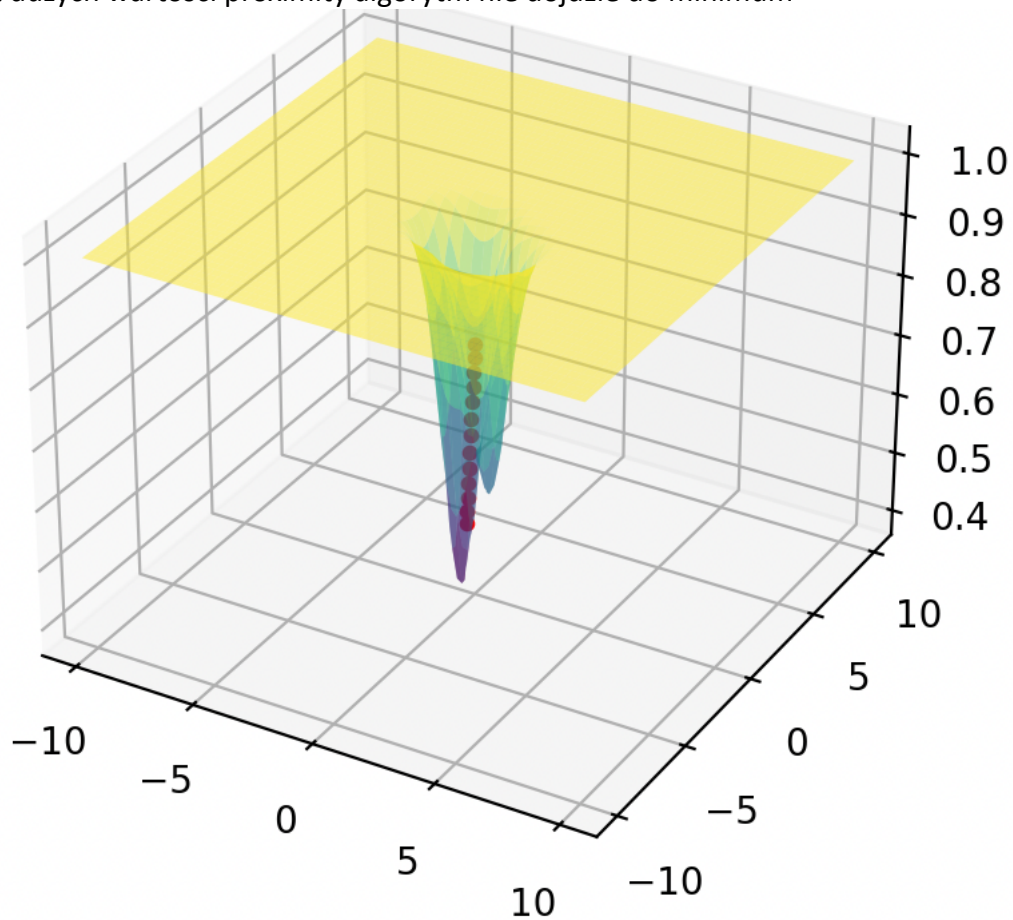


## Badanie proximity

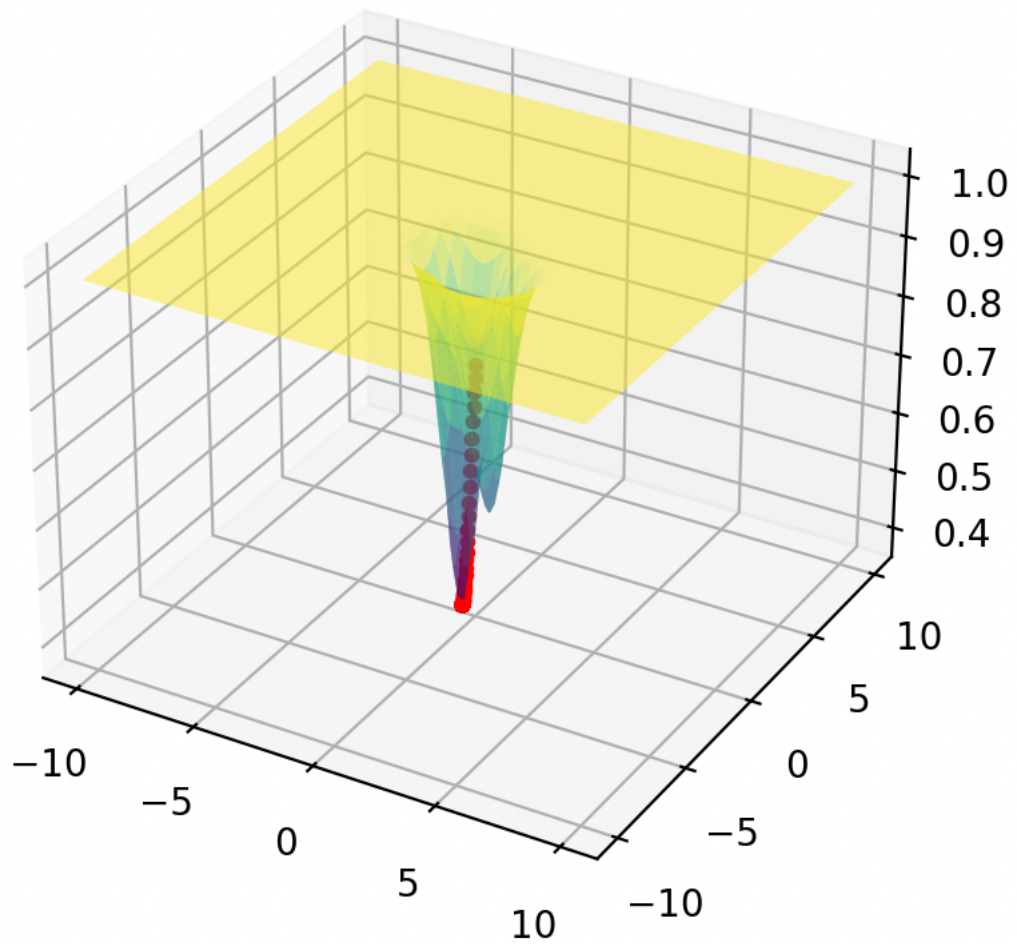
Dla zbyt dużych wartości proximity algorytm nie dojdzie do minimum



Dla zbyt małego proximity algorytm będzie na tyle blisko że w