

УДК 519.95

## ПОСТРОЕНИЕ ЯВНЫХ РАЗНОСТНЫХ СХЕМ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ С ПОМОЩЬЮ РАЗЛОЖЕНИЙ ЛАГРАНЖА–БЮРМАНА

Е. В. Ворожцов<sup>1</sup>

Предложен подход к построению явных многостадийных методов типа Рунге–Кутты для решения обыкновенных дифференциальных уравнений (ОДУ) с помощью разложения сеточных функций в ряды Лагранжа–Бюрмана. Приведены формулы для первых четырех коэффициентов этого разложения. Построены новые явные методы первого и второго порядков точности, которые применены для численного интегрирования задачи Коши для умеренно жесткой системы ОДУ. Оказалось, что  $L_2$ -норма ошибки решения, полученного по новому численному методу второго порядка точности, в 50 раз меньше, чем в случае классического метода Рунге–Кутты второго порядка точности.

**Ключевые слова:** обыкновенные дифференциальные уравнения, разложение Лагранжа–Бюрмана, методы Рунге–Кутты, жесткие системы.

**1. Введение.** Обыкновенные дифференциальные уравнения (ОДУ) имеют важное значение при моделировании многих прикладных задач. Для того чтобы найти решение ОДУ, приходится, как правило, использовать приближенные методы. Конечно-разностные методы получили здесь широкое распространение, однако при использовании этих методов приходится учитывать проблемы точности и устойчивости. Проблемы устойчивости особенно усложняются в случае жестких систем ОДУ, типичных для задач химической кинетики. В [1] было доказано, что явный линейный многостадийный метод не может быть А-устойчивым, поэтому целесообразно строить неявные многостадийные методы, а также неявные методы Рунге–Кутты, которые обладают свойством А-устойчивости; однако эти методы требуют дополнительных итераций [2]. В этой связи в литературе имеются исследования, целью которых является получение явных методов типа Рунге–Кутты с расширенной областью устойчивости [3–7]. В частности, в [3] было впервые предложено строить явные методы Рунге–Кутты второго порядка точности с расширенной областью устойчивости на отрицательной вещественной оси с применением сдвинутого многочлена Чебышева первого рода; эти явные методы были названы методами Рунге–Кутты–Чебышева (РКЧ). Абдулле [4, 5] построил также семейства РКЧ-методов второго и четвертого порядков точности. Обзор РКЧ-методов можно найти в [6, 7]. Недостаток РКЧ-методов состоит в том, что для их реализации требуется большой расход машинного времени. Это связано с тем, что РКЧ-методы требуют выполнения нескольких десятков стадий; например, РКЧ-метод, предложенный в [7], требует до 320 стадий. Таким образом, возникает необходимость в поиске других возможностей для построения явных методов, которые обладают расширенной областью устойчивости и при этом более высокой локальной точностью.

Ниже мы будем рассматривать задачу Коши вида

$$\frac{du}{dx} = f(x, u), \quad x \in [x_0, X], \quad (1)$$

$$u(x_0) = u_0, \quad (2)$$

где  $x$  — независимая переменная,  $f(x, u)$  — заданная скалярная функция,  $x_0, X$  и  $u_0$  — заданные величины и  $x_0 < X$ . Наиболее распространенным подходом при построении разностных методов различных порядков точности для решения уравнения (1) является разложение решения рассматриваемой задачи Коши в усеченный ряд Тейлора. Ряд Тейлора — это ряд по степеням функции  $x - x_0$ . Таким образом, усеченный ряд Тейлора — это некоторый многочлен. Однако из теории аппроксимации функций многочленами известно (см., например, [8]), что такие приближения приводят к осцилляциям интерполирующей функции в окрестности разрывов исходной функции, а также в областях больших градиентов решения.

<sup>1</sup> Институт теоретической и прикладной механики им. С. А. Христиановича СО РАН, ул. Институтская, 4/1, 630090, Новосибирск; ведущий науч. сотр., e-mail: vorozh@itam.nsc.ru

С другой стороны, известно, что функцию можно разложить и в степенной ряд общего вида, т.е. по степеням некоторой функции  $\varphi(x - x_0)$ . Тогда можно подобрать функцию  $\varphi(x)$  так, чтобы уменьшить амплитуду осцилляций численного решения. Задача разложения функции в ряд по степеням другой функции эффективно решается с помощью формулы Лагранжа–Бюрмана [9, 10]. В [11] было впервые предложено применять разложение Лагранжа–Бюрмана для построения разностных схем для численного интегрирования уравнений в частных производных гиперболического типа, в частности уравнений Эйлера, описывающих течения невязкого сжимаемого нетеплопроводного газа. Оказалось, что построенные таким образом разностные схемы хорошо пригодны для численного расчета течений газа с ударными волнами и другими сильными разрывами, несмотря на то что в построенные схемы не вводились ни искусственная вязкость, ни TVD-ограничители.

Для обеспечения требуемого порядка аппроксимации построенных схем во времени в [11] предлагалось использовать явные схемы Рунге–Кутты. Таким образом, разностные схемы, построенные в [11], комбинировали формулу Лагранжа–Бюрмана для аппроксимации частных производных по пространственным переменным с формулой разложения Тейлора для аппроксимации производных по времени. Однако и для аппроксимаций производных по времени тоже можно использовать формулу разложения Лагранжа–Бюрмана. Ниже в настоящей статье строятся разностные схемы для численного решения задачи Коши (1), (2), которые имеют локальные порядки точности  $p = 1, 2$ . В разделе 2 обсуждается формула разложения Лагранжа–Бюрмана. В разделе 3 представлена общая процедура построения явных методов типа Рунге–Кутты конечного порядка точности с помощью разложений сеточных функций в ряды Лагранжа–Бюрмана. В разделе 4 выводится аналог явного метода Эйлера с помощью разложения Лагранжа–Бюрмана. В разделе 5 выводится явный метод типа Рунге–Кутты второго порядка точности и приводятся примеры численного решения умеренно жесткой системы ОДУ с помощью новых явных методов.

**2. Формула разложения Лагранжа–Бюрмана.** Пусть  $u(x)$  — бесконечно дифференцируемая функция и пусть  $y = g(x)$ ,  $y_0 = g(x_0)$ ,  $g'(x_0) \neq 0$ . Тогда формула разложения Лагранжа–Бюрмана записывается в виде [9]

$$u(x) = u(x_0) + \sum_{k=1}^{\infty} \frac{(y - y_0)^k}{k!} \left\{ \frac{d^{k-1}}{dx^{k-1}} \left[ u'(x) \left( \frac{x - x_0}{g(x) - y_0} \right)^k \right] \right\}_{x=x_0}. \tag{3}$$

Очевидно, что в частном случае  $g(x) = x$ ,  $y_0 = x_0$  формула Лагранжа–Бюрмана (3) переходит в формулу Тейлора

$$u(x) = u(x_0) + \sum_{k=1}^{\infty} \left\{ \frac{d^k u(x)}{dx^k} \right\}_{x_0} \frac{(x - x_0)^k}{k!}. \tag{4}$$

