

Imię i nazwisko: Anastasiya Yahorava, Marcin Wardyński

Grupa: Piątek, 15:00

Sprawozdanie do programu eliminacji Gaussa i LU faktoryzacji

Dane wejściowe:

Suma dnia i miesiąca dla każdego z nas: Anastasiya = 5, Marcin = 27.

Ponieważ rozmiar macierzy 5 wydał się nam bardzo mały, przygotowaliśmy sprawozdanie dla różnicy liczb wyliczonej dla każdego z nas, czyli: 27 - 5 = 22

Użyta w zadaniach macierz A:

Dla przejrzystości zamieszczanych rezultatów, na potrzeby sprawozdania zmniejszamy precyzję użytych liczb do dwóch miejsc po przecinku. Chcemy jednak zwrócić uwagę, że obliczenia dokonane zostały na liczbach o maksymalnej precyzji, a jej zmniejszenie nastąpiło już po wszystkich wyliczenia podczas listowania rezultatów w konsoli.

-3.59,	-2.30,	-4.12,	-4.61,	-3.59,	-4.74,	-2.65,	-4.96,	1.94,	-4.57,	-0.70,	1.05,	3.29,	1.28,	0.46,	-3.86,	-0.17,	-2.89,	1.89,	-2.08,	3.11,	2.57,
0.81,	4.40,	-4.49,	-0.29,	-0.51,	3.52,	2.97,	2.56,	-0.29,	-0.40,	-3.80,	3.64,	3.49,	-4.53,	-3.00,	4.89,	4.89,	-4.35,	2.32,	2.08,	-0.75,	1.22,
-4.90,	-2.86,	2.31,	4.27,	-3.48,	-3.94,	0.05,	2.22,	2.09,	3.45,	-0.98,	-2.21,	3.94,	-3.94,	4.21,	2.03,	1.85,	3.86,	-3.88,	2.23,	1.19,	3.13,
0.87,	-1.77,	-1.47,	-3.37,	-1.08,	3.43,	2.70,	4.98,	1.34,	4.81,	1.82,	1.52,	4.02,	-1.03,	-2.85,	-2.79,	-4.52,	-4.98,	-4.83,	1.52,	-2.75,	1.56,
3.36,	-4.35,	-2.82,	-3.52,	-0.06,	4.89,	-1.72,	4.74,	1.78,	2.07,	-1.47,	2.14,	-0.59,	2.92,	-4.04,	-4.30,	-4.10,	-4.31,	-0.63,	1.53,	2.65,	3.79,
-1.09,	4.13,	2.72,	-1.41,	-4.56,	4.79,	2.61,	-2.50,	1.28,	-0.95,	-2.29,	2.97,	-1.05,	-4.09,	0.39,	0.91,	3.17,	-2.97,	-1.31,	4.47,	-0.24,	-1.55,
-0.80,	3.25,	-1.37,	-3.34,	1.31,	0.03,	0.37,	-0.13,	3.36,	4.08,	-2.06,	2.33,	1.82,	-3.39,	-1.46,	2.81,	-2.57,	4.30,	-2.31,	-1.46,	-4.43,	3.59,
-2.01,	2.48,	-4.34,	1.43,	-2.09,	1.75,	0.31,	0.63,	2.02,	-1.86,	-2.89,	-3.41,	3.58,	2.02,	-0.96,	1.12,	-1.11,	-2.82,	2.84,	-4.41,	-3.01,	3.43,
1.66,	1.83,	-1.08,	1.64,	2.07,	2.42,	2.02,	4.79,	-2.82,	-1.56,	1.80,	4.56,	-0.76,	3.29,	-4.00,	1.80,	0.47,	-4.04,	4.98,	1.20,	-0.03,	0.66,
4.06,	-2.16,	2.16,	-0.43,	-3.32,	0.78,	4.76,	-3.65,	2.07,	-4.54,	1.65,	3.86,	2.81,	1.44,	-4.38,	2.01,	-3.97,	0.49,	-0.46,	-1.21,	-0.43,	-3.21,
-1.94,	1.68,	0.93,	-4.61,	4.41,	-2.26,	3.67,	4.70,	0.52,	-1.06,	1.27,	2.87,	-4.85,	0.93,	-3.18,	1.58,	-4.75,	1.02,	3.31,	-2.42,	0.65,	-0.08,
3.72,	-0.36,	-2.63,	-1.69,	3.14,	-3.55,	4.27,	-3.58,	4.54,	-0.58,	-1.45,	1.81,	-3.79,	-2.96,	3.55,	3.17,	-2.02,	-4.65,	-1.31,	-0.44,	-0.29,	1.56,
4.90,	-2.40,	4.02,	1.60,	-0.55,	0.79,	1.12,	-0.47,	0.34,	1.20,	3.48,	1.10,	-0.77,	-0.32,	4.85,	2.22,	2.18,	3.61,	3.29,	5.00,	-4.93,	-2.45,
-3.78,	4.45,	3.02,	2.68,	2.66,	-0.06,	4.42,	1.18,	0.32,	0.97,	-2.15,	-1.88,	-3.35,	2.17,	-4.35,	-4.17,	-3.82,	-1.60,	-2.51,	3.97,	-4.28,	-1.94,
-1.46,	3.34,	4.96,	-4.95,	-2.73,	0.91,	2.14,	2.97,	-0.04,	1.02,	-0.58,	-3.36,	-1.45,	-4.58,	-2.80,	-0.85,	-4.68,	1.84,	4.47,	-0.10,	1.62,	2.05,
-2.62,	3.06,	-4.65,	2.12,	-4.44,	1.45,	-1.76,	4.21,	-3.59,	-0.19,	4.72,	-1.35,	1.04,	1.99,	1.76,	-1.17,	2.46,	-2.30,	1.83,	1.26,	-4.78,	-4.28,
3.47,	-1.34,	0.79,	3.59,	-1.02,	4.71,	3.74,	4.93,	-1.62,	3.37,	2.70,	-4.82,	1.63,	-3.34,	-4.55,	2.26,	2.65,	1.98,	3.31,	-1.60,	-2.39,	4.06,
2.55,	2.92,	0.85,	-3.90,	-4.98,	-3.71,	4.14,	-1.96,	-3.48,	-1.08,	3.01,	-0.80,	0.68,	-1.06,	-3.74,	2.20,	3.75,	-0.96,	-3.77,	-2.04,	-3.57,	-2.23,
1.91,	4.62,	3.95,	-2.51,	3.46,	3.49,	-1.94,	3.60,	-4.15,	-1.93,	2.25,	-4.75,	0.85,	3.87,	-1.73,	-2.09,	2.81,	1.79,	3.67,	-2.14,	2.85,	-1.51,
-3.76,	1.61,	2.08,	-0.47,	-4.76,	4.06,	3.73,	2.68,	0.28,	1.84,	0.29,	4.40,	1.86,	4.85,	-2.54,	1.43,	3.88,	-3.92,	-0.60,	3.57,	-1.72,	1.86,
-1.03,	-3.10,	-2.73,	1.02,	-0.18,	-4.55,	3.21,	3.03,	-1.23,	4.16,	4.31,	0.87,	1.18,	-0.57,	-2.29,	-1.76,	-2.91,	4.49,	-0.58,	3.91,	2.88,	-2.70,
-0.02,	-3.56,	-0.29,	0.28,	-2.24,	1.10,	1.53,	3.35,	-4.35,	1.30,	1.81,	0.29,	-2.08,	-4.17,	-0.99,	-0.29,	1.72,	-3.64,	-2.14,	-3.60,	-1.40,	4.28,

