**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

**ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ**

**(Вариант 5)**

*Выполнил студент 3 ПМ*

*Бут Дмитрий*

***Постановка задачи:***

Написать, отладить и выполнить программы решения систем линейных алгебраических уравнений, записанных в векторно-матричной форме и приведенных в таблице. В колонке х\* приведено точное решение. Решить систему методом Гаусса с выбором главного элемента и методом Зейделя.

Оценить погрешности методов.

Для метода Гаусса привести матрицу, приведенную к треугольному виду. Для метода Зейделя - преобразованную матрицу и количество итераций. Показать, что условия сходимости выполнены.

|  |  |  |  |
| --- | --- | --- | --- |
| 5,37 |  |  |  |

**Метод Гаусса.** Он состоит из двух этапов: на первом, называемом прямым ходом, матрица приводится к верхнетреугольному виду методом исключения переменных. Для уменьшения вычислительных ошибок используется выбор главного элемента — строки переставляются так, чтобы в каждом столбце максимальный по модулю элемент становился ведущим. На втором этапе, обратном ходе, решение системы находится путем последовательного подстановочного вычисления, начиная с последнего уравнения. Метод Гаусса является точным, но его сложность O(n³) делает его менее эффективным для больших систем.

Матрица, приведённая к верхнетреугольному виду:

Решение методом Гаусса: .

Погрешности метода Гаусса:

**Метод Зейделя.** Метод Зейделя — это итерационный метод, который постепенно уточняет значения переменных, используя уже обновленные значения на каждом шаге. Для сходимости метода матрицу системы желательно преобразовать так, чтобы в ней преобладали диагональные элементы. Затем выполняются последовательные итерации: каждое новое значение вычисляется с использованием уже найденных значений , что делает процесс быстрее, чем в методе простой итерации. Алгоритм завершается, когда разница между текущими и предыдущими значениями становится меньше заданной точности. При вычислении была использована точность . Метод Зейделя может быть значительно быстрее метода Гаусса, особенно для больших разреженных систем, но его сходимость без предварительного преобразования матрицы не всегда гарантирована.

Преобразованная матрица:

|  |  |
| --- | --- |
| A | b |
| |  |  |  |  | | --- | --- | --- | --- | | 4,77 | 1,03 | 0,58 | -1,17 | | 1,14 | -5,03 | 3,01 | 0,12 | | 2,11 | 1,17 | 4,89 | 0,88 | | 0,14 | -0,18 | 1,28 | 2,1 | | 12.19  -10.91  0.79 |

В принятых обозначениях D означает матрицу, у которой на главной диагонали стоят соответствующие элементы матрицы A, а все остальные нули; тогда как матрицы  и ULсодержат верхнюю и нижнюю треугольные части A, на главной диагонали которых нули.

Пусть , где , где – матрица, обратная к . Тогда при любом выборе начального приближения метод Зейделя сходится.

Имеем

|  |  |  |
| --- | --- | --- |
| L | D | U |
|  |  |  |

|  |
| --- |
|  |
|  |

Таким образом, условие сходимости метода Зейделя выполняется

Число итераций: 9

Решение, полученное методом Зейделя:

|  |
| --- |
| x |
| 2.00000507  2.00000093  -1.00000197  -0.99999906 |

Погрешности метода Зейделя:

**ПРИЛОЖЕНИЕ**

***Программа реализации метода Гаусса с выбором главного элемента***

import numpy as np

A = np.array([[1.14, -5.03, 3.01, 0.12],

                 [4.77, 1.03, 0.58, -1.17],

                 [2.11, 1.17, 4.89, 0.88],

                 [0.14, -0.18, 1.28, 2.10]])

b = np.array([-10.91, 12.19, 0.79, -3.46], dtype=float)

x\_exact = np.array([2, 2, -1, -1], dtype=float)

def gauss\_with\_pivoting(A, b):

    n = len(b)

    A = A.astype(float)

    b = b.astype(float)