Для дальнейшего нам будет удобно переписать формулу (3) в виде, максимально похожем на формулу Тейлора (4). Введем функцию  $\psi(x) = g(x) - y_0$ , так что  $\psi(x_0) = 0$ . Введем также новую переменную  $\tilde{x}$  по формуле  $x = \tilde{x} + x_0$ . Тогда получим, что  $\psi(x) = \psi(\tilde{x} + x_0)$ . Пусть  $\tilde{\psi}(\tilde{x}) = \psi(\tilde{x} + x_0)$ . Введем обозначение  $\varphi(x - x_0) = \tilde{\psi}(\tilde{x}) = \tilde{\psi}(x - x_0)$ . Тогда  $\varphi(x - x_0) = \varphi(0)$  при  $x = x_0$ , и мы должны потребовать, чтобы  $\varphi(0) = 0$ ,  $\varphi'(0) \neq 0$ , т.е. точка  $x = 0$  — нуль первого порядка функции  $\varphi(x)$ . С введением функции  $\varphi(x)$  мы можем переписать формулу Лагранжа–Бюрмана (3) в виде

$$u(x) = u(x_0) + \sum_{k=1}^{\infty} \frac{[\varphi(x - x_0)]^k}{k!} \left\{ \frac{d^{k-1}}{dx^{k-1}} \left[ u'(x) \left( \frac{x - x_0}{\varphi(x - x_0)} \right)^k \right] \right\}_{x=x_0}. \tag{5}$$

Таким образом, формула (5) позволяет разложить функцию  $u(x)$  в ряд по степеням функции  $\varphi(x - x_0)$ .

Обозначим через  $b_k = \frac{1}{k!} \left\{ \frac{d^{k-1}}{dx^{k-1}} \left[ u'(x) \left( \frac{x - x_0}{\varphi(x - x_0)} \right)^k \right] \right\}_{x=x_0}$  коэффициент при  $[\varphi(x - x_0)]^k$  в (5),

$k = 1, 2, \dots$ . В частности,  $b_1 = \left[ u'(x) \frac{x - x_0}{\varphi(x - x_0)} \right]_{x=x_0} = u'(x_0) \frac{0}{0}$ . Видно, что в выражении для  $b_1$  имеется неопределенность вида  $0/0$ . Для раскрытия этой неопределенности воспользуемся следующим представлением коэффициентов  $b_k$  [10]:

$$b_k = \lim_{x \rightarrow x_0} \frac{1}{k!} \left\{ \frac{d^{k-1}}{dx^{k-1}} \left[ u'(x) \left( \frac{x - x_0}{\varphi(x - x_0)} \right)^k \right] \right\}, \quad k = 1, 2, \dots \tag{6}$$

В частности, при  $k = 1$  имеем

$$b_1 = \lim_{x \rightarrow x_0} \left[ u'(x) \frac{x - x_0}{\varphi(x - x_0)} \right] = \left[ \lim_{x \rightarrow x_0} u'(x) \right] \left[ \lim_{x \rightarrow x_0} \frac{x - x_0}{\varphi(x - x_0)} \right] = \frac{u'(x_0)}{\varphi'(0)}. \tag{7}$$

При вычислении второго предела мы воспользовались правилом Лопиталья. Сложность вычисления пределов в выражениях (6) для  $b_k$  нелинейно возрастает с ростом  $k$ . Здесь весьма полезным оказалось применение системы *Mathematica*. Заметим, что *Mathematica*-функция `Limit[...]` не смогла вычислить даже простой предел (7):

```
TraditionalForm[Limit[u'[x]*(x-x0)/φ[x-x0],x→x0]]
```

$$\lim_{x \rightarrow x_0} \frac{(x - x_0)u'(x)}{\varphi(x - x_0)}.$$

Поэтому при реализации вычисления коэффициентов  $b_k$  в системе *Mathematica* мы сначала ввели переменную  $h = x - x_0$ , а затем разложили функцию  $\varphi(x - x_0) = \varphi(h)$  в отрезок ряда Тейлора относительно точки  $h = 0$ . После этого предел (6) находился просто с помощью подстановки  $h = 0$  в получаемые дробно-рациональные выражения. Приведем фрагмент соответствующей программы на языке системы *Mathematica*:

```
m = 4; Phi[x_] := x*Sum[(D[fi[x], {x, k}] /. x -> 0)*x^(k - 1)/k!,
{k, m}]; Do[b = (1/k!)*D[u'[x + x0]*(x/Phi[x])^k, {x, k - 1}];
b = b /. x -> 0; b = Expand[b];
Print["b(", k, ") = ", TraditionalForm[b]], {k, m}];
```

В результате символьных вычислений по этой *Mathematica*-программе для  $b_1$  была получена формула (7), а для  $b_2, b_3, b_4$  — следующие выражения:

$$b_2 = \frac{u''(x_0)}{2\varphi'(0)^2} - \frac{u'(x_0)\varphi''(0)}{2\varphi'(0)^3}, \tag{8}$$

$$b_3 = \frac{u'(x_0)\varphi''(0)^2}{2\varphi'(0)^5} - \frac{u''(x_0)\varphi''(0)}{2\varphi'(0)^4} + \frac{u^{(3)}(x_0)}{6\varphi'(0)^3} - \frac{u'(x_0)\varphi^{(3)}(0)}{6\varphi'(0)^4}, \tag{9}$$

$$b_4 = -\frac{5u'(x_0)\varphi''(0)^3}{8\varphi'(0)^7} + \frac{5u''(x_0)\varphi''(0)^2}{8\varphi'(0)^6} - \frac{u^{(3)}(x_0)\varphi''(0)}{4\varphi'(0)^5} + \frac{5u'(x_0)\varphi^{(3)}(0)\varphi''(0)}{12\varphi'(0)^6} - \frac{u''(x_0)\varphi^{(3)}(0)}{6\varphi'(0)^5} + \frac{u^{(4)}(x_0)}{24\varphi'(0)^4} - \frac{u'(x_0)\varphi^{(4)}(0)}{24\varphi'(0)^5}. \tag{10}$$

Если в формулах (7)–(10) положить в частном случае  $\varphi(x) = x$ , то легко видеть, что полученные выражения для коэффициентов  $b_k$  переходят в коэффициенты при  $(x - x_0)^k$  в формуле Тейлора (4).

**3. Общая процедура явных методов типа Рунге–Кутта.** Эта процедура предназначена для приближенного численного решения задачи Коши (1), (2). Сначала введем в промежутке  $[x_0, X]$  конечное множество точек  $x = x_j, j = 0, 1, \dots, N$ , причем  $x_0 < x_1 < \dots < x_N = X$ . Будем называть точки из множества  $G_h = \{x_0, x_1, \dots, x_N\}$  узлами расчетной сетки, а само множество  $G_h$  — сеткой. В рамках методов Рунге–Кутта приближенные значения решения  $u(x_j)$  находятся последовательно для возрастающих номеров  $j$  начиная с  $j = 0$ . Пусть значение  $u(x_j)$  известно. Тогда значение решения в следующем узле  $x_{j+1}$  вычисляется по формуле

$$u(x_{j+1}) = u(x_j) + \Delta u_{h,j}. \tag{11}$$

Формула для вычисления  $\Delta u_{h,j}$  зависит от количества стадий используемого метода Рунге–Кутта. В общем случае  $(q + 1)$ -стадийного метода сначала последовательно вычисляют величины

$$\begin{aligned} g_0 &= \varphi_j(h_j)f(x, u), \\ g_1 &= \varphi_j(h_j)f(x + \alpha_1 h_j, u + \beta_{10}g_0), \\ g_2 &= \varphi_j(h_j)f(x + \alpha_2 h_j, u + \beta_{20}g_0 + \beta_{21}g_1), \\ &\dots\dots\dots \\ g_q &= \varphi_j(h_j)f(x + \alpha_q h_j, u + \beta_{q0}g_0 + \beta_{q1}g_1 + \dots + \beta_{q,q-1}g_{q-1}), \end{aligned} \tag{12}$$

