Dokumentacja do

WSI - ćwiczenie 1 Zagadnienie przeszukiwania

i podstawowe podejścia do niego

Wykonano przez Marfenko Mykhailo

Struktura programu

Program jest podzielony na 3 pliki:

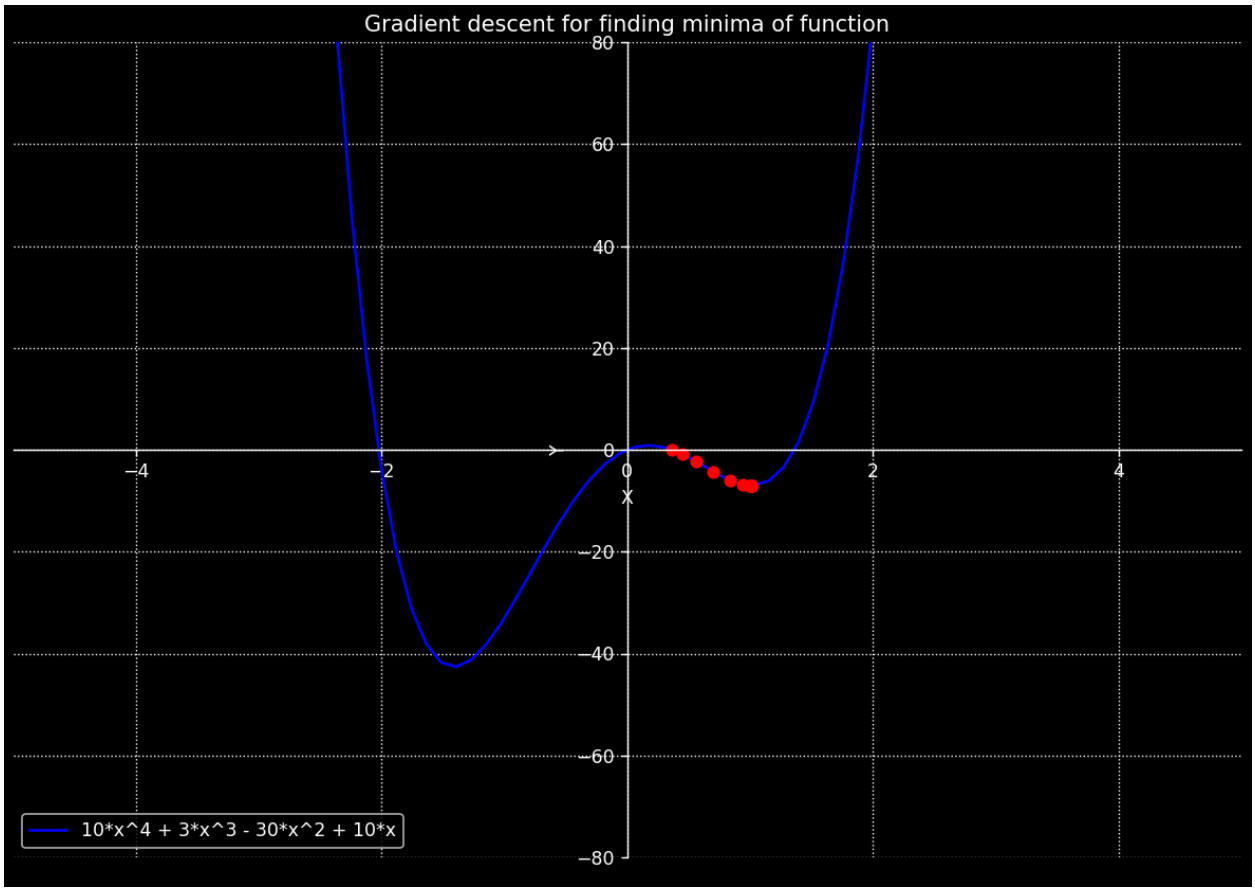
main.py – Z tego pliku wyłowuje się odpowiednie elementy programu

gradient.py – Zawiera algorytm gradient descent oraz przychowuje dane o funkcjach.

lr_tests.py – tutaj znajdują się funckje które dla badają wpływ rozmiaru kroku dla różnych (losowych) punktów początkowych dla funkcji f(x) i g(x) i wypisuje wyniki do konsoli.

