

Układy równań liniowych

Projekt 2 - Metody Numeryczne

Marcin Bajkowski, 193696, 28.04.2024r.

1. Wstęp teoretyczny

Celem projektu jest implementacja i analiza dwóch metod iteracyjnych (Jacobiiego i Gaussa-Seidla) oraz jednej metody bezpośredniej (faktoryzacja LU) rozwiązywania układów równań liniowych. Testy poszczególnych metod będą przeprowadzane na macierzach o różnych rozmiarach, które mogą powstać w wyniku dyskretyzacji równań różniczkowych i są powszechnie stosowane w takich zagadnieniach jak: elektronika, elektrodynamika, mechanika (zastosowania lotnicze, biomechanika, motoryzacja), badanie wytrzymałości materiałów i konstrukcji, symulacje odkształceń, naprężeń, przemieszczeń i drgań, akustyka, fotonika, termodynamika, dynamika płynów i wiele innych.

Metody te zostały zaimplementowane w języku Python z wykorzystaniem własnoręcznie napisanej klasy **Matrix** reprezentującej macierz oraz bibliotek: **time** do pomiaru czasu wykonywania się poszczególnych metod, **matplotlib** do wizualizacji wyników, **math** do obliczeń matematycznych, **os** do operacji na plikach i folderach oraz **copy** do kopiowania obiektów. W dalszej części sprawozdania zostaną przedstawione wyniki przeprowadzonych testów wydajnościowych, które pozwolą na porównanie zaimplementowanych metod pod względem szybkości oraz dokładności rozwiązywania układów równań liniowych. Dodatkowo każda z metod została przetestowana na macierzach o różnych rozmiarach, aby sprawdzić jak zachowuje się w zależności od wielkości danych wejściowych.

2. Wstęp praktyczny (metody)

Wszystkie wspomniane metody służą do rozwiązywania układów równań liniowych postaci $Ax = b$, gdzie A jest macierzą współczynników, x wektorem niewiadomych, a b wektorem wyrazów wolnych. W przypadku metody Jacobiiego, rozwiązanie układu równań liniowych polega na iteracyjnym przybliżaniu wartości wektora x na podstawie rekurencyjnego równania:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} * x_j^{(k)} \right), i = 1, 2, \dots, n$$

Algorytm kończy się, gdy spełniony jest warunek zbieżności, czyli gdy norma błędu jest mniejsza od zadanego epsilon. Metoda ta nie jest jednak zawsze zbieżna, więc nie gwarantuje znalezienia rozwiązania. W takim przypadku należy zastosować inną metodę iteracyjną, np. Gaussa-Seidla. W tej metodzie wartość wektora x jest obliczana na podstawie równania:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j < i} a_{ij} * x_j^{(k+1)} - \sum_{j > i} a_{ij} * x_j^{(k)} \right), i = 1, 2, \dots, n$$

Metoda Gaussa-Seidla jest zbieżna dla macierzy symetrycznych i dodatnio określonych. W przeciwnym przypadku może nie zwrócić poprawnego wyniku. Warto zauważyć, że metoda ta jest mniej kosztowna obliczeniowo niż metoda Jacobiiego, ponieważ w każdej iteracji korzysta z poprawionych wartości wektora x .

Ostatnią rozpatrywaną metodą jest faktoryzacja LU, która polega na zdekomponowaniu macierzy A na iloczyn dwóch macierzy trójkątnych: dolnej L i górnej U . Następnie rozwiązanie układu równań liniowych sprowadza się do rozwiązania dwóch układów równań liniowych z macierzami trójkątnymi. Metoda ta jest dokładna, ale wymaga więcej pamięci i czasu obliczeniowego niż metody iteracyjne. Do rozwiązania wykorzystuje się dwa wzory rekurencyjne:

$$L_{ii} * y_i = b_i - \sum_{j=1}^{(i-1)} L_{ij} * y_j, i = 1, 2, \dots, n$$

$$U_{ii} * x_i = y_i - \sum_{j=i+1}^n U_{ij} * x_j, i = n, n-1, \dots, 1$$

3. Dane wejściowe pierwszego zestawu danych

Zgodnie z treścią Zadania A zaimplementowana została macierz A o wymiarach $N \times N$ dla indeksu 193696. Dane pozwalające wyliczyć konieczne wartości dla macierzy A oraz wektora wyrazów wolnych b :

