Молдавский Государственный Университет
Факультет Математики и Информатики
Департамент Информатики

Лабораторная работа №2
по курсу “Calcul Numeric si Metode de Optimizare”

Тема: “Решение систем уравнений методом Якоби и Гаусса-Зейделя”

Выполнил: Slavov Constantin,
студент группы I2302
Проверил: I. Verlan,
doctor, conferențiar universitar

Кишинев, 2025

Условие задачи:
- Дана система уравнений:

3.738x1 + 0.195x2 + 0.275x3 + 0.136x4 = 0.815
0.519x1 + 5.002x2 + 0.405x3 + 0.283x4 = 0.191
0.306x1 + 0.381x2 + 4.812x3 + 0.418x4 = 0.423
0.272x1 + 0.142x2 + 0.314x3 + 3.935x4 = 0.352

- Решить данную систему уравнений методом Якоби и Гаусса-Зейделя

Ход работы:

Решение методом Якоби:

Метод Якоби является итерационным методом решения системы линейных уравнений вида: Ax = b, где

A - матрица коэффициентов, x - вектор неизвестных, b - вектор свободных членов.

Шаг 1: Преобразование к итерационной форме

Выразим каждую переменную через остальные:

$$x_1^{(k+1)} = \frac{1}{3.738}\left(0.815 - 0.195x_2^{(k)} - 0.275x_3^{(k)} - 0.136x_4^{(k)}\right)$$
$$x_2^{(k+1)} = \frac{1}{5.002}\left(0.191 - 0.519x_1^{(k)} - 0.405x_3^{(k)} - 0.283x_4^{(k)}\right)$$
$$x_3^{(k+1)} = \frac{1}{4.812}\left(0.423 - 0.306x_1^{(k)} - 0.381x_2^{(k)} - 0.418x_4^{(k)}\right)$$
$$x_4^{(k+1)} = \frac{1}{3.935}\left(0.352 - 0.272x_1^{(k)} - 0.142x_2^{(k)} - 0.314x_3^{(k)}\right)$$

Начнем с нулевого приближения:

$$x_1^{(0)} = 0, \quad x_2^{(0)} = 0, \quad x_3^{(0)} = 0, \quad x_4^{(0)} = 0.$$

Далее, в итерациях подставляем значения.

Итерация 1:

Подставляем $x_1^{(0)} = 0, \quad x_2^{(0)} = 0, \quad x_3^{(0)} = 0, \quad x_4^{(0)} = 0.$

$$x_1^{(1)} = \frac{0.815 - (0.195 \cdot 0) - (0.275 \cdot 0) - (0.136 \cdot 0)}{3.738} = \frac{0.815}{3.738} = 0.21803103$$

$$x_2^{(1)} = \frac{0.191 - (0.519 \cdot 0) - (0.405 \cdot 0) - (0.283 \cdot 0)}{5.002} = \frac{0.191}{5.002} = 0.03818473$$

$$x_3^{(1)} = \frac{0.423 - (0.306 \cdot 0) - (0.381 \cdot 0) - (0.418 \cdot 0)}{4.812} = \frac{0.423}{4.812} = 0.08790524$$

$$x_4^{(1)} = \frac{0.352 - (0.272 \cdot 0) - (0.142 \cdot 0) - (0.314 \cdot 0)}{3.935} = \frac{0.352}{3.935} = 0.08945362$$

Результат первой итерации:

$$x_1^{(1)} = 0.2180, \quad x_2^{(1)} = 0.0382, \quad x_3^{(1)} = 0.0879, \quad x_4^{(1)} = 0.0895$$

Итерация 2:

Подставляем значения из первой итерации:

$$x_1^{(2)} = \frac{0.815 - (0.195 \cdot 0.0382) - (0.275 \cdot 0.0879) - (0.136 \cdot 0.0895)}{3.738}$$

$$x_1^{(2)} = \frac{0.815 - 0.0075 - 0.0242 - 0.0122}{3.738} = \frac{0.7711}{3.738} = 0.2063$$

$$x_2^{(2)} = \frac{0.191 - (0.519 \cdot 0.2180) - (0.405 \cdot 0.0879) - (0.283 \cdot 0.0895)}{5.002}$$

$$x_2^{(2)} = \frac{0.191 - 0.1132 - 0.0356 - 0.0253}{5.002} = \frac{0.0169}{5.002} = 0.0034$$

$$x_3^{(2)} = \frac{0.423 - (0.306 \cdot 0.2180) - (0.381 \cdot 0.0382) - (0.418 \cdot 0.0895)}{4.812}$$

$$x_3^{(2)} = \frac{0.423 - 0.0667 - 0.0146 - 0.0374}{4.812} = \frac{0.3043}{4.812} = 0.0632$$

$$x_4^{(2)} = \frac{0.352 - (0.272 \cdot 0.2180) - (0.142 \cdot 0.0382) - (0.314 \cdot 0.0879)}{3.935}$$

$$x_4^{(2)} = \frac{0.352 - 0.0593 - 0.0054 - 0.0276}{3.935} = \frac{0.2597}{3.935} = 0.0660$$

