6 –TOM 2 – SON / 2023 - YIL / 15 - FEVRAL

УДК 518.0

# АЛГОРИТМЫ МЕТОДА ЯКОБИ ДЛЯ СОБСТВЕННЫХ ЗНАЧЕНИЙ И ВЕКТОРОВ СИММЕТРИЧЕСКИХ МАТРИЦ

**Имомов Адаш**

*adashimomov50@gmail.com*

*к.ф.-м.н., Наманганский государственный университет, Узбекистан*

**Аннотация**: *В статье рассматриваются классический метод Якоби для симметрических матриц, а также методы непосредственного определения с помощью команд математической системы **Mathcad**, двухступенчатого классического метода по определению. Для всех методов построены укрупнёные алгоритмы, состоящие из последовательности математических формул и команд математической системы **Mathcad**. Здесь **Mathcad** выступает как исполнитель укрупнённого математического алгоритма решения задачи. Для всех методов получены собственные значения и собственные векторы матриц двумя способами: методом внутренней команды **Mathcad eigenvals(A)**, **eigenvecs(A)** и методом Якоби. Только в случае совпадения результатов по этими методами укрупнённый алгоритм считается правильным. В обсуждении приведены сравнения рассмотренных методов.*

**Ключевые слова**: *алгоритм, крупная операция, алгоритмический язык, исполнитель алгоритма, **Mathcad**-исполнитель укрупнённого алгоритма, алгоритмы в **Mathcad**, классические методы Якоби, Крылова для собственных значений и векторов матриц.*

## Введение.

Прблема определения собственных значений и векторов имеет вид:

$$Ax = \lambda x, x \neq 0, \lambda, x = ?, \qquad (1)$$

где $A$ - матрица порядка $n \times n$. В этом уравнении неизвестные: **n**-скалярных величин $\lambda$ и **n** собственных векторов $x \neq 0$. Теорией и применением проблемы занимались академики С.Л. Соболев, И.А.Крылов, Д.К. Фаддеев, Г.И.Марчук, А.А. Самарский, Н.Н. Яненко, С.К. Годунов, Н.С. Бахвалов, а также Якоби, Хаусхолдер, Уилкинсон, Фробениус, Френсис и др., что укзывает на акутуальность проблемы собственных значений и векторов.

Поскольку, задача (1) однородная система линейных уравнений по **x**, решение существует тогда и только тогда когда:

$$\det(A - \lambda E) = |A - \lambda E| = (-1)^n (\lambda^n - p_1\lambda^{n-1} - \dots p_{n-1}\lambda - p_n) = 0\,. \qquad (2)$$

Алгебраическое уравнение (2) называется характеристическим уравнением (ХУ) матрицы. Здесь коэффициенты характеристического уравнения определяются через диогональные миноры [1]:

$$p_1 = \sum_{i=1}^{n} a_{ii},\ p_2 = \sum_{i<j} \begin{vmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{vmatrix},\ p_3 = \sum_{i<k<j} \begin{vmatrix} a_{ii} & a_{ik} & a_{ij} \\ a_{ki} & a_{kk} & a_{kj} \\ a_{ji} & a_{jk} & a_{jj} \end{vmatrix}, \ldots,\ p_n = \det(A). \qquad (3)$$

Классический способ поиска собственных значений состоит из двух этапов:

1) определение коэффициентов характеристичесого уравнения;

2) решение характеристического уравнения .

Для поиска собственных значений симметрических матриц разработаны итерационные методы вращений Якоби (1846), Гивенса (1959), Хаусхолдера (1964). Для любых матриц разработаны классические универсальные методы Леверрье (1840), Крылова (1931), Данилевского (1937), Фаддеева (1963), интерполяции, Рутисхаузера LU (1958), Френсиса, Кублановской QR (1961) и итерационные методы поиска экстремальных собственных значений.

