Zadanie 1 Gradient Prosty

Mikołaj Szawerda 318731

Opis zaimplementowanego algorytmu

Celem zadania było zaimplementowanie algorytmu gradientu prostego - służącego do (sub)optymalizowania zadanych funkcji matematycznych wielu zmiennych. Algorytm polega na obliczaniu wartości gradientu w zadanym punkcie - a więc kierunku wzrostu wartości funkcji - i "przejściu" do kolejnego punktu w kierunku przeciwnym do gradientu o wartość iloczynu gradientu i parametru kroku. Kluczowym jest więc odpowiednie dobranie kroku. W mojej implementacji znajdują się dwa rozwiązania - z góry ustalony krok losowany z zadanego przedziału(potencjalnie można uruchomić program dla paru wartości z przedziału) i krok dynamiczny, zmieniający się w każdej iteracji, wyszukiwany przez - "Backtrack line search". Ponieważ wybór punktu ma również ogromne znaczenie na wyniki gradientu prostego, program można uruchomić dla wielu punktów, losowanych z podanej dziedziny. Jako warunek stopu przyjąłem maksymalną ilość iteracji(n=1000), lub brak wystarczająco dużej poprawy/wartość gradientu, lub rozbieżność funkcji.

Backtrack line search

Polega na iteracyjnym zmniejszaniu potencjalnego kroku w danej iteracji, aż do osiągnięcia zadowalającej optymalizacji z bieżącego miejsca.

Planowane eksperymenty numeryczne

- Uruchomienie obu algorytmów dla zadanego punktu/kroku
- Uruchomienie obu algorytmów dla wielu kroków i punktów
- Uruchomienie obu algorytmów dla funkcji dwóch zmiennych w losowym punkcie i na "płaskim terenie"

W każdym eksperymencie program dokonuje zapisu obecnego obliczanego punktu, wartości funkcji w punkcie i iteracji, po za tym program mierzy czas wykonania.

Dwa pierwsze eksperymenty zostały uruchomione na proponowanej funkcji:

$q(x) = \sum_{i=1}^{n} \alpha^{\frac{i-1}{n-1}} x_i^2, x \in [-100, 100]^n \subset \mathbb{R}^n, n = 10$

Trzeci dla:

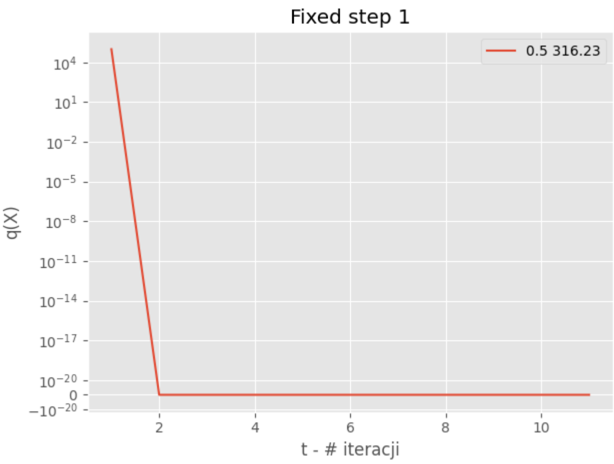
$q(x, y) = (1 - x^2 + y^3)e^{-(x^2+y^2)}$

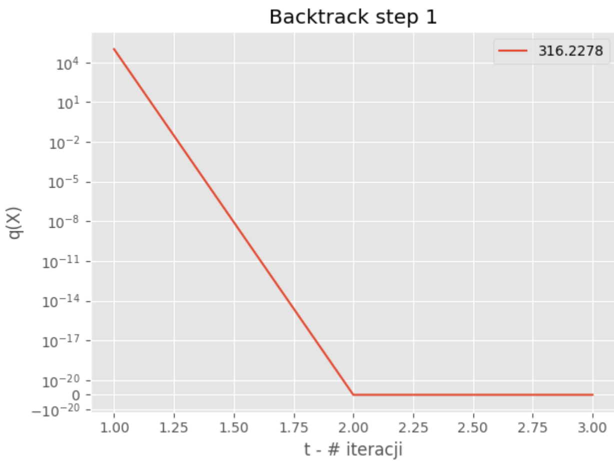
Wyniki

Wywołanie dla zadanego kroku i punktu

Etykieta w przypadku Fixed step stanowi przyjęty krok i odległość punktu startowego od Punktu X=0 - odpowiednio dla Backtrack step

