Metody Numeryczne, Projekt 2

Hleb Badzeika

Styczeń 2024

1 Krótki opis metody

Dekompozycja LDL jest metoda faktoryzacji macierzy, która rozkłada hermitowska, dodatnio określona macierz A na iloczyn trzech macierzy: dolnej trójkatnej L, diagonalnej D oraz hermitowskiego sprzeżenia dolnej trójkatnej L^H. Matematycznie, dekompozycje te można wyrazić jako:

$$A = LDL^H$$

gdzie:

- A jest hermitowska, dodatnio określona macierza.
- L jest dolna macierza trójkatna z jedynkami na głównej przekatnej.
- D jest macierza diagonalna.
- L^H jest hermitowskim sprzeżeniem L, co oznacza dolna macierz trójkatna złożona z elementów zespolonych sprzeżonych do odpowiadajacych im elementów w L.

Dekompozycja LDL jest szczególnie użyteczna do rozwiazywania równań liniowych postaci $Ax = b$, gdzie A jest macierza hermitowska. W przypadku macierzy rzeczywistych, sprzeżenie hermitowskie L^H redukuje sie do transpozycji L^T, a dekompozycja przyjmuje postać LDL^T.

Dla danej hermitowskiej, dodatnio określonej macierzy A, metoda Cholesky'ego–Banachiewicza pozwala na znalezienie takiej dolnej macierzy trójkatnej L, że $A = LDL^H$, gdzie L^H oznacza sprzeżenie hermitowskie macierzy L, a D jest macierza diagonalna.

Proces rozkładu dla macierzy pentadiagonalnej A odbywa sie według poniższego schematu:

1. Rozpoczynamy od pierwszej kolumny macierzy A i wyznaczamy elementy macierzy L oraz D.
2. Dla każdego i-tego wiersza macierzy A, obliczamy element l_{ii} jako pierwiastek kwadratowy z $a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 d_{kk}$.

3. Elementy poza przekatna macierzy L, l_{ij} dla $j < i$, obliczamy korzystajac z wzoru:
$$l_{ij} = \frac{1}{d_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}^{*} d_{kk} \right)$$
, gdzie l_{jk}^{*} oznacza sprzeżenie zespolone elementu l_{jk}.

4. Po obliczeniu wszystkich elementów macierzy L i D, możemy rozwiazać równanie $XA = B$ poprzez podstawienie $A = LDL^H$ i rozwiazanie dwóch równań trójkatnych: $LY = B$ oraz $L^H X = D^{-1} Y$.

W przypadku macierzy pentadiagonalnej, operacje te sa bardziej efektywne ze wzgledu na mniejsza liczbe niezerowych elementów w macierzy A.

2 Krótki opis programu

Implementacja w MATLAB-ie algorytmu dekompozycji LDL skupia sie na obsłudze hermitowskich, pieciоprzekatnych, dodatnio określonych macierzy. Kod składa sie z kilku funkcji:

2.1 Przeglad implementacji

Program zawiera nastepujace funkcje:

- `LDLDecomposition(A)`: Rozkłada macierz A na LDL^H.
- `solveLDLH(A, B)`: Rozwiazuje równanie $XA = B$ za pomoca dekompozycji LDL.
- Funkcje do generowania macierzy testowych: Różne funkcje do tworzenia specyficznych typów macierzy do testowania.

2.2 Opis funkcji

2.2.1 `LDLDecomposition(A)`

Wejście: Hermitowska, pieciоprzekatna, dodatnio określona macierz A.
Wyjście: Macierze L i D z rozkładu $A = LDL^H$.

Funkcja ta wykonuje rozkład LDL macierzy A, zwracajac dolna macierz trójkatna L i macierz diagonalna D, tak że $A = L * D * L^H$.

2.2.2 `solveLDLH(A, B)`

Wejście: Macierz A oraz macierz B o odpowiednich wymiarach.
Wyjście: Rozwiazanie X równania $XA = B$.

