

Raport z MN2 z zadania 16

Jakub Bazyluk

28 kwietnia 2022

Spis treści

1	Wstęp	2
2	Metoda Doolittle’a	2
2.1	Blokowa metoda Doolittle’a	2
2.2	Rozkład Doolittle’a	2
2.3	Wyznaczanie rozwiązania	3
3	Uwagi	4
4	Przykłady	4
4.1	Czas wykonywania obliczeń	4
4.2	Błąd wykonywanych obliczeń	5
5	Wnioski	6

1 Wstęp

W ramach projektu z Metod Numerycznych utworzono algorytm rozwiązywania układów równań postaci $Ax = b$, gdzie

$$A = \begin{bmatrix} A_{11} & I \\ A_{21} & A_{22} \end{bmatrix}$$

$$A_{11}, A_{21}, A_{22} \in M_{n \times n}(\mathbb{R})$$

Algorytm oparty jest na blokowym rozkładzie Doolittle'a - $A = LU$, U macierz blokowa górnotrójkątna, L macierz blokowa dolnotrójkątna oraz na klasycznym rozkładzie Doolittle'a poszczególnych macierzy.

Metodę przetestowano pod względem wskaźnika poprawności oraz czasu wykonania dla różnych macierzy, odnośnikiem była wbudowana metoda MatLab'a. Algorytm okazał się zdecydowanie szybszy oraz skuteczniejszy, szczególnie dla źle uwarunkowanych macierzy.

2 Metoda Doolittle'a

2.1 Blokowa metoda Doolittle'a

Metoda pozwalająca rozłożyć macierz A do postaci

$$\begin{bmatrix} A_{11} & I \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} \begin{bmatrix} U_1 & U_2 \\ 0 & U_3 \end{bmatrix}$$

po wymnożeniu

$$\begin{bmatrix} A_{11} & I \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \\ LU_1 & LU_2 + U_3 \end{bmatrix}$$

Zatem

$$\begin{cases} U_1 = A_{11} \\ U_2 = I \\ U_3 = A_{22} - L \\ L = A_{21}U_1^{-1} \end{cases}$$

2.2 Rozkład Doolittle'a

Szukamy rozkładu $A = LU$ W rozkładzie Doolittle'a macierze L i U są już macierzami odpowiednio dolnotrójkątną i górnotrójkątną, przy czym zakładamy, że elementy leżące na diagonalu macierzy L są równe 1. Otrzymujemy układ n^2 równań:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

Po rozpisaniu iloczynu znajdującego się po prawej stronie powyższej równości możemy porównać odpowiadające sobie elementy macierzy A i LU . Robimy to naprzemiennie, tj. raz wyznaczamy wiersz macierzy U raz kolumnę macierzy L .

Wzory ogólne na poszczególne elementy macierzy rozkładu przedstawiają się następująco:

dla wszystkich $i \in \{1, 2, \dots, n\}$:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, \text{ dla } j \in \{i, i+1, \dots, n\}$$

$$l_{ji} = \frac{1}{u_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right), \text{ dla } j \in \{i+1, i+2, \dots, n\}$$

Z ostatniego równania wynika, że metoda nie zadziała, gdy $u_{ii} = 0$.

2.3 Wyznaczanie rozwiązania

Posiadając rozkład Doolittle'a, pozostaje rozwiązanie równania $Ax = b$. Mamy zatem $LUx = b$, przyjmijmy $y = Ux$ i zacznijmy od rowiązania $Ly = b$ w postaci macierzowej:

$$\begin{bmatrix} I & 0 \\ L & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Po rowiązaniu otrzymujemy:

$$\begin{cases} y_1 &= b_1 \\ Ly_1 + y_2 &= b_2 \end{cases}$$

Zatem

$$\begin{cases} y_1 &= b_1 \\ y_2 &= b_2 - Ly_1 \end{cases}$$

Następnie rozwiązujemy $Ux = y$:

$$\begin{bmatrix} U_1 & U_2 \\ 0 & U_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Zatem

$$\begin{cases} U_1 x_1 + U_2 x_2 &= y_1 \\ U_3 x_2 &= y_2 \end{cases}$$

Do rozwiązania układu $U_i x_j = y_k$ potrzebujemy rozłożyć macierz: $U_i = L_{U_i} U_{U_i}$, a następnie podstawić $y_{U_i} = U_{U_i} x_j$ oraz rozwiązać układ równań z macierzą dolnotrójkątną a następnie z górnątrojkątną element po elemencie.

