



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 51 за 2018 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Галанин М.П., Конев С.А.

Разработка и применение
экспоненциального метода
интегрирования жёстких
систем на основе
классического метода Рунге
—Кутты

Рекомендуемая форма библиографической ссылки: Галанин М.П., Конев С.А. Разработка и применение экспоненциального метода интегрирования жёстких систем на основе классического метода Рунге—Кутты // Препринты ИПМ им. М.В.Келдыша. 2018. № 51. 24 с. doi:[10.20948/prepr-2018-51](https://doi.org/10.20948/prepr-2018-51)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-51>

Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
им. М.В. Келдыша
Российской академии наук

М.П. Галанин, С.А. Конев

Разработка и применение
экспоненциального метода интегрирования
жёстких систем
на основе классического метода Рунге—Кутты

Москва — 2018

Галанин М.П., Конев С.А.

Разработка и применение экспоненциального метода интегрирования жёстких систем на основе классического метода Рунге—Кутты

Работа посвящена разработке численного метода решения жёстких систем обыкновенных дифференциальных уравнений. На основе вычислительных формул классического четырёхстадийного метода Рунге—Кутты получен новый экспоненциальный метод RK4exp. Для метода RK4exp доказана сходимость и вычислена функция устойчивости. Теоретические выводы подтверждены результатами вычислительных экспериментов.

Ключевые слова: обыкновенные дифференциальные уравнения; численные методы; экспоненциальные методы; методы Рунге—Кутты.

Mikhail Pavlovich Galanin, Stanislav Andreevich Konev

Development of exponential integrator based on classical Runge—Kutta method and it's application for solving stiff systems

The paper deals with numerical method for solving stiff systems of ordinary differential equations. The new method RK4exp is derived on the base of classical four-stage Runge—Kutta method. The convergence of RK4exp is proved and it's stability function is calculated. Theoretical results are verified with numerical experiments.

Key words: ordinary differential equations; numerical methods; exponential integrators; Runge—Kutta methods.

Работа выполнена при финансовой поддержке РФФИ (проект № 18-01-00252).

Введение

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений (ОДУ) вида:

$$\dot{u} = f(u), \quad 0 < t < T; \quad (0.1a)$$

$$u(0) = u_0; \quad (0.1b)$$

где $u = u(t) = (u_1(t), \dots, u_d(t))^T$ — точное решение задачи, $u_0 = (u_0^1, \dots, u_0^d)^T$ — вектор начальных условий, d — размерность задачи.

Будем считать, что задача (0.1) является жёсткой [1, с. 181], т.е. величина S , определяемая как

$$S = \sup_{t \in (0, T]} \left(\max_{1 \leq i \leq d} |\operatorname{Re} \lambda_i(t)| / \min_{1 \leq i \leq d} |\operatorname{Re} \lambda_i(t)| \right), \quad (0.2)$$

велика. Здесь $\{\lambda_i(t)\}_{i=1}^d$ — собственные значения матрицы Якоби $J(t)$ правой части $f(u)$ на точном решении $u(t)$ задачи (0.1): $J(t) = f_u(u(t))$.

Целью работы является конструирование, доказательство сходимости и экспериментальное исследование численного метода решения жёстких задач Коши для систем ОДУ, основанного на классической четырёхстадийной явной схеме Рунге—Кутты [2, с. 15].

В разделе 1 представлена постановка задачи и кратко описаны возможные пути её решения. В разделе 2 описан алгоритм численного метода, отмечены основные тонкости его реализации. В разделе 3 доказана сходимость численного метода, в разделе 4 построена функция устойчивости. В разделе 5 представлены результаты вычислительных экспериментов.

1. Постановка задачи

Выделим в правой части (0.1) некоторую линейную часть, после чего задача перейдёт в следующую:

$$\dot{u} = Au + g(u), \quad 0 < t < T; \quad (1.1a)$$

$$u(0) = u_0; \quad (1.1b)$$

где $g(u) = f(u) - Au$, A — произвольная матрица: она может быть зафиксирована или вычисляться по определённому правилу во время расчёта; в зависимости от способа выделения линейной части получаются различные алгоритмы [3].

Введём на отрезке $[0, T]$ сетку $\{t_k\}_{k=0}^N$ и проинтегрируем поставленную задачу на участке $[t_n, t_{n+1}]$ — это приводит к серии интегральных уравнений вида:

$$u(t_{n+1}) = e^{\tau_n A} u(t_n) + e^{\tau_n A} \int_0^{\tau_n} e^{-\sigma A} g(u(t_n + \sigma)) d\sigma, \quad n \geq 0, \quad (1.2)$$

здесь $\tau_n = t_{n+1} - t_n$, а $\exp(\tau_n A)$ — матричная экспонента [4, с. 137]. В дальнейшем для краткости нижние индексы у шага τ_n будем опускать. Конечный вид метода определяется правилом вычисления матрицы A , способами аппроксимации интеграла (1.2) и вычисления матричных экспонент, а также связанных с ними функций. Рассмотрим несколько вариантов подобной аппроксимации и обсудим, к каким методам они приводят.

Вариант 1. Пусть A — это матрица Якоби, которая вычисляется в точке $u_n = u(t_n)$: $A = f_u(u_n)$, функция g в интеграле (1.2) (обозначим его через I) аппроксимируется с помощью значений в узле $\sigma = 0$, а интеграл от матричной экспоненты вычисляется точно:

$$I \approx \int_0^\tau e^{-\sigma A} g(u(t_n)) d\sigma = A^{-1} (I - e^{-\tau A}) g(u(t_n)),$$

что задаёт явный экспоненциальный метод Эйлера—Розенброка [3]. Фактически данный метод был реализован в работе [5]: матричная экспонента вычислялась с помощью классического четырёхстадийного метода Рунге—Кутты, а линеаризация правой части в задаче (1.1) осуществлялась лишь в некоторых точках согласно определённому критерию. Как следует из результатов работы [5], метод достаточно легко программируется и даёт удовлетворительные результаты в случае задач умеренной жёсткости. Для задач с большим числом жёсткости S он может вести себя неустойчиво и существенно проигрывает (4,2)-методу [6].

Вариант 2. Матрица A — та же, функция g в интеграле I аппроксимируется с помощью значений в узле $\sigma = \tau$, интеграл от матричной экспоненты вычисляется точно:

$$I \approx \int_0^\tau e^{-\sigma A} g(u(t_n + \tau)) d\sigma = A^{-1} (I - e^{-\tau A}) g(u(t_{n+1})),$$

что задаёт неявный экспоненциальный метод Эйлера—Розенброка [3]. Данный метод реализован в работе [7], матричная экспонента приближалась с помощью (0, 1)-аппроксимации Паде [8]. Метод показал своё устойчивое поведение даже в случае задач с большой жёсткостью, но при этом отмечалось отставание профиля численного решения от точного.

Вариант 3. Рассмотренные выше варианты аппроксимации представляют собой простейшие экспоненциальные методы [3, 9]. Основная идея, заложенная в них, — заменить функцию g в интеграле I на полином и полученное выражение вида $P(\theta) \exp(\theta A)$, где P — полином, проинтегрировать точно.

Данный подход приводит к необходимости вычислять на каждом шаге несколько значений функций от матриц вида:

$$\varphi_k(A) = \int_0^1 e^{\tau(1-\theta)A} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1,$$

что в случае матриц A больших размерностей сопряжено с решением дополнительных сложных задач [10].

Поэтому далее предложен иной метод, который совмещает в себе выделение линейной части в виде матрицы A , но при этом на каждом шаге требует вычисления только одной матричной функции — функции $\exp(\tau A/2)$.

2. Формулировка метода

Зафиксируем на отрезке $[0, T]$ сетку $\{t_k\}_{k=0}^N$. Очевидно [11, с. 205], что от решения задачи (1.1) можно перейти к решению серии задач на каждом из отрезков $[t_n, t_{n+1}]$ вида:

$$\dot{v}_n = Av_n + F_n(v_n), \quad 0 \leq t' \leq \tau_n; \quad (2.1a)$$

$$v_n(0) = 0; \quad (2.1b)$$

где $\tau_n = t_{n+1} - t_n$, функция $F_n(z) = f(u_n + z) - Az$, $u_n = u(t_n)$ — решение в узле t_n , а решение $v_n(t')$ каждой из задач связано с решением задачи (1.1) следующим образом: $u(t_{n+1}) = u(t_n) + v_n(\tau)$. Далее индекс n будет опущен. Ради общности изложения способ задания матрицы A пока не конкретизируется.

Для построения численного метода сделаем замену вида

$$w(t') = e^{-At'} v(t'), \quad (2.2)$$

после чего задача для $v(t')$ перейдёт в задачу для функции $w(t')$ (штрих у аргумента опущен):

$$\dot{w} = e^{-At} F(e^{At} w) = G(t, w), \quad (2.3a)$$

$$w(0) = 0. \quad (2.3b)$$

Основная идея данной работы — численное интегрирование задачи (2.3) с помощью классических методов Рунге—Кутты. Напомним, как выглядят расчётные формулы явного m -стадийного метода Рунге—Кутты, применённого к решению неавтономной задачи (2.3) ([1, с. 170], [2, с. 15], [3], [12, с. 139]):

$$\omega_{k+1} = \omega_k + h \sum_{i=1}^m c_i G_{k,i}, \quad (2.4a)$$

$$G_{k,i} = G(x_k + a_i h, W_{k,i}), \quad 1 \leq i \leq m; \quad (2.4b)$$

$$W_{k,i} = \omega_k + h \sum_{j=1}^{i-1} b_{ij} G_{k,j}, \quad 1 \leq i \leq m. \quad (2.4c)$$

Здесь ω_k — приближение к решению $w(x_k)$, h — шаг интегрирования, а $\{x_k\}_{k=0}^K$ — сетка узлов на отрезке $[0, \tau]$; подобные обозначения используются

с целью различения данного метода и методов, получаемых из аппроксимации (1.2). В общем случае допустимо построение, при котором каждый раз на отрезке $[0, \tau]$ строится достаточно мелкая сетка и на ней происходит интегрирование поставленной задачи методом (2.4).

