**Реферат**

Целью данной работы является изучение метода вращений, применение данного метода для вычисления собственных значений и собственных векторов симметричной матрицы и реализация этого метода на ЭВМ.

Пояснительная записка к курсовому проекту состоит из двух основных частей: теоретической и практической. В первой части рассматривается сущность метода Данилевского. Практическая часть содержит разработку программного кода для вычисления собственных значений и собственных векторов матрицы на языке программирования С Sharp.

При написании пояснительной записки было использовано 7 литературных источников.

Перечень ключевых слов: собственные значения, собственные вектора, вращение Якоби, симметричная матрица.

**Содержание**

Введение

. Теоретическая часть

.1 Собственные значения и собственные вектора матрицы

.2 Итерационный метод вращений Якоби решения симметричной полной проблемы собственных значений

.3 Метод вращений Якоби

.4 Решение обобщенной проблемы собственных значений

. Практическая часть

.1 Пример решения

.2 Реализация на ЭВМ

Заключение

Список литературы

Приложение 1. Листинг программы

Приложение 2. Пример результата работы программы

#### **Введение**

Целый ряд инженерных задач сводится к рассмотрению систем уравнений, имеющих единственное решение лишь в том случае, если известно значение некоторого входящего в них параметра. Этот особый параметр называется характеристическим, или собственным значением системы. С задачами на собственные значения инженер сталкивается в различных ситуациях. Так, для тензоров напряжений собственные значения определяют главные нормальные напряжения, а собственными векторами задаются направления, связанные с этими значениями. При динамическом анализе механических систем собственные значения соответствуют собственным частотам колебаний, а собственные вектора характеризуют моды этих колебаний. При расчете конструкций собственные значения позволяют определять критические нагрузки, превышение которых приводит к потере устойчивости. Выбор наиболее эффективного метода определения собственных значений или собственных векторов для данной инженерной задачи зависит от ряда факторов, таких, как тип уравнений, число искомых собственных значений и их характер. Алгоритмы решения задач на собственные значения делятся на две группы. Итерационные методы очень удобны и хорошо приспособлены для определения наименьшего и наибольшего собственных значений. Методы преобразований подобия несколько сложней, зато позволяют определить все собственные значения и собственные векторы. В данной работе будет рассмотрен метод вращений Якоби. Однако сначала приведем некоторые основные сведения из теории матричного и векторного исчислений, на которых базируются методы определения собственных значений.

#### **1. Теоретическая часть**

#### **.1 Собственные значения и собственные вектора матрицы**

Рассмотрим квадратную матрицу n-ого порядка:



Собственные значения l*i*  квадратной матрицы A есть действительные или комплексные числа, удовлетворяющие условию:

,

E - единичная матрица,  - собственный вектор матрицы A, соответствующий некоторому собственному значению l.

Матрица  называется характеристической матрицей матрицы A. Т.к. в матрице  по главной диагонали стоят l, а все остальные элементы равны нулю, то характеристическая матрица имеет вид:



Определитель этой матрицы называется характеристическим или вековым определителем и равен:



В развернутом виде он является многочленом n-ой степени относительно l, т.к. при вычислении этого определителя произведение элементов главной диагонали дает многочлен со старшим членом , т.е.



и называется характеристическим многочленом. Корни  этого многочлена - собственные значения или характеристические числа матрицы A. Числа  называются коэффициентами характеристического многочлена.

Ненулевой вектор  называется собственным вектором матрицы A, если эта матрица переводит вектор X в вектор

,

т.е. произведение матрицы A на вектор X и произведение характеристического числа l на вектор X есть один и тот же вектор. Каждому собственному значению  матрицы соответствует свой собственный вектор .

Для определения координат собственного вектора составляется характеристическое уравнение: . Переписав его в векторном виде и выполнив умножение, получим систему линейных однородных уравнений:



Определитель этой системы равен нулю, т.к. из этого условия были определены собственные значения матрицы A. Следовательно, система имеет бесконечное множество решений. Ее можно решить с точностью до постоянного множителя (как систему однородных уравнений). Решив эту систему, мы найдем все координаты собственного вектора X. Подставляя в систему однородных уравнений поочередно , получаем n собственных векторов.

