# *Spawozdanie z projektu z przedmiotu Metody Numeryczne*

# *Projekt : Układy równań liniowych*

Metody iteracyjne (przybliżone) służą obliczaniu przybliżonych wartości pierwiastków układu równań liniowych za pomocą kolejnych aproksymacji. W tym celu należy przyjąć pewien dowolny wektor $\mathbf{x}^{(0)}$ jako rozwiązanie początkowe i według określonego schematu tworzyć kolejno ciąg wektorów $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$,..., $\mathbf{x}^{(n)}$,... tak, aby wektor $\mathbf{x}^{(n+1)}$ lepiej przybliżał rozwiązanie dokładne od wektora $\mathbf{x}^{(n)}$.

Najprostszymi lecz nadal często wykorzystywanymi metodami iteracyjnymi są metoda Jacobiego oraz metoda Gaussa- Seidla. Obydwie polegają na podziale macierzy wejściowej na górną macierz trójkatną, dolną macierz trokątną oraz macierz diagonalną.

L -> macierz trójkątna dolna
U -> macierz trójkątna górna
D -> macierz diagonalna
b -> wektor impulsu

Schemat dla obydwu metod:

- Gauss-Seidel -> $x^{(n+1)} = (D - L)^{-1}(Ux^{(n)}) + (D - L)^{-1}b$
- Jacobi -> $x^{(n+1)} = D^{-1}(L + U)x^{(n)} + D^{-1}b$

Najpowszechniejszą metodą sprawdzanie dokładności metod iteracyjnych jest obliczanie normy drugiej z residuum przybliżonego rozwiązania :

- $r^{(n)} = Ax^{(n)} - b$

Do obliczania niewielkich macierzy można stosować również bezpośrednie metody rozwiązywania, takie jak na przykład metoda faktoryzacji LU, polegająca na stworzeniu 2 macierzy trójkątnych (górnej i dolnej), których iloczyn to macierz wejściowa. Pozwala to na uproszczenie działań przy potrzebie obliczania wyników dla układu równań dla wielu wektorów pobudzenia b.

# I Badana macierz

Badaną macierzą jest macierz **A**. Jest ona tzw. macierzą pasmową o rozmiarze 925x925. Zawiera ona pięć diagonali - główna z elementami a1, dwie sąsiednie z elementami a2 i dwie skrajne diagonale z elemetnami a3.

$$\mathbf{A} = \begin{bmatrix} a1 & a2 & a3 & 0 & 0 & 0 & 0 & \dots & 0 \\ a2 & a1 & a2 & a3 & 0 & 0 & 0 & \dots & 0 \\ a3 & a2 & a1 & a2 & a3 & 0 & 0 & \dots & 0 \\ 0 & a3 & a2 & a1 & a2 & a3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & a3 & a2 & a1 \end{bmatrix}$$

Prawa strona równania to wektor **b** o długości 925, którego n-ty element jest obliczony wg wzoru sin(5*n). A z kolei w wyniku rozwiązania układu równań otrzymujemy wektor **x**.

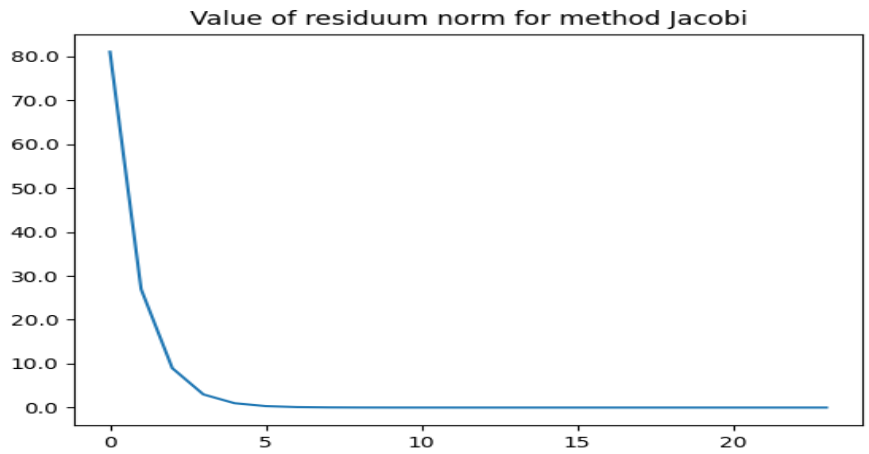
Bezpośrednio badane będą 2 wersje macierzy A:

