



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

Препринты ИПМ • Препринт № 120 за 2020 г.



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

С.А. Конев

Моделирование горения
углеводородных топлив с
помощью метода EPIRK4(3)

Рекомендуемая форма библиографической ссылки: Конев С.А. Моделирование горения углеводородных топлив с помощью метода EPIRK4(3) // Препринты ИПМ им. М.В.Келдыша. 2020. № 120. 28 с. <https://doi.org/10.20948/prepr-2020-120>
<https://library.keldysh.ru/preprint.asp?id=2020-120>

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

С.А. Конев

Моделирование горения углеводородных топлив
с помощью метода EPIRK4(3)

Москва — 2020

Конев С.А.

Моделирование горения углеводородных топлив с помощью метода EPIRK4(3)

Работа посвящена экспоненциальным методам интегрирования жёстких систем ОДУ. Описан опыт реализации известных методов семейств EPIRK3 и EPIRK4. Показано, что методы семейства EPIRK4 не обладают четвёртым порядком сходимости. Представлены обновлённые методы семейства EPIRK4, обладающие четвёртым порядком сходимости. Новые методы протестированы на примере задачи горения углеводородных топлив.

Ключевые слова: численные методы; жёсткие системы; экспоненциальные методы; сажеобразование.

Stanislav Andreevich Konev

Hydrocarbon fuels combustion simulation with EPIRK4(3) method

The paper deals with exponential methods for solving stiff systems of ODEs. The experience of implementation EPIRK3 and EPIRK4 methods is described. It is shown, that EPIRK4 methods do not converge with order four. Refined EPIRK4 methods that converge with order four are constructed. The new methods are tested with the hydrocarbon fuel combustion problem.

Key words: numerical methods; stiff systems; exponential integrators; soot formation.

Работа выполнена при финансовой поддержке РФФИ (проекты № 18-01-00252 и № 19-01-00602).

Введение

Решение многих современных задач науки и техники можно свести к решению задачи Коши для обыкновенных дифференциальных уравнений (ОДУ) или их систем. В большинстве случаев такие задачи являются жёсткими, что объясняется разномасштабностью скоростей протекания процессов системы ([1, §IV.1], [2, §5.4]). Одной из таких задач является моделирование горения углеводородных топлив (УВТ) в ламинарном диффузионном пламени [3, 4].

Для численного решения жёстких задач используют специальные методы. Например, в упомянутых работах [3, 4] используется L -устойчивый метод Розенброка четвёртого порядка с вложенным методом третьего порядка (т.н. метод ROS4(3)L — см. [1, §IV.7]).

Однако в последнее время наблюдается большой интерес вычислителей к экспоненциальным методам интегрирования жёстких систем ОДУ (см. обзор [5] и ссылки там же). Это связано с высокими вычислительными качествами экспоненциальных методов: при тех же шагах интегрирования они обеспечивают лучшую точность и меньшую трудоёмкость на шаг по сравнению с самыми устойчивыми классическими неявными методами.

В данной работе отражён опыт реализации метода EPIRK4 из работы [6] и применения его к моделированию горения углеводородных топлив. Работа имеет следующую структуру. В разделе 1 представлена формулировка методов EPIRK и даны комментарии по поводу их реализации. В разделе 2 представлены результаты работы методов семейства EPIRK4 на специальном тестовом примере. В разделе 3 представлены уточнённые условия порядка и в результате эксперимента найдены коэффициенты обновлённого метода EPIRK4. Показано, что обновлённый метод действительно имеет четвёртый порядок сходимости. В разделе 4 на примере задачи о горении углеводородных топлив (УВТ) представлено сравнение методов ROS4(3)L и EPIRK4(3). В заключении подведены итоги работы.

1. Формулировка методов EPIRK

В разделе представлена формулировка методов семейства EPIRK и даны комментарии по поводу их реализации.

1.1. Постановка задачи

Допустим, что для автономной системы ОДУ поставлена задача Коши

$$u' = F(u), \quad 0 < t \leq T, \quad (1.1a)$$

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

где $u = u(t) = (u^1(t), \dots, u^d(t))^T$ — точное решение, $u_0 = (u_0^1, \dots, u_0^d)^T$ — вектор начальных условий, $F(u) = (F^1(u), \dots, F^d(t, u))$ — вектор-функция правой части, d — размерность задачи (здесь и далее используются верхние индексы для нумерации компонент векторов).

Считаем, что задача (1.1) является жёсткой. Введём на отрезке $[0, T]$ произвольную сетку $\{t_i\}_{i=0}^N$:

$$0 = t_0 < t_1 < \dots < t_N = T$$

и построим интегральное представление решения на произвольном отрезке $[t_n, t_{n+1}]$.

Выделим линейную часть функции F в окрестности $u_n = u(t_n)$:

$$u' = F_n + J(u - u_n) + R_n(u), \quad t \in [t_n, t_{n+1}], \quad (1.2a)$$

$$u(t_n) = u_n, \quad (1.2b)$$

где $F_n = F(u_n)$, $J = F_u(u_n)$ — матрица Якоби F в точке u_n , $R_n(u) = F(u) - F_n - J(u - u_n)$ — нелинейная часть F .

Проинтегрируем (1.2) с помощью метода вариации постоянной, получим

$$u(t_{n+1}) = u(t_n) + (e^{hJ} - I)J^{-1}F_n + \int_{t_n}^{t_{n+1}} e^{J(t_{n+1}-t)} R_n(u(t)) dt, \quad (1.3)$$

где $h = t_{n+1} - t_n$, I — единичная матрица, e^{hJ} , $e^{J(t_{n+1}-t)}$ — матричные экспоненты соответствующих аргументов.

Полученное соотношение (1.3) является отправной точкой для построения экспоненциальных численных методов.

1.2. Расчётные формулы

В работе [6] предложена следующая конструкция трёхстадийного метода EPIRK: промежуточные векторы вычисляются по формулам

$$r_1 = y_n + a_{11}\varphi_{30} \left(\frac{h}{3} J \right) \frac{h}{3} F_n, \quad (1.4a)$$

$$r_2 = y_n + a_{21}\varphi_{30} \left(\frac{2h}{3} J \right) \frac{2h}{3} F_n + a_{22}\varphi_{31} \left(\frac{2h}{3} J \right) \frac{2h}{3} R_n(r_1), \quad (1.4b)$$

после чего строится решение на новом слое:

$$y_{n+1} = y_n + \varphi_{30}(hJ)hF_n + b_1\varphi_{31}(hJ)hR_n(r_1) + b_2\varphi_{32}(hJ)h(-2R_n(r_1) + R_n(r_2)). \quad (1.4c)$$

Функции $\varphi_{3j}(z)$, $j = \overline{0, 2}$ в (1.4) определены по формулам

$$\begin{aligned}\varphi_{30}(z) &= \frac{e^z - 1}{z}, \\ \varphi_{31}(z) &= 3 \frac{e^z - (1 + z)}{z^2}, \\ \varphi_{32}(z) &= \frac{3e^z(6 - z) - (6 + 5z + 2z^2)}{2z^3}.\end{aligned}$$

Перепишем (1.3), используя определение $\varphi_{30}(z)$:

$$u(t_{n+1}) = u(t_n) + \varphi_{30}(hJ)F_n + \int_{t_n}^{t_{n+1}} e^{J(t_n+h-t)} R_n(u(t)) dt. \quad (1.5)$$

Нетрудно заметить, что (1.4) является аппроксимацией (1.5) (подробный и обоснованный вывод см. в [6]).

При реализации метода (1.4) возникают следующие задачи:

- 1) Как вычислить значения функций $\varphi_{3j}(z)$, $j = \overline{0, 2}$?
- 2) Какие коэффициенты a_{ij} и b_j выбрать?
- 3) Как контролировать шаг и ошибку метода?

Рассмотрим каждый из вопросов по отдельности.

Матричные функции. Как видно из (1.4), для работы метода не требуется вычислять значения матричных функций $\varphi_{3j}(z)$ сами по себе. Необходимо вычислить только их действие на соответствующие векторы. Для решения этой задачи хорошо подходят методы крыловского пространства [7, Chapter 7]. Кратко опишем идею этого подхода.

Пусть имеются невырожденная матрица A и вектор b . Тогда возможно построить ортонормированный базис в пространстве Крылова

$$\mathcal{K}_m(A, b) = \text{Span}\{b, Ab, A^2b, \dots, A^{m-1}b\}.$$

Специальная версия алгоритма Грама—Шмидта, адаптированная к применению в пространстве $\mathcal{K}_m(A, b)$, называется алгоритмом Арнольди (см. алгоритм 1).

В результате работы алгоритма Арнольди получаем матрицы \overline{V}_m и \overline{H}_m . Матрица \overline{V}_m содержит базисные векторы $\{v_i\}_{i=\overline{1, m+1}}$ в качестве своих столбцов и имеет размеры $N \times (m + 1)$, где N — размерность вектора b . Матрица \overline{H}_m является верхней Хессенберговой, содержит коэффициенты h_{ij} разложения векторов $\{A^k b\}_{k=0}^{m-1}$ по базису $\{v_i\}_{i=1}^m$ и имеет размеры $(m + 1) \times m$.

Алгоритм 1: Арнольди

Исходные данные: Матрица A , вектор b , размерность m

Результат: Матрицы $\bar{V}_m = [v_1, \dots, v_m]$ и $\bar{H}_m = (h_{ij})_{i=\overline{1,m+1}, j=\overline{1,m}}$

$v_1 = b/\|b\|_2$;

for $j = \overline{1, m}$ **do**

$w_j = Av_j$

for $i = \overline{1, j}$ **do**

$h_{ij} = (w_j, v_i)$

$w_j = w_j - h_{ij}v_i$

end

if $h_{j+1,j} = 0$ **then**

stop

else

$v_{j+1} = w_j/h_{j+1,j}$

end

end

Если в алгоритме Арнольди произошёл останов (т.е. $h_{j+1,j} = 0$ при некотором j), то это значит, что m превышает размерность крыловского пространства, т.е. $m > \dim \mathcal{K}_m(A, b)$. Будем считать, что в таком случае значение m устанавливается в $\dim \mathcal{K}_m(A, b)$ по окончании работы алгоритма.

Вместе с \bar{V}_m и \bar{H}_m определяются матрицы V_m и H_m :

$$\begin{aligned} V_m &= \bar{V}_m(:, 1 : m), \\ H_m &= \bar{H}_m(1 : m, :), \end{aligned}$$

т.е. это первые m столбцов и m строк матриц \bar{V}_m и \bar{H}_m соответственно (используется нотация MATLAB). Тогда

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \quad (1.6)$$

где $e_m = (0, \dots, 0, 1) \in \mathbb{R}^m$. Разложение (1.6) позволяет вычислить приближение $f(A)b$, если f можно представить с помощью степенного ряда:

$$f(A)b \approx \|b\|_2 V_m f(H_m) e_1, \quad (1.7)$$

где $e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^m$. Т.е. значение функции вычисляется не в исходном пространстве, а в построенном пространстве Крылова. В предположении, что значение f в (1.7) вычисляется точно, можно оценить ошибку аппроксимации (1.7) [8, 9]:

$$\rho_m = \|b\|_2 h_{m+1,m} [f(H_m)]_{m,1} v_{m+1}, \quad (1.8)$$

где $[A]_{i,j}$ означает взятие элемента a_{ij} матрицы A .

