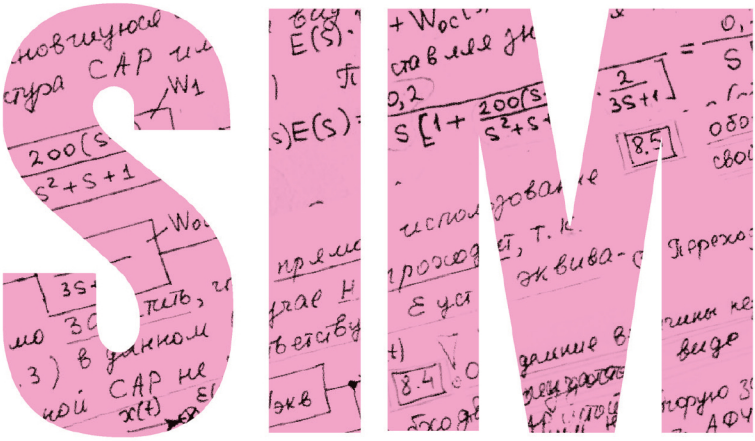
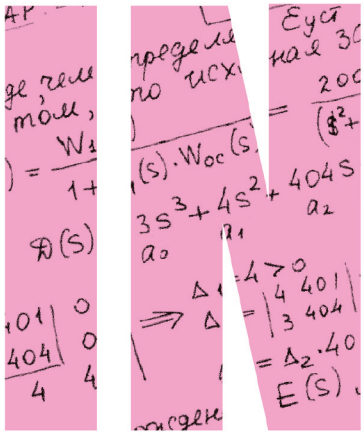
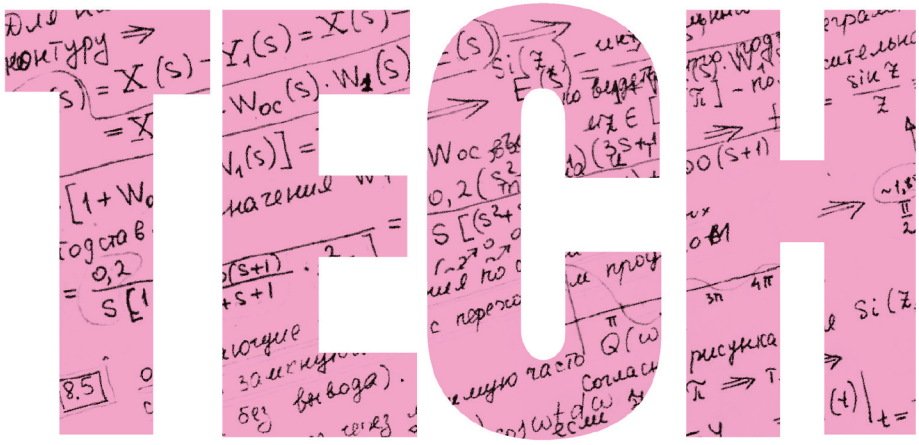




Леонид Скворцов



Численное решение обыкновенных дифференциальных и дифференциально-алгебраических уравнений

Второе издание

ДМК ПРЕСС издательство

Ссылка на дополнительные материалы книги

Книга посвящена численному решению задач с начальными условиями для обыкновенных дифференциальных и дифференциально-алгебраических уравнений. Рассматриваются явные и неявные, одношаговые и многошаговые методы, среди которых новые оригинальные методы. Особое внимание уделено решению жестких задач (в том числе и с использованием специальных явных методов), а также решению дифференциально-алгебраических задач высших индексов. Наряду с теоретическими результатами приведены результаты решения тестовых задач и рассмотрены вопросы программной реализации численных методов.

Для всех, кто интересуется численными методами решения дифференциальных и дифференциально-алгебраических уравнений.

ISBN 978-5-93700-143-6

9 785937 001436 >

Москва 2022

Леонид Скворцов

Численное решение обыкновенных дифференциальных и дифференциально-алгебраических уравнений

Второе издание, исправленное и дополненное

ДМК Пресс
издательство
Москва, 2023

УДК 517.912
ББК 22.193
С42

Скворцов Л. М.
С42 Численное решение обыкновенных дифференциальных и дифференциально-алгебраических уравнений. 2-е изд., испр. и доп. – М.: ДМК Пресс, 2022. – 236 с.

ISBN 978-5-93700-143-6

Книга посвящена численному решению задач с начальными условиями для обыкновенных дифференциальных и дифференциально-алгебраических уравнений. Рассматриваются явные и неявные, одношаговые и многошаговые методы, среди которых новые оригинальные методы. Особое внимание уделено решению жестких задач (в том числе и с использованием специальных явных методов), а также решению дифференциально-алгебраических задач высших индексов. Наряду с теоретическими результатами приведены результаты решения тестовых задач и рассмотрены вопросы программной реализации численных методов.

Для всех, кто интересуется численными методами решения дифференциальных и дифференциально-алгебраических уравнений.

УДК 517.912
ББК 22.193

ISBN 978-5-93700-143-6

Содержание

▼ Глава 3

▼ Глава 4

▼ Глава 5

▼ Глава 6

▼ Глава 7

▼ Глава 8

Предисловие

На современном этапе развития цивилизации прогресс во многих областях науки и техники определяется степенью внедрения в научно-технические разработки математического и имитационного моделирования. Замена натурных экспериментов компьютерным моделированием существенно удешевляет и ускоряет научные исследования, а также позволяет избежать трагических ошибок, вызванных критическими состояниями человеческого организма, технических объектов, окружающей среды.

Многие процессы в природе и технике описываются обыкновенными дифференциальными уравнениями (ОДУ) и дифференциальными уравнениями в частных производных. Лишь в редких случаях такие уравнения имеют аналитическое решение, поэтому приходится решать их численно. Уравнения в частных производных можно привести к системе ОДУ, применив метод прямых (method of lines), т. е. заменив пространственные производные конечными разностями. Переменные, входящие в систему ОДУ, могут быть связаны некоторыми алгебраическими соотношениями, в этом случае получаем систему дифференциально-алгебраических уравнений (ДАУ).

Таким образом, многие явления и процессы в физике, химии, астрономии, биологии, технике могут быть описаны в виде системы ОДУ или ДАУ, а моделирование этих процессов сводится к численному решению таких систем. Поэтому очевидна важность построения и реализации в виде компьютерных программ эффективных методов численного решения ОДУ и ДАУ. Компьютерные программы – решатели ОДУ и ДАУ – рассматривались в [4, 74, 75, 128]. Такие программы удобно применять, если математическая модель задана непосредственно в виде системы уравнений. Однако в различных предметных областях применяют также и другие способы представления математической модели: структурные схемы систем автоматического управления, электрические схемы в электротехнике и электронике, кинематические схемы в механике и робототехнике и т. д. Для удобства моделирования таких систем программы снабжают специализированным интерфейсом, позволяющим пользователю задавать модель в удобном виде.

Наиболее известным и популярным программным средством моделирования разнородных (т. е. содержащих элементы разной физической природы) динамических систем является пакет Simulink, входящий в систему

математических вычислений MATLAB. Среди аналогичных отечественных разработок особого внимания заслуживает программное обеспечение (ПО) «Среда динамического моделирования технических систем SimInTech». Далее будем использовать название ПО SimInTech или SimInTech (сокращение от Simulation In Technic). ПО SimInTech является результатом модернизации программного комплекса «Моделирование в технических устройствах» (ПК МВТУ) [24–26], который был разработан коллективом ученых и выпускников МГТУ им. Н. Э. Баумана под руководством О. С. Козлова. Разработка, а также дальнейшее развитие, сопровождение и распространение ПО SimInTech выполняются специалистами ООО «3В Сервис» (www.3v-services.com). Ознакомиться с ПО SimInTech можно в [22] и на сайте http://simintech.ru.

Автор этой книги участвовал в разработке ПК МВТУ и ПО SimInTech и имеет большой опыт по реализации самых различных алгоритмов. Возможность включить свои алгоритмы в современный программный продукт стала серьезным стимулом для исследований в области численного решения ОДУ и ДАУ. В настоящее время SimInTech имеет обширный набор методов решения ОДУ и ДАУ, содержащий явные и диагонально-неявные методы Рунге–Кутты, явные адаптивные методы, неявные методы Гира и Эйлера. Почти все методы являются оригинальными либо содержат оригинальные решения. Особый интерес представляют явные адаптивные методы, которые позволяют эффективно решать многие жесткие системы. Наряду с методами, реализованными в SimInTech и показавшими высокую эффективность при решении множества прикладных задач, в книге рассмотрены новые перспективные методы. Уделено внимание эффективной реализации методов и тестовому сравнению с известными решателями, среди которых решатели системы MATLAB+Simulink и RADAU5.

Глава 1 является вводной, в ней даны постановки задачи Коши для систем ОДУ и ДАУ, рассмотрены различные классы задач и методов их решения. К трудным для численного решения отнесены жесткие, колебательные и плохо обусловленные задачи, задачи с разрывами и ДАУ высших индексов. Предложены количественные меры жесткости, колебательности и неустойчивости задачи Коши, приведены значения этих мер для известных тестовых задач.

В главе 2 рассмотрены явные методы Рунге–Кутты для нежестких задач. Приведены условия порядка до 5-го включительно и даны рекомендации по выбору оптимальных коэффициентов. Рассмотрены два способа построения вложенных пар методов с оцениванием ошибки. Приведены коэффициенты известных и новых вложенных пар до 5-го порядка, а также результаты их тестового сравнения. Рассмотрен эффективный способ решения задач с разрывами.

В главе 3 рассмотрены неявные одношаговые методы низкой точности. Для реализации выбраны три метода 2-го порядка: трапеций, TR-BDF2 и Лобатто IIIC. На примере метода трапеций рассмотрены четыре схемы реализации неявных методов. Представлены детальные схемы реализации выбранных ме-

тодов и новые схемы типа Розенброка. Приведены результаты их тестового сравнения с решателями MATLAB.

Глава 4 содержит теоретические и экспериментальные результаты о сходимости методов Рунге–Кутты при решении жестких и дифференциально-алгебраических задач. Предложены простейшие модельные уравнения, объясняющие снижение точности и порядка при решении таких задач. Получены выражения для ошибок решения модельных уравнений и показано, что минимизация этих ошибок позволяет построить методы повышенной точности, свободные от снижения порядка.

В главе 5 рассмотрены диагонально-неявные методы Рунге–Кутты порядков 3, 4 и 5. Получены упрощенные условия порядка, а также функции погрешности, описывающие поведение жестких составляющих ошибки. Построены конкретные методы с минимизированными функциями погрешности. Рассмотрены схемы реализации и приведены результаты решения тестовых задач в сравнении с решателем RADAU5.

В главе 6 рассмотрены неявные методы, обладающие повышенной точностью при решении жестких задач и ДАУ. К ним относятся коллокационные методы, узлы которых выбраны из условия минимизации ошибок решения модельных уравнений, неявные методы с явными внутренними стадиями, а также двухшаговый метод 5-го порядка, который не снижает точности и порядка при решении жестких задач и ДАУ индексов 2 и 3. Приведены результаты тестового сравнения с методами Радо IIA и Лобатто IIIA.

В главе 7 рассмотрены явные одношаговые и многошаговые методы с расширенными областями устойчивости, позволяющие эффективно решать жесткие задачи с распределенным вещественным спектром матрицы Якоби. Предложен простой и эффективный способ расчета «почти оптимальных» многочленов устойчивости произвольной степени. Рассмотрены способы построения методов Рунге–Кутты с заданным многочленом устойчивости и вложенной формулой для оценивания ошибки. Построены методы порядков 2, 3 и 4; приведены результаты решения тестовых задач (в том числе и в сравнении с решателями RKC, DUMKA3, ROCK4).

В главе 8 рассмотрены явные адаптивные методы, использующие полученные на основе предварительных стадий покомпонентные оценки наибольшего по модулю собственного значения матрицы Якоби для настройки формулы интегрирования на решаемую задачу. Приведены расчетные схемы одношаговых методов порядков 1, 2, 3 и многошагового метода переменного порядка. Показано, что такие методы могут быть эффективными для решения жестких и колебательных задач. Приведены результаты численных экспериментов, которые показали, что при решении многих жестких задач явные адаптивные методы не уступают неявным методам, а иногда и превосходят их.

В качестве инструментов для исследования методов численного решения ОДУ и ДАУ автор использовал алгоритмы, реализованные в ПК МВТУ, ПО SimInTech, а также в системе компьютерных вычислений MathCAD. Такие про-

граммные инструменты заметно сокращают объем рутинной работы по построению, реализации и тестированию новых методов. Некоторые из этих программ, а также некоторые дополнительные материалы по численному решению ОДУ и ДАУ размещены на сайте ООО «3В Сервис» (http://3v-services.com/books/978-5-97060-636-0/).

Автор благодарен коллективу ООО «3В Сервис» за помощь и содействие в издании книги.

Во втором издании исправлены некоторые формулы; добавлен новый материал в разделы 5.1, 5.3, 5.4, 5.12, 5.13, 8.4, 8.9; дополнен список литературы работами [152–155].

Задача Коши и методы ее решения

ГЛАВА 1

1.1. Обыкновенные дифференциальные уравнения

Рассмотрим задачу Коши для системы ОДУ

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad 0 \leqslant t \leqslant T, \tag{1.1}$$

где t – независимая переменная, $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ – вектор переменных состояния, $\mathbf{f}(t, \mathbf{y}) = (f_1(t, \mathbf{y}), \ldots, f_n(t, \mathbf{y}))^{\mathrm{T}}$ – нелинейная векторная функция. Если решается задача моделирования во времени, то t – модельное время. Система ОДУ называется *автономной*, если правая часть не зависит от t, т. е. $\mathbf{f}(t, \mathbf{y}) = \mathbf{f}(\mathbf{y})$. Неавтономную систему (1.1) нетрудно привести к автономной, добавив уравнение $t' = 1$. Поэтому все теоретические результаты, полученные для автономных систем, справедливы также и для неавтономных систем.

Численное решение (интегрирование) задачи (1.1) сводится к нахождению последовательности векторов $\mathbf{y}_1, \ldots, \mathbf{y}_N$, аппроксимирующих истинное решение $\mathbf{y}(t)$ в дискретные моменты модельного времени $t_1, \ldots, t_N = T$. Интервал между двумя соседними моментами времени $h_i = t_{i+1} - t_i$ называется шагом интегрирования (размером шага). Размер шага может быть постоянным ($h_i = h = \text{const}$) либо переменным.

В дальнейшем будем предполагать, что задача (1.1) имеет единственное решение, а функция $\mathbf{f}$ – гладкая в любой точке решения на интервале интегрирования $[0, T]$. Тогда на всем интервале определена и непрерывна матрица Якоби системы (1.1)

$$\mathbf{J}(t) = \frac{\partial \mathbf{f}(t, \mathbf{y}(t))}{\partial \mathbf{y}}. \tag{1.2}$$

Требование гладкости правой части не всегда согласуется с реальными моделями, в составе которых могут быть различные релейные и переключательные элементы. При наличии таких элементов будем предполагать, что число переключений конечно, а весь интервал интегрирования можно разбить на несколько интервалов, на каждом из которых функция $\mathbf{f}$ остается гладкой. Тогда решения «склеиваются», т. е. на каждом последующем интервале в качестве начального условия принимается решение, полученное в конце текущего ин-

тервала. Таким образом, и в этом случае можно считать функцию $\mathbf{f}$ гладкой, а якобиан (1.2) – непрерывным.

Простейшим методом численного интегрирования является метод Эйлера, формула которого при решении задачи (1.1) имеет вид

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\mathbf{f}(t_i, \mathbf{y}_i). \tag{1.3}$$

Этот метод является *явным*, поскольку вектор переменных в очередной момент модельного времени явно выражается через уже рассчитанный вектор в предыдущий момент. Для решения ОДУ применяют также *неявные* методы, простейший из них – неявный (обратный) метод Эйлера, формула которого

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\mathbf{f}(t_{i+1}, \mathbf{y}_{i+1}). \tag{1.4}$$

В неявных методах формула шага интегрирования представляет собой систему нелинейных алгебраических уравнений, для решения которой используют итерационные методы (обычно это метод Ньютона или его модификации).

Методы решения ОДУ подразделяются также на *одношаговые* (Эйлера, Рунге–Кутты, Розенброка) и *многошаговые* (Адамса, Гира, прогноза-коррекции и др.). В одношаговых методах для нахождения $\mathbf{y}_{i+1}$ используются только векторы $\mathbf{y}_i$ и $\mathbf{y}'_i = \mathbf{f}(t_i, \mathbf{y}_i)$. В многошаговых методах используется также информация, полученная на предыдущих шагах: в k-шаговом методе это $\mathbf{y}_{i-1}, \ldots, \mathbf{y}_{i-k+1}$, $\mathbf{y}'_{i-1}, \ldots, \mathbf{y}'_{i-k+1}$. Первый шаг всегда выполняется одношаговым методом. На последующих шагах может быть произведен переход на многошаговый метод с последовательным увеличением числа используемых шагов. Не следует продолжать решение многошаговым методом при резком изменении правой части системы ОДУ, поскольку накопленная информация оказывается устаревшей. В этом случае следует отбросить всю предыдущую информацию и вновь начать решение одношаговым методом.

1.2. Точность и устойчивость численных методов

Основные характеристики методов численного решения ОДУ связаны с их точностью и устойчивостью. Размер шага выбирается исходя из точности численного решения. Ошибка интегрирования

$$\mathbf{e}_i = \mathbf{y}(t_i) - \mathbf{y}_i \tag{1.5}$$

складывается из двух составляющих: методической (или ошибки дискретизации), обусловленной неточностью метода, и ошибки округления, обусловленной ограниченностью разрядной сетки компьютера. При уменьшении размера шага методическая ошибка уменьшается, а ошибка округления возрастает. Ошибка округления обычно пренебрежимо мала и заметно сказывается лишь в некоторых исключительных случаях.

При точном выполнении всех вычислений ошибка состоит только из методической составляющей. При заданном начальном условии ошибка (1.5) называется *глобальной*, поскольку она получена в результате накопления ошибок на

всех предыдущих шагах. Ошибка, полученная на одном шаге при предположении, что все используемые предыдущие значения точные, называется *локальной*. Для одношаговых методов ошибка (1.5) будет локальной, если $\mathbf{y}_{i-1} = \mathbf{y}(t_{i-1})$. При некоторых (достаточно общих) предположениях локальная ошибка при $h \to 0$ пропорциональна h^{p+1}, где целое число p называется *порядком сходимости* метода. Глобальная ошибка получается в результате накопления локальных ошибок на всех шагах, поэтому она пропорциональна числу шагов $N = T/h$ и усредненной локальной ошибке. В результате при $h \to 0$ глобальная ошибка пропорциональна h^p. Для явного и неявного методов Эйлера $p = 1$, поэтому уменьшение размера шага в 2 раза приводит к уменьшению локальной ошибки примерно в $2^{p+1} = 4$ раза. Но при этом в 2 раза увеличивается число шагов, поэтому глобальная ошибка уменьшится только в $2^p = 2$ раза.

Порядок используемого метода следует соотносить с требуемой точностью численного решения. Чтобы убедиться в этом, рассмотрим задачу

$$y_1' = -22y_1 + 20y_2^2,\ y_2' = y_1 - y_2 - y_2^2,\quad y_1(0) = 1,\ y_2(0) = 1,\ 0 \leqslant t \leqslant 1, \tag{1.6}$$

решение которой $y_1(t) = \exp(-2t)$, $y_2(t) = \exp(-t)$. Ошибку решения оценим в виде $\varepsilon = \max(e(t_i),\ 0 \le t_i \le 1)$, где $e(t_i)$ – евклидова норма абсолютной ошибки в точке $t_i = ih$. Вычислительные затраты оценим числом вычислений правой части Nf на всем интервале. Зависимости ошибки от вычислительных затрат для явных одношаговых методов 1-го, 2-го и 4-го порядков приведены на рис. 1.1. Из этого рисунка видно, что выбор порядка метода определяется требованиями к точности. Если допустима достаточно большая ошибка, преимущество имеют методы невысоких порядков, позволяющие получить решение с малыми вычислительными затратами. А при малой допустимой ошибке следует использовать методы более высоких порядков.

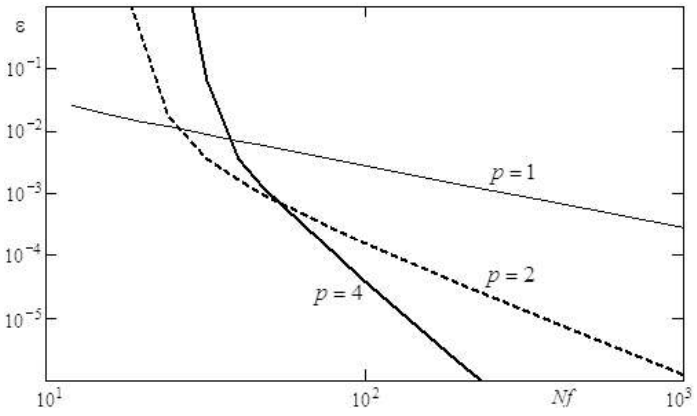


Рис. 1.1. Зависимости ошибки решения задачи (1.6) от вычислительных затрат для методов порядков 1, 2 и 4

При численном решении дифференциальные уравнения заменяются разностными. Решение полученных разностных уравнений может оказаться неустойчивым, хотя исходная система ОДУ была устойчивой. Неустойчивость проявляется как катастрофический рост ошибки численного решения при увеличении размера шага. Покажем это на примере задачи

$$y' = 50(e^{-t} - y),\ y_0 = 1,\ 0 \le t \le 3 \tag{1.7}$$

с решением $y(t) = (50/49)e^{-t} - (1/49)e^{-50t}$, которую будем решать методом Эйлера. При размере шага $h < 0.04$ численное решение сходится и почти не отличается от точного решения. Но уже при $h = 3/73 = 0.0411$ получаем быстро расходящееся решение, показанное на рис. 1.2 тонкой линией (толстой линией показано точное решение).

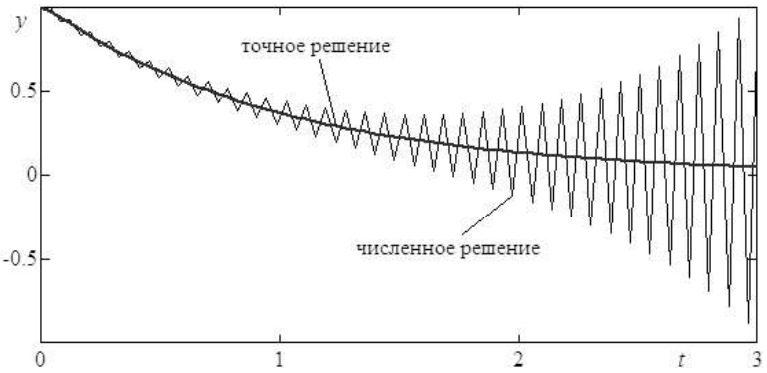


Рис. 1.2. Точное и численное решения задачи (1.7) при $h = 3/73$

Обычно для конкретной задачи и конкретного метода существует некоторое граничное значение шага $h_{\max}$, превышение которого приводит к неустойчивости численного решения. Исследуем устойчивость численных методов решения ОДУ на примере линейной системы

$$\mathbf{y}' = \mathbf{J}\mathbf{y},\ \mathbf{y}(t_0) = \mathbf{y}_0, \tag{1.8}$$

где $\mathbf{J}$ – матрица размером $n \times n$. Система (1.8) устойчива, если все собственные значения матрицы $\mathbf{J}$ имеют отрицательные действительные части. Применим для решения этой системы явный метод Эйлера (1.3). Подставив в (1.3) $\mathbf{y}_i' = \mathbf{J}\mathbf{y}_i$, получим

$$\mathbf{y}_{i+1} = (\mathbf{I} + h\mathbf{J})\mathbf{y}_i, \tag{1.9}$$

где $\mathbf{I}$ – единичная матрица. Мы получили систему линейных разностных уравнений, решение которой аппроксимирует решение исходной системы ОДУ (1.8). Полученная система (1.9) будет устойчивой, если все собственные числа матрицы $\mathbf{I} + h\mathbf{J}$, равные $1 + h\lambda_j$, по абсолютной величине меньше 1 (λ_j – собственные числа матрицы $\mathbf{J}$). Таким образом, условие устойчивости численного решения системы (1.8) методом Эйлера запишется в виде системы неравенств

$$|1 + h\lambda_j| < 1,\ j = 1, \ldots, n. \tag{1.10}$$

Вместо системы (1.8) для исследования устойчивости используют скалярное линейное уравнение (уравнение Далквиста)

$$y' = \lambda y, \tag{1.11}$$

в котором λ может быть комплексным числом. Применение одношагового метода типа Рунге–Кутты для решения этого уравнения приводит к формуле интегрирования $y_{i+1} = R(h\lambda)y_i$, где $R(z)$ называется *функцией устойчивости*. Область, задаваемая неравенством $|R(z)| \leq 1$, называется *областью устойчивости*. Функция устойчивости явного метода Эйлера $R(z) = 1 + z$, а его область устойчивости задается неравенством $|1 + z| \leq 1$ и представляет собой круг единичного радиуса с центром в точке (–1, 0). Функция устойчивости неявного метода Эйлера $R(z) = (1 - z)^{-1}$, а его область устойчивости задается неравенством $|1 - z| \geq 1$. При интегрировании устойчивой линейной системы (1.8) размер шага следует выбирать таким, чтобы все числа $h\lambda_j$ попали в область устойчивости.

Метод называется *A-устойчивым*, если его область устойчивости включает всю левую полуплоскость комплексной плоскости. Метод называется *A*(α)-*устойчивым*, если его область устойчивости включает сектор, задаваемый неравенством $|\arg(-z)| \leq \alpha$. Для методов решения жестких задач часто требуют также выполнения условия $R(\infty) = 0$. *A*- и *A*(α)-устойчивые методы, удовлетворяющие этому условию, называются, соответственно, *L*- и *L*(α)-*устойчивыми*. Приведенные определения распространяются и на многошаговые методы, в этом случае вместо неравенства $|R(z)| \leq 1$ рассматривают аналогичные неравенства для корней характеристического полинома разностной схемы, которые также зависят от $z = h\lambda$.

1.3. Жесткие задачи

Пусть все собственные числа матрицы **J** в системе (1.8) вещественные и отрицательные. Тогда условие (1.10) запишется в виде $h < 2\tau_{\min}$, где минимальная постоянная времени $\tau_{\min} = \min(\tau_i,\ i = 1, \ldots, n)$, $\tau_i = -1/\lambda_i$. Таким образом, для обеспечения устойчивости численного решения явным методом Эйлера размер шага должен быть меньше двух минимальных постоянных времени. Аналогичные условия накладываются на размер шага и при использовании других явных методов. Время переходного процесса в системе (1.8) определяется максимальной постоянной времени $\tau_{\max}$ и составляет примерно $3\tau_{\max}$. При большом разбросе постоянных времени $\tau_{\max}/\tau_{\min}$ число шагов интегрирования оказывается очень большим, что может привести к большим затратам машинного времени. В то же время размер шага неявного метода Эйлера и многих других неявных методов ограничен только требованиями к точности решения и может быть значительно больше $\tau_{\min}$.

Рассмотрим, например, задачу

$$y_1' = -y_1/\tau_1,\ \ y_2' = (y_1 - y_2)/\tau_2,\ y_1(0) = y_2(0) = 1, \tag{1.12}$$

имеющую при $\tau_1 \neq \tau_2$ решение

$$y_1 = e^{-t/\tau_1}, \quad y_2 = \frac{\tau_1}{\tau_1 - \tau_2} e^{-t/\tau_1} - \frac{\tau_2}{\tau_1 - \tau_2} e^{-t/\tau_2}.$$

При $\tau_1 >> \tau_2$ компонента решения, соответствующая постоянной τ_2, мала и быстро затухает. Несмотря на это, шаг интегрирования при использовании явных методов следует выбирать малым на всем интервале интегрирования. В некоторых случаях малой постоянной времени можно пренебречь, заменив, например, второе уравнение в (1.12) на равенство $y_2 = y_1$. Но в общем случае подобная замена может привести к появлению алгебраического уравнения вместо дифференциального. К тому же для сложных нелинейных систем выделить в явном виде малую постоянную времени не всегда возможно.

