ГЛАВА 9

Задача Коши для системы ОДУ

К обыкновенным дифференциальным уравнениям (ОДУ) и их системам сводится изучение многообразных задач в различных предметных областях (в физике, химии, биологии, медицине, технике и т.д.). Ряд содержательных задач подобного рода был рассмотрен нами в главах 2 и 3.

Конкретная задача может приводить к дифференциальному уравнению любого порядка, или к системе уравнений различных порядков. Но уравнение n-го порядка

$$u^{(n)}(t) = f(t, u', u'', \ldots, u^{(n-1)})$$

при введении новых неизвестных $u_i(t) = u^{(i)}(t)$ можно свести к эквивалентной системе n уравнений первого порядка

$$\begin{aligned} u_i'(t) &= u_{i+1}, \quad i = 0, 1, \ldots, n-2, \\ u_{n-1}'(t) &= f(t, u_0, u_1, \ldots, u_{n-1}), \end{aligned}$$

где $u_0(t) = u(t)$. Аналогично, произвольную систему дифференциальных уравнений любого порядка можно заменить эквивалентной системой уравнений первого порядка

$$u_i'(t) = f_i(t, u_1, u_2, \ldots, u_n), \quad i = 1, 2, \ldots, n, \tag{9.1}$$

записывая их для краткости в векторной форме

$$u'(t) = f(t, u(t)), \tag{9.2}$$

где u и f вектор-функции

$$u(t) = (u_1, u_2, \ldots, u_n)(t), \quad f(t, u(t)) = (f_1, f_2, \ldots, f_n)(t, u(t)).$$

Общее решение системы (9.1) зависит от n параметров $c = (c_1, c_2, \ldots, c_n)$ — постоянных интегрирования. Для выделения единственного решения необходимо наложить n дополнительных условий.

Различают три типа задач для обыкновенных дифференциальных уравнений: задачи Коши, краевые задачи и задачи на собственные значения. В случае задачи Коши (задачи с начальными условиями) дополнительные условия имеют вид

$$u_i(a) = u_{ai}, \quad i = 1, 2, \ldots, n, \tag{9.3}$$

т. е. заданы значения всех функций u_i в одной и той же точке $t = a$. Решение задачи (9.1), (9.3), как правило, требуется найти на некотором отрезке $a \leqslant t \leqslant b$ (или $b \leqslant t \leqslant a$), так что точку $t = a$ можно считать начальной точкой этого отрезка. Условия (9.3) в векторной записи имеют вид $u(a) = u_a$.

1. Методы решения. Методы решения задачи Коши разделяются на точные, приближенно-аналитические и численные.

К *точным* относятся методы, позволяющие выразить решение задачи через элементарные или специальные функции, либо представить его при помощи интегралов от них. Эти методы изучаются в курсе ОДУ; по этому поводу можно порекомендовать также справочник [11]. Однако классы уравнений, для которых разработаны методы получения точных решений, сравнительно узки и охватывают только малую часть возникающих на практике задач. Например, доказано, что решение первого из уравнений

$$u'(t) = t^2 + u^2(t), \quad u'(t) = \frac{u(t) - t}{u(t) + t}$$

не выражается через элементарные функции, а общее решение второго имеет вид

$$0.5 \ln(t^2 + u^2) + \operatorname{arctg}(u/t) = c.$$

Однако для того, чтобы воспользоваться последней формулой и вычислить $u(t)$ при конкретном t, надо численно решить это трансцендентное уравнение, что нисколько не проще, чем численно решить само дифференциальное уравнение.

К *приближенно-аналитическим* относятся методы, в которых решение получается как предел $u(t)$ некоторой последовательности функций $u_k(t)$, причем $u_k(t)$ выражается через элементарные функции или интегралы от них. Ограничиваясь конечным значением k,

получаем приближенное выражение для $u(t)$. Примерами могут служить метод разложения решения в обобщенный степенной ряд, метод Пикара, метод малого параметра. Однако эти методы удобны лишь в том случае, когда большую часть промежуточных выкладок удается провести точно. Это выполнимо для сравнительно простых задач, что сильно сужает область применения подобных методов.

Численные методы — это методы вычисления приближенных значений искомого решения $u(t)$ на некотором заданном или генерируемом в ходе решения задачи множестве точек $t_1, t_2, \ldots, t_N$, называемом сеткой узлов. Решение при этом получается в виде таблицы. Численные методы не позволяют найти общее решение системы (9.1); они могут дать только какое-то частное решение, например, решение задачи Коши (9.1), (9.3), что является основным недостатком этих методов. Он компенсируется тем, что численные методы применимы к различным уравнениям и всем типам задач для них. С появлением ЭВМ численные методы стали одним из основных способов решения конкретных практических задач для ОДУ.

Далее мы кратко рассмотрим одно семейство численных методов решения задачи Коши — явные методы Рунге–Кутты. Для простоты изложения мы ограничиваемся случаем одного уравнения первого порядка. Векторная запись задачи Коши позволяет без изменения вида формул обобщить методы на системы уравнений.

§ 1. Методы Рунге–Кутты

Пусть требуется найти приближенное решение задачи

$$u'(t) = f(t, u(t)), \quad t \in (a, b], \quad u(a) = u_a, \tag{9.4}$$

на равномерной с шагом h сетке узлов

$$\omega_h = \{t_i = a + ih,\ i = 0, 1, \ldots, N\}, \quad h = (b - a)/N.$$

Через $u_i = u(t_i)$, y_i будем обозначать соответственно точное и приближенное решение задачи в точке сетки с номером i.

1. Явный метод Эйлера. Этот метод является простейшим представителем семейства методов Рунге–Кутты. Строится метод Эйлера, например, следующими рассуждениями.

