

# Краевая задача

Цыбулин Иван

**К3.** Дана краевая задача для системы из  $n$  ОДУ

$$\begin{aligned}\frac{d\mathbf{u}}{dt} &= A(t)\mathbf{u} + \boldsymbol{\varphi}(t), \quad 0 \leq t \leq 1 \\ L_i^- \mathbf{u}(0) &= \alpha_i^-, \quad i = \overline{1, r} \\ L_i^+ \mathbf{u}(1) &= \alpha_i^+, \quad i = \overline{r+1, n}\end{aligned}$$

Рассмотреть вариант

$$\begin{aligned}L^- &= \begin{pmatrix} 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} & \alpha^- &= \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ L^+ &= \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} & \alpha^+ &= \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\end{aligned}$$

- а) Предложить алгоритм численного построения общего решения системы ОДУ.
- б) Предложить алгоритм численного построения линейного многообразия всех решений системы ОДУ, удовлетворяющих левым (или правым) условиям.
- в) Предложить эффективный алгоритм решения краевой задачи.

1). Требуется построить общее решение системы ОДУ. Оно представляет из себя сумму решений неоднородной задачи и произвольной линейной комбинации решений однородной задачи. Для этого численно решим 6 задач Коши следующего вида:

$$\begin{cases} \frac{d\mathbf{u}_0}{dt} = A\mathbf{u}_j + \boldsymbol{\varphi} \\ \mathbf{u}_0(0) = \mathbf{0} \end{cases}$$

$$\begin{cases} \frac{d\mathbf{u}_j}{dt} = A\mathbf{u}_j \\ \mathbf{u}_j(0) = \mathbf{e}_j \end{cases}$$

Здесь  $j = \overline{1, 5}$ ,  $\mathbf{e}_j$  -  $j$ -й столбец единичной матрицы. Так как начальные условия для однородных задач Коши линейно независимы, то и решения будут линейно независимы. Общее решение можно записать в виде

$$\mathbf{u} = \mathbf{u}_0 + \sum_{j=1}^5 C_j \mathbf{u}_j$$

2). С одной стороны, имея общее решение из п. 1 мы бы могли наложить краевые условия непосредственно на общее решение. При этом бы мы получили 2 уравнения на 5 неизвестных  $C_1, \dots, C_5$ , и смогли бы выразить две из них через 3 другие. Таким образом, необходимое многообразие бы задавалось тремя оставшимися неизвестными. Но это неэффективно с точки зрения вычислений, так как требует решения би задач Коши. Покажем, как можно обойтись меньшим числом. Если для построения общего решения мы брали линейно независимую систему из 5 различных начальных условий, то теперь возьмем систему из трех линейно независимых начальных условий, но при этом каждое из них должно удовлетворять левым краевым условиям. Построим многообразие начальных условий, которые удовлетворяют левым краевым условиям, т.е. просто разрешим систему

$$L^-\mathbf{u}(0) = \alpha^-$$

Базисный минор  $L^-$  находится, например, в столбцах 2 и 5. Эти компоненты будут зависимыми и будут выражаться через остальные. Приведем минор к виду единичной матрицы. Вычтем для этого из первой строки вторую

$$\begin{pmatrix} 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u^1(0) \\ u^2(0) \\ u^3(0) \\ u^4(0) \\ u^5(0) \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} u^1(0) \\ u^2(0) \\ u^3(0) \\ u^4(0) \\ u^5(0) \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 2 \end{pmatrix} + C_1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + C_2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + C_3 \begin{pmatrix} 0 \\ -2 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \mathbf{f}_0 + C_1 \mathbf{f}_1 + C_2 \mathbf{f}_2 + C_3 \mathbf{f}_3$$

Многообразие начальных условий, которое удовлетворяем левым краевым условиям найдено. Осталось найти многообразие решений краевой задачи только с левыми условиями. Оно находится с помощью решения четырех задач Коши.

$$\begin{cases} \frac{d\mathbf{u}_0}{dt} = A\mathbf{u}_j + \varphi \\ \mathbf{u}_0(0) = \mathbf{f}_0 \end{cases} \quad \begin{cases} \frac{d\mathbf{u}_j}{dt} = A\mathbf{u}_j \\ \mathbf{u}_j(0) = \mathbf{f}_j \end{cases}$$

Необходимое многообразие задается следующим выражением

$$\mathbf{u} = \mathbf{u}_0 + \sum_{j=1}^3 C_j \mathbf{u}_j$$

Заметим, что если производить аналогичные действия на правом конце отрезка, то необходимо решать только 3 задачи Коши.

