

Metody Obliczeniowe w Nauce i Technice

**Rozwiązywanie układów równań
metodami bezpośrednimi**

Mateusz Łopaciński

25.05.2022 r.

I. Dane techniczne sprzętu

Obliczenia zostały wykonane na komputerze o następujących parametrach:

- Procesor: AMD Ryzen 7 4700U (8 rdzeni, 8 wątków),
- Pamięć RAM: 16 GB 3200 MHz

II. Zadanie 1

1. Wprowadzenie

1.1. Opis problemu

Dany jest układ równań liniowych $Ax = b$. Macierz A składa się z elementów, wyznaczanych zgodnie z poniższym wzorem:

$$\begin{cases} a_{1,j} = 1 \\ a_{i,j} = \frac{1}{i+j-1} \text{ dla } i \neq 1 \end{cases} \quad \text{gdzie } i, j = 1, \dots, n$$

(1.1.1.)

Za wektor x przyjęta miała zostać dowolna n -elementowa permutacja liczb ze zbioru $\{-1, 1\}$, a następnie, wykorzystując wektor x , obliczony miał zostać wektor b .

W kolejnym kroku, dla znanych wartości macierzy A oraz wektora b (takich, jak otrzymane w poprzednim kroku zadania) obliczony miał zostać wektor x . Tak obliczony wektor x miał zostać następnie porównany z zadanym wcześniej wektorem x i sprawdzone miały zostać błędy zaokrągleń. Obliczenia miały zostać przeprowadzone dla różnych rozmiarów układu równań (różnych wartości n). Na koniec, należało sprawdzić uwarunkowanie analizowanego układu równań.

2. Kryterium pomiaru błędu

2.1. Norma maksimum

Jako kryterium, według którego wyznaczałem wartości błędów dla obliczonego wektora x , porównując go ze wcześniej zadanym wektorem x , skorzystałem z normy maksimum. Błąd zatem był wyznaczany jako maksymalna wartość spośród wartości bezwzględnych różnic kolejnych współrzędnych wektorów. Sposób wyznaczania błędu można zapisać, przy pomocy wzoru:

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

gdzie:

x_i – i . współrzędna wyznaczonego wektora x ,

\bar{x}_i – i . współrzędna zadanego wektora x .

(1.2.1.)

3. Opracowanie zagadnienia

3.1. Metoda postępowania

Program, zastosowany do rozwiązania problemu, napisałem w języku Python. Korzystałem z biblioteki numpy, która pozwala na użycie liczb zmiennoprzecinkowych o różnej precyzji. Wykorzystałem zmienne 2 różnych precyzji: `float32` oraz `float64` (odpowiednio 32-bitowy oraz 64-bitowy) do wyznaczenia znanych wartości macierzy A oraz wektora b . Obliczenia przeprowadzałem dla kolejnych wartości n z przedziału 2 – 200. Przyjęty wektor x miał postać $x = [-1, 1, -1, 1, -1, \dots]$, czyli składał się z na przemian występujących wartości -1 oraz 1 .

3.2. Wartości błędów oraz czasy obliczeń wektora x

3.2.1. Wykresy błędów obliczeń

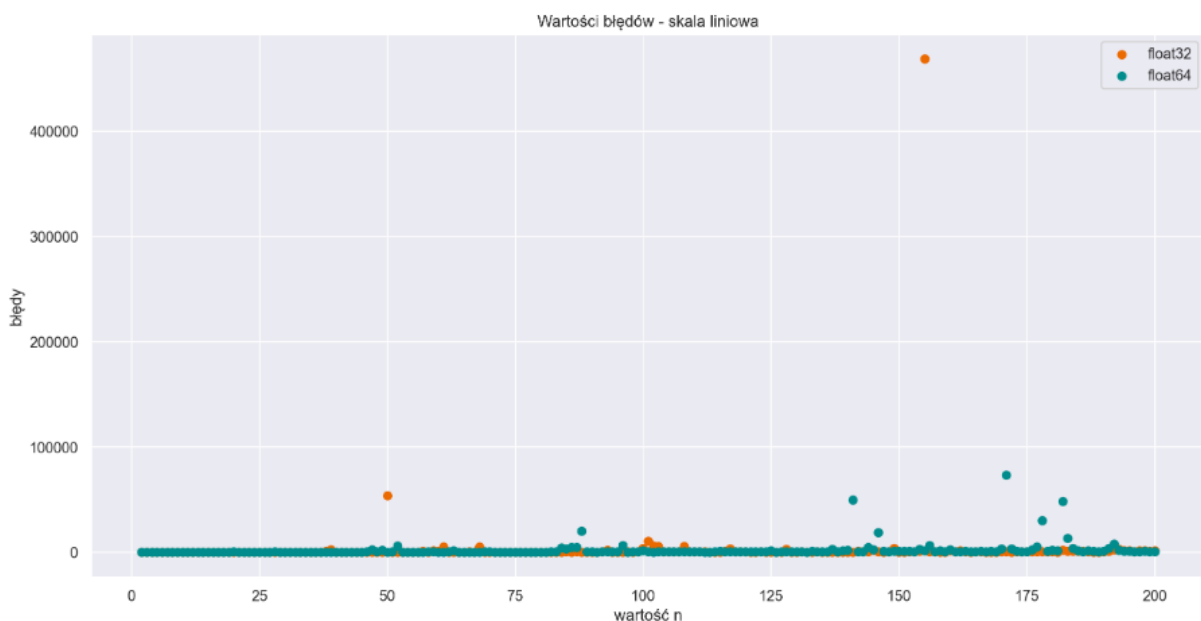
Poniżej umieściłem wykresy błędów obliczeń dla zmiennych typu `float32` oraz `float64`.

Jak możemy zauważyć, początkowo błędy rosną szybko i systematycznie (widać to dokładnie na wykresie w skali logarytmicznej), a więc zwiększanie wartości n powoduje wzrost błędów.

Możemy także zauważyć, że przy wartości n wynoszącej ok. 13, wartości błędów dla zmiennych obu typów przestają tak gwałtownie rosnąć, a błędy są coraz bardziej losowe (jest to rezultat złego uwarunkowania).

Również dla $n < 13$ widzimy, że błędy dla zmiennych typu `float32` są większe niż dla zmiennych typu `float64`. Dla większych wartości n ta tendencja już nie występuje.

- Wykres w skali liniowej dla $n = 2, \dots, 200$ (wartości błędów)



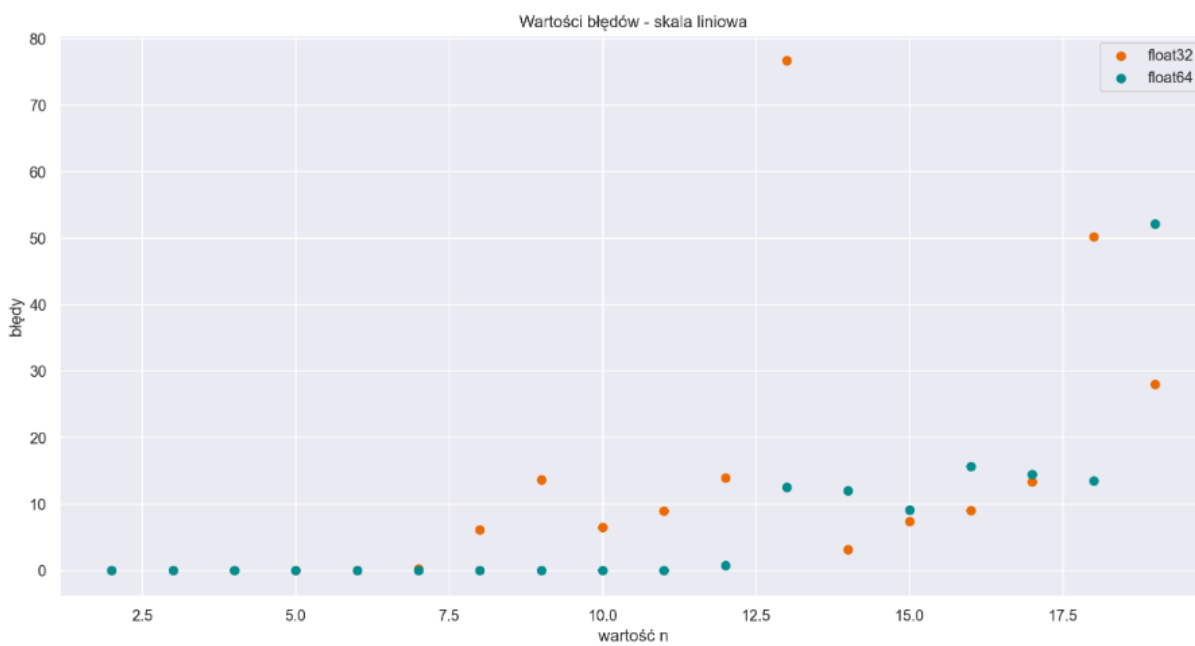
Rys. 1.3.2.1. Wykres błędów dla n od 2 do 200 (skala liniowa)

- Wykres w skali logarytmicznej dla $n = 2, \dots, 200$ (wartości błędów)



