

WSI - ćwiczenie 1

Agnieszka Wójtowicz

11 marca 2022

Spis treści

1	Polecenie oraz elementy języka	1
1.1	Polecenie Prowadzącego	1
1.2	Elementy języka	2
2	Cel zadania	2
3	Funkcja celu	2
3.1	Gradient oraz Hesjan	3
4	Wyniki	4
4.1	Metoda najszybszego spadku gradientu	4
4.2	Metoda Newtona	7
5	Wnioski	10
5.1	Metoda najszybszego spadku gradientu	10
5.2	Metoda Newtona	11
5.3	Porównanie	11

1 Polecenie oraz elementy języka

1.1 Polecenie Prowadzącego

Pierwszym tematem ćwiczeń jest zagadnienie przeszukiwania i związane z tym podejścia. Państwa zadaniem będzie minimalizacja funkcji celu i porównanie wyników dla metody najszybszego spadku gradientu i metody Newtona. W raporcie oczekiwałbym, że opisz Państwo zaobserwowane różnice w działaniu tych metod dla różnych punktów początkowych z zadanego zakresu (należałoby podać ilość iteracji i pomiary czasowe). Zadane przeze mnie funkcje posiadają minima lokalne i globalne, dlatego warto, żeby udało się je Państwu znaleźć. Istnieje możliwość, że dla danego punktu startowego funkcja nie znajdzie minimum, dlatego liczę na Państwa wnioski w tym przypadku. Wizualizacje nie są przeze mnie wymagane, chociaż bardzo Państwu pomogą w interpretacji wyników np. zmiana położenia kolejnych współrzędnych punktów.

1.2 Elementy języka

Wykorzystano język programowania Python w wersji 3.8. Skorzystano z pomocniczych bibliotek: *numpy 1.20.1*, *matplotlib 3.4.3*, *pandas 1.2.4*, *time* oraz *sympy 1.8* (w celu wyznaczenia pochodnych).

2 Cel zadania

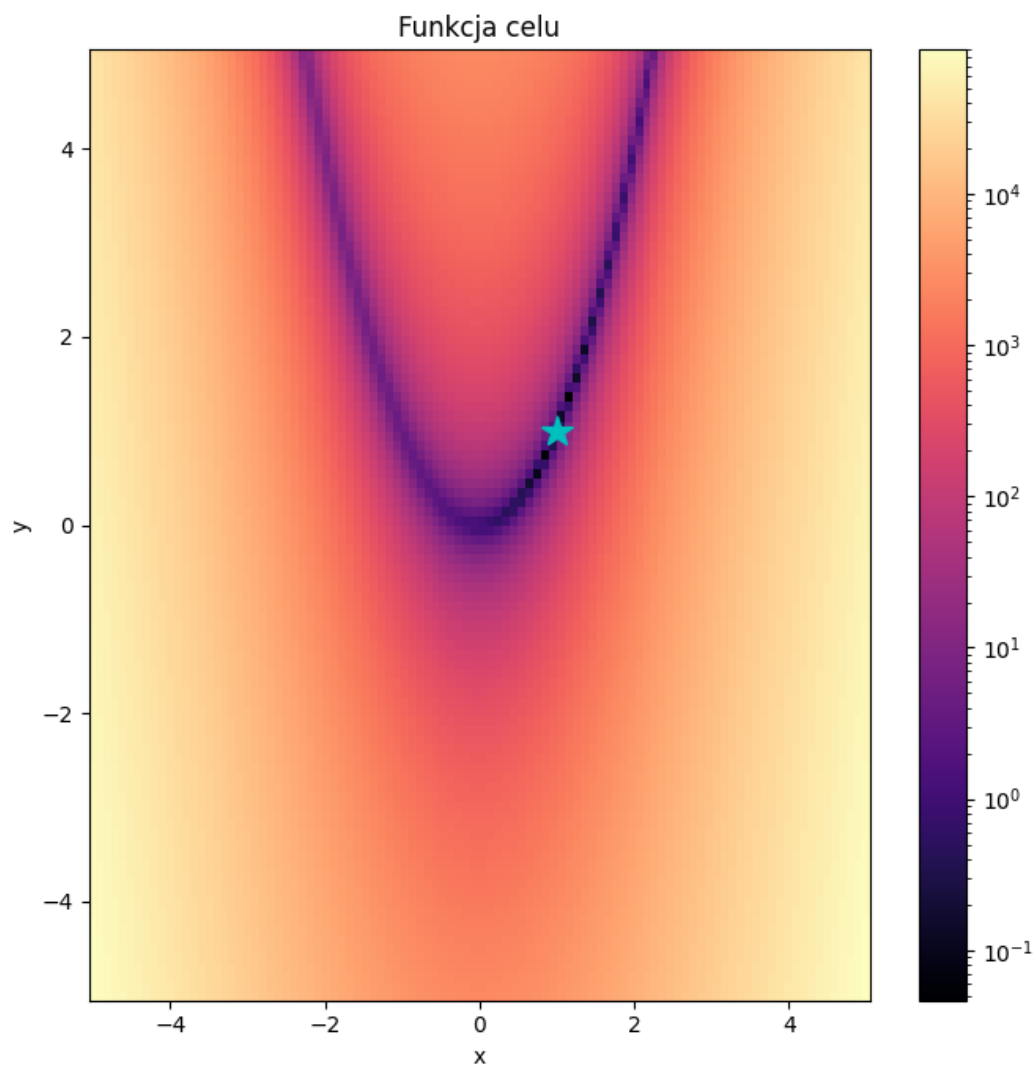
Celem zadania jest implementacja algorytmu najszybszego spadku gradientu (*Steepest Gradient Descent*) oraz metody Newtona (*Newton's Method*), a następnie prezentacja działania algorytmów w zależności od zadanych parametrów *beta* (krok uczenia) oraz położenia punktu początkowego. Porównane zostanie również działanie obu algorytmów.