#### **.2 Итерационный метод вращений Якоби решения симметричной полной проблемы собственных значений**

матрица вектор эрмитовый итерационный

Метод вращений применим к эрмитовым матрицам с комплексными элементами. Для простоты рассмотрим частный случай эрмитовых матриц - действительных симметричных.

Известно, что всякая симметричная действительная матрица  может быть приведена к диагональному виду преобразованием

, (1)

где  - некоторая ортогональная матрица; - диагональная матрица, элементами которой являются собственные числа  матрицы .

Так как для ортогональной матрицы обратная матрица совпадает с транспонированной матрицей: , то равенство (1) равносильно следующему

. (2)

На основе этого равенства можно построить ряд методов, отличающихся способом вычисления матрицы . Все методы базируются на следующем равенстве:

, (3)

где  - некоторая ортогональная матрица, которая несложно вычисляется;  - квадратная матрица с близкими к нулю недиагональными элементами, так как найти  в общем случае трудоемко:

. (4)

Естественно предполагать, что коэффициенты , , будут близки к собственным числам матрицы .

Введем обозначение для суммы квадратов модулей недиагональных элементов по строкам:

. (5)

За меру близости матрицы  к диагональному виду примем число:

. (6)

Пусть с помощью преобразования подобия с ортогональными матрицами построена последовательность матриц: . Процесс построения этих матриц называется монотонным, если .

Таких процессов может быть построено большое число. Рассмотрим метод вращений. Он достаточно прост по вычислительной схеме и обладает быстрой сходимостью.

#### **.3 Метод вращений Якоби**

В методе вращений последовательность матриц строится с помощью ортогональной матрицы вращения:

 (7)

Матрица вращений  получается из единичной матрицы заменой ее элементов на пересечениях - х и - х строк и столбцов записанными элементами.

Предположим, что преобразования доведены до шага номера  и построена матрица . Найдем в ней наибольший по модулю недиагональный элемент . Если таких элементов несколько, то рассматриваем любой из них. По индексам ,  найденного наибольшего по модулю недиагонального элемента строим матрицу вращения , в которой угол  пока не определен. Далее образуем матрицу

. (8)

Введем обозначения:

.

Ввиду особой структуры матрицы  (7) все столбцы матрицы , кроме - го и - го, будут такими же, как и в матрице . Элементы столбцов номеров  и  будут вычисляться по формулам:

;

. (9)

Аналогично строки матрицы , кроме - й и - й, будут такими же, как в матрице , а элементы ее строк - й и - й вычисляется по формулам:

;

. (10)

Равенства (9) и (10) позволяют несложно вычислить коэффициенты



,

которые в силу равенства  преобразуются к следующему виду:

. (11)

Сейчас назначим в выражении (11) такое значение угла , чтобы рассматриваемый элемент  обратился в нуль. Из этого условия получаем выражение:

,

из которого находим значение угла

 (12)

Убедимся, что полученная таким образом матрица  ближе к диагональному виду, чем предшествующая . Для этого вычислим - меру близости матрицы  к диагональному виду. Пусть матрица  имеет меру ее близости к диагональному виду . Матрица  симметрична и на основании выражений (9) - (11) можно показать, что

, (13)

так как  есть наибольший по модулю недиагональный элемент, отличный от нуля, который исключен при получении матрицы .

Тогда на основании выражения (13) верно неравенство:

.

Следовательно, мера  уменьшиться при преобразовании (8).

Оценим скорость стремления к нулю величины  при преобразованиях исходной матрицы согласно выражению (8).

По выбору элемента  как наибольшего по модулю справедливо неравенство



и, следовательно,

. (14)

С помощью этого неравенства из выражения (13) получается

, (15)

где

.

Пусть значение . Тогда величина  и из неравенства (15) вытекает цепочка неравенств

.

