УДК 681.142.352.3
517.518

РАЗЛОЖЕНИЕ КРИВЫХ НА НЕЗАВИСИМЫЕ КОМПОНЕНТЫ

Я.П. Цирулис
(Рига)

Постановка задачи. Дано m функций $f_1(\lambda),\ldots,f_m(\lambda)$, $\lambda\in(a,b)$. Требуется найти такую систему функций $x_1(\lambda),\ldots,x_n(\lambda)$ и числа a_{ij}, $i=1,\ldots,m$; $j=1,\ldots,n$, чтобы для каждого i и всех $\lambda\in(a,b)$ имели бы место равенства

$$f_i(\lambda)=\sum_j a_{ij}x_j(\lambda)$$

или, в векторной форме,

$$f(\lambda)=A\,x(\lambda), \tag{I}$$

где $f(\lambda)$ – вектор-столбец с компонентами $f_1(\lambda),\ldots,f_m(\lambda)$;

$x(\lambda)$ – вектор-столбец с компонентами $x_1(\lambda),\ldots,x_n(\lambda)$;

A – матрица $\{a_{ij}\}$ порядка $m\times n$.

Функции $x_1,\ldots,x_n$ должны, кроме того, удовлетворять некоторым дополнительным требованиям (задача $\mathfrak{A}$),

ОПРЕДЕЛЕНИЕ I. Система функций $x=\{x_1,\ldots,x_n\}$ называется решением задачи $\mathfrak{A}$, а матрица A – решающей матрицей задачи $\mathfrak{A}$, если кроме (I) выполнены еще условия:

а) функции $x_1,\ldots,x_n$ линейно независимы в промежутке (a,b);

б) столбцы матрицы A линейно независимы.

Число n называется при этом порядком решения x, а функции $x_1,\ldots,x_n$ – его компонентами.

Условие б) здесь эквивалентно условию б′):

б′) любая система функций, удовлетворяющих (I), содержит

не менее, чем n функций.

Непосредственно из этого определения следуют два предложения:

<u>Предложение I</u>. Все решения задачи $\mathfrak{A}$ имеют одинаковый порядок, равный рангу системы $\{f_1, \ldots, f_m\}$, т.е. максимальному числу линейно независимых функций в этой системе.

<u>Предложение 2</u>. Если x – решение, то x' является решением тогда и только тогда, если существует неособенная матрица C, такая, что

$$x' = Cx. \qquad (2)$$

ОПРЕДЕЛЕНИЕ 2. Решения x и x' называются эквивалентными, если матрица C в (2) содержит в каждой строке ровно один ненулевой элемент, т.е. если компоненты этих решений можно пронумеровать так, чтобы для каждого j и всех $\lambda \in (a,b)$ выполнялись равенства $x'_j(\lambda) = c_j x_j(\lambda)$.

ОПРЕДЕЛЕНИЕ 3. Решение x называется простым, если оно удовлетворяет требованию: в любом подинтервале промежутка (a,b) линейно независимы все компоненты решения x, принимающие в этом подинтервале ненулевые значения.

ОПРЕДЕЛЕНИЕ 4. Пусть x некоторое решение. Обозначим через T_j множество всех тех и только тех $\lambda \in (a, b)$, в которых не обращается в нуль компонента x_j. Решение x называется нормальным, если любое из множеств $T_i \setminus T_j$, $i \neq j$ содержит бесконечно много точек.

Мы будем интересоваться простыми решениями и выясним, при каких условиях такие существуют, а также – когда все они эквивалентны. Можно доказать следующее:

<u>Предложение 3</u>. Если задача $\mathfrak{A}$ не имеет нормальных решений, то она может иметь простые решения, но среди них обязательно будут неэквивалентные. Если у задачи существует простое нормальное решение, то все простые её решения ему эквивалентны. Если же существует нормальное, но не простое решение, то простых решений у задачи $\mathfrak{A}$ нет вообще.

Существует алгоритм, позволяющий выяснить, существуют ли у задачи нормальные решения, и найти такие. Объем статьи позволяет привести только частный случай алгоритма, дающий положительный ответ в случае, если все множества T_j –суть интервалы (a_j, b_j). Условие нормальности решения тогда означает, что при надлежащей нумерации компонент имеет место

$$a_1 < a_2 < \ldots < a_n \quad \text{и} \quad b_1 < b_2 < \ldots < b_n . \qquad (3)$$

Такое решение для краткости назовем вполне нормальным.

Алгоритм напоминает метод Гаусса для решения системы линейных алгебраических уравнений. Сначала уравнение (I) постепенно преобразовывается так, чтобы матрица A приняла "треугольную" форму (прямой ход), а потом из полученного уравнения вычисляется вполне нормальное решение x (обратный ход).