Формулы (1.7)–(1.8) можно использовать для вычисления $\varphi_{3j}(\tau J)b$ в (1.4), где $\tau = h/3, 2h/3, h$. Для этого необходимо задать величину Tol , зафиксировать размерность m , построить по алгоритму 1 базис пространства Крылова $\mathcal{K}_m(\tau A, b)$ и вычислить приближение к $\varphi_{3j}(\tau J)b$ по формуле (1.7). Процесс повторяется до тех пор, пока не выполнится условие

$$\|\rho_m\|_2 < Tol. \quad (1.9)$$

Поскольку процесс построения пространства Крылова трудоёмок, в [11] предложено ограничиться пространствами Крылова размерностей

$$\text{Dims} = \{1, 2, 3, 4, 6, 8, 11, 15, 20, 27, 36, 48\}.$$

Соответственно, условие (1.9) проверяется только при $m \in \text{Dims}$. Если хотя бы для одной функции (1.4) это условие не выполнено даже при $m_{\max} = \max \text{Dims}$, то необходимо уменьшить h и пересчитать весь шаг заново.

Заметим, что в методе (1.4) не нужно строить каждый раз новое крыловское пространство при вычислении очередной функции $\varphi_{3j}(\tau A)$. Достаточно ограничиться построением трёх крыловских пространств. Действительно, если $H_m = V_m^T A V_m$, то $\sigma H_m = V_m^T (\sigma A) V_m$. Более того, если $d = \gamma b$, то в (1.7)

$$f(A)d \approx |\gamma| \|b\|_2 V_m f(H_m) e_1,$$

т.к. базис V_m останется тем же. Из этого следует, что

- 1) для вычисления значений

$$\varphi_{30} \left(\frac{h}{3} J \right) \frac{h}{3} F_n, \quad \varphi_{30} \left(\frac{2h}{3} J \right) \frac{2h}{3} F_n, \quad \varphi_{30} (hJ) h F_n$$

можно использовать одно и то же крыловское пространство;

- 2) если для $\varphi_{30}(hJ)hF_n$ выполнено условие (1.9), то для остальных функций $\varphi_{30}(\tau J)\tau F_n$ при $\tau < h$ оно выполнено автоматически, — см. (1.8).

Аналогичные рассуждения верны для пары $\varphi_{31}(2h/3J)(2h/3)R_n(r_1)$ и $\varphi_{31}(hJ)hR_n(r_1)$. Следовательно, каждая стадия метода (1.4) определяется не вычислением векторов r_1, r_2 и y_{n+1} , а вычислением соответствующих функций в пространстве Крылова. Это отражено в табл. 1: в каждом столбце собраны векторно-матричные произведения, которые можно вычислить в одном и том же пространстве Крылова.

Коэффициенты Коэффициенты метода (1.4) выбираются так, чтобы он имел третий или четвёртый порядок аппроксимации. В оригинальной работе [6] без вывода представлены условия порядков 3 и 4. Представим их:

- условие порядка 3

$$a_{11}^2 b_1 - a_{11}^2 b_2 + 2a_{21}^2 b_2 = 2; \quad (1.10a)$$

Таблица 1. Группировка векторно-матричных произведений по стадиям

Стадия 1	Стадия 2	Стадия 3
$\varphi_{30} \left(\frac{h}{3} J \right) \frac{h}{3} F_n$	$\varphi_{31} \left(\frac{2h}{3} J \right) \frac{2h}{3} R_n(r_1)$	$\varphi_{32} (hJ) h(-2R_n(r_1) + R_n(r_2))$
$\varphi_{30} \left(\frac{2h}{3} J \right) \frac{2h}{3} F_n$	$\varphi_{31} (hJ) hR_n(r_1)$	
$\varphi_{30} (hJ) hF_n$		

- условия порядка 4

$$4a_{11}^2 b_1 - 3a_{11}^2 b_2 + 10a_{21}^2 b_2 = 12, \quad (1.10b)$$

$$2a_{11}^3 b_1 - 2a_{11}^3 b_2 + 8a_{21}^3 b_2 = 9. \quad (1.10c)$$

Автору [6] удалось построить семейство решений уравнений (1.10). На их основе построены методы третьего и четвёртого порядков аппроксимации — их коэффициенты представлены в табл. 2.

Таблица 2. Коэффициенты методов семейства EPIRK

Имя метода	4A	4B	4C	4D	3A	3B
Порядок	4	4	4	4	3	3
a_{11}	9/4	11/16	27/28	27/76	9/4	11/16
a_{21}	9/8	55/64	27/28	27/38	9/8	55/64
a_{22}	0	0	0	0	0	0
b_1	160/243	-512/3993	1568/2187	-57760/6561	32/81	512/121
b_2	128/243	8192/3993	3136/2187	23104/6561	0	0

Из табл. 2 видно, что методы 3A и 3B являются вложенными для методов 4A и 4B соответственно. В рамках данной работы реализованы методы EPIRK4A и EPIRK3A.

Контроль шага. Как следует из формулировки алгоритма EPIRK, шаг h отвечает не только за точность формул (1.4), но и за качество крыловской аппроксимации функций $\varphi_{3j}(z)$. Поэтому необходим алгоритм контроля шага, учитывающий эти особенности.

Точность метода. Если предположить, что все функции в (1.4) вычисляются точно, то шаг будет влиять только на локальную ошибку метода. Поэтому для контроля шага можно использовать стандартную стратегию [10,

§II.4]. Вычисляем 2-норму масштабированного вектора ошибки:

$$err = \sqrt{\frac{1}{d} \sum_{i=1}^d \left(\frac{E_n^i}{Atol_i + |y_n^i| \cdot Rtol_i} \right)^2}, \quad (1.11)$$

где $E_n = (E_n^1, \dots, E_n^d)$ — вектор ошибки размерности d на временном слое t_n , $\{Atol_i\}_{i=1}^d$ — массив допустимых абсолютных погрешностей по каждой координате, $\{Rtol_i\}_{i=1}^d$ — массив допустимых относительных погрешностей, $y_n = (y_n^1, \dots, y_n^d)$ — вектор численного решения на слое t_n . Вектор E_n локальной ошибки можно получить с помощью вложенного метода или с помощью правила Рунге.

С помощью err определяем шаг на новом слое

$$h_{new}^{step} = h_{old} \cdot \min\{facmax, \max\{facmin, fac \cdot (1/err)^{1/(p+1)}\}\}, \quad (1.12)$$

где $facmin$, $facmax$, fac — минимальный, максимальный и гарантийный множители изменения шага, p — порядок, с которым получен вектор ошибки E_n .

Погрешность крыловской аппроксимации. Так как для вычисления значений векторных функций в (1.4) используются методы крыловского пространства, шаг h также влияет на погрешность крыловской аппроксимации (1.7): чем больше h , тем бóльшая размерность пространства $\mathcal{K}_m(\sigma h A, b)$ требуется для достижения заданной точности, и наоборот.

Поэтому с помощью шага h можно управлять качеством крыловской аппроксимации. В работе [11] предложена следующая стратегия вычисления шага на основе размерности крыловского пространства:

$$h_{new}^{kry} = h_{old} \cdot \left(\frac{m_{opt}}{m} \right)^\alpha, \quad (1.13)$$

где m — текущая размерность пространства Крылова, m_{opt} — оптимальная размерность пространства Крылова (выбирается перед расчётом), $\alpha = 1/3$ — показатель, определённый экспериментально. Задача критерия (1.13) — поддерживать размерность пространства Крылова на уровне оптимальной.

Выше сказано, что в процессе вычисления (1.7) размерность m увеличивается автоматически, пока условие (1.9) не будет выполнено. Если же $\|\rho_m\| < Tol$, то можно попытаться уменьшить размерность пространства Крылова. Далее представлено описание такого алгоритма.

Пусть $est = \|\rho_m\|_2 / Tol$, $g(z) = z^\alpha$, где α — показатель степени из (1.13). Определим

$$\hat{m}_{new} = \lceil m_{max} \cdot g(est) \rceil,$$

где \hat{m}_{new} — промежуточная размерность, $m_{max} = \max \text{Dims}$ — максимальная размерность пространства Крылова, $\lceil \cdot \rceil$ — функция «потолок», определяемая как

$$\lceil x \rceil = \min\{m \in \mathbb{Z} : x \leq m\}.$$

В качестве новой размерности пространства Крылова выберем

$$m_{new} = \min\{m \in \text{Dims} : \hat{m}_{new} \leq m\}. \quad (1.14)$$

Идея введённого критерия в следующем: функция $g(est)$ оценивает степень малости локальной погрешности (достаточно ли он мал, чтобы уменьшить размерность). Оценка \hat{m}_{new} даёт новую размерность пространства Крылова без привязки к массиву Dims. Условно говоря, это новая размерность пространства Крылова, если бы мы могли менять размерность пространства Крылова как угодно. Формула (1.14) для m_{new} привязывает \hat{m}_{new} к массиву Dims допустимых размерностей пространства Крылова.

Формулы (1.13), (1.14) совместно управляют шагом и размерностью пространства Крылова. Они применяются только в том случае, если шаг метода (1.4) досчитан до конца, т.е. на каждой стадии выполнено условие (1.9). Если хотя бы на одной стадии это условие не выполнено, нужно уменьшить h и провести расчёт всего шага заново. Для уменьшения шага можно использовать формулу (1.12) с $p = 2$ (т.е. $\alpha = 1/(p + 1)$) и $estim$ вместо err :

$$h_{new}^{lost} = h_{old} \cdot \min\{facmax, \max\{facmin, fac \cdot (1/estim)^\alpha\}\}. \quad (1.15)$$

Таким образом, итоговый шаг вычисляется по формуле

$$h_{new} = \min\{h_{new}^{step}, h_{new}^{kry}\} \quad (1.16)$$

— если на каждой стадии выполнено условие (1.9), и по формуле (1.15) — если условие не выполнено хотя бы на одной стадии.

Замечание 1.1. В работе [6] отмечено, что при $i < j$ для вычисления $\varphi_{3j}(\tau A)b$ может потребоваться меньшая размерность пространства Крылова, чем при вычислении $\varphi_{3i}(\tau A)b$. Поэтому введём счётчик размерности m_j , $j = \overline{1, 3}$ для каждой стадии.

В таком случае h_{new}^{kry} вычисляется по формуле

$$h_{new}^{kry} = h_{old} \cdot \min_{j=1,3} \left(\frac{m_{opt}}{m_j} \right)^\alpha.$$

Погрешность вычисления функции. В приближении (1.7) считается, что значение $f(H_m)$ вычисляется точно. Если для вычисления f используется аппроксимация, то размер шага h может влиять на её качество. Например, если для f существует степенное разложение, то h надо выбрать таким, чтобы H_m попала в круг сходимости разложения. В рамках данной работы это важно при вычислении $f(z) = e^z$.

