Rachunek Macierzowy
Program 2: Algorytmy eliminacji Gaussa i LU

Paulina Dziwak, Karol Markowicz

Zadanie 1. „Algorytm eliminacji Gaussa”

Algorytm eliminacji Gaussa jest jednym z najpopularniejszych sposobów rozwiązywania układów równań liniowych. Polega on na przekształcaniu macierzy współczynników równań liniowych poprzez kombinacje linowych wierszy w celu wyeliminowania zmiennych, aż do uzyskania macierzy trójkątnej górnej lub dolnej, która jest łatwa do rozwiązania za pomocą metody substytucji wstecznej lub do przodu.

Pseudokod:

```
Funkcja gaussianElimination(A, b):
    n = długość_wektora(b)

    Dla każdego i od 0 do n-1:
        Dla każdego j od i+1 do n-1:
            factor = A[j][i] / A[i][i]
            Dla każdego k od i do n-1:
                A[j][k] -= factor * A[i][k]
            b[j] -= factor * b[i]

        mainElement = A[i][i]
        Dla każdego k od i do n-1:
            A[i][k] /= mainElement
        b[i] /= mainElement
```

Kod:

Wynik programu dla macierzy o wymiarze 9:
Input:
A matrix:
9.499585762798151 -9.236741620224073 7.925359286205563 -1.763103342568261 5.048543142086396
5.514493425045439 2.7127591052278035 8.452496711192769 -4.660362382081395
9.827473817825716 3.946815968313393 9.925907187765993 2.7602265115469784 -7.220661205054471
7.557330209617881 7.250386459454166 -1.553846916433411 -8.59443981436717
8.86220054785084 9.414745156392978 -5.016374410024936 -5.042138663741367 -6.3311077810291465
8.972467350305664 -4.132291439593418 5.530104730745329 -9.985339415527417
3.484867819894168 5.26783739524322 -3.0917724688617465 6.99633150695988 -9.364618635483687 -
8.642192310797643 2.738587255867868 7.15997628182858 0.4628587532672519
-8.200288474060788 4.972356519830845 -5.434651396955852 0.8375274064960898 -9.071664223582736
4.031893763962289 -8.090515646994316 5.604053173354718 4.764555383639809
-6.741711513701891 6.988394192735331 8.74192868545034 2.193298689774622 -8.062429585576629 -
8.704689454368799 2.444180652477659 8.473686163164153 -6.433649855839154

-8.3015168743427 -7.808815321141031 0.7059707639378026 -0.9940655578130162 9.911045294696539 -2.302849539073824 -6.038045066722832 7.78724787015441 -6.139989047067886
5.032619692151252 -8.5074278413499 -4.932826332705044 -9.919016663403477 -7.722895744582832 -7.241635967505886 6.600202119467923 6.909698185977547 8.73177020329851
-1.0143898067281256 -0.42244204770360483 -5.488949125405118 6.022860905136081 7.532905635654114 3.29926574009788 -0.31921692704107585 -1.647678930955479 3.5599977984861386

b vector:
2.526445051854026 -7.151291589725126 1.952439180001642 0.5846994333908562 6.300119571830326 6.629258001144493 9.979914282339035 7.143442882169541 -8.902467950800188

Output:
A matrix [After gaussian elimination]:
1.0 -0.9723309890412889 0.834284723997387 -0.1855979183295388 0.5314487671512239 0.5804983041093275 0.28556604182167483 0.8897752936021752 -0.49058585273603145
0.0 1.0 0.12790313724935698 0.3395095877513512 -0.9215757680757755 0.13719797866597924 0.3291268549375493 -0.7626874242882784 -0.27944866475476987
0.0 0.0 1.0 0.6468533330562666 -0.3789473685996448 -0.09201076889252033 0.856042186776022 -0.7744695136160389 0.04068446057074917
0.0 0.0 0.0 1.0 -0.6377988338621525 -1.3446597565097025 0.5351824763297921 0.5545278376255233 0.524740004441428
0.0 0.0 0.0 0.0 1.0 -1.1275884652419472 0.7989138935638528 -1.7000868546480565 -0.035816250313825106
0.0 0.0 0.0 0.0 0.0 1.0 0.013147387222021833 -1.2156496362442881 0.3031684753928081
0.0 0.0 0.0 0.0 0.0 0.0 1.0 -2.946361742595594 2.372295894264979
0.0 0.0 0.0 0.0 0.0 0.0 -3.552713678800501E-15 1.0 -4.981665274592852
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.7763568394002505E-15 1.0

