Opracowanie: Maksymilian Sulima gr 3

**MOwNiT – Rozwiązywanie układów równań liniowych metodami bezpośrednimi**

1. Sprzęt

System operacyjny:

- Windows 10 19044.2604

Język:

- Python 3.10, numpy 1.24.3, matplotlib 3.7.1, jupyter

Procesor:

- AMD Ryzen 7 4700U

1. Treść zadania

Dla danego układu równań liniowych .

2.1) Dla macierzy zadanej wzorem:

Zadaniem jest obliczenie wektora b na podstawie dowolnej permutacji wektora x, który składa się z n elementów ze zbioru {1, -1}. Następnie należy rozwiązać układ równań liniowych przy użyciu metody eliminacji Gaussa, przyjmując wektor x jako nieznaną. W celu analizy wpływu błędów zaokrągleń na rozwiązanie, należy porównać obliczone wektory x z zadanym wektorem x, korzystając z określonej normy. Eksperymenty powinny być przeprowadzone dla różnych rozmiarów układu i zmiennych różnej precyzji, w celu zbadania, jak błędy zaokrągleń wpływają na rozwiązanie.

2.2) Porównaj wyniki uzyskane w 2.1 dla macierzy:

Uzasadnij różnice w wynikach, sprawdź uwarunkowanie układów.

3) Powtórz eksperyment dla macierzy:

Obraz zawierający tekst, Czcionka, zrzut ekranu, diagram

Opis wygenerowany automatycznie

Dla parametrów:

Następnie rozwiąż zadany układ korzystając z metody przeznczonej do układów z macierzą trójdiagonalną. Porównaj wyniki otrzymane za pomocą dwóch metod pod względem zużycia pamięci, czasu i szybkości obliczeń. Porównując czas weź pod uwagę tylko czas rozwiązania układu. Opisz sposób przechowywania i wykorzystania macierzy .

W celu obliczenia błędu obliczeń posłużono się następującą metryką:

,

gdzie:

– i-ta współrzędna zadanego wektora x,

– i-ta współrzędna wyznaczonego wektora x.

W eksperymentach użyto zmiennych z biblioteki float32 i float64 z biblioteki numpy. Zmienna float32 składa się z 8 bitów wykładnika i 23 bitów mantysy. Zmienna float64 składa się 11 bitów wykładnika i 52 bitów mantysy.

1. Wykonanie eksperymentów

**2.1)** Eksperyment został policzony dla

|  |  |  |
| --- | --- | --- |
| n | float32 | float64 |
| 3 | 0,00E+00 | 2,33E-15 |
| 4 | 4,13E-04 | 1,20E-13 |
| 5 | 2,00E-03 | 2,84E-12 |
| 6 | 1,71E-01 | 2,81E-10 |
| 7 | 2,21E+00 | 6,42E-09 |
| 8 | 3,65E+00 | 5,47E-08 |
| 9 | 2,31E+00 | 2,59E-07 |
| 10 | 2,44E+00 | 1,61E-04 |
| 11 | 5,21E+01 | 6,84E-03 |
| 12 | 3,99E+01 | 8,28E-02 |
| 13 | 1,02E+02 | 1,21E+00 |
| 14 | 2,01E+01 | 3,18E+00 |
| 15 | 4,49E+01 | 1,47E+01 |
| 16 | 5,31E+00 | 1,04E+00 |
| 17 | 1,46E+01 | 1,38E+01 |
| 18 | 2,09E+00 | 1,97E+01 |
| 19 | 3,71E+00 | 2,10E+01 |
| 20 | 1,80E+01 | 4,13E+02 |

Tabela 1. Przedstawia wartości błędów w zależności od wartości n i typu zmiennych użytych do obliczeń.

Obraz zawierający tekst, diagram, linia, numer

Opis wygenerowany automatycznie

Wykres 1. Przedstawia obliczone wartości błędów w zależności od wartości n. Wykres w skali logarytmicznej.