3). Воспользуемся п. 2 и построим численно многообразие решений, удовлетворяющих правым краевым условиям. При этом необходимо решить 3 задачи Коши. Искомое многообразие будет представлено в виде

$$\mathbf{u} = \mathbf{u}_0 + \sum_{j=1}^2 C_j \mathbf{u}_j$$

На правом конце оно удовлетворяет тождеству

$$L^+ \mathbf{u}(1) \equiv \alpha^+$$

Остается только найти при каких  $C_j$  данное многообразие удовлетворяет также и левым краевым условиям. Подставив значения этого многообразия решений в левом конце отрезка в левые граничные условия, получаем линейную систему из двух уравнений на 2 неизвестные  $C_1$  и  $C_2$ .

$$L^- \mathbf{u}(0) = \alpha^-$$

$$L^{-}\mathbf{u}_0(0) + C_1 L^{-}\mathbf{u}_1(0) + C_2 L^{-}\mathbf{u}_2(0) = \alpha^{-}$$

Разрешив систему, получим элемент многообразия, который удовлетворяет левым и правым условиям, дифференциальному уравнению, т.е. является решением краевой задачи. Этот алгоритм в два раза экономичнее метода численного построения общего решения.

**К4.** *Дана нелинейная дифференциальная краевая задача второго порядка*

$$\begin{aligned} y''(x) - e^{xy'(x)} - x^2(y'(x))^3 - x &= 0, \quad x \in [0, 1] \\ y(0) = 2 \quad y'(1) + 3y(1) &= 2 \end{aligned}$$

- а) Выбрать начальное приближение  $\tilde{y}(x)$ , удовлетворяющее краевым условиям.*
- б) Линеаризовать задачу в окрестности этого приближения.*
- в) Выписать разностную задачу, аппроксимирующую линейную краевую задачу из пункта б) со вторым порядком.*
- г) Предложить алгоритм решения разностной задачи из пункта в), основанный на методе прогонки.*

а) В качестве начального приближения возьмем линейную функцию  $\tilde{y}(x) = \alpha x + \beta$ . Подстановкой в краевые условия находим

$$\beta = 2, \quad \alpha = -1.$$

Начальное приближение

$$\tilde{y}(x) = 2 - x.$$

б) Будем искать решение в виде поправки к начальному приближению

$$y(x) = \tilde{y}(x) + v(x).$$

Подставим  $y(x)$  в краевую задачу

$$\tilde{y}''(x) + v''(x) - e^{x\tilde{y}'(x) + xv'(x)} - x^2(\tilde{y}'(x) + v'(x))^3 - x = 0$$

и пренебрежем всеми членами порядка  $O(v(x)^2, v'(x)^2)$

$$\tilde{y}''(x) + v''(x) - e^{x\tilde{y}'(x)}(1 + xv'(x)) - x^2(\tilde{y}'(x))^3 - 3x^2(\tilde{y}'(x))^2v'(x) - x = 0.$$

Соберем слева все слагаемые с  $v, v', v''$ :

$$v''(x) + [e^{x\tilde{y}'(x)}x - 3x^2(\tilde{y}'(x))^2]v'(x) = \tilde{y}''(x) + x + x^2(\tilde{y}'(x))^3 - e^{xy'(x)}$$

Мы получили линейное дифференциальное уравнение для  $v$ . Для  $v$  также имеются краевые условия

$$\tilde{y}(0) + v(0) = 2, \quad \tilde{y}'(1) + v'(1) + 3\tilde{y}(1) + 3v(1) = 2.$$

Поскольку  $\tilde{y}(x)$  удовлетворяла краевым условиям,  $v(x)$  должна удовлетворять таким же, но однородным краевым условиям

$$v(0) = 0, \quad v'(1) + 3v(1) = 0.$$

Для конкретного начального приближения  $\tilde{y}(x) = 2 - x$  задача для  $v(x)$  имеет вид

$$\begin{aligned} v''(x) + [xe^{-x} - 3x^2]v'(x) &= x - x^2 - e^{-x} \\ v(0) = 0, \quad v'(1) + 3v(1) &= 0. \end{aligned}$$

в) Аппроксимируем задачу для  $v(x)$  со вторым порядком. Чтобы аппроксимировать уравнение достаточно заменить производные разностными аналогами второго порядка

$$v''(x) \sim \frac{v_{m+1} - 2v_m + v_{m-1}}{h^2}, \quad v'(x) \sim \frac{v_{m+1} - v_{m-1}}{2h},$$

где  $h = \frac{1-0}{M}$  — шаг сетки. Полученная аппроксимация уравнения имеет вид (здесь  $x_m = mh$ ):

$$\frac{v_{m+1} - 2v_m + v_{m-1}}{h^2} + [x_me^{-x_m} - 3x_m^2]\frac{v_{m+1} - v_{m-1}}{2h} = x_m - x_m^2 - e^{-x_m}$$

В трехдиагональной форме уравнение можно записать как

$$\begin{aligned} a_mv_{m-1} + b_mv_m + c_mv_{m+1} &= d_m \\ a_m &= 1 - \frac{h}{2}(x_me^{-x_m} - 3x_m^2) \\ b_m &= -2 \\ c_m &= 1 + \frac{h}{2}(x_me^{-x_m} - 3x_m^2) \\ d_m &= h^2(x_m - x_m^2 - e^{-x_m}) \end{aligned}$$