zasadzie nie będzie się poruszał, ale będzie iterował cały czas. Obiawia się to tym że kolejne punkty w zasadzie się od siebie nie różnią

Kolejno (x,y,z) dla punktów pod koniec iteracji

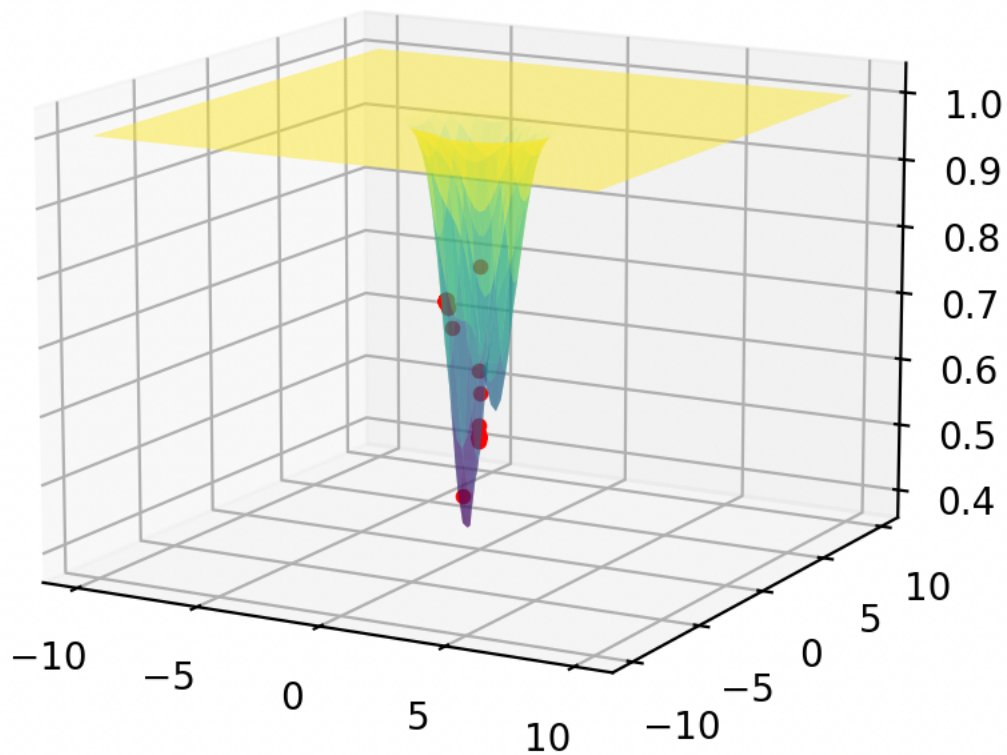
[illegible]