Мы будем интегрировать задачу (2.3), проводя всего один шаг интегрирования по методу Рунге—Кутты, т.е. в формулах (2.4) полагаем $h = \tau$, $x_0 = 0$, $x_1 = \tau$. Воспользуемся методом RK4 ($m = 4$) со следующими коэффициентами [2, с. 27]:

$$a_2 = a_3 = b_{21} = b_{32} = 1/2, \quad a_4 = b_{43} = 1, \quad c_1 = c_4 = 1/6, \quad c_2 = c_3 = 1/3,$$

остальные равны нулю. Проводя серию элементарных арифметических преобразований, получаем, что формулы (2.4) в применении к задаче (2.3) принимают вид:

$$\omega_1 = \frac{\tau}{6} \left(F_1 + 2e^{-A\tau/2}F_2 + 2e^{-A\tau/2}F_3 + e^{-A\tau}F_4 \right), \quad (2.5)$$

где векторы F_i определяются следующим образом:

$$F_1 = F(0) = f(u_n), \quad (2.6a)$$

$$F_2 = F \left((\tau/2)e^{A\tau/2}F_1 \right), \quad (2.6b)$$

$$F_3 = F \left((\tau/2)F_2 \right), \quad (2.6c)$$

$$F_4 = F \left(\tau e^{A\tau/2}F_3 \right). \quad (2.6d)$$

Для построения окончательных расчётных формул необходимо вспомнить вид замены (2.2), откуда следует точная связь $v(\tau) = \exp(A\tau)w(\tau)$. Следовательно, домножая (2.5) на $\exp(A\tau)$, получим

$$\nu_1 = \frac{\tau}{6} \left(e^{A\tau}F_1 + 2e^{A\tau/2}F_2 + 2e^{A\tau/2}F_3 + F_4 \right), \quad (2.7)$$

где ν_1 — приближение для $v(\tau)$. Полученные формулы (2.6), (2.7) задают численный метод решения задачи (2.1). Назовём этот метод RK4exp — четырёхстадийный метод Рунге—Кутты с экспоненциальной заменой.

Особенности метода. По сравнению с общим экспоненциальным методом данный метод не требует вычисления набора функций φ_k — требуется вычислить всего одну матричную экспоненту $E = \exp(A\tau/2)$, а для вычисления выражения вида $\exp(A\tau)b$ — дважды умножить вектор b на матрицу E слева.

Именно одношаговое интегрирование задачи (2.3) с шагом $h = \tau$ позволяет искать единственную матричную экспоненту $\exp(A\tau/2)$. По сути, данный подход эквивалентен аппроксимации интеграла (аналогичного (1.2)) с помощью квадратуры

$$\int_0^{\tau_n} e^{-\sigma A_n} F(v(\sigma)) d\sigma \approx \sum_{i=1}^m c_i e^{-a_i \tau_n A_n} F(v(a_i \tau_n)).$$

Заметим, что RK4exp не даёт точного решения в случае линейных задач: в этом можно убедиться, непосредственно выписав формулы (2.6), (2.7) для линейной задачи. Выписываем функцию правой части: $f(u) = Au$, следовательно $F(v) = f(y_n + v) - Av = Ay_n \equiv f_n$ — константа на отрезке $[0, \tau]$ в задаче (2.1). Поэтому (2.7) принимает следующий вид:

$$\nu_1 = \frac{\tau}{6} \left(e^{A\tau} + 4e^{A\tau/2} + 1 \right) f_n,$$

при этом точное решение задачи (2.1) при $F_n(v) \equiv f_n$ имеет вид:

$$v(\tau) = A^{-1}(e^{\tau A} - I)f_n$$

— отсюда и возникает несоответствие $v(\tau) \neq \nu_1$.

3. Доказательство сходимости

Известно, что метод RK4 имеет порядок аппроксимации $p = 4$. Желательно, чтобы численное решение задачи (2.1) имело тот же порядок аппроксимации. Следуя [12, с. 139], считаем, что метод Рунге—Кутты (2.4) имеет порядок p , если для достаточно гладких задач (2.3) выполнено:

$$\|w_1 - \omega_1\| \leq Kh^{p+1},$$

где K не зависит от h , т.е. если ряды Тейлора для точного решения $w_1 = w(h)$ и для ω_1 совпадают до члена h^p включительно.

В случае, когда матрица A постоянна на всём промежутке интегрирования, т.е. матрица A в серии задач (2.1) — одна и та же, верно следующее

Утверждение 1. Пусть решение $w(t)$ задачи (2.3) достаточно гладко, а матрица A в серии задач (2.1) не зависит от номера n . Если решение задачи (2.3) осуществляется с помощью одного шага метода Рунге—Кутты (2.4) порядка p с шагом τ , то решение задачи (2.1), получаемое по правилу $\nu_1 = \exp(A\tau)\omega_1$, также имеет порядок аппроксимации p .

Доказательство. Поскольку A — линейный оператор, действующий в конечномерном пространстве, то он ограничен, т.е. $\|A\| < \infty$, поэтому существует и ограничен оператор $\exp(A)$ [13, с. 302], т.е. выполнено

$$\|e^{\tau A}b\| \leq \|e^{\tau A}\| \|b\|,$$

где b — произвольный вектор из линейного пространства, причём

$$\|e^{\tau A}\| \leq e^{\tau\|A\|} \leq e^{T\|A\|} = C,$$

поэтому, считая, что матричная экспонента вычисляется точно, получаем:

$$\|v_1 - \nu_1\| = \|e^{\tau A}(w_1 - \omega_1)\| \leq \|e^{\tau A}\| \|w_1 - \omega_1\| \leq CK\tau^{p+1} = M\tau^{p+1}, \quad (3.1)$$

где $M = CK$ не зависит от величины шага τ и номера отрезка $[t_n, t_{n+1}]$, что доказывает сходимость метода с порядком p . \square

В случае, когда матрица A является матрицей Якоби вида $A = f_u(u_n)$, необходимо сформулировать дополнительные условия на матрицу A линейной части в (2.1), которые позволили бы сохранить порядок аппроксимации.

Утверждение 2. Пусть решение $w(t)$ задачи (2.3) достаточно гладко, матрица $A = f_u(u_n)$, а функция $g(t, w)$ правой части имеет равномерно ограниченную производную (матрицу Якоби) на множестве $[0, T] \times W$, где W — окрестность решения $w(t)$. Если решение задачи (2.3) осуществляется с помощью одного шага метода Рунге—Кутты (2.4) порядка p с шагом τ , то решение задачи (2.1), получаемое по правилу $v_1 = \exp(A\tau)\omega_1$, также имеет порядок аппроксимации p .

Доказательство. Первые шаги доказательства аналогичны предыдущему случаю, но при этом необходимо учитывать, что величина $\|\exp(\tau A)\|$ может меняться от точки к точке. Считаем, что все матричные экспоненты вычисляются точным образом, тогда с учётом равномерной ограниченности матриц Якоби получаем то же неравенство (3.1), где C — константа равномерной ограниченности. Это гарантирует сходимость метода с порядком p . \square

Обратим внимание, что для истинности данных утверждений существенно, что матричные экспоненты, возникающие в расчётных формулах, вычисляются точно. Поскольку рассматриваемый метод (2.6), (2.7) реализован авторами в среде математического пакета MATLAB, то для вычисления матричной экспоненты использована встроенная функция `expm`, в которой заложен алгоритм построения диагональной (13,13)-аппроксимации Паде с масштабированием. В [14, с. 254] показано, что такой подход позволяет вычислять матричную экспоненту с погрешностью, сопоставимой с величиной ошибок округления. Таким образом, в рамках данной работы можно считать, что `expm` вычисляет $\exp(A\tau)$ точно.

Однако матричную экспоненту можно вычислять иными способами [8]: например, в работе [5] для этого применяется тот же четырёхстадийный метод Рунге—Кутты. Поэтому, строго говоря, для такого случая необходимо обобщить утверждение 2, считая, что матрица $\exp(A\tau)$ вычисляется не точно, а с некоторым порядком q .

Утверждение 3. Пусть решение $w(t)$ задачи (2.3) достаточно гладко, матрица $A = f_u(u_n)$, а функция $g(t, w)$ правой части имеет равномерно ограниченную производную (матрицу Якоби) на множестве $[0, T] \times W$, где W — окрестность решения $w(t)$. Если решение задачи (2.3) осуществляется с помощью одного шага метода Рунге—Кутты (2.4) порядка p с шагом τ , а соответствующие матричные экспоненты вычисляются с порядком q , то решение задачи (2.1), получаемое по правилу $v_1 = \exp(A\tau)\omega_1$, имеет порядок аппроксимации $\min(p, q)$.

Аналогичное обобщение можно получить и в случае утверждения 1.

