

УДК 519.622

НЕЯВНЫЕ ГНЕЗДОВЫЕ МЕТОДЫ РУНГЕ–КУТТЫ ТИПА ГАУССА И ЛОБАТТО С ЛОКАЛЬНЫМ И ГЛОБАЛЬНЫМ КОНТРОЛЕМ ТОЧНОСТИ ДЛЯ ЖЕСТКИХ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ¹⁾

© 2020 г. Г. Ю. Куликов

1049-001 Лиссабон, пр-т Ровишку Паиш, Центр вычисл. и стохастической математики, высший техн. институт, Лиссабонский университет, Португалия

e-mail: gkulikov@math.ist.utl.pt

Поступила в редакцию 31.08.2019 г.

Переработанный вариант 31.08.2019 г.

Принята к публикации 10.03.2020 г.

Исследуется задача эффективной оценки и контроля глобальной погрешности неявных гнездовых методов Рунге–Кутты типа Гаусса и Лобатто, используемых при интегрировании жестких обыкновенных дифференциальных уравнений (ОДУ). Жесткие задачи возникают во многих прикладных областях и их точное численное решение является важным вопросом вычислительной и прикладной математики. Эффективный метод оценки глобальной ошибки, недавно разработанный для встроенных методов Рунге–Кутты, может сильно завышать глобальную ошибку при применении к жестким ОДУ и, следовательно, снижать эффективность этих решателей. В настоящей работе исследована причина завышения этой ошибки и показано, как улучшить указанные методы для решения жестких задач. Такие модификации не только повышают эффективность численного интегрирования жестких ОДУ, но и делают встроенные пары неявных гнездовых методов Рунге–Кутты с автоматическим управлением масштабированными модифицированными локальными и глобальными ошибками численного решения превосходящими жесткие встроенные решатели ОДУ в пакете MATLAB с автоматическим управлением локальной ошибкой. Библиограф. 79. Фиг. 4.

Ключевые слова: обыкновенное дифференциальное уравнение, жесткая задача, неявные гнездовые методы Рунге–Кутты типа Гаусса и Лобатто, абсолютные и масштабированные локальные и глобальные оценки ошибки, автоматический локальный и глобальный контроль точности численного решения.

DOI: 10.31857/S0044466920070078

1. ВВЕДЕНИЕ

Обыкновенные дифференциальные уравнения (ОДУ), возникающие в прикладной науке и технике, обычно представляются в виде следующей задачи Коши (ЗК):

$$x'(t) = g(t, x(t)), \quad t \in [t_0, t_{\text{end}}], \quad x(t_0) = x^0, \quad (1.1)$$

где $x(t) \in \mathbb{R}^n$ и $g: D \subset \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ – достаточно гладкая функция. Эти задачи являются типичными при математическом моделировании динамических явлений.

Точность численного решения ЗК (1.1) может иметь решающее значение для достоверности математического моделирования исследуемого явления. Например, в [1]–[15] показано, что применение адаптивных ОДУ-решателей с автоматическим управлением погрешности в непрерывно-дискретной калмановской фильтрации позволяет существенно улучшить качество оценивания стохастических систем в задачах слежения, навигации, управления, описания химических реакций, финансового моделирования и т.д., где широко используются стохастические динамические системы с непрерывным временем [16]–[24]. Поэтому настоящая статья посвящена

¹⁾ Автор выражает благодарность за финансовую поддержку португальского фонда науки и технологии – *Fundação para a Ciência e a Tecnologia*, проекты UIDB/04621/2020 и UIDP/04621/2020, выполняемые центром вычислительной и стохастической математики при высшем техническом институте Лиссабонского университета.

решению этой важной задачи и представляет вычислительные средства для обработки жестких ОДУ (1.1). Эти адаптивные численные методы эффективны для практического использования, а также они превосходят жесткие встроенные решатели ОДУ в пакете MATLAB на ряде важных тестовых задач.

Далее будем считать, что ЗК (1.1) имеет единственное решение $x(t)$ на интервале интегрирования $[t_0, t_{\text{end}}]$.

Традиционная вычислительная технология для решения уравнения (1.1) опирается на формулы Рунге–Кутты, которые привлекают внимание практиков уже более века. Такие методы эффективны для решения различных проблем, в том числе нежестких и жестких ОДУ, обратимых и гамильтоновых уравнений, задач, возникающих при полудискретизации уравнений в частных производных методом прямых и т.д. (см., например, [25]–[32]). При применении к ЗК (1.1) общая схема метода Рунге–Кутты (РК) имеет вид

$$x_{ki} = x_k + \tau_k \sum_{j=1}^l a_{ij}g(t_{kj}, x_{kj}), \quad i = 1, 2, \dots, l, \tag{1.2a}$$

$$x_{k+1} = x_k + \tau_k \sum_{j=1}^l b_jg(t_{kj}, x_{kj}), \quad k = 0, 1, \dots, K - 1, \tag{1.2б}$$

где $x_0 = x^0$, $t_{kj} := t_k + c_j\tau_k$, τ_k – постоянный или переменный размер шага численного интегрирования и векторы x_{ki} являются стадийными (промежуточными) величинами l -стадийной схемы РК (1.2). Заметим, что этот метод полностью определен его фиксированными действительными коэффициентами a_{ij} , b_j и c_j , $i, j = 1, 2, \dots, l$, которые могут быть удобно представлены в виде таблицы Бутчера

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array},$$

где A – квадратная матрица размера l , b и c суть l -мерные векторы. По форме матрицы A различают подклассы формул РК, т.е. явные, диагонально неявные и полностью неявные схемы. Явные методы являются самыми “дешевыми” и наиболее удобными для численного решения прикладных задач. Однако этот подкласс неэффективен для решения многих жестких ОДУ, которые часто возникают на практике.

Таким образом, неявные схемы РК также необходимы. Более того, некоторые полностью неявные формулы (как, например, методы Гаусса и Лобатто) являются эффективными средствами для решения обратимых и гамильтоновых задач [27]–[31].

К сожалению, явление снижения порядка аппроксимации, которое сопровождает упомянутые выше методы Гаусса и Лобатто, а также их трудоемкость снижают практическую значимость этих методов РК, см., например, [25], [26], [29].

Наше исследование будет посвящено неявным гнездовым методам Рунге–Кутты NIRK (*Nested Implicit Runge–Kutta*) типа Гаусса и Лобатто, которые наследуют преимущества классических методов Гаусса и Лобатто, но не имеют их недостатков. Эти схемы NIRK впервые были построены и изучены в [33]–[37]. Они представляют собой специальное семейство эффективных моно-неявных формул РК (MIRK), предложенных в [38]–[41]. По определению (см., например, [35]–[37]), l -уровневый метод NIRK, примененный к ЗК (1.1), может быть записан в виде

$$x_{kj}^2 = a_{j1}^2x_k + a_{j2}^2x_{k+1} + \tau_k(d_{j1}^2g(t_k, x_k) + d_{j2}^2g(t_{k+1}, x_{k+1})), \quad j = 1, 2, \tag{1.3a}$$

$$\begin{aligned} x_{kj}^i &= a_{j1}^ix_k + a_{j2}^ix_{k+1} + \tau_k(d_{j1}^ig(t_k, x_k) + d_{j2}^ig(t_{k+1}, x_{k+1})) + \\ &+ \tau_k \sum_{m=1}^{i-1} d_{j,m+2}^ig(t_{km}^{i-1}, x_{km}^{i-1}), \quad i = 3, 4, \dots, l, \quad j = 1, 2, \dots, i, \end{aligned} \tag{1.3б}$$

$$x_{k+1} = x_k + \tau_k \sum_{i=1}^l b_i g(t_{ki}^l, x_{ki}^l), \tag{1.3в}$$

где начальное значение $x_0 = x^0$, а стадийные узлы $t_{kj}^i := t_k + \tau_k c_j^i$. Приведенная формулировка предполагает, что номер уровня l подчиняется условию $l \geq 2$, хотя NIRK-схемы с $l = 1$ также существуют. В этом последнем случае формулы (1.3а) и (1.3б) отсутствуют и все такие методы полностью определяются одной формулой (1.3в). Правило трапеций TR (*Trapezoidal Rule*) является примером таких схем NIRK.

Численные методы вида (1.3) мотивированы необходимостью дешевых и эффективных решателей, которые применимы к ЗК (1.1) различного рода. Существенно, что они наследуют главную особенность всех формул MIRK и не увеличивают размер решаемых ОДУ, т.е. нелинейные задачи, возникающие при дискретизации (1.3), всегда имеют ту же размерность n , что и исходная система ОДУ вне зависимости от порядка метода NIRK и количества его этапов (см., например, [33]–[37]).

Это особенно важно для эффективного интегрирования больших жестких задач, особенно тех, которые получены с помощью полудискретизации уравнений в частных производных (УрЧП) методом прямых, поскольку приведенные выше методы Гаусса и Лобатто, а также многие полностью неявные схемы РК расширяют размер обрабатываемых нелинейных уравнений до $l \times n$, где l – число неявных стадий в используемой формуле РК (1.2).

Отметим, что другие эффективные неявные формулы РК, такие как однократно диагонально-неявные методы Рунге–Кутты (SDIRK) или однократно неявные схемы Рунге–Кутты (SIRK), имеют некоторые недостатки, ограничивающие область их применения (см., например, [35], [36]).

Кроме того, построенные формулы NIRK (1.3) типа Гаусса обладают многими привлекательными свойствами для практического использования. Например, это A -устойчивость, жесткая точность, симметричность и сопряженность симплектическими методами РК как минимум до порядка 6. Они также имеют достаточно высокие стадийные, псевдо-стадийные и классические порядки, а их некоторые стадийные величины, вычисленные на каждом шаге, достаточно точны и позволяют осуществлять плотную выдачу результатов интегрирования того же порядка, что и порядок сходимости основной формулы NIRK [35]–[37], [42], [43]. Именно поэтому методы NIRK (1.3) эффективны для решения различных ЗК вида (1.1), которые возникают в прикладных и инженерных исследованиях и охватывают нежесткие и жесткие ОДУ, а также обратимые и гамильтоновы задачи.

В [1]–[15] подробно изучено и описано применение методов NIRK для обработки математических моделей в гидромеханике и задачах нелинейной фильтрации Калмана.

В настоящей статье мы рассмотрим методы NIRK (1.3), удовлетворяющие условию

$$a_{j1}^i + a_{j2}^i = 1, \quad i = 2, 3, \dots, l, \quad j = 1, 2, \dots, i. \quad (1.4a)$$

Потребуем также, чтобы стадийные величины x_{kj}^i хорошо аппроксимировали точное решение $x(t)$ ЗК (1.1) в стадийных узлах t_{kj}^i , определяемых заданными коэффициентами

$$c_j^i = a_{j2}^i + \sum_{m=1}^{i-1} a_{jm}^i. \quad (1.4б)$$

В [35]–[37] доказано, что любой метод NIRK (1.3) с порядком сходимости $s \geq 1$ и удовлетворяющий условиям (1.4) может быть представлен в виде неявной формулы РК (1.2) с таблицей Бутчера вида

$$\begin{array}{c|ccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ c^2 & d_1^2 & 0 & 0 & \dots & 0 & a^2 b^T & d_2^2 \\ c^3 & d_1^3 & D^3 & 0 & \dots & 0 & a^3 b^T & d_2^3 \\ c^4 & d_1^4 & 0 & D^4 & \dots & 0 & a^4 b^T & d_2^4 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ c^l & d_1^l & 0 & 0 & \dots & D^l & a^l b^T & d_2^l \\ 1 & 0 & 0 & 0 & \dots & 0 & b^T & 0 \\ \hline & 0 & 0 & 0 & \dots & 0 & b^T & 0 \end{array}$$

в которой использована следующая векторная и матричная запись:

$$D^i := \begin{pmatrix} d_{i3}^i & \cdots & d_{i,i+1}^i \\ \vdots & \ddots & \vdots \\ d_{i3}^i & \cdots & d_{i,i+1}^i \end{pmatrix}, \quad d_1^i := (d_{i1}^i, \dots, d_{i1}^i)^T, \quad d_2^i := (d_{i1}^i, \dots, d_{i1}^i)^T, \\ d^i := (d_{i2}^i, \dots, d_{i2}^i)^T, \quad c^i := (c_1^i, \dots, c_i^i)^T, \quad b := (b_1, \dots, b_l)^T.$$

В настоящее время на практике принято применять адаптивные решатели для ОДУ, которые подразумевают, что методы РК (1.2), предназначенные для вычисления приближений $\{x_k\}_{k=0}^K$ к точному решению $x(t)$ ЗК (1.1), включают и механизм для автоматического построения подходящей сетки

$$\{t_k\}_{k=0}^K := \{t_{k+1} = t_k + \tau_k, k = 0, 1, \dots, K - 1, t_K = t_{\text{end}}\},$$

на которой такое численное решение вычисляется.

Это делает процедуру численного интегрирования более гибкой, эффективной и точной. Например, все встроенные в пакете MATLAB решатели ОДУ организованы именно таким образом, см. [44], [45].

С другой стороны, некоторое исключение из этого правила, т.е. когда методы с фиксированным шагом используются в реальных приложениях, также могут существовать [46]. Это касается случаев, когда гамильтоновы задачи интегрируются численно и где изменение размера шага может нарушать особые свойства геометрического численного интегрирования [27], [31]. В литературе также известно, что обсуждаемая реализация с переменным шагом интегрирования может создавать определенные неудобства для решения прикладных задач и требует непрерывного расширения реализуемой схемы РК для получения численных решений в заранее заданных точках на интервале интегрирования (см., например, [47]–[53]).

К счастью, последняя проблема решается автоматически во всех методах NIRK (1.3) из-за их специальной процедуры построения, которая позволяет применить дешевую процедуру плотной выдачи результатов численного интегрирования того же порядка аппроксимации, что и численное решение, полученное основной формулой NIRK.