Wektor b:

1.03,	3.58,	-3.11,	3.99,	-3.31,	-2.07,	0.32,	3.94,	1.13,	2.36,	2.59,	2.42,	-1.34,	-2.81,	3.45,	0.90,	2.42,	2.77,	4.27,	3.47,	1.00,	4.80,
-------	-------	--------	-------	--------	--------	-------	-------	-------	-------	-------	-------	--------	--------	-------	-------	-------	-------	-------	-------	-------	-------

Repozytorium kodu dla każdego z zadań:

<https://github.com/mwardynski/matrix-calculus/tree/prog2/prog2>

1. Eliminacja Gaussa

1.1. Pseudokod programu

```
GaussianElimination(){  
  for (i = 0; i<matrix.length;i++){  
    scale = 1 /matrix[i][i];  
    for (j = i; j < matrix.length; j++){  
      matrix[i][j] = matrix[i][j] * scale;  
    }  
    vector[i] = vector[i]*scale;  
    if |matrix[i][i]-1| < Epsilon{  
      matrix[i][i] = 1;  
    }  
    gaussianEliminationStep(i)  
  }  
  result = solveEquation()  
  return result  
}
```

Funkcje pomocnicze: gaussianEliminationStep - służy do modyfikacji kolumn solveEquation - postępowanie odwrotne Gaussa

1.2. Kod programu

<https://github.com/mwardynski/matrix-calculus/blob/prog2/prog2/gaussianElimination.js>