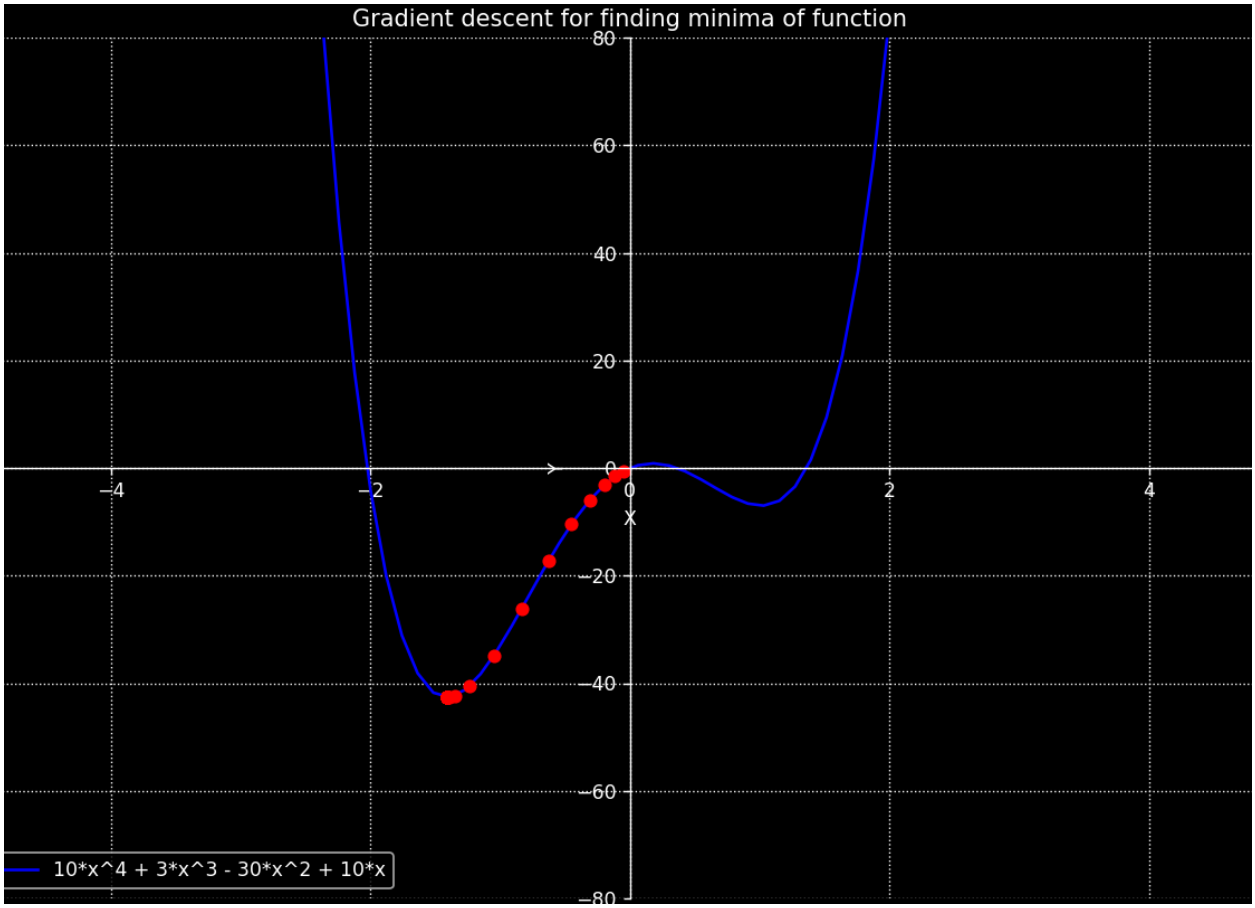
Rozpatrzymy działanie algorytmu dla pościególnych funkcji dla różnych punktów początkowych.