3 Uwagi

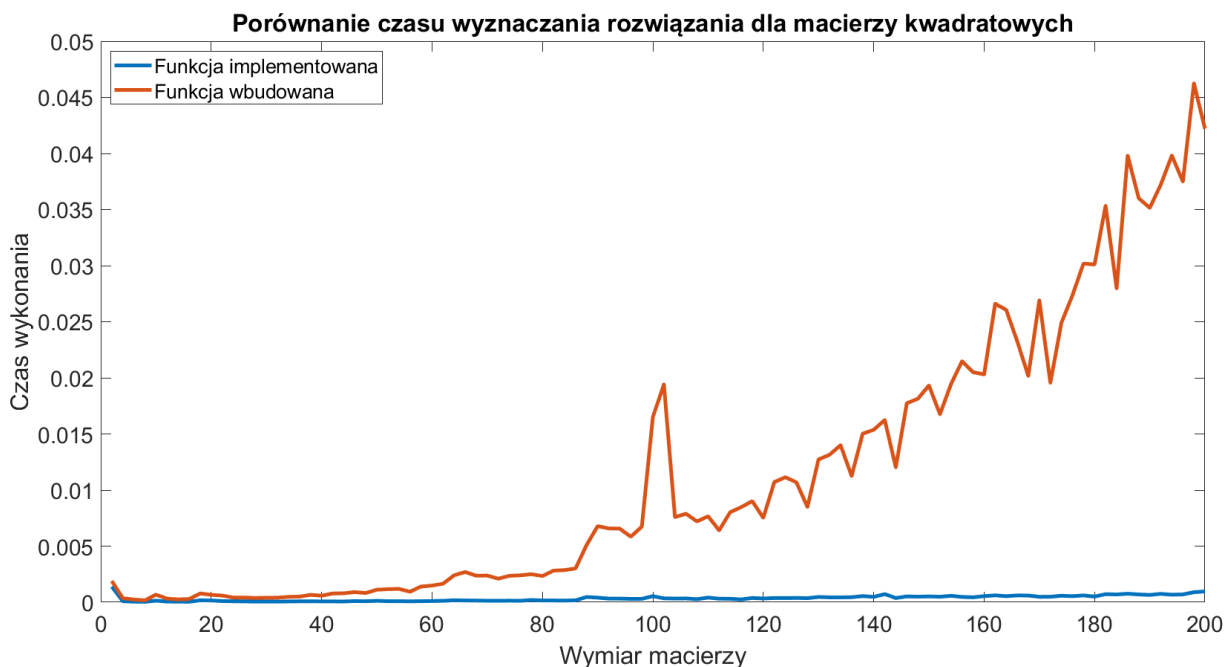
Algorytm służy do wyznaczania bardzo specyficznych układów równań, dzięki czemu jego złożoność obliczeniowa jest klasy $O(n^3)$. Algorytm nie zadziała kiedy rozkład LU lub rozkład blokowy LU nie będzie możliwy. Z tego względu macierze rzadkie nie są rozkładalne i należy skorzystać z innej metody. W szczególności należy mieć na uwadze, że macierz A_{11} musi być odwracalna.

