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

Obraz zawierający tekst, zrzut ekranu, linia, diagram

Opis wygenerowany automatycznie

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

Obraz zawierający zrzut ekranu, tekst, linia, diagram

Opis wygenerowany automatycznie

Wykres 2. Przedstawia obliczone wartości błędów w zależności od wartości n. Wykres w skali logarytmicznej.

Z wykresów 1 i 2 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 mniej dokładny. Największa wartość błędu uzyskana dla float64 wyniosła 413 dla n=20, natomiast dla float32 102 dla n = 13.

2.2) Eksperyment został policzony dla

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Wykres 3. Przedstawia obliczone wartości błędów w zależności od wartości n. Wykres w skali liniowej.

Obraz zawierający zrzut ekranu, tekst, linia, Wykres

Opis wygenerowany automatycznie

Wykres 4. Przedstawia obliczone wartości błędów w zależności od wartości n. Wykres w skali liniowej.

Na wykresach 3 i 4 można zauważyć, dużo słabszy związek między błędem, a rozmiarem macierzy dla float64. Wszystkie zmierzone błędy są rzędu od do , co w niemal każdym zastosowaniu jest wystarczające. Dla float32 widać, zwłaszcza na wykresie liniowym zależność między rozmiarem układu a dokładnością wyników jednak należy zwrócić uwagę na fakt, że największy z odnotowanych błędów wynosi co jest zupełnie nieporównywalne z największym błędem z eksperymentu 2.1, 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 utrata 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.

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznieWykres 5. Przedstawia wartości współczynnika uwarunkowania macierzy z zadań 2.1 i 2.2 w skali liniowej.

Obraz zawierający zrzut ekranu, tekst, linia, diagram

Opis wygenerowany automatycznie

Wykres 6. 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.Dodatko widać tutaj pogorszenie się uwarunkowania wraz z rozmiarem macierzy jednak tempo wzrostu może być istotnie różne.