Так как , то для меры близости матрицы  к диагональному виду  будем иметь оценку



Отсюда следует, что  со скоростью не меньшей, чем скорость сходимости геометрической прогрессии со знаменателем .

Рассмотренный метод преобразования подобия с ортогональной матрицей требует выбора среди недиагональных элементов преобразуемой матрицы наибольшего по модулю. Для этого необходимо выполнить приблизительно  операций.

Существуют и другие методы выбора наибольшего по модулю элемента, предназначенного для исключения. Например, можно выбрать наибольшую из сумм (5). Если это есть , то в качестве элемента  можно выбрать наибольший по модулю элемент из найденной строки . При таком правиле выбора элемента  необходимо выполнить примерно  операций. Отметим, что оценка (16) для такого метода выбора наибольшего по модулю элемента, предназначенного для исключения, сохраняется.

#### **.4 Решение обобщенной проблемы собственных значений**

Определение 1. Пусть  и  некоторые квадратные матрицы. Тогда задача нахождения чисел  и ненулевых векторов , в общем случае комплексных, являющихся решениями уравнения , называется обобщенной проблемой собственных значений. Искомые числа  называются собственными числами проблемы, а ненулевые векторы  - собственными векторами, соответствующими этим собственными числам .

Определение 2. Матрица  называется матричным пучком и обозначается .

Если матрица  невырожденная, то обобщенная проблема собственных значений эквивалентна стандартной проблеме собственных значений  с матрицей .

Решение ряда задач сводится к обобщенной проблеме собственных значений с симметричными действительными положительно определенными матрицами , :

. (1)

Тогда все собственные числа обобщенной проблемы положительны (неотрицательны).

Сведение обобщенной проблемы собственных значений с симметричными матрицами к стандартной проблеме собственных значений  невыгодно, так как в ней матрица  несимметричная. Как известно, симметричные матрицы имеют ряд полезных свойств, которые позволяют построить эффективные методы различных вычислений. Поэтому сознательно терять полезные свойства исходной проблемы невыгодно. В связи с этим для решения обобщенной проблемы собственных значений с симметричными матрицами разработаны специальные методы, которые, по сравнению с общими методами, позволяют решать проблему более эффективно.

Отметим, что обобщенная проблема собственных значений с симметричными действительными положительно определенными матрицами возникает, например, при определении собственных частот линейных колебательных систем с распределенными параметрами вариационными методами математической физики, в том числе методом конечных элементов. При анализе частот собственных колебаний конструкции необходимо решать уравнение (1), в котором искомая величина  явно выражена через частоту собственных колебаний конструкции;  - общая матрица жесткости конструкции;  - матрица масс;  - форма собственных колебаний.

Среди методов решения обобщенной проблемы собственных значений с симметричными матрицами одно семейство методов состоит в сведении проблемы к стандартной проблеме собственных значений с симметричной матрицей. Рассмотрим два метода этого семейства, в которых применяются теоретические результаты вычислительной математики, изложенные ранее.

Рассмотрим первый метод. С помощью преобразования подобия представим в уравнении (1) матрицу  в виде разложения на множители:

, (3)

где  - ортогональная матрица;  - диагональная матрица, ненулевые элементы которой равны: , ; ,  - собственные числа матрицы , при этом , ; ;  - диагональная матрица, ненулевые элементы которой равны: , .

Единственное и решающее использование этого свойства в рассматриваемом методе - свойство положительной определенности матрицы , состоящее в том, что все ее собственные числа положительные. Следовательно, можно получить вещественную матрицу .

Матрицы  и  могут быть найдены, например, с помощью процедуры метода вращений, изложенной ранее.

Согласно формуле (3) подставим матрицу  в уравнение (1), получим

.

Умножая полученное равенство слева на матрицу , получим:

. (4)

Представим

 (5)

и, подставляя это выражение в уравнение (4), приведем его сначала к виду

,

а затем запишем так

. (6)

Уравнение (6) можно представить в стандартном виде

, (7)

где  - симметричная матрица.