1.3. Wyniki i ich analiza

Czas wykonania: 0.5860999999999876 ms Wynik norm: 7.034157316721072e-11

Eliminacja Gaussa tworząca jedynki po przekątnej

```
1.00, 0.64, 1.15, 1.28, 1.00, 1.32, 0.74, 1.38, -0.54, 1.27, 0.19, -0.29, -0.92, -0.36, -0.13, 1.08, 0.05, 0.81, -0.53, 0.58, -0.87, -0.72,  
0.00, 1.00, -1.40, -0.34, -0.34, 0.63, 0.61, 0.37, 0.04, -0.37, -1.02, 1.00, 1.09, -1.09, -0.75, 1.04, 1.25, -1.29, 0.71, 0.41, -0.01, 0.46,  
0.00, 0.00, 1.00, 1.28, 0.18, 0.28, 0.42, 1.07, -0.07, 1.18, 0.03, -0.47, -0.10, -0.65, 0.45, 0.84, 0.21, 0.98, -0.80, 0.59, -0.37, -0.06,  
0.00, 0.00, 0.00, 1.00, -0.83, 2.62, 2.88, 5.25, 0.74, 4.68, -0.27, 0.69, 3.31, -3.41, -0.91, 1.72, -0.22, -1.50, -3.57, 2.64, -2.02, 1.43,  
0.00, 0.00, 0.00, 0.00, 1.00, -3.11, -4.02, -5.99, -0.83, -5.92, -0.98, -0.84, -4.53, 3.78, 1.41, -0.94, 1.66, 2.70, 5.18, -2.66, 3.56, -1.10,  
0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 2.24, 3.02, 0.12, 3.21, 1.28, 0.04, 2.59, -1.68, -0.67, 0.38, -1.37, -1.15, -2.86, 0.74, -2.48, 0.17,  
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.57, -0.56, 0.40, 1.20, -0.27, 0.97, -0.15, -0.12, -0.32, -0.38, -1.13, -0.72, -0.03, -0.92, -0.81,  
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.50, 1.31, -1.50, 1.86, 0.28, -2.47, -0.08, 1.71, 1.73, 1.23, -0.56, 2.03, 1.13, 1.13,  
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 2.03, -1.30, -1.24, 0.86, -2.49, 0.86, 0.95, 0.32, 1.86, -2.21, 0.12, -0.72, 0.22,  
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, -0.64, -0.67, 0.31, -1.83, 0.80, 0.33, 1.11, 0.39, -1.22, 0.71, 0.12, -0.43,
```

```

0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.38, 0.43, 1.33, -0.51, -0.52, -1.27, -0.25, 0.16, -0.60, -0.95, 0.43,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, -0.95, -0.15, 0.70, -0.25, 0.70, -1.49, 0.18, 1.29, 0.72, 0.55,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.31, -1.41, 0.25, -0.97, 1.30, -0.54, -1.13, -0.19, -0.12,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.30, -1.86, 0.70, -0.65, 1.86, 2.89, -2.22, -0.24,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, -0.29, 1.36, 0.17, 1.21, 1.72, -1.46, -0.61,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 27.61, 73.18, -2.31, 38.07, -57.80, -73.78,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 2.40, 0.10, 1.34, -1.96, -2.32,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, -0.58, 0.39, -0.63, -1.48,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.12, -0.38, 0.50,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 3.81, -1.36,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.12,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00,

```

Wektor

```

-0.29, 0.98, -0.58, 1.58, -3.72, 2.99, 2.12, -2.13, -0.06, 0.09, 0.42, -1.06, 0.42, -1.04, -0.39, -32.66, -1.12, -0.32, 0.14, 2.77, 0.20,
-75.19,

```

2. Eliminacja Gaussa z pivotingiem

2.1. Pseudokod programu

```

gaussianEliminationWithPivoting(){
  for (colIndex = 0; colIndex < matrix.length; colIndex++){
    maxRowIndex = findMaxAbsColIndex(colIndex);
    if |matrix[maxRowIndex][colIndex]| !=0{
      swapRows(colIndex,maxColIndex)
      gaussianEliminationStep(colIndex);
    }
    else{
      return null;
    }
  }
  result = solveEquation();
  return result
}

```

Funkcje pomocnicze: findMaxAbsColIndex - poszukiwanie elementu maksymalnego co do wartości bezwzględnej swapRows - zamiana wierszy miejscami gaussianEliminationStep - służy do modyfikacji kolumn, ponieważ po zamianie wartości kolumn wykonuje się krok eliminacji Gaussa solveEquation - postępowanie odwrotne Gaussa

Funkcja gaussianEliminationStep jest częścią algorytmu eliminacji Gaussa

```
gaussianEliminationStep(colIndex){
  for (rowIndex = colIndex+1; rowIndex < matrix.length; rowIndex++){
    factor = matrix[rowIndex][ColIndex]/matrix[colIndex][colIndex];

    for ( j = colIndex; j < n; j++) {
      matrix[rowIndex][j] -= factor * matrix[colIndex][j];
    }
    vector[rowIndex] -= factor * vector[colIndex];
  }
}
```

2.2. Kod programu

<https://github.com/mwardynski/matrix-calculus/blob/prog2/prog2/gaussianElimination.js>

