Ćwiczenie nr 1
Wprowadzenie do sztucznej inteligencji

Autor
Maciej Prostak

1 Wprowadzenie

Zadanie polega na minimalizacji funkcji celu za pomocą dwóch metod: spadek gradientu oraz metoda Newtona. Funkcja, na której metody będą wyliczane to Himmelblau. Następnie dla podanych przykładowych punktów zostaną przedstawione wykresy oraz działanie obydwu metod. Zostaną one ze sobą porównane pod względem prędkości działania, kroków oraz dokładnością.

1.1 Technologia

Rozwiązanie zostało zaimplementowane w języku Python (3.8.0) z wykorzystaniem bibliotek matplotlib (3.4.3) oraz numpy (1.21.2). Dodatkowo dla poprawy czytelności kodu zostały dołączone biblioteki black (21.9b0), flake8 (4.0.1) oraz isort (5.9.3).

1.2 Funkcja Himmelblau

Funkcja Himmelblau opisana wzorem (1).

$$f(x,y) = (x*x+y-11)^2 + (x+y*y-7)^2, -5 <= x,y <= 5 \tag{1}$$

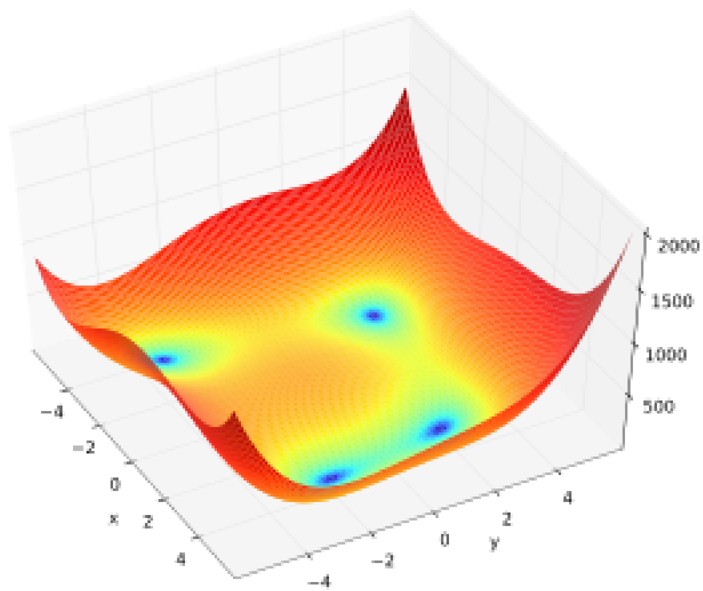

Fig. 1 Wizualizacja funkcji Himmelblau

W celu liczenia gradientu wyliczone zostały pochodne funkcji pierwszego stopnia opisane wzorami (2, 3).

$$\frac{\partial f}{\partial x}(x,y) = 4x^3 + 4xy - 42x + 2y^2 - 14 \tag{2}$$

$$\frac{\partial f}{\partial y}(x,y) = 2y + 2x^2 - 22 + 4y^3 + 4xy - 28y \tag{3}$$

Zostały również policzone pochodne drugiego stopnia do policzenia hesjanu, które opisano wzorami (4, 5, 6, 7).

$$\frac{\partial^2 f}{\partial x^2}(x,y) = 12x^2 + 4y - 42 \tag{4}$$

$$\frac{\partial^2 f}{\partial y^2}(x,y) = 12y^2 + 4x - 26 \tag{5}$$

$$\frac{\partial^2 f}{\partial x \partial y}(x,y) = 4x + 4y \tag{6}$$

$$\frac{\partial^2 f}{\partial y \partial x}(x,y) = 4x + 4y \tag{7}$$

2 Wyniki

2.1 Spadek gradientu

Na zamieszczonym rysunku (Fig. 2) widać dokładne działanie metody spadku gradientu przy różnych wartościach beta. Ustalona maksymalna liczba iteracji została ustawiona na 5000 i punkt startowy jako [0, 0]. Dla bety równej 0.0001 kolejno wyznaczane punkty zbliżają się do lokalnego minimum, lecz w niedużych krokach. Wartość Iter podana na rysunku oznacza wykonaną liczbę iteracji w algorytmie. W podanym przykładzie warunek stopu epsilon nie został uruchomiony.

