2 – SON / 2022 - YIL / 15 - OKTYABR

АЛГОРИТМЫ И ПРОГРАММЫ ПОЛНОЙ ПРОБЛЕМЫ СОБСТВЕННЫХ ЗНАЧЕНИЙ

Имомов Адаш

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

Аннотация: *В статье рассматриваются классические методы нахождения польной проблемы собственных значений матриц, такие как непосредственное нахождение с помощью двухступенчатого метода по определению, методы Леверрье, Фадеева, Крылова, интерполяции, итерационного устойчивого метода Френсиса QR. Для всех методов построены укрупнёые алгоритмы, состоящие из последовательности математических формул и команд математической системы Mathcad. Здесь Mathcad выступает как испольнитель укрупнённого математического алгоритма решения задачи. Для всех методов получены собственные значения матриц двумя способами: методом внутренной команды Mathcad eigenvals(A), и самым предлагаемым методом. Только в случае совпадения результатов по этим и методами укрупнённый алгоритм считается правильным. В обсуждении рассмотрена матрица десятого порядка, для которой по всем предлагаемыми методами получены почти совпадающие численные величины собственных значений.*

Ключевые слова: *алгоритм, крупная операция, алгоритмический язык, испольнитель алгоритма, Mathcad-исполнитель укрупнённого алгоритма, алгоритмы и программы в Mathcad для собственных значений матриц*

Введение.

Многие научные и технические задачи приводят к нахождению собственных значений и векторов матриц [1]. Например, при рассмотрении тензора напряжений собственные значения задают главные нормальные напряжения, а собственные векторы определяют ориентации, соответствующим этим нормальным направлениям. При анализе динамических систем собственные значения определяют частоты колебаний, а собственные векторы характеризуют эти формы. При анализе конструкций собственные значения используют для определения критических нагрузок изгиба или других видов нестабильности.

Формулировка задачи определения собственных значений имеет вид:

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

где A - матрица порядка $n \times n$. В этом уравнении неизвестные: n-скалярных величин λ и n собственных векторов $x \neq 0$, соответствующих каждому из этих собственных значений . Поскольку, задача (1) однородная система линейных уравнений по x, решение существует когда матрица системы имеет детерминант, равного нулю:

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

Это алгебраическое уравнение называется характеристическим (вековым) уравнением (в комментариях в алгоритмах его обозначим кратко как ХУ), где

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

Число диагональных миноров k-го порядка матрицы A равно C_n^k, и вычисление коэффициентов характеристического полинома (1) эквивалентно

вычислению определителей различных порядков в количестве [1]

$$C_n^1+C_n^2+...+C_n^n=2^n-1,\quad C_n^k=n!/k!(n-k)!, k=1..n \qquad (3)$$

Классический способ поиска обственных значений состоит из двух этапов:

1) определение коэффициентов характеристичесого уравнения;

2) решение характеристического уравнения.

Для любых матриц разработаны итерационные методы поиска экстремальных собственных значений . Для поиска всех собственных значений симметрических матриц разработаны итерационные методы вращений Якоби, Гивенса. Для произвольных матриц имеются классические методы Леверрье, Фадеева, Крылова, интерполяции, Френсиса QR.

Теоретические основы задачи изложены в [1-3]. Из зарубежных работ укажем [4]. Алгебраический подход к приближёным методам собственных значений имеется в [5,6]. Задача о собственных значений для несимметрических матриц рассмотрены в [5], где проанализированы программы, имющие к этому времени. Программы на языке Algol имеются в [6]. В работе [7] впервые появились программы методов для экстремальных собственных значений и метода интерполяции на трёх языках Бейсик, Паскаль и Фортран. Программы на языке Бейсик имеются по для экстремальных собственных значений, Якоби, QR. Появились программы -укрупнённые алгоритмы на языке Python [8]. В [9,10] имеются укрупнёные алгоритмы нахождения собственных значений, исполняемые математической системой Mathcad.

