ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра дифференциальных уравнений
и теории управления

Ю.Н. Горелов

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ (МЕТОД РУНГЕ – КУТТА)

*Утверждено Редакционно-издательским советом университета
в качестве учебного пособия*

Самара
Издательство «Самарский университет»
2006

УДК 518.0 (075)
ББК 32.96
Г 687

Рецензент д-р физ.-мат. наук, заведующий кафедрой высшей математики Поволжской государственной академии телекоммуникаций и информатики И.А. Блатов

Отв. редактор д-р физ.-мат. наук, проф., заведующий кафедрой дифференциальных уравнений и теории управления В.А. Соболев

Горелов Ю.Н.

Г 687 Численные методы решения обыкновенных дифференциальных уравнений (метод Рунге – Кутта): учеб. пособие / Ю.Н. Горелов; Федер. агентство по образованию. – Самара: Изд-во «Самарский университет», 2006. – 48 с.

В учебном пособии изложены теоретические основы и практические подходы к программной реализации численных методов решения задачи Коши или начальной задачи для обыкновенных дифференциальных уравнений первого порядка и их систем. Рассматривается процедура поэтапной разработки программной реализации численного решения с применением формул Рунге – Кутта четвертой степени (четвертого порядка точности на одном шаге интегрирования), получивших наиболее широкое применение в вычислительной математике для решения широкого круга прикладных задач в различных областях естествознания, техники и экономики.

Предназначено для студентов, изучающих курсы «Теория управления», «Теория оптимального управления» и спецкурс «Устойчивость и управление движением» по специальностям «Математика», «Прикладная математика», «Механика», выполняющих лабораторные, курсовые и дипломные работы в рамках указанных курсов, а также для аспирантов, обучающихся по специальности 05.13.18 – Математическое моделирование, численные методы и комплексы программ.

УДК 518.0 (075)
ББК 32.96

СОДЕРЖАНИЕ

Введение

Обыкновенные дифференциальные уравнения и их системы являются математическими моделями для значительного числа прикладных задач в различных областях естествознания (механика, физика и др.), техники и экономики. Как правило, эти задачи практически исключают получение аналитических решений. В первую очередь это относится к нелинейным дифференциальным уравнениям либо к системам линейных дифференциальных уравнений высокой размерности с переменными коэффициентами. В таких случаях единственная возможность их исследования или решения обычно связана с применением численных методов.

Задачи, математические модели которых содержат соответствующие дифференциальные уравнения, во многих случаях сводятся к численному решению задачи Коши или некоторого набора таких задач. Простейшим примером задачи Коши является начальная задача для дифференциального уравнения

$$\frac{dy}{dx} = f(x, y), \tag{0.1}$$

для которого требуется найти частное решение $y(x)$ на интервале $[x_0, x_f]$, удовлетворяющее заданному начальному условию:

$$y(x_0) = y_0, \tag{0.2}$$

где x_0 – начальная точка, y_0 – начальное значение и предполагается, что функция $f(x, y)$ – правая часть уравнения (0.1) – является непрерывной функцией, удовлетворяющей условиям Липшица [1,2]. Это оговаривается сейчас, а далее предполагается, что функция $f(x, y)$ такова, что решение задачи (0.1), (0.2) существует и единственно на любом заданном интервале $[x_0, x_f]$. Здесь $x_0 < x_f \le \infty$, хотя в общем случае задачу Коши можно рассматривать и на интервалах $[x_f, x_0]$, когда $-\infty \le x_f < x_0$.

Поскольку отыскание точного решения задачи Коши (0.1), (0.2) в виде $y(x)$ или, по крайней мере, значения $y(x_f)$, как правило, не представляется возможным, то эту задачу приходится решать приближенными методами вычислительной математики, которые в зависимости от формы представления получаемого решения можно разделить на две группы [1]:

1) аналитические методы, доставляющие приближенное решение уравнения (0.1) в виде некоторого аналитического выражения;
2) численные методы, доставляющие приближенное решение в виде некоторой таблицы значений искомой функции $y(x)$ в заданных точках x_i, $i = 0, 1, 2, \ldots$, совокупность которых называется сеткой на $[x_0, x_f]$, а каждая точка x_i, соответственно, – ее узлами.

В пределах настоящего введения кратко остановимся на первой группе методов, отметив лишь метод представления решения дифференциального уравнения (0.1) в виде отрезка ряда, а также метод последовательных приближений Пикара [1,2].

В теории дифференциальных уравнений рассматривается задача о представлении решения уравнения (0.1) в виде ряда

$$y(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \ldots + a_n(x - x_0)^n + \ldots , \qquad (0.3)$$

где a_i, $i = 0, 1, 2, \ldots$, – некоторые коэффициенты. Если искомое решение можно представить в виде (0.3) и, кроме того, если найти достаточно большое число коэффициентов a_i, $i = 0, 1, \ldots, n$, такое, чтобы абсолютная величина суммы остальных членов оказалась меньше наперед заданной допустимой погрешности, то соответствующий отрезок ряда (0.3) будет приближенным аналитическим представлением искомого решения.

Предположим, что искомое частное решение уравнения (0.1) может быть разложено в ряд Тейлора по степеням разности $x - x_0$:

$$y(x) = y_0 + (x - x_0)\, y_0^{(1)} + \frac{(x - x_0)^2}{2!}\, y_0^{(2)} + \ldots$$

$$+ \frac{(x - x_0)^n}{n!}\, y_0^{(n)} + o(|x - x_0|^{n+1}), \qquad (0.4)$$

где $y_0^{(k)} = \left.\dfrac{d^k y(x)}{dx^k}\right|_{x=x_0}$ – производная k-го порядка, вычисляемая в точке x_0, а $o(|x - x_0|^{n+1})$ – члены разложения степени не ниже $n+1$. Очевидно, что непосредственно из (0.2) следует $a_0 = y_0$, а из (0.1), с учетом (0.2), – $a_1 = y_0^{(1)} = f(x_0, y_0)$. Значения производных $y_0^{(k)}$ для $k \geq 2$ в (0.4), а стало быть, и соответствующие им коэффициенты a_k в (0.3) последовательно вычисляются дифференцированием уравнения (0.1) с учетом ранее полученных значений $y_0^{(k)}$. Например,

$$y_0^{(2)} = \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_0\\ y=y_0}} + f(x_0, y_0)\left.\frac{\partial f}{\partial y}\right|_{\substack{x=x_0\\ y=y_0}} \text{ и т.д. (см. также [1,2]).}$$

Такая процедура определения всех коэффициентов a_i, $i = 0, 1, \ldots, n$, требует существования непрерывных частных производных правой части $f(x, y)$ уравнения (0.1) до порядка $n-1$. Известно, что если правая часть $f(x, y)$ в окрестности точки (x_0, y_0) является аналитической функцией своих аргументов, то при значениях x, достаточно близких к значению x_0, существует единственное решение задачи Коши (0.1), (0.2), которое можно разложить в ряд Тейлора. Тогда любая частичная сумма этого ряда будет приближенным решением рассматриваемой задачи.

Аналогичным образом этот метод применяется и для обыкновенных дифференциальных уравнений порядка выше первого, а также для систем обыкновенных дифференциальных уравнений первого порядка.

Еще один аналитический метод построения решения задачи Коши (0.1), (0.2) – метод последовательных приближений. Он состоит в том, что искомое решение $y(x)$ уравнения (0.1) получают как предел последовательности функций $\tilde{y}_m(x)$, $m = 0, 1, 2, \ldots$, которые находятся по рекуррентной формуле

$$\tilde{y}_m(x) = y_0 + \int_{x_0}^{x} f(x, \tilde{y}_{m-1}(x))\, dx. \tag{0.5}$$

Известно, что в том случае, когда правая часть $f(x, y)$ уравнения (0.1) удовлетворяет условию Липшица по y, то независимо от выбора начальной функции $\tilde{y}_0(x)$ последовательные приближения, вычисляемые по формуле (0.5), сходятся на некотором интервале $[x_0, x_0 + h]$, то есть для некоторого заданного $0 < h < \infty$, к решению задачи Коши (0.1), (0.2). При этом в качестве начального приближения допустимо выбирать любую функцию $\tilde{y}_0(x)$, достаточно близкую к точному решению, в том числе и в виде какой-либо частичной суммы степенного ряда (0.4).

Если окажется, что $x_0 + h < x_f$, то далее указанная процедура может рассматриваться на подынтервалах $[x_0 + h, x_0 + 2h]$, $[x_0 + 2h, x_0 + 3h]$ и т.д., пока для некоторого $q > 1$ не будет выполнено $x_f \le x_0 + qh$. Соответственно, решение исходной задачи Коши (0.1), (0.2) в этом случае сводится к последовательному решению аналогичных задач на таких подынтервалах $[x_0 + ih, x_0 + (i+1)h]$, $i = 0, 1, 2, \ldots$, с начальными условиями: для $i = 0$ с учетом (0.2) – $y_{00} = y_0$, а для $i = 1, 2, 3, \ldots$, $y_{i0} = \tilde{y}_{i-1}(x_0 + ih)$, где $\tilde{y}_{i-1}(x)$ – приближенное решение задачи Коши на предыдущем подынтервале, то есть на $[x_0 + (i-1)h, x_0 + ih]$.

Отметим, что для разложения решения дифференциального уравнения (0.1) в степенной ряд требуется аналитичность его правой части, но при использовании метода последовательных приближений это необязательно. Поэтому область применения метода последовательных приближений по сравнению первым методом, вообще говоря, более широкая, хотя существенным недостатком здесь является необходимость вычисления на каждой итерации по формуле (0.5) все более громоздких интегралов.

Рассмотрение численных методов решения задачи Коши (0.1), (0.2) обычно начинают с изложения метода Эйлера, имеющего в основном лишь теоретическое значение. Итак, выбрав достаточно малый шаг h, построим на некотором интервале $[x_0, x_f]$ равномерную сетку в виде системы равноотстоящих узлов: $x_i = x_0 + ih$, $i = 0, 1, 2, \dots$. Приближенные значения $y_i \approx y(x_i)$ в методе Эйлера находят последовательно по формулам

$$y_{i+1} = y_i + hf(x_i, y_i),\ i = 0, 1, 2, \dots \ . \tag{0.6}$$

При этом искомая интегральная кривая $y = y(x)$ заменяется ломанной $M_0 M_1 M_2 \dots$ с вершинами $M_i(x_i, y_i)$, координаты которых вычисляются последовательно по формулам (0.6).

На практике метод Эйлера не применяется, поскольку его точность, как правило, оказывается низкой. В связи с этим более удобным считается двойной просчет, когда расчеты по формулам (0.6) повторяются, но для сетки с шагом $h/2$. Кроме того, в этом случае появляется возможность контролировать точность получаемого решения задачи (0.1), (0.2).

Соответствующие модификации метода Эйлера – улучшенные методы Эйлера [4] – будут еще описаны далее.

Основная особенность метода Эйлера – как простейшего способа численного решения задачи Коши – связана с тем, что он относится к классу так называемых одношаговых методов численного решения обыкновенных дифференциальных уравнений и их систем. В них на каждом шаге для вычисления приближенного значения $y_{i+1} \approx y(x_{i+1})$ на $[x_i, x_{i+1}]$, например по формулам (0.6), используется только информация, доступная в точке $x = x_i$, то есть значение $y_i \approx y(x_i)$, полученное ранее на предыдущем шаге (или для $i = 0$, согласно (0.2), заданное $y_0 = y(x_0)$). Если значения приближенного решения $y_{i+1} \approx y(x_{i+1})$ находят, используя информацию о нем в нескольких узлах: x_i, x_{i-1}, … , $x_{i-r+1}(r > 1)$, то такие методы решения задачи Коши относят к классу многошаговых методов [1,2].

Многие важные вопросы, связанные с численными методами решения дифференциальных уравнений и их систем (вычислительная устойчивость, глобальная точность, решение жестких дифференциальных систем и т.п.) рассматриваются, например, в монографиях [4-9]. В заключение отметим, что настоящее пособие в основном следует изложению 2-го тома книги И.С. Березина и Н.П. Жидкова Методы вычислений [1], а приведенные в нем материалы в основном материалы заимствованы из [1-4].

1. Численное решение обыкновенных дифференциальных уравнений первого порядка методом Рунге-Кутта

1.1. Идея метода Рунге – Кутта

Рассмотрим задачу Коши на некотором заданном интервале $[x_0, x_f]$, а именно: найти для дифференциального уравнения

$$\frac{dy}{dx} = f(x, y) \tag{1.1}$$

решение, которое удовлетворяет начальному условию:

$$y(x_0) = y_0, \tag{1.2}$$

где x_0, y_0 – некоторые числа, а для функции $f(x, y)$ – правой части этого уравнения – предполагается существование непрерывных частных производных до некоторого порядка $n-1$ в соответствующей области, содержащей точку (x_0, y_0). В этом случае решение уравнения (1.1), по крайней мере, в окрестности точки $x = x_0$ будет иметь непрерывные до n-го порядка производные и, стало быть, для него можно записать следующее разложение в ряд Тейлора:

$$y(x) = y_0 + (x - x_0)\, y_0^{(1)} + \frac{(x - x_0)^2}{2!}\, y_0^{(2)} + \ldots$$

$$\ldots + \frac{(x - x_0)^n}{n!}\, y_0^{(n)} + o(|x - x_0|^{n+1}), \tag{1.3}$$

где $y_0^{(k)} = \frac{d^k y(x)}{dx^k}|_{x=x_0}$, $k = 1, 2, 3, \ldots$, – производные от $y(x)$, вычисленные в точке $x = x_0$; $o(\cdot)$ – члены разложения степени не ниже $(n+1)$-й относительно $x - x_0$ или остаточный член ряда Тейлора $R_n(x - x_0)$. Если для всех x $(|x - x_0| < h)$ и некоторого h функция $y(x)$ имеет $(n+1)$-ю производную $y^{(n+1)}(x)$, то остаточный член $R_n(x - x_0)$ для всех указанных x имеет вид

$$R_n(x - x_0) = y^{(n+1)}(\xi)\frac{(x - x_0)^{n+1}}{(n+1)!},$$

где $x_0 < \xi < x$, если $x > x_0$, или $x < \xi < x_0$, если $x < x_0$. Если при этом $\left| y^{(n+1)}(x) \right| \le M_{n+1} < \infty$, то $\left| R_n(x - x_0) \right| < M_{n+1}\frac{h^{n+1}}{(n+1)!}$ или, что то же самое, погрешность при отбрасывании $R_n(x - x_0)$ в (1.3) имеет порядок h^{n+1}.