Funckcja f(x) = 10x^4 + 3x^3 − 30x^2 + 10x



Start x: 0.3; Iterations: 100; Learning rate: 0.01;

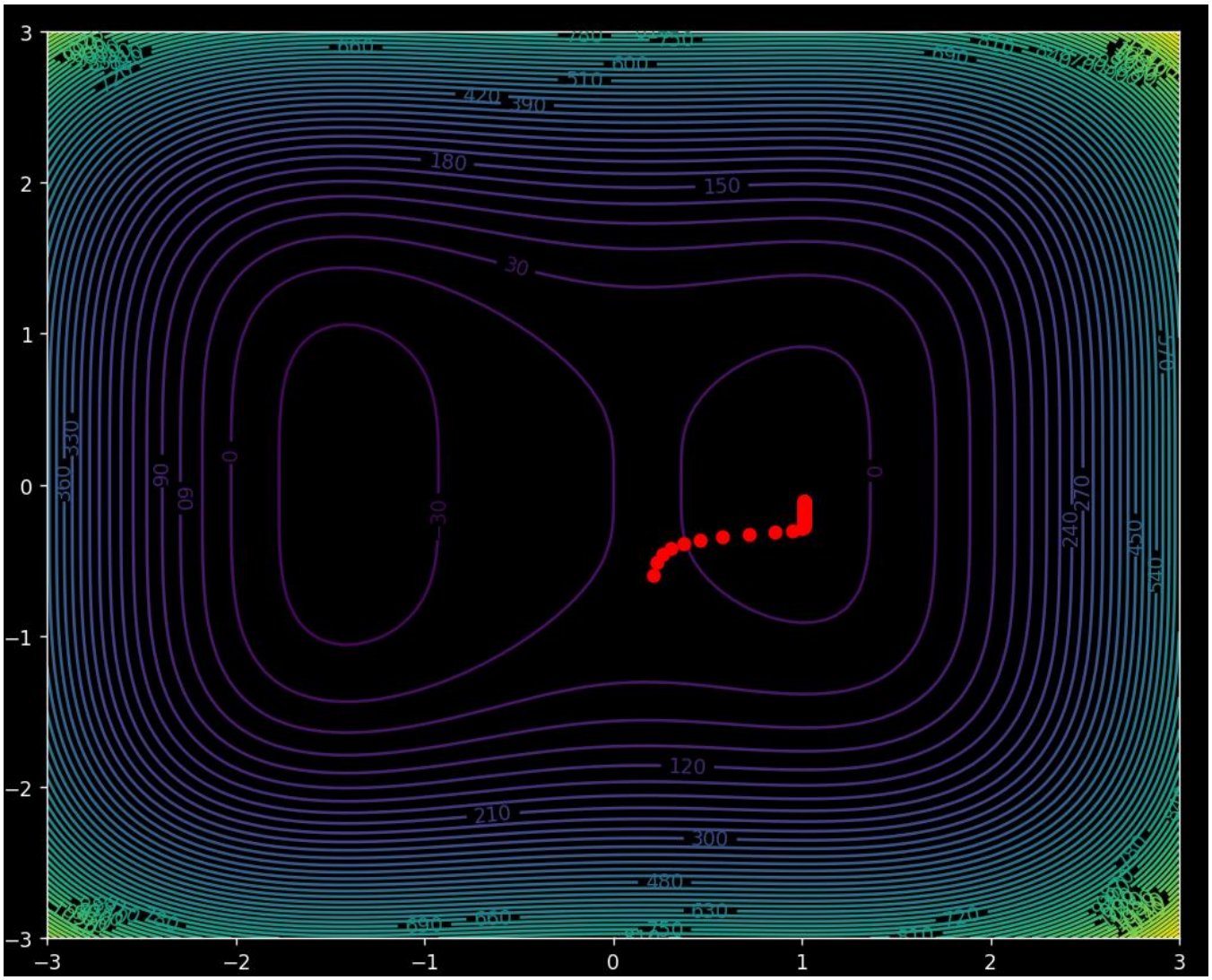
Minimum 1: ([1.012558652832393, 7.006322161247466])



Start x: 0; Iterations: 200; Learning rate: 0.005;

