Projekt 2 - Eliminacja Gaussa i LU faktoryzacja

Temat - Zaimplementować dla macierzy o rozmiarze = (miesiąc urodzenia + dzień urodzenia)
1) Algorytm eliminacji Gaussa bez pivotingu generujący jedynki na przekątnej (slajd 14)
2) Algorytm eliminacji Gaussa z pivotingiem (slajd 26)
3) Algorytm LU faktoryzacji bez pivotingu (slajd 33)
4) Algorytm LU faktoryzacji z pivotingiem (slajd 36)

Bartłomiej Jamiołkowski

W projekcie macierz ma rozmiar N = 15, ponieważ jest to suma mojego dnia i miesiąca urodzenia, czyli 10.05.

1 Algorytm eliminacji Gaussa bez pivotingu generujący jedynki na przekątnej

1.1 Pseudokod

Algorithm 1 Gaussian Elimination without Pivoting

Require: A - a square matrix $n \times n$
Require: b - a column vector $n \times 1$
Ensure: x - a column vector $n \times 1$ representing the solution to the system of equations $Ax = b$

```
for i = 0 to n − 1 do
    for j = 0 to n − 1 do
        if j == i then
            b[i] ← b[i]/A[i, i]
            A[i, :] ← A[i, :]/A[i, i]
        else if j > i then
            b[j] ← b[j] − b[i] × A[j, i]
            A[j, :] ← A[j, :] − A[i, :] × A[j, i]
        end if
    end for
end for
x ← SolveUpperTriangular(A, b)    ▷ Solve system of equations with upper triangular matrix A
return x
```

Działanie programu można wyjaśnić następująco:

1. Funkcja przyjjmuje dwie argumenty: macierz kwadratową A oraz wektor kolumnowy b.
2. Pętla zewnętrzna `for` i `in` `range`$(A.shape[0])$ iteruje po wierszach macierzy A.
3. Wewnętrzna pętla `for` j `in` `range`$(A.shape[1])$ iteruje po kolumnach macierzy A.
4. Warunek `if` $j == i$ sprawdza, czy aktualna kolumna odpowiada obecnie przetwarzanemu wierszowi. Innymi słowy sprawdza czy element macierzy jest na jej przekątnej. Jeśli tak, to wykonują się następujące czynności:
   - Element $b[i]$ dzielony jest przez odpowiadający element na przekątnej macierzy A (element diagonalny).
   - Cały wiersz $A[j, :]$ dzielony jest przez element przekątnej $A[i, i]$. Dzięki temu element na przekatnej staje się równy 1.
5. Warunek `elif` $j > i$ odnosi się do kolumn poza główną przekątną macierzy A. Dla tych kolumn wykonują się następujące czynności:
   - Elementy wektora b są aktualizowane przez odjęcie iloczynu $b[i]$ i elementu $A[j, i]$.
   - Wiersze macierzy A są aktualizowane poprzez odjęcie iloczynu wiersza $A[i, :]$ i elementu $A[j, i]$. Dzieje się to w celu wyeliminowania niezerowych wartości pod główną przekątną macierzy.
6. Proces ten kontynuuje się dla kolejnych wierszy i kolumn, aż do zakończenia przetwarzania całej macierzy A.
7. Na koniec działania funkcji zwracana jest przekształcona macierz A oraz zaktualizowany wektor b, które zawierają rozwiązania układu równań po eliminacji Gaussa w x.

1.2 Kod algorytmu w języku Python

```python
def gaussian_elimination_without_pivoting(A: np.ndarray, b: np.ndarray) -> np.ndarray:
    for i in range(A.shape[0]):
        for j in range(A.shape[1]):
            if j == i:
                b[i] /= A[i, i]
                A[j, :] /= A[i, i]
            elif j > i:
                b[j] -= b[i] * A[j, i]
                A[j, :] -= A[i, :] * A[j, i]

    return np.linalg.solve(A, b)
```

Rysunek 1: Kod algorytmu eliminacji Gaussa generujący 1 na przekątnej

1.3 Test dla macierzy gęstej o losowych wartościach (porównanie z MATLAB)

Sprawdzenie czy algorytm generuje jedynki na przekątnej.

```
[[ 1.      4.      1.      7.      2.      5.      2.      5.      3.      1.      6.      3.      8.      3.      3.     ]
 [ -0.     1.      0.25    2.75   -0.75    1.75   -0.      0.5    -0.25   -0.75    1.75    1.25    2.5    -0.5     0.75  ]
 [ -0.    -0.      1.    -49.     77.    -29.     16.     54.     75.     73.     -5.    -51.    -26.     86.     11.    ]
 [  0.     0.      0.      1.     -1.5238  0.5476 -0.4048 -1.0476 -1.4286 -1.2381  0.0952  0.8095  0.4048 -1.6429 -0.1905]
 [ -0.    -0.     -0.     -0.      1.     -0.7455 -0.7818  1.1273  1.4727  3.4     0.1455 -3.0182 -1.4545  1.5818  0.3091]
 [  0.     0.      0.      0.      0.      1.      1.4896 -1.1669 -1.5913 -4.529   0.5376  4.3803  2.6217 -1.8029 -0.1534]
 [ -0.    -0.     -0.     -0.     -0.     -0.      1.      1.6966  1.1526 -1.7068 -0.6427 -0.3792  2.4951  1.1264 -1.1695]
 [ -0.    -0.     -0.     -0.     -0.     -0.     -0.      1.      1.1178  0.8346 -0.1195 -1.6031  0.326   1.1526 -0.472 ]
 [ -0.    -0.     -0.     -0.     -0.     -0.     -0.     -0.      1.      1.442  -0.8283 -3.2603  0.7448  1.4815 -4.0592]
 [  0.     0.      0.      0.      0.      0.      0.      0.      0.      1.      0.1065  0.7497 -1.1093 -0.4437  0.9361]
 [  0.     0.      0.      0.      0.      0.      0.      0.      0.      0.      1.     10.1389 -6.6095 -3.3187 10.6662]
 [ -0.    -0.     -0.     -0.     -0.     -0.     -0.     -0.     -0.     -0.     -0.      1.     -0.619  -0.2807  0.8764]
 [  0.     0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      1.     -8.7191 -77.1361]
 [  0.     0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      1.     10.7476]
 [  0.     0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      1.    ]]
```

