

Ćwiczenie 6. Układy równań liniowych – metody bezpośrednie.

1. Informacje techniczne

Zadanie zostało wykonane w języku Python3 na komputerze z systemem Windows 11, procesorem Intel i7-11800H, 2x8GB pamięci RAM o szybkości 3200MHz.

Wykorzystane biblioteki w zadaniu:

- `import matplotlib.pyplot as plt`
- `import numpy as np`
- `import pandas as pd`
- `import time`

2. Treść ćwiczenia

Dany jest układ równań liniowych $Ax = b$.

2.1. Zadanie 1

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

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

Przyjmij wektor x jako dowolną n -elementową permutację ze zbioru $\{1, -1\}$ i oblicz wektor b . Następnie metodą eliminacji Gaussa rozwiąż układ równań liniowych $Ax = b$ (przyjmując jako niewiadomą wektor 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 x obliczony z x zadany). Przeprowadź eksperymenty dla różnych rozmiarów układu.

2.2. Zadanie 2

Powtórz eksperyment dla macierzy zadanej wzorem:

$$\begin{cases} a_{ij} = \frac{2i}{j} \text{ dla } j \geq i \\ a_{ij} = a_{ji} \text{ dla } j < i \end{cases} \quad i, j = 1, \dots, n$$

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

2.3. Zadanie 3

Powtórz eksperyment dla macierzy zadanej wzorem:

$$\begin{cases} a_{i,i} = -m * i - k \\ a_{i,i+1} = i \\ a_{i,i-1} = \frac{m}{i} \text{ dla } i > 1 \\ a_{i,j} = 0 \text{ dla } j < i - 1 \text{ oraz } j > i + 1 \end{cases} \quad i, j = 1, \dots, n$$

gdzie $m = 4$, $k = 7$.

Następnie rozwiąż układ metodą przeznaczoną do rozwiązywania układów z macierzą trójdagonalną. 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 trójdagonalną przechowywano i wykorzystywano macierz A.

3. Realizacja ćwiczenia

W celu wykonania ćwiczenia, wykorzystana została m.in. biblioteka numpy. Umożliwiła ona przeprowadzenie analizy dla różnej precyzji liczb zmiennoprzecinkowych. Sprawdzone zostały typy float32 oraz float64 (odpowiednio zapis 32-bitowy, 64-bitowy).

3.1. Obliczanie błędów

W celu wyznaczenia dokładności otrzymywanych wyników porównywane były wektory \hat{x} obliczony z x zadany. Błędy były obliczane na dwa sposoby:

- maksymalna różnica współrzędnych wektorów

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

gdzie x_i – i ta współrzędna wektora x

- różnica długości wektorów obliczana za pomocą funkcji `linalg.norm` z biblioteki numpy

$$\|x - \hat{x}\| = \sqrt{\sum_{i=1}^n (x_i - \hat{x}_i)^2}$$

3.2. Współczynnik uwarunkowania

Współczynnik uwarunkowania macierzy to miara, jak bardzo wynik operacji na macierzy może zmienić się w wyniku niewielkich zmian w jej elementach. W kontekście rozwiązywania układów równań liniowych, wysoki współczynnik uwarunkowania macierzy może prowadzić do dużych błędów w rozwiązaniu. Do obliczania współczynników uwarunkowania macierzy użyta została funkcja `linalg.cond` z biblioteki numpy.

3.3. Metoda postępowania

Zgodnie z treścią zadania na początku wybieramy nasz wektor x , będący n -elementową permutacją ze zbioru $\{1, -1\}$. Następnie dla różnej liczby równań n , tworzymy macierz A i obliczamy wektor b . Mając wektor b , obliczamy metodą Gaussa lub Thomasa wektor \hat{x} i porównujemy go z wybranym wcześniej wektorem x obliczając przedstawione błędy w punkcie 3.1. Analizę przeprowadzamy dla różnych precyzji, a w przypadku porównania metod Gaussa oraz Thomasa, mierzymy również czasy tych algorytmów. W przypadku zadań 1 i 2 obliczamy i porównujemy również współczynniki uwarunkowania macierzy.

Uwaga: w przypadku zadań 1 oraz 2 będziemy rozwiązywali układ tylko metodą Gaussa. W zadaniu 3 porównamy obie metody Gaussa i Thomasa.

4. Wyniki

Obliczenia zostały wykonane dla n (macierz $n \times n$) z przedziału $\langle 2, 200 \rangle$. W tabelach przedstawione zostały wyniki tylko dla wybranych wartości n .

4.1. Zadanie 1

4.1.1 Obliczone wartości błędów

4.1.1.1 Tabela z wartościami błędów metoda Gaussa

n	Precyzja			
	Maksymalna różnica		Długość różnicy wektorów	
	Float32	Float64	Float32	Float64
2	0.000000	0.000000e+00	0.000000	0.000000e+00
3	0.000002	0.000000e+00	0.000003	0.000000e+00
4	0.000003	1.929568e-13	0.000004	2.465697e-13
5	0.000317	6.979972e-12	0.000470	9.891256e-12
6	0.044535	2.437184e-10	0.065090	3.568660e-10
7	0.375060	8.735740e-09	0.540226	1.259790e-08
8	1.606400	7.057528e-08	2.320680	1.062511e-07
9	4.904638	5.060815e-07	8.689874	8.089723e-07
10	5.521448	7.760520e-05	7.927493	1.225307e-04
11	2.764232	6.263925e-03	4.330271	1.001833e-02
12	4.422953	6.821455e-01	6.361421	1.150363e+00
13	3.387240	2.072562e+01	4.592288	3.507859e+01
14	68.547005	2.059340e+01	120.499543	3.627481e+01
15	12.386322	1.046186e+01	24.339989	1.695537e+01
16	9.191536	1.712022e+01	17.428444	2.802811e+01
17	5.753284	1.842019e+01	10.021981	2.835910e+01
18	8.454904	1.146170e+02	15.751236	1.688305e+02
19	5.814231	4.168393e+01	9.394214	6.474019e+01
20	2.386188	7.018891e+01	5.356794	1.116974e+02
30	3.961026	3.021416e+02	8.478085	5.661564e+02
50	4.850247	2.222858e+02	14.277783	4.728852e+02
75	18.235489	1.077184e+02	71.292337	3.589158e+02
100	214.694931	1.323177e+03	731.127423	5.441312e+03
150	8.468469	6.474255e+02	36.390169	2.357166e+03
200	8.324449	2.225555e+03	36.825218	1.160170e+04