Одной из важнейших характеристик любого приближенного решения является достигнутая точность численного интегрирования, т.е. величина ее глобальной ошибки $\{\Delta x_k := x(t_k) - x_k\}_{k=0}^K$ на автоматически генерируемой сетке $\{t_k\}_{k=0}^K$.

Однако большинство программных кодов для практического применения реализуют только тот или иной вариант управления локальной ошибкой численного интегрирования. Другими словами, программный код вычисляет оценку локальной погрешности схемы РК (или любой другой схемы) и сетка $\{t_k\}_{k=0}^K$ генерируется исходя их условия, что оценка локальной ошибки приближенного решения, вычисленного в узлах такой сетки, не может превышать заданной пользователем величины допуска Tol (см., например, описание часто используемых решателей ОДУ в пакете MATLAB [44], [45]).

Последний подход эффективно реализуется для методов РК (1.2) в виде их вложенных пар, т.е. приведенная выше таблица Бутчера дополнена коэффициентами $\tilde{b}_j, j = 1, 2, \dots, l$, и принимает вид

$$\begin{array}{c} c \\ \hline A \\ \hline b^T \\ \hline \tilde{b}^T \end{array}. \tag{1.5}$$

Последняя строка таблицы (1.5) используется для вычисления второго (обычно более низкого порядка) численного решения

$$x_{k+1}^{\text{emb}} = x_k + \tau_k \sum_{j=1}^l \tilde{b}_j g(t_k + c_j \tau_k, x_{kj}), \quad k = 0, 1, \dots, K - 1. \tag{1.6}$$

Тогда локальная ошибка оценивается с помощью простой формулы

$$le_{k+1} = x_{k+1}^{\text{emb}} - x_{k+1} \tag{1.7}$$

на $(k + 1)$ -м шаге вложенной пары РК. Далее, оценка локальной погрешности (1.7) масштабируется следующим образом:

$$|e_{k+1}|_{sc} := \max_{i=1,2,\dots,n} \left\{ |e_{k+1}^i| / (\text{atol} + \text{rtol} \times |x_{k+1}^i|) \right\}, \quad (1.8)$$

где верхний индекс i означает i -й элемент в соответствующем векторе и два фиксированных параметра atol и rtol определяют соответственно вклад абсолютной и относительной ошибок в масштабированную оценку локальной погрешности. В этой статье мы ограничимся механизмом контроля ошибки, удовлетворяющим условию $\text{atol} = \text{rtol} := \text{Tol}$.

Далее мы проверяем, что оценка локальной ошибки (1.8) удовлетворяет условию

$$|e_{k+1}|_{sc} \leq 1. \quad (1.9)$$

Затем оптимальный размер τ_k^* $(k + 1)$ -го шага определяется по формуле

$$\tau_k^* := \min \left\{ 1.5, 0.8 / |e_{k+1}|_{sc}^{1/(p+1)} \right\} \tau_k, \quad (1.10)$$

где p – порядок согласованности встроенной формулы РК (1.6), 1.5 – верхняя граница увеличения размера шага, установленная в нашем механизме автоматического управления размером шага численного интегрирования, а 0.8 служит обычным коэффициентом безопасности численного интегрирования.

Если условие (1.9) не выполняется, то локальная оценка погрешности (1.8) будет считаться слишком большой и, следовательно, численное решение x_{k+1} , вычисленное на $(k + 1)$ -м шаге с помощью встроенной пары РК (1.5), будет недостаточно точным и неприемлемым. Поэтому в этом случае все значения, вычисленные на текущем шаге, будут отклонены, размер шага τ_k будет уменьшен на основе формулы (1.10) и вычисления решения x_{k+1} и соответствующей оценки локальной ошибки $|e_{k+1}|_{sc}$ будут повторены той же парой (1.5), но для скорректированного размера шага $\tau_k = \tau_k^*$.

Если условие точности (1.9) выполняется для исходного (или скорректированного) размера шага τ_k , то все найденные значения численного решения x_{k+1} и локальной ошибки $|e_{k+1}|_{sc}$ будут приняты и τ_k^* будет рассматриваться как хороший прогноз следующего размера шага в адаптивной паре РК (1.5), т.е. $\tau_{k+1} = \tau_k^*$ (см. подробное объяснение представленной процедуры управления размером шага в [25], [28], [29], [32]). Отметим, что формула (1.10) автоматически увеличит размер шага численного интегрирования, если текущий шаг принят и оценка $|e_{k+1}|_{sc}$ допущенной локальной ошибки много меньше 1.

Подчеркнем, что оценка локальной ошибки (1.6)–(1.8) является малозатратной, поскольку обе схемы РК во встроенной паре (1.5) используют одни и те же стадийные величины x_{kj} , $j = 1, 2, \dots, l$. Более того, оптимистичный пользователь надеется, что контроль масштабированной локальной оценки погрешности (1.8) позволит получить решение $\{x_k\}_{k=0}^K$ с точностью, которая более или менее соответствует заданному допуску погрешности Tol . К сожалению, последнее не всегда имеет место для большинства численных методов и ОДУ. Основная причина этого заключается в том, что локальные и глобальные ошибки согласованных численных схем, которые включают все формулы РК, слабо связаны и могут отличаться значительно. Например, как показано в [54]–[57], отношение “глобальная ошибка”/“допуск” явного и неявного решателей ОДУ в пакете MATLAB может превышать 10^4 даже для простых тестовых задач с достаточно высокими требованиями к точности вычислений.

Кроме того, в [58] дан пример жесткой задачи, для которого известный код RODAS дает для указанного отношения величину порядка 10^8 . В [56], [57] также показано, что эталонный решатель MATLAB для жестких задач ODE15s бесполезен для упомянутого примера. Таким образом, никакой стандартный локальный контроль точности не способен должным образом гарантировать глобальную погрешность используемых численных схем. Последнее привело к необходимости развития методов оценки и управления глобальной ошибкой в различных схемах численного интегрирования [59]–[65]. Современный прогресс в оценке глобальной ошибки в схемах РК в основном базируется на трех наиболее популярных методиках, т.е. на оценке погрешности с помощью псевдозадачи Задунайского, на использовании специальных формул для вычисления оценки глобальной ошибки и при использовании двух численных формул разных порядков схо-

димости (подробнее см. [47]–[49], [55]–[57], [66]–[73]). Здесь мы также можем упомянуть оценку глобальной погрешности, основанную на глобальной экстраполяции, т.е. когда два независимых численных интегрирования выполняются одним и тем же адаптивным методом, но для разных значений локального допуска To1 . Однако последний метод подразумевает, что глобальная ошибка представима в виде асимптотического ряда по степеням параметра To1 , т.е. этот метод работает в РК-схемах со свойством пропорциональности допуска (см. точное определение этого свойства и его ограничения в [74]–[77]). Тем не менее все перечисленные методы оценки глобальной ошибки довольно дорогостоящи. Обычно необходимо повторное численное интегрирование ЗК (1.1) на всем интервале $[t_0, t_{\text{end}}]$ для оценки глобальной ошибки, что, по крайней мере, вдвое увеличивает время выполнения таких расчетов с глобальным регулированием ошибки [63], [64]. Это являлось главным препятствием для широкого использования глобального контроля ошибки в используемых решателях ОДУ.

Недавно в [54] была представлена “дешевая” схема оценки глобальной ошибки, которая основана на простом суммировании оценок локальных ошибок (1.7). В этой работе также показано, что эта оценка ошибки может применяться в парах NIRK типа Гаусса (1.5). Конечно, представленный в [54] метод оценки глобальной ошибки не является асимптотически корректным, поскольку он только частично оценивает глобальную ошибку метода меньшего порядка в каждой встроенной паре, но он дешевле и достаточно хорошо работает на практике. С другой стороны, численные тесты, проведенные с жесткими примерами в [54], показывают, что упомянутая глобальная оценка ошибки может сильно завышать истинную ошибку получаемого решения. Последнее обстоятельство значительно снижает эффективность опубликованных решателей NIRK для жестких ОДУ. Для решения этой проблемы представим пары NIRK типа Гаусса и Лобатто с автоматическим управлением масштабированными модифицированными локальной и глобальной ошибками. В разд. 4 будет исследована эффективность предложенных численных методов решения жестких ОДУ и показано их превосходство над жесткими решателями ОДУ в пакете MATLAB с автоматическим контролем локальных ошибок. Подчеркнем, что эффективность любого практического метода численного интегрирования зависит как от точности и устойчивости применяемых формул дискретизации, так и от качества реализованного способа оценки погрешности и генерации сетки. Вопросы, связанные с построением и изучением формул типа Гаусса и Лобатто для NIRK, уже рассматривались и были опубликованы в [35], [37]. В настоящей работе основное внимание уделяется автоматической генерации эффективных сеток, которые позволяют вычислять приближенные решения жестких ОДУ, соответствующие установленным пользователем требованиям к их точности в автоматическом режиме.

2. МЕТОДЫ NIRK ТИПА ГАУССА С АВТОМАТИЧЕСКИМ УПРАВЛЕНИЕМ ЛОКАЛЬНОЙ И ГЛОБАЛЬНОЙ ОШИБКАМИ

Здесь мы обсуждаем две встроенные пары методов NIRK типа Гаусса. Первая состоит из формул РК порядков 2 и 4, а вторая – из встроенных методов NIRK порядков 4 и 6. Оба численных метода используют комбинированный локальный и глобальный механизм контроля ошибки численного решения и предназначены для решения различных ОДУ с требуемой точностью.

2.1. Встроенная пара NIRK порядков 2 и 4

Настоящий адаптивный вычислительный метод разработан, исследован и протестирован численно в [33]–[35], [37], [54]. Его таблица Бутчера имеет вид

$$\begin{array}{c|cccc}
 0 & 0 & 0 & 0 & 0 \\
 c_2 & \frac{6(c_2 + \theta) - 5}{12} & \frac{1 - \theta}{2} & \frac{1 - \theta}{2} & \frac{6(c_2 + \theta) - 7}{12} \\
 1 - c_2 & \frac{7 - 6(c_2 + \theta)}{12} & \frac{\theta}{2} & \frac{\theta}{2} & \frac{5 - 6(c_2 + \theta)}{12} \\
 1 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\
 \hline
 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\
 \hline
 & \frac{1}{2} & 0 & 0 & \frac{1}{2}
 \end{array} \quad (2.1)$$

Здесь фиксированный параметр $c_2 := (3 - \sqrt{3})/6$ и $\theta = 1/2 + 2\sqrt{3}/9$. Далее мы сделаем важные замечания о свойствах этого метода NIRK и особенностях его практической реализации.

Замечание 1. Параметр θ является свободным во встроенной паре NIRK (2.1) и может быть любым вещественным числом. Для любого θ метод более высокого порядка этой пары удовлетворяет упрощающим условиям $C(2)$ и $B(4)$ и имеет стадийный порядок 2 и классический порядок 4 (т.е. точно такие же, как и известный метод Гаусса порядка 4, представленный в [29, Table 5.1]). Кроме того, при $\theta = 1/2 + 2\sqrt{3}/9$ упомянутый метод NIRK дополнительно удовлетворяет упрощающему условию $C(3)$ и приобретает стадийный порядок 3 (см. доказательство в [33], [35], [37]). Напомним, что высокий стадийный порядок особенно важен для преодоления явления снижения порядка численного интегрирования, с которым можно столкнуться при решении очень жестких ОДУ [25], [29].

Замечание 2. Функция устойчивости формулы NIRK четвертого порядка в паре (2.1) является диагональной аппроксимацией Паде $R_{22}(z)$ независимо от θ . Поэтому этот метод (с любым параметром θ) эффективен для решения жестких ОДУ. Кроме того, он также подходит для решения очень жестких задач и дифференциально-алгебраических систем (см. дальнейшее объяснение в [25], [29]).

Замечание 3. Как следует из [28, Теорема 8.8], рассматриваемый метод является симметричным. Это свойство имеет решающее значение для его эффективности при численном интегрировании сингулярных краевых задач в [5]. Кроме того, это свойство делает указанный метод эффективным для решения обратимых ОДУ.

Замечание 4. Из [35] следует, что метод NIRK гауссовского типа и порядка 4 сопряжен с симплектическим методом, по крайней мере, до 6-го порядка включительно. Именно поэтому он эффективен для решения гамильтоновых задач на больших интервалах интегрирования, что подтверждено численным интегрированием задачи Кеплера (см. [34]).