4. Функция устойчивости

4.1. Вывод функции устойчивости

При решении жёстких задач наиболее остро встаёт вопрос об устойчивости численного метода, поэтому исследуем метод RK4exp с помощью стандартного теста: применим его к решению уравнения Далквиста [15, с. 26]

$$\dot{u} = \lambda u, \quad (4.1)$$

где λ — комплексное число. Цель теста — после формального применения численного метода к решению уравнения (4.1) получить выражение вида

$$y_{n+1} = R(\lambda\tau)y_n,$$

где $R(z)$ — искомая функция устойчивости метода. Докажем следующее

Утверждение 4. *Функция устойчивости метода RK4exp, определяемого формулами (2.6), (2.7), имеет вид:*

$$R(z) = 1 + \frac{z}{6} \left[1 + \left(1 + (1 - B)z + \frac{(1 - B)^2 z^2}{2} + \frac{(1 - B)^3 z^3}{4} \right) e^{Bz/2} + \left(4 + 2(1 - B)z + \frac{(1 - B)^2 z^2}{2} \right) e^{Bz} \right], \quad (4.2)$$

где $B = A/\lambda$ — частное линейной части A и величины λ .

Доказательство. Действуем по определению: в применении к решению уравнения (4.1) метод RK4exp принимает вид:

$$y_{n+1} = y_n + \nu_1,$$

где ν_1 определяется по формулам (2.6), (2.7). В обозначениях задач (0.1) и (2.1) уравнение Далквиста записывается следующим образом:

$$f(u) = \lambda u, \quad F(v) = f(y_n + v) - Av = \lambda y_n + (\lambda - A)v.$$

После этого вычисляем F_i по формулам (2.6):

$$F_1 = F(0) = f(y_n) = \lambda y_n, \\ F_2 = F\left((\tau/2)e^{A\tau/2}F_1\right) = \lambda y_n + \frac{(\lambda - A)\tau}{2}e^{A\tau/2}F_1.$$

Введём обозначения $B = A/\lambda$ и $z = \lambda\tau$, получим

$$F_2 = \lambda y_n + \frac{(1 - B)z}{2}e^{Bz/2}F_1 = \lambda \left[1 + \frac{(1 - B)z}{2}e^{Bz/2} \right] y_n.$$

Проводя аналогичные выкладки для $F_{3,4}$, можно получить следующее:

$$F_3 = \lambda \left[1 + \frac{(1-B)z}{2} + \frac{(1-B)^2 z^2}{4} e^{Bz/2} \right] y_n,$$

$$F_4 = \lambda \left[1 + (1-B)z e^{Bz/2} + \frac{(1-B)^2 z^2}{2} e^{Bz/2} + \frac{(1-B)^3 z^3}{4} e^{Bz} \right] y_n.$$

Домножим F_i на соответствующие множители, соберём их вместе и приведём подобные члены, тогда получим

$$y_{n+1} = y_n + \nu_1 = y_n + \frac{\tau}{6} \left(e^{A\tau} F_1 + 2e^{A\tau/2} F_2 + 2e^{A\tau/2} F_3 + F_4 \right) = R(z)y_n,$$

где $R(z)$ определяется по формуле (4.2). \square

4.2. Геометрическая иллюстрация области устойчивости

Рассмотрим наиболее простой для анализа случай $B = 1$, что соответствует $A = \lambda$, тогда функция $R(z)$ выглядит следующим образом:

$$R(z) = 1 + \frac{z}{6} \left(1 + 4e^{z/2} + e^z \right) = 1 + z \sum_{i=1}^4 c_i \exp(a_i z), \quad (4.3)$$

где $\{c_i\}_{i=1}^4$ и $\{a_i\}_{i=1}^4$ — коэффициенты классического явного четырёхстадийного метода Рунге—Кутты RK4 (см. раздел 2). В то же время функцией устойчивости метода RK4 является [15, с. 27]

$$R(z) = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!}.$$

Под областью устойчивости метода будем понимать область $D = \{z \in \mathbb{C} : |R(z)| < 1\}$. Обозначим через D_1 область устойчивости функции (4.3), т.е. область устойчивости метода RK4exp при $A = \lambda$, а через D_2 — область устойчивости исходного метода RK4.

Для сравнения на рис. 1 и 2 представлены геометрические иллюстрации областей D_1 и D_2 соответственно с изолиниями функций $|R(z)|$. По представленным рисункам видно, что метод RK4exp, как и RK4, не является A -устойчивым [2, с. 41], но при этом его область устойчивости значительно больше области устойчивости метода RK4 (примерно в четыре раза по действительной оси, и в два — по мнимой оси).

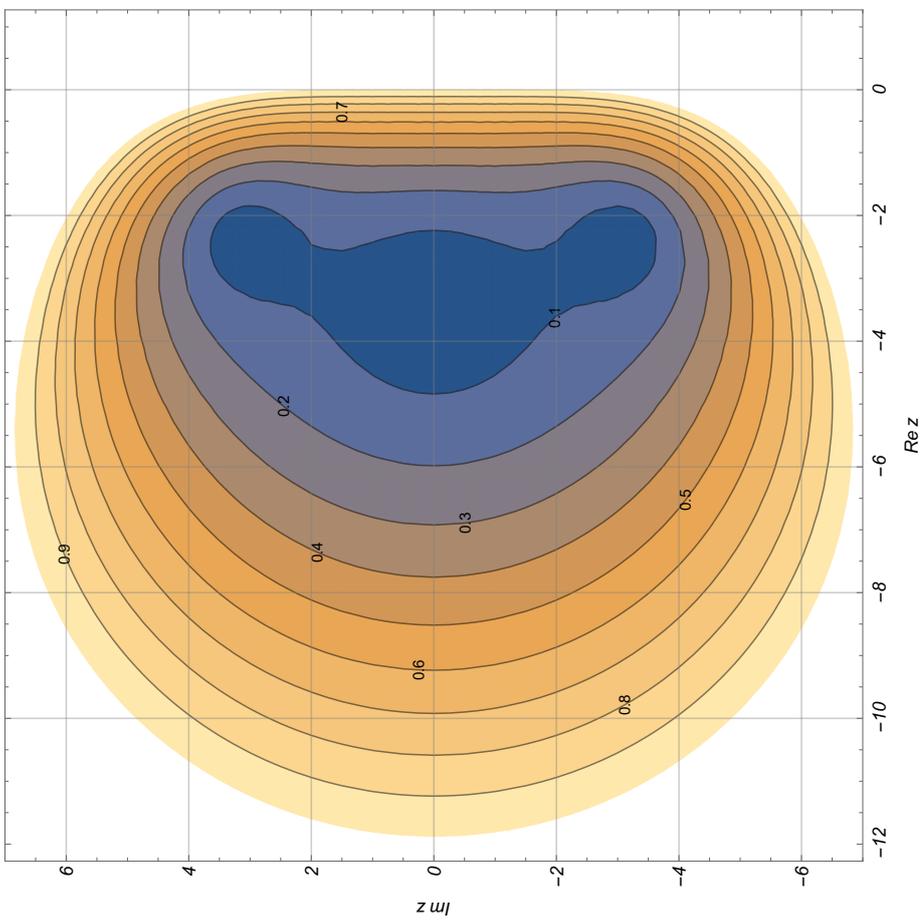


Рис. 1. Область устойчивости метода RK4exр при $B = 1$.

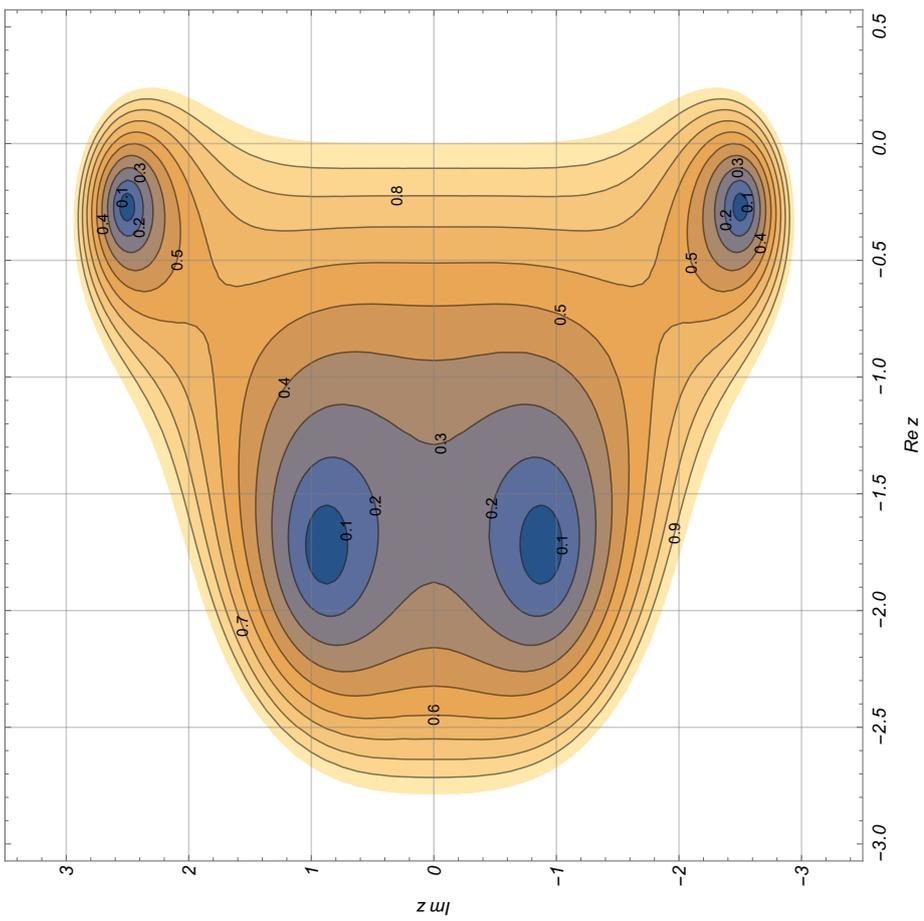


Рис. 2. Область устойчивости метода RK4.

