

Zadanie A - Wprowadzenie

Celem projektu jest implementacja i porównanie metod rozwiązywania układów równań liniowych. Analizowane będą dwie metody iteracyjne (metoda Jacobiego oraz metoda Gaussa-Seidla) oraz jedna metoda bezpośrednia (faktoryzacja LU). Analizowane układy równań będą następujące:

- I. Macierz systemowa A o wielkości $N = 963$ powstanie z liczb $a_1 = 8$, $a_2 = a_3 = -1$. Wektor pobudzenia b powstanie poprzez obliczenie n kolejnych elementów ciągu $\sin(4 \cdot n)$.
- II. Jedyną zmianą jest $a_1 = 3$. Poza tym parametry równania pozostają bez zmian.

Utworzenie macierzy A polega na ustawieniu liczb a_1 , a_2 , a_3 na odpowiednich miejscach zgodnie ze schematem poniżej:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & \dots & 0 \\ a_2 & a_1 & a_2 & a_3 & 0 & 0 & 0 & \dots & 0 \\ a_3 & a_2 & a_1 & a_2 & a_3 & 0 & 0 & \dots & 0 \\ 0 & a_3 & a_2 & a_1 & a_2 & a_3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & a_3 & a_2 & a_1 \end{bmatrix}$$

Oto przykładowa macierz 10x10:

```

8 -1 -1 0 0 0 0 0 0 0
-1 8 -1 -1 0 0 0 0 0 0
-1 -1 8 -1 -1 0 0 0 0 0
0 -1 -1 8 -1 -1 0 0 0 0
0 0 -1 -1 8 -1 -1 0 0 0
0 0 0 -1 -1 8 -1 -1 0 0
0 0 0 0 -1 -1 8 -1 -1 0
0 0 0 0 0 -1 -1 8 -1 -1
0 0 0 0 0 0 -1 -1 8 -1
0 0 0 0 0 0 0 -1 -1 8

```

Oraz przykładowe wartości wektora b :

```

-0.756802
0.989358
-0.536573
-0.287903
0.912945
-0.905578
0.270906
0.551427
-0.991779
0.745113

```

Zadanie B

W zadaniu tym implementujemy metody Jacobiego oraz Gaussa-Seidla. Metoda Jacobiego polega na iteracyjnym poprawianiu przybliżenia rozwiązania poprzez aktualizację wartości każdej zmiennej niezależnej na podstawie poprzedniej iteracji. Metoda Gaussa-Seidla działa niemal identycznie, ale podczas obliczania nowych wartości zmiennych wykorzystuje już obliczone wartości zmiennych niezależnych w tej samej iteracji.

$$\begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)})/a_{22} \\ x_3^{(k+1)} = (b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)})/a_{33} \end{cases}$$

Oto przykładowy schemat metody Jacobiego dla macierzy 3 na 3. Zauważyć możemy, że przy obliczaniu każdej kolejnej zmiennej możemy używać poprzednio już obliczonych zmiennych. Dla x_2 możemy wykorzystać poprzednio obliczone x_1 , natomiast dla x_3 możemy wykorzystać zarówno x_2 jak i x_1 . Takie podejście to właśnie metoda Gaussa-Seidla.

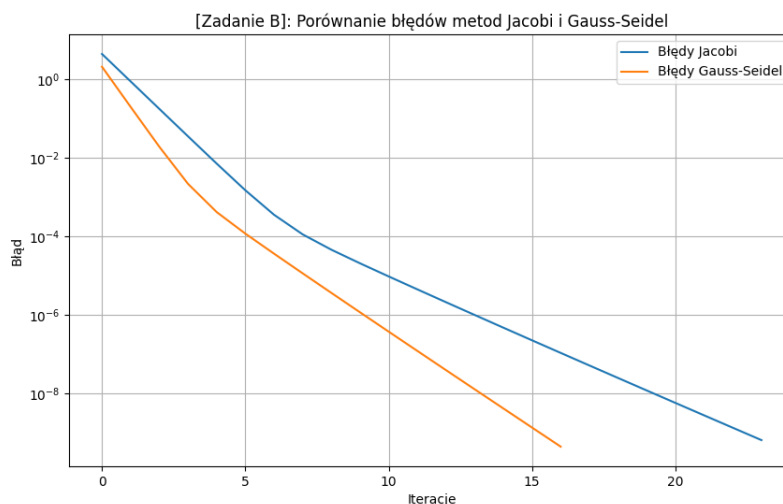
```
Jacoby's method:  
Time: 0.06 seconds  
Iterations: 23  
Error: 6.53e-10
```

```
Gauss-Seidel method:  
Time: 0.04 seconds  
Iterations: 16  
Error: 4.45e-10
```