Результат второй итерации:

$$x_1^{(2)} = 0.2063, \quad x_2^{(2)} = 0.0034, \quad x_3^{(2)} = 0.0632, \quad x_4^{(2)} = 0.0660$$

Итерация 3:
Подставляем значения из второй итерации:

$$x_1^{(3)} = \frac{0.815 - (0.195 \cdot 0.0034) - (0.275 \cdot 0.0632) - (0.136 \cdot 0.0660)}{3.738}$$

$$x_1^{(3)} = \frac{0.815 - 0.0007 - 0.0174 - 0.0090}{3.738} = \frac{0.7879}{3.738} = 0.2108$$

$$x_2^{(3)} = \frac{0.191 - (0.519 \cdot 0.2063) - (0.405 \cdot 0.0632) - (0.283 \cdot 0.0660)}{5.002}$$

$$x_2^{(3)} = \frac{0.191 - 0.1071 - 0.0256 - 0.0187}{5.002} = \frac{0.0396}{5.002} = 0.0079$$

$$x_3^{(3)} = \frac{0.423 - (0.306 \cdot 0.2063) - (0.381 \cdot 0.0034) - (0.418 \cdot 0.0660)}{4.812}$$

$$x_3^{(3)} = \frac{0.423 - 0.0631 - 0.0013 - 0.0276}{4.812} = \frac{0.3309}{4.812} = 0.0688$$

$$x_4^{(3)} = \frac{0.352 - (0.272 \cdot 0.2063) - (0.142 \cdot 0.0034) - (0.314 \cdot 0.0632)}{3.935}$$

$$x_4^{(3)} = \frac{0.352 - 0.0561 - 0.0005 - 0.0198}{3.935} = \frac{0.2756}{3.935} = 0.0700$$

Результат третьей итерации:

$$x_1^{(3)} = 0.2108, \quad x_2^{(3)} = 0.0079, \quad x_3^{(3)} = 0.0688, \quad x_4^{(3)} = 0.0700$$

Финальный ответ (после 9 итераций):

$$x_1 = 0.2101, \quad x_2 = 0.0070, \quad x_3 = 0.0680, \quad x_4 = 0.0693$$

Метод Якоби сходится к точному решению через 9 итераций.

Решение методом Гаусса-Зейделя:
Метод Гаусса-Зейделя использует последовательное обновление значений переменных внутри одной итерации.
За основу берем то же уравнение, что и в начале.

Шаг 1: Преобразование к итерационной форме
Представляем начальное приближение:

$$x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0, x_4^{(0)} = 0:$$

$$x_1^{(1)} = \frac{0.815 - (0.195 \cdot 0) - (0.275 \cdot 0) - (0.136 \cdot 0)}{3.738} = 0.2180$$

Подставляем $x_1^{(1)} = 0.2180$ в выражение для x_2:

$$x_2^{(1)} = \frac{0.191 - (0.519 \cdot 0.2180) - (0.405 \cdot 0) - (0.283 \cdot 0)}{5.002} = 0.0156$$

Теперь используем $x_1^{(1)}, x_2^{(1)}$ для вычисления x_3:

$$x_3^{(1)} = \frac{0.423 - (0.306 \cdot 0.2180) - (0.381 \cdot 0.0156) - (0.418 \cdot 0)}{4.812} = 0.0728$$

Используем $x_1^{(1)}, x_2^{(1)}, x_3^{(1)}$ для x_4:

$$x_4^{(1)} = \frac{0.352 - (0.272 \cdot 0.2180) - (0.142 \cdot 0.0156) - (0.314 \cdot 0.0728)}{3.935} = 0.0680$$

Результат первой итерации:

$$x_1^{(1)} = 0.2180, \quad x_2^{(1)} = 0.0156, \quad x_3^{(1)} = 0.0728, \quad x_4^{(1)} = 0.0680$$

То же самое делаем для остальных итераций:

Итерация 2:
Используем значения из первой итерации:

$$x_1^{(2)} = \frac{0.815 - (0.195 \cdot 0.0156) - (0.275 \cdot 0.0728) - (0.136 \cdot 0.0680)}{3.738} = 0.2094$$

$$x_2^{(2)} = \frac{0.191 - (0.519 \cdot 0.2094) - (0.405 \cdot 0.0728) - (0.283 \cdot 0.0680)}{5.002} = 0.0067$$

$$x_3^{(2)} = \frac{0.423 - (0.306 \cdot 0.2094) - (0.381 \cdot 0.0067) - (0.418 \cdot 0.0680)}{4.812} = 0.0682$$

$$x_4^{(2)} = \frac{0.352 - (0.272 \cdot 0.2094) - (0.142 \cdot 0.0067) - (0.314 \cdot 0.0682)}{3.935} = 0.0693$$