Rysunek 2: Przekształcona macierz górna trójkątna A z 1 na przekątnej

Losowanie macierzy gęstej A z wartościami losowymi.

```
Matrix A:
[[1 4 1 7 2 5 2 5 3 1 6 3 8 3 3]
 [2 4 1 3 7 3 4 8 7 5 5 1 6 8 3]
 [6 5 1 2 7 4 8 7 4 2 4 7 7 6 1]
 [7 6 2 6 5 5 5 7 4 8 5 2 5 6 2]
 [4 5 6 3 1 7 2 1 4 2 1 4 4 6 5]
 [8 6 7 7 7 8 3 1 3 8 6 3 5 5 6]
 [3 8 2 7 1 8 2 5 4 8 7 7 2 6 8]
 [5 7 3 8 5 1 3 8 2 4 1 6 5 4 6]
 [6 1 2 3 4 6 4 4 3 3 8 1 5 2 7]
 [5 2 6 7 2 7 6 5 6 5 1 4 8 1 1]
 [8 4 2 8 5 7 7 8 8 2 6 7 7 5 3]
 [7 8 3 8 4 2 7 8 1 2 5 6 8 1 8]
 [1 6 4 5 4 2 2 8 7 1 6 3 5 4 2]
 [8 8 1 8 2 4 3 7 2 7 2 7 2 3 1]
 [1 6 5 1 1 5 3 5 2 8 5 3 1 1 2]]
```

Rysunek 3: Macierz gęsta A

Losowanie wektora b prawej strony z wartościami losowymi.

```
Vector b:
[[2]
 [5]
 [1]
 [6]
 [8]
 [7]
 [5]
 [6]
 [8]
 [6]
 [8]
 [4]
 [1]
 [7]
 [2]]
```

Rysunek 4: Wektor prawej strony b

Rozwiązanie układu równań moim programem.

```
Result x:
[[ 0.1095]
 [ 1.5512]
 [-1.2596]
 [ 0.0176]
 [ 1.9039]
 [ 3.7982]
 [-0.3328]
 [ 1.5609]
 [ 0.3064]
 [-1.3155]
 [-2.3532]
 [-2.0441]
 [-1.3114]
 [-1.6014]
 [ 0.9425]]
```

Rysunek 5: Rozwiązanie układu równań x

Rozwiązanie układu równań programem MATLAB.

```
>> x = A\b

x =

    0.1095
    1.5512
   -1.2596
    0.0176
    1.9039
    3.7982
   -0.3328
    1.5609
    0.3064
   -1.3155
   -2.3532
   -2.0441
   -1.3114
   -1.6014
    0.9425
```

Rysunek 6: Rozwiązanie układu równań x w MATLAB

Porównanie wyników mojego programu i MATLAB norm(x1-x2,2).

```
>> n = norm(x-x2,2)

n =

   9.7250e-05
```

Rysunek 7: Wynik rnorm(x1 - x2, 2) x w MATLAB

Na podstawie wydrukowanych przez programy wyników można zauważyć, że w tym przypadku są one takie same. Potwierdza to wynik 9.7250e-05 w MATLAB, który wskazuje, że różnice między poszczególnymi wynikami w wektorach x1 i x2 są bardzo małe.

2 Algorytm eliminacji Gaussa z pivotingiem

2.1 Pseudokod

Algorithm 2 Gaussian Elimination with Pivoting

Require: A is a square matrix
Require: b is a vector
Ensure: Solution vector x of the linear system $Ax = b$

for $i \leftarrow 0$ **to** $A.\text{shape}[0] - 1$ **do**
  $A[\{i, -1\}] \leftarrow A[\{-1, i\}]$
  $b[i], b[-1] \leftarrow b[-1], b[i]$
  for $j \leftarrow 0$ **to** $A.\text{shape}[1] - 1$ **do**
    if $j > i$ **then**
      $b[j] \leftarrow b[j] - \frac{b[i] \times A[j,i]}{A[i,i]}$
      $A[j,:] \leftarrow A[j,:] - \frac{A[i,:] \times A[j,i]}{A[i,i]}$
    end if
  end for
end for
return solve_linear_equations(A, b)

Działanie programu można wyjaśnić następująco:

1. Funkcja `gaussian_elimination_with_pivoting` wykonuje eliminację Gaussa z częściowym wyborem elementu podstawowego w celu rozwiązania układu równań liniowych.
2. Przyjmuje dwa argumenty: `A`, tablicę numpy reprezentującą macierz współczynników równań liniowych, oraz `b`, tablicę numpy reprezentującą wektor stałych.
3. Funkcja najpierw iteruje po wierszach macierzy współczynników `A`.
4. W każdej iteracji zamienia bieżący wiersz z ostatnim wierszem w celu przeprowadzenia częściowego wyboru elementu podstawowego. Pomaga to uniknąć dzielenia przez zero i problemów z niestabilnością numeryczną.
5. Zmienia również odpowiednie elementy wektora stałego `b` zgodnie z zamianą.
6. Następnie przechodzi do eliminacji współczynników poniżej diagonalnej dla każdej kolumny, zapewniając, że macierz staje się górnotrójkątna.
7. Podczas eliminacji aktualizuje zarówno macierz współczynników `A`, jak i wektor stały `b`.
8. Po zakończeniu procesu eliminacji funkcja używa funkcji `linalg.solve` z biblioteki numpy do rozwiązania układu równań reprezentowanego przez zmodyfikowane `A` i `b`.
9. Na koniec zwraca wektor rozwiązania x uzyskany z procesu rozwiązywania.

