Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ "МЭИ"

Институт информационных и вычислительных технологий

Кафедра математического и компьютерного моделирования

Курсовая работа

по дисциплине "Методы вычислительной математики"

Студент: Воронцов А.А. гр А-14-21
Преподаватель: Вестфальский А.Е.

Москва 2024

1 Постановка задачи

Рассмотрим задачу нахождения собственных значений симметричной трёхдиагональной матрицы T размера $n \times n$, элементы которой заданы главной диагональю $a = \{a_1, a_2, \ldots, a_n\}$ и поддиагональной диагональю $b = \{b_1, b_2, \ldots, b_{n-1}\}$.

Задача формулируется как нахождение собственных значений λ, удовлетворяющих уравнению:

$$Tx = \lambda x,$$

где T — симметричная трёхдиагональная матрица, x — собственный вектор.

$$T = \begin{pmatrix} a_1 & b_1 & 0 & \ldots & 0 \\ b_1 & a_2 & b_2 & \ldots & 0 \\ 0 & b_2 & a_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & b_{n-1} \\ 0 & 0 & 0 & b_{n-1} & a_n \end{pmatrix},$$

алгоритм позволяет последовательно находить все собственные значения $\lambda_1, \lambda_2, \ldots, \lambda_n$ с заданной точностью ε.

Описание алгоритма

1. Определяем начальный интервал $[y, z]$:

Используем теорему о кругах Гершгорина или другой метод, чтобы определить начальный интервал, содержащий все собственные значения матрицы T.

2. Запускаем метод бисекции:

1. Пока длина интервала $|z - y|$ больше заданной точности ε:

   (1) Вычисляем середину интервала:
   $$x = \frac{y + z}{2}$$

   (2) Строим последовательность многочленов для x:
   $$\begin{aligned} &p_0(x) = 1, \\ &p_1(x) = a_1 - x, \\ &p_r(x) = (a_r - x)p_{r-1}(x) - b_{r-1}^2 p_{r-2}(x), \quad \text{для } r = 2, 3, \ldots, n. \end{aligned}$$

   (3) Подсчитываем число перемен знака в последовательности Штурма:
   - Получаем набор значений последовательности $\{p_0(x), p_1(x), p_2(x), \ldots, p_n(x)\}$.
   - Определяем количество перемен знака в этой последовательности.
   - Пусть результат подсчёта числа перемен знака обозначим как $a(x)$. Число $a(x)$ — это количество собственных значений матрицы T, которые меньше x.

   (4) Сравниваем $a(x)$ с k:
   - Если $a(x) \geq k$, то в интервале $[y, x]$ находится как минимум k-е собственное значение (и, возможно, больше). Обновляем верхнюю границу интервала:
   $$z = x.$$
   - Если $a(x) < k$, то в интервале $[x, z]$ содержится k-е собственное значение (и, возможно, другие, большие по порядку). Обновляем нижнюю границу интервала:
   $$y = x.$$

3. Завершение:

Когда длина интервала $|z - y|$ станет меньше требуемой точности ε, середина интервала будет приближением к k-му собственному значению:

$$\lambda = \frac{y + z}{2}.$$

Алгоритм позволяет найти k-е собственное значение симметричной матрицы, сужая интервал и используя последовательность Штурма для подсчёта количества собственных значений ниже заданного порога.

Теория и подробности были объяснены в устном докладе.

Тесты

Тест№ 1

- трёхдиагональная матрица T:

$$\begin{bmatrix} 1 & 5.6 & 0 & 0 & 0 \\ 5.6 & 2 & 6.2 & 0 & 0 \\ 0 & 6.2 & 3 & 6.1 & 0 \\ 0 & 0 & 6.1 & 4 & 6.3 \\ 0 & 0 & 0 & 6.3 & 5 \end{bmatrix}$$

Результаты вычислений

- Собственные значения, найденные методом бисекции:

$$[-7.68254, -2.95669, 2.82624, 8.91326, 13.89972]$$

- Собственные значения, найденные методом `eigh.Scipy`:

$$[-7.68254 - 2.95669, 2.826242, 8.91326, 13.89972]$$

Вывод

Метод бисекции и функция `eigh` из библиотеки `scipy` дают идентичные результаты, что подтверждает правильность вычислений.