Используя формулу Тейлора с остаточным членом в форме Лагранжа, для каждого $i = 0, 1, \ldots, N-1$ получим разложение

$$u(t_i + h) = u(t_i) + hu'(t_i) + \frac{h^2}{2}\, u''(t_i + \xi_i h), \quad \xi_i \in [0, 1]. \tag{9.5}$$

Поскольку в силу уравнения (9.4) справедливо равенство $u'(t_i) = f(t_i, u(t_i))$, то соотношение (9.5) перепишется в виде

$$u_{i+1} = u_i + hf(t_i, u_i) + h\psi_{i+1}, \quad \psi_{i+1} = \frac{h}{2}\, u''(t_i + \xi_i h). \tag{9.6}$$

В этой формуле слагаемое $h\psi_{i+1} = O(h^2)$ является малой величиной, если h достаточно мало. Отбрасывая его, придем к методу Эйлера:

$$y_{i+1} = y_i + h\, f(t_i, y_i), \quad i = 0, 1, \ldots, N-1, \quad y_0 = u_a. \tag{9.7}$$

Эти соотношения позволяют вычислить приближенное решение в точке сетки t_{i+1}, зная приближенное решение в предыдущей точке t_i. Такие численные методы называются *одношаговыми*.

1.1. Оценка точности метода Эйлера. Оценим точность метода Эйлера, считая, что функция f удовлетворяет условию Липшица

$$|f(t, \xi) - f(t, \eta)| \leqslant \lambda\, |\xi - \eta| \quad \forall\, \xi, \eta \in R,\ t \in [a, b].$$

Пусть $e_i = u_i - y_i$. Величина $|e_i|$ представляет собой абсолютную погрешность решения в i-той точке сетки. Обозначим через $\|u - y\|$ максимальную погрешность,

$$\|u - y\| = \max_{i=1,\ldots,N} |u_i - y_i|.$$

Величина ψ, определяемая согласно (9.6) формулой

$$\psi_{i+1} = \frac{u_{i+1} - u_i}{h} - f(t_i, u_i), \quad i = 0, 1, \ldots, N-1, \tag{9.8}$$

называется погрешностью аппроксимации метода Эйлера и определяет малость $\|u - y\|$. Чтобы убедиться в этом, получим уравнение для погрешности, вычитая из равенств (9.6) равенства (9.7):

$$e_{i+1} = e_i + h\,\big(f(t_i, u_i) - f(t_i, y_i)\big) + h\psi_{i+1}. \tag{9.9}$$

Воспользуемся условием Липшица и хорошо известной оценкой $1 + x \leqslant e^x,\ x > 0$. Тогда, из (9.9) следует, что

$$|e_{i+1}| \leqslant (1 + \lambda h)\,|e_i| + h\,|\psi_{i+1}| \leqslant e^{\lambda h}\,|e_i| + h\,|\psi_{i+1}|.$$

Эта оценка справедлива для любого $i = 0, 1, \ldots, N-1$. Применяя ее реккурентно для оценки $|e_i|$ в правой части и продолжая этот процесс, будем иметь ($e_0 = 0$):

$$\begin{aligned} |e_{i+1}| &\leqslant e^{\lambda h}\big(e^{\lambda h}\,|e_{i-1}| + h\,|\psi_i|\big) + h\,|\psi_{i+1}| \leqslant \ldots \leqslant \\ &\leqslant e^{(i+1)\lambda h}\,|e_0| + \sum_{k=1}^{i+1} h\,e^{(i+1-k)\lambda h}|\psi_k| \leqslant \\ &\leqslant e^{ih\lambda}\,(i+1)\,h \max_{i=1,\ldots,N} |\psi_i| \leqslant (b-a)\,e^{(b-a)\lambda}\,\|\psi\|. \end{aligned}$$

Таким образом, для любого $i = 1, 2, \ldots, N$, справедлива оценка $|e_i| \leqslant C_0\|\psi\|$ с постоянной $C_0 = (b-a)e^{(b-a)\lambda}$, не зависящей от h. Следовательно,

$$\|u - y\| \leqslant C_0\,\|\psi\|.$$

Если ввести обозначение $C(u) = 0.5 \max_{t\in[a,b]} |u''(t)|$, то из (9.6) вытекает, что $\|\psi\| \leqslant C(u)h$ и, таким образом,

$$\|u - y\| \leqslant C_0\,C(u)h. \tag{9.10}$$

В оценке (9.10) величины сомножителей при h, как правило, неизвестны. В силу этого оценка имеет теоретическое значение, но малопригодна для практической оценки точности метода Эйлера. Тем не менее, из нее следует, что с уменьшением h пропорционально уменьшается максимальная погрешность решения, а также, что погрешность может быть сделана сколь угодно малой при соответствующем выборе h (или числа точек сетки N). Из оценки (9.10) следует также, что погрешность решения больше для задач, у которых больше отрезок интегрирования и больше константа Липшица λ; может потребоваться очень малый шаг сетки для обеспечения точности, если эти числа достаточно велики. Практика вычислений подтверждает эти выводы.

Сделаем еще одно важное замечание. Время, необходимое для вычисления решения, равно примерно $T_f N$, где T_f время, необходимое для вычисления правой части f уравнения (9.4). В то же

время, согласно (9.10), справедлива оценка $\|u - y\| \leqslant c/N$, где $c = C_0 C(u)(b-a)$. Отсюда можно сделать вывод, что если мы хотим, например, в 10 раз увеличить точность полученного решения, то время вычисления решения также увеличится в 10 раз.

1.2. Об устойчивости метода Эйлера. Чтобы пояснить понятие устойчивости метода Эйлера, рассмотрим следующую модельную задачу Коши

$$u'(t) = -\lambda u, \quad t \in (0, T], \quad u(0) = u_0,$$

в случае, когда $u_0 > 0$, $\lambda > 0$. Эта задача соответствует выбору $f(t, u) = -\lambda\, u$ в (9.4), а постоянная λ определяет соответствующую f постоянную Липшица.