Задачи, подобные рассмотренной выше, получили название *жестких*. Жесткие задачи характеризуются наличием собственных значений матрицы Якоби, имеющих большие отрицательные действительные части. Соответствующие составляющие решения быстро затухают и, за исключением малых участков (пограничных слоев), пренебрежимо малы. Решение жестких задач традиционными явными методами требует больших вычислительных затрат, поэтому для их решения обычно применяют неявные методы, которые обеспечивают устойчивое интегрирование с большим размером шага. Неявные методы не свободны от недостатков, к которым относятся прежде всего сложность реализации и необходимость вычислять матрицу Якоби. В тех случаях, когда правая часть содержит разрывы или логические условия, вычисление якобиана может представлять собой сложную и далеко не тривиальную задачу. Отметим также, что при решении некоторых жестких задач применение неявных методов дает неудовлетворительные, а иногда и качественно неверные результаты (примеры таких задач приведены в разделе 8.9). Поэтому наряду с неявными методами разрабатывают специальные явные методы, пригодные для решения жестких задач.

Жесткие задачи весьма разнообразны, поэтому класс таких задач трудно поддается формальному определению, а среди существующих определений нет общепринятого. Наиболее известно определение Ламберта (см. [72]), согласно которому задача Коши называется жесткой, если на всем интервале интегрирования выполняются условия

$$\mathrm{Re}\,\lambda_i < 0, \quad i = 1, \ldots, n;$$

$$S(t) = \frac{\max(\mathrm{Re}(-\lambda_i), i = 1, \ldots, n)}{\min(\mathrm{Re}(-\lambda_i), i = 1, \ldots, n)} >> 1,$$

где величина $S(t)$ названа локальным коэффициентом жесткости. Часто приводят эквивалентное определение, введя постоянные времени $\tau_i = -1/\mathrm{Re}\,\lambda_i$: задача считается жесткой, если имеет большой разброс постоянных времени. Максимальная постоянная времени τ_{max} определяет длительность переходного процесса. Если решение гладкое и не содержит большого числа колебаний, то размер шага из соображений точности может быть выбран значительно боль-

ше τ_{min}. Но при интегрировании явными методами приходится выбирать шаг из соображений устойчивости, т. е. порядка τ_{min}. Таким образом, разброс постоянных времени характеризует вычислительные затраты при интегрировании задачи явными методами.

Определение Ламберта не охватывает всего класса жестких задач, поскольку оно применимо только к устойчивым системам, при условии что постоянные времени существенно не изменяются на интервале интегрирования. Другой недостаток этого определения виден на примере задачи

$$y' = \lambda(y - \sin t) + \cos t, \quad y(0) = 0,$$

имеющей гладкое решение $y(t) = \sin t$, не зависящее от λ. При больших отрицательных значениях λ для устойчивого решения этой задачи классическими явными методами приходится выбирать очень малый шаг интегрирования. Таким образом, данная задача проявляет свойство жесткости, хотя по определению Ламберта не является таковой, поскольку имеет только одну постоянную времени.

Авторы известной книги по численному решению жестких задач [75] Э. Хайрер и Г. Ваннер полагают, что наиболее практичным определением понятия «жесткий» является самое раннее, данное в 1952 г. Кертиссем и Хиршфельдером [97]: «Жесткие уравнения – это уравнения, для которых определенные неявные методы, в частности ФДН, дают лучший результат, обычно несравненно более хороший, чем явные методы». В настоящее время известны специальные явные методы, эффективные для многих жестких задач, поэтому под явными методами в этом определении следует понимать классические явные методы Рунге–Кутты и Адамса. Отметим, что термины «жесткие уравнения» и «жесткая задача» применимы к конкретной задаче Коши, т. е. при заданных начальных условиях и интервале интегрирования, поскольку задача, жесткая на большом интервале, может оказаться нежесткой на меньшем интервале или при других начальных условиях.

Данное определение дает практический способ оценивания жесткости задачи как отношения затрат явного метода к затратам неявного метода. Количественной мерой затрат на решение может быть машинное время, необходимое для решения задачи. В наше время нетрудно оценить жесткость задачи, поскольку современные системы математических вычислений имеют в своем составе достаточно обширные наборы методов интегрирования, включая неявные методы и классические явные методы. Однако для оценивания жесткости задачи желательно использовать меру вычислительных затрат, не зависящую от используемого метода и его программной реализации.

1.4. Меры жесткости, колебательности и неустойчивости задачи Коши

В качестве меры затрат на решение жесткой задачи явным методом можно использовать приблизительное число вычислений правой части при умеренных требованиях к точности. Для линейной задачи (1.8) это число пропорциональ-

но интервалу интегрирования T и обратно пропорционально минимальной постоянной времени $\tau_{\min}$. Поэтому для линейной задачи вычислительные затраты можно оценить числом

$$M_{\text{ж}} = \mu T,\ \mu = 1/\tau_{\min} = \max_i \operatorname{Re}(-\lambda_i).$$

Если все собственные значения матрицы **J** вещественные и отрицательные, то значение $M_{\text{ж}}$ равно минимальному числу вычислений $\mathbf{f}(t, \mathbf{y})$, необходимому для устойчивого решения задачи (1.8) явным методом Рунге–Кутты 2-го порядка с функцией устойчивости $R(z) = 1 + z + z^2/2$. Такой метод реализован в решателе RK2(1), рассмотренном в разделе (2.4). Для явного метода Эйлера $M_{\text{ж}}$ – минимальное число вычислений правой части при решении задач, удовлетворяющих условиям $\operatorname{Re}\lambda_i < 0, |\operatorname{Im}\lambda_i| \le |\operatorname{Re}\lambda_i|$.

В общем случае нелинейной неавтономной задачи (1.1) якобиан (1.2) и его спектр зависят от t, поэтому вычислительные затраты явных методов можно оценить с помощью интеграла

$$M_{\text{ж}} = \int_0^T \max(\max_i \operatorname{Re}(-\lambda_i(t)), 0)dt. \tag{1.13}$$

Величину $M_{\text{ж}}$ назовем *мерой жесткости* задачи Коши. В формуле (1.13) оцениваются вычислительные затраты, вызванные только жесткостью задачи. Реальное число шагов может значительно превышать эти оценки вследствие негладкости правой части, наличия больших собственных значений вблизи мнимой оси или в правой полуплоскости (колебательные и плохо обусловленные задачи), а также по другим причинам.

Трудности, возникающие при решении задачи Коши, в значительной степени определяются спектром матрицы Якоби. В зависимости от расположения наибольших по модулю собственных значений (в левой полуплоскости, вблизи мнимой оси, в правой полуплоскости) можно выделить жесткие, колебательные и плохо обусловленные задачи. Колебательные задачи имеют собственные значения вблизи мнимой оси, а плохо обусловленные – в правой полуплоскости. Для эффективного решения колебательных задач применяют неявные симметричные методы либо специальные методы, в том числе и явные. Отметим, однако, что характер задачи может изменяться в процессе решения, а также может быть разным для разных компонент.

По аналогии с мерой жесткости оценим также колебательность и неустойчивость задачи Коши. Величину

$$M_{\text{к}} = \int_0^T \max_i \operatorname{Im}(\lambda_i(t))dt$$

назовем *мерой колебательности*, а величину

$$M_{\text{ну}} = \int_0^T \max\left(\max_i \operatorname{Re}(\lambda_i(t)), 0\right)dt$$

назовем *мерой неустойчивости* задачи Коши.

На разных участках решения задача может иметь разный характер, поэтому имеет смысл ввести обобщенный показатель, оценивающий трудность решения задачи Коши при использовании классических методов. Такой показатель примем в виде

$$M_\Sigma = \int_0^T \max_i |\lambda_i(t)|\, dt.$$

Конечно, трудность решения задачи зависит также и от многих других причин, но нас сейчас интересуют характеристики задачи, связанные только со спектром матрицы Якоби.

Вычислим определенные выше меры для конкретного примера. Возьмем наиболее распространенный тест – осциллятор Ван-дер-Поля, уравнения которого имеют вид

$$\begin{aligned} &y_1' = y_2,\ y_2' = \mu(1 - y_1^2)y_2 - y_1, \\ &y_1(0) = 2,\ \ y_2(0) = y_{20},\ \ 0 \le t \le T. \end{aligned} \tag{1.14}$$

Здесь T – период предельного цикла, а значение y_{20} выбрано таким, чтобы начальная точка лежала на траектории предельного цикла. Для вычисления этих значений использовался ПК МВТУ [26]. Вычисленные при различных значениях μ характеристики этой задачи приведены в табл. 1.1. Здесь же приведено число вычислений правой части *Nf* при решении задачи с допуском на ошибку *Tol* = 0.01 методом RK2(1). При μ = 0 задача – чисто колебательная. При увеличении μ возрастает жесткость задачи, а также появляется неустойчивая составляющая. При μ > 10 жесткость задачи пропорциональна μ^2, а число вычислений правой части явного метода практически совпадает с мерой жесткости и на порядки больше, чем у неявного метода. При $\mu = 10^3$ решение задачи неявным методом TR-BDF2, рассмотренным в главе 3, потребовало всего 715 вычислений правой части и 10 вычислений матрицы Якоби.

Таблица 1.1. Характеристики уравнения Ван-дер-Поля (1.14) на одном цикле решения

μ	T	$y_2(0)$	$M_ж$	$M_к$	$M_{ну}$	M_Σ	Nf
0	6.28319	0	0	6.28	0	6.28	113
1	6.66329	–0.16898	9.37	4.13	3.28	13.4	182
10	19.0784	–0.0665099	323.5	4.00	12.87	331.7	653
100	162.837	-6.66654×10^{-3}	2.90×10^4	4.02	24.10	2.90×10^4	29498
1000	1614.40	-6.66667×10^{-4}	2.89×10^6	4.02	35.54	2.89×10^6	2887653

В качестве тестовой задачи обычно используется нормированное уравнение Ван-дер-Поля, полученное путем замены переменных

$$y_1(t) = x_1(t/\mu),\ \ \mu y_2(t) = x_2(t/\mu).$$

В результате получаем

$$\begin{aligned} &x_1' = x_2,\ \ x_2' = \mu^2((1 - x_1^2)x_2 - x_1), \\ &x_1(0) = 2,\ \ x_2(0) = x_{20},\ \ 0 \le t \le T. \end{aligned} \tag{1.15}$$

Такую задачу удобнее исследовать потому, что при больших μ период предельного цикла T почти не зависит от μ, а в пределе при $\mu \to \infty$ получаем $T = 3 - 2\ln(2)$.

Интересно посмотреть, как изменяются собственные числа матрицы Якоби на траектории решения. Для уравнения (1.15) при $\mu^2 = 10$ кривые изменения x_1 и собственных чисел приведены на рис. 1.3. При увеличении μ качественная картина сохраняется, но время перехода переменной x_1 от 1 до –2 сокращается, а собственные числа увеличиваются. Такая задача часто используется как тест для методов численного решения жестких ОДУ. Исследуем уравнение (1.15) при $\mu^2 = 10^6$ (такое значение используется в жесткой тестовой задаче). В этом случае получаем $x_{20} = -0.6666665432102$, $T = 1.614401125809$. Разобьем полупериод $T/2$ на 5 интервалов таким образом, чтобы границами интервалов были моменты t_i, в которые переменная x_1 принимает целые значения. В табл. 1.2 приведены значения переменных и собственных чисел λ_1, λ_2 в эти моменты, а также величины интервалов времени.

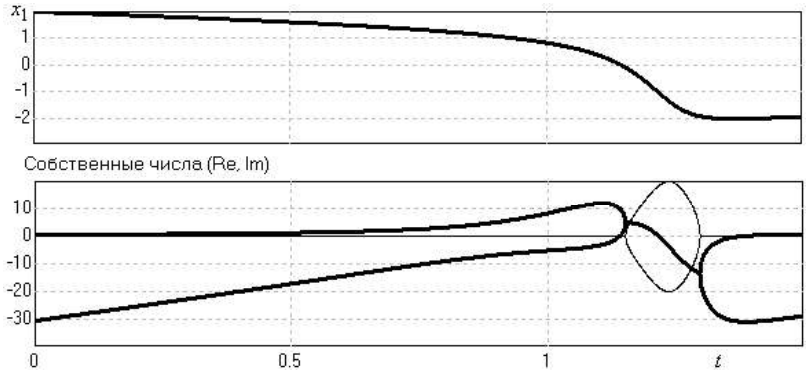


Рис. 1.3. Изменение собственных чисел матрицы Якоби на траектории решения уравнения Ван-дер-Поля (толстые линии – действительные части, тонкие – мнимые)

Таблица 1.2. Собственные числа уравнения Ван-дер-Поля (1.15) при $\mu^2 = 10^6$

i	$x_1(t_i)$	$x_2(t_i)$	λ_1	λ_2	$t_i - t_{i-1}$
0	2	−0.6667	0.5556	-3.000×10^6	---
1	1	-1.021×10^2	1.452×10^4	-1.452×10^4	0.80695
2	0	-6.669×10^5	1.000×10^6	1.000	1.323×10^{-4}
3	−1	-1.334×10^6	$j1.633 \times 10^6$	$-j1.633 \times 10^6$	9.619×10^{-7}
4	−2	-2.228×10^2	-2.974×10^2	-3.000×10^6	3.470×10^{-6}
5	−2	0.6667	0.5556	-3.000×10^6	1.117×10^{-4}

Характеристики некоторых известных тестовых задач приведены в табл. 1.3, где первые 6 тестов – нежесткие, остальные 8 – жесткие. При решении обыч-

ными явными методами все жесткие задачи требуют очень больших вычислительных затрат, а решение наиболее жесткой задачи ROBER практически невозможно получить такими методами. Неявные методы успешно и с малыми затратами решают эти задачи (результаты приведены в разделах 3.6 и 5.13). Отметим, что задача BEAM имеет чисто мнимый спектр матрицы Якоби. Поэтому формально ее можно отнести к колебательным задачам, но она проявляет свойство жесткости, поскольку эффективно может быть решена только неявными методами. Для решения задач с вещественным жестким спектром (к ним относятся тесты VDPOL, ROBER, OREGO, HIRES, CUSP и BRUSS) успешно применяют специальные явные методы, рассмотренные в главах 7 и 8.

Таблица 1.3. Характеристики тестовых задач

Задача	Источник	n	T	$M_{ж}$	$M_{к}$	$M_{ну}$	M_{Σ}
JACB	[74]	3	20	5.6	17.1	5.8	17.7
TWOB	[74]	4	20	31.0	21.9	31.0	31.0
VDPL	[74]	2	20	28.1	12.4	9.8	40.2
BRUS	[74]	2	20	205.1	3.7	20.2	227.9
LAGR	[74]	10	10	0	54.7	0	54.7
PLEY	[74, 128]	28	3	40.6	28.3	40.6	40.6
VDPOL	[75, 128]	2	2	3.84×10^{6}	4.0	35.8	3.84×10^{6}
ROBER	[75, 128]	3	10^{11}	10^{15}	0	0	10^{15}
OREGO	[75, 128]	3	360	1.13×10^{7}	1.5	27.1	1.13×10^{7}
HIRES	[75, 128]	8	321.8122	3.44×10^{4}	0.006	0	3.44×10^{4}
PLATE	[75]	80	7	6.96×10^{3}	1.02×10^{4}	0	1.08×10^{4}
BEAM	[75, 128]	80	5	0	3.2×10^{4}	0	3.2×10^{4}
CUSP	[75]	96	1.1	6.87×10^{4}	1.8	21.0	6.87×10^{4}
BRUSS	[75]	1000	10	2×10^{5}	0	0	2×10^{5}

1.5. Колебательные задачи

Колебательные задачи имеют собственные значения матрицы Якоби вблизи мнимой оси, а их решения представляют собой колебательные процессы с медленно изменяющимися амплитудой и частотой. Трудность решения таких задач обусловлена необходимостью обеспечить правильные значения амплитуды и фазы на протяжении многих периодов.

Простейшая колебательная задача имеет вид

$$x' = -\omega y, \quad y' = \omega x, \quad x_0 = 1, \quad y_0 = 0, \quad 0 \le t \le T \tag{1.16}$$

и описывает незатухающие колебания $x(t) = \cos(\omega t)$, $y(t) = \sin(\omega t)$ с амплитудой $A(t) = 1$ и фазой $\varphi(t) = \omega t$. Однако большинство используемых на практике методов дает медленно расходящееся или медленно сходящееся численное решение, фаза которого отстает или опережает фазу точного решения.

Посмотрим, как изменяются амплитуда и фаза при численном решении методом Рунге–Кутты. Представим (1.16) в виде скалярного уравнения

$u' = j\omega u, \quad u = x + jy, \quad u_0 = 1,$

где j – мнимая единица. Обозначим $H = \omega h$, тогда решение на одном шаге методом с функцией устойчивости $R(z)$ получим в виде $u_1 = R(jH)$, а в конце интервала в виде $u_N = R(jH)^N$, где $N = T/h$ – число шагов. Значения амплитуды и фазы численного решения после первого шага:

$$\tilde{A}(h) = \left|R(jH)\right|, \quad \tilde{\varphi}(h) = \arg(R(jH)).$$

Соответствующие локальные ошибки выражаются формулами

$$\delta A(h) = A(h) - \tilde{A}(h) = 1 - \left|R(jH)\right|, \quad \delta\varphi(h) = \varphi(h) - \tilde{\varphi}(h) = H - \arg(R(jH)), \tag{1.17}$$

а глобальные ошибки равны $\Delta A(T) = 1 - |R(jH)|^N$, $\Delta\varphi(T) = N\delta\varphi(T)$.

Разлагая выражения (1.17) в ряд Тейлора, получаем

$$\delta A(h) = C_A H^{q+1} + O(H^{q+3}), \quad \delta\varphi(h) = C_\varphi H^{r+1} + O(H^{r+3}),$$

где C_A и C_φ – коэффициенты ошибки по амплитуде и по фазе. Соответствующие глобальные ошибки пропорциональны H^q и H^r, где q – нечетное число, а r – четное. Значение q называют *порядком диссипативности* (*dissipation order*), а r – *порядком сдвига фазы* (*phase lag order*) [99, 147].

Для методов Рунге–Кутты не ниже 4-го порядка имеем

$$R(z) = 1 + z + z^2/2 + z^3/6 + z^4/24 + a_5 z^5 + \cdots + a_9 z^9 + O(z^{10}),$$

а выражения для локальных ошибок по амплитуде и фазе запишутся в виде

$$\delta A(h) = \left(\frac{1}{144} - a_5 + a_6\right)H^6 + \left(\frac{-1}{1152} + \frac{a_5}{6} - \frac{a_6}{2} + a_7 - a_8\right)H^8 + O(H^{10}),$$

$$\delta\varphi(h) = \left(\frac{1}{120} - a_5\right)H^5 + \left(\frac{-1}{336} + \frac{a_5}{2} - a_6 + a_7\right)H^7 + \left(\frac{1}{5184} - \frac{a_5}{24} + \frac{a_6}{6} - \frac{a_7}{2} + a_8 - a_9\right)H^9 + O(H^{11}).$$

Приведем порядки (классический p, диссипативности q и сдвига фазы r) и коэффициенты C_A и C_φ некоторых известных методов Рунге–Кутты.

Метод Ральстона:

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6}, \quad p = 3, \quad q = 3, \quad r = 4, \quad C_A = \frac{1}{24}, \quad C_\varphi = \frac{-1}{30}.$$

Классический метод Рунге–Кутты:

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}, \quad p = 4, \quad q = 5, \quad r = 4, \quad C_A = \frac{1}{144}, \quad C_\varphi = \frac{1}{120}.$$

Метод Мерсона:

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{144}, \quad p = 4, \quad q = 7, \quad r = 4, \quad C_A = \frac{1}{3456}, \quad C_\varphi = \frac{1}{720}.$$

Метод Дорманда–Принса:

$$R(z)=1+z+\frac{z^2}{2}+\frac{z^3}{6}+\frac{z^4}{24}+\frac{z^5}{120}+\frac{z^6}{600},\quad p=5,\quad q=5,\quad r=6,\quad C_A=\frac{1}{3600},\ C_\varphi=\frac{-1}{2100}.$$

Метод Лобатто IIIA 4-го порядка:

$$R(z)=\frac{1+z/2+z^2/12}{1-z/2+z^2/12},\quad p=4,\quad q=\infty,\quad r=4,\quad C_A=0,\quad C_\varphi=\frac{1}{720}.$$

Метод Радо IIA 5-го порядка:

$$R(z)=\frac{1+(2/5)z+z^2/20}{1-(3/5)z+(3/20)z^2-z^3/6},\quad p=5,\quad q=5,\quad r=6,\quad C_A=\frac{1}{7200},\quad C_\varphi=\frac{1}{42000}.$$

Метод Ральстона был предложен в [136] и вместе с вложенной формулой 2-го порядка образует метод Богацки–Шампайна с автоматическим выбором шага [86], реализованный в решателе ode23 системы MATLAB. Метод Дорманда–Принса также реализован в MATLAB в решателе ode45. Метод Мерсона реализован в одном из явных решателей SimInTech. Методы Лобатто IIIA относятся к симметричным методам [74, 75], для которых $|R(jH)| = 1$, благодаря чему они сохраняют амплитуду колебаний на любом интервале. Такие методы позволяют обеспечить правильный характер огибающей колебательного решения при моделировании высокочастотных колебательных процессов, модулированных по амплитуде. Метод Радо IIA 5-го порядка реализован в одном из наиболее эффективных решателей жестких и дифференциально-алгебраических задач RADAU5 [75]. Специальные явные методы, имеющие повышенную точность при решении колебательных задач и пригодные также и для решения жестких задач, рассмотрены в разделе 8.5.

Чтобы оценить возможности наиболее известных методов при решении колебательных задач, приведем результаты решения трех таких задач. Для их решения используем следующие явные методы: Ralston3 – метод Ральстона, Merson4 – метод Мерсона, DP5 – метод Дорманда–Принса, а также неявные методы Лобатто IIIA (Lobatto4) и Радо IIA (Radau5). Цифра в обозначении метода показывает его порядок. Чтобы вычислительные затраты всех методов были примерно одинаковы, размер шага явного метода выбираем таким, чтобы на одном периоде колебаний длиной T выполнялось 120 вычислений правой части, т. е. принимаем $h = Ts/120$, где s – число стадий, совпадающее с числом вычислений правой части на одном шаге. Наш опыт показывает, что при эффективной реализации трудоемкость выполнения одной неявной стадии в 2...3 раза больше, чем явной стадии, поэтому размер шага неявного метода принимаем в 2.5 раза больше, чем у явного метода с таким же числом стадий. Первая стадия метода Лобатто – явная и не требует вычислений, поэтому ее не учитываем.

Первая задача – простейший линейный тест

$$y_1' = y_2,\quad y_2' = -y_1,\quad y_1(0) = 0,\quad y_2(0) = 1,\quad 0 \le t \le NT, \tag{1.18}$$

где $T = 2\pi$ – период колебаний, N – число периодов на интервале интегрирования. Мера колебательности этой задачи $M_к = 2\pi N$. Ошибку численного решения вычисляем по формуле

$$error = \max\left(\sqrt{e_1^2(t) + e_2^2(t)},\ 0 \le t \le NT\right), \quad e_i(t) = y_i(t) - \tilde{y}_i(t), \quad i = 1,2, \tag{1.19}$$

где $y_1(t) = \sin(t)$, $y_2(t) = \cos(t)$ – точное решение, а $\tilde{y}_i(t)$ – численное решение. Полученные ошибки при трех значениях числа периодов N приведены в табл. 1.4. Видно, что методы более высокого порядка имеют преимущество, а ошибки решения всех методов возрастают пропорционально интервалу интегрирования. Методы Merson4 и Lobatto4 показывают близкие результаты, что вполне объяснимо, поскольку они имеют одинаковые значения C_φ при $p = r = 4$ и доминировании ошибки по фазе.

Таблица 1.4. Ошибки решения колебательной задачи (1.18)

Метод	h	Ошибка		
		$N = 1$	$N = 10$	$N = 100$
Ralston3	$T/40$	1.01×10^{-3}	1.01×10^{-2}	9.65×10^{-2}
Merson4	$T/24$	4.20×10^{-5}	4.20×10^{-4}	4.20×10^{-3}
DP5	$T/20$	5.52×10^{-6}	5.52×10^{-5}	5.52×10^{-4}
Lobatto4	$T/24$	4.08×10^{-5}	4.08×10^{-4}	4.08×10^{-3}
Radau5	$T/16$	8.09×10^{-6}	8.09×10^{-5}	8.09×10^{-4}

Вторая задача – нелинейное уравнение маятника $\alpha'' = -\sin(\alpha)$, где α – угол отклонения маятника от вертикальной оси. Обозначив $y_1 = \alpha$, $y_2 = \alpha'$, получим систему ОДУ

$$y_1' = y_2, \quad y_2' = -\sin(y_1), \quad y_1(0) = \pi/2, \quad y_2(0) = 0, \quad 0 \le t \le NT, \tag{1.20}$$

где $T = 7.416298709205$ – период колебаний. Задача имеет колебательное решение с амплитудой $\pi/2$ (y_1) и $\sqrt{2}(y_2)$, которое мало отличается от синусоиды с такими же амплитудой, периодом и фазой (разность не превышает 0.036). Эта задача не имеет аналитического решения, но известно точное решение в отдельных точках:

$$\mathbf{y}(kT) = (\pi/2, 0)^{\mathrm{T}}, \quad \mathbf{y}((k + 1/4)T) = (0, -\sqrt{2})^{\mathrm{T}}, \quad \mathbf{y}((k + 1/2)T) = (-\pi/2, 0)^{\mathrm{T}},$$
$$\mathbf{y}((k + 3/4)T) = (0, \sqrt{2})^{\mathrm{T}}, \quad k = 0, 1, 2, \ldots,$$

в которых мы и вычисляли ошибку, используя формулу (1.19).

Мера колебательности этой задачи $M_к = 4.14N$, что немного меньше, чем у задачи (1.18). Ошибки решения приведены в табл. 1.5. На этот раз ошибки всех методов, за исключением Lobatto4, пропорциональны квадрату интервала интегрирования, а метод Lobatto4 сохраняет линейную зависимость ошибки от интервала интегрирования. Аналогично ведут себя и другие симметричные методы Лобатто и Гаусса, что подтверждает преимущество симметричных методов при решении колебательных задач.

Таблица 1.5. Ошибки решения уравнения маятника (1.20)

Метод	h	Ошибка		
		$N = 1$	$N = 10$	$N = 100$
Ralston3	$T/40$	2.15×10^{-3}	1.82×10^{-1}	3.06
Merson4	$T/24$	2.39×10^{-5}	1.85×10^{-3}	2.31×10^{-1}
DP5	$T/20$	3.89×10^{-5}	2.81×10^{-3}	2.78×10^{-1}
Lobatto4	$T/24$	3.27×10^{-5}	4.22×10^{-4}	4.32×10^{-3}
Radau5	$T/16$	4.59×10^{-5}	4.08×10^{-3}	4.27×10^{-1}

Третий тест – простейшая задача двух тел, одно из которых неподвижно, а второе движется по круговой орбите. Уравнения имеют вид

$$x'' = -x/r^3, \quad y'' = -y/r^3, \quad r = \sqrt{x^2 + y^2}.$$

Примем $x_0 = y_0' = 0$, $y_0 = x_0' = 1$, тогда период обращения $T = 2\pi$ и решение $x(t) = \sin(t)$, $y(t) = \cos(t)$. На интервале $0 \le t \le NT$ задача имеет колебательность $M_к = 6.28N$ (такую же, как и первый тест), но при этом проявляет неустойчивость ($M_{ну} = M_ж = 8.89N$). Ошибки решения приведены в табл. 1.6. Видно, что и на этот раз симметричный метод Lobatto4 имеет преимущество. При решении этой задачи, как и задачи (1.20), ошибка метода Lobatto4 и других симметричных методов пропорциональна интервалу интегрирования, а остальные методы демонстрируют более быстрый (примерно квадратичный) рост ошибки. Резюмируя, можно сказать, что для эффективного решения колебательных задач на больших интервалах следует использовать методы высоких порядков, а если задача нелинейная, то преимущество имеют симметричные методы.

