

MOFIT Lab4 - Równanie Poissona: relaksacja i nadrelaksacja

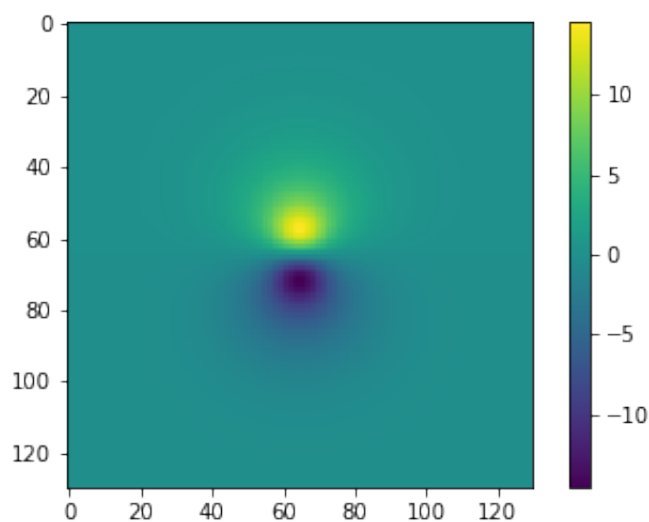
Jan Malczewski

maj 2022

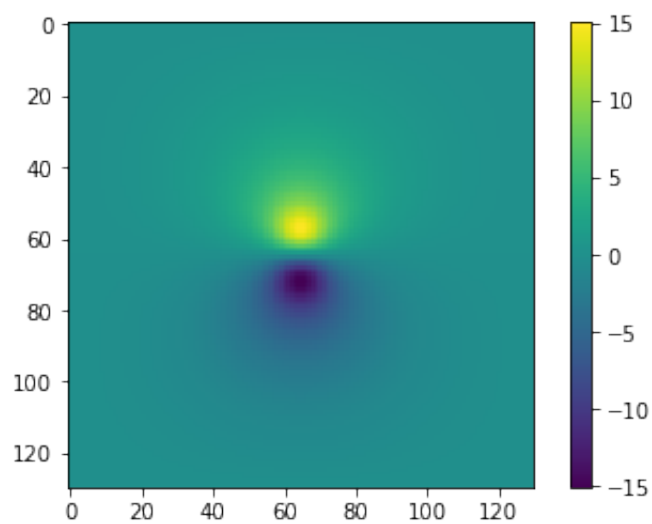
1 Wstęp

Do wykonania zadań użyto języka Python 3 oraz bibliotek numpy oraz matplotlib. Poniżej zamieszczam wykresy wraz z adnotacjami do konkretnych zadań. Wszystkie wartości na osiach zostały zapisane w układzie SI. Optymalne w mierzyliśmy poprzez utworzenie macierzy różnic sąsiednich wartości a oraz wyszukanie w której komórce najwcześniej występuje wartość poniżej buforu $\sigma = 0.00005$.

