

Metody Numeryczne

Project 2. Układy równań liniowych

Pavel Harelik 196766, grupa 2

Wstęp

Celem projektu jest implementacja i analiza dwóch metod iteracyjnych (Jacobiego i Gaussa-Seidla) oraz jednej metody bezpośredniej (faktoryzacja LU) rozwiązywania układów równań liniowych. Testy poszczególnych metod będą przeprowadzane na układach równań które zostaną obliczone zgodnie z pewnymi zasadami, ale są zbliżone do tych spotykanych w praktyce. Często dla takich równań stosuje się macierze rzadkie, ale dla uproszczenia użyjemy tzw. formatu pełnego.

Do wykonania obliczeń i wizualizacji użyto programu Matlab.

Zadanie A

Stwórzmy macierze A i b z równania postaci $Ax = b$, aby znaleźć dla nich rozwiązanie x.

Dla mojego indeksu otrzymujemy następujące parametry :

a1 = 12

a2 = a3 = -1

N = 966

f = 7

Korzystając z nich, otrzymujemy macierz A pierwsze 15 elementów której wygląda następująco:

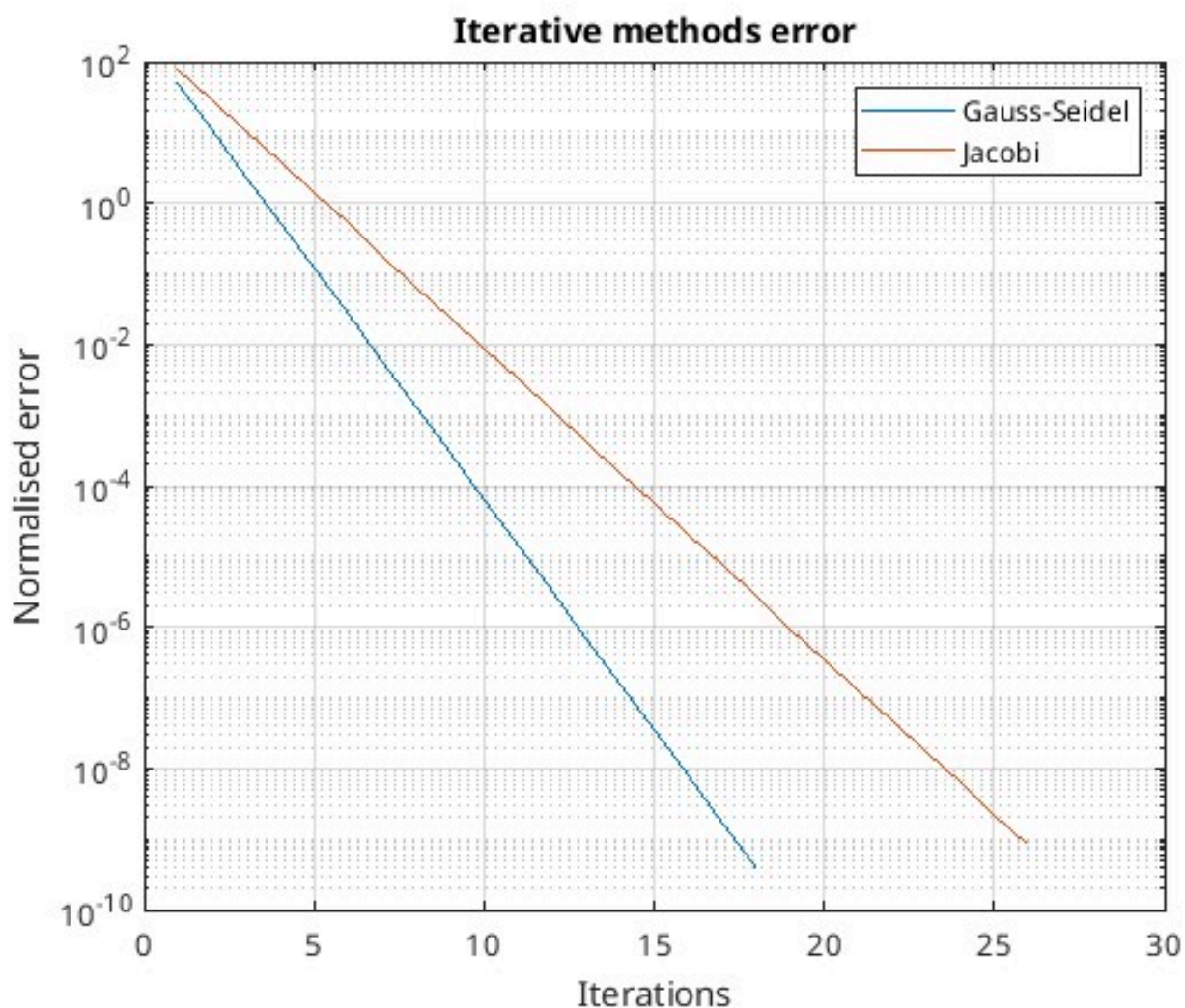
12	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0
-1	12	-1	-1	0	0	0	0	0	0	0	0	0	0	0
-1	-1	12	-1	-1	0	0	0	0	0	0	0	0	0	0
0	-1	-1	12	-1	-1	0	0	0	0	0	0	0	0	0
0	0	-1	-1	12	-1	-1	0	0	0	0	0	0	0	0
0	0	0	-1	-1	12	-1	-1	0	0	0	0	0	0	0
0	0	0	0	-1	-1	12	-1	-1	0	0	0	0	0	0
0	0	0	0	0	-1	-1	12	-1	-1	0	0	0	0	0
0	0	0	0	0	0	-1	-1	12	-1	-1	0	0	0	0
0	0	0	0	0	0	0	-1	-1	12	-1	-1	0	0	0
0	0	0	0	0	0	0	0	-1	-1	12	-1	-1	0	0
0	0	0	0	0	0	0	0	0	-1	-1	12	-1	-1	0
0	0	0	0	0	0	0	0	0	0	-1	-1	12	-1	-1
0	0	0	0	0	0	0	0	0	0	0	-1	-1	12	-1
0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	12

