МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Департамент математического и компьютерного моделирования

ОТЧЕТ
по лабораторной работе

по дисциплине «Вычислительная математика»
по направлению подготовки «01.03.02 Прикладная математика и информатика»
шифр и название направления

образовательная программа «Математика и компьютерные технологии»
название образовательной программы

на тему **«LU-Разложение»**

Выполнил студент гр. Б9122
Пелагеев Даниил Иванович

Проверил
Журавлев Павел Викторович

____
(оценка)

г. Владивосток
2024

Оглавление

Введение

Вычисление решения системы линейных алгебраических уравнений (СЛАУ) является фундаментальной задачей численного анализа и прикладной математики. Методы решения СЛАУ применяются во всех областях науки и техники, начиная с решения уравнений механики сплошной среды и заканчивая экономическими моделями. Одним из часто используемых подходов является разложение матрицы на произведение двух треугольных матриц (LU-разложение). Этот метод позволяет упростить процесс решения и многократно использовать одно и то же разложение при решении СЛАУ с разными векторами правых частей.

В данном отчёте рассматривается реализация и применение LU-разложения для решения заданных систем линейных уравнений. Основное внимание уделено практической реализации, оценке точности полученного решения и анализу невязки.

1 Постановка задачи

Требуется решить задачу нахождения вектора неизвестных x для уравнений:

$$Ax = b,$$

где A — квадратная невырожденная матрица размера $n \times n$, b — заданный вектор правых частей, а x — искомый вектор. Необходимо:

а) Реализовать на языке Python метод решения СЛАУ с использованием LU-разложения;

б) Проверить работоспособность реализации на нескольких тестовых системах;

в) Оценить погрешность полученно решения.

2 Теоретическое описание метода

2.1 LU-разложение

LU-разложение[1] представляет собой факторизацию квадратной матрицы A в произведение нижнетреугольной матрицы L и верхнетреугольной матрицы U:

$$A = LU,$$

где матрица L имеет диагональные элементы, равные 1, а U — верхнетреугольная.

Тот же метод Гаусса, но записанный через разложение матриц:

2.2 Определения матриц

$$D_1 := \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -d_{21} & 1 & 0 & \dots & 0 \\ -d_{31} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -d_{n1} & 0 & 0 & \dots & 1 \end{pmatrix};$$

$$A_1 = D_1 A.$$

Элементы d_{ij} вычисляются так, чтобы занулить элементы ниже главной диагонали в k-м столбце преобразованной матрицы. Это означает:

$$d_{ij} = \frac{a_{ij}}{a_{jj}}, \quad i > j,\ j = k,$$

где a_{jj} — текущий диагональный элемент. Знак минус в D_k возникает из-за того, что элементарные преобразования для зануления элементов столбца выполняются с использованием вычитания строк.

$$D_2 := \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & -d_{32} & 1 & \dots & 0 \\ 0 & -d_{42} & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \end{pmatrix};$$

$$A_2 = D_2 A_1.$$

Последовательное преобразование матриц:

$$A_{n-1} = D_{n-1} A_{n-2}.$$

$$U = A_{n-1} = D_{n-1} A_{n-2} = D_{n-1} D_{n-2} A_{n-3} = \dots$$

$$U = D_{n-1} D_{n-2} \dots D_1 A.$$

$$D = D_{n-1} D_{n-2} \dots D_1.$$

$$U = DA.$$

Структура матрицы U:

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{pmatrix}.$$

$$L := D^{-1}.$$

Структура матрицы L:

$$L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ d_{21} & 1 & 0 & \dots & 0 \\ d_{31} & d_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \dots & 1 \end{pmatrix};$$

Тогда:

$$A = LU.$$

Далее решение исходной системы линейных алгебраических уравнений (СЛАУ) $Ax = b$ сводится к двум простым шагам:

а) Решается вспомогательная система $Ly = b$ методом прямого хода;

б) Решается система $Ux = y$ методом обратного хода.

2.3 Ограничения и замечания

— Матрица A должна быть невырожденной, то есть $\det(A) \neq 0$;

— При отсутствии выбора главного элемента (Pivoting) могут возникать численные проблемы. Для более устойчивого решения на практике часто используют LUP-разложение[2];

— В данной работе рассматривается базовая реализация без выбора ведущих элементов.

3 Практическая часть

3.1 Описание реализации

В ходе практической части был реализован следующий алгоритм:

Функция `lu(A, b)`, где:

— На вход подаются матрица A и вектор b;

— Выполняется построение матриц L и U;

— Решается система $Ly = b$, затем $Ux = y$;

— Возвращается решение x.