4 Przykłady

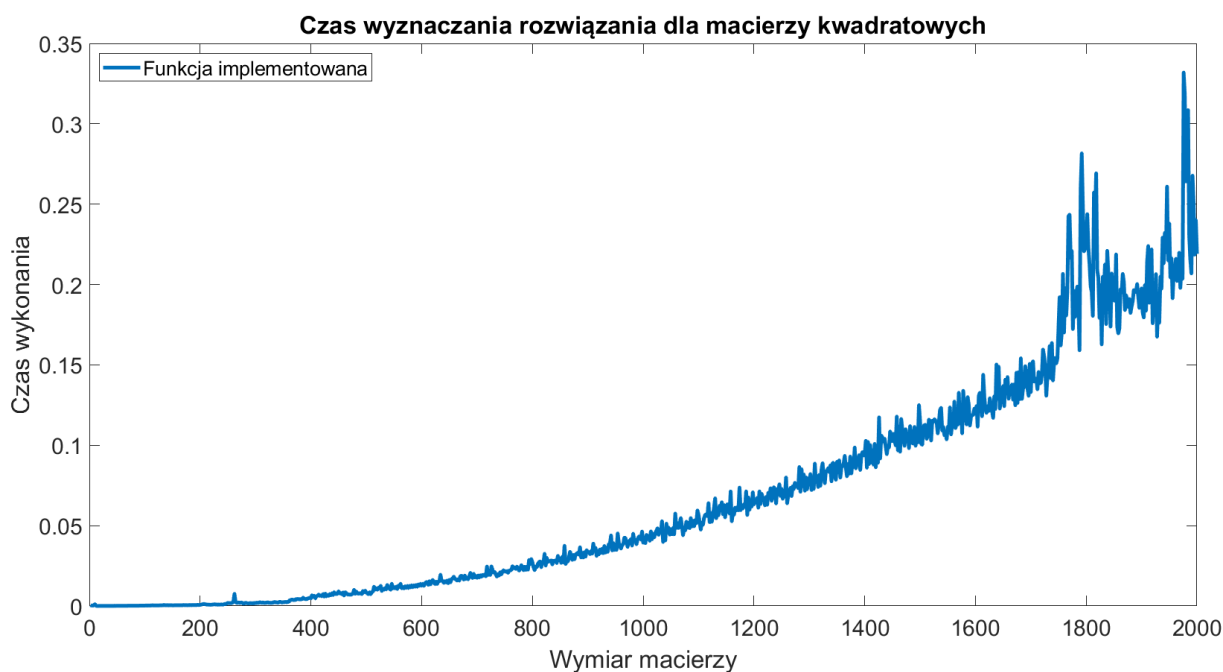
W ramach testów obliczeniowych przeprowadzono eksperymenty na losowych macierzach różnych rozmiarów i zbadano dla nich błąd obliczeń oraz czas potrzebny do ich wykonania. Wyniki zestawiono z wynikami analogicznych testów dla metody wbudowanej MatLab'a.

4.1 Czas wykonywania obliczeń

Porównanie metody implementowanej do metody wbudowanej względem czasu wykonywania obliczeń dla losowo wygenerowanych macierzy $A \in \mathbb{C}^{n \times n}$, dla $n = 1, 2, 3, \dots, 200$.