Например, в MATLAB для вычисления $e^{\tau A}$ используется диагональная аппроксимация Паде $R_{q,q}(z)$ в комбинации с известным алгоритмом масштабирования [12]: перед вычислением $R_{q,q}(z)$ выбирается p , такой, что

$$\|\tau A/2^p\| < 1,$$

после чего

$$[R_{q,q}(\tau A/2^p)]^{2^p}$$

используется в качестве аппроксимации для матричной экспоненты. В случае больших p это может привести к нарастанию погрешности из-за округлений при матричных умножениях.

Тогда размер шага h (а через него — и τ) можно ограничить косвенно, задав p_{\max} , такой, что если выполнено $\|\tau A/2^{p_{\max}}\| < 1$, то выполняется алгоритм масштабирования, если не выполнено — уменьшается шаг h .

Все указанные замечания учтены при реализации методов на языке среды MATLAB.

2. Тестирование метода

В оригинальной работе [6] проведён полный анализ сходимости метода: определены порядки сходимости, экспериментально подтверждена сходимость крыловских аппроксимаций. Поэтому в данном разделе мы ограничимся экспериментальным определением порядка сходимости реализованных методов.

2.1. Постановка задачи

В силу своей формулировки метод (1.4) даёт точное решение линейных задач. Поэтому для экспериментального определения порядка сходимости рассмотрим следующую задачу из коллекции [13]:

$$u_1' = \alpha u_1^2 u_2, \quad (2.1a)$$

$$u_2' = -\alpha u_1 u_2^2, \quad (2.1b)$$

с начальными условиями $u_1(0) = u_2(0) = 1$. Параметр α определяет жёсткость задачи. Решение (2.1) выражается в элементарных функциях и является аналитической функцией:

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

Наша цель — экспериментальное определение порядка сходимости метода. Для целей тестирования проведём расчёт задачи (2.1) в отрезке $[0, 1]$ на последовательности сгущающихся равномерных сеток G_1, G_2, G_3 с шагами h, qh, q^2h , $q < 1$. Получим решения $y_h^{(1)}$, $y_{qh}^{(2)}$ и $y_{q^2h}^{(3)}$ соответственно (далее нижние индексы опущены).

Чтобы оценить порядок сходимости в каждой точке сетки G_1 , спроектируем $y^{(2)}$ и $y^{(3)}$ на сетку G_1 и построим в каждой точке сетки G_1 оценку [2,

Раздел 4.7]:

$$p = \frac{1}{\ln q} \ln \left| \frac{y^{(3)} - y^{(2)}}{y^{(2)} - y^{(1)}} \right|. \quad (2.2)$$

Построенная сеточная функция оценивает порядок сходимости метода в каждом узле сетки G_1 .

Замечание 2.1. Условия порядка методов EPIRK (1.10) получены в предположении, что матричные функции в (1.4) вычисляются точно. Поэтому при оценке порядка необходимо вычислять эти функции максимально точно.

Т.е. необходимо вычислять значения матричных функций «по определению», без использования методов крыловского пространства, либо установить размерность m всех пространств Крылова равной размерности d системы.

В тестовой задаче (2.1) $d = 2$, поэтому вычисление матричных функций не представляет труда.

2.2. Метод EPIRK3A

Положим $\alpha = 1$ в задаче (2.1) и выполним расчёт с помощью метода EPIRK3A (см. табл. 2) при $h = 0.01$. Затем решим задачу на более мелких сетках с параметром сгущения $q = 0.1$ и построим интерполированный график функции-оценщика (2.2). На рис. 1 представлены графики наблюдаемого порядка p для каждой из компонент задачи (2.1) в зависимости от времени.

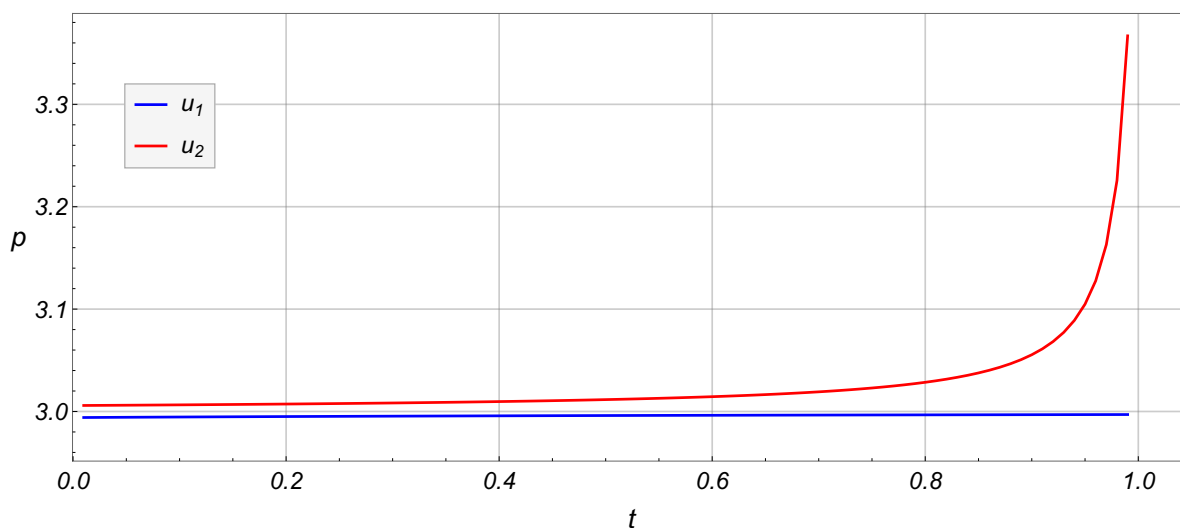


Рис. 1. Поточечная оценка порядка сходимости метода EPIRK3A

Из рис. 1 видно, что экспериментально наблюдаемый порядок сходимости EPIRK3A близок к теоретическому. Резкое возрастание порядка сходимости для u_2 в конце отрезка объясняется тем, что u_2 экспоненциально убывает и оценщик становится чувствительным к ошибкам округления.

2.3. Метод EPIRK4A

С помощью метода EPIRK4A решим задачу (2.1) при тех же параметрах и построим оценки сходимости (2.2). Получим графики, аналогичные графикам рис. 1 — см. рис. 2.

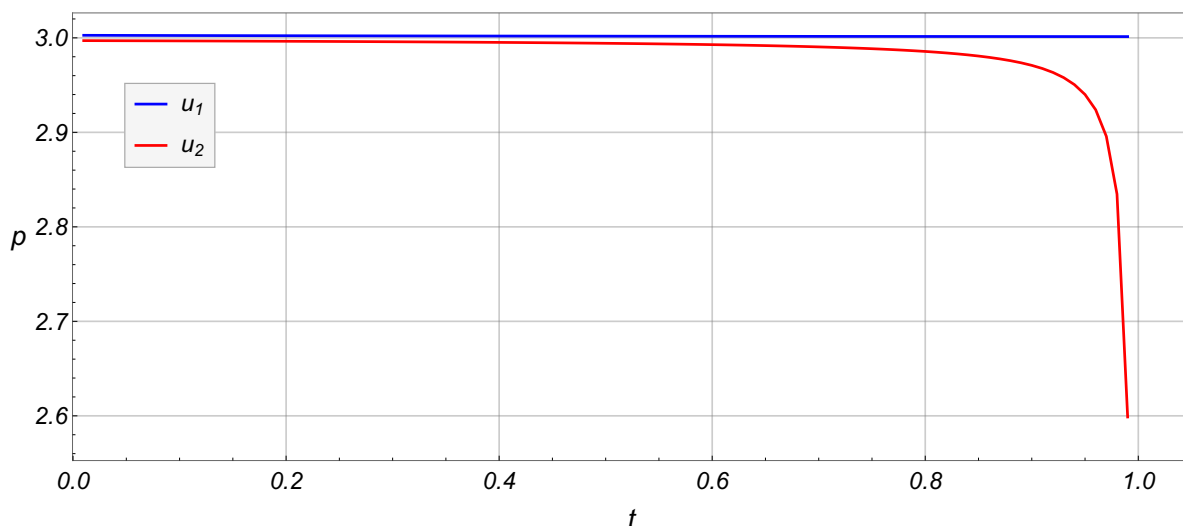


Рис. 2. Поточечная оценка порядка сходимости метода EPIRK4A

Из рис. 2 видно, что наблюдаемый порядок сходимости метода EPIRK4A отличается от теоретического. У этого явления может быть несколько причин:

- **Недостаточная гладкость решения.** Если порядок гладкости решения q меньше порядка аппроксимации p метода, то порядок сходимости может отличаться от p .

Однако выше отмечено, что (2.1) имеет бесконечно дифференцируемое аналитическое решение.

- **Неточное вычисление функций $\varphi_{3j}(z)$.** Согласно замечанию 2.1, качество вычисления матричных функций может влиять на качество сходимости метода.

В данной работе методы EPIRK реализованы на языке MATLAB, матричная экспонента вычисляется с помощью `expm`, остальные функции $\varphi_{3j}(z)$ — по определению. Задача (2.1) имеет малую размерность и при $\alpha = 1$ жёсткой не является. Поэтому допустимо считать, что матричные функции вычисляются практически точно.

- **Некорректные условия порядка.** Если условия (1.10) отличаются от истинных условий порядка, то метод может иметь, вообще говоря, любой порядок сходимости.

Разрешению этого вопроса посвящён следующий раздел.

3. Коррекция методов

В данном разделе представлены обновлённые условия порядка метода (1.4) и найдены их решения. Найденные решения определяют коэффициенты a_{ij} и b_i , новых методов семейства EPIRK. Показано, что теоретический порядок сходимости новых методов совпадает с экспериментальным.

3.1. Условия порядка

В результате сравнения производных точного решения (1.5) и производных численного решения (1.4) вплоть до порядка 4 автором обнаружено, что

- метод (1.4) имеет порядок аппроксимации $p = 2$ при произвольных коэффициентах a_{ij} и b_i ;
- условие третьего порядка аппроксимации метода (1.4) имеет вид

$$(b_1 - b_2)a_{11}^2 + 2b_2a_{21}^2 = 2, \quad (3.1a)$$

т.е. совпадает с (1.10a);

- условия четвёртого порядка аппроксимации метода (1.4) имеют вид

$$2b_1a_{11}^2 - b_2a_{11}^2 + 2b_2a_{21}^2 = 3, \quad (3.1b)$$

$$2(b_1 - b_2)a_{11}^3 + 8b_2a_{21}^3 = 9, \quad (3.1c)$$

$$2(b_1 - b_2)a_{11}^2 + 8b_2a_{21}^2 = 9. \quad (3.1d)$$

Нетрудно установить, что условие (3.1c) совпадает с условием (1.10c), а сумма условий (3.1b) и (3.1d) совпадает с (1.10b). Т.е. в работе [6] представлены ослабленные условия порядка 4.

