Lab6

May 25, 2022

1 Układy równań liniowych - metody bezpośrednie

1.1 Bartosz Kucharz

Dany jest układ równań liniowych $\mathbf{A}x$=$\mathbf{b}$.

2 Używana postać błędu obliczania x

$$\max_{i=1}^{n} |x_i - \bar{x}_i|$$

$$x - wektor\ zadany\ na\ pocztku$$

$$\bar{x} - wektor\ obliczony$$

3 Uwarunkowanie macierzy

Uwarunkowanie macierzy było obliczane za pomocą funkcji z biblioteki numpy (numpy.linalg.cond), która korzysta z rozkładu wedlug wartości osobliwych (SVD).

4 Zad 1

Elementy macierzy **A** o wymiarze $n \times n$ są określone wzorem:

$$\begin{cases} a_{1j} = 1 \\ a_{ij} = \dfrac{1}{i+j-1} \quad dla \quad i \neq 1 \end{cases} \qquad i, j = 1, \ldots, n$$

Przyjmij wektor $\boldsymbol{x}$ jako dowolną n–elementową permutację ze zbioru { 1, -1 } i oblicz wektor **b**.

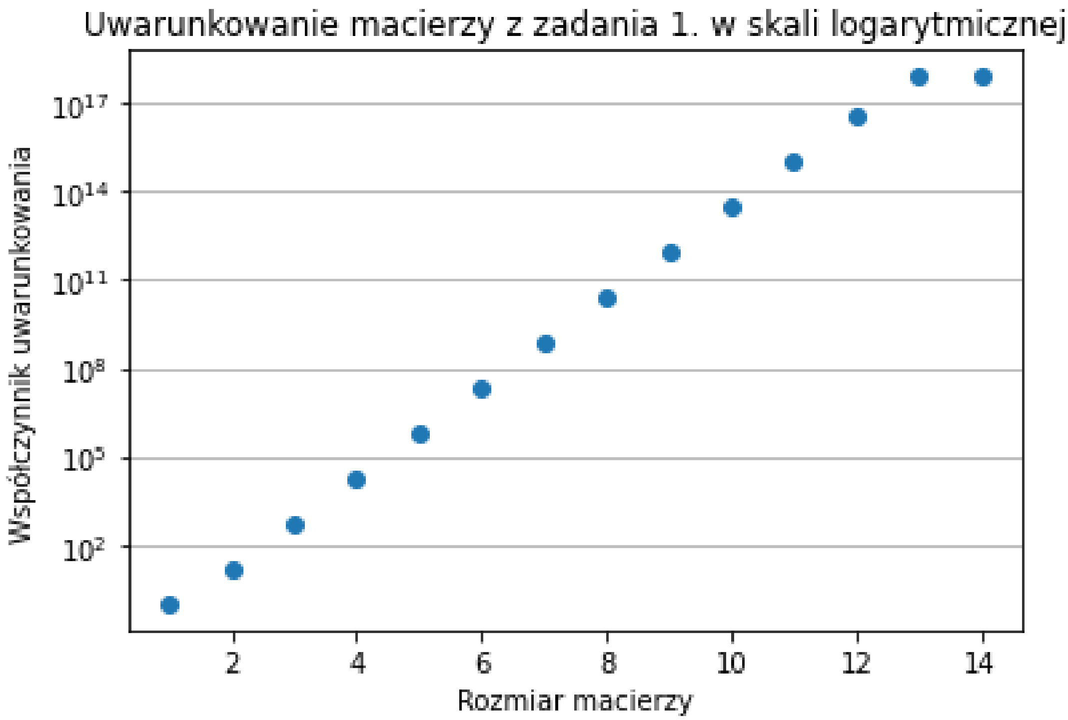
Następnie metodą eliminacji Gaussa rozwiąż układ równań liniowych $\mathbf{A}\boldsymbol{x}=\mathbf{b}$ (przyjmując jako niewiadomą wektor $\boldsymbol{x}$). Przyjmij różną precyzję dla znanych wartości macierzy **A** i wektora **b**. Sprawdź, jak błędy zaokrągleń zaburzają rozwiązanie dla różnych rozmiarów układu (porównaj – zgodnie z wybraną normą – wektory $\boldsymbol{x}$ *obliczony* z $\boldsymbol{x}$ *zadany*). Przeprowadź eksperymenty dla różnych rozmiarów układu.