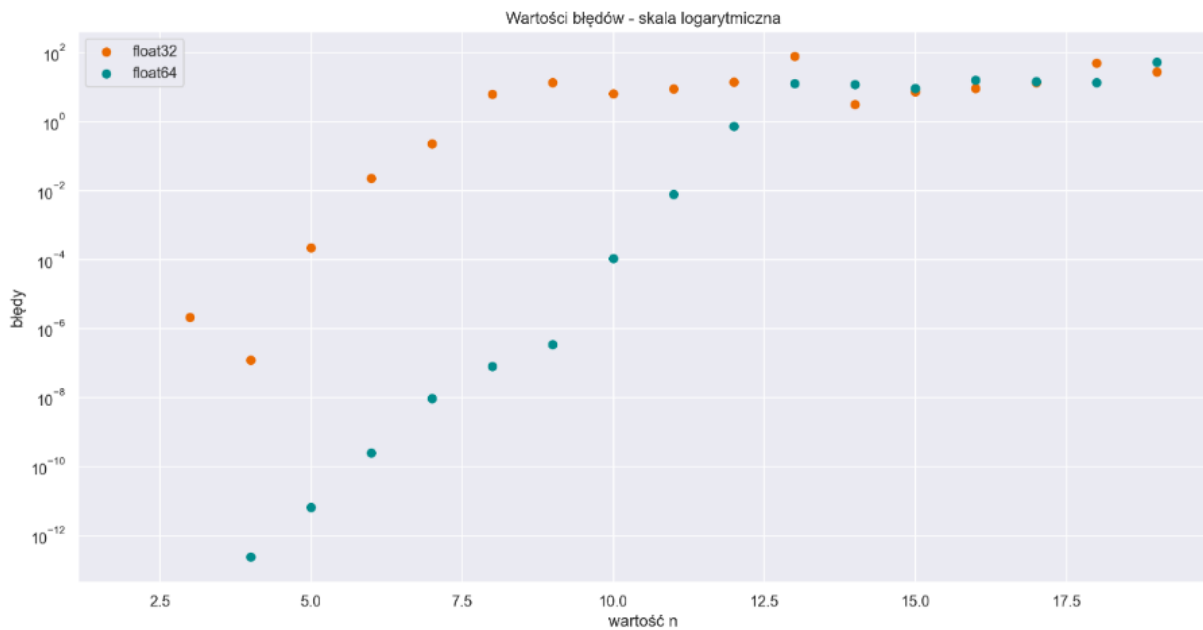
Rys. 1.3.2.2. Wykres błędów dla n od 2 do 200 (skala logarytmiczna)

- Wykres w skali liniowej dla $n = 2, \dots, 20$ (wartości błędów)



Rys. 1.3.2.3. Wykres błędów dla n od 2 do 20 (skala liniowa)

▪ Wykres w skali logarytmicznej dla $n = 2, \dots, 20$ (wartości błędów)

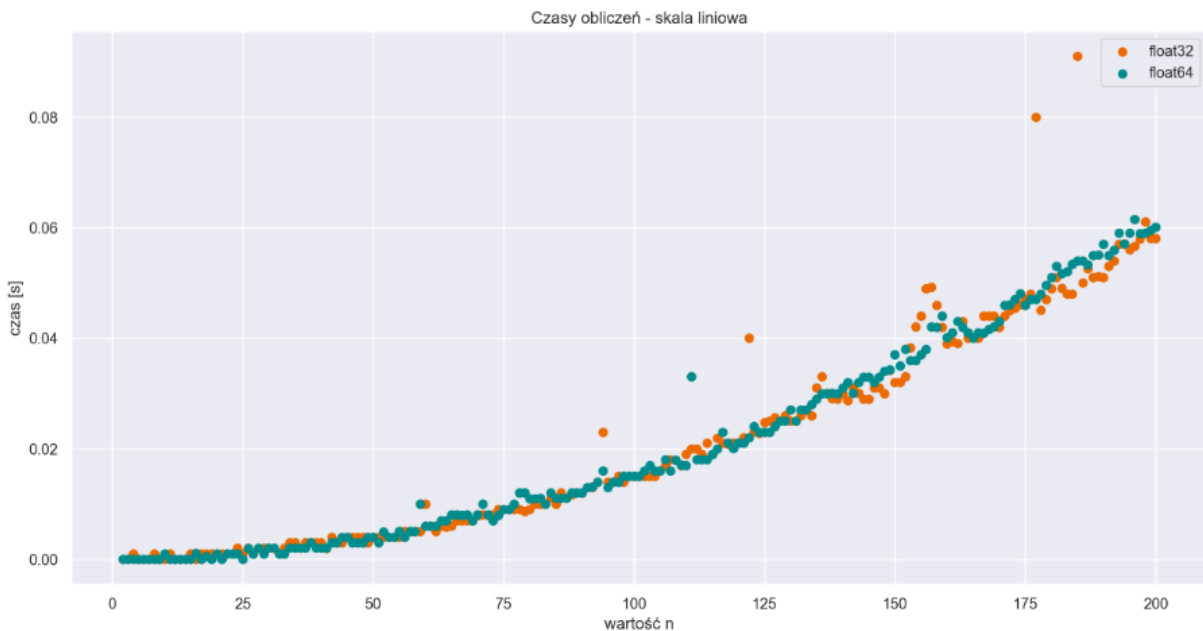


Rys. 1.3.2.4. Wykres błędów dla n od 2 do 20 (skala logarytmiczna)

3.2.2. Wykres czasów obliczeń

Poniżej umieściłem wykres czasów obliczeń dla zmiennych typu float32 oraz float64.

Ponieważ algorytm eliminacji Gaussa ma złożoność wielomianową (dokł. $O(n^3)$), obserwujemy wykres o kształcie zgodnym z oczekiwaniami. Widzimy, że zwiększanie wartości n powoduje wzrost czasu obliczeń. W przypadku zmiennych obu precyzji, wzrost czasu obliczeń jest bardzo zbliżony.



Rys. 1.3.2.5. Wykres czasów obliczeń wektora dla n od 2 do 200 (skala liniowa)

3.2.3. Zestawienie błędów oraz czasów obliczeń

W tabeli zamieszczonej niżej znajdują się błędy oraz czasy obliczeń dla typów float32 oraz float64. W pierwszej tabeli umieściłem rezultaty dla wartości n od 2 do 50, natomiast w kolejnej, znajdują się wartości dla $n = 10, 20, \dots, 200$.

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	0,00000E+00	0,00000E+00	9,94444E-04	0,00000E+00
3	2,08616E-06	0,00000E+00	0,00000E+00	0,00000E+00
4	1,19209E-07	2,36478E-13	0,00000E+00	0,00000E+00
5	2,15411E-04	6,51357E-12	1,01066E-03	0,00000E+00
6	2,23271E-02	2,48666E-10	0,00000E+00	0,00000E+00
7	2,25690E-01	9,43695E-09	0,00000E+00	0,00000E+00
8	6,13139E+00	7,98740E-08	0,00000E+00	0,00000E+00
9	1,36161E+01	3,42392E-07	0,00000E+00	0,00000E+00
10	6,49310E+00	1,05293E-04	0,00000E+00	1,01018E-03
11	8,97018E+00	7,64796E-03	9,97782E-04	0,00000E+00
12	1,39067E+01	7,19864E-01	0,00000E+00	0,00000E+00
13	7,66610E+01	1,25322E+01	0,00000E+00	0,00000E+00
14	3,12784E+00	1,19926E+01	0,00000E+00	0,00000E+00
15	7,34302E+00	9,06911E+00	9,98497E-04	0,00000E+00
16	9,04856E+00	1,56372E+01	0,00000E+00	1,02735E-03
17	1,33554E+01	1,44139E+01	1,00255E-03	0,00000E+00
18	5,02067E+01	1,35004E+01	9,95636E-04	5,38349E-04
19	2,79621E+01	5,21564E+01	9,98259E-04	0,00000E+00
20	9,07050E+00	4,03456E+02	9,99689E-04	9,42469E-04
21	2,95725E+01	3,50990E+01	1,00017E-03	0,00000E+00
22	1,10628E+01	3,17171E+01	9,99928E-04	9,89199E-04
23	2,59931E+01	2,82740E+01	9,99928E-04	1,00350E-03
24	8,16652E+01	5,61458E+01	2,01344E-03	1,00827E-03
25	1,07535E+01	1,08448E+02	9,97305E-04	0,00000E+00
26	1,68614E+01	4,41860E+01	2,01368E-03	1,98412E-03
27	1,29021E+01	4,92055E+01	1,01280E-03	9,89676E-04
28	4,82137E+01	6,22804E+02	1,98746E-03	1,99866E-03
29	8,50908E+01	2,09196E+02	1,99914E-03	1,00017E-03
30	1,90770E+02	7,30665E+01	2,00152E-03	1,99938E-03
31	1,89783E+02	9,16385E+01	1,98865E-03	2,00033E-03
32	1,97260E+02	3,50967E+01	1,00231E-03	1,00017E-03
33	5,00214E+01	3,39045E+01	2,01178E-03	9,99451E-04
34	2,33014E+01	2,27654E+02	2,98715E-03	2,00057E-03
35	2,18943E+01	2,10305E+02	3,01790E-03	1,99938E-03
36	1,89606E+01	1,30375E+02	2,00033E-03	1,99938E-03
37	2,23685E+01	1,65740E+02	2,98500E-03	2,00033E-03

38	8,90380E+02	1,61622E+02	3,01147E-03	2,99931E-03
39	2,31892E+03	6,09164E+01	2,98834E-03	1,99986E-03
40	8,61740E+01	8,95872E+01	3,01123E-03	1,99842E-03
41	2,10694E+02	2,25021E+02	1,98865E-03	1,99842E-03
42	7,98876E+01	1,22925E+02	3,99041E-03	3,00002E-03
43	3,73795E+01	1,28931E+02	2,99859E-03	2,99907E-03
44	2,38405E+01	1,60580E+02	2,99859E-03	4,01998E-03
45	1,47576E+01	1,67700E+02	4,00090E-03	3,98517E-03
46	2,48722E+01	3,37335E+02	3,99995E-03	3,01480E-03
47	2,51772E+01	2,15019E+03	3,99971E-03	2,99120E-03
48	3,91334E+01	5,85814E+02	3,99971E-03	3,01027E-03
49	3,86557E+01	1,95478E+03	2,99859E-03	3,99923E-03
50	5,36307E+04	8,68574E+01	3,99971E-03	3,99160E-03

Tabela. 1.3.2.1. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla n od 2 do 50

