Министр науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа № 3

По дисциплине «Прикладная математика»

4 семестр

Выполнили: Шуляк Георгий Владимирович
М32031

Климачёва Екатерина Николаевна М32041

Проверила: Гомозова Валерия Эдуардовна

Санкт-Петербург, 2022 г.

LU-разложение - это представление матрицы A в виде произведения двух матриц, A = LU, где L - нижняя треугольная матрица, U - верхняя треугольная матрица.

LU-разложение существует только в том случае, когда матрица A обратима, а все ведущие (угловые) главные миноры этой матрицы невырождены.

Алгоритм нахождения LU-разложения:

1. Матрица U заполняется 0;
2. Матрица L заполняется 1;
3. В цикле *i* от 1 до *n* и цикле *j* от 1 до *n*:

- Если $i \leq j$, то $u_{ij} = a_{ij} - \sum_{k=1}^{i-1}(l_{ik} * u_{kj})$,
- Если $i > j$, то $l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1}(l_{ik} * u_{kj})}{u_{jj}}$.

Решение СЛАУ

LU-разложение матрицы A может быть использовано для решения СЛАУ с вектором *b* в правой части: $Ax = b$, $A = LU$. Тогда решение СЛАУ находится из системы:

$Ly = b$

$Ux = y$

Обратная матрица через LU-разложение

Сначала находим LU- разложение матрицы A, затем находим L^{-1} и U^{-1}, а потом перемножаем полученные обратные матрицы, чтобы получить обратную исходной матрицы: $A^{-1} = L^{-1} U^{-1}$.

Чтобы найти обратную матрицу A^{-1} нужно решить СЛАУ: $Ly = E$, $UA^{-1} = y$.

Протестируем алгоритм LU-разложения на конкретной матрице:

```
Input matrix A:
 [[-2  4  3]
 [ 0  1 -1]
 [ 3  0 -2]]
```

```
L matrix:
 [[ 1.   0.   0. ]
 [ 0.   1.   0. ]
 [-1.5  6.   1. ]]
```

```
U matrix:
 [[-2.   4.   3. ]
 [ 0.   1.  -1. ]
 [ 0.   0.   8.5]]
```

Обратная матрица с использованием LU-разложения:

```
Inverse matrix A:
 [[ 0.11764706 -0.47058824  0.41176471]
 [ 0.17647059  0.29411765  0.11764706]
 [ 0.17647059 -0.70588235  0.11764706]]
```

Решение системы с использованием LU-разложения:

```
Matrix A:
 [[ 2  1  1]
 [ 1 -1  0]
 [ 3 -1  2]]
Vector b:
 [2, -2, 2]
Solution:
 [-1.  1.  3.]
```

Метод Зейделя

При вычислении очередного *(k+1)* приближения к неизвестному x_i при $i > 1$ используются уже найденные *(k+1)* приближения к неизвестным $x_1, x_2, \dots, x_{i-1}$, а не k, как в методе Якоби.

$$\begin{cases} 10x_1 + x_2 + x_3 = 12, \\ 2x_1 + 10x_2 + x_3 = 13, \\ 2x_1 + 2x_2 + 10x_3 = 14. \end{cases}$$

$\xi = 0,00001$

```
1. [1.2   1.06   0.948]
2. [0.9992   1.00536   0.999088]
3. [0.9995552  1.00018016 1.00005293]
4. [0.99997669 0.99999937 1.00000479]
5. [0.99999958 0.9999996  1.00000016]
6. [1.00000002 0.99999998 1.        ]
[1.00000002 0.99999998 1.        ]
Total iterations: 6
```

Исследование методов на матрицах, число обусловленности которых регулируется за счет изменения диагонального преобладания

$k = 3, \xi = 0,001$

```
Matrix A:
 [[ 2.001 -1.    -1.   ]
 [-1.     2.001 -1.   ]
 [-1.    -1.     2.001]]
Vector b:
 [-3, 8, 2]
Solution:  [2330.9791096  2334.64474671 2332.64560535]
Total iterations: 8308
```

$k = 4, \xi = 0,001$

```
Matrix A:
 [[ 3.0001 -1.     -1.     -1.    ]
 [-1.      3.0001 -1.     -1.    ]
 [-1.     -1.      3.0001 -1.    ]
 [-1.     -1.     -1.      3.0001]]
Vector b:
 [-3, 8, 2, 0]
Solution:  [17491.31268171 17494.06273796 17492.56290045 17492.06303794]
Total iterations: 116329
```