Таблица 1.6. Ошибки решения задачи двух тел

Метод	h	Ошибка		
		$N = 1$	$N = 10$	$N = 100$
Ralston3	$T/40$	2.12×10^{-3}	2.14×10^{-1}	2.04
Merson4	$T/24$	5.11×10^{-4}	2.78×10^{-2}	1.92
DP5	$T/20$	2.50×10^{-4}	1.19×10^{-2}	1.02
Lobatto4	$T/24$	2.69×10^{-4}	2.67×10^{-3}	2.67×10^{-2}
Radau5	$T/16$	1.65×10^{-4}	1.54×10^{-2}	1.40

1.6. Плохо обусловленные задачи

Рассмотренные выше колебательные задачи описываются в общем случае системой 2-го порядка

$$\mathbf{y}'' = \mathbf{F}(t, \mathbf{y}), \tag{1.21}$$

которую можно преобразовать в систему 1-го порядка

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}' \end{bmatrix}' = \begin{bmatrix} \mathbf{y}' \\ \mathbf{F}(t, \mathbf{y}) \end{bmatrix}, \quad \begin{bmatrix} \mathbf{y}(t_0) \\ \mathbf{y}'(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_0' \end{bmatrix}.$$

Матрица Якоби такой системы имеет вид

$$\mathbf{J}(t,\mathbf{y}) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{F}_\mathbf{y} & \mathbf{0} \end{bmatrix}, \quad \mathbf{F}_\mathbf{y} = \frac{\partial \mathbf{F}(t,\mathbf{y})}{\partial \mathbf{y}},$$

а ее собственные значения равны $\lambda_i = \pm\sqrt{\mu_i}$, где μ_i – собственные значения матрицы $\mathbf{F}_\mathbf{y}$. Таким образом, если спектр матрицы Якоби содержит собственное число с отрицательной действительной частью, то он содержит также и число с такой же положительной действительной частью. Тогда $M_{\text{ну}} = M_{\text{ж}}$, а система ОДУ является неустойчивой. Наличие собственных чисел с положительной действительной частью вносит дополнительные трудности при численном решении, поскольку малейшее отклонение от точного решения может привести к быстрому росту ошибки.

Задачи с большим значением $M_{\text{ну}}$ будем называть плохо обусловленными. В частности, многие задачи, описываемые уравнениями 2-го порядка вида (1.21) или более общего вида $\mathbf{y}'' = \mathbf{F}(t, \mathbf{y}, \mathbf{y}')$, могут быть не только колебательными, но и плохо обусловленными. Рассмотрим, например, задачу

$y''_1 = 2y_1y_2, \quad y''_2 = 30(y_2 - y_1^2) + 6y_1^2y_2,$

$y_1(0) = y_2(0) = 1, \quad y'_1(0) = -1, \quad y'_2(0) = -2, 0 \leqslant t \leqslant T$

с решением $y_1(t) = (1 + t)^{-1}$, $y_2(t) = (1 + t)^{-2}$. При любом T задача имеет $M_{\text{ну}} = M_{\text{ж}} = (5.48...5.74)T$. При умеренных требованиях к точности все решатели системы MATLAB и ПО SimInTech дают правильное решение этой задачи на интервале $0 \leq t \leq 1$. Но при $T = 10$ ни один из этих решателей не смог обеспечить правильного решения на всем интервале даже при минимально возможном допуске на ошибку. Например, при допуске на ошибку $Tol = 10^{-14}$ и $T = 1$ ошибка метода Мерсона не превысила заданного допуска. Но уже при $T = 4$ ошибка была 1.6×10^{-4}, а при $T = 7$ ошибка в конце интервала достигла 22.3.

К плохо обусловленным можно также отнести многие задачи небесной механики, трудность решения которых связана в первую очередь с наличием собственных чисел матрицы Якоби в правой полуплоскости. Одна из них – задача Аренсторфа, рассмотренная в [74]. Рассматриваются два тела с массами $1 - \mu$ и μ, участвующие в совместном движении в некоторой плоскости, и движущееся в той же плоскости третье тело пренебрежимо малой массы. Уравнения имеют вид:

$x'' = x + 2y' - \mu'(x + \mu)/D_1 - \mu(x - \mu')/D_2,$

$y'' = y - 2x' - \mu' y/D_1 - \mu y/D_2,$

$D_1 = ((x + \mu)^2 + y^2)^{3/2}, \quad D_2 = ((x - \mu')^2 + y^2)^{3/2},$

$\mu = 0.012277471, \quad \mu' = 1 - \mu,$

$x(0) = 0.994, \quad x'(0) = y(0) = 0,$

$y'(0) = -2.00158510637908252240537862224,$

$T = 17.06521656015796255889172062 49.$

Начальные условия тщательно подобраны, чтобы получить замкнутую орбиту с периодом T. Траектория решения показана на рис. 1.4. На интервале $0 \le t \le T$ задача имеет $M_{\text{ну}} = M_{\text{ж}} = 31.4$ и $M_{\text{к}} = 30.7$. Для получения приемлемого решения на таком интервале методами 4-го порядка пришлось выполнить 16 000 шагов размером $h = T/16000$. При этом ошибка в конечной точке равна 2.41×10^{-3} у метода Merson4 и 1.33×10^{-3} у метода Lobatto4. Но уже на двух периодах при таком же размере шага ошибка была 0.794 у метода Merson4 и 0.401 у метода Lobatto4. Мы видим, что ошибка возрастает значительно быстрее, чем при решении рассмотренных ранее колебательных задач, при этом симметричный метод Lobatto4 не имеет ощутимого преимущества.

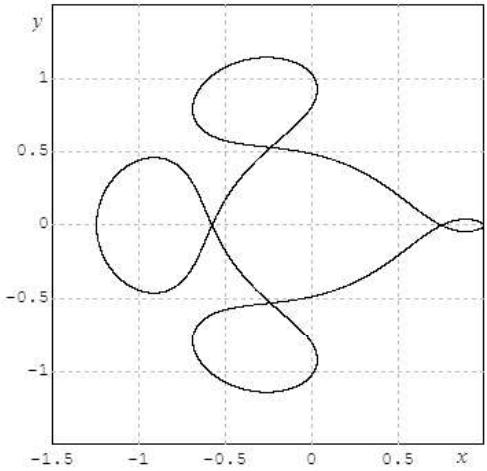


Рис. 1.4. Орбита Аренсторфа

Поскольку движение по полученной траектории крайне неравномерно, значительно более эффективно решение этой задачи с автоматическим выбором размера шага. При задаваемом допуске на ошибку $Tol = 10^{-4}$ потребовалось выполнить 197 шагов и 1005 вычислений правой части метода Мерсона для получения решения на одном периоде с ошибкой 2.10×10^{-3}. А на двух периодах потребовалось 378 шагов и 1910 вычислений правой части, но ошибка в этом случае составила уже 0.336. В [74] было показано, что для эффективного решения задачи Аренсторфа и многих подобных задач следует использовать методы высоких порядков, например метод Дорманда–Принса 8-го порядка или экстраполяционный метод переменного порядка, реализованные в решателях DOPRI8 и ODEX.

При решении плохо обусловленных задач важно убедиться в достоверности полученного результата. Для этого следует повторить расчет на более частой сетке (уменьшив размер шага или значение *Tol*) либо использовать другой метод. Если решение не изменяется, то это повышает вероятность получения действительно правильного решения. Однако существуют задачи, при реше-

нии которых разными методами и с различными значениями *Tol* будет получен один и тот же неправильный результат. Это жесткие локально-неустойчивые задачи. Одна из таких задач имеет вид:

$y_1' = y_2, \quad y_2' = \mu(1 - y_1^2)(y_1 + y_2),$
$y_1(0) = 2, \quad y_2(0) = 0, \quad 0 \le t \le 3.$

Если решать эту задачу при $\mu \ge 10^8$ и умеренных требованиях к точности одним из известных неявных решателей, то наверняка будет получено неправильное монотонно затухающее решение (точное решение – периодическое). Для получения правильного результата придется задать очень малое значение *Tol*. В главе 8 рассмотрены специальные явные методы, позволяющие эффективно решать такие задачи.

1.7. Задачи с разрывами

До сих пор мы рассматривали задачи, в которых функция $\mathbf{f}(t, \mathbf{y})$ является гладкой. Однако в практических задачах часто встречаются зависимости, содержащие разрывы (реле, люфт, гистерезис, цифровой регулятор и т. д.). При решении таких задач следует уменьшать шаг в окрестности разрыва, поэтому эффективность их решения в значительной степени определяется алгоритмом выбора размера шага.

Простейшая задача с нелинейностью релейного типа имеет вид:

$$y_1' = y_2, \quad y_2' = -2\text{sign}(y_1), \quad y_1(0) = 1, \quad y_2(0) = 0, \; 0 \leqslant t \leqslant 8. \tag{1.22}$$

Она имеет периодическое решение с периодом 4, состоящее из отрезков парабол (y_1) и прямых (y_2). При решении задачи методом Мерсона с шагом $h = 0.1$ требуется выполнить 400 вычислений правой части, при этом ошибка в конце интервала равна 0.937. При шаге $h = 0.01$ затраты возрастают в 10 раз, а ошибка равна 0.0946, т. е. уменьшается пропорционально размеру шага. Мы видим, что даже при выборе размера шага, обеспечивающего точное попадание в точки переключения ($t = 1, 3, 5, 7$), результаты оказываются неудовлетворительными. И дело тут не только в ошибках округления, поскольку для точного прохождения точки разрыва следует изменять значение y_2' не в процедуре вычисления правой части, а в отдельной процедуре, вызываемой после выполнения очередного шага непосредственно перед точкой разрыва.

Поскольку в общем случае при решении задач с разрывами невозможно заранее знать, в какие моменты модельного времени случаются разрывы, следует использовать методы с переменным размером шага. При этом можно использовать два способа:

1) применять алгоритм управления размером шага на основе получаемой на каждом шаге оценки локальной ошибки. В этом случае можно использовать обычные решатели, не внося в них никаких изменений;
2) при определении размера шага использовать не только оценку ошибки, но и прогноз ближайшего момента разрыва. Интегрирование до очеред-

ной точки разрыва выполняется обычным образом, далее управление передается программе, изменяющей некоторые переменные. После этого интегрирование возобновляется с новыми значениями переменных.

Решение задачи (1.22) с автоматическим выбором шага оказалось значительно более эффективным. При задаваемой точности $Tol = 10^{-4}$ мы получили ошибку $e = 6.85 \times 10^{-4}$ и число вычислений функции $Nf = 615$ при использовании первого способа и $e = 3.66 \times 10^{-15}$, $Nf = 96$ при использовании второго способа. Для решения задачи вторым способом была введена дискретная переменная s, которая описывает состояние реле и в начальный момент равна 1. Тогда второе уравнение в (1.22) запишется в виде $y_2' = -2s$, где переменная s изменяется (меняет знак) только после выполнения успешного шага непосредственно перед моментом разрыва.

Второй способ не только более эффективен, но и позволяет решать задачи, которые принципиально невозможно решать первым способом. Одна из таких задач – скачущий мяч, который при отскоке от пола меняет направление движения, уменьшая или сохраняя скорость. Уравнения имеют вид:

$$y' = v, \quad v' = -g, \quad \text{if } (y \le 0) \text{ and } (v < 0) \text{ then } v = -kv, \tag{1.23}$$
$$0 < k \le 1, \quad y(0) = 1, \quad v(0) = 0.$$

В отличие от (1.22), эти уравнения не могут быть описаны в подпрограмме вычисления правой части. Задачи вида (1.22) или (1.23) часто описывают в терминах событийного моделирования [41, 43, 142]. Событием называется выполнение условий, при которых происходит скачкообразное изменение переменных, параметров или даже структуры моделируемой системы. При наличии таких условий, кроме решения дифференциальных уравнений, необходимо решать две задачи: локализация события (какое событие и в какой момент должно произойти) и реализация события, т. е. выполнение вычислений, составляющих суть события. При решении уравнений (1.23) локализация события сводится к определению момента касания мяча пола (т. е. момента, когда $y = 0$ при $v < 0$), а его реализация заключается в вычислении нового значения скорости по формуле $v = -kv$. В общем случае локализация события сводится к решению нелинейного алгебраического уравнения. Мы вернемся к таким задачам в разделе 2.9.

1.8. Одношаговые методы Рунге–Кутты

Один шаг s-стадийного метода Рунге–Кутты для решения задачи Коши (1.1) задается формулами:

$$\mathbf{y}_1 = \mathbf{y}_0 + h\sum_{i=1}^{s} b_i \mathbf{F}_i, \quad \mathbf{F}_i = \mathbf{f}(t_0 + c_i h, \mathbf{Y}_i), \quad \mathbf{Y}_i = \mathbf{y}_0 + h\sum_{j=1}^{s} a_{ij}\mathbf{F}_j, \quad i = 1,\ldots,s \tag{1.24}$$

(приводим формулы первого шага, поскольку на последующих шагах используются точно такие же формулы). Коэффициенты метода можно представить в виде таблицы Бутчера:

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & \dots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} = \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^{\mathrm{T}} \end{array}.$$

Метод является явным, если $a_{ij} = 0$ при $j \geq i$, в противном случае он неявный. В случае явного метода формулы (1.24) могут быть непосредственно реализованы. Для неявного метода эти формулы задают систему нелинейных алгебраических уравнений относительно стадийных значений $\mathbf{Y}_i$, размер которой в общем случае равен произведению числа стадий s на число уравнений в системе ОДУ n. Среди неявных методов Рунге–Кутты наиболее просто реализуются диагонально-неявные (DIRK – Diagonally Implicit Runge–Kutta), у которых матрица $\mathbf{A}$ имеет нижнюю треугольную форму. В этом случае система алгебраических уравнений размера sn распадается на s последовательно решаемых систем размера n. Обычно также требуют, чтобы все ненулевые диагональные элементы матрицы $\mathbf{A}$ были равны между собой, что позволяет выполнять только одно LU-разложение матрицы $\mathbf{I} - h\gamma\mathbf{J}$ на шаге интегрирования, где γ – диагональный элемент матрицы $\mathbf{A}$, $\mathbf{J}$ – матрица Якоби системы ОДУ. Такие методы называют однократно диагонально-неявными (SDIRK – Singly DIRK).

Простейшими методами Рунге–Кутты являются методы Эйлера: явный (1.3) и неявный (1.4), которые имеют таблицы Бутчера

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad \text{и} \quad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}.$$

Классический метод Рунге–Кутты 4-го порядка имеет таблицу

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

(для явных и диагонально-неявных методов обычно опускают нулевые элементы матрицы $\mathbf{A}$).

К неявным методам Рунге-Кутты второго порядка относятся методы средней точки и трапеций с таблицами Бутчера

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array} \quad \text{и} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}.$$

Эти методы имеют одинаковую функцию устойчивости $R(z)=\dfrac{1+z/2}{1-z/2}$ и являются A-устойчивыми (но не L-устойчивыми). Метод трапеций реализован в системе MATLAB (решатель ode23t). Примером L-устойчивого метода DIRK второго порядка является метод

$$\begin{array}{c|ccc} 0 & 0 & & \\ 2\gamma & \gamma & \gamma & \\ 1 & (1-\gamma)/2 & (1-\gamma)/2 & \gamma \\ \hline & (1-\gamma)/2 & (1-\gamma)/2 & \gamma \end{array} \qquad \gamma = 1-\sqrt{2}/2.$$

Его можно интерпретировать как последовательное применение правила трапеций и формулы дифференцирования назад 2-го порядка, поэтому он получил название TR-BDF2. Этот метод реализован в системе MATLAB под названием ode23tb и в ПО SimInTech, в котором реализованы также методы DIRK третьего и четвертого порядков (DIRK3 и DIRK4).

Для уменьшения вычислительных затрат при реализации неявных методов было предложено ограничить решение алгебраических уравнений одной ньютоновской итерацией. Например, применяя одну итерацию при решении алгебраических уравнений в неявном методе Эйлера, получим метод, задаваемый формулой

$$\mathbf{y}_1 = \mathbf{y}_0 + (\mathbf{I} - h\mathbf{J})^{-1} h\mathbf{f}(t_0, y_0).$$

Такие методы получили название линейно-неявных. Применительно к диагонально-неявным методам Рунге-Кутты методы такого типа были предложены Розенброком [139]. L-устойчивый метод Розенброка второго порядка реализован в системе MATLAB под названием ode23s.

1.9. Многошаговые методы

В общем случае линейные k-шаговые методы задаются формулами вида

$$\mathbf{y}_{i+1} = \sum_{j=1}^{k} a_j \mathbf{y}_{i+1-j} + h\sum_{j=0}^{k} b_j \mathbf{f}_{i+1-j}, \quad \mathbf{f}_l = \mathbf{f}(t_l, \mathbf{y}_l). \tag{1.25}$$

Среди них наиболее известны и популярны явные и неявные методы Адамса, а также неявные методы, основанные на формулах численного дифференцирования. Неявные методы имеют $b_0 \neq 0$. Для неявных методов действует второй барьер Далквиста: если метод A-устойчив, то его порядок $p \leq 2$.

Методы Адамса имеют $a_j = 0$ при $j > 1$ и получены из условия максимального порядка при заданном k. Явные методы Адамса имеют порядок k, а неявные – порядок $k + 1$. При $k = 1$ получаем, соответственно, явный метод Эйлера и неявный метод трапеций. При $k = 2$ и постоянном размере шага формула явного ме-

тода $\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{h}{2}(3\mathbf{f}_i - \mathbf{f}_{i-1})$, а неявного метода $\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{h}{12}(5\mathbf{f}_{i+1} + 8\mathbf{f}_i - \mathbf{f}_{i-1})$. При $k \geq 2$ явные методы Адамса имеют очень ограниченные области устойчивости, поэтому самостоятельно они не применяются. Области устойчивости неявных методов Адамса при $k \geq 2$ также ограничены, что делает неэффективным их использование для решения жестких задач. В то же время сочетание явных и неявных формул Адамса позволяет построить весьма эффективные методы прогноза-коррекции. При $k = 2$ и $h = \text{const}$ формулы такого метода имеют вид:

- прогноз: $\hat{\mathbf{y}}_{i+1} = \mathbf{y}_i + \frac{h}{2}(3\mathbf{f}_i - \mathbf{f}_{i-1}), \quad \hat{\mathbf{f}}_{i+1} = \mathbf{f}(t_{i+1}, \hat{\mathbf{y}}_{i+1})$;
- коррекция: $\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{h}{12}(5\hat{\mathbf{f}}_{i+1} + 8\mathbf{f}_i - \mathbf{f}_{i-1}), \quad \mathbf{f}_{i+1} = \mathbf{f}(t_{i+1}, \mathbf{y}_{i+1})$.

Формулы Адамса удобны для реализации явных методов переменного порядка и шага, например в решателе ode113 системы MATLAB реализованы формулы порядка от 2-го до 13-го. Явные многошаговые методы, основанные на формулах Адамса, рассмотрены в разделах 8.6–8.8.

Для решения жестких задач используют методы вида (1.25), основанные на формулах дифференцирования назад (ФДН, или BDF – backward differentiation formulas), в которых производная в точке t_{i+1} аппроксимируется по значениям $\mathbf{y}_{i+1-j}$, $j = 0,\ldots, k$. ФДН порядка k имеет вид

$$\mathbf{y}_{i+1} = \sum_{j=1}^{k} a_j \mathbf{y}_{i+1-j} + hb_0 \mathbf{f}(t_{i+1}, \mathbf{y}_{i+1}).$$

При $k = 1$ получаем $b_0 = 1$, что соответствует неявному методу Эйлера, а при $k = 2$ получаем L-устойчивый метод $\mathbf{y}_{i+1} = \frac{4}{3}\mathbf{y}_i - \frac{1}{3}\mathbf{y}_{i-1} + \frac{2}{3}h\mathbf{f}(t_{i+1}, \mathbf{y}_{i+1})$. ФДН до 6-го порядка включительно являются $L(\alpha)$-устойчивыми. На их основе Ч. В. Гир (C. W. Gear) разработал метод переменного порядка и шага и реализовал его в программе DIFSUB, которая была опубликована в 1971 году [102]. Метод Гира до сих пор считается одним из самых эффективных для жестких задач. Он реализован в SimInTech, а в системе MATLAB в решателе ode15s реализован метод NDF (Numerical Differential Formulas), который практически является модификацией метода Гира.

1.10. Явные методы для жестких задач

Наряду с неявными методами для решения жестких задач успешно применяют специальные явные методы, позволяющие эффективно решать многие задачи с вещественным жестким спектром. Принципы построения таких методов рассмотрим на примере явного двухстадийного метода Рунге–Кутты 1-го порядка с функцией устойчивости $R(z) = 1 + z + dz^2$. На рис. 1.5 приведены области устойчивости такого метода при различных значениях d. При $d > 1/8$ область

устойчивости односвязная, а ее длина равна $1/d$. Значение $d = 1/4$ соответствует двум шагам метода Эйлера. Уменьшая d, можно увеличить длину области. При $d < 1/8$ область устойчивости перестает быть односвязной и состоит из двух разделенных областей, наиболее удаленная из которых позволяет обеспечить стабилизацию метода в жесткой части спектра.

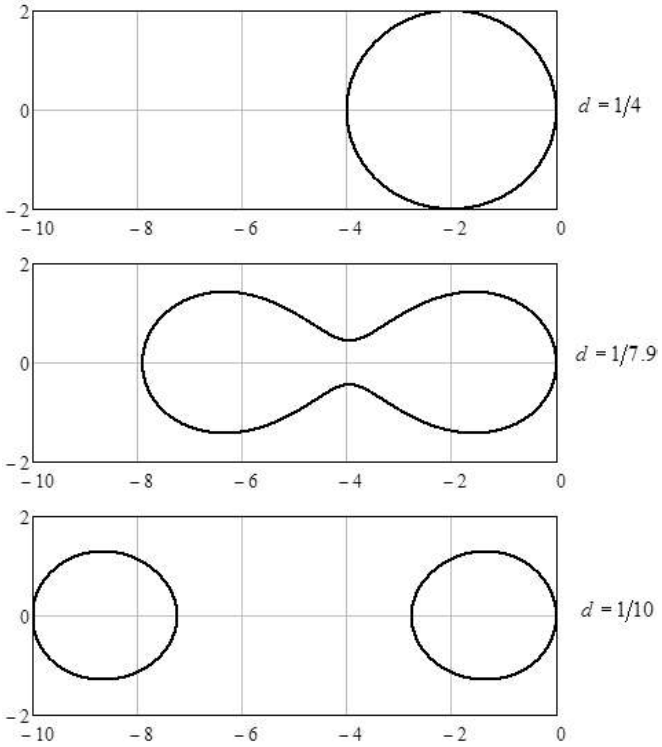


Рис. 1.5. Области устойчивости двухстадийного метода

Таким образом, можно сформулировать два способа построения явных методов для жестких задач. Первый способ основан на максимальном расширении области устойчивости. Построение многочленов устойчивости таких методов выполняется исходя из условия чебышевского альтернанса, т. е. чередования равных по модулю максимальных и минимальных значений многочлена. Методы с расширенными областями устойчивости рассмотрены в главе 7.

Идея второго способа заключается в получении на основе предварительных стадий оценок наибольших по модулю собственных значений матрицы Якоби, которые используются в заключительной формуле для стабилизации расчетной схемы в полученных точках жесткого спектра. Такие методы, названные адаптивными, рассмотрены в главе 8.

1.11. Дифференциально-алгебраические уравнения

Часто уравнения математической модели представлены не в нормальной форме Коши (1.1), а в виде системы ДАУ:

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}),\ \mathbf{y}(t_0) = \mathbf{y}_0, \tag{1.26а}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{y}, \mathbf{z}),\ \mathbf{z}(t_0) = \mathbf{z}_0. \tag{1.26б}$$

Предполагаем, что размерность векторной функции $\mathbf{g}$ совпадает с размерностью вектора $\mathbf{z}$, а начальные условия $\mathbf{y}_0$, $\mathbf{z}_0$ согласованы (для ДАУ индекса 1 это означает, что они удовлетворяют алгебраической подсистеме (1.26б)). Будем называть компоненты вектора $\mathbf{y}$ дифференциальными переменными, а вектора $\mathbf{z}$ – алгебраическими переменными.

Для численного решения уравнений (1.26) можно использовать два способа [75]. Первый из них – метод пространства состояний – основан на приведении уравнений к нормальной форме (1.1) путем численного решения алгебраической подсистемы (1.26б) при заданном векторе $\mathbf{y}$. Подставляя затем полученное значение вектора $\mathbf{z}$ в (1.26а), получаем искомые значения производных. Метод пространства состояний позволяет разделить задачи решения дифференциальных и алгебраических уравнений, поэтому его можно применять в сочетании с любым методом интегрирования. Но его нельзя использовать при решении задач высших индексов, когда алгебраическая подсистема вырождена.

Второй способ – метод ε-вложения – основан на совместном решении дифференциальной и алгебраической подсистем и может быть интерпретирован как решение сингулярно возмущенной задачи

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}),\ \mathbf{y}(t_0) = \mathbf{y}_0,$$

$$\varepsilon\mathbf{z}' = \mathbf{g}(\mathbf{y}, \mathbf{z}),\ \mathbf{z}(t_0) = \mathbf{z}_0$$

при $\varepsilon \to 0$. Применяя метод Рунге–Кутты, получим формулы одного шага интегрирования системы (1.26) в виде:

$$\mathbf{y}_1 = \mathbf{y}_0 + h\sum_{i=1}^{s} b_i \mathbf{Y}_i',\quad \mathbf{z}_1 = \mathbf{z}_0 + h\sum_{i=1}^{s} b_i \mathbf{Z}_i', \tag{1.27а}$$

$$\mathbf{Y}_i = \mathbf{y}_0 + h\sum_{j=1}^{s} a_{ij} \mathbf{Y}_j',\quad \mathbf{Z}_i = \mathbf{z}_0 + h\sum_{j=1}^{s} a_{ij} \mathbf{Z}_j', \tag{1.27б}$$

$$\mathbf{Y}'_i = \mathbf{f}(\mathbf{Y}_i, \mathbf{Z}_i),\ \ \mathbf{0} = \mathbf{g}(\mathbf{Y}_i, \mathbf{Z}_i),\ \ i = 1,\ldots, s. \tag{1.27в}$$

Формулы (1.27б, в) задают систему нелинейных алгебраических уравнений относительно векторов $\mathbf{Y}_i'$, $\mathbf{Z}_i'$, $i = 1,\ldots, s$ (векторы стадийных значений $\mathbf{Y}_i$, $\mathbf{Z}_i$ нетрудно исключить). Решая эти уравнения, находим векторы $\mathbf{Y}_i'$, $\mathbf{Z}_i'$, которые подставляем в (1.27а). Метод ε-вложения позволяет решать задачи высших индексов, но его можно использовать только в сочетании с неявным методом интегрирования, поскольку для явных методов система алгебраических уравнений (1.27б, в) будет вырожденной.

Среди неявных методов Рунге–Кутты для решения ДАУ обычно применяют жесткоточные методы, у которых последняя строка матрицы $\mathbf{A}$ совпадает с $\mathbf{b}^{\mathrm{T}}$. В этом случае нет необходимости находить векторы $\mathbf{Z}'_i$, а формулы (1.27) принимают вид:

$$\mathbf{Y}_i = \mathbf{y}_0 + h\sum_{j=1}^{s} a_{ij}\mathbf{Y}'_j, \quad \mathbf{y}_1 = \mathbf{Y}_s, \quad \mathbf{z}_1 = \mathbf{Z}_s,$$

$$\mathbf{Y}'_i = \mathbf{f}(\mathbf{Y}_i, \mathbf{Z}_i), \quad \mathbf{0} = \mathbf{g}(\mathbf{Y}_i, \mathbf{Z}_i), \quad i = 1, \ldots, s.$$

Преимущество жесткоточных методов заключается в том, что они обеспечивают точное выполнение алгебраического соотношения (1.26б). Для жесткоточных методов при решении задач индекса 1 метод ε-вложения идентичен методу пространства состояний.