2.3. Wyniki i ich analiza

Czas wykonania 0.5423999999999296 ms Wynik norm: 3.177124443021102e-11 Macierz po pivotingu i krokach eliminacji Gaussa

```
-4.90, -2.86, 2.31, 4.27, -3.48, -3.94, 0.05, 2.22, 2.09, 3.45, -0.98, -2.21, 3.94, -3.94, 4.21, 2.03, 1.85, 3.86, -3.88, 2.23, 1.19, 3.13,
0.00, 6.65, 1.24, -0.61, 5.35, 2.98, 4.38, -0.53, -1.30, -1.69, -1.40, -0.18, -6.38, 5.21, -7.59, -5.74, -5.25, -4.57, 0.48, 2.25, -5.20, -4.36,
0.00, 0.00, 7.31, 5.38, 0.20, -0.80, 4.64, 1.32, 1.41, 3.31, 1.40, -1.25, -1.88, -0.15, 3.05, -0.29, -0.12, 3.85, -0.21, 9.01, -7.85, -2.76,
0.00, 0.00, 0.00, -8.41, -5.16, 0.59, -2.85, 2.01, -0.52, -0.53, -0.08, -1.99, 2.29, -6.61, -0.73, 2.29, -1.87, 1.73, 5.42, -6.49, 8.29, 5.18,
0.00, 0.00, 0.00, 0.00, -9.29, 1.13, -2.30, 5.85, -2.84, 1.86, 7.44, -2.43, 3.03, -3.73, 7.12, 2.87, 3.82, 3.42, 6.74, 2.78, -3.92, -2.28,
0.00, 0.00, 0.00, 0.00, 0.00, -7.17, 4.61, -5.62, -0.37, -0.79, -2.50, 0.71, 1.75, 0.57, -5.19, 1.86, 3.95, -1.79, -12.20, -3.22, -0.11, 0.77,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 7.40, 2.84, 0.85, 5.18, 1.30, 1.46, 3.16, 1.55, -5.70, -3.61, -2.77, -6.35, -12.06, 4.10, -8.06, -1.06,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -10.51, 3.00, -4.39, -7.48, 5.75, -0.26, -2.65, 0.27, 4.11, 7.34, -2.26, -6.97, -1.61, 7.32, 1.74,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -6.35, -3.32, 0.33, -1.11, -6.80, -2.12, -2.48, -0.16, 2.31, -1.98, 3.00, -6.47, 0.75, 2.60,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, -6.55, 1.03, -1.14, -0.38, 3.83, -5.62, -0.94, -9.18, 2.98, 8.18, -5.91, 3.61, -2.03,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -8.12, 2.11, 1.03, -11.94, 2.15, 9.34, 9.61, 1.31, 4.16, -1.34, 7.23, 9.14,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 8.31, -4.50, -6.21, 10.61, 10.11, 14.62, 1.13, 1.78, 4.28, -0.12, 0.37,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -11.65, -9.18, 11.93, 3.60, 3.88, -3.37, 6.38, 0.49, 0.30, 4.10,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 13.38, -10.02, -8.89, -3.54, -1.72, -9.43, 3.55, -2.50, -4.83,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, -5.84, 0.13, -7.10, 6.22, 1.30, -6.19, -4.22, 3.17,
0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 6.28, -2.23, 4.67, 11.37, -7.54, -1.01, 8.20,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 16.19, -2.34, -13.15, 6.88, 8.46, -6.89,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -5.70, -0.41, -1.29, 6.58, 4.25,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 14.70, 0.39, -3.61, 10.13,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, -6.28, -3.48, 10.70,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 17.42, 2.07,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.07,
```

-3.11, -0.41, -4.78, 6.93, 2.74, -1.79, 0.48, -5.21, 4.16, 4.39, 2.23, -4.30, 1.78, -11.70, 2.29, 13.45, -21.33, 0.74, 6.86, 6.70, 2.51, -4.95,

[illegible]

```

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      0      0

Columns 19 through 23
0      0      0      0      61.2375
0      0      0      0      43.6278
0      0      0      0      48.6434
0      0      0      0      60.3056
0      0      0      0      -43.4204
0      0      0      0      -64.4211
0      0      0      0      9.6630
0      0      0      0      46.9962
0      0      0      0      90.0302
0      0      0      0      103.0871
0      0      0      0      -51.6678
0      0      0      0      89.2561
0      0      0      0      5.8722
0      0      0      0      -15.2275
0      0      0      0      6.6994
0      0      0      0      -127.3849
0      0      0      0      63.2447
0      0      0      0      -19.5860
1.0000      0      0      0      58.0483
0      1.0000      0      0      -134.1442
0      0      1.0000      0      9.0815
0      0      0      1.0000      -75.1890

```

3. LU faktoryzacja

3.1. Pseudokod programu

```

lu_decompose(A, b) {
    n = A.length
    L = construct_L(n)
    U = copy_mx(A)

    for (let row = 1; row < n; row++) {
        for (let i = 0; i < row; i++) {
            let multiplier = U[row][i] / U[i][i]
            L[row][i] = multiplier
            for (let j = 0; j < n; j++) {
                U[row][j] = U[row][j] - multiplier * U[i][j]
            }
            b[row] = b[row] - multiplier * b[i]
        }
    }

    return L, U;
}

```

Program wykorzystuje dwie funkcje pomocnicze:

- `copy_mx` - kopiuje zadaną macierz
- `construct_L` - tworzy macierz L o rozmiarze identycznym do danej macierzy A, a następnie uzupełnia wszystkie jej pola zerami, a po przekątnej wstawia jedyinki.

3.2. Kod programu

<https://github.com/mwardynski/matrix-calculus/blob/prog2/prog2/lu.js>

3.3. Wyniki i ich analiza

Po wywołaniu funkcji `lu_decompose(A, b)`, otrzymamy następujące macierze:

L:

```
1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.23, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
1.36, 0.07, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.24, -0.60, -0.69, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.94, -1.68, -1.90, 4.94, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.30, 1.24, 1.29, -5.89, -2.45, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.22, 0.97, 0.58, -3.51, -0.61, -0.49, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.56, 0.97, 0.39, 0.57, 0.28, -0.27, 0.63, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.46, 0.20, -0.23, 1.09, 0.51, -0.30, 0.14, -0.08, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-1.13, -1.23, -1.10, 2.17, -0.65, 1.98, -3.44, -1.60, -5.32, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.54, 0.75, 0.87, -5.07, -0.46, -1.40, 1.72, 0.22, 1.74, 0.31, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-1.04, -0.71, -1.29, 3.09, 1.01, 0.23, -2.07, -0.53, -5.92, 1.08, 0.50, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-1.36, -1.43, -1.12, 2.62, -0.21, 1.55, -2.65, -2.16, -3.24, 0.49, -0.63, -1.07, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
1.05, 1.77, 2.04, -5.77, -0.73, -1.25, 2.04, 2.25, 1.07, 0.06, 1.15, 1.77, -1.42, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.41, 1.10, 1.52, -8.67, -2.99, 0.84, 0.08, -2.69, 2.16, -0.02, -0.28, -3.58, 2.57, 2.06, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.73, 1.22, 0.60, 0.35, -0.09, 0.27, 0.26, -1.06, 3.40, -0.74, -1.48, -2.49, 2.56, 0.90, 1.02, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.97, -0.92, -0.98, 4.09, 0.49, 0.79, -1.52, -0.38, -1.40, 0.11, -0.31, -0.56, 0.63, -0.48, 0.48, -13.36, 1.00, 0.00, 0.00, 0.00, 0.00,
-0.71, 0.33, -0.03, -3.11, -2.19, 2.64, -4.11, -5.66, -1.84, 0.26, -1.46, -4.89, 4.44, 5.92, 2.26, 71.91, -4.62, 1.00, 0.00, 0.00, 0.00,
-0.53, 0.88, 0.78, -5.92, -1.52, -0.20, 1.43, -1.46, 4.27, -0.40, -0.37, -3.12, 1.38, 0.19, -0.12, -22.47, 1.46, -0.19, 1.00, 0.00, 0.00,
1.05, 1.04, 1.45, -4.72, -1.74, 0.20, 0.91, 0.92, 1.73, -0.23, 0.24, 0.77, -0.64, 2.07, 0.08, 67.85, -4.43, 0.90, -0.46, 1.00, 0.00,
0.29, -0.63, -0.60, 3.84, 1.32, -0.21, -1.11, -0.38, -1.06, 0.05, -0.15, 0.36, 0.49, 1.52, 0.81, 22.71, -1.42, 0.24, -0.71, -0.44, 1.00,
0.01, -0.91, -0.63, 2.82, 0.41, 0.28, -0.63, -0.00, 1.34, -0.36, 0.01, 0.20, 0.20, 0.45, 0.49, 20.21, -1.30, 0.22, -0.62, -2.39, 0.83, 1.00,
```

U:

```
-3.59, -2.30, -4.12, -4.61, -3.59, -4.74, -2.65, -4.96, 1.94, -4.57, -0.70, 1.05, 3.29, 1.28, 0.46, -3.86, -0.17, -2.89, 1.89, -2.08, 3.11, 2.57,
0.00, 3.88, -5.42, -1.33, -1.32, 2.44, 2.37, 1.44, 0.15, -1.44, -3.96, 3.87, 4.23, -4.24, -2.89, 4.02, 4.86, -5.01, 2.75, 1.61, -0.05, 1.80,
0.00, 0.00, 8.32, 10.65, 1.52, 2.35, 3.50, 8.88, -0.56, 9.79, 0.26, -3.91, -0.86, -5.39, 3.78, 7.01, 1.73, 8.16, -6.66, 4.95, -3.06, -0.51,
0.00, 0.00, 0.00, 2.05, -1.70, 5.37, 5.89, 10.76, 1.51, 9.58, -0.55, 1.41, 6.78, -6.99, -1.87, 3.53, -0.44, -3.07, -7.31, 5.40, -4.13, 2.92,
0.00, 0.00, 0.00, 0.00, 5.65, -17.55, -22.72, -33.84, -4.70, -33.43, -5.54, -4.76, -25.59, 21.32, 7.97, -5.32, 9.35, 15.23, 29.26, -15.02, 20.10, -6.20,
0.00, 0.00, 0.00, 0.00, 0.00, -11.14, -24.92, -33.64, -1.34, -35.74, -14.31, -0.50, -28.83, 18.75, 7.45, -4.19, 15.23, 12.81, 31.81, -8.22, 27.67, -1.85,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -8.79, -5.00, 4.88, -3.51, -10.54, 2.38, -8.50, 1.29, 1.02, 2.77, 3.37, 9.92, 6.29, 0.29, 8.05, 7.11,
```

```

0.00, 0.00, 0.00, 0.00, -0.00, -0.00, 0.00, -3.93, -1.97, -5.14, 5.90, -7.33, -1.10, 9.72, 0.30, -6.72, -6.81, -4.84, 2.21, -7.97, -4.45, -4.44,
0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, -2.60, -5.26, 3.37, 3.21, -2.23, 6.47, -2.23, -2.48, -0.84, -4.84, 5.74, -0.31, 1.86, -0.56,
0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, -20.85, 13.27, 13.89, -6.39, 38.09, -16.59, -6.90, -23.12, -8.10, 25.42, -14.71, -2.54, 9.03,
0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -14.07, -5.32, -6.02, -18.70, 7.24, 7.26, 17.88, 3.50, -2.29, 8.50, 13.35, -6.11,
-0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 8.88, -8.44, -1.35, 6.23, -2.19, 6.20, -13.20, 1.63, 11.43, 6.39, 4.85,
-0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -11.39, -3.53, 16.08, -2.85, 11.03, -14.76, 6.14, 12.85, 2.16, 1.35,
0.00, 0.00, 0.00, 0.00, -0.00, -0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 3.42, 1.01, -6.34, 2.40, -2.23, 6.36, 9.88, -7.57, -0.84,
-0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -17.99, 5.30, -24.48, -2.98, -21.78, -30.89, 26.20, 10.97,
0.00, 0.00, 0.00, 0.00, -0.00, -0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.24, 6.72, 17.82, -0.56, 9.27, -14.08, -17.97,
0.00, 0.00, 0.00, 0.00, -0.00, -0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 102.64, 246.31, 9.77, 137.36, -201.47, -238.16,
-0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, -0.00, -122.09, 70.26, -48.22, 76.78, 180.82,
-0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, -26.81, -3.34, 10.27, -13.45,
-0.00, 0.00, 0.00, 0.00, -0.00, -0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 2.22, 8.46, -3.03,
-0.00, 0.00, 0.00, 0.00, 0.00, -0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 18.35, 2.17,
-0.00, 0.00, 0.00, 0.00, -0.00, -0.00, 0.00, 0.00, -0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, -0.20,