Результат второй итерации:

$$x_1^{(2)} = 0.2094, \quad x_2^{(2)} = 0.0067, \quad x_3^{(2)} = 0.0682, \quad x_4^{(2)} = 0.0693$$

Итерация 3:
Используем значения из второй итерации:

$$x_1^{(3)} = \frac{0.815 - (0.195 \cdot 0.0067) - (0.275 \cdot 0.0682) - (0.136 \cdot 0.0693)}{3.738} = 0.2101$$

$$x_2^{(3)} = \frac{0.191 - (0.519 \cdot 0.2101) - (0.405 \cdot 0.0682) - (0.283 \cdot 0.0693)}{5.002} = 0.0069$$

$$x_3^{(3)} = \frac{0.423 - (0.306 \cdot 0.2101) - (0.381 \cdot 0.0069) - (0.418 \cdot 0.0693)}{4.812} = 0.0680$$

$$x_4^{(3)} = \frac{0.352 - (0.272 \cdot 0.2101) - (0.142 \cdot 0.0069) - (0.314 \cdot 0.0680)}{3.935} = 0.0693$$

Результат третьей итерации:

$$x_1^{(3)} = 0.2101, \quad x_2^{(3)} = 0.0069, \quad x_3^{(3)} = 0.0680, \quad x_4^{(3)} = 0.0693$$

Финальный ответ (после 6 итераций):

$$x_1 = 0.2101, \quad x_2 = 0.0070, \quad x_3 = 0.0680, \quad x_4 = 0.0693$$

Метод Гаусса-Зейделя сходится быстрее, чем метод Якоби и сходится за 6 итераций.

*Реализация решения данных уравнений на примере языка программирования **Python**.*

Для метода Якоби:

```
def jacobi_method(A, b, tol=1e-6, max_iterations=100):
    n = len(A)
    x = [0.0 for _ in range(n)]
    x_new = [0.0 for _ in range(n)]

    for iteration in range(max_iterations):
        for i in range(n):
            s = sum(A[i][j] * x[j] for j in range(n) if j != i)
            x_new[i] = (b[i] - s) / A[i][i]

        # Проверка на сходимость
        if max(abs(x_new[i] - x[i]) for i in range(n)) < tol:
            break
```

```
x = x_new[:]
    return x


# Пример использования
A = [
    [3.738, 0.195, 0.275, 0.136],
    [0.519, 5.002, 0.405, 0.283],
    [0.306, 0.381, 4.812, 0.418],
    [0.272, 0.142, 0.314, 3.935]
]

b = [0.815, 0.191, 0.423, 0.352]

result = jacobi_method(A, b)
print("Решение методом Якоби:", result)
```

Данный код реализует метод Якоби для решения системы линейных уравнений.
На каждом шаге вычисляется новое значение каждой переменной, **используя только значения из предыдущей итерации**.
Процесс повторяется, пока изменения между итерациями не станут меньше заданной точности.
Метод прост, но обычно сходится медленнее.

Вывод результата кода на экране:

```
Решение методом Якоби: [0.21014735144605226, 0.006957922930841733, 0.06797482270680735,
0.06925199413810146]
```

Получается тот же ответ, что и после всех произведенные выше итераций.

Для метода Гаусса-Зейделя:

```
def gauss_seidel_method(A, b, tol=1e-6, max_iterations=100):
    n = len(A)
    x = [0.0 for _ in range(n)]

    for iteration in range(max_iterations):
        x_old = x[:]
        for i in range(n):
            s = sum(A[i][j] * x[j] for j in range(n) if j != i)
            x[i] = (b[i] - s) / A[i][i]
```

```
        if max(abs(x[i] - x_old[i]) for i in range(n)) < tol:
            break
    return x

# Пример использования
A = [
    [3.738, 0.195, 0.275, 0.136],
    [0.519, 5.002, 0.405, 0.283],
    [0.306, 0.381, 4.812, 0.418],
    [0.272, 0.142, 0.314, 3.935]
]

b = [0.815, 0.191, 0.423, 0.352]

result = gauss_seidel_method(A, b)
print("Решение методом Гаусса-Зейделя:", result)
```

Этот код решает ту же систему уравнений, но по методу Гаусса-Зейделя.
В отличие от Якоби, **как только вычисляется новая переменная, она сразу используется для вычисления следующих переменных в текущей итерации**.
Благодаря этому метод часто сходится быстрее.

Вывод результата кода на экране:

```
Решение методом Гаусса-Зейделя: [0.2101475855858421, 0.006958234582366248,
0.06797512723785827, 0.06925224840083036]
```

Получается тот же ответ, что и после всех произведенные выше итераций.