- c - przedostatnia cyfra numeru indeksu ($c = 9$),
- d - ostatnia cyfra numeru indeksu ($d = 6$),
- e - czwarta cyfra numeru indeksu ($e = 6$),
- f - trzecia cyfra numeru indeksu ($f = 3$),
- $a1 = 5 + e$ - współczynniki macierzy A ($a1 = 11$),
- $a2 = a3 = -1$ - współczynniki macierzy A ,

Rozmiar macierzy definiowany jest wzorem $N = 9cd$ co w przypadku indeksu 193696 daje $N = 996$. Z kolei wektor wyrazów wolnych b o długości N dla n -tego elementu jest równy $\sin(n \cdot (f + 1)) = \sin(n)$.

$$A_{996 \times 996} = \begin{pmatrix} 11 & -1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 11 & -1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 11 & -1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 11 & -1 & -1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 11 & -1 & -1 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 11 & -1 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 11 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 11 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & 11 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & -1 & 11 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & -1 & -1 & 11 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & -1 & 11 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & -1 & 11 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & -1 & 11 \end{pmatrix}$$

$$b_{996} = \begin{pmatrix} -0.7568 \\ 0.9894 \\ -0.5366 \\ -0.2879 \\ 0.9129 \\ -0.9056 \\ 0.2709 \\ \vdots \\ 0.9997 \\ -0.6365 \\ -0.1677 \\ 0.8557 \\ -0.951 \\ 0.3875 \\ 0.4444 \end{pmatrix}$$

4. Rezultaty metod rozwiązywania układów równań liniowych dla macierzy zdefiniowanej w zadaniu A

Zadania B polegało na zaimplementowaniu metod iteracyjnych rozwiązywania układów równań liniowych: Jacobiego i Gaussa-Seidla (uzupełnione o Zadanie D) oraz porównanie ich efektywności pod względem liczby iteracji potrzebnych do uzyskania rozwiązania o normie residuum mniejszej niż zadanego epsilon. Wartość epsilon została ustalona na 10^{-9} . Wyniki prezentują się następująco:

Metoda	Liczba iteracji	Czas obliczeń [s]	Norma residuum
Jacobi	25	6.7150	$8.278375316734148 \times 10^{-10}$
Gauss-Seidel	17	3.8724	$3.8123459240370307 \times 10^{-10}$
Faktoryzacja LU	—	17.6411	$2.34135805584451 \times 10^{-15}$

Table 1: Porównanie norm residuum dla metod iteracyjnych i bezpośredniej

Wyniki testów pokazują, że metoda Gaussa-Seidla jest szybsza (3.8724s) i dokładniejsza (17 iteracji) od metody Jacobiego (6.7150s dla 25 iteracji), co jest zgodne z oczekiwaniami. Różnicę w czasie obliczeń wynosi 2.8426s, co jest znaczącą różnicą dla tak krótkich czasów. Z kolei metoda faktoryzacji LU okazała się najdokładniejsza ($2.34135805584451 \times 10^{-15}$), ale jednocześnie najwolniejsza (17.6411s), co wynika z konieczności faktoryzacji macierzy A na macierze L i U .

Na podstawie tabeli można stwierdzić, że metoda Gaussa-Seidla jest najlepszym wyborem dla rozwiązywania układów równań liniowych, ponieważ jest zarówno szybsza, jak i dokładniejsza od metody Jacobiego. Metoda faktoryzacji LU jest dokładna, ale kosztowna obliczeniowo, dlatego nie jest zalecana dla dużych macierzy.