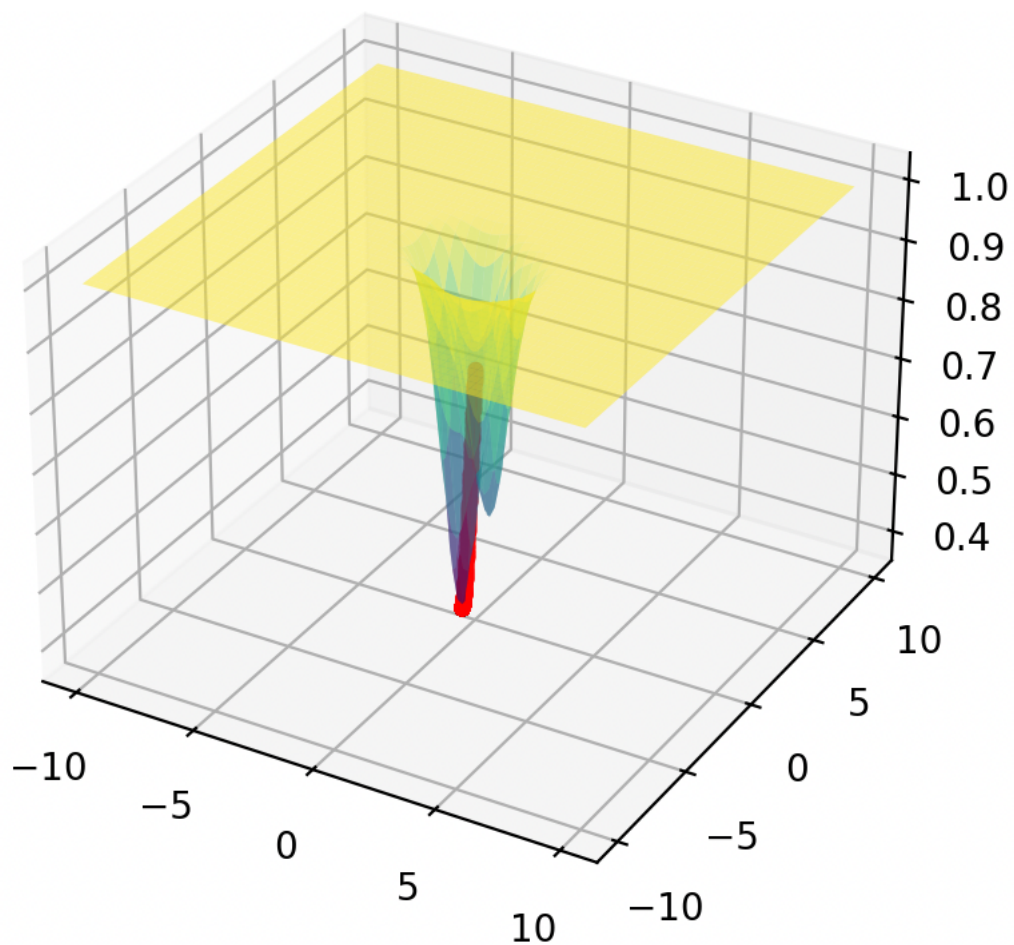
---

#### Badanie długości kroku

Dla zbyt dużego kroku przeskakuje on punkt minimalny, a więc dochodzi do odbijania się po ścianach