Точное решение этой задачи есть $u(t) = u_0\, e^{-\lambda t}$. Оно положительно и монотонно убывает с ростом t. Формула метода Эйлера принимают вид $y_{i+1} = y_i - \lambda h\, y_i$ или $y_{i+1} = q\, y_i$, $i = 0, 1, \ldots, N-1$, где $q = 1 - \lambda\, h$. Следовательно,

$$y_i = q\, y_{i-1} = q^2\, y_{i-2} = \ldots = q^i\, u_0.$$

Отсюда следует, что, если $h > 2/\lambda$, то $|q| > 1$ и $|y_i|$ возрастает с ростом i, что кардинально расходится с качественным поведением точного решения. К тому же, в этом случае, при $q < 0$ приближенное решение меняет знак в каждой точке сетки. В этом случае говорят, что приближенное решение ведет себя неустойчиво (метод неустойчив). Условие устойчивости метода Эйлера имеет вид $|q| < 1$ или

$$h < h_0, \quad h_0 = 2/\lambda. \tag{9.11}$$

При выполнении несколько более сильного условия $0 < q < 1$ (при $h < 1/\lambda$) приближенное решение является положительным и монотонно убывающим, т. е. качественно верно воспроизводится как монотонность, так и положительность точного решения. Ясно, что условие (9.11) практически не является ограничительным, если λ не слишком велико: для обеспечения точности решения должно выбраться достаточно малое h. Условие (9.11) является ограничительным, а метод практически непригодным, если λ очень большое. В этом случае необходимо выбирать очень малое h для получения хотя бы качественно правильного решения (например, при $\lambda \sim 10^6$ должно быть $h \lesssim 10^{-6}$).

Отмеченное выше понятие устойчивости метода Эйлера для модельной задачи переносится на задачу Коши общего вида. Сформулированное выше условие устойчивости (9.11), при соответствующем определении λ, является характерным для всего класса методов Рунге–Кутты и при их использовании необходимо выбирать шаг h достаточно малым как с целью обеспечения устойчивости метода, так и с целью достижения требуемой точности приближенного решения.

2. Определение семейства методов Рунге–Кутты. Формулы, требующие q вычислений правой части f на одном шаге интегрирования (q-стадийные явные методы Рунге–Кутты), имеют следующий вид:

$$y_{i+1} = y_i + h\big(b_1k_1 + b_2k_2 + \ldots + b_qk_q\big), \tag{9.12}$$

где

$$\begin{aligned}
&k_1 = f(t_i, y_i),\\
&k_2 = f(t_i + c_2h, y_i + a_{21}hk_1),\\
&\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots\\
&k_q = f(t_i + c_qh, y_i + a_{q1}hk_1 + a_{q2}hk_2 + \ldots + a_{q,q-1}hk_{q-1}).
\end{aligned} \tag{9.13}$$

При фиксированном q конкретный метод определяется выбором коэффициентов b_j, c_j, $a_{\ell j}$. Всегда выполняется равенство

$$b_1 + b_2 + \ldots + b_q = 1, \tag{9.14}$$

так что при $q = 1$ приходим к методу Эйлера. Коэффициенты q-стадийного явного метода Рунге–Кутты принято располагать в виде следующей треугольной таблицы:

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_q & a_{q1} & a_{q2} & \ldots & a_{q,q-1} & \\
\hline
 & b_1 & b_2 & \ldots & b_{q-1} & b_q
\end{array}$$

Величины k_j, определенные выше, зависят, в частности, от h и y_i; укажем эту зависимость в виде $k_j = k_j(h, y_i)$. Так, если $u_i = u(t_i)$

есть значение точного решения в узле сетки t_i, то $k_1(h, u_i) = f(t_i, u_i)$, $k_2(h, u_i) = f(t_i + c_2 h, u_i + a_{21} h k_1(t_i, u_i))$, и т.д. Величину

$$\psi_{i+1} = \psi_{i+1}(h) = \frac{u(t_i + h) - u_i}{h} - \sum_{j=1}^{q} b_j \, k_j(h, u_i) \tag{9.15}$$

называют *погрешностью аппроксимации* в точке сетки t_{i+1}, и рассматривают ее как функцию шага сетки h.

Разложим $\psi_{i+1}(h)$ в ряд Тейлора:

$$\psi_{i+1}(h) = \psi_{i+1}(0) + \psi'_{i+1}(0)h + \ldots + \psi^{(m)}_{i+1}(0)\frac{h^m}{m!} + o(h^m),$$

и подберем коэффициенты b_j, c_j, $a_{\ell j}$ так, чтобы при возможно большем m выполнялись равенства

$$\psi_{i+1}(0) = \psi'_{i+1}(0) = \ldots = \psi^{(m-1)}_{i+1}(0) = 0. \tag{9.16}$$

Тогда

$$\psi_{i+1} = \psi^{(m)}_{i+1}(0)\frac{h^m}{m!} + o(h^m), \tag{9.17}$$

и говорят, что *порядок погрешности аппроксимации* метода равен m ($\psi_{i+1} = O(h^m)$), а первое слагаемое в правой части (9.17) называют главным членом погрешности аппроксимации.

Нетрудно видеть, что условие первого порядка аппроксимации (первое равенство $\psi_{i+1}(0) = 0$ в (9.16)) совпадает с условием (9.14), поскольку, в силу уравнения, $k_j(0, u_i) = f(t_i, u_i) = u'(t_i)$,

$$\psi_{i+1}(0) = u'(t_i) - \sum_{j=1}^{q} b_j \, k_j(0, u_i) = \Big(1 - \sum_{j=1}^{q} b_j\Big) u'(t_i).$$

Для произвольного метода Рунге–Кутты (9.12) аналогично методу Эйлера доказывается, что (см., напр., [12, с. 221])

$$\|u - y\| \leqslant C_0 \, \|\psi\|,$$

с постоянной C_0, зависящей on $b - a$, λ, q, а также от коэффициентов метода, но не зависящей от h. Из этой оценки следует, что если порядок погрешности аппроксимации равен m, т. е. $\|\psi\| \leqslant C(u)\, h^m$, то

$$\|u - y\| \leqslant C_0 \, C(u) h^m.$$

В этом случае говорят, что *порядок точности метода* равен m; видно, что порядок точности совпадает с порядком погрешности аппроксимации, чем и обусловлен способ выбора коэффициентов метода.