Согласно определению Гира и др. (см. [75]), *индекс дифференцирования* системы (1.26) есть наименьшее число аналитических дифференцирований, требующихся для того, чтобы из уравнений (1.26) можно было бы получить систему ОДУ в форме Коши. При этом каждое дифференцирование понижает индекс на 1. Продифференцировав алгебраическую подсистему (1.26б) и обозначив $\mathbf{g}_\mathbf{y} = \partial\mathbf{g}/\partial\mathbf{y}$, $\mathbf{g}_\mathbf{z} = \partial\mathbf{g}/\partial\mathbf{z}$, получим

$$\mathbf{0} = \mathbf{g}_\mathbf{y}\mathbf{y}' + \mathbf{g}_\mathbf{z}\mathbf{z}'. \tag{1.28}$$

Если матрица $\mathbf{g}_\mathbf{z}$ обратима, то из (1.26), (1.28) можно получить систему в нормальной форме Коши:

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0,$$

$$\mathbf{z}' = -\mathbf{g}_\mathbf{z}^{-1}\mathbf{g}_\mathbf{y}\mathbf{f}(\mathbf{y}, \mathbf{z}), \; \mathbf{z}(t_0) = \mathbf{z}_0.$$

Таким образом, если матрица $\mathbf{g}_\mathbf{z}$ обратима в любой точке на траектории решения, то система (1.26) имеет индекс 1, а в противном случае (если матрица $\mathbf{g}_\mathbf{z}$ вырождена) индекс системы больше 1. Системы высших индексов (2 и выше) возникают при решении многих прикладных задач. Например, уравнения механической системы со связями, сформированные методом Лагранжа, имеют индекс 3. Такие системы наиболее трудны для численного решения и могут быть решены только неявными методами.

В качестве примера рассмотрим систему ДАУ

$$x' = u, \quad y' = v, \quad u' = -xz, \quad v' = -1 - yz, \tag{1.29а}$$

$$0 = x^2 + y^2 - 1, \tag{1.29б}$$

описывающую колебания маятника в декартовой системе координат. Продифференцировав алгебраическое уравнение (1.29б), получим

$$0 = xu + yv, \tag{1.30}$$

а продифференцировав (1.30), получим уравнение

$$0 = -z(x^2 + y^2) - y + u^2 + v^2, \tag{1.31}$$

из которого можно выразить алгебраическую переменную z через дифференциальные переменные x, y, u, v. Таким образом, система (1.29а), (1.31) имеет индекс 1, система (1.29а), (1.30) – индекс 2, а исходная система (1.29) – индекс 3.

Трудность решения систем ДАУ высших индексов обусловлена тем, что порядок сходимости численного решения оказывается ниже классического порядка метода. Например, при решении ДАУ индекса 3 методом Радо IIA 5-го порядка обеспечивается только второй порядок сходимости алгебраических переменных. Если же понизить индекс ДАУ путем аналитических дифференцирований, то возникает другая трудность – полученная система будет плохо обусловленной. Это проявляется в медленном расхождении численного решения, при котором алгебраическое соотношение (1.29б) перестает выполняться (явление сноса [75]). В случае уравнений маятника (1.29) можно преодолеть указанные трудности, перейдя к другой системе координат и сформировав уравнения относительно угла отклонения, т. е. в виде (1.20). Однако в общем случае подобное преобразование может быть практически невыполнимым. Вопросы повышения эффективности методов численного решения ДАУ высших индексов рассмотрены в главах 4, 5, 6.

Явные методы Рунге–Кутты для нежестких задач

ГЛАВА 2

2.1. Условия порядка и коэффициенты погрешности

Один шаг явного s-стадийного метода Рунге–Кутты выполняется путем последовательных вычислений по формулам:

$$\begin{gathered}\mathbf{Y}_i = \mathbf{y}_0 + h\sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j, \quad \mathbf{F}_i = \mathbf{f}(t_0 + c_i h, \mathbf{Y}_i), \quad i = 1,\ldots,s, \\ \mathbf{y}_1 = \mathbf{y}_0 + h\sum_{i=1}^{s} b_i \mathbf{F}_i.\end{gathered} \tag{2.1}$$

При реализации алгоритмов с автоматическим выбором размера шага наряду с вектором $\mathbf{y}_1$ вычисляют еще одно численное приближение

$$\hat{\mathbf{y}}_1 = \mathbf{y}_0 + h\left(\sum_{i=1}^{s} \hat{b}_i \mathbf{F}_i + \hat{b}_{s+1}\mathbf{f}(t_0 + h, \mathbf{y}_1)\right), \tag{2.2}$$

которое позволяет оценить локальную ошибку численного решения в виде нормы вектора $\mathbf{y}_1 - \hat{\mathbf{y}}_1$. Формулу (2.2) обычно называют *вложенной*, а вместе с (2.1) – вложенной парой формул (методов) Рунге–Кутты (embedded pair of Runge–Kutta formulas). Таблицу Бутчера пары (2.1), (2.2) будем представлять в виде

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline \mathbf{y}_1 & b_1 & b_2 & \cdots & b_{s-1} & b_s \\ \hat{\mathbf{y}}_1 & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_{s-1} & \hat{b}_s & \hat{b}_{s+1} \end{array} . \tag{2.3}$$

Возможны два типа вложенных пар. Пары первого типа имеют $\hat{b}_{s+1} = 0$, тогда входящие в пару методы можно представить в виде

$$\begin{array}{c|c}\mathbf{c} & \mathbf{A}\\ \hline \mathbf{y}_1 & \mathbf{b}^{\mathrm{T}}\end{array} \quad \text{и} \quad \begin{array}{c|c}\mathbf{c} & \mathbf{A}\\ \hline \hat{\mathbf{y}}_1 & \hat{\mathbf{b}}^{\mathrm{T}}\end{array}.$$

Пары второго типа имеют $\hat{b}_{s+1} \neq 0$, а входящие в них методы имеют вид:

$$\begin{array}{c|c}\mathbf{c} & \mathbf{A}\\ \hline \mathbf{y}_1 & \mathbf{b}^{\mathrm{T}}\end{array} \quad \text{и} \quad \begin{array}{c|c}\hat{\mathbf{c}} & \hat{\mathbf{A}}\\ \hline \hat{\mathbf{y}}_1 & \hat{\mathbf{b}}^{\mathrm{T}}\end{array}, \quad \hat{\mathbf{c}} = \begin{bmatrix}\mathbf{c}\\ 1\end{bmatrix}, \quad \hat{\mathbf{A}} = \begin{bmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{b}^{\mathrm{T}} & 0\end{bmatrix}.$$

Формально метод (2.3) второго типа является (s + 1)-стадийным, но по вычислительным затратам он эквивалентен s-стадийному методу, поскольку значение $\mathbf{f}(t_0 + h, \mathbf{y}_1)$, вычисленное на последней стадии, используется на первой стадии следующего шага. Поэтому вложенные пары второго типа называют FSAL (First Stage As Last), а первого типа – non-FSAL. Отметим, однако, что если после очередного шага осуществляется обработка событий, т. е. пересчет некоторых параметров или переменных, то в этом случае FSAL-методы также требуют вычисления правой части на первой стадии следующего шага. Обозначим порядок основной формулы через p, а вложенной формулы – через $\hat{p}$. Обычно вложенные методы имеют $\hat{p} = p - 1$. Ниже покажем, что такой выбор наиболее рационален.

Определение порядка аппроксимации и коэффициентов погрешности методов Рунге–Кутты сводится к сравнению рядов Тейлора для точного и численного решений (вывод условий порядка изложен в [74]). Для наглядного представления получаемых при разложении в ряд элементарных дифференциалов используют корневые деревья, при этом существует взаимно-однозначное соответствие между множеством элементарных дифференциалов и множеством деревьев. Для упрощения вывода неавтономную систему $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ преобразуют к автономной форме

$$\begin{bmatrix}t\\ \mathbf{y}\end{bmatrix}' = \begin{bmatrix}1\\ \mathbf{f}(t, \mathbf{y})\end{bmatrix}.$$

Чтобы численные решения этих двух систем совпадали, следует принять $\mathbf{c} = \mathbf{A}\mathbf{e}$, $\mathbf{e} = (1,\ldots,1)^{\mathrm{T}}$. Такое предположение позволяет упростить запись условий порядка, а вместе с условием $\mathbf{b}^{\mathrm{T}}\mathbf{e} = 1$ обеспечивает *первый стадийный порядок*. Это означает, что стадийные значения $\mathbf{Y}_i$, а также $\mathbf{y}_1$ имеют порядок не ниже первого. Все применяемые на практике методы Рунге–Кутты имеют первый стадийный порядок.

Условия, обеспечивающие порядок p метода Рунге–Кутты, запишутся в виде

$$\gamma(T_{ij})\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(T_{ij}) = 1, \quad i = 1,\ldots, p, \ j = 1,\ldots, N_i, \tag{2.4}$$

где T_{ij} – корневое дерево порядка i с порядковым номером j, N_i – число деревьев порядка i. Порядок дерева равен числу его вершин. Вывод величин $\gamma(T_{ij})$ и $\boldsymbol{\Phi}(T_{ij})$ изложен в [74]. В табл. 2.1 приведены значения этих величин для деревьев до 5-го порядка включительно. В выражениях для $\boldsymbol{\Phi}(T_{ij})$ предполагается поком-

понентное выполнение операций с векторами. *Коэффициенты погрешности* являются невязками условий порядка и имеют вид

$$e(T_{ij}) = 1 - \gamma(T_{ij})\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(T_{ij}), \quad i = 1, 2,..., \quad j = 1,..., N_i .$$

При построении метода порядка p его коэффициенты выбирают из условий (2.4), а оставшиеся свободные коэффициенты обычно используют для минимизации коэффициентов погрешности $e(T_{p+1,j})$, $j = 1,..., N_{p+1}$. Отметим, что коэффициент погрешности $e(T_{p+1,1})$ определяет точность квадратурной формулы, т. е. точность решения уравнения $y' = f(t)$, а коэффициент $e(T_{p+1,\,j})$ при $j = N_{p+1}$ определяет точность решения линейной автономной задачи $\mathbf{y}' = \mathbf{J}\mathbf{y}$.

Таблица 2.1. Деревья и значения $\gamma(T_{ij})$ и $\Phi(T_{ij})$

i	T_{ij}	Граф	$\gamma(T_{ij})$	$\Phi(T_{ij})$
1	T_1		1	$\mathbf{e}$
2	T_{21}		2	$\mathbf{c}$
3	T_{31}		3	$\mathbf{c}^2$
	T_{32}		6	$\mathbf{Ac}$
4	T_{41}		4	$\mathbf{c}^3$
	T_{42}		8	$\mathbf{c}(\mathbf{Ac})$
	T_{43}		12	$\mathbf{Ac}^2$
	T_{44}		24	$\mathbf{A}^2\mathbf{c}$
5	T_{51}		5	$\mathbf{c}^4$
	T_{52}		10	$\mathbf{c}^2(\mathbf{Ac})$
	T_{53}		15	$\mathbf{c}(\mathbf{Ac}^2)$
	T_{54}		30	$\mathbf{c}(\mathbf{A}^2\mathbf{c})$
	T_{55}		20	$(\mathbf{Ac})^2$
	T_{56}		20	$\mathbf{Ac}^3$
	T_{57}		40	$\mathbf{A}(\mathbf{c}(\mathbf{Ac}))$

Окончание табл. 2.1

i	T_{ij}	Граф	$\gamma(T_{ij})$	$\Phi(T_{ij})$
5	T_{58}		60	$\mathbf{A}^2\mathbf{c}^2$
	T_{59}		120	$\mathbf{A}^3\mathbf{c}$

2.2. Требования к параметрам методов

Выбор параметров методов Рунге–Кутты производят исходя из условий обеспечения заданного порядка и необходимых свойств устойчивости. Ряд параметров, называемых свободными, остается при этом незадействованным. Обычно подбором этих параметров стремятся минимизировать коэффициенты погрешности. Среди методов, построенных по этому принципу, – методы Богацки–Шампайна [86] и Дорманда–Принса [98]. Существуют и другие подходы. По ряду причин желательно, чтобы среди коэффициентов метода не было больших по модулю чисел. В [19] было предложено более строгое требование *интерполяционности*: все коэффициенты должны принадлежать интервалу [0, 1]. Покажем, что это требование можно получить из условия минимизации максимального отклонения численного решения от истинного решения на всех стадиях при заданных двухсторонних ограничениях на значения правой части.

Согласно [19], условие интерполяционности формулируется в виде системы неравенств

$$0 \le c_i \le 1, \tag{2.5а}$$

$$0 \le b_i \le 1, \tag{2.5б}$$

$$0 \le a_{ij} \le 1,\ i, j = 1,\ldots, s. \tag{2.5в}$$

Отметим, что при выполнении условия 1-го стадийного порядка вместо (2.5б, в) достаточно потребовать неотрицательности коэффициентов b_i, a_{ij}. Ограничения (2.5а) наиболее очевидны: они означают, что значения независимой переменной на всех стадиях не должны выходить за пределы интервала $[t_0, t_0 + h]$. Как правило, в известных методах это условие выполняется. Ограничения (2.5б, в) также имеют определенный смысл, но они редко выполняются в методах порядка 4 и выше.

Рассмотрим более подробно, что означают ограничения на параметры b_i, a_{ij}. Для этого предположим, что дифференциальное уравнение скалярное и что функция $f(t, y)$ удовлетворяет неравенству

$$\alpha \le f(t, y) \le \beta. \tag{2.6}$$

Предположим сначала, что ограничение (2.6) выполняется в некоторой достаточно большой области изменения переменных t, y, а затем уточним размеры этой области. Вычисленное согласно (2.1) значение y_1 будет максимально

возможным, если $F_i = \beta$ при $b_i > 0$ и $F_i = \alpha$ при $b_i < 0$. Значение y_1 будет минимально возможным, если $F_i = \alpha$ при $b_i > 0$ и $F_i = \beta$ при $b_i < 0$. Пусть сумма всех отрицательных b_i равна $-\bar{b}$, тогда сумма всех положительных b_i равна $1 + \bar{b}$. При выполнении (2.6) интервал неопределенности, которому может принадлежать значение y_1, задается неравенством

$$y_0 + h(\alpha - \bar{b}(\beta - \alpha)) \leqslant y_1 \leqslant y_0 + h(\beta + \bar{b}(\beta - \alpha)),$$

а его размер равен $h(1 + 2\bar{b})(\beta - \alpha)$. Размер интервала будет минимальным при $\bar{b} = 0$, т. е. при выполнении условия (2.5б).

Пусть теперь сумма отрицательных коэффициентов среди a_{ij}, $j = 1,\ldots, s$ равна $-\bar{a}_i$. Тогда интервалы неопределенности стадийных значений Y_i задаются неравенствами

$$y_0 + h(c_i\alpha - \bar{a}_i(\beta - \alpha)) \leqslant Y_i \leqslant y_0 + h(c_i\beta + \bar{a}_i(\beta - \alpha)),$$

а их размеры равны $h(c_i + 2\bar{a}_i)(\beta - \alpha)$ и будут минимальными, если все $\bar{a}_i$ равны 0, т. е. при выполнении условия (2.5в). При выполнении условия интерполяционности интервалы неопределенности численного решения на всех стадиях совпадают с интервалами неопределенности точного решения и задаются неравенствами

$$y_0 + h\alpha \leq y_1 \leq y_0 + h\beta; \quad y_0 + h\alpha c_i \leq Y_i \leq y_0 + h\beta c_i, \quad i = 1,\ldots, s.$$

Теперь можно сформулировать следующее утверждение. Пусть для всех значений t, y, удовлетворяющих неравенствам

$$t_0 \leq t \leq t_0 + h, \quad \min(y_0, y_0 + h\alpha) \leq y \leq \max(y_0, y_0 + h\beta), \tag{2.7}$$

выполняется ограничение (2.6). Тогда если метод Рунге–Кутты обладает свойством интерполяционности, то для численного решения на одном шаге гарантируется выполнение неравенства

$$y_0 + h\alpha \leq y_1 \leq y_0 + h\beta. \tag{2.8}$$

Если же метод не обладает этим свойством, то можно построить такую функцию $f(t, y)$, что при выполнении в области (2.7) ограничения (2.6) неравенство (2.8) не выполняется. Приведенное утверждение распространяется на векторный случай, тогда все неравенства следует рассматривать покомпонентно. Например, неравенство (2.6) для i-й компоненты запишется в виде $\alpha_i \leq f_i(t, \mathbf{y}) \leq \beta_i$. Таким образом, свойство интерполяционности гарантирует ограниченность в определенных пределах численного решения при ограниченности правой части.

Нарушение интерполяционности i-й стадии можно оценить величиной

$$d_i = \max(\bar{a}_i, \bar{a}_i - c_i, \bar{a}_i + c_i - 1),$$

а нарушение интерполяционности метода Рунге–Кутты – величиной

$$D = \max(\bar{b}, d_1, d_2, \ldots, d_s). \tag{2.9}$$

Для методов, обладающих свойством интерполяционности, $D = 0$, а в противном случае $D > 0$. Многие методы удовлетворяют условию (2.5а), тогда

$$D = \max(\bar{b}, \bar{a}_1, \bar{a}_2, ..., \bar{a}_s).$$

Методы, обладающие интерполяционностью, рассматривались в [2, 3, 57]. Среди известных методов это метод Ральстона 3-го порядка [136] и классический метод Рунге–Кутты 4-го порядка. Метод Ральстона имеет также и малые значения коэффициентов погрешности. Методы более высоких порядков, сочетающие интерполяционность и высокую точность, построить весьма сложно. Например, метод Дорманда–Принса имеет малые коэффициенты погрешности, но интерполяционность сильно нарушена ($D = 11.89$). Поэтому при построении методов следует определять свободные параметры путем минимизации коэффициентов погрешности при ограничении сверху на значение D. Во многих работах (например, в [134]) нарушение интерполяционности пары Рунге–Кутты оценивается величиной максимального коэффициента $D_\infty = \max\left(|a_{ij}|, |b_i|, |\hat{b}_i|, |c_i|\right)$.

Определенные требования предъявляются и к вложенной формуле, коэффициенты которой также не должны сильно выходить за пределы интервала [0, 1]. Чтобы оценка ошибки была пропорциональна глобальной ошибке, порядок вложенного метода должен быть на 1 меньше порядка основного метода. Нежелательно, чтобы среди коэффициентов погрешности порядка $\hat{p} + 1$ вложенного метода были нулевые или близкие к нулю, поскольку это ослабляет контроль некоторых компонент ошибки. Желательно, чтобы это были числа примерно одного порядка.

2.3. Управление размером шага

На очередном шаге мы получаем оценку локальной ошибки $\mathbf{y}_1 - \hat{\mathbf{y}}_1$. Если эта оценка заметно недооценивает либо переоценивает ошибку, то следует использовать оценку в виде $K(\mathbf{y}_1 - \hat{\mathbf{y}}_1)$, где параметр K настраивают так, чтобы реальная ошибка для определенного круга задач примерно соответствовала задаваемому допуску. В этом случае во вложенной формуле (2.2) вместо ранее выбранного $\hat{\mathbf{b}}$ следует задать $K\hat{\mathbf{b}} + (1 - K)\mathbf{b}$. Для получения оценки ошибки обычно используют вложенные методы, но можно использовать и экстраполяцию по Ричардсону [74], тогда оценку ошибки получаем в виде разности двух решений, полученных в результате выполнения одного шага размера h и двух шагов размера $h/2$.

Размер следующего шага следует принять таким, чтобы ошибка не превышала заданного допустимого значения. Обычно задают допуск на относительную ошибку *Rtol* и на абсолютную ошибку *Atol*. Ненулевое значение *Atol* позволяет предотвратить неоправданное уменьшение размера шага, когда одна из переменных изменяет знак. Поскольку переменные могут иметь разный масштаб,

правильно было бы задавать значение *Atol* для каждой из переменных. Однако для сложных систем это приводит к дополнительным трудностям, поэтому обычно задают одинаковое и достаточно малое значение *Atol* для всех переменных.

Для каждой переменной вычисляем масштабный коэффициент

$$d_i = Atol + Rtol \times \max(|y_{0i}|, |y_{1i}|), \tag{2.10}$$

а нормированную оценку ошибки принимаем в виде

$err = \max_i(|y_{1i} - \hat{y}_{1i}|/d_i)$.

Возможно применение и других норм, например

$$err = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_{1i} - \hat{y}_{1i}}{d_i}\right)^2}.$$

Шаг считается успешным, если $err \le 1$. При малых h зависимость оценки ошибки от h имеет вид $err = Ch^{\min(p,\hat{p})+1}$. Чтобы оценка ошибки на следующем шаге была допустимой, следует задать размер следующего шага $h_{new} = wh$, где $w = fac \times err^{-\alpha}$, $\alpha = 1/(\min(p, \hat{p}) + 1)$, $fac = 0.5...0.9$. Множитель *fac* можно задать в виде $fac = \overline{err}^{\alpha}$, где $\overline{err} = 0.25...0.4$ – ожидаемое значение *err* на следующем шаге (в используемых ниже решателях задаем $fac = 0.3^{\alpha}$). Следует также ограничить максимальное и минимальное значения w, задав, например, $w_{max} = 4$, $w_{min} = 1/4$. В результате получаем формулу для расчета размера шага в виде

$h_{new} = wh$, $w = \min(w_{max}, \max(w_{min}, fac \times err^{-\alpha}))$.

Если на очередном шаге $err > 1$, то шаг отбрасывается и вычисления повторяются с новым размером шага h_{new}.

Мы привели алгоритм управления размером шага на основе оценки локальной ошибки, который принято считать стандартным. Несмотря на свою простоту, он достаточно эффективен и используется во многих решателях. Более продвинутые алгоритмы, например построенные с использованием методов теории автоматического управления, приведены в [75, 144]. Все эти алгоритмы контролируют локальную ошибку и не гарантируют, что глобальная ошибка также будет удовлетворять заданному допуску. Поэтому применяемые на практике решатели с автоматическим выбором размера шага не всегда обеспечивают достоверность полученного решения. И если есть сомнения в полученном результате, то следует произвести контрольный расчет с более высокой точностью либо использовать другой решатель.

Наиболее надежный способ получения оценки глобальной ошибки основан на глобальной экстраполяции по Ричардсону. Его идея состоит в том, что расчет производится на сетке, число узлов которой последовательно удваивается. Это позволяет получить зависимость оценки ошибки от шага сетки и выбрать шаг, при котором достигается требуемая точность. Такой подход развивается в работах Н. Н. Калиткина и его учеников [17, 18]. Если решение наряду с гладкими участками содержит участки быстрого изменения переменных, то такая

стратегия выбора шага сопряжена с очень большими вычислительными затратами. В подобных случаях следует применять неравномерную сетку, но заранее невозможно определить, на каких участках и насколько нужно сгущать сетку. Справиться с этой ситуацией мог бы метод, автоматически генерирующий подходящие неравномерные сетки в процессе расчета. В [21] предложено выполнять интегрирование не по независимой переменной t, а по преобразованной переменной (длине дуги). При этом равномерная сетка по длине дуги преобразуется в неравномерную сетку по t, причем шаг сетки тем меньше, чем быстрее изменяется решение. Среди других работ по алгоритмам управления размером шага с использованием оценок глобальной ошибки отметим работы Г. Ю. Куликова и Р. Вайнера [9, 122, 123].

2.4. Методы 1-го и 2-го порядков

Будем обозначать пары методов через ИМЯ$p(\hat{p})$ для non-FSAL-методов и ИМЯ$p(\hat{p})$F для FSAL-методов. Две простейшие пары имеют вид:

$$\text{RK1(2)F:}\quad \begin{array}{c|cc} 0 & & \\ \hline \mathbf{y}_1 & 1 & \\ \hat{\mathbf{y}}_1 & 1/2 & 1/2 \end{array} \qquad \text{RK2(1):}\quad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline \mathbf{y}_1 & 1/2 & 1/2 \\ \hat{\mathbf{y}}_1 & 1 & 0 \end{array} \tag{2.11}$$

и отличаются между собой перестановкой $\mathbf{y}_1$ и $\hat{\mathbf{y}}_1$.

Используем эти методы для решения задачи

$$y_1' = y_2 - y_1(y_1^2 + y_2^2 - 1),\ y_2' = -y_1 - y_2(y_1^2 + y_2^2 - 1),\\ y_1(0) = 0,\ y_2(0) = 1,\ 0 \leqslant t \leqslant 2\pi \tag{2.12}$$

с точным решением $y_1(t) = \sin t$, $y_2(t) = \cos t$. Результаты (ошибка и число вычислений функции Nf) при различных значениях допуска на ошибку $Tol = Atol = Rtol$ приведены в табл. 2.2. В первую очередь нас интересует зависимость реальной ошибки от заданного допуска Tol. Очевидно, что такая зависимость должна быть прямо пропорциональной, т. е. при уменьшении Tol в 10 раз реальная ошибка также должна уменьшаться в 10 раз. Ошибка метода RK2(1) действительно обладает этим свойством, но ошибка метода RK1(2)F пропорциональна $Tol^{1/2}$. Посмотрим, почему так происходит. Ошибка e, оценка ошибки err и размер шага h связаны отношениями пропорциональности $e \propto h^p$ и $err \propto h^{\min(p,\hat{p})+1}$, откуда $e \propto err^{1/\beta}$, $\beta = \dfrac{\min(p,\hat{p})+1}{p}$. А поскольку управление размером шага производится по оценке локальной ошибки err, то имеем $e \propto Tol^{1/\beta}$. Ошибка будет пропорциональна задаваемому допуску, только если $\beta = 1$, откуда $\hat{p} = p - 1$. Обычно вложенные формулы имеют именно такое значение $\hat{p}$. Если же получить вложенную формулу такого порядка не удается, то можно воспользоваться следующим приемом: вместо $Rtol$ и $Atol$ использовать в формуле (2.10) величины $Rtol' = Rtol^{\beta}$ и $Atol' = Atol \times Rtol^{\beta-1}$. Для метода RK1(2)F в табл. 2.2

это соответствует замене *Tol* в виде 10^{-k} на $10^{-k/2}$. Заметим, однако, что методы 1-го порядка применяют очень редко вследствие низкой точности.

Таблица 2.2. Результаты решения задачи (2.12)

Tol	RK1(2)F		RK2(1)	
	Ошибка	***Nf***	**Ошибка**	***Nf***
10^{-2}	2.98×10^{-2}	59	1.22×10^{-2}	122
10^{-3}	9.09×10^{-3}	181	1.32×10^{-3}	364
10^{-4}	2.87×10^{-3}	562	1.35×10^{-4}	1128
10^{-5}	9.08×10^{-4}	1769	1.36×10^{-5}	3540
10^{-6}	2.87×10^{-4}	5581	1.37×10^{-6}	11164

Двухстадийные методы 2-го порядка образуют однопараметрическое семейство с таблицей Бутчера

$$\begin{array}{c|cc} 0 & & \\ c_2 & c_2 & \\ \hline & 1-\dfrac{1}{2c_2} & \dfrac{1}{2c_2} \end{array}.$$

Коэффициенты погрешности этих методов: $e(T_{31}) = 1 - 3c_2/2$, $e(T_{32}) = 1$. Метод RK2(1) имеет $e(T_{31}) = -1/2$ и не является оптимальным по точности. Оптимальный метод получим, задав $c_2 = 2/3$, тогда $e(T_{31}) = 0$. Соответствующая non-FSAL-пара имеет таблицу Бутчера

$$\begin{array}{c|cc} 0 & & \\ 2/3 & 2/3 & \\ \hline \mathbf{y}_1 & 1/4 & 3/4 \\ \hat{\mathbf{y}}_1 & 1 & 0 \end{array}.$$

Обозначим этот метод RK2(1)b и сравним его с RK2(1).

При решении гладких задач метод RK2(1)b действительно немного более точен. Но если решение содержит участки быстрого изменения некоторых переменных или правая часть содержит разрывы, то более точным оказывается метод RK2(1). Рассмотрим, например, задачу

$$y_1' = y_2,\ y_2' = -2\mathrm{sign}(y_1 - 1),\quad y_1(0) = 2,\quad y_2(0) = 0,\quad 0 \le t \le 4. \tag{2.13}$$

Она имеет периодическое решение с периодом 4, состоящее из отрезков парабол (y_1) и прямых (y_2). Кривые решения приведены на рис. 2.1. Методы порядка 2 и выше точно проходят гладкие участки, поэтому ошибка определяется только выбором размера шага вблизи разрыва. При допуске на ошибку $Tol = 10^{-4}$ метод RK2(1) имеет ошибку в конце интервала $e = 9.08\times10^{-5}$ и число вычислений функции $Nf = 909$, а метод RK2(1)b имеет $e = 2.02\times10^{-2}$ и $Nf = 861$. Графики изменения размера шага приведены на рис. 2.2 и показывают, что

метод RK2(1)b недостаточно точно локализует точку разрыва или даже совсем ее пропускает. Это объясняется тем, что если разрыв случается внутри интервала $[t_n + c_2h, t_n + h]$, то на оценке ошибки это никак не сказывается. Чтобы эффективно решать такие задачи, следует использовать методы, имеющие одну из абсцисс $c_i = 1$ (обычно это последняя абсцисса c_s), либо FSAL-методы. Далее будем рассматривать только такие методы.