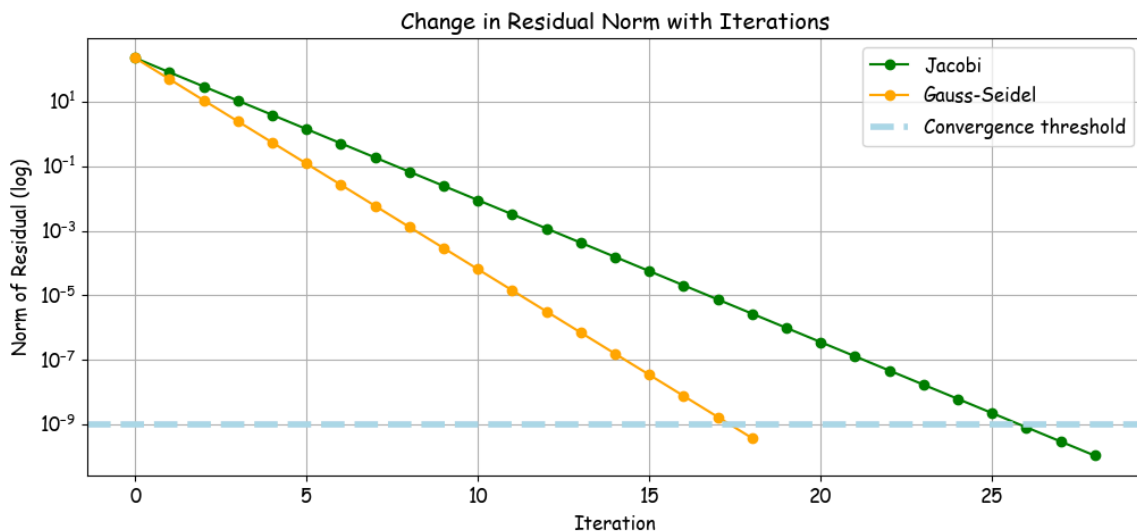


Figure 1: Porównanie norm residuum dla metod iteracyjnych

Wykres przedstawia porównanie norm residuum dla metod iteracyjnych: Jacobiego i Gaussa-Seidla. Jak widać, metoda Gaussa-Seidla osiąga mniejszą normę residuum w krótszym czasie niż metoda Jacobiego, co potwierdza wyniki z tabeli.

5. Dane wejściowe drugiego zestawu danych

Czy metody iteracyjne dla takich wartości elementów macierzy A zbiegają się? Dla obu metod przedstaw na wykresie jak zmienia się norma residuum w kolejnych iteracjach (oś y w skali logarytmicznej).

Zadanie C polegało na stworzeniu układu równań z nowymi wartościami współczynników macierzy A ($a_1 = 3$, $a_2 = a_3 = -1$) oraz obliczeniu wektora wyrazów wolnych b zgodnie z treścią Zadania A.

$$A_{996 \times 996} = \begin{pmatrix} 3 & -1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 3 & -1 & -1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 3 & -1 & -1 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 3 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & -1 & -1 & 3 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & -1 & 3 \end{pmatrix} b_{996} = \begin{pmatrix} -0.7568 \\ 0.9894 \\ -0.5366 \\ -0.2879 \\ 0.9129 \\ -0.9056 \\ 0.2709 \\ \vdots \\ 0.9997 \\ -0.6365 \\ -0.1677 \\ 0.8557 \\ -0.951 \\ 0.3875 \\ 0.4444 \end{pmatrix}$$

6. Rezultaty metod rozwiązywania układów równań liniowych dla macierzy zdefiniowanej w zadaniu C

Zadanie C polegało na zastosowaniu wcześniej zaimplementowanych metod iteracyjnych do rozwiązania układu równań liniowych z nowymi wartościami współczynników macierzy A ($a_1 = 3$, $a_2 = a_3 = -1$) oraz obliczeniu wektora wyrazów wolnych b zgodnie z treścią Zadania A. Zadanie zostało uzupełnione dodatkowo o wyniki dla metody faktoryzacji LU (Zadanie D). Poniżej przedstawiono porównanie ich efektywności pod względem liczby iteracji potrzebnych do uzyskania rozwiązania o normie residuum mniejszej niż zadanego epsilon oraz czasu obliczeń. Wartość epsilon pozostała niezmienna (10^{-9}), a limit błędu został ustalony na 10^9 .