Левое краевое условие для  $v(x)$  аппроксимируется тривиально и точно

$$v_0 = 0.$$

Правое краевое условие необходимо аппроксимировать со вторым порядком но только с использованием двух крайних точек. Аппроксимируем его таким образом:

$$\frac{v_M - v_{M-1}}{h} + 3v_M + h\varphi(v_M, v_{M-1}) = 0,$$

где  $\varphi$  — пока не известная добавка, обеспечивающая второй порядок аппроксимации. Исследуем правок краевое условие на аппроксимацию

$$\frac{[v]_M - [v]_{M-1}}{h} + 3[v]_M + h\varphi([v]_M, [v]_{M-1}) = \delta_M,$$

Разложим  $v(x)$  в окрестности  $x = 1$  ( $m = M$ )

$$[v]_{M-1} = [v]_M - h[v']_M + \frac{h^2}{2}[v'']_M + O(h^3)$$

Получаем

$$[v' + 3v]_M - \frac{h}{2}[v'']_M + h\varphi([v]_M, [v]_{M-1}) = \delta_M.$$

Первое слагаемое содержит в точности правое краевое условие для  $v$ , следовательно,

$$\delta_M = h\varphi([v]_M, [v]_{M-1}) - \frac{h}{2}[v'']_M.$$

Чтобы  $\delta_M = O(h^2)$  необходимо, чтобы  $\varphi([v]_M, [v]_{M-1})$  совпадала с  $\frac{1}{2}[v'']_M$  хотя бы с точностью  $O(\tau)$ . Аппроксимировать  $v''$  по значениям в двух точках  $v_M, v_{M-1}$  невозможно, поэтому понизим порядок производной пользуясь тем, что  $v''$  удовлетворяет также дифференциальному уравнению

$$v''(x) + (xe^{-x} - 3x^2)v'(x) = x - x^2 - e^{-x}$$

В точке  $x_M = 1$  оно принимает вид

$$v''(1) + (e^{-1} - 3)v'(1) = -e^{-1}$$

$$v''(1) = (3 - e^{-1})v'(1) - e^{-1}$$

$$[v'']_M = (3 - e^{-1})[v'(1)]_M - e^{-1} = (3 - e^{-1})\frac{[v]_M - [v]_{M-1}}{h} - e^{-1} + O(h)$$

Таким образом, в качестве добавки  $\varphi$  можно взять

$$\varphi(v_M, v_{M-1}) = \frac{3 - e^{-1}}{2} \frac{v_M - v_{M-1}}{h} - \frac{e^{-1}}{2}$$

и граничное условие принимает вид

$$\frac{v_M - v_{M-1}}{h} + 3v_M + h \left( \frac{3 - e^{-1}}{2} \frac{v_M - v_{M-1}}{h} - \frac{e^{-1}}{2} \right) = 0.$$

Разностная задача целиком

$$\begin{cases} \frac{v_{m+1} - 2v_m + v_{m-1}}{h^2} + [x_m e^{-x_m} - 3x_m^2] \frac{v_{m+1} - v_{m-1}}{2h} = x_m - x_m^2 - e^{-x_m}, & m = 1, \dots, M-1 \\ v_0 = 0 \\ \frac{v_M - v_{M-1}}{h} + 3v_M + h \left( \frac{3 - e^{-1}}{2} \frac{v_M - v_{M-1}}{h} - \frac{e^{-1}}{2} \right) = 0 \end{cases}$$

г) Разностная задача является трехдиагональной системой линейных уравнений относительно  $v_m$

$$\begin{cases} b_0 v_0 + c_0 v_1 = d_0 \\ a_m v_{m-1} + b_m v_m + c_m v_{m+1} = d_m \\ a_M v_{M-1} + b_M v_M = d_M \end{cases}$$

Метод прогонки для нее можно реализовать следующим образом: сначала система приводится к двухдиагональной форме (считаем, что  $c_M = 0$ )

$$\begin{aligned} \tilde{b}_0 &= 1, \quad \tilde{c}_0 = \frac{c_0}{b_0}, \quad \tilde{d}_0 = \frac{d_0}{b_0} \\ \tilde{b}_m &= 1, \quad \tilde{c}_m = \frac{c_m}{b_m - a_m \tilde{c}_{m-1}}, \quad \tilde{d}_m = \frac{d_m - a_m \tilde{d}_{m-1}}{b_m - a_m \tilde{c}_{m-1}}, \quad m = 1, 2, \dots, M \end{aligned}$$

В результате этого преобразования система принимает вид

$$\begin{cases} v_0 + \tilde{c}_0 v_1 = \tilde{d}_0 \\ v_m + \tilde{c}_m v_{m+1} = \tilde{d}_{m+1} \\ v_M = \tilde{d}_M, \end{cases}$$

из которого решение находится обратной подстановкой

$$\begin{aligned} v_M &= \tilde{d}_M \\ v_m &= \tilde{d}_m - \tilde{c}_m v_{m+1}, \quad m = M-1, \dots, 0 \end{aligned}$$

PS. Такая форма прогонки редко встречается в литературе, однако ее проще всего реализовать в программе. Она устроена так, что величины с тильдой можно записывать поверх величин без тильды, так как те более не нужны для вычислений. Величины  $\tilde{b}$  можно не менять, так как они в формулах не используются.