W tym zadaniu wykorzystamy implementację wyżej wymienionych metod iteracyjnych w celu rozwiązania układu równań numer I. Wyniki są przedstawione poniżej:

Zauważyć możemy, że obydwie metody wykonały się w bardzo krótkim czasie (poniżej 0.1s). Metoda Jacobiego w celu otrzymania wyniku o błędzie residualnym mniejszym od 10^{-9} potrzebowała 23 iteracji. Dla osiągnięcia tak samo dokładnego rozwiązania metoda Gaussa-Seidla potrzebowała natomiast jedynie 16 iteracji.

Na wykresie poniżej zobaczyć możemy, że błąd residualny dla obu metod bardzo szybko maleje. Ponadto widać, że błąd dla metody Gaussa-Seidla malał o wiele szybciej od metody Jacobiego:



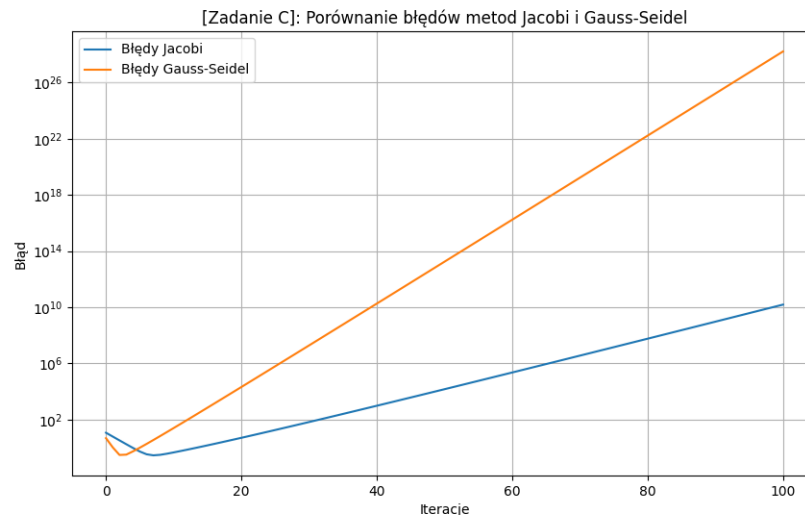
WNIOSKI: Obydwie metody wykonują się bardzo szybko. Zauważyć możemy, że metoda Gaussa-Seidla była szybsza oraz wymagała mniej iteracji w celu osiągnięcia satysfakcjonującego nas wyniku. Obydwie metody bardzo efektywnie zbiegają do rozwiązania dokładnego.

Zadanie C

Teraz naszym zadaniem jest rozwiązać układ równań numer II za pomocą poprzednio już zaimplementowanych metod iteracyjnych: Gaussa-Seidla oraz Jacobiego. Wyniki możemy zobaczyć po prawej stronie. Widać na nich, że pomimo wykorzystania swojego limitu iteracji (100) obie metody nie osiągnęły wystarczająco dobrego rozwiązania. Stało się coś odwrotnego, metody te znalazły rozwiązanie dalekie od dokładnego.

```
Jacoby's method:  
Time: 0.24 seconds  
Iterations: 100  
Error: 1.56e+10  
  
Gauss-Seidel method:  
Time: 0.23 seconds  
Iterations: 100  
Error: 1.67e+28
```

Na wykresie poniżej łatwo zauważyć, że nasze metody zamiast zbliżać się do naszego rozwiązania, oddalały się od niego. Błąd residualny ciągle rósł, więc gdybyśmy nie ustawili górnego limitu iteracji, nasze metody prawdopodobnie nigdy nie osiągnęłyby satysfakcjonującej dokładności, a błąd nadal by wzrastał w nieskończoność:



WNIOSKI: Metody iteracyjne nie zawsze zapewniają nam osiągnięcie wystarczająco przybliżonego rozwiązania. Dla niektórych układów równań metody te mogą okazać się rozbieżne, dlatego tak ważne jest wprowadzenie limitu ilości iteracji i porzucanie tej metody w przypadku, gdy metody te nie będą zbliżać się do rozwiązań optymalnych.