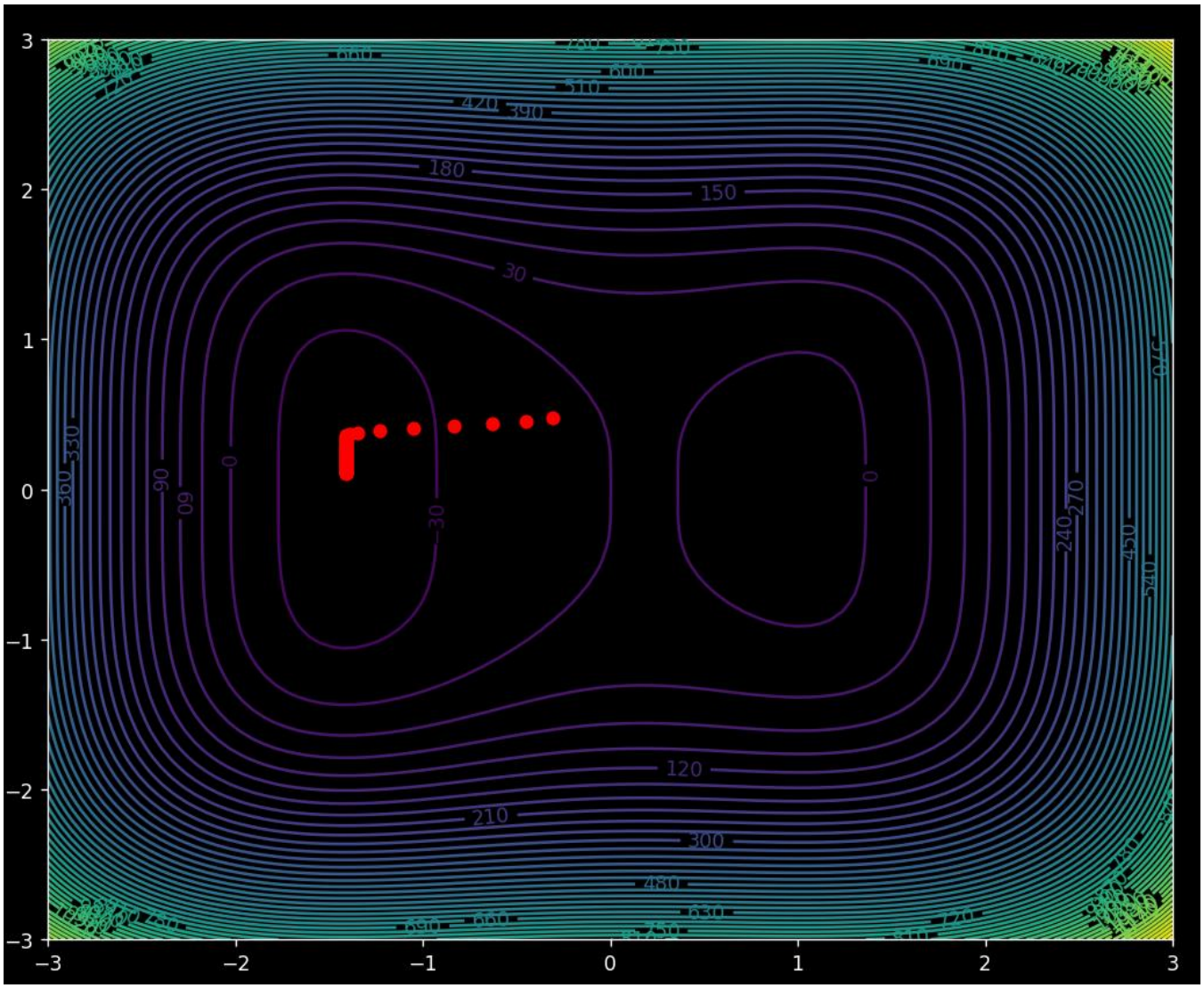
Minimum 2: ([-1.4123706144776538, -42.62767877897754])

Funckcja f(x) = 10y^4+10x^4 + 3x^3 − 30x^2 + 10x



Start (x, y): (0.2, -1); Iterations: 100; Learning rate: 0.01;

Minimum 1: ([1.012558652832393, -0.1090387674880599, -7.00490857036499])



