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

ОТЧЁТ по лабораторной работе № 6
«Проблема собственных значений матрицы»

Вариант № 8

Выполнила: студент гр. Б9122-02.03.01сцт
Ф.И.О.
Ильяхова Алиса Алексеевна

Проверил: преподаватель
Ф.И.О.
Павленко Елизавета Робертовна

Владивосток

2024

Цели работы:

Найти методом простых итераций одно из собственных значений и все собственные значения симметричной положительно определенной матрицы А методом вращений (Якоби).

Ход работы:

1. Реализовать метод итераций по следующей схеме:
   a. Задаем нормированное начальное приближение x^0 и точность нахождения собственного значения ε.
   b. Находим значение вектора $y^{k+1} = Ax^k$.
   c. Осуществляем нормировку $x^{k+1} = y^{k+1} / ||y^{k+1}||$.
   d. Вычисляем значение $\lambda^{k+1} = \max(x^{k+1}_i / x^k_i)$, где $i = 0, n - 1$, n - размерность матрицы А.
   e. Повторяем шаги (a) - (d) до тех пор, пока не будет выполняться условие $|\lambda^{k+1} - \lambda^k| \leq \varepsilon$.
2. Реализовать метод вращений:
   a. Задаём точность нахождения собственных значений ε.
   b. Поскольку по условию исходная матрица симметричная, мы можем рассматривать либо верхний треугольник матрицы, либо нижний. Рассмотрим верхнюю треугольную наддиагональную часть матрицы А. В ней выделяем максимальный по модулю элемент a_{ij}, $i \neq j$.
   c. Найти угол поворота по формуле $\varphi = ½ * \text{arctg}(2a_{ij} / a_{ii} – a_{jj})$ (если $a_{ii} = a_{jj}$, то угол $\varphi = \pi/4$).
   d. Составить матрицу вращения T_{ij}, где:
   $t_{ii} = t_{jj} = \cos(\varphi)$, $t_{ij} = -t_{ji} = -\sin(\varphi)$, $t_{kk} = 1$, $k \neq ii, ij, ji, jj$.
   Остальные элементы матрицы вращений равны нулю.
   e. Вычислить новое приближение $A^{k+1} = (T^k_{ij})^T * A^k * T^k_{ij}$
   Замечание: в целях оптимизации перемножать полностью на матрицы T не нужно, поскольку по факту у матрицы А поменяются только компоненты ii, ij, ji и jj.
   f. Повторять шаги (a) - (e) до тех пор, пока не будет выполняться условие $\sum_{i \neq j} |a^k_{ij}|^2 \leq \varepsilon$, то есть сумма квадратов внедиагональных элементов матрицы A^{k+1} не должна превышать заданную на первом шаге точность.
3. Необходимо сгенерировать симметричную, положительно определённую матрицу А/размерностей n = 3, n = 5, n = 7. Изучить, сколько требуется шагов обоим методам для каждого случая при $\varepsilon = 10^{-3}, 10^{-7}$.
4. Вывести полученные собственные значения матрицы A^{k+1}.
5. Для каждой из матриц пункта 2 вывести число итераций, потребовавшихся для нахождения значений.
6. Подготовить отчёт о выполненной работе.

Основное:

1) Функции:

Функция **simple_iteration** реализует метод простых итераций для нахождения собственного значения и соответствующего собственного вектора симметричной положительно определенной матрицы.

В бесконечном цикле:

- yk вычисляется как произведение матрицы A на текущий вектор xk.
- Новый вектор xk1 нормализуется так же, как и начальный, чтобы его длина была равна 1.

Затем вычисляется новое собственное значение lambda_k1. Оно определяется как отношение скалярного произведения xk1 и произведения матрицы A на xk1 к скалярному произведению xk1 с самим собой. Это позволяет оценить, насколько хорошо текущий вектор xk1 является собственным вектором.

Здесь проверяется, достигнута ли заданная точность. Если разница между новым и старым значениями собственного значения меньше или равна epsilon, цикл завершает свою работу: if np.abs(lambda_k1 - lambda_k) <= epsilon: break.

Обновляются значения вектора и собственного значения для следующей итерации:

xk = xk1

lambda_k = lambda_k1

Возвращает: найденное собственное значение и количество итераций, которые потребовались для достижения сходимости.

Функция **rotation_method** реализует метод вращений (метод Якоби) для нахождения всех собственных значений симметричной положительно определенной матрицы A .

В бесконечном цикле:

- Ищет максимальный элемент вне диагонали матрицы Ak.
- Если максимальное значение меньше ε - цикл прерывается.
- Вычисляет угол вращения φ на основе найденных индексов максимального элемента.
- Создает матрицу вращения T.
- Обновляет матрицу Ak с помощью преобразования T.

