Дисциплина: Численные методы
Лабораторное задание **№3**, Вариант **№3**

Отчёт
Тема: «Применение точных методов решения систем линейных алгебраических уравнений»

Выполнил:
студент 3 курса 61 группы
Вафин А.Р.

Проверила:
старший преподаватель
Фролова О.А.

1. Постановка задачи

Применение метода прямых итераций с исчерпыванием для определения пары со третьим максимальным по модулю собственным значением симметричной матрицы простой структуры.

Входные параметры основной процедуры:
N – размерность матрицы;

A – двумерный массив размерности $N \times N$;

ε (лямбда не рисуется почему-то) – точность определения второго минимального по модулю собственного значения;

ε_g – точность определения собственного вектора, соответствующего второму минимальному по модулю собственному значению;

n – минимальное по модулю собственное значение;

x_n – собственный вектор, соответствующий минимальному по модулю собственному значению;

M – максимально допустимое число итераций.

Выходные параметры основной процедуры:

λ – второе минимальное по модулю собственное значение;

x – собственный вектор, соответствующий второму минимальному по модулю собственному значению;

K – число выполненных итераций;

r – мера точности полученной пары $(лямбда, x)$.

2. Метод решения

Для того чтобы составить симметричную матрицу размерности N, имеющую заранее известные собственные значения, можно поступить следующим образом. Пусть Λ=diag(λ_i) – диагональная матрица размерности $N \times N$, λ_i– собственные значения конструируемой матрицы A, ω – случайным образом сгенерированный и пронормированный вектор ($|\omega|$=1) размерности N. Образуем с помощью вектора (столбца) ω матрицу Хаусхолдера:

$$H = E - 2\omega\omega^T,$$

являющуюся симметричной и ортогональной. Тогда в качестве тестируемой матрицы можно взять матрицу

$$A = H\Lambda H^T,$$

у которой все собственные значения (элементы диагонали матрицы Λ) и все соответствующие им собственные векторы (столбцы матрицы H) известны.

Если пара (λ_n, x_n) найдена, то степенной метод можно применить для вычисления пары (λ_{n-1}, x_{n-1}). Введем в рассмотрение матрицу

$$A^{(1)} = A - \lambda_n x_n x_n^T. \qquad (2.2.1)$$

После выбора начального приближения $x^{(0)}$ итерационный процесс организуется по схеме

$$\begin{cases} v^{(k)} = x^{(k)} \Big/ \left\| x^{(k)} \right\| \\ x^{(k+1)} = A^{(1)} v^{(k)}, \end{cases} \quad \mathrm{k} = 0, 1, 2, \ldots \qquad (2.2.2)$$

При этом $v^{(k)} \to \pm x_{n-1}$, $\sigma^{(\mathrm{k})} = v^{(k)^T} x^{(k+1)} \to \lambda_{n-1}$ при $\mathrm{k} \to \infty$.

В формулу вычисления матрицы для текущего шага подставляется матрица, вычисленная на предыдущем шаге, и тогда последним шагом будет вычисление нужной матрицы.

Для вычисления абсолютного значения угла между двумя векторами можно использовать формулу через скалярное произведение: $cos\ \theta = \frac{\vec{a} * \vec{b}}{\|a\| \|b\|}$, где $\vec{a} * \vec{b}$ – скалярное произведение векторов, $\|a\|$, $\|b\|$ – длины (нормы) векторов, θ – угол между векторами, который найдём по формуле $\arccos(\cos\theta)$.

Для вычисления меры точности решения вычислим вектор отклонения $r = A *$

$x -$ $* x$ и найдём первую норму вектора: $\|r\| = \max_i |r_i|$

3. Описание основных процедур

В программе реализованы следующие процедуры и функции:

1) generate_random_vector(long double* omega, int N)

, где omega ω – вектор для построения матрицы Хаусхолдера, N – размерность вектора.

Функция создаёт вектор ω для построения матрицы Хаусхолдера H.

2) generate_symmetric_matrix(long double** A, long double** H, long double* eigenvalues, int N)

, где A – будущая матрица. H – генерируемая в процессе матрица Хаусхолдера, N - размерность будущей матрицы A, eigenvalues – вектор заранее известных собственных значений.