Известно, что всегда $m \leqslant q$ и не существует m-стадийных методов с порядком точности m при $m \geqslant 5$. В следующей таблице указано минимальное число стадий ($q_{\min}$), которое необходимо для обеспечения соответствующего порядка точности метода.

m	1	2	3	4	5	6	7	8
$q_{\min}$	1	2	3	4	6	7	9	11

Время работы T_q программы, реализующей методы с q-стадиями, определяется величиной qT_fN, где T_f время, необходимое для вычисления правой части f уравнения (9.4) при заданных аргументах. Максимальная погрешность решения оценивается сверху величиной C/N^m. Увеличивая число узлов, скажем, в 10 раз, мы получаем в 10^m раз более точное решение. Отсюда ясно, в чем выгода от использования методов высокого порядка точности. С другой стороны, если в вычислениях не нужна очень большая точность, то методы со средним показателем m (с $m = 4$), могут оказаться предпочтительными, поскольку они требуют меньшего времени работы программы и имеют достаточную точность.

Рассмотрим, как получаются конкретные методы при небольших значениях q. При больших значениях q рассмотрения аналогичны; возрастает лишь сложность проводимых при этом вычислений.

3. Методы Рунге–Кутты при $q = 2$. Формулы имеют вид

$$y_{i+1} = y_i + h\big(b_1k_1 + b_2k_2\big),$$

где

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + c_2h, y_i + a_{21}hk_1). \end{aligned}$$

Попытаемся удовлетворить соотношениям (9.16) при возможно большем значении m. В данном случае

$$\psi_{i+1}(h) = \frac{u(t_i + h) - u_i}{h} - \big(b_1k_1(h, u_i) + b_2k_2(h, u_i)\big). \qquad (9.18)$$

Как мы видели выше, равенство $\psi_{i+1}(0) = 0$ выполняется, если

$$b_1 + b_2 = 1.$$

Разложим $u(t_i + h)$ в ряд Тейлора в точке t_i. Тогда

$$\frac{u(t_i + h) - u_i}{h} = u_i' + \frac{h}{2!}\, u_i'' + \frac{h^2}{3!}\, u_i''' + \dots \tag{9.19}$$

Согласно уравнению $u'(t) = f(t, u(t))$; поэтому

$$u''(t) = f_t'(t, u(t)) + f_u'(t, u(t))u'(t) = f_t'(t, u(t)) + f_u'(t, u(t))f(t, u(t)).$$

С учетом этих формул нетрудно видеть, что равенство

$$0 = \psi_{i+1}'(0) = \left\{(1 - 2c_2b_2)f_t' + (1 - 2a_{21}b_2)f f_u'\right\}\Big|_{t=t_i, u=u_i}$$

дает еще два уравнения для определения коэффициентов

$$1 - 2c_2b_2 = 0, \quad 1 - 2a_{21}b_2 = 0.$$

Вычисления показывают, что выбором коэффициентов условию $\psi_{i+1}''(0) = 0$ удовлетворить не удается. Это легко увидеть на примере уравнения с $f(t, u) = u$. Тогда в выражении (9.18) второе слагаемое линейно зависит от h и его вторая производная по h равна нулю. Из формулы (9.19) легко видеть, что вторая производная по h от первого слагаемого в (9.18) при $h = 0$ равна $1/3u'''(t_i)$. Поэтому

$$\psi_{i+1}''(0) = 1/3\, u'''(t_i).$$

Следовательно, $m = 2$ и четыре коэффициента метода необходимо определить из трех уравнений

$$\begin{aligned} b_1 + b_2 &= 1, \\ 2c_2b_2 &= 1, \\ 2a_{21}b_2 &= 1. \end{aligned}$$

Принимая, например, c_2 за свободный параметр, найдем:

$$a_{21} = c_2, \quad b_2 = \frac{1}{2c_2}\,, \quad b_1 = 1 - \frac{1}{2c_2}\,. \tag{9.20}$$

Таким образом, мы получили однопараметрическое семейство формул Рунге–Кутты второго порядка точности. В частности, полагая $c_2 = 1;\ 2/3;\ 1/2$ в (9.20), придем к трем конкретным формулам.

4. Методы Рунге–Кутты при $q = 3$. Формулы имеют вид

$$y_{i+1} = y_i + h\big(b_1 k_1 + b_2 k_2 + b_3 k_3\big),$$

где

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + c_2 h, y_i + a_{21} h k_1), \\ k_3 &= f(t_i + c_3 h, y_i + a_{31} h k_1 + a_{32} h k_2). \end{aligned}$$

Аналогично тому, как это было сделано выше, получается следующая система из шести уравнений для определения восьми коэффициентов метода при $m = 3$ (см., напр., [12, с. 224]):

$$\begin{aligned} c_2 &= a_{21}, \\ c_3 &= a_{31} + a_{32}, \\ b_1 + b_2 + b_3 &= 1, \\ 2(c_2 b_2 + c_3 b_3) &= 1, \\ 3(\alpha_2^2 b_2 + \alpha_3^2 b_3) &= 1, \\ 6 c_2 a_{32} b_3 &= 1. \end{aligned}$$

Эта система имеет два семейства решений: двухпараметрическое со свободными параметрами c_2 и c_3, причем, $c_2 \neq c_3$ и $c_2 \neq 2/3$, и однопараметрическое со свободным параметром a_{32} (при $c_2 = c_3 = 2/3$). В качестве примера укажем коэффициенты при $c_2 = 1/2$, $c_3 = 1$: $b_1 = b_3 = 1/6$, $b_2 = 4/6$, $a_{21} = 1/2$, $a_{31} = -1$, $a_{32} = 2$.