b vector [After gaussian elimination]:
0.2659531810058472 -0.7232017136527192 -0.8586466574343429 -0.019748858163851676 -1.0690208332121143 -0.8479888042085357 -2.0766383360684313 0.9542657745217425 -0.13514066127644073

Solution:
-0.023573040790424674 -0.3975705518409805 0.27013065063376746 -0.45061492940659387 -0.36566797686892705 -0.4531712007840293 -0.9279985034768234 0.2810402350553829 -0.13514066127644073

Rozwiązanie z Matlaba:

```
>> eliminacja_gaussa
A matrix [After gaussian elimination]:
    1.0000   -0.9723    0.8343   -0.1856    0.5314    0.5805    0.2856    0.8898   -0.4906
         0    1.0000    0.1279    0.3395   -0.9216    0.1372    0.3291   -0.7627   -0.2794
         0         0    1.0000    0.6469   -0.3789   -0.0920    0.8560   -0.7745    0.0407
         0         0         0    1.0000   -0.6378   -1.3447    0.5352    0.5545    0.5247
         0         0         0         0    1.0000   -1.1276    0.7989   -1.7001   -0.0358
         0         0         0         0         0    1.0000    0.0131   -1.2156    0.3032
         0         0         0         0         0         0    1.0000   -2.9464    2.3723
         0         0         0         0         0         0   -0.0000    1.0000   -4.9817
         0         0         0         0         0         0         0    0.0000    1.0000

b vector [After gaussian elimination]:
    0.2660   -0.7232   -0.8586   -0.0197   -1.0690   -0.8480   -2.0766    0.9543   -0.1351

Solution:
   -0.0236
   -0.3976
    0.2701
   -0.4506
   -0.3657
   -0.4532
   -0.9280
    0.2810
   -0.1351
```

Rozwiązanie jest prawidłowe, aczkolwiek zauważalny jest problem z bardzo małymi wartościami które w matlabie są przedstawiane jako 0, a w programie jako bardzo mała

liczba. Problem ten wynika z obliczeń na bardzo małych wartościach i ograniczeń związanych z precyzją arytmetyki zmiennoprzecinkowej na komputerze.

Zadanie 2. „Algorytm eliminacji Gaussa z pivotingiem”

Algorytm eliminacji Gaussa z pivotingiem jest ulepszoną wersją podstawowego algorytmu eliminacji Gaussa. Pivoting polega na zmianie kolejności wierszy macierzy w celu uniknięcia dzielenia przez zero oraz minimalizacji błędów numerycznych.

Działanie tego algorytmu:

1) **Wybór pivotu:** W każdej iteracji eliminacji, wybierany jest pivot. Jest to element macierzy, który służy do wyeliminowania współczynników poniżej niego. W zwykłym przypadku, pivotem jest element na przekątnej lub poza nią o największej wartości bezwzględnej w danej kolumnie.

2) **Zamiana wierszy:** Jeśli pivot nie znajduje się na przekątnej, dokonuje się zamiany wierszy tak, aby pivot znalazł się na przekątnej. To zapobiega dzieleniu przez zero i minimalizuje błędy numeryczne.

3) **Eliminacja współczynników**: Po wyborze pivotu i, jeśli konieczne, zamianie wierszy, przeprowadzana jest eliminacja współczynników w danej kolumnie. Wiersze poniżej aktualnego pivotu są przekształcane w taki sposób, aby wyzerować elementy poniżej niego.

4) **Powtarzanie kroków:** Krok 1-3 jest powtarzany dla każdej kolumny macierzy, aż do uzyskania macierzy trójkątnej górnej lub dolnej.