```

Matlab nie dysponuje zwykłą LU faktoryzacją, a jedynie jej wariantem z pivotingiem, więc porównanie dla tego przypadku pomijamy.

4. LU faktoryzacja z pivotingiem

4.1. Pseudokod programu

```

lup_decompose(A) {
    n = A.length
    U = copy_mx(A)
    L = create_empty_mx(n)
    P = create_empty_mx(n)
    fill_diagonal(P)

    for (k = 0; k < n - 1; k++) {
        max_idx = find_row_idx_with_max_val(U, k)
        if (max_idx != k) {
            swap_rows(U, k, max_idx)
            swap_rows(P, k, max_idx)
            swap_rows(L, k, max_idx)
        }

        for (i = k + 1; i < n; i++) {
            factor = U[i][k] / U[k][k];
            L[i][k] = factor;
            for (j = k; j < n; j++) {
                U[i][j] -= factor * U[k][j];
            }
        }
    }
    fill_diagonal(L)

    return L, U, P;
}

```


Powyższy kod wykorzystuje trochę więcej funkcji pomocniczych, niż zwykła LU faktoryzacja. Oto one wraz z krótkim opisem:

- `copy_mx` - jak i w LU faktoryzacji, funkcja ta kopiuje zadaną macierz
- `create_empty_mx` - funkcja tworzy macierz kwadratową o zadanej wielkości i uzupełnia ją zerami
- `fill_diagonal` - funkcja uzupełnia przekątną macierzy jedynekami
- `find_row_idx_with_max_val` - zwraca numer wiersza z elementem o największej wartości bezwzględnej z macierzy U i kolumny k odliczając od wiersza k
- `swap_rows` - w zadanej macierzy zamienia miejscami wiersze o podanych numerach

4.2. Kod programu

<https://github.com/mwardynski/matrix-calculus/blob/prog2/prog2/lup.js>

4.3. Wyniki i ich analiza

Po wywołaniu funkcji `lup_decompose(A)`, otrzymamy następujące macierze:

L:

```
1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.77, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-1.00, -0.79, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.30, 0.63, 0.48, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.54, 0.69, -0.92, -0.62, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.52, 0.21, 0.24, 0.34, 0.67, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.18, -0.34, -0.09, 0.28, -0.17, -0.52, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.22, 0.72, 0.18, 0.34, 0.63, -0.39, 0.38, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.00, -0.53, 0.05, 0.04, -0.09, -0.39, 0.72, 0.08, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.83, -0.68, 0.67, 0.11, 0.23, 0.03, 0.72, 0.62, -0.04, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.17, 0.59, -0.66, -0.52, 0.73, -0.01, 0.50, 0.06, -0.47, 0.33, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.40, 0.42, -0.07, 0.67, -0.76, 0.22, 0.18, -0.75, -0.11, 0.64, -0.26, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.76, -0.38, -0.06, -0.19, -0.17, 0.72, 0.27, -0.25, -0.93, 0.49, 0.25, -0.15, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.69, -0.95, -0.01, 0.13, -0.36, -0.74, 0.74, -0.12, -0.08, 0.26, 0.59, -0.13, 0.51, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.16, 0.56, -0.33, 0.23, -0.01, 0.19, -0.11, -0.12, -0.76, -0.52, 0.00, 0.25, 0.05, -0.15, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.77, 0.57, -0.05, 0.37, 0.35, -0.66, 0.85, 0.72, 0.39, 0.09, -0.21, 0.46, -0.35, 0.53, -0.75, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.71, -0.51, 0.42, -0.48, 0.36, -0.51, 0.79, 0.05, 0.26, 0.12, 0.51, -0.94, 0.22, -0.28, 0.52, 0.97, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00,
-0.34, 0.13, -0.06, -0.42, 0.21, -0.09, 0.20, -0.40, 0.07, 0.49, 0.48, 0.62, -0.35, 0.20, 0.17, -0.19, -0.15, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.73, -0.03, -0.79, 0.42, -0.15, 0.34, 0.05, 0.35, -0.05, 0.40, -0.65, 0.19, 0.09, -0.02, -0.53, -0.66, -0.16, -0.76, 1.00, 0.00, 0.00, 0.00,
-0.39, 0.53, 0.57, 0.43, -0.15, -0.11, -0.75, -0.53, 0.23, -0.18, 0.02, -0.01, -0.87, -0.36, -0.91, 0.13, 0.23, -0.02, -0.53, 1.00, 0.00, 0.00,
0.21, -0.38, -0.38, -0.23, -0.16, 0.36, 0.53, -0.45, 0.18, -0.06, -0.31, 0.28, 0.12, -0.18, 0.73, -0.40, -0.04, -0.91, 0.76, -0.80, 1.00, 0.00,
0.41, 0.55, -0.82, -0.52, 0.66, -0.09, 0.28, 0.29, -0.55, 0.50, 0.53, -0.73, 0.42, 0.20, -0.85, 0.90, 0.18, -0.52, -0.30, 0.54, -0.27, 1.00,
```

