

Metody numeryczne, projekt 2: Układy równań liniowych

Jakub Stachowicz
198302

29 kwietnia 2025

Spis rzeczy

1	Wstęp	3
1.1	Abstrakt	3
1.2	Analizowany układ równań	3
1.3	Wektor residuum	3
2	Algorytmy	3
2.1	Metody iteracyjne	3
2.1.1	Metoda Jacobiego	3
2.1.2	Metoda Gaussa-Seidla	4
2.2	Metoda bezpośrednia (faktoryzacja LU)	4
3	Analiza	4
3.1	Metody iteracyjne	4
3.2	Metoda bezpośrednia (faktoryzacja LU)	7
3.3	Analiza wydajności	7
4	Podsumowanie	8
5	Bibliografia	8

1 Wstęp

1.1 Abstrakt

Celem projektu jest implementacja i analiza trzech sposobów rozwiązywania układów równań liniowych, dwóch iteracyjnych (metoda Jacobiego i metoda Gaussa-Seidla) i jednej bezpośredniej (faktoryzacja LU).

1.2 Analizowany układ równań

W ogólności, układ równań liniowych ma następującą postać:

$$Ax = b, \quad (1)$$

gdzie A jest macierzą systemową, b jest wektorem pobudzenia, a x jest wektorem rozwiązań.

Macierz A , która została przyjęta do użycia przy przeprowadzaniu analizy powyższych algorytmów, stanowi macierz pasmowa rozmiaru $N \times N$ o następującej strukturze¹:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ a_2 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & \dots & 0 \\ a_3 & a_2 & a_1 & a_2 & a_3 & 0 & 0 & 0 & \dots & 0 \\ 0 & a_3 & a_2 & a_1 & a_2 & a_3 & 0 & 0 & \dots & 0 \\ 0 & 0 & a_3 & a_2 & a_1 & a_2 & a_3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & a_1 \end{bmatrix} \quad (2)$$

Jako wektor b , w tym samym celu, przyjęto wektor rozmiaru N o wartościach będących kolejnymi elementami ciągu zadanego wzorem:

$$c_n = \sin(n \cdot (f + 1)), \quad (3)$$

gdzie f stanowi trzecią cyfrę indeksu autora sprawozdania ($f = 8$).

1.3 Wektor residuum

W metodach iteracyjnych korzysta się z wektora residuum, który obliczany jest w sposób następujący:

$$r = Ax - b. \quad (4)$$

Wyznaczona norma euklidesowa tego wektora pozwala stwierdzić, czy wyniki kolejnych iteracji zbiegają się do rozwiązania. Pozwala ona także przerwać działania iteracyjne, jeśli uzyskany zostanie wynik w wyznaczonej tolerancji.

W dalszej części sprawozdania jako sygnał do zaprzestania obliczeń przyjęto normę residuum na poziomie mniejszym niż 10^{-9} .

2 Algorytmy

2.1 Metody iteracyjne

Metody iteracyjne wykorzystują rozbięcie macierzy A na trzy podmacierze: macierz trójkątną dolną L , trójkątną górną U i diagonalną D . Macierz A stanowi ich sumę.

$$A = L + U + D \quad (5)$$

2.1.1 Metoda Jacobiego

Po przekształceniu wzoru (5) i podstawieniu go do wzoru (1):

$$Dx = -(L + U)x + b \quad (6)$$

$$x = -D^{-1}(L + U)x + D^{-1}b, \quad (7)$$

¹Wartość N oraz elementów a_1, a_2 i a_3 zdefiniowane są w dalszej części opracowania.

otrzymujemy następujący wzór na oszacowanie wektora rozwiązań x w $(k + 1)$ -szej iteracji, na bazie wektora z k -tej iteracji.

$$x^{(k+1)} = -D^{-1}(L + U)x^{(k)} + D^{-1}b = M_J x^{(k)} + w_J. \quad (8)$$

Jako wektor początkowy przyjmuje się wektor długości N o zawartości jedynek.

2.1.2 Metoda Gaussa-Seidla

Po przekształceniach wzoru (5) i podstawieniu go do wzoru (1):

$$(D + L)x = -Ux + b \quad (9)$$

otrzymujemy następujący wzór na oszacowanie wektora rozwiązań x w $(k + 1)$ -szej iteracji, na bazie wektora z k -tej iteracji.

$$x^{(k+1)} = -(D + L)^{-1}(Ux^{(k)}) + -(D + L)^{-1}b = M_{GS}x^{(k)} + w_{GS}. \quad (10)$$

Jako wektor początkowy, podobnie jak w metodzie Jacobiego, przyjmuje się wektor długości N o zawartości jedynek.