где  $h_j = x_{j+1} - x_j, j = 0, \dots, N - 1; q \geq 0, \alpha_1, \dots, \alpha_q; \beta_{10}, \beta_{20}, \beta_{21}, \dots, \beta_{q0}, \beta_{q1}, \dots, \beta_{q,q-1}$  — параметры метода Рунге–Кутты. Затем приращение решения  $\Delta u_{h,j}$  вычисляется по формуле

$$\Delta u_{h,j} = \sum_{i=0}^q A_i g_i, \tag{13}$$

где  $A_0, A_1, \dots, A_q$  — параметры метода.

Наряду с приближенной формулой (13) для приращения решения мы можем выписать “точную” формулу для приращения решения  $\Delta u$  с использованием формулы разложения Лагранжа–Бюрмана (5):

$$\Delta u_j = u(x_j + h_j) - u(x_j) = \sum_{k=1}^m \frac{h_j^k}{k!} \left\{ \frac{d^{k-1}}{dx^{k-1}} \left[ u'(x) \left( \frac{x - x_j}{\varphi_j(x - x_j)} \right)^k \right] \right\}_{x=x_j}, \tag{14}$$

где  $m$  — заданное конечное натуральное число,  $m \geq q + 1$ . Теперь составим разность

$$\delta u_j = \Delta u_j - \Delta u_{h,j} \tag{15}$$

и потребуем, чтобы выполнялось соотношение

$$\delta u_j = O(h_j^p), \tag{16}$$

где  $p$  — максимальный возможный показатель для заданного количества стадий  $q$ . Максимизация показателя  $p$  в (16) осуществляется за счет подходящего подбора параметров  $(\alpha), (\beta), (A)$  в (12) и (13).

Заметим, что вид функции  $\varphi_j$ , входящей в (12), может меняться при переходе от  $j$ -го узла к следующему  $(j + 1)$ -му узлу. За счет этого можно добиваться повышения точности и устойчивости метода типа Рунге–Кутты (11)–(13).

В простейшем случае равномерной сетки, когда  $x_{j+1} - x_j = h = \text{const}$  для всех  $j$ , можно также брать одну и ту же функцию  $\varphi(x)$  во всех узлах сетки  $G_h$ . Тогда можно вычислить значение  $\varphi(h)$  один раз, до начала вычислений по методу Рунге–Кутты (11)–(13). Из (12) следует, что предлагаемый метод типа Рунге–Кутты потребует объема вычислений, не превосходящего объема вычислений по классическому методу Рунге–Кутты, для которого  $\varphi(h) = h$ .

**4. Метод первого порядка точности.** При  $m = 1$  в (14) из (12) получаем с учетом (7) формулу

$$u(x_j + h_j) = u(x_j) + \frac{u'(x_j)}{\varphi_j'(0)} \varphi_j(h_j). \tag{17}$$

С учетом (1) можем переписать равенство (17) в виде

$$u(x_j + h_j) = u(x_j) + \frac{\varphi_j(h_j)}{\varphi_j'(0)} f(x_j, u(x_j)). \tag{18}$$

В частном случае  $\varphi_j(x) = x$  разностная схема (18) переходит в известный явный метод Эйлера первого порядка точности [2, 12, 13]:

$$u(x_j + h_j) = u(x_j) + h_j f(x_j, u(x_j)). \tag{19}$$

Поэтому схему (18) можно рассматривать как аналог метода Эйлера (19). В дальнейшем метод (18) будем называть, для краткости, методом LB1 (т.е. это метод первого порядка точности, полученный с применением разложения сеточной функции в ряд Лагранжа–Бюрмана).

Введем обозначение

$$\gamma = \frac{\varphi_j(h_j)}{h_j \varphi_j'(0)} = h_j \tilde{\varphi}_j \left( \frac{x - x_j}{h_j} \right) \Big|_{x=x_j+h_j} \left( h_j^2 \left[ \frac{d}{dx} \tilde{\varphi}_j \left( \frac{x - x_j}{h_j} \right) \right] \Big|_{x=x_j} \right)^{-1}. \tag{20}$$

С введением коэффициента  $\gamma$  можем переписать схему (18) в виде:

$$u(x_j + h_j) = u(x_j) + \gamma h_j f(x_j, u(x_j)). \tag{21}$$

Для того чтобы можно было применять метод LB1 в практических расчетах, необходимо задать функцию  $\varphi(x)$ . В [11] для газодинамических расчетов использовалась функция

$$\varphi_j(x - x_j) = h_j \tilde{\varphi}_j(\xi), \quad (22)$$

где  $\xi = (x - x_j)/h_j$ ,

$$\tilde{\varphi}_j(\xi) = \text{th}(\beta\xi), \quad (23)$$

$\beta$  — положительная постоянная, задаваемая пользователем. Выбор функции  $\text{th}(\beta\xi)$  связан с тем, что за счет подбора постоянной  $\beta$  она может моделировать и быстрые, и медленные компоненты решения. Наличие этих двух разных компонент характерно для жестких систем ОДУ. Тогда в случае задания функции  $\tilde{\varphi}_j(\xi)$  по формуле (23) получим  $\gamma = \frac{1}{\beta} \text{th}(\beta)$ .

Для того чтобы оценить локальную погрешность метода LB1, положим  $m = 2$  в формуле (14):  $u(x_j + h_j) = u(x_j) + \frac{u'(x_j)}{\varphi_j'(0)} \varphi_j(h_j) + r_2 + O(h^3)$ , где в соответствии с (8)

$$r_2 = \left[ \frac{u''(x_j)}{2\varphi_j'(0)^2} - \frac{u'(x_j)\varphi_j''(0)}{2\varphi_j'(0)^3} \right] \varphi_j^2(h). \quad (24)$$

Из (24) видно, что в общем случае, когда функция  $\varphi_j(x)$  не является ни четной, ни нечетной и когда  $\varphi_j''(0) \neq 0$ , локальная погрешность  $r_2$  метода LB1 может быть как больше, так и меньше, чем в случае метода Эйлера (19), что зависит и от выбора функции  $\varphi_j(x)$ , и от локального поведения решения  $u(x)$ . Теперь рассмотрим частный случай, когда  $\varphi_j(x)$  — нечетная функция. Тогда, как известно, выполняются равенства  $\varphi_j^{(2m)}(0) = 0$ ,  $m = 1, 2, \dots$ . Заметим, что и функция  $\varphi_j(x) = x$ , используемая в разложении Тейлора, тоже является нечетной. Таким образом, в случае нечетной функции  $\varphi_j(x)$  формула (24) принимает более простую форму  $r_2 = \frac{1}{2} u''(x_j) \left[ \frac{\varphi_j(h)}{\varphi_j'(0)} \right]^2$ . В случае метода (19) соответствующая погрешность