ZADANIE D

W tym zadaniu wykorzystamy metodę bezpośrednią w celu sprawdzenia, czy ona również okaże się rozbieżna dla przypadku z zadania C. Implementowana przez nas metoda to faktoryzacja LU. Polega ona na znalezieniu dwóch macierzy trójkątnych: górnej U oraz dolnej L, tak aby wynik mnożenia macierzy L z U był równy naszej macierzy systemowej A. Teraz do rozwiązania układu równań pozostało nam wykonać kilka kroków, które pozwolą nam znaleźć wektor rozwiązań:

- Tworzymy wektor pomocniczy: $y = Ux$
- Rozwiązujemy układ równań: $Ly = b$ za pomocą podstawiania wprzód
- Rozwiązujemy układ równań: $Ux = y$ za pomocą podstawiania wstecz

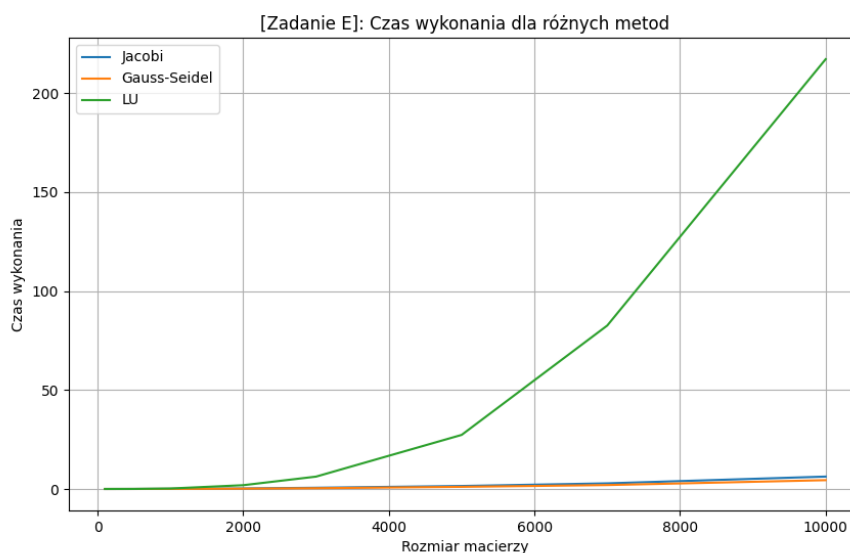
Otrzymany wektor x jest poszukiwanym przez nas wektorem rozwiązań. Jak widać na wynikach po prawej, metoda ta osiągnęła wynik o marginalnym błędzie residualnym rzędu 10^{-13} .

```
LU factorization:  
Time: 0.28 seconds  
Error: 2.19e-13
```

WNIOSKI: Faktoryzacja LU znalazła niemal dokładne rozwiązanie badanego w zadaniu C układu równań. Oznacza to, że metody bezpośrednie pozwalają nam na znalezienie rozwiązania dokładnego w każdym przypadku, a marginalne błędy wynikają głównie z błędów zaokrągleń komputera.

