Rozwiązywanie układu równań liniowych $Ax = b$, gdzie $A(n \times n)$ jest macierzą symetryczną dodatnio określoną

Kornel Tłaczała

8 stycznia 2024

Projekt nr 2

1 Opis probelmu

Cel projektu

Niech $A \in \mathbb{R}^{n \times n}$ będzie dodatnio określoną macierzą symetryczną postaci:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix}, \quad A_{ij} \in \mathbb{R}^{p \times p} \quad \wedge \quad n = 2p$$

Zadaniem jest rozwiązać układ równań liniowych $Ax = b$ korzystając z blokowego rozkładu UDU^T macierzy A ($A = UDU^T$).

Metoda rozwiązania

Szukanie macierzy U i D

Jeżeli przyjmiemy, że U jest macierzą o postaci:

$$U = \begin{bmatrix} I & U_{12} \\ 0 & I \end{bmatrix}$$

oraz D jest macierzą o postaci:

$$D = \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix}$$

to zgodnie ze schematem mnożenia macierzy blokowych otrzymujemy:

$$A = UDU^T$$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \begin{bmatrix} I & U_{12} \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ U_{12}^T & I \end{bmatrix} =$$
$$= \begin{bmatrix} D_{11} & U_{12} \cdot D_{22} \\ 0 & D_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ U_{12}^T & I \end{bmatrix} =$$
$$= \begin{bmatrix} D_{11} + U_{12} \cdot D_{22} \cdot U_{12}^T & U_{12} \cdot D_{22} \\ D_{22} \cdot U_{12}^T & D_{22} \end{bmatrix}$$

Na tej podstawie łatwo będzie znaleźć macierze U oraz D za pomocą następującego układu równań, wyznaczając po kolei:

$$\begin{cases} D_{22} = A_{22} \\ D_{22} \cdot U_{12}^T = A_{12}^T, \quad U_{12} = (U_{12}^T)^T \\ D_{11} + U_{12} \cdot D_{22} \cdot U_{12}^T = A_{11} \end{cases}$$

Szukanie wektora x

Jesteśmy zatem w stanie wyznaczyć macierze U oraz D. Będziemy mogli to wykorzystać aby policzyć $Ax = b$. Będziemy liczyli według następującego schematu:

$$\begin{aligned} DU^T x &= b \\ Uz &= b \quad \Longrightarrow \quad DU^T x = z \\ Dy &= z \quad \Longrightarrow \quad U^T x = y \\ U^T x &= y \end{aligned}$$

Rozwiązując po kolei równania możemy wyznaczyć najpierw $z(1)$, potem $y(2)$ i na koniec $x(3)$. Takie rozwiązanie upraszcza poszczególne obliczenia, ponieważ w równaniach (1) oraz (3) mamy do czynienia z macierzami górną trójkątną oraz dolną trójkątną co bardzo ułatwia, oraz przyśpiesza obliczenia. Równanie (2) natomiast można rozwiązać korzystając z następującej zależności:

$$Dy = z$$
$$\begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

zatem:

$$\begin{cases} D_{11} \cdot y_1 = z_1 \\ D_{22} \cdot y_2 = z_2 \end{cases}$$

Wiemy, że $det(D_{ii}) \neq 0$, bo w przeciwnym wypadku $det(D) = 0 \implies det(UDU^T) = 0 \implies det(A) = 0$. Wtedy jednak macierz A miałaby wartość własną równą 0, przez co nie mogłaby być dodatnio określona. Wiemy natomiast, że macierz A jest dodatnio określona, zatem $det(D_{ii}) \neq 0$.

Możemy więc rozbić macierz D_{ii} na iloczyn LU otrzymując $L_{Dii}U_{Dii} \cdot y_i = z_i$. To z kolei łatwo policzyć wykorzystując analogiczne podejście, tj. obliczając v_i z $L_{Dii} \cdot v_i = z_i$, a następnie podstawiając do $U_{Dii} \cdot y_i = v_i$. Na koniec oczywiście:

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

2 Opis programu obliczeniowego

W celu implementacji opisanej metody wykorzystałem kilka przedstawionych poniżej funkcji. Wszystkie zostały zwektoryzowane w celu zwiększenia wydajności wykonania.

Funkcje obliczeniowe

splitZ(z)

Przyjmuje wektor z, zwraca 2 mniejsze wektory z_1 oraz z_2. Przydatna przy obliczaniu y.

splitD(D)

Przyjmuje macierz D, zwraca 2 mniejsze macierze D_{11} oraz D_{22}. Przydatna przy obliczaniu y.