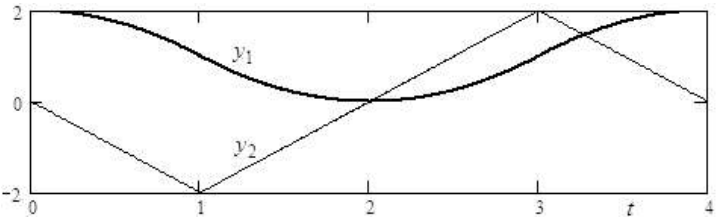


Рис. 2.1. Решение задачи (2.13)

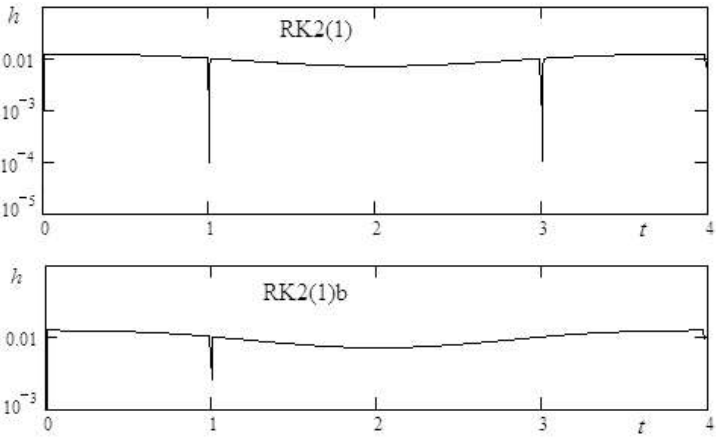


Рис. 2.2. Изменение размера шага при решении задачи (2.13)

Построим теперь оптимальную FSAL-пару, приняв $c_2 = 2/3$. Как и в паре RK2(1), пусть коэффициент ошибки вложенного метода $e(T_{21}) = 1$, тогда коэффициенты вложенной формулы находим из уравнений $\hat{b}_1 + \hat{b}_2 + \hat{b}_3 = 1$, $1 - 2(\hat{b}_2 c_2 + \hat{b}_3) = 1$, откуда

$$\hat{b}_1 = 1 + \hat{b}_3/2, \quad \hat{b}_2 = -3\hat{b}_3/2.$$

Свободный параметр $\hat{b}_3$ определим из условия минимизации нормы $\|\mathbf{b} - \hat{\mathbf{b}}\|_\infty$, тогда $\hat{b}_3 = -1/2$ и искомая пара RK2(1)F имеет вид

$$\begin{array}{c|ccc} 0 & & & \\ 2/3 & 2/3 & & \\ \hline \mathbf{y}_1 & 1/4 & 3/4 & \\ \hat{\mathbf{y}}_1 & 3/4 & 3/4 & -1/2 \end{array}. \qquad (2.14)$$

Для тестового сравнения из рассмотренных пар оставим RK2(1) (2.11) и RK2(1)F (2.14). Подобным образом выберем по одному представителю non-FSAL- и FSAL-пар порядков 3(2), 4(3) и 5(4), а затем сравним их эффективность при решении тестовых задач.

2.5. Методы 3-го порядка

Рассмотрим трехстадийные методы с таблицей Бутчера

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & c_2 & & \\ c_3 & c_3 - a_{32} & a_{32} & \\ \hline & 1 - b_2 - b_3 & b_2 & b_3 \end{array}.$$

Коэффициенты методов 3-го порядка находим из соотношений:

$b_2c_2 + b_3c_3 = 1/2$, $b_2c_2^2 + b_3c_3^2 = 1/3$, $b_3a_{32}c_2 = 1/6$.

Такие методы образуют два семейства: двухпараметрическое семейство с коэффициентами

$$a_{32} = \frac{c_3(c_3 - c_2)}{c_2(2 - 3c_2)}, \quad b_2 = \frac{3c_3 - 2}{6c_2(c_3 - c_2)}, \quad b_3 = \frac{3c_2 - 2}{6c_3(c_2 - c_3)}$$

и свободными параметрами c_2, c_3 и однопараметрическое семейство с коэффициентами

$$c_2 = c_3 = 2/3, \quad b_2 = \frac{3}{4} - \frac{1}{4a_{32}}, \quad b_3 = \frac{1}{4a_{32}}$$

и свободным параметром a_{32}.

Коэффициенты погрешности трехстадийных методов 3-го порядка находим по формулам:

$$e(T_{41}) = 1 + 2c_2c_3 - \frac{4}{3}(c_2 + c_3), \quad e(T_{42}) = 1 - \frac{4}{3}c_3, \quad e(T_{43}) = 1 - 2c_2, \quad e(T_{44}) = 1.$$

Чтобы получить надежную оценку ошибки, для non-FSAL-метода зададим c_3 =1, тогда $e(T_{42})$ = −1/3, а при c_2 =1/2 получим $e(T_{41})$ = $e(T_{43})$ = 0. Таким образом, значения c_2 =1/2, c_3 =1 являются оптимальными для non-FSAL-метода. При таких параметрах коэффициенты вложенной формулы можно задать в виде $\hat{\mathbf{b}} = (\hat{b}_1, 1 - 2\hat{b}_1, \hat{b}_1)^{\mathrm{T}}$. Выбрав $\hat{b}_1 = 1/4$, получим пару RK3(2) в виде

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline \mathbf{y}_1 & 1/6 & 2/3 & 1/6 \\ \hat{\mathbf{y}}_1 & 1/4 & 1/2 & 1/4 \end{array}. \qquad (2.15)$$

Коэффициенты погрешности 3-го порядка вложенной формулы этой пары: $e(T_{31}) = -1/8$, $e(T_{32}) = -1/2$.

Найдем теперь оптимальные параметры FSAL-метода путем минимизации коэффициентов погрешности. Если минимизировать значение $|e(T_{41})| + |e(T_{42})| + |e(T_{43})|$, то получим $c_2 = 1/2$, $c_3 = 3/4$, $e(T_{41}) = 1/12$, $e(T_{42}) = e(T_{43}) = 0$. Метод с такими параметрами был предложен в [136] и считается оптимальным среди трехстадийных методов. К тому же он удобен для реализации. Соответствующая вложенная пара предложена в [86] и известна как метод Богацки–Шампайна, который реализован в решателе ode23 системы MATLAB. Обозначим этот метод через BS3(2)F, его таблица коэффициентов имеет вид:

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 3/4 & 0 & 3/4 & & \\ \hline \mathbf{y}_1 & 2/9 & 1/3 & 4/9 & \\ \hat{\mathbf{y}}_1 & 7/24 & 1/4 & 1/3 & 1/8 \end{array}. \qquad (2.16)$$

Все коэффициенты неотрицательны, т. е. как основной, так и вложенный методы обладают интерполяционностью. Кроме этого, коэффициенты погрешности вложенного метода равны между собой: $e(T_{31}) = e(T_{32}) = -1/8$, – что также является достоинством этой пары.

2.6. Методы 4-го порядка

Вывод четырехстадийных методов 4-го порядка подробно изложен в [74]. При условии что все узлы различные, эти методы образуют двухпараметрическое семейство со свободными коэффициентами c_2 и c_3. Остальные коэффициенты находим по формулам:

$$b_2 = \frac{2c_3 - 1}{12c_2(1-c_2)(c_3-c_2)}, \quad b_3 = \frac{2c_2 - 1}{12c_3(1-c_3)(c_2-c_3)}, \quad b_4 = \frac{6c_2c_3 - 4(c_2+c_3)+3}{12(1-c_2)(1-c_3)},$$

$$c_4 = 1, \quad a_{32} = \frac{c_3(c_3-c_2)}{2c_2(1-2c_2)}, \quad a_{42} = \frac{b_2(1-c_2)-b_3a_{32}}{b_4}, \quad a_{43} = \frac{b_3(1-c_3)}{b_4},$$

$$a_{21} = c_2, \quad a_{31} = c_3 - a_{32}, \quad a_{41} = 1 - a_{42} - a_{43}, \quad b_1 = 1 - b_2 - b_3 - b_4.$$

Коэффициенты погрешности такого метода:

$$e(T_{51}) = -\frac{1}{4} + \frac{5}{12}(c_2 + c_3) - \frac{5}{6}c_2c_3, \quad e(T_{52}) = \frac{5}{12}c_3 - \frac{1}{4}, \quad e(T_{53}) = \frac{5}{8}c_2 - \frac{1}{4},$$

$$e(T_{54}) = -\frac{1}{4}, \quad e(T_{55}) = 1 - \frac{5(b_4 + b_3(3-4c_3)^2)}{144b_3b_4(1-c_3)^2}, \quad e(T_{56}) = -4e(T_{51}),$$

$$e(T_{57}) = -4e(T_{52}), \quad e(T_{58}) = -4e(T_{53}), \quad e(T_{59}) = 1.$$

В [74, упражнение II.3.1] приведены значения коэффициентов c_2 и c_3, полученные путем минимизации различных норм коэффициентов погрешности. При минимизации величины $q = \sum_{i=1}^{9} |e(T_{5i})|$ получаем c_2 = 0.3995, c_3 = 0.6, q = 1.553. Практически такое же значение q (отличие в 6-м знаке), но более удобные коэффициенты метода получаем, задав c_2 = 0.4, c_3 = 0.6 (такой метод предложен в [136], он имеет $e(T_{52}) = e(T_{53}) = e(T_{57}) = e(T_{58}) = 0$). Заметим, что для классического метода Рунге–Кутты q = 2.229.

Для четырехстадийных методов невозможно построить non-FSAL-пару порядков 4(3). Поэтому построим FSAL-пару на основе метода, имеющего $c_2 = 0.4$, $c_3 = 0.6$. Вложенная формула имеет 5 коэффициентов, а чтобы был 3-й порядок, достаточно четырех из них, поэтому задаем $\hat{b}_4 = 0$, а остальные коэффициенты находим из условий порядка. Полученная пара имеет вид

0					
2/5	2/5				
3/5	−3/20	3/4			
1	19/44	−15/44	10/11		
$\mathbf{y}_1$	11/72	25/72	25/72	11/72	
$\hat{\mathbf{y}}_1$	5/36	5/12	5/18	0	1/6

Добавление в основной метод пятой стадии увеличивает число свободных параметров, что дает больше возможности для минимизации коэффициентов погрешности. Рассмотрим четырехпараметрическое семейство методов, имеющих $b_2 = 0$ и 2-й порядок всех стадий, за исключением второй. Такое предположение является обычным для методов высоких порядков, упрощая их построение. Коэффициенты таких методов удовлетворяют соотношениям

$$\sum_{j=2}^{i-1} a_{ij}c_j = c_i^2/2, \quad i = 3,\ldots,s, \tag{2.17}$$

из которых определяем коэффициенты a_{i2}.

Построенное на основе условий порядка и (2.17) семейство пятистадийных методов 4-го порядка имеет свободные параметры c_2, c_3, c_4 и a_{53}. Остальные коэффициенты находим по формулам

$$b_3 = \frac{2c_4 - 1}{12c_3(c_3 - c_4)(c_3 - 1)}, \quad b_4 = \frac{2c_3 - 1}{12c_4(c_4 - c_3)(c_4 - 1)}, \quad b_5 = \frac{3 + 6c_3c_4 - 4(c_3 + c_4)}{12(1 - c_3)(1 - c_4)},$$

$$a_{54} = \frac{1 - 6(b_3c_3^3 + b_4c_4^2c_3 + b_5c_3)}{12b_5c_4(c_4 - c_3)}, \quad a_{43} = \frac{1 - 12b_5(a_{53}c_3^2 + a_{54}c_4^2)}{12b_4c_3^2}, \quad b_2 = 0, \quad c_5 = 1, \qquad (2.18)$$

$$a_{i2} = \frac{1}{2c_2}\left(c_i^2 - 2\sum_{j=3}^{i-1} a_{ij}c_j\right), \quad a_{i1} = c_i - \sum_{j=2}^{i-1} a_{ij}, \quad b_1 = 1 - b_3 - b_4 - b_5.$$

Коэффициенты погрешности методов этого семейства связаны соотношениями:

$e(T_{51}) = e(T_{52}) = e(T_{55})$, $e(T_{56}) = e(T_{57}) = -4e(T_{51})$,
$e(T_{58}) = -4e(T_{53})$, $e(T_{59}) = -4e(T_{54})$.

Чтобы их минимизировать, достаточно минимизировать $e(T_{51})$, $e(T_{53})$ и $e(T_{54})$. Для упрощения расчета мы задали $e(T_{51}) = e(T_{53}) = e(T_{54})$, в результате получили двухпараметрическое семейство со свободными коэффициентами c_2 и c_3, а остальные коэффициенты находим по формулам

$$c_4 = \frac{c_3}{2(1 - 2c_3)}, \quad a_{53} = \frac{(1 - c_3)(4 - 23c_3 + 36c_3^2 - 8c_3^3)}{2c_3^2(1 - 4c_3)(3 - 12c_3 + 11c_3^2)} \qquad (2.19)$$

и (2.18). Коэффициенты погрешности этих методов

$e(T_{5i}) = (5c_3 - 2)/8$, $i = 1,\ldots, 5$; $e(T_{5j}) = -4e(T_{51}) = (2 - 5c_3)/2$, $j = 6,\ldots, 9$.

При $c_3 \to 2/5$ эти коэффициенты $\to 0$, но в этом случае $|b_4| \to \infty$, $|b_5| \to \infty$. Известно, что пятистадийные методы не могут иметь 5-й порядок, но они могут иметь сколь угодно малые коэффициенты погрешности 5-го порядка. Однако в этом случае получаем большие значения коэффициентов b_4 и b_5, что неизбежно сказывается на точности метода.

При $c_2 = c_3 = 1/3$ формулы (2.18), (2.19) задают известный метод Мерсона, который имеет $e(T_{5i}) = -1/24$, $i = 1,\ldots, 5$; $e(T_{5j}) = 1/6$, $j = 6,\ldots, 9$ при показателе (2.9) $D = 1.5$. Мы выбрали $c_2 = 1/4$, $c_3 = 3/8$, что позволило уменьшить коэффициенты погрешности до значений $e(T_{5i}) = -1/64$, $i = 1,\ldots, 5$; $e(T_{5j}) = 1/16$, $j = 6,\ldots, 9$ при $D = 1.74$. На основе этого метода построены две вложенные пары: RK4(3) с таблицей коэффициентов

0					
1/4	1/4				
3/8	3/32	9/32			
3/4	3/8	−9/8	3/2		
1	−7/27	2	−40/27	20/27	
$\mathbf{y}_1$	7/54	0	64/135	8/27	1/10
$\hat{\mathbf{y}}_1$	5/27	0	8/27	14/27	0

(2.20)

и RK4(3)F с таблицей

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
1/4 & 1/4 & & & & & \\
3/8 & 3/32 & 9/32 & & & & \\
3/4 & 3/8 & -9/8 & 3/2 & & & \\
1 & -7/27 & 2 & -40/27 & 20/27 & & \\
\hline
\mathbf{y}_1 & 7/54 & 0 & 64/135 & 8/27 & 1/10 & \\
\hat{\mathbf{y}}_1 & 1/18 & 0 & 32/45 & 0 & -1/10 & 1/3
\end{array}
. \qquad (2.21)
$$

Отличие этих пар заключается в том, что RK4(3) имеет коэффициенты погрешности вложенной формулы $e(T_{4i})$ = (1/16, 1/16, 1/16, –5/16), а у RK4(3)F они одинаковы и равны –1/12.

2.7. Методы 5-го порядка

Среди методов Рунге–Кутты средней точности популярны вложенные пары порядков 5(4). Наиболее известен среди них метод Дорманда–Принса с минимизированными коэффициентами погрешности $e(T_{6i})$ [74, 98], который реализован в решателе ode45 системы MATLAB. Таблица Бутчера этого метода (обозначим его DP5(4)F) имеет вид:

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
\hline
\mathbf{y}_1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hat{\mathbf{y}}_1 & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40}
\end{array}
. \qquad (2.22)
$$

Этот метод, как и многие другие известные методы 5-го порядка, имеет большие значения некоторых коэффициентов, поэтому в [57] была предпринята попытка найти аналогичные методы с малыми значениями показателя D.

Поиск осуществлялся путем минимизации значения D, для чего использовался режим «Оптимизация» ПК МВТУ. Были исследованы два пятипараметрических семейства (в первом $b_2 = 0$, а во втором $b_2 = b_3 = 0$), но ни в одном из них не было найдено методов, имеющих $D = 0$. Наименьшее значение D было получено среди методов первого семейства, которое и будем рассматривать. В этом случае задаем $b_2 = 0$, $c_6 = 1$, а в качестве свободных параметров принимаем c_2, c_3, c_4, c_5, a_{42}. Остальные коэффициенты находим по формулам

$$a_{32} = \frac{c_3^2}{2c_2}, \quad a_{43} = \frac{c_4^2 - 2a_{42}c_2}{2c_3}, \quad \begin{bmatrix} b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} c_3 & c_4 & c_5 & 1 \\ c_3^2 & c_4^2 & c_5^2 & 1 \\ c_3^3 & c_4^3 & c_5^3 & 1 \\ c_3^4 & c_4^4 & c_5^4 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1/2 \\ 1/3 \\ 1/4 \\ 1/5 \end{bmatrix}$$

$$a_{52} = \frac{b_3(c_3 - 1)a_{32} + b_4(c_4 - 1)a_{42}}{b_5(1 - c_5)}, \quad a_{54} = \frac{2 - 5c_3}{120b_5c_4(1 - c_5)(c_4 - c_3)},$$

$$a_{53} = \frac{c_5^2 - 2(a_{52}c_2 + a_{54}c_4)}{2c_3}, \quad a_{63} = \frac{b_3(1 - c_3) - b_4a_{43} - b_5a_{53}}{b_6},$$

$$a_{64} = \frac{b_4(1 - c_4) - b_5a_{54}}{b_6}, \quad a_{65} = \frac{b_5(1 - c_5)}{b_6}, \quad a_{62} = \frac{c_6^2 - 2(a_{63}c_3 + a_{64}c_4 + a_{65}c_5)}{2c_2},$$

$$a_{i1} = c_i - \sum_{j=2}^{i-1} a_{ij}, \quad i = 3,\ldots,6; \quad b_1 = 1 - b_3 - b_4 - b_5 - b_6.$$

В результате оптимизации, проводимой из различных начальных точек, было найдено минимальное значение $D = 0.203$. В [57] приведен метод с близким значением $D = 0.3$, но более удобными коэффициентами. Однако для полученных методов невозможно построить вложенную формулу 4-го порядка. К тому же они имеют большие коэффициенты погрешности. Коэффициенты вложенной пары порядков 5(4) должны удовлетворять дополнительному условию, из которого получаем

$a_{42} = c_4^2(3c_3 - 2c_4)/(2c_2c_3)$.

Поскольку коэффициент a_{42} исключается из числа свободных параметров, имеем четырехпараметрическое семейство методов 5-го порядка с оцениванием погрешности по формуле 4-го порядка. Методы этого семейства исследовались в работах [98, 110, 134]. В [57] среди методов этого семейства был осуществлен поиск методов с малыми значениями D и коэффициентов погрешности.

В результате компромисса между точностью, малым значением D и надежностью оценки ошибки найден метод, который обозначим через RK5(4):

0						
1/5	1/5					
3/10	3/40	9/40				
3/5	3/10	−9/10	6/5			
4/5	32/135	−2/5	16/27	10/27		
1	−23/27	5/2	−14/27	−35/27	7/6	
$\mathbf{y}_1$	41/432	0	80/189	25/216	5/16	3/56
$\hat{\mathbf{y}}_1$	2/27	0	14/27	−5/54	1/2	0

. (2.23)

Этот метод принадлежит трехпараметрическому семейству non-FSAL-методов, удовлетворяющих дополнительному условию $c_4 = c_3/(10c_3^2 - 8c_3 + 2)$. Сравним основные показатели приведенных методов. DP5(4)F имеет $\|e(T_{6i})\|_2 = 0.204$ и $D = 11.89$, а RK5(4) имеет $\|e(T_{6i})\|_2 = 0.237$ и $D = 2.67$. Таким образом, у обоих методов примерно одинаковые по величине коэффициенты погрешности, но значение D значительно меньше у RK5(4).

2.8. Тестовое сравнение методов

Для сравнения мы выбрали 8 методов порядков 2, 3, 4 и 5, по 2 метода (non-FSAL и FSAL) каждого порядка. Это методы RK2(1) (2.11), RK2(1)F (2.14), RK3(2) (2.15), BS3(2)F (2.16), RK4(3) (2.20), RK4(3)F (2.21), RK5(4) (2.23) и DP5(4)F (2.22). Приведем результаты решения четырех задач. Это 3 задачи из [74] (VDPL, BRUS, PLEI) и задача с разрывами RELAY. В качестве точного решения в конце интервала принимаем численное решение, полученное при $Tol = 10^{-14}$ методом 5-го порядка.

VDPL – нежесткое уравнение Ван-дер-Поля:

$$y_1' = y_2,\ y_2' = (1 - y_1^2)y_2 - y_1,\ \ y_1(0) = 2,\ y_2(0) = 0,\ 0 \le t \le 20.$$

Точное решение: $y_1(20) = 2.0081497621749$, $y_2(20) = -0.0425088752731$.

BRUS – «брюсселятор»:

$$y_1' = 2 + y_1^2 y_2 - 9.533y_1,\ y_2' = 8.533y_1 - y_1^2 y_2,$$
$$y_1(0) = 1,\ y_2(0) = 4.2665,\ 0 \le t \le 20.$$

Точное решение: $y_1(20) = 6.8702504297725$, $y_2(20) = 6.852468904325$.

PLEY – задача небесной механики «Плеяды»:

$$x_i'' = \sum_{j \ne i} m_j (x_j - x_i)/r_{ij}^{3/2},\quad y_i'' = \sum_{j \ne i} m_j (y_j - y_i)/r_{ij}^{3/2},$$
$$r_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2,\ \ i, j = 1,\ldots, 7$$
$$\mathbf{x}(0) = (3, 3, -1, -3, 2, -2, 2)^{\mathrm{T}},\ \ \mathbf{y}(0) = (3, -3, 2, 0, 0, -4, 4)^{\mathrm{T}},$$
$$\mathbf{x}'(0) = (0, 0, 0, 0, 0, 1.75, -1.5)^{\mathrm{T}},\ \ \mathbf{y}'(0) = (0, 0, 0, -1.25, 1, 0, 0)^{\mathrm{T}},\ \ 0 \le t \le 3.$$

Точное решение в конце интервала приведено в [128].

RELAY – задача с нелинейностью релейного типа:

$$y_1' = y_2,\ y_2' = \mathrm{sign}(1 - y_1) + (1 - y_1)^3,\ \ y_1(0) = 2,\ y_2(0) = 0,\ 0 \le t \le 8.$$

Точное решение: $y_1(8) = 1.3749395055449$, $y_2(8) = 1.3191814402608$.

Графики решений задач VDPL, BRUS и PLEI приведены в [74], а задачи RELAY – на рис. 2.3.

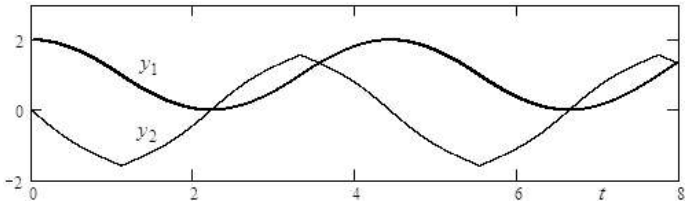


Рис. 2.3. Решение задачи RELAY

Мы задавали допуск на ошибку $Tol = 10^{-2}, 10^{-3}, \ldots, 10^{-6}$ для методов 2-го и 3-го порядков и $Tol = 10^{-3}, 10^{-4}, \ldots, 10^{-8}$ для методов 4-го и 5-го порядков. При этом принимаем $Rtol = Tol$ для всех задач; $Atol = 0.1Tol$ для BRUS и $Atol = Tol$ для остальных задач. Начальный шаг принимаем $h_0 = 0.01T \times Tol^{1/p}$, где T – величина интервала интегрирования. Ошибку вычисляем в виде

$$error = \sqrt{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \Big/ \sum_{i=1}^{n} y_i^2},$$

где y_i – точное, а $\tilde{y}_i$ – численное решение по i-й компоненте в конце интервала. Объем вычислений оцениваем числом вычислений правой части Nf. Результаты вычислений для каждой задачи приведены на рис. 2.4–2.7. Среди методов 2-го и 3-го порядков преимущество у BS3(2)F (за исключением задачи RELAY), а среди методов 4-го и 5-го порядков ни один не имеет очевидного преимущества. Для более наглядного сравнения этих методов на рис. 2.8 представлены зависимости общих затрат на решение четырех задач от усредненной (для каждого значения Tol) ошибки. Видно, что при $error > 10^{-3}$ метод BS3(2)F эффективнее, чем DP5(4)F. В интервале $10^{-6} < error < 10^{-3}$ методы 4-го и 5-го порядков примерно равнозначны, а при $error < 10^{-6}$ более эффективны методы 5-го порядка.