3 Funkcja celu

Dla mnie, jako osoby z nazwiskiem na "W" zadana została funkcja bananowa Rosenbrocka postaci:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2, \quad (1)$$

której wykres dla zadanego zakresu wartości x oraz y widnieje poniżej (Rysunek 1. Zaznaczono na nim wyznaczone analitycznie jedynym minimum (globalne) funkcji wynoszące 0, dla punktu $[1, 1]$. Funkcja nie jest funkcją wypukłą.



Rysunek 1: Funkcja celu w zadanych zakresach wartości.

3.1 Gradient oraz Hesjan

Wyznaczone (korzystając z biblioteki *sympy*) gradient oraz hesjan funkcji celu 1 znajdują się poniżej.

$$GRAD = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} 2(200x^3 - 200xy + x - 1) \\ 200(y - x^2) \end{bmatrix} \quad (2)$$

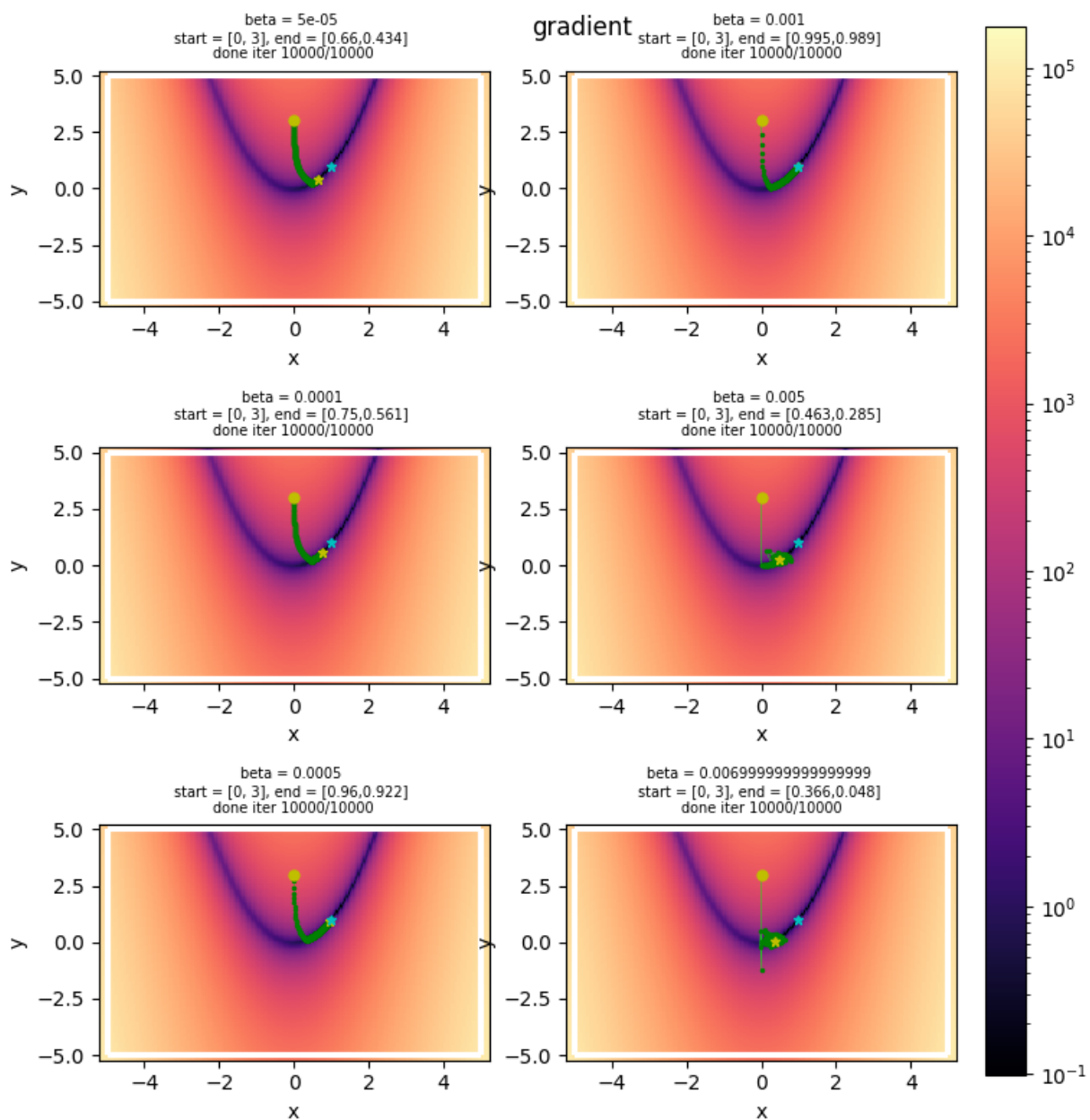
$$HES = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 1200x^2 - 400y + 2 & -400x \\ -400x & 200 \end{bmatrix} \quad (3)$$