2.2 Kod algorytmu w języku Python

```python
def gaussian_elimination_with_pivoting(A: np.ndarray, b: np.ndarray) -> np.ndarray:
    for i in range(A.shape[0]):
        A[[i, -1]] = A[[-1, i]]
        b[i], b[-1] = b[-1], b[i]

        for j in range(A.shape[1]):
            if j > i:
                b[j] -= b[i] * A[j, i] / A[i, i]
                A[j, :] -= A[i, :] * A[j, i] / A[i, i]

    return np.linalg.solve(A, b)
```

Rysunek 8: Kod algorytmu eliminacji Gaussa z pivotingiem

2.3 Test dla macierzy gęstej o losowych wartościach (porównanie z MATLAB)

Sprawdzenie czy algorytm generuje rózne wartości na przekątnej.

```
[[  4.       3.       3.       3.       7.       8.       3.       4.       3.       7.       5.       3.       5.       4.       2.     ]
 [  0.       2.25     2.25    -0.75    -6.75    -7.      -1.75    -1.      -0.75    -5.75     1.75     0.25    -0.25    -3.       1.5    ]
 [  0.       0.       2.       1.3333 -10.      -6.2222   3.4444  -3.8889   4.3333 -11.1111   1.5556   3.2222   1.7778  -3.6667   4.3333]
 [  0.       0.       0.       5.3333  -6.      -3.8889   8.2778  -5.0556  12.8333  -9.4444   0.2222   7.3889   1.1111   0.8333   5.8333]
 [  0.       0.       0.       0.       3.125   -0.6042   2.1146  -0.3854   5.5938  -1.8958  -3.7083  -3.0521  -9.5417   2.8438   0.9063]
 [  0.       0.       0.       0.       0.     -31.3667  25.2833 -13.7167  63.05   -46.6333 -38.7333 -26.7833 -88.0667  17.45    20.55   ]
 [  0.       0.       0.       0.       0.       0.      -9.2257   3.7163  -6.1762   6.684   -1.5643   4.7235  12.6956  -2.604   -6.7394]
 [  0.       0.       0.       0.      -0.       0.       0.      -3.3181  -1.8553   3.1053   0.7947   3.3155  -1.7376  -0.7384  -0.4047]
 [  0.       0.       0.       0.      -0.       0.       0.       0.       1.7652   6.6123  -4.8382   6.8345  -0.3859  -1.0329  -8.451 ]
 [  0.       0.       0.       0.      -0.       0.       0.       0.       0.      28.3552 -27.5825  26.267    0.3845 -10.5802 -33.0322]
 [  0.       0.       0.       0.       0.       0.      -0.       0.       0.       0.      35.3671   6.0673   8.5433  19.0587  -2.8069]
 [  0.       0.       0.       0.       0.       0.      -0.       0.       0.       0.       0.       5.5791   2.9702  11.7551   0.624 ]
 [  0.       0.       0.       0.      -0.       0.      -0.       0.       0.       0.       0.       0.       4.3253  17.7074   7.7382]
 [  0.       0.       0.       0.       0.       0.      -0.       0.       0.       0.       0.       0.       0.      39.5507  19.3541]
 [  0.       0.       0.       0.       0.       0.      -0.       0.      -0.       0.       0.       0.       0.       0.      -8.602 ]]
```

Rysunek 9: Przekształcona macierz górna trójkątna A

Losowanie macierzy gęstej A z wartościami losowymi.

```
Matrix A:
[[5 6 6 3 2 3 2 4 3 3 8 4 6 2 4]
 [4 1 3 5 3 8 8 1 8 1 5 6 7 3 5]
 [4 7 4 5 4 1 3 3 8 4 6 6 3 5 4]
 [7 5 8 3 6 8 6 5 7 2 7 1 1 4 6]
 [8 6 2 3 5 3 4 6 7 8 3 1 8 6 7]
 [8 4 8 3 8 8 1 8 8 5 1 2 5 4 2]
 [8 4 6 3 3 4 2 3 6 6 2 6 7 2 4]
 [6 8 6 4 3 5 8 6 8 7 8 7 5 4 1]
 [1 8 4 7 8 4 4 3 1 5 3 5 2 1 5]
 [6 4 6 6 8 1 7 3 2 2 4 5 4 8 5]
 [2 8 4 4 2 6 2 5 4 2 4 6 8 8 8]
 [3 5 6 8 8 7 1 2 8 6 6 5 7 6 8]
 [3 3 8 8 3 2 2 1 1 5 2 6 3 3 8]
 [2 7 1 5 4 5 2 2 4 5 7 1 7 6 4]
 [4 3 3 3 7 8 3 4 3 7 5 3 5 4 2]]
```

Rysunek 10: Macierz gęsta A

Losowanie wektora b prawej strony z wartościami losowymi.

```
Vector b:
[[6]
 [4]
 [2]
 [1]
 [3]
 [2]
 [6]
 [7]
 [2]
 [3]
 [4]
 [7]
 [6]
 [6]
 [6]]
```

Rysunek 11: Wektor prawej strony b

Rozwiązanie układu równań moim programem.

```
Result x:
[[-0.5959]
 [-0.0755]
 [ 0.9576]
 [-0.4452]
 [-0.0824]
 [-0.0339]
 [ 0.2256]
 [-0.6851]
 [-0.1346]
 [ 0.7566]
 [ 0.239 ]
 [-0.0865]
 [ 0.8089]
 [ 0.1552]
 [-0.2545]]
```

Rysunek 12: Rozwiązanie układu równań x

Rozwiązanie układu równań programem MATLAB.

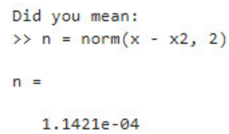
```
>> x = A\b

x =

   -0.5959
   -0.0755
    0.9576
   -0.4452
   -0.0824
   -0.0339
    0.2256
   -0.6851
   -0.1346
    0.7566
    0.2390
   -0.0865
    0.8089
    0.1552
   -0.2545
```

Rysunek 13: Rozwiązanie układu równań x w MATLAB

Porównanie wyników mojego programu i MATLAB norm(x1-x2,2).



Rysunek 14: Wynik norm(x1 - x2, 2) x w MATLAB

Na podstawie wydrukowanych przez programy wyników można zauważyć, że w tym przypadku są one takie same. Potwierdza to wynik 1.1421e-04 w MATLAB, który wskazuje, że różnice między poszczególnymi wynikami w wektorach x1 i x2 są bardzo małe.

