

4. Численные методы решения систем линейных уравнений. Метод Гаусса последовательного исключения

Методы решения систем линейных алгебраических уравнений можно разделить на две группы: прямые и итерационные.

Прямые методы используют конечные соотношения (формулы) для вычисления неизвестных. Они дают решение после выполнения заранее известного числа операций.

Итерационные методы – это методы последовательных приближений. В них необходимо задать некоторое приближенное решение – начальное приближение. После этого с помощью некоторого алгоритма проводится один цикл вычислений, называемый итерацией. В результате итерации находят новое приближение. Итерации проводятся до получения решения с требуемой точностью.

Одним из наиболее популярных прямых методов решения СЛУ является Метод Гаусса последовательного исключения (Классический метод Гаусса).

Метод Гаусса последовательного исключения

Суть классического метода Гаусса заключается в том, чтобы на основе коэффициентов при переменных в исходной СЛУ составить расширенную матрицу коэффициентов, и, при помощи элементарных линейных преобразований векторов типа сдвига, масштабирования и перестановки, привести матрицу к треугольному виду.

$$AX = B \Rightarrow \tilde{A}X = \tilde{B}$$

Помимо решения СЛУ, метод Гаусса можно также использовать для нахождения ранга матрицы, определения совместности системы.

К минусам данного метода можно отнести вычислительную неустойчивость для плохо обусловленных матриц коэффициентов. Однако, такие матрицы встречаются относительно нечасто. Данная проблема частично решается различными модификациями классического метода Гаусса.

Решение СЛУ методом Гаусса

Решение СЛУ методом Гаусса последовательного исключения можно разделить на два этапа:

1. Этап прямого хода;
2. Этап обратного хода.

Рассмотрим эти этапы и используемые в них формулы по отдельности.

Этап прямого хода

1. К матрице коэффициентов A присоединяют столбец свободных членов, получая расширенную матрицу коэффициентов;
2. При помощи элементарных преобразований, матрицу приводят к эквивалентной ей треугольной матрице \tilde{A} с единичными диагональными элементами.

Для преобразования элементов на этом этапе используют следующие формулы:

1. $\tilde{a}_{1j} = \frac{a_{1j}}{a_{11}}$, где $j = 1 \div (n+1)$;
2. $\tilde{a}_{kj} = a_{kj} - \tilde{a}_{1j} \cdot a_{k1}$, где $k = 2 \div n, j = 1 \div (n+1)$;
3. $i = 1 \div (n-1)$.

Или в общем виде:

1. $\tilde{a}_{ij} = \frac{a_{ij}}{a_{ii}}$;
2. $\tilde{a}_{kj} = a_{kj} - \tilde{a}_{ij} \cdot a_{ki}$;
3. $i = 1 \div (n-1)$;
4. $k = (i+1) \div n$;
5. $j = i \div (n+1)$,

где \tilde{a} - текущее значение a ; a - предыдущее значение a ; a_{ii} - ведущий элемент; a_{ki} - коэффициент преобразования; ведущая строка определяется индексом.

Этап обратного хода

На этом этапе происходит подстановка полученных на этапе прямого хода коэффициентов в исходную СЛУ и находятся корни уравнений.

На этапе обратного хода используют следующие рекуррентные формулы:

$$x_n = \frac{a_{n(n+1)}}{\tilde{a}_{nn}}; x_i = a_{i(n+1)} - \sum_{j=i+1}^n \tilde{a}_{ij} x_j,$$

где i изменяется в обратном порядке $i = (n-1) \div 1$.

Постановка и решение задачи с применением классического метода Гаусса

Задача

Составить программу для решения СЛУ классическим методом Гаусса. Решить данную систему линейных уравнений с помощью программы. Решить СЛУ методом Гаусса вручную и сверить результаты. Система линейных уравнений:

$$\begin{cases} x + y + z = 6 \\ x - y + 2z = 5 \\ 2x - y - z = -3 \end{cases}$$

Решение

Для решения поставленной задачи напомним программу на языке Python.