Теоретические основы задачи изложены в [1-7]. Алгебраический подход к приближёным методам собственных значений имеется в [8,9]. Задача о собственных значений для несимметрических матриц рассмотрены в [8]. Программы на языке Algol имеются в [9]. В работе [10] имеются программы методов для экстремальных собственных значений и метода интерполяции на трёх языках Бейсик, Паскаль и Фортран. Программы на языке Бейсик имеются и в [5] для экстремальных собственных значений, Якоби, QR. Появились программы на языке Python [11]. В [12,13] имеются укрупнёные алгоритмы нахождения собственных значений в Mathcad. В [14] имеются большое количество программ в Mathcad. В [15] изложены алгоритмы и программы в Mathcad классического метода, методов Крылова, Леверрье, Фаддеева, интерполяции, многообещающего итерационного метода QR.

В математических системах решение задачи можно найти 4-мя способами:

1) на основе команд-система-большой калькулятор;

2) на основе математического алгоритма-система испольнитель алгоритма;

3) на основе внутренного языка-система среда программирования;

4) на основе интерактивного способа-система-обучающая программа .

Для решения задач методов вычислений важную роль играет второй способ решения задач, так как в этом способе задача решается только с помощью математического алгоритма решения задачи. Matchad выступает в качестве испольнителя *укрупнённого алгоритма*, а пользователь играет роль создателя алгоритма. *Под укрупнённым алгоритмом мы понимаем последовательность математических формул, и команд математической системы, дающую решение задачи*. Укрупнённые команды, это, вычисления степеней матрицы, определителя, обратной матрицы, интеграла, решение дифференциального уравнения, и т.д. [12-13]. Наши мысли, мы демонстрировали примерами: нахождение собственных значений и векторов матриц классическими методами акад. Крылова [2], Леверрье, акад.Фаддеева и Данилевского [3], Хаусхолдера[6]. В этих методах сначала находиться вековое уравнение, и затем, собственные значения и собственные векторы. Адекватность результатов наших алгоритмов мы проверяем внутренней функцией системы по нахождению собственных значений и векторов матрицы. Обозначим через -матрицу, единичную матрицу.

**I. Методы решения. Укрупнённые алгоритмы и программы.**

**1. Метод команд системы Mathcad.** Вычислим собственные значения

и векторы по команде Mathcad и проверим правильность результата по формуле **:**

ORIGIN: =1 $A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$ $n := cols(A)$ $E := identity(n)$ //задание матриц A,E

**// ,v**

//проверка правильности **,v**

**2. Классический универсальный метод**. Этапы метода:

1) Вычисление векового уравнения: .

2) Решение векового уравнения: .

Ранее, этот метод применялся только для матриц малых порядков: **.**

Для двух этапного процесса составим укрупнённый алгоритм:

ORIGIN: =1 $A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$ $n := cols(A)$ $E := identity(n)$ //задание матриц A,E

// вычисление ХУ по команде $p(x) := D(x)\,coeffs, x \rightarrow \begin{bmatrix} 160 & 64 & -28 & -8 & 1 \end{bmatrix}^T$ // выделение коэффициентов ХУ

$r := polyroots(p)$ $r = \begin{bmatrix} -2.828 & -2 & 2.828 & 10 \end{bmatrix}^T$ // вывод собственных значений

//проверка

**3.** Метод Якоби (1846) для симметрических матриц.

Метод используется для решения полной проблемы собственных значений симметрической матрицы и основан на преобразовании подобия исходной матрицы A с помощью ортогональной матрицы H. Две матрицы $A$ и $B$ называются подобными (A~B)или B~A), если $B = H^{-1}AH$ или $A = HAH^{-1}$, где H — невырожденная матрица. В методе вращений в качестве H берется ортогональная матрица, такая, что $HH^T = H^T H = E \Leftrightarrow H^T = H^{-1}$, которая зависит от угла $\varphi^{(k)}$:

$$H^{(k)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\varphi^{(k)} & 0 & 0 & -\sin\varphi^{(k)} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \sin\varphi^{(k)} & 0 & 0 & \cos\varphi^{(k)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Для преобразованной матрицы А сохраняется собственные значения. При реализации метода вращений преобразование подобия применяется к исходной матрице А многократно:

$$A^{(k+1)} = (H^{(k)})^{-1} A^{(k)} H^{(k)} = (H^{(k)})^T A^{(k)} H^{(k)}, k = 0,1,...$$

