

Laboratorium

Metod Obliczeniowych Optymalizacji

Grupa dzień tygodnia, godz: czwartek 14:15

ĆWICZENIE 3

Optymalizacja wielowymiarowa bez ograniczeń

Lp.	Nazwisko i imię	Obecność ocena	Sprawozdanie ocena	Ocena średnia	Uwagi
1	Zawiślak Łukasz				
2	Klimczak Krzysztof				

Ćwiczenie oddano:

Podpis prowadzącego:.....

Cel zadania

Celem zadania trzeciego była implementacja programu pozwalającego na prześledzenie procesu odnajdowania minimum funkcji, dla metody Fletchera-Reevesa (metoda gradientów sprzężonych).

Wprowadzenie

Zadaniem programu jest odnalezienie minimum funkcji dwóch zmiennych wykorzystując do tego celu metodę Fletchera-Reevesa. Algorytm ten należy do klasy metod gradientów sprzężonych, co oznacza że w procesie minimalizacji wykorzystywane są gradienty funkcji celu pozwalające na otrzymanie kolejnych sprzężonych kierunków poszukiwań (Poprzez sprzężenie rozumiana jest zależność kolejnego kierunku poszukiwań, od poprzednio otrzymanego).

Zasada działania algorytmu Fletchera-Reevesa, sprowadza się do minimalizacji funkcji celu na kolejno otrzymywanych sprzężonych kierunkach poszukiwań, aż do momentu osiągnięcia zakładanej dokładności. Przy czym metoda minimalizacji na kierunku nie jest narzucona.

Na potrzeby tego zadania w zaimplementowanym algorytmie do celów minimalizacji kierunkowej wykorzystywane jest kryterium Armijo, ze wzoru:

$$f(\lambda + \lambda_k) \leq f(\lambda) + c_1 \cdot \lambda_k \cdot f'(\lambda)$$

Gdzie:

λ – dotychczas wykonany krok

λ_k – krok k-tej iteracji

c_1 – stała kryterium Armijo $c_1 \in (0,1)$

Sprzężenie kolejnych kierunków poszukiwań osiągane jest dzięki zastosowaniu współczynnika α , przedstawionego wzorem:

$$\alpha_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}$$

$$d_k = -\nabla(f(x_k)) + \alpha_k d_{k-1}$$

Gdzie:

k – k-ta iteracja

x_k – przybliżone rozwiązanie w k-tej iteracji

∇f – gradient funkcji celu w zadanym punkcie

d_k – kierunek poszukiwań w k-tej iteracji

Uwaga: dla $k \bmod 2 == 0$ wartość współczynnika α przyjmuje wartość 0.

Przyjęto że algorytm poszukiwania minimum kończy działanie gdy zostanie osiągnięta zadana dokładność, dla warunku stopu przedstawionego wzorem:

$$\|x_k - x_{k-1}\| < \varepsilon$$

Gdzie:

ε – oczekiwana dokładność

Algorytm

1. Przygotowanie wstępne, tworzenie i alokacja zmiennych.
2. Sprawdzenie warunku stopu. Jeśli prawdziwe, kończy algorytm.
3. $k=k+1$;
4. Obliczenie nowego kierunku poszukiwań d_k , z uwzględnieniem $\alpha=0$, gdy $k \bmod 2 == 0$

5. Wyliczeni kroku minimalizacji λ , korzystając z kryterium armijo.
6. Wyliczenie przybliżonego rozwiązania: $x_{k+1} = x_k + \lambda d_k$. Przejście do kroku 2

Testy

Testy zostały przeprowadzone dla dwóch różnych funkcji dwóch zmiennych. Poprawność wyników sprawdzono korzystając z serwisu wolframAlpha. Jako że wydajność algorytmu w głównej mierze zależy od zastosowanej metody minimalizacji kierunkowej, zarzucono próby porównywania wydajności, skupiając się na zaprezentowania przebiegu działania algorytmu.