4 Wyniki

Dla obu metod, wybrano parametr minimalnej dopuszczalnej zmiany wartości funkcji po stawieniu kroku $\epsilon = 10E - 12$ oraz maksymalną ilość iteracji $\max_iter = 10\,000$ i $40\,000$ w przypadku metody spadku gradientu, oraz 5000 dla metody Newtona (mocno przeszacowaną do góry, by niezależnie od wielkości β dać algorytmowi dużą szansę osiągnięcia celu). We wstępnych próbach zauważono również, że w przypadku algorytmu Newtona, rząd wielkości β powinien być znacznie większy niż dla metody spadku gradientu, stąd dla metod dobierano parametry z diametralnie innych zakresów wartości.

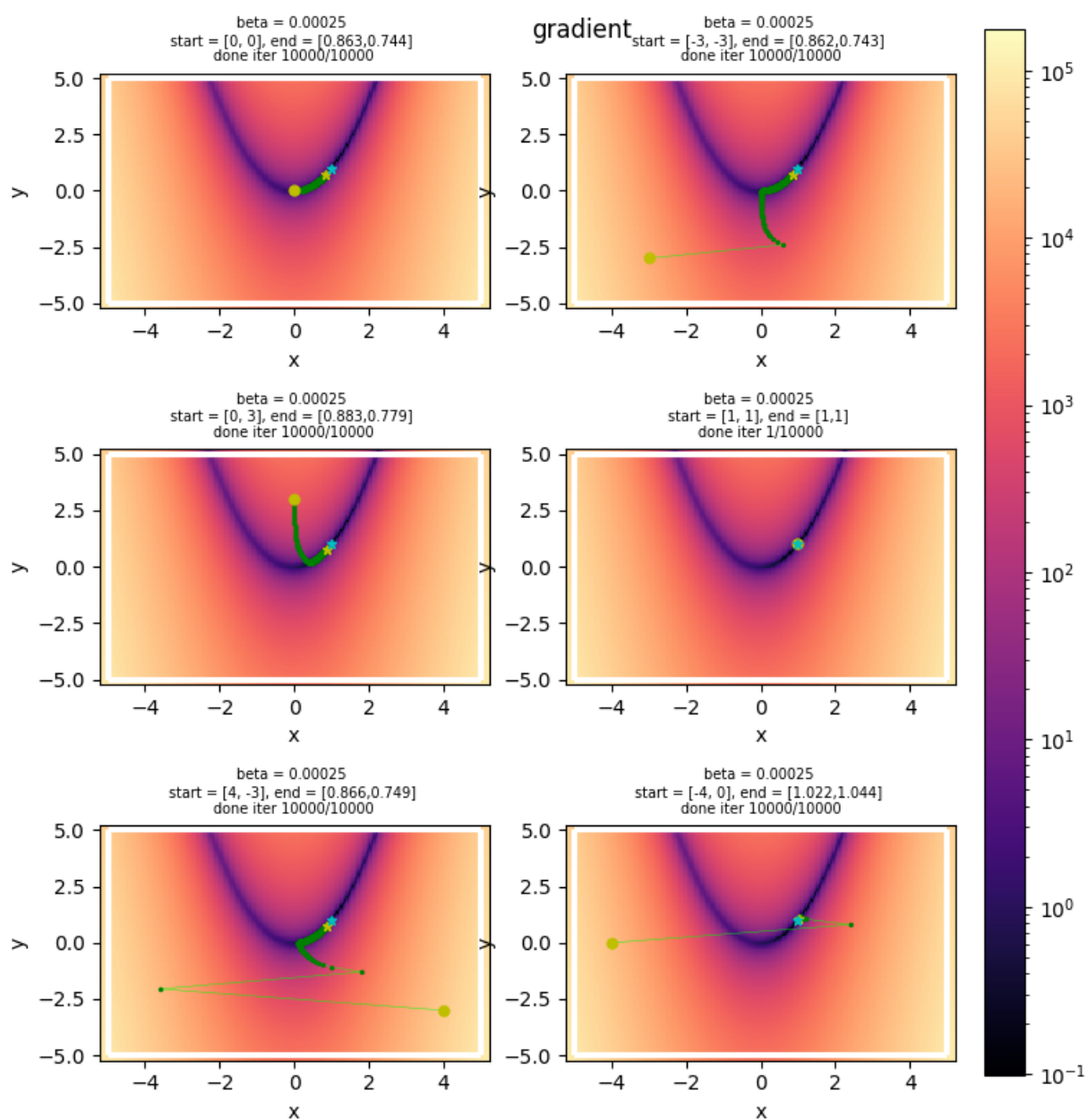
4.1 Metoda najszybszego spadku gradientu

Rysunek 2 przedstawia kilka wywołań metody spadku gradientu dla kilku różnych wartości parametru β dla danego punktu początkowego.

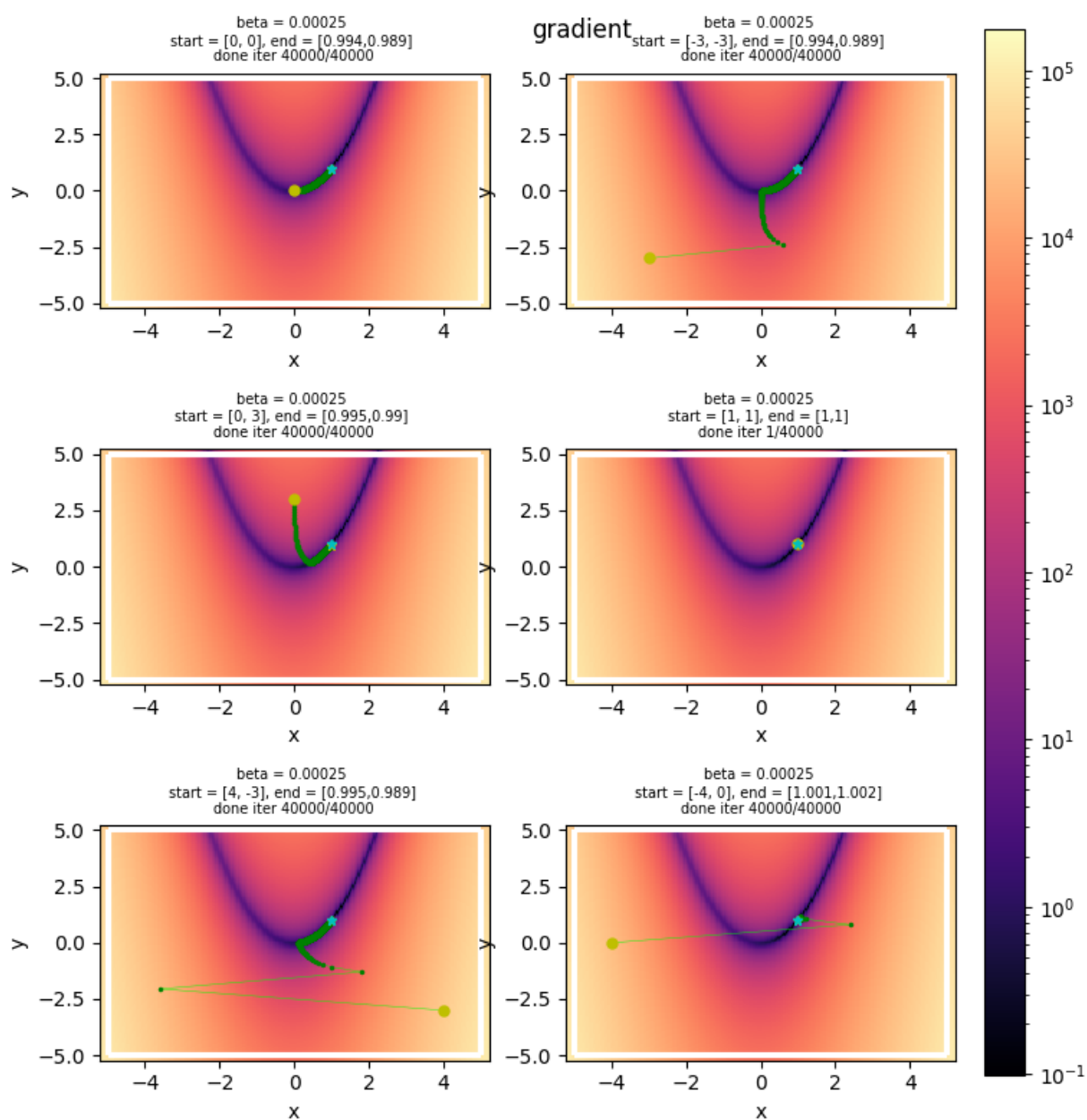


Rysunek 2: Metoda najszybszego spadku gradientu dla różnych wartości parametru β .

Rysunki 3 i 4 przedstawiają kilka wywołań metody spadku gradientu dla kilku różnych punktów początkowych oraz dwóch liczb maksymalnych iteracji.



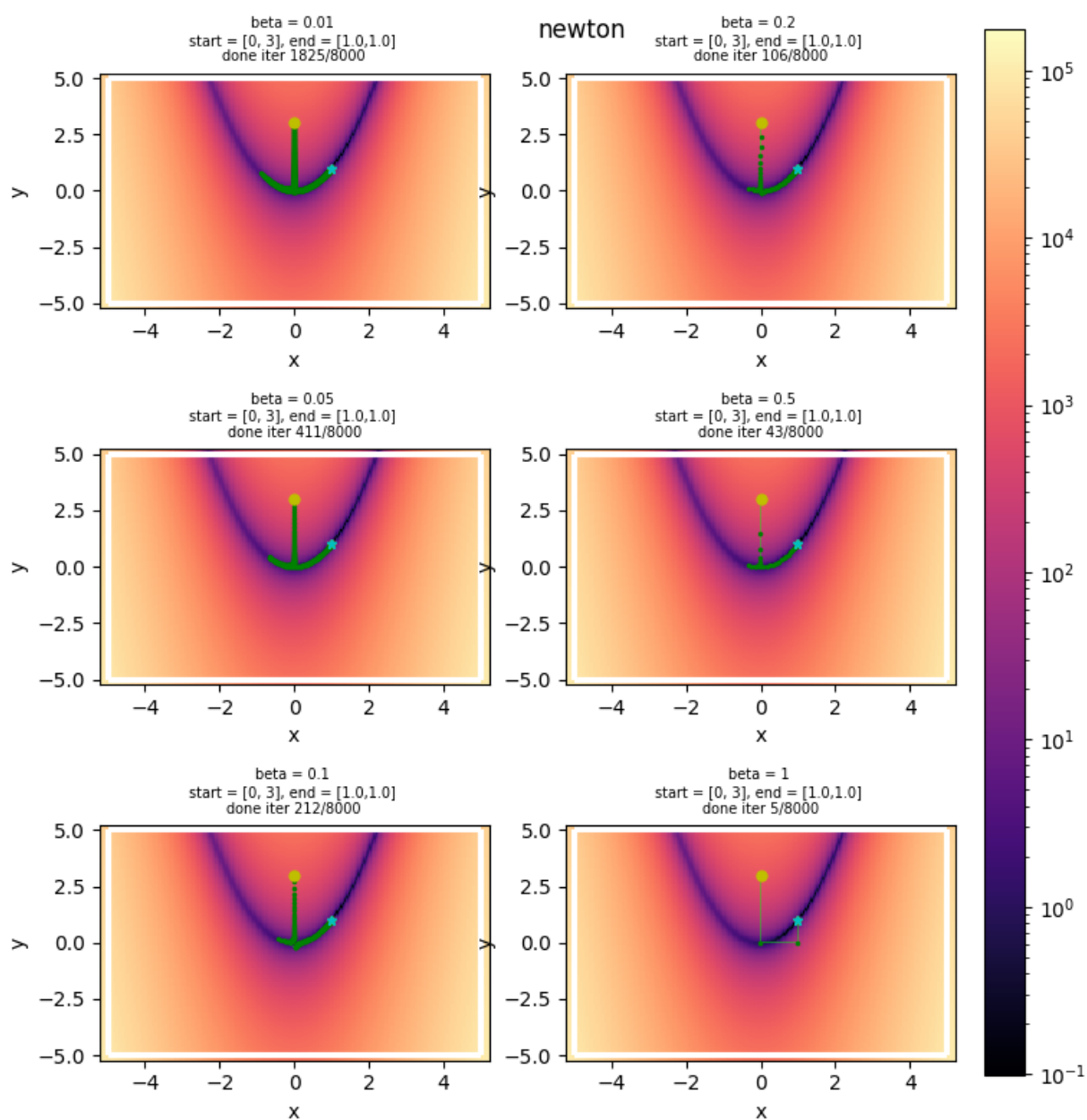
Rysunek 3: Metoda najszybszego spadku gradientu dla kilku różnych punktów początkowych.



Rysunek 4: Metoda najszybszego spadku gradientu dla kilku różnych punktów początkowych, dla większej liczby iteracji.

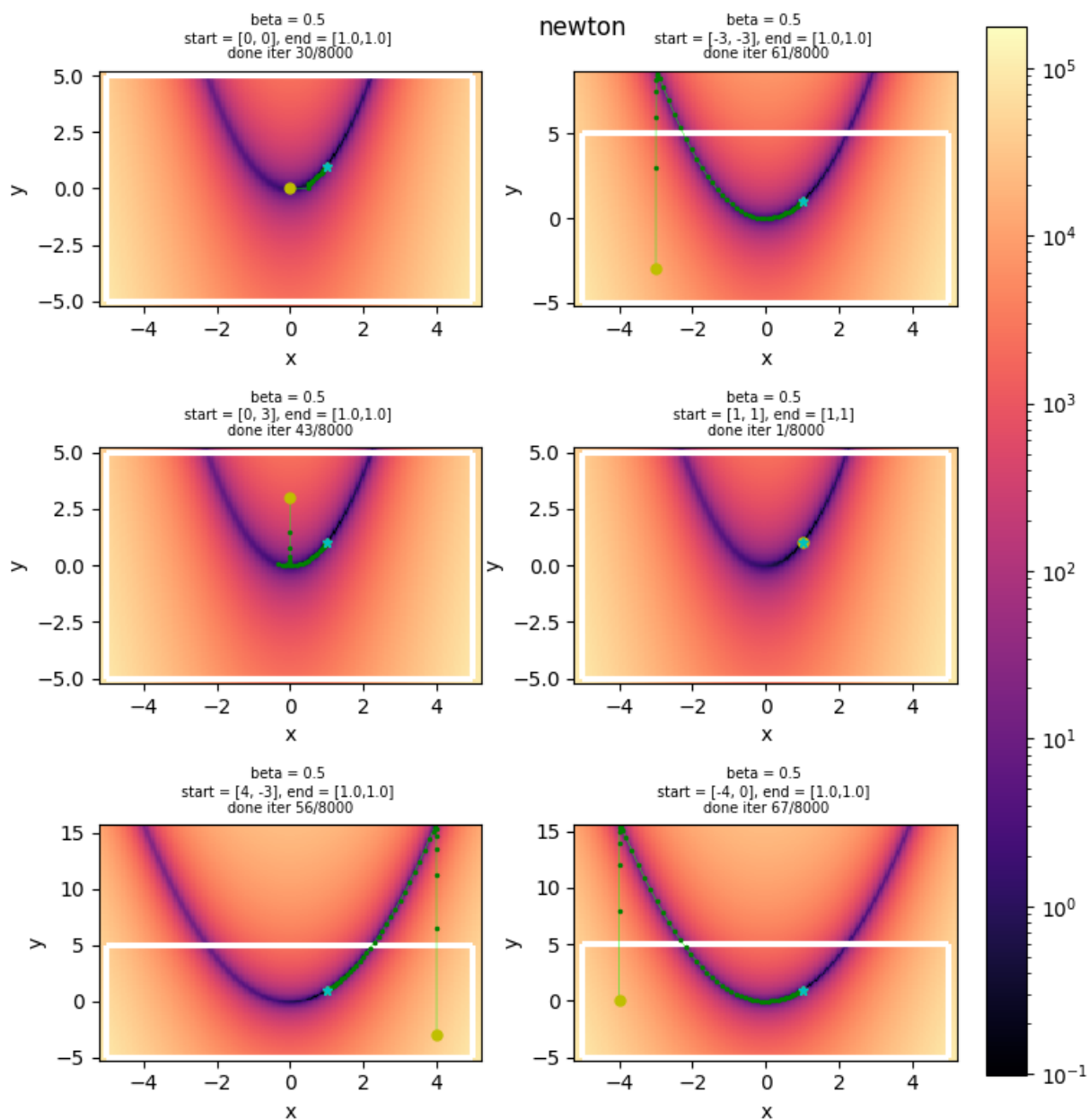
4.2 Metoda Newtona

Rysunek 5 przedstawia kilka wywołań algorytmu Newtona dla kilku różnych wartości parametru β dla danego punktu początkowego.



Rysunek 5: Metoda Newtona dla kilku wartości parametru β

Rysunek 6 przedstawia kilka wywołań metody Newtona dla kilku różnych punktów początkowych.



Rysunek 6: Metoda Newtona dla kilku wartości punktów początkowych

Tabela 1 zawiera parametry opisujące działanie algorytmów dla różnych parametrów początkowych oraz wartości parametrów β . Czas jednej iteracji wyznaczono dzieląc całkowity czas działania algorytmu na liczbę iteracji.

Metoda	Iteracje	Beta	Czas iteracji [ms]	Pkt początkowy	Pkt końcowy
Newton	1825	0,01	2,29	[0, 3]	[1,0; 1,0]
Newton	411	0,05	2,25	[0, 3]	[1,0; 1,0]
Newton	212	0,1	2,31	[0, 3]	[1,0; 1,0]
Newton	106	0,2	2,50	[0, 3]	[1,0; 1,0]
Newton	43	0,5	2,27	[0, 3]	[1,0; 1,0]
Newton	5	1	2,00	[0, 3]	[1,0; 1,0]
Newton	30	0,5	2,19	[0, 0]	[1,0; 1,0]
Newton	43	0,5	2,13	[0, 3]	[1,0; 1,0]