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
10	6,49310E+00	1,05293E-04	0,00000E+00	1,01018E-03
20	9,07050E+00	4,03456E+02	9,99689E-04	9,42469E-04
30	1,90770E+02	7,30665E+01	2,00152E-03	1,99938E-03
40	8,61740E+01	8,95872E+01	3,01123E-03	1,99842E-03
50	5,36307E+04	8,68574E+01	3,99971E-03	3,99160E-03
60	7,35694E+02	8,12254E+01	1,00083E-02	5,99289E-03
70	9,47871E+01	3,31843E+02	7,84516E-03	8,01921E-03
80	3,42387E+01	1,48167E+02	9,00888E-03	1,09897E-02
90	1,55563E+02	4,49830E+02	1,19991E-02	1,20003E-02
100	3,64353E+03	1,16070E+03	1,50387E-02	1,50115E-02
110	2,52172E+02	3,99993E+02	1,90265E-02	1,70050E-02
120	2,40383E+02	6,11717E+02	2,10009E-02	2,10180E-02
130	1,11719E+02	6,00331E+02	2,49968E-02	2,70123E-02
140	2,14922E+02	1,87185E+03	2,99990E-02	3,09987E-02
150	1,49628E+02	1,03845E+03	3,19793E-02	3,69897E-02
160	1,59070E+03	2,11011E+03	3,90019E-02	4,00891E-02
170	2,69813E+02	3,28303E+03	4,19898E-02	4,30002E-02
180	2,45832E+02	2,00272E+03	4,89926E-02	5,09894E-02
190	3,55095E+02	7,11754E+02	5,09889E-02	5,70247E-02
200	1,45109E+03	6,23205E+02	5,80215E-02	6,00557E-02

Tabela. 1.3.2.2. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla wybranych wartości n

III. Zadanie 2

1. Wprowadzenie

1.1. Opis problemu

Problem jest analogiczny do tego z zadania 1. W tym przypadku jednak obliczenia należało wykonać dla macierzy zdanej wzorem:

$$\begin{cases} a_{1,j} = 1 \\ a_{i,j} = \frac{1}{i+j-1} \text{ dla } i \neq 1 \end{cases} \quad \text{gdzie } i, j = 1, \dots, n$$

(2.1.1.)

2. Kryterium pomiaru błędu

Analogicznie, jak w zadaniu 1., do pomiaru błędu wykorzystałem normę maksimum, opisaną w punkcie 2.1..

3. Opracowanie zagadnienia

3.1. Metoda postępowania

Postępowanie jest analogiczne, jak w poprzednim zadaniu.

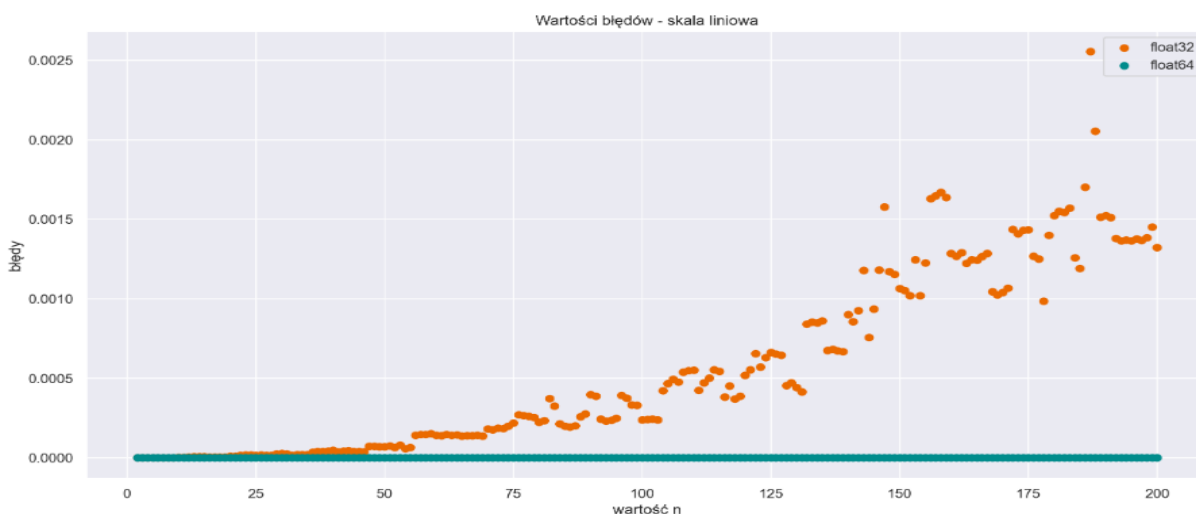
3.2. Wartości błędów oraz czasy obliczeń wektora x

3.2.1. Wykresy błędów obliczeń

Poniżej umieściłem wykresy błędów obliczeń dla zmiennych typu float32 oraz float64.

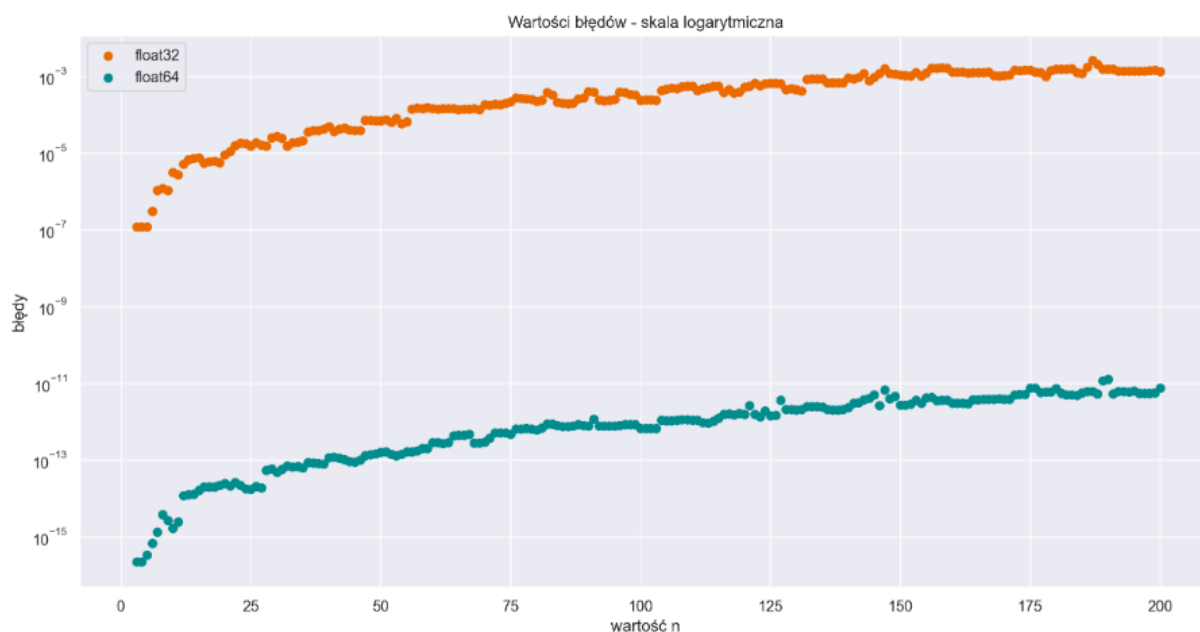
Jak możemy zauważyć, wartości błędów dla zmiennych typu float32 rosną znacznie szybciej niż dla bardziej dokładnego typu, jakim jest float64. Analizując wykres w skali logarytmicznej, możemy dojść do wniosku, że największy wpływ na błąd ma w tym przypadku błąd zaokrąglenia, związany faktem, iż typ 32-bitowy jest w stanie pomieścić mniej cyfr liczby niż typ 64-bitowy.

- Wykres w skali liniowej dla $n = 2, \dots, 200$ (wartości błędów)



Rys. 2.3.2.1. Wykres błędów dla n od 2 do 200 (skala liniowa)

▪ Wykres w skali logarytmicznej dla $n = 2, \dots, 200$ (wartości błędów)

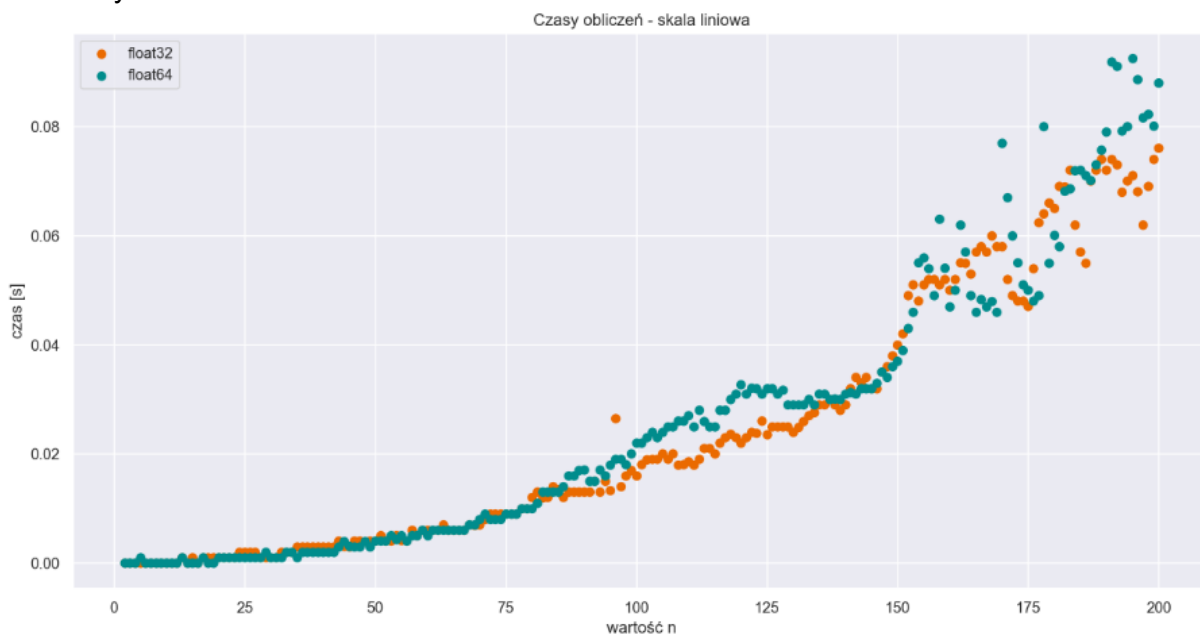


Rys. 2.3.2.2. Wykres błędów dla n od 2 do 200 (skala logarytmiczna)

3.2.2. Wykres czasów obliczeń

Poniżej umieściłem wykresy czasów obliczeń dla zmiennych typu float32 oraz float64.