Test 1

Dla funkcji postaci :

$$f(x, y) = (x - 2)^4 + (x - 2y)^2$$

Znalezione przez system wolfram alpha rozwiązanie:

$$\min = f(2, 1) = 0$$

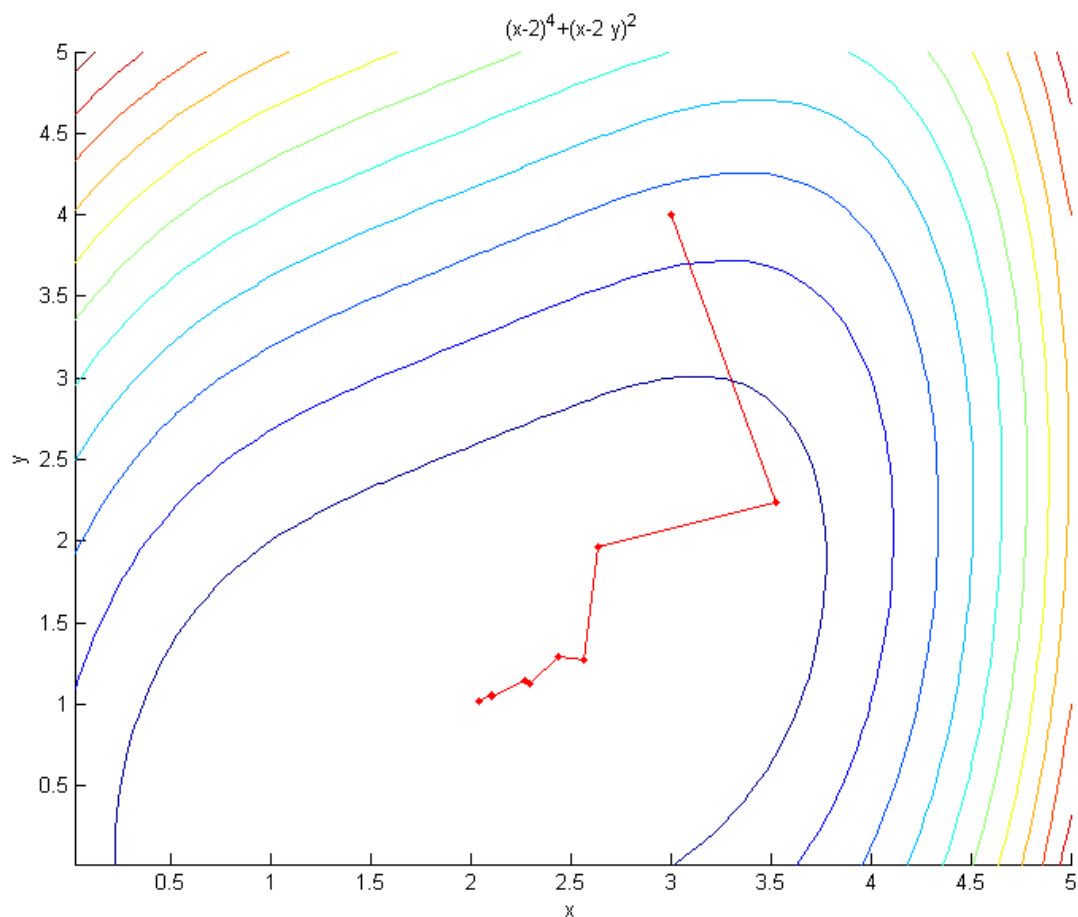
Test 2

Dla funkcji postaci :

$$f(x, y) = 10(y - x^2)^2 + (1 - x)^2$$

Znalezione przez system wolfram alpha rozwiązanie:

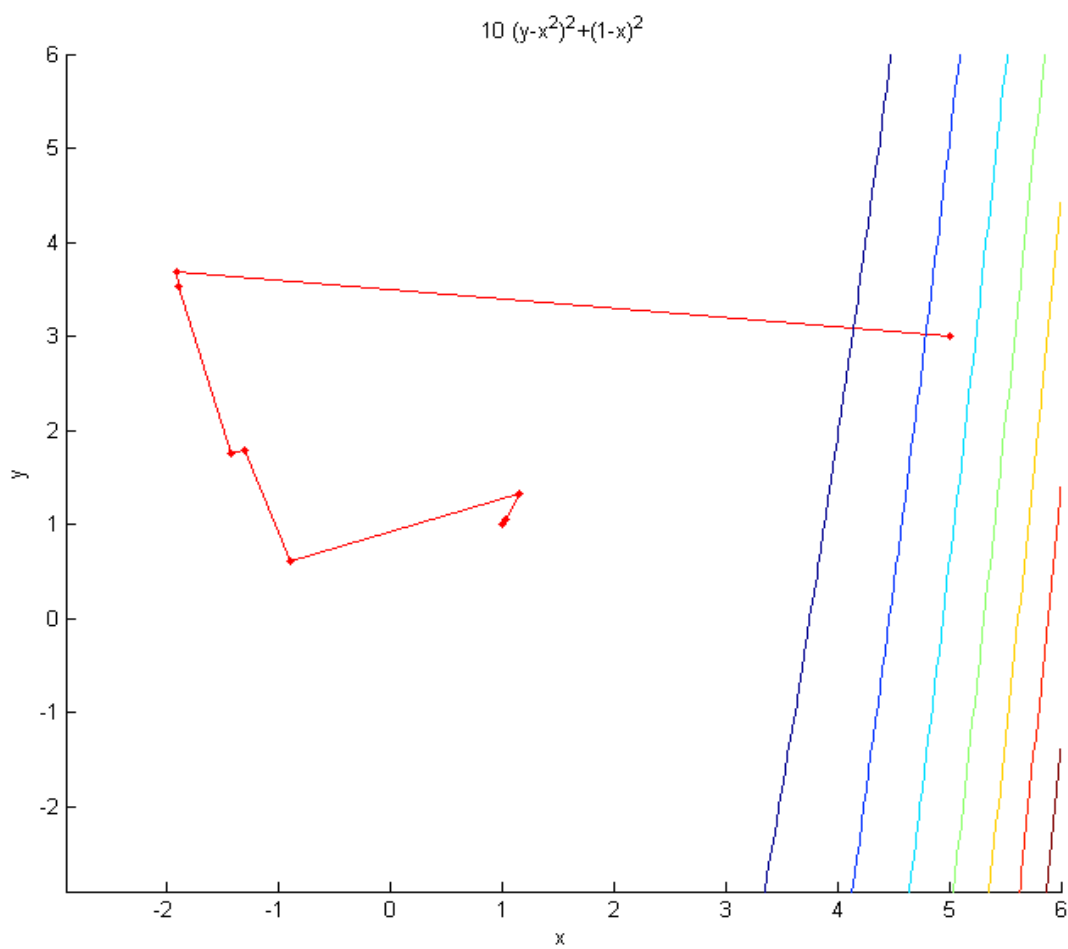
$$\min = f(1, 1) = 0$$



Rysunek 1. Graficzne rozwiązanie test 1, punkt startowy [3,4], dokładność 0.001.

Tabela 1. Kroki algorytmu test 1, punkt startowy [3,4], dokładność 0.001.