Zwiększenie parametru beta w kolejnych testach, na ogół pozwalają na skrócenie liczby iteracji (przy tym czasu wykonania całego algorytmu), z wyjątkiem gdy wartość będzie za duża i kolejno wyznaczone punkty nigdy nie przybliżą nas do podążanego punktu, a jedynie "krążyć" wokół niego. Na wykresie z parametrem beta równym 0.03 można zauważyć, że liczba iteracji osiągnęła założone maksimum.

2.2 Metoda Netwona

Działanie metody Newtona w punkcie [3,4] dla różnych wartości parametru beta zostało zwizualizowane na rysunku (Fig. 3). Można zauważyć pewne podobieństwo zachowania tego algorytmu do opisanego w podrozdziale 2.1, jakim jest zmniejszenie ilości iteracji przy odpowiednim zwiększeniu wartości beta. Przy odpowiednio dużej wartości beta również widzimy, że warunek stopu epsilon nigdy nie zostaje uruchomiony.

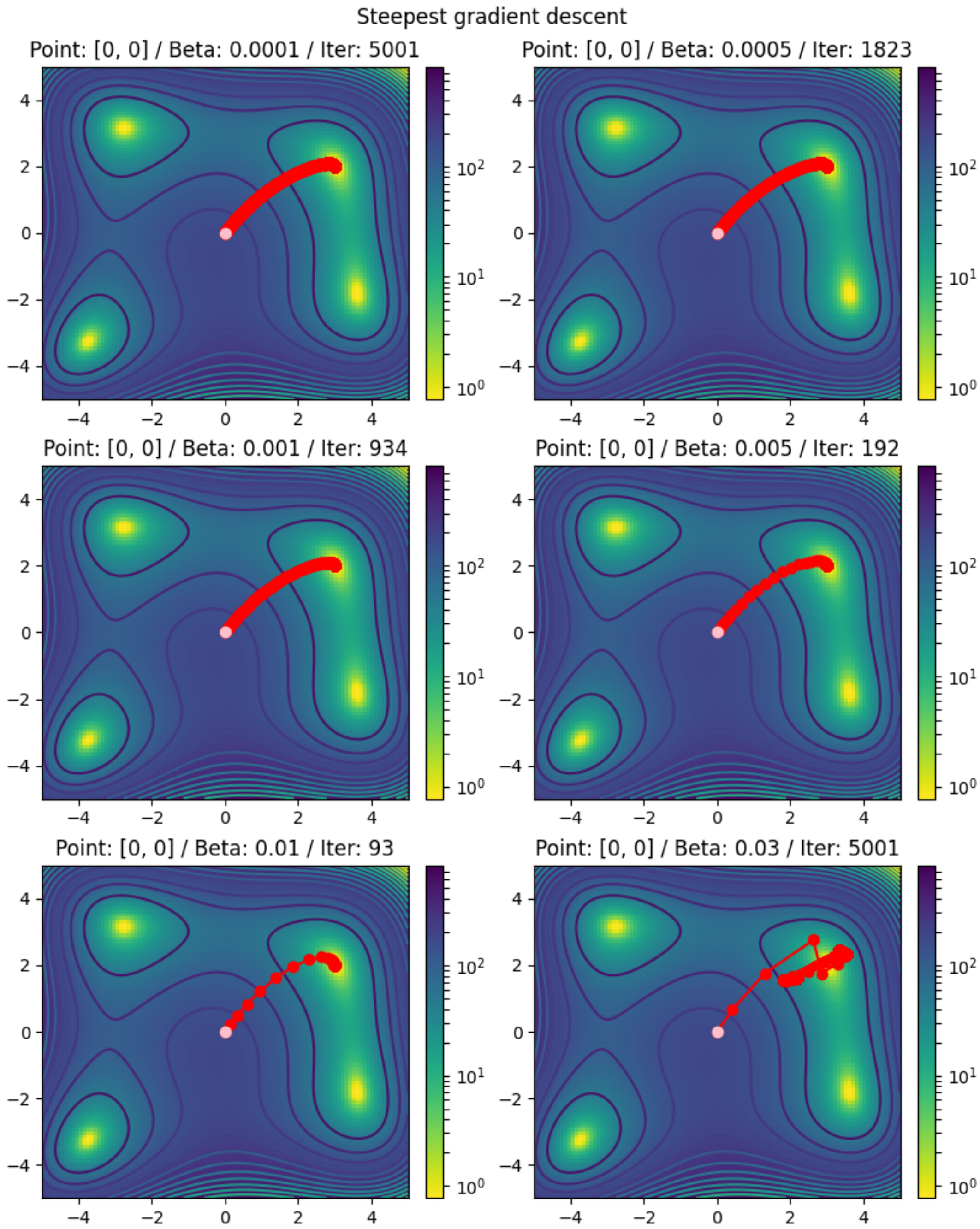

Fig. 2 Wyniki spadku gradientu dla różnych wartości beta

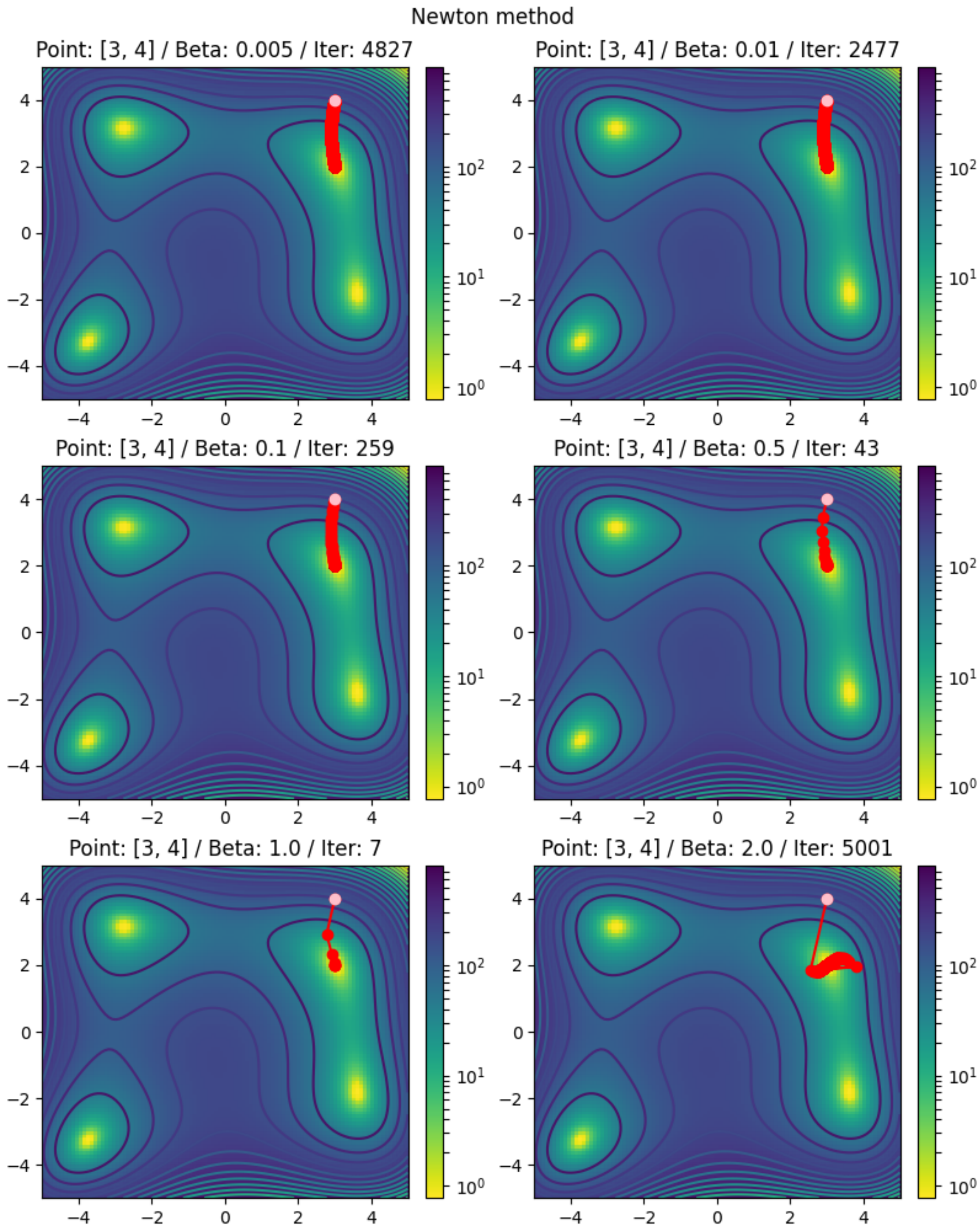

Fig. 3 Wyniki metody Newtona dla różnych wartości beta

2.3 Testy w różnych punktach

Punkty, z których będą uruchamiane metody to:

$$\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}4\\4\end{bmatrix}, \begin{bmatrix}-3\\2\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4\\0\end{bmatrix} \tag{8}$$