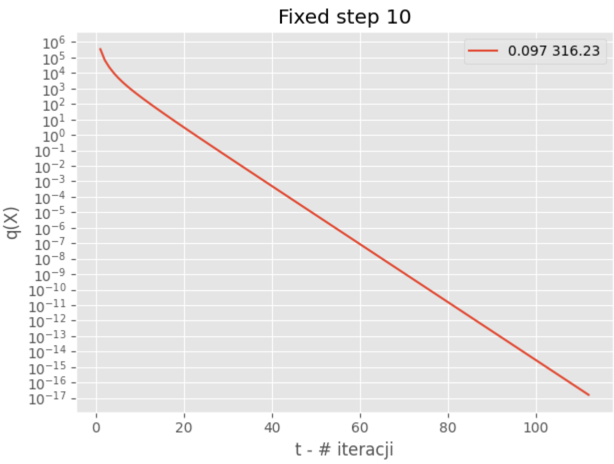
α=1

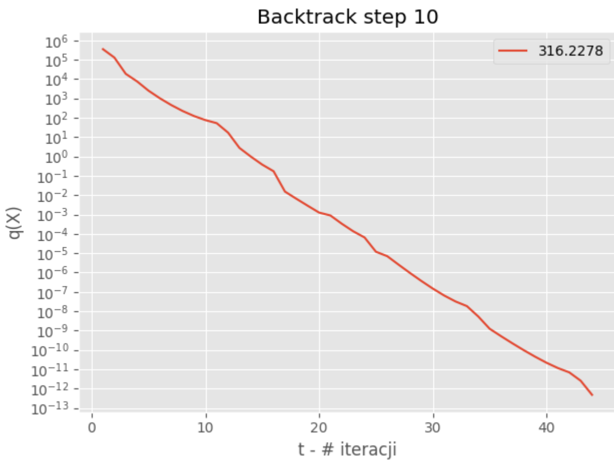


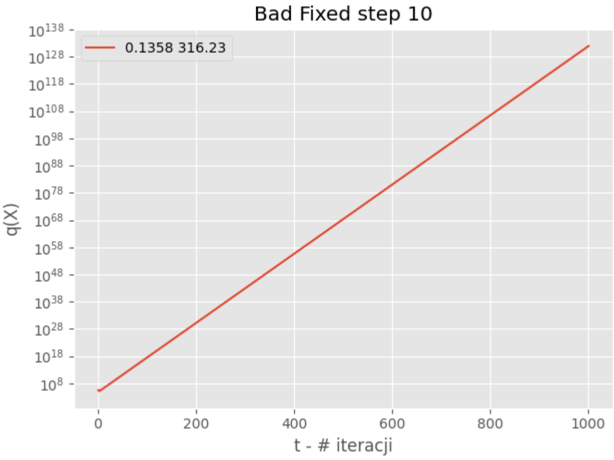



```
----------
Experiment:Fixed step 0.5 316.23
Result: 0.0
Execution time: 0.01041413500206545
----------
Experiment:Backtrack step 316.22776601683
Result: 0.0
Execution time: 0.007631012998899678
```