Так как в уравнении (7) матрица  симметричная, то для нахождения ее собственных чисел и собственных векторов можно второй раз применить метод вращений. В результате применения метода вращений получаются множители разложения для матрицы  в виде:

 (8)

где  - ортогональная матрица;  - диагональная матрица.

На основании выражений (7), (8) запишем равенство

,

из которого следует

, (9)

где  - ортогональная матрица, равная произведению обратимых матриц.



Результат двойного применения метода вращений можно представить в виде преобразования подобия пары исходных матриц ,  (пучка матриц ) с помощью ортогональной матрицы , которое дает  - матрицу собственных чисел исходной проблемы (1).

Собственные векторы проблемы (7) связаны с векторами исходной проблемы (1) соотношением (5).

Рассмотрим второй метод. Представим матрицу  в виде произведения двух треугольных матриц, полученных по схеме метода квадратных корней

, (10)

где  - правая треугольная матрица разложения по схеме метода квадратных корней (разложение Холецкого, см. формулы (11) подраздела).

Подставляя выражение (10) для матрицы  в уравнение (1) и умножая его слева на матрицу , получим

.

Затем, подставляя в это уравнение , получим

, (11)

где  - симметричная матрица.

Так как в уравнении (11) матрица  симметричная, то для решения получившейся стандартной проблемы собственных значений  можно применить метод вращений.

Замечания. 1. Если собственные числа или собственные векторы матрицы найдены недостаточно точно, то их можно уточнить существующими методиками уточнения отдельного собственного значения и соответствующего собственного вектора.

. Разработаны методы уточнения приближенного решения полной проблемы собственных значений.

. Разработаны методики ускорения сходимости при решении проблем собственных значений как частичных, так и полной.

#### **2. Практическая часть**

#### **.1 Пример решения**

Найти собственные значения и собственные вектора для матрицы

. Положим .

. Выделим максимальный по модулю элемент в над диагональной части: . Так как , то процесс продолжается.

3. Находим угол поворота:



. Сформируем матрицу вращения:



. Выполним первую итерацию:



Положим и перейдем к пункту 2.

(1). Максимальный по модулю над диагональный элемент

Так как , процесс продолжается.

(1). Найдем угол поворота:



(1). Сформируем матрицу вращения:



(1). Выполним вторую итерацию:



Положим и перейдем к пункту 2.

(2) Максимальный по модулю над диагональный элемент

.

(2). Найдем угол поворота:



(2). Сформируем матрицу вращения



(2). Выполним третью итерацию и положим и перейдем к пункту 2:



(3). Максимальный по модулю над диагональный элемент

(3). Найдем угол поворота:



(3). Сформируем матрицу вращения:



(3). Выполним четвертую итерацию и положим и перейдем к пункту 2:



(4). Так как , процесс повторяется

(4). Найдем угол поворота



(4). Сформируем матрицу вращения:



(4). Выполним пятую итерацию и положим и перейдем к пункту 2:



(5). Так как наибольший по модулю над диагональный элемент удовлетворяет условию , процесс завершается.

Собственные значения: 

Для нахождения собственных векторов вычислим



отсюда



или после нормировки



#### **.2 Реализация на ЭВМ**

Алгоритм метода вращений

1) Положить и задать .

) Выделить в верхней треугольной над диагональной части матрицы максимальный по модулю элемент .

Если для всех , процесс завершить. Собственные значения определяются по формуле

.

Собственные векторы находятся как i-e столбцы матрицы, получающейся в результате перемножения:



Если , процесс продолжается.

) Найти угол поворота по формуле.

) Составить матрицу вращения .

) Вычислить очередное приближение

Положить и перейти к пункту 2.

Замечания:

Используя обозначение

,

можно в пункте 3 алгоритма вычислять элементы матрицы вращения по формулам

,



Контроль правильности выполнения действий по каждому повороту осуществляется путем проверки сохранения следа преобразуемой матрицы.