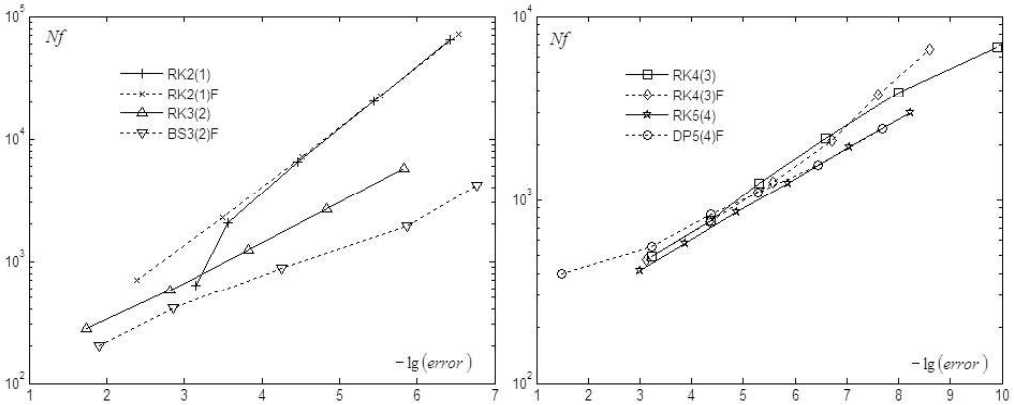


Рис. 2.4. Диаграмма «точность – объем вычислений» для задачи VDPL

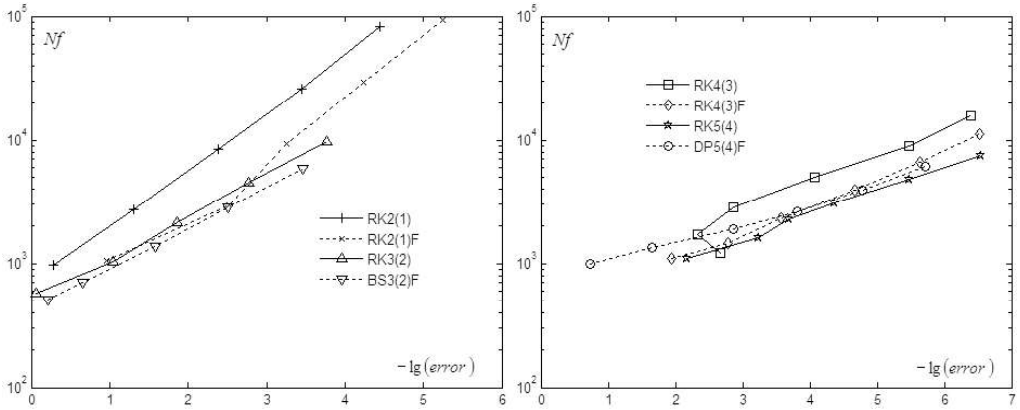


Рис. 2.5. Диаграмма «точность – объем вычислений» для задачи BRUS

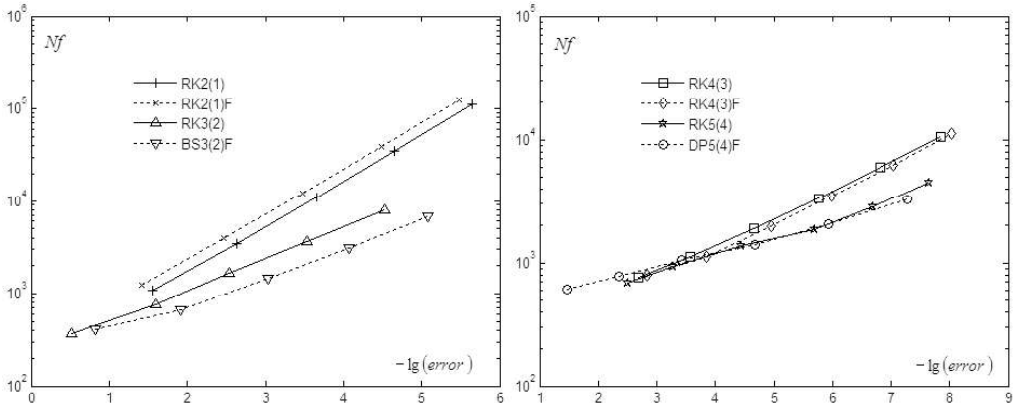


Рис. 2.6. Диаграмма «точность – объем вычислений» для задачи PLEI

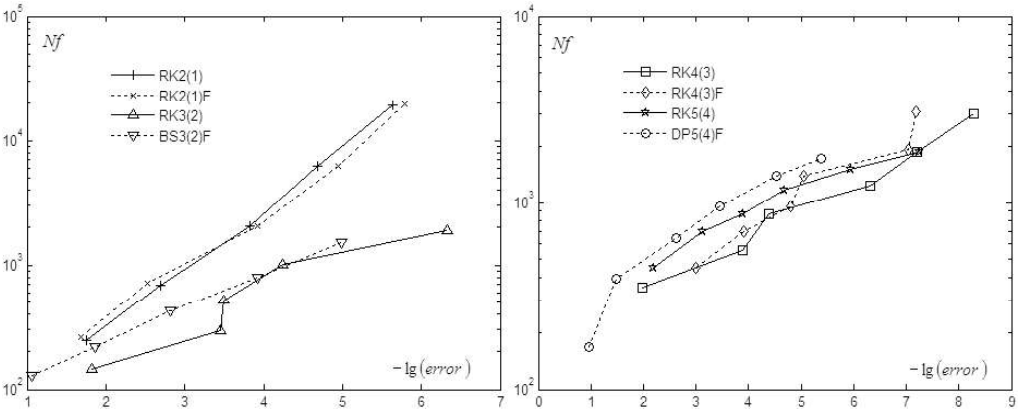


Рис. 2.7. Диаграмма «точность – объем вычислений» для задачи RELAY

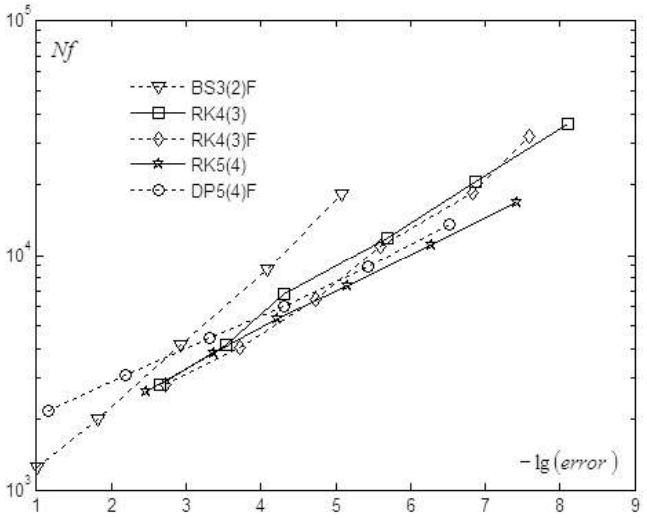


Рис. 2.8. Диаграмма «точность – объем вычислений» для четырех задач

2.9. Решение задач с разрывами

Задачу RELAY мы решали, используя обычные решатели со стандартной процедурой управления шагом. При этом размер шага уменьшается вблизи точки разрыва на несколько порядков, как это видно на рис. 2.2, что приводит к заметному увеличению вычислительных затрат. Покажем, что можно повысить эффективность решения таких задач, используя процедуру локализации точки разрыва. Запишем уравнения с разрывами в виде:

$$\mathbf{y}' = \begin{cases} \mathbf{f}_{\mathrm{I}}(\mathbf{y}), & g(\mathbf{y}) < 0, \\ \mathbf{f}_{\mathrm{II}}(\mathbf{y}), & g(\mathbf{y}) \geq 0. \end{cases} \tag{2.24}$$

В общем случае в точке разрыва может изменяться не только правая часть, но и значения некоторых переменных.

Пусть на очередном шаге $g(\mathbf{y}_n) < 0$, тогда шаг выполняем, используя $\mathbf{f}_{\mathrm{I}}(\mathbf{y})$. Пусть также $g(\mathbf{y}_{n+1}) > 0$, тогда точка разрыва находится внутри интервала $[t_n, t_n + h]$. Таким образом, локализация точки разрыва сводится к нахождению нуля функции

$$g(\theta) = g(\mathbf{y}(t_n + \theta h)), \quad 0 < \theta < 1. \tag{2.25}$$

Наиболее экономичный способ решения этой задачи основан на построении интерполяционного многочлена (интерполянта), аппроксимирующего функцию $\mathbf{y}(t_n + \theta h)$. Такой многочлен используется для *плотной выдачи* (dense output) [74], т. е. для выдачи результатов в любых промежуточных точках. Для этого используют *непрерывную вложенную формулу*

$$\mathbf{y}(t_n + \theta h) = \mathbf{y}_n + h\sum_{i=1}^{s} \beta_i(\theta)\mathbf{F}_i, \tag{2.26}$$

порядок которой должен быть не ниже $p-1$. Предполагаем, что в этой формуле может быть использовано также и значение $\mathbf{f}_{n+1} = \mathbf{f}(\mathbf{y}_{n+1})$, тогда матрицу $\mathbf{A}$ дополняем строкой $\mathbf{b}^{\mathrm{T}}$ и принимаем $\mathbf{F}_s = \mathbf{f}_{n+1}$. Условия, обеспечивающие порядок r такой формулы, запишутся в виде:

$$\gamma(T_{ij})\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Phi}(T_{ij}) = \theta^i, \quad i = 1,\ldots,r,\ j = 1,\ldots,N_i, \tag{2.27}$$

где $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_s)^{\mathrm{T}}$, N_i – число различных деревьев порядка i, а значения $\gamma(T_{ij})$ и $\boldsymbol{\Phi}(T_{ij})$ при $i \le 5$ приведены в табл. 2.1. Решая линейные уравнения (2.27), получаем коэффициенты β_i в виде многочленов от θ степени r, а подставляя их значения в (2.26), а затем (2.26) в (2.25), получаем зависимость $g(\theta)$.

Для уточнения точки разрыва находим решение θ^* уравнения (2.25), после чего принимаем $t_{n+1} = t_n + \theta^* h$, а решение системы (2.24) в очередной точке принимаем в виде (2.26) при $\theta = \theta^*$. На первой стадии следующего шага вычисляем $\mathbf{F}_1 = \mathbf{f}_{\mathrm{II}}(\mathbf{y}_{n+1})$ и продолжаем решение, используя уже $\mathbf{f}_{\mathrm{II}}(\mathbf{y})$ в качестве правой части. Таким образом, локализация точки разрыва сводится к решению уравнения (2.25), которое можно выполнить, используя, например, квадратичную интерполяцию по трем точкам в сочетании с последовательным уменьшением интервала неопределенности. Для сложных систем ОДУ при простой зависимости $g(\mathbf{y})$ такая процедура требует относительно небольших вычислительных затрат по сравнению с вычислением правой части.

Приведем непрерывные вложенные формулы. Для методов 2-го и 3-го порядков можно использовать формулу 2-го порядка, построенную по значениям $\mathbf{y}_n$, $\mathbf{y}_{n+1}$ и $\mathbf{f}_n = \mathbf{f}(\mathbf{y}_n)$ в виде

$$\mathbf{y}(t_n + \theta h) = \mathbf{y}_n + \theta h\mathbf{f}_n + \theta^2[(\mathbf{y}_{n+1} - \mathbf{y}_n) - h\mathbf{f}_n].$$

Добавив $\mathbf{f}_{n+1}$, получим формулу 3-го порядка:

$$\mathbf{y}(t_n + \theta h) = \mathbf{y}_n + \theta h\mathbf{f}_n + \theta^2[3(\mathbf{y}_{n+1} - \mathbf{y}_n) - 2h\mathbf{f}_n - h\mathbf{f}_{n+1}] + \theta^3[2(\mathbf{y}_n - \mathbf{y}_{n+1}) + h(\mathbf{f}_n + \mathbf{f}_{n+1})].$$

Формулы более высоких порядков получаем для каждого конкретного метода в виде (2.26), используя условия (2.27). Для метода RK4(3)F (2.21) коэффициенты такой формулы 4-го порядка:

$$\beta_1(\theta) = \theta - \frac{5}{2}\theta^2 + \frac{68}{27}\theta^3 - \frac{8}{9}\theta^4, \quad \beta_2(\theta) = 0, \quad \beta_3(\theta) = \frac{64}{15}\theta^2 - \frac{896}{135}\theta^3 + \frac{128}{45}\theta^4,$$

$$\beta_4(\theta) = -\frac{8}{3}\theta^2 + \frac{176}{27}\theta^3 - \frac{32}{9}\theta^4, \quad \beta_5(\theta) = -\frac{21}{10}\theta^2 + \frac{23}{5}\theta^3 - \frac{12}{5}\theta^4,$$

$$\beta_6(\theta) = 3\theta^2 - 7\theta^3 + 4\theta^4$$

(предполагаем, что $\mathbf{F}_6 = \mathbf{f}_{n+1}$). Коэффициенты формулы 4-го порядка для метода DP5(4)F (2.22) приведены в [74]. Для метода RK5(4) не удалось построить непрерывную формулу 4-го порядка, используя только значения внутренних стадий, поэтому использовали также и $\mathbf{F}_7 = \mathbf{f}_{n+1}$. Появившийся свободный коэффициент

$\beta_7(\theta)$ выбран из условия минимизации погрешности, в результате получены коэффициенты формулы 4-го порядка для RK5(4) (2.23) в виде:

$$\beta_1(\theta) = \theta - \frac{149}{48}\theta^2 + \frac{1625}{432}\theta^3 - \frac{25}{16}\theta^4, \quad \beta_2(\theta) = 0, \quad \beta_3(\theta) = \frac{110}{21}\theta^2 - \frac{1810}{189}\theta^3 + \frac{100}{21}\theta^4,$$

$$\beta_4(\theta) = -\frac{25}{8}\theta^2 + \frac{1825}{216}\theta^3 - \frac{125}{24}\theta^4, \quad \beta_5(\theta) = \frac{15}{16}\theta^2 - \frac{35}{16}\theta^3 + \frac{25}{16}\theta^4,$$

$$\beta_6(\theta) = -\frac{81}{56}\theta^2 + \frac{199}{56}\theta^3 - \frac{115}{56}\theta^4, \quad \beta_7(\theta) = \frac{3}{2}\theta^2 - 4\theta^3 + \frac{5}{2}\theta^4.$$

Методы с интерполяцией для локализации точки разрыва использовались для решения задачи RELAY. Мы убедились, что для метода порядка p использование интерполянтов порядков p и $p - 1$ дает очень близкие результаты, тогда как применение интерполянта порядка $p - 2$ приводит к заметному снижению эффективности. Это объясняется тем, что порядок глобальной ошибки на 1 меньше порядка локальной ошибки. Зависимости вычислительных затрат от ошибки решения задачи RELAY для методов без интерполяции и этих же методов с интерполяцией – RK2(1)Fi, BS3(2)Fi, RK4(3)Fi, RK5(4)i – приведены на рис. 2.9. Видно, что преимущество от использования интерполяции возрастает при повышении порядка метода.

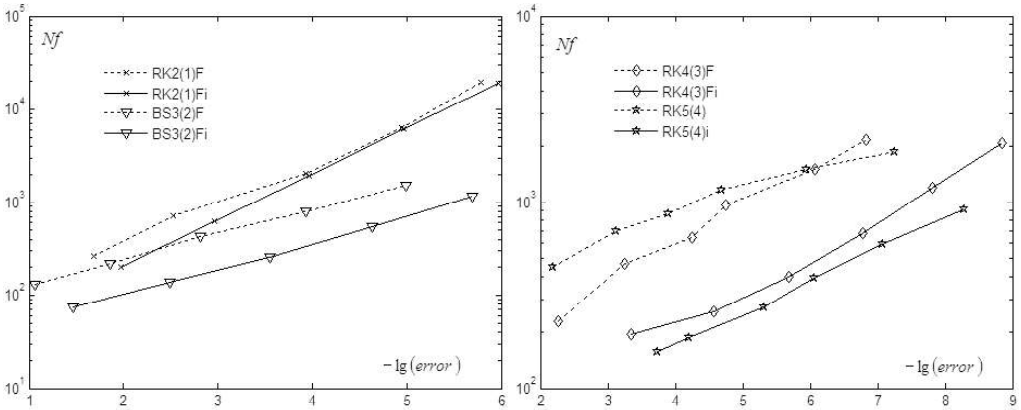


Рис. 2.9. Диаграмма «точность – объем вычислений»
при решении задачи RELAY без интерполяции и с интерполяцией

Неявные методы Рунге–Кутты и Розенброка 2-го порядка

ГЛАВА 3

3.1. Методы и их свойства

Неявные методы Рунге–Кутты обычно применяют для решения жестких и дифференциально-алгебраических уравнений. Важность эффективного решения таких задач обусловлена тем, что сложные многофункциональные объекты содержат элементы разной физической природы, и поэтому протекающие в них процессы имеют разный временной масштаб. А это означает, что уравнения, описывающие такие процессы, почти наверняка являются жесткими и могут содержать не только дифференциальные, но и алгебраические соотношения. Мы уже убедились, что классические явные методы не подходят для эффективного решения таких задач. Некоторые классы жестких задач могут быть эффективно решены специальными явными методами, но наиболее универсальным и надежным средством решения жестких уравнений остаются неявные методы.

Решение большинства инженерных задач не требует высокой точности вычислений. Поэтому имеет смысл применять методы низких порядков, которые при умеренных требованиях к точности более эффективны, чем методы высоких порядков. Необходимость разработки эффективных методов низкой точности отмечалась в [103]. Такие методы могут быть востребованы при решении больших задач, а также при моделировании в реальном времени. Методы низких порядков достаточно просты, поэтому на них удобно отлаживать различные процедуры, входящие в состав решателя ОДУ. Опыт эффективной реализации таких методов может быть полезен и при реализации более сложных методов высоких порядков.

Для решения многих ОДУ и ДАУ успешно применяют простые неявные методы 2-го порядка (например, из четырех неявных решателей системы MATLAB три имеют 2-й порядок). Поэтому рассмотрение неявных методов начнем именно с них. Выберем 5 наиболее известных методов 2-го порядка: средней точки MP2, трапеций TR2, диагонально-неявные методы SDIRK2 и TR-BDF2, метод Лобатто IIIC, который обозначим как Lobatto2. Коэффициенты этих методов задаются следующими таблицами:

$$\text{MP2:}\ \begin{array}{c|cc} 1/2 & 1/2 \\ \hline & 1 \end{array} \qquad \text{TR2:}\ \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \qquad \text{SDIRK2:}\ \begin{array}{c|cc} \gamma & \gamma & \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array},\ \gamma = 1-\frac{\sqrt{2}}{2}$$

$$\text{TR-BDF2:}\ \begin{array}{c|cc} 0 & 0 & \\ 2\gamma & \gamma & \gamma \\ 1 & \frac{1-\gamma}{2} & \frac{1-\gamma}{2} & \gamma \\ \hline & \frac{1-\gamma}{2} & \frac{1-\gamma}{2} & \gamma \end{array},\ \gamma = 1-\frac{\sqrt{2}}{2} \qquad \text{Lobatto2:}\ \begin{array}{c|cc} 0 & 1/2 & -1/2 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

(нулевые коэффициенты диагонально-неявных методов опущены). Метод TR-BDF2 принадлежит к классу однократно диагонально-неявных методов с явной первой стадией (ESDIRK) и может быть интерпретирован как последовательное применение правила трапеций (TR) и формулы дифференцирования назад 2-го порядка (BDF2).

Основные характеристики этих методов приведены в табл. 3.1. Здесь r – число неявных стадий, q – стадийный порядок (наименьший порядок на всех стадиях, включая формулу интегрирования). Все методы являются A-устойчивыми, а SDIRK2, TR-BDF2 и Lobatto2 также и L-устойчивы. За исключением MP2, все методы обладают свойством жесткой точности, т. е. последняя стадия совпадает с формулой интегрирования.

Таблица 3.1. Характеристики методов 2-го порядка

Метод	$R(\infty)$	r	q	Жесткая точность	$e(T_{31})$	$e(T_{32})$
MP2	–1	1	1	–	0.25	–0.5
TR2	–1	1	2	+	–0.5	–0.5
SDIRK2	0	2	1	+	–0.061	–0.243
TR-BDF2	0	2	2	+	–0.243	–0.243
Lobatto2	0	2	1	+	–0.5	1

Исследуем функции устойчивости, которые у методов MP2 и TR2 одинаковы и задаются формулой

$$R_1(z) = \frac{1+z/2}{1-z/2},$$

у методов SDIRK2 и TR-BDF2 также одинаковы и имеют вид

$$R_2(z) = \frac{1+(1-2\gamma)z}{(1-\gamma z)^2},$$

а функция устойчивости метода Lobatto2

$$R_3(z) = \frac{1}{1 - z + z^2/2}.$$

Графики этих функций приведены на рис. 3.1, где приведена также и «идеальная» функция устойчивости $R(z) = \exp(z)$, обеспечивающая точное решение уравнения Далквиста $y' = \lambda y$. При малых значениях z согласованность с экспонентой определяется коэффициентом погрешности $e(T_{32})$ и лучше всего у функции $R_2(z)$. Но при больших z лучше всего согласуется с экспонентой функция погрешности $R_3(z)$ метода Lobatto2, которая монотонно приближается к 0 при $z \to -\infty$. Метод Lobatto2 является также и $L2$-устойчивым (т. е. он L-устойчив и $R(z) = O(z^{-2})$ при $z \to \infty$ [16]).

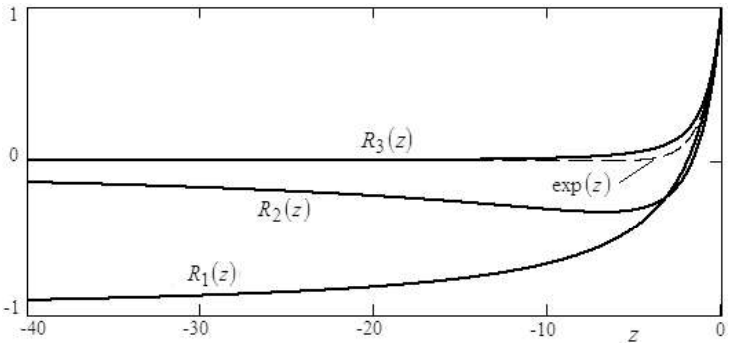


Рис. 3.1. Функции устойчивости неявных методов 2-го порядка

Поскольку наименьшие значения коэффициентов погрешности имеет метод SDIRK2, то следует ожидать, что он будет более точным при решении нежестких задач на заданной сетке. Но для жестких задач только малых коэффициентов ошибки в сочетании с A-устойчивостью оказывается недостаточно для эффективного и точного решения. В этом случае существенное влияние на точность решения оказывают вид функции устойчивости, стадийный порядок и жесткая точность. Чтобы исследовать зависимость точности численного решения от жесткости задачи, удобно использовать тесты с известным гладким решением, не зависящим от параметра жесткости. Приведем результаты решения задачи, которая использовалась для исследования сходимости методов в зависимости от жесткости в [8, 12, 15, 54] и других работах.

Задача Капса задается уравнениями

$$y_1' = -(\mu + 2)y_1 + \mu y_2^2, \quad y_2' = y_1 - y_2 - y_2^2,$$
$$y_1(0) = 1, y_2(0) = 1, 0 \leqslant t \leqslant 1, \qquad (3.1)$$

и имеет точное решение $y_1(t) = \exp(-2t)$, $y_2(t) = \exp(-t)$. Собственные числа матрицы Якоби в начальной точке равны $\lambda_{1,2} = -\frac{\mu+5}{2} \pm \frac{1}{2}\sqrt{\mu^2 + 6\mu + 1}$ и мало изменяются на интервале решения. При больших μ имеем $\lambda_1 \approx -(\mu + 4)$, $\lambda_2 \approx -1$.

Таким образом, значение μ вполне можно использовать как меру жесткости задачи. Зависимости ошибок решения задачи Капса от μ приведены на рис. 3.2. Ошибка вычислялась по формуле $e = \max\left(\sqrt{e_1(t)^2 + e_2(t)^2}, 0 \le t \le 1\right)$, где $e_i(t)$ – ошибка по i-й компоненте при размере шага $h = 1/30$.

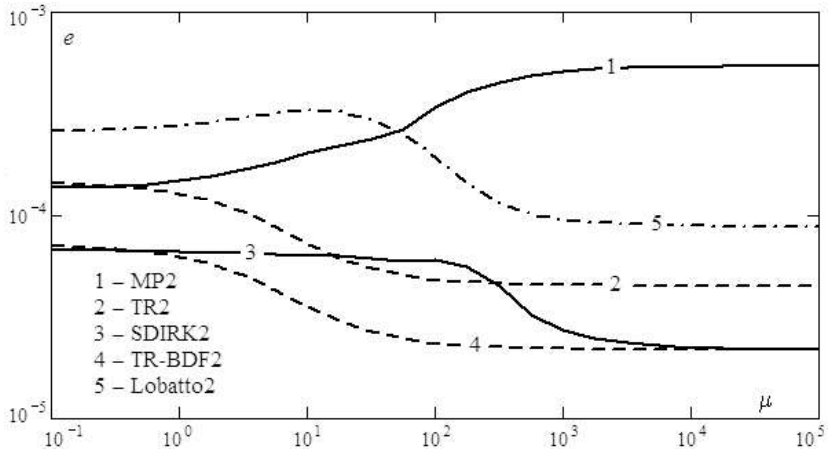


Рис. 3.2. Ошибки решения задачи Капса в зависимости от жесткости μ

Сравним методы с одинаковыми функциями устойчивости. По сравнению с TR2, метод MP2 имеет меньшее значение $e(T_{31})$, но только при малых μ он имеет незначительное преимущество. При увеличении μ ошибка метода MP2 увеличивается, а метода TR2 – уменьшается. Такое поведение ошибок объясняется тем, что при $\mu \to \infty$ первое уравнение в (3.1) вырождается в алгебраическое соотношение $y_1 - y_2^2 = 0$, точное выполнение которого обеспечивают жесткоточные методы, к которым относится TR2. При выполнении этого соотношения второе уравнение принимает вид $y_2' = -y_2$, тогда ошибка метода TR2 обусловлена фактически только ошибкой решения этого уравнения. Метод MP2 не является жесткоточным, а для таких методов ошибка возрастает при увеличении жесткости.

Метод SDIRK2 также имеет незначительное преимущество, по сравнению с TR-BDF2, только при малых μ, что объясняется меньшим значением $e(T_{31})$. Оба метода являются жесткоточными и имеют одинаковые функции устойчивости, поэтому их ошибки при больших μ практически совпадают. Но в диапазоне умеренных значений μ метод TR-BDF2 заметно более точен, что объясняется его более высоким стадийным порядком.

Среди рассмотренных методов Lobatto2 имеет самые большие коэффициенты погрешности, которые в 2 раза больше, чем у MP2. Поэтому при малых μ ошибка метода Lobatto2 примерно в 2 раза больше, чем у MP2, но при больших μ ошибка метода Lobatto2 уменьшается, поскольку он является жесткоточным. Небольшой «горб» при умеренных значениях μ объясняется тем, что

стадийный порядок метода Lobatto2 на 1 меньше его классического порядка. Более детальный анализ поведения ошибок методов Рунге–Кутты при решении жестких задач выполнен в следующей главе, где получены аналитические зависимости ошибок от жесткости для простейших модельных уравнений.

Подведем предварительные итоги. Среди рассмотренных методов наиболее точным при решении задачи Капса оказался метод TR-BDF2, ошибка которого во всем диапазоне изменения μ примерно в 2 раза меньше, чем у TR2 (это соответствует соотношению коэффициентов ошибок этих методов). Но метод TR2 имеет только одну неявную стадию, поэтому его затраты на одном шаге меньше, чем у TR-BDF2. В то же время каждый из этих методов имеет свои достоинства и свои недостатки. Например, TR2 может быть более эффективным при решении колебательных задач и неэффективным при решении некоторых жестких задач и ДАУ высших индексов, поскольку не является L-устойчивым и имеет $R(\infty) = -1$. Поэтому для реализации с автоматическим выбором шага мы выбрали наиболее точные методы TR2 и TR-BDF2, а также метод Lobatto2, функция устойчивости которого хорошо согласуется с экспонентой.

3.2. Схемы реализации

Эффективность решения задачи Коши определяется не только выбранным методом, но и его программной реализацией. В первую очередь это относится к неявным методам, схема реализации которых не задается однозначно коэффициентами формулы интегрирования. Выполнение одного шага неявного метода сводится к численному решению системы нелинейных алгебраических уравнений, для этого обычно применяют итерации Ньютона. Вычислительная схема должна также содержать условия перерасчета матрицы Якоби и прекращения итераций.

Вычислительные затраты неявного метода можно оценить следующими величинами: 1) число вычислений функции $\mathbf{f}(t, \mathbf{y})$; 2) число вычислений матрицы Якоби; 3) число LU-факторизаций матрицы; 4) число решений линейных уравнений с факторизованной матрицей коэффициентов. Для сложных нелинейных задач наибольшие затраты связаны с вычислением якобиана, поэтому решение алгебраических уравнений производят модифицированным методом Ньютона, в котором матрица Якоби «заморожена» в течение нескольких шагов интегрирования. В этом случае процедура LU-факторизации выполняется не более одного раза на шаге интегрирования (после изменения величины шага или после перерасчета якобиана). Второй по трудоемкости обычно является процедура вычисления функции, число обращений к которой определяется числом итераций и существенно зависит от выбранного начального значения. Для уменьшения числа итераций применяют специальную формулу прогноза начального значения (нетривиальный предиктор). Условия прекращения итераций и обновления матрицы Якоби формируются на основе оценки ошибки решения алгебраических уравнений и оценки скорости сходимости итераций.

Кроме этого, на каждом шаге оценивается ошибка численного решения, которая используется при выборе размера следующего шага. Вопросы эффективной реализации неявных методов Рунге–Кутты рассматривались в работах [28, 58, 65, 66, 75, 82, 105, 111, 112, 124, 133].

Таким образом, эффективная схема реализации неявного метода Рунге–Кутты должна использовать замороженную матрицу Якоби и нетривиальный прогноз. Для методов низких порядков потребуем также, чтобы схема допускала не более одного вычисления функции на каждой неявной стадии, не ухудшая при этом заметно свойств точности и устойчивости исходного метода.

Наряду с хорошо известными схемами реализации рассмотрим также предложенную в [58] схему, идея которой заключается в том, что прогноз выполняется не только для переменных, но и для производных. Формулу прогноза можно получить как значение интерполяционного многочлена, построенного по уже вычисленным стадийным значениям (могут быть использованы значения, полученные на предыдущих шагах). В этом случае прогноз для переменных представляет собой линейную комбинацию предыдущих значений переменных. Прогноз для производных получаем в виде такой же линейной комбинации предыдущих значений производных. После окончания итераций определение производных выполняется из формулы интегрирования и не требует вычисления правой части.