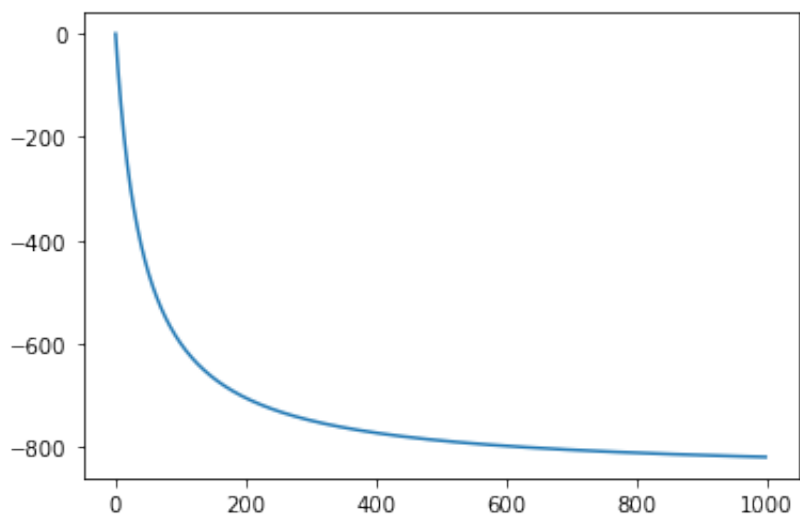
2 Wyniki



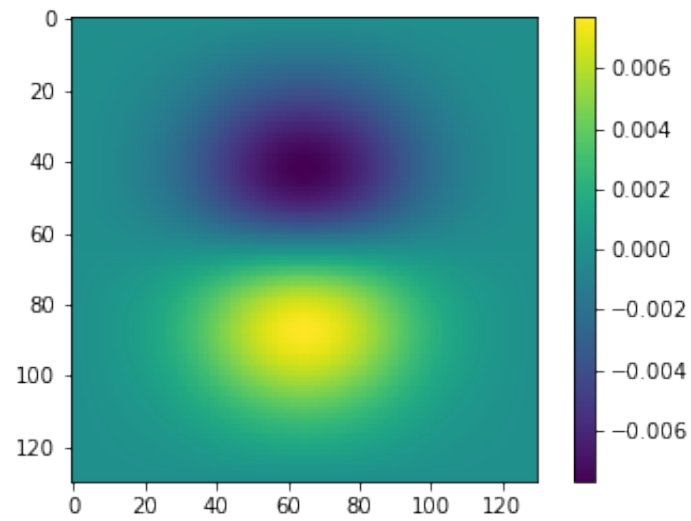
Rysunek 1: **Zad. 1** $u(i,j)$ po 1000 iteracji.



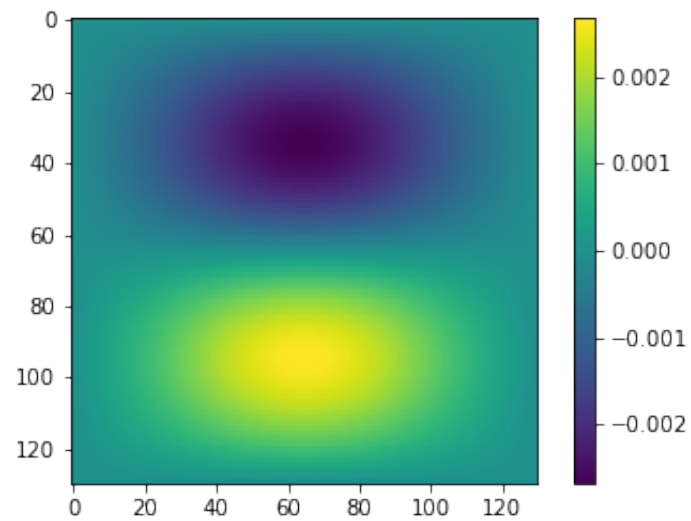
Rysunek 2: **Zad. 1** $u(i,j)$ po 2000 iteracji.