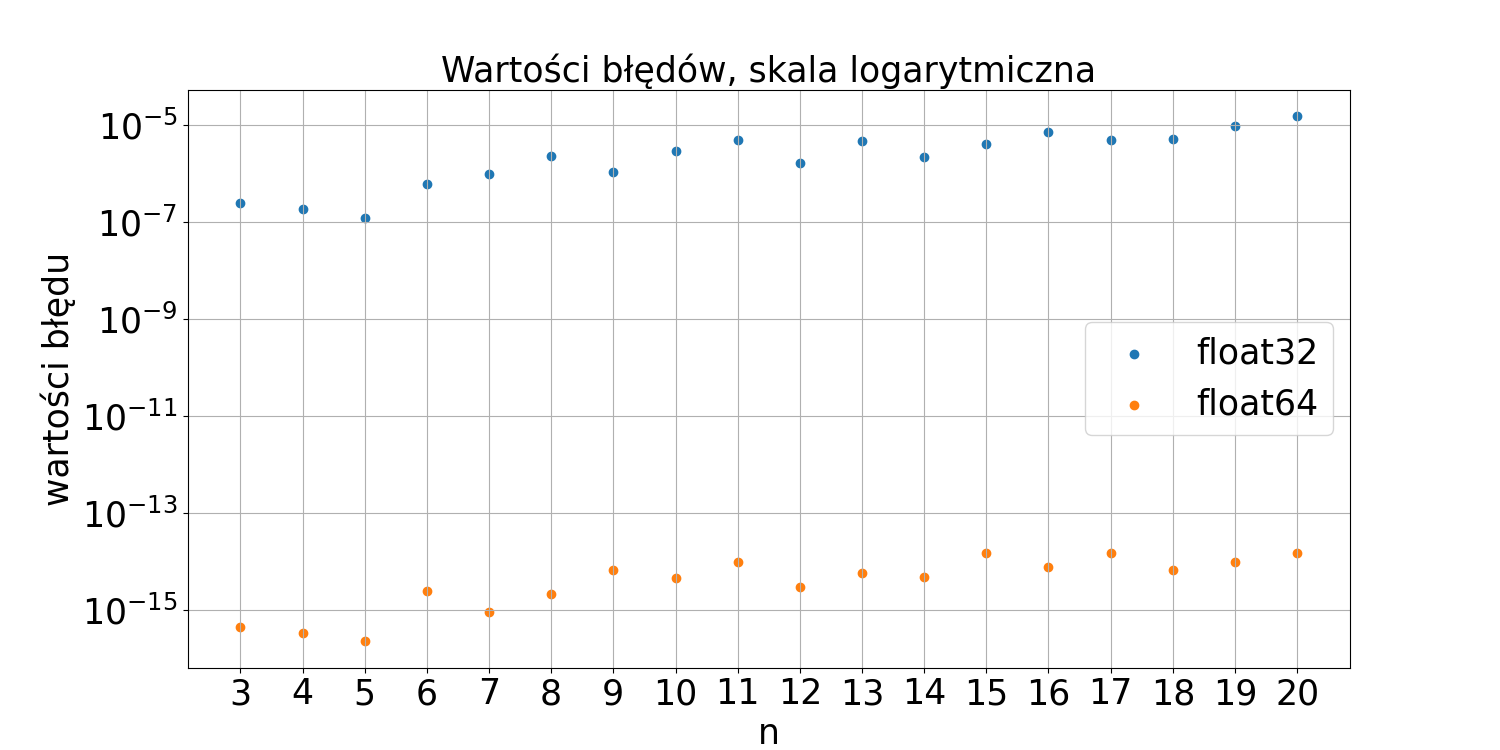
Z tabeli 1 i wykresu 1 możemy wczytać, że niedokładności rosną wraz ze wzrostem rozmiaru macierzy. Dla małych n wynik jest dość dokładnym, dla n = 3. Z kolei dla float32 wartość błędu wyniosła 0, dlatego też nie ma go na wykresie o skali logarytmicznej. Dla float32 dla n=7 pierwszy raz wartość błędu przekroczyła 1, natomiast dla float64 tą wartością jest n=13. Widać tutaj, że użycie zmiennych większej precyzji zmniejsza wielkość błędu obliczeniowego dla małych n. Jednak wraz ze wzrostem wielkości układu float64 zaczyna być coraz mniej dokładny. Od n = 17 dokładność dla zmiennej większej precyzji jest nawet mniejsza od zmiennej mniejszej precyzji.