В предложенной в [58] схеме число вычислений функции на каждой неявной стадии на единицу меньше числа итераций. При использовании нетривиального прогноза для сходимости бывает достаточно двух итераций, т. е. одного вычисления функции на каждой неявной стадии. Покажем также, что можно построить работоспособную схему, в которой одна из неявных стадий вообще не требует вычислений правой части.

Рассмотрим четыре схемы реализации на примере метода трапеций. Один шаг решения задачи Коши методом трапеций задается формулой

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}(\mathbf{f}_n + \mathbf{f}_{n+1}), \tag{3.2а}$$

$$\mathbf{f}_{n+1} = \mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}), \tag{3.2б}$$

которая представляет собой систему нелинейных алгебраических уравнений относительно $\mathbf{y}_{n+1}$. Итерационный процесс вычисления этого вектора запишется в виде:

$$\mathbf{y}^i_{n+1} = \mathbf{y}^{i-1}_{n+1} + \left(\mathbf{I} - \frac{h}{2}\mathbf{J}\right)^{-1}\left[\frac{h}{2}(\mathbf{f}_n + \mathbf{f}^{i-1}_{n+1}) - (\mathbf{y}^{i-1}_{n+1} - \mathbf{y}_n)\right], \quad i = 1,\ldots,N; \tag{3.3а}$$

$$\mathbf{f}^i_{n+1} = \mathbf{f}(t_{n+1}, y^i_{n+1}), \quad i = 1,\ldots,N-1, \tag{3.3б}$$

где $\mathbf{I}$ – единичная матрица, $\mathbf{J}$ – аппроксимация матрицы Якоби $\partial\mathbf{f}/\partial\mathbf{y}$, N – число итераций. Будем предполагать, что матрица $\mathbf{J}$ сохраняется постоянной в течение нескольких шагов интегрирования и не пересчитывается в процессе итераций.

Перед началом итераций нужно задать начальные значения $\mathbf{y}^0_{n+1}$, $\mathbf{f}^0_{n+1}$, а после окончания итераций следует вычислить значение $\mathbf{y}_{n+1}$, а также значение $\mathbf{f}_{n+1}$, которое будет использовано на следующем шаге (если текущий шаг не последний). Таким образом, схема реализации одного шага метода трапеций задается формулами (3.3) и формулами вычисления $\mathbf{y}^0_{n+1}$, $\mathbf{f}^0_{n+1}$, $\mathbf{y}_{n+1}$ и $\mathbf{f}_{n+1}$. Рассмотрим конкретные схемы.

Тривиальная схема (T). Проще всего задать стартовые значения в виде

$$\mathbf{y}^0_{n+1} = \mathbf{y}_n, \quad \mathbf{f}^0_{n+1} = \mathbf{f}_n \tag{3.4}$$

(тривиальный прогноз). Приняв также

$$\mathbf{y}_{n+1} = \mathbf{y}^N_{n+1}, \quad \mathbf{f}_{n+1} = \mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}), \tag{3.5}$$

получим простейшую схему (3.3)–(3.5), которую назовем *тривиальной*.

Стандартная схема (S). Можно уменьшить число итераций, если использовать нетривиальный прогноз. Применяя экстраполяцию первого порядка для переменных, получаем

$$\mathbf{y}^0_{n+1} = \mathbf{y}_n + w(\mathbf{y}_n - \mathbf{y}_{n-1}), \quad \mathbf{f}^0_{n+1} = \mathbf{f}(t_{n+1}, \mathbf{y}^0_{n+1}), \tag{3.6}$$

где $w = h/\overline{h}$ – отношение размера текущего шага к размеру предыдущего шага. Вектор производных $\mathbf{f}_{n+1}$ будем определять из (3.2а), а не из (3.2б), тогда

$$\mathbf{y}_{n+1} = \mathbf{y}^N_{n+1}, \quad \mathbf{f}_{n+1} = \frac{2}{h}(\mathbf{y}_{n+1} - \mathbf{y}_n) - \mathbf{f}_n. \tag{3.7}$$

Такой прием применяется при реализации неявных методов с явной первой стадией; в [112] он назван «сглаженная первая стадия» (smoothed first stage). Схему, задаваемую формулами (3.3), (3.6), (3.7), назовем *стандартной*. Подобные схемы обычно используют в программных реализациях неявных методов. В стандартной схеме, как и в тривиальной, число вычислений функции равно числу итераций.

Схема с прогнозом для производных (D). Прогноз для переменных выполняем следующим образом: делаем предварительный прогноз для производных $\hat{\mathbf{f}}_{n+1} = \mathbf{f}_n + w(\mathbf{f}_n - \mathbf{f}_{n-1})$, который подставляем в формулу интегрирования (3.2а), в результате получаем явную формулу Адамса 2-го порядка. Окончательный прогноз получаем в виде:

$$\mathbf{y}^0_{n+1} = \mathbf{y}_n + h\mathbf{f}_n + \frac{h}{2}w(\mathbf{f}_n - \mathbf{f}_{n-1}), \quad \mathbf{f}^0_{n+1} = \mathbf{f}(t_{n+1}, \mathbf{y}^0_{n+1}), \tag{3.8}$$

а формулы (3.3), (3.8), (3.7) задают схему *с прогнозом для производных*.

Экономичная схема (E). В предложенной в [58] схеме прогноз выполняется не только для переменных, но и для производных, т. е. вместо (3.6) принимаем

$$\mathbf{y}^0_{n+1} = \mathbf{y}_n + w(\mathbf{y}_n - \mathbf{y}_{n-1}), \quad \mathbf{f}^0_{n+1} = \mathbf{f}_n + w(\mathbf{f}_n - \mathbf{f}_{n-1}). \tag{3.9}$$

Подобный прогноз применялся в [84, формула (15)] при реализации тета-метода, задаваемого формулой $\mathbf{y}_{n+1} = \mathbf{y}_n + (1 - \theta)h\mathbf{f}_n + \theta h\mathbf{f}_{n+1}$. В схеме, задаваемой формулами (3.3), (3.9), (3.7), число вычислений функции на единицу меньше

числа итераций. Численные эксперименты показали, что такая схема позволяет минимизировать число вычислений функции и матрицы Якоби, поэтому назовем ее *экономичной*.

В формуле (3.3) итерации выполняются относительно переменных. Однако можно выполнять итерации и относительно производных, которые связаны с переменными формулой интегрирования (3.2а). Такие итерации запишутся в виде:

$$\begin{aligned}\mathbf{f}_{n+1}^{i} &= \mathbf{f}_{n+1}^{i-1} + \left(\mathbf{I} - \frac{h}{2}\mathbf{J}\right)^{-1}\left(\mathbf{f}(t_{n+1}, y_{n+1}^{i-1}) - \mathbf{f}_{n+1}^{i-1}\right),\\ \mathbf{y}_{n+1}^{i} &= \mathbf{y}_{n} + \frac{h}{2}(\mathbf{f}_{n} + \mathbf{f}_{n+1}^{i}), \quad i = 1,\ldots,N.\end{aligned} \tag{3.10}$$

В некоторых случаях итерации в виде (3.10) более удобны, чем в виде (3.3). Например, при реализации схемы D прогноз для итераций задаем в виде

$$\mathbf{f}_{n+1}^{0} = \mathbf{f}_{n} + w(\mathbf{f}_{n} - \mathbf{f}_{n-1}), \quad \mathbf{y}_{n+1}^{0} = \mathbf{y}_{n} + \frac{h}{2}(\mathbf{f}_{n} + \mathbf{f}_{n+1}^{0}),$$

а финальные значения получаем непосредственно из (3.10), т. е. в виде

$$\mathbf{y}_{n+1} = \mathbf{y}_{n+1}^{N}, \quad \mathbf{f}_{n+1} = \mathbf{f}_{n+1}^{N}.$$

Ниже покажем, что при $N = 1$ такие схемы задают методы типа Розенброка.

Оценим эффективность рассмотренных схем при решении задачи Капса (3.1) и условии, что матрица Якоби вычисляется только один раз в начальной точке. Ошибки решения этой задачи различными схемами метода трапеций с шагом $h = 1/30$ приведены в табл. 3.2, где nf – число вычислений функции на одном шаге (для схем T, S и D $nf = N$, а для схемы E $nf = N - 1$). В последней строке приведены ошибки при точной реализации метода трапеций, т. е. при выполнении итераций до сходимости. Такие ошибки получены схемой T при $nf = 4$, схемой S при $nf = 3$, схемами D и E при $nf = 2$. Отметим, что самая медленная сходимость наблюдается при умеренной жесткости ($\mu = 10^1, 10^2$).

Таблица 3.2. Ошибки схем реализации метода трапеций

Схема	nf	μ				
		10^0	10^1	10^2	10^3	10^4
T	1	3.21×10^{-3}	6.30×10^{-3}	8.09×10^{-3}	8.34×10^{-3}	8.37×10^{-3}
	2	1.36×10^{-4}	1.43×10^{-4}	9.35×10^{-5}	5.17×10^{-5}	4.61×10^{-5}
	3	1.29×10^{-4}	7.39×10^{-5}	4.87×10^{-5}	4.58×10^{-5}	4.55×10^{-5}
S	1	2.50×10^{-4}	3.97×10^{-4}	6.68×10^{-4}	1.02×10^{-3}	1.07×10^{-3}
	2	1.28×10^{-4}	7.12×10^{-5}	4.63×10^{-5}	4.54×10^{-5}	4.54×10^{-5}
D	1	1.29×10^{-4}	7.86×10^{-5}	5.56×10^{-5}	5.35×10^{-5}	5.31×10^{-5}
E	1	1.29×10^{-4}	7.71×10^{-5}	4.93×10^{-5}	4.58×10^{-5}	4.55×10^{-5}
Точная	> 3	1.29×10^{-4}	7.36×10^{-5}	4.85×10^{-5}	4.58×10^{-5}	4.55×10^{-5}

Из приведенных результатов видно, что самыми эффективными являются схемы D и E, которые обеспечивают приемлемую сходимость при одном вычислении функции на шаге, тогда как тривиальная схема требует трех вычислений, а стандартная схема – двух вычислений. В то же время прогноз по формуле (3.8) представляет собой явную формулу Адамса 2-го порядка и не является устойчивым, что может отрицательно сказаться при решении очень жестких задач. Это видно и из табл. 3.2, где при $\mu = 10^3, 10^4$ ошибка схемы E совпадает с ошибкой точной схемы, а ошибка схемы D немного отличается от точной. Численные эксперименты с переменным шагом подтвердили, что при решении очень жестких задач (например, теста ROBER) схема E более надежна и эффективна, чем схема D. Поэтому приводим экономичные схемы (E-схемы) выбранных методов.

3.3. Метод трапеций

Выше уже была рассмотрена E-схема реализации метода трапеций с прогнозом 1-го порядка (3.9). Но более эффективной для сложных задач оказалась схема с прогнозом 2-го порядка, которую мы и рассмотрим. Вместо значений переменных и производных удобно использовать их приращения. Обозначим

$w_1 = h/(t_n - t_{n-1}), \quad w_2 = (t_n - t_{n-1})/(t_{n-1} - t_{n-2}),$

$\Delta\mathbf{y}_n = \mathbf{y}_n - \mathbf{y}_{n-1}, \quad \Delta\mathbf{f}_n = \mathbf{f}_n - \mathbf{f}_{n-1}$

тогда формулы прогноза запишутся в виде:

$\Delta\mathbf{y}^0_{n+1} = \alpha_1\Delta\mathbf{y}_n + \alpha_2\Delta\mathbf{y}_{n-1}, \quad \mathbf{f}^0_{n+1} = \mathbf{f}_n + \alpha_1\Delta\mathbf{f}_n + \alpha_1\Delta\mathbf{f}_{n-1}.$

$\alpha_1 = w_1 + w_1w_2\,(1 + w_1)/(1 + w_2), \quad \alpha_2 = -w_1w_2^2(1 + w_1)/(1 + w_2)$

(на 1-м шаге принимаем $\alpha_1 = \alpha_2 = 0$, а на 2-м шаге принимаем $\alpha_1 = w_1$, $\alpha_2 = 0$). Если изменялся размер шага или обновлялась матрица Якоби, то вычисляем $\mathbf{D} = \mathbf{I} - \frac{h}{2}\mathbf{J}$ и выполняем LU-факторизацию матрицы $\mathbf{D}$. Далее выполняем 1-ю итерацию, которая не нуждается в вычислении правой части:

$$\mathbf{D}(\Delta\mathbf{y}^1_{n+1} - \Delta\mathbf{y}^0_{n+1}) = \frac{h}{2}(\mathbf{f}_n + \mathbf{f}^0_{n+1}) - \Delta\mathbf{y}^0_{n+1}, \quad \mathbf{y}^1_{n+1} = \mathbf{y}_n + \Delta\mathbf{y}^1_{n+1}.$$

После этого выполняем одну или две итерации с вычислением правой части:

$$\mathbf{D}(\Delta\mathbf{y}^i_{n+1} - \Delta\mathbf{y}^{i-1}_{n+1}) = \frac{h}{2}\left[\mathbf{f}_n + \mathbf{f}(t_{n+1}, \mathbf{y}^{i-1}_{n+1})\right] - \Delta\mathbf{y}^{i-1}_{n+1}, \quad \mathbf{y}^i_{n+1} = \mathbf{y}_n + \Delta\mathbf{y}^i_{n+1}, \quad i = 2, N,$$

где $N \le 3$ – общее число итераций. После выполнения итераций принимаем

$$\mathbf{y}_{n+1} = \mathbf{y}^N_{n+1}, \quad \mathbf{f}_{n+1} = \frac{2}{h}\Delta\mathbf{y}_{n+1} - \mathbf{f}_n$$

(вектор производных $\mathbf{f}_{n+1}$ вычисляем из формулы интегрирования).

Локальную ошибку оцениваем по формуле

$$\delta\mathbf{y} = \begin{cases} \Delta\mathbf{y}_{n+1} - h\mathbf{f}_n, & n = 0, \\ \Delta\mathbf{y}_{n+1} - w_1\Delta\mathbf{y}_n, & n > 0, \end{cases}$$

что соответствует вложенной формуле

$$\hat{\mathbf{y}}_{n+1} = \begin{cases} \mathbf{y}_n + h\mathbf{f}_n, & n = 0, \\ \mathbf{y}_n + w_1\Delta\mathbf{y}_n, & n > 0, \end{cases}$$

имеющей 1-й порядок. Нормированную оценку ошибки принимаем в виде $\delta = \mathrm{err}(\delta\mathbf{y})$, где

$$\mathrm{err}(\delta\mathbf{y}) = K \times \max_i \left(\frac{|\delta y_i|}{Atol + Rtol \times \max\left(|y_{ni}|, |y_{n+1,i}|\right)} \right), \tag{3.11}$$

Atol – допустимая абсолютная ошибка, *Rtol* – допустимая относительная ошибка. Задаем $K = 0.2w_1/(1 + w_1)$, тогда оценка ошибки пропорциональна норме оценки 2-й производной, вычисленной при $t = t_{n+1}$ по значениям $\mathbf{y}_{n+1}$, $\mathbf{y}_n$ и $\mathbf{y}_{n-1}$. На первом шаге $K = 0.2$. Если $\delta \le 1$, то шаг считается успешным, в противном случае шаг отбрасывается. Размер нового шага задаем в виде:

$$h_{\text{new}} = w_{\text{new}} h, \quad w_{\text{new}} = \begin{cases} 1, & |1 - w_0| \le 0.2, \\ w_0, & |1 - w_0| > 0.2, \end{cases} \quad w_0 = \max\left(1/8, \min(4, 0.55\delta^{-0.5})\right). \tag{3.12}$$

Размер шага сохраняется при его прогнозируемом изменении менее чем на 20%, что позволяет уменьшить число LU-факторизаций матрицы **D**.

Сформируем условия выполнения дополнительной (3-й) итерации и обновления якобиана. После выполнения 2-й итерации оцениваем ошибку шага $\delta = \mathrm{err}(\delta\mathbf{y})$ и ошибку итераций $\delta_1 = \mathrm{err}(\Delta\mathbf{y}^2_{n+1} - \Delta\mathbf{y}^1_{n+1})$; 3-я итерация выполняется, если $\delta_1 > K_1\delta$. Подходящее значение K_1 подбиралось экспериментально, при этом оказалось, что в эффективной схеме это значение должно уменьшаться при уменьшении допустимой ошибки. По результатам серии экспериментов выбрано значение $K_1 = Rtol^{0.5}$. После выполнения 3-й итерации вычисляем $\delta_2 = \mathrm{err}(\Delta\mathbf{y}^3_{n+1} - \Delta\mathbf{y}^2_{n+1})$ и $\delta = \mathrm{err}(\delta\mathbf{y})$ при новом значении $\mathbf{y}_{n+1}$. Обновление матрицы Якоби возможно только после успешного шага при условии выполнения 3-й итерации. В этом случае оцениваем скорость сходимости итераций в виде $\theta = \delta_2/\delta_1$, тогда оценка ошибки последней итерации равна $\delta_2\theta/(1 - \theta)$. Матрица Якоби обновляется при выполнении хотя бы одного из двух условий: $\theta > \theta_{\max}$ или $\delta_2\theta/(1 - \theta) > K_2\delta$, где $\theta_{\max} = 0.5$, $K_2 = 0.2$.

3.4. Метод TR-BDF2

Один шаг метода TR-BDF2 [112] задается формулами:

$$\begin{aligned} &\mathbf{Y} = \mathbf{y}_n + h\gamma(\mathbf{f}_n + \mathbf{F}), \quad \mathbf{F} = \mathbf{f}(t_n + ch, \mathbf{Y}), \quad \mathbf{y}_{n+1} = \mathbf{y}_n + h[a(\mathbf{f}_n + \mathbf{F}) + \gamma\mathbf{f}_{n+1}], \\ &\mathbf{f}_{n+1} = \mathbf{f}(t_n + h, \mathbf{y}_{n+1}), \quad \gamma = 1 - \sqrt{2}/2, \quad c = 2\gamma, \quad a = (1 - \gamma)/2. \end{aligned} \tag{3.13}$$

Этот метод реализован в решателе ode23tb системы MATLAB, а также в одном из решателей ПО SimInTech. Метод формально имеет 3 стадии, но 1-я (явная) стадия не требует вычислений, поскольку результатом ее выполнения являются значения $\mathbf{y}_n$ и $\mathbf{f}_n$, полученные на последней стадии предыдущего шага. Поэтому рассмотрим только неявные стадии.

Будем использовать прогноз 2-го порядка по трем последним точкам. Ограничимся одним вычислением правой части на 2-й стадии и одним либо двумя вычислениями на 3-й стадии. Тогда E-схема реализации метода TR-BDF2 запишется в следующем виде.

Прогноз 2-й стадии:

$$\Delta\mathbf{Y}^0_{n+1} = \alpha_1(\Delta\mathbf{y}_n - \Delta\mathbf{Y}_n) + \alpha_2\Delta\mathbf{y}_n, \quad \mathbf{F}^0_{n+1} = \mathbf{f}_n + \alpha_1(\mathbf{f}_n - \mathbf{F}_n) + \alpha_2\Delta\mathbf{f}_n,$$
$$\alpha_1 = w(1+wc)/(1-c), \quad \alpha_2 = -w(1-c+wc), \quad w = h/(t_n - t_{n-1}).$$

2-я стадия:

$$\mathbf{D} = (\mathbf{I} - h\gamma\mathbf{J}), \quad \mathbf{D}(\Delta\mathbf{Y}^1_{n+1} - \Delta\mathbf{Y}^0_{n+1}) = h\gamma(\mathbf{f}_n + \mathbf{F}^0_{n+1}) - \Delta\mathbf{Y}^0_{n+1},$$

$$\mathbf{D}(\Delta\mathbf{Y}_{n+1} - \Delta\mathbf{Y}^1_{n+1}) = h\gamma\left[\mathbf{f}_n + \mathbf{f}(t_n + ch, \mathbf{y}_n + \Delta\mathbf{Y}^1_{n+1})\right] - \Delta\mathbf{Y}^1_{n+1}, \quad \mathbf{F}_{n+1} = \frac{1}{h\gamma}\Delta\mathbf{Y}_{n+1} - \mathbf{f}_n.$$

Прогноз 3-й стадии:

$$\Delta\mathbf{y}^0_{n+1} = \beta_1\Delta\mathbf{Y}_{n+1} + \beta_2(\Delta\mathbf{y}_n - \Delta\mathbf{Y}_n), \quad \mathbf{f}^0_{n+1} = \mathbf{f}_n + \beta_1(\mathbf{F}_{n+1} - \mathbf{f}_n) + \beta_2(\mathbf{f}_n - \mathbf{F}_n),$$
$$\beta_1 = (1-c+w)/[c(1-c+wc)], \quad \beta_2 = -w^2(1-c+wc).$$

3-я стадия:

$$\mathbf{D}(\Delta\mathbf{y}^1_{n+1} - \Delta\mathbf{y}^0_{n+1}) = h\left[a(\mathbf{f}_n + \mathbf{F}_{n+1}) + \gamma\mathbf{f}^0_{n+1}\right] - \Delta\mathbf{y}^0_{n+1}, \quad \mathbf{y}^1_{n+1} = \mathbf{y}_n + \Delta\mathbf{y}^1_{n+1},$$
$$\mathbf{D}(\Delta\mathbf{y}^i_{n+1} - \Delta\mathbf{y}^{i-1}_{n+1}) = h\left[a(\mathbf{f}_n + \mathbf{F}_{n+1}) + \gamma\mathbf{f}(t_n + h, \mathbf{y}^{i-1}_{n+1})\right] - \mathbf{y}^{i-1}_{n+1}, \ \mathbf{y}^i_{n+1} = \mathbf{y}_n + \Delta\mathbf{y}^i_{n+1}, \ i = 2, N,$$

$$\mathbf{y}_{n+1} = \mathbf{y}^N_{n+1}, \quad \mathbf{f}_{n+1} = \frac{1}{\gamma h}\Delta\mathbf{y}_{n+1} - \frac{a}{\gamma}(\mathbf{f}_n + \mathbf{F}_{n+1}).$$

На 1-м шаге принимаем $\alpha_1 = \alpha_2 = \beta_2 = 0$, $\beta_1 = 0.5/\gamma$. После выполнения 2-й итерации 2-й стадии вычисляем $\delta_1 = \mathrm{err}(\Delta\mathbf{y}^2_{n+1} - \Delta\mathbf{y}^1_{n+1})$ и $\delta = \mathrm{err}(\Delta\mathbf{y}^2_{n+1} - \Delta\mathbf{Y}_{n+1}/c)$. Норму ошибки принимаем в виде (3.11), где $K = 0.3$. Если $\delta_1 > K_1\delta$, где $K_1 = Rtol^{0.5}$, то выполняем 3-ю итерацию, после чего вычисляем $\delta_2 = \mathrm{err}(\Delta\mathbf{y}^3_{n+1} - \Delta\mathbf{y}^2_{n+1})$, $\theta = \delta_2/\delta_1$ и $\delta = \mathrm{err}(\Delta\mathbf{y}^3_{n+1} - \Delta\mathbf{Y}_{n+1}/c)$. Размер следующего шага принимаем согласно формуле (3.12). При $\delta \le 1$ шаг считается успешным, в противном случае шаг отбрасывается. Матрица Якоби обновляется после успешного шага, если на 2-й стадии было сделано 3 итерации и при этом $\theta > 0.5$ или $\delta_2\theta/(1-\theta) > 0.2\delta$.

3.5. Метод Лобатто IIIC

Метод Лобатто IIIC 2-го порядка задается формулами:

$$\begin{aligned}&\mathbf{Y} = \mathbf{y}_n + \frac{h}{2}(\mathbf{F} - \mathbf{f}_{n+1}), \quad \mathbf{F} = \mathbf{f}(t_n, \mathbf{Y}),\\ &\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}(\mathbf{F} + \mathbf{f}_{n+1}), \quad \mathbf{f}_{n+1} = \mathbf{f}(t_n + h, \mathbf{y}_{n+1}).\end{aligned} \tag{3.14}$$

Рассмотрим сначала простейшую схему реализации метода (3.14) применительно к автономной системе $\mathbf{y}' = \mathbf{f}(\mathbf{y})$. Используя тривиальный прогноз $\mathbf{Y}^0 = \mathbf{y}^0_{n+1} = \mathbf{y}_n$ и ограничившись одной итерацией, получим:

$$\begin{bmatrix}\mathbf{I} - 0.5h\mathbf{J} & 0.5h\mathbf{J}\\ -0.5h\mathbf{J} & \mathbf{I} - 0.5h\mathbf{J}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{Y}\\ \mathbf{y}_{n+1} - \mathbf{y}_n\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ h\mathbf{f}_n\end{bmatrix}, \quad \Delta\mathbf{Y} = \mathbf{Y} - \mathbf{y}_n, \quad \mathbf{f}_{n+1} = \mathbf{f}(\mathbf{y}_{n+1}). \tag{3.15}$$

Исключив $\Delta\mathbf{Y}$ из (3.15), получим формулу интегрирования в виде:

$$\left(\mathbf{I} - h\mathbf{J} + \frac{h^2}{2}\mathbf{J}^2\right)(\mathbf{y}_{n+1} - \mathbf{y}_n) = \left(\mathbf{I} - \frac{h}{2}\mathbf{J}\right)h\mathbf{f}_n. \tag{3.16}$$

Эта формула задает ABC-схему [8], т. е. схему вида

$$(\mathbf{I} + Ah\mathbf{J} + Bh^2\mathbf{J}^2)(\mathbf{y}_{n+1} - \mathbf{y}_n) = (\mathbf{I} + Ch\mathbf{J})h\mathbf{f}_n\,.$$

Обозначим $\alpha = (1+j)/2$, $\bar{\alpha} = (1-j)/2$, где j – мнимая единица. Тогда схему (3.16) можно представить в виде:

$$(\mathbf{I} - \alpha h\mathbf{J})(\mathbf{I} - \bar{\alpha}h\mathbf{J})(\mathbf{y}_{n+1} - \mathbf{y}_n) = \mathrm{Re}\left((\mathbf{I} - \bar{\alpha}h\mathbf{J})h\mathbf{f}_n\right),$$

откуда

$$\mathbf{y}_{n+1} - \mathbf{y}_n = \mathrm{Re}((\mathbf{I} - \alpha h\mathbf{J})^{-1}h\mathbf{f}_n). \tag{3.17}$$

Схема (3.17) известна как комплексная схема Розенброка (CROS) [1, 19, 139]. Таким образом, все три схемы (3.15), (3.16) и (3.17) эквивалентны. Отметим, что в этих схемах предполагается вычисление матрицы Якоби на каждом шаге, поскольку в противном случае не обеспечивается 2-й порядок.

Значительно более эффективной оказалась E-схема. В отличие от CROS, она сохраняет порядок при замораживании матрицы Якоби и одном вычислении правой части на каждом шаге. Кроме этого, она не требует приведения системы ОДУ к автономной форме. Рассмотрим такую схему.

На i-й итерации для 1-й стадии принимаем $\mathbf{Y}^{i-1} = \mathbf{y}_n$, $\mathbf{F}^{i-1} = \mathbf{f}_n$, тогда эта стадия не требует вычислений правой части. Эксперименты показали, что такой прием позволяет существенно сократить вычислительные затраты без снижения точности. Прогноз 2-й стадии выполняется так же, как и прогноз метода трапеций. Далее выполняем две или три итерации, 1-я из которых не требует вычисления правой части:

$$\mathbf{M}=\begin{bmatrix}\mathbf{I}-0.5h\mathbf{J} & 0.5h\mathbf{J}\\ -0.5h\mathbf{J} & \mathbf{I}-0.5h\mathbf{J}\end{bmatrix},$$

$$\mathbf{M}\begin{bmatrix}\Delta\mathbf{Y}\\ \Delta\mathbf{y}^1_{n+1}-\Delta\mathbf{y}^0_{n+1}\end{bmatrix}=\begin{bmatrix}0.5h(\mathbf{f}_n-\mathbf{f}^0_{n+1})\\ 0.5h(\mathbf{f}_n+\mathbf{f}^0_{n+1})-\Delta\mathbf{y}^0_{n+1}\end{bmatrix},\quad \mathbf{y}^1_{n+1}=\mathbf{y}_n+\Delta\mathbf{y}^1_{n+1},$$

$$\mathbf{M}\begin{bmatrix}\Delta\mathbf{Y}\\ \Delta\mathbf{y}^i_{n+1}-\Delta\mathbf{y}^{i-1}_{n+1}\end{bmatrix}=\begin{bmatrix}0.5h\left(\mathbf{f}_n-\mathbf{f}(t_n+h,\mathbf{y}^{i-1}_{n+1})\right)\\ 0.5h\left(\mathbf{f}_n+\mathbf{f}(t_n+h,\mathbf{y}^{i-1}_{n+1})\right)-\Delta\mathbf{y}^{i-1}_{n+1}\end{bmatrix},\quad \mathbf{y}^i_{n+1}=\mathbf{y}_n+\Delta\mathbf{y}^i_{n+1},\ i=2,N.$$

Результат выполнения шага принимаем в виде:

$\mathbf{y}_{n+1}=\mathbf{y}^N_{n+1}$, $\mathbf{Y}=\mathbf{y}_n+\Delta\mathbf{Y}$, $\mathbf{f}_{n+1}=h^{-1}(\mathbf{y}_{n+1}-\mathbf{Y})$.