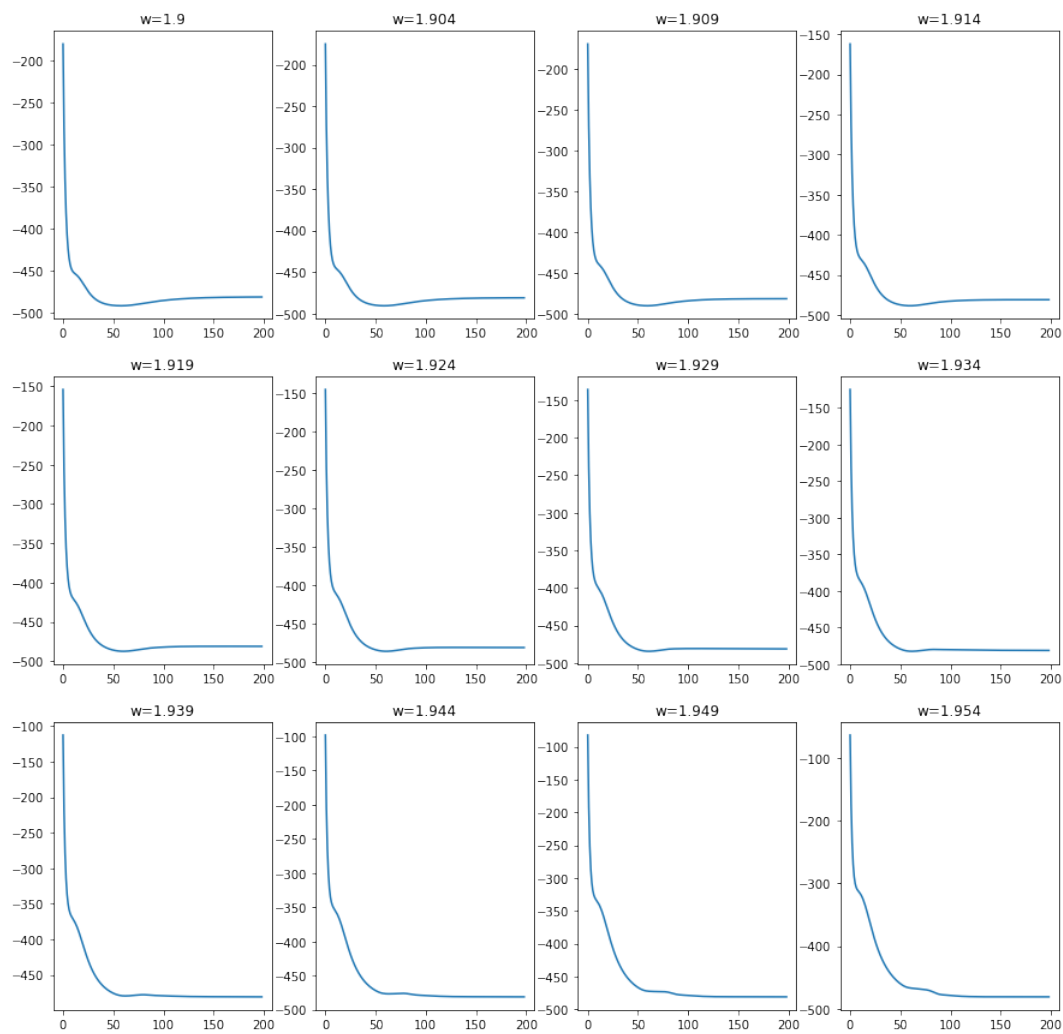
Rysunek 3: **Zad. 1** a od numeru iteracji.



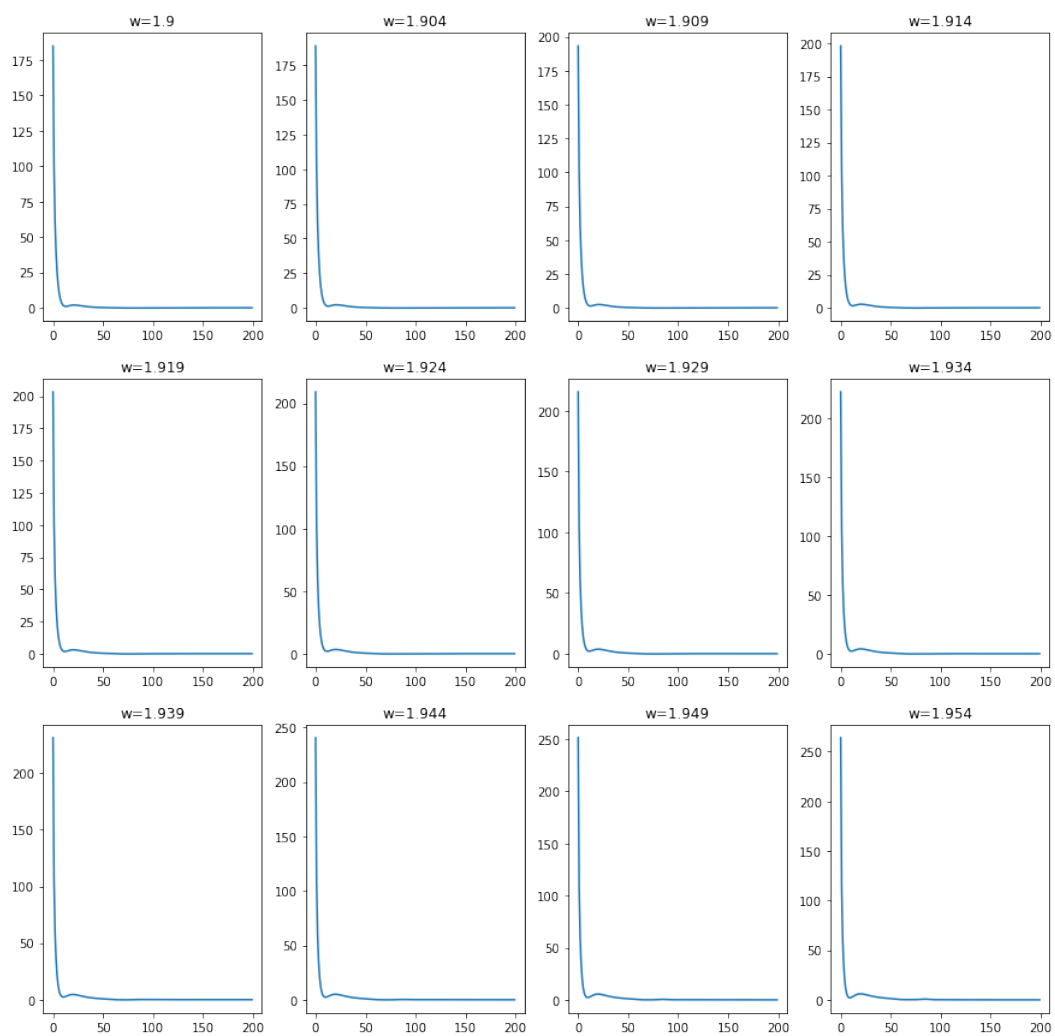
Rysunek 4: **Zad. 1** $\delta(x, y) = \rho'(x, t) - \rho(x, y)$ dla 1000 iteracji.