4.1 Wyniki

Błąd dla poszczególnych precyzji

n	float64	float32	float16
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	3.1086244689504383e-15	3.973643e-06	0.01563
4	3.774758283725532e-15	5.3077936e-05	0.1157
5	9.452993943170895e-12	0.0016908527	2.465
6	6.228835965534547e-11	0.03423637	1.93
7	2.600175677943842e-09	1.4587183	3.662
8	4.265395663061167e-08	3.9666986	5.266
9	5.43500676464248e-07	3.4150329	7.844
10	7.283076397545108e-05	1.0647228	0.5723
11	0.0017260893048604514	0.92691535	1.365
12	0.08406431026016954	6.1354117	2.15
13	2.046072692726407	67.049286	1.465
14	0.9690465098251082	4.846507	2.3

4.2 Uwarunkowanie układu



4.3 Wnioski

Błąd rozwiązania rośnie bardzo szybko, dlatego już dla rozmiarów powyżej 10, wyniki stają się bezużyteczne. Powodem tak złych wyników jest sama macierz A. Jak można zaobserwować na powyższym wykresie uwarunkowanie tej macierzy rośnie wykładniczo i osiąga bardzo duże wartośći.

5 Zad2

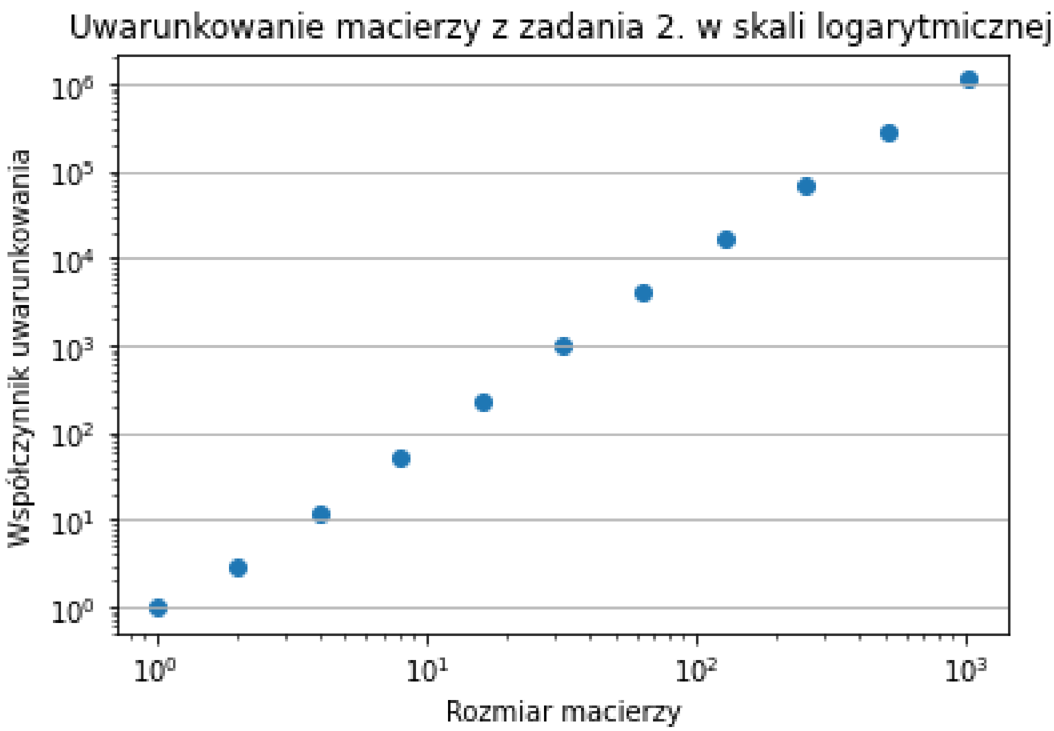
Powtórz eksperyment dla macierzy zadanej wzorem:

$$\begin{cases} a_{ij} = \dfrac{2i}{j} & dla \quad j \ge i \\ a_{ij} = a_{ji} & dla \quad j < i \end{cases} \qquad i, j = 1, \ldots, n$$