3.2 Тестовые примеры

Для тестирования метода были использованы следующие матрицы:

Пример 1.

$$A_1 = \begin{bmatrix} 0.411 & 0.421 & -0.333 & 0.313 & -0.141 & -0.381 & 0.245 \\ 0.241 & 0.705 & 0.139 & -0.409 & 0.321 & 0.0625 & 0.101 \\ 0.123 & -0.239 & 0.502 & 0.901 & 0.243 & 0.819 & 0.321 \\ 0.413 & 0.309 & 0.801 & 0.865 & 0.423 & 0.118 & 0.183 \\ 0.241 & -0.221 & -0.243 & 0.134 & 1.274 & 0.712 & 0.423 \\ 0.281 & 0.525 & 0.719 & 0.118 & -0.974 & 0.808 & 0.923 \\ 0.246 & -0.301 & 0.231 & 0.813 & -0.702 & 1.223 & 1.105 \end{bmatrix}.$$

$$\mathbf{b}_1 = \begin{bmatrix} 0.096 \\ 1.252 \\ 1.024 \\ 1.023 \\ 1.155 \\ 1.937 \\ 1.673 \end{bmatrix}.$$

Результат 1.

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.586375 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.29927 & -0.79669 & 1 & 0 & 0 & 0 & 0 \\ 1.00487 & -0.24894 & 1.40425 & 1 & 0 & 0 & 0 \\ 0.586375 & -1.02123 & 0.33829 & 11.3252 & 1 & 0 & 0 \\ 0.683698 & 0.517669 & 0.891324 & 1.29885 & -0.376198 & 1 & 0 \\ 0.59854 & -1.20703 & 0.960619 & 6.06925 & 0.114446 & 0.926633 & 1 \end{bmatrix}.$$

$$U_1 = \begin{bmatrix} 0.411 & 0.421 & -0.333 & 0.313 & -0.141 & -0.381 & 0.245 \\ 0 & 0.458136 & 0.334263 & -0.592535 & 0.403679 & 0.285909 & -0.0426618 \\ 0 & 0 & 0.867961 & 0.335261 & 0.606804 & 1.1608 & 0.213691 \\ 0 & 0 & 0 & -0.0678191 & -0.186925 & -1.05803 & -0.373887 \\ 0 & 0 & 0 & 0 & 3.68062 & 12.8171 & 4.39784 \\ 0 & 0 & 0 & 0 & 0 & 6.08181 & 2.72719 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0596219 \end{bmatrix}.$$

$$\mathbf{x}_1 = \begin{bmatrix} 11.09196963 \\ -2.51573632 \\ 0.72098648 \\ -2.54467447 \\ -1.60482658 \\ 3.62397366 \\ -4.94958981 \end{bmatrix}.$$

Невязка решения:$\|\mathbf{b}_1 - A_1\mathbf{x}_1\|_2 = 8.18898849141959 \times 10^{-15}$.

Пример 2.

$$A_2 = \begin{bmatrix} 1.231 & -0.231 & 0.613 & -0.314 & 0.281 & 0.271 & -0.301 \\ -0.421 & 1.052 & 0.128 & 0.523 & -0.328 & 0.813 & 0.291 \\ 0.319 & -0.123 & 1.402 & 0.319 & 0.714 & -0.213 & 0.134 \\ -0.314 & 0.413 & 0.217 & 1.118 & 0.412 & -0.319 & 0.284 \\ 0.213 & -0.328 & 0.913 & 0.251 & 1.105 & 0.114 & -0.213 \\ -0.231 & 0.412 & -0.214 & 0.319 & 0.522 & 1.312 & 0.319 \\ 0.142 & 0.318 & -0.231 & 0.216 & -0.213 & 0.421 & 1.089 \end{bmatrix}.$$

$$\mathbf{b}_2 = \begin{bmatrix} 0.823 \\ 1.231 \\ 0.912 \\ 1.132 \\ 0.926 \\ 1.543 \\ 0.874 \end{bmatrix}.$$

Результат 2.

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.341998 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.259139 & -0.0648911 & 1 & 0 & 0 & 0 & 0 \\ -0.255077 & 0.363903 & 0.198008 & 1 & 0 & 0 & 0 \\ 0.17303 & -0.296023 & 0.716871 & 0.15213 & 1 & 0 & 0 \\ -0.187652 & 0.378883 & -0.179357 & 0.223497 & 1.43352 & 1 & 0 \\ 0.115353 & 0.354211 & -0.333035 & 0.308368 & -0.194542 & 1.00135 & 1 \end{bmatrix}.$$