Замечание 5. Встроенная пара NIRK (2.1) является неявной, поэтому ее практическая эффективность в значительной степени зависит от используемых итераций. Следующий упрощенный метод Ньютона рекомендован в [33], [37], [78]:

$$x_{k1}^{2,\ell-1} = a_{11}^2 \bar{x}_k + a_{12}^2 x_{k+1}^{\ell-1} + \tau_k [d_{11}^2 g(t_k, \bar{x}_k) + d_{12}^2 g(t_{k+1}, x_{k+1}^{\ell-1})], \quad (2.2a)$$

$$x_{k2}^{2,\ell-1} = a_{21}^2 \bar{x}_k + a_{22}^2 x_{k+1}^{\ell-1} + \tau_k [d_{21}^2 g(t_k, \bar{x}_k) + d_{22}^2 g(t_{k+1}, x_{k+1}^{\ell-1})], \quad (2.2b)$$

$$Q_2(\tau_k J_{k+1})(x_{k+1}^\ell - x_{k+1}^{\ell-1}) = -x_{k+1}^{\ell-1} + \bar{x}_k + \tau_k [b_1 g(t_{k1}^2, x_{k1}^{2,\ell-1}) + b_2 g(t_{k2}^2, x_{k2}^{2,\ell-1})], \quad (2.2в)$$

здесь $\ell = 1, 2, \dots, N$, с начальным приближением $x_{k+1}^0 = \bar{x}_k$, вычисленным в узле сетки $t_{k+1}^0 = t_k$, где \bar{x}_k соответствует численному решению ЗК (1.1), полученному за N итераций упрощенного метода Ньютона (2.2) в предыдущей точке t_k , т.е. $\bar{x}_k := x_k^N$, $k = 1, 2, \dots, K$. Частная производная (якобиан) $J_{k+1} := \partial_x g(t_{k+1}^0, x_{k+1}^0)$ правой части ОДУ (1.1) по переменной x вычисляется для начального значения x_{k+1}^0 в точке t_{k+1}^0 , и матрица $Q_2(\tau_k J_{k+1}) := (I_n - \tau_k J_{k+1}/4)^2$ является значением матричного полинома второй степени, в котором I_n обозначает единичную матрицу размера n . Отметим, что итерации (2.2) используют так называемую гнездовую форму (1.3) метода NIRK (2.1), а ее постоянные коэффициенты легко вычисляются для $\theta = 1/2 + 2\sqrt{3}/9$ по следующим формулам: $b_1 = b_2 := 1/2$, $c_1^2 := (3 - \sqrt{3})/6$, $c_2^2 := (3 + \sqrt{3})/6$, $a_{11}^2 = a_{22}^2 := 1/2 + 2\sqrt{3}/9$, $a_{12}^2 = a_{21}^2 := 1/2 - 2\sqrt{3}/9$, $d_{11}^2 = -d_{22}^2 := (3 + \sqrt{3})/36$, $d_{12}^2 = -d_{21}^2 := (-3 + \sqrt{3})/36$.

Подчеркнем, что представленная схема Ньютона (2.2) использует тривиальный предиктор, означающий, что численное решение \bar{x}_k с предыдущего шага метода NIRK используется в качестве его начального приближения. В этом случае $J_{k+1} := \partial_x g(t_k, \bar{x}_k)$. Далее, в [78] доказано, что двух итераций в каждой точке сетки достаточно для обеспечения сходимости порядка 4 при $\tau_k \rightarrow 0$. Кроме того, векторы $\bar{x}_{k+1}^N := x_{k+1}^N, x_{k1}^{2,N}, x_{k2}^{2,N}$ (заметим, что формулы (2.2a) и (2.2b) применяются еще один раз для вычисления стадийных величин, соответствующих искомому численному решению \bar{x}_{k+1}) и вектор \bar{x}_k , вычисленный на предыдущем шаге численного решения, могут быть использованы для построения полинома Эрмита третьей (или более высокой) степени, который можно применить для плотной выдачи результатов численного интегрирования с четвертым порядком точности. Кроме того, последний полином может быть использован для вычисления улучшенного начального приближения в итерациях (2.2). Тогда число итерационных шагов, требуемое для обеспечения сходимости четвертого порядка понижается до $N = 1$.

Также стоит отметить, что замена точного якобиана правой части формулы NIRK порядка 4 во встроенной паре (2.1) на ее аппроксимацию полиномом второй степени $Q_2(\tau_k J_{k+1})$ не влияет на устойчивость нашего метода и, следовательно, ее можно использовать для решения жестких

ОДУ [33], [78]. Кроме того, эффективное решение линейной задачи (2.2в) осуществляется в виде последовательного решения двух линейных систем с одной и той же матрицей коэффициентов $I_n - \tau_k J_{k+1}/4$.

Замечание 6. Схема оценки локальной ошибки (1.6), (1.7) для встроеной пары методов NIRK (2.1) принимает вид

$$le_{k+1} = x_{k+1}^{emb} - x_{k+1} = \frac{\tau_k}{2} [g(t_k, \bar{x}_k) - g(t_{k1}^2, x_{k1}^{2,N}) - g(t_{k2}^2, x_{k2}^{2,N}) + g(t_{k+1}, \bar{x}_{k+1})], \quad (2.3)$$

где векторы $\bar{x}_{k+1} := x_{k+1}^N, x_{k1}^{2,N}, x_{k2}^{2,N}$ вычисляются упрощенным методом Ньютона (2.2) и \bar{x}_k является численным решением, найденным на предыдущем шаге численного интегрирования. Затем оценка локальной ошибки le_{k+1} масштабируется по формуле (1.8) с $atol = rtol := Tol$. К сожалению, как показано в [34], [35], формула для оценки погрешности (2.3) может существенно переоценивать локальную ошибку для очень жестких ОДУ, поскольку функция устойчивости TR встроеного в метод NIRK (2.1) имеет вид

$$R_{emb}(z) = \frac{1 + 1/2z + 1/12z^2 + 1/12z^3}{1 - 1/2z + 1/12z^2}, \quad (2.4)$$

где $z := \lambda \tau_k$. Очевидно, что функция устойчивости (2.4) неограничена в полуплоскости комплексных чисел z с отрицательной вещественной частью.

Для устранения этого недостатка в цитируемых работах предлагается использовать модифицированную оценку локальной погрешности \tilde{le}_{k+1} , полученную в качестве решения линейной системы

$$Q_3(\tau_k J_{k+1}) \tilde{le}_{k+1} = le_{k+1}, \quad (2.5)$$

где матрица коэффициентов вычисляется с помощью матричного полинома третьей степени $Q_3(\tau_k J_{k+1}) := (I_n - \tau_k J_{k+1}/4)^3$. Отметим, что эта модифицированная оценка локальной ошибки довольно дешева, потому что линейная задача (2.5) подразумевает последовательное решение трех линейных систем с одной и той же матрицей коэффициентов $I_n - \tau_k J_{k+1}/4$. Кроме того, последняя матрица коэффициентов уже была вычислена и факторизована в итерациях (2.2). Положительным обстоятельством является то, что функция устойчивости, соответствующая модифицированной оценке локальной ошибки \tilde{le}_{k+1} , имеет вид

$$R_{mod}(z) = \frac{1 - 1/4z - 5/48z^2 + 19/192z^3 + 1/128z^4 - 1/768z^5}{1 - 5/4z + 31/48z^2 - 11/64z^3 + 3/128z^4 - 1/768z^5}, \quad (2.6)$$

где $z := \lambda \tau_k$ и $R_{mod}(z) := R_{NIRK4}(z) + Q_3^{-1}(z)[R_{emb}(z) - R_{NIRK4}(z)]$, а $R_{NIRK4}(z)$ соответствует функции устойчивости метода NIRK порядка 4 во встроеной паре (2.1), который мы обозначаем NIRK4 в этой статье. Отметим также, что $R_{NIRK4}(z) \equiv R_{22}(z)$ (см. замечание 2). Очевидно, что функция устойчивости (2.6) ограничена в левой комплексной полуплоскости и, следовательно, лучше подходит для управления локальной ошибкой в случае жесткой и очень жесткой ЗК (1.1). Исползованная идея модифицированной оценки локальной ошибки принадлежит Shampine (см. дальнейшее объяснение в [29, с. 123]). Затем управление локальной ошибкой реализуется, как описано во Введении.

Замечание 7. Оценка глобальной ошибки в паре NIRK (2.1) основана на простой идее из [54], т.е. истинная ошибка Δx_{k+1} встроеного решения x_{k+1}^{emb} в узле сетки t_{k+1} определяется по формуле

$$\Delta x_{k+1} = \Delta x_k - le_{k+1} \quad (2.7)$$

с начальной ошибкой интегрирования Δx_0 , равной нулю, так как начальное значение решения ОДУ (1.1) известно точно. В формуле (2.7) учтен знак локальной ошибки, т.е. формулы для оценки локальной ошибки, использованные здесь и в [54], имеет разные знаки. Последний член в (2.7) находится по формуле (2.3). Затем вычисленные оценки локальной и глобальной погрешностей совместно регулируются с помощью достаточно сложного механизма управления размером шага численного интегрирования, описанным в [54, Алгоритм 3.2]. Цитируемая статья показывает разумную производительность пары NIRK (2.1) с комбинированным локальным и глобальным контролем точности, а также показывает значительную переоценку ошибки этого метода для жестких ОДУ.

Причина завышения этой ошибки связана с формой функции устойчивости $R_{emb}(z)$ встроеного TR, заданной формулой (2.4). Кроме того, переоценка локальной ошибки, обсуждаемая в

замечании 6, усиливается формулой суммирования (2.7). Вот почему глобальный контроль точности, разработанный в [54], неэффективен для решения жестких ОДУ. Чтобы устранить этот недостаток, необходимо снова обратиться к идее Shampine и заменить исходную оценку локальной ошибки le_{k+1} , использованную в формуле (2.7), на ее модифицированную версию \tilde{le}_{k+1} , полученную с помощью решения линейной задачи (2.5). Другими словами, теперь мы оцениваем модифицированную глобальную ошибку $\Delta\tilde{x}_{k+1}$ и масштабируем ее следующим образом:

$$\Delta\tilde{x}_{k+1} = \Delta\tilde{x}_k - \tilde{le}_{k+1}, \quad |\Delta\tilde{x}_{k+1}|_{sc} := \max_{i=1,2,\dots,n} \{|\Delta\tilde{x}_{k+1}^i| / (\text{atol} + \text{rtol} \times |\bar{x}_{k+1}^i|)\}, \quad (2.8)$$

где верхний индекс i означает i -й элемент в соответствующем векторе, а два фиксированных параметра atol и rtol определяют вклад абсолютной и относительной глобальных ошибок в масштабированную глобальную погрешность соответственно. Масштабированная глобальная ошибка (2.8) проверяется на соответствие глобальному условию точности

$$|\Delta\tilde{x}_{k+1}|_{sc} \leq 1. \quad (2.9)$$

Для обеспечения выполнения условия точности (2.9) на практике мы используем алгоритм автоматического управления размером шага численного интегрирования, представленный в [54, Алгоритм 3.2], но сформулированный теперь для масштабированных модифицированных оценок локальной и глобальной ошибок, вычисленных в замечаниях 6 и 7. Насколько нам известно, идея масштабированного глобального контроля точности численного решения является довольно новой и впервые обсуждается в явных и неявных равнозначных блочных схемах [55], [56]. Эффективность пары NIRK гауссовского типа порядков 2 и 4 с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок будет продемонстрирована на жестких и крупномасштабных тестовых задачах и сравнена с жесткими встроенными решателями ОДУ пакета MATLAB в разд. 4.

2.2. Встроенная пара NIRK порядков 4 и 6

Этот адаптивный вычислительный метод разработан, изучен и протестирован численно в [35]–[37], [54]. Его таблица Бутчера имеет вид

$$\begin{array}{c|ccccccc}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 c_2 & \frac{c_3}{6} & 0 & 0 & \frac{35-40c_3}{108} & \frac{56-64c_3}{108} & \frac{35-40c_3}{108} & \frac{c_3-1}{6} \\
 c_3 & \frac{c_2}{6} & 0 & 0 & \frac{35-40c_2}{108} & \frac{56-64c_2}{108} & \frac{35-40c_2}{108} & \frac{c_2-1}{6} \\
 c_4 & \frac{20c_6-3}{200} & \frac{18c_3-9}{100} + \theta & \theta & \frac{v_1-5\theta}{360} & \frac{v_1-8\theta}{225} & \frac{v_1-5\theta}{360} & \frac{3-20c_4}{200} \\
 c_5 & \frac{1}{32} & \frac{\sqrt{27}}{32} & -\frac{\sqrt{27}}{32} & \frac{5}{36} & \frac{2}{9} & \frac{5}{36} & -\frac{1}{32} \\
 c_6 & \frac{20c_4-3}{200} & -\theta & \frac{18c_2-9}{100} - \theta & \frac{v_2+5\theta}{360} & \frac{v_2+8\theta}{225} & \frac{v_2+5\theta}{360} & \frac{3-20c_6}{200} \\
 1 & 0 & 0 & 0 & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} & 0 \\
 \hline
 & 0 & 0 & 0 & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} & 0 \\
 \hline
 & \frac{1}{6} & 0 & 0 & 0 & \frac{2}{3} & 0 & \frac{1}{6}
 \end{array}, \quad (2.10)$$

где $c_2 := (3 - \sqrt{3})/6$, $c_3 := (3 + \sqrt{3})/6$, $c_4 := (5 - \sqrt{15})/10$, $c_5 := 1/2$, $c_6 := (5 + \sqrt{15})/10$, $v_1 := 120c_4 - 18c_3 - 1$, $v_2 := 120c_6 - 18c_2 - 1$ и $\theta = (36c_6 - 18c_3 - 9)/200$.

Ниже мы суммируем свойства метода (2.10) и особенности его практической реализации в виде следующих замечаний.

Замечание 8. Параметр θ является свободным во встроенной паре NIRK (2.10) и может быть любым вещественным числом. Для любого θ метод более высокого порядка этой пары удовлетворяет упрощающим условиям $C(3)$ и $B(6)$. В [35] также доказано, что он имеет стадийный порядок 3 и классический порядок 6

(т.е. точно такой же, как метод Гаусса порядка 6, представленный в [29, табл. 5.2]). Кроме того, при $\theta = (36c_6 - 18c_3 - 9)/200$ упомянутый метод NIRK дополнительно удовлетворяет упрощающему условию $C(3, 5)$, которое означает, что его стадийные величины третьего уровня удовлетворяют упрощающему условию $C(5)$ (см. подробности в [35]). Поэтому их можно использовать для плотной выдачи результатов численного интегрирования с шестым порядком точности.

Замечание 9. Функция устойчивости формулы NIRK шестого порядка в таблице (2.10) является диагональной аппроксимацией Паде $R_{33}(z)$ в независимости от θ . Таким образом, этот метод NIRK (с любым θ) эффективен для решения жестких ОДУ. Кроме того, он также жестко точен и, следовательно, подходит для решения очень жестких задач и дифференциально-алгебраических систем (см. дальнейшее объяснение в [25], [29]).