3 Algorytm LU faktoryzacji bez pivotingu

3.1 Pseudokod

Algorithm 3 LU factorization without pivoting

Require: A is a square matrix
Require: b is a vector
Ensure: Solution vector x of the linear system $Ax = b$

$L \leftarrow$ identity_matrix(A.shape[0])
for i **in** range(A.shape[0]) **do**
  for j **in** range(A.shape[1]) **do**
    if $j > i$ **then**
      $L[j,i] \leftarrow A[j,i]/A[i,i]$
      $B[j] \leftarrow B[j] - B[i] \times A[j,i]/A[i,i]$
      $A[j,:] \leftarrow A[j,:] - A[i,:] \times A[j,i]/A[i,i]$
    end if
  end for
end for
$U \leftarrow A$.copy()
return L, U

Działanie programu można wyjaśnić następująco:

1. Funkcja `lu_factorization_without_pivoting` przeprowadza faktoryzację LU bez wyboru elementu podstawowego dla danej macierzy kwadratowej A.
2. Przyjmuje dwa argumenty: `A`, tablicę numpy reprezentującą macierz współczynników, oraz `B`, tablicę numpy reprezentującą wektor stałych.
3. Inicjuje dolnotrójkątną macierz `L` jako macierz identycznościową tej samej wielkości co `A`.
4. Funkcja iteruje po wierszach i kolumnach macierzy `A`.

5. Dla każdego elementu `A[j, i]` w `A`, gdzie `j` jest większe niż `i`, oblicza odpowiednią wartość w dolnotrójkątnej macierzy `L` używając wzoru `L[j, i] = A[j, i] / A[i, i]`.
6. Aktualizuje elementy wektora stałych `B`, odejmując `B[i] * A[j, i] / A[i, i]` od `B[j]`.
7. Przeprowadza eliminację Gaussa na wierszach `A` poniżej diagonali, aktualizując odpowiednio `A` i `B`.
8. Po zakończeniu procesu eliminacji funkcja tworzy górnotrójkątną macierz `U` poprzez skopiowanie `A`.
9. Na koniec zwraca krotkę `(L, U)` zawierającą dolnotrójkątną macierz `L` i górnotrójkątną macierz `U`.

3.2 Kod algorytmu w języku Python

```python
def lu_factorization_without_pivoting(A: np.ndarray, B: np.ndarray) -> tuple:
    L = np.eye(A.shape[0])

    for i in range(A.shape[0]):
        for j in range(A.shape[1]):
            if j > i:
                L[j, i] = A[j, i] / A[i, i]
                B[j] -= B[i] * A[j, i] / A[i, i]
                A[j, :] -= A[i, :] * A[j, i] / A[i, i]

    U = A.copy()
    return L, U
```

Rysunek 15: Kod algorytmu LU faktoryzacji bez pivotingu

3.3 Test dla macierzy gęstej o losowych wartościach (porównanie z MATLAB)

Losowanie macierzy gęstej A z wartościami losowymi.

```
Matrix A:
[[2 1 1 8 5 6 6 5 7 7 6 7 3 5 2]
 [7 6 5 4 7 3 4 1 6 7 8 5 7 5 4]
 [3 8 4 7 4 3 5 4 5 7 4 4 7 2 7]
 [4 3 7 3 1 2 2 4 6 7 4 6 2 7 8]
 [2 5 6 7 6 6 4 3 5 8 2 7 4 4 3]
 [1 4 6 2 6 7 7 5 5 8 7 7 8 2 2]
 [4 5 5 1 3 6 6 4 1 1 5 4 7 7 1]
 [1 4 1 6 4 8 6 6 4 6 5 1 5 3 5]
 [1 5 2 2 8 6 1 5 5 6 6 4 4 6 7]
 [8 8 4 3 4 4 1 6 6 8 1 4 3 5 6]
 [7 1 8 8 8 8 5 4 3 5 1 4 2 4 3]
 [8 7 3 4 3 7 3 6 6 3 3 3 4 7 1]
 [8 1 2 7 1 2 6 1 5 2 8 4 7 5 3]
 [8 1 1 6 7 4 3 5 1 4 4 8 6 3 6]
 [1 8 3 5 6 7 4 3 3 8 1 7 2 3 3]]
```

Rysunek 16: Macierz gęsta A

Losowanie wektora b prawej strony z wartościami losowymi.

```
Vector b:
[[6]
 [6]
 [2]
 [5]
 [1]
 [1]
 [8]
 [7]
 [2]
 [3]
 [4]
 [5]
 [3]
 [7]
 [3]]
```

Rysunek 17: Wektor prawej strony b

Wyznaczenie macierzy L moim programem.

```
Lower triangular matrix L
 [[ 1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 3.5     1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 1.5     2.6     1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 2.      0.4    -3.1429  1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 1.      1.6    -1.8571  0.8136  1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.5     1.4    -2.4286  0.9661  1.6577  1.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 2.      1.2    -0.8571  0.3559  0.2148  1.8449  1.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.5     1.4     1.1429 -0.1695  0.1711  1.5069  1.1208  1.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.5     1.8     0.8571 -0.0452  1.4183  0.8608  0.7486  3.6278  1.      0.      0.      0.      0.      0.      0.    ]
 [ 4.      1.6     1.7143 -0.5028 -0.9508  1.4834  1.7251  3.3156  0.746   1.      0.      0.      0.      0.      0.    ]
 [ 3.5    -1.     -4.2857  1.1412  0.4183 -0.1182  0.0366 -0.1606  0.9532  3.2698  1.      0.      0.      0.      0.    ]
 [ 4.      1.2     2.     -0.6441 -1.3691  2.4826  2.4955  2.6079  0.0541 -1.5554 -0.7344  1.      0.      0.      0.    ]
 [ 4.     -1.2     0.1429 -0.3503 -2.0749 -0.085   0.1395 -2.4005 -0.7363 -0.3641 -0.0361 -0.0091  1.      0.      0.    ]
 [ 4.     -1.2     0.8571 -0.5876 -0.9642 -0.1494  0.3543  1.9415  1.9033  5.8562  1.8281 -2.3144  1.9187  1.      0.    ]
 [ 0.5     3.      1.4286 -0.0508  0.9027  1.3055  1.0245  1.2261 -0.1018 -0.2622 -0.1789 -0.5492 -0.908   0.9559  1.    ]]
```