Start (x, y): (-0.2, 0.5); Iterations: 200; Learning rate: 0.005;

Minimum 2: ([-1.4123706144776538, 0.10851037605730116, -42.62629239007186])

Badanie wpływu rozmiaru kroku dla różnych (losowych) punktów początkowych:

Rozpatrzymy funkcję f(x):

Niech x_values będą punktami początkowymi x dla funkcji f(x) zaś lr_values – znaczeniami learning rate:

```
x_values = [-3, -2.5, -2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3]
lr_values = [0.00005, 0.0005, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.0175, 0.025, 0.05, 0.1]
```

Teraz możemy zbadać poprawność algorytmu dla każdego punktu początkowego oraz learning rate. Wszystkie testy będą mieli iterations = 1000. Możliwe są 3 wyniki: 1) learning rate jest za małe co powoduje że algorytm nie zdąży dojść do minimuma i da nie poprawną odpowiedż. 2) Learning rate jest dobrym i z danego punktu początkowego znaleźliśmy minimum. 3) learning rate jest za wielki co powoduje że algorytm da błąd obliczeniowy (Overflow Error) albo będzie skakać i nie dojdzie do minimum.

Wyniki:

Weźmiemy taki kawałek wyniku:

```
Start_x: -3
Lr is too LOW / High: 5e-05, difference is 0.00024207205985260494
OK for lr: 0.0005
OK for lr: 0.001
OK for lr: 0.0025
OK for lr: 0.005
Lr is too HIGH: 0.0075
Lr is too HIGH: 0.01
Lr is too HIGH: 0.0175
Lr is too HIGH: 0.025
Lr is too HIGH: 0.05
Lr is too HIGH: 0.1

Start_x: -2.5
Lr is too LOW / High: 5e-05, difference is 0.00021209037139313658
OK for lr: 0.0005
OK for lr: 0.001
OK for lr: 0.0025
OK for lr: 0.005
OK for lr: 0.0075
OK for lr: 0.01
Lr is too HIGH: 0.0175
Lr is too HIGH: 0.025
Lr is too HIGH: 0.05
Lr is too HIGH: 0.1

Start_x: -2
Lr is too LOW / High: 5e-05, difference is 0.0001563307909917544
OK for lr: 0.0005
OK for lr: 0.001
OK for lr: 0.0025
OK for lr: 0.005
OK for lr: 0.0075
OK for lr: 0.01
OK for lr: 0.0175
Lr is too HIGH: 0.025
Lr is too HIGH: 0.05
Lr is too HIGH: 0.1

Start_x: -1.5
OK for lr: 5e-05
OK for lr: 0.0005
OK for lr: 0.001
OK for lr: 0.0025
OK for lr: 0.005
OK for lr: 0.0075
OK for lr: 0.01
Lr is too LOW / High: 0.0175, difference is 0.06570622366239998
Lr is too LOW / High: 0.025, difference is 0.06292786468985945
Lr is too HIGH: 0.05
Lr is too HIGH: 0.1
```

1. Zauważymy, że lr = 0.00005 prawie zawsze jest za małe dla 1000 iteracji żeby dojść do minimum, ale jeżeli startować z punktu -1.5, to jest OK. Tak jest dlatego, że ten punkt jest bardzo bliski do minimum.
2. Prawie zawsze lr = 0.0075 daje poprawny wynik, ale jeżeli startować z punktu -3, to ten lr jest już za duży, bo „slope" is too steep, i nawet przy małym lr ono „idzie za daleko". Dlatego dla danej funkcji w punktach dalszych od 0 trzeba brać mniejsze lr.
3. I tak samo lr około 0.025 jest dobry w tych przypadkach gdy start x jest blisko minimum. Czyli dla tej funkcji gdy „slope" is NOT too steep.
4. Także zauważmy, że np. Dla start_x = -1.5 Lr = 0.0175 I 0.025 jest za Wysokim, ale nie daje błąd obliczeniowy. Bo nie wychodzi za daleko, a tylko w tym samym miejscu robi „back and forth" i nie dochodzi za 1000 iteracji do minimum.

Dla funkcji g(x) wszystko jest analogicznie.