Funkcja ta wykorzystuje rozkład LDL macierzy A do rozwiazania równania $XA = B$, stosujac do tego celu metode podstawienia w przód i wstecz.

2.3 Funkcje do generowania macierzy testowych

Opracowane funkcje służa do generowania macierzy dla konkretnych przypadków testowych, każda z nich tworzy macierz o unikalnych właściwościach:

2.3.1 generateWellConditionedMatrix(n)

Wejście: Rozmiar macierzy n.
Wyjście: Dobrze uwarunkowana, hermitowska, piecioprzekatna, dodatnio określona macierz.
Funkcja ta generuje dobrze uwarunkowana macierz, zapewniajac umiarkowane wartości własne dla stabilności numerycznej.

2.3.2 generateIllConditionedMatrix(n)

Wejście: Rozmiar macierzy n.
Wyjście: Źle uwarunkowana, hermitowska, piecioprzekatna, dodatnio określona macierz.
Ta funkcja tworzy macierz źle uwarunkowana poprzez zastosowanie szerokiego zakresu wartości własnych, co czyni ja numerycznie trudniejsza do rozwiazania.

2.3.3 generateLargeMatrix(n)

Wejście: Rozmiar macierzy n.
Wyjście: Duża, hermitowska, pieciоprzekatna, dodatnio określona macierz.
Funkcja generuje duża macierz, co pozwala na ocene wydajności i skalowalności algorytmu.

2.3.4 generateMatrixWithSmallElements(n)

Wejście: Rozmiar macierzy n.
Wyjście: Macierz z małymi elementami, hermitowska, pieciоprzekatna, dodatnio określona.
Ta funkcja tworzy macierz z bardzo małymi liczbami, testujac precyzje algorytmu w warunkach bliskich precyzji maszynowej.

2.3.5 generateMatrixWithRandomNoise(n)

Wejście: Rozmiar macierzy n.
Wyjście: Macierz z losowym szumem, hermitowska, pieciоprzekatna, dodatnio określona.
Funkcja dodaje losowy szum do dobrze uwarunkowanej macierzy, symulujac tym samym rzeczywiste dane, które moga nie być idealnie ustrukturyzowane.

2.3.6 generateMatrixWithStructuredPattern(n)

Wejście: Rozmiar macierzy n.
Wyjście: Macierz o określonym wzorze strukturalnym, hermitowska, piecioprzekatna, dodatnio określona. Ta funkcja tworzy macierz z wykorzystaniem specyficznego, powtarzajacego sie wzoru liczbowego, zachowujac przy tym jej własności hermitowskie i dodatnio określone.