Zadanie E

W tym zadaniu porównamy czas wyznaczania rozwiązań dla omówionych wcześniej 3 wybranych metod. Wykorzystamy w tym celu macierz z zadania B, natomiast będziemy zwiększali jego wielkość. Sprawdzimy czasy poszukiwania rozwiązań dla wielkości macierzy $N = \{100, 500, 1000, 2000, 3000, 5000, 7000, 10.000\}$. Wyniki są przedstawione na wykresie poniżej:



Na wykresie możemy zauważyć, że czas obliczania rozwiązań dla metod iteracyjnych znacząco odbiegają od czasu faktoryzacji LU. Dla większych rozmiarów macierzy faktoryzacja LU może szukać rozwiązania nawet 40-krotnie dłużej od metod iteracyjnych. Ponadto zauważyć możemy nieznacznie lepsze czasy dla metod Gaussa-Seidla w porównaniu z metodą Jacobiego. W tabeli poniżej możemy zobaczyć normę błędu oraz czas wykonania dla każdej z metod dla każdej kolejnej wielkości macierzy.

N	Jacobi Time	Jacobi Error	Gauss-Seidel Time	Gauss-Seidel Error	LU Time	LU Error
100	5.12e-04	8.06e-10	3.62e-04	5.85e-10	2.63e-04	7.04e-16
500	1.60e-02	5.66e-10	1.05e-02	3.98e-10	2.02e-02	1.81e-15
1000	6.28e-02	7.69e-10	4.39e-02	5.48e-10	2.40e-01	2.55e-15
2000	2.57e-01	7.46e-10	1.74e-01	5.45e-10	1.91e+00	3.60e-15
3000	6.16e-01	8.34e-10	3.86e-01	7.57e-10	6.24e+00	4.45e-15
5000	1.51e+00	7.33e-10	1.08e+00	5.16e-10	2.73e+01	5.81e-15
7000	2.84e+00	9.77e-10	1.98e+00	9.64e-10	8.26e+01	6.86e-15
10000	6.25e+00	7.99e-10	4.45e+00	5.83e-10	2.17e+02	8.22e-15

Zauważyć możemy, że pomimo o wiele lepszych czasów dla metod iteracyjnych posiadają one całkiem gorszy błąd residualny w porównaniu z metodą bezpośrednią, natomiast wynika to z góry założonej granicy błędu, po której przekroczenie metody te kończą się. Zapewne moglibyśmy ustawić surowsze wymagania co do dokładności rozwiązania nie tracąc przy tym jakoś mocno na czasie wykonania.

WNIOSKI: Metody iteracyjne znajdują przybliżone rozwiązania znacznie szybciej niż metody bezpośrednie, natomiast precyzja rozwiązania może być niższa od metod bezpośrednich. Oznacza to, że mogą być bardzo przydatne przy rozwiązywaniu układów równań w przypadku bardzo dużych macierzy. Są one szczególnie przydatne, gdy nie zależy nam na bardzo dokładnych rozwiązaniach.

Zadanie F – Podsumowanie

Metody iteracyjne rozwiązywania układów równań liniowych okazują się być nieporównywalnie szybsze dla dużej ilości równań i niewiadomych. Są one ponadto bardzo elastyczne – jeśli zależy nam na dokładnych rozwiązaniach, możemy ustawić surowszą normę błędu, natomiast jeśli zależy nam na jak najszybszym czasie wykonania możemy pogodzić się z uzyskiwaniem mniej dokładnych wyników, jednocześnie znacząco poprawiając czas wykonywania obliczeń. Niestety metody iteracyjne nie sprawdzą się w każdym przypadku – są macierze dla których metody iteracyjne są rozbieżne i zamiast wraz z kolejnymi iteracjami zbliżać się do rozwiązania, będą się od niego oddalać – i to w bardzo szybkim tempie. Dlatego nie należy bagatelizować potęgi metod bezpośrednich, które mimo nieraz o wiele dłuższego czasu wykonania, gwarantują nam otrzymanie niemal dokładnego wyniku (ze względu na błędy zaokrągleń) w każdym przypadku. Bardzo dobrym połączeniem w takim razie byłoby wykonywanie początkowo metody iteracyjnej, i ewentualne zastosowanie metody bezpośredniej, gdy metoda iteracyjna zawiedzie. Ponadto faktoryzacja LU zadziała bardzo dobrze dla układów równań, które wykonujemy wielokrotnie dla tej samej macierzy A , lecz różnych macierzy b , ponieważ głównym obciążeniem obliczeniowym tej metody jest właśnie znajdowanie macierzy LU (złożoność tego to n^3).