Опишем вспомогательные функции для печати в консоль матрицы.

```
import typing as tp

from tabulate import tabulate

MatrixRow = tp.List[tp.Union[int, float]]
Matrix = tp.List[MatrixRow]

def output_matrix(matrix: Matrix, digits: int = 4) -> None:
    """Вывод матрицы"""
    matrix = [[round(i, digits) for i in row] for row in matrix]
    print(tabulate(matrix))
```

Реализуем этап прямого хода.

```
def gauss_method(matrix: Matrix) -> Matrix:
    """Метод Гаусса. Алгоритм последовательного исключения неизвестных"""
    n = len(matrix)

    for k in range(n - 1):
        matrix[k] = [i / matrix[k][k] for i in matrix[k]]

        for i in range(k + 1, n):
            div = (matrix[i][k] / matrix[k][k]) * (-1)
            l = [i * div for i in matrix[k]]
            matrix[i] = [
                element1 + element2 for (element1, element2) in zip(matrix[i], l)
            ]

    matrix[n - 1] = [i / matrix[n - 1][n - 1] for i in matrix[n - 1]]
    return matrix
```

Опишем этап обратного хода.

```
def backward_iteration(matrix: Matrix) -> MatrixRow:
    """Обратный ход"""
    l = []
    n = len(matrix) - 1

    for i in range(n, -1, -1):
        res = (matrix[i][-1] - sum(matrix[i][i+1:-1])) / matrix[i][i]
        l.append(res)

        for j in range(i - 1, -1, -1):
            matrix[j][i] = matrix[j][i] * res

    return l[::-1]
```

Добавим точку входа в программу.

```
if __name__ == "__main__":
    matrix = [
        [1, 1, 1, 6],
        [1, -1, 2, 5],
        [2, -1, -1, -3],
    ]
    print("Расширенная матрица коэффициентов:")
    output_matrix(matrix)

    matrix = gauss_method(matrix)
    print(f'\nПрямой ход. {gauss_method.__doc__}:')
    output_matrix(matrix)

    answers = backward_iteration(matrix)
    print("\nОтвет после применения обратного хода:")
    print("; ".join(f"x{i} = {round(x, 4)}" for i, x in enumerate(answers, 1)))
```

Запустим программу и получим результат.

Расширенная матрица коэффициентов:

```
- - - -
1  1  1  6
1 -1  2  5
2 -1 -1 -3
- - - -
```

Прямой ход. Метод Гаусса. Алгоритм последовательного исключения неизвестных:

```
- - - - -
1  1  1  6
-0  1 -0.5 0.5
-0 -0  1  3
- - - - -
```

Ответ после применения обратного хода:

x1 = 1.0; x2 = 2.0; x3 = 3.0

□

Решим задачу вручную и сравним результаты вычислений.

Задача:

$$\begin{cases} x+y+z=6 \\ x-y+2z=5 \\ 2x-y-z=-3 \end{cases}$$

Прямой ход:

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 1 & -1 & 2 & 5 \\ 2 & -1 & -1 & -3 \end{array} \right) \xrightarrow{\substack{II-I \\ III-2I}} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -2 & 1 & -1 \\ 0 & -3 & -3 & -15 \end{array} \right) \xrightarrow{III \cdot (-1/3)}$$

$$\sim \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -2 & 1 & -1 \\ 0 & 1 & 1 & 5 \end{array} \right) \xrightarrow{II \leftrightarrow III} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & 1 & 5 \\ 0 & -2 & 1 & -1 \end{array} \right) \xrightarrow{IV+2II} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & 1 & 5 \\ 0 & 0 & 3 & 9 \end{array} \right)$$

Обратный ход:

$$\begin{cases} x+y+z=6 \\ y+z=5 \\ 3z=9 \end{cases} ; \begin{cases} x+y+z=6 \\ y=5-z \\ z=3 \end{cases} ; \begin{cases} x=6-2-3 \\ y=2 \\ z=3 \end{cases}$$

$$\begin{cases} x=1 \\ y=2 \\ z=3 \end{cases} ; \quad \underline{\text{Ответ:}} \quad x=1; y=2; z=3.$$

Результаты вычислений совпадают. Решение верное.

Вывод

В ходе решения поставленной задачи была написана программа на языке программирования Python для решения СЛУ классическим методом Гаусса. С помощью полученной программы была решена данная система линейных уравнений и получены её корни. СЛУ также была решена методом Гаусса вручную. Результаты двух решений совпали. Корни СЛУ: $x=1; y=2; z=3$.