2.4 Analiza i testowanie

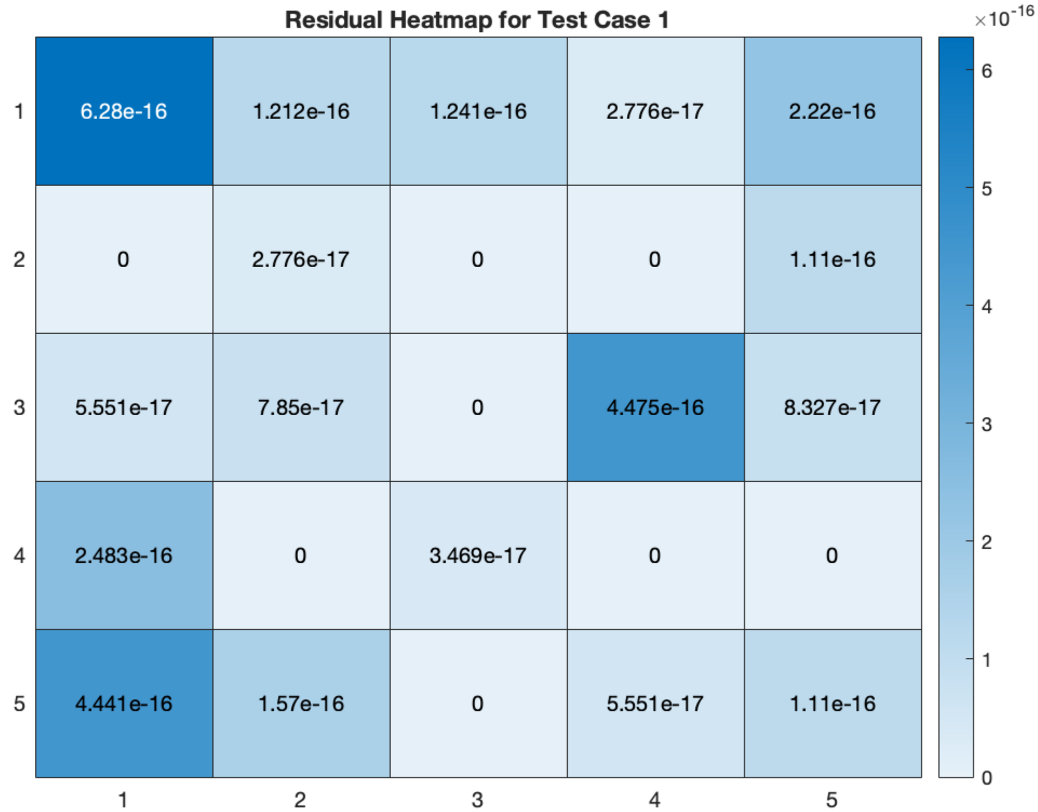
Do analizy i testowania wydajności algorytmu wykorzystano szereg przypadków testowych. Każdy przypadek został rozwiazany przy użyciu opracowanego algorytmu, a nastepnie dokładnie zanalizowano wyniki, biorac pod uwage błedy i czas obliczeń. Dodatkowo, wykorzystano techniki wizualizacji, takie jak mapy ciepła, do lepszego zrozumienia rozkładu błedów.

3 Ciekawe przykłady

Analiza czasu wykonania jest zrobiana dla macierzy 1000x1000 dla dokładniejszych wyników. Widać że różnica w czasie wykonania jest niewielka, około 5%. Widać również że pattern błedu resztkowego też jest prawie taki sam dla wszystkich testcasów (oprócz ostatniego, o nim później), dlatego możemy powiedzieć, że jak już mamy macierz A z zadanymi warunkami, to algorytm działa stabilne.