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 ползователь играет роль создателя алгоритма. *Под укрупнённым алгоритмом мы понимаем последовательность* *математических формул, и команд математической системы, дающую решение задачи*. Именно, эти команды математической системы и являются крупной командой, например, внутренняя функция вычисления степеней матрицы, определителя, обратной матрицы, интеграла, решение дифференциального уравнения, и т.д. Наши мысли, мы демонстрируем примерами: нахождение собственных значений матрицы классическими методоми непосредственного определения, Леверрье, Фадеева, интерполяции и Крылова, QR-алгоритма. Во всех этих методах сначала находиться вековое уравнение, а затем, решив его найдём собственные значения. Результаты наших алгоритмов мы проверяем внутренной функцией eigenvals(A) по нахождению собственных значений.

Результаты вычислений по алгоритмам и внутренной функции одинаковые.

В дальнейшем обозначим через $A=[a_{ij}]_{i,j=1}^{n}$-матрицу, E – единичная матрица. В теории собственных значений важное значение имеют известные разложения матриц по Жордану, Шуру [2]. Всякая матрица подобна жордановой и шуровой формам: $A=SJS^{*}\,(A=SDS^{*}), A=QRQ^{*}$, где J-жорданова форма, D-диогональная форма, R-верхняя треугольная форма, которые содержать все собственные значения, Q-ортогональная матрица. QR-алгоритм есть конструктивный способ построения разложения $A=QRQ^{*}$ [2].

1. Методы решения. Укрупнённые алгоритмы и программы.

Укрупнённый алгоритм 1. Метод непосредственного развёртывания векового определителя. Этапы метода:

1) Вычисление векового уравнения: $D(\lambda)=|A-\lambda E|=(-1)^{n}(\lambda^{n}-\sum_{i=1}^{n}p_{i}\lambda^{i})=0$.

2) Решение векового уравнения: $D(\lambda)=|A-\lambda E|=0, \lambda=\lambda_{i}, i=1..n$.

Ранее, этот метод применялся только для матриц малых порядков: $n\leq 3$.

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

$D(x):=|E-x*A|$ $D(x)\rightarrow 160+64*x-28*x^{2}-8*x^{3}+x^{4}$ // построение ХУ

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

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

Данный метод универсальный, он находить все собственные значения.

Укрупнённый алгоритм 2. Метод Леверрье. [1]. Алгоритм метода:

1)вычисляются степени матрицы $A^{k}, k=1..n$;

2) вычисляются следы: $s_{k}=trace(A^{k})=a_{11}^{k}+..+a_{nn}^{k}, k=1..n$;

3) вычисляются коэффициенты ХУ по формулам Ньютона:

$p_1 = -s_1, p_k = -(s_k + p_1 s_{k-1} + .. + p_{k-1} s_1)/k, k = 2..n$;

4) решив вековое уравнение $\lambda : D(\lambda) = 0$ найдем собственные значения;

5) с помощью внутренней функции проверим результат.

Этапы вычислений трудоемкие. Mathcad упрощает задачу:

ORIGIN: =1 $A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$ $n := cols(A)$ $E := identity(n)$ $D(x) := |A - E * x|$

//матрицы $k := 1..n$ $s_k := tr(A^k)$ $s^T = (8\ 120\ 992\ 1.014*10^4)$ $p_1 := -s_1$ //вычисление следа матрицы

$k := 2..n$ $p_k := -(s_k + \sum_{i=1}^{k-1} p_i * s_{k-i})/k$ $p^T = (-8\ -28\ 64\ 160)$ // часть коэффициентов p_i

$i := 1..n$ $q_i := p_{n-i+1}$ $q^T = (160\ 64\ -28\ -8)$ $P := stack(q,1)$ $P^T = (160\ 64\ -28\ -8\ 1)$ //все p_i

$r := polyroots(P)$ $r^T = (-2.828\ -2\ 2{,}828\ 10)$ //внутренняя команда и все λ

3.Программа метода Фадеева. Академик Д.К. Фадеев видоизменил метод Леверрье, который в Mathcad записывается так: $i := 1..n$ $A_i := AB_{i-1}$ $p_i := tr(A_i)/i$ $B_i := A_i - p_i E$, где p_i-коэффициенты характеристического полинома, $A^{-1} = B_{n-1}/p_n$ -обратная матрица, на правильность вычислений показывает равенство $B_n = 0$.

