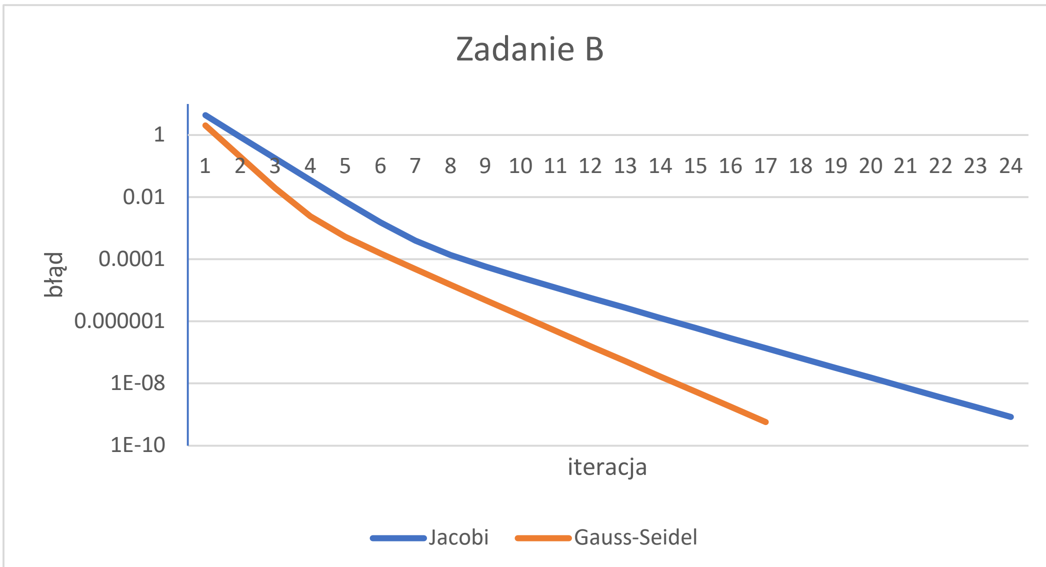
Sprawozdanie do projektu 2 – Układy równań liniowych

Adrian Ściepura 193350

- Wstęp teoretyczny
  - Celem projektu było rozwiązanie układów równań liniowych zapisanych w postaci macierzy za pomocą różnych sposobów. Zastosowane metody w projekcie:
  - Metoda Jacobiego
    - Metoda iteracyjna
    - Wzór użyty do implementacji
    - $x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^k$ - $\sum_{j=i+1}^{n} a_{ij}x_j^k)/a_{ii}$
  - Metoda Gaussa-Seidla
    - Metoda iteracyjna
    - Wzór użyty do implementacji
    - $x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)}$ - $\sum_{j=i+1}^{n} a_{ij}x_j^k)/a_{ii}$
  - Metoda faktoryzacji LU
    - Metoda bezpośrednia
    - Podział na macierz trójkątną górną i trójkątną dolną
  - Jakoś powyższych metod została wyznaczona za pomocą obliczenia normy wektora residuum
- Założenia projektowe
  - Język implementacji: C++ (standard 20)
  - Sposób tworzenia wykresów: importowanie wygenerowanych plików .csv do programu Excel
  - Struktury utworzone w projekcie są generyczne – pozwala to na bardzo łatwą podmianę typu znajdującego się w macierzach – typ ten jest weryfikowany na etapie kompilacji (ograniczenie do typów zmiennoprzecinkowych)
- Zadania
  - Zadanie A
    - Stworzona została macierz o rozmiarze 950, z wartościami na głównej przekątnej równymi 8. Dwie równoległe przekątne

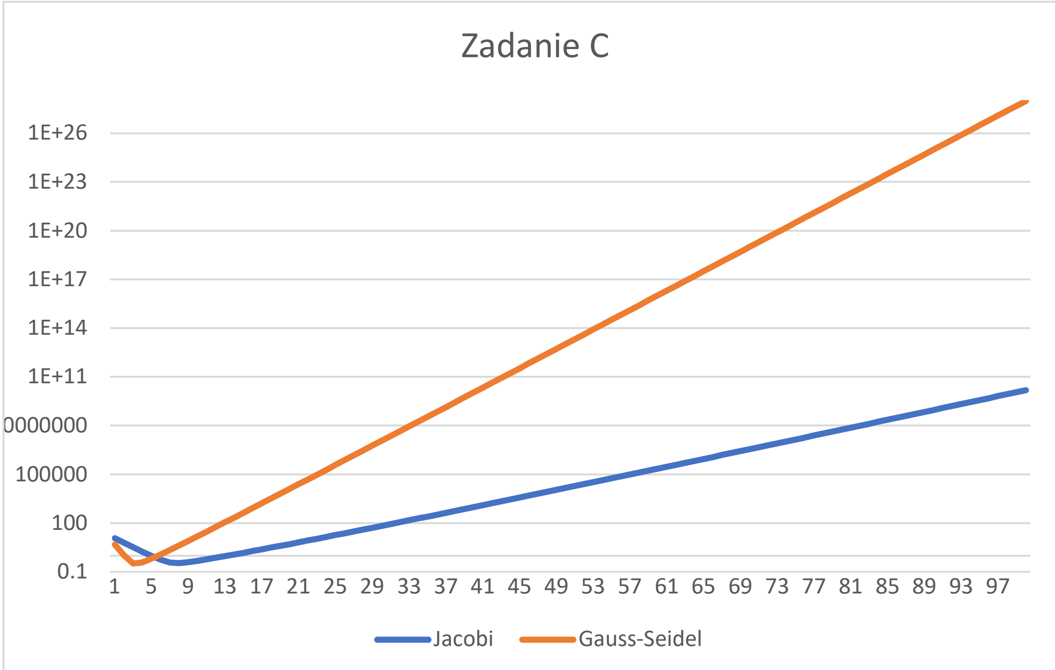
wypełnione zostały wartością -1. Taki typ macierzy nazywamy wstęgową lub pasmową. Wektor x został zainicjowany wartościami 0. Natomiast wektor b wartościami wyrażonymi wzorem sin(i * 4), gdzie i jest indexem kolejnej komórki.