Tabela 4.1.1.1 Wartości obliczonych błędów dla zadania 1

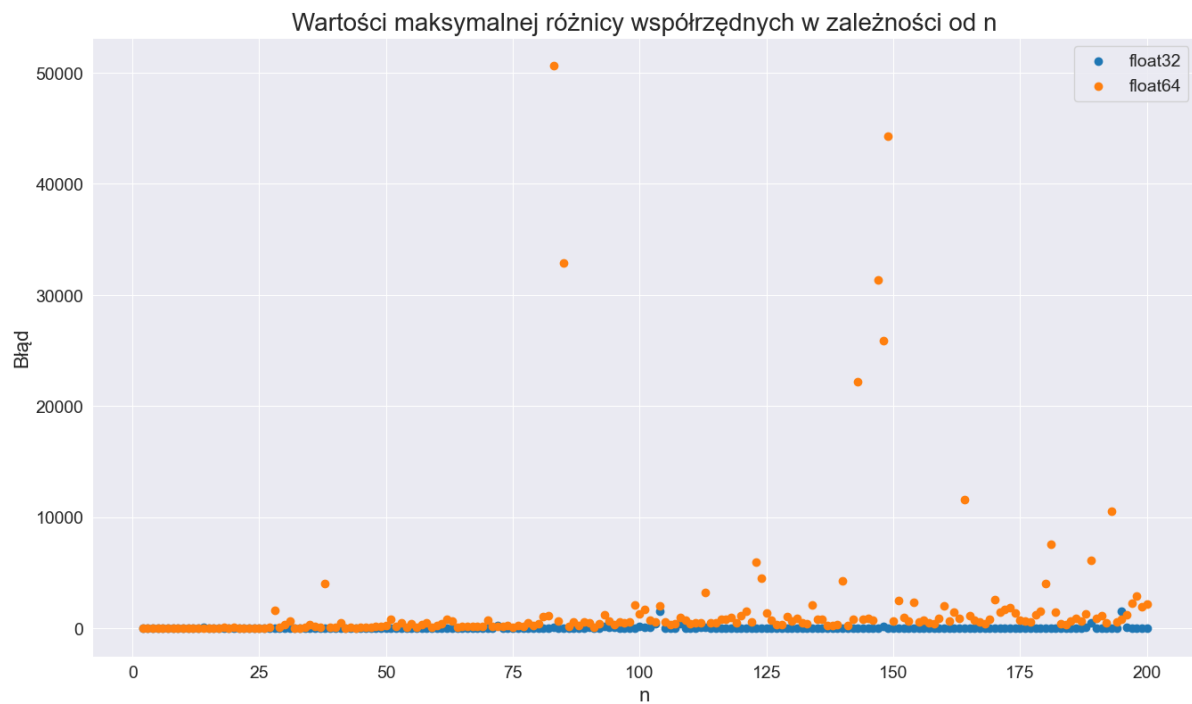
Jak możemy zobaczyć w tabeli 4.1.1.1 dla n mniejszego od 13 dla obu rodzajów błędów, są one znacznie mniejsze dla float64 niż dla float32. Różnica ta liniowo maleje aż dla $n = 13$ mamy ten sam rząd wyników, a dla większych n otrzymujemy zazwyczaj większe błędy dla float64.

Dla float32 błędy przestają szybko rosnąć przy $n = 10$, a dla float64 dla $n = 13$, później stają się one bardziej losowe, co wynika z złego uwarunkowania macierzy z zadania 1.

4.1.1.2 Wykresy

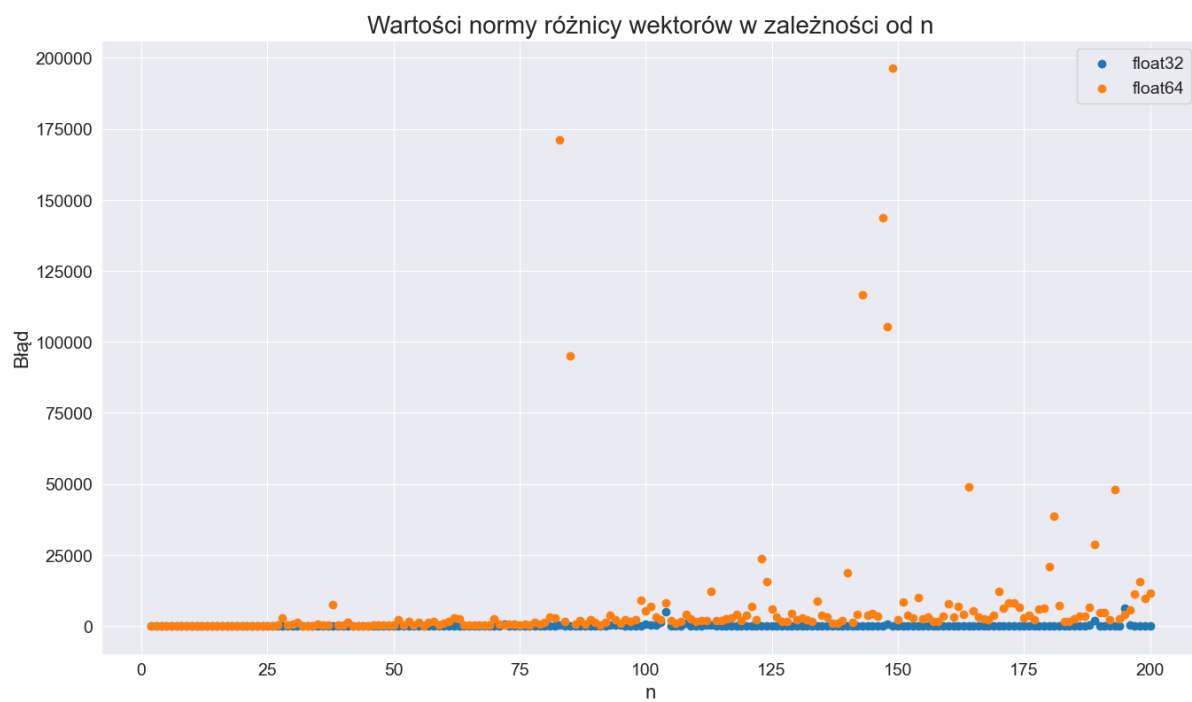
Na poniższych wykresach zestawione zostały dane z tabeli 4.1.1.1.

Wykres wartości maksymalnej różnicy dla zadania 1



Wykres 4.1.1.2.1

Wykres wartości normy różnicy wektorów dla zadania 1



Wykres 4.1.1.2.2

4.2. Zadanie 2

Przeprowadzona została taka sama analiza co w podpunkcie 4.1. Następnie w celu wyjaśnienia, skąd biorą się tak duże różnice w otrzymywanych błędach dla układów z zadania 1 i 2 obliczone zostały współczynniki uwarunkowania macierzy i zestawione ze sobą.