Czasy obliczeń dla zmiennych obu typów są bardzo zbliżone i podobne do tych, jakie uzyskałem dla układu równań z 1. zadania.



Rys. 2.3.2.3. Wykres czasów obliczeń wektora dla n od 2 do 200 (skala liniowa)

3.2.3. Zestawienie błędów oraz czasów obliczeń

Tak, jak poprzednio, zamieściłem poniżej 2 tabele. W pierwszej z nich znajdują się rezultaty dla n od 2 do 50, a w drugiej dla $n = 10, 20, \dots, 200$.

Widzimy, że w przypadku tego układu równań, precyzja uzyskanych wyników jest bardzo wysoka, a błędy, dla zmiennych typu float64 są rzędu 10^{-13} dla $n = 200$.

Podobnie, jak widzieliśmy na wykresach, możemy zobaczyć w poniższych tabelach, że błędy dla typu float32 są znacznie większe niż dla float64.

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
3	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
4	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
5	1,19209E-07	3,33067E-16	0,00000E+00	9,99928E-04
6	2,98023E-07	6,66134E-16	0,00000E+00	0,00000E+00
7	1,07288E-06	1,33227E-15	0,00000E+00	0,00000E+00
8	1,19209E-06	3,77476E-15	0,00000E+00	0,00000E+00
9	1,07288E-06	2,66454E-15	0,00000E+00	0,00000E+00
10	3,09944E-06	1,66533E-15	0,00000E+00	0,00000E+00
11	2,74181E-06	2,44249E-15	0,00000E+00	0,00000E+00
12	5,12600E-06	1,17684E-14	0,00000E+00	0,00000E+00
13	6,67572E-06	1,24345E-14	1,00088E-03	9,91106E-04
14	7,33137E-06	1,28786E-14	0,00000E+00	0,00000E+00
15	7,45058E-06	1,62093E-14	9,99451E-04	0,00000E+00
16	5,48363E-06	2,02061E-14	0,00000E+00	0,00000E+00
17	5,90086E-06	2,02061E-14	9,97066E-04	1,01113E-03
18	6,19888E-06	2,02061E-14	9,80377E-04	0,00000E+00
19	5,60284E-06	2,15383E-14	1,01209E-03	0,00000E+00
20	9,05991E-06	2,46470E-14	9,94921E-04	1,00017E-03
21	1,09076E-05	2,13163E-14	9,94205E-04	1,00040E-03
22	1,54376E-05	2,62013E-14	1,00231E-03	1,00303E-03
23	1,79410E-05	2,19824E-14	1,00231E-03	9,89199E-04
24	1,77026E-05	1,79856E-14	2,00009E-03	9,88483E-04
25	1,50800E-05	1,73195E-14	1,99676E-03	9,96351E-04
26	1,85370E-05	2,06501E-14	1,99223E-03	9,85146E-04
27	1,60933E-05	1,86517E-14	1,99819E-03	9,99928E-04
28	1,51396E-05	5,32907E-14	1,00183E-03	9,99928E-04
29	2,49147E-05	5,92859E-14	1,00899E-03	1,99986E-03
30	2,69413E-05	4,86278E-14	9,92298E-04	9,99689E-04
31	2,38419E-05	5,63993E-14	1,00827E-03	1,00017E-03
32	1,50800E-05	6,99441E-14	2,01488E-03	9,98974E-04
33	1,87159E-05	6,70575E-14	1,98388E-03	2,00009E-03
34	1,88947E-05	6,86118E-14	2,00701E-03	1,99962E-03

35	2,06232E-05	6,32827E-14	2,96926E-03	9,89199E-04
36	3,51667E-05	8,68194E-14	3,00860E-03	1,99986E-03
37	3,89814E-05	8,28226E-14	3,00908E-03	1,99986E-03
38	3,86834E-05	8,06022E-14	2,98262E-03	2,00009E-03
39	4,26173E-05	7,79377E-14	2,99931E-03	2,00033E-03
40	4,81009E-05	1,14575E-13	2,99716E-03	1,99986E-03
41	3,65973E-05	1,18350E-13	3,00908E-03	1,99986E-03
42	4,17233E-05	1,12577E-13	3,00765E-03	1,99986E-03
43	4,46439E-05	1,01696E-13	4,00686E-03	2,99978E-03
44	3,97563E-05	9,01501E-14	3,01480E-03	3,98946E-03
45	3,87430E-05	8,90399E-14	2,98762E-03	3,01075E-03
46	3,80278E-05	1,00142E-13	4,01759E-03	2,99907E-03
47	7,11083E-05	1,30562E-13	4,00853E-03	3,00002E-03
48	7,12872E-05	1,35447E-13	3,98326E-03	3,97944E-03
49	7,00355E-05	1,44329E-13	4,01783E-03	2,99883E-03
50	6,89626E-05	1,58762E-13	4,01711E-03	3,99923E-03

Tabela. 2.3.2.1. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla n od 2 do 50

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
10	3,09944E-06	1,66533E-15	0,00000E+00	0,00000E+00
20	9,05991E-06	2,46470E-14	9,94921E-04	1,00017E-03
30	2,69413E-05	4,86278E-14	9,92298E-04	9,99689E-04
40	4,81009E-05	1,14575E-13	2,99716E-03	1,99986E-03
50	6,89626E-05	1,58762E-13	4,01711E-03	3,99923E-03
60	1,42455E-04	2,82219E-13	5,99980E-03	4,98986E-03
70	1,80244E-04	2,95097E-13	6,99997E-03	8,00991E-03
80	2,23160E-04	6,05405E-13	1,20127E-02	1,00117E-02
90	3,96729E-04	7,64944E-13	1,29950E-02	1,70288E-02
100	2,39253E-04	6,66356E-13	1,60003E-02	2,20249E-02
110	5,50270E-04	1,09135E-12	1,85983E-02	2,70109E-02
120	5,17368E-04	1,54921E-12	2,20242E-02	3,27213E-02
130	4,42505E-04	2,03171E-12	2,40176E-02	2,89917E-02
140	9,00149E-04	2,34979E-12	2,90105E-02	3,09546E-02
150	1,06287E-03	2,71927E-12	4,00014E-02	3,69990E-02
160	1,28424E-03	3,06777E-12	4,99797E-02	4,70030E-02
170	1,03962E-03	3,78164E-12	5,80125E-02	7,69944E-02
180	1,52260E-03	7,36389E-12	6,50294E-02	6,00295E-02
190	1,52230E-03	1,27474E-11	7,20117E-02	7,90308E-02

Tabela. 2.3.2.2. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla wybranych wartości n

3.3. Współczynnik uwarunkowania macierzy A

3.3.1. Sposób wyznaczania współczynnika uwarunkowania

Współczynnik uwarunkowania określa, w jakim stopniu błąd reprezentacji danych wejściowych wpływa na błąd wyniku. Jeżeli wskaźnik uwarunkowania dla danego problemu jest duży, wówczas nawet niewielki błąd danych może spowodować, że wynik będzie znacznie odbiegał od przewidywanego. Wskaźnik uwarunkowania oblicza się ze wzoru:

$$\kappa(A) = \|A^{-1}\| \cdot \|A\| \quad (2.3.1.)$$

Norma jest określona poniższym wzorem:

$$\|A\| = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{i,j}| \quad (2.3.2.)$$

3.3.2. Porównanie współczynników uwarunkowania macierzy A z zadania 1 i zadania 2

Zamieszczona niżej tabela przedstawia współczynniki uwarunkowania dla macierzy A wyznaczonej zgodnie ze wzorem z zadania 1 oraz ze wzorem z zadania 2. Od razu daje się zauważyć znaczącą różnicę w wartościach współczynników uwarunkowania, które są dużo mniejsze dla macierzy A z 2. zadania.