5. Формулировка задачи Коши с событиями. Выше мы рассмотрели задачу Коши

$$u'(t) = f(t, u(t)), \quad t \in (a, b], \quad u(a) = u_a,$$

предполагая, что интервал интегрирования (a, b) задан и конечен. В таком виде формулируется большинство задач, встречающихся в приложениях. Однако имеется немало задач, в формулировке которых точка b неизвестна, а, в свою очередь, определяется решением задачи. Такие задачи ставятся следующим образом:

требуется найти функцию $u(t)$ и число t_f такие, что $u(t)$ удовлетворяет условию $u(a) = u_a$ и уравнению

$$u'(t) = f(t, u(t)), \quad t \in (a, t_f], \tag{9.21}$$

а $t_f > a$ такое минимальное число, при котором

$$\Phi(t, u(t))\big|_{t=t_f} = 0. \tag{9.22}$$

Здесь Φ — заданная функция двух переменных, которую будем называть событием. Часто эта функция «снабжается направленностью» и интегрирование прекращается, когда выполнено условие (9.22) и при этом функция $\phi(t) = \Phi(t, u(t))$ была возрастающей (или убывающей). *При формулировке задач с событием всегда предполагается, что $\phi(t)$ является непрерывной и меняет знак при $t = t_f$.*

В общем случае Φ может быть вектор функцией

$$\Phi(t, u(t)) = (\Phi_1(t, u(t)), \ldots, \Phi_m(t, u(t))), \quad m \geqslant 1.$$

В этом случае интегрирование прекращается, если выполнено, например, хотя бы одно из «направленных» условий $\Phi_i(t, u(t))\big|_{t=t_f} = 0$ (наступит хотя бы одно событие).

В качестве примера «задачи с событием» рассмотрим задачу баллистики — задачу о полете снаряда. Система уравнений в этом случае имеет вид

$$\begin{aligned} x' &= v\cos(\theta), \\ y' &= v\sin(\theta), \\ v' &= -C\rho S v^2/(2m) - g\sin(\theta), \\ \theta' &= -g\cos(\theta)/v. \end{aligned}$$

Она дополняются начальными условиями

$$x(0) = 0, \quad y(0) = 0, \quad \theta(0) = \theta_0, \quad v(0) = v_0.$$

Здесь $x = x(t)$, $y = y(t)$ — координаты центра масс снаряда в момент времени t в системе координат xOy, в которой лежит траектория снаряда, а $y = 0$ есть уровень земли; $v = v(t)$ есть скорость снаряда в момент времени t; $\theta = \theta(t)$ — угол между касательной к траектории снаряда и осью Ox. В соответствии с начальными условиями, снаряд вылетает из точки с координатами $(0, 0)$ под углом θ_0 к горизонту с начальной скоростью v_0. В данной задаче интерес представляют время t_f и дальность L полета снаряда. Время полета снаряда можно определить как тот момент времени, когда снаряд упадет на землю (или

падая достигнет некоторой высоты h_0 — подрыв снаряда на заданной высоте). Ясно, что интервал интегрирования $(0, t_f]$ в данном случае неизвестен, имеется одно «направленное» событие, которое определяется следующим образом

$$\Phi(t, u(t)) = y(t), \quad \text{функция } \ t \to \Phi(t, u(t)) \ \text{ убывает}.$$

§ 2. Программирование методов Рунге–Кутты

1. Задачи без событий. Рассмотрим вопросы, связанные с программной реализацией q-стадийного метода Рунге–Кутты, предназначенного для решения произвольной задачи Коши для системы ОДУ, имея в виду рассмотренные выше методы с постоянным шагом интегрирования h. Будем предполагать, что число уравнений n и число узлов N таковы, что в памяти ЭВМ можно разместить матрицу размера $(N + 1) \times n$.

Выберем идентификатор RKq для обозначения имени этой программы (функции). Какие параметры необходимо передать этой программе, а какие получить в качестве результата ее выполнения? Поскольку мы ведем речь о задаче вида

$$u'(t) = f(t, u(t)), \quad t \in (a, b], \quad u(a) = y_0,$$

где $f : R \times R^n \to R^n$ — заданная функция, $y_0 \in R^n$ — заданный вектор, а a, b — заданные числа, то задача определяется только исходными данными

$$f, \quad a, \ b, \ y_0,$$

или тремя данными, если a, b поместить, например, в массив $tspan$, длины два. Ясно, что эти данные надо передать RKq. Кроме того метод Рунге–Кутты определяется параметром N ($h = (b-a)/N$); его тоже необходимо передать RKq. Будем считать (это общепринято), что f в свою очередь реализуется в виде программы (функции); она должна иметь два входных параметра (t, y), $y \in R^n$, $t \in R$, и один выходной —значение вектор-функции f —массив длины n.[1]

[1] В зависимости от выбранного языка программирования, возможно как RKq, так и f, необходимо передавать также параметр n — размерность задачи. Далее мы будем предполагать, что имеется функция $numel()$, которая для одномерного массива возвращает число его элементов.

Метод Рунге–Кутты генерирует векторы y_i длины n, которые являются приближениями к $u(t_i) = (u_1(t_i), \ldots, u_n(t_i))$, $i = 0, 1, \ldots, N$. Будем считать, что эти векторы сохраняются в матрице Y, размера $(N+1) \times n$, i-тая строка которого равна y_i.[2] Будем считать также, что массив $T = (t_0, t_1, \ldots, t_N)$ используется для хранения сетки.