#### **Заключение**

При выполнении данной курсовой работы был изучен метод вращения Якоби решения симметрически полной проблемы собственных значений.

Этот метод при небольших размерах матриц дает неплохие результаты, и позволяет с большой точностью вычислять собственные пары. При больших размерах матриц реализация метода наталкивается на существенные потери машинных ресурсов (из-за необходимости поиска максимального элемента матрицы, и перемножения матриц большой размерности). Так же резко сужает возможность применить данный метод на практике то, что он может быть применен только к симметрическим, вещественным матрицам. Ввиду вышесказанного, этот метод на практике применяется редко, чаще применяется циклический метод Якоби с барьерами. Скорость сходимости которого так же асимптотически квадратичная.

#### **Список литературы**

Бахвалов Н.С. Численные методы. М.: Наука, 1987. 600 с.

Бусленко H. П., Голенко Д.И., Соболь И.М., Срагович В.Г., Шрейдер Ю.А.. Метод статистических испытаний (метод Монте-Карло). М.: Физматгиз, 1962. 332 с.

Бусленко Н.П., Шрейдер Ю.А. Метод статистических испытаний (Монте-Карло) и его реализация на цифровых вычислительных машинах. М.: Государственное издательство физико-математической литературы, 1961. 226 с.

Демидович Б.П., Марон И.А. Основы вычислительной математики. М.: Наука, 1966. 664 с.

Ермаков С.М. Метод Монте-Карло и смежные вопросы. М.: Наука, 1975. 294 с.

Калиткин Н.Н. Численные методы. М.: Наука, 1978. 512 с.

Копчёнова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. М.: Наука, 1972. 368 с.

#### **Приложение 1. Листинг программы**

using System;System.Collections.Generic;System.Linq;System.Text;Числаки

{RotationMethod

{static double Max(double[,] matrix, out int index\_i, out int index\_j)

{\_i = 0;\_j = 1;max = Math.Abs(matrix[1,0]);(int i = 0; i < matrix.GetLength(0); i++)(int j = i + 1; j < matrix.GetLength(1); j++)(Math.Abs(matrix[i, j]) > max)

{= Math.Abs(matrix[i, j]);\_i = i;\_j = j;

}max;

}static double[,] CreateRotationMatrix(int rows, int columns, int i, int j, double cos\_phi, double sin\_phi)

{[,] matrix\_h = new double[rows, columns];\_h[i,i] = cos\_phi;\_h[j,j] = cos\_phi;\_h[i,j] = - sin\_phi;\_h[j,i] = sin\_phi;(int k = 0; k < rows; k++)(int l = 0; l < columns; l++)

{(k == l && k != i && k != j)

{\_h[k, l] = 1;

}

}matrix\_h;

}static double[,] Transpose(double[,] matrix)

{[,] matrix\_transpose = new double[matrix.GetLength(1), matrix.GetLength(0)];(int i = 0; i < matrix.GetLength(0); i++)(int j = 0; j < matrix.GetLength(1); j++)\_transpose[j, i] = matrix[i, j];matrix\_transpose;

}static double[,] Multiply(double[,] matrix\_1, double[,] matrix\_2)

{(matrix\_1.GetLength(0) != matrix\_2.GetLength(1)) throw new Exception("Некорректная размерность матриц");[,] matrix\_multiply = new double[matrix\_1.GetLength(0), matrix\_2.GetLength(1)];(int i = 0; i < matrix\_multiply.GetLength(0); i++)(int j = 0; j < matrix\_multiply.GetLength(1); j++)(int k = 0; k < matrix\_1.GetLength(0); k++)

{\_multiply[i, j] += matrix\_1[i, k] \* matrix\_2[k, j];

}matrix\_multiply;

}static double[] Rotation(double[,] matrix, double eps, out double[,] matrix\_vector)