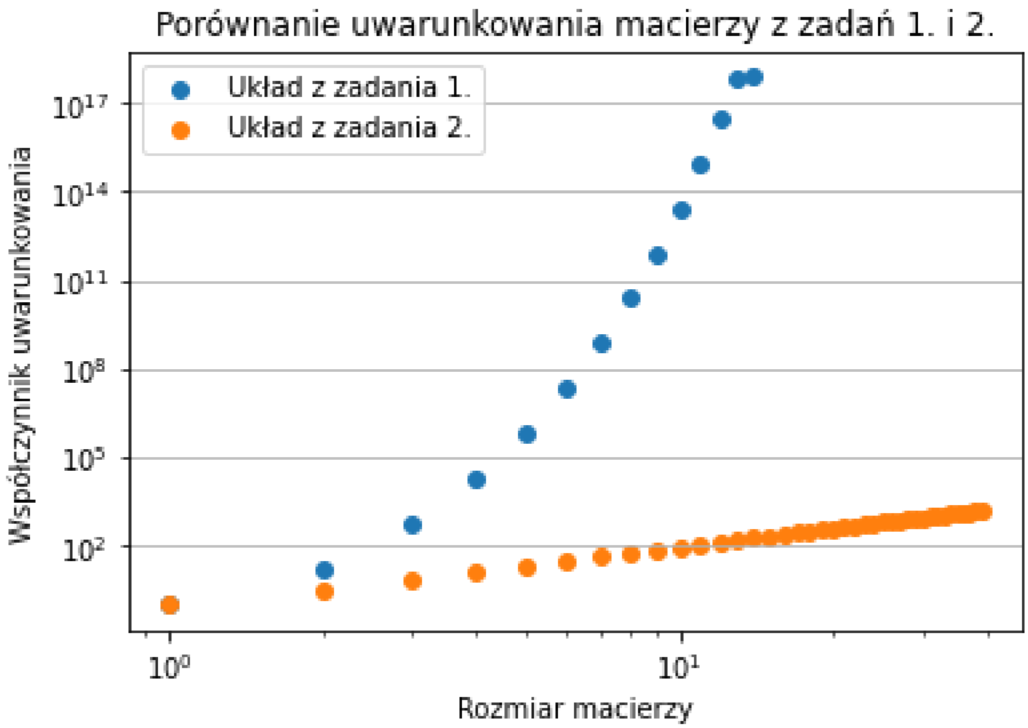
Porównaj wyniki z tym, co otrzymano w przypadku układu z punktu 1). Spróbuj uzasadnić, skąd biorą się różnice w wynikach. Sprawdź uwarunkowanie obu układów.

5.1 Wyniki

Błąd dla poszczególnych precyzji			
n	float64	float32	float16
5	3.108624468950438e-16	2.861023e-07	0.000977
10	1.254552017826427e-15	4.3511392e-07	0.00581
15	1.080617077301819e-15	1.1881192e-06	0.012985
25	8.895106873296755e-15	5.5122377e-06	0.05316
50	2.26885177312397e-14	1.9313096e-05	0.1305
75	7.403855306620244e-14	3.0659834e-05	0.339
100	1.0485501356072291e-13	5.9408547e-05	0.589
200	3.81314979591707e-13	0.0001768002	6.44



<matplotlib.legend.Legend at 0x7fa3ccb11cd0>



5.2 Wnioski

W odróżnieniu od macierzy z zadania 1. obecna macierz utrzymuje niski współczynnik uwarunkowania. Ma to odzwierciedlenie w wyliczonych błędach, które nawet dla rozmiaru macierzy równego 200 są małe. W odróżnieniu od przypadku z zadania 1. tutaj można lepiej zaobserwować różnice w dokładności obliczeń w zależnośći od precyzji.