α=10

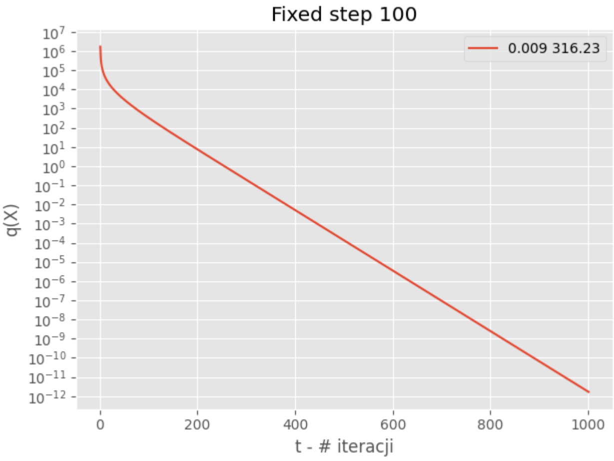


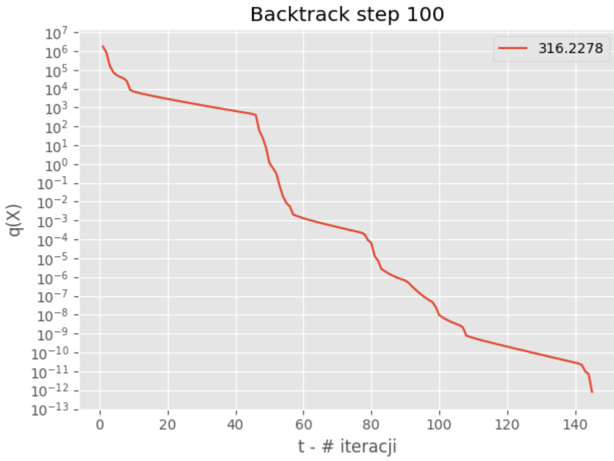





```
----------
Experiment:Fixed step 0.097 316.23
Result: 1.608346515453692e-17
Execution time: 0.13126453799850424
----------
Experiment:Backtrack step 316.22776601683
Result: 4.793633982835726e-13
Execution time: 0.23740183299742057
```

α=100

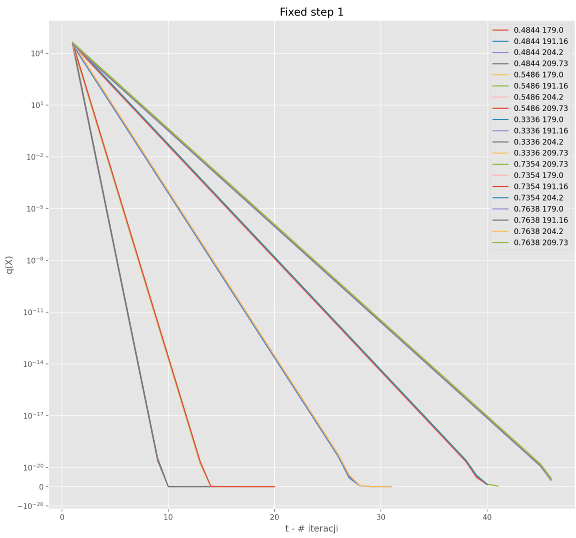


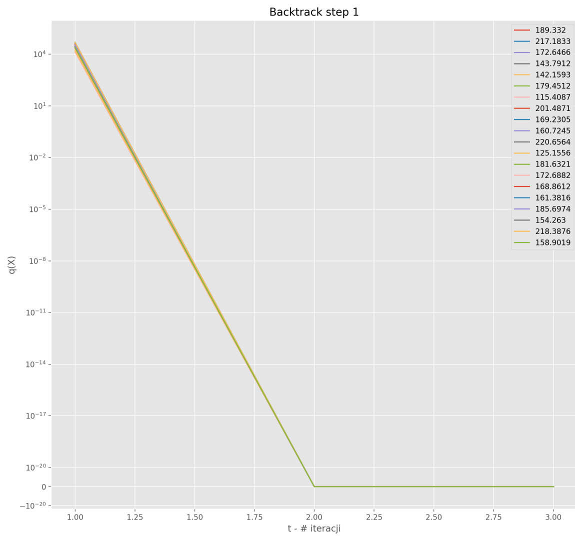



```
----------
Experiment:Fixed step 0.009 316.23
Result: 1.6709966306609213e-12
Execution time: 1.0560820499995316
----------
Experiment:Backtrack step 316.22776601683
Result: 8.036478306381011e-13
Execution time: 1.1040850710014638
```