Newton	56	0,5	2,21	[4, -3]	[1,0; 1,0]
Newton	61	0,5	2,92	[-3, -3]	[1,0; 1,0]
Newton	1	0,5	1,22	[1, 1]	[1; 1]
Newton	67	0,5	2,21	[-4, 0]	[1,0; 1,0]
Gradient	10000	0.,00005	1,39	[0, 3]	[0,66; 0,434]
Gradient	10000	0,0001	1,41	[0, 3]	[0,75; 0,561]
Gradient	10000	0,0005	1,40	[0, 3]	[0,96; 0,922]
Gradient	10000	0,001	1,40	[0, 3]	[0,995; 0,989]
Gradient	10000	0,005	1,01	[0, 3]	[0,463; 0,285]
Gradient	10000	0,007	1,39	[0, 3]	[0,366; 0,048]
Gradient	10000	0,00025	1,40	[0, 0]	[0,863; 0,744]
Gradient	10000	0,00025	1,39	[0, 3]	[0,883; 0,779]
Gradient	10000	0,00025	1,41	[4, -3]	[0,866; 0,749]
Gradient	10000	0,00025	1,39	[-3, -3]	[0,862; 0,743]
Gradient	1	0,00025	0,74	[1, 1]	[1; 1]
Gradient	10000	0,00025	1,40	[-4, 0]	[1,022; 1,044]
Gradient	40000	0,00025	1,46	[0, 0]	[0,994; 0,989]
Gradient	40000	0,00025	1,41	[0, 3]	[0,995; 0,99]
Gradient	40000	0,00025	1,41	[4, -3]	[0,995; 0,989]
Gradient	40000	0,00025	1,41	[-3, -3]	[0,994; 0,989]
Gradient	1	0,00025	0,72	[1, 1]	[1, 1]
Gradient	40000	0,00025	1,42	[-4, 0]	[1,001; 1,002]

Tabela 1: Parametry opisujące wykonanie programu dla różnych parametrów wywołania.