Докажем экспериментально, что условия (3.1) задают метод четвёртого порядка аппроксимации. Для этого построим метод, коэффициенты которого являются решением (3.1).

Т.к. a_{22} не участвует в (3.1), положим $a_{22} = 0$. Затем вычтем (3.1a) из (3.1b) и (3.1d), получим

$$(3.1b) - (3.1a) : \quad b_1a_{11}^2 = 1, \quad (3.1b')$$

$$(3.1d) - 2 \cdot (3.1a) : \quad b_2a_{21}^2 = \frac{5}{4}. \quad (3.1d')$$

Затем подставим (3.1b') и (3.1d') в (3.1a) и (3.1c), получим:

$$2 \cdot 1 - b_2a_{11}^2 + 2 \cdot \frac{5}{4} = 3 \implies b_2a_{11}^2 = \frac{3}{2}, \quad (3.1a')$$

$$2 \cdot \left(1 - \frac{3}{2}\right) a_{11} + 8 \cdot \frac{5}{4} a_{21} = 9 \implies -a_{11} + 10a_{21} = 9. \quad (3.1c')$$

Из (3.1d') и (3.1a') выведем

$$\frac{5}{4a_{21}^2} = b_2 = \frac{3}{2a_{11}^2} \implies |a_{21}| = \sqrt{\frac{5}{6}} |a_{11}|. \quad (3.2)$$

Уравнение (3.2) совместно с (3.1c') однозначно определяет коэффициенты a_{11} и a_{21} . Рассмотрим два варианта раскрытия модуля.

Вариант «+». Раскроем в (3.2) модуль со знаком «+» и решим совместно с (3.1c'), получим:

$$\begin{cases} -a_{11} + 10a_{21} = 9, \\ \sqrt{5/6} a_{11} - a_{21} = 0. \end{cases} \iff \begin{cases} a_{11} = \frac{10}{9\sqrt{5/6} - 1}, \\ a_{21} = \sqrt{5/6} a_{11}. \end{cases} \quad (3.3a)$$

Из (3.1b') и (3.1a') находим b_1, b_2 :

$$b_1 = \frac{1}{a_{11}^2}, \quad b_2 = \frac{3}{2} b_1. \quad (3.3b)$$

Вариант «-». Раскроем в (3.2) модуль со знаком «-» и решим совместно с (3.1c'), получим

$$\begin{cases} -a_{11} + 10a_{21} = 9, \\ \sqrt{5/6} a_{11} + a_{21} = 0. \end{cases} \iff \begin{cases} a_{11} = -\frac{9}{10\sqrt{5/6} + 1}, \\ a_{21} = -\sqrt{5/6} a_{11}. \end{cases}$$

Коэффициенты b_1 и b_2 находятся по формуле (3.3b). С точки зрения программной реализации предпочтительнее вариант «+», т.к. $a_{11} > 0, a_{21} > 0$.

Замечание 3.1. Здесь и далее метод, соответствующий варианту «+», будем называть методом EPIRK4. Этот метод реализован автором на языке среды MATLAB.

3.2. Тестирование метода

Определим экспериментально наблюдаемый порядок сходимости метода EPIRK4: решим тестовую задачу (2.1) на отрезке $[0, 1]$ с параметром $\alpha = 1$ и шагами h, qh, q^2h , где $h = 0.01, q = 0.1$, и построим оценку (2.2). На рис. 3 представлен интерполированный график оценки: видно, что экспериментальный порядок практически совпадает с теоретическим.

Следовательно, построенный метод EPIRK4 обладает четвёртым порядком сходимости, и условия (3.1b)–(3.1d) действительно являются условиями порядка 4.

Замечание 3.2. Видно, что ни один метод семейства EPIRK из табл. 2 не удовлетворяет полному набору условий (3.1b)–(3.1d), а следовательно, не может обладать четвёртым порядком сходимости. Это и объясняет результаты рис. 2.

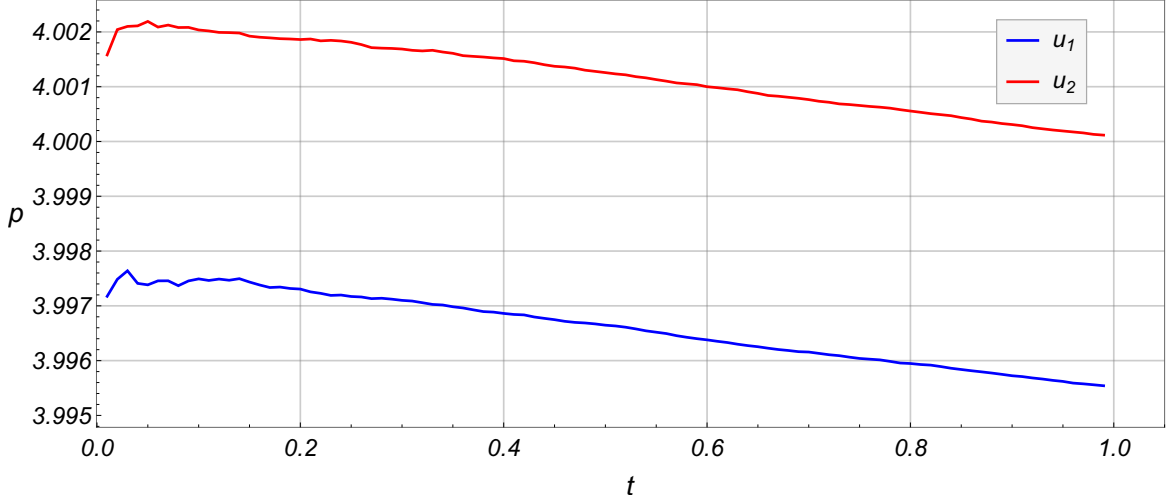


Рис. 3. Поточечная оценка порядка сходимости метода EPIRK4

3.3. Вложенный метод

Для оценки локальной ошибки метода EPIRK4 построим вложенный метод порядка 3. Для этого зафиксируем в условии (3.1a) значения коэффициентов a_{11} и a_{21} из (3.3a), получим линейное уравнение на b_1 и b_2 .

Чтобы однозначно определить коэффициенты b_i , дополним (3.1a) одним из условий порядка 5:

$$(b_1 - b_2)a_{11}^4 + 8b_2a_{21}^4 = \frac{54}{5}. \quad (3.4)$$

Совместное решение (3.1a) и (3.4) даёт

$$b_1 = \frac{5a_{11}^4 - 27a_{11}^2 + 54a_{21}^2 - 40a_{21}^4}{5a_{11}^2a_{21}^2(a_{11}^2 - 4a_{21}^2)}, \quad b_2 = \frac{5a_{11}^2 - 27}{5a_{21}^2(a_{11}^2 - 4a_{21}^2)}, \quad (3.5)$$

где значения a_{11} и a_{21} берутся из (3.3a). Построенный метод с коэффициентами (3.5) будем называть методом EPIRK3.

На рис. 4 представлена поточечная оценка порядка сходимости метода EPIRK3 для каждой компоненты решения задачи (2.1) (используется тот же набор параметров, что и при анализе EPIRK4).

Тот факт, что метод EPIRK4 используется с вложенным оценщиком EPIRK3, отразим в названии EPIRK4(3).

4. Моделирование горения УВТ

В работе [6] проведено исчерпывающее исследование сходимости и точности методов семейства EPIRK на некоторых модельных задачах. В текущей работе исследуем производительность и эффективность метода EPIRK4(3) на примере одной конкретной задачи.

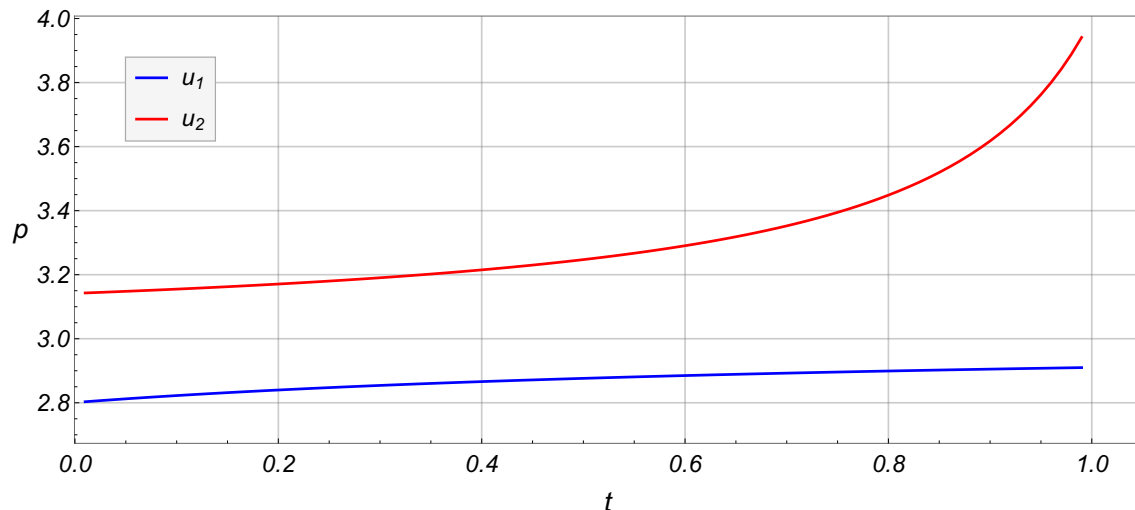


Рис. 4. Поточечная оценка порядка сходимости метода EPIRK3

В работе [3] описана система ОДУ — задача (3.1)–(3.3), — моделирующая процесс горения углеводородных топлив в ламинарном диффузионном пламени. С её помощью оценим производительности метода EPIRK4(3). Проведём серию расчётов задачи с различными параметрами точности и посмотрим, как они влияют на результаты работы метода EPIRK4(3). Для сравнения сопоставим результаты с результатами метода ROS4(3)L.

4.1. Качественные результаты

Проверим результаты, полученные методом EPIRK4(3), на качественное соответствие результатам, полученным в [3] ранее. На рис. 5–8 представлены первые четыре компоненты решения, полученного методом ROS4(3)L. Задача решалась для случая изооктана на отрезке $[0, 0.1]$ с параметрами расчёта

$$\text{AbsTol} = \text{RelTol} = 10^{-12},$$

определяющими точность каждой из компонент, т.е. в (1.11) задано

$$\text{A}tol_i = \text{AbsTol}, \quad \text{R}tol_i = \text{RelTol}, \quad i = \overline{1, d}.$$

На рис. 9 представлен график изменения шага интегрирования в зависимости от времени t .