Приведём укрупнённый алгоритм.

ORIGIN:=1 n:=4 A:= $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$ E:=identity(n) //матрицы, ХУ уравнение по

D(x):=(-1)^n $|A - E * x| \rightarrow 160 + 64x - 28x^2 - 8x^3 + x^4$ // ХУ команде Mathcad

$C_1 := A$ $p_1 := tr(C_1)$ $B_1 := C_1 - E * p_1$ // первый шаг метода Фадеева

Дальше приведём подпрограмму метода Фадеева, которая находить все $\lambda = \lambda_i$.

```
Fadeev(A,B,C,n) := | C1 ← A
                   | p1 ← tr(C1)
                   | B1 ← C1 − p1 · E
                   | for i ∈ 2..n
                   |    | Ci ← A · B(i−1)
                   |    | pi ← tr(Ci) / i
                   |    | Bi ← Ci − pi · E
                   | for j ∈ 1..n
                   |    qj ← −p(n−j+1)
                   | Q ← stack(q,1)
                   | r ← polyroots(Q)
                   | Q
                   | r
r := Fadeev(A,B,C,n)                         r^T = ( −2.828  −2  2.828  10 )
```

Укрупнённый алгоритм 4. Метод интерполяции. [1]. Идея метода: $D(i)=|A-i*E|=(-1)^n*(i^n-\sum_{j=1}^{n}p_j i^{n-j})=d_i, i=1..n$. Это даст нам систему линейных алгебраических уравнений относительно коэффициентов векового уравнения $Mp=b$. Решив её построим характеристическое уравнение, решив её в свою очередь найдём собственные значения:

ORIGIN:=1 n:=4 E:=identity(n) $A:=\begin{bmatrix}1&2&3&4\\2&3&4&1\\3&4&1&2\\4&1&2&3\end{bmatrix}$ i:=1..n j:=1..n $D_i:=|A-i*E|$ $d_i:=(-1)^n*D_i$

$d=\begin{bmatrix}189\\128\\-35\\-288\end{bmatrix}$ $M_{i,j}:=i^{n-j}$ $M=\begin{bmatrix}1&1&1&1\\8&4&2&1\\27&9&3&1\\64&16&4&1\end{bmatrix}$ $p=M^{-1}*b$ $b_i:=i^n-d_i$ $b=\begin{bmatrix}-188\\-112\\116\\544\end{bmatrix}$ // $Mp=b$

$p^T=\begin{bmatrix}8&28&-64&-160\end{bmatrix}$ $q_i=-p_{n-i+1}$ $q^T=\begin{bmatrix}160&64&-28&-8\end{bmatrix}$ $P:=stack(q,1)$ $P^T=[160\ 64\ -28\ -8\ 1]$

$r:=polyroots(P)$ $r=\begin{bmatrix}-2.88&-2&2.828&10\end{bmatrix}$ $r1:=eigenvals(A)$ $r1^T=\begin{bmatrix}-2.828&-2&2.828&10\end{bmatrix}$

Укрупнённый алгоритм 5. Метод Крылова. [1].