I wektor b pierwsze 11 elementów którego wygląda następująco

0.8415	0.9093	0.1411	-0.7568	-0.9589	-0.2794	0.6570	0.9894	0.4121	-0.5440	-1.0000
--------	--------	--------	---------	---------	---------	--------	--------	--------	---------	---------

Zadanie B

Do rozwiązania układu równań zaimplementowano dwie iteracyjne metody rozwiązywania równań. Jako kryterium uznania rozwiązania za skończone przyjęto przecięcie dokładności 10^{-9} . Rozwiązując układ równań opisany w punkcie A otrzymano następujące wyniki:

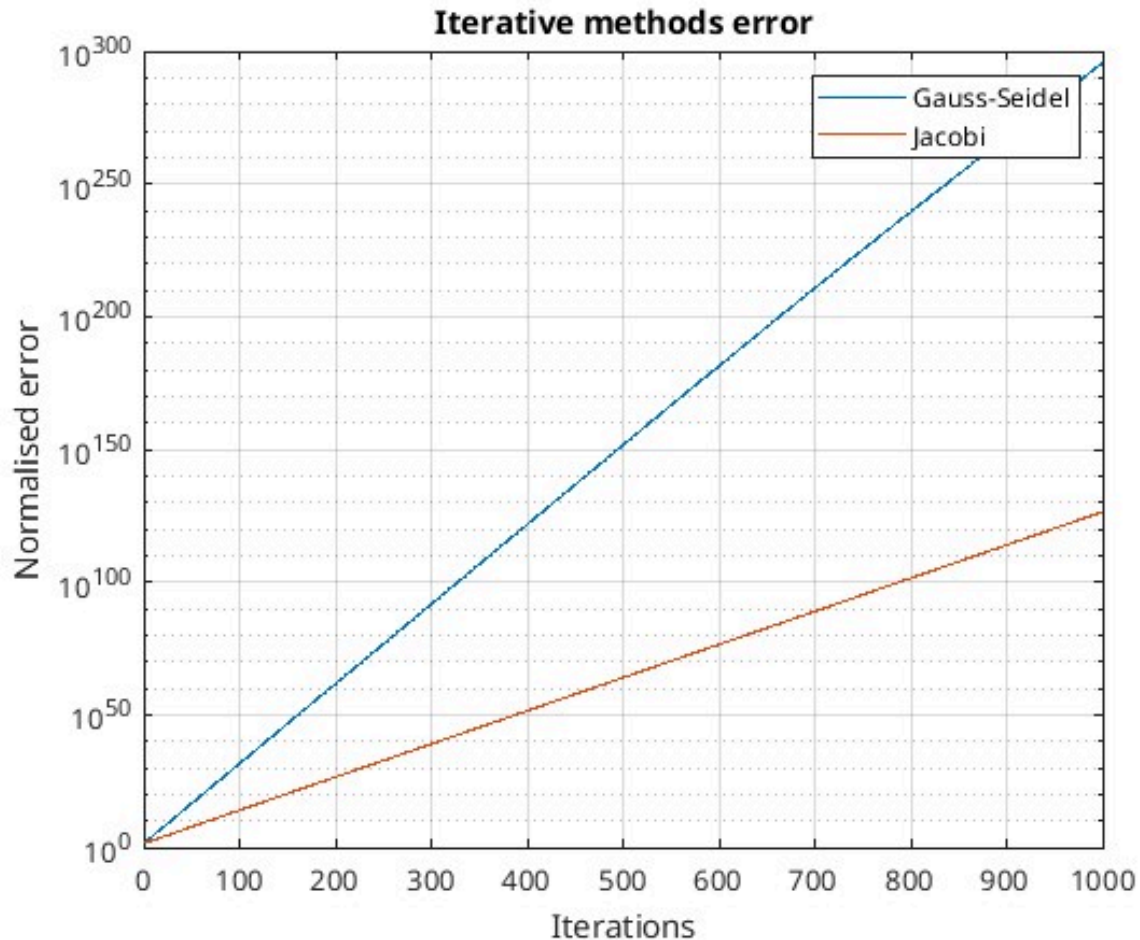


	Metody	
Parametr	Jacobiego	Gaussa–Seidla
Liczba iteracji	26	18
Czas trwania[s]	0.3203	0.2250

Można zauważyć, że błąd rozwiązania metody Gaussa-Seidla zmniejsza się szybciej niż błąd metody Jacobiego. Z tego powodu metoda ta wymaga mniejszej liczby iteracji i krótszego czasu obliczeń. Z tego możemy wywnioskować, że lepiej jest używać metody Jacobiego do rozwiązywania równań tego rodzaju.

Zadanie C

Do obliczeń w tym zadaniu bierzemy macierze jak w zadaniu B, ale zmieniamy wartości głównej przekątnej w macierzy A z 12 na 3. Uruchommy algorytmy iteracyjne dla otrzymanych macierzy. Otrzymamy następujące wyniki:



Wykres pokazuje, że znormalizowany błąd rośnie z każdą iteracją, co oznacza, że rozwiązanie nie ma tendencji do bycia bardziej dokładnym, ale odwrotnie. Dlatego te metody iteracyjne nie są odpowiednie do znajdowania rozwiązań tego typu równań

Zadanie D

Spróbujmy rozwiązać równania z zadania C metodą bezpośrednią. Do znalezienia rozwiązania wykorzystamy metodę faktoryzacji LU. Otrzymujemy następujące wyniki:

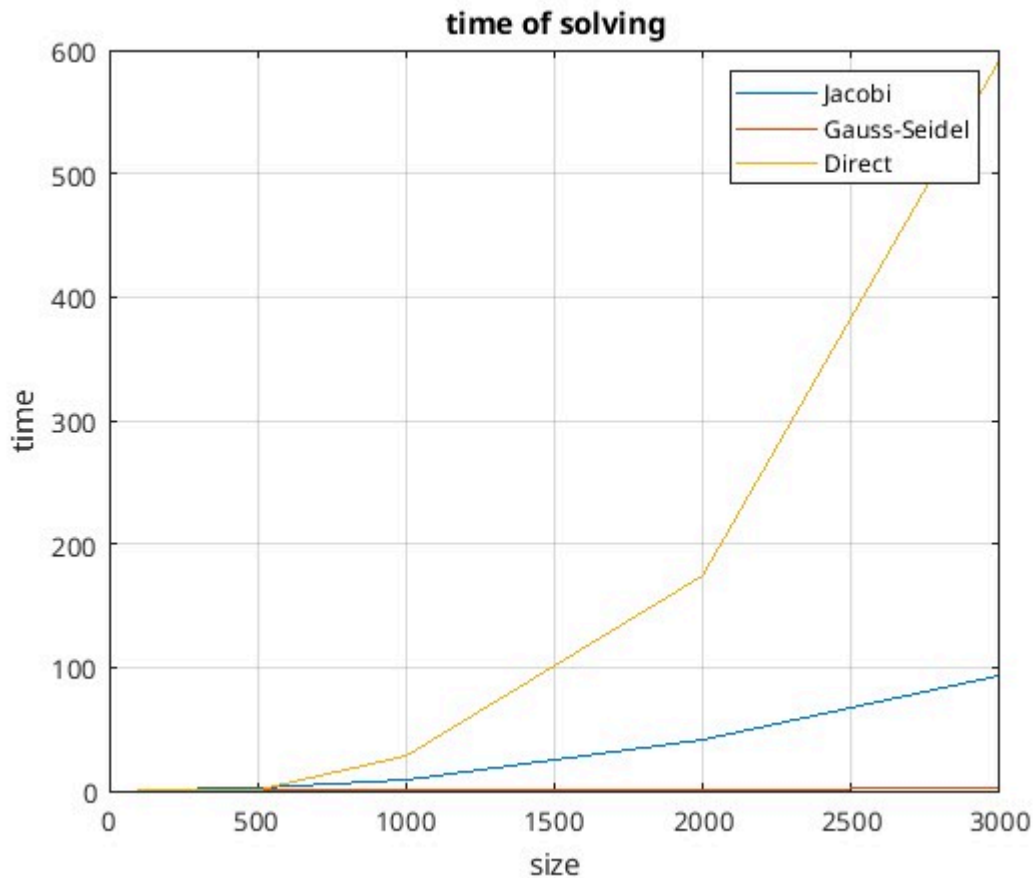
Znormalizowany błąd = $7.6643e-10$

Czas liczenia = 24.85s

Na podstawie tych danych możemy powiedzieć, że metoda ta może być używana do znajdowania rozwiązań podobnych równań, ponieważ daje odpowiedź z dużą dokładnością.

Zadanie E

Znajdźmy zależność czasu obliczeń od rozmiaru macierzy, dla których ma zostać znalezione rozwiązanie.



Z wykresu łatwo zauważyć, że wraz ze wzrostem rozmiaru macierzy, metoda bezpośrednia zaczyna obliczać znacznie dłużej, podczas gdy metody iteracyjne spowalniają znacznie mniej.

Zadanie F

Po wykonaniu powyższych obliczeń można dojść do następujących wniosków. Chociaż metody iteracyjne wykonują obliczenia dość szybko i nie zależą tak bardzo od rozmiaru danych wejściowych, mogą nie działać we wszystkich przypadkach. Pozwalają one również na znalezienie rozwiązań o określonej dokładności, co może dodatkowo zmniejszyć liczbę iteracji, a w konsekwencji czas wykonania. Warto również zauważyć, że metoda Gaussa-Seidla wykazuje lepszą wydajność niż metoda Jacobiego.

Metoda bezpośrednia ma jednak swoją zaletę. Pozwala na znalezienie rozwiązania niezależnie od zawartości macierzy. Jest również stosunkowo łatwiejsza w implementacji. Daje jej to pierwszeństwo przed metodami iteracyjnymi, gdy rozmiar danych nie jest duży, a szybkość nie jest ważna.