до тех пор все недиагональные элементы обращались в нуль или же стали меньше чем заданного малого числа $\varepsilon$. Максимальный недиагональный элемент $A_{ij}^{(k)} = \max\{|A_{ij}^{(k)}|, i < j\}$ назовьём ведущим элементом k-ой итерации. Алгоритм метода Якоби изобразим в следующей таблице:

| Шаги | Проделываемые операции в одной итерации |
|---|---|
| 1 | Задание $k = 0, A^{(0)} = A, \varepsilon$-номер, матрица шага и точность расчётов |
| 2 | Определение ведущего элемента: $A_{ij}^{(k)} = \max\{|A_{ij}^{(k)}|, i < j, i \neq j\}$. Если $|A_{ij}^{(k)}| < \varepsilon$ для всех $i < j$, то положить $\lambda_i(A^{(k)}) = A_{ij}^{(k)}, i = 1..n$ и $X^{(k)} = H^{(0)}...H^{(k-1)}, AX^{(j)} = \lambda_j X^{(j)}$.Если $|A_{ij}^{(k)}| > \varepsilon$, процесс продолжается. |
| 3 | Определение угла поворота $p_k = tg2\varphi^{(k)} = 2A_{ij}^{(k)} / (A_{ii}^{(k)} - A_{jj}^{(k)})$, $s_k = \sin\varphi^{(k)} = signp_k \sqrt{0.5(1 - 1/\sqrt{1 + p_k^2})}, c_k = \cos\varphi^{(k)} = \sqrt{0.5(1 + 1/\sqrt{1 + p_k^2})}$ |
| 4 | Составление матрицу вращения $H^{(k)}$ |
| 5 | Вычисление очередного приближения $A^{(k+1)} = (H^{(k)})^T A^{(k)} H^{(k)}$. Вычисление меру близости $A^k$ к диагональной форме: $t(A^{(k+1)}) = t(A^{(k)}) - (A_{ij}^{(k)})^2$ . Положить k=k+1 и перейти к п.2 |

Пример 1. Итерация №1. Укажем вычисления в таблице:

| Шаги | Операции | Процесс вычисления |
|---|---|---|
| 1 | Задание $k = 0, A^{(0)} = A, \varepsilon$ | $A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}, \varepsilon = 10^{-10}$ |
| 2 | Ведущий элемент: $A_{ij}^{(k)} = \max\{|A_{ij}^{(k)}|, i < j, i \neq j\}$, | $A_{ij} = A_{12} = 1$.Так как $|A_{1,2}| > \varepsilon$ ,то процесс продолжается |

| | | |
|---|---|---|
| | Если $\left|A_{ij}^{(k)}\right| \le \varepsilon$, $i<j$, стоп иначе перейти п.3 | |
| 3 | Угол поворота $p_k = tg2\varphi^{(k)} = 2\dfrac{A_{ij}^{(k)}}{A_{ii}^{(k)} - A_{jj}^{(k)}}$ | $p_k = 2(A_{ij}^{(k)})/(A_{ii}^{(k)} - A_{jj}^{(k)}) = -2$ <br> $s_k = \sin\varphi^{(k)} = signp_k\sqrt{0.5(1-1/\sqrt{1+p_k^2})} = 0.58471028466$ <br> $c_k = \cos\varphi^{(k)} = \sqrt{0.5(1+1/\sqrt{1+p_k^2})} = 0.81124218518$ |
| 4 | Матрица вращения $H^{(k)}$ | $H^{(0)} = \begin{bmatrix} c_0 & -s_0 \\ s_0 & c_0 \end{bmatrix} = \begin{bmatrix} 0.81124218518 & 0.58471028466 \\ -0.58471028466 & 0.81124218518 \end{bmatrix}$ |
| 5 | Очередное приближение $A^{(k+1)} = (H^{(k)})^T A^{(k)} H^{(k)}$. $t(A^{(k+1)}) = t(A^{(k)}) - (A_{ij}^{(k)})^2 \to 0$ Положить k=k+1 и перейти к п.2 | $A^{(1)} = (H^{(0)})^T A^{(0)} H^{(0)} =$ <br> $\begin{bmatrix} 0.81124218518 & 0.58471028466 \\ -0.58471028466 & 0.81124218518 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} *$ <br> $\begin{bmatrix} 0.81124218518 & 0.58471028466 \\ -0.58471028466 & 0.81124218518 \end{bmatrix} = \begin{bmatrix} 9.3245553203 & 0 \\ 0 & -3.3245553203 \end{bmatrix}$ |