имеет вид  $r_{2,RK1} = \frac{1}{2} h^2 u''(x_j)$ . Рассмотрим отношение  $\frac{r_2}{r_{2,RK1}} = \left[ \frac{\varphi_j(h)}{\varphi_j'(0)h} \right]^2$ . Для того чтобы выполнялось неравенство  $\frac{r_2}{r_{2,RK1}} < 1$ , нужно, чтобы

$$\frac{\varphi_j(h)}{\varphi_j'(0)h} < 1. \quad (25)$$

Возьмем для примера функцию  $\varphi_j(x)$  в виде (22), (23). С учетом (9) в член погрешности порядка малости  $O(\varphi_j^3(h_j))$  войдет произведение  $\varphi_j^{(3)}(0)\varphi_j^3(h_j)$ . Однако если использовать для  $\varphi_j(x)$  формулы (22) и (23), то получим  $\varphi_j^{(3)}(0) = -\frac{2\beta^3}{h_j^2}$ ; следовательно,  $\varphi_j^{(3)}(0)\varphi_j^3(h_j) = O(h_j)$ . Для того чтобы это произведение имело порядок малости  $O(h_j^3)$ , достаточно обеспечить, чтобы  $\varphi_j^{(3)}(0) = O(1)$ . Мы поступим следующим образом: предположим, что в уравнении (1) осуществлено обезразмеривание так, что  $X - x_0 = O(1)$ . Вместо функции (22), (23) введем в рассмотрение следующую функцию:

$$\varphi_j(x) = b(x + b_1 x^3), \quad (26)$$

где  $b$  и  $b_1$  — постоянные, причем  $b > 0$ . Тогда  $\varphi_j^{(3)}(0) = 6bb_1$  и  $\varphi_j^{(k)} = 0$ ,  $k > 3$ . Постоянную  $b_1$  подберем из требования, чтобы  $\varphi_j(h) > 0$ . Для этого постоянная  $b_1$  должна удовлетворять неравенству  $b_1 > -h^{-2}$ . Далее, для того чтобы выполнялось неравенство (25), необходимо, чтобы  $b_1 < 0$  в (26).

Рассмотрим уравнение Далаквиста [2, 14]

$$u'(x) = \lambda u(x), \quad u(x_0) = u_0, \quad \lambda \in \mathbb{C}. \quad (27)$$

Применяя метод (21) к уравнению (27), получим

$$u(x_j + h_j) = (1 + \gamma h_j \lambda) u(x_j). \quad (28)$$

Пусть  $z = \lambda h_j$  и  $R(z) = 1 + \gamma z$ . Тогда можем переписать (28) в виде  $u(x_j + h_j) = R(z)u(x_j)$ . Область абсолютной устойчивости определяется как область в комплексной  $z$ -плоскости, в которой  $|R(z)| \leq 1$ .

Многочлен  $R(z)$ , который возникает при применении метода Рунге–Кутта к уравнению (27), называется многочленом устойчивости. В частности, из условия  $|1 + \gamma z| \leq 1$  получаем следующее условие абсолютной устойчивости схемы LB1 на отрицательной вещественной оси:  $-2 \leq \gamma z \leq 0$ . Отсюда следуют неравенства

$$0 \leq h \leq \frac{2}{\gamma|\lambda|}. \tag{29}$$

Тем самым приходим к такому заключению: для того чтобы область абсолютной устойчивости в случае метода LB1 была больше, чем в случае метода RK1, должны выполняться неравенства

$$0 < \gamma < 1. \tag{30}$$

Теперь посмотрим, при каких значениях  $b$  и  $b_1$  в (26) будут выполняться неравенства (30). Воспользовавшись формулой (20), получаем из (30) неравенства

$$-1 < b_1 h^2 < 0. \tag{31}$$

Чтобы получить ограничение сверху на  $|b_1|$ , подставим в (29) величину  $\gamma = 1 + b_1 h^2$  и пренебрежем членом  $b_1 h^2$ . Тогда находим, что максимально допустимое с точки зрения абсолютной устойчивости значение шага  $h$  имеет вид  $h_{\max} = \frac{2}{|\lambda|}$ . Подставив это значение вместо  $h$  в (31), получим искомые неравенства  $-\frac{1}{4}|\lambda|^2 < b_1 \leq 0$ .

**5. Метод второго порядка точности.** Рассмотрим двухстадийный метод Рунге–Кутта. С этой целью положим  $q = 1$  в (12). Тогда приращение  $\Delta u_{h,j}$  в (11) вычисляется с учетом (13) по формулам

$$\Delta u_{h,j} = A_0 g_0 + A_1 g_1, \tag{32}$$

$$g_0 = \varphi_j(h_j) f(x, u), \quad g_1 = \varphi_j(h_j) f(x + \alpha_1 h_j, u + \beta_{10} g_0). \tag{33}$$

Для того чтобы сравнить величину (32) с величиной (14), разложим  $g_1$  в ряд Лагранжа–Бюрмана в точке  $(x, u)$  как функцию двух переменных по степеням функций  $\varphi_j(x - x_j)$  и  $\varphi_j(u - u_j)$  (заметим, что при выводе классических явных методов Рунге–Кутта функцию  $g_1(x, u)$  разлагают в ряд Тейлора по степеням переменных  $(x - x_j)$  и  $(u - u_j)$  [2, 12]). Общая формула разложения в ряд функции двух переменных по степеням двух других произвольных функций двух переменных дана, например, в [15]. В соответствии с предложенной в разделе 3 конструкцией (12) метода типа Рунге–Кутта, в нашем случае необходимо разложить функцию  $g_1(x, u)$  в ряд по степеням одной и той же функции  $\varphi(\cdot)$ , но с разными аргументами. Тогда общая формула Лагранжа заметно упрощается:

$$g_1(x, u) = g_1(x_j, u_j) + \underbrace{\sum_{l=0}^m \sum_{k=0}^m}_{l+k>0} B_{l,k}(x_j, u_j) \frac{[\varphi(x - x_j)]^l [\varphi(u - u_j)]^k}{l!k!}. \tag{34}$$

Здесь для краткости мы ввели обозначение  $\varphi(\cdot) \equiv \varphi_j(\cdot)$ ; тогда

$$B_{l,k}(x_j, u_j) = \left\{ D_1^{l-1} D_2^{k-1} \left[ \left( \frac{x - x_j}{\varphi(x - x_j)} \right)^l \left( \frac{u - u_j}{\varphi(u - u_j)} \right)^k D_1 D_2 g_1(x, u) \right] \right\}_{\substack{x=x_j \\ u=u_j}}. \tag{35}$$