Пусть $D(\lambda)=|A-\lambda E|=(-1)^n(p_1+p_2\lambda+...+p_{n-1}\lambda+p_n\lambda^{n-1}+\lambda^n)=0$ вековое уравнение матрицы A. По теореме Гамильтона-Келли сама матрица удовлетворяет вековое уравнение: $D(A)=0: D(A)=|A-AE|=0$. Поэтому имеем

$$D(A)=|A-AE|=(-1)^n(p_1+p_2A+...+p_{n-1}A^{n-1}+p_nA^n)=0 \Rightarrow (p_1+p_2A+...+p_{n-1}A^{n-1}+p_nA^n)=0.$$

Отсюда получаем $p_1E+p_2A+...+p_nA^{n-1}=-A^n$.Последнее уравнение умножим на произвольный ненулевой вектор $x\in R^n$ и получаем векторное уравнение:

$p_1 Ex + p_2 Ax + ... + p_n A^{n-1} x = -A^n x$.

Вводя векторы $x^{(k)} := A^{(k)} * x, k = 1..n$, для неизвестных коэффициентов векового уравнения получаем систему линейных уравнений:

$$\begin{bmatrix} x_{11} & x_{12} & . & x_{1n} \\ x_{21} & x_{22} & . & x_{2n} \\ . & . & . & . \\ x_{n1} & x_{n2} & . & x_{nn} \end{bmatrix} * \begin{bmatrix} p_1 \\ p_2 \\ . \\ p_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ . \\ d_n \end{bmatrix}, x_{ij} = (A^j * x)_i .$$

Укрупнённый алгоритм в Mathcad.

$$\text{ORIGIN:=1} \quad \text{n:=4} \quad A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix} \quad x := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad E := identity(n)$$

$$\text{k:=1..n} \quad B^{<k>} := A^{k-1} * x \quad B = \begin{bmatrix} 1 & 1 & 30 & 240 \\ 0 & 2 & 24 & 244 \\ 0 & 3 & 22 & 256 \\ 0 & 4 & 24 & 260 \end{bmatrix} \quad d := (-1) * A^n * x \quad p := B^{-1} * d$$

$$p^T = [160 \quad 64 \quad -28 \quad -8] \quad P := stack(p,1) \quad P^T = [160 \quad 64 \quad -28 \quad -8 \quad 1]$$

r:=polyroots(P) s:=eigenvals(A)

$r = [-2.828 \quad -2 \quad 2.828 \quad 10] \quad s = [-2.828 \quad -2 \quad 2.828 \quad 10]$

Укрупнённый алгоритм 6. Программа QR –алгоритма. [2, 3]. Последовательность матриц

$$A_m =: Q_m * R_m, A_{m+1} := R_m * Q_m, m = 1, 2, ..., A_1 = A,$$

стремится к матрице собственных значений в главной диагонали, где Q-ортогональная, R-верхнее треугольная матрицы. Этот метод позволяет найти все собственные значения и является устойчивым методом. Здесь =: означает “правая часть по определению равно”, а-:= означает “левая часть по определению равно” . Приведём укрупнённый QR-алгоритм метода:

$$ORIGIN := 1 \quad A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix} \quad p := eigenvals(A) \quad p = \begin{bmatrix} -2.828 \\ -2 \\ 2.828 \\ 10 \end{bmatrix} \text{// реш в командах}$$

$$M := qr(A) \quad r := row(M) \quad s := cols(M) \quad r = 4 \quad s = 8 \quad R := identity(r) \quad Q := identity(r) \quad n := 4$$

Составим подпрограмму метода QR в инутренноя языке Mathcade:

$$QR(A,n,Q,R,M) := \left| \begin{array}{l} \text{for } k \in 1..n \\ \quad \left| \begin{array}{l} M \leftarrow qr(A) \\ Q \leftarrow \text{submatrix}(M,1,r,1,r) \\ R \leftarrow \text{submatrix}(M,1,r,r+1,2\cdot r) \\ A \leftarrow Q^{-1}\cdot A\cdot Q \end{array} \right. \\ Q \\ R \\ M \end{array} \right.$$

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad z := QR(A,n,Q,R,M)$$

$$z = \begin{pmatrix} 1 & 0 & 0.00002 & 0 & 10 & -0.00011 & 0.00011 & -0 \\ -0.00001 & -0.70469 & 0.70401 & 0.08822 & 0 & -2.82705 & 0.00275 & 0.06235 \\ 0.00002 & -0.70332 & -0.70951 & 0.04394 & 0 & 0 & 2.82705 & -0.06229 \\ -0 & 0.09352 & -0.03108 & 0.99513 & 0 & 0 & 0 & -2.00195 \end{pmatrix}$$

Здесь в матрице z на месте треугольной R матрицы в главной диагонали расположены собственные значения матрицы A.