Ведущий недиагональный элемент равен нулю и точность достигнута за одну итерацию. Найдены следующие собственные значения и векторы:

$\lambda_1$= 9,3245553203;$\lambda_2$= -3,3245553203,

$$X = H^{(0)} = \begin{bmatrix} X^1 \\ X^2 \end{bmatrix} = \begin{bmatrix} 0.81124218518 & 0.5847102847 \\ -0.5847102847 & 0.81124218518 \end{bmatrix}$$

Пример 2. Итерация №1.

| Шаги | Операции | Процесс вычисления |
|---|---|---|
| 1 | Данные итерации $k=1, A^{(0)} = A, \varepsilon = 0.001$ | $A0 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$ $r0 = r(A0) = \sum_{i<j}(A0_{i,j})^2 = 50$ |
| 2 | Ведущий элемент: $A_{ij}^{(k)} = \max\{\left|A_{ij}^{(k)}\right|, i<j, i \ne j\}$, Если $\left|A_{ij}^{(k)}\right| \le \varepsilon$, $i<j$, стоп иначе перейти к п.3 | $A_{ij} = A_{14} = 4$. Так как $\left|A_{1,4}^{(0)}\right| = 4 > \varepsilon = 0.001$, то процесс продолжается |
| 3 | Угол поворота $p_k = tg2\varphi^{(k)} = 2\dfrac{A_{ij}^{(k)}}{A_{ii}^{(k)} - A_{jj}^{(k)}}$ | $p_1 = 2A_{14}^{(0)}/(A_{11}^{(0)} - A_{44}^{(0)}) = -4$ <br> $s_1 = \sin\varphi^{(1)} = signp_k\sqrt{0.5(1-1/\sqrt{1+p_1^2})} = -0.6154$ <br> $c_1 = \cos\varphi^{(1)} = \sqrt{0.5(1+1/\sqrt{1+p_1^2})} = 0.7882$ |

| | | |
|---|---|---|
| 4 | Матрица вращения $H^{(0)}$ | $H^{(0)}=\begin{bmatrix} 0.7882 & 0 & 0 & -0.6154 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.6154 & 0 & 0 & 0.7882 \end{bmatrix}$ |
| 5 | Первая итерация $A^{(k+1)}=(H^{(k)})^T A^{(k)} H^{(k)}$. $t(A^{(k+1)})=t(A^{(k)})-2(A_{ij}^{(k)})^2 \to 0$ Положить k=k+1 и перейти к п.2 | $A^{(1)}=(H^{(0)})^T A^{(0)} H^{(0)}=\begin{bmatrix} 5.6381 & 2.1918 & 3.5954 & 1.9403 \\ 2.1918 & 3 & 4 & -0.4426 \\ 1.1338 & 4 & 1 & -0.2698 \\ 1.9403 & -0.4426 & -0.2698 & -1.628 \end{bmatrix}$ $r1=\sum_{i<j}(A1_{i,j})^2=37.7647$ |

Итерация №2.

