Opracowanie: Maksymilian Sulima gr 3

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

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.

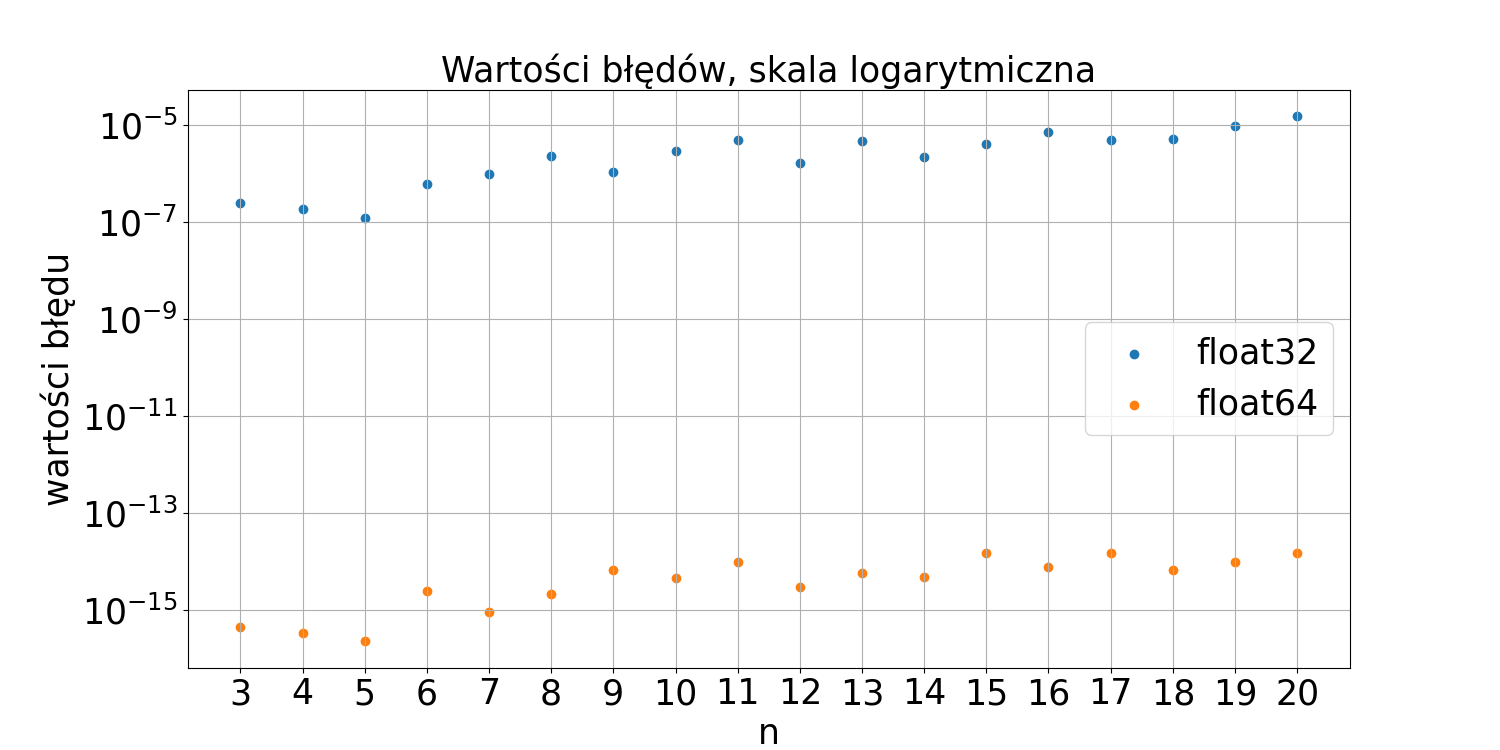
Z tabeli 1 i wykresu 1 możemy wczytać, że niedokładności rosną wraz ze wzrostem rozmiaru macierzy. Dla małych n wynik jest dość dokładnym dla n = 3 dla float32 wartość błędu wyniosła 0 dlatego nie ma go na widoku o skali logarytmicznej. Dla float32 dla n = 7 pierwszy raz wartość błędu przekroczyła 1 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 błędy 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 wielkościach błędów możemy się dopatrzeć w różnicach między zadanymi macierzami. Przyjrzymy się przykładowej macierzy z 2.1:

Natomiast macierz odpowiednia macierz 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 w aż tak drastyczny sposób. Odbija się to na wartościach współczynnika przez który mnóżmy 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:

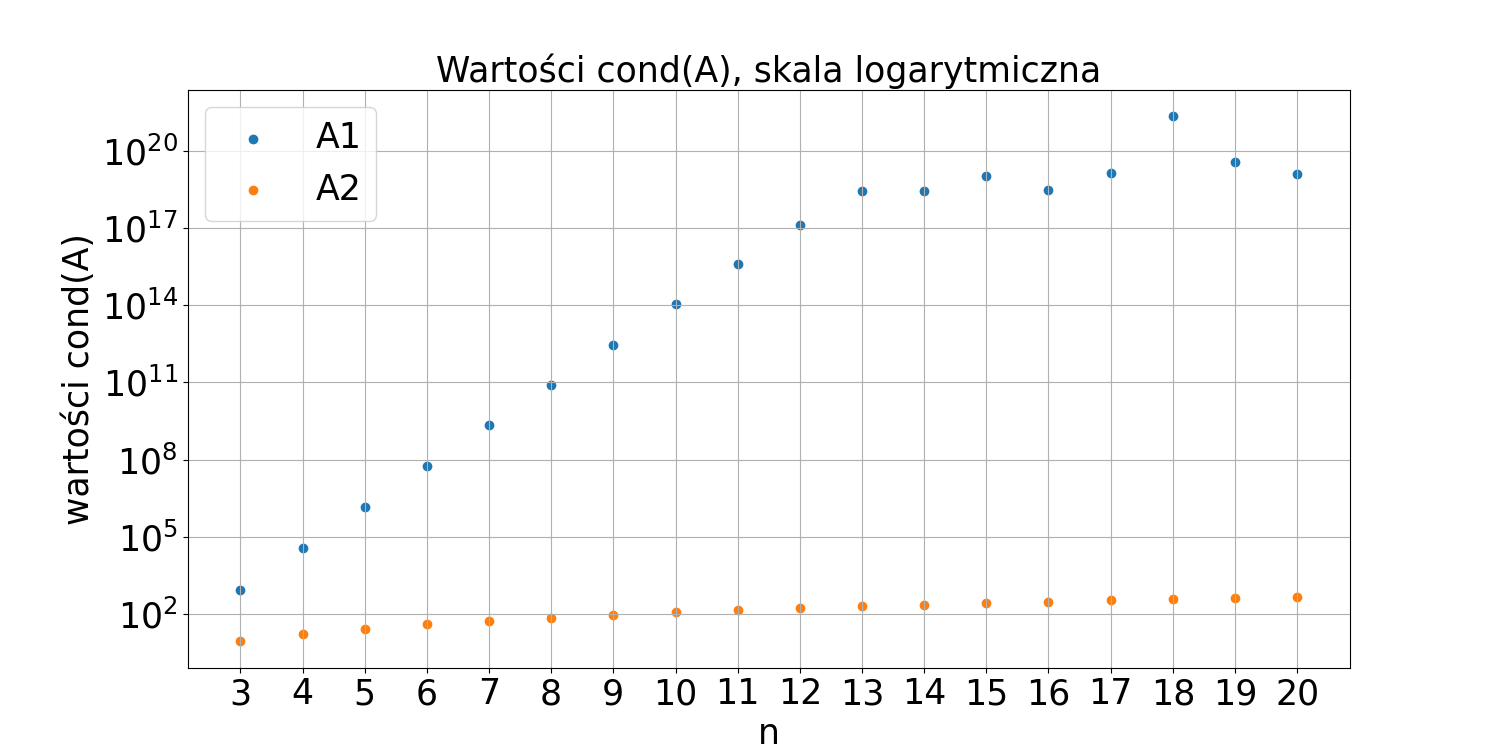
gdzie:

– 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 | 2,1 | 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.

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.