Wyniki po wprowadzonych zmianach prezentują się następująco:

Metoda	Liczba iteracji	Czas obliczeń [s]	Norma residuum	Zbieżność
Jacobi	60	15.9803	1301276180.7303445	×
Gauss-Seidel	24	5.8992	1029036099.3947304	×
Faktoryzacja LU	—	17.6019	$1.9769795577221488 \times 10^{-13}$	—

Table 2: Porównanie norm residuum dla metod iteracyjnych i bezpośredniej

Wyniki testów pokazują, że metoda Gaussa-Seidla jest szybsza (5.8992s) i dokładniejsza (24 iteracje) od metody Jacobiego (15.9803s dla 60 iteracji), co jest zgodne z oczekiwaniami. Różnica w czasie obliczeń wynosi 9.0811s, co jest ogromną różnicą dla tak krótkich czasów. Metoda faktoryzacji LU okazała się najdokładniejsza ($1.9769795577221488 \times 10^{-13}$), ale jednocześnie najwolniejsza (17.6019s), co wynika z konieczności faktoryzacji macierzy A na macierze L i U . Co więcej, Jacobi okazał się być szybszy od faktoryzacji LU o tylko 1.3784s, co jest niewielką różnicą w porównaniu do Gaussa-Seidla.

Warty uwagi jest fakt, iż Jacobi i Gauss-Seidel nie zbiegają się dla nowych wartości współczynników macierzy A w porównaniu do poprzednich wartości zdefiniowanych w Zadaniu A.

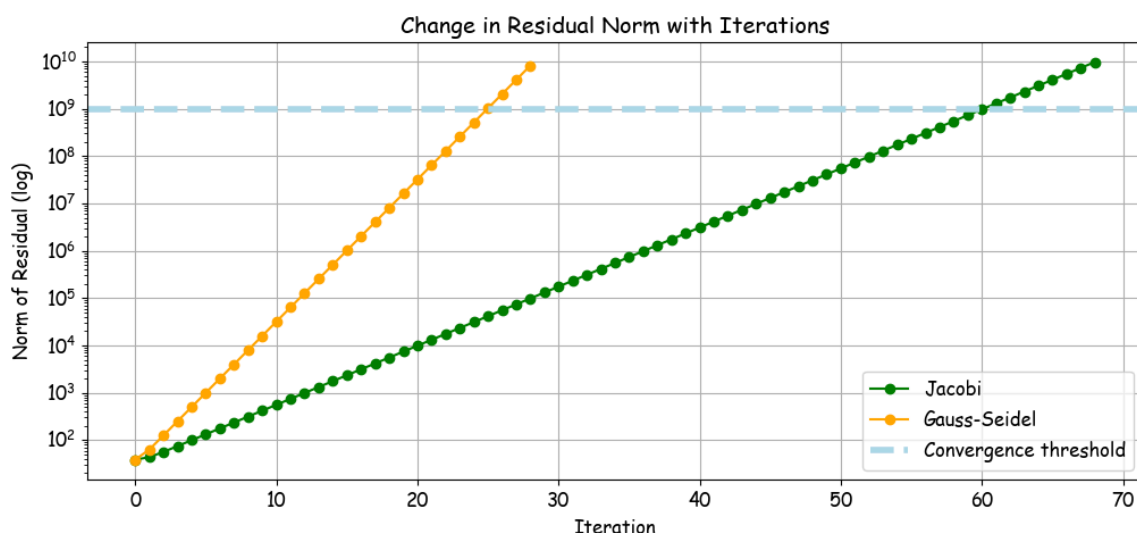


Figure 2: Porównanie norm residuum dla metod iteracyjnych

Wykres przedstawia porównanie norm residuum dla metod iteracyjnych: Jacobiego i Gaussa-Seidla. Jak widać, metoda Gaussa-Seidla osiąga mniejszą normę residuum w krótszym czasie niż metoda Jacobiego, co potwierdza wyniki z tabeli.

7. Porównanie czasów obliczeń dla metod rozwiązywania układów równań liniowych

Zadanie E polegało na utworzeniu wykresu prezentującego, jak zmienia się czas potrzebny na wyznaczenie rozwiązania dla trzech badanych metod (metoda Jacobiego, metoda Gaussa-Seidla oraz faktoryzacja LU) w zależności od liczby niewiadomych N . Przyjęto, że wartości N miały przyjmować kolejne wartości z sekwencji $\{100, 500, 1000, 2000, 3000\}$, które odzwierciedlały rosnący rozmiar macierzy opisanej w Zadaniu A.