Замечание. На самом деле область устойчивости D_1 является многосвязной — она содержит в себе три компоненты связности. На рис. 1 показана только самая крупная компонента связности. На рис. 3, 4 представлены остальные две компоненты связности области D_1 .

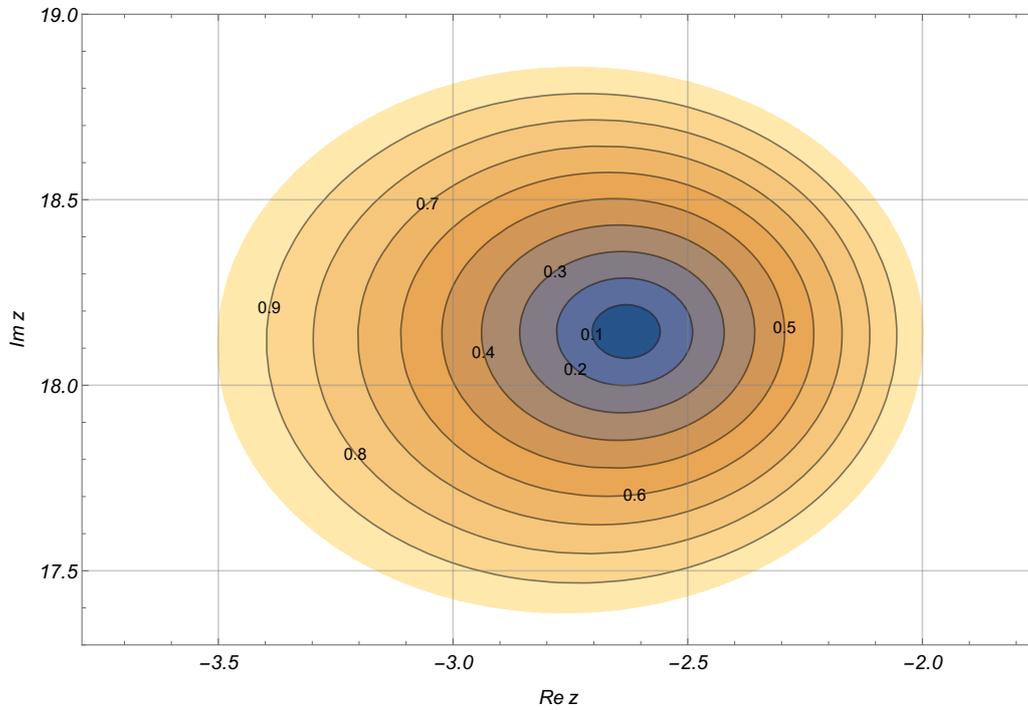


Рис. 3. Вторая компонента связности области D_1 .

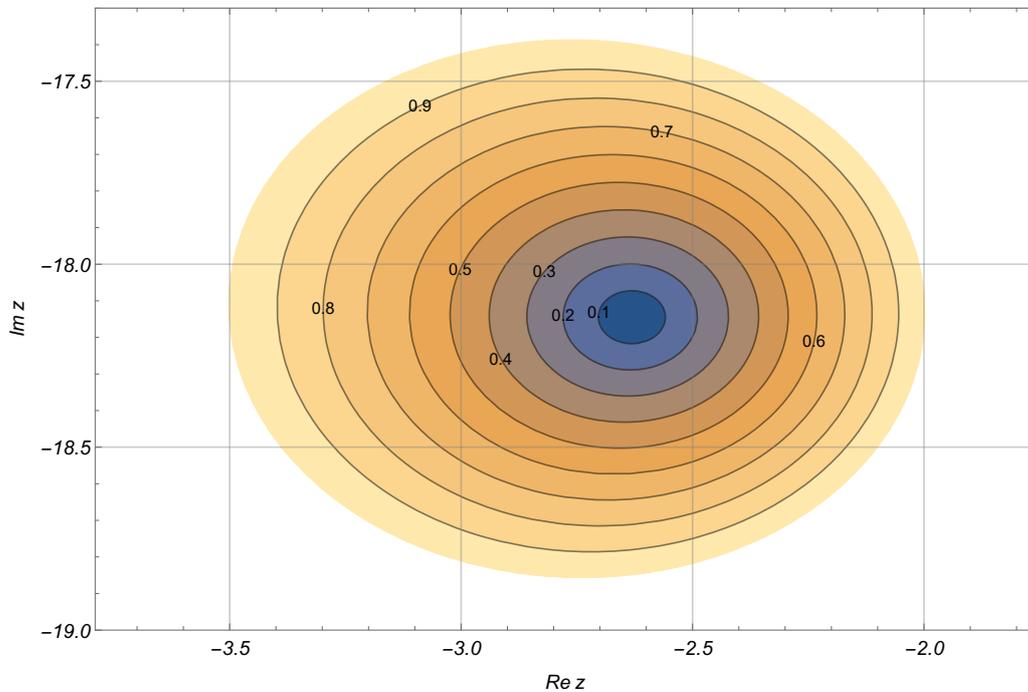


Рис. 4. Третья компонента связности области D_1 .

5. Результаты вычислительных экспериментов

В данном разделе представлены результаты работы сконструированного метода RK4exp на некоторых тестовых задачах из [5, 16]; среди них: линейные; нелинейные, допускающие аналитическое решение в замкнутом виде; и нелинейные, не допускающие представления аналитического решения в замкнутом виде. В последнем случае для проверки использовались результаты работы эталонного метода ode15s из пакета MATLAB, полученные с высокой степенью точности.

При решении тестовых задач применялись следующие способы задания матрицы A :

- 1) матрица $A = f_u(u_0)$ — матрица Якоби, вычисленная в единственной точке (т.е. на векторе начальных условий);
- 2) матрица $A = f_u(y_n)$ — матрица Якоби, пересчитываемая в каждый момент времени t_n (вычисляется на векторе $y_n \approx u(t_n)$);
- 3) матрица A вычисляется по иному правилу — этот случай отмечен отдельно в одном из примеров ниже.

5.1. Линейные задачи

Задача 1. Данная задача представляет собой «Тест 1» из работы [16] и определяется системой вида:

$$\begin{aligned}
 \dot{u}_1 &= \mu_0 u_1, \\
 \dot{u}_2 &= \dot{u}_1 - \mu_1 u_1 + (\mu_1 + \nu_1) u_2 - \nu_1 u_3, \\
 \dot{u}_3 &= \dot{u}_1 - (\mu_1 + \nu_1) u_1 + 2\nu_1 u_2 + (\mu_1 - \nu_1) u_3, \\
 \dot{u}_4 &= \dot{u}_3 - \mu_2 u_3 + (\mu_2 + \nu_2) u_4 - \nu_2 u_5, \\
 \dot{u}_5 &= \dot{u}_3 - (\mu_2 + \nu_2) u_3 + 2\nu_2 u_4 + (\mu_2 - \nu_2) u_5.
 \end{aligned} \tag{5.1}$$

При $u_2(0) = u_3(0)$, $u_4(0) = u_5(0)$ система имеет аналитическое решение вида:

$$\begin{aligned}
 u_1(t) &= u_1(0) e^{\mu_0 t}, \\
 u_2(t) &= u_1(t) + (u_2(0) - u_1(0)) e^{\mu_1 t} \cos \nu_1 t, \\
 u_3(t) &= u_1(t) + \sqrt{2} (u_2(0) - u_1(0)) e^{\mu_1 t} \sin(\nu_1 t + \pi/4), \\
 u_4(t) &= u_3(t) + (u_4(0) - u_2(0)) e^{\mu_2 t} \cos \nu_2 t, \\
 u_5(t) &= u_3(t) + \sqrt{2} (u_4(0) - u_2(0)) e^{\mu_2 t} \sin(\nu_2 t + \pi/4).
 \end{aligned}$$

Собственные значения матрицы правой части: $\lambda_1 = \mu_0$, $\lambda_{2,3} = \mu_1 \pm i\nu_1$, $\lambda_{3,4} = \mu_2 \pm i\nu_2$. Выбор параметров $\mu_0, \mu_1, \mu_2, \nu_1, \nu_2$ влияет на характер поведения системы. В таблице 1 представлены наборы параметров и начальных условий из работы [16] (при описании типов задач использована терминология [2, с. 40]).

Таблица 1. Наборы параметров для задачи 1

№	Тип задачи	μ_0	μ_1	μ_2	ν_1	ν_2	$u_1(0)$	$u_2(0)$	$u_4(0)$
1	плохо обусловленная	10	4	5	20π	10^2	0.1	1.0	0.5
2	мягкая	-2	1	-1	1	10	1.0	1.5	2.5
3	быстро осциллирующая	-2	1	-1	1	10^3	0.5	0.8	2.0
4	жёсткая	-10^2	-1	-10^4	1	10	10.0	11.0	111.0
5	жёстко осциллирующая	-10^4	1	-10^2	1	10^3	100.0	101.0	201.0

Во всех случаях расчёты проводились на отрезке $[0, 1]$. Использовались сетки с постоянным шагом $\tau = 10^{-k}$, $1 \leq k \leq 6$, при этом в каждом случае матрица A вычислялась единственный раз по правилу $A = f_u(u_0)$. Качество работы алгоритма оценивалось с помощью следующих величин:

$$\Delta = \max_{0 \leq i \leq N} \|y_i - u_i\|, \quad \delta = \max_{0 \leq i \leq N} \frac{\|y_i - u_i\|}{\|u_i\|},$$

представляющих собой поточечные максимумы (на сетке с узлами $\{t_i\}_{i=0}^N$) абсолютной и относительной ошибки соответственно; здесь y_i — значения численного решения на сетке, u_i — значения точного решения, спроектированные на ту же самую сетку. В качестве нормы $\|\cdot\|$ использовалась кубическая норма [1, с. 40].