Пусть решение задачи Коши (1.1), (1.2) вначале отыскивается только в точке $x = x_0 + h$, где $x_0 + h \le x_f$, а $h > 0$ – некоторое достаточно малое число. Тогда, обозначая $h = x - x_0$ и отбрасывая в разложении (1.3) члены $o(|x - x_0|^{n+1})$ или $R_n(x - x_0)$, разложение (1.3) можно переписать в виде

$$y(x_0 + h) \cong y_0 + h y_0^{(1)} + \frac{h^2}{2!} y_0^{(2)} + \ldots + \frac{h^n}{n!} y_0^{(n)}.$$

Обозначив $\Delta y_0(h) \cong y(x_0 + h) - y_0$, отсюда получим следующее выражение для приращения искомого решения на интервале $[x, x_0 + h]$:

$$\Delta y_0 = h y_0^{(1)} + \frac{h^2}{2!} y_0^{(2)} + \ldots + \frac{h^n}{n!} y_0^{(n)}. \tag{1.4}$$

Производные, входящие в выражение (1.4), можно непосредственно вычислить, как было уже отмечено во введении, по формулам последовательного дифференцирования уравнения (1.1). Однако получаемые при этом формулы – даже в операторной форме [1] – оказываются чрезмерно громоздкими, что снижает в конечном счете их практическую ценность. В связи с этим К. Рунге предложил (впоследствии В. Кутта развил идею метода) вместо вычислений по формуле (1.4) для Δy_0 искать линейные комбинации следующего вида:

$$\Delta y_0(h) = p_{r1} k_1(h) + p_{r2} k_2(h) + \ldots + p_{rr} k_r(h), \tag{1.5}$$

где p_{ri} $(i = 1, 2, \ldots, r)$ – некоторые постоянные коэффициенты; а $k_i(h)$ – функции, вычисляемые по формулам:

$$k_i(h) = h f(\xi_i, \eta_i), \; i = 1, 2, \ldots, r; \tag{1.6}$$

здесь $\xi_i = x_0 + \alpha_i h$ и $\eta_i = y_0 + \beta_{i1} k_1(h) + \beta_{i2} k_2(h) + \ldots + \beta_{i,i-1} k_{i-1}(h)$ (для заданного $r \ge 1$ и $\forall i = 1, 2, \ldots, r$); α_i и β_{ij} – некоторые постоянные (причем $\alpha_1 = 0$, $\beta_{11} = 0$). Таким образом, функции (1.6) имеют такой вид:

$$
\begin{aligned}
&k_1(h) = h f(x_0, y_0); \\
&k_2(h) = h f(x_0 + \alpha_2 h, y_0 + \beta_{21} k_1); \\
&k_3(h) = h f(x_0 + \alpha_3 h, y_0 + \beta_{31} k_1 + \beta_{32} k_2); \\
&\quad \vdots \qquad\qquad\qquad\qquad \vdots \\
&k_r(h) = h f(x_0 + \alpha_r h, y_0 + \beta_{r1} k_1 + \beta_{r2} k_2 + \ldots + \beta_{r,r-1} k_{r-1}).
\end{aligned} \tag{1.7}
$$

Выбрав величину h, которую называют также шагом интегрирования, и зная α_i, β_{ij} и p_{ri}, по формулам (1.7) можно последовательно вычислить функции $k_i(h)$, $i = 1, 2, \ldots, r$, а затем по формуле (1.5) – искомое значение $\Delta y_0(h)$ или, что то же самое, искомое значение $y(x_0 + h) = y_0 + \Delta y_0(h)$.

Если теперь в качестве начальных условий для уравнения (1.1) взять следующую начальную точку $x_1 = x_0 + h$ и вместо (1.2) начальное значение

$$y(x_1) = y_1 = y(x_0 + h),$$

то можно получить по тем же формулам значение искомого решения и в точке $x_2 = x_1 + h = x_0 + 2h$, а именно: значение $y(x_2) = y_2 = y(x_0 + 2h)$. Повторяя указанный процесс далее, получим таблицу значений искомого решения дифференциального уравнения – $y(x_m)$, где $x_m = x_0 + mh$, $m = 0, 1, 2, \ldots$. В связи с тем, что процедура построения такой таблицы является пошаговой (и на каждом следующем шаге используется только информация, полученная на предыдущем), метод Рунге – Кутта относят к классу одношаговых методов численного интегрирования обыкновенных дифференциальных уравнений.

1.2. Формулы Рунге – Кутта

Укажем теперь условия, которым подчиняется выбор постоянных α_i, β_{ij} и p_{ri} $(i = 2, 3, \ldots, r,\ j = 1, 2, \ldots, r-1)$ в (1.5), (1.7), определяющих при заданном $r \geq 1$ формулу Рунге – Кутта соответствующей степени, а именно, r-й степени. Как известно [1], эти условия состоят в том, чтобы разложение (1.4) и линейная комбинация (1.5) совпадали до возможно более высоких степеней h для произвольных правых частей уравнения (1.1) – $f(x, y)$ и любых значений шага интегрирования – h. При этом функция ошибки

$$\varphi_r(h) = y(x_0 + h) - y_0 - [p_{r1}k_1(h) + p_{r2}k_2(h) + \ldots + p_{rr}k_r(h)]$$

будет удовлетворять следующим условиям:

$$\varphi_r(0) = \varphi_r^{(1)}(0) = \varphi_r^{(2)}(0) = \ldots \varphi_r^{(s)}(0) = 0; \quad \varphi_r^{(s+1)}(0) \neq 0,$$

где $s \geq 1$ – некоторое число. Очевидно, что любой выбор постоянных α_i, β_{ij} и p_{ri} $(i = 2, 3, \ldots, r,\ j = 1, 2, \ldots, r-1)$ и p_{ri} $(i = 1, 2, \ldots, r)$ необходимо, в первую очередь, подчинить условию максимума числа s с учетом произвола в задании $f(x, y)$ и h. Тогда погрешность вычисления приращения

$\Delta y_0(h)$ и, соответственно, значения $y(x_0+h)$ на интервале $[x, x_0+h]$, то есть для одного шага (называемая также локальной погрешностью метода), будет определяться остаточным членом в форме Лагранжа

$$R_r(h) = \frac{h^{s+1}\varphi_r^{(s+1)}(\xi)}{(s+1)!}, \quad 0 \le \xi \le h.$$

При достаточно малых h главный член этой погрешности пропорционален h^{s+1}, и в связи с этим число s обычно называют порядком рассматриваемой формулы Рунге – Кутта. Отметим, что в общем случае $r \ge s$.

1.3. Примеры формул Рунге – Кутта различных степеней

Рассмотрим некоторые частные случаи формул Рунге-Кутта (1.5) – (1.7), полученных для различных r и при соответствующем выборе α_i, β_{ij} и p_{ri} ($i = 2, 3, \dots, r$, $j = 1, 2, \dots, r-1$) [1].

1.3.1. Вначале рассмотрим случай $r = 1$. При этом имеет место

$$\varphi_1(h) = y(x_0+h) - y(x_0) - p_{11} f(x_0, y_0);$$

$$\varphi_1^{(1)}(h) = y^{(1)}(x_0+h) - p_{11} f(x_0, y_0).$$

Отсюда следует, что $\varphi_1^{(1)}(0) = y^{(1)}(x_0) - p_{11} f(x_0, y_0) = 0$, то есть для произвольной правой части $f(x, y)$ в (1.1) возможно только $p_{11} = 1$. Далее имеет место $\varphi_1^{(2)}(0) = y^{(2)}(x_0)$, то есть в силу произвола $f(x, y)$ в общем случае $y^{(2)}(x_0) \ne 0$, и, стало быть, здесь $s = 1$. Поэтому для $r = 1$ существует единственная формула Рунге – Кутта:

$$\Delta y_0 = h f(x_0, y_0), \tag{1.8}$$

погрешность которой (на одном шаге интегрирования) будет равна

$$R_1(h) = \frac{h^2}{2}\varphi_1^{(2)}(\xi), \quad x_0 \le \xi \le x_0 + h,$$

или, что то же самое, погрешность формулы (1.8) имеет порядок h^2.

Процедура численного решения дифференциального уравнения (1.1) с начальными условиями (1.2), основанная на применении формулы (1.8), – метод Эйлера (или метод ломаных), описанный во введении.

1.3.2. В случае $r = 2$ необходимые и достаточные условия обращения в нуль первых двух производных функции $\varphi_2(h)$ при $h = 0$ имеют вид системы следующих уравнений:

$$p_{21} + p_{22} = 1;\ p_{22}\alpha_2 = \frac{1}{2};\ p_{22}\beta_{21} = \frac{1}{2}. \tag{1.9}$$

Здесь производная $\varphi_2^{(3)}(0)$, вообще говоря, в нуль не обращается. Решение системы (1.9) с учетом какого-либо дополнительного условия доставляет формулы интегрирования, имеющие порядок точности h^3. Например, здесь можно взять $\alpha_2 = \beta_{21} = 1$, тогда $p_{21} = p_{22} = \frac{1}{2}$ и, стало быть, имеет место:

$$\Delta y_0 = \frac{1}{2}(k_1 + k_2);\ k_1 = hf(x_0, y_0);\ k_2 = hf(x_0 + h, y_0 + k_1). \tag{1.10}$$

Формулы (1.10) отвечают методу Эйлера – Коши (второй улучшенный метод Эйлера [3]).

Еще одна формула будет получена, если взять $\alpha_2 = \beta_{21} = \frac{1}{2}$, тогда $p_{22} = 1$, $p_{21} = 0$ и, стало быть, имеем (первый улучшенный метод Эйлера [3], или модифицированный с пересчетом):

$$\Delta y_0 = k_2;\ k_1 = hf(x_0, y_0);\ k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1). \tag{1.11}$$

Кроме того, подходят также значения: $p_{21} = \frac{1}{4}$, $p_{22} = \frac{3}{4}$, $\alpha_2 = \beta_{21} = \frac{2}{3}$. Отсюда следует

$$\Delta y_0 = \frac{1}{4}(k_1 + 3k_2);\ k_1 = hf(x_0, y_0);\ k_2 = hf(x_0 + \frac{2}{3}h, y_0 + \frac{2}{3}k_1). \tag{1.12}$$

Очевидно, что приведенные варианты формул Рунге – Кутта (1.10) – (1.12), имеющих порядок точности h^3, отнюдь не исчерпывают всего множества допустимых решений системы (1.9). В связи с этим отметим, что выбор той или иной формулы зачастую обусловлен только удобством программирования из-за предполагаемого произвола в задании правой части уравнения (1.1). Учет каких-либо особенностей функции $f(x, y)$ может существенно ограничить множество практически допустимых решений системы (1.9).

1.3.3. В том случае, когда $r = 3$, вообще говоря, нельзя приравнять к нулю четвертую производную от функции $\varphi_3(h)$ (при $h = 0$); то есть здесь $s = 3$. Соответственно, условия $\varphi_3^{(1)}(0) = \varphi_3^{(2)}(0) = \varphi_3^{(3)}(0) = 0$ сводятся к таким условиям на коэффициенты α_2, α_3, β_{21}, β_{31}, β_{32}, p_{31}, p_{32} и p_{33}:

$$\alpha_2 = \beta_{21}\,;\ \alpha_3 = \beta_{31} + \beta_{22}\,;\ p_{31} + p_{32} + p_{33} = 1\,;$$

$$p_{32}\alpha_2 + p_{33}\alpha_3 = \frac{1}{2};\ p_{32}\alpha_2^2 + p_{33}\alpha_3^2 = \frac{1}{3};\ p_{33}\beta_{22}\alpha_2 = \frac{1}{6}. \qquad (1.13)$$

Решение системы (1.13) – при каких-либо дополнительных условиях – доставляют формулы Рунге – Кутта, погрешность которых имеет порядок h^4. Далее приводятся некоторые варианты таких формул.

Во-первых, для $\alpha_2 = \beta_{21} = \frac{1}{2}$, $\alpha_3 = 1$ получим $\beta_{31} = -1$, $\beta_{32} = 2$, $p_{31} = \frac{1}{6}$, $p_{32} = \frac{2}{3}$, $p_{33} = \frac{1}{6}$. Отсюда следует

$$\Delta y_0 = \frac{1}{6}(k_1 + 4k_2 + k_3)\,, \qquad (1.14)$$

где

$$k_1 = hf(x_0, y_0)\,,\quad k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1)\,,$$

$$k_3 = hf(x_0 + h, y_0 - k_1 + 2k_2)\,.$$

Во-вторых, для $\alpha_2 = \beta_{21} = \frac{1}{3}$, $\alpha_3 = \frac{2}{3}$ получим $\beta_{31} = 0$, $\beta_{32} = \frac{2}{3}$, $p_{31} = \frac{1}{4}$, $p_{32} = 0$, $p_{33} = \frac{3}{4}$ или (формула метода Рунге – Кутта – Гейне):

$$\Delta y_0 = \frac{1}{4}(k_1 + 3k_3)\,, \qquad (1.15)$$

где

$$k_1 = hf(x_0, y_0)\,,\quad k_2 = hf(x_0 + \frac{1}{3}h, y_0 + \frac{1}{3}k_1)\,,$$

$$k_3 = hf(x_0 + \frac{2}{3}h, y_0 + \frac{2}{3}k_2)\,.$$

В-третьих, пусть $\alpha_2 = \beta_{21} = \frac{1}{2}$, $\alpha_3 = \frac{3}{4}$. Тогда $\beta_{31} = 0$, $\beta_{32} = \frac{3}{4}$, $p_{31} = \frac{2}{9}$, $p_{32} = \frac{1}{3}$, $p_{33} = \frac{4}{9}$, и, стало быть, имеем

$$\Delta y_0 = \frac{1}{9}(2k_1 + 3k_2 + 4k_3), \tag{1.16}$$

где

$$k_1 = hf(x_0, y_0), \quad k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1),$$

$$k_3 = hf(x_0 + \frac{3}{4}h, y_0 + \frac{3}{4}k_2).$$

1.3.4. Наконец, рассмотрим случай, когда $r = 4$, так как он получил наиболее широкое применение в решении прикладных задач. Здесь удается обеспечить равенство нулю только первых четырех производных функции $\varphi_4(h)$ (при $h = 0$), а ее пятая производная в силу произвольности правой части уравнения (1.1) при $h = 0$ тождественно в нуль не обращается ни при каких значениях постоянных α_2, α_3, α_4, β_{21}, β_{31}, β_{32}, β_{41}, β_{42}, β_{43}, p_{41}, p_{42}, p_{43} и p_{44}. Они удовлетворяют следующей системе уравнений [1]:

$$\alpha_2 = \beta_{21};\ \alpha_3 = \beta_{31} + \beta_{22};\ \alpha_4 = \beta_{41} + \beta_{42} + \beta_{43};$$

$$p_{41} + p_{42} + p_{43} + p_{44} = 1;$$

$$p_{42}\alpha_2 + p_{43}\alpha_3 + p_{44}\alpha_4 = \frac{1}{2};\ p_{42}\alpha_2^2 + p_{43}\alpha_3^2 + p_{44}\alpha_4^2 = \frac{1}{3};$$

$$p_{42}\alpha_2^3 + p_{43}\alpha_3^3 + p_{44}\alpha_4^3 = \frac{1}{4};$$

$$p_{43}\beta_{32}\alpha_2 + p_{44}\beta_{42}\alpha_2 + p_{44}\beta_{43}\alpha_3 = \frac{1}{6};$$

$$p_{43}\beta_{32}\alpha_2\alpha_3 + p_{44}\beta_{42}\alpha_2\alpha_4 + p_{44}\beta_{43}\alpha_3\alpha_4 = \frac{1}{8};$$

$$p_{43}\beta_{32}\alpha_2^2 + p_{44}\beta_{42}\alpha_2^2 + p_{44}\beta_{43}\alpha_3^2 = \frac{1}{12};\ p_{44}\beta_{43}\beta_{32}\alpha_2 = \frac{1}{24}. \tag{1.17}$$

Отсюда при дополнительных условиях следуют варианты формул Рунге-Кутта, погрешность которых имеет порядок h^5.