To zadanie pozwala na zrozumienie, jak skalowanie rozmiaru macierzy wpływa na czas wykonania każdej z badanych metod, co jest istotne w kontekście optymalizacji wydajności obliczeniowej. Co więcej pozwala na przyszły dobór odpowiedniej metody w zależności od rozmiaru danych wejściowych, problemu do rozwiązania, dostępnych zasobów obliczeniowych, wymagań dotyczących dokładności rozwiązania oraz czasu obliczeń.

Wyniki dla poszczególnych metod przedstawiono w tabeli poniżej:

Rozmiar macierzy	Jacobi [s]	Gauss-Seidel [s]	Faktoryzacja LU [s]
100	0.0617	0.0392	0.0181
500	1.5391	0.9294	2.1670
1000	6.6282	3.8385	17.7077
1500	15.3376	9.2717	60.4937
2000	27.6398	16.6547	144.8885
2500	43.4565	26.2444	281.8669
3000	62.7243	37.5745	485.6950

Table 3: Porównanie czasów obliczeń dla metod rozwiązywania układów równań liniowych

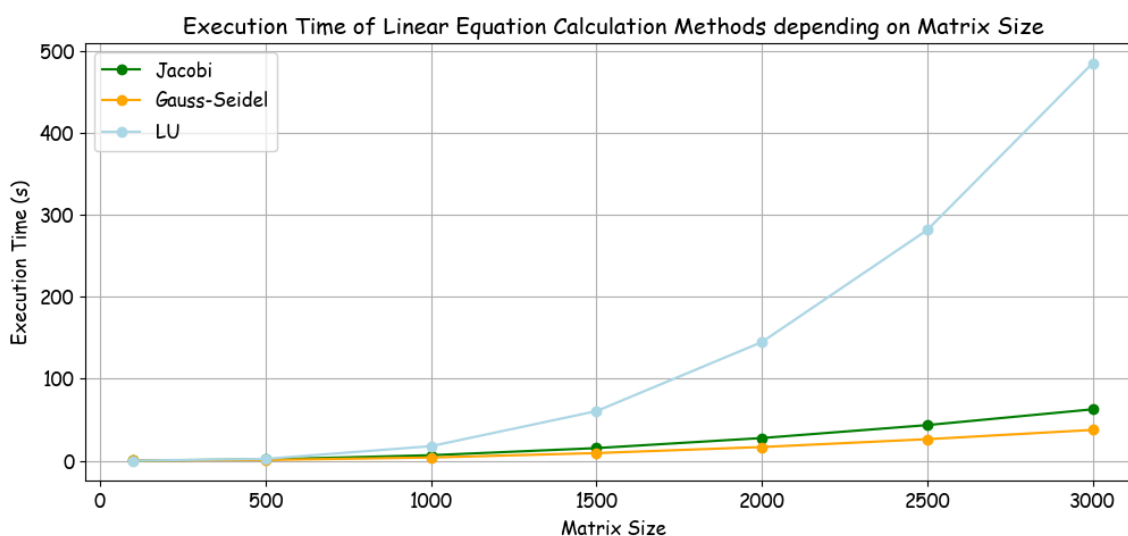


Figure 3: Porównanie czasów obliczeń dla metod rozwiązywania układów równań liniowych

Wyniki testów pokazują, że metoda Gaussa-Seidla jest najszybsza dla wszystkich rozmiarów macierzy, co jest zgodne z wynikami z poprzednich testów. Metoda Jacobiego jest wolniejsza od metody Gaussa-Seidla, ale szybsza od faktoryzacji LU. W przypadku macierzy o rozmiarze 100×100 czasy obliczeń wynoszą odpowiednio: 0.0617s dla metody Jacobiego, 0.0392s dla metody Gaussa-Seidla i 0.0181s dla metody faktoryzacji LU. Dla macierzy o rozmiarze 3000×3000 czasy obliczeń wynoszą odpowiednio: 62.7243s dla metody Jacobiego, 37.5745s dla metody Gaussa-Seidla i 485.6950s dla metody faktoryzacji LU. Porównując czas potrzebny na wyznaczenie rozwiązania dla najmniejszej, jak i największej macierzy, można zauważyć, że czas obliczeń rośnie wraz z rozmiarem macierzy dla każdej z metod, co jest zgodne z oczekiwaniami. Różnice w czasie obliczeń dla najmniejszej macierzy wynosi 0.0211s na korzyść metody faktoryzacji LU w porównaniu z metodą Gaussa-Seidla, co jest niewielką różnicą. Natomiast dla największej macierzy różnica w czasie obliczeń wynosi 448.1205s na korzyść metody Gaussa-Seidla w porównaniu z metodą faktoryzacji LU, co jest znaczącą różnicą dla tak dużych czasów. Porównując metody Jacobiego i Gaussa-Seidla można zauważyć, że czas obliczeń rośnie liniowo wraz z rozmiarem macierzy i jest on większy średnio o 65% w stosunku do metody Jacobiego. Natomiast metoda faktoryzacji LU ma złożoność obliczeniową $O(n^3)$, co sprawia, że czas obliczeń rośnie kwadratowo wraz z rozmiarem macierzy.