Wywołanie z wieloma punktami i krokami

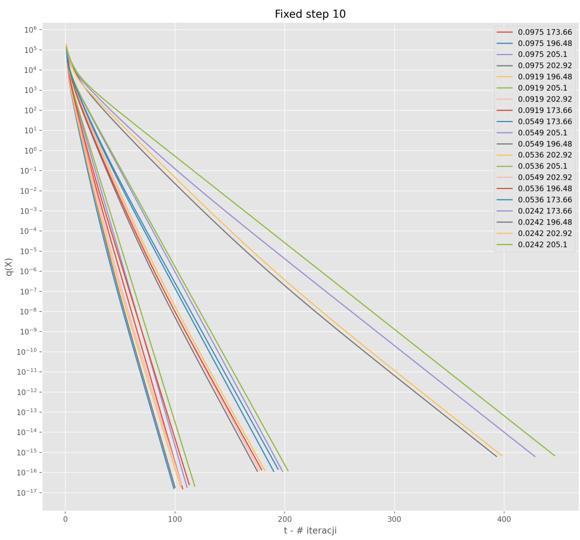
α=1

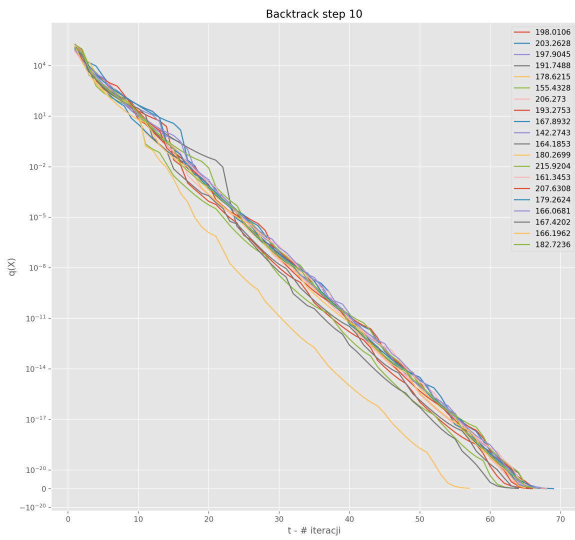





```
----------
Experiment: Random fixed step
Result: 1.9944734169218877e-44
Execution time: 0.25882955999986734
----------
Experiment: Backtrack dynamic step
Result: 0.0
Execution time: 0.0845379429993045
```

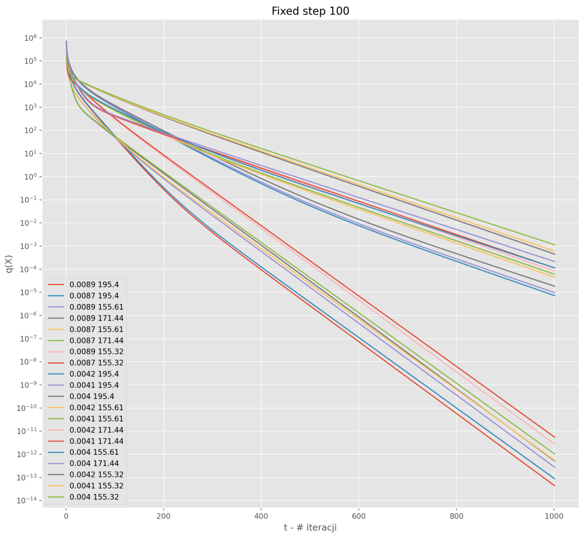
α=10

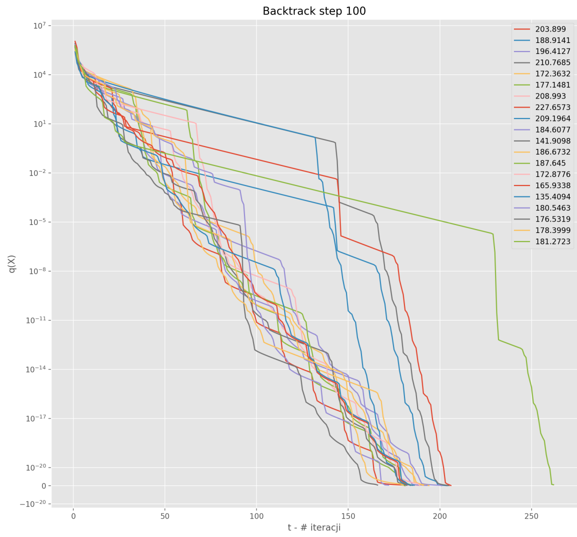





```
----------
Experiment: Random fixed step
Result: 1.5778196364329622e-17
Execution time: 1.1685714550003468
----------
Experiment: Backtrack dynamic step
Result: 5.561213341395163e-23
Execution time: 2.0900338419996842
```