2.2 Metoda bezpośrednia (faktoryzacja LU)

Metoda ta polega na utworzeniu takich macierzy trójkątnych L i U , żeby zachodziło równanie $LUx = b$.

Tworzy się wówczas wektor pomocniczy $y = Ux$, a następnie rozwiązuje układ równań $Ly = b$ za pomocą podstawiania wprzód, a następnie – układ $Ux = y$ za pomocą podstawiania wstecz.

Sednem tego algorytmu jest faktoryzacja macierzy A na macierze L i U . Można tego dokonać stosując poniższy algorytm (w Pythonie):

```
def LU_decomposition(A, m):
    U = copy.copy(A)
    L = np.eye(m)
    for i in range(2, m + 1):
        for j in range(1, i):
            L[i - 1, j - 1] = U[i - 1, j - 1] / U[j - 1, j - 1]
            U[i - 1, :] = U[i - 1, :] - L[i - 1, j - 1] * U[j - 1, :]
    return L, U
```

Taki też został zastosowany w dalszej analizie tego algorytmu.

3 Analiza

3.1 Metody iteracyjne

Pierwsza analiza

Jako rozmiar macierzy N przyjęto liczbę $1200 + 10c + d$, gdzie c jest przedostatnią cyfrą numeru indeksu autora, a d – ostatnią. Ostatecznie $N = 1202$. Jako a_2 i a_3 przyjęto -1 , a jako a_1 – liczbę o 5 większą od czwartej cyfry numeru indeksu autora. Ostatecznie $a_1 = 8$.

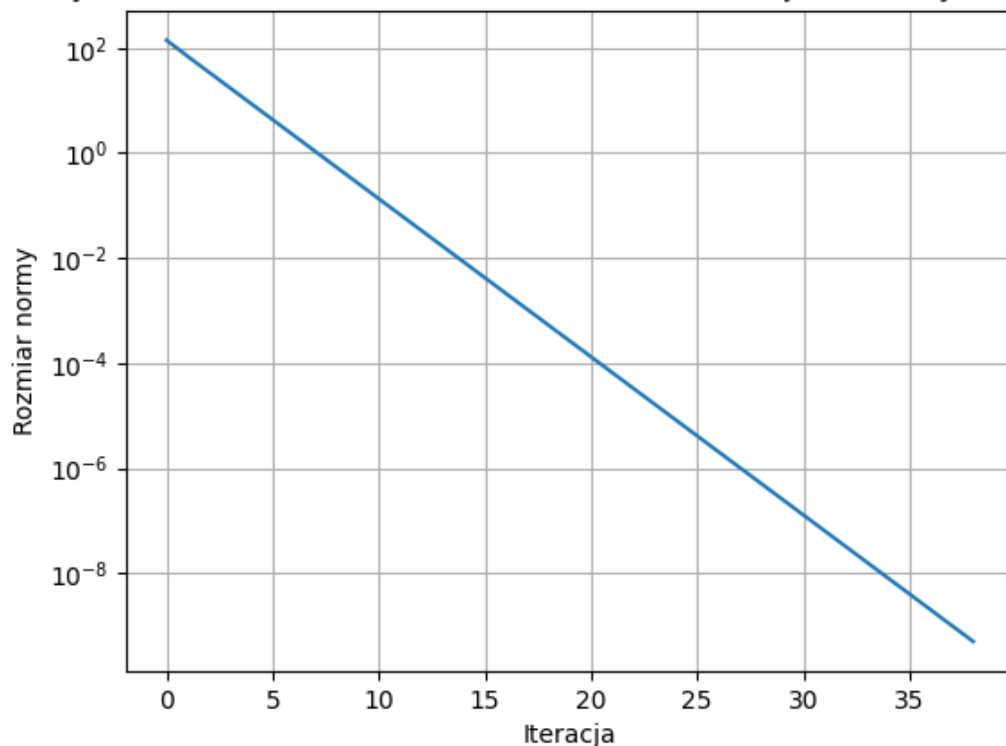
Zarówno metoda Jacobiego, jak i metoda Gaussa-Seidla poradziły sobie z rozwiązaniem zadanego układu równań.