4.2.1 Obliczone wartości błędów

4.2.1.1 Tabela z wartościami błędów metoda Gaussa.

n	Precyzja			
	Maksymalna różnica		Długość różnicy wektorów	
	Float32	Float64	Float32	Float64
2	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
3	1.192093e-07	2.220446e-16	1.192093e-07	2.482534e-16
4	1.192093e-07	2.220446e-16	1.332800e-07	2.482534e-16
5	1.192093e-07	2.220446e-16	2.149076e-07	3.330669e-16
6	2.384186e-07	4.440892e-16	3.039252e-07	5.978734e-16
7	1.013279e-06	6.661338e-16	1.231669e-06	8.950904e-16
8	1.132488e-06	4.551914e-15	1.383804e-06	5.564422e-15
9	1.013279e-06	2.331468e-15	1.233110e-06	2.817430e-15
10	2.920628e-06	2.331468e-15	4.336417e-06	3.306527e-15
11	2.563000e-06	4.218847e-15	3.729460e-06	6.745005e-15
12	2.384186e-06	1.287859e-14	3.918524e-06	2.127750e-14
13	3.814697e-06	1.398881e-14	5.852800e-06	2.234887e-14
14	5.006790e-06	1.465494e-14	8.834567e-06	2.376240e-14
15	5.781651e-06	1.554312e-14	1.124602e-05	2.548657e-14
16	4.470348e-06	1.865175e-14	8.528881e-06	3.409542e-14
17	5.960464e-06	1.887379e-14	1.135371e-05	3.481749e-14
18	7.629395e-06	2.020606e-14	1.517661e-05	3.564456e-14
19	6.794930e-06	2.042810e-14	1.248031e-05	3.709208e-14
20	5.424023e-06	3.019807e-14	1.199594e-05	4.395852e-14
30	1.394749e-05	6.461498e-14	3.195532e-05	1.181835e-13
50	4.196167e-05	1.625367e-13	1.106974e-04	3.665881e-13
75	1.239777e-04	3.538281e-13	3.130642e-04	1.139206e-12
100	2.617836e-04	1.071587e-12	7.505489e-04	3.068362e-12
150	6.015301e-04	3.041900e-12	2.398110e-03	9.479360e-12
200	8.723140e-04	6.788792e-12	3.997014e-03	2.488266e-11

Tabela 4.2.1.1 Wartości obliczonych błędów dla zadania 2

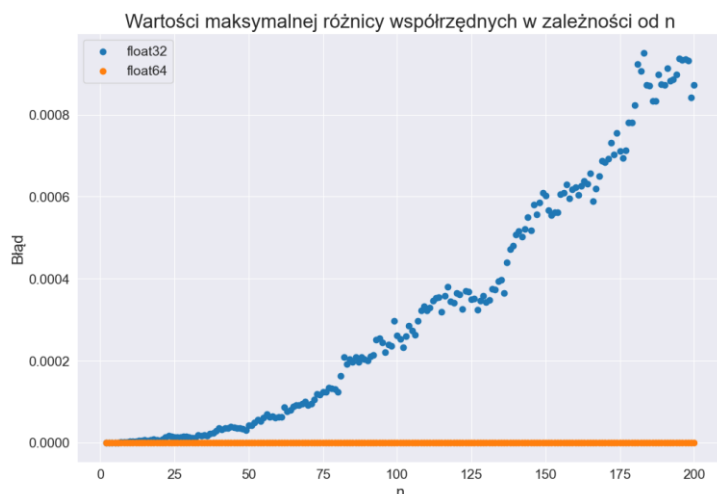
Jak od razu możemy zobaczyć w tabeli 4.2.1.1 uzyskane błędy są znacznie mniejsze niż w tabeli 4.1.1.1. Nawet dla $n = 200$ otrzymujemy wyniki rzędu 10^{-4} dla float32 oraz 10^{-12} dla float64. Dla obu precyzji otrzymujemy wysoką dokładność, jednak w przypadku float64 jest

ona kilka rzędów większa. Otrzymywane błędy zwiększają się co raz szybciej wraz z wzrostem n . Nie zachodzi jednak sytuacja jak w przypadku zadania 1, że ich wartości z powodu złego uwarunkowania stawały się losowe.

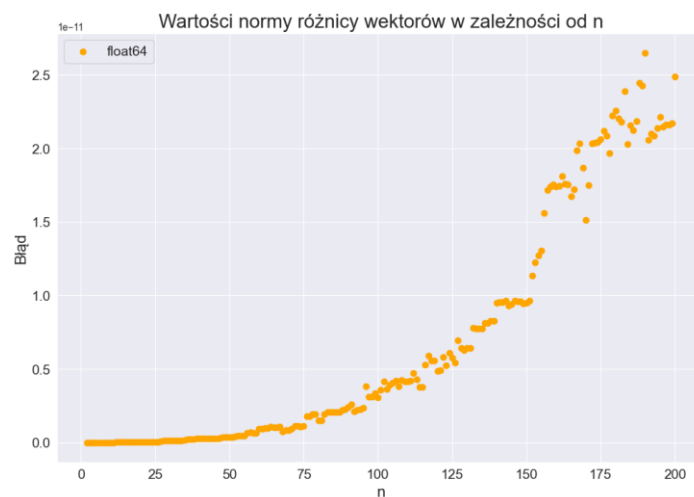
4.2.1.2 Wykresy

Na poniższych wykresach przedstawione zostały dane z tabeli 4.2.1.1.

Wykresy wartości maksymalnej różnicy dla zadania 2



Wykresy wartości normy różnicy wektorów dla zadania 2



4.2.2 Porównanie uwarunkowania układów z zadania 1 i zadania 2.

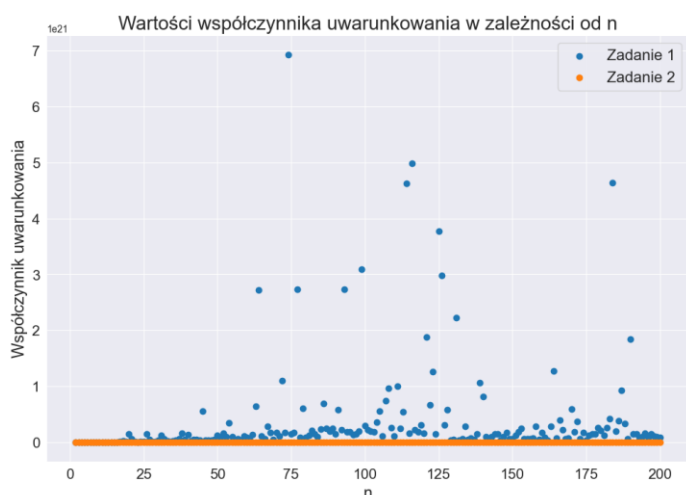
Zgodnie z podpunktem 3.2 obliczony został współczynnik uwarunkowania macierzy z zadań 1 i 2.