LUdecomposition(A)

Przyjmuje macierz A, zwraca macierze L oraz U, takie, że $LU = A$. Funkcja wykorzystuje rozkład Doolittle'a.

linsolveForUpper(U, B)

Przyjmuje macierz górną trójkątnąU oraz macierz lub wektor B, zwraca wektor lub macierz X taki, że $UX = B$. Wykorzystuje charakterystykę macierzy górnej trójkątnej, by przyśpieszyć rozwiązywanie układu równań.

linsolveForLower(L, B)

Przyjmuje macierz dolną trójkątnąL oraz macierz lub wektor B, zwraca wektor lub macierz X taki, że $LX = B$. Wykorzystuje charakterystykę macierzy dolnej trójkątnej, by przyśpieszyć rozwiązywanie układu równań.

linsolveLU(A, B)

Przyjmuje macierz$A$ oraz macierz lub wektor B, zwraca wektor lub macierz X taki, że $AX = B$. Wykorzystuje LUdecomposition aby otrzymać potrzebne macierze L oraz U a następnie wykorzystuje linsolveForLower() oraz linsolveForUpper() by policzyć X.

UDUTdecomposition(A)

Przyjmuje macierz$A$, zwraca macierze U oraz D takie, że $UDU^T = A$. Wykorzystuje linsolveLU() aby otrzymać potrzebną macierz U_{12}.

linsolveUDUT(A, b)

Przyjmuje macierz$A$ oraz wektor b, zwraca wektor x, takie, że $Ax = b$. Wykorzystuje UDUTdecomposition(), linsolveForUpper(), linsolveForLower(), splitZ(), splitD() oraz linsolveLU().

Funkcje testujące

getMatrix(p, variance)

Przyjmuje wartości p, oraz *variance*. Tworzy macierz $(2 * p \times 2 * p)$ wykorzystując funkcję randn(). Mnoży tą macierz przez *variance* i ją zwraca.

testfor(p, variance)

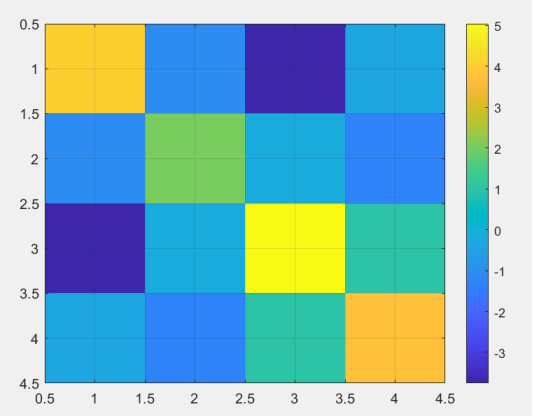
Przyjmuje wartości p, oraz *variance*. Tworzy macierz $A(2*p \times 2*p)$ wykorzystując funkcję getMatrix(). Tworzy wektor $b(2 * p)$. Liczy $Ax = b$ wykorzystując funkcję linsolveUDUT() oraz wbudowaną funcję linsolve(). Porównuje wyniki i czas wykonania. Rezultaty wypisuje w konsoli.

3 Przykłady obliczeniowe

Przykład 1

```
Testing for:
p =      2

variance =      1


matrix =

    9.8160    0.3566    2.0805   -2.6352
    0.3566    3.4551   -0.9534    0.2301
    2.0805   -0.9534    3.2923   -0.5416
   -2.6352    0.2301   -0.5416    4.0670


vector =

 -133.3678
  112.7492
   35.0179
  -29.9066

Result comparison for first 5 indices:
    myResult    matlabResult    AbsoluteError    RelativeError
    ________    ____________    _____________    _____________

    -30.344       -30.344        3.5527e-15       1.1708e-16
     48.375        48.375        7.1054e-15       1.4688e-16
     39.798        39.798        7.1054e-15       1.7854e-16
    -24.452       -24.452                 0                0

My time (over 1000 repetitions):     0.0734

BuiltIn function time (over 1000 repetitions):    5.5130e-04
```