8. Podsumowanie

Celem projektu było zaimplementowanie i analiza dwóch metod iteracyjnych (Jacobiego i Gaussa-Seidla) oraz jednej metody bezpośredniej (faktoryzacja LU) rozwiązywania układów równań liniowych. Testy poszczególnych metod zostały przeprowadzone na macierzach o różnych rozmiarach, aby porównać ich wydajność pod względem szybkości oraz dokładności rozwiązywania układów równań liniowych.

Wyniki testów pozwoliły na stwierdzenie, że metoda Gaussa-Seidla jest najszybsza i najdokładniejsza dla rozwiązywania układów równań liniowych. Metoda Jacobiego jest wolniejsza od metody Gaussa-Seidla, ale szybsza od faktoryzacji LU. Metoda faktoryzacji LU jest najdokładniejsza, ale najwolniejsza, co wynika z konieczności faktoryzacji macierzy A na macierze L i U .

W zależności od rozmiaru macierzy i wymagań dotyczących dokładności rozwiązania można wybrać odpowiednią metodę, aby zoptymalizować czas obliczeń. W przypadku metody Jacobiego i Gaussa-Seidla czas obliczeń rośnie liniowo wraz z rozmiarem macierzy, co jest związane z koniecznością wykonania większej liczby iteracji. Natomiast metoda faktoryzacji LU ma złożoność obliczeniową $O(n^3)$, co sprawia, że czas obliczeń rośnie kwadratowo wraz z rozmiarem macierzy. Porównując metody, można stwierdzić, że metoda Gaussa-Seidla jest najlepszym wyborem dla rozwiązywania układów równań liniowych, ponieważ jest zarówno szybsza, jak i dokładniejsza od metody Jacobiego. Metoda faktoryzacji LU jest dokładna, ale kosztowna obliczeniowo, dlatego nie jest zalecana dla dużych macierzy. W zależności od rozmiaru macierzy i wymagań dotyczących dokładności rozwiązania można wybrać odpowiednią metodę, aby zoptymalizować czas obliczeń.

Warto także zauważyć, że metody iteracyjne nie zawsze zbiegają się dla wszystkich wartości współczynników macierzy, co może prowadzić do braku rozwiązania.

W przypadku macierzy o ogromnych rozmiarach, takich jak 3000×3000 , metoda Gaussa-Seidla jest najszybsza i w porównaniu z metodą faktoryzacji LU osiąga znacząco lepsze wyniki. Różnice te mogą być nawet kilkunastukrotne w zależności od rozmiaru macierzy. Dla testowanych danych największa różnica wyniosła 448.1205s na korzyść metody Gaussa-Seidla w porównaniu z metodą faktoryzacji LU (tym samym metoda faktoryzacji LU była wolniejsza o 1192,5877%).

Warto zauważyć, że metody zostały napisane w języku Python z wykorzystaniem własnoręcznie napisanej klasy **Matrix** reprezentującej macierz. Wykorzystanie gotowych bibliotek takich jak **numpy** mogłoby przyspieszyć obliczenia, ale nie pozwoliłoby na pełne zrozumienie działania algorytmów. Innym sposobem na przyspieszenie obliczeń byłoby przepisanie kodu na przykład do języka C++, który jest znacznie szybszy od Pythona, ponieważ jest językiem kompilowanym, a nie interpretowanym. Dodatkowo każda funkcja w Pythonie jest obiektem, co wprowadza dodatkowe narzuty czasowe, które mogą wpłynąć na czas obliczeń.