Таблица 2. Результаты численного решения RK4exp задачи 1

№	Шаг интегрирования τ					
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
1	$1.49 \cdot 10^{10}$	$2.94 \cdot 10^0$	$3.44 \cdot 10^{-4}$	$3.45 \cdot 10^{-8}$	$4.57 \cdot 10^{-11}$	$1.65 \cdot 10^{-10}$
	$6.62 \cdot 10^6$	$1.28 \cdot 10^{-2}$	$1.91 \cdot 10^{-6}$	$1.94 \cdot 10^{-10}$	$6.69 \cdot 10^{-14}$	$2.62 \cdot 10^{-13}$
2	$1.79 \cdot 10^{-3}$	$1.83 \cdot 10^{-7}$	$1.82 \cdot 10^{-11}$	$2.41 \cdot 10^{-14}$	$1.46 \cdot 10^{-13}$	$2.69 \cdot 10^{-13}$
	$9.47 \cdot 10^{-4}$	$9.53 \cdot 10^{-8}$	$9.66 \cdot 10^{-12}$	$1.20 \cdot 10^{-14}$	$7.30 \cdot 10^{-14}$	$1.41 \cdot 10^{-13}$
3	$8.89 \cdot 10^{19}$	$2.25 \cdot 10^{93}$	$2.32 \cdot 10^{-1}$	$2.17 \cdot 10^{-5}$	$2.17 \cdot 10^{-9}$	$4.14 \cdot 10^{-13}$
	$4.92 \cdot 10^{19}$	$1.24 \cdot 10^{93}$	$2.21 \cdot 10^{-1}$	$2.13 \cdot 10^{-5}$	$2.13 \cdot 10^{-9}$	$4.29 \cdot 10^{-13}$
4	$1.56 \cdot 10^{24}$	$3.13 \cdot 10^{121}$	$7.12 \cdot 10^1$	$2.13 \cdot 10^{-2}$	$1.34 \cdot 10^{-6}$	$1.28 \cdot 10^{-10}$
	$3.06 \cdot 10^{24}$	$6.15 \cdot 10^{121}$	$7.08 \cdot 10^0$	$6.43 \cdot 10^{-4}$	$4.07 \cdot 10^{-8}$	$3.89 \cdot 10^{-12}$
5	$1.56 \cdot 10^{24}$	$3.15 \cdot 10^{121}$	$7.11 \cdot 10^1$	$2.13 \cdot 10^{-2}$	$1.34 \cdot 10^{-6}$	$1.28 \cdot 10^{-10}$
	$4.15 \cdot 10^{23}$	$8.37 \cdot 10^{120}$	$1.22 \cdot 10^0$	$1.46 \cdot 10^{-4}$	$9.36 \cdot 10^{-9}$	$8.95 \cdot 10^{-13}$

В таблице 2 представлены результаты работы алгоритма для каждого из пяти случаев: в каждой из ячеек представлены значения Δ (верхняя величина) и δ (нижняя величина), соответствующие номеру случая и величине шага интегрирования τ . Обсудим каждый из случаев.

Случай 1. Несмотря на плохую обусловленность задачи, алгоритм RK4exp справляется с поставленной задачей — отчасти благодаря наличию $\exp(\tau A)$ в расчётных формулах в явном виде. По изменению величин Δ и δ при уменьшении шага τ видно, что метод действительно имеет четвёртый

порядок сходимости, как то гарантирует утверждение 1. Нарастание погрешности на самых мелких шагах объясняется тем, что наибольший вклад в погрешность начинает вносить машинная арифметика, а не ошибка дискретизации алгоритма.

Случай 2. Задача является хорошо обусловленной, поэтому уже при относительно крупном шаге $\tau = 10^{-4}$ достигается абсолютная погрешность порядка 10^{-14} , при дальнейшем уменьшении шага наблюдается преобладание ошибок машинной арифметики.

Случай 3. Задача не подпадает под определение жёсткой (см. (0.2)), но при этом результаты работы RK4exp очень похожи на результаты для случаев 4 и 5. Если перейти к комплексному аргументу $z = \lambda\tau$, подобные явления описываются единым образом: при крупных величинах шага τ аргумент z оказывается далёк от области устойчивости RK4exp (см. рис. 1).

Случаи 4,5. Несмотря на относительно небольшую область устойчивости, алгоритм достаточно быстро достигает величины погрешности, всего на один-два порядка превышающей порядок машинного нуля.

Во всех трёх случаях 3, 4, 5 алгоритм продемонстрировал хорошее качество работы и сходимость с четвёртым порядком.

Задача 2. «Тест 2» из работы [16] задаётся линейной системой с двумя жордановыми клетками вида:

$$\begin{aligned} \dot{u}_1 &= \mu_1 u_1, \\ \dot{u}_2 &= u_1 + \mu_1 u_2, \\ \dot{u}_3 &= \mu_2 u_3, \\ \dot{u}_4 &= u_3 + \mu_2 u_4, \\ \dot{u}_5 &= 2u_4 + \mu_2 u_5, \\ \dot{u}_6 &= 3u_5 + \mu_2 u_6. \end{aligned} \tag{5.2}$$

Особенностью системы является наличие двух кратных собственных значений $\lambda_1 = \mu_1$ и $\lambda_2 = \mu_2$ (кратностей 2 и 4 соответственно) у матрицы правой части.

Для тестирования использовался следующий набор параметров: начальные условия $u_1(0) = u_2(0) = 1$, $u_3(0) = u_4(0) = u_5(0) = u_6(0) = 10^3$, параметры жёсткости $\mu_1 = -1$, $\mu_2 = -10^4$. Результаты представлены в таблице 3, величины Δ и δ имеют тот же смысл, что и в таблице 2: это поточечные максимумы абсолютной и относительной погрешностей соответственно.

Таблица 3. Результаты численного решения RK4exp задачи 2

	Шаг интегрирования τ					
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Δ	$1.56 \cdot 10^{25}$	$3.15 \cdot 10^{122}$	$7.12 \cdot 10^2$	$2.13 \cdot 10^{-1}$	$1.34 \cdot 10^{-5}$	$1.28 \cdot 10^{-9}$
δ	$2.12 \cdot 10^{26}$	$4.27 \cdot 10^{123}$	$7.12 \cdot 10^3$	$5.19 \cdot 10^{-3}$	$3.35 \cdot 10^{-7}$	$3.21 \cdot 10^{-11}$

По результатам, представленным в таблице 3, видно, что уменьшение погрешности задачи началось с величины шага $\tau = 10^{-3}$: в этом случае ком-

плексная величина $z = \lambda_2 \tau = -10$, как видно на рис. 1, попадает внутрь области устойчивости метода.

5.2. Нелинейные задачи

При решении нелинейных задач метод вычисления матрицы A сильно влияет на качество сходимости алгоритма, поэтому для каждой нелинейной задачи представлены результаты работы как минимум двух вариантов алгоритма RK4exp: с фиксированной матрицей $A = f_u(u_0)$ и матрицей A , пересчитываемой в каждой точке по правилу $A = f_u(y_n)$.

Задача 3. Рассмотрим следующую задачу Коши с нелинейной правой частью [5, 7]:

$$\begin{aligned} \dot{u}_1 &= \alpha u_1^2 u_2, \\ \dot{u}_2 &= -\alpha u_1 u_2^2; \end{aligned} \quad (5.3)$$

при $u_1(0) = u_2(0) = 1$ задача имеет аналитическое решение

$$\begin{aligned} u_1(t) &= e^{\alpha t}, \\ u_2(t) &= e^{-\alpha t}. \end{aligned}$$

Далее представлены результаты решения (5.3) с помощью RK4exp на отрезке $[0, 1]$ при различных значениях $\alpha = 1, 10, 100$. Все задачи решались на равномерных сетках.

Вариант 1. Матрица A вычислялась единственный раз: $A = f_u(u_0)$, где u_0 — вектор начальных условий. Результаты работы данного варианта RK4exp представлены в таблице 4: информация в ячейках таблицы расположена по тому же принципу, что и в таблице 2; в некоторых ячейках проставлены прочерки, поскольку для соответствующих величин шагов τ расчёт не производился.

Таблица 4. Результаты работы RK4exp с фиксированной $A = f_u(u_0)$

		Шаг интегрирования τ					
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
α	1	$7.90 \cdot 10^{-5}$	$7.83 \cdot 10^{-9}$	$7.93 \cdot 10^{-13}$	$3.42 \cdot 10^{-14}$	$4.22 \cdot 10^{-14}$	—
		$2.91 \cdot 10^{-5}$	$2.88 \cdot 10^{-9}$	$2.92 \cdot 10^{-13}$	$1.41 \cdot 10^{-14}$	$1.72 \cdot 10^{-14}$	—
	10	$1.29 \cdot 10^5$	$2.20 \cdot 10^4$	$2.18 \cdot 10^4$	$1.99 \cdot 10^4$	$3.84 \cdot 10^3$	$5.40 \cdot 10^{-1}$
		$5.86 \cdot 10^1$	$1.00 \cdot 10^0$	$9.90 \cdot 10^{-1}$	$9.01 \cdot 10^{-1}$	$1.74 \cdot 10^{-1}$	$2.45 \cdot 10^{-5}$
	100	—	$2.68 \cdot 10^{43}$	$2.68 \cdot 10^{43}$	$2.68 \cdot 10^{43}$	$2.68 \cdot 10^{43}$	$2.68 \cdot 10^{43}$
		—	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$

Из таблицы 4 видно, что в случае $\alpha = 100$ данный вариант алгоритма не позволил получить удовлетворительных результатов даже при шаге интегрирования $\tau = 10^{-6}$. Чтобы объяснить это, рассмотрим матрицу Якоби правой

части на точном решении:

$$f_u(u(t)) = J(t) = \begin{pmatrix} 2\alpha u_1 u_2 & \alpha u_1^2 \\ -\alpha u_2^2 & -2\alpha u_1 u_2 \end{pmatrix} = \begin{pmatrix} 2\alpha & \alpha e^{2\alpha t} \\ -\alpha e^{-2\alpha t} & -2\alpha \end{pmatrix},$$

в нашем случае

$$A = f_u(u_0) = J(0) = \begin{pmatrix} 2\alpha & \alpha \\ -\alpha & -2\alpha \end{pmatrix}.$$

Видно, что уже при малых t (при $\alpha = 100$) матрицы A и $J(t)$ значительно отличаются, поэтому градиент функции $F[v] = f(u_n + v) - Av$ при удалении от точки $t = 0$ сильно растёт, а потому интегрирование задачи (2.3) требует мелкого шага.