4.2.2.1 Tabela z obliczonymi współczynnikami uwarunkowania

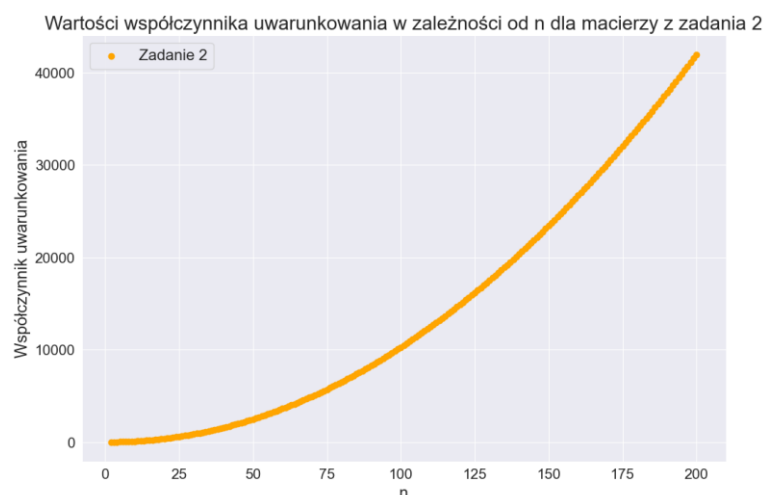
n	Zadanie 1	Zadanie 2
2	1.409572e+01	3.000000
3	4.829221e+02	6.663295
4	1.703290e+04	12.206264
5	5.918771e+05	19.655490
6	2.037851e+07	29.040429
7	6.980461e+08	40.387341
8	2.383968e+10	53.717162
9	8.126917e+11	69.046612
10	2.767444e+13	86.389316
11	9.440475e+14	105.756602
12	3.296147e+16	127.158056
13	7.996306e+17	150.601909
14	5.948189e+17	176.095322
15	7.497220e+17	203.644586
16	6.157433e+17	233.255285
17	1.423288e+19	264.932412
18	1.749228e+19	298.680465
19	1.481043e+19	334.503518
20	1.451349e+20	372.405284
30	3.053671e+19	866.387472
50	1.219880e+20	2487.031745
75	1.407996e+20	5709.230997
100	2.934614e+20	10268.203175
150	5.951074e+19	23413.044186
200	8.594893e+19	41942.698975

Tabela 4.2.2.1 Wartości współczynników uwarunkowania

4.2.2.2 Wykresy



Wykres 4.2.2.2.1



Wykres 4.2.2.2.2

Jak możemy zobaczyć w tabeli 4.2.2.1 oraz na powyższych wykresach, współczynnik uwarunkowania w przypadku zadania 1 bardzo gwałtownie rośnie aż do $n = 13$, a później, podobnie jak w przypadku błędów, wyniki są bardziej losowe. W przypadku zadania 2 współczynnik uwarunkowania jest kilka lub kilkanaście rzędów mniejszy i rośnie co raz szybciej wraz z wzrostem n .

Widzimy zatem, że układ z zadania 2 jest znacznie lepiej uwarunkowany niż układ z zadania 1, co ma przełożenie na dokładność rozwiązań układów równań z obu tych zadań.

Współczynnik uwarunkowania jest kilkanaście rzędów mniejszy – otrzymywane wyniki są stabilniejsze i kilka rzędów dokładniejsze.

4.3. Zadanie 3

Na początku przeprowadzona została analiza jak dla zadań 1 i 2. Następnie tej samej analizie została poddana metoda Thomasa, a na końcu obie metody zostały z sobą porównane.

4.3.1 Obliczone wartości błędów

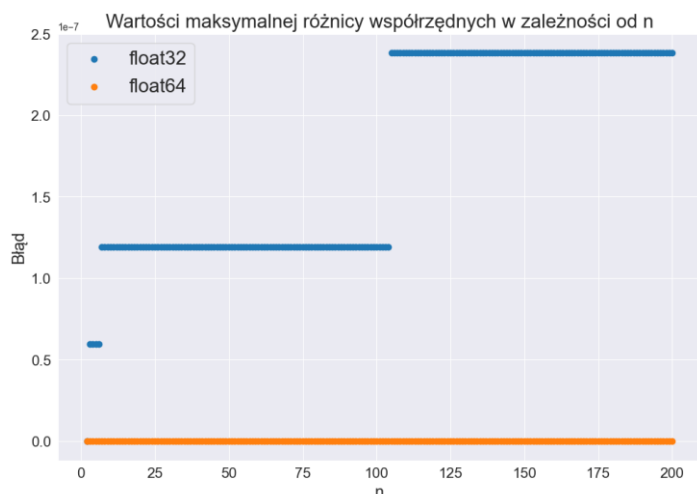
4.3.1.1 Tabela z wartościami błędów metoda Gaussa

n	Precyzja			
	Maksymalna różnica		Długość różnicy wektorów	
	Float32	Float64	Float32	Float64
2	0.000000e+00	1.110223e-16	0.000000e+00	1.110223e-16
3	5.960464e-08	1.110223e-16	5.960464e-08	1.110223e-16
4	5.960464e-08	1.110223e-16	5.960464e-08	1.570092e-16
5	5.960464e-08	2.220446e-16	5.960464e-08	2.482534e-16
6	5.960464e-08	2.220446e-16	5.960464e-08	2.482534e-16
7	1.192093e-07	2.220446e-16	1.332800e-07	2.719480e-16
8	1.192093e-07	2.220446e-16	1.788139e-07	2.719480e-16
9	1.192093e-07	2.220446e-16	1.788139e-07	2.719480e-16
10	1.192093e-07	2.220446e-16	1.884864e-07	2.937374e-16
11	1.192093e-07	2.220446e-16	1.884864e-07	2.937374e-16
12	1.192093e-07	2.220446e-16	1.884864e-07	3.682193e-16
13	1.192093e-07	4.440892e-16	1.976862e-07	5.661049e-16
14	1.192093e-07	4.440892e-16	1.884864e-07	5.661049e-16
15	1.192093e-07	4.440892e-16	1.884864e-07	5.661049e-16
16	1.192093e-07	2.220446e-16	2.230202e-07	3.140185e-16
17	1.192093e-07	2.220446e-16	2.230202e-07	3.140185e-16
18	1.192093e-07	2.220446e-16	2.308478e-07	3.140185e-16
19	1.192093e-07	2.220446e-16	2.308478e-07	3.140185e-16
20	1.192093e-07	2.220446e-16	2.308478e-07	4.002966e-16
30	1.192093e-07	2.220446e-16	3.209808e-07	4.154074e-16
50	1.192093e-07	2.220446e-16	4.298152e-07	7.928578e-16
75	1.192093e-07	2.220446e-16	5.196212e-07	9.222205e-16
100	1.192093e-07	2.220446e-16	5.685921e-07	1.011461e-15
150	2.384186e-07	3.330669e-16	7.633119e-07	1.318317e-15
200	2.384186e-07	3.330669e-16	8.658075e-07	1.514142e-15

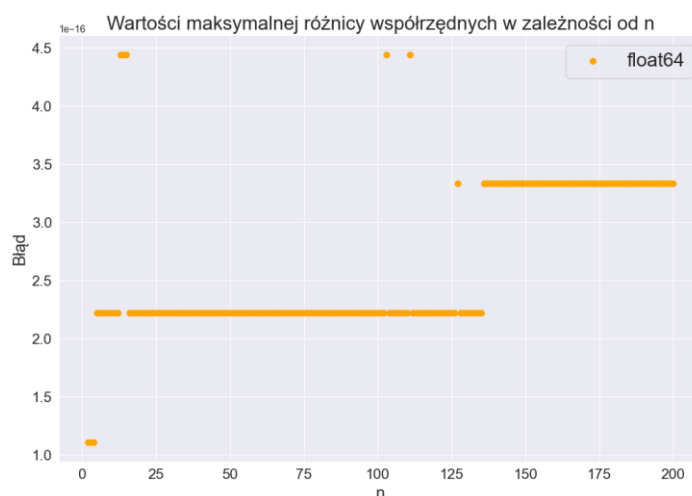
Tabela 4.3.1.1 Wartości obliczonych błędów dla zadania 3 metoda Gaussa

4.3.1.2 Wykresy

Wykresy wartości maksymalnej różnicy dla zadania 3 metoda Gaussa

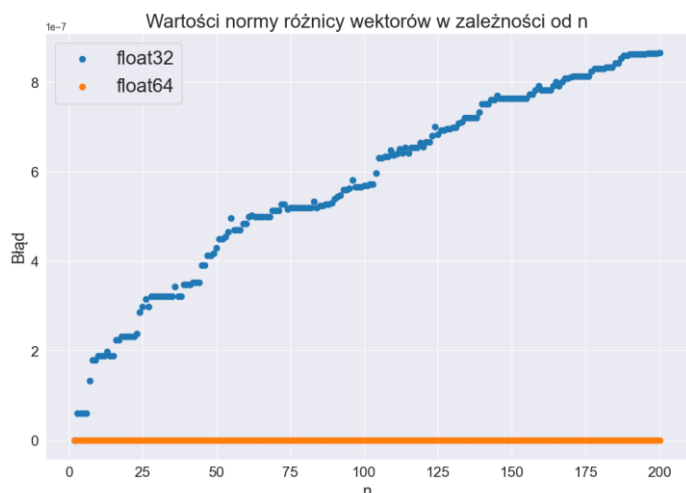


Wykres 4.3.1.2.1

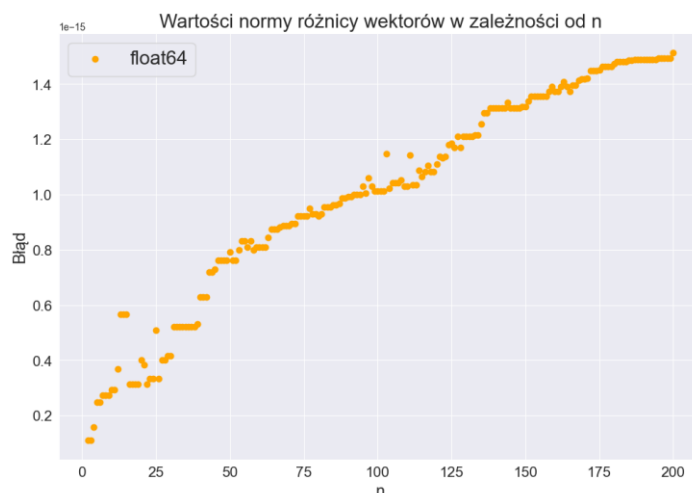


Wykres 4.3.1.2.2

Wykresy wartości normy różnicy wektorów dla zadania 3 metoda Gaussa



Wykres 4.1.1.2.3



Wykres 4.1.1.2.4

Jak możemy zobaczyć w tabeli 4.3.1.1 i na powyższych wykresach, w przypadku układu równań z zadania 3 otrzymywane błędy są znacznie mniejsze nawet od tych z zadania 2. Wykresy błędów maksymalnej różnicy współrzędnych ma charakterystyczną „schodkową” postać. Dla float64 błędy są o 10^8 , 10^9 mniejsze. W przypadku normy różnicy wektorów błąd na początku rośnie szybciej, a później co raz wolniej – kształtem przypomina wykres pierwiastka kwadratowego.

4.3.2. Metoda Thomasa

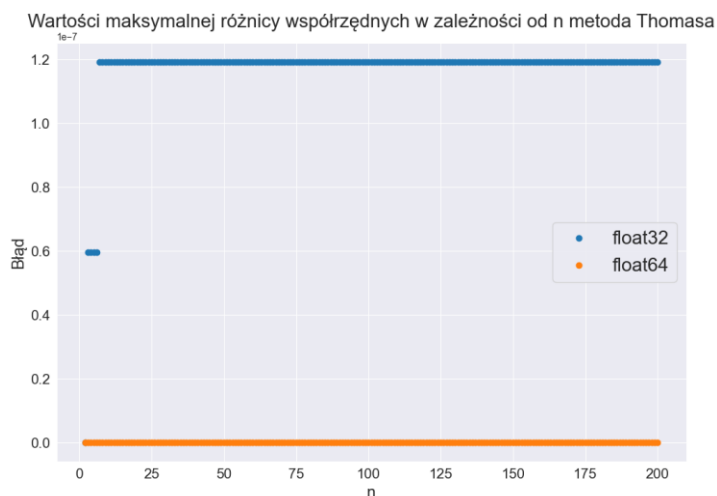
4.3.2.1 Tabela z wartościami błędów metoda Thomasa

n	Precyzja			
	Maksymalna różnica		Długość różnicy wektorów	
	Float32	Float64	Float32	Float64
2	0.000000e+00	1.110223e-16	0.000000e+00	1.110223e-16
3	5.960464e-08	1.110223e-16	5.960464e-08	1.110223e-16
4	5.960464e-08	1.110223e-16	5.960464e-08	1.570092e-16
5	5.960464e-08	2.220446e-16	5.960464e-08	2.482534e-16
6	5.960464e-08	2.220446e-16	5.960464e-08	2.482534e-16
7	1.192093e-07	2.220446e-16	1.332800e-07	2.719480e-16
8	1.192093e-07	2.220446e-16	1.788139e-07	2.719480e-16
9	1.192093e-07	2.220446e-16	1.788139e-07	2.719480e-16
10	1.192093e-07	2.220446e-16	1.884864e-07	2.937374e-16
11	1.192093e-07	2.220446e-16	1.884864e-07	2.937374e-16
12	1.192093e-07	2.220446e-16	1.884864e-07	3.682193e-16
13	1.192093e-07	4.440892e-16	2.308478e-07	5.661049e-16
14	1.192093e-07	2.220446e-16	2.308478e-07	4.299875e-16
15	1.192093e-07	2.220446e-16	2.308478e-07	4.299875e-16
16	1.192093e-07	2.220446e-16	2.598106e-07	4.299875e-16
17	1.192093e-07	2.220446e-16	2.598106e-07	4.299875e-16
18	1.192093e-07	2.220446e-16	2.665601e-07	4.299875e-16
19	1.192093e-07	2.220446e-16	2.665601e-07	4.299875e-16
20	1.192093e-07	2.220446e-16	2.665601e-07	4.965068e-16
30	1.192093e-07	2.220446e-16	3.625609e-07	6.377746e-16
50	1.192093e-07	2.220446e-16	4.878850e-07	8.671119e-16
75	1.192093e-07	2.220446e-16	5.930587e-07	1.005350e-15
100	1.192093e-07	2.220446e-16	6.364035e-07	1.093443e-15
150	1.192093e-07	4.440892e-16	7.952251e-07	1.522261e-15
200	1.192093e-07	4.440892e-16	8.940697e-07	1.672720e-15