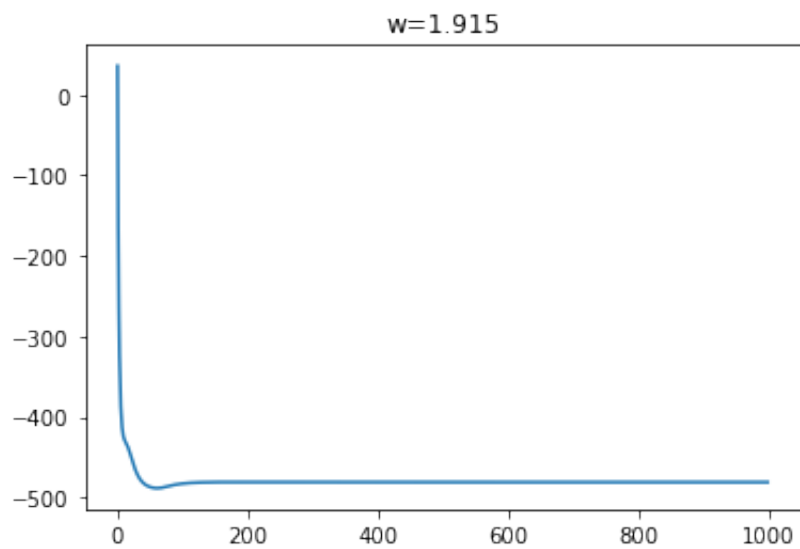
Rysunek 5: **Zad. 1** $\delta(x, y) = \rho'(x, t) - \rho(x, y)$ dla 2000 iteracji.



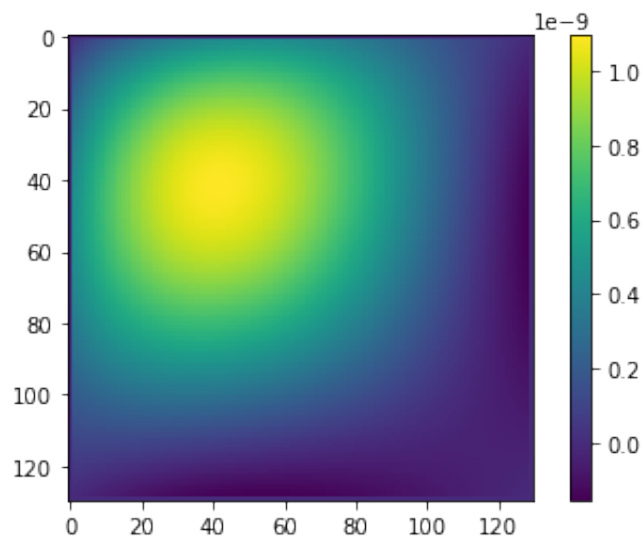
Rysunek 6: **Zad. 2** a dla 200 iteracji w zależności od współczynnika w .



Rysunek 7: **Zad. 2** Różnica sąsiednich wartości a dla 200 iteracji w zależności od współczynnika w .



Rysunek 8: **Zad. 3** a dla 1000 iteracji w pobliżu optimum.



Rysunek 9: **Zad. 3** $\delta(x, y)$ dla $w = 1.915$.