Замечание 10. Переставив строки в таблице (2.10) так, чтобы коэффициенты c_i монотонно возрастали, из [28, Теорема 8.8] следует, что этот метод симметричен. Таким образом, он подходит для решения обратимых 3К вида (1.1).

Замечание 11. В замечании 8 сказано, что формула NIRK порядка 6 во встроенной паре (2.10) по своим свойствам аналогична методу Гаусса порядка 6, но значительно “дешевле” последнего. Поэтому есть надежда, что она будет эффективна для решения гамильтоновых уравнений на больших интервалах интегрирования. Последнее было подтверждено численно интегрированием задачи Кеплера на интервале $[0, 10^6]$ в [36].

Замечание 12. Встроенная пара NIRK (2.10) является неявной и, следовательно, ее эффективность в значительной степени зависит от используемой итерации. Следующий упрощенный метод Ньютона разработан в [36], [37] для реализации этого вычислительного инструмента:

$$x_{kj}^{2,\ell-1} = a_{j1}^2 \bar{x}_k + a_{j2}^2 x_{k+1}^{\ell-1} + \tau_k [d_{j1}^2 g(t_k, \bar{x}_k) + d_{j2}^2 g(t_{k+1}, x_{k+1}^{\ell-1})], \quad j = 1, 2, \quad (2.11a)$$

$$x_{kj}^{3,\ell-1} = a_{j1}^3 \bar{x}_k + a_{j2}^3 x_{k+1}^{\ell-1} + \tau_k [d_{j1}^3 g(t_k, \bar{x}_k) + d_{j2}^3 g(t_{k+1}, x_{k+1}^{\ell-1})] + \tau_k \sum_{m=1}^2 d_{j,m+2}^3 g(t_{km}^2, x_{km}^{2,\ell-1}), \quad j = 1, 2, 3, \quad (2.11б)$$

$$Q_3(\tau_k J_{k+1})(x_{k+1}^\ell - x_{k+1}^{\ell-1}) = -x_{k+1}^{\ell-1} + \bar{x}_k + \tau_k \sum_{i=1}^3 b_i g(t_{ki}^3, x_{ki}^{3,\ell-1}), \quad \ell = 1, 2, \dots, N, \quad (2.11в)$$

с начальным приближением $x_{k+1}^0 = \bar{x}_k$, вычисленным в узле сетки $t_{k+1}^0 = t_k$, где \bar{x}_k соответствует численному решению 3К (1.1), полученному за N итераций упрощенного метода Ньютона (2.10) в предыдущей точке t_k , т.е. $\bar{x}_k := x_k^N$, $k = 1, 2, \dots, K$, с якобианом $J_{k+1} := \partial_x g(t_{k+1}, x_{k+1}^0)$ и матрицей $Q_3(\tau_k J_{k+1}) := (I_n - \tau_k J_{k+1}/6)^3$, являющейся значением матричного полинома третьей степени. Итерации (2.11) эксплуатируют гнездовое представление (1.3) метода (2.10), в котором коэффициенты легко находятся для $\theta = (36c_6 - 18c_3 - 9)/200$ в следующем виде: $b_1 = b_3 := 5/18$, $b_2 := 4/9$, $c_1^2 := (3 - \sqrt{3})/6$, $c_2^2 := (3 + \sqrt{3})/6$, $a_{11}^2 = a_{22}^2 := 1/2 + 2\sqrt{3}/9$, $a_{12}^2 = a_{21}^2 := 1/2 - 2\sqrt{3}/9$, $d_{11}^2 = -d_{22}^2 := (3 + \sqrt{3})/36$, $d_{12}^2 = -d_{21}^2 := (-3 + \sqrt{3})/36$, $c_1^3 := (5 - \sqrt{15})/10$, $c_2^3 := 1/2$, $c_3^3 := (5 + \sqrt{15})/10$, $a_{11}^3 = a_{32}^3 := (125 + 39\sqrt{15})/250$, $a_{12}^3 = a_{31}^3 := (125 - 39\sqrt{15})/250$, $a_{21}^3 = a_{22}^3 := 1/2$, $d_{11}^3 = -d_{32}^3 := (7 + 2\sqrt{15})/200$, $d_{12}^3 = -d_{31}^3 := (-7 + 2\sqrt{15})/200$, $d_{13}^3 = -d_{34}^3 := (18\sqrt{15} + 15\sqrt{3})/1000$, $d_{14}^3 = -d_{33}^3 := (18\sqrt{15} - 15\sqrt{3})/1000$, $d_{21}^3 = -d_{22}^3 := 1/32$, $d_{23}^3 = -d_{24}^3 := 3\sqrt{3}/32$.

Как и в приведенной выше встроенной паре NIRK порядков 2 и 4, упрощенная схема Ньютона (2.11) тоже применяет тривиальный предиктор, означающий, что численное решение \bar{x}_k с предыдущего шага метода NIRK используется в качестве его начального приближения. В этом случае $J_{k+1} := \partial_x g(t_k, \bar{x}_k)$. Далее, в [36] доказано, что трех итераций в каждой точке сетки достаточно для обеспечения сходимости максимального шестого порядка при $\tau_k \rightarrow 0$. Кроме того, векторы $\bar{x}_{k+1}^N := x_{k+1}^N, x_{k1}^{3,N}, x_{k2}^{3,N}, x_{k3}^{3,N}$ (заметим, что формулы (2.11a) и (2.11б) применяются еще один раз для вычисления стадийных величин, соответствующих искомому численному решению \bar{x}_{k+1}) и вектор \bar{x}_k , вычисленный на предыдущем шаге численного решения, могут быть использованы для построения полинома Эрмита пятой (или более высокой) степени, который можно применить для плотной выдачи результатов численного интегрирования с шестым порядком точности. Кроме того, последний полином может быть использован для вычисления улучшенного начального приближения в итерациях (2.11). Тогда число итерационных шагов, требуемое для обеспечения сходимости шестого порядка, понижается до $N = 1$ при $\tau_k \rightarrow 0$. Последнее вытекает из [78, Теорема 2.3].

Также стоит отметить, что замена точного якобиана правой части формулы NIRK шестого порядка во встроенной паре (2.10) на его аппроксимацию полиномом третьей степени $Q_3(\tau_k J_{k+1})$ приемлемо при решении жестких ОДУ [36]. В этом случае эффективное решение линейной задачи (2.11в) осуществляется в виде последовательного решения трех линейных систем с одной и той же матрицей коэффициентов $I_n - \tau_k J_{k+1}/6$.

Замечание 13. Общая методика оценки локальной ошибки (1.6), (1.7), реализованная во встроенной паре (2.10), приобретает вид

$$le_{k+1} = \frac{\tau_k}{3} \left[\frac{1}{2} g(t_k, \bar{x}_k) - \frac{5}{6} g(t_{k1}^3, x_{k1}^{3,N}) + \frac{2}{3} g(t_{k2}^3, x_{k2}^{3,N}) - \frac{5}{6} g(t_{k3}^3, x_{k3}^{3,N}) + \frac{1}{2} g(t_{k+1}, \bar{x}_{k+1}) \right], \tag{2.12}$$

где векторы $\bar{x}_{k+1} := x_{k+1}^N, x_{k1}^{3,N}, x_{k2}^{3,N}, x_{k3}^{3,N}$ были уже вычислены с помощью итераций (2.11), а \bar{x}_k – численное решение, полученное на предыдущем шаге этой пары NIRK типа Гаусса. Затем оценка локальной ошибки le_{k+1} масштабируется по формуле (1.8) с $atol = rtol := Tol$. Опять же, легко видеть, что формула (2.12) может сильно переоценить локальную ошибку для очень жестких ОДУ, потому что функция устойчивости встроенной формулы РК в паре NIRK (2.10) имеет вид

$$R_{emb}(z) = \frac{1 + 1/2z + 1/10z^2 + 1/120z^3 + 1/2880z^5}{1 - 1/2z + 1/10z^2 - 1/120z^3}, \tag{2.13}$$

где $z := \lambda \tau_k$. Очевидно, что функция устойчивости (2.13) неограничена в комплексной полуплоскости чисел z с отрицательной вещественной частью.

Опять же, мы используем идею Shampine и вычисляем оценку модифицированной локальной ошибки \tilde{le}_{k+1} , решая линейную систему

$$Q_2(\tau_k J_{k+1}) \tilde{le}_{k+1} = le_{k+1}, \tag{2.14}$$

матрица коэффициентов которой определяется матричным многочленом второй степени $Q_2(\tau_k J_{k+1}) := (I_n - \tau_k J_{k+1}/6)^2$. Подчеркнем, что модифицированная оценка локальной погрешности достаточно дешева, поскольку линейная задача (2.14) подразумевает последовательное решение двух линейных систем с одной и той же матрицей коэффициентов $I_n - \tau_k J_{k+1}/6$. Кроме того, последняя матрица коэффициентов уже была вычислена и факторизована в итерациях (2.11). Теперь функция устойчивости, соответствующая модифицированной оценке локальной ошибки \tilde{le}_{k+1} , приобретает вид

$$R_{mod}(z) = \frac{1 + 1/6z - 7/180z^2 - 1/90z^3 + 1/1728z^5}{1 - 5/6z + 53/180z^2 - 1/8z^3 + 1/180z^4 - 1/4320z^5}, \tag{2.15}$$

где $z := \lambda \tau_k$ и $R_{mod}(z) := R_{NIRK6}(z) + Q_2^{-1}(z)[R_{emb}(z) - R_{NIRK6}(z)]$, а $R_{NIRK6}(z)$ – функции устойчивости метода NIRK шестого порядка во встроенной паре (2.10), которая далее обозначается NIRK6.

Из замечания 9 следует, что $R_{NIRK6}(z) \equiv R_{33}(z)$. Очевидно, что функция устойчивости (2.15) ограничена в левой комплексной полуплоскости и, следовательно, лучше соответствует управлению локальной ошибкой в жестких и очень жестких ОДУ (1.1). Затем управление локальной ошибкой осуществляется так же, как описано в разд. 1.

Замечание 14. Как уже говорилось в замечании 7, причина переоценки глобальной ошибки, наблюдаемая для встроенной пары (2.10) с управлением глобальной ошибкой на основе методики из [54], связана с формой функции устойчивости $R_{emb}(z)$ (2.13) встроенной квадратуры Лобатто порядка 4, используемой в формуле оценки погрешности (2.12). Кроме того, эта переоценка локальной ошибки усиливается формулой суммирования (2.7). Поэтому контроль глобальной ошибки, разработанный в [54], неэффективен для жестких ОДУ. Чтобы устранить этот недостаток, мы следуем идее Shampine и заменяем исходную оценку локальной ошибки le_{k+1} в оценке глобальной погрешности (2.7) на значение модифицированной оценки локальной погрешности \tilde{le}_{k+1} , полученное с помощью решения линейной задачи (2.14). Другими словами, теперь мы оцениваем модифицированную глобальную ошибку по формуле

$$\Delta \tilde{x}_{k+1} = \Delta \tilde{x}_k - \tilde{le}_{k+1}, \tag{2.16}$$

которая масштабируется в виде:

$$|\Delta \tilde{x}_{k+1}|_{sc} := \max_{i=1,2,\dots,n} \left\{ |\Delta \tilde{x}_{k+1}^i| / \left(atol + rtol \times |\bar{x}_{k+1}^i| \right) \right\}, \tag{2.17}$$

где верхний индекс i означает i -й элемент в соответствующем векторе, а два фиксированных параметра $atol$ и $rtol$ определяют вклад абсолютной и относительной глобальных ошибок в масштабированную глобальную погрешность соответственно. Напомним, что в данной работе изучается механизм управления глобальной ошибкой, удовлетворяющий условию $atol = rtol := Tol$. Масштабированная модифицированная оценка глобальной погрешности, полученная при помощи формул (2.16) и (2.17) должна удовлетворять условию глобальной точности (2.9). Последнее обеспечивается с помощью автоматического управления размером шага численного интегрирования, разработанного в [54, Алгоритм 3.2], но сформулированного теперь для масштабированных модифицированных оценок локальной и глобальной ошибок, вычисленных в замечаниях 13 и 14 соответственно. Эффективность пары NIRK гауссовского типа порядков 4 и 6 с представленным выше управлением масштабированными модифицированными оценками локальной и глобальной ошибок будет продемонстрирована на жестких и крупномасштабных тестовых задачах в сравнении с жесткими встроенными решателями ОДУ пакета MATLAB в разд. 4.

3. МЕТОДЫ NIRK ТИПА ЛОБАТТО С АВТОМАТИЧЕСКИМ УПРАВЛЕНИЕМ ЛОКАЛЬНОЙ И ГЛОБАЛЬНОЙ ОШИБКАМИ

Рассмотрим одну встроенную пару NIRK, основанную на двух формулах Лобатто порядков 2 и 4. Подробное описание этого инструмента численного интегрирования представлено в [37]. Приведенная пара NIRK типа Лобатто задается следующей таблицей Бутчера:

$$\begin{array}{c|ccc}
 0 & 0 & 0 & 0 \\
 \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\
 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
 \hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
 \hline
 & \frac{1}{2} & 0 & \frac{1}{2}
 \end{array} . \tag{3.1}$$

Далее мы суммируем наиболее важные свойства метода (3.1) и особенности его практической реализации в виде следующих замечаний.

Замечание 15. Сопоставив коэффициенты таблицы метода (3.1) с коэффициентами метода Лобатто порядка 4, приведенными в [29, табл. 5.7], приходим к выводу, что формула NIRK четвертого порядка во встроенной паре (3.1) является формулой Лобатто IIIA порядка 4. Именно поэтому она удовлетворяет упрощающим условиям $C(3)$ и $B(4)$ и, следовательно, эта формула NIRK имеет стадийный порядок 3 и классический порядок 4.