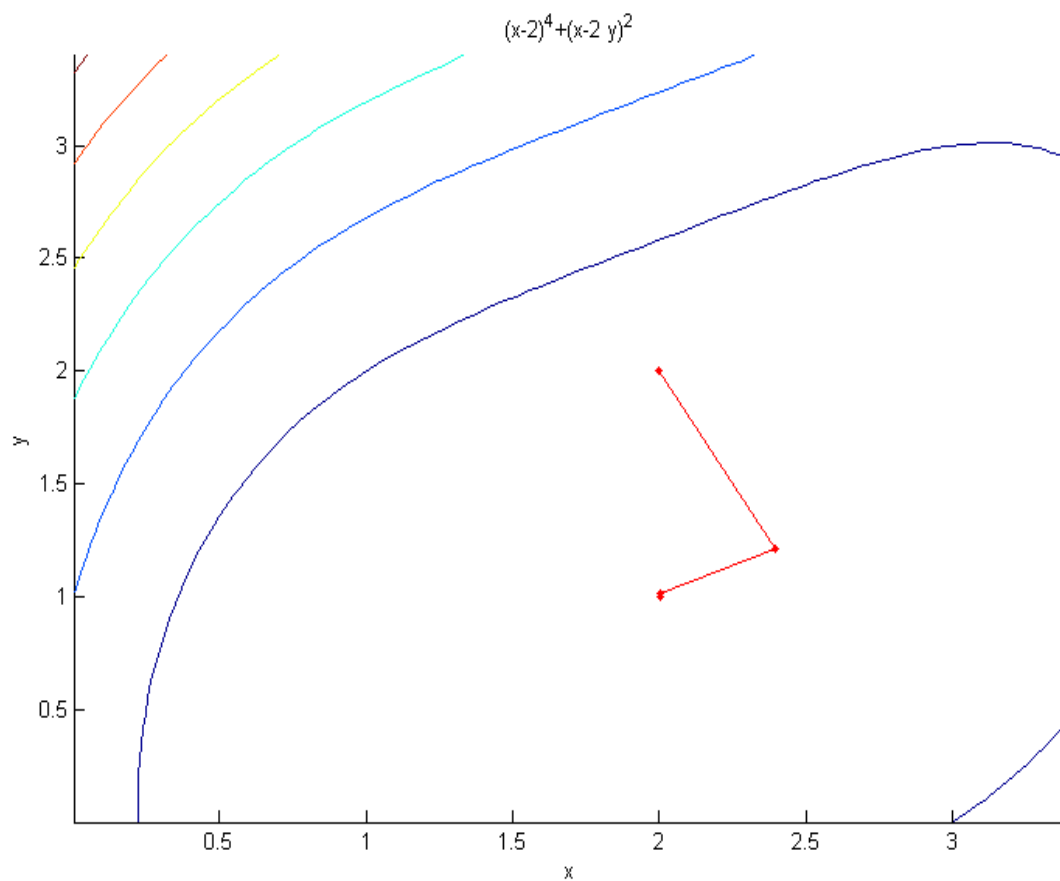
k	x_k	x_{k-1}	$f(x_k)$	$f(x_{k-1})$	Stop	α	d_k	λ
1	[3.000000,4.000000]	[3.530246,2.232511]	26.000000	6.357148	1.845313	1.786884	[16.721296, -55.737689]	0.031711
2	[3.530246,2.232511]	[2.636954,1.964524]	6.357148	1.834105	0.932624	0.000000	[-12.463686, -3.739101]	0.071672
3	[2.636954,1.964524]	[2.566475,1.273470]	1.834105	0.103355	0.694638	0.171955	[-0.592684, -5.811328]	0.118915
4	[2.566475,1.273470]	[2.435941,1.286783]	0.103355	0.055057	0.131211	0.000000	[-0.766187, 0.078141]	0.170368
5	[2.435941,1.286783]	[2.294423,1.126940]	0.055057	0.009158	0.213488	0.516227	[-0.451673, -0.510158]	0.313320
6	[2.294423,1.126940]	[2.274406,1.144661]	0.009158	0.005892	0.026734	0.000000	[-0.183176, 0.162173]	0.109274
7	[2.274406,1.144661]	[2.107796,1.046752]	0.005892	0.000339	0.193249	0.106086	[-0.072252, -0.042459]	2.305970
8	[2.107796,1.046752]	[2.104433,1.052475]	0.000339	0.000119	0.006638	0.000000	[-0.033595, 0.057169]	0.100108
9	[2.104433,1.052475]	[2.038740,1.019150]	0.000119	0.000002	0.073662	0.003794	[-0.003650, -0.001851]	18.000000
10	[2.038740,1.019150]	[2.038609,1.019357]	0.000002	0.000002	0.000245	0.000000	[-0.001113, 0.001762]	0.117649



Rysunek 2. Graficzne rozwiązanie test 2, punkt startowy [5,3], dokładność 0.001.

Tabela 2. Kroki algorytmu test 2, punkt startowy [5,3], dokładność 0.001.