| Шаги | Операции | Процесс вычисления |
|---|---|---|
| 1 | Данные итерации $k=2, A^{(1)}=A, \varepsilon=0.001$ | $A^{(1)}=\begin{bmatrix} 5.6381 & 2.1918 & 3.5954 & 1.9403 \\ 2.1918 & 3 & 4 & -0.4426 \\ 1.1338 & 4 & 1 & -0.2698 \\ 1.9403 & -0.4426 & -0.2698 & -1.628 \end{bmatrix}$ |
| 2 | Ведущий элемент: $A_{ij}^{(k)}=\max\{\lvert A_{ij}^{(k)}\rvert, i<j, i\neq j\}$ Если $\lvert A_{ij}^{(k)}\rvert \leq \varepsilon$, $i<j$, стоп иначе перейти к п.3 | $A_{ij}=A_{23}=4>\varepsilon$. Процесс продолжается |
| 3 | Угол поворота $p_k=tg2\varphi^{(k)}=2\dfrac{A_{ij}^{(k)}}{A_{ii}^{(k)}-A_{jj}^{(k)}}$ | $p_k=2A_{ij}^{(k)}/(A_{ii}^{(k)}-A_{jj}^{(k)})=4$ $\sin\varphi^{(k)}=signp_k\sqrt{0.5(1-1/\sqrt{1+p_k^2})}=0.6154$ $\cos\varphi^{(k)}=\sqrt{0.5(1+1/\sqrt{1+p_k^2})}=0.7882$ |
| 4 | Матрица вращения $H^{(1)}$ | $\mathrm{H}^{(1)}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.7882 & -0.6154 & 0 \\ 0 & 0.6154 & 0.7882 & 0 \\ 0 & 0 & -0.4477 & 1 \end{bmatrix}$ |
| 5 | Вторая итерация $A^{(k+1)}=(H^{(k)})^T A^{(k)} H^{(k)}$. $t(A^{(k+1)})=t(A^{(k)})-2(A_{ij}^{(k)})^2 \to 0$ Положить k=k+1 и перейти к п.2 | $A^{(2)}=(H^{(2)})^T A^{(1)} H^{(2)}=\begin{bmatrix} 5.638 & 3.9403 & 1.4851 & 1.9403 \\ 3.9403 & 6.1231 & 0 & -0.5149 \\ 1.4851 & 0 & -2.1231 & 0.05970 \\ 1.9403 & -0.5149 & 0.0597 & -0.638 \end{bmatrix}$ $r2=\sum_{i<j}(A2_{i,j})^2=21.7547$ |

**Итерация №3**

| Шаги | Операции | Процесс вычисления |
|---|---|---|
| 1 | Данные итерации $k=3, A^{(2)}=A, \varepsilon=0.001$ | $A^{(2)}=\begin{bmatrix} 5.638 & 3.9403 & 1.4851 & 1.9403 \\ 3.9403 & 6.1231 & 0 & -0.5149 \\ 1.4851 & 0 & -2.1231 & 0.05970 \\ 1.9403 & -0.5149 & 0.0597 & -0.638 \end{bmatrix}$ |
| 2 | Ведущий элемент: $A_{ij}^{(k)}=\max\{\lvert A_{ij}^{(k)}\rvert, i<j, i\neq j\}$, Если $\lvert A_{ij}^{(k)}\rvert \le \varepsilon$, $i<j$, стоп иначе перейти к п.3 | $A_{ij}=A_{12}=6.1231>\varepsilon$. Так как $A_{12}>\varepsilon=0.001$, то процесс продолжается |
| 3 | Угол поворота $p_k=tg2\varphi^{(k)}=2\dfrac{A_{ij}^{(k)}}{A_{ii}^{(k)}-A_{jj}^{(k)}}$ | $p_k=2A_{ij}^{(k)}/(A_{ii}^{(k)}-A_{jj}^{(k)})=-0.4851$ <br> $\sin\varphi^{(k)}=signp_k\sqrt{0.5(1-1/\sqrt{1+p_k^2})}=-0.685$ <br> $\cos\varphi^{(k)}=\sqrt{0.5(1+1/\sqrt{1+p_k^2})}=0.7285$ |
| 4 | Матрица вращения $H^{(2)}$ | $H^{(2)}=\begin{bmatrix} 0.7285 & -0.685 & 0 & 0 \\ 0.685 & 0.7285 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| 5 | Третья итерация $A^{(k+1)}=(H^{(k)})^T A^{(k)} H^{(k)}$. $t(A^{(k+1)})=t(A^{(k)})-2(A_{ij}^{(k)})^2 \to 0$ Положить k=k+1 и перейти к п.2 | $A^{(3)}=(H^{(2)})^T A^{(2)} H^{(2)}=\begin{bmatrix} 9.7985 & 0.4842 & 1.0819 & 1.0608 \\ 0.4842 & 1.9626 & -1.0173 & -1.7043 \\ 1.0819 & -1.0173 & -2.1231 & 0.0597 \\ 1.0608 & -1.7043 & 0.0597 & -1.638 \end{bmatrix}$ <br> $r(A^{(3)})=\sum_{i<j}(A3_{i,j})^2=6.4733$ |

Аналогично, вычисляем итерации с 4-го по 10-й, опуская вычисления переходим к 11-й итерации. Критерий правильности вычислений заключается в уменьшении суммы квадратов недиагональных элементов.