Во-первых, одна из наиболее распространенных формул Рунге – Кутта четвертой степени и, соответственно, четвертого порядка точности (это так называемый стандартный метод Рунге – Кутта, правило «одной шестой») получается при $\alpha_2 = \alpha_3 = \beta_{21} = \beta_{32} = \frac{1}{2}$, $\alpha_4 = 1$, $\beta_{31} = \beta_{41} = \beta_{42} = 0$, $\beta_{43} = 1$, $p_{41} = \frac{1}{6}$, $p_{42} = \frac{1}{3}$, $p_{43} = \frac{1}{3}$, $p_{44} = \frac{1}{6}$, а именно:

$$\Delta y_0 = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{1.18}$$

где

$$k_1 = hf(x_0, y_0), \quad k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1),$$

$$k_3 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2), \; k_4 = hf(x_0 + h, y_0 + k_3).$$

Во-вторых, при $\alpha_2 = \beta_{21} = \frac{1}{3}$, $\alpha_3 = \frac{2}{3}$, $\alpha_4 = 1$, $\beta_{31} = -\frac{1}{3}$, $\beta_{42} = -1$, $\beta_{32} = \beta_{41} = \beta_{43} = 1$, $p_{41} = \frac{1}{8}$, $p_{42} = \frac{3}{8}$, $p_{43} = \frac{3}{8}$, $p_{44} = \frac{1}{8}$ получаем следующую формулу (правило «трех восьмых»):

$$\Delta y_0 = \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4), \tag{1.19}$$

где

$$k_1 = hf(x_0, y_0), \quad k_2 = hf(x_0 + \frac{1}{3}h, y_0 + \frac{1}{3}k_1),$$

$$k_2 = hf(x_0 + \frac{2}{3}h, y_0 - \frac{1}{3}k_1 + k_2), \; k_4 = hf(x_0 + h, y_0 + k_1 - k_2 + k_3).$$

В-третьих, при $\alpha_2 = \beta_{21} = \frac{1}{4}$, $\alpha_3 = \frac{1}{2}$, $\alpha_4 = 1$, $\beta_{31} = 0$, $\beta_{32} = \frac{1}{2}$, $\beta_{41} = 1$, $\beta_{42} = -2$, $\beta_{43} = 2$, $p_{41} = \frac{1}{6}$, $p_{42} = 0$, $p_{43} = \frac{2}{3}$, $p_{44} = \frac{1}{6}$ получим

$$\Delta y_0 = \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4), \tag{1.20}$$

где

$$k_1 = hf(x_0, y_0), \quad k_2 = hf(x_0 + \frac{1}{4}h, y_0 + \frac{1}{4}k_1),$$

$$k_3 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2), \; k_4 = hf(x_0 + h, y_0 + k_1 - 2k_2 + 2k_3).$$

Погрешности формул (1.18) – (1.20) на одном шаге интегрирования оцениваются величиной

$$R_4(h) = \frac{h^5}{120}\varphi_4^{(5)}(\xi), \quad x_0 \le \xi \le x_0 + h.$$

Другие варианты формул метода Рунге – Кутта для четвертого порядка точности приведены в Приложении.

1.3.5. Отметим, что при $r = 5$ увеличение порядка точности формул Рунге-Кутты не происходит; здесь оказывается возможным только $s = 4$. Поэтому такие формулы практического применения не находят. Однако можно получить соответствующие формулы, имеющие погрешность порядка h^6, но для этого необходимо выбирать $r \geq 6$ (получаемые при этом формулы численного интегрирования методом Рунге – Кутта, как правило, оказываются громоздкими и неудобными для практического применения).

В случае $r > 4$ соответствие между r и s нарушается. Метод Рунге – Кутта пятого порядка точности удается построить только при $r = 6$ (это формула Рунге – Кутта – Фельдберга [2]), шестого – при $r = 7$, седьмого – при $r = 9$, а в общем случае при $s > 7$ имеет место такая оценка: $r \geq s + 2$.

Формулы Рунге – Кутта – Фельдберга имеют следующий вид [4]:

$$\Delta \boldsymbol{y}_m = \frac{16}{35}\boldsymbol{k}_1 + \frac{6656}{12825}\boldsymbol{k}_3 + \frac{28561}{56430}\boldsymbol{k}_4 - \frac{9}{50}\boldsymbol{k}_5 + \frac{2}{55}\boldsymbol{k}_6, \tag{1.21}$$

где

$$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m), \quad \boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{4}h, \boldsymbol{y}_m + \frac{1}{4}\boldsymbol{k}_1),$$

$$\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{3}{8}h, \boldsymbol{y}_m + \frac{3}{32}\boldsymbol{k}_1 + \frac{9}{32}\boldsymbol{k}_2),$$

$$\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + \frac{12}{13}h, \boldsymbol{y}_m + \frac{1932}{2197}\boldsymbol{k}_1 - \frac{7200}{2197}\boldsymbol{k}_2 + \frac{7296}{2197}\boldsymbol{k}_3),$$

$$\boldsymbol{k}_5 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \frac{439}{216}\boldsymbol{k}_1 - 8\boldsymbol{k}_2 + \frac{3680}{513}\boldsymbol{k}_3 - \frac{845}{4104}\boldsymbol{k}_4),$$

$$\boldsymbol{k}_6 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m - \frac{8}{27}\boldsymbol{k}_1 + 2\boldsymbol{k}_2 - \frac{3544}{2565}\boldsymbol{k}_3 + \frac{1859}{4104}\boldsymbol{k}_4 - \frac{11}{40}\boldsymbol{k}_5).$$

1.4. Метод Рунге-Кутта для системы обыкновенных дифференциальных уравнений

Метод Рунге – Кутта применим и для решения систем обыкновенных дифференциальных уравнений первого порядка. В этой связи вначале здесь рассмотрим систему только двух уравнений

$$\frac{dy}{dx} = f(x, y, z), \quad \frac{dz}{dx} = g(x, y, z), \tag{1.21}$$

решение которой должно удовлетворять следующим начальным условиям:

$$y(x_0) = y_0; \quad z(x_0) = z_0, \tag{1.22}$$

где y_0 и z_0 – некоторые заданные начальные значения для искомого решения системы (1.21), то есть $y(x)$ и $z(x)$.

Предполагая достаточную гладкость правых частей системы (1.21), то есть функций $f(x, y, z)$ и $g(x, y, z)$, разложим ее искомое решение в ряд Тейлора в окрестности точки $x = x_0$:

$$y(x) = y_0 + (x - x_0)\, y_0^{(1)} + \frac{(x - x_0)^2}{2!}\, y_0^{(2)} + \ldots + \frac{(x - x_0)^{n+1}}{(n+1)!}\, y_0^{(n+1)} + o(\left|x - x_0\right|^{n+1}),$$

$$z(x) = z_0 + (x - x_0)\, z y_0^{(1)} + \frac{(x - x_0)^2}{2!}\, z_0^{(2)} + \ldots + \frac{(x - x_0)^{n+1}}{(n+1)!}\, z_0^{(n+1)} + o(\left|x - x_0\right|^{n+1}),$$

где $y_0^{(k)} = \dfrac{d^k y(x)}{dx^k}\Big|_{x=x_0}$, $z_0^{(k)} = \dfrac{d^k z(x)}{dx^k}\Big|_{x=x_0}$, $k = 1, 2, 3, \ldots$, – производные k-го порядка.

Обозначая $\Delta y_0(h) \cong y(x_0 + h) - y_0$, $\Delta z_0(h) \cong z(x_0 + h) - z_0$, а также $x - x_0 = h$, перепишем приведенные разложения с точностью до членов $o(\left|h\right|^{n+1})$ в виде

$$\Delta y_0 = h y_0^{(1)} + \frac{h^2}{2!}\, y_0^{(2)} + \ldots + \frac{h^{n+1}}{(n+1)!}\, y_0^{(n+1)},$$

$$\Delta z_0 = h z_0^{(1)} + \frac{h^2}{2!}\, z_0^{(2)} + \ldots + \frac{h^{n+1}}{(n+1)!}\, z_0^{(n+1)}. \tag{1.23}$$

Следуя методу Рунге – Кутта [1], введем такие линейные комбинации для соответствующих приближений Δy_0 и Δz_0:

$$\Delta y_0 = p_{r1} k_1(h) + p_{r2} k_2(h) + \ldots + p_{rr} k_r(h);$$

$$\Delta z_0 = q_{r1} l_1(h) + q_{r2} l_2(h) + \ldots + q_{rr} l_r(h)\,. \tag{1.24}$$

Здесь p_{ri}, q_{ri} $(i = 1, 2, \ldots, r)$ – некоторые постоянные, а функции $k_i(h)$ и $l_i(h)$ вводятся так (для заданного $r \ge 1$ и всех $i = 1, 2, \ldots, r$):

$$k_i(h) = hf(\xi_i, \eta_i, \varsigma_i)\,,\;\; l_i(h) = hg(\tilde{\xi}_i, \tilde{\eta}_i, \tilde{\varsigma}_i)\,,\; i = 1, 2, \ldots, r\,, \tag{1.25}$$

где $\quad \xi_i = x_0 + \alpha_i h \quad (\alpha_1 = 0); \qquad \tilde{\xi}_i = x_0 + \tilde{\alpha}_i h \qquad (\tilde{\alpha}_1 = 0);$

$$\eta_i = y_0 + \beta_{i1} k_1(h) + \beta_{i2} k_2(h) + \ldots + \beta_{i,\,i-1} k_{i-1}(h) \;\; (\beta_{i1} = 0);$$

$$\tilde{\eta}_i = y_0 + \tilde{\beta}_{i1} k_1(h) + \tilde{\beta}_{i2} k_2(h) + \ldots + \tilde{\beta}_{i,\,i-1} k_{i-1}(h) \;\; (\tilde{\beta}_{i1} = 0);$$

$$\varsigma_i = y_0 + \gamma_{i1}\, l_1(h) + \gamma_{i2}\, l_2(h) + \ldots + \gamma_{i,\,i-1}\, l_{i-1}(h) \;\; (\gamma_{i1} = 0);$$

$$\tilde{\varsigma}_i = y_0 + \tilde{\gamma}_{i1}\, l_1(h) + \tilde{\gamma}_{i2}\, l_2(h) + \ldots + \tilde{\gamma}_{i,\,i-1} l_{i-1}(h) \;\; (\gamma_{i1} = 0).$$

Как и ранее, задача заключается в таком выборе постоянных в (1.24), (1.25): α_i, β_{ij}, γ_{ij}, $\tilde{\alpha}_i$, $\tilde{\beta}_{ij}$, $\tilde{\gamma}_{ij}$, p_{ri}, q_{ri}, для которого разложения функций $\varphi_r(h) = \Delta y_0(h) - \sum\limits_{i=1}^{r} p_{ri} k_i(h)$, $\psi_r(h) = \Delta z_0(h) - \sum\limits_{i=1}^{r} q_{ri}\, l_i(h)$ по степеням h начинаются с возможно более высоких степеней.

Например, в [1, С.317] приведена полученная (для случая $r = 4$) система 17-ти уравнений, которым должны удовлетворять перечисленные выше постоянные в формулах Рунге – Кутта, имеющих порядок точности h^5. Значительное упрощение как этих формул, так и соответствующих им вычислительных алгоритмов достигается в общем случае с принятием следующих дополнительных условий:

$$\alpha_i = \tilde{\alpha}_i;\; \beta_{ij} = \tilde{\beta}_{ij} = \gamma_{ij} = \tilde{\gamma}_{ij};\; p_{ri} = q_{ri},\; i = 1, 2, \ldots, r \;\, (j = 1, 2, \ldots, r-1). \tag{1.26}$$

Тогда, выбирая случай $r = 4$ и, к примеру, такой набор постоянных с учетом (1.18), (1.26): $\alpha_2 = \alpha_3 = \beta_{21} = \beta_{32} = \gamma_{21} = \gamma_{32} = \frac{1}{2}$; $\alpha_4 = \beta_{43} = \gamma_{43} = 1$; $\beta_{31} = \beta_{41} = \beta_{42} = \gamma_{31} = \gamma_{41} = \gamma_{42} = 0$; $\;p_{41} = p_{44} = \frac{1}{6}$; $p_{42} = p_{43} = \frac{1}{3}$, получим из (1.24) и (1.25)

$$\Delta y_0 = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\,,$$

$$\Delta z_0 = \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4)\,, \tag{1.27}$$

где

$$k_1 = hf(x_0, y_0, z_0), \ \ l_1 = hg(x_0, y_0, z_0),$$

$$k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1, z_0 + \frac{1}{2}l_1), \ \ l_2 = hg(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1, z_0 + \frac{1}{2}l_1),$$

$$k_3 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2, z_0 + \frac{1}{2}l_2), \ \ l_3 = hg(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2, z_0 + \frac{1}{2}l_2),$$

$$k_4 = hf(x_0 + h, y_0 + k_3, z_0 + l_2), \ \ l_4 = hg(x_0 + h, y_0 + k_3, z_0 + l_2).$$

Следует отметить случай, когда в (1.21) $\frac{dz}{dx} = g(x, y, z) = 1$. Очевидно, что это уравнение с учетом (1.22) сразу же интегрируется; в результате получим $z(x) = x - x_0$. Этот случай можно рассматривать как замену независимой переменной в уравнении (1.1) на новую переменную, осуществляемую присоединением к уравнению (1.1) уравнения $\frac{dz}{dx} = 1$ с начальным условием $z(x_0) = x_0$; тогда задача Коши (1.1), (1.2) сводится к задаче (1.21), (1.22), в которой правые части не содержат независимой переменной. Такой прием часто используется в стандартных программах численного решения задачи Коши, поскольку это позволяет автоматически вычислять необходимые значения независимой переменной интегрированием, а не отдельным суммированием соответствующих долей шага.