Вариант 2. Учитывая результаты предыдущего варианта, произведём расчёт методом RK4exp с помощью переменной матрицы A , определяемой на каждом $(n + 1)$ -ом шаге по правилу $A = f_u(y_n)$, где y_n — численное решение на n -ом шаге. Результаты представлены в таблице 5.

Таблица 5. Результаты работы RK4exp с переменной $A = f_u(y_n)$

		Шаг интегрирования τ					
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
α	1	$6.76 \cdot 10^{-6}$	$7.54 \cdot 10^{-10}$	$6.93 \cdot 10^{-14}$	$3.55 \cdot 10^{-14}$	$4.22 \cdot 10^{-14}$	—
		$2.49 \cdot 10^{-6}$	$2.77 \cdot 10^{-10}$	$2.55 \cdot 10^{-14}$	$1.46 \cdot 10^{-14}$	$1.72 \cdot 10^{-14}$	—
	10	$1.21 \cdot 10^4$	$1.38 \cdot 10^{-1}$	$5.70 \cdot 10^{-5}$	$5.96 \cdot 10^{-9}$	$2.39 \cdot 10^{-9}$	$5.85 \cdot 10^{-9}$
		$5.51 \cdot 10^{-1}$	$8.78 \cdot 10^{-6}$	$2.59 \cdot 10^{-9}$	$2.71 \cdot 10^{-13}$	$1.09 \cdot 10^{-13}$	$2.66 \cdot 10^{-13}$
	100	—	$2.69 \cdot 10^{43}$	$4.83 \cdot 10^{40}$	$1.95 \cdot 10^{35}$	$6.56 \cdot 10^{31}$	$1.73 \cdot 10^{32}$
		—	$1.00 \cdot 10^0$	$1.80 \cdot 10^{-3}$	$9.44 \cdot 10^{-9}$	$2.44 \cdot 10^{-12}$	$6.43 \cdot 10^{-12}$

Результаты таблицы 5 намного лучше представленных в таблице 4; в том числе удалось получить удовлетворительные результаты для случая $\alpha = 100$ — это объясняется тем, что построение A в каждой точке позволяет избежать нарастания отклонения A от $J(t)$. Также метод сходится с четвертым порядком аппроксимации, как это следует из утверждения 2.

Вариант 3. Как отмечено выше, в общем случае матрица A не обязана быть матрицей Якоби — её можно построить иным способом. Воспользуемся тем фактом, что нам известно решение в явном виде: оно также является решением следующей линейной задачи:

$$\begin{aligned} \dot{u} &= B_\alpha u, \\ u(0) &= u_0; \end{aligned}$$

где $B_\alpha = \text{diag}(\alpha, -\alpha)$ — диагональная матрица. В некотором смысле можно считать, что B_α — это матрица собственных значений для задачи (5.3), поэтому положим $A = B_\alpha$ и проведём аналогичную серию расчётов. Результаты

Таблица 6. Результаты работы RK4exp с фиксированной $A = B_\alpha$

		Шаг интегрирования τ					
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
α	1	$8.62 \cdot 10^{-6}$	$9.45 \cdot 10^{-10}$	$8.48 \cdot 10^{-14}$	$3.42 \cdot 10^{-14}$	$4.22 \cdot 10^{-14}$	—
		$3.17 \cdot 10^{-6}$	$3.48 \cdot 10^{-10}$	$3.13 \cdot 10^{-14}$	$1.41 \cdot 10^{-14}$	$1.72 \cdot 10^{-14}$	—
	10	$1.42 \cdot 10^4$	$3.84 \cdot 10^{-1}$	$7.34 \cdot 10^{-5}$	$7.66 \cdot 10^{-9}$	$2.88 \cdot 10^{-9}$	$5.84 \cdot 10^{-9}$
		$6.43 \cdot 10^{-1}$	$1.74 \cdot 10^{-5}$	$3.33 \cdot 10^{-9}$	$3.48 \cdot 10^{-13}$	$1.31 \cdot 10^{-13}$	$2.65 \cdot 10^{-13}$
	100	—	$2.69 \cdot 10^{43}$	$3.36 \cdot 10^{40}$	$5.14 \cdot 10^{35}$	$6.35 \cdot 10^{31}$	$1.93 \cdot 10^{32}$
		—	$1.00 \cdot 10^0$	$1.25 \cdot 10^{-3}$	$1.91 \cdot 10^{-8}$	$2.37 \cdot 10^{-12}$	$7.18 \cdot 10^{-12}$

представлены в таблице 6: видно, что они значительно лучше представленных в таблице 4 и при этом незначительно уступают по качеству результатам из таблицы 5.

Следовательно, если в более общем случае удастся построить такую же «оптимальную» матрицу A , это позволит не проводить линеаризацию правой части на каждом шаге интегрирования, что существенно уменьшит объём вычислительной работы. Таким образом, возникает задача: как наиболее эффективно построить матрицу A , применение которой позволит достичь наивысшей точности? В дальнейшем планируется представить различные методы построения матриц подобного класса и исследовать их эффективность при решении жёстких систем ОДУ.

Задача 4. Нелинейный осциллятор Ван дер Поля:

$$\begin{aligned} \dot{u}_1 &= u_2, \\ \dot{u}_2 &= \mu^2((1 - u_1^2)u_2 - u_1); \end{aligned} \quad (5.4)$$

с начальными условиями $u_1(0) = 2$, $u_2(0) = 0$, параметр μ^2 определяет жёсткость задачи. Поскольку точное решение для задачи в общем виде неизвестно, в качестве «эталонного» использовалось численное решение, полученное с помощью функции `ode15s` из пакета MATLAB.

Представленный в разделе 2 алгоритм RK4exp не содержит в себе методики подбора шага — для его работы необходимо предварительно определить сетку $\{t_k\}_{k=0}^N$. Для RK4exp сетка строилась с помощью алгоритма, реализованного в работе [7] (назовём его IER1, т.е. неявный метод Эйлера—Розенброка первого порядка): задача (5.4) решалась методом IER1, построенная им сетка G_0 использовалась в качестве исходной для RK4exp. На эту же сетку проектировалось эталонное решение, полученное с помощью `ode15s` с параметрами `AbsTol = 1e-15` и `RelTol = 1e-15`.

Подобные действия объясняются следующим образом: решение задачи (5.4) характеризуется тем, что в определённые моменты времени t^* одна из его компонент очень резко меняет своё значение (фактически происходит «скачок»), поэтому главная сложность для алгоритма — это вычисление значений решения именно в окрестностях подобных точек, что требует локального сгущения сетки. В качестве иллюстрации на рис. 5 представлен

график изменения компоненты u_1 решения задачи (5.4) на отрезке $[0, 10]$. Компонента u_2 испытывает «скачки» намного больше и поэтому на рисунке не представлена.

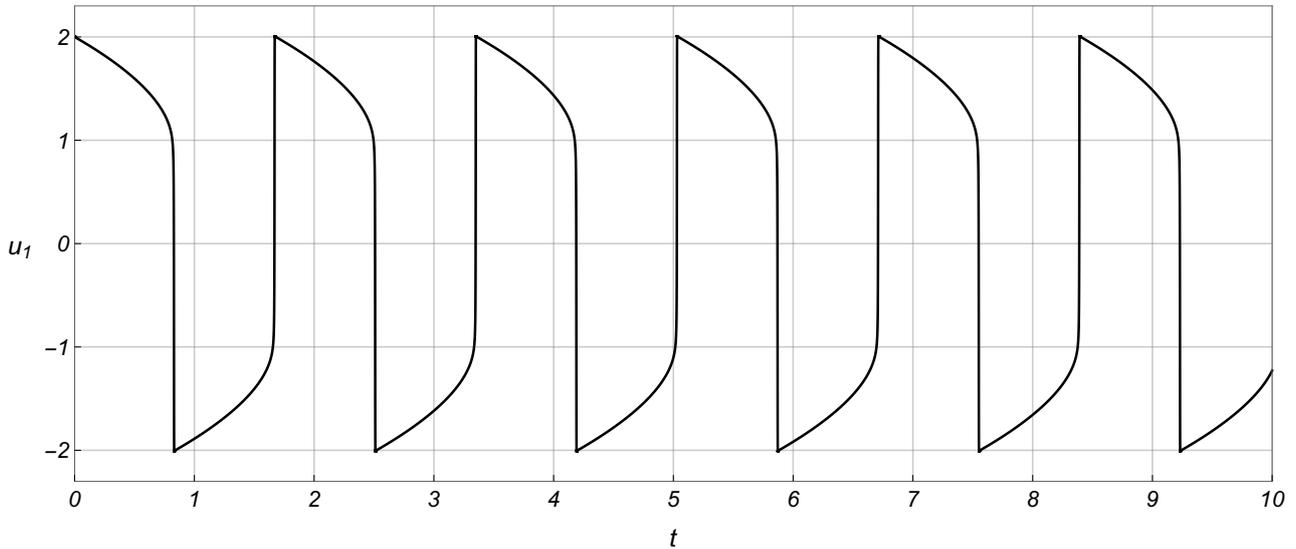


Рис. 5. Компонента u_1 решения задачи (5.4) при $t \in [0, 10]$.

