	-47,9639175112724	1,64425814887170
49	-46,7912923715343	-46,5371612354930	-47,1841230751918	-47,4643993576421	0,168861034033394
56	-46,8425455907201	-46,6487067869517	-47,1195504099170	-47,3208861094141	0,107993545830060
63	-46,6753627563172	-47,4359508804805	-47,9061311397970	-47,8656247725561	161,247413912414
70	-46,6018218539203	-46,0483746495023	-47,6987798128799	-47,9856539411664	1,04997164262935
77	-46,9831170714585	-46,9657406933900	-47,0146025487196	-47,0431717580900	0,00189648637478373
84	-46,5208010358510	-45,8001407861780	-47,5593587077647	-47,9818234036051	1,91648865471456
91	-46,7407971191840	-46,4264086004280	-47,2887564783831	-47,6516711594910	0,387891241254151
98	-46,8350620310581	-46,6448993092275	-47,1336533790372	-47,3410331775054	0,102749831935425
105	-46,9747788373675	-46,9503509909614	-47,0127265776586	-47,0358354950875	0,00395278817794351
112	-46,9817085443403	-46,9636485840259	-47,0093188134108	-47,0256180097826	0,00237843900524028
119	-46,9828688753212	-46,9658714869185	-47,0087490334415	-47,0239096899728	0,00216945674274844
126	-46,9830748632629	-46,9662659750984	-47,0086479234429	-47,0236065289620	0,00213395012262919
133	-46,9831117899997	-46,9663366892733	-47,0086297991681	-47,0235521859747	0,00212763554122124
140	-46,9831184212009	-46,9663493877890	-47,0086265445052	-47,0235424273290	0,00212650321326697
147	-46,9831196123834	-46,9663516688567	-47,0086259598618	-47,0235406743581	0,00212629986230105

Widać, że algorytm zbiega do oczekiwanego minimum. Uzyskiwana jest wartość funkcji celu rzędu 10^{-3} . Chwilowe zakłócenie w 63 iteracji (nagły wzrost wartości funkcji celu do wartości ≈ 161 wynika z konstrukcji pliku `pownad` — algorytm startuje drugi raz z punktu przesuniętego o wektor jedynej w stosunku do ostatnio znalezionej. Z tego wynika to zaburzenie w pracy.

Co ważne, nie zaobserwowano efektu degeneracji bazy, który objawił się w tym przykładzie w trakcie ostatnich laboratoriów.

4 Podsumowanie

W trakcie laboratorium zbadano działanie drugiej metody Powella dla dwóch funkcji.

Dla doliny bananowej ponownie dała ona dość zadowalający efekt — wyznaczono minimum z nieco gorszą dokładnością niż za pomocą pierwszej metody Powella, niemniej jednak było to relatywnie dobre przybliżenie. Obliczenia trwały nieco dłużej niż w przypadku pierwszej metody Powella.

Dla drugiej funkcji uzyskano znacznie lepsze przybliżenie minimum niż w przypadku pierwszej metody Powella. Uzyskana obecnie wartość funkcji celu jest o trzy rzędy wielkości mniejsza (oczekiwana wartość to 0). Jest to wynikiem wyeliminowania efektu degeneracji bazy kierunków.