5) **Rozwiązanie układu równań**: Po uzyskaniu macierzy trójkątnej, można łatwo znaleźć rozwiązania układu równań za pomocą metody substytucji wstecznej (dla macierzy trójkątnej górnej) lub substytucji do przodu (dla macierzy trójkątnej dolnej).

Pseudokod:

```
Funkcja gaussianEliminationWithPivoting(A, b):
    n = długość_wektora(b)

    Dla każdego r od 0 do n-1:
        // Wybieramy pivot
        maxIndex = r
        maxValue = |A[r][r]|
        Dla każdego c od r+1 do n-1:
            Jeśli |A[c][r]| > maxValue:
                maxIndex = c
                maxValue = |A[c][r]|

        // Zamieniamy
        Jeśli maxIndex != r:
            Zamień_wiersze(A, r, maxIndex)
            Zamień_wartości(b, r, maxIndex)

        Wywołaj gaussianElimination(A, b)
```

Kod

Wynik programu dla macierzy o wymiarze 9:
A matrix [After gaussian elimination]:
1.0 0.2037733997281862 -0.8500809674265932 -0.6167620680736758 0.5967860989141234 -0.3614992971081078 -0.5212294535317776 0.2124355103803917 -0.14287365640981828
0.0 1.0 0.01638790915030734 1.0104543164780004 -0.028351005987258926 -0.21473511924130836 1.6452594542268144 -0.05244604211779072 -0.4376717947298523
0.0 0.0 1.0 2.176746078870957 2.486826504414561 -2.017466865530107 3.438594046077781 -4.614044684229588 -0.6320380425305615
0.0 0.0 0.0 1.0 0.8194861716021609 -0.2805363807175816 1.0036303930512387 -1.0170392884322002 0.012560142890155643
0.0 0.0 0.0 0.0 1.0 -0.7071880016381843 0.3087580745134118 -0.6644857829817912 -0.13714423250608987
0.0 0.0 0.0 0.0 0.0 1.0 -0.2533327419058072 1.8947643071263778 1.4631482938642066
0.0 0.0 0.0 0.0 0.0 0.0 1.0 -0.3759202410154367 -0.19858306312242122
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.3976495438671264
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0

b vector [After gaussian elimination]:
0.21665457588484813 0.17642486972395705 1.821837263813264 0.5221573095550025 0.35003612609736356 -1.0402376278605634 0.988971983070382 -0.9258653092446084 1.634724120059884

Solution:
-1.564383043193184 0.7734129234446838 -2.674283027250514 -1.176140801043726 -0.8818862521672488 -0.263399189098823 0.7211830580459433 -1.575912609935011 1.634724120059884

Rozwiązanie z Matlaba:

```
A = [
-8.341966522648791 7.314885645946639 1.3956112936586464 -1.7123728019334106 8.305936838683522 -7.95377280427285 5.316754256250851 7.597985793359538 -3.3158779149780866;
8.293268789132554 9.82844375668115 -6.9165770193657945 3.108604985911306 4.718572973967499 -4.745631785798974 9.067241825467896 1.3349528744747765 -4.746879865284283;
3.304658130010152 -4.518978190740659 -0.8960446022352269 -2.9351137243294634 7.088745515483904 -4.11109817662505 -3.3940415770296717 -8.245781906952796 0.537423932494562;
1.0265597244230875 1.615441030269654 -7.244872924323442 6.604835788536196 0.8437849953316423 6.691958399806698 -0.4025851871679169 9.578141192898194 3.527811253560433;
-9.886900961821329 -2.0146874217662063 8.40466633447599 6.097865484052537 -5.900365055355645 3.5741077482758854 5.1533439854529 38 -2.1003288519048997 1.4125776900771623;
6.877010579926782 5.55930754895785 5.691916008376099 -5.453613084902509 -3.0897129784726403 -2.9424881746517713 7.004990927395053 0.5120987844202336 1.9854772542572867;
3.7933091352829464 7.065571781018416 -1.4013139649910773 7.483292051960444 -2.195087951294095 -6.1810939705019 3 -1.479514976229357 -7.505438607220462 -9.095266302166628;
-4.556049907512161 2.003056559215125 9.88939699125272 2.743604124699324 5.244772753662048 2.129379247203243 -2.323499211523936 -2.8760906869892366 9.914937481899123;
-8.337188530003685 -6.559651547974672 0.8347229148897704 -1.0901594872117677 4.082775279790411 8.194736819589345 7.1140287098297 04 5.025278543650051 -0.7863366062489074;
];


b = [-1.8006383470387952 3.2326077606304153 3.4404355809235465 -0.8743375728036362 -2.142042334698897 -4.275826144744743 -6.066902885044496 -7.118497217316713 -2.81422387265426];

x= GEPP(A,b)
```

```
Solution:
   -1.5644
    0.7734
   -2.6743
   -1.1761
   -0.8819
   -0.2634
    0.7212
   -1.5759
    1.6347
```