Кроме того, это же оказывается удобным в том случае, когда момент x_f, отвечающий моменту окончания интегрирования, не задан, а определяется из некоторого дополнительного условия.

Процедура построения формул Рунге – Кутта применима и для систем, содержащих любое конечное число обыкновенных дифференциальных уравнений первого порядка. А именно, рассмотрим теперь начальную задачу для следующей системы:

$$\begin{aligned} \frac{dy^1}{dx} &= f^1(x, y^1, y^2, \dots, y^n); \\ \frac{dy^2}{dx} &= f^2(x, y^1, y^2, \dots, y^n); \\ &\vdots \\ \frac{dy^n}{dx} &= f^n(x, y^1, y^2, \dots, y^n), \end{aligned} \tag{1.28}$$

с начальными условиями

$$y^1(x_0) = y_0^1; \ \ y^2(x_0) = y_0^2; \ \dots; \ y^n(x_0) = y_0^n. \tag{1.29}$$

Вводя вектор-столбцы $\boldsymbol{y}=(y^1, y^2, \dots, y^n)$ и $\boldsymbol{f}=(f^1, f^2, \dots, f^n)$, задачу (1.28), (1.29) можно представить в векторно-матричном виде

$$\frac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(x, \boldsymbol{y}); \tag{1.30}$$

$$\boldsymbol{y}(x_0) = \boldsymbol{y}_0, \tag{1.31}$$

где $\boldsymbol{y}_0 = (\boldsymbol{y}_0^1, \boldsymbol{y}_0^2, \dots, \boldsymbol{y}_0^n)$ – вектор-столбец начальных значений искомого решения в точке $x = x_0$.

Очевидно, что в предположении достаточной гладкости компонент правой части уравнения (1.30) в виде вектор-функции $\boldsymbol{f}(x, \boldsymbol{y})$, искомое решение также можно разложить в ряд Тейлора в окрестности точки $x = x_0$, а именно:

$$\boldsymbol{y}(x) = \boldsymbol{y}_0 + h\boldsymbol{y}_0^{(1)} + \frac{h^2}{2!}\,\boldsymbol{y}_0^{(2)} + \dots + \frac{h^{n+1}}{(n+1)!}\,\boldsymbol{y}_0^{(n+1)} + o(|h|^{n+1}),$$

где для фиксированного $x \neq x_0$ введено обозначение $h = x - x_0$.

Соответственно, с учетом предположений, аналогичных (1.27), для $\Delta\boldsymbol{y}_0(h) = \boldsymbol{y}(x_0 + h) - \boldsymbol{y}_0$ следует отыскивать такую линейную комбинацию:

$$\Delta\boldsymbol{y}_0(h) = p_{r1}\boldsymbol{k}_1(h) + p_{r2}\boldsymbol{k}_2(h) + \dots + p_{rr}\boldsymbol{k}_r(h), \tag{1.32}$$

где вектор-функции $\boldsymbol{k}_i(h)$, $i = 1, 2, \dots, r$, вводятся следующим образом:

$$\begin{aligned}
&\boldsymbol{k}_1(h) = h\boldsymbol{f}(x_0, \boldsymbol{y}_0);\\
&\boldsymbol{k}_2(h) = h\boldsymbol{f}(x_0 + \alpha_2 h, \boldsymbol{y}_0 + \beta_{21}\boldsymbol{k}_1);\\
&\boldsymbol{k}_3(h) = h\boldsymbol{f}(x_0 + \alpha_3 h, \boldsymbol{y}_0 + \beta_{31}\boldsymbol{k}_1 + \beta_{32}\boldsymbol{k}_2);\\
&\quad\vdots \qquad\qquad\qquad\qquad\qquad \vdots\\
&\boldsymbol{k}_r(h) = h\boldsymbol{f}(x_0 + \alpha_r h, \boldsymbol{y}_0 + \beta_{r1}\boldsymbol{k}_1 + \beta_{r2}\boldsymbol{k}_2 + \dots + \beta_{r,r-1}\boldsymbol{k}_{r-1}).
\end{aligned} \tag{1.33}$$

Таким образом, назначив величину h и задавая соответствующий набор постоянных α_i, β_{ij} и p_{ri}, в (1.32) и (1.33) можно получить соответствующие формулы Рунге – Кутта для вычисления искомого приближенного решения системы (1.21) $y(x_0 + h) = y_0 + \Delta y_0(h)$.

2. Программная реализация численного решения задачи Коши для системы обыкновенных дифференциальных уравнений первого порядка

2.1. Основные варианты численного решения задачи Коши

Рассмотрим задачу Коши на интервале $[x_0, x_f]$ ($x_0 < x_f < \infty$, где x_f – некоторое заданное или определяемое из каких-либо условий число) для векторного дифференциального уравнения

$$\frac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(x, \boldsymbol{y}) \tag{2.1}$$

с начальным условием

$$\boldsymbol{y}(x_0) = \boldsymbol{y}_0, \tag{2.2}$$

где $\boldsymbol{y} = (y^1, y^2, \dots, y^n)$, $\boldsymbol{f} = (f^1, f^2, \dots, f^n)$, а $\boldsymbol{y}_0 = (\boldsymbol{y}_0^1, \boldsymbol{y}_0^2, \dots, \boldsymbol{y}_0^n)$ – вектор начальных значений искомого решения $\boldsymbol{y} = \boldsymbol{y}(x)$ в точке $x = x_0$.

В общем случае решение задачи Коши (2.1), (2.2) – вектор-функция $\boldsymbol{y} = \boldsymbol{y}(x)$, $\forall x \in [x_0, x_f]$. Практически такое решение задачи Коши можно построить только в случае применения аналитических методов решения. Для случая численного решения задачи Коши можно построить лишь ее приближенное решение в виде некоторой таблицы значений искомой функции $y(x)$ в заданных точках x_i, $i = 0, 1, 2, \dots$, совокупность которых называется сеткой на интервале $[x_0, x_f]$, а точки x_i – ее узлами. Наиболее удобной является равномерная сетка на $[x_0, x_f]$ с шагом h, когда ее узлы задаются так: $x_i = x_0 + ih$, $i = 0, 1, 2, \dots$.

Построение таблицы значений $\boldsymbol{y}_i = \boldsymbol{y}(x_i)$, $i = 0, 1, 2, \dots$, для заданного шага h, как приближенного решения методом Рунге – Кутта задачи Коши (2.1), (2.2), сводится к решению последовательности задач Коши для дифференциального уравнения (2.1) на подынтервалах $[x_0 + ih, x_0 + (i+1)h]$. При этом начальные условия задаются так: для $i = 0$ или для первого подынтервала $[x_0, x_0 + h]$ это значение $\boldsymbol{y}_0$, согласно (2.2). Пусть на первом подынтервале приближенное решение представляется в виде вектор-функции $\tilde{\boldsymbol{y}}_1(x)$, для которой выполняется условие $\tilde{\boldsymbol{y}}_1(x_0) = \boldsymbol{y}_0$. Тогда, обозначая $\boldsymbol{y}_1 = \tilde{\boldsymbol{y}}_1(x_0 + h)$, получим начальное значение для второго подынтервала $[x_0 + h, x_0 + 2h]$, на котором приближенное решение – вектор-функции $\tilde{\boldsymbol{y}}_2(x)$. Очевидно, что $\boldsymbol{y}_2 = \tilde{\boldsymbol{y}}_2(x_0 + 2h)$ – начальное значение для третьего подынтервала $[x_0 + 2h, x_0 + 3h]$ и т.д.

В общем случае, в силу возможного произвола в задании x_f, для некоторого $i = i_f$ получим: $x_0 + (i_f - 1)h < x_f \le x_0 + i_f h$. Поэтому в таблицу значений $\boldsymbol{y}_i$, $i = 0, 1, 2, \dots, i_f - 1$, следует включать также и значение $\boldsymbol{y}(x_f)$. Тем более, что в качестве одного из вариантов численного решения задачи Коши (2.1), (2.2) может быть определение только одного значения $\boldsymbol{y}(x_f)$. Очевидно, что на i_f-м подынтервале для вычисления $\boldsymbol{y}(x_f)$ необходимо выбирать шаг $h_f = x_f - x_0 - (i_f - 1)h \le h$. Но сетку на интервале $[x_0, x_f]$ можно в самом начале задать так, чтобы $h = (x_f - x_0)/i_f$, где i_f – некоторое заданное число подынтервалов разбиения. При этом следует отметить, что выбор числа i_f обусловлен в первую очередь требованиями ограничения локальной погрешности соответствующей формулы метода Рунге – Кутта, которая будет пропорциональна величине h^{s+1}, где s – порядок точности применяемой формулы.

Таким образом, в качестве *основного варианта* численного решения задачи Коши (2.1), (2.2) методом Рунге – Кутта (для заданного $r \ge 1$ или, что то же самое, заданной степени соответствующей формулы метода) примем вариант построения таблицы с заданным шагом $h > 0$ со значениями ее приближенного решения $\boldsymbol{y}_i = \boldsymbol{y}(x_i)$, $i = 0, 1, 2, \dots,$ на некотором заданном интервале $[x_0, x_f]$. В указанную таблицу должно быть включено также и значение $\boldsymbol{y}(x_f)$.

Отметим, что результаты решения задачи Коши (2.1), (2.2) в рамках указанного варианта можно использовать и для построения приближенных аналитических решений $\tilde{\boldsymbol{y}}_i(x)$, при необходимости, на каждом из подынтервалов $[x_{i-1}, x_i]$, $i = 1, 2, 3, \dots,$ используя только соответствующие значения $\boldsymbol{y}_{i-1}$ и $\boldsymbol{y}_i$. Это же возможно и при необходимости построения таблицы с меньшим шагом, чем шаг, который был выбран для численного решения задачи Коши.

Например, аналитическое выражение для $\tilde{\boldsymbol{y}}_i(x)$ можно отыскивать в следующем виде:

$$\tilde{\boldsymbol{y}}_i(x) = \boldsymbol{a}_{i0} + (x - x_{i-1})\boldsymbol{a}_{i1} + (x - x_{i-1})^2 \boldsymbol{a}_{i2} + (x - x_{i-1})^3 \boldsymbol{a}_{i3}, \qquad (2.3)$$

где $\boldsymbol{a}_{ik}$, $k = 0, 1, 2, 3$, – вектор-коэффициенты, подлежащие определению. Для вычисления коэффициентов $\boldsymbol{a}_{ik}$, $k = 0, 1, 2, 3$, вектор-функции (2.3) можно потребовать выполнения следующих условий: во-первых, при $x = x_{i-1}$

$$\tilde{\boldsymbol{y}}_i(x_{i-1}) = \boldsymbol{y}_{i-1}; \quad \tilde{\boldsymbol{y}}_i'(x_{i-1}) = \boldsymbol{f}(x_{i-1}, \boldsymbol{y}_{i-1}); \qquad (2.4)$$

во-вторых, при $x = x_i$

$$\tilde{\boldsymbol{y}}_i(x_i) = \boldsymbol{y}_i; \quad \tilde{\boldsymbol{y}}'_i(x_i) = \boldsymbol{f}(x_i, \boldsymbol{y}_i). \tag{2.5}$$

С учетом выражения (2.3), непосредственно из условий (2.4) следует: $\boldsymbol{a}_{i0} = \boldsymbol{y}_{i-1}$; $\boldsymbol{a}_{i1} = \boldsymbol{f}(x_{i-1}, \boldsymbol{y}_{i-1})$. Учитывая условия (2.5) и обозначая $x_i - x_{i-1} = h$, в соответствии с (2.3) получим:

$$\boldsymbol{y}_i = \boldsymbol{a}_{i0} + h\boldsymbol{a}_{i1} + h^2\boldsymbol{a}_{i2} + h^3\boldsymbol{a}_{i3}; \quad \boldsymbol{f}(x_i, \boldsymbol{y}_i) = \boldsymbol{a}_{i1} + 2h\boldsymbol{a}_{i2} + 3h^2\boldsymbol{a}_{i3}.$$

Если ввести следующие обозначения:

$$\boldsymbol{b}_i = \frac{\boldsymbol{y}_i - \boldsymbol{y}_{i-1} - h\,\boldsymbol{f}(x_{i-1}, \boldsymbol{y}_{i-1})}{h^2}; \qquad \boldsymbol{c}_i = \frac{\boldsymbol{f}(x_i, \boldsymbol{y}_i) - \boldsymbol{f}(x_{i-1}, \boldsymbol{y}_{i-1})}{h},$$

то получим $\boldsymbol{a}_{i2} = 3\boldsymbol{b}_i - \boldsymbol{c}_i$, $\boldsymbol{a}_{i3} = (\boldsymbol{c}_i - 2\boldsymbol{b}_i)/h$.

Еще одним вариантом численного решения задачи Коши (2.1), (2.2), который часто возникает при решении широкого круга прикладных задач, является вариант построения таблицы с заданным шагом $h > 0$ значений ее приближенного решения $\boldsymbol{y}_i = \tilde{\boldsymbol{y}}(x_i)$, $i = 0, 1, 2, \dots$, при условии, что правая граница интервала $[x_0, x_f]$, то есть $x_f > x_0$, непосредственно определяется в ходе решения задачи, исходя из некоторых дополнительных условий. В общем случае совокупность таких условий можно записать в виде:

$$\psi_k(x_f, \boldsymbol{y}(x_f)) = 0, \ k = 1, 2, \dots, l, \tag{2.6}$$

где ψ_k – некоторые заданные функции, а l – число условий. При этом предполагается, что значение x_f определяется при выполнении хотя бы одного из указанных условий (2.6), а именно, когда решение уравнения (2.1) – вектор-функция $\boldsymbol{y}(x)$ – достигает хотя бы одного из многообразий $\psi_k(x, \boldsymbol{y}) = 0$. Поэтому условия (2.6) не являются условиями, налагаемыми на решение уравнения (2.1) $\boldsymbol{y}(x)$ на правой границе интервала $[x_0, x_f]$. Иначе требование выполнения всех перечисленных выше условий (2.6) фактически сводит задачу Коши к соответствующей краевой задаче. Таким образом, условия (2.6) суть условия «останова» процедуры решения задачи (2.1), (2.2).