6 Zad3

Powtórz eksperyment dla jednej z macierzy zadanej wzorem poniżej (macierz i parametry podane w zadaniu indywidualnym). Następnie rozwiąż układ metodą przeznaczoną do rozwiązywania układów z macierzą trójdiagonalną. Porównaj wyniki otrzymane dwoma metodami (czas, dokładność obliczeń i zajętość pamięci) dla różnych rozmiarów układu. Przy porównywaniu czasów należy pominąć czas tworzenia układu. Opisz, jak w metodzie dla układów z macierzą trójdiagonalną przechowywano i wykorzystywano macierz A.

b) (m,k - parametry zadania):

$$\begin{cases} a_{i,i} = -m \cdot i - k \\ a_{i,i+1} = i \\ a_{i,i-1} = \dfrac{m}{i} \quad dla \;\; i > 1 \\ a_{i,j} = 0 \qquad dla \;\; j < i - 1 \quad oraz \quad j > i + 1 \end{cases} \qquad\qquad i, j = 1, \ldots, n$$

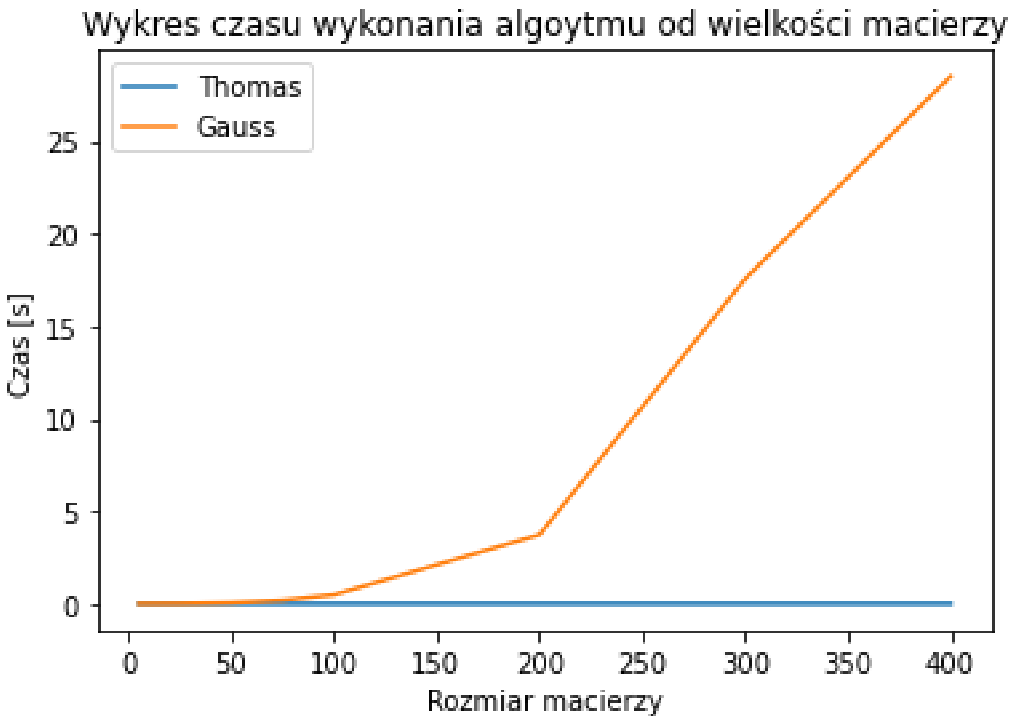
$$k = 6, m = 3$$

6.1 Wyniki

6.2 Czasy rozwiazania układu równań w zależnośći od algorytmu i rozmiaru układu

Czasy wykonywania alogorytmów w sekundach		
n	Gauss	Thomas
5	0.0001327991485595703	5.7220458984375e-05
10	0.0010221004486083984	0.00013303756713867188
15	0.0020911693572998047	0.00011515617370605469
25	0.0070953369140625	0.00014472007751464844
50	0.05832362174987793	0.00023651123046875
75	0.19103789329528809	0.0003674030303955078
100	0.4847218990325928	0.0009531974792480469
200	3.736295461654663	0.0008718967437744141
300	17.579717874526978	0.0015783309936523438
400	28.51323890686035	0.0017549991607666016

```
<matplotlib.legend.Legend at 0x7fa3ccbf3400>
```