Схематически заголовок программы RKq можно определить в виде

$$function \quad [T, Y] = RKq(f, tspan, y0, N).$$

Здесь в квадратных скобках указываются выходные параметры, а в круглых —входные.[3] Вместо параметра N в функцию можно передавать шаг сетки h.

Для заданной матрицы Y под $Y(i,:)$ будем понимать i-тую строку Y, а для заданных одномерных массивов x, y, длины n, и чисел c, d, под $x = y$, $z = c * x + d * y$, $y = 0$ будем понимать реализацию покомпонентных операций

$$x_i = y_i, \quad z_i = c * x_i + d * y_i, \quad y_i = 0, \quad i = 1, 2, \ldots, n.$$

Будем считать также, что команда $Y = zeros(m, n)$ определяет матрицу размера $m \times n$, состоящую из нулей; под $x(i)$ будем понимать i-тый элемент массива (вектора). Ниже приведена в схематической форме реализация трехстадийного метода (для примера); числа типа $alpha2$, $beta21$ и т.д. должны быть заменены на конкретные значения.

```
function [T,Y]=RK3(f,tspan,y0,N)
T=zeros(N+1,1);
Y=zeros(N+1,n);

t=tspan(1);
y=y0(:);      % y column vector
h=(tspan(2)-tspan(1))/N;

T(1)=t;
Y(1,:)=y';
for i=1:N
  k1=f(t,y);

  yp=y+(beta21*h)*k1;
  k2=f(t+alpha2*h,yp);
```

[2] Тогда j-тый столбец Y содержит приближение к j-той компоненте решения $u_j(t)$ во всех точках сетки.

[3] Такой синтаксис языка программирования принят в MatLab.

```
  yp=y+(beta31*h)*k1+(beta32*h)*k2;
  k3=f(t+alpha3*h,yp);

  y=y+h*(p1*k1+p2*k2+p3*k3);

  t=t+h;
  T(i+1)=t;
  Y(i+1,:)=y';
end
```

Для каждой задачи Коши должна быть написана своя программа вычисления правой части; шаблон этой программы с именем fct выглядит следующим образом:

```
function dy=fct(t,y)
n=numel(y);
dy=zeros(n,1);

dy(1)=...;
dy(2)=...;
.....
dy(n)=...;
```

Здесь вместо многоточия в $dy(i) = ...$; нужно записать выражение функции $f_i(t, y)$ в терминах переменных $t, y(1), \ldots, y(n)$.

2. Задачи с событиями. В этом случае задача определяется как данными f, a, b, y_0, которые мы обсудили выше, так и функцией Φ, определяющей события. Для упрощения изложения будем считать, что Φ есть скалярная функция. Предположим также, что она определяется функцией $events$, имеющей описание

```
function [val,dir]=events(t,y)
```

Эта функция возвращает величину val $= \Phi(t, y)$ и направленность Φ в переменной dir: dir $= 1$, если событие наступает при возрастании функции Φ; dir $= -1$, если событие наступает при убывании Φ; наконец, dir $= 0$, если Φ не снабжено направленностью (событие фиксируется при val $= 0$ как при убывании, так и возрастании Φ). Например, в задаче о полете снаряда, описанной выше, эта функция определяется следующим образом, если y есть вектор $(x(t), y(t), v(t), \theta(t))$:

```
function [val,dir]=events(t,y)
val=y(2);
dir=-1;
```

При написании аналога функции $RK3$ информация об этой функции должна быть передана в качестве входного параметра. При этом потребуется внести также следующие очевидные изменения в $RK3$.

1) Вместо параметра N в функцию необходимо передавать параметр h, поскольку значение N до решения задачи не известно.

2) Заранее неизвестно, сколько памяти необходимо выделить для хранения выходного параметра Y. При программировании в MatLab эта трудность преодолевается, если убрать в $RK3$ следующие строки: $T = zeros(1, N+1)$; и $Y = zeros(N+1, n)$;, а строки $T(i+1) = t$; $Y(i+1, :) = y'$; заменить на $T = [T; t]$; $Y = [Y; y']$;

3) Основной цикл с перечислением $for\ \ i = 1 : N$ необходимо заменить на цикл типа $while$, условие выхода из которого зависит от наступления события.

4) Функция $events$, определяющая событие, должна быть вычислена как до основного цикла, так и внутри него после определения y_{i+1}. В этом случае можно считать, что внутри цикла известны значения $\Phi_i = \Phi(t_i, y_i)$ и $\Phi_{i+1} = \Phi(t_{i+1}, y_{i+1})$.

Пусть на i-том шаге известны $\Phi_i = \Phi(t_i, y_i)$, $\Phi_{i+1} = \Phi(t_{i+1}, y_{i+1})$ и $\Phi_i \Phi_{i+1} < 0$. В этом случае при некотором $t_{fh} \in [t_i, t_{i+1}]$ функция Φ обращается в нуль. Обсудим вопрос о том, как экономично определить точку t_{fh} — приближение к t_f.

Можно принять $t_{fh} = (t_i + t_{i+1})/2$. В этом случае время наступления события будет определено с точностью $O(h)$. На порядок по h точнее время t_f можно определить, используя линейную (лагранжеву) интерполяцию функции Φ (методом секущих). Точнее, пусть $L(\Phi, t)$ есть линейная на $[t_i, t_{i+1}]$ функция, определяемая условиями интерполяции $L(\Phi, t_i) = \Phi_i$, $L(\Phi, t_{i+1}) = \Phi_{i+1}$:

$$L(\Phi, t) = \Phi_i + (\Phi_{i+1} - \Phi_i)(t - t_i)/h.$$

Определим t_{fh} так, что $L(\Phi, t_{fh}) = 0$. Получим

$$t_{fh} = t_i - h\,\Phi_i/(\Phi_{i+1} - \Phi_i).$$

Приближенное решение y_f в этой точке определим как $y_f = L(y, t_{fh})$, т. е. положим

$$y_f = y_i + (y_{i+1} - y_i)(t_{fh} - t_i)/h.$$

Можно использовать также квадратичную или кубическую эрмитову интерполяцию Φ и y (в точке t_i известны значения y_i, $y'_i = f(t_i, y_i)$; аналогично в точке t_{i+1} известны значения y_{i+1}, $y'_{i+1} = f(t_{i+1}, y_{i+1})$). Отметим, что применение того или иного способа определения t_f должно быть согласовано, по-возможности, с точностью реализуемого метода Рунге–Кутты. Приближенное решение y_f в точке t_{fh} определяется как значение соответствующего интерполяционного полинома в точке t_{fh}.