Индикацией выполнения какого-либо из условий (2.6) в ходе решения задачи Коши может служить смена знака соответствующей функции ψ_k на текущем или, для определенности, на m-ом шаге интегрирования, то есть на подынтервале $[x_{m-1}, x_m]$, когда, например, хотя бы для одного из $1 \le k \le m$ выполняются условия: $\psi_k(x_{m-1}, \tilde{\boldsymbol{y}}(x_{m-1})) > 0$ и $\psi_k(x_m, \tilde{\boldsymbol{y}}(x_m)) < 0$. Тогда определение точки $x = \tilde{x}_f$, для которой выполняется условие

$$|ш_k(\widetilde{x}_f, \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f))| \le \varepsilon_k, \tag{2.7}$$

где ε_k – заданная точность выполнения k-го условия (2.6), равносильно решению уравнения $ш_k(x, \widetilde{\boldsymbol{y}}_m(x)) = 0$. В связи с этим следует отметить, что функции $ш_k$ $(1 \le k \le l)$ могут быть существенно нелинейными функциями. Поэтому решение соответствующего уравнения целесообразно отыскивать методом последовательных приближений.

Например, в качестве начального приближения здесь можно выбирать точку $x = \widetilde{x}_f^0$, вычисляемую с учетом указанных выше знаков для функции $ш_k$ в точках x_{m-1} и x_m, а также с учетом дополнительного обозначения $h_0 = h$, так:

$$\widetilde{x}_f^0 = x_{m-1} + \frac{h_0\, ш_k(x_{m-1}, \widetilde{\boldsymbol{y}}_m(x_{m-1}))}{ш_k(x_{m-1}, \widetilde{\boldsymbol{y}}_m(x_{m-1})) - ш_k(x_m, \widetilde{\boldsymbol{y}}_m(x_m))}. \tag{2.8}$$

Если для точки $\widetilde{x}_f^0$, которая находится по формуле (2.8), условие (2.7) не выполняется, то рассматриваемая процедура может быть продолжена в зависимости от знака $ш_k(\widetilde{x}_f^0, \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f^0))$. А именно, если $ш_k(\widetilde{x}_f^0, \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f^0)) < 0$, то следует выбрать шаг интегрирования равный

$$h_1 = \frac{h_0\, ш_k(x_{m-1}, \widetilde{\boldsymbol{y}}_m(x_{m-1}))}{ш_k(x_{m-1}, \widetilde{\boldsymbol{y}}_m(x_{m-1})) - ш_k(x_m, \widetilde{\boldsymbol{y}}_m(x_m))} \tag{2.9}$$

и, соответственно, принять: $x_m = \widetilde{x}_f^0$, $\widetilde{\boldsymbol{y}}_m(x_m) = \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f^0)$.

Если же для точки $\widetilde{x}_f^0$ будет получено $ш_k(\widetilde{x}_f^0, \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f^0)) > 0$, то интегрирование уравнения (2.1) следует продолжить с шагом (2.9) до тех пор, пока функция $ш_k$ не изменит знак, либо не будет выполнено условие (2.7).

В связи с изложенной процедурой обеспечения выполнения условий «останова» (2.6) следует отметить, что на подынтервале $[x_{m-1}, x_m]$ могут быть одновременно выполнены несколько условий: $ш_{k_q}(x_{m-1}, \widetilde{\boldsymbol{y}}(x_{m-1})) > 0$; $ш_{k_q}(x_m, \widetilde{\boldsymbol{y}}(x_m)) < 0$, где k_q, $q = 1, 2, \ldots, \nu$ $(1 \le \nu \le l)$, – соответствующие номера условий (2.6). В этом случае точкой «останова» $\widetilde{x}_f$ может выбираться либо точка, для которой это значение является наименьшим, либо точка, определяемая из условия (2.7) с минимальным номером $k_1 < k_2 < \ldots < k_\nu$.

2.2. Программная реализация формул Рунге – Кутта четвертой степени для численного решения задачи Коши на заданном интервале

2.2.1. В настоящем подразделе поэтапно будет изложена процедура разработки программы численного решения задачи Коши (2.1), (2.2) для формул Рунге-Кутта четвертой степени (четвертого порядка точности), так как они получили наибольшее применение для решения прикладных задач. Построение программной реализации метода Рунге – Кутта будет проведено для варианта решения задачи Коши на заданном интервале $[x_0, x_f]$, когда точка $x_f > x_0$ фиксирована. Кроме того, предполагается, что на этом интервале вводится равномерная сетка, шаг которой задан и равен h.

Таким образом, интервал $[x_0, x_f]$ разбивается на подынтервалы: $[x_0 + (i-1)h, x_0 + ih]$, $i = 1, 2, 3, \ldots$. Причем для первого из них $[x_0, x_0 + h]$ начальное условие задается в соответствии с (2.2), то есть $\widetilde{\boldsymbol{y}}_1(x_0) = \boldsymbol{y}_0$, где $\widetilde{\boldsymbol{y}}_1(x)$ – приближенное решение задачи Коши (2.1), (2.2) на этом подынтервале. Соответственно, на остальных подынтервалах $[x_0 + (i-1)h, x_0 + ih]$ (для $i > 1$) начальные условия будут задаваться как значения получаемых решений на предшествующих подынтервалах, то есть в виде

$$\widetilde{\boldsymbol{y}}_i(x_0 + (i-1)h) = \boldsymbol{y}_{i-1} \quad (i = 2, 3, 4, \ldots), \tag{2.10}$$

где $\boldsymbol{y}_{i-1} = \widetilde{\boldsymbol{y}}_{i-1}(x_0 + ih)$, а $\widetilde{\boldsymbol{y}}_i(x)$ – приближенное решение задачи Коши (2.1), (2.10) на i-м подынтервале.

В силу произвола при задании значений h и x_f для некоторого $i = i_f$ (число подынтервалов разбиения на интервале $[x_0, x_f]$), в общем случае имеет место: $x_0 + (i_f - 1)h < x_f \le x_0 + i_f h$. Отсюда для i_f-го подынтервала $[x_0 + (i_f - 1)h, x_f]$ шаг сетки будет равен $h_f = x_f - x_0 - (i_f - 1)h \le h$.

В результате приближенного численного решения задачи Коши (2.1), (2.2) на интервале $[x_0, x_f]$ с использованием выбранной формулы Рунге – Кутта будут получены соответствующие значения: $\boldsymbol{y}_i$, $i = 0, 1, \ldots, i_f - 1$, где $x_i = x_0 + ih$, а также $\boldsymbol{y}_{i_f}(x_{i_f})$.

Совокупность определяемых соответствующей формулой Рунге-Кутта однотипных вычислительных операций, которые выполняются на каждом подынтервале разбиения интервала $[x_0, x_f]$, далее будем называть шагом интегрирования. Следовательно, i-й шаг интегрирования будет соответствовать решению i-й задачи Коши (2.1), (2.10) на i-м подынтервале $[x_0 + (i-1)h, x_0 + ih]$.

С учетом соотношений (1.32), (1.33) значения $\Delta \boldsymbol{y}_i$ или, в конечном счете, значения $\boldsymbol{y}_i = \boldsymbol{y}_{i-1} + \Delta \boldsymbol{y}_i$, $i = 1, 2, \ldots, i_f$, на каждом шаге интегрирования для $r = 4$ определяются по формулам Рунге – Кутта:

$$\Delta \boldsymbol{y}_i = p_{41}\boldsymbol{k}_1 + p_{42}\boldsymbol{k}_2 + p_{43}\boldsymbol{k}_3 + p_{44}\boldsymbol{k}_4, \tag{2.11}$$

где вектор-коэффициенты $\boldsymbol{k}_j$, $j = 1, 2, 3, 4$, вычисляются так:

$$\boldsymbol{k}_1 = h\boldsymbol{f}(x_0 + (i-1)h,\ \boldsymbol{y}_{i-1}); \tag{2.12}$$

$$\boldsymbol{k}_2 = h\boldsymbol{f}(x_0 + (i-1)h + \alpha_2 h,\ \boldsymbol{y}_{i-1} + \beta_{21}\boldsymbol{k}_1); \tag{2.13}$$

$$\boldsymbol{k}_3 = h\boldsymbol{f}(x_0 + (i-1)h + \alpha_3 h,\ \boldsymbol{y}_{i-1} + \beta_{31}\boldsymbol{k}_1 + \beta_{32}\boldsymbol{k}_2); \tag{2.14}$$

$$\boldsymbol{k}_4 = h\boldsymbol{f}(x_0 + (i-1)h + \alpha_4 h,\ \boldsymbol{y}_{i-1} + \beta_{41}\boldsymbol{k}_1 + \beta_{42}\boldsymbol{k}_2 + \beta_{43}\boldsymbol{k}_3). \tag{2.15}$$

2.2.2. Для изложения соответствующего вычислительного алгоритма необходима конкретизация приведенных соотношений (2.11) – (2.15), то есть требуется задавать как величину h, так и значения постоянных в применяемой формуле Рунге-Кутта: α_2, α_3, α_4, β_{21}, β_{31}, β_{32}, β_{41}, β_{42}, β_{43}, p_{41}, p_{42}, p_{43}, p_{44}.

Итак, в соответствии с (2.11) – (2.15), алгоритм последовательного (пошагового) решения следующих задач Коши: для $i = 1$ – (2.1), (2.2); для $i = 2, 3, 4, \ldots, i_f$ – (2.1), (2.10) – можно записать в следующем виде.

Шаг 1: инициализация локальных переменных программы, в том числе: а) начальных условий для первого шага интегрирования в виде: $\xi_1 = \xi_0 = x_0$; $\boldsymbol{z}_1 = \boldsymbol{z}_0 = \boldsymbol{y}_0$; б) обнуление приращений $\Delta \boldsymbol{y} = 0$; в) обнуление счетчика шагов интегрирования: $step = 0$.

Шаг 2: $step = step + 1$ ($i = step$, номер текущего шага интегрирования).

Шаг 3: если $\xi_0 + h > x_f$, то $h = x_f - \xi_0$ (коррекция шага интегрирования).

Шаг 4: вычисление: $\boldsymbol{k}_1 = h\boldsymbol{f}(\xi_1, \boldsymbol{z}_1)$; $\Delta \boldsymbol{y} = \Delta \boldsymbol{y} + p_{41}\boldsymbol{k}_1$.

Шаг 5: $\xi_2 = \xi_0 + \alpha_2 h$; $\boldsymbol{z}_2 = \boldsymbol{z}_0 + \beta_{21}\boldsymbol{k}_1$.

Шаг 6: вычисление: $\boldsymbol{k}_2 = h\boldsymbol{f}(\xi_2, \boldsymbol{z}_2)$; $\Delta \boldsymbol{y} = \Delta \boldsymbol{y} + p_{42}\boldsymbol{k}_2$.

Шаг 7: $\xi_3 = \xi_0 + \alpha_3 h$; $\boldsymbol{z}_3 = \boldsymbol{z}_0 + \beta_{31}\boldsymbol{k}_1 + \beta_{32}\boldsymbol{k}_2$.

Шаг 8: вычисление: $\boldsymbol{k}_3 = h\boldsymbol{f}(\xi_3, \boldsymbol{z}_3)$; $\Delta \boldsymbol{y} = \Delta \boldsymbol{y} + p_{43}\boldsymbol{k}_3$.

Шаг 9: $\xi_4 = \xi_0 + \alpha_4 h$; $\boldsymbol{z}_4 = \boldsymbol{z}_0 + \beta_{41}\boldsymbol{k}_1 + \beta_{42}\boldsymbol{k}_2 + \beta_{43}\boldsymbol{k}_3$.

Шаг 10: вычисление: $\boldsymbol{k}_4 = h\boldsymbol{f}(\xi_4, \boldsymbol{z}_4)$; $\Delta \boldsymbol{y} = \Delta \boldsymbol{y} + p_{44}\boldsymbol{k}_4$.

Шаг 11: вычисление: $\boldsymbol{y}_i = \boldsymbol{z}_0 + \Delta \boldsymbol{y}$ (решение i-й задачи Коши (2.1), (2.10)).

Шаг 12: если $\xi_0 = x_f$ (или $\xi_0 \ge x_f$), то выход из программы.

Шаг 13: формирование начальных условий для следующего шага интегрирования: $\xi_0 = \xi_0 + h$; $\xi_1 = \xi_0$; $\boldsymbol{z}_0 = \boldsymbol{y}_i$; $\boldsymbol{z}_1 = \boldsymbol{z}_0$.

Шаг 14: переход к шагу 2.

Очевидно, что в приведенном выше алгоритме (далее – Алгоритм -1) шаги №№ 4 – 10, связанные с вычислением вектор-коэффициентов $\boldsymbol{k}_j$ (2.12) – (2.15), можно реализовать с помощью одного цикла (далее – это основной цикл программы); список значений параметра для этого цикла, очевидно, имеет вид: $j = 1, 2, 3, 4$. Кроме того, следует учесть, что ξ_1, ξ_2, ξ_3, ξ_4 в Алгоритме -1 – значения одной и той же переменной ξ. И то же самое относится к векторам $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, $\boldsymbol{z}_3$, $\boldsymbol{z}_4$, которые располагаются в одном и том же рабочий массив $\boldsymbol{z}$, но с различными значениями компонент (или элементов). При этом переменная ξ_0 и вектор $\boldsymbol{z}_0$ предназначаются для хранения соответствующих значений в начале текущего шага интегрирования до его завершения. В результате модификаций вышеприведенного Алгоритма -1 получим Алгоритм -2, запись которого имеет следующий вид.

Шаг 1: инициализация локальных переменных программы, в том числе: а) начальных условий для первого шага интегрирования в виде: $\xi_0 = x_0$; $\xi = x_0$; $\boldsymbol{z}_0 = \boldsymbol{y}_0$; $\boldsymbol{z} = \boldsymbol{y}_0$ б) обнуление приращений $\Delta\boldsymbol{y} = 0$; в) обнуление счетчика шагов интегрирования: $step = 0$.

Шаг 2: $step = step + 1$ ($i = step$, номер текущего шага интегрирования).

Шаг 3: если $\xi_0 + h > x_f$, то $h = x_f - \xi_0$ (коррекция шага интегрирования).

Шаг 4: $j = 0$ (начало основного цикла).

Шаг 5: $j = j + 1$ (счетчик подшагов текущего шага интегрирования).

Шаг 6: вычисление: $\boldsymbol{k}_j = h\boldsymbol{f}(\xi, \boldsymbol{z})$; $\Delta\boldsymbol{y} = \Delta\boldsymbol{y} + p_{4j}\boldsymbol{k}_j$.

Шаг 7: если $j < 4$, то переход к шагу 12.

Шаг 8: вычисление: $\boldsymbol{y}_i = \boldsymbol{z}_0 + \Delta\boldsymbol{y}$.

Шаг 9: формирование начальных условий для следующего шага интегрирования: $\xi_0 = \xi_0 + h$; $\xi = \xi_0$; $\boldsymbol{z}_0 = \boldsymbol{y}_i$; $\boldsymbol{z} = \boldsymbol{z}_0$.