Здесь и ниже  $D_1 = \frac{\partial}{\partial x}$ ,  $D_2 = \frac{\partial}{\partial u}$ . Из формул (34) и (35) следует, что сложность вычисления разложения Лагранжа функции двух переменных нелинейно растет с увеличением  $m$ . Оказалось весьма эффективным применение системы *Mathematica*, однако при символьном вычислении коэффициентов  $B_{l,0}$  ( $l \geq 1$ ) и  $B_{0,k}$  ( $k \geq 1$ ) возникает трудность, связанная с наличием в (35) производных отрицательного порядка. Для того чтобы устранить этот недостаток, представим сумму, стоящую в правой части равенства (34), в виде трех сумм  $g_1(x, u) = g_1(x_j, u_j) + S_1 + S_2 + S_3$ , где

$$S_1 = \sum_{l=1}^m B_{l,0} \frac{[\varphi(x - x_j)]^l}{l!}, \quad S_2 = \sum_{k=1}^m B_{0,k} \frac{[\varphi(u - u_j)]^k}{k!}, \quad S_3 = \sum_{l=1}^m \sum_{k=1}^m B_{l,k} \frac{[\varphi(x - x_j)]^l [\varphi(u - u_j)]^k}{l!k!}.$$

Теперь преобразуем формулу для  $B_{j,0}$  с использованием соотношения  $D_2^{-1}D_1D_2 = D_2^{-1}D_2D_1 = D_1$  :

$$B_{l,0} = \left\{ D_1^{l-1} D_2^{-1} \left[ \left( \frac{x-x_j}{\varphi(x-x_j)} \right)^l D_1 D_2 g_1 \right] \right\}_{\substack{x=x_j \\ u=u_j}} = \left\{ D_1^{l-1} \left[ \left( \frac{x-x_j}{\varphi(x-x_j)} \right)^l D_1 g_1 \right] \right\}_{\substack{x=x_j \\ u=u_j}}, \quad l \geq 1. \quad (36)$$

Аналогично,

$$B_{0,k} = \left\{ D_2^{k-1} \left[ \left( \frac{u-u_j}{\varphi(u-u_j)} \right)^k D_2 g_1 \right] \right\}_{\substack{x=x_j \\ u=u_j}}, \quad k \geq 1. \quad (37)$$

Благодаря формулам (36) и (37) символьное вычисление всего разложения (34) становится вполне ясным, см. ниже соответствующую *Mathematica*-программу:

```
m = 2; B = Table[0, {j, m + 1}, {k, m + 1}];
Phi[x_] := x*Sum[(D[fi[x], {x, k}] /. x -> 0)*x^(k - 1)/k!, {k, m + 1}];

(* Computation of B[[0, k]], k = 1, ..., m *)
Do[b = D[(y/Phi[y])^k * f^(0,1)[x,y], {y, k-1}]; b = b /. {x -> 0, y -> 0}; B[[1, k + 1]] = b, {k, m}];
(* Computation of B[[j, 0]], j = 1, ..., m *)
Do[b = D[(x/Phi[x])^j * f^(1,0)[x, y], {x, j - 1}]; b = b /. {x -> 0, y -> 0}; B[[j + 1, 1]] = b, {j, m}];
(* Computation of B[[j, k]] for j >= 1, k >= 1 *)
w[x_, y_, j_, k_] := (x/Phi[x])^j * (y/Phi[y])^k * f^(1,1)[x, y];

Do[Do[b = D[w[x, y, j, k], {x, j - 1}, {y, k - 1}]; b = b /. {x-> 0, y -> 0};
B[[j + 1, k + 1]] = b, {j, m}], {k, m}];

B = B /. {f^(i,j)[0, 0] -> f^(i,j)[x0, u0]};
(* Computation of the entire bivariate Lagrange expansion *)

df = f[x0, u0] + Sum[B[[j + 1, 1]]*fi[x - x0]^j/j!, {j, m}] +
Sum[B[[1, k + 1]]*fi[u - u0]^k/k, {k, m}] +
Sum[B[[j + 1, k + 1]]*fi[x - x0]^j*fi[u - u0]^k/(j! k!), {j, m}, {k, m}];
```

После этого величина (32) вычисляется с помощью следующих *Mathematica*-команд:

```
g0 = fi[h]*f[x0,u0]; g1 = fi[h]*df /. {x-> x0 + a1*h, u-> u0 + b10*g0};
g1 = g1 /. {fi[b10 f[x0, u0] fi[h]]-> fi'[0]*b10*f[x0, u0]*fi[h],
fi[a1 h] -> fi'[0]*a1*h}; duh = Expand[A0*g0 + A1*g1];
```

Величина  $\Delta u_j$  (14) вычислялась с помощью компактной рекурсивной *Mathematica*-программы, представленной в [16].

В результате символьных вычислений было получено следующее выражение для величины (15):

$$\delta u_j = r_1 \varphi(h) + r_2 \varphi^2(h) + r_3 \varphi^3(h), \quad (38)$$

где мы ввели для краткости обозначение  $h \equiv h_j$  и

$$r_1 = -A_0 - A_1 + [\varphi'(0)]^{-1}, \quad r_2 = r_{2,0}f + r_{2,1}f_x + r_{2,2}ff_u, \\ r_3 = r_{3,0}f + r_{3,1}f_{xx} + r_{3,2}ff_{xu} + r_{3,3}f^2f_{uu} + r_{3,4}ff_{fu} + r_{3,5}ff_u^2 + r_{3,6}f^2f_u + r_{3,7}f_x + r_{3,8}f_x f_u.$$

Здесь

$$r_{2,0} = -\frac{\varphi''(0)}{2[\varphi'(0)]^3}, \quad r_{2,1} = \frac{1}{2[\varphi'(0)]^2} - \frac{\alpha_1 A_1 h}{\varphi(h)}, \quad r_{2,2} = \frac{1}{2[\varphi'(0)]^2} - A_1 \beta_{10}, \\ r_{3,0} = \frac{[\varphi''(0)]^2}{2[\varphi'(0)]^5} - \frac{\varphi^{(3)}(0)}{6[\varphi'(0)]^4}, \quad r_{3,1} = \frac{1}{6[\varphi'(0)]^3} - \frac{1}{2} \alpha_1^2 A_1 \frac{h^2}{\varphi^2(h)}, \quad r_{3,2} = \frac{1}{3[\varphi'(0)]^3} - \alpha_1 A_1 \beta_{10} \frac{h}{\varphi(h)}, \\ r_{3,3} = \frac{1}{6[\varphi'(0)]^3} - \frac{1}{2} A_1 \beta_{10}^2, \quad r_{3,4} = \frac{\varphi''(0)}{2[\varphi'(0)]^4}, \quad r_{3,5} = \frac{1}{6[\varphi'(0)]^3}, \\ r_{3,6} = \frac{A_1 \beta_{10}^2 \varphi''(0)}{2\varphi'(0)}, \quad r_{3,7} = \frac{\varphi''(0)}{2[\varphi'(0)]^4} + \frac{\alpha_1^2 A_1 h^2 \varphi''(0)}{2\varphi^2(h)\varphi'(0)}, \quad r_{3,8} = \frac{1}{6[\varphi'(0)]^3}. \quad (39)$$

С целью обеспечения выполнения соотношения  $\delta u_j = O(h^3)$  потребуем, чтобы параметры  $A_0, A_1, \alpha_1, \beta_{10}$  удовлетворяли уравнениям  $r_1 = 0, r_{21} = r_{22} = 0$ . Будем предполагать, что  $\varphi(x)$  — нечетная функция. Тогда получаем из (39) следующие определяющие уравнения для параметров  $A_0, A_1, \alpha_1, \beta_{10}$ :

$$A_0 + A_1 = \frac{1}{\varphi'(0)}; \quad \alpha_1 A_1 h = \frac{\varphi(h)}{2[\varphi'(0)]^2}; \quad A_1 \beta_{10} = \frac{1}{2[\varphi'(0)]^2}. \tag{40}$$

Число этих уравнений меньше, чем число неизвестных. Следуя [12], выразим  $A_0, \alpha_1, \beta_{10}$  как функции параметра  $A_1$ :

$$A_0 = \frac{1}{\varphi'(0)} - A_1; \quad \alpha_1 = \frac{\varphi(h)}{2[\varphi'(0)]^2 A_1 h}; \quad \beta_{10} = \frac{1}{2A_1[\varphi'(0)]^2}. \tag{41}$$

Семейство схем (32), (33), (41) будем называть методом LB2. В частном случае, когда  $\varphi(x) = x$ , равенства (41) переходят в равенства, полученные в [12] для классического явного метода Рунге–Кутты второго порядка точности.