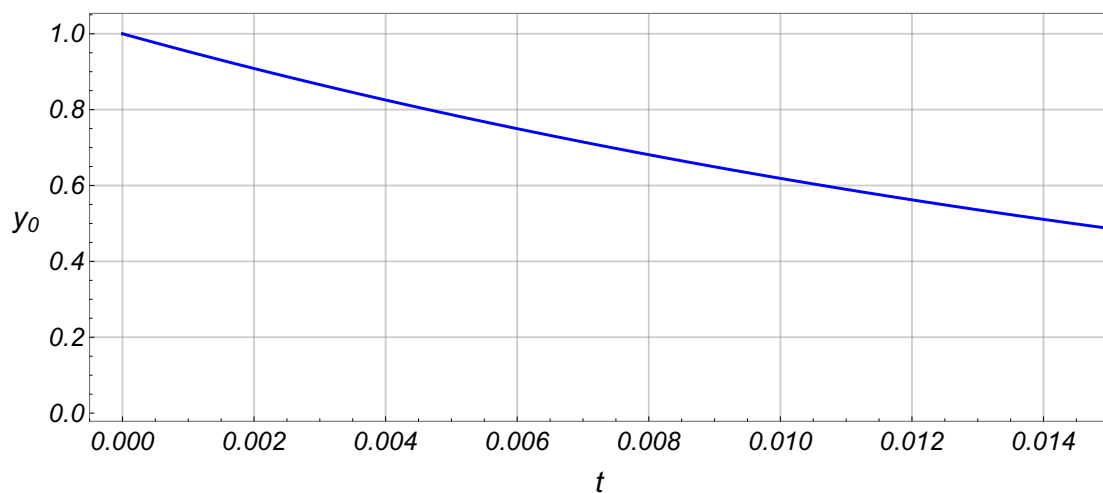


Рис. 5. Компонента решения y_0 , полученная методом ROS4(3)L

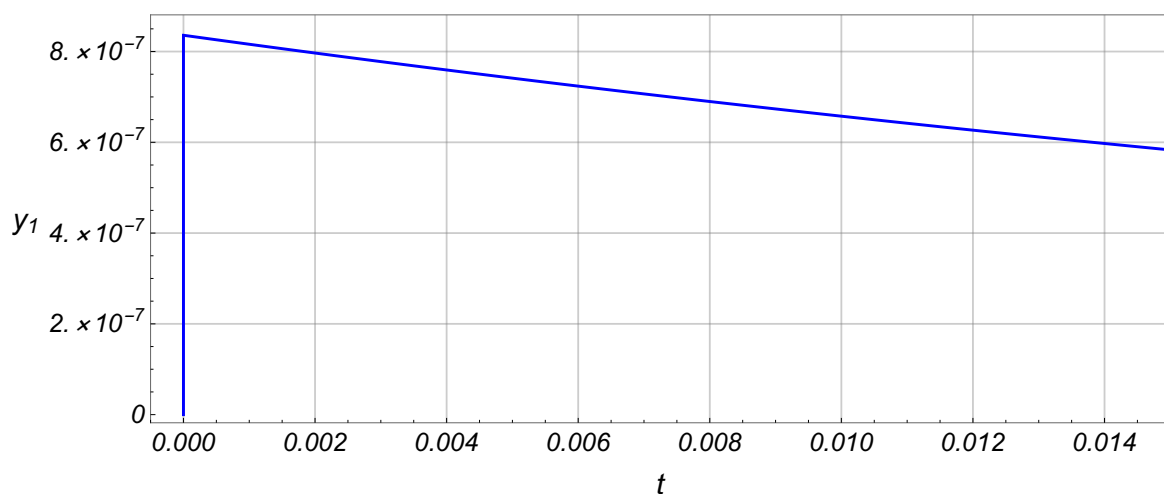


Рис. 6. Компонента решения y_1 , полученная методом ROS4(3)L

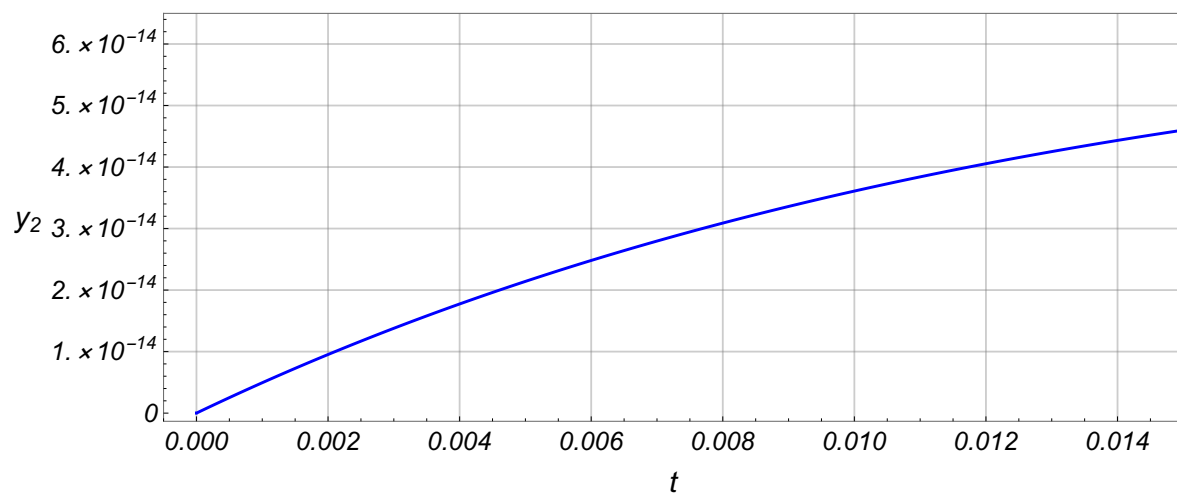


Рис. 7. Компонента решения y_2 , полученная методом ROS4(3)L

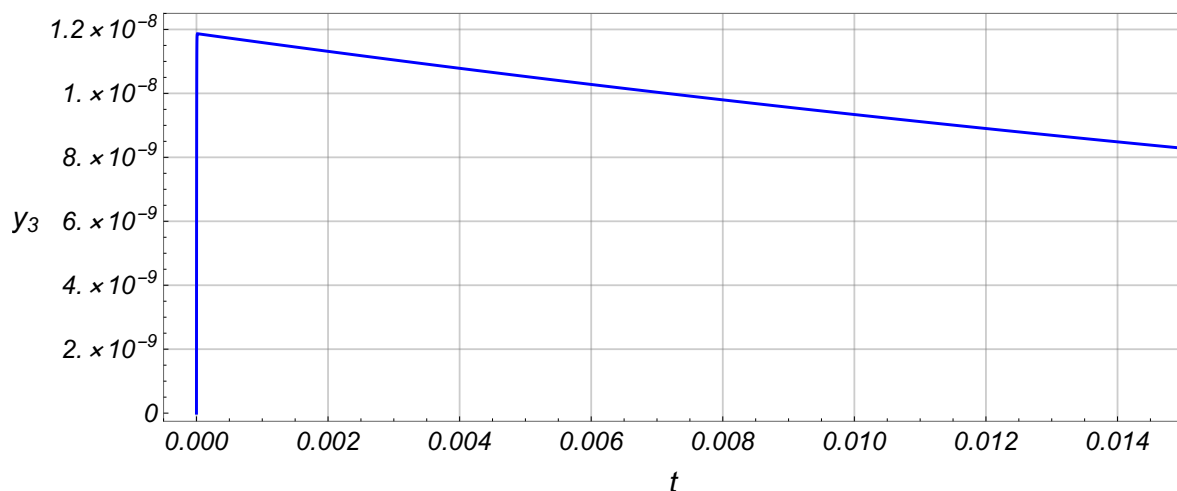


Рис. 8. Компонента решения y_3 , полученная методом ROS4(3)L

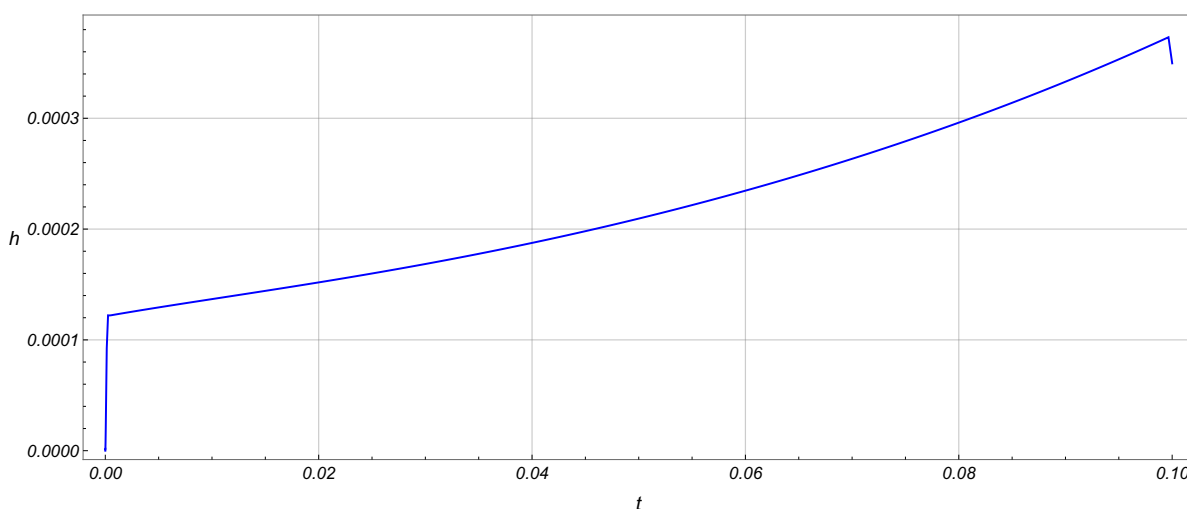


Рис. 9. Размер шага h , выбираемого методом ROS4(3)L, в зависимости от времени t

Сравним полученные результаты с результатами работы метода EPIRK4(3). Выберем те же значения параметров расчёта **Abstol** и **RelTol** и дополним их условиями $Tol = 10^{-12}$ в (1.9) и $m_{opt} = 8$ в (1.13). Результаты представлены на рис. 10–13: видно, что качественно картина совпадает с графиками рис. 5–8.

Рис. 14 содержит график изменения шага h методом EPIRK4(3). Для удобства на том же рисунке представлен график изменения шага для метода ROS4(3)L: видно, что метод EPIRK4(3) выбирает более мелкие шаги интегрирования.

На графике рис. 14 видна серия пиков, иллюстрирующих резкое изменение шага. Причина в том, что плавное изменение шага обеспечивается только формулой (1.12). В методе EPIRK4(3) управление шагом также осуществляется с помощью формул (1.15) и (1.13), связанных с крыловской аппроксимацией. Поэтому шаг h может меняться не так плавно, как в случае ROS4(3)L.