Metoda Jacobiego okazała się szybsza czasowo (rozwiązanie podała w 202 ms w przeciwieństwie do 1431 ms), jednak dotarcie do zadowalającej odpowiedzi (tj. takiej, w której norma residuum wynosi mniej niż 10^{-9}) zajęło jej więcej iteracji (38, w przeciwieństwie do 24).

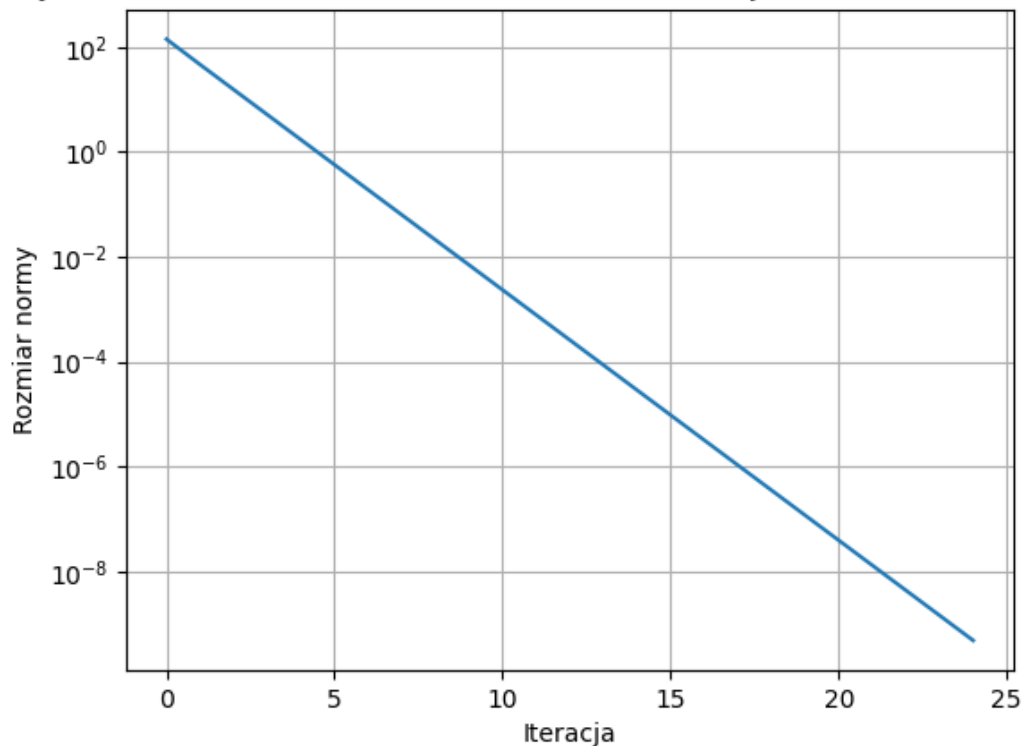
Obie te metody zbiegały się w podobny sposób – wykładniczo – co przedstawiają wykresy 1 i 2 (w skali logarytmicznej).

Metody iteracyjne nie są jednak uniwersalnym sposobem rozwiązywania układów równań liniowych, o czym świadczy druga analiza.

Wykres 1: Norma residuum w zależności od iteracji (Metoda Jacobiego)



Wykres 2: Norma residuum w zależności od iteracji (Metoda Gaussa-Seidla)