Мы можем теперь попытаться найти  $A_1$  из требования минимизации члена  $r_3$  в (38). Для этого подставим в  $r_{3,1}$  и  $r_{3,2}$  выражения для  $\alpha_1$  и  $\beta_{10}$  и потребуем, чтобы  $r_{3,1} = r_{3,2} = 0$ . Отсюда получаем следующее выражение для  $A_1$ :  $A_1 = \frac{3}{[4\varphi'(0)]}$ . При этом значении  $A_1$  также получаем, что  $r_{3,3} = 0$ , и тогда, в случае нечетной функции  $\varphi(x)$ , выражение для  $r_3$  упрощается к виду  $r_3 = r_{3,5} f f_u^2 + r_{3,8} f_x f_u$ .

В случае классического метода Рунге–Кутты второго порядка точности и  $A_1 = 3/4$  локальная погрешность имеет вид [12]  $r_{3,RK2} = \frac{h^3}{6} f_u(f_x + f f_u)$ . Рассмотрим отношение  $\frac{r_{3,5} f f_u^2 + r_{3,8} f_x f_u}{r_{3,RK2}}$ . Тогда, как и в случае метода LB1, приходим к выводу, что описанное выше семейство методов типа Рунге–Кутты второго порядка точности имеет при выполнении неравенства (25) меньшую локальную погрешность, чем классический метод Рунге–Кутты с  $A_1 = 3/4$ .

Возвращаясь к оптимальному по точности методу второго порядка точности (32), (33), (41) со значением  $A_1 = \frac{3}{[4\varphi'(0)]}$ , можем записать расчетные формулы этого метода в виде

$$g_0 = \varphi(h) f(x, u), \quad g_1 = \varphi(h) f\left(x + \frac{2\varphi(h)}{3\varphi'(0)}, u + \frac{2}{3\varphi'(0)} g_0\right), \quad \Delta u_h = \frac{g_0 + 3g_1}{4\varphi'(0)}. \tag{42}$$

Полином устойчивости  $R(z)$  для рассматриваемой схемы второго порядка точности запишется с учетом (40) и обозначения (20) в виде

$$R(z) = 1 + (A_0 + A_1) \frac{\varphi(h)}{h} z + A_1 \left[\frac{\varphi(h)}{h}\right]^2 \beta_{10} z^2 = 1 + \gamma z + \frac{1}{2} \gamma^2 z^2. \tag{43}$$

Заметим, что при  $\gamma < 1$  коэффициенты многочлена (43) при  $z^k$  убывают с ростом  $k$  быстрее, чем в случае классического метода Рунге–Кутты второго порядка точности (для которого  $\gamma = 1$ ). Как указывалось в [6], это является предпосылкой для увеличенного интервала абсолютной устойчивости на отрицательной вещественной оси. Из неравенства  $|R(z)| \leq 1$  получаем следующее условие абсолютной устойчивости метода LB2 на отрицательной вещественной оси:

$$-\frac{2}{\gamma} \leq \operatorname{Re}(\lambda h) \leq 0. \tag{44}$$

Например, при  $\gamma = 0.1$  получаем интервал устойчивости, который в десять раз больше, чем в случае классического метода Рунге–Кутты второго порядка точности.

В качестве примера рассмотрим систему [17]

$$\frac{du}{dt} = J u, \tag{45}$$

где  $t$  — время и  $J$  — матрица второго порядка:

$$J = \begin{pmatrix} -1000 & 999 \\ 1 & -2 \end{pmatrix}. \tag{46}$$



Собственными значениями матрицы  $J$  являются  $\lambda_1 = -1001$ ,  $\lambda_2 = -1$ . Таким образом, система (45), (46) — пример умеренно жесткой задачи. Решение  $\mathbf{u} = (n_1, n_2)$  системы (45), (46) записывается в виде

$$\begin{aligned} n_1(t) &= 0.999(n_1(0) - n_2(0))e^{-1001t} + (0.001n_1(0) + 0.999n_2(0))e^{-t}, \\ n_2(t) &= -0.001(n_1(0) - n_2(0))e^{-1001t} + (0.001n_1(0) + 0.999n_2(0))e^{-t}. \end{aligned} \quad (47)$$

Разделив первое уравнение системы (45), (46) на  $10^3$ , получим

$$10^{-3} \frac{dn_1}{dt} = -n_1 + 0.999n_2, \quad (48)$$

$$\frac{dn_2}{dt} = n_1 - 2n_2. \quad (49)$$

Уравнение (48) содержит малый параметр  $\varepsilon = 10^{-3}$ . Соответствующее уравнению (48) вырожденное уравнение получается в пределе при  $\varepsilon \rightarrow 0$ :

$$-n_1 + 0.999n_2 = 0. \quad (50)$$

Из (50) имеем

$$n_1 = 0.999n_2. \quad (51)$$

Подставляя это значение в (49), получаем обыкновенное дифференциальное уравнение

$$\frac{dn_2}{dt} = -1.001n_2. \quad (52)$$

Очевидно,  $\lambda_1 = -1.001 < 0$ . Условие  $\operatorname{Re} \lambda_j < 0$  является достаточным для устойчивости решения вырожденной системы. Для достаточно малого параметра  $\varepsilon$  касательные к интегральным кривым почти параллельны оси  $n_1$ , т.е. наклон  $\left| \frac{dn_1}{dt} \right|$  велик. У интегральной кривой  $n_1 = n_1(t)$  имеются два участка различного поведения. Первый участок с быстрым изменением искомой функции отражает стремление интегральной кривой к графику функции  $\bar{n}_1 = \bar{n}_1(t)$ , полученному из решения вырожденной системы (51), (52). Этот участок называется пограничным слоем.

На втором участке производные решения значительно меньше, а интегральная кривая практически совпадает с графиком  $\bar{n}_1(t)$ .

Вернемся к решению (47). Внутри пограничного слоя переменная  $n_1$  ведет себя заметно активнее, чем  $n_2$ . Поэтому иногда  $n_1(t)$  называют быстрой компонентой, а  $n_2(t)$  — медленной. После прохождения пограничного слоя производные вектора решения невелики и определяются экспонентой с показателем  $\lambda_2$ .