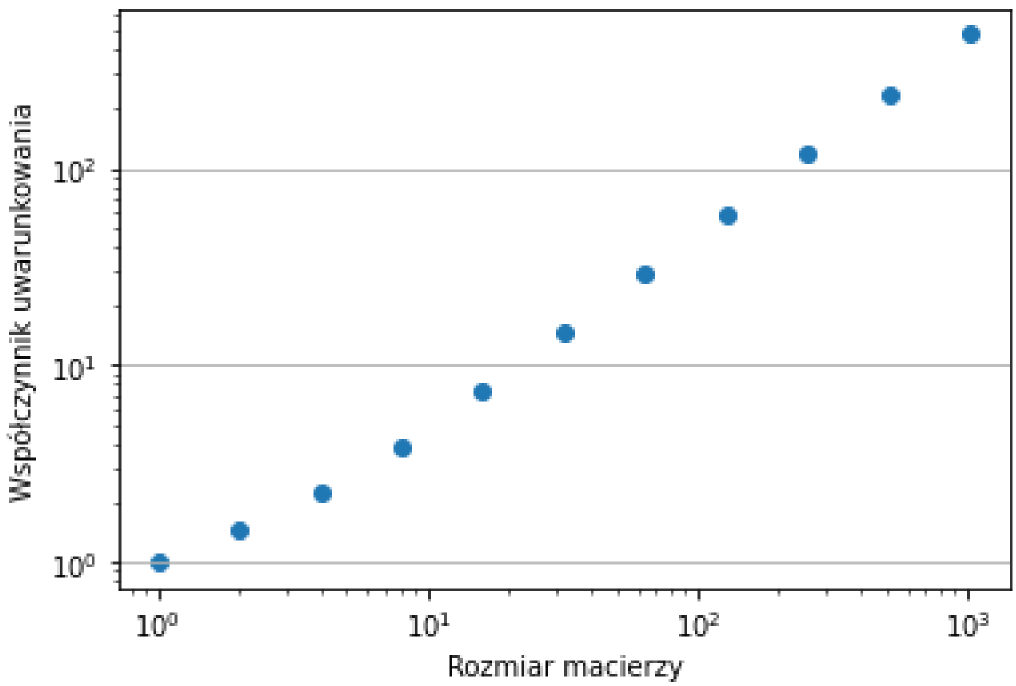


6.3 Dokładność rozwiazania układu równań w zależnośći od algorytmu i rozmiaru układu

Dokładności obliczeń algorytmów dla typu:float64		
n	Gauss	Thomas
5	0.0	0.0
10	1.1102230246251566e-17	0.0
15	7.401486830834377e-17	0.0
25	7.993605777301127e-17	0.0
50	6.439293542825907e-17	0.0
75	6.957397620984315e-17	0.0
100	7.993605777301127e-17	0.0
200	8.992806499463769e-17	0.0
300	8.178642948071987e-17	0.0
400	8.271161533457416e-17	0.0

Dokładności obliczeń algorytmów dla typu:float32		
n	Gauss	Thomas

5	1.1102230246251565e-16	0.0
10	6.661338147750939e-17	0.0
15	6.661338147750939e-17	0.0
25	7.549516567451065e-17	0.0
50	6.661338147750939e-17	0.0
75	9.177843670234627e-17	0.0
100	7.105427357601002e-17	0.0
200	7.771561172376095e-17	0.0
300	7.734553738221924e-17	0.0
400	8.18789480661053e-17	0.0



6.4 Wnioski

Przeprowadzone testy czasowe zgadzają się z założeniami, że algorytm Thomasa ma złożoność obliczeniową $O(n)$, a elminacja Gaussa $O(n^3)$. Dla algorytmu Thomasa macierz była przechowywana jako trzy tablice odpowiadające trzem przekątnym, dzięki czemu złożoność pamięciowa algorytmu Thomasa wynosi $O(n)$ i jest mniejsza od złożoności pamięciowej eliminacji Gaussa, która jest równa $O(n^2)$. Jeśli chodzi o dokładność obu metod, to nie udało się zaobserwować różnic. Błędy algorytmu Thomasa są równe 0, a eliminacj Gaussa bardzo bliskie zeru. Jest to wynikiem bardzo dobrego uwarunkowania macierzy.