Тест№ 2

- Полная трёхдиагональная матрица T (размер 100×100):

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 2 & 0 & \dots & 0 \\ 0 & 0 & 3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 100 \end{bmatrix}$$

Результаты вычислений

- Собственные значения :

$$[1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0,$$

$$21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,$$

$$40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0, 50.0, 51.0, 52.0, 53.0, 54.0, 55.0, 56.0, 57.0, 58.0,$$

$$59.0, 60.0, 61.0, 62.0, 63.0, 64.0, 65.0, 66.0, 67.0, 68.0, 69.0, 70.0, 71.0, 72.0, 73.0, 74.0, 75.0, 76.0, 77.0,$$

$$78.0, 79.0, 80.0, 81.0, 82.0, 83.0, 84.0, 85.0, 86.0, 87.0, 88.0, 89.0, 90.0, 91.0, 92.0, 93.0, 94.0, 95.0, 96.0,$$

$$97.0, 98.0, 99.0, 100.0]$$

Вывод

Результаты совпадают с ожидаемыми собственными значениями диагональной матрицы, что подтверждает корректность вычислений.

Заключение

В процессе работы для трёхдиагональной матрицы T была успешно реализована процедура вычисления собственных значений методом бисекции. Алгоритм показал корректные результаты, соответствующие теоретическим ожиданиям.

Программные коды

```
import numpy as np
from scipy.linalg import eigh

def create_matrix(a, b):
    """
    Создаёт полную симметричную трёхдиагональную матрицу из диагоналей a и b.
    a - главная диагональ, b - поддиагональная диагональ.
    """
    n = len(a)
    T = np.zeros((n, n))  # Создаём пустую матрицу

    # Заполняем главную диагональ
    for i in range(n):
        T[i, i] = a[i]

    # Заполняем поддиагональ и наддиагональ
    for i in range(n-1):
        T[i, i+1] = b[i]
        T[i+1, i] = b[i]  # Симметрия

    return T

def sturm_sequence_count(a, b, x):
    """
    Подсчитывает число перемен знаков в последовательности Штурма
    для трехдиагональной симметричной матрицы с главной диагональю a и поддиагональю b.
    """
    n = len(a)
    p = [1, a[0] - x]  # Начальные условия

    for i in range(1, n):
        p_next = (a[i] - x) * p[-1] - (b[i-1]**2) * p[-2]
        p.append(p_next)

    # Подсчитываем число перемен знаков
    count = 0
    for i in range(1, len(p)):
        if p[i-1] * p[i] < 0:
            count += 1
    return count

def gershgorin_bounds(a, b):
    """
    Оценивает начальный интервал [y, z] для собственных значений с помощью теоремы Гершгорина.
```

```
    a - главная диагональ, b - поддиагональные элементы.
    """
    n = len(a)
    y = min(a[i] - (abs(b[i-1]) if i > 0 else 0) - (abs(b[i]) if i < n-1 else 0) for i in range(n))
    z = max(a[i] + (abs(b[i-1]) if i > 0 else 0) + (abs(b[i]) if i < n-1 else 0) for i in range(n))
    return y, z

def bisection_method(a, b, k, epsilon=1e-6):
    """
    Метод бисекции для нахождения k-го собственного значения трехдиагональной симметричной матрицы.
    a - главная диагональ, b - поддиагональные элементы.
    """
    # Используем теорему Гершгорина для нахождения начального интервала
    y, z = gershgorin_bounds(a, b)

    while abs(z - y) > epsilon:
        x = (y + z) / 2
        count = sturm_sequence_count(a, b, x)

        if count >= k:
            z = x
        else:
            y = x

    return (y + z) / 2

def generate_symmetric_tridiagonal_matrix(n, low=-10, high=10):
    """
    Генерирует случайную симметричную трехдиагональную матрицу размера n x n.
    Возвращает главную диагональ a и поддиагональную диагональ b.
    """
    a = np.random.uniform(low, high, n)       # Главная диагональ
    b = np.random.uniform(low, high, n-1)     # Поддиагональные элементы
    a= np.array([ 1.0,2.0,3,4,5])
    b=np.array([5.6,6.2,6.1,6.3])

    a=np.zeros(n)

    for i in range(len(a)):
        a[i]=a[i-1]+1
    b=np.zeros(n)

    return a, b


np.random.seed(42)  # Фиксируем случайность для воспроизводимости

    # Генерация симметричной трёхдиагональной матрицы
n = 100  # Размер матрицы
a, b = generate_symmetric_tridiagonal_matrix(n)
T = create_matrix(a, b)  # Полная матрица для наглядности

print("Полная трёхдиагональная матрица T:")
print(T)

print("\nГлавная диагональ a:")
```

```
print(a)
print("Поддиагональная диагональ b (симметричная):")
print(b)

    # Нахождение всех собственных значений методом бисекции
eigenvalues_bisection = []
for k in range(1, n + 1):
    eigenvalue = bisection_method(a, b, k)
    eigenvalues_bisection.append(eigenvalue)



eigenvalues, eigenvectors = eigh(T)
print("Собственные значения:", sorted(eigenvalues))
```