{[] vector\_eigenvals = new double[matrix.GetLength(0)];[,] matrix\_a = matrix;i, j;count = 0;max = Max(matrix\_a, out i, out j);.WriteLine("Max={0:0.###},\ti={1:0.###},\tj={2:0.###}", max, i, j);cos\_phi = Math.Sqrt(0.5 \* (1 + 1 / Math.Sqrt(1 + (2 \* matrix\_a[i, j] / (matrix[i, i] - matrix[j, j])) \* (2 \* matrix\_a[i, j] / (matrix[i, i] - matrix[j, j])))));sin\_phi = Math.Sqrt(0.5 \* (1 - 1 / Math.Sqrt(1 + (2 \* matrix\_a[i, j] / (matrix[i, i] - matrix[j, j])) \* (2 \* matrix\_a[i, j] / (matrix[i, i] - matrix[j, j])))));.WriteLine("Cos\_phi={0}\tSin\_phi={1}", cos\_phi, sin\_phi);title\_h = "Матрица H\_" + count + ":";[,] matrix\_h = CreateRotationMatrix(matrix\_a.GetLength(0), matrix\_a.GetLength(1), i, j, cos\_phi, sin\_phi);(title\_h, matrix\_h);title\_a = "Матрица A\_" + count + ":";\_a = Multiply(Multiply(Transpose(matrix\_h), matrix\_a), matrix\_h);(title\_a, matrix\_a);\_vector = matrix\_h;(max > eps)

{++;= Max(matrix\_a, out i, out j);.WriteLine("Max={0:0.###},\ti={1:0.###},\tj={2:0.###}", max, i, j);\_phi = Math.Sqrt(0.5 \* (1 + 1 / Math.Sqrt(1 + (2 \* matrix\_a[i, j] / (matrix\_a[i, i] - matrix\_a[j, j])) \* (2 \* matrix\_a[i, j] / (matrix\_a[i, i] - matrix\_a[j, j])))));\_phi = Math.Sqrt(0.5 \* (1 - 1 / Math.Sqrt(1 + (2 \* matrix\_a[i, j] / (matrix\_a[i, i] - matrix\_a[j, j])) \* (2 \* matrix\_a[i, j] / (matrix\_a[i, i] - matrix\_a[j, j])))));.WriteLine("Cos\_phi={0}\tSin\_phi={1}", cos\_phi, sin\_phi);\_h = "Матрица H\_" + count + ":";\_h = CreateRotationMatrix(matrix\_a.GetLength(0), matrix\_a.GetLength(1), i, j, cos\_phi, sin\_phi);(title\_h, matrix\_h);\_a = "Матрица A\_" + count + ":";\_a = Multiply(Multiply(Transpose(matrix\_h), matrix\_a), matrix\_h);(title\_a, matrix\_a);\_vector = Multiply(matrix\_vector, matrix\_h);

}(int l = 0; l < vector\_eigenvals.GetLength(0); l++)\_eigenvals[l] = matrix\_a[l, l];vector\_eigenvals;

}void PrintMatrix(string header, Array matrix)

{i = 0;.WriteLine();.WriteLine(header);(object x in matrix)

{(i == matrix.GetLength(1))

{.WriteLine();= 0;

}++;.Write("{0,11:0.#######}", x);

}.WriteLine();.WriteLine();

}void PrintVector(string header, string index, Array vector)

{i = 1;.WriteLine();.WriteLine(header);(object x in vector)

{.WriteLine("{0,6}[{1}]={2}", index, i, x);++;

}

}void Main()

{

{[,] matrix = { { 5, 1, 2 }, { 1, 4, 1 }, { 2, 1, 3 } };eps = 1e-3;[,] matrix\_vector = new double[matrix.GetLength(0), matrix.GetLength(1)];[] vector\_eigenvals = new double[matrix.GetLength(0)];("Исходная матрица:", matrix);\_eigenvals = Rotation(matrix, eps, out matrix\_vector);("Собственные значения", "lambda", vector\_eigenvals);("Собственные вектора:", matrix\_vector);

Console.ReadLine();

}(Exception e)

{.WriteLine(e.Message);.ReadLine();

}

}

}

}

#### **Приложение 2. Пример результата работы программы**