Rozwiązania są zgodne, a ponadto zauważalne jest zastępowanie bardzo małych wartości przez zera, zarówno w Javie, jak i w Matlabie.

Zadanie 3 „Algorytm LU faktoryzacji bez pivotingu”

Algorytm polega na stopniowym redukowaniu macierzy do postaci trójkątnej dolnej. W każdym kroku wykonuje się odejmowanie wielokrotności jednego wiersza od kolejnych wierszy w celu uzyskania zerowych wartości poniżej przekątnej. W porównaniu do algorytmów z pivotingiem, algorytm bez pivotingu jest mniej kosztowny obliczeniowo, co jest istotne w przypadku dużych macierzy.

Pseudokod:

```
Funkcja luDecomposition(A, U, L):
    n = liczba_wierszy(A)

    Dla każdego k od 0 do n-1:
        Skopiuj elementy od k-tej kolumny do ostatniej kolumny macierzy A do
k-tej kolumny macierzy U
        Dla każdego i od k+1 do n-1:
            Oblicz współczynnik L[i][k] jako A[i][k] / U[k][k]
            Dla każdego j od k+1 do n-1:
                Oblicz element A[i][j] jako A[i][j] - L[i][k] * U[k][j]

    Dla każdego i od 0 do n-1:
        Ustaw L[i][i] na 1
```

Kod

Wynik programu dla macierzy o wymiarze 9:

Input:

```
A matrix:
-8.3419665226487911 7.3148856459466391 1.3956112936586464 -1.7123728019334106 8.3059368386835221 -7.9537728042728501 5.3167542562508511 7.5979857933595381 -3.3158779149780866
8.2932687891325541 9.8284437566811501 -6.9165770193657945 3.1086049859113061 4.7185729739674991 -4.7456317857989741 9.0672418254678961 1.3349528744747651 -4.7468798652842831
3.3046581300101521 -4.5189781907406591 -0.8960446022352269 -2.9351137243294634 7.0887455154839041 -4.1110981766250501 -3.3940415770296717 -8.2457819069527961 0.5374239324945621
1.0265597244230851 1.6154410302696541 -7.2448729243234421 6.6048357885361961 0.8437849953316423 6.6919583998066981 -0.4025851871679169 9.5781411928981941 3.5278111253560431
-9.8869009618213291 -2.0146874217662063 8.4046663344759961 6.0978654840525371 -5.9003650553556451 3.5741077482758841 5.1533439854529381 -2.1003288519048997 1.4125776909771623
6.8770105799267821 5.5593075489578501 5.6919160083760991 -5.4536130849025091 -3.0897129784726403 -2.9424881746517713 7.0049909273950531 0.5120987844202336 1.9854772542572867
3.7933091352829464 7.0655717810184161 -1.4013139649910773 7.4832920519604441 -2.1950879512940951 -6.1810939705019301 -1.4795149762293571 -7.5054386072204621 -9.0952663021666281
-4.5560499075121611 2.0030565592151251 9.8893969912527221 2.7436041246993241 5.2447727536620481 2.1293792472032431 -2.3234992115239361 -2.8760906869923661 9.9149374818991231
-8.3371885300036851 -6.5596515479746721 0.8347229148897041 -1.0901594872117771 4.0827752797904111 8.1947368195893451 7.1140287098297041 5.0252785436500511 -0.7863366062489074

b vector:
-1.8006383470387952 3.2326077606304153 3.4404355809235465 -0.8743375728036362 -2.1420423346988971 -4.2758261447447431 -6.0669028850444961 -7.1184972173167131 -2.8142238726542601
```