$$U_2 = \begin{bmatrix} 1.231 & -0.231 & 0.613 & -0.314 & 0.281 & 0.271 & -0.301 \\ 0 & 0.972998 & 0.337645 & 0.415613 & -0.231898 & 0.905682 & 0.188058 \\ 0 & 0 & 1.26506 & 0.427339 & 0.626134 & -0.224456 & 0.224204 \\ 0 & 0 & 0 & 0.802046 & 0.444085 & -0.53501 & 0.0943924 \\ 0 & 0 & 0 & 0 & 0.471315 & 0.577509 & -0.280334 \\ 0 & 0 & 0 & 0 & 0 & 0.271149 & 0.612246 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.435061 \end{bmatrix}.$$

$$\mathbf{x}_2 = \begin{bmatrix} 0.73933376 \\ 0.64757658 \\ 0.12413174 \\ 0.86438883 \\ 0.57054077 \\ 0.62736704 \\ 0.24100584 \end{bmatrix}.$$

Невязка решения:$\|\mathbf{b}_2 - A_2\mathbf{x}_2\|_2 = 1.5700924586837752 \times 10^{-16}$.

Пример 3.

$$A_3 = \begin{bmatrix} 0.531 & 0.621 & -0.211 & 0.213 & 0.431 & 0.112 & 0.125 \\ 0.331 & 1.105 & 0.112 & -0.209 & 0.621 & -0.221 & 0.211 \\ -0.213 & -0.331 & 1.211 & 0.401 & 0.201 & 0.713 & 0.321 \\ 0.311 & 0.213 & 0.419 & 1.165 & 0.523 & -0.118 & 0.193 \\ 0.121 & -0.121 & -0.321 & 0.214 & 1.372 & 0.512 & 0.413 \\ 0.212 & 0.412 & 0.719 & 0.218 & -0.671 & 1.312 & 0.823 \\ -0.126 & -0.213 & 0.314 & 0.713 & -0.812 & 1.002 & 1.015 \end{bmatrix}.$$

$$\mathbf{b}_3 = \begin{bmatrix} 1.012 \\ 0.923 \\ 1.131 \\ 0.874 \\ 1.564 \\ 1.213 \\ 0.932 \end{bmatrix}.$$

Результат 3.

$$L_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.623352 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0.40113 & -0.114081 & 1 & 0 & 0 & 0 & 0 \\ 0.585687 & -0.209935 & 0.514412 & 1 & 0 & 0 & 0 \\ 0.227872 & -0.365662 & -0.159313 & 0.151389 & 1 & 0 & 0 \\ 0.399247 & 0.228539 & 0.647741 & -0.106678 & -0.813005 & 1 & 0 \\ -0.237288 & -0.0914392 & 0.247976 & 0.841546 & -0.61496 & 1.33413 & 1 \end{bmatrix}.$$

$$U_3 = \begin{bmatrix} 0.531 & 0.621 & -0.211 & 0.213 & 0.431 & 0.112 & 0.125 \\ 0 & 0.717898 & 0.243527 & -0.341774 & 0.352335 & -0.290815 & 0.133081 \\ 0 & 0 & 1.15414 & 0.447451 & 0.414082 & 0.72475 & 0.386323 \\ 0 & 0 & 0 & 0.738324 & 0.131528 & -0.617469 & -0.0510018 \\ 0 & 0 & 0 & 0 & 1.44868 & 0.589078 & 0.502446 \\ 0 & 0 & 0 & 0 & 0 & 1.27735 & 0.895493 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.118231 \end{bmatrix}.$$

$$\mathbf{x}_3 = \begin{bmatrix} 9.42193428 \\ -5.26418664 \\ 2.74015438 \\ -3.51368396 \\ 0.44158435 \\ -3.91772088 \\ 6.82449553 \end{bmatrix}.$$

Невязка решения:$\|\mathbf{b}_3 - A_3\mathbf{x}_3\|_2 = 1.683736582170148 \times 10^{-15}$.

Программа показала корректное поведение на заданных примерах, а вычисленные невязки были достаточно малы, что свидетельствует о правильности решения.

Заключение

В данной работе был реализован алгоритм LU-разложения с использованием языка программирования Python и библиотеки NumPy. Алгоритм успешно применён для решения системы линейных уравнений, позволяя разложить исходную матрицу на две треугольные матрицы L и U, что значительно упрощает вычисления.

Полученные результаты демонстрируют, что использование LU-разложения является эффективным методом для решения систем линейных уравнений, особенно при наличии больших матриц.

Список использованных источников

1. *Белов, С.А.* Численные методы линейной алгебры: Лабораторный практикум / С.А. Белов, Н.Ю. Золотых // *Издательство Нижегородского государственного университета.* — 2005. — Дата обращения: 10.12.2024.