Итерация №11

| Шаги | Операции | Процесс вычисления |
|---|---|---|
| 1 | Данные итерации $k=11, A^{(10)}=A, \varepsilon=0.001$ | $A^{(10)}=\begin{bmatrix} 9.9999 & -0.0261 & 0.0217 & -0 \\ -0.0261 & 2.828 & -0.0514 & -0.0001 \\ 0.0217 & -0.0514 & -1.9994 & 0.0005 \\ -0 & -0.0001 & 0.0005 & -2.8284 \end{bmatrix}$ |
| 2 | Ведущий элемент: $A_{ij}^{(k)}=\max\{\lvert A_{ij}^{(k)}\rvert, i<j, i\neq j\}$ Если $\lvert A_{ij}^{(k)}\rvert \le \varepsilon$, $i<j$, стоп иначе перейти к п.3 | $A_{ij}=A_{23}=-0.0514>\varepsilon$. Так как $\lvert A_{2,3}^{(k)}\rvert>\varepsilon=0.001$, то процесс продолжается |

| | | |
|---|---|---|
| 3 | Угол поворота $p_k = tg2\varphi^{(k)} = 2\dfrac{A_{ij}^{(k)}}{A_{ii}^{(k)} - A_{jj}^{(k)}}$ | $p_k = 2A_{ij}^{(k)}/(A_{ii}^{(k)} - A_{jj}^{(k)}) = -0.0213$ <br> $\sin\varphi^{(k)} = signp_k\sqrt{0.5(1-1/\sqrt{1+p_k^2})} = -0.0107$ <br> $\cos\varphi^{(k)} = \sqrt{0.5(1+1/\sqrt{1+p_k^2})} = 0.9999$ |
| 4 | Составление матрицу вращения $H^{(11)}$ | $H^{(11)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9999 & 0.0107 & 0 \\ 0 & -0.0107 & 0.9999 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| 5 | Одинадцатая итерация $A^{(k+1)} = (H^{(k)})^T A^{(k)} H^{(k)}$. $t(A^{(k+1)}) = t(A^{(k)}) - 2(A_{ij}^{(k)})^2 \to 0$ Положить k=k+1 и перейти к п.2 | $A^{(11)} = (H^{(10)})^T A^{(10)} H^{(10)} = \begin{bmatrix} 9.9999 & -0.0264 & 0.0214 & -0 \\ -0.264 & 2.8285 & 0 & -0.0001 \\ 0.0214 & 0 & -2 & 0.0005 \\ -0 & -0.0001 & 0.0005 & -2.8284 \end{bmatrix}$ <br> $r(A^{(11)}) = \sum_{i<j}(A4_{i,j})^2 = 0.0012$ |