Для облегчения программирования при выполнении заданий студент может ограничится лишь линейной интерполяцией.

§ 3. Решение задачи Коши для системы ОДУ в MatLab

Для решения задачи Коши для системы ОДУ в MatLab имеются две функции ode23 и ode45, основанные на явных методах Рунге–Кутты. Эти функции хорошо документированы в Help системе MatLab и их использование для решения как задач без событий, так и с событиями, не вызывает трудностей.

В ode45 реализован один из так называемых вложенных методов Рунге–Кутты. Эти методы позволяет экономично и автоматически выбирать шаг интегрирования с целью обеспечения требуемой точности приближенного решения задачи.

1. О вложенных методах Рунге–Кутты. Рассмотрим q-стадийный явный метод Рунге–Кутты, определяемый следующей таблицей:

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_q	a_{q1}	a_{q2}	$\dots$	$a_{q,q-1}$	
	b_1	b_2	$\dots$	b_{q-1}	b_q
	$\hat{b}_1$	$\hat{b}_2$	$\dots$	$\hat{b}_{q-1}$	$\hat{b}_q$

Эта таблица имеет два набора коэффициентов $b : \{b_i\}$ и $\{\hat{b}_i\}$. Предположим, что приближенное решение y_i в точке t_i найдено с требуемой

точностью и решение в точке $t_{i+1} = t_i + h$ находится по формуле

$$y_{i+1} = y_i + h\big(b_1k_1 + b_2k_2 + \ldots + b_qk_q\big), \tag{9.23}$$

имеющей порядок погрешности аппроксимации равный m. Здесь k_i вычисляются согласно формулам (9.13). Предположим также, что коэффициенты $\{\hat{b}_i\}$ таковы, что формула

$$\hat{y}_{i+1} = \hat{y}_i + h\big(\hat{b}_1k_1 + \hat{b}_2k_2 + \ldots + \hat{b}_qk_q\big)$$

с теми же коэффициентами k_i, что и в (9.23), имеет порядок погрешности аппроксимации равный $m + 1$. При этих предположениях за q вычислений правой части одновременно получаются два приближенных решения, точность которых отличается на порядок по h.

Пусть $\|\cdot\|_1$, $\|\cdot\|_2$ есть некоторые нормы в R^n. В силу малости h, величину $err_{i+1} = \|y_{i+1} - \hat{y}_{i+1}\|_1/\|\hat{y}_{i+1}\|_2$ можно принять за меру локальной относительной погрешности решения y_{i+1} в точке t_{i+1}; второе значение $\hat{y}_{i+1}$ используется при этом лишь для определения локальной погрешности.

Вложенные методы обычно называют по фамилии автора с указанием порядка основного и вспомогательного метода; например, метод Мерсона 4(5). На практике широко используются различные вложенные методы, среди которых выделяются методы Дормана–Принса, известные при различных m. В функции ode45 реализованы формулы Дормана–Принса 4(5), которые определяются следующей таблицей

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
y	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
$\hat{y}$	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Этот метод имеет семь стадий, однако практически — шесть (как отмечалось ранее, метод пятого порядка не может иметь меньше,

чем шесть стадий). Действительно, поскольку $a_{7,i} = b_i$ для всех i, то нетрудно видеть, что коэффициент k_7 на предыдущем шаге совпадает с коэффициентом k_1 на следующем шаге. Этот факт используется при программировании метода.

2. Об автоматическом выборе шага. Процедура интегрирования с автоматическим выбором шага определяется следующим образом. Пусть решение в точке t_i найдено с некоторым шагом h. Выполняется следующий цикл.

1) С использованием вложенных формул вычисляется y_{i+1} и величина погрешности $err = \|y_{i+1} - \hat{y}_{i+1}\|_1 / \|\hat{y}_{i+1}\|_2$.

2) Если $err \leqslant tol$, т. е. погрешность меньше или равна заданной допустимой погрешности, то решение в точке t_{i+1} считается найденным, а шаг интегрирования — выполненным успешно. Решение y_{i+1} каким-либо образом сохраняется (например, в массиве Y). Далее вычисляется новый шаг интегрирования $h = \min(\gamma\, h, hmax)$, где $\gamma \geqslant 1$, и h принимается в качестве текущего для следующего шага и интегрирование продолжается. Здесь $hmax$ —максимальный шаг (задается пользователем);

3) Если $err > tol$, т. е. погрешность оказывается больше желаемой, то полученное решение в точке t_{i+1} отбрасывается как неточное, а шаг интегрирования считается неудачным. Вычисляется новый шаг интегрирования, равный $\gamma\, h$, где теперь $\gamma < 1$, и вычисления повторяются с шага 1).

Для выбора γ, как правило, используют формулу

$$\gamma = \min(facmax, \max(facmin, fac\,(tol/err)^{1/(m+1)}).$$

Здесь $facmax$ ($facmin$) — максимальный коэффициент увеличения (уменьшения) шага ($facmax = 1.5, \ldots, 5$). Эти коэффициенты страхуют от резкого увеличения или уменьшения шага. Для шагов, выполняемых непосредственно после отбрасывания решения, рекомендуется выбирать $facmax = 1$; fac — страховочный коэффициент ($facmax = 0.8, \ldots, 0.9$).