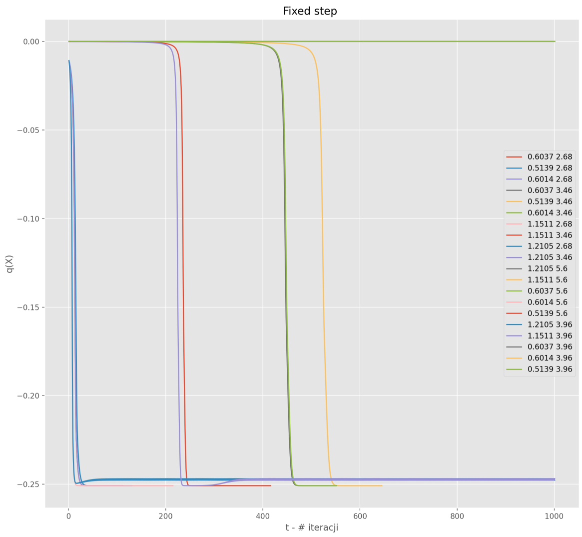
α=100

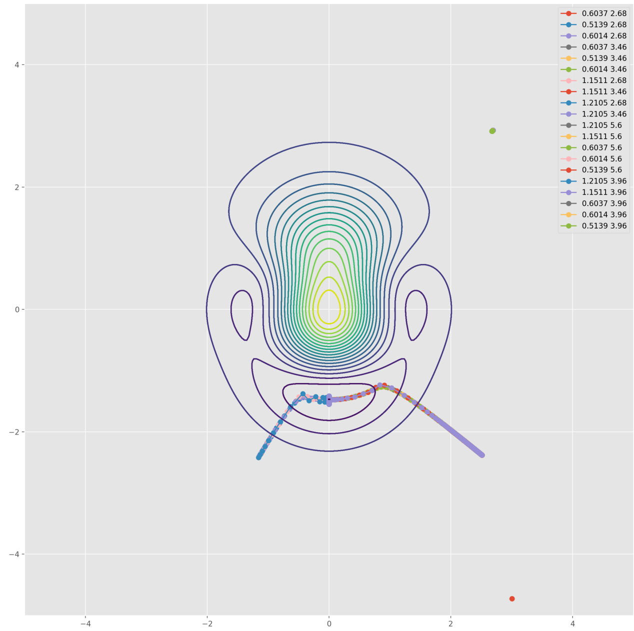


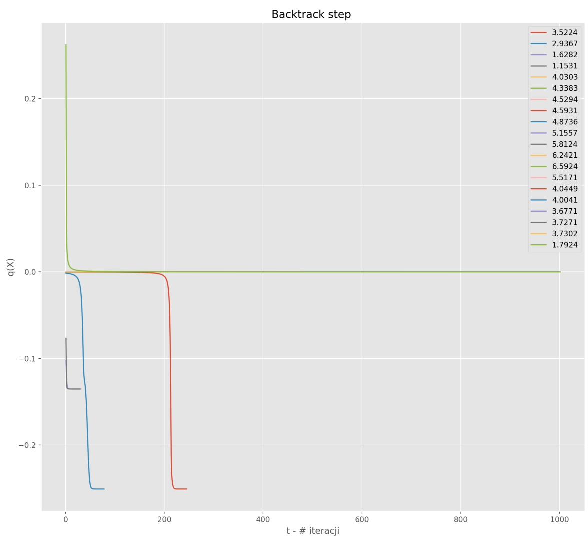


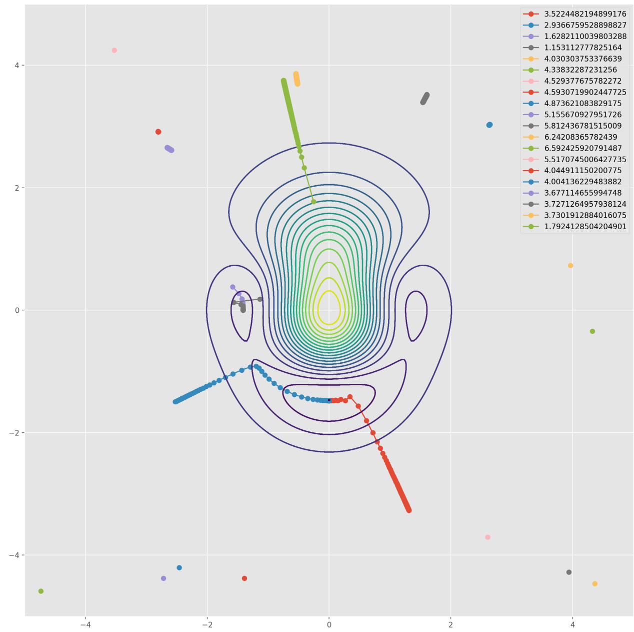

```
----------
Experiment: Random fixed step
Result: 4.387609704989057e-14
Execution time: 6.41408557899922
----------
Experiment: Backtrack dynamic step
Result: 3.210670718600228e-23
Execution time: 9.911294301000453
```

Wywołanie dla funkcji z wieloma punktami krytycznymi i "płaskim" obszarem










```
----------
Experiment: Random fixed step
Result: -0.2508093291388027
Execution time: 1.915485173001798
----------
Experiment: Backtrack dynamic step
Result: -0.250809329056063
Execution time: 1.3285440649997327
```

Analiza

Szybkość zbieżności

Przy odpowiednim dobraniu kroku, oba algorytmy w przypadku testowej funkcji osiągają zadowalającą wartość. Stały krok wymaga jednak testowania wielu potencjalnych wartości, stwarza ryzyko wpadnięcia w oscylację, lub dążenia do nieskończoności. Z wykresów można wyciągnąć wniosek o dużym wpływie wartości kroku na szybkość zbieżności - wykres stanowią proste o różnych stopniach nachylenia. Dynamiczny krok osiągał średnio zadowalającą optymalizację przy mniejszej liczbie iteracji.

Dokładność