Największa wartość błędu uzyskana dla float64 wyniosła 413 dla n = 20, natomiast dla float32 102 dla n = 13.

**2.2)** Eksperyment został policzony dla

|  |  |  |
| --- | --- | --- |
| n | float32 | float64 |
| 3 | 2,38E-07 | 4,44E-16 |
| 4 | 1,79E-07 | 3,33E-16 |
| 5 | 1,19E-07 | 2,22E-16 |
| 6 | 5,96E-07 | 2,44E-15 |
| 7 | 9,54E-07 | 8,88E-16 |
| 8 | 2,26E-06 | 2,11E-15 |
| 9 | 1,07E-06 | 6,66E-15 |
| 10 | 2,80E-06 | 4,55E-15 |
| 11 | 4,77E-06 | 9,66E-15 |
| 12 | 1,61E-06 | 2,89E-15 |
| 13 | 4,53E-06 | 5,55E-15 |
| 14 | 2,15E-06 | 4,66E-15 |
| 15 | 4,05E-06 | 1,49E-14 |
| 16 | 7,15E-06 | 7,55E-15 |
| 17 | 4,83E-06 | 1,49E-14 |
| 18 | 5,07E-06 | 6,66E-15 |
| 19 | 9,42E-06 | 9,55E-15 |
| 20 | 1,48E-05 | 1,49E-14 |
| 50 | 7,36E-05 | 9,95E-14 |
| 100 | 3,99E-04 | 6,34E-13 |
| 200 | 9,60E-04 | 2,02E-12 |

Tabela 2. Przedstawia wartości błędów w zależności od wartości n i typu zmiennych użytych do obliczeń.



Wykres 2. Przedstawia obliczone wartości błędów w zależności od wartości n. Wykres w skali logarytmicznej. (Błędy dla zostały pomięte na wykresie ze względu na czytelność)

W tabeli 2 i wykresie 2 można zauważyć, że zmiana wielkości zmierzonego błędu jest nieporównywalna w stosunku do układu z pkt 2.1.

Wszystkie zmierzone błędy dla float64 są rzędu od do w przeciwieństwie do poprzedniego układu równań, dla którego błąd już dla n=20 wynosi ponad 400.

Natomiast dla float32 największy z odnotowanych błędów wynosi co jest zupełnie nieporównywalne z największym błędem z poprzedniego eksperymentu, który wyniósł ponad 100.

Przyczynę takich różnic w wielkości błędów możemy się dopatrzeć w różnicach między zadanymi macierzami. Przyjrzyjmy się przykładowej macierzy z zadania 2.1:

Natomiast macierz odpowiednia macierz z zadania 2.2 wygląda następująco:

Widać tutaj, że macierz z 2.1 jest źle uwarunkowana, ponieważ wraz ze wzrostem jej rozmiaru jedynie zwiększy się amplituda jej elementów. Natomiast wartości z macierzy z 2.2 nie różnią się od siebie aż w tak drastyczny sposób. Odbija się to na wartościach współczynnika, przez który mnożymy odejmowane wiersze. W zadaniu 2.1 dla n = 15 zaobserwowano wartości od -30 do 15, wraz z wzrostem ilości obliczeń. Prowadzi to do zwiększenia różnicy między elementami ,co w arytmetyce zmiennoprzecinkowej często skutkuje utratą dokładności. W odpowiedniej macierzy z zadaniu 2.2 wszystkie wartości mieszczą się w przedziale .