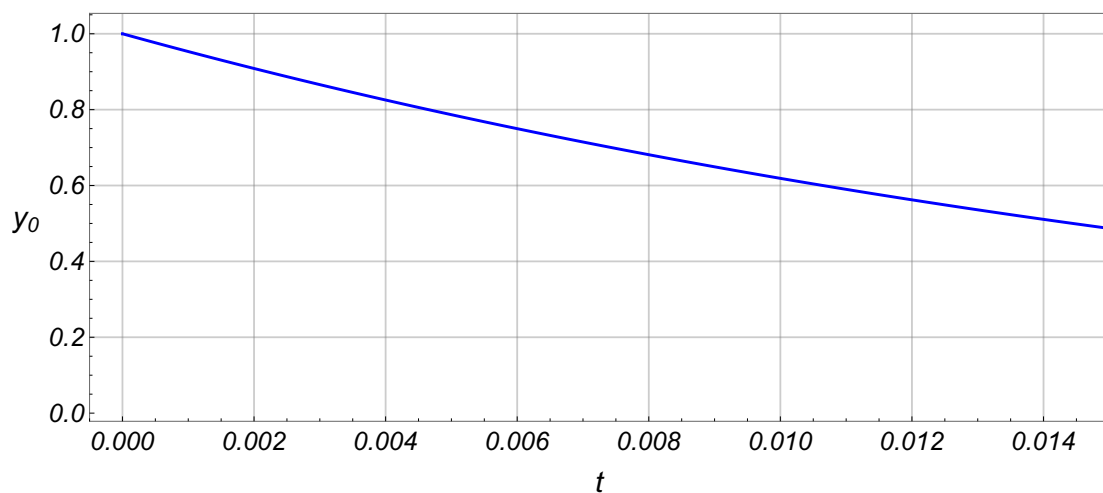


Рис. 10. Компонента решения y_0 , полученная методом EPIRK4(3)

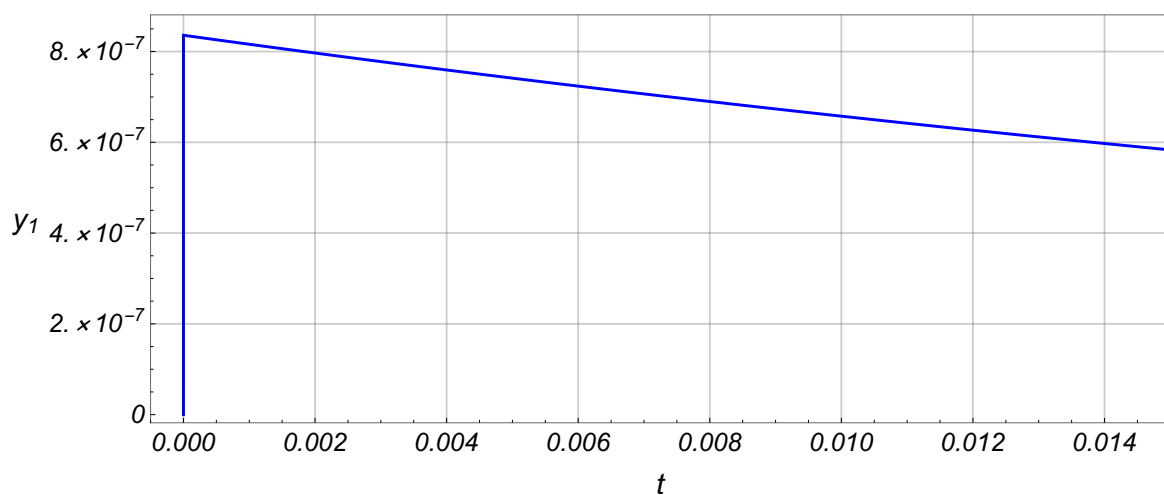


Рис. 11. Компонента решения y_1 , полученная методом EPIRK4(3)

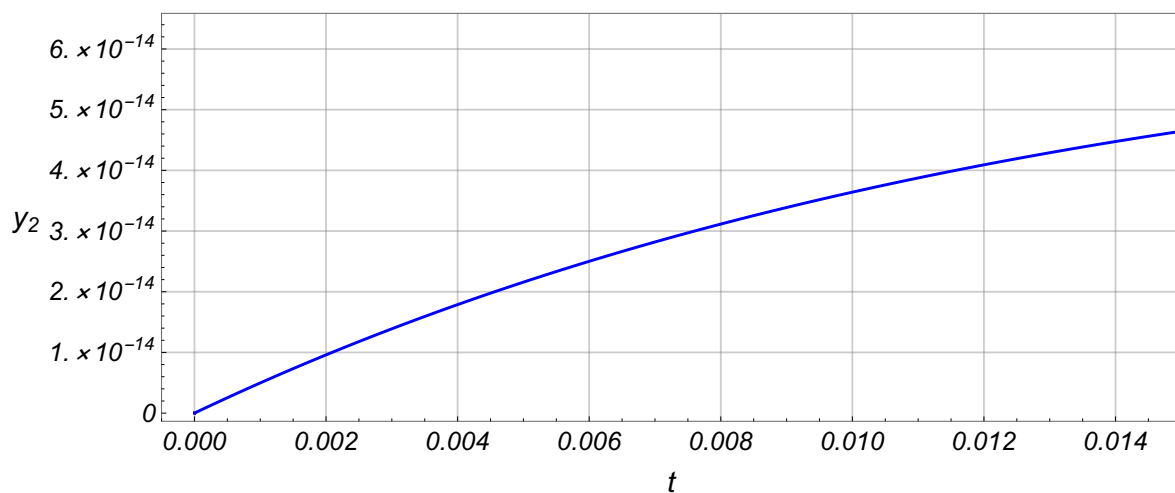


Рис. 12. Компонента решения y_2 , полученная методом EPIRK4(3)

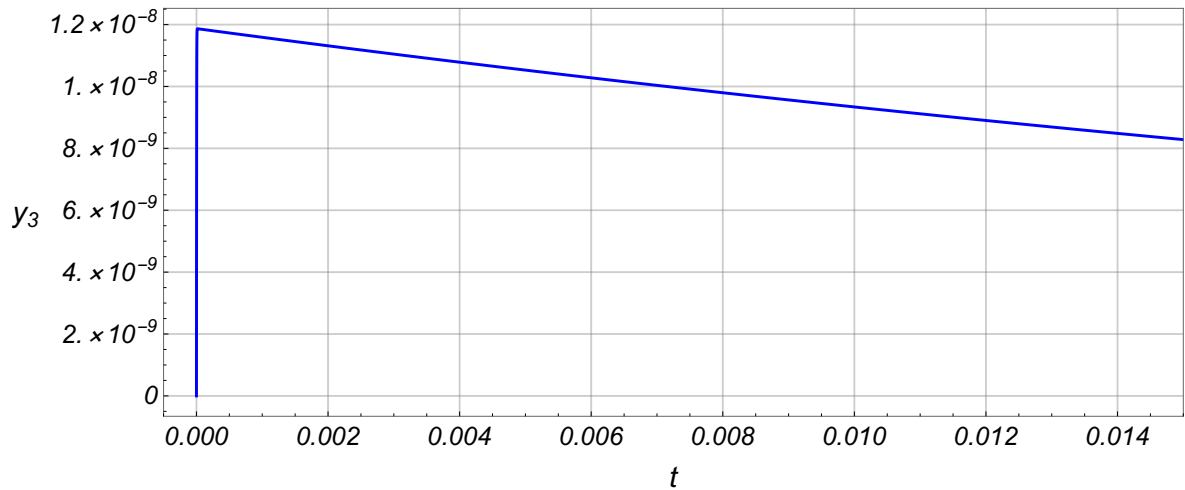


Рис. 13. Компонента решения y_3 , полученная методом EPIRK4(3)

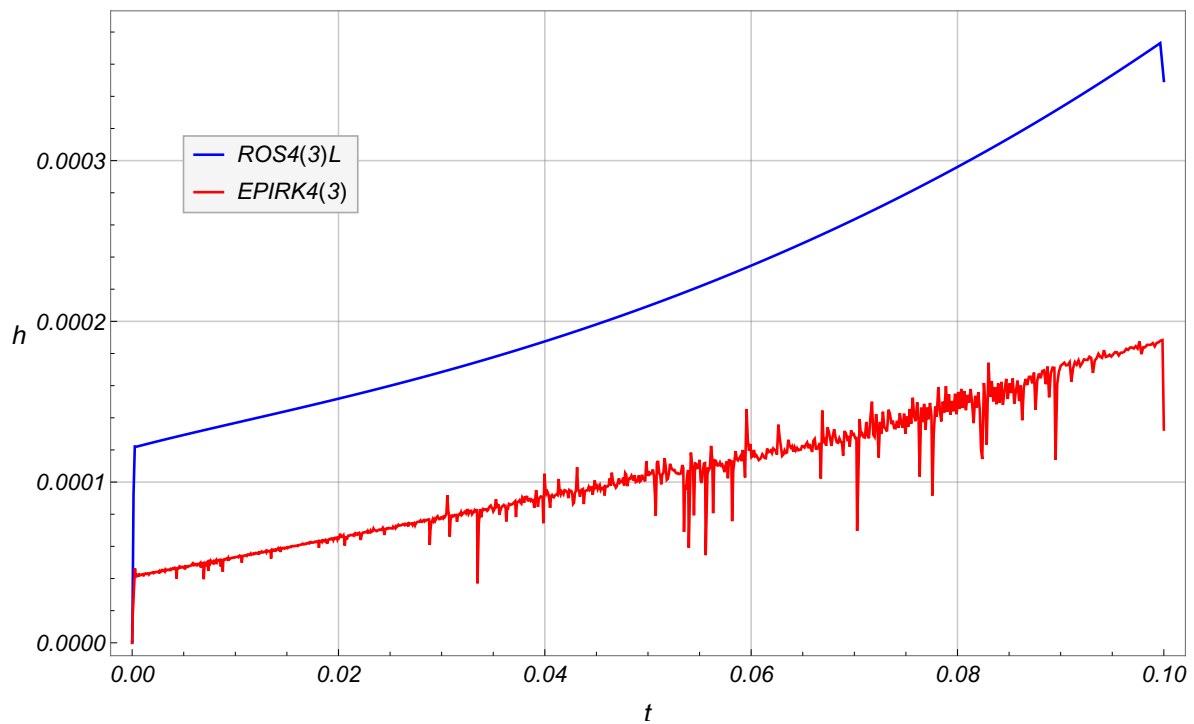


Рис. 14. Размер шага h , выбираемого методом EPIRK4(3), в зависимости от времени t

4.2. Время расчёта

Ранее показано, что EPIRK4(3) передаёт качественную структуру решения. Теперь определим трудоёмкость метода.

Количество шагов. Сначала сравним количества шагов, за которые EPIRK4(3) и ROS4(3)L решают одну и ту же задачу, и как на их количество влияют параметры расчёта.

Решать будем ту же задачу, что и выше. В качестве параметров точности

ROS4(3)L выберем

$$\text{AbsTol} = \text{RelTol} = \varepsilon_1, \quad \varepsilon_1 = 10^{-14} \dots 10^{-6}.$$

Для метода EPIRK4(3) дополнительно зададим максимальную погрешность крыловской аппроксимации — параметр Tol в (1.9) — и оптимальную размерность крыловского пространства: $Tol = 10^{-12}$, $m_{\text{opt}} = 8$. Затем проведём дополнительный расчёт с $m_{\text{opt}} = 4$.