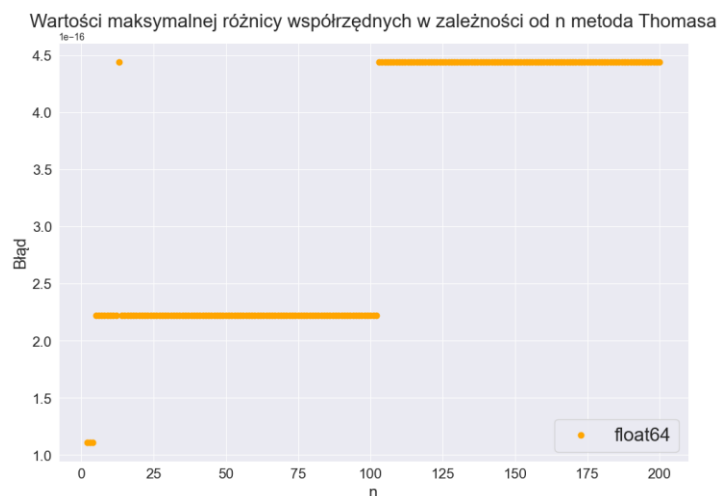
Tabela 4.3.2.1 Wartości obliczonych błędów dla zadania 3 metoda Thomasa

4.3.1.2 Wykresy

Wykresy wartości maksymalnej różnicy dla zadania 3 metoda Thomasa

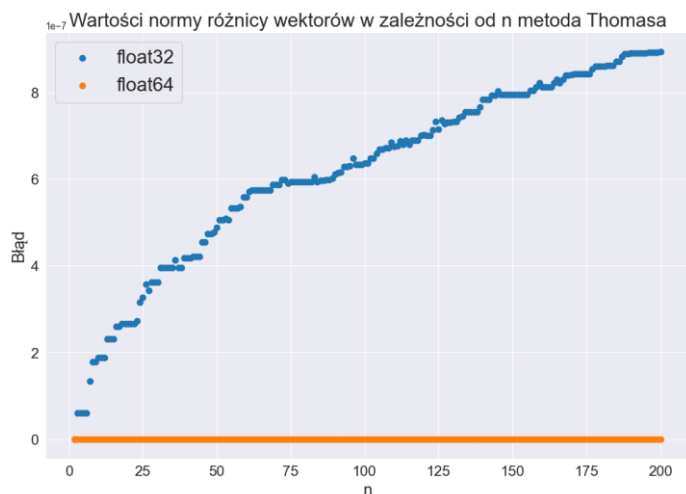


Wykres 4.3.1.2.1

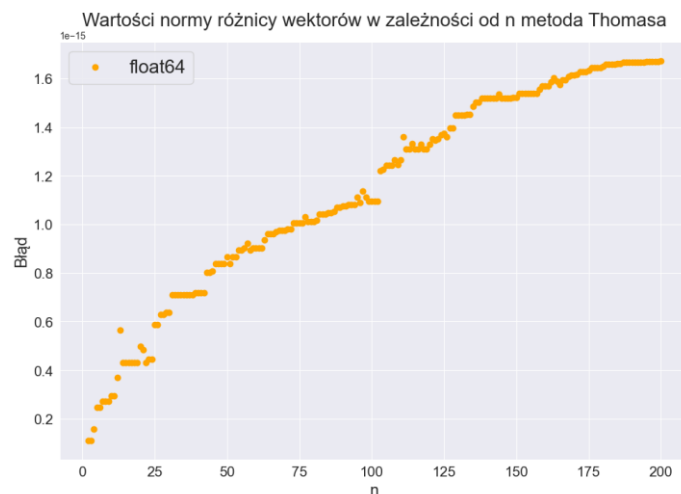


Wykres 4.3.1.2.2

Wykresy wartości normy różnicy wektorów dla zadania 3 metoda Thomasa



Wykres 4.3.1.2.3



Wykres 4.3.1.2.4

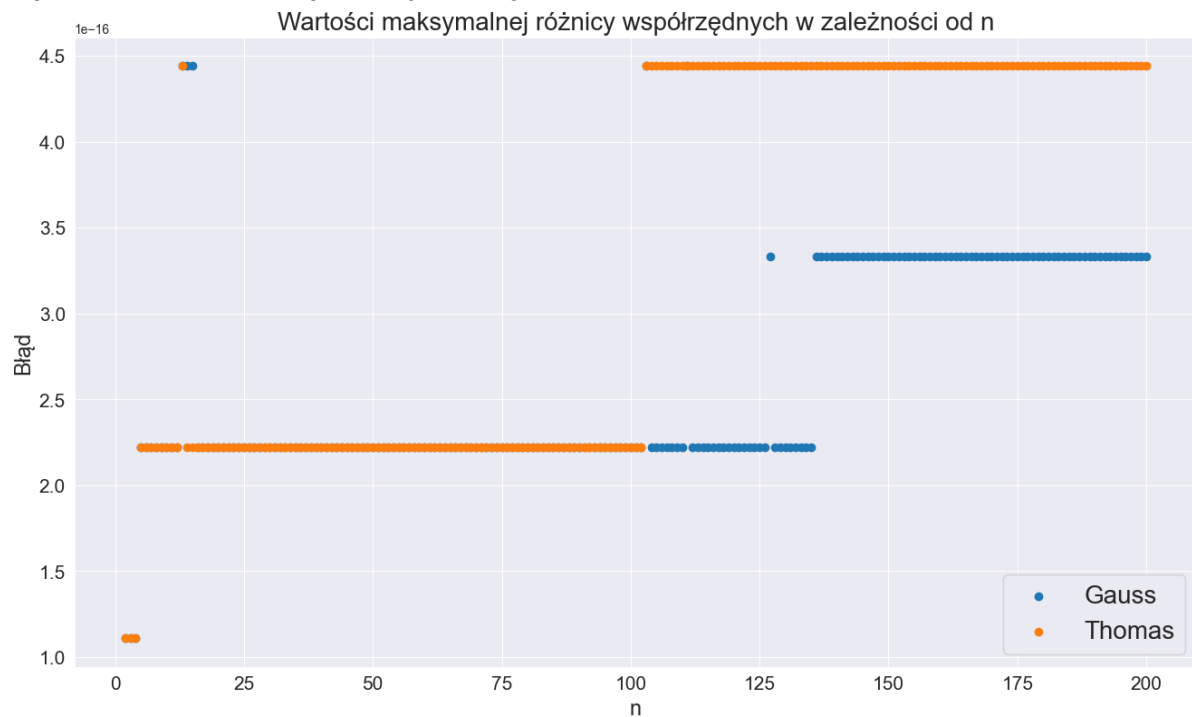
Dla metody Thomasa otrzymujemy podobne wyniki jak dla metody Gaussa. Jedynie na wykresach 4.3.1.2.1 oraz 4.3.1.2.2 możemy zobaczyć mniejszą ilość „schodków”. Otrzymywane błędy są praktycznie niezmiennie dla każdego n .