Замечание 16. Хорошо известно, что функция устойчивости формулы РК более высокого порядка в паре NIRK (3.1) является диагональной аппроксимацией Паде $R_{22}(z)$ (см., например, [26, лемма 3.4.5]). Поэтому этот метод может быть использован для решения жестких ОДУ. Кроме того, он также жестко точен и поэтому подходит для решения очень жестких задач и дифференциально-алгебраических систем (см. дальнейшее объяснение в [25], [29]).

Замечание 17. Из [28, Теорема 8.8] следует, что рассматриваемый метод симметричен. Поэтому он может быть эффективен для решения обратимых ОДУ.

Замечание 18. В [35, Лемма 2] следует, что метод NIRK типа Лобатто и порядка 4 сопряжен с симплектическим методом, по крайней мере, до 6-го порядка включительно. Именно поэтому он подходит для решения гамильтоновых задач на больших интервалах интегрирования. Последнее было подтверждено численно интегрированием задачи Кеплера на интервале $[0, 10^5]$ в [37].

Замечание 19. Встроенная пара NIRK (3.1) является неявной, поэтому ее практическая эффективность в значительной степени зависит от реализованной итерации. Кроме того, учитывая ее гнездовую структуру, для реализации численной формулы (3.1) в [37] рекомендован следующий упрощенный метод Ньютона:

$$x_{k1}^{2,\ell-1} = a_{11}^2 \bar{x}_k + a_{12}^2 x_{k+1}^{\ell-1} + \tau_k [d_{11}^2 g(t_k, \bar{x}_k) + d_{12}^2 g(t_{k+1}, x_{k+1}^{\ell-1})], \tag{3.2a}$$

$$Q_2(\tau_k J_{k+1})(x_{k+1}^\ell - x_{k+1}^{\ell-1}) = -x_{k+1}^{\ell-1} + \bar{x}_k + \tau_k [b_1 g(t_k, \bar{x}_k) + b_2 g(t_{k1}^2, x_{k1}^{2,\ell-1}) + b_3 g(t_{k+1}, x_{k+1}^{\ell-1})], \tag{3.2b}$$

$\ell = 1, 2, \dots, N$, с начальным приближением $x_{k+1}^0 = \bar{x}_k$, вычисленным в узле сетки $t_{k+1}^0 = t_k$, где \bar{x}_k соответствует численному решению ЗК (1.1), полученному за N итераций упрощенного метода Ньютона (3.2) в предыдущей точке t_k , т.е. $\bar{x}_k := x_k^N$, $k = 1, 2, \dots, K$, с якобианом $J_{k+1} := \partial_{xg}(t_{k+1}^0, x_{k+1}^0)$ и матрицей

$Q_2(\tau_k J_{k+1}) := (I_n - \tau_k J_{k+1}/4)^2$, являющейся значением матричного полинома третьей степени. Отметим, что итерации (3.2) показывают, что метод Лобатто IIIA порядка 4 допускает гнездовое представление вида (1.3) со следующими постоянными коэффициентами: $b_1 = b_3 := 1/6$, $b_2 := 2/3$, $c_1^2 := 1/2$, $a_{11}^2 = a_{12}^2 := 1/2$, $d_{11}^2 = -d_{12}^2 := 1/8$.

Опять же, упрощенный метод Ньютона (3.2) применяет тривиальный предиктор, означающий, что численное решение \bar{x}_k с предыдущего шага метода NIRK используется в качестве его начального приближения. В этом случае $J_{k+1} := \partial_x g(t_k, \bar{x}_k)$. Кроме того, формулы NIRK типа Гаусса и Лобатто и порядка 4 используют одну и ту же аппроксимацию якобиана $Q_2(\tau_k J_{k+1})$. Поэтому замечание 5 доказывает, что двух итераций достаточно для обеспечения сходимости с максимальным порядком 4 при $\tau_k \rightarrow 0$. Далее, векторы $\bar{x}_{k+1} := x_{k+1}^N, x_{k+1}^{2,N}$ (напомним, что формула (3.2a) применяется еще один раз для вычисления стадийной величины, соответствующей искомому численному решению \bar{x}_{k+1}) и вектор \bar{x}_k , вычисленный на предыдущем шаге численного решения, могут быть использованы для построения полинома Эрмита третьей (или более высокой) степени, который можно применить для плотной выдачи результатов численного интегрирования с четвертым порядком точности. Кроме того, последний полином может быть использован для вычисления улучшенного начального приближения в итерациях (3.2). Тогда число итерационных шагов, требуемое для обеспечения сходимости четвертого порядка, понижается до $N = 1$, что следует из [78, Теоремы 2.3].

Отметим, что замена точного якобиана правой части метода Лобатто IIIA на его аппроксимацию матричным полиномом второй степени $Q_2(\tau_k J_{k+1})$ подходит для решения жестких ОДУ, поскольку функция устойчивости итерации (3.2) с тривиальным предиктором и $N = 1$ задается формулой

$$R(z) = \frac{(1 + 1/4z)^2}{(1 - 1/4z)^2}, \quad (3.3)$$

где $z := \lambda \tau_k$. Функция устойчивости (3.3) очевидно является A -устойчивой. Кроме того, напомним, что эффективное решение линейной задачи (3.2б) осуществляется в виде последовательно-го решения двух линейных систем с одной и той же матрицей коэффициентов $I_n - \tau_k J_{k+1}/4$.

Замечание 20. Общая схема оценки локальной ошибки (1.6), (1.7), реализованная во встроенной паре NIRK типа Лобатто (3.1), приводит к формуле оценки ошибки

$$le_{k+1} = x_{k+1}^{\text{emb}} - x_{k+1} = \frac{\tau_k}{3} [g(t_k, \bar{x}_k) - 2g(t_{k1}, x_{k1}^{2,N}) + g(t_{k+1}, \bar{x}_{k+1})], \quad (3.4)$$

где векторы $\bar{x}_{k+1} := x_{k+1}^N$ и $x_{k1}^{2,N}$ были уже вычислены с помощью итераций (3.2), а \bar{x}_k – численное решение, полученное на предыдущем шаге этой пары NIRK типа Гаусса. Затем оценка локальной ошибки le_{k+1} масштабируется по формуле (1.8) с $\text{atol} = \text{rtol} := \text{ToI}$. Опять же, формула оценки ошибки (3.4) может сильно завывать локальную ошибку для очень жестких ОДУ, поскольку функция устойчивости встроенного TR в паре NIRK (3.1) принимает вид

$$R_{\text{emb}}(z) = \frac{1 + 1/2z + 1/12z^2 + 1/12z^3}{1 - 1/2z + 1/12z^2}. \quad (3.5)$$

Очевидно, что функция устойчивости (3.5) неограничена в комплексной полуплоскости чисел z с отрицательной вещественной частью. С другой стороны, эта функция устойчивости и функция устойчивости (2.4) встроенного TR в паре NIRK гауссовского типа (2.1) совпадают. Таким образом, модифицированная схема оценки локальной погрешности, представленная в замечании 6, также применима для преобразования оценки ошибки (3.4). Затем мы просто заменяем оценку локальной ошибки le_{k+1} ее модифицированным значением \tilde{le}_{k+1} , найденным путем решения линейной задачи (2.5). Замечание 6 объясняет, что это преобразование устраняет отмеченный недостаток и делает обсуждаемый контроль локальной ошибки подходящим для жестких ОДУ.

Замечание 21. Оценка и управление глобальной погрешностью во встроенной паре NIRK типа Лобатто (3.1) выполняются точно так же, как и в паре NIRK типа Гаусса (2.1) (см. замечание 7). Заметим, что все теоретические условия, лежащие в основе дешевого метода оценки глобальной ошибки, разработанного в [54], справедливы и для встроенной пары (3.1). Эффективность пары NIRK типа Лобатто по-

рядков 2 и 4 с представленным выше управлением масштабированными модифицированными оценками локальной и глобальной ошибок будет продемонстрирована на жестких и крупномасштабных тестовых задачах в сравнении с жесткими встроенными решателями ОДУ пакета MATLAB в следующем разделе.

4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Выполним ряд численных тестов для изучения трех пар NIRK, представленных в разд. 2 и 3, на жестких и крупномасштабных задачах. Сначала мы проверим их способность находить численные решения с заданной точностью. Требуемая точность вычислений в наших экспериментах определяется допуском на погрешность To1 . Другими словами, мы хотим, чтобы масштабированная глобальная погрешность найденного численного решения не превышала эту границу. Такая информация представлена на приведенных ниже фигурах для точности интегрирования. Для оценки эффективности наших адаптивных решателей ОДУ мы также изучим точность, достигнутую каждым методом, в зависимости от времени его работы. На практике нас интересуют инструменты численного интегрирования, которые обеспечивают более высокую точность интегрирования за более короткое время работы. Такие данные представлены на фигурах для эффективности численного решения. Отметим, что наши результаты (т.е. допущенные ошибки численного интегрирования) показаны в сравнении с используемыми допусками To1 и затраченным временем процессора в двойном логарифмическом масштабе.

На представленных фигурах используются следующие сокращения для NIRK: пара NIRK типа Гаусса (2.1) с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок отмечена как NIRK4 (2) g; пара NIRK типа Гаусса (2.10) с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок показана аббревиатурой NIRK6 (4) g; пара NIRK типа Лобатто (3.1) с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок указана как NIRK4 (2) l. Мы сравним также точность и эффективность наших решателей NIRK с жесткими встроенными кодами ODE15s, ODE23s и ODE23tb пакета MATLAB. Для корректности сравнения все вычисления выполнены в MATLAB.

Дополнительно следует иметь в виду, что для повышения эффективности разработанных кодов NIRK все упрощенные итерации Ньютона, реализованные в решателях NIRK, используют интерполяционный полином второй степени, построенный по значениям численного решения, найденного в двух предыдущих узлах сетки. Этот полином применяется для более точного предсказания начального приближения. Поэтому первые два шага в каждом коде NIRK выполняются с тривиальным предиктором, но с увеличенным числом допустимых итераций. Кроме того, каждый упрощенный метод Ньютона использует переменное число итераций в узлах сетки, начиная с третьего шага в каждом решателе NIRK. Другими словами, мы должны решить две задачи. Первая из них – убедиться, что численные решения \bar{x}_k , полученные парами NIRK типа Гаусса и Лобатто (2.1) и (3.1), сходятся с порядком 4 (или численное решение \bar{x}_k из пары NIRK типа Гаусса (2.10) сходится с порядком 6) при $\tau_k \rightarrow 0$. Вторая задача состоит в том, чтобы представить некоторые аргументы в пользу того, что итерационные ошибки, возникающие в примененных итерациях, незначительны по сравнению с ошибками дискретизации, имеющими место в использованных схемах NIRK. Для достижения обеих целей, начиная с третьего шага в каждом коде NIRK выполняется следующий расчет. Мы выполняем одну итерацию в формулах NIRK четвертого порядка (или три итерации в методе шестого порядка) без контроля точности. Это должно гарантировать, что найденное численное решение сходится с предопределенным теоретическим порядком. Затем мы начинаем мониторить невязку следующей итерации и так далее. Таким образом, вычислив очередную итерацию x_{k+1}^ℓ , мы проверяем ее точность с помощью критерия остановки

$$\max_{i=1, \dots, n} \{ |(x_{k+1}^\ell)^i - (x_{k+1}^{\ell-1})^i| / (1 + |(x_{k+1}^\ell)^i|) \} \leq \varepsilon_l / 10, \quad (4.1)$$

где верхний индекс i означает i -й элемент в соответствующем векторе, а ε_l является верхней границей для масштабированных оценок локальных ошибок, допустимых в нашем механизме управления размером шага численного интегрирования, представленном в [54, Алгоритм 3.2]. Критерий остановки (4.1) подразумевает, что масштабированная невязка вычисленного итерационного приближения должна быть в 10 раз меньше максимально допустимой масштабированной оценки локальной ошибки в соответствующем методе NIRK. Если этот критерий выполняется для некоторой итерации x_{k+1}^ℓ с разрешенным номером итерации ℓ , то мы останавливаем ите-

рацию и возвращаем итоговое решение $\bar{x}_{k+1} = x_{k+1}^\ell$ с первым надстрочным индексом ℓ , для которого выполнено условие (4.1). Однако, если условие остановки (4.1) не выполняется для 20 последовательных итераций x_{k+1}^ℓ , то последняя итерация $\bar{x}_{k+1} = x_{k+1}^{21}$ будет принята как итерационное решение в паре NIRK четвертого порядка (или $\bar{x}_{k+1} = x_{k+1}^{23}$ будет принято как итерационное решение в методе шестого порядка), а затем реализованные средства управления локальной и глобальной ошибками выяснят, является ли это решение достаточно точным.

Наш первый жесткий пример довольно прост и взят из [46]. Он сформулирован в виде следующего ОДУ.