На рис. 15 представлены зависимости числа шагов методов EPIRK4(3) и ROS4(3)L от значения $-\lg \varepsilon_1$ (используется логарифмический масштаб).

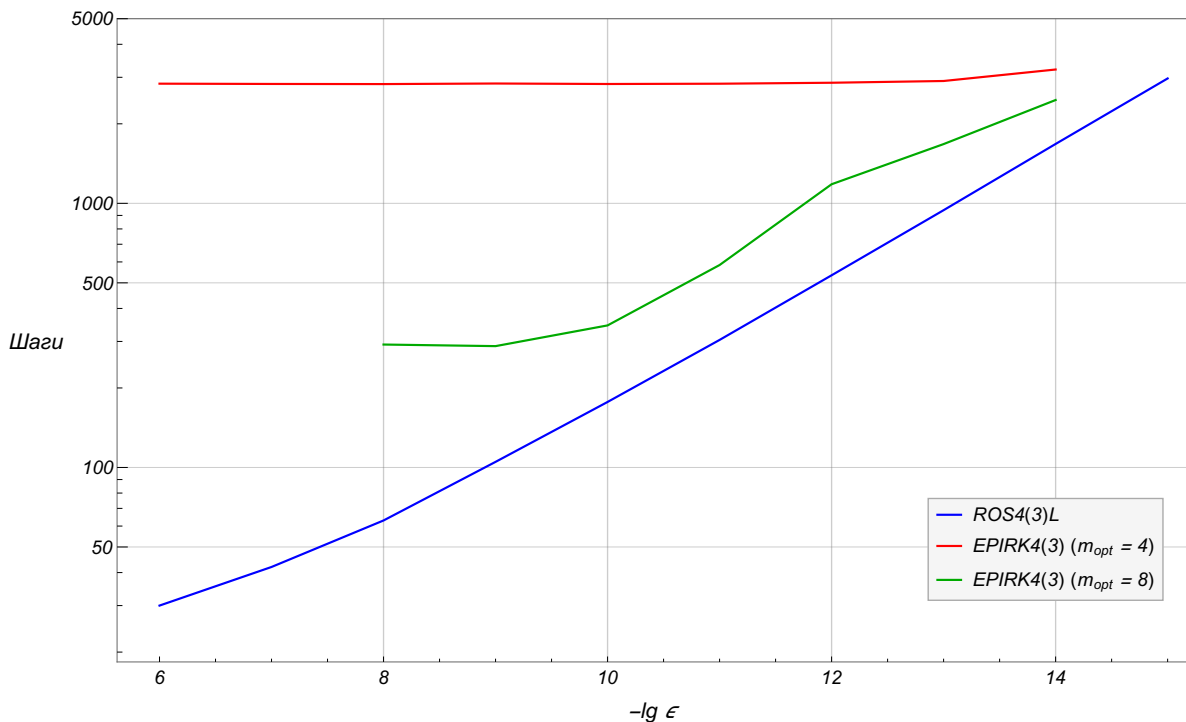


Рис. 15. Количества шагов методов ROS4(3)L и EPIRK4(3) в зависимости от $-\lg \varepsilon_1$

По рис. 15 видно, что метод EPIRK4(3) решает задачу за большее число шагов, чем ROS4(3)L. Более того,

- количество шагов метода EPIRK4(3) при $m_{\text{opt}} = 4$ практически никак не меняется при изменении ε_1 ;
- количество шагов метода EPIRK4(3) при $m_{\text{opt}} = 8$ растёт с уменьшением ε_1 и примерно вдвое больше количества шагов ROS4(3)L.

Объясним это. И шаг h , и размерность m влияют на ошибку (1.8) крыловской аппроксимации. Чем меньше h и чем больше m , тем меньше ошибка. В то же время параметр m_{opt} участвует в работе в критерия (1.13). Задача критерия — поддерживать размерность пространства Крылова m на уровне оптимальной m_{opt} за счёт управления шагом h .

Таким образом, при расчёте EPIRK4(3) поддерживает m на одном уровне. Поэтому влиять на ошибку (1.13) он может только за счёт шага h . Следовательно, при $m_{\text{opt}} = 4$ требуется шаг меньше, чем при $m_{\text{opt}} = 8$, — именно это и наблюдается на рис. 15.

Но почему кривая для случая $m_{\text{opt}} = 4$ не зависит от ε_1 , а кривая для случая $m_{\text{opt}} = 8$ — зависит? Выше отмечено, что при расчёте методом EPIRK4(3) в (1.9) установлен $Tol = 10^{-12}$. Поэтому алгоритм пытается выполнить условие (1.9) на каждом шаге.

Как следует из предыдущего текста, EPIRK4(3) может выполнить условие (1.9) только уменьшив шаг. Судя по всему, в случае $m_{\text{opt}} = 4$ методу требуется выбирать достаточно мелкий шаг, чтобы выполнить это условие, даже если другие условия выполнены. А в случае $m_{\text{opt}} = 8$ размерность m оказывается достаточно большой, так что условие (1.9) можно выполнить в более широком диапазоне шагов h . Поэтому выбираются более крупные шаги.

На рис. 16 представлен обновлённый сравнительный график, когда параметр $Tol = \varepsilon_1$: в этом случае количество шагов метода EPIRK4(3) уменьшается, но EPIRK4(3) всё равно требуется больше шагов, чтобы решить задачу.

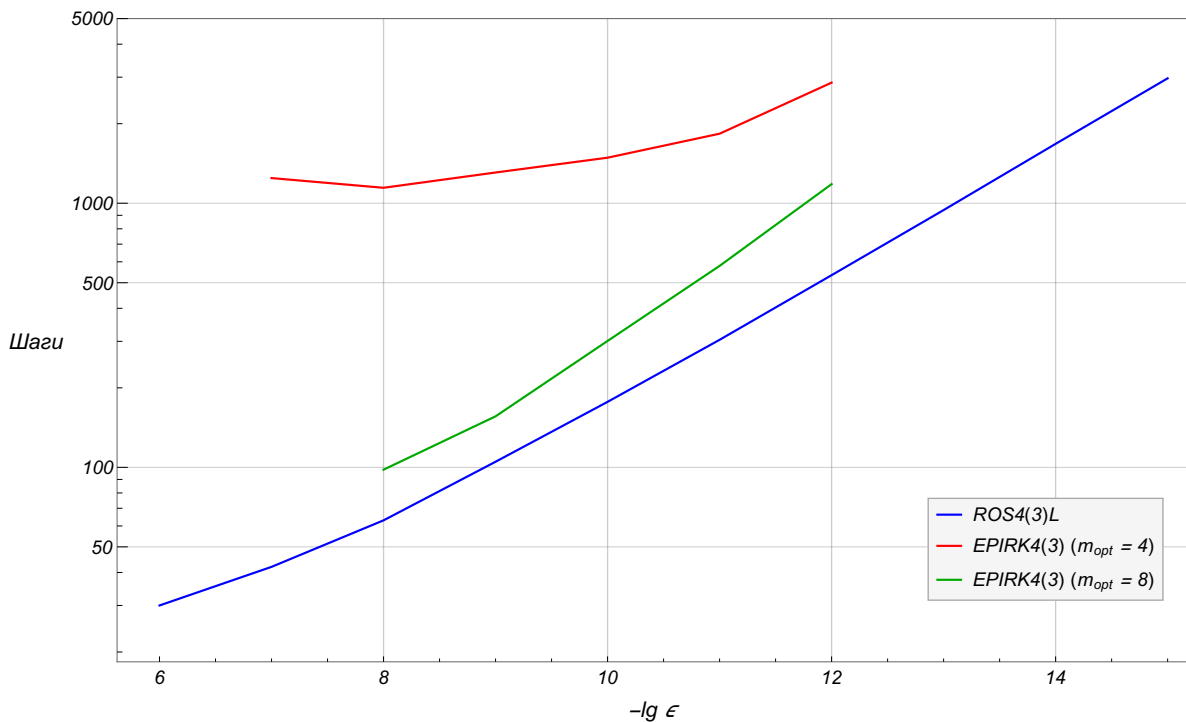


Рис. 16. Количество шагов методов ROS4(3)L и EPIRK4(3) в зависимости от $-\lg \varepsilon_1$

Рис. 15–16 отражают ещё один интересный факт: при увеличении размерности крыловского пространства m_{opt} количество шагов метода EPIRK4(3) становится сопоставимо с количеством шагов ROS4(3)L.

Объяснить это можно так: каждая стадия метода ROS4(3)L требует обращения оператора вида $(I - \tau A)$ в пространстве размерности d системы. Т.е. в

методе ROS4(3)L вычисляется значение функции $(I - \tau A)^{-1}b$ в пространстве \mathbb{R}^d . В то же время в методе EPIRK4(3) вычисляется значение функций $e^{\tau A}b$ и $\varphi_{3j}(\tau A)b$ в пространстве Крылова размерности $m < d$.

Оператор $(I - \tau A)^{-1}$ является $(0, 1)$ -аппроксимацией Паде для оператора $e^{\tau A}$. Ошибка отклонения $(I - \tau A)^{-1}$ от $e^{\tau A}$ складывается из

- ошибки аппроксимации Паде;
- ошибки крыловской аппроксимации.

При увеличении m_{opt} уменьшается ошибка крыловской аппроксимации, а следовательно, отклонение $(I - \tau A)^{-1}$ от $e^{\tau A}$, поэтому размеры шагов и их количества становятся сопоставимыми.

Время расчёта. Ещё одной характеристикой быстродействия метода является время расчёта. На рис. 17–18 представлены диаграммы времени исполнения расчёта методов ROS4(3)L и EPIRK4(3) на тестовой ЭВМ в зависимости от значения $-\lg \varepsilon_1$, где $\text{AbsTol} = \text{RelTol} = \varepsilon_1$. Рис. 17 соответствует случаю, когда в методе EPIRK4(3) в (1.9) параметр $Tol = 10^{-12}$, рис. 18 — когда $Tol = \varepsilon_1$.