Rysunek 18: Wyznaczona macierz L

Wyznaczenie macierzy U moim programem.

```
Upper triangular matrix U
[[  2.        1.        1.        8.        5.        6.        6.        5.        7.        7.        6.        7.        3.        5.        2.     ]
 [  0.        2.5       1.5     -24.      -10.5     -18.      -17.      -16.5     -18.5     -17.5     -13.      -19.5      -3.5     -12.5      -3.     ]
 [  0.        0.       -1.4      57.4      23.8      40.8      40.2      39.4      42.6      42.       28.8      44.2      11.6      27.       11.8    ]
 [  0.        0.        0.      177.       70.      125.4286 123.1429 124.4286 133.2857 132.       87.7143 138.7143  33.8571  86.8571  42.2857]
 [  0.        0.        0.        0.        5.0508   2.5278  -0.3269  -3.6586  -1.7215  -0.3898  -1.0751   0.4334   0.5981  -1.5206  -6.6877]
 [  0.        0.        0.        0.        0.        2.9185   7.0019   7.14      4.9434   4.1208   9.1841   3.4123   5.8706   1.1793   4.0911]
 [  0.        0.        0.        0.        0.        0.       -7.8213  -9.1038 -10.477  -10.502  -14.6478  -4.4757  -7.8673   2.3784 -10.4481]
 [  0.        0.       -0.       -0.        0.        0.        0.        2.7319   4.8935   7.001    4.9149  -2.403    0.7506  -2.3183   8.5717]
 [  0.        0.        0.        0.        0.        0.        0.        0.       -7.4132 -16.5646  -7.5671  12.4998  -2.3479  14.5542 -14.1142]
 [  0.        0.        0.        0.        0.        0.        0.        0.        0.        3.1513  -7.4943   2.8907  -1.5667  -8.0813  -8.4605]
 [  0.        0.       -0.       -0.        0.        0.        0.        0.        0.        0.       24.9035 -30.2454   7.2877   3.4564  41.4712]
 [  0.        0.        0.       -0.        0.        0.        0.        0.        0.        0.        0.       -9.4932   1.7686 -11.7754   2.7025]
 [  0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        3.6215  -4.592    1.0786]
 [  0.        0.       -0.       -0.        0.        0.        0.        0.        0.        0.        0.        0.        0.       -6.8751  -4.8649]
 [  0.        0.        0.        0.        0.        0.        0.        0.        0.        0.       -0.        0.        0.        0.        8.0641]]
```

Rysunek 19: Wyznaczona macierz U

Wyznaczenie macierzy L, U i P programem MATLAB.

```
>> [L, U, P] =lu(A)

L =

    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0
    0.8750    0.8571    1.0000         0         0         0         0         0         0         0         0         0         0         0         0
    0.1250   -1.0000    0.0805    1.0000         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000   -0.1609   -0.0811    1.0000         0         0         0         0         0         0         0         0         0         0
    0.3750   -0.7143    0.1724    0.9915   -0.3807    1.0000         0         0         0         0         0         0         0         0         0
    0.1250   -0.4286    0.7471    0.2226   -0.2055   -0.2142    1.0000         0         0         0         0         0         0         0         0
    0.8750    0.1429    0.2874    0.0288    0.2524    0.7983    0.6203    1.0000         0         0         0         0         0         0         0
    0.2500    0.1429    0.0460    0.7781    0.3566   -0.1898   -0.2085   -0.6843    1.0000         0         0         0         0         0         0
    1.0000    0.1429   -0.1149    0.0770    0.0127   -0.9862   -0.1308   -0.4431    0.0036    1.0000         0         0         0         0         0
    0.5000    0.1429    0.8506   -0.0859   -0.8802    0.4676    0.0537   -0.1758    0.1449   -0.5839    1.0000         0         0         0         0
    0.1250   -0.4286   -0.0575    0.8799    0.1263   -0.8270   -0.1150   -0.8935    0.3456    0.2615    0.3564    1.0000         0         0         0
    0.2500   -0.4286    0.6667    0.7872   -0.3455    0.5406   -0.1571   -0.0286    0.3406   -0.2234    0.3741   -0.1246    1.0000         0         0
    0.5000   -0.1429    0.4368   -0.0920   -0.3207   -0.6578    0.8472    0.0065   -0.8663    0.7537   -0.1354   -0.0873   -0.9084    1.0000         0
    0.1250   -0.5714    0.0575    0.4486    0.6187   -0.0402    0.1760    0.0362    0.7027    0.6761    0.6975    0.5071    0.4028    0.0244    1.0000


U =

    8.0000    8.0000    4.0000    3.0000    4.0000    4.0000    1.0000    6.0000    6.0000    8.0000    1.0000    4.0000    3.0000    5.0000    6.0000
         0   -7.0000   -2.0000    4.0000   -3.0000   -2.0000    5.0000   -5.0000   -1.0000   -6.0000    7.0000         0    4.0000         0   -3.0000
         0         0    6.2143    1.9464    7.0714    6.2143   -0.1607    3.0357   -1.3929    3.1429   -5.8750    0.5000   -4.0536   -0.3750    0.3214
         0         0         0    8.4684    1.9310    4.0000    8.8879   -2.9943    1.3621    0.7471    8.3477    6.4598    5.9511    2.4052   -0.7759
         0         0         0         0    7.2945    3.3244   -2.3051    4.2457   -4.1137    2.5663   -4.2684    4.6043   -1.1697   -1.8653    2.9888
         0         0         0         0         0   -3.7006   -1.4658    2.2402   -0.6406   -0.5915   -0.2638   -2.2385    3.0851   -2.9051    4.4587
         0         0         0         0         0         0    6.3719    1.8578    3.5764    2.3148   11.4726    5.1551   11.4636    0.1142    1.4660
         0         0         0         0         0         0         0   -8.3339    0.5849   -1.6788    1.7437   -1.4026   -4.4819    3.3825   -6.1146
         0         0         0         0         0         0         0         0    7.1383    3.4378    3.5825   -1.0006   -2.4396    4.3482   -2.5807
         0         0         0         0         0         0         0         0         0   -4.9083    1.7368   -3.6496    2.0847    0.4282   -2.6234
         0         0         0         0         0         0         0         0         0         0    4.7659    6.7193    1.5827    4.9508    3.3228
         0         0         0         0         0         0         0         0         0         0         0   -9.3433    0.6169   -2.2736    2.0741
         0         0         0         0         0         0         0         0         0         0         0         0    3.3639   -1.3732   -1.4028
         0         0         0         0         0         0         0         0         0         0         0         0         0    4.9254   -0.8530
         0         0         0         0         0         0         0         0         0         0         0         0         0         0    3.9627


P =

     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0
     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0
```