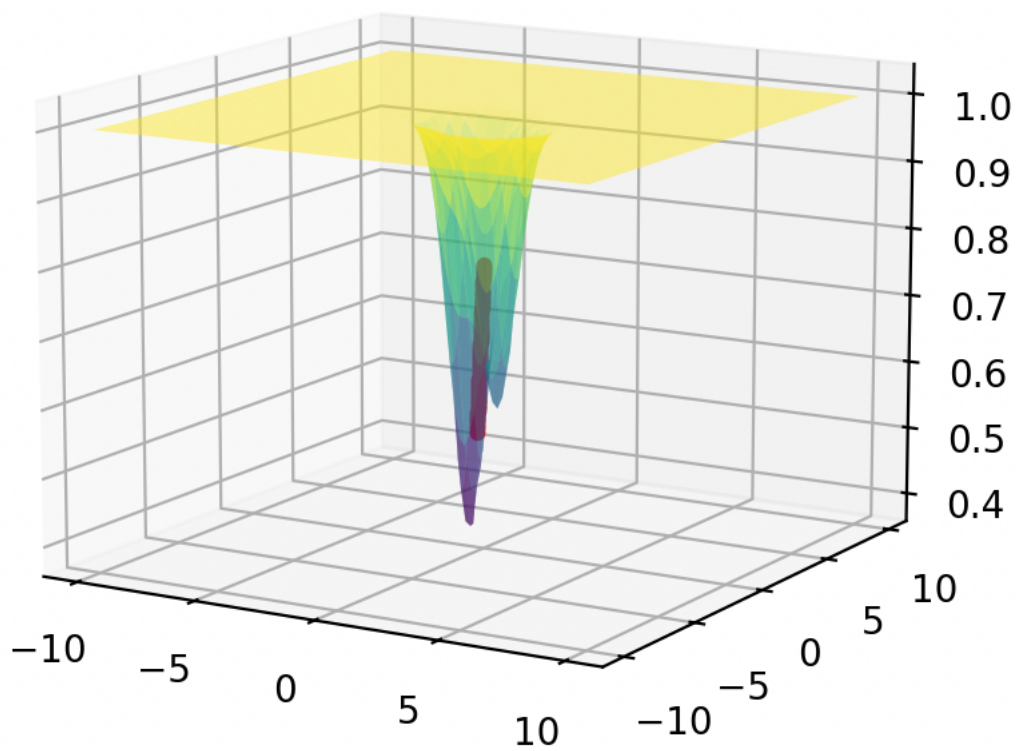
Dla dostatecznie małego kroku algorytm trafia do minimum. Za zmniejszaniem długości kroku musi iść zwiększanie. Ilości iteracji, bo dochodzimy do przypadku gdy algorytm nie zbliża się do minimum(opisany niżej)



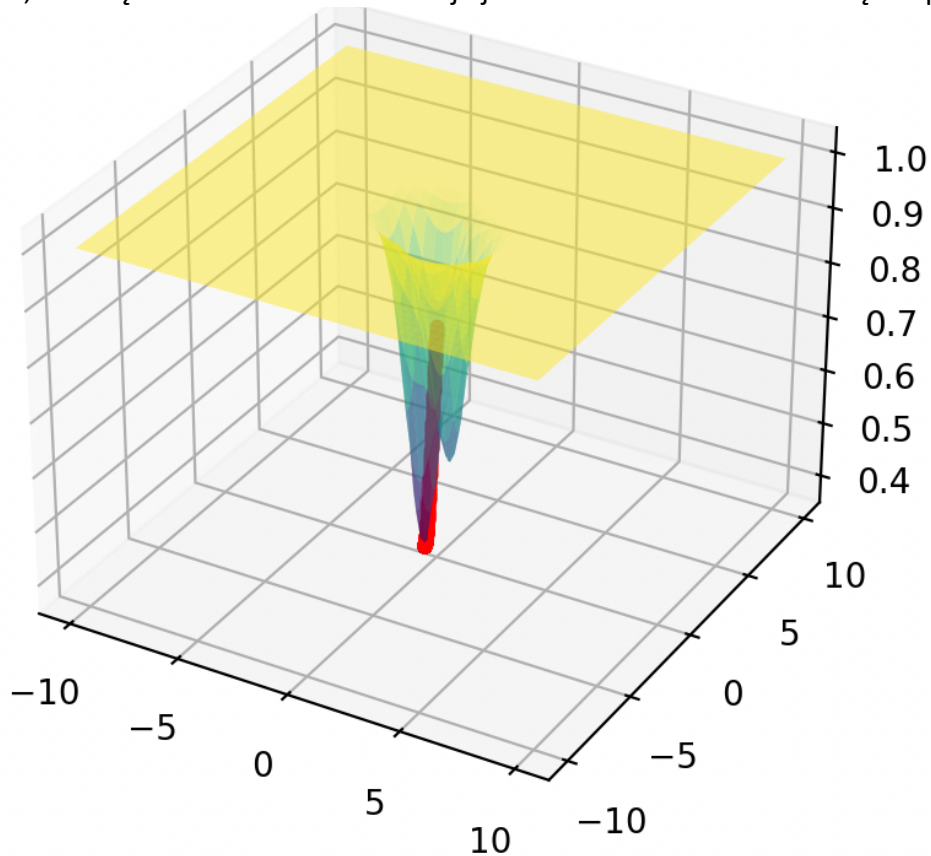
---

#### Badanie zmiany ilości wykonywanych kroków

Dla zbyt małej ilości kroków algorytm nie dojdzie do minimum, podając zbyt duży wynik



