Metody numeryczne – projekt 2

Projekt zaimplementowałem w C++ . Wykorzystałem aspekty programowania obiektowego, dzięki czemu miałem możliwość użycia przeciążania operatorów. To pozwoliło na łatwe zapisywanie działań macierzowych podobnie jak to jest w MATLABie. Przykładowo:

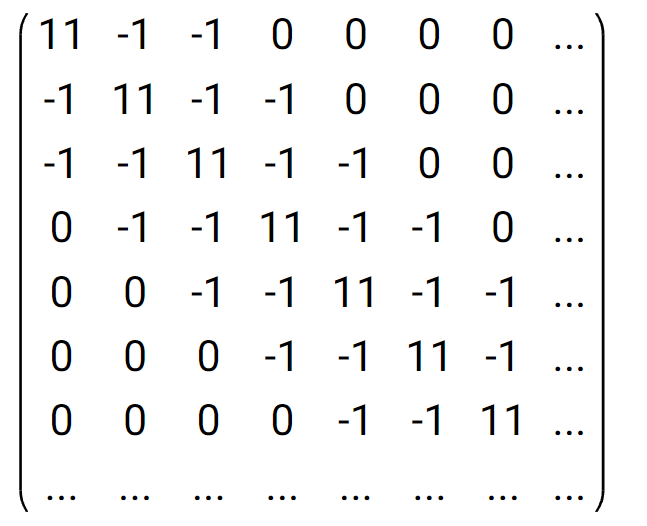
C++: (!(A \* -b) \* C) % D

MATLAB: ((A \* -b).**'** \* C) \ D

Rysowanie wykresów jest zrobione przy wykorzystaniu zewnętrznej instalacji MATLABa. Komendy do rysowania wykresów są wywoływane z poziomu programu w C++ przy pomocy funkcji system().

# Zadanie A

Dla zadania A, macierz ze współczynnikami równań układu wygląda następująco:



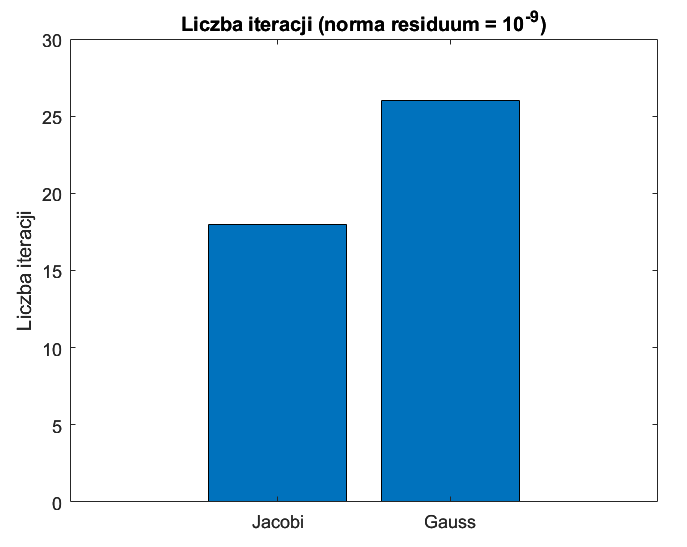
Na głównej diagonali jest a1 = 5 + 6 = 11

# Zadanie B

Liczba iteracji potrzebna do uzyskania normy residuum równej zależnie od metody

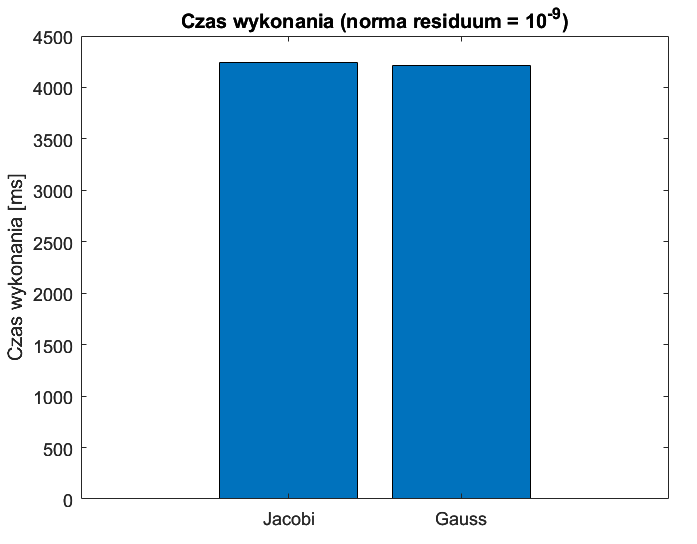
#### Liczba iteracji:

* **Jacobi**: 18
* **Gauss**: 26



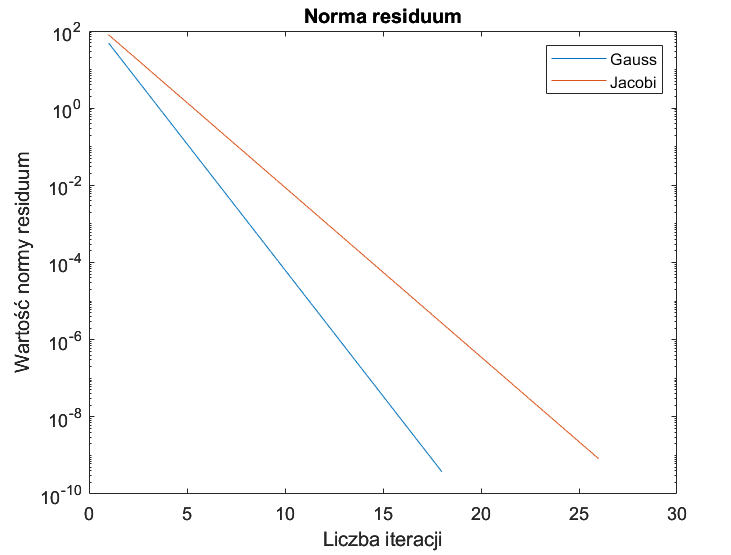
#### Czas wykonania:

* **Jacobi**: 4240ms
* **Gauss**: 4212ms



Widać, ze pomimo znacznej różnicy w liczbie iteracji, to czas jest bardzo zbliżony do siebie. Czas wykonania metody Jacobiego względem Gaussa to mniej niż 1%.

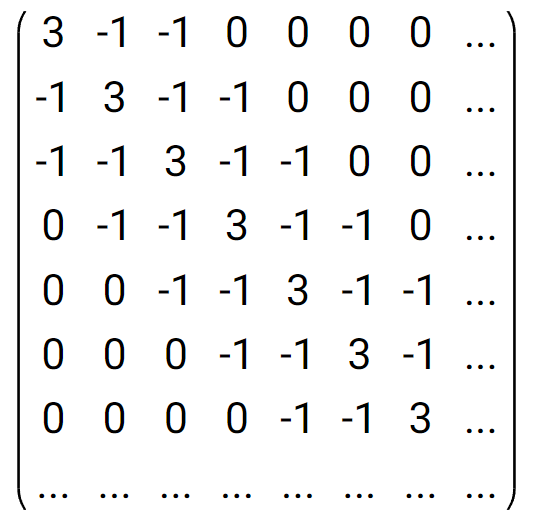
#### Wykres residuum względem iteracji:



W metodzie Gaussa norma residuum maleje szybciej niż w metodzie Jacobiego, dzięki czemu metoda Gaussa potrzebuje mniej iteracji, żeby uzyskać tą samą normę residuum.

# Zadanie C

W tym zadaniu macierz współczynników wygląda następująco:



Główna diagonala wynosi 3.

Dla takiego układu równań metody iteracyjne nie zbiegają się. Z kolejnymi iteracjami metod norma residuum wzrasta. Aby to wykryć sprawdzałem czy z każdą iteracją norma residuum maleje, jeśli nie to wyrzucałem wyjątek czasu wykonywania:

terminate called after throwing an instance of 'std::runtime\_error'

what(): methods are not converging d

# Zadanie D

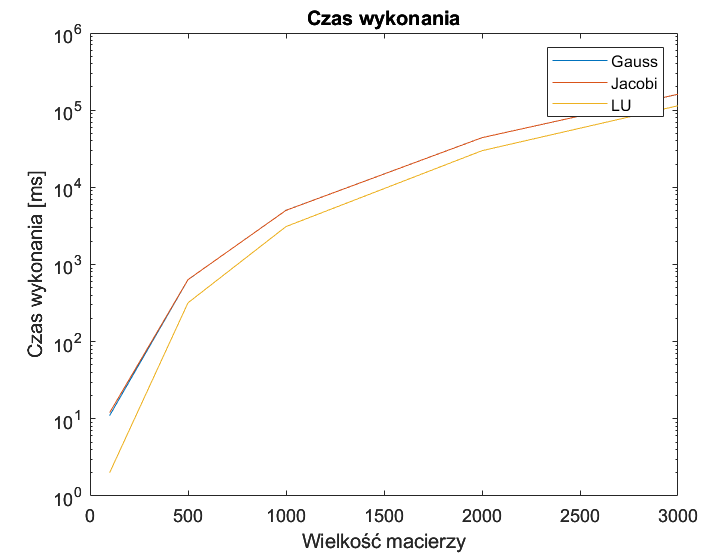
W przypadku faktoryzacji LU, norma residuum po użyciu metody wynosiła:

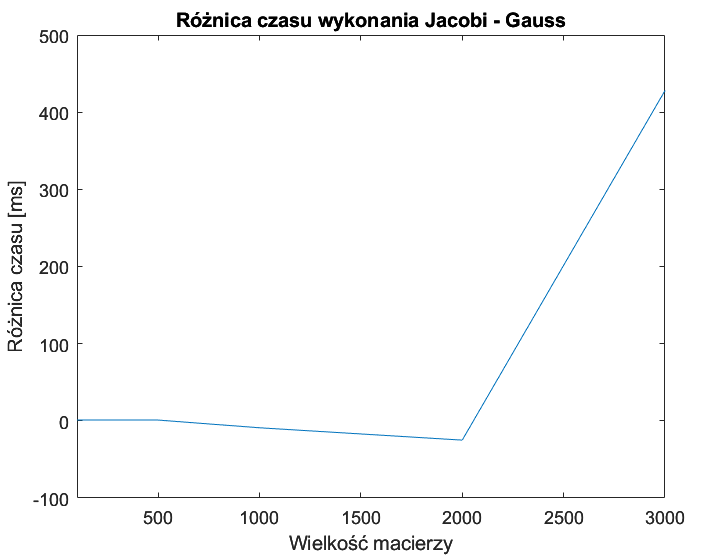
8.57584e-14

Jest to o pare rzędów wielkości mniejsza niż wskazana norma residuum dla metod iteracyjnych.

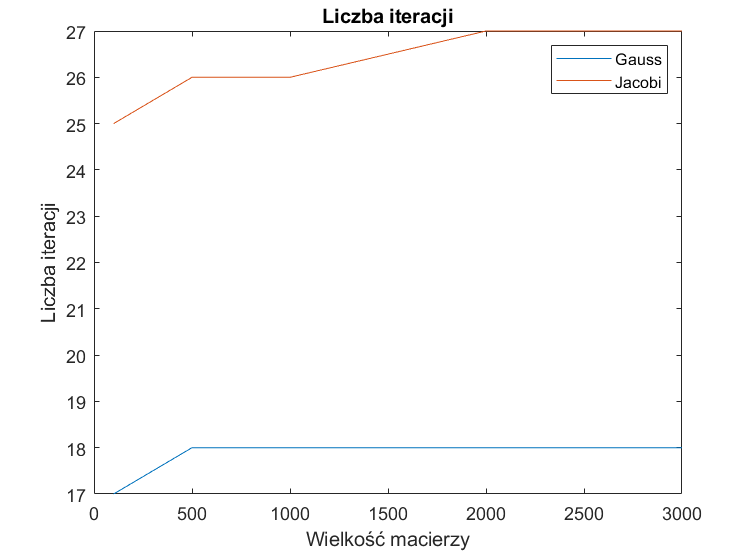
# Zadanie E

Dla kolejnych wielkości macierzy zostały wykonane metody: (100, 500, 1000, 2000, 3000).



Jak widać metody iteracyjne mają prawie identyczne czasy wykonania. Natomiast metoda faktoryzacji LU jest szybsza od każdej metody iteracyjnej. 

Czasy wykonania metod iteracyjnych różnią się od siebie w setkach milisekund.



Liczba iteracji stabilizuje wraz ze zwiększeniem wielkości macierzy dla metody Gaussa. Liczba iteracji względem wielkości macierzy nie zwiększa się liniowo dla obu metod.

# Podsumowanie

Metody iteracyjne są wolniejsze niż metoda faktoryzacji LU. Metoda Jacobiego potrzebuje więcej iteracji niż metoda Gaussa do rozwiązania układu równań dla takiej samej normy residuum. Mimo, to te dwie metody w przypadku implementacji w C++ praktycznie nie różnią się od siebie czasem wykonania. Liczba iteracji do rozwiązania układu równań nie zwiększa się liniowo względem liczby równań.