    # Прямой ход с выбором главного элемента

    for k in range(n):

        max\_row = np.argmax(np.abs(A[k:n, k])) + k

        A[[k, max\_row]] = A[[max\_row, k]]  # Переставляем строки

        b[[k, max\_row]] = b[[max\_row, k]]

        for i in range(k+1, n):

            factor = A[i, k] / A[k, k]

            A[i, k:] -= factor \* A[k, k:]

            b[i] -= factor \* b[k]

    # Обратный ход

    x = np.zeros(n)

    for i in range(n-1, -1, -1):

        x[i] = (b[i] - np.dot(A[i, i+1:], x[i+1:])) / A[i, i]

    return x, A

# Решение методом Гаусса

x\_gauss, A\_triangular = gauss\_with\_pivoting(A.copy(), b.copy())

abs\_error\_gauss = np.linalg.norm(x\_gauss - x\_exact, ord=np.inf)

rel\_error\_gauss = abs\_error\_gauss/np.linalg.norm(x\_gauss)\*100

# Вывод результатов

print("Решение методом Гаусса:", x\_gauss)

print("Матрица после приведения к верхнетреугольному виду:\n", A\_triangular)

print(f"Погрешности метода Гаусса:", abs\_error\_gauss, rel\_error\_gauss)

***Программы реализации метода Зейделя***

import numpy as np

A = np.array([[1.14, -5.03, 3.01, 0.12],

                 [4.77, 1.03, 0.58, -1.17],

                 [2.11, 1.17, 4.89, 0.88],

                 [0.14, -0.18, 1.28, 2.10]])

b = np.array([-10.91, 12.19, 0.79, -3.46], dtype=float)

x\_exact = np.array([2, 2, -1, -1], dtype=float)

def make\_strongly\_diagonally\_dominant(A, b):

    n = len(A)

    for i in range(n):

        row = max(range(i, n), key=lambda r: abs(A[r, i]) / sum(abs(A[r, :])) if sum(abs(A[r, :])) != 0 else 0)

        if row != i:

            A[[i, row]] = A[[row, i]]

            b[[i, row]] = b[[row, i]]

    return A, b

def seidel\_method(A, b, tol=1e-4, max\_iter=100):

    A, b = make\_strongly\_diagonally\_dominant(A, b)

    n = len(b)

    x = np.zeros(n)

    x\_old = np.copy(x)

    for iteration in range(max\_iter):

        for i in range(n):

            sum1 = sum(A[i][j] \* x[j] for j in range(i))

            sum2 = sum(A[i][j] \* x\_old[j] for j in range(i + 1, n))

            x[i] = (b[i] - sum1 - sum2) / A[i][i]

        if np.linalg.norm(x - x\_old, ord=np.inf) < tol:

            return x, iteration + 1

        x\_old = np.copy(x)

    return x, max\_iter

# Решение методом Зейделя

x\_seidel, iterations = seidel\_method(A.copy(), b.copy())

abs\_error\_seidel = np.linalg.norm(x\_seidel - x\_exact, ord=np.inf)

rel\_error\_seidel = abs\_error\_seidel/np.linalg.norm(x\_seidel)\*100

# Проверка условия сходимости

strongA, strongB = make\_strongly\_diagonally\_dominant(A, b)

D = np.diag(np.diag(strongA))

L = np.tril(strongA, -1)

U = np.triu(strongA, 1)

B = -np.linalg.inv(D + L) @ U

spectral\_radius = max(abs(np.linalg.eigvals(B)))

convergence = spectral\_radius < 1

print("\nРешение методом Зейделя:", x\_seidel)

print("Число итераций:", iterations)

print("Погрешности метода Зейделя:", abs\_error\_seidel, rel\_error\_seidel)

print("\nПреобразованная матрица для метода Зейделя:\n", strongA, strongB)

print(f"Спектральный радиус матрицы B:{spectral\_radius:.3f}")

print("Условие сходимости выполнено:", convergence)

print(f'{D=}')

print(f'{L=}')

print(f'{U=}')

print(f'{B=}')