Задача I:

$$\begin{aligned}x_1'(t) &= \lambda[\cos^2(t)\sin(t) + 2\cos(t) - (2 + x_1(t)x_2(t))x_1(t)] - x_2(t), \\x_2'(t) &= x_1(t) + x_2(t) - \sin(t)\end{aligned}$$

с начальным значением $x(0) = (1, 0)^\top$. Задача I решается на интервале $[0, 5]$. Ее жесткость определяется фиксированным параметром λ , который далее выбирается равным $\lambda = 10^6$. Это соответствует жесткому поведению приведенного тестового примера. Задача I имеет точное решение в замкнутой форме

$$x_1(t) = \cos(t), \quad x_2(t) = \sin(t), \quad (4.2)$$

которое не зависит от жесткости. Это точное решение используется для оценки максимальной масштабированной глобальной ошибки, допущенной рассматриваемыми кодами на сетках, генерируемых в автоматическом режиме, по формуле

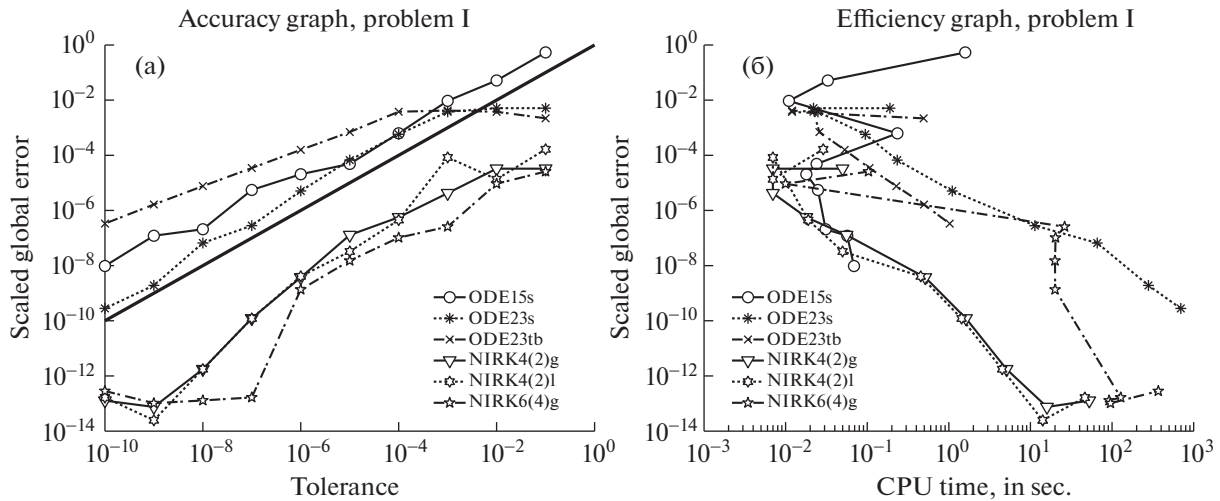
$$|\Delta x(t)|_{sc} := \max_{k=1,2,\dots,K} \max_{i=1,2,\dots,n} \{|x_{\text{exact}}^i(t_k) - x_k^i| / (1 + |x_{\text{exact}}^i(t_k)|)\}, \quad (4.3)$$

где $x_{\text{exact}}(t_k) := (x_1(t_k), x_2(t_k))^\top$ — значение точного вектора решения с его компонентами, вычисленными по формулам (4.2) в узле сетки t_k , x_k — это численное решение, полученное соответствующей парой NIRK (или решателем MATLAB) в той же самой точке сетки t_k , K — это количество узлов в сгенерированной сетке (оно специфично для каждого рассматриваемого кода), n — размер Задачи I (т.е. $n = 2$ в данном конкретном случае), а верхний индекс i означает i -ю компоненту в соответствующем векторе.

В первом эксперименте мы установили допуски $\text{Tol} := 10^{-1}, 10^{-2}, \dots, 10^{-10}$ на ошибку численного интегрирования во всех рассматриваемых кодах. Подчеркнем, что параметр точности ε_g , используемый при выборе размера шага в [54, Алгоритм 3.2], который реализован во всех парах NIRK, здесь обозначается как Tol . Решатели ОДУ ODE15s, ODE23s и ODE23tb в пакете MATLAB применяются с опциями $\text{AbsTol} := \text{Tol}$, $\text{RelTol} := \text{Tol}$ и $\text{MaxStep} := 10^{-1}$. Такое же ограничение на максимальный размер шага интегрирования установлено в наших парах NIRK с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок.

Задача I решена численно как кодами NIRK NIRK4 (2) g, NIRK6 (4) g и NIRK4 (2) l, так и с помощью встроенных в MATLAB решателей ODE15s, ODE23s и ODE23tb с вышеуказанными ограничениями на ошибку. Затем допущенная максимальная масштабированная глобальная ошибка вычислена по формуле (4.3) для каждого исследуемого решателя, и эти ошибки показаны в сравнении с используемыми допусками Tol и процессорным временем на фиг. 1. На левом графике представлено исследование точности этих решателей ОДУ, а на правой диаграмме показана их эффективность.

На фиг. 1а показано, что точность, достигнутая парами NIRK, лучше согласуется с нашими требованиями, представленными на каждом графике точности в виде жирной прямой линии, соответствующей условию “Масштабированная Глобальная Ошибка” = “Допуск” особенно при высоких требованиях к точности вычислений. В этой статье мы хотим получить численные решения, масштабированные глобальные ошибки которых (4.3) лежат ниже этой линии. В целом решатели NIRK удовлетворяют наш запрос в отличие от встроенных кодов пакета MATLAB. Это неудивительно, потому что локальный контроль ошибки, реализованный в ODE15s, ODE23s и ODE23tb, не может должным образом регулировать их глобальные ошибки. Наиболее точное



Фиг. 1. Диаграммы точности и эффективности для жестких решателей ОДУ с локальным контролем ошибки из пакета MATLAB и для трех пар NIRK с масштабированным модифицированным локальным и глобальным контролем ошибки, протестированных на задаче I.

численное интегрирование среди рассмотренных кодов MATLAB выполняется с помощью ODE23s. С другой стороны, из диаграммы эффективности фиг. 1 следует, что это наименее эффективный ОДУ-решатель для задачи I. Среди кодов NIRK наиболее эффективными являются NIRK4 (2) g и NIRK4 (2) l, которые на удивление так же эффективны, как ODE15s (этот решатель ОДУ переменного порядка в пакете MATLAB многими практиками признается эталоном для интегрирования жестких задач) и даже превосходят пару NIRK NIRK6 (4) g более высокого порядка. Последнее объясняется простотой задачи I для высокоточного численного интегрирования и тем, что метод шестого порядка выполняет более крупные шаги по сравнению с методами четвертого порядка. Это требует большего числа повторных циклов интегрирования задачи I, начиная с начального узла сетки t_0 , кодом NIRK6 (4) g. Чтобы увидеть преимущества нашего кода NIRK более высокого порядка, нам потребуются более сложные тестовые примеры, рассмотренные ниже.

Следующей жесткой системой ОДУ, использованной в этой статье, является известный осциллятор Ван-дер-Поля [29, с. 144]. Он индуцирует устойчивые колебания и может быть сформулирован в виде следующего ОДУ.

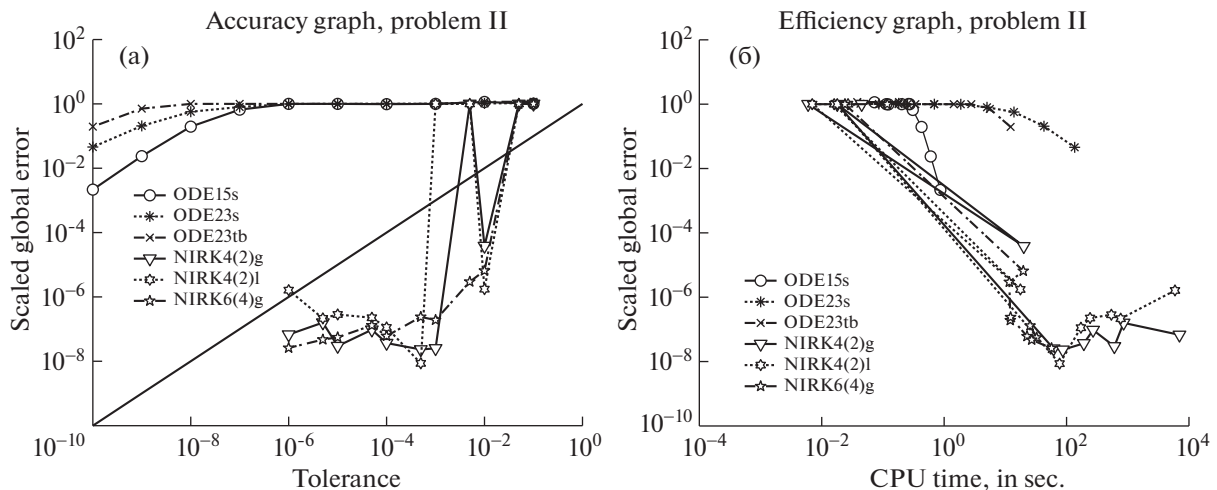
Задача II:

$$x_1'(t) = x_2(t), \quad x_2'(t) = \lambda[(1 - x_1^2(t))x_2(t) - x_1(t)]$$

с начальным значением $x(0) = (2, 0)^T$. Опять же, жесткость задачи II определяется постоянным параметром λ , который выбирается $\lambda = 10^6$ (см. дополнительную информацию в [29, с. 144]). К сожалению, эта тестовая задача не имеет точного решения в замкнутом виде. Поэтому мы используем численное решение, вычисленное с высокой точностью в конечной точке его интервала интегрирования в качестве нашего эталонного решения. Другими словами, итоговая масштабированная глобальная ошибка оценивается для задачи II по формуле

$$|\Delta x(t_{\text{end}})|_{\text{sc}} := \max_{i=1,2,\dots,n} \{|x_{\text{ref}}^i(t_{\text{end}}) - x_{\text{end}}^i| / (1 + |x_{\text{ref}}^i(t_{\text{end}})|)\}, \tag{4.4}$$

где $x_{\text{ref}}(t_{\text{end}})$ обозначает эталонное решение, вычисленное в конечной точке интервала интегрирования, x_{end} означает численное решение, полученное исследуемой парой NIRK (или решателем MATLAB) в той же конечной точке, n – размерность второго тестового примера, а верхний индекс i отмечает i -ю компоненту в соответствующем векторе. Кроме того, чтобы выявить максимально возможные ошибки, мы берем в качестве конечной точки интервала интегрирования импульсную точку генератора Ван-дер-Поля, где совершается максимальная ошибка. Точнее, мы решаем задачу II на интервале $[0, t_6]$, где $t_6 := 1.614286811415814$ и используем численное ре-

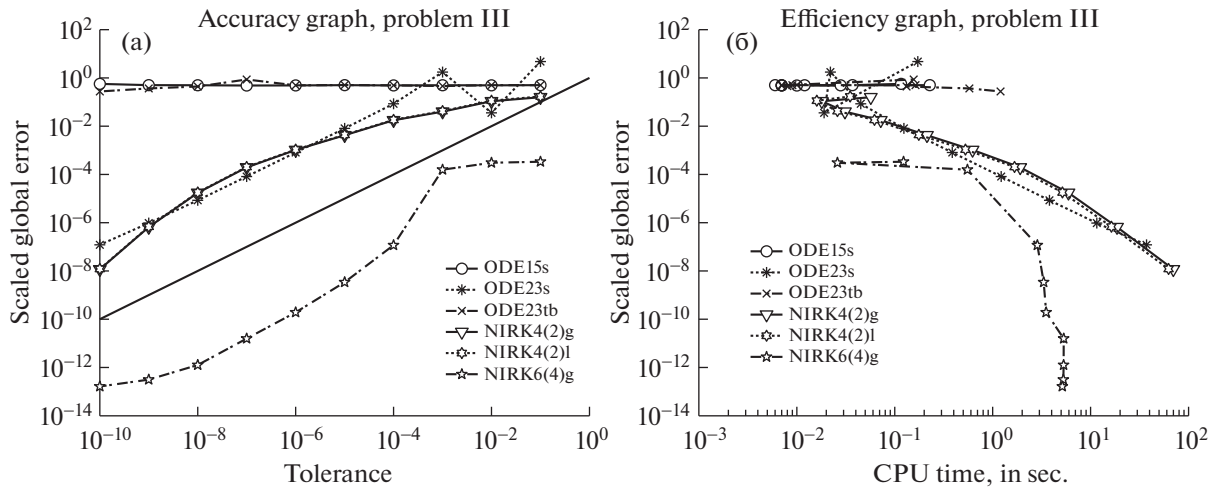


Фиг. 2. Диаграммы точности и эффективности для жестких решателей ОДУ с локальным контролем ошибки из пакета MATLAB и для трех пар NIRK с масштабированным модифицированным локальным и глобальным контролем ошибки, протестированных на задаче II.

шение, найденное с высокой точностью в *t6* и представленное в [79, табл. 8] в качестве нашего эталонного решения. Мы также уменьшаем диапазон допусков для решателей NIRK следующим образом: $\text{Tol} := 10^{-1}, 5 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}, \dots, 10^{-6}$, поскольку реализованный механизм управления глобальной ошибкой может потребовать слишком высокой точности ϵ_l для управления локальной погрешностью, при более высоких требованиях Tol к точности численного интегрирования задачи II. Напомним, что допуск ϵ_l на локальную ошибку устанавливается автоматически в [54, Алгоритм 3.2]. С другой стороны, все решатели MATLAB оперируют с теми же самыми допусками, что и в первом эксперименте. Итоговые погрешности этого теста представлены на фиг. 2.

Из диаграммы точности фиг. 2 следует, что все коды MATLAB не способны вычислить более или менее точные численные решения для осциллятора Ван-дер-Поля. Более того, мы не видим даже реальной сходимости этих кодов до тех пор, пока требования к точности численного решения не становятся очень высокими, т.е. когда $\text{Tol} \leq 10^{-8}$. Кроме того, среди решателей MATLAB для наиболее точного численного решения, полученного решателем ODE15s для $\text{Tol} := 10^{-10}$, соотношение “Истинная ошибка”/“Допуск на ошибку” составляет около 10^7 , что вряд ли приемлемо на практике. Напротив, все коды NIRK с масштабированным модифицированным локальным и глобальным контролем ошибки дают численные решения, соответствующие заданным требованиям к точности вычислений (см. фиг. 2a). Можно также наблюдать некоторую переоценку ошибки в решателях NIRK. Это результат разности на два порядка во встроенных методах NIRK, т.е. глобальная ошибка оценивается для метода, который является на два порядка менее точным, чем метод, используемый для вычисления итогового численного решения. Однако эта переоценка не приводит к существенному снижению эффективности кодов NIRK для задачи II. Последнее следует из диаграммы эффективности на фиг. 2. Теперь мы видим, что лучшим методом интегрирования осциллятора Ван-дер-Поля среди рассмотренных в данной работе является NIRK6(4)g, который обеспечивает требуемую точность численного решения за минимальное процессорное время.

