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.