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:  
 – размерность матрицы;

– двумерный массив размерности ;

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

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

– минимальное по модулю собственное значение;

– собственный вектор, соответствующий минимальному по модулю собственному значению;

*M* – максимально допустимое число итераций.

Выходные параметры основной процедуры:

*l* – 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

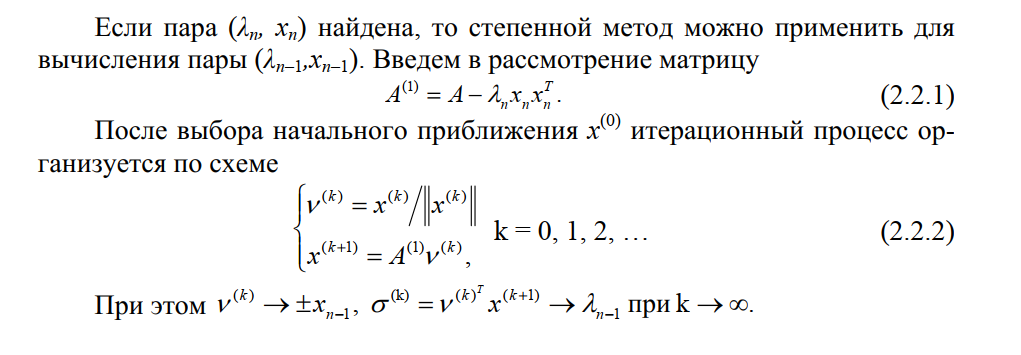
**2. Метод решения**

Для того чтобы составить симметричную матрицу размерности N, имеющую заранее известные собственные значения, можно поступить следующим образом. Пусть =diag() – диагональная матрица размерности *N* × *N*, – собственные значения конструируемой матрицы А, ω – случайным образом сгенерированный и пронормированный вектор (=1) размерности N. Образуем с помощью вектора (столбца) ω матрицу Хаусхолдера:

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

,

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



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: , где – скалярное произведение векторов, – длины (нормы) векторов, – угол между векторами, который найдём по формуле

Для вычисления меры точности решения вычислим вектор отклонения и найдём первую норму вектора:

**3. Описание основных процедур**

В программе реализованы следующие процедуры и функции:

1) generate\_random\_vector(long double\* omega, int N)

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

Функция создаёт вектор 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  | Точность () | Сред. оценка точности собств. значений | Сред. оценка точности собств. векторов | Сред. мера точности *r* | Среднее число операций |
| 1 | 10 |  |  | 9.2705e-11 | 9.10215e-06 | 4.74451e-06 | 24.4 |
| 2 | 10 |  |  | 7.66609e-14 | 0 | 1.01465e-07 | 146 |
| 3 | 10 |  |  | 3.91057e-07 | 7.74732e-06 | 0.000705944 | 87.5 |
| 4 | 10 |  |  | 5.53779e-14 | 0 | 5.04961e-07 | 117.4 |
| 5 | 30 |  |  | 6.95984e-11 | 7.85854e-06 | 3.06147e-06 | 37.1 |
| 6 | 30 |  |  | 6.55725e-16 | 0 | 4.21643e-09 | 48.2 |
| 7 | 30 |  |  | 1.73947e-08 | 8.75143e-06 | 0.000207941 | 48.9 |
| 8 | 30 |  |  | 2.05036e-13 | 0 | 4.54156e-07 | 116.5 |
| 9 | 50 |  |  | 3.10098e-10 | 7.78397e-06 | 5.23488e-06 | 31.3 |
| 10 | 50 |  |  | 4.19456e-16 | 0 | 4.86583e-09 | 59 |
| 11 | 50 |  |  | 1.3657e-07 | 8.17931e-06 | 0.000364423 | 63.7 |
| 12 | 50 |  |  | 7.61613e-14 | 0 | 2.42428e-07 | 198.4 |