2. *Сеченов, П. А.* Сравнение быстродействия численных методов Гаусса и LUP-разложения в задаче нахождения равновесного химического состава / П. А. Сеченов // *Вестник ВГТУ.* — 2023. — no. 2. — Дата обращения: 10.12.2024.

Приложение А   Пример кода

Ниже приведён пример кода на языке Python:

```
import numpy as np
from tabulate import tabulate


def lu(A, b):
    n = len(A)
    L = np.zeros((n, n))
    U = np.zeros((n, n))
    y = np.zeros_like(b)

    for i in range(n):
        for k in range(i, n):
            sum_upper = sum(L[i][j] * U[j][k] for j in range(i))
            U[i][k] = A[i][k] - sum_upper

        L[i][i] = 1
        for k in range(i + 1, n):
            sum_lower = sum(L[k][j] * U[j][i] for j in range(i))
            L[k][i] = (A[k][i] - sum_lower) / U[i][i]

    print(tabulate(L), tabulate(U))

    for i in range(n):
        sum_forward = sum(L[i][j] * y[j] for j in range(i))
        y[i] = b[i] - sum_forward

    x = np.zeros_like(y)

    for i in reversed(range(n)):
        sum_backward = sum(U[i][j] * x[j] for j in range(i + 1, n))
        x[i] = (y[i] - sum_backward) / U[i][i]

    return x

def main():
    A1 = np.array([
        [0.411, 0.421, -0.333, 0.313, -0.141, -0.381, 0.245],
        [0.241, 0.705, 0.139, -0.409, 0.321, 0.0625, 0.101],
        [0.123, -0.239, 0.502, 0.901, 0.243, 0.819, 0.321],
        [0.413, 0.309, 0.801, 0.865, 0.423, 0.118, 0.183],
        [0.241, -0.221, -0.243, 0.134, 1.274, 0.712, 0.423],
        [0.281, 0.525, 0.719, 0.118, -0.974, 0.808, 0.923],
        [0.246, -0.301, 0.231, 0.813, -0.702, 1.223, 1.105],
```

```python
    ])
    b1 = np.array([0.096, 1.252, 1.024, 1.023, 1.155, 1.937, 1.673])

    A2 = np.array([
        [1.231, -0.231, 0.613, -0.314, 0.281, 0.271, -0.301],
        [-0.421, 1.052, 0.128, 0.523, -0.328, 0.813, 0.291],
        [0.319, -0.123, 1.402, 0.319, 0.714, -0.213, 0.134],
        [-0.314, 0.413, 0.217, 1.118, 0.412, -0.319, 0.284],
        [0.213, -0.328, 0.913, 0.251, 1.105, 0.114, -0.213],
        [-0.231, 0.412, -0.214, 0.319, 0.522, 1.312, 0.319],
        [0.142, 0.318, -0.231, 0.216, -0.213, 0.421, 1.089],
    ])
    b2 = np.array([0.823, 1.231, 0.912, 1.132, 0.926, 1.543, 0.874])

    A3 = np.array([
        [0.531, 0.621, -0.211, 0.213, 0.431, 0.112, 0.125],
        [0.331, 1.105, 0.112, -0.209, 0.621, -0.221, 0.211],
        [-0.213, -0.331, 1.211, 0.401, 0.201, 0.713, 0.321],
        [0.311, 0.213, 0.419, 1.165, 0.523, -0.118, 0.193],
        [0.121, -0.121, -0.321, 0.214, 1.372, 0.512, 0.413],
        [0.212, 0.412, 0.719, 0.218, -0.671, 1.312, 0.823],
        [-0.126, -0.213, 0.314, 0.713, -0.812, 1.002, 1.015],
    ])
    b3 = np.array([1.012, 0.923, 1.131, 0.874, 1.564, 1.213, 0.932])

    x1 = lu(A1, b1)

    print("Solution of the first system Ax = b:", x1, end="\n\n\n".)

    residual1 = b1 - np.dot(A1, x1)
    residual_norm1 = np.linalg.norm(residual1)
    print("Solution inequality (vector norm b - Ax):", residual_norm1, end="\n\n")

    x2 = lu(A2, b2)

    print("Solution of the second system Ax = b:", x2, end="\n\n\n".)

    residual2 = b2 - np.dot(A2, x2)
    residual_norm2 = np.linalg.norm(residual2)
    print("Solution inequality (vector norm b - Ax):", residual_norm2, end="\n\n")

    x3 = lu(A3, b3)

    print("Solution of the third system Ax = b:", x3, end="\n\n\n".)
```

```

    residual3 = b3 - np.dot(A3, x3)
    residual_norm3 = np.linalg.norm(residual3)
    print("Solution inequality (vector norm b - Ax):", residual_norm3,
        end="\n\n")

if __name__ == "__main__":
    main()
```