При решении системы алгебраических уравнений вида $\mathbf{M}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}$ можно воспользоваться тем фактом, что она эквивалентна системе с комплексными коэффициентами $(\mathbf{I}-\alpha h\mathbf{J})(\mathbf{x}+j\mathbf{y})=\mathbf{a}+j\mathbf{b}$, где $\alpha=(1+j)/2$, что позволяет примерно в 2 раза сократить количество операций. Определение числа итераций, обновление матрицы Якоби и управление размером шага выполняем точно так же, как и при реализации метода трапеций.

3.6. Численные эксперименты

На основе рассмотренных E-схем были написаны компьютерные программы, которые обозначим через TRE (метод трапеций), TB2E (метод TR-BDF2) и Lob2E (метод Лобатто IIIC 2-го порядка). Приведем результаты решения 4 тестов из [75] (VDPOL, ROBER, HIRES и BRUSS). Для сравнения приводим также результаты решателей ode23t и ode23tb системы MATLAB, реализующих, соответственно, методы трапеций и TR-BDF2. Допустимую ошибку обозначим через *Tol*. Для всех задач задаем допустимую относительную ошибку *Rtol* = *Tol*, а абсолютную ошибку принимаем в виде *Atol* = *Tol* для VDPOL и BRUSS, *Atol* = $10^{-12}\times$ *Tol* для ROBER и *Atol* = $10^{-4}\times$ *Tol* для HIRES. Начальный размер шага принимаем в виде h_0 = *Tol* для BRUSS и $h_0=10^{-2}\times$ *Tol* для остальных задач. Точность решения оцениваем по формуле

$$scd=-\lg\left(\max_i\left(\frac{|y_i-\tilde{y}_i|}{|y_i|}\right)\right), \tag{3.18}$$

где y_i – точное, а $\tilde{y}_i$ – численное решение по i-й компоненте в конечной точке интервала интегрирования. Величина *scd* (significant correct digits) является оценкой относительной ошибки и показывает число правильных значащих цифр среди компонент численного решения. Поскольку задачи не имеют аналитического решения, в качестве точного принимаем численное решение, полученное при очень малом значении *Tol*. Обозначим также: *Nf* – число вычислений правой части, *Nstp* – число успешных шагов, *Nbad* – число отброшенных шагов,

NJ – число вычислений матрицы Якоби, *NLU* – число LU-факторизаций, *Nsol* – число решений алгебраических уравнений с факторизованной матрицей коэффициентов.

VDPOL – осциллятор Ван-дер-Поля:

$y_1' = y_2, \quad y_2' = 10^6((1 - y_1^2)y_2 - y_1),$

$y_1(0) = 2, \quad y_2(0) = 0, \quad 0 \le t \le 2.$

Точное решение в конце интервала:

$\mathbf{y}(2) = (1.70616773217047, -0.89280970102481)^T.$

Эта задача является наиболее популярной среди жестких тестов, поэтому в табл. 3.3 приводим наиболее полную информацию о ее численном решении. Чтобы посмотреть соответствие ошибки задаваемому допуску, для решателей TB2E и ode23tb приведены результаты в широком диапазоне изменения *Tol*. Решатель TB2E показывает хорошее соответствие между задаваемой и фактической ошибкой, тогда как точность ode23tb при $Tol < 10^{-2}$ ниже заданной. При одинаковой фактической ошибке TB2E требует заметно меньших вычислительных затрат (сравним оба решателя при $Tol = 10^{-2}$; TB2E при $Tol = 10^{-4}$ и ode23tb при $Tol = 10^{-6}$). Приводим также показатель экономичности итераций $nf = Nf/(r \times Nstp)$, где r – число неявных стадий для TR2E и TB2E и $r = 1$ для Lob2E. Число *nf* показывает среднее число вычислений правой части на одной неявной стадии. При $Tol < 10^{-3}$ методы 2-го порядка менее эффективны, чем методы более высоких порядков, поэтому результаты решения других задач приводим только для значений $Tol = 10^{-2}, 10^{-3}$.

Таблица 3.3. Результаты решения задачи VDPOL

Решатель	*Tol*	*scd*	*Nf*	*Nstp*(*Nbad*)	*NJ*	*NLU*	*Nsol*	*nf*
TB2E	10^{-2}	2.07	692	295(12)	13	143	1305	1.13
	10^{-3}	3.14	2047	933(3)	13	191	3918	1.09
	10^{-4}	4.08	6450	2966(2)	11	227	12385	1.09
	10^{-5}	5.01	20279	9464(2)	11	258	39210	1.07
	10^{-6}	6.01	63902	29931(1)	9	272	123765	1.07
ode23tb	10^{-2}	2.06	1035	193(62)	27	141	1247	2.03
	10^{-3}	2.23	1857	384(63)	31	182	2260	2.08
	10^{-4}	2.72	3550	802(64)	31	200	4385	2.05
	10^{-5}	3.39	6675	1709(54)	33	229	8405	1.89
	10^{-6}	4.09	13436	3755(47)	35	248	17 203	1.77
TR2E	10^{-2}	2.40	479	353(13)	15	144	844	1.30
	10^{-3}	3.12	1414	1150(3)	13	207	2566	1.23
Lob2E	10^{-2}	2.56	488	361(16)	15	145	846	1.29
	10^{-3}	3.49	1445	1178(3)	13	206	2625	1.22
ode23t	10^{-2}	0.73	677	239(65)	23	148	676	2.23
	10^{-3}	2.25	1345	520(62)	24	197	1344	2.31

ROBER – реакция Робертсона, описывается уравнениями:

$y_1' = -0.04y_1 + 10^4 y_2 y_3, \quad y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3\cdot 10^7 y_2^2,$

$y_3' = 3\cdot 10^7 y_2^2, \quad y_1(0) = 1, \quad y_2(0) = 0, \quad y_3(0) = 0, \quad 0 \le t \le 10^{11}.$

Точное решение при $t = 10^{11}$:

$\mathbf{y} = (2.08334014970126 \times 10^{-8},\ 8.33336077033471 \times 10^{-14},\ 0.999999979166505)^{\mathrm{T}}.$

Эта задача – самая жесткая из рассмотренных (мера жесткости $M_{ж} = 10^{15}$). Трудность ее решения обусловлена также большим интервалом интегрирования, на протяжении которого размер шага увеличивается более чем на 14 порядков. Результаты приведены в табл. 3.4. Решатели TRE и ode23t, реализующие метод трапеций, оказались малопригодными для решения этой задачи. Этот факт можно объяснить тем, что метод трапеций не является L-устойчивым.

Таблица 3.4. Результаты решения задачи ROBER

Решатель	*Tol* = 10^{-2}				*Tol* = 10^{-3}			
	scd	***Nf***	***NJ***	***NLU***	***scd***	***Nf***	***NJ***	***NLU***
TR2E	1.05	31 322	8	1 346	2.10	6 460	9	1 742
TB2E	2.80	521	10	137	4.80	1 575	9	177
Lob2E	1.64	382	11	154	2.26	1 079	10	187
ode23t	0.14	95 845	8	803	1.99	36 540	8	860
ode23tb	1.84	573	13	116	2.43	1 150	12	144

HIRES – модель химической реакции с участием восьми реагентов:

$y_1' = -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007,$

$y_2' = 1.71y_1 - 8.75y_2,$

$y_3' = -10.03y_3 + 0.43y_4 + 0.035y_5,$

$y_4' = 8.32y_2 + 1.71y_3 - 1.12y_4,$

$y_5' = -1.745y_5 + 0.43y_6 + 0.43y_7,$

$y_6' = -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7,$

$y_7' = 280y_6y_8 - 1.81y_7,$

$y_8' = -y_7',$

$\mathbf{y}(0) = (1, 0, 0, 0, 0, 0, 0, 0.0057)^{\mathrm{T}}, \quad 0 \le t \le 321.8122.$

Точное решение в конце интервала:

$$\begin{aligned}\mathbf{y} = (&0.7371312573325668 \times 10^{-3},\ 0.1442485726316185 \times 10^{-3},\\ &0.5888729740967575 \times 10^{-4},\ 0.1175651343283149 \times 10^{-2},\\ &0.2386356198831331 \times 10^{-2},\ 0.6238968252742796 \times 10^{-2},\\ &0.2849998395185769 \times 10^{-2},\ 0.2850001604814231 \times 10^{-2})^{\mathrm{T}}.\end{aligned}$$

Интервал интегрирования выбран таким, что в конце интервала переменные y_7 и y_8 изменяются очень быстро. В связи с этим не все решатели позволяют

обеспечить заданную точность. Результаты решения этой задачи приведены в табл. 3.5.

Таблица 3.5. Результаты решения задачи HIRES

Решатель	*Tol* = 10^{-2}				*Tol* = 10^{-3}			
	scd	*Nf*	*NJ*	*NLU*	*scd*	*Nf*	*NJ*	*NLU*
TR2E	2.79	179	9	57	3.00	560	8	72
TB2E	2.17	250	9	53	3.13	805	8	71
Lob2E	2.20	184	10	59	2.82	555	8	74
ode23t	1.48	233	12	51	2.18	402	9	67
ode23tb	1.56	278	12	53	1.88	539	14	62

BRUSS – система ОДУ, полученная в результате дискретизации уравнений в частных производных, описывающих химическую реакцию с диффузией:

$$u_i' = 1 + u_i^2 v_i - 4u_i + \frac{\alpha}{\Delta x^2}(u_{i-1} - 2u_i + u_{i+1}),$$

$$v_i' = 3u_i - u_i^2 v_i + \frac{\alpha}{\Delta x^2}(v_{i-1} - 2v_i + v_{i+1}),$$

$u_0 = u_{N+1} = 1$, $v_0 = v_{N+1} = 3$, $\alpha = 1/50$, $\Delta x = 1/(N+1)$,

$u_i(0) = 1 + \sin(2\pi x_i)$, $v_i(0) = 3$, $x_i = i\Delta x$, $i = 1,\ldots,N$, $0 \le t \le 10$.

Задаем $N = 500$, в результате получаем большую жесткую задачу ($M_ж = 2 \times 10^5$), содержащую 1000 переменных. Результаты решения приведены в табл. 3.6.

Таблица 3.6. Результаты решения задачи BRUSS

Решатель	*Tol* = 10^{-2}				*Tol* = 10^{-3}			
	scd	*Nf*	*NJ*	*NLU*	*scd*	*Nf*	*NJ*	*NLU*
TR2E	1.90	83	1	21	3.14	182	1	31
TB2E	2.14	103	1	23	3.27	232	1	29
Lob2E	1.45	84	1	23	2.52	182	1	32
ode23t	1.90	122	7	26	2.72	192	2	28
ode23tb	2.07	165	4	22	2.64	279	2	26

Приведенные в табл. 3.3–3.6 результаты показывают преимущество E-схем, реализованных в решателях TR2E, TB2E и Lob2E, по сравнению со схемами, реализованными в решателях ode23t и ode23tb системы MATLAB. E-схемы почти всегда обеспечивают заданную точность расчета, тогда как решатели MATLAB часто дают результаты с точностью ниже заданной (одной из причин является 3-й порядок вложенной формулы решателей MATLAB). Кроме этого, при одинаковой фактической точности E-схемы требуют заметно меньших вычислительных затрат.

3.7. Методы типа Розенброка

Выше уже было показано, что простейшую схему реализации метода Лобатто IIA 2-го порядка можно представить как комплексную схему Розенброка. Попробуем применить такой же подход к реализации диагонально-неявных методов. Один шаг s-стадийного метода DIRK для решения автономной системы $\mathbf{y}' = \mathbf{f}(\mathbf{y})$, $\mathbf{y}(t_0) = \mathbf{y}_0$ задается формулой

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\sum_{i=1}^{s} b_i \mathbf{F}_i, \tag{3.19}$$

где векторы $\mathbf{F}_i$ определяются из уравнений

$$\mathbf{F}_i = \mathbf{f}\left(\mathbf{y}_n + h\sum_{j=1}^{i} a_{ij}\mathbf{F}_j\right), \quad i = 1,\ldots,s.$$

Ограничив численное решение этих уравнений одной итерацией метода Ньютона и выбрав в качестве начальных приближений векторы

$$\mathbf{F}_i^0 = \sum_{j=1}^{i-1} \beta_{ij}\mathbf{F}_j, \tag{3.20}$$

т. е. используя схему с прогнозом для производных, получим формулы методов типа Розенброка:

$$(\mathbf{I} - ha_{ii}\mathbf{J})(\mathbf{F}_i - \mathbf{F}_i^0) = \mathbf{f}\left(\mathbf{y}_n + h\sum_{j=1}^{i-1}\alpha_{ij}\mathbf{F}_j\right) - \mathbf{F}_i^0, \quad \alpha_{ij} = a_{ij} + a_{ii}\beta_{ij}. \tag{3.21}$$

Обычно методы Розенброка представляют в виде:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \sum_{i=1}^{s} b_i \mathbf{k}_i, \quad \mathbf{k}_i = h\mathbf{f}\left(\mathbf{y}_n + \sum_{j=1}^{i-1}\alpha_{ij}\mathbf{k}_j\right) + h\mathbf{J}\sum_{j=1}^{i}\gamma_{ij}\mathbf{k}_j, \quad i = 1,\ldots,s. \tag{3.22}$$

Такая форма записи удобна для теоретического исследования, но неудобна для реализации, поскольку требует на каждой стадии не только решения линейной системы с матрицей $\mathbf{I} - h\gamma_{ii}\mathbf{J}$, но и умножения матрицы $\mathbf{J}$ на вектор. Последнего можно избежать, если преобразовать формулу вычисления $\mathbf{k}_i$ (3.22) к виду:

$$(\mathbf{I} - h\gamma_{ii}\mathbf{J})(\mathbf{k}_i + \mathbf{u}_i) = h\mathbf{f}\left(\mathbf{y}_n + \sum_{j=1}^{i-1}\alpha_{ij}\mathbf{k}_j\right) + \mathbf{u}_i, \quad \mathbf{u}_i = \sum_{j=1}^{i-1}\frac{\gamma_{ij}}{\gamma_{ii}}\mathbf{k}_j. \tag{3.23}$$

Если принять $\mathbf{k}_i = h\mathbf{F}_i$, $\gamma_{ij} = -\gamma_{ii}\beta_{ij}$, $\gamma_{ii} = a_{ii}$, то из (3.23) получим (3.21). Таким образом, формулы (3.19)–(3.21) действительно описывают методы Розенброка, которые можно рассматривать как простейшие D-схемы реализации диагонально-неявных методов Рунге–Кутты.

Приведем примеры таких схем 2-го порядка. За основу возьмем однопараметрическое семейство L-устойчивых методов SDIRK 2-го порядка

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h[(1-b_2)\mathbf{F}_1 + b_2\mathbf{F}_2],\quad \mathbf{F}_1 = \mathbf{f}(\mathbf{y}_n + h\gamma\mathbf{F}_1),$$
$$\mathbf{F}_2 = \mathbf{f}(\mathbf{y}_n + h(a_{21}\mathbf{F}_1 + \gamma\mathbf{F}_2)),\quad \gamma = 1 - \sqrt{2}/2,\quad b_2 = (1/2 - \gamma)/a_{21}, \tag{3.24}$$

которое порождает двухпараметрическое семейство схем Розенброка:

$$(\mathbf{I} - h\gamma\mathbf{J})\mathbf{F}_1 = \mathbf{f}(\mathbf{y}_n),\quad (\mathbf{I} - h\gamma\mathbf{J})(\mathbf{F}_2 - \beta_{21}\mathbf{F}_1) = \mathbf{f}(\mathbf{y}_n + h\alpha_{21}\mathbf{F}_1) - \beta_{21}\mathbf{F}_1,$$
$$\mathbf{y}_{n+1} = \mathbf{y}_n + h[(1-b_2)\mathbf{F}_1 + b_2\mathbf{F}_2],\ \gamma = 1 - \sqrt{2}/2,\quad b_2 = \frac{1-2\gamma}{2(\alpha_{21} - \gamma\beta_{21})}. \tag{3.25}$$

В данном случае выполнение условий 2-го порядка исходного метода (3.24) привело к выполнению условий 2-го порядка схем (3.25):

$$b_1 + b_2 = 1,\quad \gamma + b_2(\alpha_{21} - \gamma\beta_{21}) = 1/2. \tag{3.26}$$

Но для схем более высоких порядков выполнение условий порядка метода Рунге–Кутты не гарантирует выполнения аналогичных условий для построенной на его основе схемы Розенброка (для схем до 5-го порядка такие условия приведены в [75]).

Рассмотрим конкретные схемы семейства (3.25). Задав $\alpha_{21} = 0$, получим схему, которая при любом ненулевом β_{21} приводится к виду:

$$(\mathbf{I} - h\gamma\mathbf{J})\mathbf{F}_1 = \mathbf{f}(\mathbf{y}_n),\quad (\mathbf{I} - h\gamma\mathbf{J})\mathbf{F}_2 = \mathbf{F}_1,\quad \mathbf{y}_{n+1} = \mathbf{y}_n + h[\gamma\mathbf{F}_1 + (1-\gamma)\mathbf{F}_2]. \tag{3.27}$$

Эта схема принадлежит к классу (m, k)-методов [38, 40]. На каждом шаге такого метода выполняются m решений алгебраических уравнений и k вычислений правой части, причем m может быть больше k (в обычных методах Розенброка $m = k$). Таким образом, схема (3.27) является (2, 1)-методом и требует всего одного вычисления правой части на шаге интегрирования.

Коэффициент β_{21} задает формулу прогноза (3.20) для 2-й стадии. Наиболее точный прогноз получаем при $\beta_{21} = 1$, поэтому в следующих двух схемах принимаем именно такое значение. Недостатком многих схем Розенброка является то, что они обеспечивают заданный порядок только при вычислении матрицы Якоби на каждом шаге. Чтобы сохранялся порядок при неточной матрице Якоби, должны выполняться дополнительные условия, а схемы, удовлетворяющие этим условиям, получили название W-методов [75]. Для двухстадийных схем 2-го порядка к условиям (3.26) нужно добавить условие

$$1 - b_2\beta_{21} = 0, \tag{3.28}$$

откуда при $\beta_{21} = 1$ получаем $\alpha_{21} = 0.5$, $b_2 = 1$. Соответствующая схема имеет вид:

$$(\mathbf{I} - h\gamma\mathbf{J})\mathbf{F}_1 = \mathbf{f}(\mathbf{y}_n),\quad (\mathbf{I} - h\gamma\mathbf{J})(\mathbf{F}_2 - \mathbf{F}_1) = \mathbf{f}(\mathbf{y}_n + 0.5h\mathbf{F}_1) - \mathbf{F}_1,\quad \mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{F}_2. \tag{3.29}$$

На основе этой схемы построен решатель ode23s системы MATLAB [141].

Приведенные схемы (3.27) и (3.29) не обладают свойством жесткой точности, что приводит к заметному снижению точности при повышении жесткости задачи. Условия жесткой точности [75] для схемы (3.25) запишутся в виде:

$$b_1 = \alpha_{21} - \gamma\beta_{21},\quad b_2 = \gamma,\quad \alpha_{21} = 1, \tag{3.30}$$

откуда $\beta_{21} = 1$, $b_1 = 1 - \gamma$, и полученная схема имеет вид:

$$(\mathbf{I} - h\gamma\mathbf{J})\mathbf{F}_1 = \mathbf{f}(\mathbf{y}_n), \quad (\mathbf{I} - h\gamma\mathbf{J})(\mathbf{F}_2 - \mathbf{F}_1) = \mathbf{f}(\mathbf{y}_n + h\mathbf{F}_1) - \mathbf{F}_1,$$
$$\mathbf{y}_{n+1} = \mathbf{y}_n + h[(1-\gamma)\mathbf{F}_1 + \gamma\mathbf{F}_2]. \tag{3.31}$$

Эта схема является жесткоточной, но не принадлежит к W-методам, поскольку условие (3.28) не выполняется.

Обозначим схему (3.27) через (2, 1), схему (3.29) – через W2 и схему (3.31) – через SA2 (Stiffly Accurate). Все три схемы имеют одинаковую функцию устойчивости, поэтому при решении линейных автономных задач с постоянным размером шага они показывают одинаковые результаты. Положение существенно изменяется при решении нелинейных задач. Покажем это на примере задачи Капса (3.1). Ошибки при $h = 1/30$ для этих схем, а также для схем, рассмотренных ниже, приведены в табл. 3.7. Уже из этих результатов видно, что эти три схемы малопригодны для расчета с замороженной матрицей Якоби ($NJ = 1$). Это подтвердили также и оценки порядка, полученные путем сравнения ошибок при размерах шага h и $h/2$. При расчете с замороженной матрицей $\mathbf{J}$ схема W2 действительно сохраняет порядок, но только при малой жесткости. При больших μ порядок сохраняет схема SA2, но это обусловлено исключительно особенностью решаемой задачи, которая для жесткоточных методов вырождается в линейное автономное уравнение при $\mu \to \infty$. Во всех остальных случаях порядок снижается до 1-го.

Таблица 3.7. Ошибки решения задачи Капса схемами Розенброка

Схема	*NJ*	Ошибка		
		$\mu = 1$	$\mu = 10^3$	$\mu = 10^6$
(2,1)	30	1.18×10^{-4}	1.01×10^{-3}	1.08×10^{-3}
	1	3.25×10^{-3}	1.45×10^{-2}	1.67×10^{-2}
W2	30	7.77×10^{-5}	1.72×10^{-4}	1.48×10^{-4}
	1	6.09×10^{-5}	8.61×10^{-3}	1.17×10^{-2}
SA2	30	7.22×10^{-5}	2.97×10^{-5}	1.98×10^{-5}
	1	1.31×10^{-3}	8.19×10^{-4}	1.01×10^{-4}
TB2R	30	6.27×10^{-5}	2.22×10^{-5}	2.21×10^{-5}
	1	6.09×10^{-5}	1.76×10^{-5}	2.29×10^{-5}
CROS	30	2.40×10^{-4}	9.98×10^{-4}	1.06×10^{-3}
	1	3.44×10^{-3}	1.61×10^{-2}	1.66×10^{-2}
MCROS	30	2.83×10^{-4}	1.08×10^{-4}	9.78×10^{-5}
	1	3.00×10^{-4}	3.56×10^{-4}	3.56×10^{-4}

Чтобы снижения порядка не происходило как для нежестких, так и для жестких задач, схема должна принадлежать к W-методам и быть жесткоточной. При $p = s = 2$ для схем Розенброка вида (3.25) этого добиться невозможно, поскольку уравнения (3.26), (3.28), (3.30) не имеют решения. Еще один недостаток традиционных схем – они требуют приведения уравнений к автономному виду, что означает использование в схеме производных $\partial\mathbf{f}/\partial t$ [75]. Можно изба-

виться от этих недостатков, если добавить явную первую стадию (подобно методам ESDIRK). В этом случае имеем $\gamma_{11} = 0$, тогда 1-я стадия запишется в виде $\mathbf{F}_1 = \mathbf{f}(\mathbf{y}_n)$. Такие схемы были предложены в [140].

Приведем схему с явной первой стадией, построенную на основе метода TR-BDF2 (3.13). Для неавтономной задачи она запишется в виде:

$$
\begin{aligned}
&(\mathbf{I}-h\gamma\mathbf{J})(\mathbf{F}-\mathbf{f}_n) = \mathbf{f}(t_n+2h\gamma,\, y_n+2h\gamma\mathbf{f}_n)-\mathbf{f}_n, \quad \mathbf{f}^0_{n+1} = \beta_{31}\mathbf{f}_n+\beta_{32}\mathbf{F},\\
&(\mathbf{I}-h\gamma\mathbf{J})(\mathbf{f}_{n+1}-\mathbf{f}^0_{n+1}) = \mathbf{f}\left(t_n+h,\, \mathbf{y}_n+h(\alpha_{31}\mathbf{f}_n+\alpha_{32}\mathbf{F})\right)-\mathbf{f}^0_{n+1},\\
&\mathbf{y}_{n+1} = \mathbf{y}_n+h\frac{1-\gamma}{2}(\mathbf{f}_n+\mathbf{F})+h\gamma\mathbf{f}_{n+1},\\
&\gamma = 1-\frac{\sqrt{2}}{2}, \quad \beta_{32} = 1+\frac{\sqrt{2}}{2}, \quad \alpha_{32} = \frac{1-\gamma}{2}+\gamma\beta_{32}, \quad \alpha_{31} = 1-\alpha_{32}, \quad \beta_{31} = 1-\beta_{32}.
\end{aligned}
\tag{3.32}
$$

Коэффициент β_{32} является свободным и выбран из условия линейной экстраполяции прогноза $\mathbf{f}^0_{n+1}$ по значениям $\mathbf{f}_n$ и $\mathbf{F}$ (в [140] $\beta_{32} = -\gamma_{32}/\gamma = 3/2 + \sqrt{2}$). В отличие от схем, предложенных в [140], мы не вычисляем $\mathbf{f}(\mathbf{y}_n)$ на явной стадии, а используем значение $\mathbf{f}_n$, полученное на предыдущем шаге, т. е. используем сглаженную первую стадию [112]. Это позволяет сэкономить одно вычисление функции и делает схему более эффективной. Например, при решении задачи VDPOL такой прием позволил сократить вычислительные затраты примерно в 3 раза, а решение задачи ROBER вообще невозможно было получить на всем интервале без использования этого приема.

Обозначим схему (3.32) через TB2R (TR-BDF2-Rosenbrock). Результаты решения задачи Капса этой схемой приведены в табл. 3.7. В отличие от рассмотренных ранее схем Розенброка, она сохраняет точность и порядок при замораживании матрицы Якоби.

Чтобы сформировать критерий обновления матрицы Якоби, необходимо оценить скорость сходимости итераций, но это невозможно сделать только на основании данных, полученных схемой (3.32). Поэтому в схемах с замороженной матрицей необходимо добавить еще одну итерацию, которую не обязательно выполнять на каждом шаге. При построении схем мы исходили из того, что вычисление матрицы Якоби и последующие операции с ней вносят наибольший вклад в вычислительные затраты. Но существуют задачи (невысокого порядка либо с матрицей Якоби специальной структуры), для которых вычисление якобиана по трудоемкости сравнимо с вычислением правой части. Для этих задач вполне оправдано применение схем Розенброка с вычислением матрицы Якоби на каждом шаге. Такие схемы можно также применять при решении линейных задач или при малом изменении якобиана на траектории решения. В этом случае матрицу Якоби можно вычислять только один раз в начале интервала (решатель должен иметь соответствующую опцию).

В табл. 3.8 приведены результаты решения жестких задач решателем TB2R, реализующим схему (3.32). В отличие от TB2E, в нем отсутствует контроль сходимости итераций, но можно выбрать режим вычисления матрицы Якоби (на

каждом шаге либо один раз на всем интервале). Приведенные результаты показывают, что схема вполне работоспособна и может быть использована для решения жестких задач.

Таблица 3.8. Результаты решателя TB2R

Задача	*Tol* = 10^{-2}				*Tol* = 10^{-3}			
	scd	*Nf*	*NJ*	*NLU*	*scd*	*Nf*	*NJ*	*NLU*
VDPOL	2.08	625	293	312	3.11	1923	957	961
ROBER	2.37	481	237	240	3.31	1461	727	730
HIRES	2.18	217	108	108	3.11	685	342	342
BRUSS	1.53	73	1	20	3.18	225	1	32

В заключение этого раздела построим схему Розенброка на основе метода Лобатто IIIC 2-го порядка (3.14). