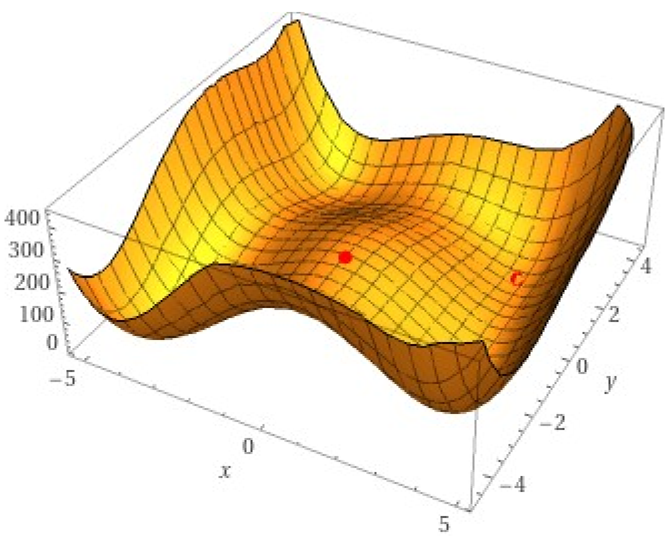
Ćwiczenie pierwsze WSI_23Z
Przeszukiwanie przestrzeni
Dawid Bartosiak 318361

1.Wstęp

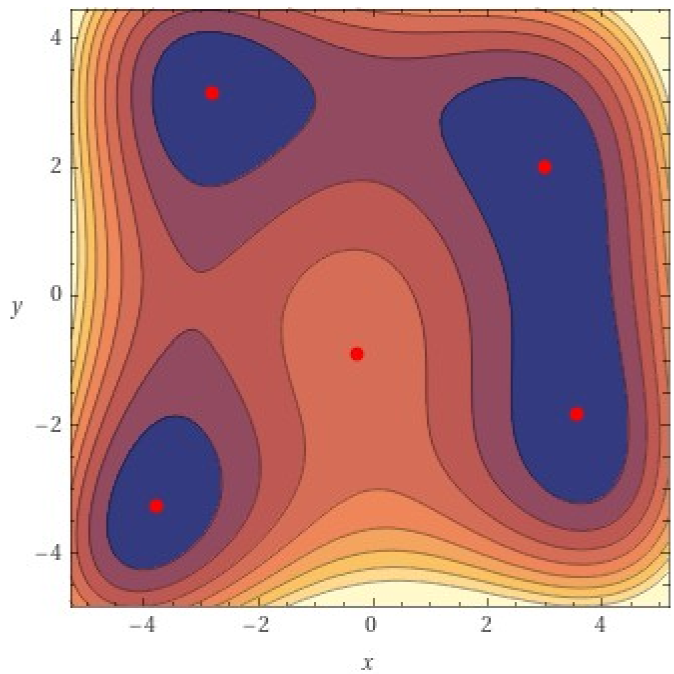
Zadaniem które miałem wykonać była minimalizacja funkcji celu i porównanie wyników dla metody najszybszego spadku gradientu i metody Newtona. Funkcją którą otrzymałem do minimalizacji była funkcja Himmelblau postaci:

$$f(x, y)=(x^2+y-11)^2+(x+y^2-7)^2, -5 \leq x \leq 5, -5 \leq y \leq 5$$

Funkcja ta wygląda następująco:



Rysunek 1: Wizualizacja 3d funkcji celu



Rysunek 2: Wizualizacja 2d funkcji celu

Widać, że mamy punkty które możemy określić jako minima funkcji oraz maxima (Zaznaczone wszystkie na czerwono).

Globalne minima (wartości własne hesjanu dodatnie), a zarazem miejsca zerowe to punkty (x, y): (3, 2), (-3.78, -3.28), (-2.85, 3.13), (3.58, -1.85)
Lokalne maximum (wartości własne hesjanu ujemne) mamy w punkcie (-0.27, -0.92) i wynosi ono 181.62.
Dodatkowo funkcja ma punkty siodłowe (takie których wartości własne hesjanu są różnych znaków): (-3.07, -0.08), (0.09, 2.88), (3.39, 0.07), (-0.13, -1.95),

2. Implementacja

Do implementacji metod użyłem wzorów z prezentacji dostarczonej razem z zadaniem przez Prowadzącego. Następnie wyliczyłem gradient i hesjan w punkcie (x, y):

$$\nabla q(x,y)=[4x^3+4xy-42x-14+2y^2, 2x^2+4xy+4y^3-26y-22]$$

$$H^{-1}(x,y)=\begin{bmatrix}12x^2+4y-42 & 4x+4y \\ 4x+4y & 4x+12y^2-26\end{bmatrix}$$

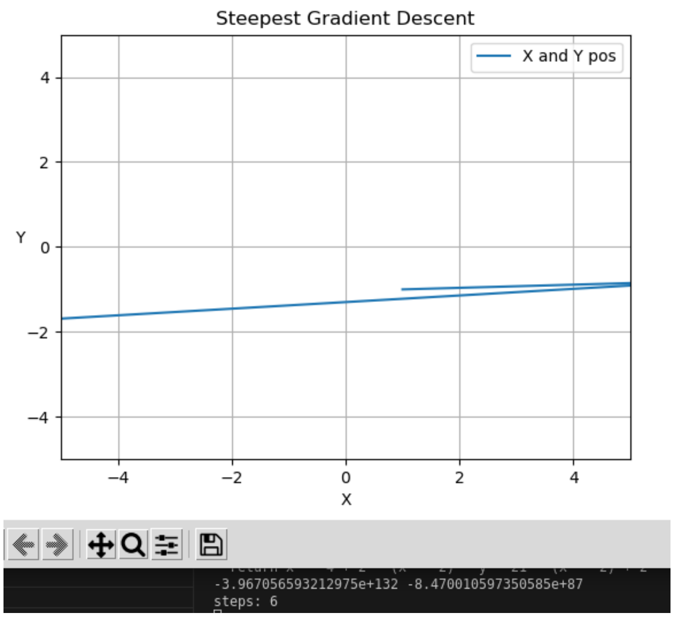
W celu testowania różnych ustawień używam zmiennych takich jak beta, eps, start_pt.

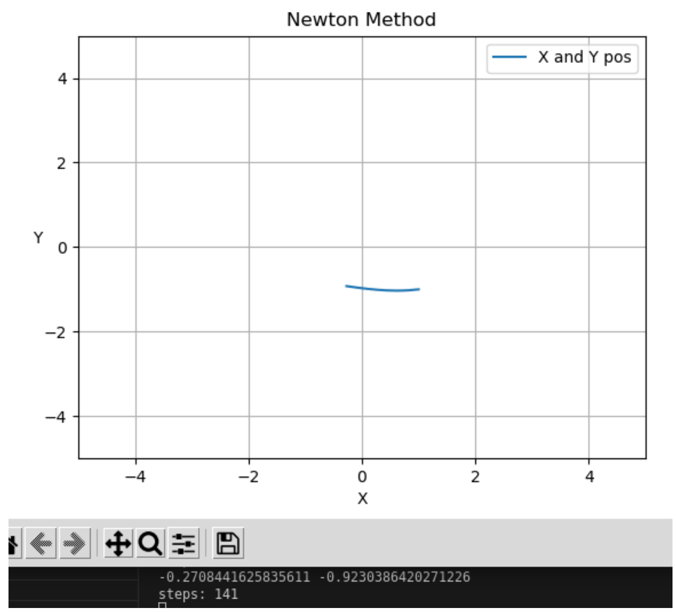
3. Testowanie

Eps który ustawiłem na testy wynosi 10^-12, tak aby nie wykonywać nieokreślenie dużej liczby małych, niepotrzebnych kroków. Równolegle będę testować dla tych samych parametrów metodę spadku gradientu i metodę Newtona.

Dobór bety:

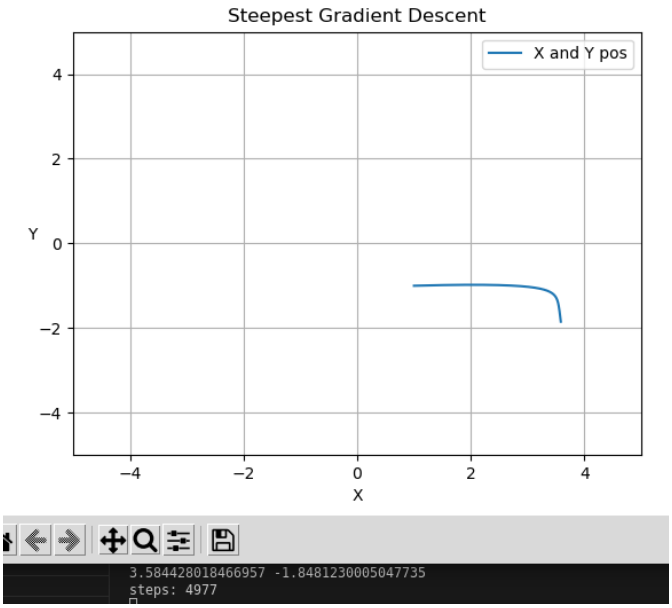
a) Beta 0.1, punkt startowy (1, -1):

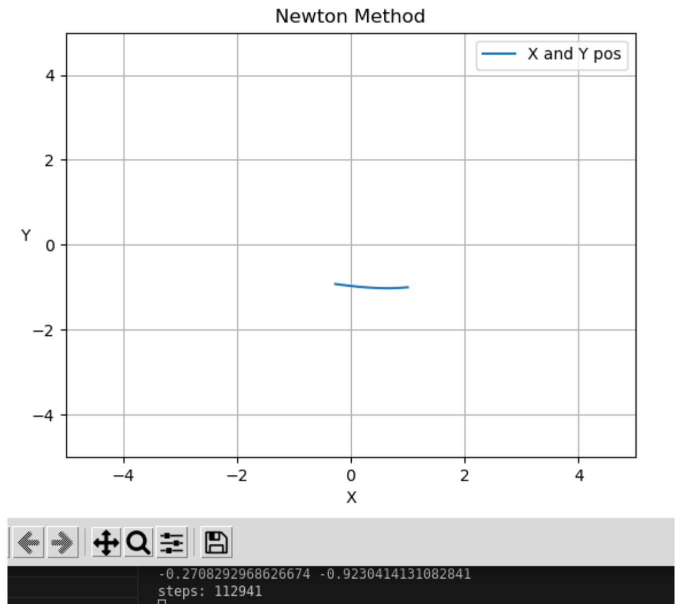




Dla tak dużej bety metoda spadku gradientu nie działa poprawnie, dochodzi w niej do przepełnienia zmiennych (RuntimeWarning: overflow encountered in scalar power). Metoda Newtona za to trafia na maksimum lokalne.

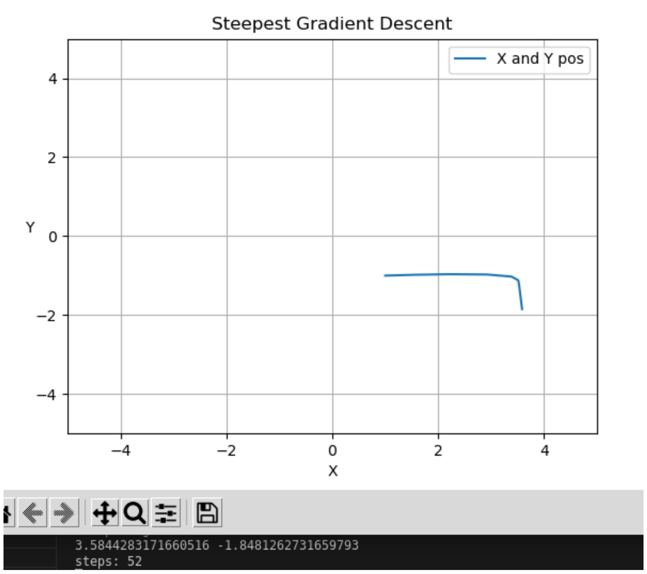
b) Beta 0.0001, punkt startowy (1, -1):

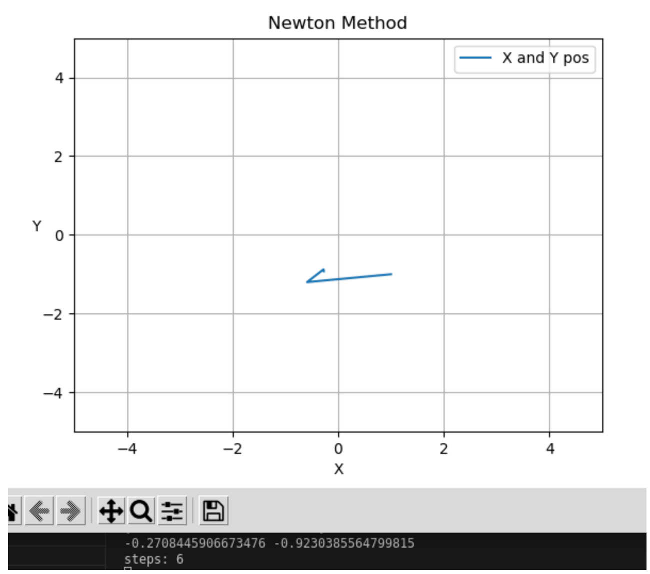




Dla małej bety obydwie metody działają wolno, szczególnie metoda Newtona, nadal jednak trafia ona na maximum lokalne. Metoda spadku gradientu trafia poprawnie na minimum lokalne.

c) Beta 0.01, punkt startowy (1, -1):

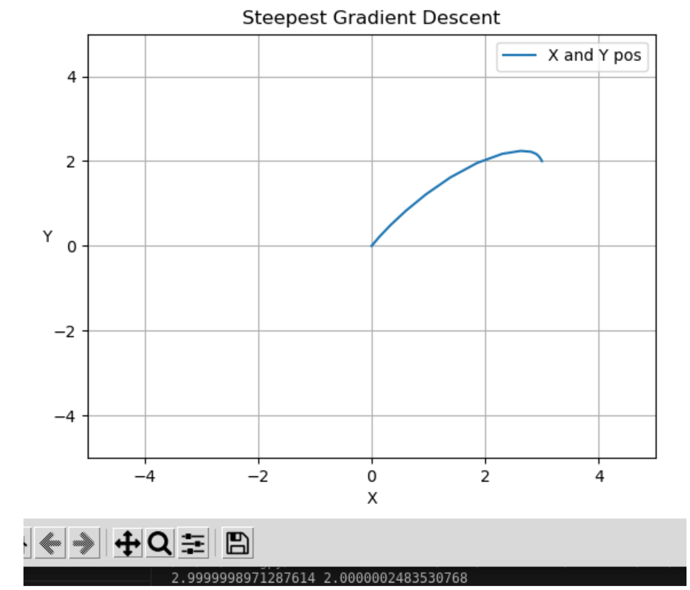


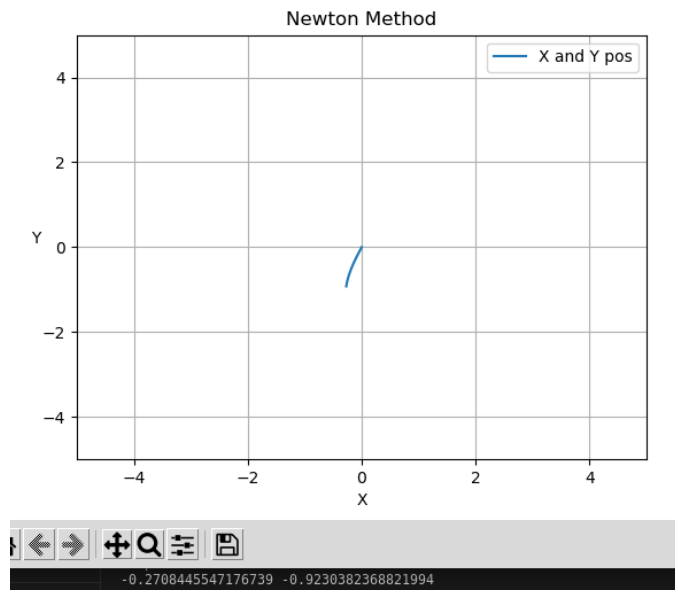


Dodałem mnożnik bety dla metody Newtona –100. Dzieki temu metoda Newtona działa znacznie szybciej a metoda spadku gradientu jest dalej stabilna. Z taką betą będę przeprowadzać dalsze eksperymenty (na różnych punktach)

Testy różnych punktów

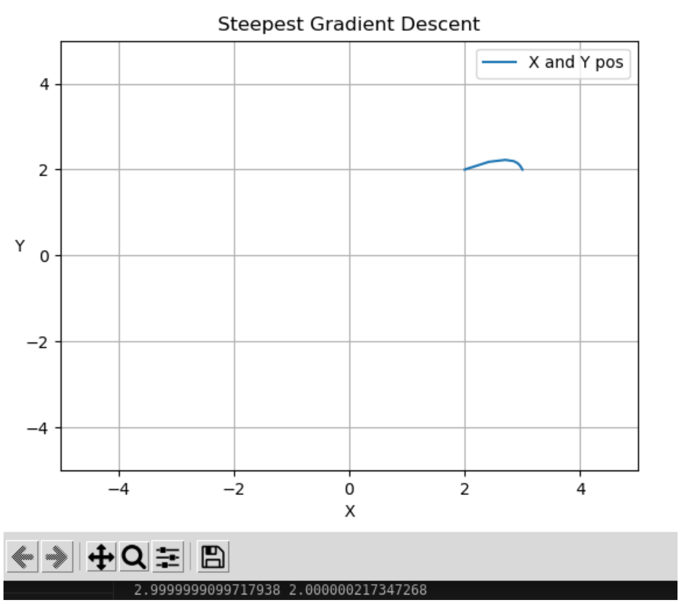
a) Punkt startowy (0, 0):

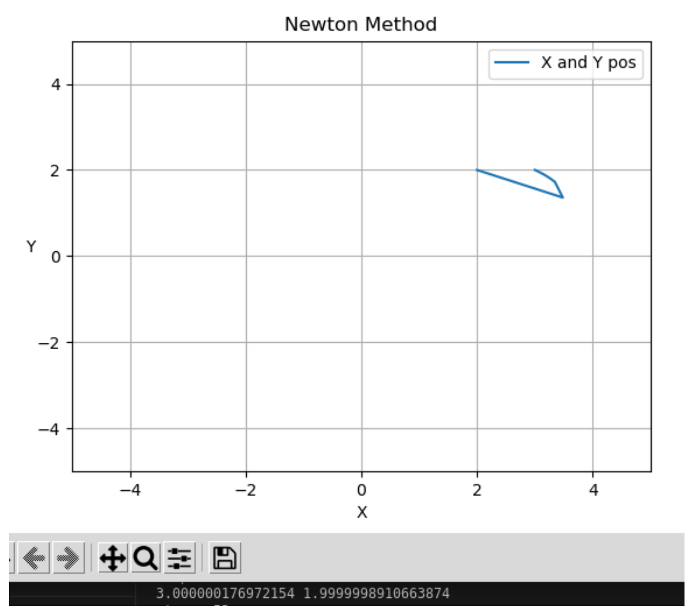




Metoda spadku gradientu działa poprawnie (dociera do minimum), a metoda Newtona trafia na punkt siodłowy.

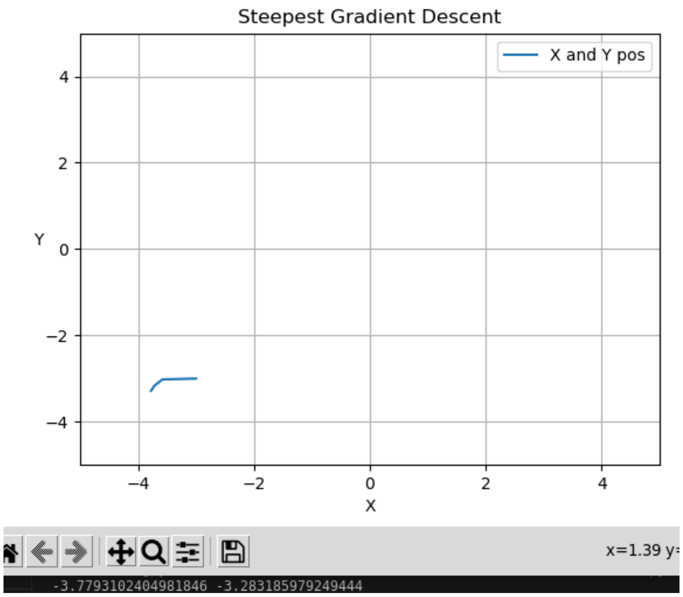
b) Punkt startowy (2, 2):

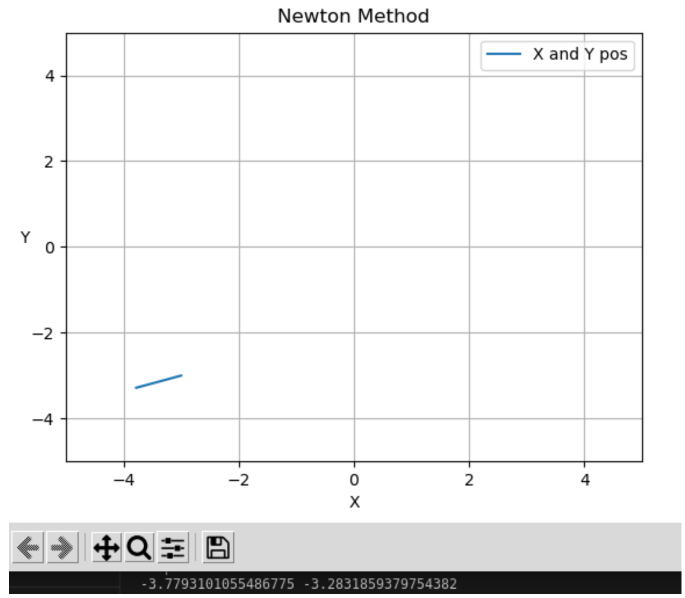




Jest to pierwszy punkt który działa poprawnie, obydwie metody docierają do minimum lokalnego w niedużej ilości kroków, jednak w metodzie Newtona wymagało to zmniejszenia mnożnika bety do 25.

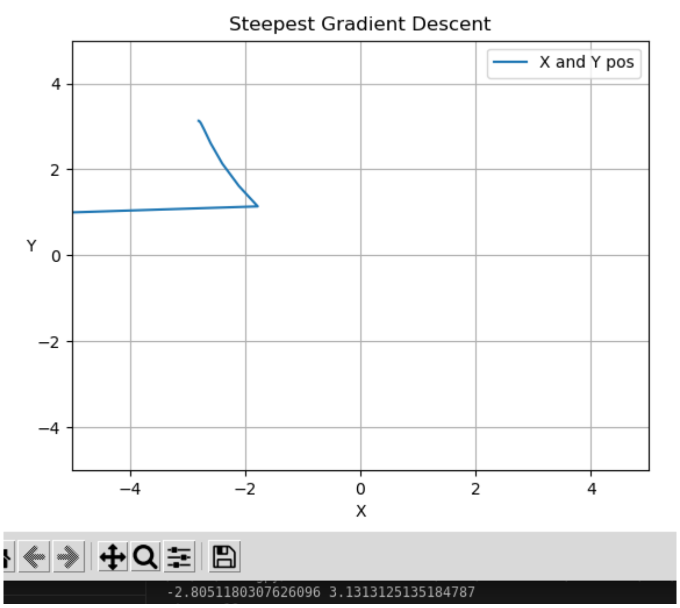
c) Punkt startowy (-3, -3):

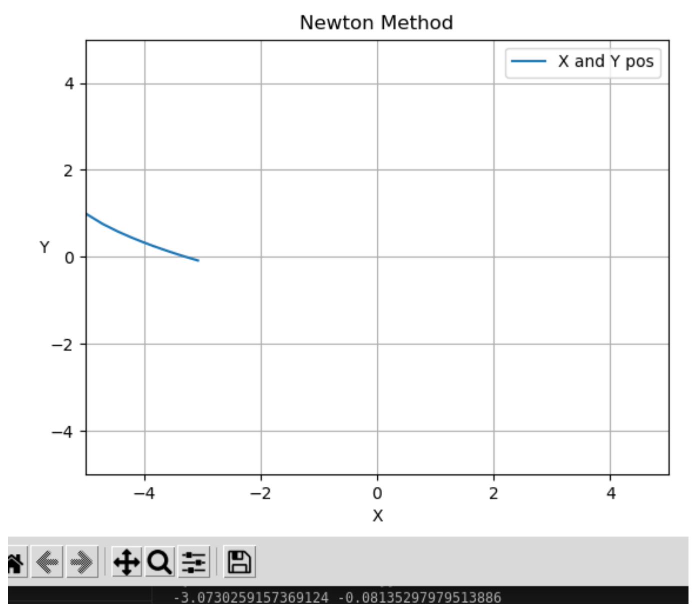




Ponownie poprawny punkt.

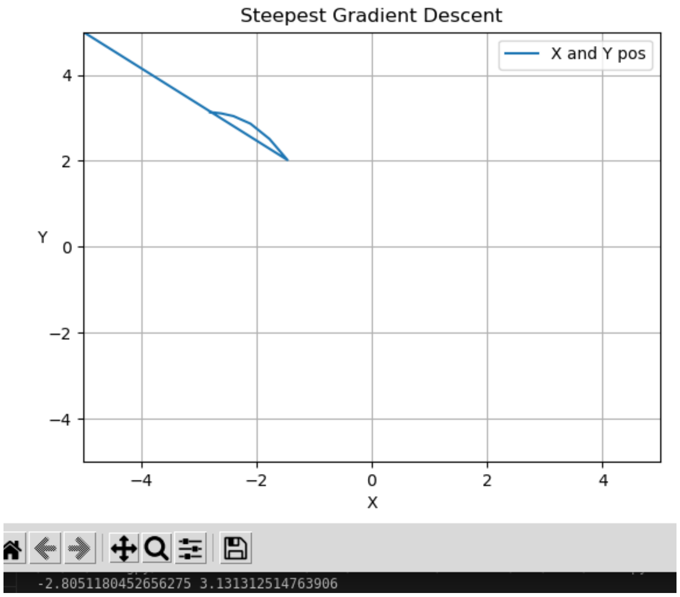
d) Punkt startowy (-5, 1):