U:

```
-4.90, -2.86, 2.31, 4.27, -3.48, -3.94, 0.05, 2.22, 2.09, 3.45, -0.98, -2.21, 3.94, -3.94, 4.21, 2.03, 1.85, 3.86, -3.88, 2.23, 1.19, 3.13,
0.00, 6.65, 1.24, -0.61, 5.35, 2.98, 4.38, -0.53, -1.30, -1.69, -1.40, -0.18, -6.38, 5.21, -7.59, -5.74, -5.25, -4.57, 0.48, 2.25, -5.20, -4.36,
0.00, 0.00, 7.31, 5.38, 0.20, -0.80, 4.64, 1.32, 1.41, 3.31, 1.40, -1.25, -1.88, -0.15, 3.05, -0.29, -0.12, 3.85, -0.21, 9.01, -7.85, -2.76,
```

P:

Wywołanie funkcji w Matlabie zwraca takie same wyniki, a jedyna różnica bierze się z błędu zaokrąglenia operacji zmiennoprzecinkowych.

Poniżej wyniki uzyskane z Matlaba, a właściwie z Octave:

L:

1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.7710	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1.0000	-0.7903	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.2981	0.6290	0.4782	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5352	0.6904	-0.9221	-0.6202	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.5200	0.2147	0.2445	0.3408	0.6703	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.1785	-0.3435	-0.0862	0.2801	-0.1723	-0.5178	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.2220	0.7160	0.1807	0.3434	0.6336	-0.3853	0.3834	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.0046	-0.5328	0.0493	0.0392	-0.0882	-0.3939	0.7243	0.0797	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0300	-0.6821	0.6742	0.1110	0.2288	0.0341	0.7226	0.6172	-0.0352	1.0000	0	0	0	0	0	0	0	0	0	0	0	0
-0.1656	0.5097	-0.6622	-0.5159	0.7290	-0.0078	0.5014	0.0557	-0.4669	0.3338	1.0000	0	0	0	0	0	0	0	0	0	0	0
0.3965	0.4225	-0.0696	0.6744	-0.7564	0.2162	0.1757	-0.7547	-0.1158	0.6355	-0.2562	1.0000	0	0	0	0	0	0	0	0	0	0
-0.7587	-0.3810	-0.0559	-0.1914	-0.1676	0.7182	0.2695	-0.2482	-0.9328	0.4914	0.2491	-0.1516	1.0000	0	0	0	0	0	0	0	0	0
-0.6865	-0.9486	-0.0086	0.1336	-0.3563	-0.7427	0.7432	-0.1244	-0.0825	0.2555	0.5883	-0.1258	0.5071	1.0000	0	0	0	0	0	0	0	0
0.1633	0.5585	-0.3340	0.2262	-0.0128	0.1916	-0.1083	-0.1190	-0.7586	-0.5163	0.0038	0.2484	0.0539	-0.1539	1.0000	0	0	0	0	0	0	0
0.7684	0.5738	-0.0548	0.3690	0.3483	-0.6592	0.8546	0.7192	0.3878	0.0945	-0.2148	0.4554	-0.3491	0.5388	-0.7535	1.0000	0	0	0	0	0	0
-0.7090	-0.5862	0.4184	-0.4827	0.3610	-0.5070	0.7904	0.0529	0.2558	0.1171	0.5859	-0.9364	0.2197	-0.2799	0.5182	0.9720	1.0000	0	0	0	0	0
-0.3395	0.1293	-0.0627	-0.4169	0.2084	-0.0918	0.1999	-0.4021	0.0747	0.4885	0.4762	0.6196	-0.3453	0.1957	0.1709	-0.1853	-0.1500	1.0000	0	0	0	0
0.7327	-0.0308	-0.7896	0.4164	-0.1535	0.3443	0.0481	0.3512	-0.0456	0.3998	-0.6497	0.1889	0.0909	-0.0238	-0.5349	-0.6558	-0.1639	-0.7617	1.0000	0	0	0
-0.3909	0.5263	0.5744	0.4299	-0.1502	-0.1052	-0.7462	-0.5259	0.2346	-0.1781	0.0203	-0.0130	-0.8710	-0.3647	-0.0955	0.1313	0.2293	-0.0225	-0.5266	1.0000	0	0
0.2112	-0.3757	-0.3764	-0.2281	-0.1569	0.3613	0.5290	-0.4537	0.1816	-0.0557	-0.3856	0.2797	0.1191	-0.1777	0.7345	-0.3952	-0.0377	-0.9139	0.7605	-0.0049	1.0000	0
0.4096	0.5491	-0.8163	-0.5240	0.6611	-0.0887	0.2841	0.2884	-0.5492	0.4959	0.5304	-0.7266	0.4198	0.1976	-0.0457	0.0974	0.1752	-0.5242	-0.3007	0.5395	-0.2744	1.0000

