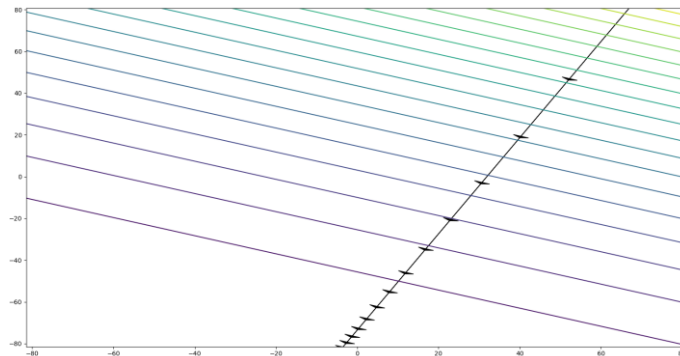


Funkcja f1:

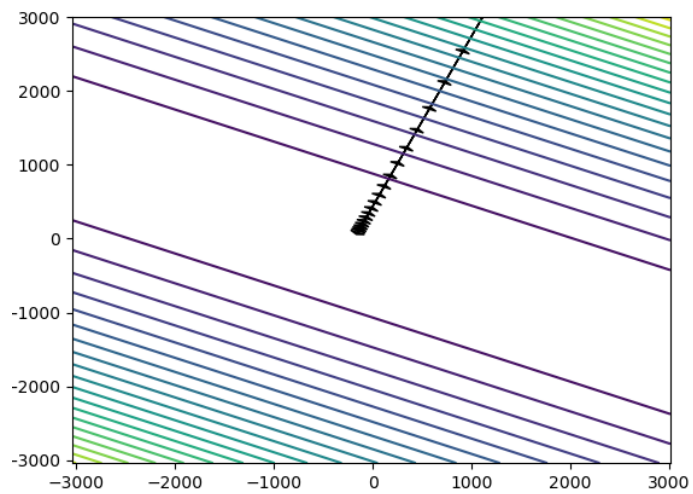
$\beta = 0.00000001$

$n = 20$

minimum:



maksimum:

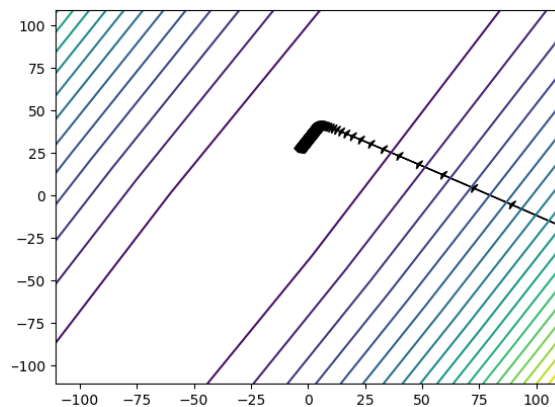


Funkcja f2:

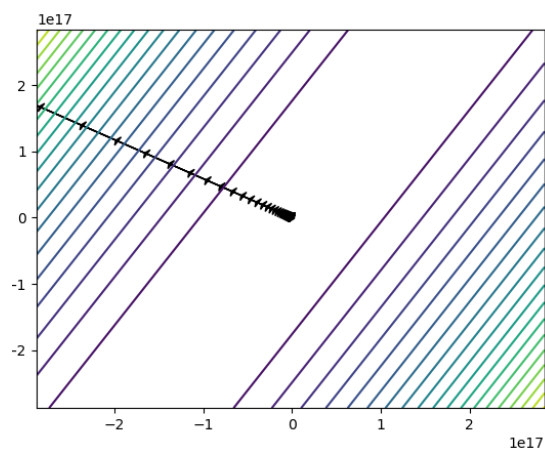
$\beta = 0.5$

$n = 200$

minimum:



maksimum:

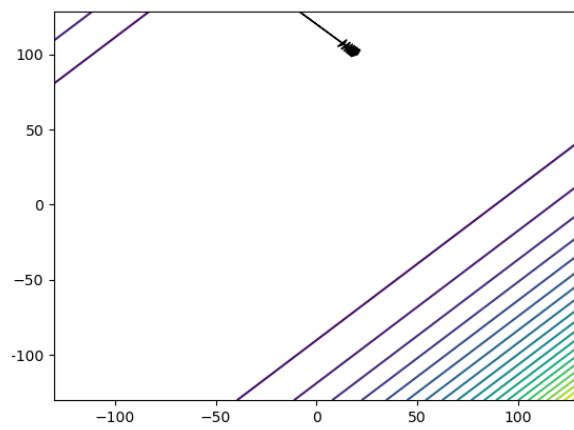


Funkcja f3:

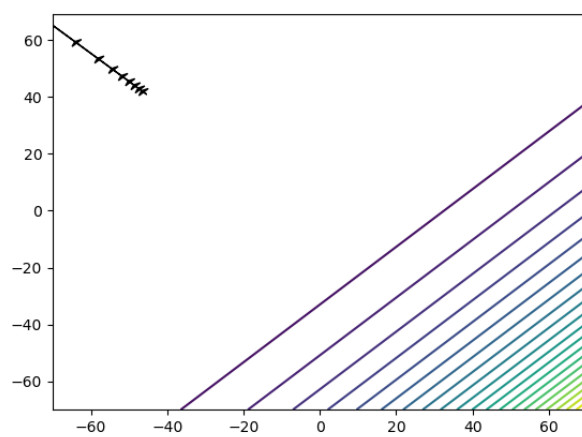
beta = 0.00003

n = 20

minimum:



maksimum:

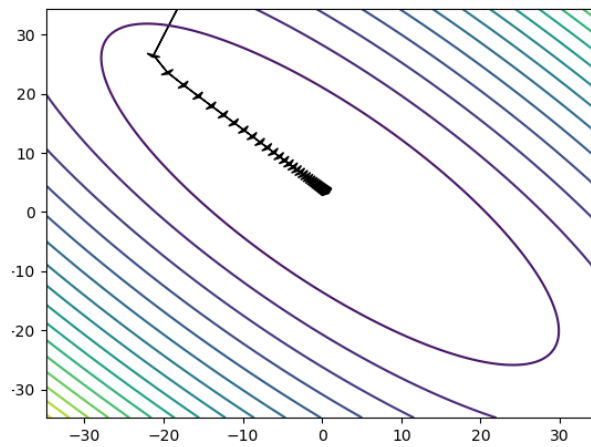


Funkcja booth:

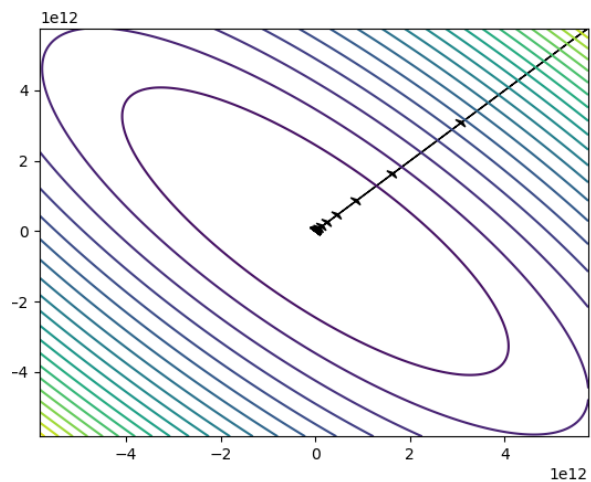
$\beta = 0.00003$

$n = 20$

minimum:



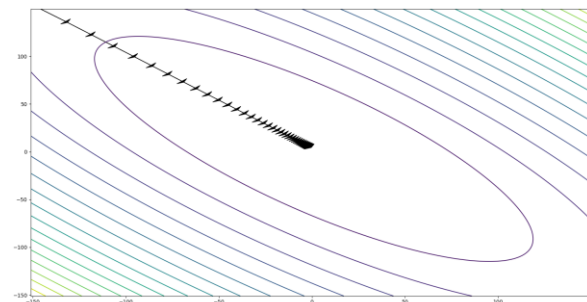
maximum:



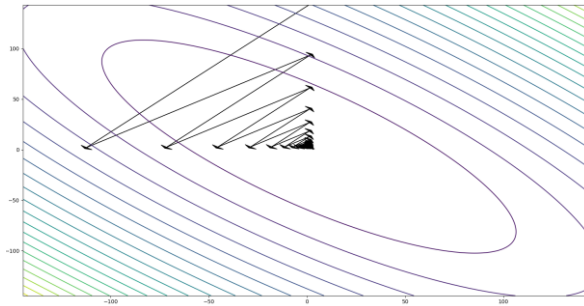
Pytania

1. Jak wartość parametru β wpływa na szybkość dojścia do optimum i zachowanie algorytmu?

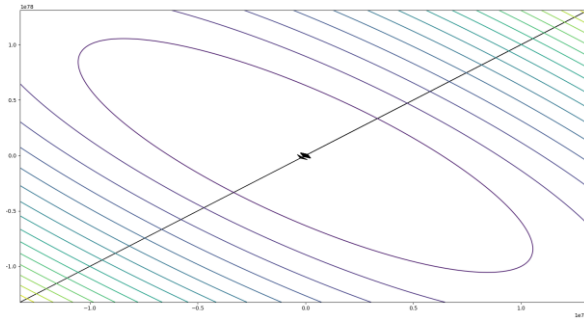
$\beta = 0.05$



$\beta = 0.1$



beta = 5



Im mniejsza beta, tym nasza precyzja jest większa, ale za to nasz czas dojścia do optimum się zwiększa. Z kolei jeżeli będziemy brali coraz większą betę, nasza precyzja się zmniejszy (będziemy mieli bardzo duże skoki), czas dojścia może się dużo zmniejszyć, jednakże istnieje duże ryzyko przeskoczenia naszego optimum, przez co możemy go nigdy nie oddać.

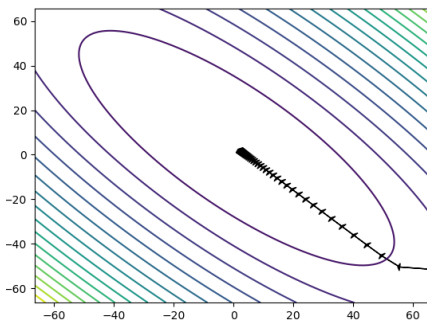
2. Zalety/wady algorytmu?

Zalety:

- prosta implementacja
- prosta implementacja dla funkcji wielowymiarowych (nie tylko 2 wymiary)

Wady:

- trudne dobranie parametru beta
- nie dąży prosto do celu



3. Wnioski

Algorytm może wydawać się na bardzo prosty i przyjemny, ponieważ łatwo go zaimplementować. Jednakże jego wykorzystanie nie należy do najłatwiejszych, ponieważ musimy odpowiednio dostosować parametr β , który może sprawiać wiele kłopotów, nie może być zbyt mały, żeby obliczenia nie trwały za długo, nie może być za duży, ponieważ wtedy możemy wcale nie otrzymać wyników. Lepszą alternatywą jest algorytm Newtona, który pojawił się na wykładzie.