Output:

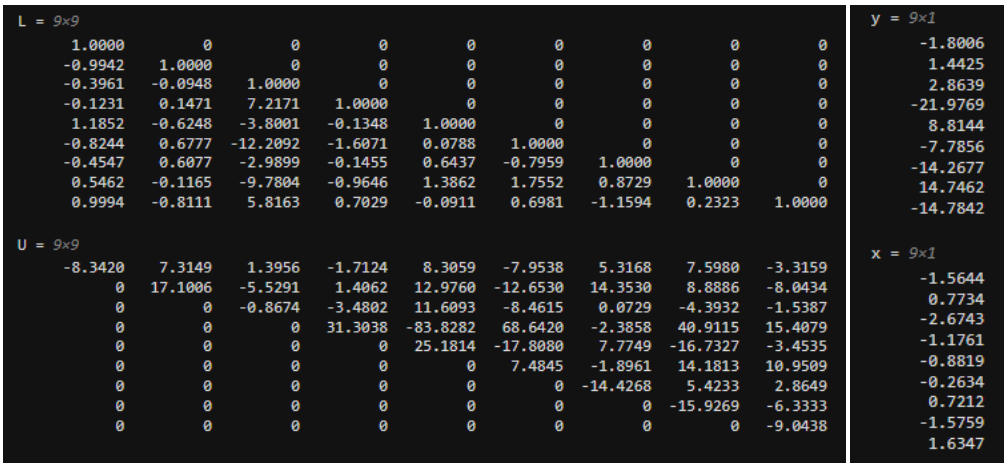
```
L matrix:
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
-0.9941623197139403 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
-0.3961485725263781 -0.09480334506426231 1.0 0.0 0.0 0.0 0.0 0.0 0.0
-0.1230596792298236 0.1471062111873509 7.2170828429678351 1.0 0.0 0.0 0.0 0.0 0.0
1.1852002684232759 -0.6247894640710597 -3.8001296028062805 -0.1347780297509574 1.0 0.0 0.0 0.0 0.0
-0.8243872186798411 0.6777298565793431 -12.209190455940241 -1.6070945387040621 0.0787739027419678 1.0 0.0 0.0 0.0
-0.4547260079483599 0.6076876750007932 -2.9898770243857911 -0.1455141206989901 0.6436768216413125 -0.7958834630094161 1.0 0.0 0.0
0.5461601763975091 -0.1164894494439245 3 -9.7804247495946891 -0.9645707341157173 1.3861683412370431 1.7552450302259477 0.8728900016458475 1.0 0.0
0.9994272342579731 -0.8111016702502051 5.8162689844101521 0.7028965471782441 -0.0910901778883912 4 0.6981419401459278 -1.1594379654170441 0.2323250825873839 6 1.0

U matrix:
-8.3419665226487911 7.3148856459466391 1.3956112936586464 -1.7123728019334106 8.3059368386835221 -7.9537728042728501 5.3167542562508511 7.5979857933595381 -3.3158779149780866
0.0 17.1006274388976641 -5.5291128582431421 1.4062284689261271 12.9760224089110581 -12.6529730073725231 14.3529585702112071 8.8885840559546581 -8.0434007451271211
0.0 0.0 -0.8673535746499362 -3.4801526026958531 11.6093008670161861 -8.4615180853435991 0.0728915160824401 6 -4.3931831794483471 -1.5386976665261811
0.0 0.0 0.0 31.303796438317411 -83.828228684978171 68.641977588411721 -2.3857805785591691 40.911547914541541 15.4079031192558771
0.0 0.0 0.0 0.0 25.1813628558060321 -17.807957676523471 7.7749491089822261 -16.7326576137888671 -3.4534786823168781
0.0 0.0 0.0 0.0 0.0 7.4844845506950841 -1.8960649929792273 14.1813341838958281 10.9509108008025131
0.0 0.0 0.0 0.0 0.0 0.0 -14.4267939584144071 5.4233238619271871 2.8649169352979681
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -15.9268986512092281 -6.3333239838712781
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -9.0438406991780291

y:
-1.8006383470387952 1.4424809645724517 2.8638672907408631 -21.9768889481499381 8.8143867038983911 -7.7856424547740751 -14.2676950304008981 14.7461629450093491 -14.7841845289255791

x:
-1.5643830431931811 0.7734129234446781 -2.6742830272505134 -1.1761408010437222 -0.8818862521672527 -0.2633991890988121 0.7211830580459441 -1.5759126099350761 1.6347241200598851
```