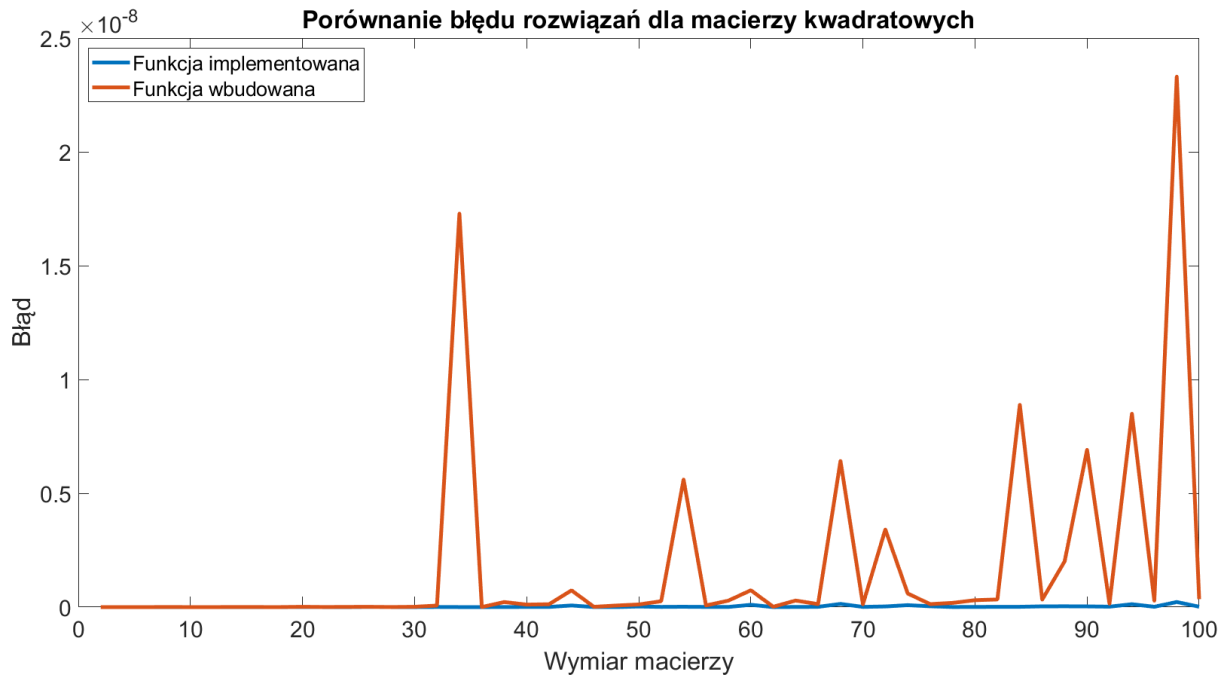
Powyższy wykres ilustruje przewagę implementowanej metody. Wzrost czasu potrzebnego do wykonania obliczeń dla metody wbudowanej jest wykładniczy i rośnie bardzo szybko już dla macierzy wymiaru 150 – 200, podczas gdy dla metody implementowanej wzrost jest właściwie niezauważalny. Aby przyjrzeć się dokładniej zmienności czasu wykonywania obliczeń dla metody implementowanej należy spojrzeć na osobny wykres dla znacznie większych macierzy.



Widzimy, że czas wykonywanie obliczeń znacząco rośnie, jednak nie jest to wzrost wykładniczy. Metoda implementowana nadaje się zatem do wyznaczania rozwiązań dla dużych macierzy. Nieregularności na wykresie dla macierzy wymiaru 1500–2000 można tłumaczyć ich złym uwarunkowaniem.

4.2 Błąd wykonywanych obliczeń

Porównanie metody implementowanej do metody wbudowanej względem błędu obliczeń dla losowo wygenerowanych macierzy $A \in \mathbb{C}^{n \times n}$, dla $n = 1, 2, 3, \dots, 100$.



Powyższy wykres dobitnie pokazuje przewagę implementowanej metody. Ze względu na nieporównywalną różnicę między błędami dla danych metod, na wykresie przedstawiono jedynie wyniki dla macierzy $A \in \mathbb{C}^{n \times n}$, dla $n = 1, 2, 3, \dots, 100$.

5 Wnioski

Podstawowym i kluczowym wnioskiem jest potwierdzenie ogromnej przewagi implementowanej metody względem zarówno czasu jak i błędu obliczeń. Wykresy przedstawione w przykładach ilustrują różnicę w wynikach implementowanej metody oraz metody wbudowanej MatLab'a już dla niewielkich rozmiarów macierzy. Metoda implementowana radzi sobie z bardzo dużymi macierzami w rozsądnym czasie, zachowując bardzo wysoką dokładność obliczeń. Jej ograniczeniem jest konkretny, sprecyzowany w projekcie format macierzy. W przypadku nie dostarczenia odpowiedniej macierzy metoda nie zadziała. Ważne jest również to, żeby istniały rozkłady Doolittle'a danych macierzy. Nie jest to niestety własność każdej macierzy kwadratowej, chociaż przy testach na losowych macierzach problem ten właściwie nie występował. Jeśli te ograniczenia są spełnione to metoda działa niezawodnie i szybko, nawet dla macierzy o bardzo złym uwarunkowaniu, co jest bardzo pożądaną cechą.

Literatura

- [1] Notatki do wykładu Metody Numeryczne 2021
- [2] https://pl.wikipedia.org/wiki/Metoda_LU