(a) Caption for the first figure



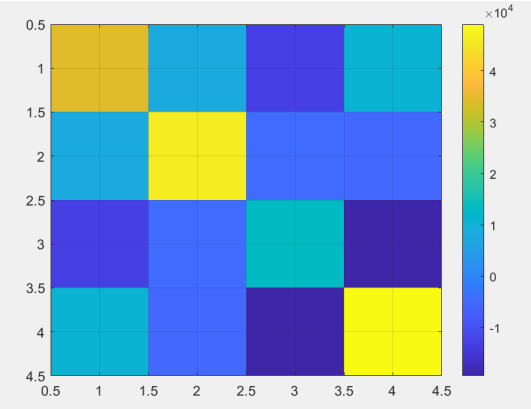
(b) Caption for the second figure

Rysunek 1: $A1 \cdot x = b$

Przykład 2

```
Command Window
  Testing for:
  p =      2

  variance =        10000


  matrix =

     1.0e+04 *

      3.4125    0.8076   -1.2967    1.0769
      0.8076    4.6151   -0.4657   -0.5320
     -1.2967   -0.4657    1.3360   -1.9385
      1.0769   -0.5320   -1.9385    4.9034


  vector =

    -13.0285
     18.3689
    -47.6153
     86.2022

  Result comparison for first 5 indices:
       myResult      matlabResult     AbsoluteError     RelativeError
      ____________    ____________    ______________    _____________

       -0.0028415      -0.0028415                 0                0
        0.0002278       0.0002278       1.8974e-19       8.329e-16
       -0.0064505      -0.0064505       8.6736e-19       1.3446e-16
      -0.00014337     -0.00014337       2.1684e-19       1.5125e-15

  My time (over 1000 repetitions):      0.1103

  BuiltIn function time (over 1000 repetitions):      0.0017

fx >>
```

(a) Caption for the first figure



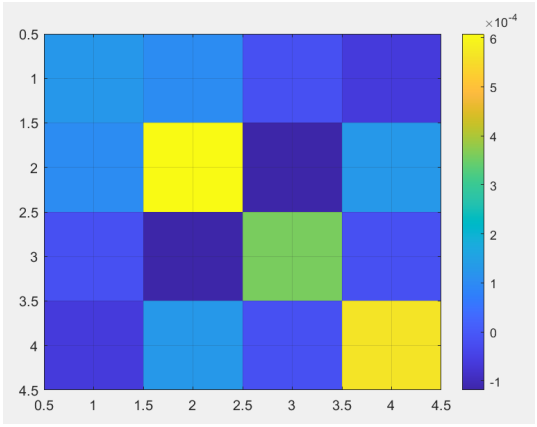
(b) Caption for the second figure

Rysunek 2: $A1 \cdot x = b$

Przykład 3

```
Command Window
  Testing for:
  p =      2

  variance =    1.0000e-04


  matrix =

     1.0e-03 *

      0.1277    0.1008   -0.0189   -0.0638
      0.1008    0.6078   -0.1186    0.1284
     -0.0189   -0.1186    0.3610   -0.0166
     -0.0638    0.1284   -0.0166    0.5652


  vector =

    152.6977
     46.6914
    -20.9713
     62.5190

  Result comparison for first 5 indices:
       myResult      matlabResult     AbsoluteError     RelativeError
    ____________    ____________    ______________    ______________

      1.5732e+06      1.5732e+06                 0                 0
     -2.6666e+05     -2.6666e+05                 0                 0
           -47488          -47488       1.4552e-11        3.0643e-16
      3.4736e+05      3.4736e+05                 0                 0

  My time (over 1000 repetitions):     0.0745

  BuiltIn function time (over 1000 repetitions):    5.9440e-04

fx >>
```

(a) Caption for the first figure



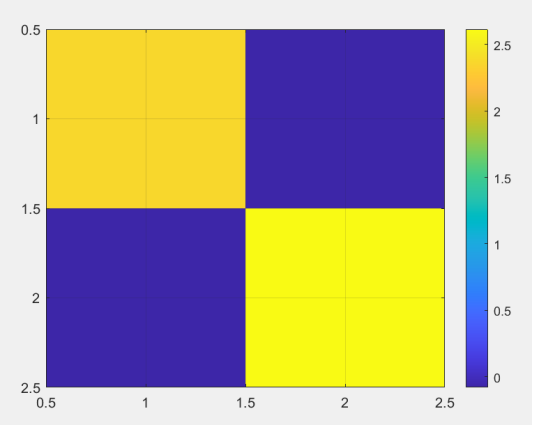
(b) Caption for the second figure

Rysunek 3: $A1 \cdot x = b$

Przykład 4

```
Command Window
  Testing for:
  p =      1

  variance =      1


  matrix =

      2.3665   -0.0805
     -0.0805    2.6141


  vector =

     13.5175
     51.5246

  Result comparison for first 5 indices:
      myResult    matlabResult    AbsoluteError    RelativeError
      ________    ____________    _____________    _____________

       6.3897        6.3897        8.8818e-16         1.39e-16
       19.907        19.907        3.5527e-15        1.7847e-16

  My time (over 1000 repetitions):     0.0687

  BuiltIn function time (over 1000 repetitions):    6.5090e-04

fx >>
```