- macierz nr 1, gdzie a1 = 12, oraz a2 = a3 = -1

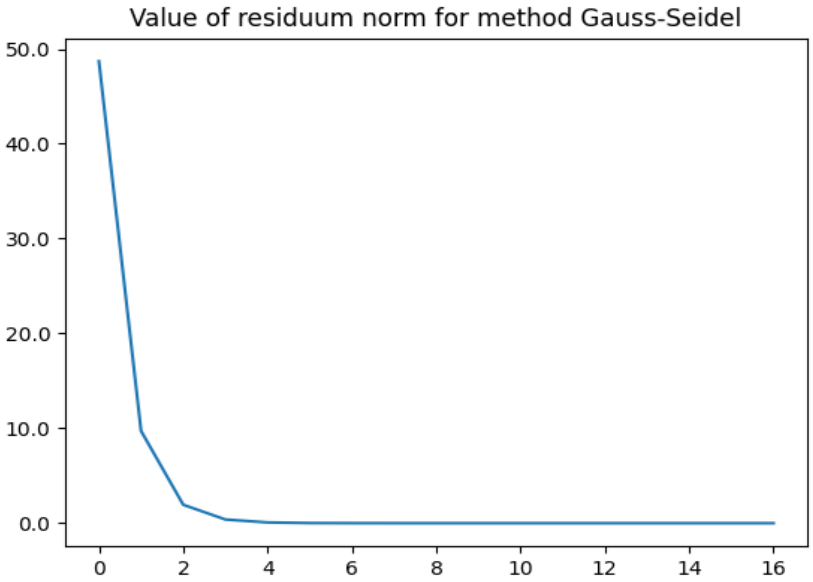
- macierz nr 2 , gdzie a1 = 3, oraz a2 = a3 = -1

# II Metody iteracyjne - macierz nr 1

1. Metoda Jacobiego potrzebowała 24 iteracji, aby osiągnąć wymagany poziom normy błędu rezydualnego. Trwało to 8.52 s. Czas ten jest około 10 – krotnie dłuższy od analogicznej operacji przeprowadzonej w matlabie. Jest to spowodowane specyficzną budową języka Python, w którym to operujemy na listach zamiast na tablicach. Powoduje to znacząco dłuższy czas dostępu do danych, co wpływa na negatywnie na czas trwanoia algorytmu.

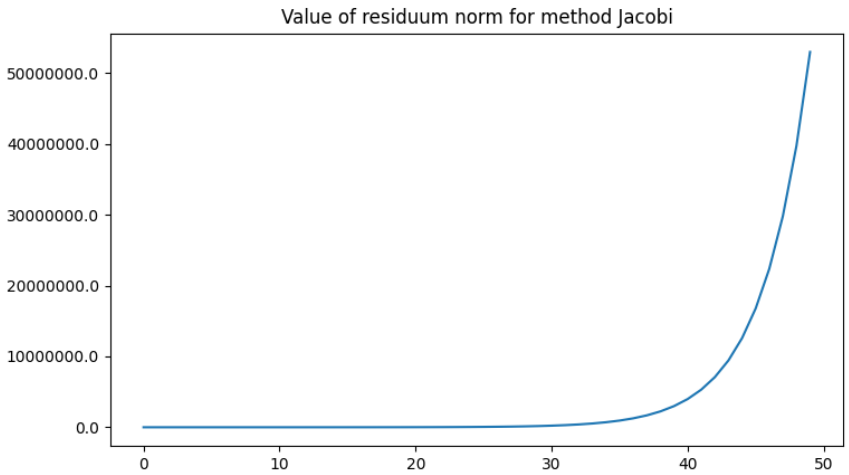


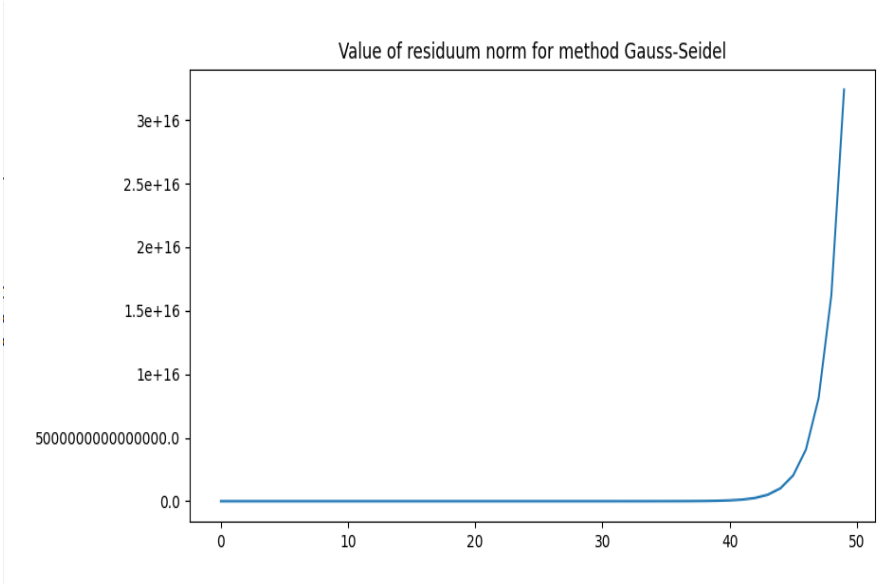
2. Metoda Gaussa-Seidla potrzebowała 17 iteracji, aby osiągnąć wymagany poziom normy błędu rezydualnego. Trwało to 7.59 s. Można zauważyć, że metoda ta jest bardziej wydajna od metody Jacobiego zarówno pod względem czasu działania jak i liczby iteracji.
   Warto też zaznaczyć, że metoda Gaussa-Seidla uzyskała początkowy błąd rezydualny o około 26% niższy niż metoda Jacobiego.