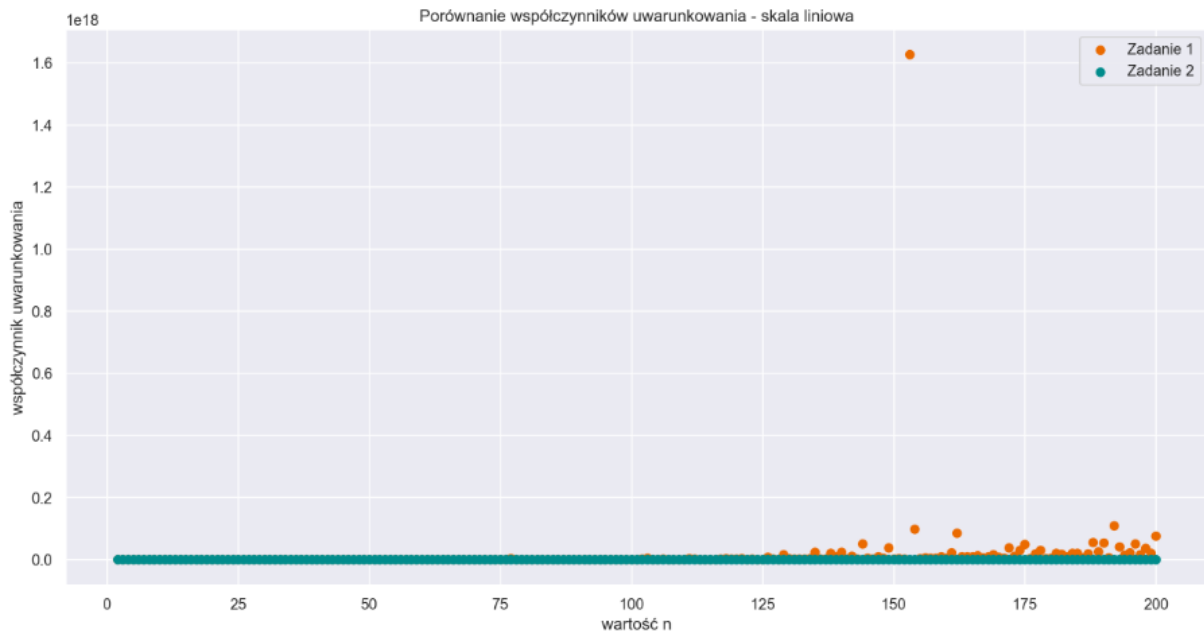
n	Zadanie 1	Zadanie 2	n	Zadanie 1	Zadanie 2	n	Zadanie 1	Zadanie 2
2	8,00000E+00	1,00000E+00	17	3,04549E+11	7,07762E+00	32	1,22792E+12	1,31434E+01
3	2,16000E+02	1,44444E+00	18	1,18749E+14	7,48503E+00	33	8,30490E+11	1,35474E+01
4	2,88000E+03	1,83333E+00	19	3,75040E+11	7,88957E+00	34	7,68770E+11	1,39533E+01
5	2,80000E+04	2,23333E+00	20	4,00389E+11	8,28957E+00	35	6,30767E+12	1,43542E+01
6	2,26800E+05	2,64444E+00	21	2,96931E+11	8,69862E+00	36	1,81944E+12	1,47616E+01
7	1,62994E+06	3,03175E+00	22	7,79582E+11	9,09968E+00	37	8,38998E+11	1,51652E+01
8	1,28621E+07	3,44841E+00	23	4,61263E+11	9,50547E+00	38	7,17348E+12	1,55687E+01
9	1,12000E+08	3,84921E+00	24	5,02202E+11	9,91063E+00	39	1,24365E+12	1,59747E+01
10	8,84144E+08	4,24921E+00	25	1,08204E+12	1,03106E+01	40	6,43340E+12	1,63764E+01
11	6,47379E+09	4,65943E+00	26	2,37439E+12	1,07194E+01	41	2,90544E+12	1,67829E+01
12	4,40794E+10	5,05522E+00	27	1,80712E+13	1,11216E+01	42	5,36701E+12	1,71869E+01
13	1,34767E+11	5,46548E+00	28	4,15528E+12	1,15263E+01	43	1,93906E+12	1,75900E+01
14	2,45922E+11	5,86890E+00	29	5,50325E+11	1,19319E+01	44	3,01970E+12	1,79962E+01
15	1,73331E+11	6,26890E+00	30	6,48358E+11	1,23319E+01	45	4,26510E+12	1,83985E+01
16	1,08497E+11	6,67840E+00	31	3,32009E+12	1,27404E+01	46	9,78917E+11	1,88043E+01

Tabela. 2.3.3.1. Współczynniki uwarunkowania dla macierzy A wyznaczonej zgodnie ze wzorem z zadania 1. i ze wzorem z zadania 2 (dla kilku początkowych wartości n)

3.3.3. Wykresy wartości współczynników uwarunkowania macierzy A z zadania 1 i zadania 2

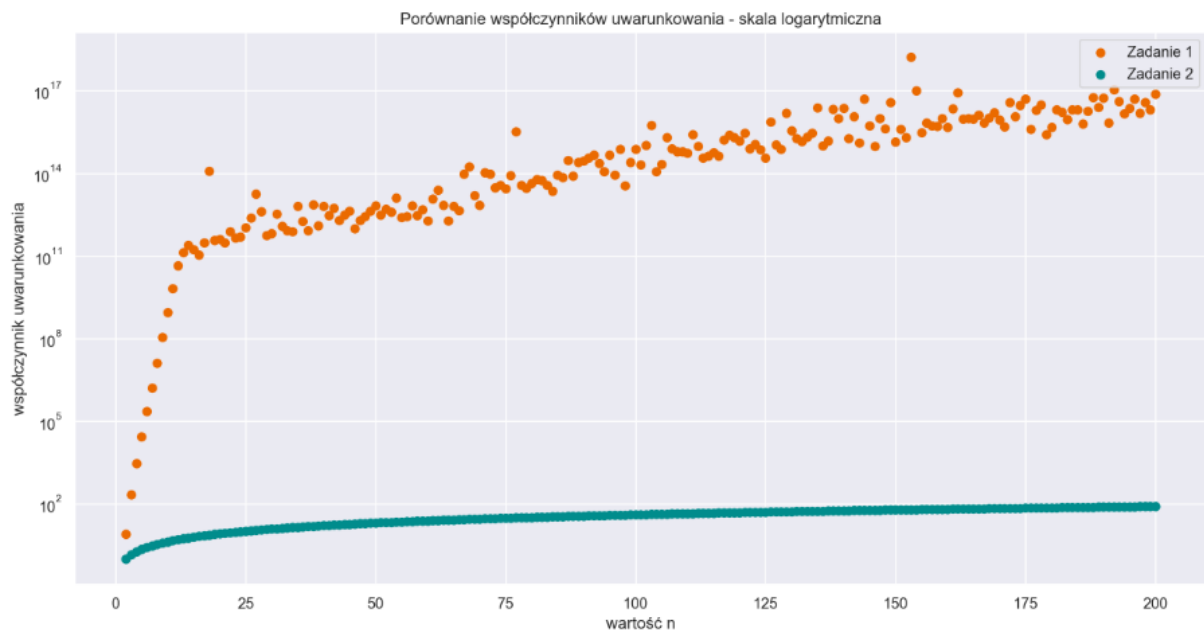
Na wykresach dokładnie widzimy, że uwarunkowanie macierzy A , w przypadku problemu z 1. zadania, jest znacznie gorsze niż w przypadku macierzy z zadania 2.

- Wykres w skali liniowej



Rys. 2.3.3.1. Wykres współczynników uwarunkowania w zależności od n dla macierzy A z zadania 1 i z zadania 2. (w skali liniowej)

- Wykres w skali logarytmicznej



Rys. 2.3.3.2. Wykres współczynników uwarunkowania w zależności od n dla macierzy A z zadania 1 i z zadania 2. (w skali logarytmicznej)

IV. Zadanie 3

1. Wprowadzenie

1.1. Opis problemu

Problem polegał na powtórzeniu eksperymentu z zadań poprzednich. Macierz A , która została mi przydzielona w zadaniu indywidualnym, wypełniana jest zgodnie z poniższymi wzorami:

$$\begin{cases} a_{1,j} = k \\ a_{i,i+1} = \frac{1}{i+m} \\ a_{i,i-1} = \frac{k}{1+m+i} \text{ dla } i > 1 \\ a_{i,j} = \frac{1}{i+j-1} \text{ dla } j < i-1 \text{ oraz } j > i+1 \end{cases} \quad \text{gdzie } i, j = 1, \dots, n$$

(3.1.1.)

Parametry, dla których przeprowadzałem analizę, są równe: $k = 8$, $m = 3$.

W kolejnym kroku należało rozwiązać układ równań metodą przeznaczoną do rozwiązywania układów z macierzą trójdziagonalną. Wykorzystałem do tego celu algorytm Thomasa. Po wykonaniu obliczeń, należało porównać błędy oraz czasy obliczeń i złożoność pamięciową dla metody Gaussa i metody Thomasa. Na koniec należało opisać, w jaki sposób w metodzie dla układów z macierzą trójdziagonalną (Thomasa) przechowywana i wykorzystywana była macierz A .

2. Kryterium pomiaru błędu

Analogicznie, jak w zadaniu 1., do pomiaru błędu wykorzystałem normę maksimum, opisaną w punkcie 2.1..

3. Opracowanie – metoda Gaussa

3.1. Metoda postępowania

Postępowanie jest analogiczne, jak w poprzednim zadaniu.

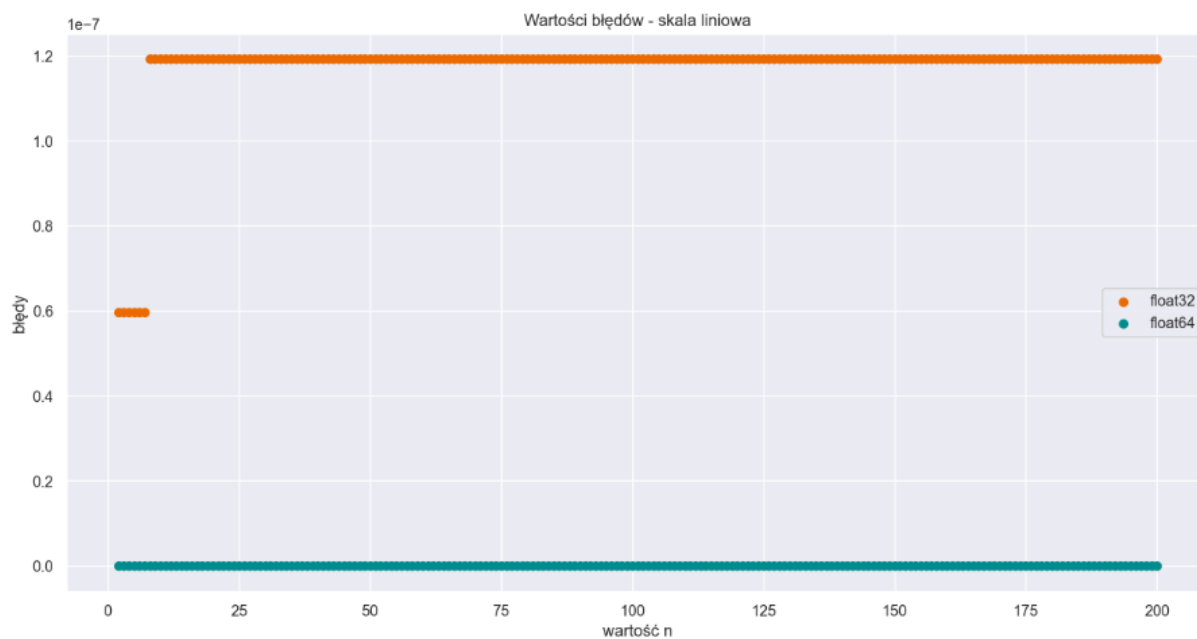
3.2. Wartości błędów oraz czasy obliczeń wektora x

3.2.1. Wykresy błędów obliczeń

Poniżej umieściłem wykresy błędów obliczeń dla zmiennych typu `float32` oraz `float64`.