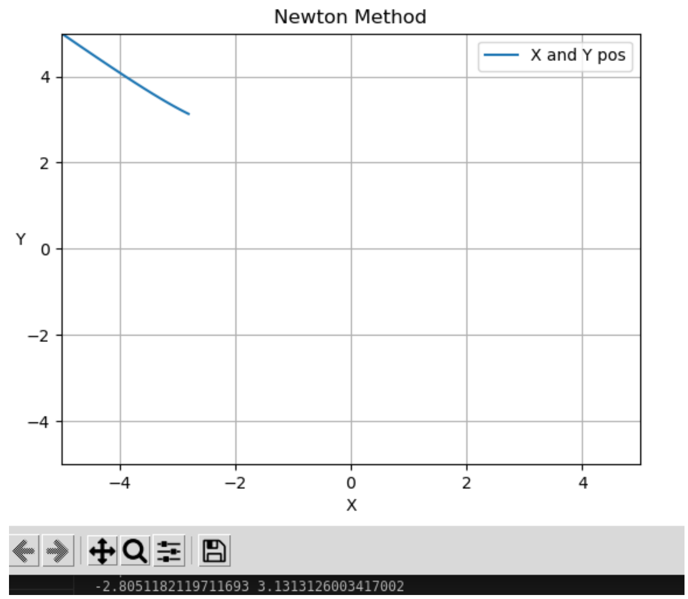




Metoda spadku gradientu ponownie działa poprawnie a metoda newtona trafia na punkt siodłowy (ponownie hesjan ma różne znaki).

e) Punkt startowy (-5, 5):





Kolejny z punktów który działa poprawnie. Na metodzie spadku gradientu widać dziwną trajektorię, która nie pojawia się w metodzie Newtona.

Czasy:
Porównania czasowe algorytmów wykonuję z pewnym założeniem – obydwie metody muszą zbiegać do tego samego punktu. Zatem tabela prezentuje się następująco:

Metoda	**Punkt Startowy**	**Czas (s)**	**Ilość kroków**
Newtona	(-5,-5)	0.000256538391113	6
spadku gradientu	(-5,-5)	0.000151157379150	16
Newtona	(5,-5)	0.000491380691528	8
spadku gradientu	(5,-5)	0.000225305557250	55
Newtonaz	(-5,5)	0.000469923019409	6
spadku gradientu	(-5,5)	0.000101804733276	20
Newtona	(5,5)	0.000607490539551	7
spadku gradientu	(5,5)	0.000215291976929	60
Newtona	(-3, -3)	0.000275135040283	6
spadku gradientu	(-3, -3)	0.000142097473145	15
Newtona (β=0.35)	(2, 2)	0.002579927444458	40
spadku gradientu	(2, 2)	0.000201463699341	49

Patrząc na powyższą tabelę wnioski nasuwają się same. Metoda Newtona wykonuje mniej operacji gdyż możemy sobie w niej pozwolić na wyższą betę, jednak mimo to trwa ona dłużej z uwagi na znacznie bardziej złożone obliczenia (mnożenie macierzy przez wektor, więcej liczenia pochodnych przy każdej iteracji, co i tak zostało uproszczone już do wykonywania prostych operacji jak mnożenie, dodawanie czy odejmowanie z uwagi na obliczenie pochodnych wcześniej i przepisanie ich do programu).

4. Wnioski

Na skutek przeprowadzonych eksperymentów udało mi się dostać optymalną wartość bety. W testach różnych punktów udało się znaleźć kilka punktów siodłowych na których zatrzymuje się metoda Newtona oraz miejsce które jest maksimum lokalnym. Dzieje się tak gdyż metoda Newtona dąży do najbliższego punktu o zerowym gradiencie, a punkt siodłowy i maksimum lokalne weryfikowałem na podstawie znaków hesjanu. W przypadku funkcji Himmelblau metoda spadku gradientu jest optymalna pod względem docierania do minimum oraz czasu działania, zaś metoda Newtona działa dla niej znacznie gorzej – wolniej oraz zatrzymuje się na punktach siodłowych, jednak wykonuje mniej iteracji. Zawieranie w sprawozdaniu kolejnych wykresów nie ma większego sensu, gdyż nie zaobserwujemy już nic nowego poza pokazywaniem kolejnych punktów które uprzednio wypisałem na wstępie. Modyfikacja dodatkowego warunku stopu jakim jest maksymalna liczba iteracji wydaje się zbędna z uwagi na pokazanie przeze mnie ilości kroków wykonywanych w poszczególnych przypadkach, więc z góry oszacować możemy, że obydwie metody w przypadku obrania wartości maksymalnej ilości iteracji mniejszej niż rzeczywista, zatrzymają się nie docierając do właściwego dla nich miejsca.