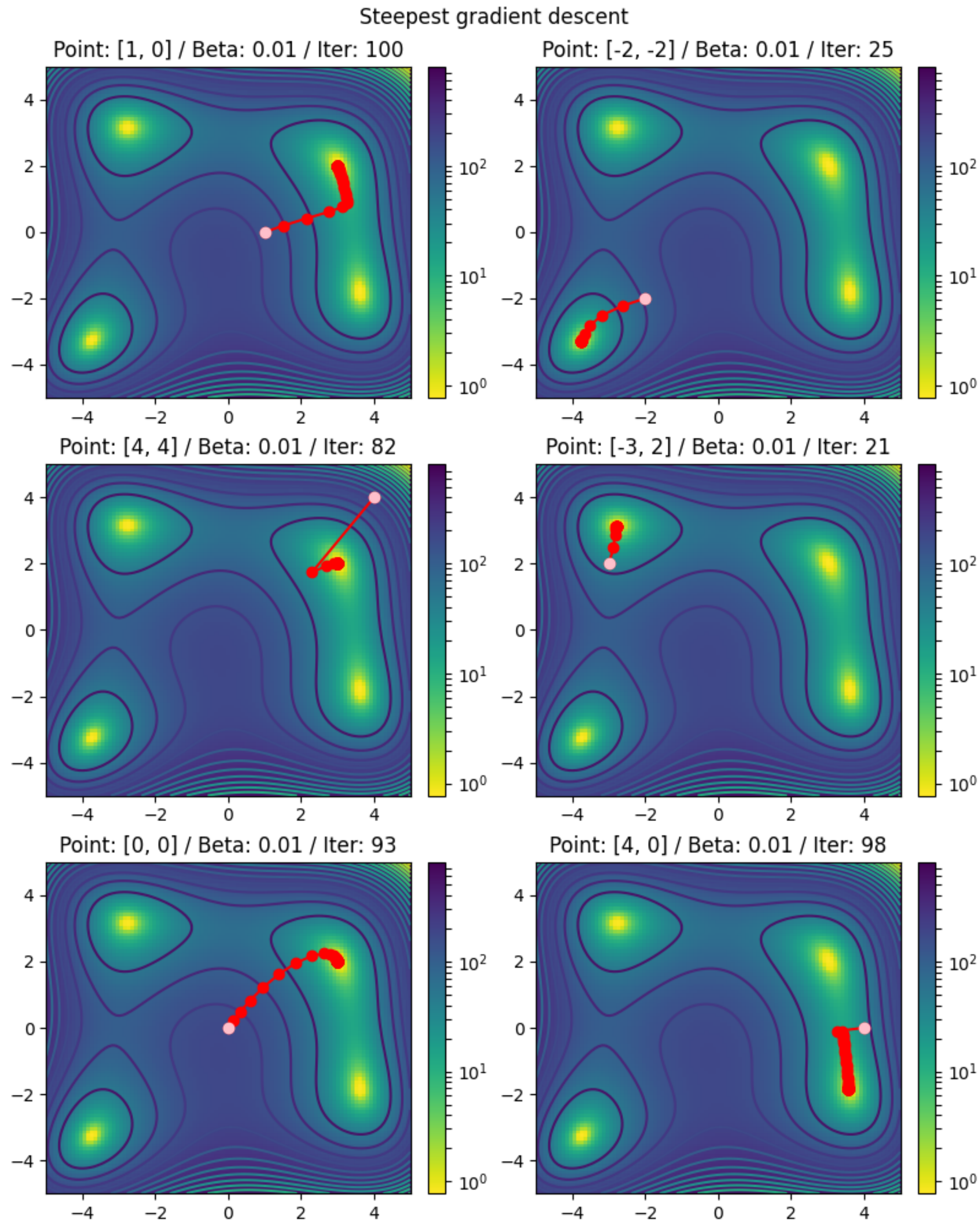

Fig. 4 Wyniki metody spadku gradientu dla różnych punktów

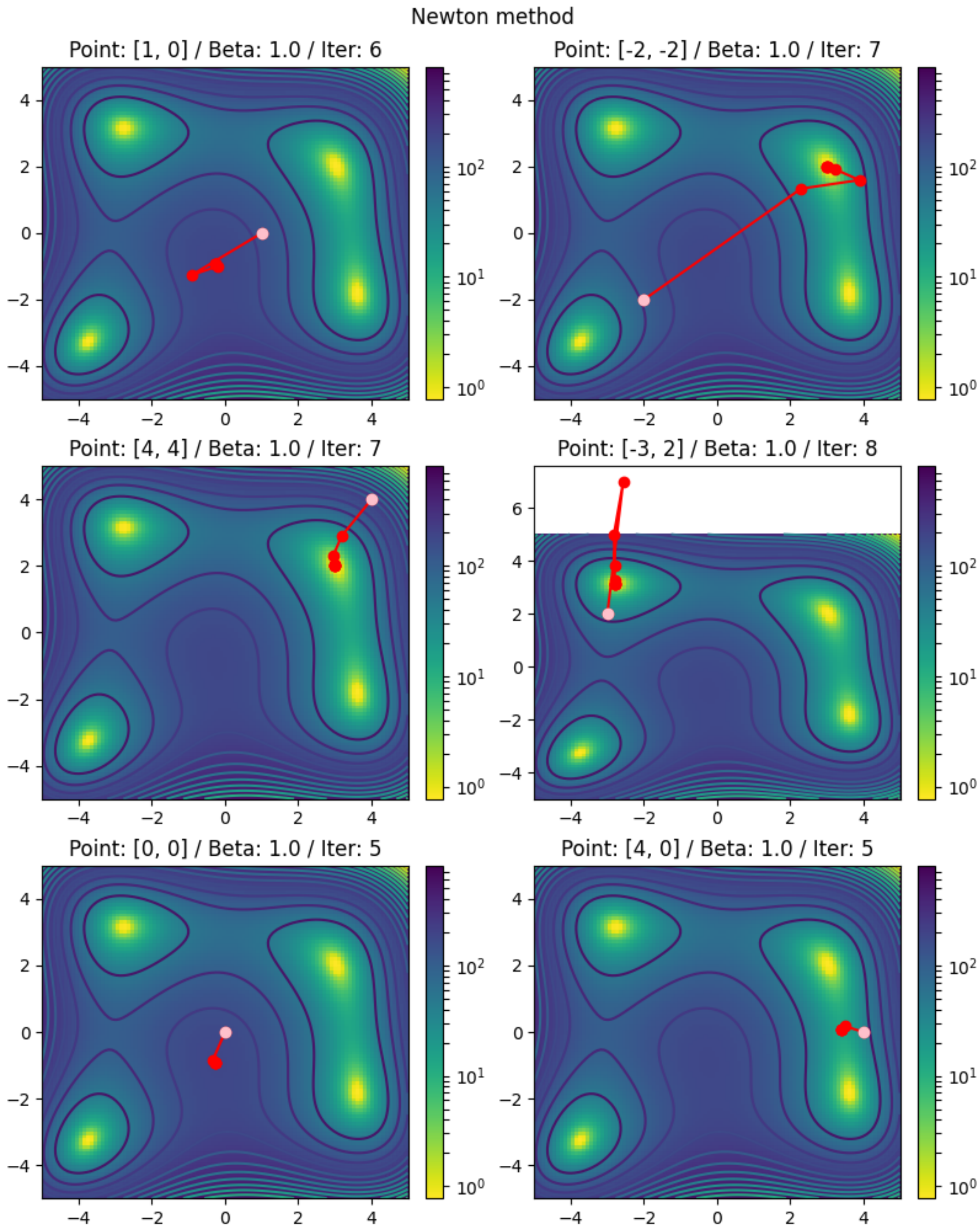

Fig. 5 Wyniki metody Newtona dla różnych punktów

W metodzie spadku gradientu można zauważyć, że algorytm zawsze znajduje minima funkcji, lecz w metodzie Newtona punkty zatrzymują się w pewnych punktach różniących się od minimum. Po zbadaniu znalezionych punktów okazało się, że pochodna wszystkich zmiennych funkcji w tych punktach jest zbliżona do zera. Są to punkty siodłowe (ang. Saddle Points), na które metoda Newtona jest podatna.

2.4 Czasy wykonania