## III Metody iteracyjne - macierz nr 2

Dla macierzy nr 2 zarówno metoda Gaussa-Seidla jak i metoda Jacobiego nie zbiegły się. Norma z błędu rezydualnego w obydwu przypadkach dąży do nieskończoności. Poniższe wykresy przedstawiają wartość normy z błędu rezydualnego dla pierwszych 50 iteracji obydwu algorytmów.






# IV Metoda bezpośrednia – macierz nr 2

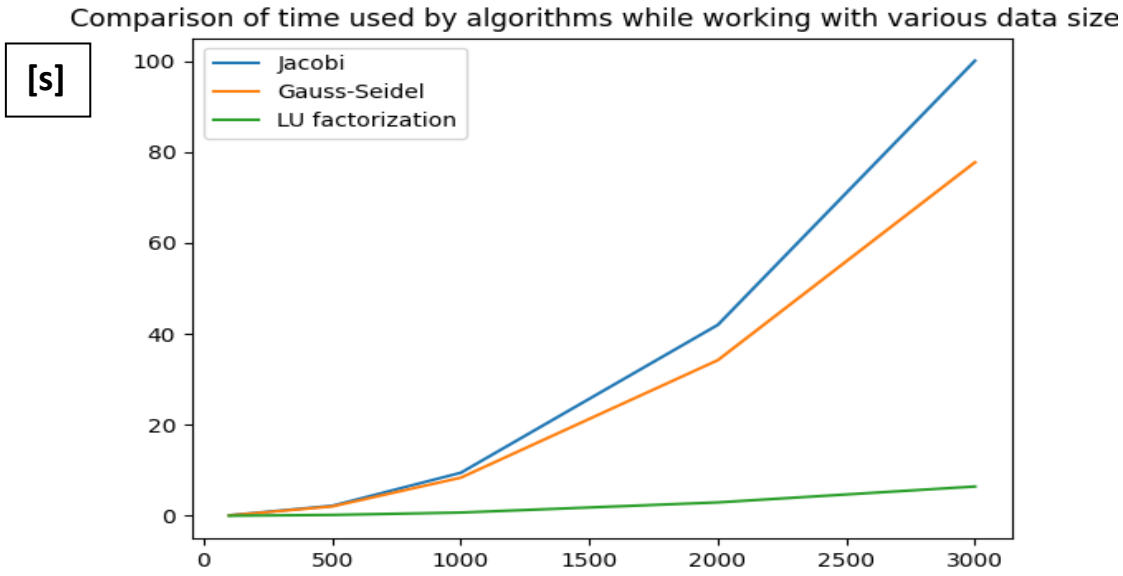
Używając metody faktoryzacji LU uzyskujemy normę z błędu resydualnego na poziomie 6.187846782787582e-13, zatem metoda charakteryzuje się dużą dokładnością obliczeń.

# V Zależność czasowa od rozmiaru macierzy oraz obserwacje

W celu porównania wyżej omawianych metod dla różnych rodzajów macierzy przeprowadzono próby na macierzach o rozmiarach 100x100, 500x500, 1000x1000, 2000x2000 i 3000x3000. Dla kązdego rozmiaru macierzy policzono wynik metodą Jacobiego, Gaussa-Seidla i faktoryzacją LU. Jak widać na wykresie metoda Jacobiego jest wolniejsza od metody Gaussa-Seidla dla wszystkich rozmiarów macierzy. Co może wydawać się na pierwszy rzut oka nietypowe, to niezwykle dobre wyniki faktoryzacji LU, o teoretycznej złożoności O(n^3). Powodem jest zaimplementowanie przeze mnie specjalnej wersji faktoryzacji LU dostosowanej do operacji na tzw. macierzach pasmowych (banded matrices). Pozwoliło to uzyskać złożoność O(a*n^2), gdzie a to ilość zapełnionych przekątnych poza główną. Jak zatem widać dobrze przystosowana prostsza metoda może mieć znacząco lepsze parametry od metod bardziej zaawansowanych przystosowanych do przypadków ogólnych.

Klasyczna faktoryzacja LU potrzebowała by około N/(a+1) (N -> rozmiar macierzy) raza więcej kroków niż wersja advanced, co dla macierzy 3000x3000 oznaczałoby 600 razy więcej operacji. Zatem z czasu około 8s koniec końców wyszłoby około 4800 s, czyli około 80 minut. Można zatem stwierdzić, że faktoryzacja LU nie jest optymalną metodą rozwiązywania dla bardzo dużych macierzy.

Ponadto, w celu usprawnienia metod itearcyjnych warto skorzystać ich charakterystyki i przeprowadzić dla nich operacje przeznaczone dla macierzy rzadkich. Inną opcją byłoby użycie biblioteki NumPy, która różni się niskopoziomową implementacją od standardowego Pythona, co pozwoliło by skrócić czas odwoływania do pamięci do minimum.




**made by Piotr Michalski, 184725**

**09.04.2023**