Rysunek 20: Wyznaczone macierze L, U i P programem MATLAB

Porównanie otrzymanych macierzy L i U mojego programu i MATLAB.

```
>> n = norm(L - L2, 2)

n =

   11.6661
```

Rysunek 21: Wynik norm(L1 - L2, 2) w MATLAB

```
>> n2 = norm(U - U2, 2)

n2 =

  405.4887
```

Rysunek 22: Wynik norm(U1 - U2, 2) w MATLAB

Porównywane są tylko macierze L i U, ponieważ w części prezentacji o tej wersji algorytmu tylko te macierze były prezentowane.

Norma macierzowa 2-norm (lub norma spektralna) macierzy jest największą wartością osobliwą (wartością własną) macierzy. Wartość 11.6661 oznacza, że odchylenie (różnica) między macierzami L i L2 jest dość znaczące.

Wartość 405.4887 oznacza, że odchylenie (różnica) między macierzami U i U2 jest znaczące, mierząc je w kontekście tej konkretnej normy. Im większa wartość, tym większa różnica między macierzami. Można zatem powiedzieć, że większe róznice występują w macierzach U niż L.

4 Algorytm LU faktoryzacji z pivotingiem

4.1 Pseudokod

Algorithm 4 LU factorization with pivoting

$P \leftarrow$ identity matrix of size A.shape[0]
$L \leftarrow$ identity matrix of size A.shape[0]
$U \leftarrow$ copy of matrix A
for $i \leftarrow 0$ **to** A.shape$[0] - 1$ **do**
    max_element_row_index $\leftarrow$ index of maximum element in $[i, A$.shape$[0])$ using |U[k, i]|
    if $i \neq$ max_element_row_index **then**
        Swap rows i and max_element_row_index in P
        Swap rows i and max_element_row_index in the first i columns of L
        Swap rows i and max_element_row_index in U
    end if
    for $j \leftarrow i + 1$ **to** A.shape$[0] - 1$ **do**
        $L[j, i] \leftarrow U[j, i]/U[i, i]$
        $U[j, i:] \leftarrow U[j, i:] - (U[j, i]/U[i, i]) \times U[i, i:]$
    end for
end for
return L, U, P

Działanie programu można wyjaśnić następująco:

1. Definicja funkcji: `lu_factorization_with_pivoting(A: np.ndarray) -> tuple`. Funkcja przyjmuje macierz numpy `A` i zwraca trójkę zawierającą macierze L, U i P.
2. Inicjalizacja macierzy pomocniczych:
   - `P = np.eye(A.shape[0])`: Inicjalizuje macierz permutacji jako jednostkową macierz o rozmiarze odpowiadającym liczbie wierszy macierzy A.
   - `L = np.eye(A.shape[0])`: Inicjalizuje dolną trójkątną macierz L jako jednostkową macierz o rozmiarze odpowiadającym liczbie wierszy macierzy A.
   - `U = A.copy()`: Tworzy kopię macierzy A, która będzie górnotrójkątną macierzą U.
3. Pętla główna:
   - Iteruje przez wiersze macierzy A używając zmiennej `i` w zakresie od 0 do liczby wierszy minus 1.
   - Znajduje wiersz z maksymalną wartością bezwzględną w kolumnie `i` od `i` do ostatniego wiersza.
   - Jeśli wiersz z maksymalną wartością bezwzględną nie jest aktualnie badanym wierszem `i`, dokonuje zamiany wierszy w macierzach P, L i U.
   - Wewnętrzna pętla:
     - Iteruje przez wiersze poniżej `i` (oznaczane przez `j` od `i+1` do końca).
     - Oblicza wartości współczynników do aktualizacji macierzy L i U.
     - Aktualizuje wartości w macierzy L i U.
4. Zwrócenie wyników:
   - Zwraca trójkę zawierającą macierze L, U i P.

4.2 Kod algorytmu w języku Python

```python
def lu_factorization_with_pivoting(A: np.ndarray) -> tuple:
    P = np.eye(A.shape[0])
    L = np.eye(A.shape[0])
    U = A.copy()

    for i in range(A.shape[0]):
        max_element_row_index = max(range(i, A.shape[0]), key=lambda k: abs(U[k, i]))
        if i != max_element_row_index:
            P[[i, max_element_row_index]] = P[[max_element_row_index, i]]
            L[[i, max_element_row_index], :i] = L[[max_element_row_index, i], :i]
            U[[i, max_element_row_index]] = U[[max_element_row_index, i]]

        for j in range(i + 1, A.shape[0]):
            L[j, i] = U[j, i] / U[i, i]
            U[j, i:] -= U[j, i] / U[i, i] * U[i, i:]

    return L, U, P
```