Шаг 10: если $\xi_0 \ge x_f$, то выход из программы.

Шаг 11: переход к шагу 2 (на следующий шаг интегрирования).

Шаг 12: вычисление ($j < 4$): $\xi = \xi_0 + \alpha_{j+1} h$ и $\boldsymbol{z} = \boldsymbol{z}_0 + \sum_{l=1}^{j} \beta_{j+1,l}\boldsymbol{k}_l$;

и переход к шагу 5 (продолжение выполнения основного цикла).

2.2.3. Текст программы, реализующей Алгоритм -2 (рис. 1), записан с использованием простейших операторов ФОРТРАН, а именно, следующих операторов: **DO** – оператор цикла (для выполнения группы операторов, из которых последний имеет метку); **IF (…) THEN** – условный оператор, который выполняется только при выполнении соответствующего условия; **GO TO** – оператор безусловной передачи управления на указанный (помеченный) оператор; **CALL** – оператор вызова подпрограммы; **CONTINUE** – «пустой» оператор. В тексте программы, для удобства ее чтения, используются строки комментариев, отмеченные «Comment», и приведены только исполняемые операторы.

В таблице 1 приведен список всех используемых в программе переменных и соответствующих им идентификаторов.

Таблица 1

№ п/п	Переменная, массив или константа	Идентификатор	Примечание
1	x_0	X0	левая граница интервала
2	x_f	XFIN	правая граница интервала ($x_f > x_0$)
3	$\boldsymbol{y}_0$	Y0 (1:N)	вектор начальных условий (2.2)
	n	N	порядок системы (2.1)
4	h	HINT	шаг интегрирования
5	α_2, α_3, α_4	ALPHA (2:4)	константы формулы Рунге-Кутта
6	β_{21}, β_{31}, β_{32}, β_{41}, β_{42}, β_{43}	BETTA (2:4, 1:3)	константы формулы Рунге-Кутта
7	p_{41}, p_{42}, p_{43}, p_{44}	P4 (1:4)	константы формулы Рунге-Кутта
8	$\boldsymbol{k}_1$, $\boldsymbol{k}_2$, $\boldsymbol{k}_3$, $\boldsymbol{k}_4$	K (1:4, 1:N)	вектор-коэффициенты (2.12) – (2.15)
9	ξ	T	независимая переменная
10	ξ_0	T0	независимая переменная на начало текущего шага интегрирования
11	$\boldsymbol{z}$	Z (1:N)	рабочий массив
12	$\boldsymbol{z}_0$	Z0 (1:N)	массив начальных условий для текущего шага интегрирования
13	$\boldsymbol{y}_i$	YI (1:N)	массив решений задач (2.1), (2.10)
14	$\Delta\boldsymbol{y}$	DY (1:N)	массив для накопления приращений
15	$\boldsymbol{f}(\xi, \boldsymbol{z})$	PR (1:N)	вектор правых частей системы (2.1)
16		STEP	счетчик шагов интегрирования
17	i, j, l	I, J, L	параметры циклов
18		SUM	вспомогательная переменная

№ п/п	метка	оператор
	Comment	**Шаг 1**
01		T0 = X0
02		T = X0
03		DO 10 I = 1, N
04		Z0 (I) = Y0 (I)
05		Z (I) = Y0 (I)
06	10	DY (I) = 0.
07		STEP = 0
	Comment	**Шаг 2**
08	20	STEP = STEP + 1
	Comment	**Шаг 3**
09		**IF** (T0 + HINT > XFIN) **THEN** HINT = XFIN – T0
	Comment	**Шаг 4 и 5** (начало основного цикла по j)
10	25	**DO** 90 J = 1, 2, 3, 4
	Comment	**Шаг 6**
11		**CALL** CALCPR (T, Z, PR)
12		**DO** 30 I = 1, N
13		K (J, I) = HINT ∗ PR (I)
14	30	DY (I) = DY (I) + P4 (J) ∗ K (J, I)
	Comment	**Шаг 7**
15		**IF** (J < 4) **THEN GO TO** 60
	Comment	**Шаг 8**
16		**DO** 40 I = 1, N
17	40	YI (I) = Z0 (I) + DY (I)
	Comment	**Шаг 9**
18		T0 = T0 + HINT
19		T = T0
20		**DO** 50 I = 1, N
21		Z0 (I) = YI (I)
22	50	Z (I) = Z0 (I)
	Comment	**Шаг 10**
23		**IF** (T0 ≥ XFIN) **THEN GO TO** 100
	Comment	**Шаг 11**
24		**GO TO** 20
	Comment	**Шаг 12**
25	60	**CONTINUE**
26		T = T0 + ALPHA (J + 1) ∗ HINT
27		**DO** 80 I = 1, N
28		SUM = 0.
29		**DO** 70 L = 1, J
30	70	SUM = SUM + BETTA (J + 1, L) ∗ K (L, I)
31	80	Z (I) = Z0 (I) + SUM
	Comment	Конец основного цикла по j
32	90	**CONTINUE**
33	100	**CONTINUE**

Рис. 1. Текст программы RUNGE 4 .2 (Алгоритм -2)

2.2.4. Очевидно, что текст программы RUNGE 4.2 можно было бы представить и в более компактной форме, но только с применением алгоритмических языков программирования высокого уровня (Паскаль, С + +, VBA и т.д.). Хотя наглядность программной реализации Алгоритма - 2 при этом, по-видимому, была бы ниже. Тем не менее, приведенная программа допускает существенную модификацию при используемых здесь средствах программирования.

В связи с этим напомним, что в подразделе 1.4 при рассмотрении применения метода Рунге – Кутта для системы, состоящей из двух дифференциальных уравнений (1.21), а именно:

$$\frac{dy}{dx} = f(x, y, z), \quad \frac{dz}{dx} = g(x, y, z),$$

был отмечен случай, когда второе из этих уравнений имеет вид $\frac{dz}{dx} = 1$. Так как это уравнение элементарно интегрируется: $z(x) = x - x_0$, то это можно рассматривать как замену независимой переменной $x = z$. Поэтому для рассматриваемой системы (2.1) также можно ввести соответствующую замену независимой переменной: $x = y^{n+1}$. Тогда задача Коши (2.1), (2.2) на интервале $[x_0, x_f]$ $(x_0 < x_f < \infty)$ будет переформулирована для следующей дифференциальной системы:

$$\frac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(\boldsymbol{y}, y^{n+1}); \quad \frac{dy^{n+1}}{dx} = 1, \tag{2.16}$$

с начальными условиями

$$\boldsymbol{y}(x_0) = \boldsymbol{y}_0, \; y^{n+1}(x_0) = x_0. \tag{2.17}$$

Таким образом, правые части системы (2.16) не содержат независимой переменной, то есть она становится автономной. Но тогда, в соответствии с общими формулами Рунге – Кутта (1.33), не требуется явное вычисление приращений для независимой переменной: $x = x_0 + \alpha_j h$, для $j = 2, 3, 4$ (см. оператор № 26 в программе RUNGE 4.2 на рис.1).

Соответствующий вариант программы RUNGE 4.2 (без описания Алгоритма -3, но с использованием тех же обозначений) приведен на рис.2. При этом предполагается, что N – порядок системы (2.16), то есть он равен $n+1$, и здесь последняя переменная предполагается как преобразованная независимая переменная. В подпрограмме вычисления правых частей CALCPR (Z, PR) входной параметр T для независимой переменной также исключается.

№ п/п	**метка**	**оператор**
Comment		**Шаг 1**
01		DO 10 I = 1, N
02		Z0 (I) = Y0 (I)
03		Z (I) = Y0 (I)
04	10	DY (I) = 0.
05		STEP = 0
Comment		**Шаг 2**
06	20	STEP = STEP + 1
Comment		**Шаг 3**
07		**IF** (Z0 (N) + HINT > XFIN) **THEN** HINT = XFIN – Z0 (N)
Comment		**Шаг 4 и 5** (начало основного цикла по j)
08	25	**DO** 90 J = 1, 2, 3, 4
Comment		**Шаг 6**
09		**CALL** CALCPR (Z, PR)
10		**DO** 30 I = 1, N
11		K (J, I) = HINT ∗ PR (I)
12	30	DY (I) = DY (I) + P4 (J) ∗ K (J, I)
Comment		**Шаг 7**
13		**IF** (J < 4) **THEN** **GO TO** 60
Comment		**Шаг 8**
14		**DO** 40 I = 1, N
15	40	YI (I) = Z0 (I) + DY (I)
Comment		**Шаг 9**
16		**DO** 50 I = 1, N
17		Z0 (I) = YI (I)
18	50	Z (I) = Z0 (I)
Comment		**Шаг 10**
19		**IF** (Z0 (N) ≥ XFIN) **THEN** **GO TO** 100
Comment		**Шаг 11**
20		**GO TO** 20
Comment		**Шаг 12**
21	60	**CONTINUE**
22		**DO** 80 I = 1, N
23		SUM = 0.
24		**DO** 70 L = 1, J
25	70	SUM = SUM + BETTA (J + 1, L) ∗ K (L, I)
26	80	Z (I) = Z0 (I) + SUM
Comment		Конец основного цикла по j
27	90	**CONTINUE**
28	100	**CONTINUE**

Рис. 2. Текст программы RUNGE 4 .3 (Алгоритм -3)

2.3. О программной реализации численного решения задачи Коши на интервале с нефиксированным правым концом

В отличие от рассмотренного в подразделе 2.2 варианта численного решения задачи Коши (2.1), (2.2) на каком-либо заданном интервале $[x_0, x_f]$ (при условии, что $x_f > x_0$), здесь будет рассматриваться вариант ее решения, в котором не предполагается явного задания правой границы интервала $[x_0, x_f]$, то есть значение x_f не фиксировано. Такая постановка задачи Коши возникает при решении широкого круга прикладных задач, в которых значение x_f определяется исходя из каких-либо дополнительных условий «останова», например в виде условий (2.6), процедуры численного решения дифференциального уравнения (2.1). В связи с этим представляет интерес соответствующая модификация программы RUNGE 4.3 (рис.2). Однако, вначале необходимо обсудить подходы к построению алгоритма определения значения x_f.

2.3.1. Поскольку значение x_f в рассматриваемом здесь варианте решения задачи Коши определяется при выполнении хотя бы одного из условий (2.6): $ш_k(x_f, \boldsymbol{y}(x_f)) = 0$ $(k = 1, 2, \ldots, l)$ с заданной точностью $\varepsilon_k > 0$, то есть с учетом (2.7): $|ш_k(\widetilde{x}_f, \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f))| \le \varepsilon_k$, то доработка программы RANGE 4.3 требуется только в той ее части, которая непосредственно связана с определением условий выхода из нее или, что то же самое, условий завершения решения задачи Коши. Поэтому в первую очередь следует дать описание дополнительных исходных данных, которые будут необходимы для соответствующей модификации программы. Во-первых, это вспомогательная программа (подпрограмма, процедура), с помощью которой будут вычисляться значения функций $ш_k(x, \boldsymbol{y})$, $k = 1, 2, \ldots, l$, для заданных x и $\boldsymbol{y}$. Эта программа, как и программа для вычисления правых частей уравнения (2.1), – вызываемая программа, предназначенная для вычисления значений функций $ш_k(x, \boldsymbol{y})$, $k = 1, 2, \ldots, l$. Поскольку они по существу являются условиями «останова» решения рассматриваемой задачи, далее их можно называть *функциями выхода*, а программу их вычисления – подпрограммой вычисления функций выхода.

В частности, следует отметить, что одну из функций выхода можно задавать, например, в виде $ш_1(x, \boldsymbol{y}) = x_f^* - x$, где x_f^* – желаемое значение для x_f (значение x_f^* может вводиться и с целью временного ограничения решения задачи). В том случае, когда $l = 1$ и функция $ш_1(x, \boldsymbol{y})$ является единственной функцией выхода, рассматриваемый вариант численного решения задачи Коши фактически сводится к фиксированию правой границы интервала $[x_0, x_f]$ с точностью, задаваемой константой точности

$\varepsilon_1 \geq 0$. В общем случае, когда $l > 1$, одну из функций выхода всегда можно задавать в виде: $ш_1(x, \boldsymbol{y}) = x_f^* - x$, и в случае, если введена замена: $x = y^{n+1}$, то эту функцию можно записать так: $ш_1(\boldsymbol{y}) = x_f^* - y^{n+1}$.

Помимо функций выхода и их числа l, дополнительными исходными данными также будут и значения точностей для них, то есть константы $\varepsilon_k > 0$, $k = 1, 2, \ldots, l$.

Далее перейдем к описанию алгоритма, который реализует «останов» программы. С учетом условий (2.7) алгоритм состоит в том, чтобы в начале каждого шага интегрирования, быть может за исключением первого, осуществлять: во-первых, проверку выполнения условий (2.7), поскольку выполнение хотя бы одного из них означает завершение решения задачи, и, в частности, получения приближенного значения $\widetilde{x}_f$; во-вторых, проверку функций выхода на смену знака. Например, пусть на начало $m+1$-го шага, то есть по завершении m-го шага, хотя бы для одной из функций выхода $ш_k$ ($1 \leq k \leq l$) будет выполнено условие

$$ш_k(x_{m-1}, \boldsymbol{y}_{m-1}) \cdot ш_k(x_m, \boldsymbol{y}_m) < 0. \qquad (2.18)$$

Но тогда это означает, что k-е условие в (2.6) выполняется для некоторой точки $x = \widetilde{x}_f$, то есть имеет место:

$$ш_k(\widetilde{x}_f, \widetilde{\boldsymbol{y}}_m(\widetilde{x}_f)) = 0, \quad x_{m-1} < \widetilde{x}_f < x_m,$$

где $\boldsymbol{y}_{m-1}$ – начальное условие на подынтервале $[x_{m-1}, x_m]$, а $\boldsymbol{y}_m = \widetilde{\boldsymbol{y}}_m(x_m)$ – полученное приближенное решение. Начальное приближение для точки $\widetilde{x}_f$ из (2.18) можно вычислять по формуле (2.8).

В общем случае, например, на том же m-м шаге, условия (2.18) могут быть выполнены для нескольких функций выхода $ш_{k_q}$, где $q = 1, 2, \ldots, \nu$ ($1 \leq \nu \leq l$), а $k_1 < k_2 < \ldots < k_\nu$. В связи с этим здесь возможны два варианта определения приближенного значения $\widetilde{x}_f$. В первом варианте, можно предполагать лексикографическое упорядочение функций выхода по их номерам, определять начальное приближение $\widetilde{x}_f$ по формуле (2.8) только для функции $ш_{k_1}$. Во втором – можно определить $\widetilde{x}_f$ для всех $ш_{k_q}$, $q = 1, 2, \ldots, \nu$, а затем выбрать тот номер k_q, для которого либо $\widetilde{x}_f$, либо $h_1 < h$ являются минимальными.