4.3.3. Porównanie metody Gaussa z metodą Thomasa

Aby porównać otrzymane wyniki dla obu metod, zostały one zestawione na jednym wykresie. Pokazane zostały tylko obliczenia dla float64 ponieważ w przypadku float32 sytuacja była identyczna.

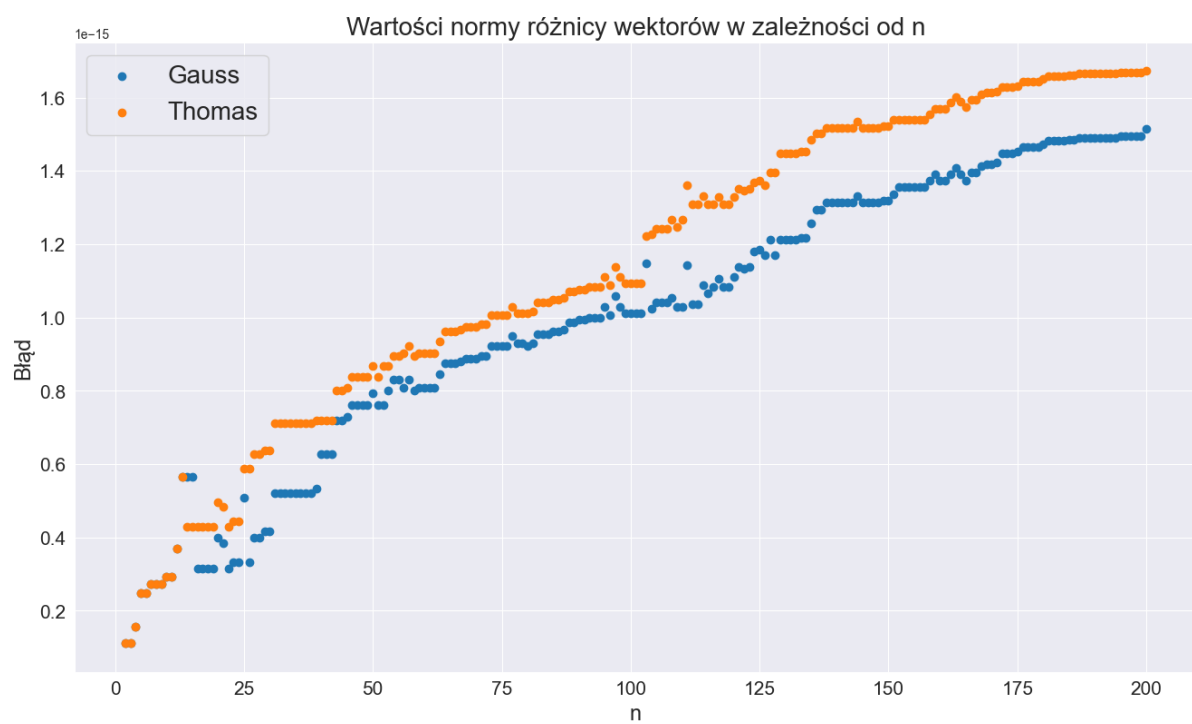
4.3.3.1. Porównanie dokładności obu metod.

Wykres wartości maksymalnej różnicy dla zadania 3



Wykres 4.3.3.1.1

Wykres wartości normy różnicy wektorów dla zadania 3



Wykres 4.3.3.1.2

Na powyższych wykresach widać, że dla obu metod otrzymywane wyniki są do siebie bardzo zbliżone, a ich wykresy mają podobny kształt. Możemy zauważyć, że dla metoda Gaussa jest nieco bardziej dokładna i otrzymywane błędy są trochę mniejsze.

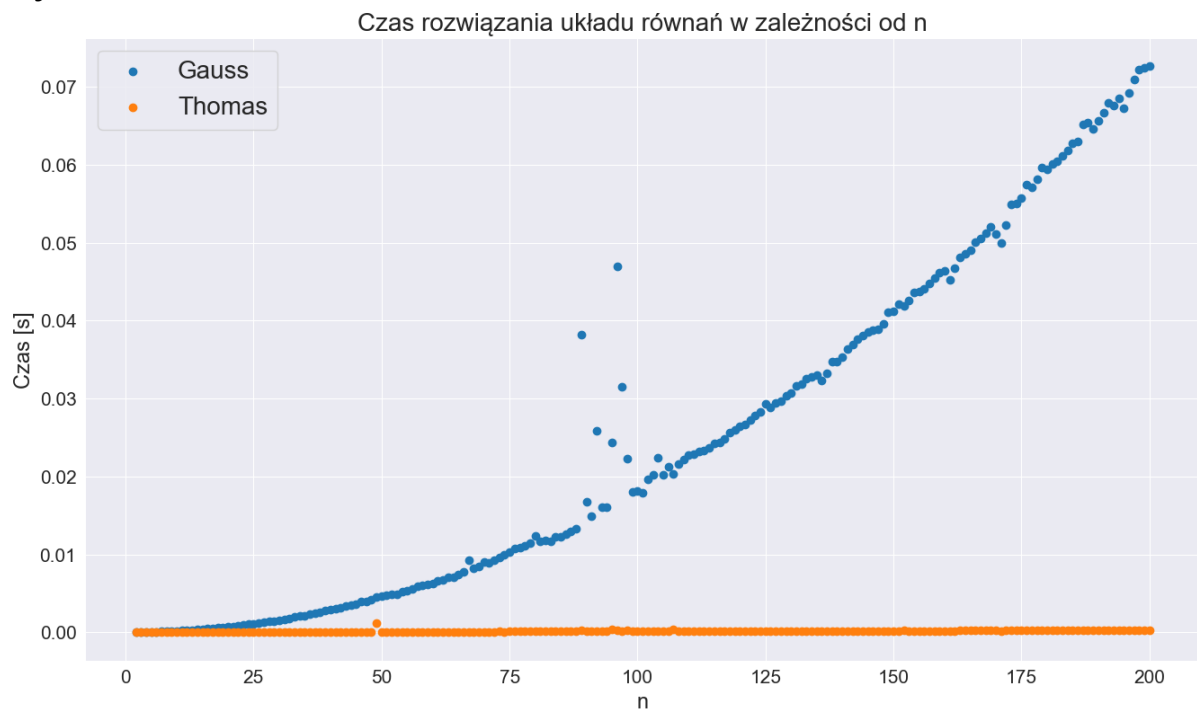
4.3.3.2. Porównanie czasów trwania obu metod.