На следующем жестком примере мы покажем, что разработанная стратегия оценки глобальной ошибки дешева, но не асимптотически корректна (см. объяснение в [54]). Это означает, что существуют жесткие ЗК, для которых коды NIRK могут автоматически не обеспечить требуемую точность численного интегрирования. Для этого рассмотрим наш наиболее сложный тестовый пример в [58], который представлен в виде следующего ОДУ.



Фиг. 3. Диаграммы точности и эффективности для жестких решателей ОДУ с локальным контролем ошибки из пакета MATLAB и для трех пар NIRK с масштабированным модифицированным локальным и глобальным контролем ошибки, протестированных на задаче III.

Задача III:

$$\begin{aligned}
 x_1'(t) &= \lambda[x_2^2(t) - x_1(t)] + 2x_1(t)/x_2(t), & x_2'(t) &= x_1(t) - x_2^2(t) + 1, \\
 x_3'(t) &= -50[x_2(t) - 2]x_3(t)
 \end{aligned}$$

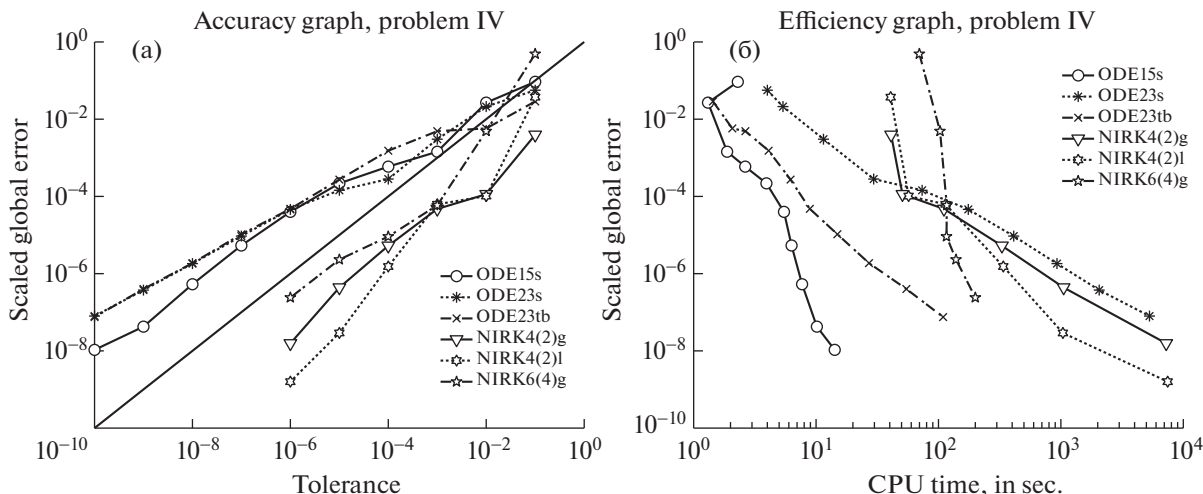
с начальными значениями $x(0) = (1, 1, \exp\{-25\})^T$. Задача III решается на интервале $[0, 2]$. Постоянный параметр λ регулирует жесткость в этом тестовом примере. Мы полагаем $\lambda = 10^6$. В процитированной статье дано точное решение $x_{\text{exact}}(t) := (x_1(t), x_2(t), x_3(t))^T$ для задачи III. Это точное решение не зависит от жесткости, и его компоненты определяются следующим образом:

$$x_1(t) = (t + 1)^2, \quad x_2(t) = t + 1, \quad x_3(t) = \exp\{-25(t - 1)^2\}. \tag{4.5}$$

Формулы (4.5) используются для оценки выходной масштабированной глобальной ошибки по формуле (4.3). Все коды применяются с $\text{To1} := 10^{-1}, 10^{-2}, \dots, 10^{-10}$ в третьем численном эксперименте и на фиг. 3 показана допущенная глобальная погрешность в зависимости от используемых допусков на ошибку и времени численного интегрирования.

График точности фиг. 3 показывает, что все рассматриваемые коды не в состоянии обеспечить требуемую точность численного интегрирования задачи III, за исключением NIRK6(4)g. Кроме того, решатели MATLAB ODE15s и ODE23tb практически бесполезны для этого тестового примера, поскольку их точность не зависит от уровня допуска на локальную ошибку, установленного в третьем вычислительном эксперименте, т.е. они находят численные решения с аналогичной ошибкой для всех значений порога ошибки To1, примененных здесь. Остальные три кода ODE23s, NIRK4(2)g и NIRK4(2)l показывают более или менее одинаковые и лучшие результаты. Таким образом, диаграммы точности и эффективности на фиг. 3 показывают, что наиболее точным и эффективным кодом для решения задачи III снова является NIRK6(4)g.

Наконец, мы протестируем наши адаптивные коды NIRK4(2)g, NIRK4(2)l и NIRK6(4)g в экстремальных условиях интегрирования крупномасштабных задач, возникающих при полудискретизации уравнений в частных производных (УрЧП). Наглядным примером для этого является известный двумерный брюсселатор с диффузией и периодическими граничными условиями [29, с. 151, 152]. Следуя цитированной книге, мы приходим к следующей задаче.



Фиг. 4. Диаграммы точности и эффективности для жестких решателей ОДУ с локальным контролем ошибки из пакета MATLAB и для трех пар NIRK с масштабированным модифицированным локальным и глобальным контролем ошибки, протестированных на задаче IV.

Задача IV:

$$\begin{aligned} \partial_t u(t, x, y) &= 1 + u^2(t, x, y)v(t, x, y) - 4.4u(t, x, y) + \alpha[\partial_{xx}^2 u(t, x, y) + \partial_{yy}^2 u(t, x, y)] + f(t, x, y), \\ \partial_t v(t, x, y) &= 3.4u(t, x, y) - u^2(t, x, y)v(t, x, y) + \alpha[\partial_{xx}^2 v(t, x, y) + \partial_{yy}^2 v(t, x, y)], \end{aligned}$$

где $x \in [0, 1]$, $y \in [0, 1]$, $t \in [0, 6]$. Периодические граничные условия подразумевают

$$u(t, x, y) = u(t, x + 1, y) = u(t, x, y + 1), \quad v(t, x, y) = v(t, x + 1, y)v(t, x, y + 1).$$

Начальные значения для этого УрЧП принимаются равными

$$u(0, x, y) = 22y(1 - y)^{3/2}, \quad v(0, x, y) = 27x(1 - x)^{3/2}.$$

Неоднородный член $f(t, x, y)$ в правой части задачи IV полагаем в виде

$$f(t, x, y) = \begin{cases} 5, & \text{если } (x - 0.3)^2 + (y - 0.6)^2 \leq 0.1^2 \text{ и } t \geq 1.1, \\ 0 & \text{иначе.} \end{cases}$$

В цитируемой книге также говорится, что представленный тестовый пример будет жестким, если постоянный параметр $\alpha \geq 0.1$. Вот почему мы принимаем $\alpha = 0.1$ в этом эксперименте.

Полудискретизация задачи IV означает, что мы берем сетку из 50 точек по каждой переменной. Пространственные производные второго порядка в правой части этого УрЧП заменяются центральными разделенными разностями. Это приводит к ОДУ размера 5000. В конечном итоге мы получаем достаточно большую тестовую задачу, удобную для исследования наших решателей NIRK в условиях, приближенных к решению реальных практических задач. Опять же, задача IV не имеет точного решения в замкнутом виде. Таким образом, мы используем эталонное решение, вычисленное с высокой точностью в конечной точке интервала интегрирования $[0, 6]$ для оценки полученной масштабированной глобальной ошибки по формуле (4.4) с $n = 5000$. Требования к точности всех кодов NIRK определяются параметром $Tol := 10^{-1}, 10^{-2}, \dots, 10^{-6}$, тогда как решатели MATLAB тестируются с теми же самыми допусками, что и в приведенных выше тестовых примерах. Глобальные ошибки всех решателей ОДУ, наблюдаемые в последнем эксперименте, показаны на фиг. 4.

Прежде всего отметим, что задача IV достаточно проста и не содержит сильно нелинейных членов. Поэтому даже жесткие решатели MATLAB с только локальным контролем ошибок способны обеспечить приемлемые численные решения. Тем не менее все эти коды не в состоянии обеспечить требуемую точность численного интегрирования и занижают истинную погрешность примерно на два порядка при высоких требованиях к точности численного интегрирования. Напротив, NIRK4 (2) g, NIRK4 (2) l и NIRK6 (4) g работают точно и возвращают числен-

ные решения, удовлетворяющие наперед заданным условиям точности, в автоматическом режиме (см. левый график фиг. 4). Однако такие достаточно малые масштабированные глобальные ошибки достигаются ценой повторных численных решений задачи IV, начиная с начальной точки интервала интегрирования. Эти дополнительные численные интегрирования требуют увеличения времени счета и, следовательно, снижают их эффективность, представленную на правой диаграмме фиг. 4. Последняя фигура также показывает, что наш код NIRK6 (4) σ превосходит оставшиеся два NIRK4 (2) σ и NIRK4 (2) 1 по эффективности численного интегрирования задачи IV. Во-первых, его механизм оценки глобальной погрешности лучше оценивает истинную ошибку и приводит к ее меньшей переоценке, наблюдаемой по сравнению с другими решателями NIRK на диаграмме точности фиг. 4. Во-вторых, эта переоценка ошибки кодов NIRK4 (2) σ и NIRK4 (2) 1 является причиной их меньшей эффективности, которая отмечена на фиг. 4б. Производительность методов NIRK четвертого порядка почти сравнима с производительностью наименее эффективного кода MATLAB ODE23s.

В конечном счете представленные численные тесты подтверждают успешность обновленных пар NIRK с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок для численного интегрирования сложных жестких и крупномасштабных ЗК в практических расчетах, а также их превосходство над жесткими встроенными интеграторами MATLAB при решении ОДУ с наперед заданной точностью в автоматическом режиме.

В итоге адаптивный решатель NIRK6 (4) σ , по-видимому, является наиболее предпочтительным методом для решения жестких моделей ОДУ, возникающих часто в прикладной науке и технике. Первое успешное применение рассмотренных методов NIRK для обработки детерминистических и стохастических математических моделей в гидромеханике и нелинейной фильтрации Калмана описано в работах [1]–[15].

5. ЗАКЛЮЧЕНИЕ

В данной работе обсуждаются вычислительные методы для решения задач Коши вида (1.1), которые являются традиционным инструментом для описания математических динамических моделей с непрерывным временем в различных исследованиях. Точность их численного решения может иметь решающее значение для адекватности и согласованности используемого метода математического моделирования. В центре нашего исследования находятся адаптивные методы NIRK с автоматическим контролем глобальной ошибки, разработанные в [54]. К сожалению, приведенная схема оценки глобальной ошибки обнаруживает серьезную переоценку ошибки для жестких примеров, рассмотренных в [54], что снижает ее практическую эффективность. Именно поэтому в данной работе был исследован источник этой переоценки ошибки и найден способ решения указанной проблемы. Это привело к парам NIRK типа Гаусса и Лобатто с автоматическим управлением масштабированными модифицированными оценками локальной и глобальной ошибок, которые гораздо более эффективны для решения жестких ЗК, чем адаптивные решатели ОДУ, представленные в [54]. Кроме того, сложные тесты, выполненные в разд. 4, не только подтвердили высокую практическую эффективность наших вычислительных методов, но и выявили их превосходство над жесткими встроенными кодами ODE15s, ODE23s и ODE23tb пакета MATLAB для большинства используемых тестовых примеров. Все это создает твердую основу для дальнейшего изучения новых адаптивных методов NIRK в практических вычислениях. Например, решатели NIRK6 (4) σ , NIRK4 (2) σ и NIRK4 (2) 1, разработанные в этой статье, использованы для повышения эффективности непрерывно-дискретных обобщенного, кубатурного и сигма-точечного методов фильтрации Калмана, применяемых для оценки состояния стохастических систем с непрерывным временем в моделях радиолокационного слежения как с гауссовским, так и с негауссовским шумом [3], [6], [7], [11], [14], [15], а также в электротехнических и химических инженерных исследованиях [2], [4], [8]–[10], [12], [13].

СПИСОК ЛИТЕРАТУРЫ

1. Kulikov G. Yu., Kulikova M. V. Accurate numerical implementation of the continuous-discrete extended Kalman filter // IEEE Trans. Automat. Contr. 2014. V. 59 (1). P. 273–279.
2. Kulikov G. Yu., Kulikova M. V. High-order accurate continuous-discrete extended Kalman filter for chemical engineering // Eur. J. Contr. 2015. V. 21. P. 14–26.
3. Kulikov G. Yu., Kulikova M. V. The accurate continuous-discrete extended Kalman filter for radar tracking // IEEE Trans. Signal Process. 2016. V. 64 (4). P. 948–958.