В любом случае повторные итерации для $\lambda = 1, 2, 3, \ldots$ на подынтервалах $[x_{m-1}, x_{m-1} + h_\lambda]$ позволяют получить требуемое решение, если только постоянные $\varepsilon_{k_q} > 0$ достаточно малы, хотя при этом «активные» функции могут быть различными (меняться от итерации к итерации).

В заключение обсуждения алгоритма «останова», в связи с возможностью выполнения хотя бы одного условия (2.7), особо следует отметить, что этот алгоритм не исключает тех случаев, когда какое-либо из условий (2.6) все-таки может выполняться внутри подынтервала $[x_{m-1}, x_m]$, при том, что ни одно из условие (2.18) не выполняется, то есть имеет место:

$$ш_k(x_{m-1}, \boldsymbol{y}_{m-1}) \cdot ш_k(x_m, \boldsymbol{y}_m) > 0.$$

Для исключения таких ситуаций можно рекомендовать выбор меньшего шага интегрирования или организацию дополнительной проверки внутри подынтервала с использованием аналитических выражений для $\tilde{\boldsymbol{y}}_i(x)$ вида (2.3). Дополнительно отметим, что выполнение одного из условий (2.7) может быть также вызвано тем, что точности $\varepsilon_k > 0$ для функций выхода, которые не меняют знака на $[x_{m-1}, x_m]$, заданы слишком грубо. Очевидно, что и в этом случае необходимо дополнительное исследование поведения функций выхода, в том числе с использованием выражений (2.3).

Переходя собственно к описанию алгоритма «останова», укажем, что в программе RANGE 4.3 его аналог реализуется оператором № 07, что соответствует шагу № 3 Алгоритма -3. Кроме того, для работы алгоритма «останова» (далее – Алгоритм-4) перед началом первого шага следует вычислить значения всех функций выхода $\psi_{k0} = ш_k(\boldsymbol{y}_0)$.

Итак, в соответствии с изложенным Алгоритм - 4, включающий в себя процедуру повторных итераций для обеспечения выполнения условий (2.6) с требуемой точностью, будет иметь следующий вид.

Шаг 1: (см. описание Алгоритма -2 на стр.27);
iter = 0 – обнуление счетчика итераций.

Шаг 2: вычисление $\tilde{\psi}_k = ш_{k0} = ш_k(\boldsymbol{y}_0)$, $k = 1, 2, \ldots, l$.

Шаг 3: $step = step + 1$ ($i = step$, номер текущего шага интегрирования).

Шаг 4: $j = j + 1$ (счетчик подшагов текущего шага интегрирования и начало основного цикла).

Шаг 5: если $j > 1$ или $i = 1$, то переход к шагу 8.

Шаг 6: вычисление значений функций выхода $ш_k(\boldsymbol{y}_{i-1})$, $k = 1, 2, \ldots, l$, на начало i-го шага.

Шаг 7: проверка выполнения условий $|ш_k(\boldsymbol{y}_{i-1})| \le \varepsilon_k$, $k = 1, 2, \ldots, l$;
если хотя бы одно из условий выполняется (первое в порядке возрастания номеров *k*), то выход из программы.

Шаг 8: проверка выполнения условий (2.18); если нет ни одного условия, которое выполняется, то переход к шагу 10.

Шаг 9: корректировка шага интегрирования по формуле (2.9);
iter = *iter* +1 (счетчик итераций); переход к шагу 3.

Шаг 10: сохранение значений функций выхода для шага 8 ($ш_k(\boldsymbol{y}_{i-1}) \to \tilde{ш}_k$).

Шаг 11: вычисление правых частей (вызов программы CALCPR).

Шаг 12: выполнение шагов №№ 6 –12 программы RUNGE 4.3.

2.3.2. Для описания программной реализации численного решения задачи Коши – в виде соответствующей модификации программы RANGE 4.3 с учетом выше изложенного в подразделе 2.3.1 алгоритма «останова» – необходимо в дополнение к ранее введенным описаниям переменных (см. таблицу 1) ввести следующие идентификаторы и переменные:

- массив $\mathrm{EPS}(1:\mathrm{M})$ для чисел $\varepsilon_k > 0$, $k = 1, 2, \dots, m$, где M – идентификатор для числа используемых функций выхода;
- массив $\mathrm{FV}(1:\mathrm{M})$ для текущих значений функций выхода на начало текущего шага интегрирования ($\mathrm{FV}(k) = \psi_k, k = 1, 2, \dots, m$);
- массив $\mathrm{FV1}(1:\mathrm{M})$, в котором хранятся значения функций выхода на начало предыдущего шага интегрирования;
- ITER идентификатор переменной *iter* (счетчик итераций).

Для вычисления текущих значений функций выхода используется вспомогательная программа (подпрограмма, процедура) $\mathrm{CALCFV(Y, FV)}$, где $\mathrm{Y}(1:\mathrm{N})$ – входные параметры, а $\mathrm{FV}(1:\mathrm{M})$ – выходной параметр этой программы и массив.

Текст программы RUNGE 4.4 (Алгоритм-4) приведен на рис. 3.

```
№ п/п  метка    оператор
       Comment  Шаг 1
  01            DO  10  I = 1, N
  02            Z0 (I) = Y0 (I)
  03            Z (I) = Y0 (I)
  04     10     DY (I) = 0.
  05            STEP = 0
                ITER = 0
       Comment  Шаг 2
  06            CALL  CALCFV (Z, FV1)
       Comment  Шаг 3
  07     20     STEP = STEP + 1
       Comment  Шаг 4 (начало основного цикла по j)
  08     25     DO  90  J = 1, 2, 3, 4
       Comment  Шаг 5
  09            IF (J > 1 .ИЛИ. STEP = 1)  GO TO 300
```

Рис. 3. Текст программы RUNGE 4.4 (Алгоритм-4)
(продолжение на стр.36)

№ п/п	метка	оператор
	Comment	**Шаг 6**
		CALL CALCFV (Z, FV1)
	Comment	**Шаг 7** (проверка условий (2.7))
10		**DO** 26 I2 = 1, M
11		**IF** (ABS (FV (I2) ≤ EPS (I2)) **THEN GO TO** 100
12	26	**CONTINUE**
	Comment	**Шаг 8** (проверка условий (2.18))
13		**DO** 27 I2 = 1, M
		M1 = I2
14		**IF** (FV (I2) ∗ FV1 (I2) > 0) **THEN GO TO** 27
	Comment	**Шаг 9**
		HINT = HINT ∗ FV1 (M1) / (FV1 (M1) – FV (M1)
		ITER = ITER + 1
		GO TO 20
	27	**CONTINUE**
	Comment	**Шаг 10**
		DO 28 I2 = 1, M
	28	FV1 (I2) = FV (I2)
	Comment	**Шаг 11**
09	300	**CALL** CALCPR (Z, PR)
10		**DO** 30 I = 1, N
11		K (J, I) = HINT ∗ PR (I)
12	30	DY (I) = DY (I) + P4 (J) ∗ K (J, I)
	Comment	**Шаг 7**
13		**IF** (J < 4) **THEN GO TO** 60
	Comment	**Шаг 8**
14		**DO** 40 I = 1, N
15	40	YI (I) = Z0 (I) + DY (I)
	Comment	**Шаг 9**
16		**DO** 50 I = 1, N
17		Z0 (I) = YI (I)
18	50	Z (I) = Z0 (I)
	Comment	**Шаг 10**
19		**IF** (T0 ≥ XFIN) **THEN GO TO** 100
	Comment	**Шаг 11**
20		**GO TO** 20
	Comment	**Шаг 12**
21	60	**CONTINUE**
22		**DO** 80 I = 1, N
23		SUM = 0.
24		**DO** 70 L = 1, J
25	70	SUM = SUM + BETTA (J + 1, L) ∗ K (L, I)
26	80	Z (I) = Z0 (I) + SUM
	Comment	Конец основного цикла по j
27	90	**CONTINUE**
28	100	**CONTINUE**

Рис. 3. Текст программы RUNGE 4 .4 (Алгоритм -4)
(начало на стр.35)

2.4. О выборе шага интегрирования при численном решении обыкновенных дифференциальных уравнений и их систем. Правило Рунге

2.4.1. При численном решении задач Коши (1.1), (1.2) или (2.1), (2.2) обычно требуется найти их приближенное решение с некоторой заданной точностью, которая определяется условиями решаемой прикладной задачи. В вычислительной математике [1-5] построение соответствующих алгоритмов, как правило, сопровождается оценками порядка точности на один шаг интегрирования (локальная погрешность). Для формул Рунге – Кутта такая погрешность пропорциональна величине выбираемого шага, так как функция ошибки для формулы r-й степени $\varphi_r(h)$ на шаге имеет главный член, то есть для нее справедливо представление

$$\varphi_r(h) = w(x, y)h^{s+1} + o(h^{s+2}),$$

где s – порядок точности ($s \le r$) формулы, а $w(x, y)$ – некоторая функция, определенная в области (здесь для первого шага): $x_0 \le x \le x_0 + h$; $|y| < \infty$. Согласно формуле Тейлора, функцию $\varphi_r(h)$ также можно записать в виде

$$\varphi_r(h) = \frac{\varphi_r^{(s+1)}(0)}{(s+1)!}h^{s+1} + \frac{\varphi_r^{(s+2)}(\lambda h)}{(s+2)!}h^{s+2}, \quad 0 < \lambda < 1,$$

то есть главный член погрешности на одном шаге интегрирования будет равен $\frac{\varphi_r^{(s+1)}(0)}{(s+1)!}h^{s+1}$. Таким образом, если $y(x)$ – точное решение задачи (1.1), (1.2) на подынтервале $[x_0, x_0 + h]$, а $\tilde{y}_{(h)}$ – приближенное решение, получаемое по соответствующей формуле Рунге – Кутта с шагом h, то имеем

$$\tilde{y}_{(h)}(x_0 + h) - y(x_0 + h) \approx \frac{\varphi_r^{(s+1)}(0)}{(s+1)!}h^{s+1}.$$

Очевидно, что в силу предполагаемой малости h погрешность той же формулы на следующем шаге интегрирования будет иметь тот же главный член, то есть в результате (после двух шагов) получим

$$\tilde{y}_{(h)}(x_0 + 2h) - y(x_0 + 2h) \approx 2\frac{\varphi_r^{(s+1)}(0)}{(s+1)!}h^{s+1}.$$

Если же применить все ту же формулу Рунге – Кутта, но с шагом $2h$, то, обозначая через $\tilde{y}_{(2h)}$ соответствующее приближенное решение, получим

$$\tilde{y}_{(2h)}(x_0+2h)-y(x_0+2h)\approx\frac{\varphi_r^{(s+1)}(0)}{(s+1)!}(2h)^{s+1}.$$

Сравнивая полученные выражения для $\tilde{y}_{(h)}(x_0+2h)$ и $\tilde{y}_{(2h)}(x_0+2h)$, представление для главного члена погрешности на шаге можно записать в следующем виде:

$$\tilde{y}_{(h)}(x_0+2h)-y(x_0+2h)\approx\frac{\tilde{y}_{(2h)}(x_0+2h)-\tilde{y}_{(h)}(x_0+2h)}{2^s-1}. \quad (2.19)$$

Таким образом, при достаточно малом шаге интегрирования [2] (а также малой вычислительной погрешности) приближенное решение задачи Коши (1.1), (1.2), получаемое методом Рунге – Кутта (точнее, по соответствующей формуле этого метода), будет близко к точному решению этой же задачи. То же самое справедливо и в общем случае для задачи Коши (2.1), (2.2).

Формально, при $h\to 0$ погрешность метода Рунге – Кутта также будет стремиться к нулю, но в действительности величина шага интегрирования должна быть ограничена и снизу $h>h_{min}$, так как реальная погрешность может существенно отличаться от главного члена в силу различного рода причин [2], которые обычно относят к так называемым вычислительным погрешностям. Очевидно, что $h_{min}<h_{max}$, где h_{max} – максимально допустимый шаг интегрирования для конкретной системы (1.1) или (2.1), который в общем случае должен быть существенно меньше характерного изменения решения этой системы. Последнее характеризуется либо предельными значениями производной $\dfrac{f(x,y)}{\partial y}$ для правой части (1.1) на интервале интегрирования – $[x_0,x_f]$, либо предельными значениями максимального (по модулю) собственного числа матрицы Якоби $\dfrac{\boldsymbol{f}(x,\boldsymbol{y})}{\partial \boldsymbol{y}}$ для уравнения (2.1). В том случае, когда на интервале интегрирования $[x_0,x_f]$ указанные характеристики изменяются существенно, применяются различные процедуры автоматического выбора шага интегрирования. С этой целью в них используют аппроксимации для главного члена погрешности на шаге, называемые *контрольными членами*.

На практике в процедуре выбора шага интегрирования используется следующий прием (иногда называемый экстраполяцией по Ричардсону). Как и выше при определении главного члена погрешности в методе Рунге – Кутта, вычисляется значение $\delta=|\tilde{y}_{(2h)}(x_0+2h)-\tilde{y}_{(h)}(x_0+2h)|$. Для ε_0 и ε_1 – заданных мер погрешности (обычно, $\varepsilon_1<\varepsilon_0$ и $\varepsilon_0/\varepsilon_1\approx 2^s$, где $s \le r$ – порядок формулы Рунге – Кутта) анализируются следующие неравенства. Во-первых, если $\delta > \varepsilon_0$, то шаг интегрирования уменьшают (обычно, вдвое); во-вторых, если $\delta < \varepsilon_1$, то увеличивают (также вдвое), и, наконец, если окажется, что выполняются неравенства $\varepsilon_1 \le \delta \le \varepsilon_0$, то шаг интегрирования не меняется.

