МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет» (ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент математического и компьютерного моделирования

ОТЧЕТ по лабораторной работе № 2

по дисциплине

«Вычислительная математика»

Направление подготовки
02.03.01 «Математика и компьютерные науки»

Вариант № 3

Выполнил(а): студент гр. Б9122-02.03.01сцт
Бекболот Отгонцэцэг
Проверил: преподаватель
_____ Ф.И.О.

Владивосток
2024

Цель работы :

Целью данной работы является изучение и реализация метода LU-разложения для решения систем линейных алгебраических уравнений (СЛАУ).

Ход работы:

1. **LU-разложение матрицы A:**

Начнем с выполнения LU-разложения матрицы A. Это означает, что мы представим данную матрицу в виде произведения двух матриц: L (нижняя треугольная) и U (верхняя треугольная). Для достижения этой цели можно использовать алгоритм Гаусса с выбором главного элемента для повышения числовой устойчивости.

2. **Решение системы $L_y = b$:**

Далее необходимо решить вспомогательную систему с использованием нижней треугольной матрицы L. Это делается методом прямой подстановки: начиная с первого уравнения, мы последовательно находим элементы вектора y, используя уже вычисленные значения.

3. **Решение системы $U_x = y$:**

Затем, с найденным вектором y, переходим к решению системы уравнений с верхней треугольной матрицей U. Здесь мы применяем метод обратной подстановки, начиная с последнего уравнения, что позволяет итеративно находить значения вектора решений x .

4. **Сравнение с точными значениями:**

Полученные результаты вектора x следует сопоставить с известными точными решениями, если таковые доступны. Это сравнение поможет оценить точность вычислений, используя метрики, такие как относительная ошибка или норма разности.

5. **Отладка алгоритма и решение дополнительных систем:**

Завершив основные вычисления, необходимо провести тестирование и отладку написанного программного обеспечения для поиска и устранения возможных ошибок. По завершении отладки целесообразно применить разработанный алгоритм для решения других систем линейных алгебраических уравнений (СЛАУ) с целью проверки его универсальности и точности.

Код программы:

```python
import numpy as np

class LU:
    def __init__(self, A):
        self.A = A
        self.L = None
        self.U = None

    def decompose(self):

        n = self.A.shape[0]
        self.L = np.eye(n)
        self.U = np.zeros_like(self.A)

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

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

    def forward_substitution(self, b):

        n = self.L.shape[0]
        y = np.zeros(n)
        for i in range(n):
            y[i] = (b[i] - sum(self.L[i, j] * y[j] for j in range(i))) / self.L[i, i]
        return y

    def backward_substitution(self, y):

        n = self.U.shape[0]
        x = np.zeros(n)
        for i in range(n - 1, -1, -1):
            x[i] = (y[i] - sum(self.U[i, j] * x[j] for j in range(i + 1, n))) / self.U[i, i]
        return x

    def solve(self, b):

        self.decompose()
        y = self.forward_substitution(b)
        x = self.backward_substitution(y)
        return self.L, self.U, x

    def print_results(self, name, x):

        print(f"{name}:")
        print("Матрица L:")
        print(self.L)
        print("\nМатрица U:")
        print(self.U)
        print("\nВектор решений x:")
        print(x)
        print("\n" + "=" * 30 + "\n")


# Определение систем уравнений
A_main = np.array([[5, 2, 3],
                   [1, 6, 1],
                   [3, -4, -2]], dtype=float)

b_main = np.array([3, 5, 8], dtype=float)

A_I = np.array([[13.14, -2.12, 1.17, 0],
                [-2.12, 6.3, -2.45, 0],
                [1.17, -2.45, 4.6, 0],
                [0, 0, 0, 1]], dtype=float)

b_I = np.array([1.27, 2.13, 3.14, 0], dtype=float)

A_II = np.array([[4.31, 0.26, 0.61, 0.27],
                 [0.26, 2.32, 0.18, 0.34],
                 [0.61, 0.18, 3.20, 0.31],
                 [0.27, 0.34, 0.31, 5.17]], dtype=float)

b_II = np.array([1.02, 1.00, 1.34, 1.27], dtype=float)


# Решение систем

# Решение основной системы
lu_main = LU(A_main)
L_main, U_main, x_main = lu_main.solve(b_main)
lu_main.print_results("Основная система", x_main)

# Решение системы I
```

```python

    # Решение основной системы
    lu_main = LU(A_main)
    L_main, U_main, x_main = lu_main.solve(b_main)
    lu_main.print_results("Основная система", x_main)

    # Решение системы I
    lu_I = LU(A_I)
    L_I, U_I, x_I = lu_I.solve(b_I)
    lu_I.print_results("Система I", x_I)

    # Решение системы II
    lu_II = LU(A_II)
    L_II, U_II, x_II = lu_II.solve(b_II)
    lu_II.print_results("Система II", x_II)
```

Полученные результаты выполнение кода:

```
Основная система:
Матрица L:
[[ 1.          0.          0.        ]
 [ 0.2         1.          0.        ]
 [ 0.6        -0.92857143  1.        ]]

Матрица U:
[[ 5.          2.          3.        ]
 [ 0.          5.6         0.4       ]
 [ 0.          0.         -3.42857143]]

Вектор решений x:
[ 2.  1. -3.]

==============================

Система I:
Матрица L:
[[ 1.          0.          0.          0.        ]
 [-0.16133942  1.          0.          0.        ]
 [ 0.0890411  -0.37953137  1.          0.        ]
 [ 0.          0.          0.          1.        ]]

Матрица U:
[[13.14       -2.12        1.17        0.        ]
 [ 0.          5.95796043 -2.26123288  0.        ]
 [ 0.          0.          3.63761311  0.        ]
 [ 0.          0.          0.          1.        ]]

Вектор решений x:
[0.12996615 0.80016894 1.07572903 0.        ]

==============================

Система II:
Матрица L:
[[1.         0.         0.         0.        ]
 [0.06032483 1.         0.         0.        ]
 [0.14153132 0.06214507 1.         0.        ]
 [0.06264501 0.14048089 0.08105905 1.        ]]

Матрица U:
[[4.31       0.26       0.61       0.27      ]
 [0.         2.30431555 0.14320186 0.3237123 ]
 [0.         0.         3.1047666  0.25166942]
 [0.         0.         0.         5.08721037]]

Вектор решений x:
[0.15331773 0.35835406 0.35066487 0.19304791]
```

Вывод:

В результате выполнения метода *LU*-разложения для различных систем линейных уравнений были получены соответствующие матрицы *L* и *U*, а также точные векторы решений.

1. Основная система: Решение было найдено успешно: $x = [2, 1, -3]$, что совпадает с ожидаемыми значениями.
2. Система *I*: Решение *x* оказалось [0.12996615, 0.80016894, 1.07572903, 0], что соответствует данной системе. Значения решений близки к ожидаемым.
3. Система *II*: Решение *x* было найдено как [0.15331773, 0.35835406, 0.35066487, 0.19304791] . Это решение также показало хорошую сходимость.

Эти результаты демонстрируют высокую эффективность метода LU-разложения для решения систем линейных уравнений. Вполне возможно, что для более сложных систем данный метод может стать основным инструментом.