Co ciekawe, w przeciwieństwie do układów równań z poprzednich zadań, w przypadku układu równań z tego zadania, nie obserwujemy dużych wartości błędów pomiaru, a błędy prawie nie zwiększają się podczas zwiększania wartości n (poza typem `float32`, dla którego, w przypadku, gdy n ma wartość mniejszą niż ok. 10, błędy są zauważalnie mniejsze niż dla większych wartości n).

- Wykres w skali liniowej dla $n = 2, \dots, 200$ (wartości błędów)



Rys. 3.3.2.1. Wykres błędów dla n od 2 do 200 (skala liniowa)

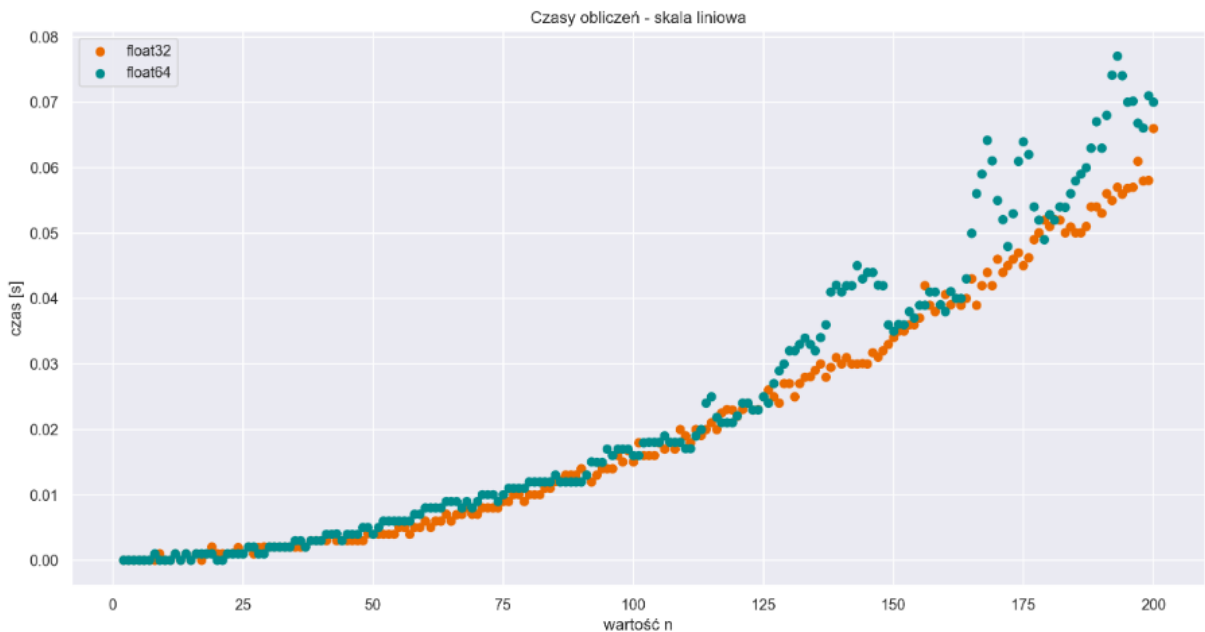
- Wykres w skali logarytmicznej dla $n = 2, \dots, 200$ (wartości błędów)



Rys. 3.3.2.2. Wykres błędów dla n od 2 do 200 (skala logarytmiczna)

3.2.2. Wykres czasów obliczeń

Poniżej umieściłem wykres czasów obliczeń dla zmiennych typu float32 oraz float64.



Rys. 3.3.2.3. Wykres czasów obliczeń wektora dla n od 2 do 200 (skala liniowa)

3.2.3. Zestawienie błędów oraz czasów obliczeń

Zestawienie analogiczne, jak w poprzednich zadaniach.

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	5,96046E-08	2,22045E-16	0,000000E+00	0,000000E+00
3	5,96046E-08	2,22045E-16	0,000000E+00	0,000000E+00
4	5,96046E-08	0,000000E+00	0,000000E+00	0,000000E+00
5	5,96046E-08	1,11022E-16	0,000000E+00	0,000000E+00
6	5,96046E-08	1,11022E-16	0,000000E+00	0,000000E+00
7	5,96046E-08	1,11022E-16	0,000000E+00	0,000000E+00
8	1,19209E-07	2,22045E-16	0,000000E+00	9,99451E-04
9	1,19209E-07	2,22045E-16	9,86576E-04	0,000000E+00
10	1,19209E-07	2,22045E-16	0,000000E+00	0,000000E+00
11	1,19209E-07	2,22045E-16	0,000000E+00	0,000000E+00
12	1,19209E-07	2,22045E-16	1,00017E-03	1,00017E-03
13	1,19209E-07	2,22045E-16	0,000000E+00	0,000000E+00
14	1,19209E-07	2,22045E-16	9,97782E-04	9,97782E-04
15	1,19209E-07	2,22045E-16	0,000000E+00	0,000000E+00
16	1,19209E-07	2,22045E-16	1,00803E-03	9,99212E-04
17	1,19209E-07	2,22045E-16	0,000000E+00	1,00231E-03
18	1,19209E-07	2,22045E-16	9,92298E-04	9,98497E-04
19	1,19209E-07	2,22045E-16	1,99199E-03	1,00112E-03

20	1,19209E-07	2,22045E-16	1,00899E-03	0,00000E+00
21	1,19209E-07	2,22045E-16	9,98259E-04	0,00000E+00
22	1,19209E-07	2,22045E-16	9,99451E-04	1,00064E-03
23	1,19209E-07	2,22045E-16	1,01256E-03	9,99689E-04
24	1,19209E-07	2,22045E-16	1,98197E-03	9,99928E-04
25	1,19209E-07	2,22045E-16	9,96828E-04	1,00207E-03
26	1,19209E-07	2,22045E-16	2,02131E-03	1,99676E-03
27	1,19209E-07	2,22045E-16	9,85622E-04	2,01178E-03
28	1,19209E-07	2,22045E-16	1,98317E-03	9,91583E-04
29	1,19209E-07	2,22045E-16	1,99676E-03	9,94444E-04
30	1,19209E-07	2,22045E-16	2,00152E-03	1,99986E-03
31	1,19209E-07	2,22045E-16	1,99771E-03	2,01464E-03
32	1,19209E-07	2,22045E-16	1,99699E-03	2,00129E-03
33	1,19209E-07	2,22045E-16	1,99580E-03	1,99938E-03
34	1,19209E-07	2,22045E-16	2,00057E-03	1,99652E-03
35	1,19209E-07	2,22045E-16	1,99866E-03	2,99621E-03
36	1,19209E-07	2,22045E-16	2,00009E-03	2,99907E-03
37	1,19209E-07	2,22045E-16	1,99890E-03	1,99580E-03
38	1,19209E-07	2,22045E-16	3,01266E-03	2,99692E-03
39	1,19209E-07	2,22045E-16	3,00932E-03	3,00407E-03
40	1,19209E-07	2,22045E-16	3,00741E-03	2,99525E-03
41	1,19209E-07	2,22045E-16	2,98023E-03	4,00043E-03
42	1,19209E-07	2,22045E-16	3,99256E-03	3,99899E-03
43	1,19209E-07	2,22045E-16	2,98810E-03	4,00758E-03
44	1,19209E-07	2,22045E-16	3,01075E-03	2,99740E-03
45	1,19209E-07	2,22045E-16	3,00050E-03	4,00043E-03
46	1,19209E-07	2,22045E-16	3,01051E-03	3,99327E-03
47	1,19209E-07	2,22045E-16	2,99048E-03	3,99756E-03
48	1,19209E-07	2,22045E-16	3,01051E-03	5,02086E-03
49	1,19209E-07	2,22045E-16	4,00519E-03	5,00393E-03
50	1,19209E-07	2,22045E-16	3,98970E-03	4,00114E-03

Tabela. 3.3.2.1. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla n od 2 do 50

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
10	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
20	1,19209E-07	2,22045E-16	1,00899E-03	0,00000E+00
30	1,19209E-07	2,22045E-16	2,00152E-03	1,99986E-03
40	1,19209E-07	2,22045E-16	3,00741E-03	2,99525E-03
50	1,19209E-07	2,22045E-16	3,98970E-03	4,00114E-03
60	1,19209E-07	2,22045E-16	5,98907E-03	7,99775E-03

70	1,19209E-07	2,22045E-16	7,00045E-03	8,99076E-03
80	1,19209E-07	2,22045E-16	1,00188E-02	1,19958E-02
90	1,19209E-07	2,22045E-16	1,39995E-02	1,19801E-02
100	1,19209E-07	2,22045E-16	1,50409E-02	1,60224E-02
110	1,19209E-07	2,22045E-16	1,90113E-02	1,70295E-02
120	1,19209E-07	2,22045E-16	2,19941E-02	2,20320E-02
130	1,19209E-07	2,22045E-16	2,69859E-02	3,20010E-02
140	1,19209E-07	2,22045E-16	3,00190E-02	4,10175E-02
150	1,19209E-07	2,22045E-16	3,40102E-02	3,50208E-02
160	1,19209E-07	2,22045E-16	4,06313E-02	3,80383E-02
170	1,19209E-07	2,22045E-16	4,60091E-02	5,49963E-02
180	1,19209E-07	2,22045E-16	5,10097E-02	5,27821E-02
190	1,19209E-07	2,22045E-16	5,30150E-02	6,30012E-02
200	1,19209E-07	2,22045E-16	6,59950E-02	7,00004E-02