Zbadajmy współczynniki uwarunkowania obu macierzy, aby potwierdzić naszą tezę. Współczynnik uwarunkowania macierzy wyraża się wzorem:

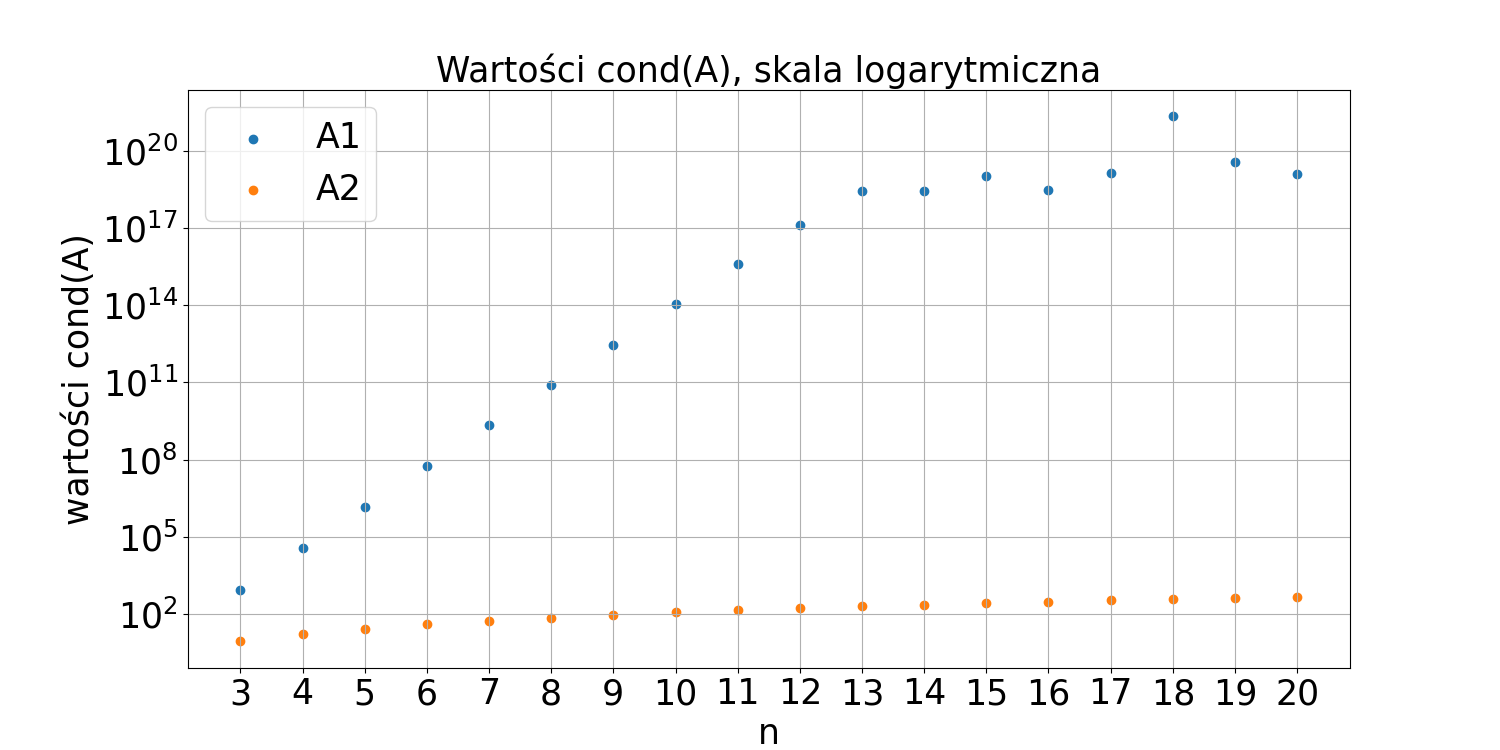
gdzie:

– oznacza normę z macierzy wyrażoną wzorem:

Większa wartość wskazuje na gorsze uwarunkowanie macierzy tzn. małe błędy w dokładności danych mogą skutkować dużymi błędami w wynikach. Współczynnik uwarunkowania został policzony za pomocą numpy.linalg.cond.

