WSI 1

Adam Sokołowski 324892

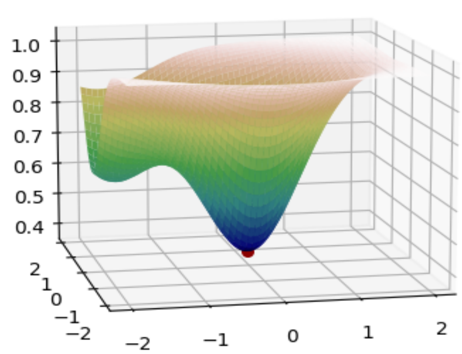
March 2024

1 Cel ćwiczenia

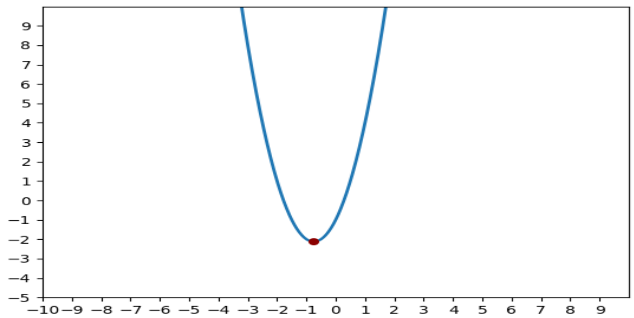
Znalezienie minimum zadanych funkcji jedno i dwu wymiarowych przy użyciu gradientu prostego, oraz przeprowadzenie eksperymentów które pozwolą zbadanie wpływu rozmiaru kroku dla losowych punktów startowych.

2 Przeprowadzone eksperymenty

Na początku aby sprawdzić poprawność działania programu znalazłem minima funkcji oraz naniosłem je na poniższe wykresy.



Rysunek 1: Wykres funkcji g z zaznaczonym minimum



Rysunek 2: Funkcja f z zaznaczonym minimum

Po zobaczeniu, że algorytm działa poprawnie dla obu funckji sprawdziłem ręcznie dla jakich bet oraz punktów startowych przestaje działać. Przestawał on działać wtedy gdy punkty startowe (tylko w przypadku funkcji g) były zbyt blisko drugiego minimum lokalnego(nie było ono globalne) lub zbyt daleko oddalone od szukanego minimum. Zbyt dalekie oddalenie skutkowało tym, że metoda którą obliczałem gradient zwracała 0, bo funkcja była bardzo "płaska".

```
(array([100, 100]), 0, 1.0)
```

Rysunek 3: Wynik dla punktu za daleko oddalonego od minimum, gdzie pierwsza wartość to minimum, druga to liczba iteracji, a trzecia wartość w minimum dla funkcji g

Bety dla obu funckji musiały być w jakimś przedziale, bo dla zbyt małych bet algorytm potrzebował bardzo dużo iteracji do znalezienia minimum, a dla zbyt dużych nie znajdował minimum i wynik zbiegał do nieskończoności.

```
(array([-2.37308055e+15]), 29, array([1.12630226e+31]))
```

Rysunek 4: Wynik dla za dużej bety dla funkcji f

beta	avg	std
0.1	37	2
0.11	33	2
0.12	29	1
0.13	26	1
0.14	23	1
0.15	21	1
0.16	19	1
0.17	17	1
0.18	15	1
0.19	14	1
0.2	12	1
0.21	11	1
0.22	9	1
0.23	8	0
0.24	6	0
0.25	1	0
0.26	6	0
0.27	8	0
0.28	9	1
0.29	11	1
0.3	12	1
0.31	14	1
0.32	15	1
0.33	17	1
0.34	19	1
0.35	21	1
0.36	23	1
0.37	26	1
0.38	29	2
0.39	33	2
0.4	37	2
0.41	43	2
0.42	49	3
0.43	58	3
0.44	69	4

(a) funkcja f

beta	avg	std
0.2	130	267
0.25	103	298
0.3	85	129
0.35	72	124
0.4	63	150
0.45	55	110
0.5	49	110
0.55	44	71
0.6	40	93
0.65	36	93
0.7	33	80
0.75	31	84
0.8	28	93
0.85	26	65
0.9	24	70
0.95	24	75
1	25	61
1.05	25	60
1.1	25	53
1.15	26	39
1.2	27	49
1.25	29	50
1.3	31	51
1.35	34	44
1.4	38	40
1.45	45	46

(b) funkcja g

Rysunek 5: Średnia liczba iteracji i odchylenie standardowe dla danej bety dla 1000 losowych punktów

Dla funkcji f zdecydowałem się testować wpływ bet i punktów startowych wybranych losowo na przedziałach: $\beta \in [0.1, 0.45]$, $x0 \in [-1000, 1000]$.
Dla funkcji g: $\beta \in [0.2, 1.5]$, $x0_1 \in [-0.3, 2.5]$, $x0_2 \in [-2, 1.8]$.

3 Wnioski i wyniki

3.1 Minima funkcji i wartości w nich

Dla funkcji f minimum to -0.75, a wartość w nim to -2.125.
Dla funkcji g minimum to -0.02195, a wartość w nim to 0.39277.

3.2 Wnioski

Najlepsza beta funkcji f to 0.25. Widać to na zdjęciu po lewej stronie Rysunku 5. Średnia liczba iteracji to 1, a odchylenie standardowe to zero.

Najlepsza beta funkcji g znajduje się gdzieś na przedziale [0.9, 1.1]. Średnie liczby iteracji delikatnie się różnią przy różnych wywołaniach programu, tak samo jak odchylenia standardowe. Można jednak zauważyć, że dla większych beta odchylenia są mniejsze, zatem większe bety lepiej sobie radzą z dalej oddalonymi punktami startowymi. Można dobierać bety w zależności od problemu.