Как показано в работе [7], упомянутый алгоритм IER1 автоматически сгущает сетку в окрестностях скачков и разрежает её в иных областях, поэтому он использовался для построения сетки. Для демонстрации сходимости метода RK4exр полученная сетка последовательно сгущалась в $q = 10$ раз. Это значит, что если G_0 — исходная сетка, а G_1 — полученная в результате сгущения G_0 в q раз, то верно:

- 1) $G_0 \subset G_1$;
- 2) для любых x_i, x_{i+1} из G_0 существуют $y_j \in G_1$, $1 \leq j \leq q - 1$, такие, что $y_j = x_i + jh_i$, где $h_i = |x_{i+1} - x_i|/q$.

Расчёты проводились для случая $\mu^2 = 1000$ как «оптимального» в смысле жёсткости задачи и объёма вычислительной работы. Результаты работы метода RK4exр в случае фиксированной $A = f_u(u_0)$ и переменной $A = f_u(y_n)$ представлены в таблице 7 (в столбцах под номерами I и II соответственно); сетка G_0 создана в результате работы алгоритма IER1, а G_1 и G_2 — её последовательные сгущения в $q = 10$ раз; поскольку все они имеют переменный шаг, в таблице дополнительно указаны минимальный τ_{\min} и максимальный τ_{\max} шаги сетки.

Как и в предыдущем примере, результаты работы алгоритма с фиксированной матрицей оказались хуже. При этом по результатам, представленным для обоих случаев, сложно сказать, что метод имеет четвёртый порядок сходимости. Объясняется это всё теми же «скачками» решения: наибольшее изменение испытывает вторая компонента решения u_2 , поэтому наибольший вклад в величину погрешности вносит разность $|u_2 - y_2|$. Ниже представлены

Таблица 7. Результаты работы RK4exр для осциллятора Ван дер Поля

$\mu^2 = 10^3$	Сетка					
	G_0		G_1		G_2	
	I	II	I	II	I	II
Δ	$1.34 \cdot 10^3$	$1.34 \cdot 10^3$	$4.68 \cdot 10^2$	$3.11 \cdot 10^0$	$5.10 \cdot 10^{-2}$	$9.75 \cdot 10^{-3}$
δ	$6.20 \cdot 10^2$	$8.61 \cdot 10^2$	$7.97 \cdot 10^{-1}$	$8.11 \cdot 10^{-3}$	$1.99 \cdot 10^{-4}$	$9.38 \cdot 10^{-5}$
τ_{\max}	$1.00 \cdot 10^{-3}$		$1.00 \cdot 10^{-4}$		$1.00 \cdot 10^{-5}$	
τ_{\min}	$3.13 \cdot 10^{-5}$		$3.13 \cdot 10^{-6}$		$3.13 \cdot 10^{-7}$	

графики, демонстрирующие качество сходимости компоненты y_2 численного решения.

На рис. 6 представлены графики численных решений поставленной задачи с помощью подпрограммы `ode15s` и метода IER1 на сетке G_0 : заметно отставание профиля численного решения y_2 от профиля эталонного решения u_2 в окрестности точки «скачка» $t = 8.39$. В работе [7] показано, что дальнейшее измельчение сетки G_0 не приводит к существенному сокращению отставания численного решения y_2 , полученного методом IER1, от точного.

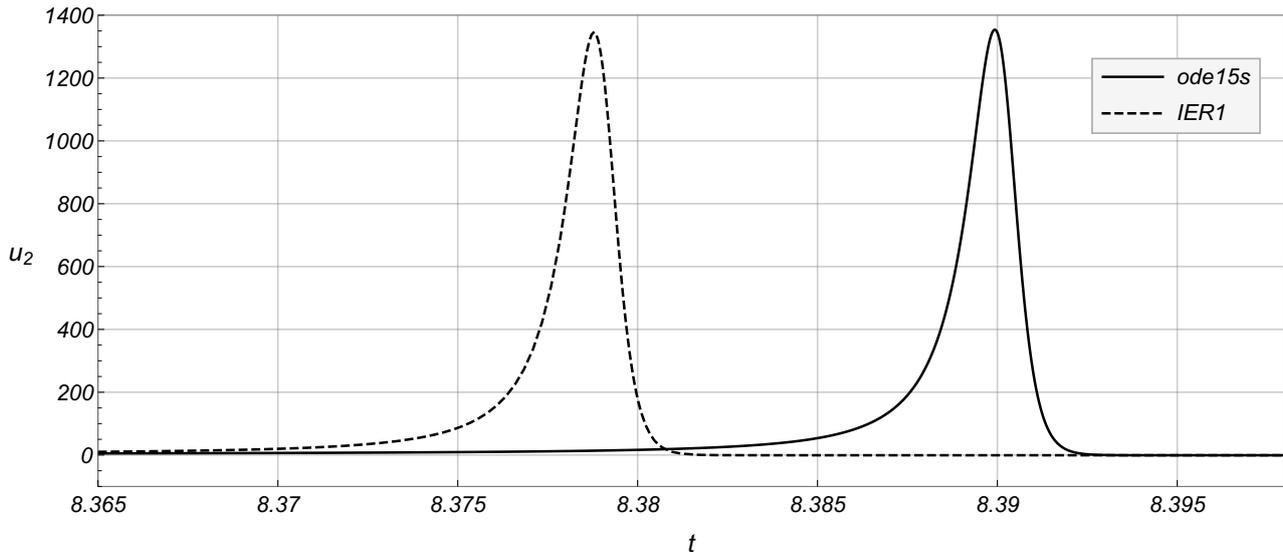


Рис. 6. Решение u_2 и приближение к нему, полученное с помощью IER1 при $t = 8.39$.

На рис. 7 показаны профили u_2 эталонного решения и численного решения y_2 , полученного методом RK4exр на сетке G_0 (матрица A фиксирована): видно, что профиль численного решения отстаёт от эталонного решения; на рис. 8, 9 показаны случаи решения той же задачи методом RK4exр на сетках G_1, G_2 соответственно в сравнении с эталонным решением (A фиксирована); на рис. 9 для обозначения решения по методу RK4exр используются дополнительные маркеры, поскольку расхождение профилей на сетке G_2 практически отсутствует.

Подобная картина наблюдается и в случае переменной $A = f_u(y_n)$, но качество приближённого решения оказывается выше.

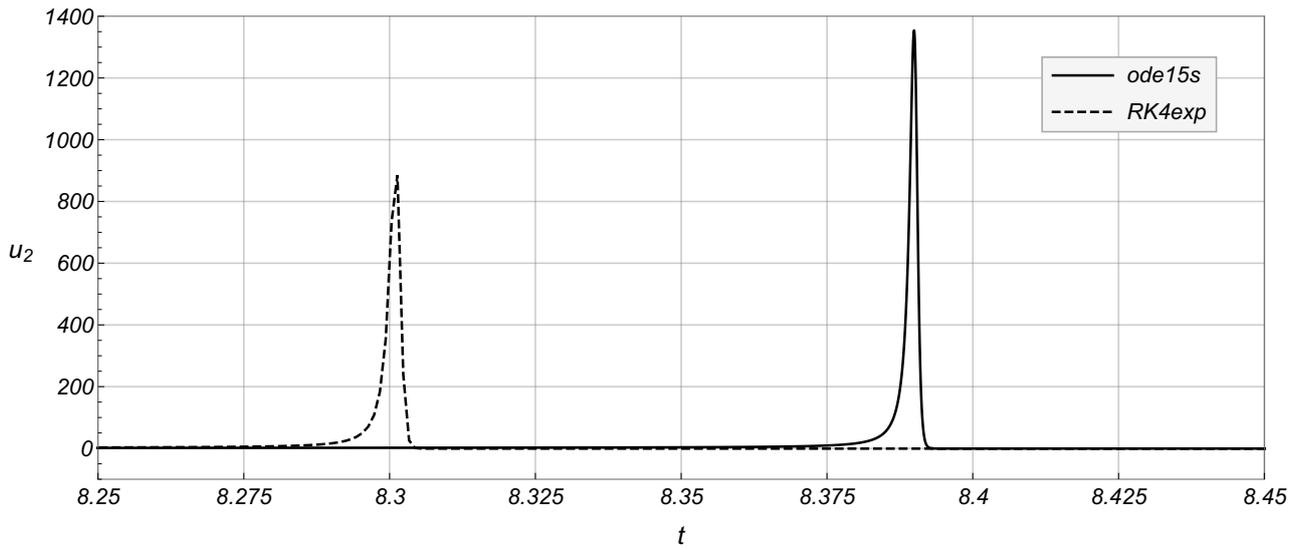


Рис. 7. Решение u_2 и решение по методу RK4exp на сетке G_0 при $t = 8.39$.