Возвращает: диагональные элементы обновленной матрицы (собственные значения) и количество итераций.

Функция **generate_symmetric_positive_definite_matrix** генерирует случайную симметричную положительно определенную матрицу размером n × n.

1. Задает размеры матриц и значения точности.

2. Генерирует симметричную положительно определенную матрицу для каждого размера.
3. Для каждой матрицы и каждого значения точности:
    - Выполняет метод степенной итерации и метод вращений.
    - Выводит найденные собственные значения и количество итераций для каждого метода.

2) Результаты:

Для матрицы 3x3 с точностями 10^{-3} и 10^{-7}:

```
М-ца A (3x3):
[[3.62209774 0.27449258 0.62781864]
 [0.27449258 3.81627943 0.23664827]
 [0.62781864 0.23664827 3.51347757]]

ε: 0.001
Метод простых итераций:
Собственное значение: 4.415869784957062,
Итерации: 9
Метод вращений:
Собственные значения: [4.4165795  3.59781128 2.93746396],
Итерации: 4

ε: 1e-07
Метод простых итераций:
Собственное значение: 4.416579353342248,
Итерации: 32
Метод вращений:
Собственные значения: [4.41657951 3.59781127 2.93746396],
Итерации: 6
```

Для матрицы 5x5 с точностями 10^{-3} и 10^{-7}:

```
М-ца A (5x5):
[[5.53442258 0.46543906 0.58875339 0.15340793 0.32380074]
 [0.46543906 5.52495203 0.40145092 0.76518068 0.56899883]
 [0.58875339 0.40145092 5.05250911 0.7213502  0.49609191]
 [0.15340793 0.76518068 0.7213502  5.22544634 0.49576015]
 [0.32380074 0.56899883 0.49609191 0.49576015 5.98365435]]

ε: 0.001
Метод простых итераций:
Собственное значение: 7.496837759252444,
Итерации: 11
Метод вращений:
Собственные значения: [7.49763242 5.44893129 5.27146848 4.87949411 4.22345811],
Итерации: 19

ε: 1e-07
Метод простых итераций:
Собственное значение: 7.497632328411537,
Итерации: 24
Метод вращений:
Собственные значения: [7.49763243 5.44893134 5.27146849 4.87949469 4.22345748],
Итерации: 27
```

Для матрицы 7x7 с точностями 10^{-3} и 10^{-7}:

```
М-ца A (7x7):
[[7.02295554 0.53975148 0.45897783 0.1716869  0.82779487 0.46911887
  0.5244291 ]
 [0.53975148 7.11181465 0.27266273 0.51799971 0.37591972 0.2466301
  0.16650111]
 [0.45897783 0.27266273 7.83705706 0.69579387 0.33105035 0.538119
  0.68140746]
 [0.1716869  0.51799971 0.69579387 7.24194889 0.41409598 0.81585997
  0.57760092]
 [0.82779487 0.37591972 0.33105035 0.41409598 7.18125245 0.59276636
  0.19034272]
 [0.46911887 0.2466301  0.538119   0.81585997 0.59276636 7.53486725
  0.31679686]
 [0.5244291  0.16650111 0.68140746 0.57760092 0.19034272 0.31679686
  7.56391196]]
```

```
ε: 0.001
Метод простых итераций:
Собственное значение: 10.211692235183593,
Итерации: 8
Метод вращений:
Собственные значения: [10.21240561  7.59981259  7.25989709  6.98865936  6.95309844  6.44727479
  6.03265991],
Итерации: 38

ε: 1e-07
Метод простых итераций:
Собственное значение: 10.212405749822887,
Итерации: 23
Метод вращений:
Собственные значения: [10.21240585  7.5998132   7.25989763  6.9886593   6.95309731  6.44727479
  6.0326597 ],
Итерации: 58
```

3) Выводы:

1. Методы нахождения собственных значений:

Метод простых итераций:

Этот метод основан на итерационном процессе, который использует начальный вектор и матрицу для нахождения собственного значения.

В процессе работы метода вектор нормализуется, и вычисляется новое собственное значение до тех пор, пока разница между последовательными значениями не станет меньше заданного порога точности (epsilon).

Метод показывает хорошую сходимость для симметричных положительно определённых матриц, однако его эффективность может зависеть от выбора начального вектора.

Метод вращений:

Этот метод использует вращательные преобразования для приведения матрицы к диагональному виду.

Он находит максимальные элементы вне диагонали и применяет соответствующие вращения, чтобы уменьшить их до заданной точности.

Метод вращений также обеспечивает получение собственных значений и является более универсальным для больших матриц.