Приведем шаблон основного цикла описанной выше процедуры интегрирования.

```
while (t<tfinal) && (h>=hmin) % The main loop
  if (t+h)>tfinal
    h = tfinal - t;
  end;

  % Compute two solution
  ...........

  y1=y+h*(...);
  y2=y+h*(...);

  err = norm(y1-y2)/norm(y2); % Truncation error

  % Update the solution only if the error is acceptable
  if err <= tol
    t = t + h;
    y = y1;
    T = [T; t];
    Y = [Y; y'];
  end
  ...........

  % Step control
  if err == 0.0, err = 10*eps;  end
  gamma = min(facmax, max(facmin, fac*(tol/err)^(1/(m+1))));
  h     = min(gamma*h,hmax);
  ...........
end
```

Более подробно о вложенных методах Рунге–Кутты и проблеме автоматического выбора шага можно прочитать, например, в книге [13].

§ 2. Система типа хищник-жертва. Модель Мак-Артура

1. Определение модели [7, 22]. Рассмотрим динамику популяции двух видов, взаимодействующих между собой по типу хищник-жертва, при наличии внутривидовой конкуренции жертв за ограниченные ресурсы и при учете фактора насыщения хищника. Обозначим через $x = x(t)$ и $y = y(t)$ плотности популяций жертв и хищников в момент времени t. Уравнения имеют следующий вид:

$$\begin{aligned} x' &= a(1 - x/K)x - \frac{bxy}{1 + Ax}, \\ y' &= -cy + \frac{d}{1 + Ax}\, xy, \end{aligned} \tag{11.4}$$

где a, b, c, d, A, K — неотрицательные числа. Структура уравнений следующая.

- Слагаемое $a(1-x/K)x$ определяет скорость размножения жертв в отсутствии хищников. При малых x скорость определяется величиной a и, таким образом, a характеризует норму рождаемости при малой плотности популяции. При большей плотности (до величины K) популяция растет, при $x > K$ — уменьшается в размерах (скорость отрицательна). Таким образом, это слагаемое описывает ограниченность ресурсов: окружающая среда может обеспечивать существование только популяции плотности меньшей K.

- Слагаемое $bxy/(1 + Ax)$ описывает влияние хищников на популяцию жертв. Функция $bx/(1 + Ax)$ характеризует количество жертв, убиваемых одним хищником в единицу времени (реакция хищника на плотность популяции жертвы). Здесь учтено, что при большой плотности жертв хищник убивает лишь b/A жертв в единицу времени; т. е., перестает убивать, когда насыщается.

- Второе уравнение определяет изменение популяции хищников. Постоянная c определяется естественной нормой смертности хищников. Второе слагаемое характеризует прирост хищников в зависимости от плотности жертв (функция $d\,x/(1 + Ax)$). При большой плотности жертв скорость прироста хищников опреде-

ляется величиной d/A и, таким образом, d/A характеризует норму рождаемости при благоприятных для хищников условиях.

2. Безразмерная форма уравнений. Вводя безразмерные величины

$$X = \left(\frac{d}{c}\right) x, \quad Y = \left(\frac{b}{a}\right) y\,, \quad \tau = at, \quad \alpha = \frac{Ac}{d}\,, \quad \varepsilon = \frac{c}{Kd}, \quad \gamma = \frac{c}{a}\,,$$

преобразуем уравнения (11.4) к виду

$$\begin{aligned} X' &= (1 - \varepsilon X)\, X - \frac{XY}{1 + \alpha X}\,, \\ Y' &= \gamma \left(\frac{X}{1 + \alpha X} - 1\right) Y\,, \end{aligned} \tag{11.5}$$

и дополним их начальными условиями

$$X(0) = X_0, \quad Y(0) = Y_0. \tag{11.6}$$

Задание

1. Напишите программу интегрирования задачи Коши для системы из n уравнений первого порядка вида

$$y' = f(t, y), \quad y(0) = y_0, \quad y(t) \in R^n,$$

на произвольном отрезке $[a, b]$, используя метод Рунге – Кутты 3-го порядка точности с постоянным шагом h:

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + h/3, y_n + h/3k_1), \\ k_3 &= f(t_n + 2/3h, y_n + 2/3hk_2), \\ y_{n+1} &= y_n + h(k_1 + 3k_3)/4. \end{aligned}$$

2. Протестируйте программу на примере системы уравнений

$$\begin{aligned} y_1' &= -\sin(t)/(1 + e^{2t})^{1/2} + y_1(y_1^2 + y_2^2 - 1), \\ y_2' &= \cos(t)/(1 + e^{2t})^{1/2} + y_2(y_1^2 + y_2^2 - 1), \end{aligned}$$

на отрезке $[0, 5]$ с точным решением (проверьте!)

$$y_1 = \cos(t)/(1 + e^{2t})^{1/2}, \quad y_2 = \sin(t)/(1 + e^{2t})^{1/2}.$$

3. Найдите стационарные решения (состояния равновесия) системы (11.5) и определите их тип. Как они зависят от параметров задачи?
4. Решите систему уравнений (11.5), (11.6) при помощи разработанной программы. Для значений параметра α из интервала $[0.1, 0.9]$ рассчитайте динамику популяции при следующих исходных данных

$$\varepsilon = 0.1, \quad \gamma = 1, \quad X_0 = 3, \quad Y_0 = 1.$$

Определите те значения параметра α при которых в системе появляются и исчезают устойчивые автоколебания (предельный цикл). Постройте графики наиболее характерных решений в координатах (X, Y), $(X(t), t)$ и $(Y(t), t)$ и дайте их интерпретацию. Соответствует ли вычисленная картина фазовой плоскости качественной теоретической, полученной в п. 3.

Замечание 3. Как вариант, можно пункт 1 задания заменить на функцию MatLab ode45.