Druga analiza

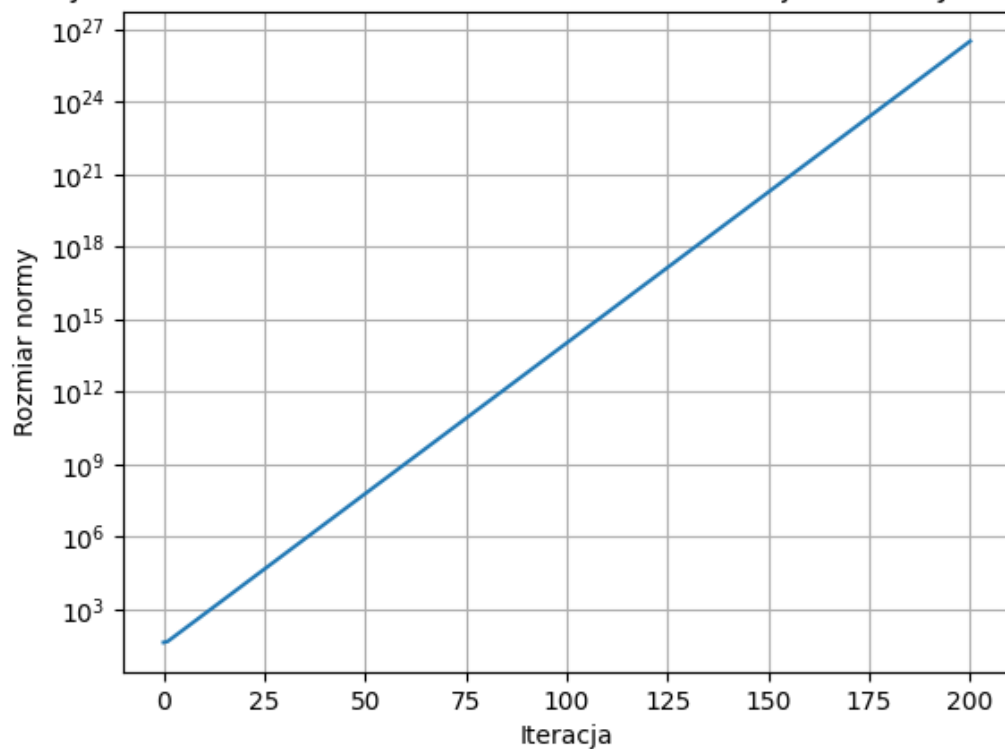
Niech N pozostanie takie jak w pierwszej analizie, podobnie a_2 i a_3 , ale niech a_1 wynosi 3.

Dla obu podanych metod, wraz ze wzrostem liczby iteracji, wykładniczo rośnie wartość normy residuum, co świadczy o rozbieżności. Ogromny rozmiar tejże rozbieżności zaobserwować można na wykresach

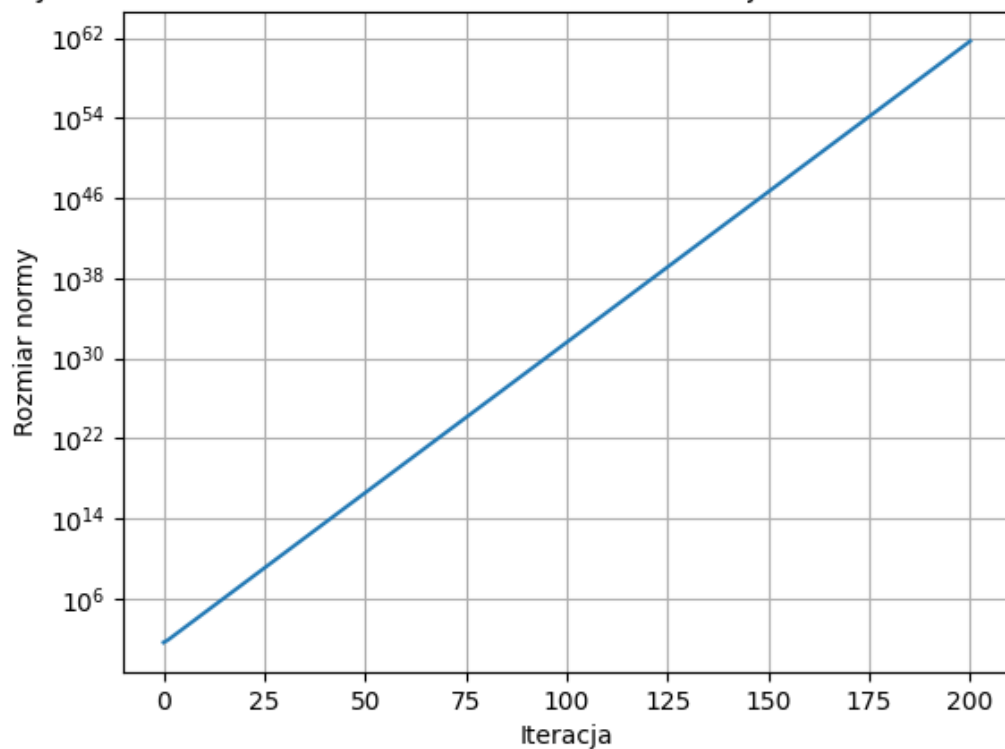
3 i 4 (w skali logarytmicznej).