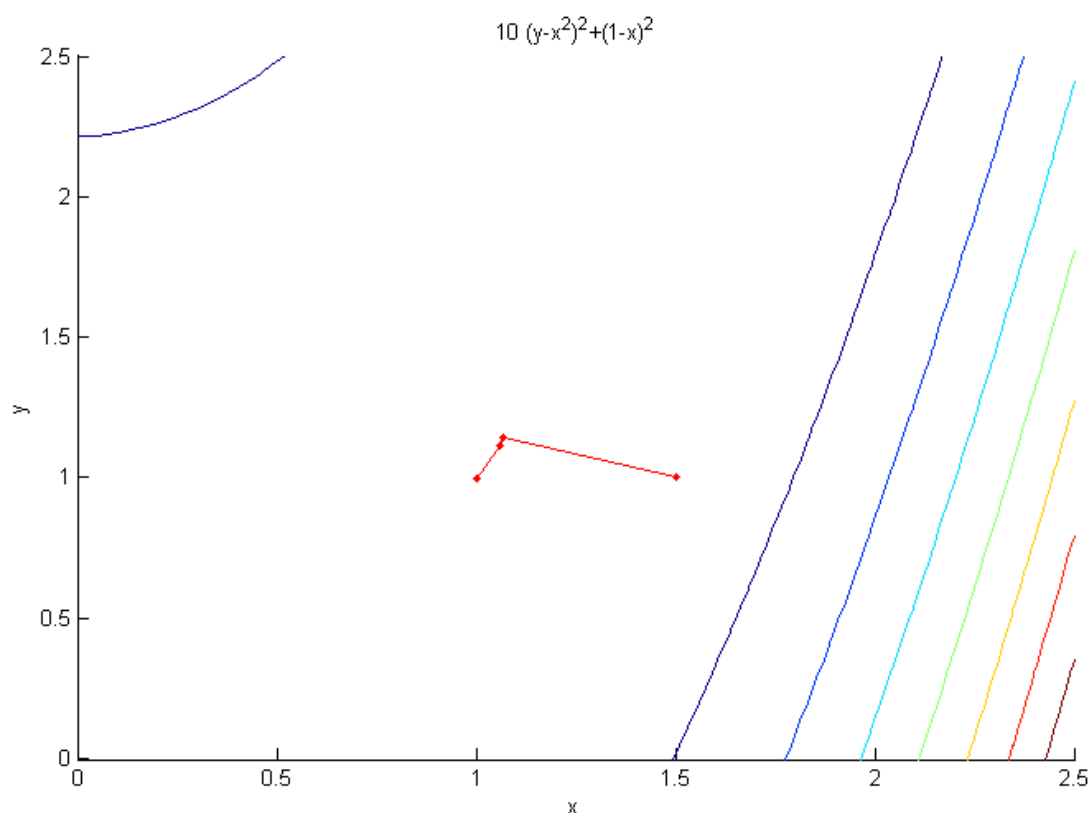
k	x_k	x_{k-1}	$f(x_k)$	$f(x_{k-1})$	Stop	α	d_k	λ
1	[5.000000,3.000000]	[-1.901114, 3.688859]	4856.000000	8.472151	6.935409	20.550993	[-94996.781440, 9482.436919]	0.000073
2	[-1.901114,3.688859]	[-1.887074, 3.524611]	8.472151	8.348469	0.164847	0.000000	[0.127584, -1.492472]	0.110051
3	[-1.887074,3.524611]	[-1.420093, 1.759256]	8.348469	6.519442	1.826075	32.622857	[12.686616, -47.960004]	0.036809
4	[-1.420093,1.759256]	[-1.307272, 1.789100]	6.519442	5.387728	0.116702	0.000000	[19.462003, 5.148169]	0.005797
5	[-1.307272,1.789100]	[-0.888547, 0.608391]	5.387728	3.894671	1.252759	0.006782	[0.556029, -1.567874]	0.753064
6	[-0.888547,0.608391]	[1.150566, 1.331536]	3.894671	0.023268	2.163543	0.000000	[10.214642, 3.622495]	0.199626
7	[1.150566,1.331536]	[1.151507, 1.329002]	0.023268	0.023046	0.002704	0.000229	[0.057161, -0.153874]	0.016472
8	[1.151507,1.329002]	[1.149069, 1.328096]	0.023046	0.022820	0.002601	0.000000	[-0.163350, -0.060664]	0.014925
9	[1.149069,1.328096]	[1.032474, 1.054478]	0.022820	0.002383	0.297425	0.897122	[-0.089127, -0.209158]	1.308191
10	[1.032474,1.054478]	[1.027394, 1.056642]	0.002383	0.000763	0.005522	0.000000	[-0.540912, 0.230477]	0.009391
11	[1.027394,1.056642]	[1.000657, 1.000607]	0.000763	0.000005	0.062087	0.001669	[-0.010354, -0.021699]	2.582351
12	[1.000657,1.000607]	[1.000366, 1.000746]	0.000005	0.000000	0.000323	0.000000	[-0.029696, 0.014161]	0.009823



Rysunek 3. Graficzne rozwiązanie test 1, punkt startowy [2,2], dokładność 0.001.