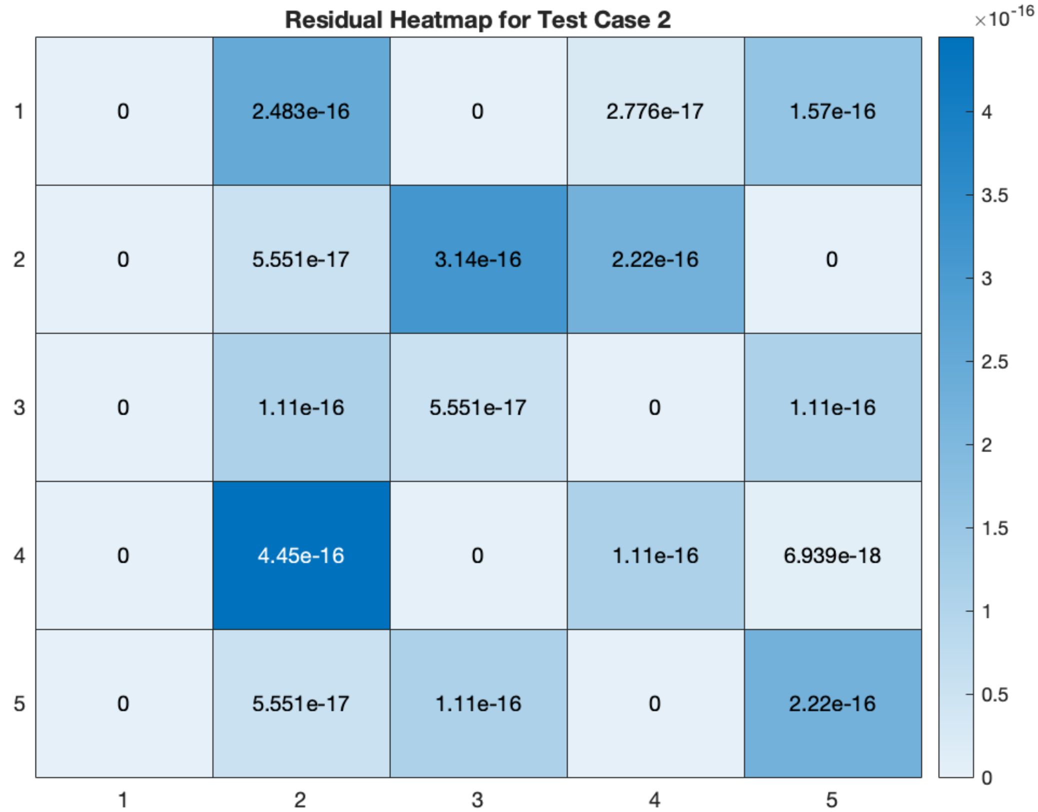
3.1 Dobrze uwarunkowana macierz

Czas wykonania: 0.9597 s



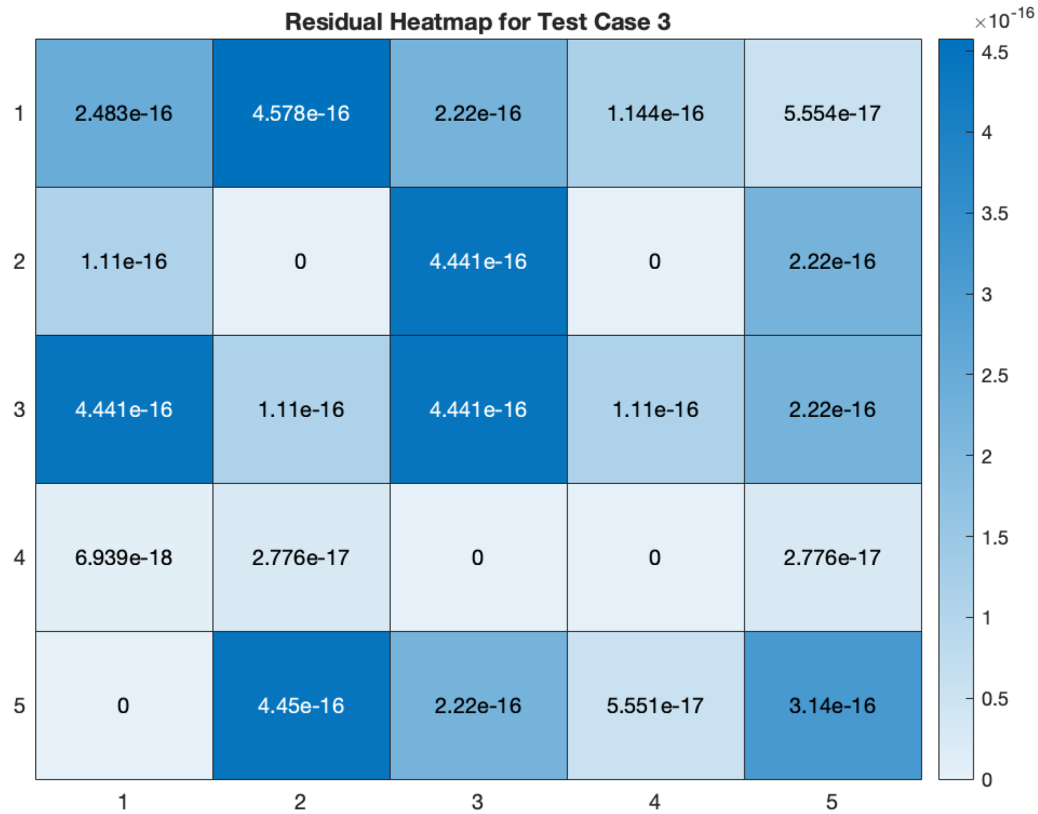
3.2 Zle uwarunkowana macierz

Czas wykonania: 0.8776 s



3.3 Macierz z dużymi elementami

Czas wykonania: 0.9469 s

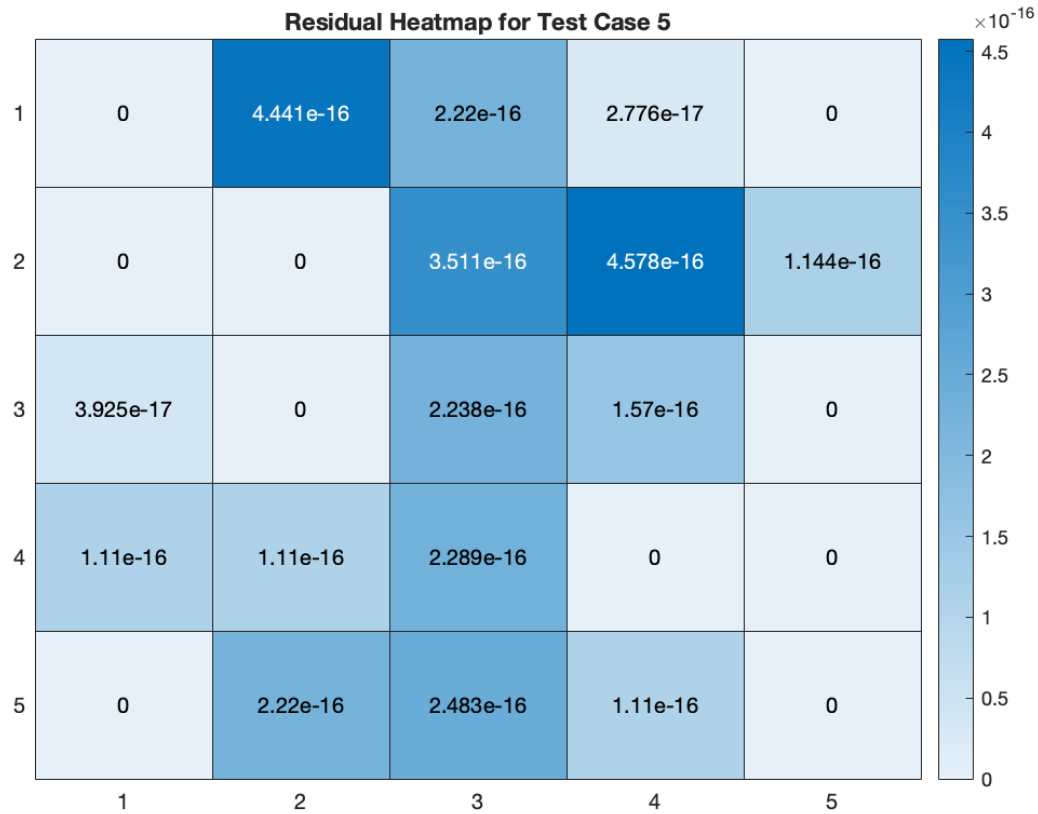


3.4 Macierz z małymi elementami

Czas wykonania: 0.8844 s

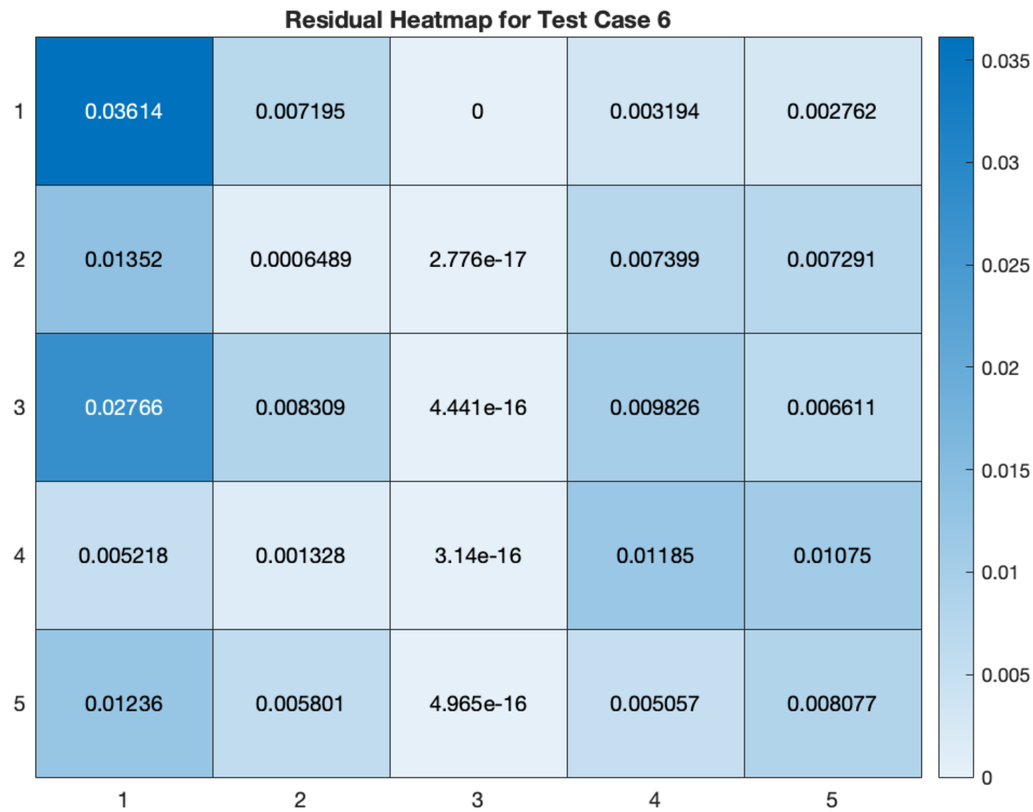
3.5 Macierz z wyróżnionym patternem

Czas wykonania: 0.8319 s



3.6 Macierz z dodatkowum szumem

Czas wykonania: 0.8371 s