Rozwiązanie z Matlaba:



```
L = 9×9
    1.0000         0         0         0         0         0         0         0         0
   -0.9942    1.0000         0         0         0         0         0         0         0
   -0.3961   -0.0948    1.0000         0         0         0         0         0         0
   -0.1231    0.1471    7.2171    1.0000         0         0         0         0         0
    1.1852   -0.6248   -3.8001   -0.1348    1.0000         0         0         0         0
   -0.8244    0.6777  -12.2092   -1.6071    0.0788    1.0000         0         0         0
   -0.4547    0.6077   -2.9899   -0.1455    0.6437   -0.7959    1.0000         0         0
    0.5462   -0.1165   -9.7804   -0.9646    1.3862    1.7552    0.8729    1.0000         0
    0.9994   -0.8111    5.8163    0.7029   -0.0911    0.6981   -1.1594    0.2323    1.0000

U = 9×9
   -8.3420    7.3149    1.3956   -1.7124    8.3059   -7.9538    5.3168    7.5980   -3.3159
         0   17.1006   -5.5291    1.4062   12.9760  -12.6530   14.3530    8.8886   -8.0434
         0         0   -0.8674   -3.4802   11.6093   -8.4615    0.0729   -4.3932   -1.5387
         0         0         0   31.3038  -83.8282   68.6420   -2.3858   40.9115   15.4079
         0         0         0         0   25.1814  -17.8080    7.7749  -16.7327   -3.4535
         0         0         0         0         0    7.4845   -1.8961   14.1813   10.9509
         0         0         0         0         0         0  -14.4268    5.4233    2.8649
         0         0         0         0         0         0         0  -15.9269   -6.3333
         0         0         0         0         0         0         0         0   -9.0438
```

```
y = 9×1
   -1.8006
    1.4425
    2.8639
  -21.9769
    8.8144
   -7.7856
  -14.2677
   14.7462
  -14.7842

x = 9×1
   -1.5644
    0.7734
   -2.6743
   -1.1761
   -0.8819
   -0.2634
    0.7212
   -1.5759
    1.6347
```

Macierze są zgodne.

Zadanie 4 „Algorytm LU z pivotingiem”

Algorytm faktoryzacji LU z pivotingiem charakteryzuje się większą niezawodnością w porównaniu do standardowego algorytmu bez pivotingu, szczególnie w kontekście macierzy źle uwarunkowanych lub obciążonych dużymi błędami numerycznymi. Dzięki zastosowaniu pivotingu minimalizuje się ryzyko wystąpienia błędów numerycznych, co przekłada się na poprawę stabilności obliczeń.

Pseudokod:

```
Funkcja luDecompositionWithPivoting(A, U, L, P):
    n = liczba_wierszy(A)

    Dla każdego k od 0 do n-1:
        maxRowIndex = k
        maxVal = wartość_bezwzględna(A[k][k])
        Dla każdego i od k+1 do n-1:
            Jeśli wartość_bezwzględna(A[i][k]) > maxVal:
                maxVal = wartość_bezwzględna(A[i][k])
                maxRowIndex = i

        Jeśli maxRowIndex != k:
            Zamień wiersze k i maxRowIndex w macierzach A i P
            Dla każdego i od 0 do k-1:
                temp = L[k][i]
                L[k][i] = L[maxRowIndex][i]
                L[maxRowIndex][i] = temp

        Skopiuj elementy od k-tej kolumny do ostatniej kolumny macierzy A do
k-tej kolumny macierzy U
        Dla każdego i od k+1 do n-1:
            Oblicz współczynnik L[i][k] jako A[i][k] / A[k][k]
            Dla każdego j od k do n-1:
                Oblicz element A[i][j] jako A[i][j] - L[i][k] * U[k][j]
```

Kod:

Wynik programu dla macierzy o wymiarze 9:
Input:

```
A matrix:
-8.341966522648791 7.314885645946639 1.3956112936586464 -1.7123728019334106 8.305936838683522 -7.95377280427285 5.316754256250851 7.597985793359538 -3.3158779149780866
8.293268789132554 9.82844375668115 -6.9165770193657945 3.108604985911306 4.718572973967499 -4.745631785798974 9.067241825467896 1.3349528744747765 -4.746879865284283
3.304658130010152 -4.518978190740659 -0.8960446022352269 -2.9351137243294634 7.088745515483904 -4.11109817662505 -3.3940415770296717 -8.245781906952796 0.537423932494562
1.0265597244230875 1.615441030269654 -7.244872924323442 6.604835788536196 0.8437849953316423 6.691958399806698 -0.4025851871679169 9.578141192898194 3.527811253560433
-9.886900961821329 -2.0146874217662063 8.40466633447599 6.097865484052537 -5.900365055355645 3.5741077482758854 5.153343985452938 -2.1003288519048997 1.4125776909771623
6.877010579926782 5.55930754895785 5.691916008376099 -5.453613084902509 -3.0897129784726403 -2.9424881746517713 7.004990927395053 0.5120987844202336 1.9854772542572867
3.7933091352829464 7.065571781018416 -1.4013139649910773 7.483292051960444 -2.195087951294095 -6.18109397050193 -1.479514976229357 -7.505438607220462 -9.095266302166628
-4.556049907512161 2.003056559215125 9.88939699125272 2.743604124699324 5.244772753662048 2.129379247203243 -2.323499211523936 -2.8760906869892366 9.914937481899123
-8.337188530003685 -6.559651547974672 0.8347229148897704 -1.0901594872117677 4.082775279790411 8.194736819589345 7.114028709829704 5.025278543650051 -0.7863366062489074

b vector:
-1.8006383470387952 3.2326077606304153 3.4404355809235465 -0.8743375728036362 -2.142042334698897 -4.275826144744743 -6.066902885044496 -7.118497217316713 -2.81422387265426
```