U:

-4.8968	-2.8616	2.3111	4.2650	-3.4787	-3.9406	0.0518	2.2161	2.0933	3.4492	-0.9759	-2.2081	3.9373	-3.9428	4.2066	2.0294	1.8486	3.8559	-3.8784	2.2303	1.1916	3.1342
0	6.6531	1.2377	-0.6109	5.3451	2.9773	4.3824	-0.5284	-1.2972	-1.6929	-1.3961	-0.1773	-6.3824	5.2050	-7.5945	-5.7374	-5.2459	-4.5729	0.4807	2.2528	-5.2022	-4.3601
0	0	7.3081	5.3785	0.1964	-0.0011	4.6377	1.3241	1.4101	3.3080	1.3961	-1.2517	-1.8755	-0.1507	3.0499	-0.2889	-0.1202	3.8503	-0.2121	9.0061	-7.8470	-2.7574
0	0	0	-0.4103	-5.1553	0.5910	-2.8493	2.0058	-0.5214	-0.5260	-0.0799	-1.9924	2.2897	-6.6119	-0.7271	2.2938	-1.8706	1.7270	5.4232	-6.4949	8.2898	5.1842
0	0	0	0	-9.2872	1.1326	-2.3018	5.8524	-2.0422	1.8585	7.4406	-2.4338	3.0266	-3.7344	7.1155	2.0665	3.0204	3.4182	6.7405	2.7823	-3.9159	-2.2784
0	0	0	0	0	-7.1668	4.6096	-5.6238	-0.3709	-0.7946	-2.4969	0.7077	1.7468	0.5700	-5.1934	1.8562	3.9476	-1.7949	-12.2020	-3.2168	-0.1130	0.7704
0	0	0	0	0	0	7.3998	2.8380	0.8515	5.1041	1.2991	1.4594	3.1587	1.5491	-5.7036	-3.6077	-2.7736	-6.3500	-12.0557	4.1033	-8.0559	-1.0579
0	0	0	0	0	0	0	-10.5089	3.0025	-4.3946	-7.4766	5.7485	-0.2509	-2.6523	0.2709	4.1111	7.3423	-2.2624	-6.9679	-1.0063	7.3211	1.7443
0	0	0	0	0	0	0	0	-6.3490	-3.3181	0.3321	-1.1001	-6.0029	-2.1232	-2.4841	-0.1573	2.3112	-1.9782	3.0018	-6.4695	0.7472	2.6028
0	0	0	0	0	0	0	0	0	-6.5407	1.0265	-1.1412	-0.3003	3.8273	-5.6242	-0.9353	-9.1798	2.9766	0.1002	-5.9063	3.6076	-2.0267
0	0	0	0	0	0	0	0	0	0	-8.1244	2.1103	1.0333	-11.9361	2.1507	9.3439	9.6137	1.3127	4.1500	-1.3449	7.2264	9.1447
0	0	0	0	0	0	0	0	0	0	0	8.3081	-4.4988	-6.2115	10.6119	10.1092	14.6106	1.1277	1.7771	4.2791	-0.1153	0.3712
0	0	0	0	0	0	0	0	0	0	0	-11.6550	-9.1820	11.9272	3.6048	3.8778	-3.3678	6.3020	0.4949	0.2964	4.1000	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	13.3049	-10.0232	-0.8949	-3.5374	-1.7230	-9.4332	3.5513	-2.5044
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5.8382	0.1327	-7.1041	6.2233	1.2995	-6.1931	-4.2165
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.2776	-2.2200	4.6726	11.3741	-7.5376	-1.0052
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.1943	-2.3429	-13.1536	6.8773	8.4591
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5.7016	-0.4003	-1.2883	6.5762	4.2529
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14.7024	0.3892	-3.0091	10.1291
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-6.2825	-3.4799	10.6991

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.4211	2.6788
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0658

P:

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	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	1	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	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	1	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	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	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	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	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	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	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	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	1	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
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	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	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

5. Wnioski

Przyglądając się wynikom, można dopatrzeć się małych rozbieżności pomiędzy wartościami z naszej implementacji oraz Matlabem. Są one spowodowane ograniczoną dokładnością obliczeń zmiennoprzecinkowych. Arytmetyka zmiennoprzecinkowa przy takiej ilości zróżnicowanych operacji jest ciężka do uniknięcia i nawet zaczynając od macierzy A wypełnionej wyłącznie wartościami całkowitymi, na pewną ilość operacji zmiennoprzecinkowych i tak się natkniemy. Jednakże można zauważyć, że dominująca ilość wartości całkowitych w macierzy A pozytywnie wpływa na różnicę pomiędzy wynikami naszej implementacji a Matlabem. Z tego wniosek, że możemy pokusić się o prosty trick i wymnożyć wszystkie kolumny macierzy A , oraz wektor wyrazów wolnych b , przez wartość wystarczająco dużą, aby pozbyć się liczb po przecinku. To podejście, choć trywialne, może przynieść dla niektórych przypadków dobre rezultaty, jednakże jest to broń obusieczna, gdyż z problemu niedokładności obliczeń zmiennoprzecinkowych, łatwo możemy wpaść w jeszcze gorszy problem, a mianowicie: przekroczenie zakresu liczb całkowitych.