Rysunek 23: Kod algorytmu LU faktoryzacji z pivotingiem

4.3 Test dla macierzy gęstej o losowych wartościach (porównanie z MATLAB)

Losowanie macierzy gęstej A z wartościami losowymi.

```
Matrix A:
[[7 6 6 5 1 8 8 5 7 6 4 5 1 8 6]
 [1 4 2 6 4 7 2 4 1 7 4 7 5 4 5]
 [3 7 2 4 7 5 1 8 1 8 1 8 1 8 1]
 [6 4 1 8 5 3 8 7 1 5 3 4 1 2 4]
 [6 5 1 4 6 7 2 6 4 7 7 2 8 8 2]
 [8 8 8 7 8 1 6 6 7 6 4 2 4 3 3]
 [8 7 8 4 2 3 4 5 7 1 4 3 5 2 1]
 [5 5 4 6 4 1 3 3 7 3 4 5 2 1 5]
 [7 5 1 6 4 1 8 5 2 4 5 7 5 4 5]
 [3 3 1 5 7 3 5 8 2 8 1 1 1 5 2]
 [8 1 1 6 7 4 1 2 5 5 3 5 2 8 1]
 [3 4 4 2 1 3 1 2 4 6 7 7 4 4 5]
 [3 5 8 3 5 1 8 8 7 7 3 6 1 6 1]
 [2 3 5 6 7 6 7 3 6 7 8 7 1 6 2]
 [3 5 5 6 4 8 5 2 7 8 6 4 6 4 2]]
```

Rysunek 24: Macierz gęsta A

Wyznaczenie macierzy L moim programem.

```
L:
[[ 1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 1.      1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.375  -0.5714  1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.25   -0.1429 -0.4     1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.75    0.2857  0.6     0.5766  1.      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.875   0.1429 -0.     -0.2218  0.7034  1.      0.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.125  -0.4286  0.4     0.9879  0.7785  0.2023  1.      0.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.375  -0.2857 -0.6     0.129  -0.4664  0.05   -0.5992  1.      0.      0.      0.      0.      0.      0.      0.    ]
 [ 0.625  -0.      0.2     0.3306  0.5886  0.0435  0.6318 -0.165   1.      0.      0.      0.      0.      0.      0.    ]
 [ 1.      0.1429 -0.2    -0.6089  0.2167  0.704   0.2326  0.3445 -0.1238  1.      0.      0.      0.      0.      0.    ]
 [ 0.375  -0.1429 -0.     -0.1734  0.167   0.419   0.3632  0.03    0.1246 -0.7045  1.      0.      0.      0.      0.    ]
 [ 0.75    0.1429  0.8    -0.3952  0.0218  0.3143 -0.3774 -0.1857  0.1855  0.0013  0.6353  1.      0.      0.      0.    ]
 [ 0.875   0.2857  0.8    -0.1089  0.7511  0.0527 -0.534  -0.6298  0.2168 -0.5908  0.5379 -0.4104  1.      0.      0.    ]
 [ 0.375  -0.2857 -0.     0.6976  0.5706  0.4795  0.6178 -0.39    0.4353 -0.4624  0.0024  0.2464  0.1694  1.      0.    ]
 [ 0.375  -0.      0.4     0.4637  0.0408 -0.2632 -0.4105  0.4913  0.1079 -0.2126 -0.0902  0.6473 -0.3694  0.1729  1.    ]]
```

Rysunek 25: Wyznaczona macierz L

Wyznaczenie macierzy U moim programem.

```
U:
[[  8.       8.       8.       7.       8.       1.       6.       6.       7.       6.       4.       2.       4.       3.       3.    ]
 [  0.      -7.      -7.      -1.      -1.       3.      -5.      -4.      -2.      -1.      -1.       3.      -2.       5.      -2.    ]
 [  0.       0.      -5.       0.8036   3.4286   6.3393  -4.1071   3.4643  -2.7679   5.1786  -1.0714   8.9643  -1.6429   9.7321  -1.2679]
 [  0.       0.       0.       4.4286   6.2286   8.7143   3.1429   2.3143   2.8571   7.4286   6.4286  10.5143  -0.9429   9.8571   0.4571]
 [  0.       0.       0.       0.      -6.3629  -7.4355   5.5806   0.2298  -3.6653  -6.6048  -2.7782  -9.7984   0.1008 -13.2016   2.8185]
 [  0.       0.       0.       0.       0.      13.8593   0.2357   0.673    4.3726   7.1863   4.0228  12.0456  -2.4943  16.1331   1.7795]
 [  0.       0.       0.       0.       0.       0.      -6.7468  -2.4514  -0.4791   0.0988  -1.5021  -0.7468   5.6577  -0.8504   1.2692]
 [  0.       0.       0.      -0.       0.       0.       0.       4.9918  -0.4411   3.232   -2.6551   4.5089   1.6264   3.397    0.47  ]
 [  0.       0.       0.       0.      -0.       0.       0.       0.       4.4311  -0.1957   1.56     4.9404  -3.1168   2.0866   0.7668]
 [  0.       0.       0.       0.       0.       0.       0.       0.       0.      -4.0867   3.0702   1.6418  -0.1449  -2.9765  -3.9152]
 [  0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       7.8443   5.7686   1.2617  -1.4057  -0.8768]
 [  0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.     -10.5449   9.2232  -2.8176   1.596 ]
 [  0.       0.       0.       0.       0.       0.      -0.       0.       0.       0.       0.       0.      11.081    1.3735   1.4207]
 [  0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.      -2.7454  -5.8537]
 [  0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       1.3235]]
```

Rysunek 26: Wyznaczona macierz U

Wyznaczenie macierzy P moim programem.

```
P:
[[ 0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.]]
```