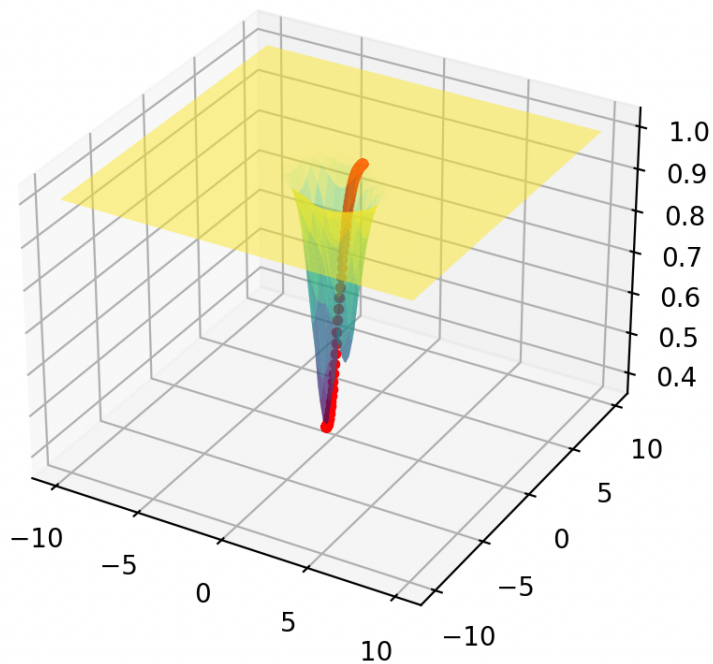
Dla dużej ilości iteracji ( $i$  na tyle małego proximity że nie ma wpływu) algorytm znajduje minimum, ale większość czasu każda iteracja jest tak mała że w zasadzie się nie przemieszcza



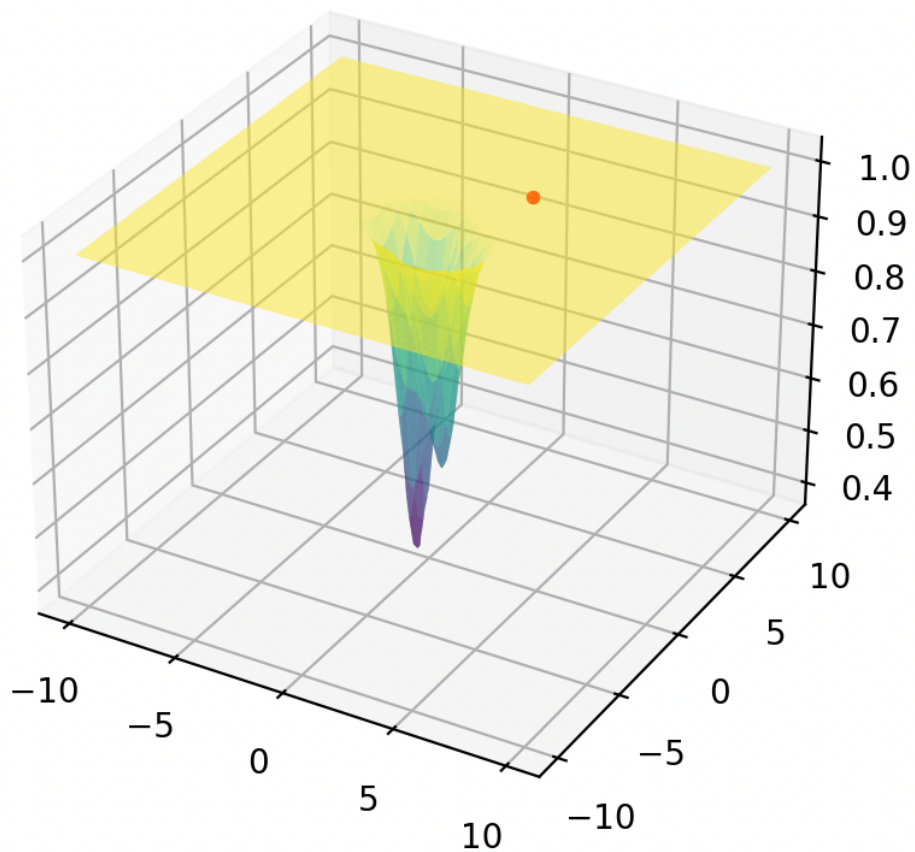


### Badanie różnych punktów startowych

Dla punktu (1,1), który jest na granicy dla odpowiednio dużego kroku znajduje prawidłowo minimum



Jeśli zaczniemy w miejscu gdzie płaszczyna jest płaska algorytm utknie i się nie ruszy, bo gradient będzie wektorem zerowym



Może się zdarzyć że natrafimy na minimum lokalne nie będące minimum globalnym