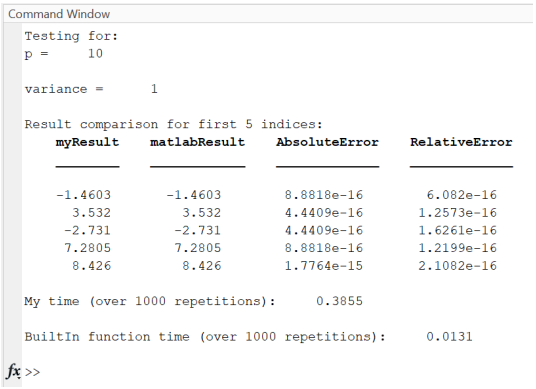
(a) Caption for the first figure



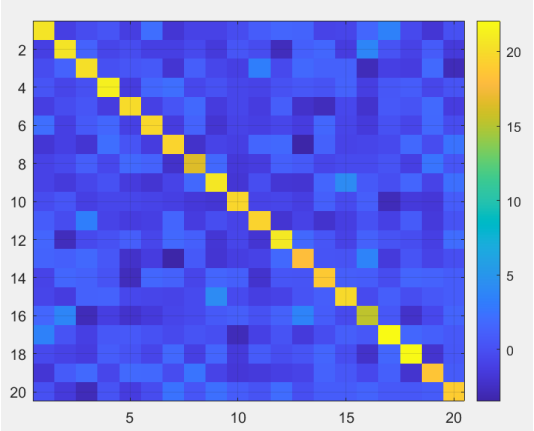
(b) Caption for the second figure

Rysunek 4: $A1 \cdot x = b$

Przykład 5



(a) Caption for the first figure



(b) Caption for the second figure

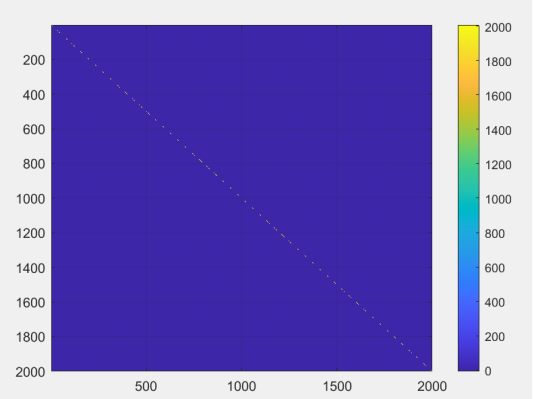
Rysunek 5: $A1 \cdot x = b$

Przykład 6

Command Window

```
Testing for:
p =         1000

variance =      1

Result comparison for first 5 indices:
    myResult     matlabResult    AbsoluteError    RelativeError
    ________     ____________    _____________    _____________

     0.013146       0.013146       3.4694e-18       2.6392e-16
    -0.014667      -0.014667       1.5613e-17       1.0644e-15
     0.012978       0.012978       7.6328e-17       5.8811e-15
    -0.047198      -0.047198       1.7347e-16       3.6755e-15
    0.0013669      0.0013669        9.541e-18       6.9802e-15

My time (over 1 repetition):     6.2192

BuiltIn function time (over 1 repetition):     0.0659

fx >>
```

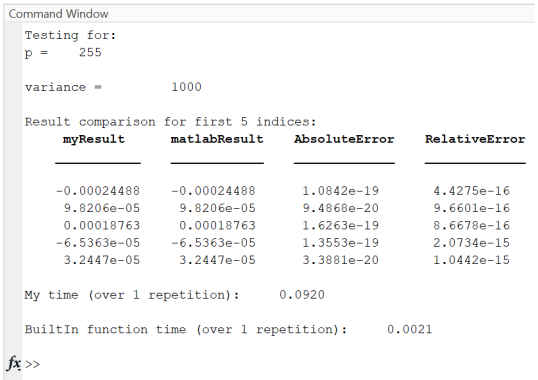
(a) Caption for the first figure



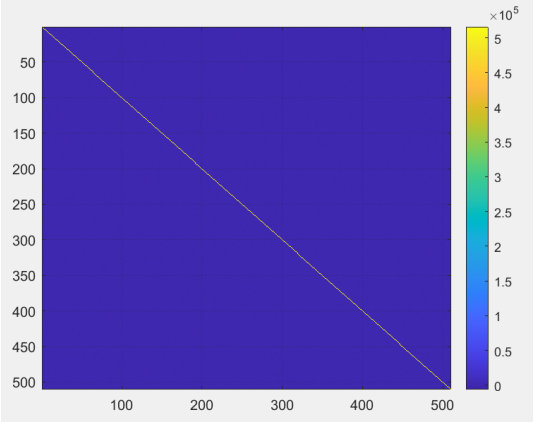
(b) Caption for the second figure

Rysunek 6: $A1 \cdot x = b$

Przykład 7



(a) Caption for the first figure



(b) Caption for the second figure

Rysunek 7: $A1 \cdot x = b$