Wykonywanie algorytmów zostaje przerwane po 200 iteracjach, jako przyjęty limit długości obliczeń w przypadku nieosiągnięcia warunku zadowalającej normy residuum.

Wykres 3: Norma residuum w zależności od iteracji (Metoda Jacobiego)



Wykres 4: Norma residuum w zależności od iteracji (Metoda Gaussa-Seidla)



3.2 Metoda bezpośrednia (faktoryzacja LU)

Do rozwiązania układu równań zastosowanego w drugiej analizie metod iteracyjnych zastosowana została również metoda bezpośrednia (faktoryzacja LU).

Obliczenia trwały 4468 ms, jednak metoda ta zwróciła poprawny wynik. Norma residuum wyniosła $1,174 \cdot 10^{-12}$.

Oznacza to, że ta metoda, choć będąca zwykle wolniejsza od metod iteracyjnych, ma większe szanse powodzenia znalezienia poprawnego rozwiązania, gdyż nie bazuje na pewnej heurystyce związanej ze stabilnością obliczeniową.

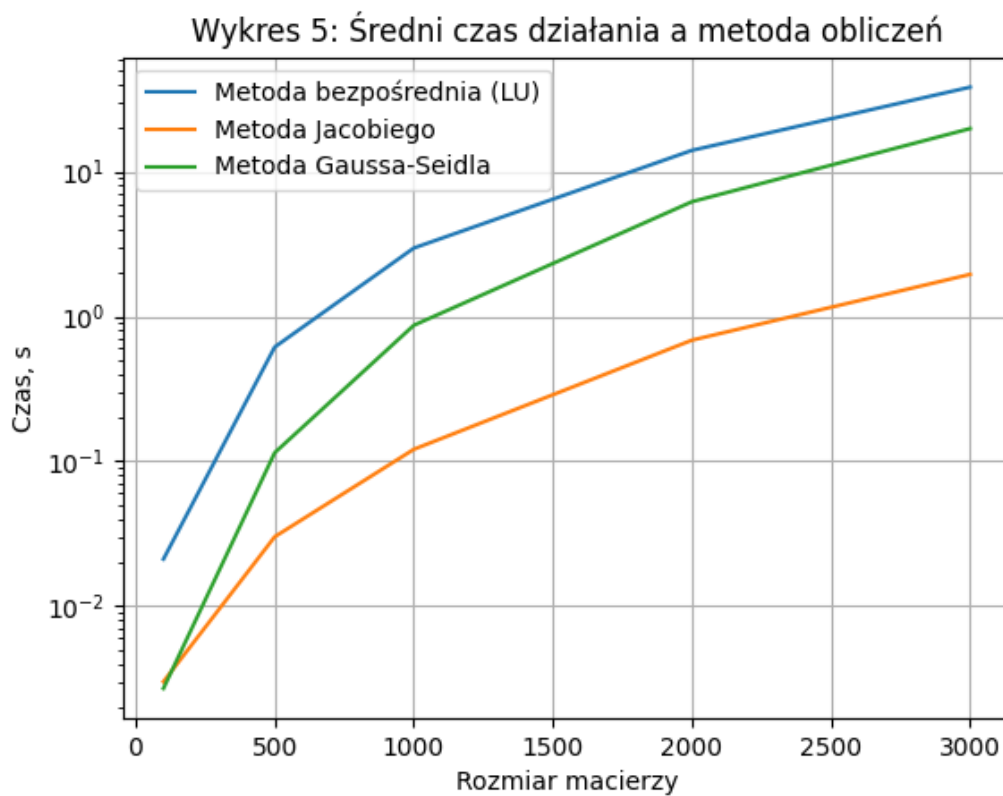
3.3 Analiza wydajności

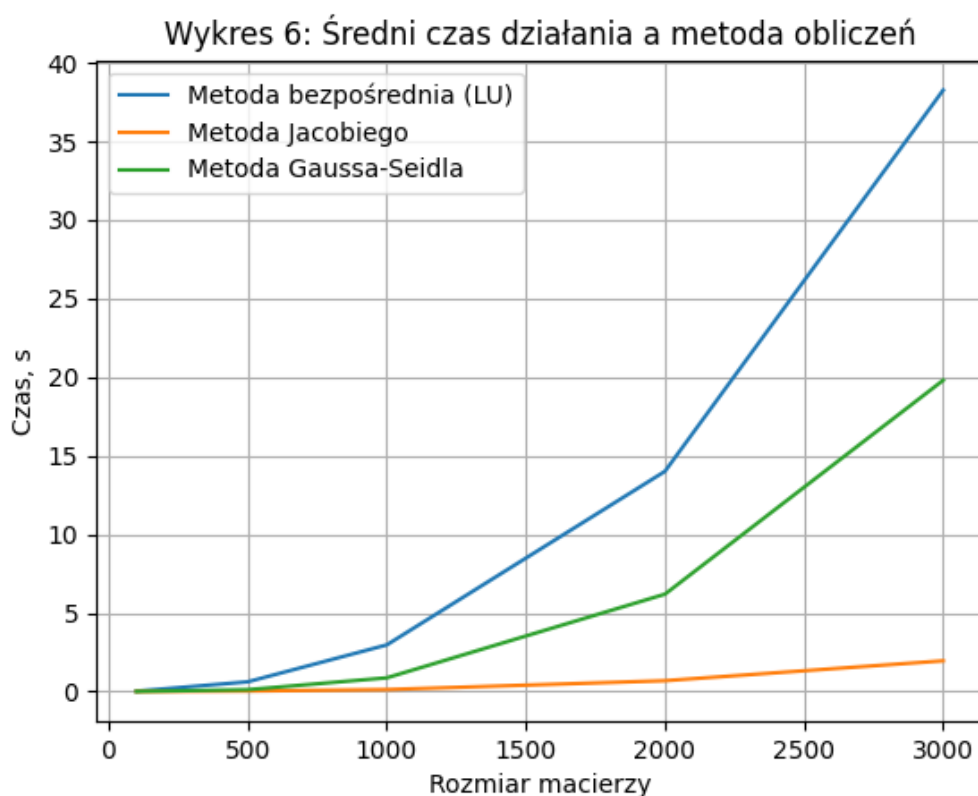
Podczas analizy wydajności wszystkich trzech metod skonstruowano ciąg macierzy o rozmiarach $N = 100, 500, 1000, 2000, 3000$. Parametry a_n zdefiniowano w ten sam sposób, co w pierwszej analizie metod iteracyjnych (sekcja 3.1).

Wszystkie te układy równań zostały poprawnie rozwiązane przez wszystkie trzy analizowane algorytmy.