Output:

```
L matrix:
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.8437392621673 5 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
-0.6955678636291229 0.4612388267762478 1.0 0.0 0.0 0.0 0.0 0.0 0.0
-0.3836701864346537 0.6980328603518614 0.4093968127203571 1.0 0.0 0.0 0.0 0.0 0.0
0.4608167842588424 0.3251844162593661 7 0.555491392472387 0.0781911349355524 6 1.0 0.0 0.0 0.0 0.0
-0.1038302829559221 6 0.155994835950269 -0.3871301463512295 0.6562127232490257 -0.13393968581730495 1.0 0.0 0.0 0.0
-0.334246104292055 1 -0.5759866723647722 -0.0965395642184323 -0.3372958495867574 6 0.7433079457105612 -0.9411456183899486 1.0 0.0 0.0
-0.8388137821100211 0.9027971147500161 0.3724290076797482 4 0.9910924827552483 0.0805589505409490 7 0.5312190097866946 0.9899316577924171 1.0 0.0
0.8432559972227979 -0.5391997314326313 -0.6582198546712692 -0.6260047695025398 0.19423900374005024 0.259371375612475 9 0.7072942072530589 0.9577596775092219 1.0

U matrix:
-9.886900961821329 -2.0146874217662063 8.40466633447599 6.097865484052537 -5.900365055355645 3.5741077482758854 5.153343985452938 -2.1003288519048997 1.4125776909771623
0.0 9.01475652468549 9 -5.695735678154891 -6.857381326243648 13.284306497007309 -10.969387838709753 0.9686756042702385 9.370115709174575 -4.507725173717216
0.0 0.0 14.165026256982955 1.9507567002116524 -13.321055237874035 4.603053893376908 10.142700594662166 -5.270683644475236 5.047158771535518
0.0 0.0 0.0 13.810905165302085 -8.278167017554292 0.962702192251575 -4.3308872272021 1 -12.69411975480246 -7.4730527737863
0.0 0.0 0.0 0.0 11.690921348361314 1.4172127952270823 -10.308790715185884 -1.0348524791521996 8.51051317588627
0.0 0.0 0.0 0.0 0.0 10.114291411678904 5.369151862540336 14.049368337983237 12.375376530369401
0.0 0.0 0.0 0.0 0.0 0.0 11.12052921989756 5.650465236774631 1.700903055745881
0.0 0.0 0.0 0.0 0.0 0.0 0.0 -7.3156425229831 -2.909061912359179
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -9.043840699178027

P matrix:
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 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 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 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 0.0 0.0 0.0 1.0

y:
-1.8006383470387952 4.7518770309911655 -0.0037807740542685764 -4.880607302249039 -2.4737939088808476 -2.334141433351674 -5.936291294184773 -0.7645878500934407 4.2255265978398455

x:
-0.2422962892922337 0.5986079825345989 0.3453457066806684 -0.7977183652589026 -0.4152262565543896 0.2613892140242437 -0.6098586955121411 0.2903067128072435 7 -0.4672270043659541
```

Rozwiązanie z Matlaba:

```
Macierz L:
    1.0000         0         0         0         0         0         0         0         0
    0.8437    1.0000         0         0         0         0         0         0         0
   -0.6956    0.4612    1.0000         0         0         0         0         0         0
   -0.3837    0.6980    0.4094    1.0000         0         0         0         0         0
    0.4608    0.3252    0.5555    0.0782    1.0000         0         0         0         0
   -0.1038    0.1560   -0.3871    0.6562   -0.1339    1.0000         0         0         0
   -0.3342   -0.5760   -0.0965   -0.3373    0.7433   -0.9411    1.0000         0         0
   -0.8388    0.9028    0.3724    0.9911    0.0806    0.5312    0.9899    1.0000         0
    0.8433   -0.5392   -0.6582   -0.6260    0.1942    0.2594    0.7073    0.9578    1.0000
Macierz U:
   -9.8869   -2.0147    8.4047    6.0979   -5.9004    3.5741    5.1533   -2.1003    1.4126
         0    9.0148   -5.6957   -6.8574   13.2843  -10.9694    0.9687    9.3701   -4.5077
         0         0   14.1650    1.9508  -13.3211    4.6031   10.1427   -5.2707    5.0472
         0         0         0   13.8109   -8.2782    0.9627   -4.3309  -12.6941   -7.4731
         0         0         0         0   11.6909    1.4172  -10.3088   -1.0349    8.5105
         0         0         0         0         0   10.1143    5.3692   14.0494   12.3754
         0         0         0         0         0         0   11.1205    5.6505    1.7009
         0         0         0         0         0         0         0   -7.3156   -2.9091
         0         0         0         0         0         0         0         0   -9.0438
Macierz P:
     0     0     0     0     1     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     1     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     1     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     1
```

```
y = 9×1

   -1.8006
    4.7519
   -0.0038
   -4.8806
   -2.4738
   -2.3341
   -5.9363
   -0.7646
    4.2255

x = 9×1

   -0.2423
    0.5986
    0.3453
   -0.7977
   -0.4152
    0.2614
   -0.6099
    0.2903
   -0.4672
```

Wyniki są zgodne.