Tabela. 3.3.2.2. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla wybranych wartości n

4. Opracowanie – metoda Thomasa

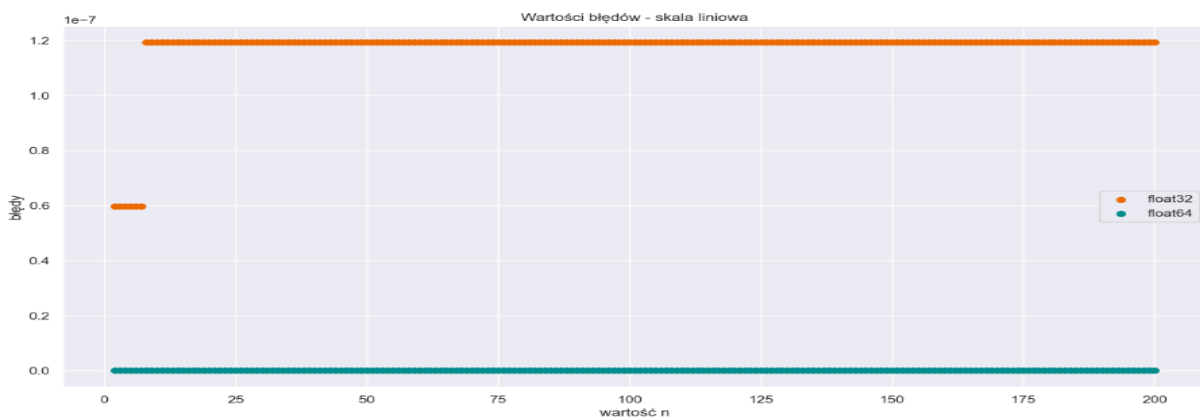
4.1. Metoda postępowania

Do obliczenia wektora x , wykorzystuję algorytm Thomasa, który działa analogicznie do eliminacji Gaussa. Algorytm ten wykorzystuje fakt, iż poza trzema głównymi przekątnymi, pozostałe wartości w macierzy są zerami. Algorytm Thomasa, zamiast iterować po wszystkich wierszach pod przekątną główną, operuje jedynie na wartości głównej przekątnej i wartości z przekątnej pod nią (z tej samej kolumny). Dzięki temu, dla każdego pivota (kolejnej wartości z przekątnej głównej) wykonane zostanie $O(1)$ operacji, a ponieważ przekątna główna ma n elementów, finalnie złożoność algorytmu Thomasa jest rzędu $O(n)$ (a więc jest dużo niższa niż złożoność algorytmu Gaussa, który działa ze złożonością $O(n^3)$).

4.2. Wartości błędów oraz czasy obliczeń wektora x

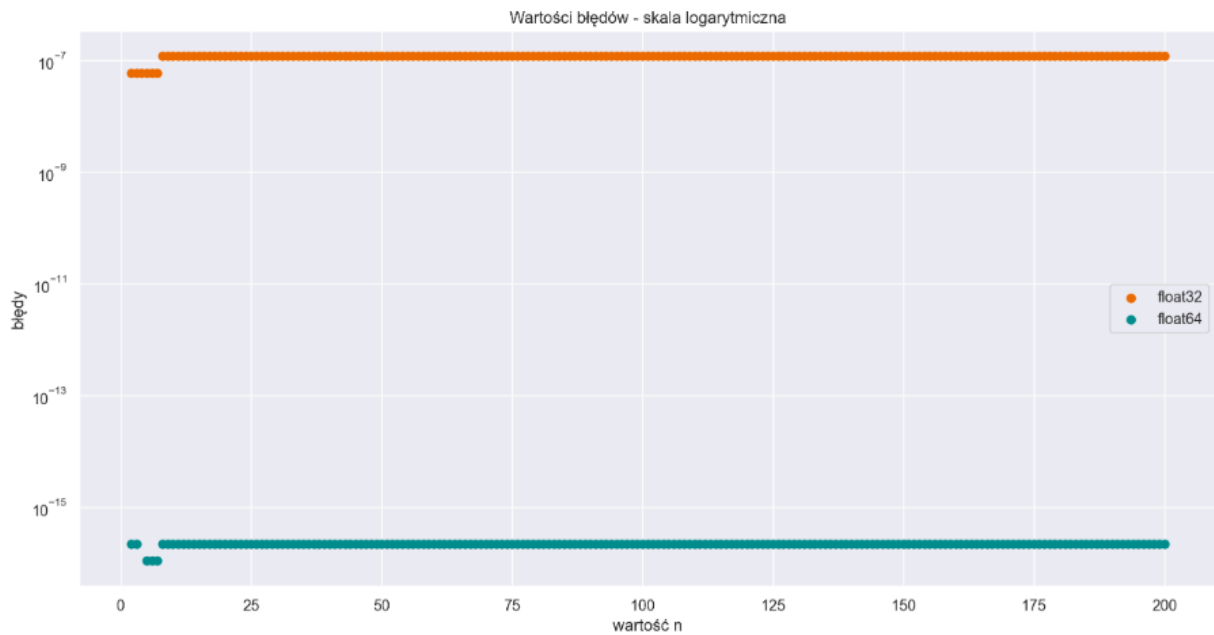
4.2.1. Wykresy błędów obliczeń

- Wykres w skali liniowej dla $n = 2, \dots, 200$ (wartości błędów)



Rys. 3.4.2.1. Wykres błędów dla n od 2 do 200 (skala liniowa)

▪ Wykres w skali logarytmicznej dla $n = 2, \dots, 200$ (wartości błędów)



Rys. 3.4.2.2. Wykres błędów dla n od 2 do 200 (skala logarytmiczna)

4.2.2. Wykres czasów obliczeń

Wykres czasów obliczeń jest bardzo ciekawy z tego względu, że dla większości wartości n , czas ten jest bliski zeru, jednak istnieje również część wartości, dla których czas obliczeń znacząco odstaje od pozostałych (jest zauważalnie dłuższy).



Rys. 3.4.2.3. Wykres czasów obliczeń wektora dla n od 2 do 200 (skala liniowa)

4.2.3. Zestawienie błędów oraz czasów obliczeń

Zestawienie analogiczne, jak w poprzednich zadaniach.

Jak możemy zauważyć, czas obliczeń jest tak niewielki, że często otrzymujemy wartość czasu równą 0.

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	5,96046E-08	2,22045E-16	0,00000E+00	0,00000E+00
3	5,96046E-08	2,22045E-16	0,00000E+00	0,00000E+00
4	5,96046E-08	0,00000E+00	0,00000E+00	0,00000E+00
5	5,96046E-08	1,11022E-16	0,00000E+00	0,00000E+00
6	5,96046E-08	1,11022E-16	0,00000E+00	0,00000E+00
7	5,96046E-08	1,11022E-16	0,00000E+00	0,00000E+00
8	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
9	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
10	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
11	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
12	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
13	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
14	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
15	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
16	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
17	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
18	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
19	1,19209E-07	2,22045E-16	9,99928E-04	0,00000E+00
20	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
21	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
22	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
23	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
24	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
25	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
26	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
27	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
28	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
29	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
30	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
31	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
32	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
33	1,19209E-07	2,22045E-16	9,99451E-04	0,00000E+00
34	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
35	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
36	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
37	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00

38	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
39	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
40	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
41	1,19209E-07	2,22045E-16	1,00636E-03	0,00000E+00
42	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
43	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
44	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
45	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
46	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
47	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
48	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
49	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00
50	1,19209E-07	2,22045E-16	0,00000E+00	0,00000E+00

Tabela. 3.4.2.1. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla n od 2 do 50

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
10	3,09944E-06	1,66533E-15	0,00000E+00	0,00000E+00
20	9,05991E-06	2,46470E-14	9,94921E-04	1,00017E-03
30	2,69413E-05	4,86278E-14	9,92298E-04	9,99689E-04
40	4,81009E-05	1,14575E-13	2,99716E-03	1,99986E-03
50	6,89626E-05	1,58762E-13	4,01711E-03	3,99923E-03
60	1,42455E-04	2,82219E-13	5,99980E-03	4,98986E-03
70	1,80244E-04	2,95097E-13	6,99997E-03	8,00991E-03
80	2,23160E-04	6,05405E-13	1,20127E-02	1,00117E-02
90	3,96729E-04	7,64944E-13	1,29950E-02	1,70288E-02
100	2,39253E-04	6,66356E-13	1,60003E-02	2,20249E-02
110	5,50270E-04	1,09135E-12	1,85983E-02	2,70109E-02
120	5,17368E-04	1,54921E-12	2,20242E-02	3,27213E-02
130	4,42505E-04	2,03171E-12	2,40176E-02	2,89917E-02
140	9,00149E-04	2,34979E-12	2,90105E-02	3,09546E-02
150	1,06287E-03	2,71927E-12	4,00014E-02	3,69990E-02
160	1,28424E-03	3,06777E-12	4,99797E-02	4,70030E-02
170	1,03962E-03	3,78164E-12	5,80125E-02	7,69944E-02
180	1,52260E-03	7,36389E-12	6,50294E-02	6,00295E-02
190	1,52230E-03	1,27474E-11	7,20117E-02	7,90308E-02

Tabela. 3.4.2.2. Błędy obliczeń oraz czasy obliczeń dla zmiennych typu float32 oraz float64 dla wybranych wartości n

5. Porównanie metody Gaussa i metody Thomasa

5.1. Porównanie błędów

Zamieszczone niżej tabele zawierają porównanie błędów obliczeń wartości wektora x , przy pomocy metody Gaussa oraz metody Thomasa. Jak możemy łatwo zauważyć, błędy są takie same dla obu metod obliczania rozwiązania układu równań.