Dla każdego rozmiaru macierzy badany algorytm uruchamiany był 10 razy, następnie obliczona średnia czasu wykonania została umieszczona na wykresach (5 – w skali logarytmicznej, 6 – w skali naturalnej).

Zaobserwować można silny wzrost czasu wykonania metody bezpośredniej. Metody iteracyjne są szybsze, zwłaszcza (na sprzecznie i oprogramowaniu autora sprawozdania) metoda Jacobiego.





4 Podsumowanie

Na podstawie przeprowadzonych analiz można stwierdzić, że metody iteracyjne (Jacobiego i Gaussa-Seidla) są znacznie szybsze w znajdowaniu rozwiązań układów równań niż metoda bezpośrednia (faktoryzacja LU). Czas działania wynika z ich złożoności obliczeń, która dla metody bezpośredniej wynosi $O(n^3)$, a w przypadku metod iteracyjnych – $O(n^2)$.

Metody iteracyjne jednak nie zawsze znajdują poprawne rozwiązanie wyniku, są przez to mniej uniwersalne. Warto jednak z nich skorzystać i poznać zwrócony przez nie wynik, wraz z normą. Nakład pracy jest znacznie mniejszy niż w przypadku metody bezpośredniej, więc warto jest zaryzykować, a następnie skorzystać z metody bezpośredniej tylko wówczas, jeśli z normy residuum wynika, iż rozwiązanie się nie zbiegło.

5 Bibliografia

1. Fotyga G., 2024: Wykład „*Układy Równań Liniowych, Metody Numeryczne*”.
2. Fotyga G., 2025: Instrukcja „*Metody Numeryczne, Projekt 2 – Układy równań liniowych*”.
3. Fotyga G., Sypek P., 2025: Instrukcja „*Metody numeryczne, laboratorium 3. Metody rozwiązywania równania macierzowego*”.