|  |  |  |
| --- | --- | --- |
| n | Zad 2.1 | Zad 2.2 |
| 3 | 8,64E+02 | 8,67E+00 |
| 4 | 3,79E+04 | 1,65E+01 |
| 5 | 1,44E+06 | 2,68E+01 |
| 6 | 5,63E+07 | 3,97E+01 |
| 7 | 2,23E+09 | 5,46E+01 |
| 8 | 8,16E+10 | 7,24E+01 |
| 9 | 2,84E+12 | 9,24E+01 |
| 10 | 1,09E+14 | 1,15E+02 |
| 11 | 3,95E+15 | 1,40E+02 |
| 12 | 1,36E+17 | 1,67E+02 |
| 13 | 2,73E+18 | 1,97E+02 |
| 14 | 2,75E+18 | 2,29E+02 |
| 15 | 1,04E+19 | 2,63E+02 |
| 16 | 2,90E+18 | 3,01E+02 |
| 17 | 1,35E+19 | 3,40E+02 |
| 18 | 2,21E+21 | 3,82E+02 |
| 19 | 3,57E+19 | 4,26E+02 |
| 20 | 1,22E+19 | 4,73E+02 |

Tabela 3. Przedstawia wartość czynnika uwarunkowania dla macierzy z zadań 2.1 i 2.2 w zależności od ich wielkości.



Wykres 3. Przedstawia wartości współczynnika uwarunkowania macierzy z zadań 2.1 i 2.2 w skali logarytmicznej. A1 oznacza macierz z zadania 2.1, a A2 macierz z zadania 2.2.

Obliczenia potwierdziły naszą tezę: macierz z zadnia 2.1 jest wyraźnie gorzej uwarunkowana. Wartości jej współczynnika zaczynają się na 800, a kończą się na , podczas gdy współczynnik dla macierzy z zadnia 2.2 ma najmniejszą wartość równą 8,(6), a największą 472. Dodatkowo widać tutaj ogólne pogorszenie się uwarunkowania wraz z rozmiarem macierzy, jednak tempo wzrostu może być istotnie różne.

**3)** Eksperyment został policzony dla:

,

metodą Thomasa i Gaussa. Ta pierwsza jest metodą przeznaczoną do rozwiązywania specyficznie układów z macierzą trójdiagonalną i charakteryzuje się złożonością obliczeniową w przeciwieństwie do metody Gaussa, która jest ogólna i ma złożoność obliczeniową . Czas w zadaniu został zmierzony za pomocą metody time() z modułu time w pythonie.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Gauss | | Thomas | |
| n | czas[s] float32 | czas[s] float64 | czas[s] float32 | czas[s] float64 |
| 3 | 0,000 | 0,000 | 0,000 | 0,000 |
| 13 | 0,001 | 0,001 | 0,000 | 0,000 |
| 23 | 0,006 | 0,003 | 0,000 | 0,000 |
| 33 | 0,009 | 0,008 | 0,000 | 0,000 |
| 43 | 0,018 | 0,018 | 0,000 | 0,001 |
| 53 | 0,031 | 0,029 | 0,000 | 0,000 |
| 63 | 0,051 | 0,050 | 0,000 | 0,000 |
| 73 | 0,076 | 0,077 | 0,001 | 0,000 |
| 83 | 0,129 | 0,111 | 0,001 | 0,000 |
| 93 | 0,211 | 0,157 | 0,000 | 0,000 |
| 103 | 0,272 | 0,313 | 0,000 | 0,000 |
| 113 | 0,348 | 0,290 | 0,001 | 0,000 |
| 123 | 0,385 | 0,421 | 0,001 | 0,001 |
| 133 | 0,480 | 0,485 | 0,001 | 0,000 |
| 143 | 0,579 | 0,593 | 0,000 | 0,001 |
| 153 | 0,716 | 0,717 | 0,000 | 0,000 |
| 163 | 0,869 | 0,872 | 0,001 | 0,001 |
| 173 | 1,039 | 1,069 | 0,001 | 0,001 |
| 183 | 1,226 | 1,213 | 0,000 | 0,001 |
| 193 | 1,431 | 1,457 | 0,000 | 0,000 |
| 203 | 1,644 | 1,703 | 0,000 | 0,001 |
| 213 | 1,917 | 1,968 | 0,000 | 0,001 |
| 223 | 2,257 | 2,246 | 0,001 | 0,000 |
| 233 | 2,485 | 2,474 | 0,001 | 0,001 |
| 243 | 2,781 | 2,898 | 0,000 | 0,001 |
| 253 | 3,180 | 3,184 | 0,000 | 0,001 |
| 263 | 3,625 | 3,614 | 0,001 | 0,001 |
| 273 | 4,045 | 4,014 | 0,000 | 0,001 |
| 283 | 4,401 | 4,501 | 0,001 | 0,001 |
| 293 | 4,925 | 4,956 | 0,001 | 0,001 |
| 303 | 5,407 | 5,469 | 0,001 | 0,001 |
| 313 | 5,999 | 5,985 | 0,001 | 0,002 |
| 323 | 6,518 | 6,668 | 0,000 | 0,000 |
| 333 | 7,201 | 7,171 | 0,001 | 0,001 |
| 343 | 8,025 | 8,070 | 0,001 | 0,001 |