Tabela z wartościami czasów metod Gaussa i Thomasa.

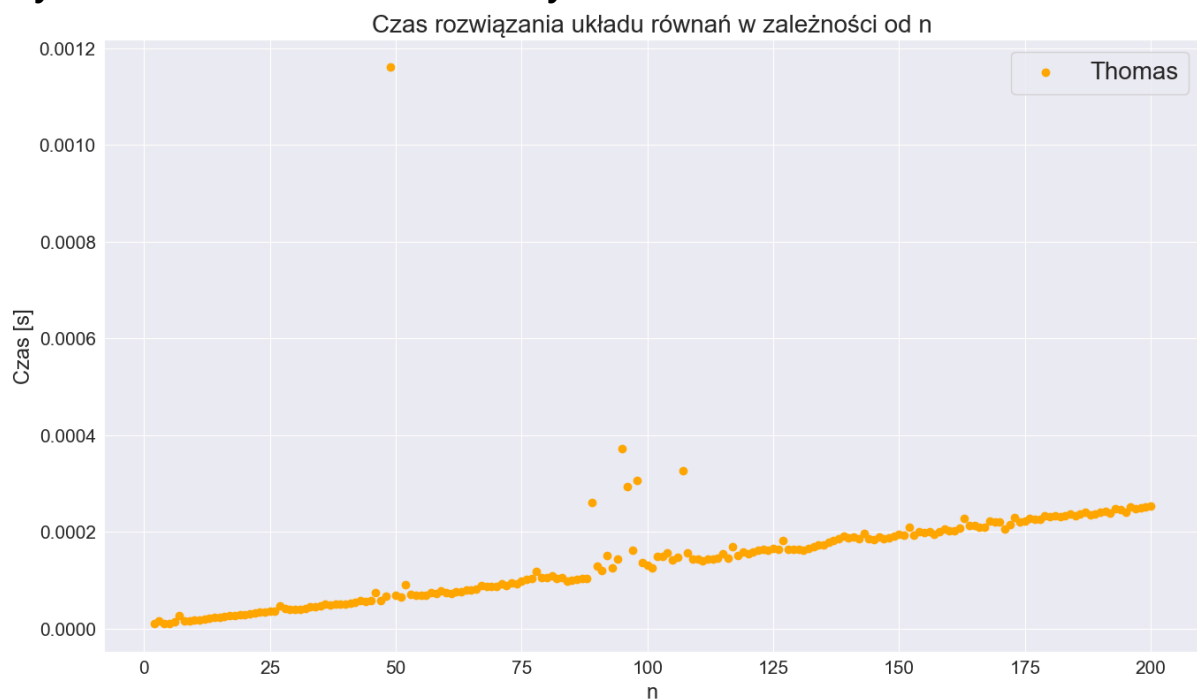
n	Gauss [s]	Thomas [s]
2	0.000053	0.000012
3	0.000034	0.000012
4	0.000033	0.000012
5	0.000072	0.000017
6	0.000087	0.000016
7	0.000178	0.000029
8	0.000133	0.000034
9	0.000159	0.000017
10	0.000186	0.000019
11	0.000221	0.000019
12	0.000276	0.000022
13	0.000334	0.000021
14	0.000357	0.000023
15	0.000408	0.000023
16	0.000502	0.000026
17	0.000540	0.000028
18	0.000588	0.000030
19	0.000750	0.000033
20	0.000748	0.000035
30	0.001593	0.000041
50	0.004640	0.000065
75	0.011142	0.000119
100	0.017951	0.000130
150	0.042300	0.000201
200	0.085645	0.000263

Tabela 4.2.2.1 Czasy trwania metod Gaussa i Thomasa

Wykres wartości czasów trwania metod Gaussa i Thomasa



Wykres wartości czasów trwania metody Thomasa



W przeciwieństwie do dokładności, czas trwania obu metod znacząco się różni. Metoda Thomasa jest znacznie szybsza od Gaussa, wyniki są mniejsze o 2 rzędy wielkości. Dodatkowo na wykresie 4.3.3.2.2 możemy zobaczyć, że czas rośnie liniowo w zależności od n dla metody

Thomasa, a w przypadku Gaussa, wykres 4.3.3.2.1. zależności czasu trwania od n ma kształt bardziej zbliżony do hiperboli.

Możemy również zauważyć, że dla niektórych n z przedziału $<80, 100>$ obie metody potrzebują nieznacznie więcej czasu niż dla innych n z swojego otoczenia.

4.3.3.3. Porównanie zajętości pamięci obu metod.

W przypadku metody Thomasa nie tworzymy macierzy o wymiarach $n \times n$. Ponieważ jest ona specjalnie stworzona dla macierzy trójdzielnych, co oznacza, że tylko trzy rzędy z każdego wiersza są niezerowe. Dlatego, zamiast przechowywać całą macierz, przechowywane są tylko te trzy rzędy, co daje nam złożoność pamięciową $O(3n) = O(n)$. Są one przechowywane w tablicy dwuwymiarowej $n \times 3$. W kolejnych kolumnach tej tablicy mamy: przekątną poniżej głównej przekątnej, główna przekątna i przekątna powyżej głównej przekątnej.

Algorytm Thomasa działa tylko na tych niezerowych wartościach, dlatego został dostosowany tak, aby móc używać stworzonej tablicy o wymiarach $n \times 3$.

Jeżeli chodzi o metodę Gaussa, to wymaga ona przechowywania całej macierzy o wymiarach $n \times n$, co prowadzi do złożoności pamięciowej $O(n^2)$.

5. Wnioski

- Duży wpływ na dokładność otrzymywanych wyników ma współczynnik uwarunkowania macierzy. W przypadku zadania 1, gdzie współczynnik uwarunkowania był dużo większy i szybciej rósł niż w zadaniu 2, otrzymywane wyniki były znacznie mniej dokładne.
- Dla float64 otrzymywane wyniki były zazwyczaj o kilka rzędów lepsze niż dla float32. Jednak w przypadku złego uwarunkowania jak w zadaniu 1 miało to miejsce tylko dla małych wartości n . Dla większych wyniki były podobne i trochę losowe – nie przedstawiały rzeczywistych wartości.
- Układ z zadania 3 był najlepiej uwarunkowany, błędy były bardzo małe i dodatkowo prawie nie zmieniały się wraz z rosnącą wartością n .
- Metoda Thomasa dla macierzy trójdzielnych pozwala nam osiągnąć niższą złożoność pamięciową niż metoda Gaussa, ponieważ przechowujemy tylko niezerowe elementy macierzy.
- Metoda Thomasa jest znacznie szybsza od metody Gaussa. W przypadku tej pierwszej czas trwania rośnie liniowo w zależności od n , a w przypadku drugiej – asymptotycznie.