Положим k=12 и итерация №12

| Шаги | Операции | Процесс вычисления |
|---|---|---|
| 1 | Данные итерации $k = 12, A^{(11)} = A, \varepsilon = 0.001$ | $A^{(11)} = \begin{bmatrix} 9.9999 & -0.0264 & 0.0214 & -0. \\ -0.0264 & 2.8285 & 0. & -0.0001 \\ 0.0214 & 0. & -2. & 0.0005 \\ 0. & -0.0001 & 0.0005 & -2.8284 \end{bmatrix}$ |
| 2 | Ведущий элемент: $A_{ij}^{(k)} = \max\{\lvert A_{ij}^{(k)}\rvert, i<j, i \neq j\}$ Если $\lvert A_{ij}^{(k)}\rvert \le \varepsilon$, $i<j$, стоп иначе перейти к п.3 | $A_{ij} = A_{12} = -0.0264$. Так как $\lvert 0.0264\rvert > \varepsilon$ то процесс продолжается |
| 3 | Угол поворота $p_k = tg2\varphi^{(k)} = 2\dfrac{A_{ij}^{(k)}}{A_{ii}^{(k)} - A_{jj}^{(k)}}$ | $p_k = 2A_{ij}^{(k)}/(A_{ii}^{(k)} - A_{jj}^{(k)}) = -0.0074$ <br> $s_{12} = \sin\varphi^{(k)} = signp_k\sqrt{0.5(1-1/\sqrt{1+p_k^2})} = -0.0037$ <br> $c_{12} = \cos\varphi^{(k)} = \sqrt{0.5(1+1/\sqrt{1+p_{12}^2})} = 1$ |
| 4 | Матрица вращения $H^{(12)}$ | $H^{(12)} = \begin{bmatrix} 1 & 0.0037 & 0 & 0 \\ -0.0037 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| 5 | Двенадцатая итерация $A^{(k+1)} = (H^{(k)})^T A^{(k)} H^{(k)}$. $t(A^{(k+1)}) = t(A^{(k)}) - 2(A_{ij}^{(k)})^2 \to 0$ Положить k=k+1 и перейти к п.2 | $A^{(12)} = (H^{(12)})^T A^{(11)} H^{(12)} = \begin{bmatrix} 10 & -0.0403 & 0.0251 & 0.061 \\ -0.0403 & 2.8284 & 0.001 & -0.0026 \\ 0.0251 & 0.001 & -2 & 0 \\ -0.0061 & -0.0026 & 0 & -2.8384 \end{bmatrix}$ <br> $r12 = \sum_{i<j}(A12_{i,j})^2 = 0.0005 < 0.001$ |

**Мы нашли то же решение, полученное ранее:**

$$\lambda = \begin{bmatrix} 10 \\ 2.8284 \\ -2 \\ -2.8284 \end{bmatrix}, X = (H^{(0)})^T * ... * (H^{(11)})^T = \begin{bmatrix} -0.6533 & -0.5 & 0.2706 & 0.5 \\ -0.2706 & 0.5 & -0.6533 & 0.5 \\ 0.6533 & -0.5 & -0.2706 & 0.5 \\ 0.2706 & 0.5 & 0.6533 & 0.5 \end{bmatrix} // \lambda, X$$ по Якоби

Можно составить программу во внутреннем алгоритмическом языке Mathcad, метода Якоби, состоящей из трёх подпрограмм:

| 1 -П. программа | 2 -П. программа | 3 -П. программа |
|---|---|---|
| $M(A,m) :=$ <br> for $i \in 1..n$ <br> $D_{i,i} \leftarrow A_{i,i}$ <br> $AD \leftarrow A - D$ <br> $m \leftarrow \max(AD)$ <br> for $j \in 1..n$ <br> for $i \in 1..n$ <br> $I \leftarrow i$ if $AD_{i,j} = m$ <br> $J \leftarrow$ if $AD_{i,j} = m$ <br> $\begin{pmatrix} I \\ J \\ m \end{pmatrix}$ <br><br> Вычисления: $M := M(A,m)$ <br> M=[1 4 4] | $U(A) :=$ <br> $H \leftarrow E$ <br> $I \leftarrow M(A,m)_1$ <br> $J \leftarrow M(A,m)_2$ <br> $x \leftarrow A - D$ <br> $m \leftarrow 2 * A_{I,J}$ <br> $y \leftarrow A_{I,I} - A_{J,J}$ <br> $z \leftarrow \sqrt{x^2 + y^2}$ <br> $sign \leftarrow (x/y)/\lvert x/y \rvert$ <br> $s \leftarrow sign * \sqrt{(z-y)/(2*z)}$ <br> $c \leftarrow \sqrt{(z+y)/(2*z)}$ <br> $H_{I,I} \leftarrow c$ <br> $H_{J,J} \leftarrow c$ <br> $H_{I,J} \leftarrow s$ <br> $H_{J,I} \leftarrow -s$ <br> $H$ | $Y(A,\varepsilon) :=$ <br> $k \leftarrow 0$ <br> while $M(A,m)_3 > \varepsilon$ <br> $k \leftarrow k+1$ <br> $H \leftarrow U(A)$ <br> $B \leftarrow H^T * A * H$ <br> $A \leftarrow B$ <br> for $i \in 1..n$ <br> $\lambda_i \leftarrow A_{i,i}$ <br> $\lambda$ <br><br> Вычисления: <br> $\lambda := Y(A,\varepsilon)$ <br> $\lambda = [9.999\ 2.828 - 2 - 2.828]$ |