5 Wnioski

5.1 Metoda najszybszego spadku gradientu

Metoda gradientu nie znalazła minimum globalnego za każdym razem - założona maksymalna liczba iteracji często nie była wystarczająco duża. Dla punktów w okolicy minimum, funkcja się wypłaszcza, a więc jej gradient ma bardzo małe wartości. Dlatego "kroki" wykonywane przez algorytm są coraz mniejsze i mimo dużej ilości iteracji, algorytm nie osiąga minimum globalnego. Dla bardzo małych wartości beta (np. 0,0001), po założonej maksymalnej ilości iteracji $max_iter = 10\ 000$, algorytm znajdował się stosunkowo daleko (w porównaniu do wywołań z innymi parametrami) od minimum globalnego. Natomiast w przypadku zbyt dużych wartości (np. 0,007), nie był w stanie zbliżyć się do minimum

i "krążył" wokół punktu (patrz: Rysunek 4).

Podczas eksperymentów zauważono, że dla niektórych punktów początkowych, metoda gradientu uciekała po kilku iteracjach do punktów bardzo oddalonych (do nieskończoności w kierunku x oraz y). Wynikało to ze zbyt dużej wartości parametru β dla danego umiejscowienia punktu. Powodowała ona "przeskakiwanie" algorytmu coraz dalej od minimum. Zachowanie podobne do opisywanego widoczne jest na Rysunku 3, dla punktu startowego $[4, -3]$ - gdzie "przeskoki" do nieskończoności zaszłyby, gdyby parametr β był większy.

W wypadku metody spadku gradientu problematyczne okazało się dobranie odpowiedniego parametru β - dla zbyt małych wartości, mimo bardzo dużej ilości iteracji, algorytm nie osiągał minimum. Dla zbyt dużych wartości, pojawiał się opisany wyżej problem. Dodatkowo, w zależności od punktu początkowego, skuteczna okazywała się inna wartość parametru β . Przykładowo, w przypadku punktu $[0,3]$, który jako startowy wybrano dla wywołań algorytmu przedstawionych na Rysunku 2, wartość $\beta = 0,00025$ dla danej założonej maksymalnej ilości iteracji $max_iter = 10000$, jest zbyt mała by osiągnąć minimum, jednak zwiększenie jej do $0,0003$ powodowało, w przypadku niektórych punktów, eksplozję współrzędnych do nieskończoności. Dla $max_iter = 40\,000$, w większości przypadków algorytm zbliżył się znacznie do minimum. Wartość parametru β musi być więc indywidualnie dobierana w zależności od punktu początkowego, a im jest mniejsza, tym większa będzie ilość iteracji konieczna do osiągnięcia punktu docelowego.

5.2 Metoda Newtona

Niezależnie od punktu początkowego oraz parametru β , funkcja znalazła minimum globalne co widoczne jest na rysunku 6 oraz w Tabeli 1, w znacznie mniejszej niż założonej maksymalnej liczbie kroków - aktywowany był warunek stopu związany z epsilon.

Zauważyć można, że wraz ze zwiększającą się wartością parametru β spada liczba kroków potrzebna do osiągnięcia przez algorytm minimum.

5.3 Porównanie

Jako główne różnice między algorytmami wymienić należy:

- czas wykonywania jednej iteracji
- ilość iteracji wymaganych do osiągnięcia celu

Metoda gradientu charakteryzuje się mniejszą ilością obliczeń wykonywanych na jeden krok, gdyż w przypadku metody Newtona, prócz gradientu wyznaczyć należy Hessian - macierz kwadratową o wymiarze n na n . Czas wykonywania jednej iteracji jest w przypadku metody Newtona średnio dwukrotnie większy, niż dla metody spadku gradientu.

Zaletą metody Newtona jest możliwość osiągnięcia celu w znacznie mniejszej ilości kroków - wynika to być może z tego, że wykorzystuje ona informacje o zakrzywieniu funkcji zawartą w Hessjanie, dzięki czemu kroki stawiane są po bardziej "bezpośredniej drodze".

Dodatkowo, w dla analizowanej funkcji celu, w przypadku metody najszybszego spadku gradientu, odpowiedni dobór parametru β był o wiele bardziej problematyczny niż dla metody Newtona, a wyniki znacznie różniły się w zależności od wybranego punktu początkowego.