Dla spadku gradientu mediana czasu wykonania na podanych wcześniej wynikach jednej iteracji to 5 mikrosekund, jednak dla metody Newtona wynosi ona 27 mikrosekund. Są to jednak bardzo małe odstępy czasowe, więc należy te wyniki jednak traktować z pewnym dystansem. Sugeruje to jednak, że na powyższych przeprowadzonych próbach metoda gradientu jest ponad pięcokrotnie szybsza.

W podanych poniżej tabelach możemy zauważyć, że udało się uzyskać podobne czasy wykonania obydwu metod dla beta = 0.01 w przypadku spadku gradientu oraz dla beta = 1.0 dla metody Newtona. Mimo, że liczba iteracji jest znacznie mniejsza, to czas kalkulacji każdego kroku jest o wiele większa.

Punkt	Beta	Liczba Iteracji	Czas w mikrosekundach
(1, 0)	0.01	100	747
(-2, -2)	0.01	25	277
(4, 4)	0.01	82	623
(-3, 2)	0.01	21	260
(0, 0)	0.01	93	671
(4, 0)	0.01	98	732

Tabela 1 Wyniki czasowe dla spadku gradientu.

Punkt	Beta	Liczba Iteracji	Czas w mikrosekundach
(1, 0)	1.0	6	742
(-2, -2)	1.0	7	549
(4, 4)	1.0	7	573
(-3, 2)	1.0	8	619
(0, 0)	1.0	5	469
(4, 0)	1.0	5	476

Tabela 2 Wyniki czasowe dla metody Newtona.

3 Wnioski

Metoda spadku gradientu wykazała się znacznie lepszymi wynikami niż metoda Newtona. Po pierwsze czas obliczeń na każdą iterację są znacznie mniejsze i obliczenia matematyczne są uproszczone. Metoda spadku gradientu nie jest również podatna na punkty siodłowe, które można zauważyć przy metodzie Newtona.