Tabela 2. Kroki algorytmu test 1, punkt startowy [2,2], dokładność 0.001.

k	x_k	x_{k-1}	$f(x_k)$	$f(x_{k-1})$	Stop	α	d_k	λ
1	[2.000000,2.000000]	[2.395067, 1.209866]	4.000000	0.024969	0.883397	0.078125	[4.312500, -8.625000]	0.091610
2	[2.395067,1.209866]	[2.004289, 1.014478]	0.024969	0.000608	0.436903	0.000000	[-0.197318, -0.098658]	1.980454
3	[2.004289,1.014478]	[2.004288, 1.002144]	0.000608	0.000000	0.012334	0.250060	[-0.000006, -0.123341]	0.100000
4	[2.004288,1.002144]	[2.004288, 1.002144]	0.000000	0.000000	0.000000	0.000000	[-0.000000, -0.000001]	0.000000



Rysunek 4. Graficzne rozwiązanie test 2, punkt startowy [1.5,1], dokładność 0.001.

Tabela 2. Kroki algorytmu test 1, punkt startowy [2,2], dokładność 0.001.

k	x_k	x_{k-1}	$f(x_k)$	$f(x_{k-1})$	Stop	α	d_k	λ
1	[1.500000,1.000000]	[1.067490, 1.142273]	15.875000	0.004630	0.455310	4.542941	[-421.263857, 138.573528]	0.001027
2	[1.067490,1.142273]	[1.057550, 1.112134]	0.004630	0.003706	0.031736	0.000000	[-0.018066, -0.054781]	0.550175
3	[1.057550,1.112134]	[1.000366, 0.997141]	0.003706	0.000129	0.128427	48.297310	[-1.253260, -2.520202]	0.045628
4	[1.000366,0.997141]	[0.998944, 0.997848]	0.000129	0.000001	0.001587	0.000000	[-0.144434, 0.071805]	0.009840
5	[0.998944,0.997848]	[0.998944, 0.997848]	0.000001	0.000001	0.000000	0.000033	[0.000378,0.000847]	0.000000

Podsumowanie

Metoda gradientów sprzężonych okazała się, prosta w swoich założeniach, łatwa w implementacji, oraz poprawna w działaniu. Wszystkie uzyskane rozwiązania, w dużym stopniu pokrywały się z wartościami zwróconymi przez system wolfram Alpha.

Należy zauważyć że metoda gradientów sprzężonych służy jedynie do uzyskania ciągu kierunków poszukiwania, więc wydajność tej metody w dużej mierze zależy do zastosowanego algorytmu minimalizacji na kierunku.

Niezależnie od położenia punktu startowego wyniki okazały się poprawne, nie mniej jednak warto zwrócić uwagę że liczba iteracji zmniejsza się jeśli pierwsze przybliżenie znajduje się bliżej właściwego rozwiązania, jakość rozwiązania również wzrasta. Wzrost jakości rozwiązania może wynikać z powszechnego użycia przybliżonych wartości pochodnych; większa liczba iteracji oznacza akumulację błędów wynikających z oszacowania pochodnych, stąd różnica w jakości wyniku.

Literatura

- [1] - Arkadiusz Tomczyk. Wykład 3 Optymalizacja, 30 marca 2011 [dostęp: 5 Kwietnia 2011]. Dostępne w Internecie: <http://ftims.edu.p.lodz.pl/file.php/33/wyklad3.pdf>
- [2] - Arkadiusz Tomczyk. Wykład 6 Optymalizacja, 14 kwietnia 2011 [dostęp: 3 Maj 2011]. Dostępne w Internecie: <http://ftims.edu.p.lodz.pl/file.php/33/wyklad6.pdf>
- [3] – Wojciech Grega. Metody Optymalizacji, wykład 4, 30 marca 2011 [dostęp: 5 Kwietnia 2011]. Dostępne w Internecie: <http://aq.ia.agh.edu.pl/Aquarium/Dydaktyk/Wyklady/MO/2006-07/Wyklad04.PDF>
- [4] - Jacek Grabowski. Praca dyplomowa. Pakiet ćwiczeń dla potrzeb laboratorium metod optymalizacji, Wrocław 2006 [dostęp: 3 Maj 2011]. Dostępne w Internecie: http://www.iit.pwr.wroc.pl/~brzostowski/optymalizacja/numeryczne_metody_optymalizacji.pdf
- [5] - Dokumentacja klasy FletcherReeves. [dostęp: 3 Maj 2011] Dostęp w Internecie: <http://eduo2.studio4plus.com/methods/classFletcherReeves.html>