Мы провели численные расчеты задачи (45), (46) по LB2-методу (42) и, для сравнения, по классическому методу Рунге–Кутты, получаемому из (42) в частном случае, когда  $\varphi(x) = x$ :

$$g_0 = hf(x, u), \quad g_1 = hf\left(x + \frac{2}{3}h, u + \frac{2}{3}g_0\right), \quad \Delta u_h = \frac{1}{4}(g_0 + 3g_1). \quad (53)$$

В случае системы (45), (46)  $\mathbf{u}$  — вектор-столбец,  $\mathbf{u} = (n_1, n_2)^T$ , где  $T$  — операция транспонирования. Метод (53) будем в дальнейшем называть RK2-методом. Расчеты по обоим методам проводились на одной и той же равномерной сетке в интервале  $0 \leq t \leq 0.2$ . Шаг сетки задавался с учетом (44) по формуле

$$h = \frac{2\theta}{\lambda_{\max}}, \quad (54)$$

где  $\theta$  — множитель надежности,  $0 < \theta \leq 1$ ,  $\lambda_{\max} = \max(|\lambda_1|, |\lambda_2|) = 1001$ . С целью проведения расчетов по обоим методам на одинаковой сетке не реализовывалась какая-либо стратегия вычисления переменного шага  $h_j = t_{j+1} - t_j$ ,  $t_j = jh$  (некоторые из этих стратегий упоминаются, например, в [2, 7]). Пусть  $n_1^{\text{ex}}(t)$ ,  $n_2^{\text{ex}}(t)$  — точное решение (47) системы (45), (46). Введем абсолютные локальные ошибки  $\delta n_{1j}$ ,  $\delta n_{2j}$  по формулам

$$\delta n_{kj} = |n_{kj} - n_{kj}^{\text{ex}}|, \quad k = 1, 2, \quad j = 0, \dots, N, \quad (55)$$

а также сеточные аналоги  $L_2$ -нормы ошибок численного решения:

$$\Delta n_k = \|n_k - n_k^{\text{ex}}\|_2 = \left[ \frac{1}{t_N - t_0} \sum_{j=0}^{N-1} (n_{kj} - n_{kj}^{\text{ex}})^2 (t_{j+1} - t_j) \right]^{0.5}, \quad k = 1, 2. \quad (56)$$

Таблица 1  
Влияние коэффициента  $b_1$  на точность метода LB2

|              |                       |                       |                       |                       |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $b_1$        | $-10^4$               | $-5 \times 10^4$      | $-7.5 \times 10^4$    | $-10^5$               |
| $\Delta n_1$ | $3.77 \times 10^{-2}$ | $3.00 \times 10^{-2}$ | $2.89 \times 10^{-2}$ | $3.02 \times 10^{-2}$ |
| $\Delta n_2$ | $1.98 \times 10^{-3}$ | $1.03 \times 10^{-2}$ | $1.56 \times 10^{-2}$ | $2.09 \times 10^{-2}$ |

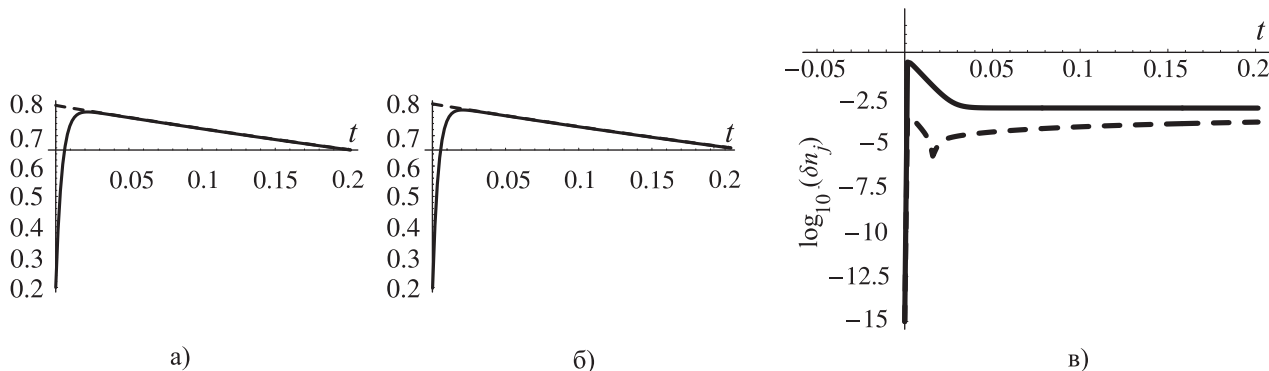


Рис. 1. Численное решение задачи (45), (46): а) метод RK2 (53); б) метод LB2 (42) при  $b = 4$ ,  $b_1 = -10^4$  в (26); в) локальные ошибки (55), полученные при расчетах по методу RK2

При использовании в (54) значения  $\theta = 0.8$  были получены по методу RK2 (53) следующие значения ошибок (56):  $\Delta n_1 = 4.11 \times 10^{-2}$ ,  $\Delta n_2 = 8.89 \times 10^{-5}$ .

В табл. 1 приведены значения ошибок (56) при фиксированном значении  $b = 4$  в (26), но при различных  $b_1$ . Из этой таблицы видно, что ошибка  $\Delta n_1$  сначала уменьшается с ростом  $|b_1|$  и достигает минимума при  $b_1 \approx -7.5 \times 10^4$ , при этом она остается меньшей, чем в случае классического метода RK2. В то же время ошибка  $\Delta n_2$  намного больше, чем в случае метода RK2.

На рис. 1а и 1б функции  $n_1$  и  $n_2$  показаны соответственно сплошной и штриховой линиями. Ошибка  $\delta n_1$  иллюстрируется на рис. 1в сплошной линией, а ошибка  $\delta n_2$  — штриховой.

С целью дальнейшего повышения точности метода LB2 (42) рассмотрим следующую его небольшую модификацию. В случае нечетной функции  $\varphi(x)$  имеем  $\varphi(h) = \varphi'(0)h + O(h^3)$ . Тогда  $\varphi'(0) = \frac{\varphi(h)}{h} + O(h^2)$ .

Заменим в выражении для  $\Delta u_h$  в (42)  $\varphi'(0)$  по формуле  $\varphi'(0) = \frac{\varphi(h)}{h}$ :

$$\begin{aligned}
 g_0 &= \varphi(h)f(x, u), \\
 g_1 &= \varphi(h)f\left(x + \frac{2\varphi(h)}{3\varphi'(0)}, u + \frac{2}{3\varphi'(0)}g_0\right), \\
 \Delta u_h &= \frac{(g_0 + 3g_1)h}{4\varphi(h)}.
 \end{aligned}
 \tag{57}$$

Ясно, что эта замена не понижает порядка локальной аппроксимации метода LB2. Назовем метод (57) методом LB2M, т.е. методом LB2 модифицированным.