- Zadanie B
  - Rozwiązano układ przedstawiony w zadaniu A za pomocą metody Jacobiego i Gaussa-Seidla. Metody wykonały się dość szybko z poniższymi rezultatami:



  - Po powyższym wykresie możemy stwierdzić, iż metoda Gaussa-Seidla szybciej odnalazła wartość dla określonego progu błędu
- Zadanie C
  - W tym zadaniu również zastosowano metody Jacobiego i Gaussa-Seidla, jednakże macierz A uległa delikatnej zmianie.

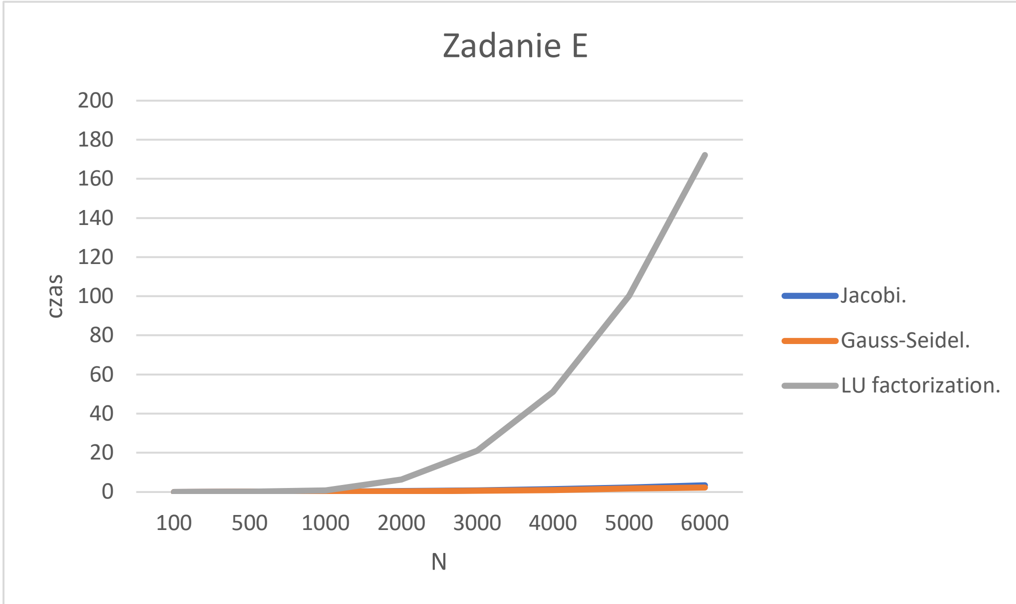
Rezultat działania przedstawia wykres:



- Na jego podstawie możemy ocenić iż metody iteracyjne się nie zbiegły i błąd rósł w bardzo szybkim tempie. Dodatkowo porównując obie metody można zauważyć że błąd rozwiązania Gaussa-Seidla rósł szybciej ponieważ korzystał on z zaktualizowanych już wartości wektora x.

- Zadanie D
  - Zaimplementowano metodę faktoryzacji LU, która w przeciwieństwie do poprzednich jest metodą bezpośrednią. Oznacza to że wynik jej działania będzie dokładny, jednakże wiąże się to z dłuższym czasem wykonania. Dla macierzy zdefiniowanej w zadaniu C norma residuum dla metody faktoryzacji LU wyniosła **2.76078e-13.** W porównaniu z metodami iteracyjnymi jest to ogromna różnica.
- Zadanie E
  - Wykonano przedstawione wcześniej metody iteracyjne oraz bezpośrednią na macierzach wielkości: `100`, `500`, `1000`, `2000`, `3000`, `4000`, `5000`, `6000`. Czas wykonania wszystkich metod

przedstawia poniższy wykres



- Możemy zaobserwować że wzrost rozmiaru macierzy miał bardzo delikatny wpływ na metody iteracyjne. Natomiast metoda faktoryzacji LU, która jest metodą bezpośrednią potrzebowała za każdym razem coraz więcej czasu. Ponadto czas wykonania rósł wielomianowo.

- Podsumowanie
  - Podsumowując metody iteracyjne są bardzo przydatne jeżeli chodzi o czas wykonania. Jeżeli chcemy otrzymać dokładniejszy wynik możemy obniżyć dopuszczalny próg. Jednakże wiązać się to będzie z dłuższym wykonaniem (nadal krótszym od metod bezpośrednich). Jednakże musimy przy tym wszystkim pamiętać o warunku zbieżności funkcji. Wykonanie metod iteracyjnych dla rozbieżnych funkcji jest bezcelowe, ponieważ otrzymany wynik będzie bezużyteczny i zmarnujemy jedynie czas i moc obliczeniową.
  - Rozwiązaniem problemu rozbieżnych funkcji są metody bezpośrednie. Jednakże są one bardzo wolne i wymagają dużej ilości pamięci do przechowywania dodatkowych macierzy (pamięć można zredukować stosując pewne optymalizacje, jednak nadal jest to więcej niż metody iteracyjne)