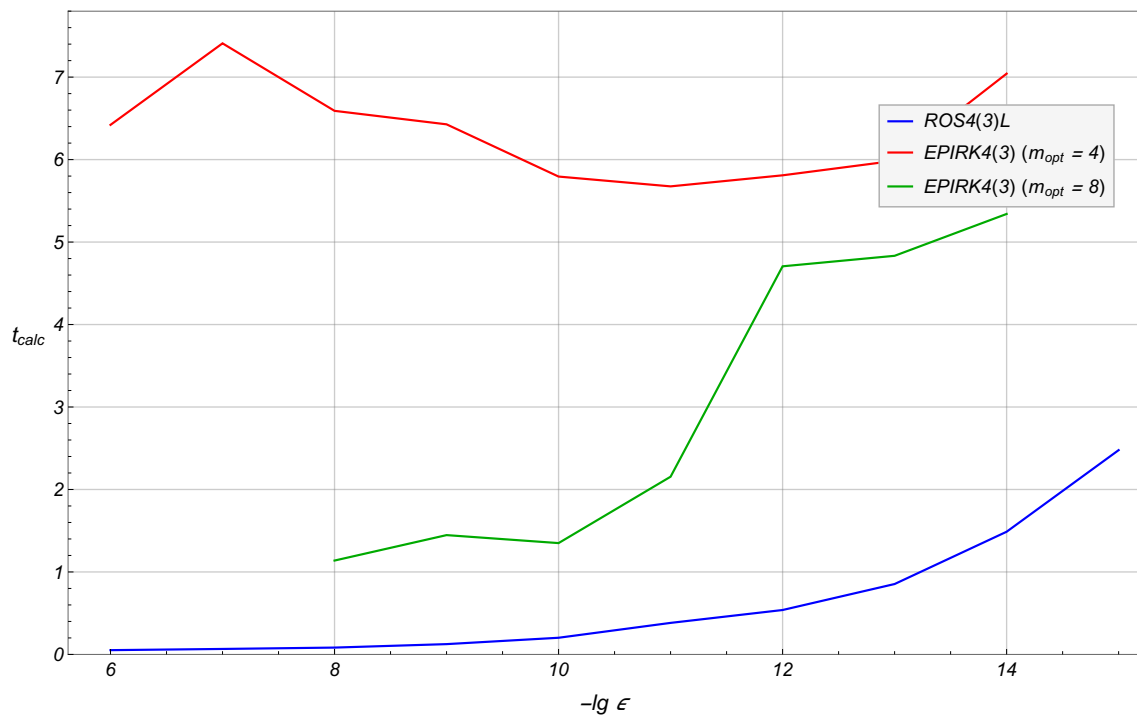


Рис. 17. Время расчёта методов ROS4(3)L и EPIRK4(3) в зависимости от $-\lg \varepsilon_1$

Результаты рисунков примерно соответствуют результатам рис. 15–16: время расчёта метода EPIRK4(3) находится в прямой зависимости с количеством выполненных шагов.

Итог. Видно, что решения, получаемые методами ROS4(3)L и EPIRK4(3), качественно похожи. При этом метод ROS4(3)L ведёт расчёт быстрее и с более крупными шагами.

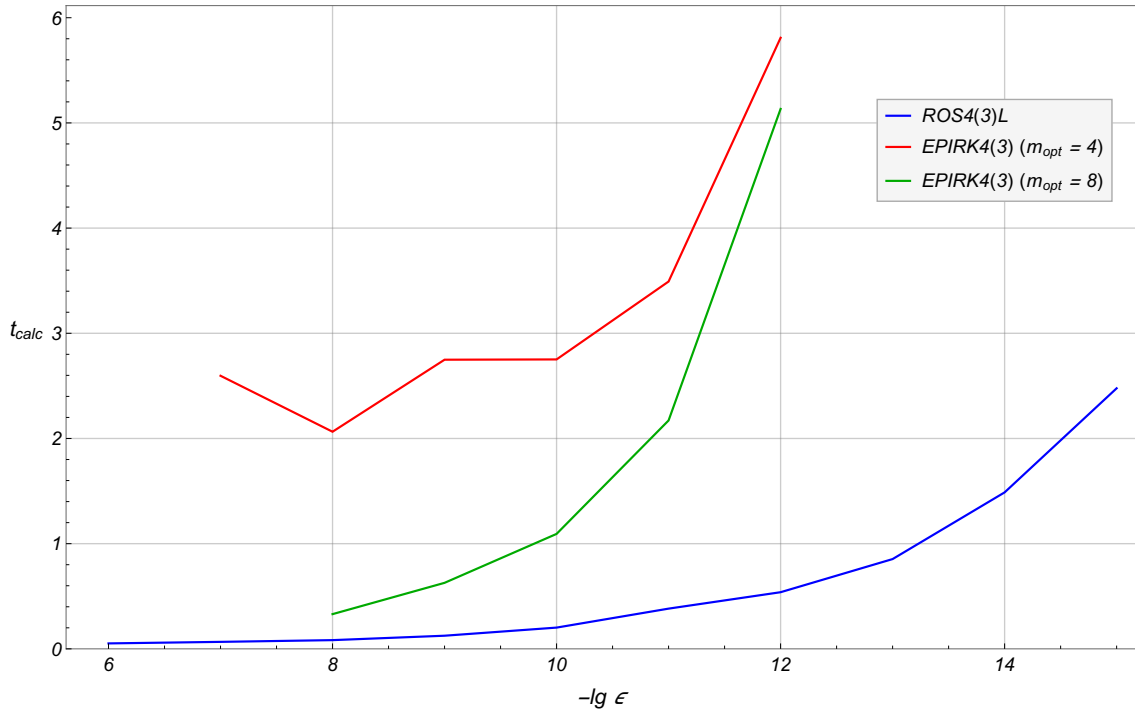


Рис. 18. Время расчёта методов ROS4(3)L и EPIRK4(3) в зависимости от $-\lg \epsilon_1$

Авторская реализация метода EPIRK4(3) не дала существенного выигрыша в расчёте по сравнению с методом ROS4(3)L. Это можно объяснить следующими причинами.

- **Использование неточно заданной матрицы Якоби.** В реализации методов ROS4(3)L и EPIRK4(3) матрица Якоби вычисляется по формуле центральных разностей с шагом $h = 10^{-5}$. Формулы (1.4) метода и его условия порядка (3.1) получены в предположении, что матрица Якоби вычисляется точно. Поэтому в случае неточной матрицы Якоби порядок сходимости и качество расчёта могут снижаться.
- **Тривиальная реализация матричных функций φ_{3j} .** В данной работе вычисление функций $\varphi_{3j}(z)$ реализовано следующим образом:
 - 1) функция e^{hA} вычисляется с помощью функции `expm` пакета MATLAB;
 - 2) функции $\varphi_{3j}(z)$ при $z = hA$ вычисляются по определению. Например, для $\varphi_{30}(z)$:

$$\varphi_{30}(z) = \frac{e^z - 1}{z} \implies \varphi_{30}(hA) = (e^{hA} - I)A^{-1}/h,$$

где I — единичная матрица соответствующей размерности. Другие функции φ_{3j} вычисляются аналогично.

Такой подход может приводить к существенному нарастанию погрешности при малых $\|A\|$, что сказывается на величине шага.

- **Размерность пространства Крылова.** Ранее сказано, что с помощью шага h метод EPIRK4(3) управляет погрешностью крыловской аппроксимации. Т.е. в алгоритме одновременно накладывается несколько условий на шаг.
- **Размерность задачи.** Рассматриваемая задача имеет относительно малую размерность $M = 15$ и заполненную матрицу Якоби. В таком случае выигрыш от применения крыловских аппроксимаций (по сравнению с решением линейной системы с помощью LU -разложения) может быть не таким большим.

В работе [6] продемонстрирован выигрыш от вычисления e^{hA} методами пространств Крылова по сравнению с вычислением $(I - hA)^{-1}$ теми же методами для достаточно больших задач.

Всё это говорит о необходимости дальнейшего развития программной реализации метода EPIRK4(3) и других экспоненциальных методов для решения задач горения углеводородных топлив.

Заключение

В работе представлена конструкция методов семейств EPIRK3 и EPIRK4 из работы [6] и обсуждены особенности их реализации. Обнаружено, что методы семейства EPIRK4 не могут обладать четвёртым порядком сходимости. Представлен расширенный набор условий порядка 4 и построен соответствующий метод EPIRK4(3).

Эффективность и быстродействие метода протестированы на примере задачи горения углеводородных топлив. Авторская реализация метода не позволила добиться выигрыша в быстродействии или величине шага. Необходимы дальнейшие исследования и более аккуратная реализация вычислительного алгоритма.

Список литературы

- [1] Hairer E., Wanner G. Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. Berlin. Springer. 2010. 614 p. DOI:10.1007/978-3-642-05221-7
- [2] Галанин М.П., Савенков Е.Б. Методы численного анализа математических моделей. М.: Изд-во МГТУ им. Н.Э.Баумана. 2010. 591 с.
- [3] Галанин М.П., Исаев А.В., Конев С.А. О моделировании образования сажи при диффузионном горении углеводородных топлив // Препринты ИПМ им. М.В. Келдыша. 2019. № 110. 32 с. DOI:10.20948/prepr-2019-110

- [4] *Галанин М.П., Исаев А.В., Конев С.А.* Математическая модель образования сажи при диффузионном горении толуола // Математическое моделирование. 2020. Т. 32. № 6. С. 66–80. DOI:10.20948/mm-2020-06-05
- [5] *Галанин М.П., Конев С.А.* О методе Лоусона и условиях жёсткого порядка для экспоненциальных методов //Препринты ИПМ им. М.В. Келдыша. 2018. № 261. 26 с. DOI:10.20948/prepr-2018-261
- [6] *Tokman M.* Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods //J. Comput. Phys. 2006. V. 213. № 2. pp. 748–776. DOI:10.1016/j.jcp.2005.08.032
- [7] *Saad Y.* Iterative methods for sparse linear systems. Philadelphia. SIAM. 2003. 528 p. DOI:10.1137/1.9780898718003
- [8] *Saad Y.* Analysis of some Krylov subspace approximations to the matrix exponential operator //SIAM J. Numer. Anal. 1992. V 29. pp. 209–228. DOI:10.1137/0729014
- [9] *Hochbruck M., Lubich Ch.* On Krylov subspace approximations to the matrix exponential operator //SIAM J. Numer. Anal. 1997. V. 34. pp. 1991–1925. DOI:10.1137/S0036142995280572
- [10] *Hairer E., Nørsett S.P., Wanner G.* Solving Ordinary Differential Equations I. Nonstiff Problems. Berlin. Springer. 1993. 528 p. DOI:10.1007/978-3-540-78862-1
- [11] *Hochbruck M., Lubich Ch., Selhofer H.* Exponential integrators for large systems of differential equations //SIAM J. Sci. Comput. 1998. V. 19. pp. 1552–1574. DOI:10.1137/S1064827595295337
- [12] *Moler C., van Loan C.* Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later //SIAM Rev. 2003. V. 45. № 1. pp. 3–49. DOI:10.1137/S00361445024180
- [13] *Арушанян О.Б., Залёткин С.Ф., Калиткин Н.Н.* Тесты для вычислительного практикума по обыкновенным дифференциальным уравнениям //Вычислительные методы и программирование. 2002. Т. 3. № 3. с. 11–19. URL: <http://mi.mathnet.ru/vmp761>

Содержание

Введение	3
1. Формулировка методов EPIRK	3
1.1. Постановка задачи	3
1.2. Расчётные формулы	4
2. Тестирование метода	11
2.1. Постановка задачи	11
2.2. Метод EPIRK3A	12
2.3. Метод EPIRK4A	13
3. Коррекция методов	14
3.1. Условия порядка	14
3.2. Тестирование метода	15
3.3. Вложенный метод	16
4. Моделирование горения УВТ	16
4.1. Качественные результаты	17
4.2. Время расчёта	21
Заключение	26