Rysunek 27: Wyznaczona macierz P

Wyznaczenie macierzy L, U i P programem MATLAB.

```
>> [L, U, P] =lu(A)

L =

    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0
    0.3750   -0.5714    1.0000         0         0         0         0         0         0         0         0         0         0         0         0
    0.2500   -0.1429   -0.4000    1.0000         0         0         0         0         0         0         0         0         0         0         0
    0.7500    0.2857    0.6000    0.5766    1.0000         0         0         0         0         0         0         0         0         0         0
    0.8750    0.1429         0   -0.2218    0.7034    1.0000         0         0         0         0         0         0         0         0         0
    0.1250   -0.4286    0.4000    0.9879    0.7785    0.2023    1.0000         0         0         0         0         0         0         0         0
    0.3750   -0.2857   -0.6000    0.1290   -0.4664    0.0500   -0.5992    1.0000         0         0         0         0         0         0         0
    0.6250         0    0.2000    0.3306    0.5886    0.0435    0.6318   -0.1650    1.0000         0         0         0         0         0         0
    1.0000    0.1429   -0.2000   -0.6089    0.2167    0.7040    0.2326    0.3445   -0.1238    1.0000         0         0         0         0         0
    0.3750   -0.1429         0   -0.1734    0.1670    0.4190    0.3632    0.0300    0.1246   -0.7045    1.0000         0         0         0         0
    0.7500    0.1429    0.8000   -0.3952    0.0218    0.3143   -0.3774   -0.1857    0.1855    0.0013    0.6353    1.0000         0         0         0
    0.8750    0.2857    0.8000   -0.1089    0.7511    0.0527   -0.5340   -0.6298    0.2168   -0.5908    0.5379   -0.4104    1.0000         0         0
    0.3750   -0.2857         0    0.6976    0.5706    0.4795    0.6178   -0.3900    0.4353   -0.4624    0.0024    0.2464    0.1694    1.0000         0
    0.3750         0    0.4000    0.4637    0.0408   -0.2632   -0.4105    0.4913    0.1079   -0.2126   -0.0902    0.6473   -0.3694    0.1729    1.0000


U =

    8.0000    8.0000    8.0000    7.0000    8.0000    1.0000    6.0000    6.0000    7.0000    6.0000    4.0000    2.0000    4.0000    3.0000    3.0000
         0   -7.0000   -7.0000   -1.0000   -1.0000    3.0000   -5.0000   -4.0000   -2.0000   -1.0000   -1.0000    3.0000   -2.0000    5.0000   -2.0000
         0         0   -5.0000    0.8036    3.4286    6.3393   -4.1071    3.4643   -2.7679    5.1786   -1.0714    8.9643   -1.6429    9.7321   -1.2679
         0         0         0    4.4286    6.2286    8.7143    3.1429    2.3143    2.8571    7.4286    6.4286   10.5143   -0.9429    9.8571    0.4571
         0         0         0         0   -6.3629   -7.4355    5.5806    0.2298   -3.6653   -6.6048   -2.7782   -9.7984    0.1008  -13.2016    2.8185
         0         0         0         0         0   13.8593    0.2357    0.6730    4.3726    7.1863    4.0228   12.0456   -2.4943   16.1331    1.7795
         0         0         0         0         0         0   -6.7468   -2.4514   -0.4791    0.0988   -1.5021   -0.7468    5.6577   -0.8504    1.2692
         0         0         0         0         0         0         0    4.9918   -0.4411    3.2320   -2.6551    4.5089    1.6264    3.3970    0.4700
         0         0         0         0         0         0         0         0    4.4311   -0.1957    1.5600    4.9404   -3.1168    2.0866    0.7668
         0         0         0         0         0         0         0         0         0   -4.0867    3.0702    1.6418   -0.1449   -2.9765   -3.9152
         0         0         0         0         0         0         0         0         0         0    7.8443    5.7686    1.2617   -1.4057   -0.8768
         0         0         0         0         0         0         0         0         0         0         0  -10.5449    9.2232   -2.8176    1.5960
         0         0         0         0         0         0         0         0         0         0         0         0   11.0810    1.3735    1.4207
         0         0         0         0         0         0         0         0         0         0         0         0         0   -2.7454   -5.8537
         0         0         0         0         0         0         0         0         0         0         0         0         0         0    1.3235


P =

     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0
```

Rysunek 28: Wyznaczone macierze L, U i P programem MATLAB

Porównanie otrzymanych macierzy L, U i Pmojego programu oraz MATLAB.

```
>> n1 = norm(L - L2, 2)

n1 =

   1.4566e-04
```

Rysunek 29: Wynik norm(L1 - L2, 2) w MATLAB

```
>> n1 = norm(U - U2, 2)

n1 =

   1.7845e-04
```

Rysunek 30: Wynik norm(U1 - U2, 2) w MATLAB

```
>> n3 = norm(P - P2, 2)

n3 =

     0
```

Rysunek 31: Wynik norm(P1 - P2, 2) w MATLAB

Norma macierzowa 2-norm (lub norma spektralna) macierzy jest największą wartością osobliwą (wartością własną) macierzy. Wartość 1.4566e-04 oznacza, że odchylenie (różnica) między macierzami L i L2 jest bardzo mała.

Wartość 1.7845e-04 oznacza, że odchylenie (różnica) między macierzami U i U2 jest bardzo mała.

Wartość 0 oznacza, że nia ma odchylenia (różnica) między macierzami P i P2.

5 Dlaczego niektóre wyniki się zgadzają z MATLABem a niektóre nie?

Jedyne otrzymane wyniki, które zauważalnie nie zgadzały się z wynikami MATLAB były dla algorytmu LU faktoryzacji bez pivotingu. Uważam, że przyczyny wspominanych różnic wynikaja z:

- mogą występować różnice w implementacji operacji arytmetycznych, co może prowadzić do różnic w precyzji obliczeń;

- mimo, że algorytmy są podobne, drobne różnice w implementacji mogą prowadzić do różnic w wynikach. To może wynikać z różnic w implementacji operacji arytmetycznych czy optymalizacji kodu;

- różnice w wynikach mogą się również pojawić ze względu na różnice w sposobie zaokrąglania liczb i obsługi punktów stałoprzecinkowych. W MATLAB zauważyłem, że liczby były zaokrąglane do 4 miejsc po przecinku, a w Python poza zaokrągleniem wyniku nie miało to miejsca,

- sposób wczytywania i przetwarzania danych może się różnić między Python a MATLAB, co może wpływać na wyniki.

6 Czy da się wylosować taką macierz żeby wszystkie wyniki się zgadzały?

Uważam, że przy stworzeniu odpowiednich warunków jest możliwe uzyskanie zgodnych wyników. Przykładowo można:

- wygenerować macierz i wektor tak, aby spełniały określone warunki, które sprawią, że wyniki działania algorytmów będą takie same.
- z ustawić konkretne wartości macierzy i wektora prawych stron w taki sposób, aby były one identyczne dla obu algorytmów.
- wybrać proste dane testowe, dla których wyniki algorytmów są dobrze znane i można je łatwo zweryfikować.