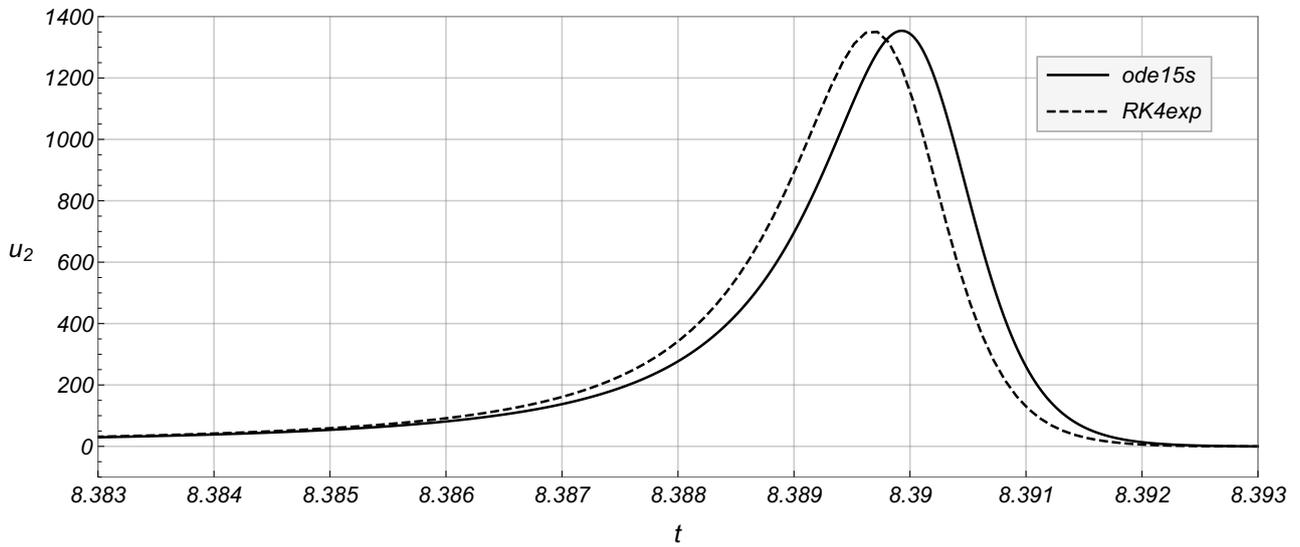


Рис. 8. Решение u_2 и решение по методу RK4exp на сетке G_1 при $t = 8.39$.

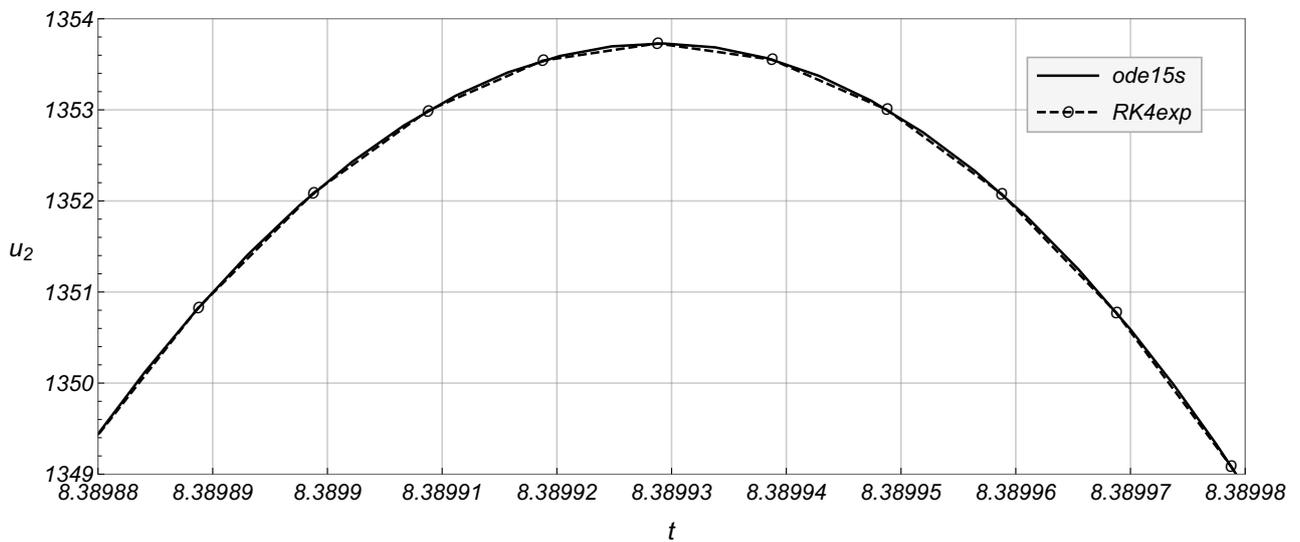


Рис. 9. Решение u_2 и решение по методу RK4exp на сетке G_2 при $t = 8.39$.

Таким образом, метод RK4exp не только позволил добиться качественной передачи численным решением эталонного профиля, но и практически ликвидировал их расхождение. Подобного результата не удавалось добиться при решении данной задачи методом IER1 даже на более подробных сетках [7].

Заключение

Представлена технология экспоненциальной замены, с помощью которой на основе классического метода RK4 получен его экспоненциальный аналог RK4exp. В достаточно общем случае доказано, что экспоненциальный аналог сохраняет аппроксимационные свойства метода, на котором он основан — это позволяет относительно просто конструировать экспоненциальные аналоги уже известных методов Рунге—Кутты. В общем случае свойства устойчивости получаемых экспоненциальных методов исследованы не были, но в частном случае для метода RK4exp построена функция устойчивости и получены геометрические иллюстрации области устойчивости.

Результаты вычислительных экспериментов для линейных и части нелинейных задач согласуются с выводами утверждений 1–2. Несоответствие результатов расчётов для нелинейного осциллятора Ван дер Поля теоретическим выводам объяснено наличием «скачков» точного решения; достаточно высокое качество приближения профиля эталонного решения численным в окрестности точки скачка наглядно проиллюстрировано с помощью серии рисунков.

При решении нелинейной задачи с экспоненциальным решением обнаружен следующий факт: если в версии алгоритма RK4exp с фиксированной матрицей A использовать специальную матрицу B_α , то это позволяет решать задачу почти так же эффективно, как и в случае алгоритма с переменной матрицей A , приближающей матрицу Якоби. Описанное явление приводит к постановке следующей задачи: как в общем случае построить подобную специальную матрицу A , позволяющую решить задачу с наименьшими вычислительными затратами?

Список литературы

- [1] *Галанин М.П., Савенков Е.Б.* Методы численного анализа математических моделей. М.: Изд-во МГТУ им. Н.Э.Баумана. 2010. 591 с.
- [2] Численные методы. В 2 кн. Кн. 2. Методы математической физики /Н.Н. Калиткин, П.В. Корякин. М.: Издательский центр «Академия». 2013. 304 с.
- [3] *Hochbruck M.* A short course on exponential integrators //Matrix Functions and Matrix Equations. Singapore: World Scientific Publishing Company. 2015. pp. 28–49.
- [4] *Филлипов А.Ф.* Введение в теорию дифференциальных уравнений. М.: КомКнига. 2007. 240 с.
- [5] *Галанин М.П., Ходжаева С.Р.* Методы решения жестких обыкновенных дифференциальных уравнений. Результаты тестовых расчетов //Препринты ИПМ им. М.В. Келдыша. 2013. № 98. 29 с.
- [6] *Новиков Е.А.* Исследование $(m, 2)$ -методов решения жёстких систем. //Вычислительные технологии. 2007. Т.12. №5. С.103–115.
- [7] *Галанин М.П., Конев С.А.* Об одном численном методе решения обыкновенных дифференциальных уравнений //Препринты ИПМ им. М.В. Келдыша. 2017. № 18. 28 с.
- [8] *Moler C., van Loan C.* Nineteen dubious ways to compute the exponential of a matrix, 25 years later //SIAM Review. 2003. V. 45. № 1. pp. 3–49.
- [9] *Tokman M.* Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. Journal of Computational Physics. 213, 2006. pp. 748–776.
- [10] *Niesen J., Wright W.* Algorithm 919: A Krylov subspace algorithm for evaluating the φ -functions appearing in exponential integrators. ACM Trans. Math. Softw. 38, 3. Article 22 (April 2012). 19 p.
- [11] *Федоренко Р.П.* Введение в вычислительную физику. Долгопрудный: Издательский дом «Интеллект». 2008. 504 с.
- [12] *Хайрер Э., Нёрсетт С., Ваннер Г.* Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. М.: Мир. 1990. 512 с.
- [13] *Кострикин А.И.* Введение в алгебру. Часть II. Линейная алгебра. М.: Физико-математическая литература. 2000. 368 с.
- [14] *Higham N.J.* Function of Matrices. Theory and computation. Philadelphia: SIAM. 2007. 425 p.

- [15] *Хайрер Э., Ваннер Г.* Решение обыкновенных дифференциальных уравнений. Жёсткие и дифференциально-алгебраические задачи. М.: Мир. 1999. 685 с.
- [16] *Арушанян О.Б., Залеткин С.Ф., Калиткин Н.Н.* Тесты для вычислительного практикума по обыкновенным дифференциальным уравнениям //Вычислительные методы и программирование. 2002. Т.3. С.11–19.

Содержание

Введение	3
1. Постановка задачи	3
2. Формулировка метода	5
3. Доказательство сходимости	7
4. Функция устойчивости	9
4.1. Вывод функции устойчивости	9
4.2. Геометрическая иллюстрация области устойчивости	10
5. Результаты вычислительных экспериментов	13
5.1. Линейные задачи	13
5.2. Нелинейные задачи	16
Заключение	22