Algorytm gradientu pozwala uzyskać dość dokładną zoptymalizowaną wartość - wartości gradientu zmniejszające się w pobliżu minimum pozwalają w tym obszarze dokładniej zbliżać się do punktu krytycznego - wpływ jednak ponownie wywiera krok, który może być za duży i spowodować na algorytmie ciągłe przeskakiwanie minimum. Algorytm z backtrackingiem jako jedyny stopień swobody posiada punkt startowy, co sprawia, że można go wywołać dla wielu potencjalnych punktów prowadzących do minimum, zwiększając szansę optymalizacji.

Zachowanie w punktach krytycznych

Mankament gradientu prostego stanowi nieodpowiedni wybór punktu startowego - punkt krytyczny, lub "płaski" obszar, sprawiają, że gradient jest bliski zeru, powodując "stanie w miejscu" algorytmu. Próby zwiększania kroku nie przynoszą żadnych rezultatów.

Szybkość wykonania

Ponieważ gradient prosty polega głównie na odejmowaniu wektorów, jest on wydajnym rozwiązaniem. Możliwości równoległego przetwarzania wielu punktów i kroków dodatkowo zmniejszają czasy wykonania. Porównując oba algorytmy, pojedyncze wywołania backtrackingu były wolniejsze, ze względu na dodatkową pętlę wykonującą się w każdej iteracji(jednakże pozwala ona zmniejszyć ilość iteracji algorytmu).

Ograniczenia

Niewątpliwą wadą gradientu prostego są ograniczenia jakie narzuca na optymalizowany obiekt w celu skutecznej optymalizacji - ciągłość, różniczkowalność, ograniczenie, wypukłość.