**4.** Обсуждение результатов. **Мы рассмотрели самый первый** итерационный метод Якоби для симметрических матриц. Этот метод был придуман в эру решения математических задач ручным способом. Симметрическую проблему собственных значений можно было решить только для матриц небольших размеров $n \le 3$. Практика показывает, что итерации дают хороший результат для двумерных матриц на первой же итерации, для трёхмерных матриц за 5-6-итераций, для четырёхмерных матриц за 10-12 итераций. В каждой итерации выполнить следующие операции: 1) задать матрицу шага итерации; 2) определить ведущий элемент и проверить критерий окончания цикла; 3)найти уголь поворота и вычислить элементы матрицы вращения $c_k = \cos\varphi^{(k)}$, $s_k = \sin\varphi^{(k)}$, 4) задать матрицу вращения $H^{(k)}$; 5)вычислить очередную итерацию $A^{(k+1)} = (H^{(k)})^T A^{(k)} H^{(k)}$. Эти вычисления трудоёмкие для ручного вычисления. Автор получил с помощью математической системы приемлимый результат за 12 итераций.

Для сравнения вычислены собственные векторы и значения матрицы четвёртого порядка универсальными методами команд и классическим методом . Универсальные методы более удобны, чем метод Якоби.

**V.** Заключение. В работе исследован классические метод Якоби, определения собственных значений и векторов для симметрических матриц

Методы даёт хорошие результаты для матриц порядков . Для проверки результатов работы составлены укрупненные алгоритмы в Mathcad . Результат алгоритма совпадает с результатоми по внутренной команде вычисления собственных значений и векторов Mathcad.

**VI. Благодарственность**. Автор благодарить членов

кафедры «Информатика» НамГУ М.Дадахонова, Ш.Болтабоева, С. Иномиддинова, С.Настинова за подддержку и обсуждения результатов.

## ЛИТЕРАТУРА:

1.Крылов В.Н., Бобков В.В., Монастирный П.И. Вычислительные методы. М.: Наука, т.1,1976,304 с.

2. Крылов И.А. «О численном решении уравнения, которым в технических вопросах определяются частоты малых колебаний материальных систем». ИАН, ОМЕН, 1931, №4, 491-539 с.

3. Данилевский А.М. О численном решении векового уравнения. Матем. сборник, 1937. №2, [44], 169-171 с.

4. Шарый С.П. Курс вычислительных методов . ИВТ СОРАН, Н.:2018.-607 с.

5. Шуп Т.Е. Прикладные численные методы в физике и технике.М.:ВШ, 1990.-256 с.

6.Burden R.L. Numerical Analysis. Books Cole. Boston, USA.-2010.-895 p.

8.Икрамов Х.Д. Несимметрическая проблема собственных значений. М.:Наука, 1991.-240 с.

9.Уилкинсон Дж., Райнш К. Справочник алгоритмов на языке Алгол. М.:Машиностроение, 1976.-390 с.

10.Мудров Е.М. Программы для ПК на языке Бейсик, Паскаль, Фортране. Томск, МП Раско, 1992.-272 с.

11.Kiusalaas J. Numerical Methods in Ingineering with Phyton 3. NY.-2013.-438p.

12.Имомов А.Решение польной проблемы собственных значений матриц в Mathcad. Научный вестник НамГУ, 2021, №5, с. 62-67.

13.Имомов А. Mathcad-испольнитель укрупнённых алгоритмов. Научный вестник НамГУ, 2022, №1, с. 62-69.

14. Охорзин В.А. Прикладная математика в системе Mathcad. СПб, 2008.-350с.

15. Имомов А. Алгоритмы и программы собственных значений матриц. Современное состояние и перспективы применения цифровых технологий и искусственного интеллекта в управлении. : г. Самарканд, 26-27 октября 2022 г. : в 2 ч. Ч. 2 Ташкент: Изд-во НИИ РЦТИИ, 2022. – 394 с.