2.4.2. Для примера автоматического выбора шага интегрирования, приведем соотношения метода Кутта – Мерсона [3, Т.3, С.158], известного как пятиэтапный метод Рунге – Кутта 4-го порядка точности:

$$k_1 = hf(x_0, y_0); \quad k_2 = hf(x_0 + \frac{1}{3}h, y_0 + \frac{1}{3}k_1);$$

$$k_3 = hf(x_0 + \frac{1}{3}h, y_0 + \frac{1}{6}k_1 + \frac{1}{6}k_2);$$

$$k_4 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{8}k_1 + \frac{3}{8}k_2);$$

$$k_5 = hf(x_0 + h, y_0 + \frac{1}{2}k_1 - \frac{3}{2}k_2 + 2k_4).$$

Соответственно, приближенное решение вычисляется по формуле

$$y^1(x_0 + h) = y_0 + \frac{1}{6}k_1 + \frac{2}{3}k_4 + \frac{1}{6}k_5.$$

Кроме того, дополнительно вычисляется вспомогательная величина

$$y^2(x_0 + h) = y_0 + \frac{1}{2}k_1 - \frac{3}{2}k_3 + 2k_4,$$

с помощью которой оценивается погрешность метода на одном шаге таким выражением: $\delta = 0{,}2|\, y^1(x_0 + h) - y^2(x_0 + h)|$. Эта величина также служит для автоматического выбора шага. Если ε_* – заданная точность вычислений, то при $\delta > \varepsilon_*$ шаг интегрирования уменьшается в два раза. При $\frac{1}{64}\varepsilon_* > \delta$ шаг увеличивается вдвое, а при $\frac{1}{64}\varepsilon_* < \delta \le \varepsilon_*$ – шаг интегрирования не меняется. Этот способ оценки погрешности и выбора шага интегрирования называется методом вложенных форм, поскольку формула для $y^2(x_0 + h)$ является «вложенной» в формулу вычисления $y^1(x_0 + h)$, а именно:

$$y^1(x_0 + h) = y_0 + \frac{1}{6}k_1 + \frac{2}{3}k_4 + \frac{1}{6}hf(x_0 + h, y^2(x_0 + h)).$$

2.4.3. Во многих прикладных задачах решение задачи Коши получают на больших промежутках. Поэтому погрешность используемого метода численного решения дифференциальных уравнений на одном шаге интегрирования как его точностная характеристика, вообще говоря, не является достаточной для обоснования целесообразности применения этого метода. Более эффективной («достаточной») его точностной характеристикой как для заданного уравнения (1.1), так и для интервала $[x_0, x_f]$, на котором решается задача Коши, будет следующая величина: $Q_{max} = \max\limits_{1 \le k \le i_f} |Q_k|$. Здесь Q_k вычисляются так: $Q_k = \tilde{y}_k(x_k) - y(x_k)$, $k = 1, 2, 3, \ldots$, где $y(x)$ – решение задачи (1.1), (1.2), а $\tilde{y}_k(x)$ – ее решения, которые удовлетворяют условиям: $\tilde{y}_k(x_k) = y_k$, где y_k – значения, получаемые по формуле численного интегрирования на подынтервалах $[x_{k-1}, x_k]$. Соответственно, можно ввести в рассмотрение и решение $\tilde{y}_0(x)$, которое можно рассматривать как решение задачи Коши (1.1), (1.2) для возмущенного начального условия.

Погрешность Q_k можно представить в виде

$$Q_k = \tilde{y}_k(x_k) - y(x_k) = \tilde{y}_k(x_k) - \tilde{y}_0(x_k) + \tilde{y}_0(x_k) - y(x_k) =$$

$$= \sum_{j=1}^{k} [\tilde{y}_j(x_k) - \tilde{y}_{j-1}(x_k)] + [\tilde{y}_0(x_k) - y(x_k)]. \qquad (2.20)$$

В [2] показано, что разности решений одного и того же дифференциального уравнения в одной точке могут быть выражены через их разности в другой точке, например:

$$Y_2(\beta) - Y_2(\beta) = \left(Y_2(\alpha) - Y_2(\alpha)\right) exp\left\{ \int_{\alpha}^{\beta} f_y(x, \bar{y}(x))\, dx \right\},$$

где $\bar{y}(x)$ заключено между $Y_2(x)$ и $Y_1(x)$. Если здесь принять $\alpha = x_{j-1}$, $\beta = x_j$, $Y_1(x) = y_{j-1}(x)$ и $Y_2(x) = y_j(x)$, а также ввести $\bar{y}_j(x)$, которое заключено между $y_{j-1}(x)$ и $y_j(x)$, то получим

$$y_j(x_k) - y_{j-1}(x_k) = \left(y_j(x_j) - y_{j-1}(x_j)\right) exp\left\{ \int_{x_j}^{x_k} f_y(x, \bar{y}_j(x))\, dx \right\}.$$

Точно так же можно получить выражение для $\tilde{y}_0(x_k) - y(x_k)$. Таким образом можно построить оценки для всех Q_k (2.20). Например, в [2] приведены результаты соответствующего анализа оценок для величины Q_{i_f}, где i_f – число шагов на интервале $[x_0, x_f]$. Получаемые при этом оценки – априорные, поэтому они не могут быть достаточными.

По-видимому, наиболее эффективными точностными характеристиками методов интегрирования являются апостериорные оценки. В общем случае при их построении сравниваются получаемые решения задачи Коши (1.1), (1.2) или (2.1), (2.2) на сетках с разными шагами. Обычно удобно задавать эти сетки так: одну, например, с шагом h (выбранным или исследуемым), а другую – с шагом $h/2$, или как в случае построения представления для главного члена в виде (2.19), то есть с шагами h и $2h$. В основе такого подхода лежит *правило Рунге* [1-5] – метод оценки погрешности формул численного интегрирования (при вычислении определенных интегралов). Существо указанного правила заключается в выделении главного члена погрешности по результатам расчетов с двумя различными шагами (не обязательно кратными в общем случае). При этом не используется явное выражение для главного члена погрешности, а предполагается лишь факт его существования. Правило Рунге используется и при численном решении дифференциальных уравнений в виде схем Ричардсона и Филиппова [2].

Библиографический список

1. Березин, И.С. Методы вычислений: Т.2 / И.С. Березин, Н.П. Жидков. – М.: ГИФМЛ, 1960. – 620 с.
2. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – 4-е изд. – М.: БИНОМ. Лаборатория знаний, 2006. – 636 с.
3. Математическая энциклопедия: в 5 т. – М.: Советская энциклопедия, 1984.
4. Копченова, Н.В. Вычислительная математика в примерах и задачах / Н.В. Копченова, И.А. Марон. – М.: Наука, 1972. – 368 с.
5. Расчет и анализ движения летательных аппаратов. Инженерный справочник / С.А. Горбатенко [и др.]. – М.: Машиностроение, 1971. – 352 с.
6. Милн, В.Э. Численное решение дифференциальных уравнений / В.Э. Милн. – М.;Л.: Иностр. лит., Л, 1955. – 292 с.
7. Бабушка, И. Численные процессы решения дифференциальных уравнений / И. Бабушка, Э. Витасек, М. Прагер. – М.: Мир, 1969. – 368 с.
8. Ортега, Дж. Введение в численные методы решения дифференциальных уравнений / Дж. Ортега, У. Пул. – М.: Наука, 1986. – 288 с.
9. Современные методы решения обыкновенных дифференциальных уравнений / под ред. Дж. Хола и Дж. Уайта. – М.: Мир, 1979. – 312 с.

Приложение

Формулы метода Рунге-Кутта
(до четвертого порядка точности)

Таблица 2

Номер/ порядок формулы	Название метода (формулы)	Формула для вычисления $\Delta \boldsymbol{y}_m = \boldsymbol{y}_{m+1} - \boldsymbol{y}_m$	Формулы для вычисления коэффициентов $\boldsymbol{k}_i$
1	2	3	4
1/1	Метод Эйлера, метод ломаных	$\Delta \boldsymbol{y}_m = \boldsymbol{k}_1$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$
2/2	Улучшенный метод Эйлера	$\Delta \boldsymbol{y}_m = \frac{1}{2}(\boldsymbol{k}_1 + \boldsymbol{k}_2)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \boldsymbol{k}_1)$
3/2	Метод Эйлера с пересчетом (модифицированный)	$\Delta \boldsymbol{y}_m = \boldsymbol{k}_2$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$
4/2		$\Delta \boldsymbol{y}_m = \frac{1}{4}(\boldsymbol{k}_1 + 3\boldsymbol{k}_2)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{2}{3}h, \boldsymbol{y}_m + \frac{2}{3}\boldsymbol{k}_1)$
5/3		$\Delta \boldsymbol{y}_m = \frac{1}{6}(\boldsymbol{k}_1 + 4\boldsymbol{k}_2 + \boldsymbol{k}_3)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m - \boldsymbol{k}_1 + 2\boldsymbol{k}_2)$

Продолжение таблицы 2

1	2	3	4
6/3	Метод Рунге – Кутта-Гейне	$\Delta \boldsymbol{y}_m = \frac{1}{4}(\boldsymbol{k}_1 + 3\boldsymbol{k}_3)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{3}h, \boldsymbol{y}_m + \frac{1}{3}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{2}{3}h, \boldsymbol{y}_m + \frac{2}{3}\boldsymbol{k}_2)$
7/3		$\Delta \boldsymbol{y}_m = \frac{1}{9}(2\boldsymbol{k}_1 + 3\boldsymbol{k}_2 + 4\boldsymbol{k}_3)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{3}{4}h, \boldsymbol{y}_m + \frac{3}{4}\boldsymbol{k}_2)$
8/4	Стандартный метод Рунге – Кутта (правило «1/6»)	$\Delta \boldsymbol{y}_m = \frac{1}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)$, контрольный член: $\boldsymbol{E} = \boldsymbol{k}_1 - \boldsymbol{k}_2 - \boldsymbol{k}_3 + \boldsymbol{k}_4$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \boldsymbol{k}_3)$
9/4	Метод Рунге – Кутта (правило «3/8»)	$\Delta \boldsymbol{y}_m = \frac{1}{8}(\boldsymbol{k}_1 + 3\boldsymbol{k}_2 + 3\boldsymbol{k}_3 + \boldsymbol{k}_4)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{3}h, \boldsymbol{y}_m + \frac{1}{3}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{2}{3}h, \boldsymbol{y}_m - \frac{1}{3}\boldsymbol{k}_1 + \boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \boldsymbol{k}_1 - \boldsymbol{k}_2 + \boldsymbol{k}_3)$

Продолжение таблицы 2

1	2	3	4
10/4		$\Delta \boldsymbol{y}_m = \frac{1}{8}(\boldsymbol{k}_1 + 3\boldsymbol{k}_2 + 3\boldsymbol{k}_3 + \boldsymbol{k}_4)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{4}h, \boldsymbol{y}_m + \frac{1}{4}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \boldsymbol{k}_1 - 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3)$
11/4	Первый метод Гилла	$\Delta \boldsymbol{y}_m = \frac{1}{6}(\boldsymbol{k}_1 + 2\frac{\sqrt{2}-1}{\sqrt{2}}\boldsymbol{k}_2 +$ $+ 2\frac{\sqrt{2}+1}{\sqrt{2}}\boldsymbol{k}_3 + \boldsymbol{k}_4)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{\sqrt{2}-1}{\sqrt{2}}\boldsymbol{k}_1 +$ $+ \frac{\sqrt{2}-1}{\sqrt{2}}\boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m - \frac{1}{\sqrt{2}}\boldsymbol{k}_2 + \frac{\sqrt{2}+1}{\sqrt{2}}\boldsymbol{k}_3)$
12/4	Второй метод Гилла	$\Delta \boldsymbol{y}_m = \frac{1}{6}(\boldsymbol{k}_1 + 3\boldsymbol{k}_2 + \boldsymbol{k}_3 + \boldsymbol{k}_4)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m - \frac{1}{2}\boldsymbol{k}_1 + \boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_2 + \frac{1}{2}\boldsymbol{k}_3)$

Продолжение таблицы 2

1	2	3	4
13/4	Метод Рунге – Кутта – Мерсона	$\Delta \boldsymbol{y}_m = \frac{1}{6}(\boldsymbol{k}_1 + 4\boldsymbol{k}_4 + \boldsymbol{k}_5)$, контрольный член Мерсона $o(h^5)$: $\boldsymbol{E} = \frac{1}{5}\left(\boldsymbol{k}_1 - \frac{9}{2}\boldsymbol{k}_3 + 4\boldsymbol{k}_4 + \frac{1}{2}\boldsymbol{k}_5\right)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{3}h, \boldsymbol{y}_m + \boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{1}{3}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1 + \frac{1}{2}\boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{3}{8}\boldsymbol{k}_1 + \frac{9}{8}\boldsymbol{k}_3)$; $\boldsymbol{k}_5 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m + \frac{3}{2}\boldsymbol{k}_1 - \frac{9}{2}\boldsymbol{k}_3 + 6\boldsymbol{k}_4)$
14/4	Метод Рунге – Кутта – Ингланда	$\Delta \boldsymbol{y}_m = \frac{1}{6}(\boldsymbol{k}_1 + 4\boldsymbol{k}_3 + \boldsymbol{k}_4)$, оценка локальной ошибки метода $o(h^5)$: $\boldsymbol{E} = \frac{1}{336}(-42\boldsymbol{k}_1 - 224\boldsymbol{k}_3 - 21\boldsymbol{k}_4 + 162\boldsymbol{k}_5 + 125\boldsymbol{k}_6)$	$\boldsymbol{k}_1 = h\boldsymbol{f}(x_m, \boldsymbol{y}_m)$; $\boldsymbol{k}_2 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{2}\boldsymbol{k}_1)$; $\boldsymbol{k}_3 = h\boldsymbol{f}(x_m + \frac{1}{2}h, \boldsymbol{y}_m + \frac{1}{4}\boldsymbol{k}_1 + \frac{1}{4}\boldsymbol{k}_2)$; $\boldsymbol{k}_4 = h\boldsymbol{f}(x_m + h, \boldsymbol{y}_m - \boldsymbol{k}_2 + 2\boldsymbol{k}_3)$; $\boldsymbol{k}_5 = h\boldsymbol{f}(x_m + \frac{28}{27}h, \boldsymbol{y}_m + \frac{7}{27}\boldsymbol{k}_1 + \frac{10}{27}\boldsymbol{k}_2 + \frac{1}{27}\boldsymbol{k}_3)$; $\boldsymbol{k}_6 = h\boldsymbol{f}(x_m + \frac{1}{5}h, \boldsymbol{y}_m + \frac{28}{625}\boldsymbol{k}_1 - \frac{125}{625}\boldsymbol{k}_2 + \frac{546}{625}\boldsymbol{k}_3 + \frac{54}{625}\boldsymbol{k}_4 - \frac{378}{625}\boldsymbol{k}_5)$

Учебное издание

Горелов Юрий Николаевич

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ (МЕТОД РУНГЕ – КУТТА)

Учебное пособие

Редактор Ю.В. Яценко
Компьютерная верстка, макет Н.П. Бариновой

Подписано в печать 19.06.06.Формат $60 \times 90/16$. Бумага офсетная.
Усл. печ. л. 2,8 . уч.-изд. л. 3,0. Гарнитура «Times New Roman».
Тираж 100 экз. Заказ №
Издательство «Самарский университет»
443011, Самара, ул. Акад. Павлова, 1, тел. (846) 334 54 23.
Отпечатано на УОП СамГУ

ДЛЯ ЗАМЕТОК