Wartość n	Metoda Gaussa		Metoda Thomasa	
	float32	float64	float32	float64
2	5,96046E-08	2,22045E-16	5,96046E-08	2,22045E-16
3	5,96046E-08	2,22045E-16	5,96046E-08	2,22045E-16
4	5,96046E-08	0,00000E+00	5,96046E-08	0,00000E+00
5	5,96046E-08	1,11022E-16	5,96046E-08	1,11022E-16
6	5,96046E-08	1,11022E-16	5,96046E-08	1,11022E-16
7	5,96046E-08	1,11022E-16	5,96046E-08	1,11022E-16
8	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
9	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
10	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
11	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
12	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
13	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
14	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
15	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
16	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
17	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
18	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
19	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
20	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
21	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
22	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
23	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
24	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
25	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
26	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
27	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
28	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
29	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
30	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
31	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
32	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
33	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
34	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
35	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16

36	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
37	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
38	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
39	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
40	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
41	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
42	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
43	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
44	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
45	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
46	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
47	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
48	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
49	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
50	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16

Tabela. 3.5.1.1. Porównanie błędów obliczeń wykonanych, przy pomocy metody Gaussa oraz metody Thomasa dla n od 2 do 50

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
10	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
20	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
30	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
40	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
50	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
60	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
70	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
80	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
90	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
100	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
110	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
120	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
130	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
140	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
150	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
160	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
170	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
180	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
190	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16
200	1,19209E-07	2,22045E-16	1,19209E-07	2,22045E-16

Tabela. 3.5.1.2. Porównanie błędów obliczeń wykonanych, przy pomocy metody Gaussa oraz metody Thomasa dla wybranych wartości n

5.2. Porównanie czasów obliczeń

Zamieszczone niżej tabele zawierają porównanie czasów obliczeń wartości wektora x , przy pomocy metody Gaussa oraz metody Thomasa. Widzimy, że czasy obliczeń są znacznie mniejsze w przypadku metody Thomasa niż w przypadku metody Gaussa.

Wartość n	Metoda Gaussa		Metoda Thomasa	
	float32	float64	float32	float64
2	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
3	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
4	9,98735E-04	0,00000E+00	0,00000E+00	0,00000E+00
5	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
6	0,00000E+00	0,00000E+00	0,00000E+00	1,00398E-03
7	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
8	9,99212E-04	0,00000E+00	0,00000E+00	0,00000E+00
9	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
10	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
11	1,00255E-03	0,00000E+00	0,00000E+00	0,00000E+00
12	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
13	0,00000E+00	9,99689E-04	0,00000E+00	0,00000E+00
14	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
15	9,98259E-04	1,00207E-03	0,00000E+00	0,00000E+00
16	0,00000E+00	0,00000E+00	9,97543E-04	0,00000E+00
17	9,99689E-04	9,88245E-04	0,00000E+00	0,00000E+00
18	9,99928E-04	1,05453E-03	0,00000E+00	9,99451E-04
19	1,00040E-03	9,45330E-04	0,00000E+00	0,00000E+00
20	9,99928E-04	0,00000E+00	0,00000E+00	0,00000E+00
21	1,00183E-03	0,00000E+00	0,00000E+00	0,00000E+00
22	9,99451E-04	9,46999E-04	0,00000E+00	0,00000E+00
23	1,01376E-03	1,07551E-03	0,00000E+00	0,00000E+00
24	1,97673E-03	9,71317E-04	0,00000E+00	0,00000E+00
25	1,01089E-03	9,99928E-04	0,00000E+00	0,00000E+00
26	1,01042E-03	1,00017E-03	0,00000E+00	0,00000E+00
27	2,01368E-03	9,99928E-04	0,00000E+00	0,00000E+00
28	1,00112E-03	9,85146E-04	0,00000E+00	0,00000E+00
29	1,00756E-03	1,99676E-03	0,00000E+00	0,00000E+00
30	2,01201E-03	2,00057E-03	0,00000E+00	1,00183E-03
31	1,97816E-03	2,01774E-03	0,00000E+00	0,00000E+00
32	2,00009E-03	1,99461E-03	1,00493E-03	0,00000E+00
33	3,01266E-03	1,98698E-03	0,00000E+00	0,00000E+00
34	2,97523E-03	2,01821E-03	0,00000E+00	0,00000E+00
35	2,01201E-03	2,00033E-03	0,00000E+00	0,00000E+00
36	3,00288E-03	1,99699E-03	0,00000E+00	0,00000E+00
37	3,01313E-03	3,99423E-03	0,00000E+00	0,00000E+00

38	2,99668E-03	1,99962E-03	0,00000E+00	0,00000E+00
39	3,00312E-03	2,00009E-03	0,00000E+00	0,00000E+00
40	2,99549E-03	1,99461E-03	0,00000E+00	0,00000E+00
41	3,00026E-03	2,01845E-03	0,00000E+00	0,00000E+00
42	2,98238E-03	2,98190E-03	0,00000E+00	0,00000E+00
43	3,00026E-03	3,01862E-03	0,00000E+00	0,00000E+00
44	3,99995E-03	2,98166E-03	0,00000E+00	0,00000E+00
45	3,98135E-03	3,02172E-03	0,00000E+00	0,00000E+00
46	2,99573E-03	2,98929E-03	0,00000E+00	0,00000E+00
47	2,99835E-03	2,99501E-03	1,01423E-03	0,00000E+00
48	4,00352E-03	4,02641E-03	0,00000E+00	0,00000E+00
49	4,00496E-03	4,98152E-03	0,00000E+00	1,00255E-03
50	3,99232E-03	4,02069E-03	0,00000E+00	0,00000E+00

Tabela. 3.5.2.1. Porównanie czasów obliczeń wykonanych, przy pomocy metody Gaussa oraz metody Thomasa dla n od 2 do 50

Wartość n	Błędy		Czasy obliczeń [s]	
	float32	float64	float32	float64
2	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
10	0,00000E+00	0,00000E+00	0,00000E+00	0,00000E+00
20	9,99928E-04	0,00000E+00	0,00000E+00	0,00000E+00
30	2,01201E-03	2,00057E-03	0,00000E+00	1,00183E-03
40	2,99549E-03	1,99461E-03	0,00000E+00	0,00000E+00
50	3,99232E-03	4,02069E-03	0,00000E+00	0,00000E+00
60	5,99051E-03	5,99980E-03	1,00398E-03	0,00000E+00
70	7,00045E-03	7,99966E-03	0,00000E+00	0,00000E+00
80	1,00191E-02	1,90101E-02	0,00000E+00	0,00000E+00
90	1,39885E-02	1,19977E-02	9,96828E-04	0,00000E+00
100	1,46759E-02	1,80039E-02	0,00000E+00	1,00398E-03
110	1,80073E-02	2,30043E-02	1,00517E-03	0,00000E+00
120	2,12951E-02	2,20413E-02	1,00541E-03	9,97543E-04
130	2,70078E-02	5,29988E-02	0,00000E+00	1,00112E-03
140	3,29955E-02	5,00000E-02	1,00756E-03	9,92298E-04
150	3,29866E-02	4,79941E-02	0,00000E+00	0,00000E+00
160	4,29890E-02	4,00007E-02	1,00040E-03	1,00374E-03
170	4,50084E-02	6,20031E-02	9,98974E-04	1,00493E-03
180	5,10080E-02	5,80389E-02	9,97782E-04	1,00160E-03
190	5,40197E-02	5,70209E-02	0,00000E+00	1,00136E-03

Tabela. 3.5.2.2. Porównanie czasów obliczeń wykonanych, przy pomocy metody Gaussa oraz metody Thomasa dla wybranych wartości n

6. Sposób przechowywania i wykorzystywania macierzy A

W przypadku metody dla macierzy trójdzielnej (metody Thomasa), zdecydowałem się przechowywać macierz w taki sam sposób jak w przypadku metody Gaussa. Oczywiście, ponieważ przechowywana jest cała macierz kwadratowa o wymiarach $n \times n$, dla obu metod otrzymujemy kwadratową złożoność pamięciową. Ponieważ macierz trójdzielna zawiera wartości tylko na głównej przekątnej oraz przekątnej nad nią i pod nią, można by było przechowywać jedynie te wartości w postaci 3 kolumn, odpowiednio modyfikując algorytm Thomasa w taki sposób, żeby uwzględnić przesunięcie kolejnych wierszy o 1 kolumnę w lewo względem poprzedniego wiersza. Tym sposobem, dałoby się otrzymać złożoność pamięciową rzędu $O(n)$.

Jeżeli założylibyśmy, że dane, które otrzymujemy na wejściu, nie są wliczane do złożoności pamięciowej oraz, że funkcja, przy pomocy której rozwiązujemy układ równań, nie może modyfikować danych wejściowych (macierze mają pozostać niezmienione), wówczas, w przypadku metody Gaussa, konieczne jest skopiowanie całej macierzy A oraz wektora b (mamy złożoność pamięciową $O(n^2)$), natomiast, w przypadku metody Thomasa, wystarczy jedynie zaalokować $O(n)$ pamięci na rozwiązanie i obliczane wartości.

V. Wnioski

- Przeprowadzona analiza pokazała, jak niebagatelne znaczenie ma uwarunkowanie problemu, które w znaczący sposób wpływa na dokładność uzyskiwanych wyników,
- Bardzo duży wpływ na dokładność wyniku mają także błędy zaokrągleń, spowodowane niewystarczającą precyzją zastosowanego typu zmiennych. Mogliśmy dokładnie zaobserwować na wykresie Rys. 2.3.2.1., jak bardzo różniące się od siebie rozwiązania otrzymujemy dla zmiennych typu `float32` oraz dla zmiennych `float64`.
- Metoda Gaussa pozwala na stosunkowo łatwe wyznaczanie rozwiązania układu równań, jednakże ma wysoką złożoność obliczeniową oraz pamięciową,
- Metoda Thomasa pozwala na bardzo szybkie obliczenie rozwiązania układu równań, jednakże może być zastosowana wyłącznie dla szczególnego typu układu równań, w którym wszystkie wartości poza leżącymi na głównej przekątnej lub na przekątnej pod albo nad główną przekątną, muszą mieć wartość równą 0.