Widać, że dodajac szum do danych. Otrzymujemy błed na kilka porzadków gorszy niż w pozostałych testcase'ach. To może świadczyć o tym że ten algorytm może być nie najlepszym dla wykorzystania w praktyce na nieoprzygotowanych danych.

4 Zastosowanie metody w praktycznym użyciu

Dekompozycja LDL znajduje wiele praktycznych zastosowań, zwłaszcza w dziedzinach takich jak uczenie maszynowe i nauka o danych. Poniżej przedstawiono kilka przykładów takich zastosowań:

4.1 Optymalizacja w Uczeniu Maszynowym

W uczeniu maszynowym dekompozycja LDL jest wykorzystywana do rozwiazywania problemów optymalizacji, szczególnie w algorytmach opartych na metodach kwadratów najmniejszych i optymalizacji wypukłej. Na przykład, w algorytmach takich jak Maszyny Wektorów Wspierajacych (SVM), dekompozycja LDL może być używana do efektywnego rozwiazywania równań liniowych, które sa kluczowe dla znalezienia hiperpłaszczyzny rozdzielajacej.

4.2 Analiza Głównych Składowych (PCA)

W naukach o danych, szczególnie w analizie głównych składowych (PCA), dekompozycja LDL może być stosowana do efektywnej faktoryzacji macierzy kowariancji. Pozwala to na redukcje wymiarowości danych, zachowujac przy tym istotne informacje statystyczne, co jest szczególnie ważne w przypadku dużych zbiorów danych.