4. Kulikov G.Yu., Kulikova M.V. Estimating the state in stiff continuous-time stochastic systems within extended Kalman filtering // *SIAM J. Sci. Comput.* 2016. V. 38 (6). P. A3565–A3588.
5. Kulikov G.Yu., Lima P.M., Morgado M.L. Analysis and numerical approximation of singular boundary value problems with the p -Laplacian in fluid mechanics // *J. Comput. Appl. Math.* 2014. V. 262. P. 87–104.
6. Kulikov G.Yu., Kulikova M.V. Accurate cubature and extended Kalman filtering methods for estimating continuous-time nonlinear stochastic systems with discrete measurements // *Appl. Numer. Math.* 2017. V. 111. P. 260–275.
7. Kulikov G.Yu., Kulikova M.V. Accurate continuous-discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking // *Signal Process.* 2017. V. 139. P. 25–35.
8. Kulikov G.Yu., Kulikova M.V. Accurate state estimation of stiff continuous-time stochastic models in chemical and other engineering // *Math. Comput. Simulation.* 2017. V. 142. P. 62–81.
9. Kulikov G.Yu., Kulikova M.V. Square-root Kalman-like filters for estimation of stiff continuous-time stochastic systems with ill-conditioned measurements // *IET Control Theory Appl.* 2017. V. 11 (9). P. 1420–1425.
10. Kulikov G.Yu., Kulikova M.V. Moore-Penrose-pseudo-inverse-based Kalman-like filtering methods for estimation of stiff continuous-discrete stochastic systems with ill-conditioned measurements // *IET Control Theory Appl.* 2018. V. 12 (16). P. 2205–2212.
11. Kulikov G.Yu., Kulikova M.V. Estimation of maneuvering target in the presence of non-Gaussian noise: A coordinated turn case study // *Signal Process.* 2018. V. 145. P. 241–257.
12. Kulikov G.Yu., Kulikova M.V. Practical implementation of extended Kalman filtering in chemical systems with sparse measurements // *Russian J. Numer. Anal. Math. Modelling.* 2018. V. 33 (1). P. 41–53.
13. Kulikov G.Yu., Kulikova M.V. Numerical robustness of extended Kalman filtering based state estimation in ill-conditioned continuous-discrete nonlinear stochastic chemical systems // *Int. J. Robust Nonlinear Control.* 2019. V. 29 (5). P. 1377–1395.
14. Kulikov G.Yu., Kulikova M.V. Square-root accurate continuous-discrete extended-unscented Kalman filtering methods with embedded orthogonal and J -orthogonal QR decompositions for estimation of nonlinear continuous-time stochastic models in radar tracking // *Signal Process.* 2020. V. 166. P. 107253.
15. Kulikov G.Yu., Kulikova M.V. NIRK-based Cholesky-factorized square-root accurate continuous-discrete unscented Kalman filters for state estimation in nonlinear continuous-time stochastic models with discrete measurements // *Appl. Numer. Math.* 2020. V. 147. P. 196–221.
16. Aggoun L., Elliot R.J. *Measure Theory and Filtering: Introduction and Applications.* Cambridge University Press, Cambridge, U.K., 2005.
17. Åström K.J. *Introduction to Stochastic Control Theory.* Academic Press, New York, 1970.
18. Bar-Shalom Y., Li X.-R., Kirubarajan T. *Estimation with Applications to Tracking and Navigation.* Wiley, New York, 2001.
19. Crassidis J.L., Junkins J.L. *Optimal Estimation of Dynamic Systems,* CRC Press LLC, New York, 2004.
20. Grewal M.S., Andrews A.P. *Kalman Filtering: Theory and Practice.* Prentice Hall, New Jersey, 2001.
21. Grewal M.S., Weill L.R., Andrews A.P. *Global Positioning Systems. Inertial Navigation and Integration,* Wiley, New York, 2001.
22. Jazwinski A.H. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.
23. Øksendal B. *Stochastic Differential Equations: An Introduction with Applications.* Springer, New York, 2003.
24. Rawlings J.B., Mayne D.Q. *Model Predictive Control: Theory and Design.* Bob Hill Publishing, LLC, Madison, Wisconsin, 2013.
25. Butcher J.C. *Numerical Methods for Ordinary Differential Equations.* John Wiley and Sons, Chichester, 2008.
26. Dekker K., Verwer M.P. *Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations.* North-Holland, Amsterdam, 1984.
27. Hairer E., Lubich C., Wanner G. *Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations.* Springer-Verlag, Berlin, 2002.
28. Hairer E., Nørsett S.P., Wanner G. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer-Verlag, Berlin, 1993.
29. Hairer E., Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, 1996.
30. Jackiewicz Z. *General Linear Methods for Ordinary Differential Equations.* John Wiley and Sons, Hoboken, 2009.
31. Sanz-Serna J.M., Calvo M.P. *Numerical Hamilton Problems.* Chapman and Hall, London, 1994.
32. Shampine L.F. *Numerical Solution of Ordinary Differential Equations.* Chapman and Hall, New York, 1994.
33. Kulikov G.Yu., Shindin S.K. On a family of cheap symmetric one-step methods of order four, in: V.N. Alexandrov, et al. (Eds.), *Computational Science – ICCS 2006, Proceedings, Part I,* 6th International Conference, Reading, UK, May 28–31, 2006, *Lecture Notes in Computer Science,* vol. 3991, Springer-Verlag, Berlin, 2006. P. 781–785.

34. *Kulikov G.Yu., Shindin S.K.* Numerical tests with Gauss-type nested implicit Runge–Kutta formulas, in: Y. Shi, et al. (Eds.), Computational Science – ICCS 2007, Proceedings, Part I, 7th International Conference, Beijing, China, May 27–30, 2007, Lecture Notes in Computer Science, vol. 4487, Springer-Verlag, Berlin, 2007. P. 136–143.
35. *Kulikov G.Yu., Shindin S.K.* Adaptive nested implicit Runge–Kutta formulas of Gauss type // *Appl. Numer. Math.* 2009. V. 59 (3–4). P. 707–722.
36. *Kulikov G.Yu.* Automatic error control in the Gauss-type nested implicit Runge–Kutta formula of order 6 // *Russian J. Numer. Anal. Math. Modelling.* 2009. V. 24 (2). P. 123–144.
37. *Kulikov G.Yu.* Embedded symmetric nested implicit Runge–Kutta methods of Gauss and Lobatto types for solving stiff ordinary differential equations and Hamiltonian systems // *Comput. Math. Math. Phys.* 2015. V. 55 (6). P. 983–1003.
38. *Cash J.R.* A class of implicit Runge–Kutta methods for the numerical solution of stiff ordinary differential equations // *J. ACM.* 1975. V. 22 (4). P. 504–511.
39. *Cash J.R.* On a class of implicit Runge–Kutta procedures // *IMA J. Numer. Anal.* 1977. V. 19 (4). P. 455–470.
40. *Cash J.R.* On a note of the computational aspects of a class of implicit Runge–Kutta procedures // *IMA J. Numer. Anal.* 1977. V. 20 (4). P. 425–441.
41. *Cash J.R., Singhal A.* Mono-implicit Runge–Kutta formulae for the numerical integration of stiff differential systems // *IMA J. Numer. Anal.* 1982. V. 2. P. 211–227.
42. *Skvortsov L.M.* How to avoid accuracy and order reduction in Runge–Kutta methods as applied to stiff problems // *Comput. Math. Math. Phys.* 2017. V. 57 (7). P. 1124–1139.
43. *Skvortsov L.M.* Implicit Runge–Kutta methods with explicit internal stages // *Comput. Math. Math. Phys.* 2018. V. 58 (3). P. 307–321.
44. *Shampine L.F., Reichelt M.W.* The MATLAB ODE suite // *SIAM J. Sci. Comput.* 1997. V. 18. P. 1–22.
45. *Higham D., Higham N.* MATLAB Guide. SIAM, Philadelphia, 2005.
46. *Kulikov G.Yu., Weiner R.* Doubly quasi-consistent parallel explicit peer methods with built-in global error estimation // *J. Comput. Appl. Math.* 2010. V. 233 (9). P. 2351–2364.
47. *Baker T.S., Dormand J.R., Gilmore J.P., Prince P.J.* Continuous approximation with embedded Runge–Kutta methods // *Appl. Numer. Math.* 1996. V. 22. P. 51–62.
48. *Dormand J.R., Prince P.J.* Runge–Kutta triples // *Comput. Math. Appl.* 1986. V. 12A. P. 1007–1017.
49. *Dormand J.R., Prince P.J.* Practical Runge–Kutta processes // *SIAM J. Sci. Statist. Comput.* 1989. V. 10. P. 977–989.
50. *Enright W.H.* Analysis of error control strategies for continuous Runge–Kutta methods // *SIAM J. Numer. Anal.* 1989. V. 26. P. 588–599.
51. *Enright W.H., Jackson K.R., Nørsett S.P., Thomsen P.G.* Interpolants for Runge–Kutta formulas // *ACM Trans. Math. Software.* 1986. V. 12. P. 193–218.
52. *Higham D.J.* Highly continuous Runge–Kutta interpolants // *ACM Trans. Math. Software.* 1991. V. 17. P. 368–386.
53. *Sharp P.W., Verner J.H.* Generation of high-order interpolants for explicit Runge–Kutta pairs // *ACM Trans. Math. Software.* 1998. V. 24. P. 13–29.
54. *Kulikov G.Yu.* Cheap global error estimation in some Runge–Kutta pairs // *IMA J. Numer. Anal.* 2013. V. 33 (1). P. 136–163.
55. *Weiner R., Kulikov G.Yu.* Local and global error estimation and control within explicit two-step peer triples // *J. Comput. Appl. Math.* 2014. V. 262. P. 261–270.
56. *Kulikov G.Yu., Weiner R.* A singly diagonally implicit two-step peer triple with global error control for stiff ordinary differential equations // *SIAM J. Sci. Comput.* 2015. V. 37 (3). P. A1593–A1613.
57. *Weiner R., Kulikov G.Yu., Beck S., Bruder J.* New third- and fourth-order singly diagonally implicit two-step peer triples with local and global error controls for solving stiff ordinary differential equations // *J. Comput. Appl. Math.* 2017. V. 316. P. 380–391.
58. *Kulikov G.Yu., Weiner R.* Global error estimation and control in linearly-implicit parallel two-step peer-methods // *J. Comput. Appl. Math.* 2011. V. 236 (6). P. 1226–1239.
59. *Aïd R., Levacher L.* Numerical investigations on global error estimation for ordinary differential equations // *J. Comput. Appl. Math.* 1997. V. 82. P. 21–39.
60. *Shampine L.F.* Global error estimation for stiff ODEs // *Lecture Notes in Mathematics.* 1984. V. 1066. P. 159–168.
61. *Shampine L.F.* Error estimation and control for ODEs // *J. Sci. Comput.* 2005. V. 25. P. 3–16.
62. *Shampine L.F., Watts H.A.* Global error estimation for ordinary differential equations // *ACM Trans. Math. Software.* 1976. V. 2. P. 172–186.
63. *Skeel R.D.* Thirteen ways to estimate global error // *Numer. Math.* 1986. V. 48. P. 1–20.

64. *Skeel R.D.* Global error estimation and the backward differentiation formulas // *Appl. Math. Comput.* 1989. V. 31. P. 197–208.
65. *Stetter H.J.* Global error estimation in ODE-solvers, in: J. Hinze (Ed.), *Numerical integration of differential equations and large linear systems*, Proceedings, Bielefeld 1980. *Lecture Notes in Mathematics*, vol. 968, Springer-Verlag, Berlin, 1982. P. 269–279.
66. *Dormand J.R., Duckers R.R., Prince P.J.* Global error estimation with Runge–Kutta methods // *IMA J. Numer. Anal.* 1984. V. 4. P. 169–184.
67. *Dormand J.R., Gilmore J.P., Prince P.J.* Globally embedded Runge–Kutta schemes // *Ann. Numer. Math.* 1994. V. 1. P. 97–106.
68. *Dormand J.R., Lockyer M.A., McGorrigan N.E., Prince P.J.* Global error estimation with Runge–Kutta triples // *Comput. Math. Appl.* 1989. V. 18. P. 835–846.
69. *Lang J., Verwer J.G.* On global error estimation and control for initial value problems // *SIAM J. Sci. Comput.* 2007. V. 29. P. 1460–1475.
70. *Macdougall T., Verner J.H.* Global error estimators for order 7, 8 Runge–Kutta pairs // *Numer. Algorithms.* 2002. V. 31. P. 215–231.
71. *Makazaga J., Murua A.* New Runge–Kutta based schemes for ODEs with cheap global error estimation // *BIT.* 2003. V. 43. P. 595–610.
72. *Shampine L.F., Baca L.S.* Global error estimates for ODEs based on extrapolation methods // *SIAM J. Sci. Stat. Comput.* 1985. V. 6. P. 1–14.
73. *Tirani R.* A parallel algorithm for the estimation of the global error in Runge–Kutta methods // *Numer. Algorithms.* 2002. V. 31. P. 311–318.
74. *Calvo M., Higham D.J., Montijano J.I., Rández L.* Step size selection for tolerance proportionality in explicit Runge–Kutta codes // *Adv. Comput. Math.* 1997. V. 7. P. 361–382.
75. *Calvo M., González-Pinto S., Montijano J.I.* Global error estimation based on the tolerance proportionality for some adaptive Runge–Kutta codes // *J. Comput. Appl. Math.* 2008. V. 218. P. 329–341.
76. *Higham D.J.* Global error versus tolerance for explicit Runge–Kutta methods // *IMA J. Numer. Anal.* 1991. V. 11. P. 457–480.
77. *Higham D.J.* The tolerance proportionality of adaptive ODE solvers // *J. Comput. Appl. Math.* 1993. V. 45. P. 227–236.
78. *Kulikov G.Yu., Merkulov A.I., Shindin S.K.* Asymptotic error estimate for general Newton-type methods and its application to differential equations // *Russian J. Numer. Anal. Math. Modelling.* 2007. V. 22 (6). P. 567–590.
79. *Kulikov G.Yu., Shindin S.K.* One-leg variable-coefficient formulas for ordinary differential equations and local-global step size control // *Numer. Algorithms.* 2006. V. 43 (1). P. 99–121.