2. Генерация симметричных положительно определённых матриц:

Для тестирования методов были сгенерированы случайные симметричные положительно определённые матрицы с использованием функции generate_symmetric_positive_definite_matrix. Это обеспечивало корректность тестирования, так как такие матрицы гарантированно имеют положительные собственные значения.

3. Результаты эксперимента:

Для различных размеров матриц (3x3, 5x5, 7x7) и значений точности (1e-3, 1e-7) были получены собственные значения и количество итераций, необходимых для достижения сходимости.

В большинстве случаев оба метода показывали хорошую сходимость, однако метод вращений часто требовал больше итераций для достижения высокой точности по сравнению с методом простых итераций.

4. Сравнение методов:

Сходимость: Метод простых итераций может быть более быстрым для нахождения одного собственного значения, в то время как метод вращений более устойчив и может находить все собственные значения одновременно.

Эффективность: Метод вращений может быть предпочтительнее для больших матриц из-за своей способности обрабатывать все собственные значения за одну итерацию, в то время как метод простых итераций требует повторных запусков для каждого значения.

5. Заключение:

Оба метода продемонстрировали свою работоспособность и эффективность при нахождении собственных значений симметричных положительно определённых матриц.

4) Листинг программы:

```
import numpy as np

def simple_iteration(A, x0, epsilon):
    n = A.shape[0]  #размерность матрицы
    xk = x0 / np.linalg.norm(x0)  #нормализация начального вектора
    lambda_k = 0  #начальное значение собственного значения
    iterations = 0  #счётчик итераций

    while True:
        yk = A @ xk  #умножение матрицы на вектор
        xk1 = yk / np.linalg.norm(yk)  #нормализация нового вектора

        #вычисление нового собственного значения
        lambda_k1 = np.dot(xk1, A @ xk1) / np.dot(xk1, xk1)

        iterations += 1

        #проверка на сходимость
        if np.abs(lambda_k1 - lambda_k) <= epsilon:
            break

        xk = xk1  #обновление вектора
        lambda_k = lambda_k1  #обновление собственного значения

    return lambda_k1, iterations  #возврат найденного собственного значения и
количества итераций

def rotation_method(A, epsilon):
    n = A.shape[0]
    Ak = A.copy()  #копия матрицы для модификации
    iterations = 0

    while True:
        max_val = 0  #максимальное значение вне диагонали
        i, j = -1, -1  #индексы максимального элемента

        #поиск максимального вне диагонального элемента
        for k in range(n):
            for l in range(k + 1, n):
                if abs(Ak[k, l]) > max_val:
                    max_val = abs(Ak[k, l])
```

```
                    i, j = k, l

        #если максимальное значение меньше заданной точности, выходим из цикла
        if max_val < epsilon:
            break

        aii = Ak[i, i]  #элемент на главной диагонали
        ajj = Ak[j, j]  #элемент на главной диагонали
        aij = Ak[i, j]  #элемент вне диагонали

        if aii == ajj:
            phi = np.pi / 4  #угол для вращения
        else:
            phi = 0.5 * np.arctan2(2 * aij, aii - ajj)  #вычисление угла
вращения

        #создание матрицы вращения
        T = np.eye(n)
        T[i, i] = T[j, j] = np.cos(phi)
        T[i, j] = -np.sin(phi)
        T[j, i] = np.sin(phi)

        #обновление матрицы Ak
        Ak = T.T @ Ak @ T

        iterations += 1

    return np.diag(Ak), iterations  #возврат собственных значений и количества
итераций

def generate_symmetric_positive_definite_matrix(n):
    A = np.random.rand(n, n)  #генерация случайной матрицы
    return (A + A.T) / 2 + n * np.eye(n)  #обеспечение симметричности и
положительной определенности

sizes = [3, 5, 7]  #размеры матриц
epsilons = [1e-3, 1e-7]  #значения точности

for n in sizes:
    A = generate_symmetric_positive_definite_matrix(n)  #генерация матрицы
    print(f"М-ца A ({n}x{n}):\n{A}\n")  #вывод матрицы

    for epsilon in epsilons:
        #метод степенной итерации
        x0 = np.random.rand(n)  #начальный вектор
        eigenvalue_power, iterations_power = simple_iteration(A, x0, epsilon)
```

```
        #метод вращений
        eigenvalues_rotation, iterations_rotation = rotation_method(A.copy(), 
epsilon)

        print(f"ε: {epsilon}")
        print(f"Метод простых итераций:\nСобственное значение: 
{eigenvalue_power},\nИтерации: {iterations_power}")
        print(f"Метод вращений:\nСобственные значения: 
{eigenvalues_rotation},\nИтерации: {iterations_rotation}\n")
```