Исследование методов на матрице Гильберта:

$k = 3, \xi = 0,00001$

```
Matrix A:
 [[1.         0.5        0.33333333]
 [0.5        0.33333333 0.25      ]
 [0.33333333 0.25       0.2       ]]
Vector b:
 [2, 3, 1]
Solution:  [ -59.99992741  323.99962952 -299.99965788]
Total iterations: 708
```

$k = 4, \xi = 0,001$

```
Matrix A:
 [[1.         0.5        0.33333333 0.25      ]
 [0.5        0.33333333 0.25       0.2       ]
 [0.33333333 0.25       0.2        0.16666667]
 [0.25       0.2        0.16666667 0.14285714]]
Vector b:
 [4, 1, -2, 0]
Solution:  [   -535.96762307   6119.65216207 -14699.18411816   9519.47845132]
Total iterations: 10099
```

$k = 5, \xi = 0,001$

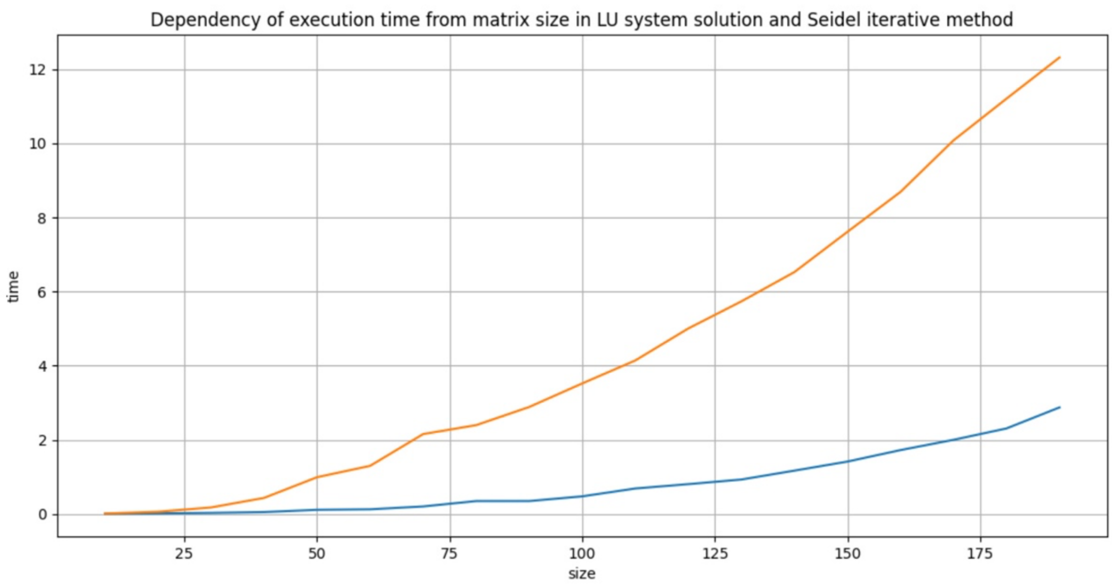
```
Matrix A:
 [[1.         0.5        0.33333333 0.25       0.2       ]
 [0.5        0.33333333 0.25       0.2        0.16666667]
 [0.33333333 0.25       0.2        0.16666667 0.14285714]
 [0.25       0.2        0.16666667 0.14285714 0.125     ]
 [0.2        0.16666667 0.14285714 0.125      0.11111111]]
Vector b:
 [1, 2, -1, 4, 3]
Solution:  [  -5334.84557754   97917.16529501 -416417.92315847   622141.91907413
 -301761.21222902]
Total iterations: 246779
```

Анализ эффективности методов на матрицах разной размерности

```
K =  10
Lu time:  0.008000612258911133
Seidel time:  0.012983322143554688

K =  50
Lu time:  0.12401628494262695
Seidel time:  1.2640714645385742

K =  100
Lu time:  0.8116505146026611
Seidel time:  5.486283779144287
```



метод LU-разложения

метод Зейделя

Выводы

Таким образом, мы реализовали LU-разложение, с использованием которого нашли нахождение обратной матрицы, решение систем. Также реализовали итерационный метод решения СЛАУ - метод Зейделя и проанализировали его на разных матрицах, по итогам наблюдений установили, что данный алгоритм может работать не совсем корректно на некоторых матрицах. Метод Зейделя предназначен для матриц со строгим диагональным преобладанием. Также проанализировали эффективность методов на матрицах разной размерности и выяснили, что прямой метод во всех случаях работает быстрее.