2. Обсуждения результатов.

Рассмотрим симметрическую матрицу десятого порядка:

$A=[a_{i,j}]_{i,j=1}^{n}, a_{1,1}=a_{10,10}=1; a_{i,i}=4, i=2..9; a_{i+1,i}=a_{i,i+1}=1; a_{i,j}=0$ – для остальных i,j.

Вычислим собственные значения этой матрицы выше приведёнными укрупнёными алгоритмами методов, заполнив следующую таблицу:

№/ метод	команда системы	классическое определение	Леверрье	Фадеев	Интер-поляция	Крылов	QR
1	1.502	1.502	1.492	1.492	1.492	1.492	5.895
2	1.498	1.498	1.508	1.508	1.508	1.508	5.511
3	2.182	2.182	2.181	2.181	2.181	2.181	5.118
4	2.622	2.622	2.624	2.624	2.624	2.624	4.522
5	3.206	3.206	3.204	3.204	3.204	204	3.863
6	3.863	3.863	6.864	6.864	3.864	3.864	3.206
7	4.522	4.522	4.524	4.524	4.524	4.524	2.622
8	5.118	5.118	5.1	5.112	5.112	5.112	2.182
9	5.591	5.591	5.598	5.598	5.508	5.508	1.501
10	5.895	5.895	5.893	5.893	5.893	5.893	1.499

Таблица 1. Сравнительная таблица собственных значений.

z =

	11	12	13	14	15	16	17	18	19	20
1	5.895	0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	5.591	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	5.118	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	4.522	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	3.863	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	3.206	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	2.622	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.182	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.501	0.004
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.499

Таблица 2.Вывод метода итерациии QR для собственных значений.

Собственные значения расположены по диогонали таблицы. Отсюда видно, для матрицы десятого порядка метод команд математической системы Mathcad, двухступенческий классический метод по определению дают одинаковые результаты. Аналогично, методы Леверрье-Фадеева, Крылова и интерполяции дают одинаковые результаты. Ближе к этим методам даёт результат устойчивый для всех собственных значений многообещающий итерационный метод QR.

3. Заключение.

В работе исследованы классические методы вычисления всех собственных значений действительных матриц такие как двухступенчатый метод по определению, методы Леверрье, Фадеева, Крылова, интерполяции и Френцисса QR. Все методы дают хорошие результаты для матриц порядков $n \le 10$. Для проверки результатов работы составлены укрупненные алгоритмы в математической системе Mathcad , которые состоят из математических формул, внутренних команд математической системы. Mathcad является исполнтелем этих укрупнённых алгоритмов. Результат каждого укупнённого алгоритма совпадает с результатом вутренной команды вычисления сабственных значений математической системы.

ЛИТЕРАТУРА:

1.Демидович Б.П. Основы вычислительной математики. М.:Наука, 1966.-644.

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

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

4.Burden R.L. Numerical Analysis. Books Cole. Boston, USA.-2010.-895 pr.

5. Tohirjon o‘g‘li A. U., Nizomiddinovich I. S., Tojiddin o‘g‘li N. S. INTEGRAL TENGLAMALARNI MATHCAD DASTURIDA TAQRIBIY YECHISH USULLARI

//INNOVATION IN THE MODERN EDUCATION SYSTEM. – 2022. – Т. 2. – №. 18. – С. 880-883..

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

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

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

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