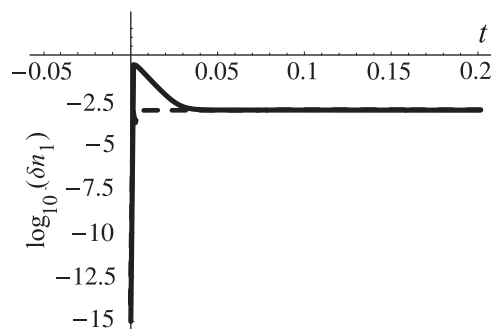


Рис. 2. Локальная ошибка  $\delta n_1$  для метода RK2 (сплошная линия) и метода LB2M (штриховая линия) при  $b = 4$ ,  $b_1 = -1.47 \times 10^5$  в (26)

Таблица 2  
Влияние коэффициента  $b_1$  на точность метода LB2M

|              |                       |                       |                       |                       |                       |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $b_1$        | $-10^4$               | $-5 \times 10^4$      | $-10^5$               | $-1.47 \times 10^5$   | $-2 \times 10^5$      |
| $\Delta n_1$ | $3.66 \times 10^{-2}$ | $2.24 \times 10^{-2}$ | $9.60 \times 10^{-3}$ | $8.10 \times 10^{-4}$ | $9.61 \times 10^{-4}$ |
| $\Delta n_2$ | $8.85 \times 10^{-5}$ | $9.02 \times 10^{-5}$ | $9.62 \times 10^{-5}$ | $1.04 \times 10^{-4}$ | $1.13 \times 10^{-4}$ |

В табл. 2 приведены значения ошибок (56) при фиксированном значении  $b = 4$  в (26), но при различ-

ных  $b_1$ . Из этой таблицы видно, что медленная компонента решения  $n_2$  считается с той же точностью, что и при использовании классического метода RK2. Точность счета быстрой компоненты  $n_1$  намного выше, чем по методу RK2.

Из табл. 2 также следует, что погрешность  $\Delta n_1$  достигает своего минимума при  $b_1 = -1.47 \times 10^5$ , при этом она примерно в 50 раз ниже, чем в случае метода RK2.

На рис. 2 показана локальная ошибка  $\delta n_1$  для случаев метода RK2 и LB2M. Видно, что вне пограничного слоя обе ошибки практически совпадают. В области пограничного слоя поведение локальной ошибки — монотонное в случае метода LB2M, в отличие от метода RK2, и именно за счет малой локальной ошибки метода LB2M внутри пограничного слоя получается суммарный выигрыш этого метода в терминах интегральной ошибки  $\Delta n_1$ .

В [14] описан пакет `OrderStar`, позволяющий получать графические изображения областей абсолютной устойчивости как явных, так и неявных методов численного интегрирования ОДУ.

Применение пакета `OrderStar` к методу RK2 и к методу LB2M (при  $b = 4$ ,  $b_1 = -1.47 \times 10^5$  в (26)) реализуется следующим образом:

```
<< NumericalMath`OrderStar`
fi[x_] := b*(x - b1*x^3); b = 4; b1 = 147000; h = 0.8*2.0/1001;
hj = fi[h]; b2 = h*b/hj; gam = 1.0/b2;
RLB2 = 1 + gam*z + (1/2)*(gam*z)^2; RRK2 = 1 + z + (1/2)*z^2;
stbLB2 = OrderStar[RLB2, 1, OrderStarZeros -> {False, False}, OrderStarSubPlots -> False,
PlotRange -> {{-3.5, 0.2}, {-3, 3}}, ContourShading -> False, DisplayFunction -> Identity];
stbRK2 = OrderStar[RRK2, 1, OrderStarZeros -> {False, False}, OrderStarSubPlots -> False,
PlotRange -> {{-3.5, 0.2}, {-3, 3}}, ContourShading -> False, DisplayFunction -> Identity];
plots = {stbLB2, stbRK2}; labels = Graphics[{Text["LB2M", {-3, 2.5}],
Text["RK2", {-2.3, 1.3}]}]; Show[Join[plots, {labels}], DisplayFunction -> $DisplayFunction];
```

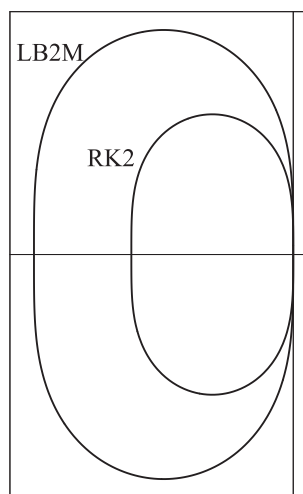


Рис. 3. Области абсолютной устойчивости методов RK2 и LB2M

Полученные области абсолютной устойчивости показаны на рис. 3, откуда следует, что область абсолютной устойчивости метода LB2M значительно больше, чем область абсолютной устойчивости метода RK2. Интервал абсолютной устойчивости методов LB2 и LB2M на отрицательной вещественной оси увеличивается с увеличением  $|b_1|$ .

#### СПИСОК ЛИТЕРАТУРЫ

1. *Dahlquist G.* A special stability problem for linear multistep methods // BIT. 1963. **3**. 27–43.
2. *Холл Дж., Уатт Дж.* Современные численные методы решения обыкновенных дифференциальных уравнений. М.: Мир, 1979.
3. *Houwen van der P., Sommeijer B.* On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values // Z. Angew. Math. Mech. 1980. **60**. 479–485.
4. *Abdulle A.* Chebyshev methods based on orthogonal polynomials. Ph.D. Doctoral Dissertation No. 3266, Dept. Math., Univ. of Geneva, 2001.

5. *Abdulle A.* Fourth order Chebyshev methods with recurrence relation // *SIAM J. Sci. Comput.* 2002. **23**. 2041–2054.
6. *Hundsdorfer W., Verwer J.* Numerical solutions of time-dependent advection–diffusion–reaction equations. Heidelberg: Springer-Verlag, 2007.
7. *Martin-Vaquero J., Janssen B.* Second-order stabilized explicit Runge–Kutta methods for stiff problems // *Computer Phys. Communications.* 2009. **180**. 1802–1810.
8. *Пинчук В.И., Шу Ч.-В.* Численные методы высоких порядков для задач аэрогидродинамики. Новосибирск: Изд-во Сиб. отд-ния РАН, 2000.
9. *Абрамовиц М., Стиган И.* Справочник по специальным функциям. М.: Наука, 1979.
10. *Лаврентьев М.А., Шабат Б.В.* Методы теории функций комплексного переменного. М.: Наука, 1973.
11. *Ворожцов Е.В.* Построение разностных схем для гиперболических законов сохранения с помощью разложения Лагранжа–Бюрмана // Труды Междунар. конф. по вычислительной математике / Ред. Г.А. Михайлов, В.П. Ильин, Ю.М. Лаевский. Часть I. Новосибирск: Прайс-куррьер, 2004. 443–448.
12. *Крылов В.И., Бобков В.В., Монастырский П.И.* Вычислительные методы. Т. II. М.: Наука, 1977.
13. *Butcher J.C.* The numerical analysis of ordinary differential equations. Chichester: Wiley, 1987.
14. *Sofroniou M.* Order stars and linear stability theory // *J. Symbolic Computation.* 1996. **21**. 101–131.
15. *Consul P.C., Famoye F.* Lagrangian probability distributions. Berlin: Birkhäuser, 2006.
16. *Strampp W., Ganzha V., Vorozhtsov E.* Höhere Mathematik mit Mathematica. Band 3: Differentialgleichungen und Numerik. Braunschweig/Wiesbaden: Vieweg, 1997.
17. *Ракитский Ю.В., Устинов С.М., Черноруцкий И.Г.* Численные методы решения жестких систем. М.: Наука, 1979.

Поступила в редакцию  
27.04.2010

---