Tabela 4. Przedstawia zależność czasu wykonania w sekundach od rozmiaru układu równań dla zmiennych różnych dokładności przy zastosowaniu metody Gaussa lub Thomasa.

Czasy przedstawione w tabeli 4 wyraźnie pokazują w praktyce różnice w złożoności obliczeniowej obu metod. Czas wykonania obliczeń używając metody Thomasa dla każdego zmierzonego oscyluje na granicy błędu pomiarowego. Natomiast czas wykonania metody Gaussa rośnie zauważalnie dochodząc do przekraczając nawet 8s dla n=343.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | float32 | | float64 | |
| n | Gauss | Thomas | Gauss | Thomas |
| 3 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 |
| 13 | 1,19E-07 | 5,96E-08 | 2,22E-16 | 1,11E-16 |
| 23 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |
| 33 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |
| 43 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 53 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 63 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 73 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 83 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 93 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 103 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 113 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 123 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |
| 133 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |
| 143 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 153 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 163 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 3,33E-16 |
| 173 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 183 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 193 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 2,22E-16 |
| 203 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |
| 213 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |
| 223 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 233 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 243 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 3,33E-16 |
| 253 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 263 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 273 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 283 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 3,33E-16 |
| 293 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 303 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 313 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 3,33E-16 |
| 323 | 1,19E-07 | 2,38E-07 | 2,22E-16 | 3,33E-16 |
| 333 | 1,19E-07 | 1,19E-07 | 2,22E-16 | 3,33E-16 |
| 343 | 1,19E-07 | 1,79E-07 | 2,22E-16 | 2,22E-16 |

Tabela 5. Przedstawia błędy obliczeń dla metod Gaussa i Thomasa w zależności od wielkości macierzy i użytego typu danych.

Dokładności obliczeń dla obu metod są identyczne, czynnik determinujący dokładność to typ precyzji użytej.

Dzięki dostosowaniu metody Thomasa do obliczeń na macierzy trójdiagonalnej użyto reprezentacji macierzy, jako listy 3-elementowych tupli. Dzięki temu złożoność pamięciowa spadła z kwadratowej (w metodzie Gaussa) do liniowej.

1. Wnioski

Przy rozwiazywaniu układu równań metodą Gaussa, zagadnieniem kluczowym dla dokładności obliczeń jest uwarunkowanie zadanej macierzy. Metoda ta jest w stanie dawać bardzo niedokładne wyniki (zadanie 2.1) jak i bardzo dokładne (zadanie 2.2). Dlatego przed użyciem tej metody kluczowe jest zbadanie uwarunkowania układu, który ma zostać rozwiązany. Użycie zmiennej większej precyzji pomaga poprawić dokładność tylko dla małych macierzy, a dla większych błędy mniejszej i większej precyzji są tych samych rzędów wielkości.

Rozwiązanie układu równań liniowych z wykorzystaniem metody przeznaczonej do układów z macierzą trójdiagonalną może być bardziej efektywne pod względem zużycia pamięci i czasu obliczeń w porównaniu do metody eliminacji Gaussa. W przypadku macierzy trójdiagonalnych, można wykorzystać specyficzną strukturę macierzy do zoptymalizowania procesu rozwiązywania układu równań.