Функция строит симметричную матрицу размерности N, имеющую заранее известные собственные значения описанным выше способом.

4) straight_iteration_exhaust(long double** A, int N, long double lambda_1, long double* x_1, long double lambda_2, long double* x_2, long double& lambda_3, long double* x_3, long double epsilon, int M, int& K, long double& r, long double& avg_vec, long double lambda_true, long double* x_true, long double& avg_lambda)
где A – матрица A из алгоритма, lambda_1 – первое собственное значение, x_1 – первый собственный вектор, epsilon – требуемая точность, lambda_2 и x_2 – вторые собственное значение и вектор, N – размерность матрицы и векторов, M – максимальное количество итераций алгоритма, K – количество текущих итераций алгоритма, r – мера точности, lambda_3 и x_3 – третьи собств. значение и вектор, которые ищем; lambda_true и x_true - действительные третьи собств. значение и собств. вектор, avg_vec и avg_lambda – высчитываемые оценки точности собств. вектора и собств. значения.

Функция представляет алгоритм прямых итераций с исчерпыванием, описанный выше.

5) mat_vec_mult(long double** A, long double* x, long double* result, int N)

, где A – умножаемая матрица, x – вектор, на который умножают матрицу A, result – вектор-результат, N – размерность матрицы и векторов.

Функция умножает матрицу и вектор заданной размерности, результат складывает в вектор-результат.

6) generate_ort_vector(long double* input_vec, int N, long double* out_vec)

, где input_vec – исходный вектор, к которому строим ортогональный вектор, N – размерность векторов, out_vec – построенный ортогональный к первому вектор.

Функция строит ортогональный к первому второй вектор.

7) dot_product(long double* a, long double* b, int N)

, где a и b – вектора, N – их размерность.

Функция ищет скалярное произведение векторов.

8) vec_length(long double* vec, int N)

, где vec – вектор, N – его размерность.

Функция ищет длину вектора.

9) angle_vectors(long double* a, long double* b, int N)

, где a и b – вектора, N – их размерность.

Функция ищет абсолютное значение угла между двумя векторами по формулам выше.

10) sort_array_abs(long double* arr, int size)

, где arr – массив чисел, size – его размер.

Функция сортирует массив по абсолютным значениям его элементов.

11) generate_random_eigenvalues(long double* eigenvalues, long double lower_bound, long double upper_bound)

, где eigenvalues – массив, lower_bound и upper_bound – границы создаваемых элементов.

Функция заполняет массив собственных чисел матрицы рандомно-сгенерированными значениями в заданных границах.

№ теста	Размерность системы N	Диапазон значений λ	Точность (ε (лямбда) ε_g)	Сред. оценка точности собств. значений	Сред. оценка точности собств. векторов	Сред. мера точности r	Среднее число операций
1	10	$-2 \div 2$	10^{-5}	9.2705e-11	9.10215e-06	4.74451e-06	24.4
2	10	$-2 \div 2$	10^{-8}	7.66609e-14	0	1.01465e-07	146
3	10	$-50 \div 50$	10^{-5}	3.91057e-07	7.74732e-06	0.000705944	87.5
4	10	$-50 \div 50$	10^{-8}	5.53779e-14	0	5.04961e-07	117.4
5	30	$-2 \div 2$	10^{-5}	6.95984e-11	7.85854e-06	3.06147e-06	37.1
6	30	$-2 \div 2$	10^{-8}	6.55725e-16	0	4.21643e-09	48.2
7	30	$-50 \div 50$	10^{-5}	1.73947e-08	8.75143e-06	0.000207941	48.9
8	30	$-50 \div 50$	10^{-8}	2.05036e-13	0	4.54156e-07	116.5
9	50	$-2 \div 2$	10^{-5}	3.10098e-10	7.78397e-06	5.23488e-06	31.3
10	50	$-2 \div 2$	10^{-8}	4.19456e-16	0	4.86583e-09	59
11	50	$-50 \div 50$	10^{-5}	1.3657e-07	8.17931e-06	0.000364423	63.7
12	50	$-50 \div 50$	10^{-8}	7.61613e-14	0	2.42428e-07	198.4