Начало алгоритма. Введем обозначения $f^1 = f$, $f_i^1 = f_i, i=1,\dots m$.

Начало прямого хода. Допустим, что уже построен вектор f^k с компонентами $f_1^k, \dots, f_m^k$, $k \geq 1$.

Выполним следующие операции.

I. Пусть $U_i^k = \{\lambda : f_i^k(\lambda) \neq 0\}$. Положим $a_k = \min d_i$, $i = k, \dots, m$, где $d_i = \inf U_i^k$, если это множество не пустое, и $d_i = b$ в противном случае.

2. Если $a_k = b$, то перейти к 6.

3. Если $k = m$, то перейти к 7.

4. Если $a' > a_k$ -точка, обладающая свойством: все функции f_i^k; $i = k, \dots, m$ в промежутке (a_k, a') пропорциональны какой-нибудь одной из них (можно считать, что это функция f_k^k; в противном случае достаточно изменить нумерацию функций f_i^k):

$$\frac{f_i^k(\lambda)}{f_k^k(\lambda)} = p_i^k, \; i = k, \dots, m, \quad \text{для всех} \quad \lambda \in (a_k, a')$$

Если такой точки a' не существует, перейти к 8.

5. Строим новый вектор f^{k+1} следующим образом:

$$f_i^{k+1} = \begin{cases} f_i^k, & \text{если } i \leq k, \\ f_i^k - p_i^k f_k^k, & \text{если } i > k. \end{cases}$$

Теперь можно опять перейти к началу прямого хода.

6. Этим прямой ход алгоритма заканчивается. В данном случае функции $f_k, f_{k+1}, \dots, f_m$ оказались линейными комбинациями $f_1, \dots, f_{k-1}$, т.е. $n = k - 1 \leq m - 1$. Кроме того, нетрудно убедиться, что $a_1 < a_2 < \dots$. Нужно перейти к обратному ходу.

7. И в этом случае прямой ход заканчивается. Все функции $f_1, \dots, f_k$ линейно независимы, т.е. $n = k = m$, а

также $a_1 < a_2 < \ldots$ Нужно перейти к обратному ходу.

8. В этом случае прямой ход обрывается безрезультатно.

Вполне нормального решения задача не имеет, но она может иметь нормальные решения. Для нахождения их нужно пользоваться обобщением описываемого алгоритма. Этот вопрос мы здесь не будем рассматривать подробнее.

Обратный ход. После прямого хода имеем вектор f^n. Пусть B -матрица, связывающая f^n с искомым решением x : $f^n = Bx$. Можно показать, что $b_{ij} = 0$, если $i > j$. Строки с номером, большим n (если такие имеются), можно отбросить, так как они состоят из нулей, а соответствующие функции f_i^n тождественно равны нулю. Положим поэтому $x^n = f^n$, $x_i^n = f_i^n$, $i = 1, \ldots, n$. Далее обратный ход ведется аналогично прямому.

Начало обратного хода. Допустим, что уже построен вектор x^l с компонентами $x_1^l, \ldots, x_n^l$, $l \le n$.

1. Положим $b_l = \max \sup T_i^k$, $i = 1, \ldots, l$,

здесь $T_i^k = \{\lambda : x_i^l \lambda \neq 0\}$ (эти множества не пустые!).

2. Если $l = 1$, то перейти к 6.

3. Пусть $b' < b_l$ -точка, обладающая свойством: все функции x_i^l, $i \le l$, в промежутке (b', b_l) пропорциональны функции x_l^l (обязательно найдется такая точка $b' < b_l$, что в интервале (b', b_l) функция x_l^l принимает ненулевые значения):

$$\frac{x_i^l(\lambda)}{x_l^l(\lambda)} = q_i^l, \quad i = 1, \ldots, l, \text{ для всех } \lambda \in (b', b_l).$$

Если такой точки b' не существует, перейти к 7.

4. Строим новый вектор x^{l+1} следующим образом:

$$x_i^{l+1} = \begin{cases} x_i^l, & \text{если } i \ge l, \\ x_i^l - q_i^l x_l^l, & \text{если } i < l. \end{cases}$$

5. Вернуться к началу обратного хода.

6. Обратный ход закончен. x^1 является вполне нормальным решением задачи, а множествами T_j являются T_j^1. В нашем случае $T_j = (a_j, b_j)$, причем выполняются (3).

7. Обратный ход обрывается безрезультатно. Задача не имеет вполне нормального решения, но может иметь нормальное. Для выяснения этого нужен более сложный алгоритм.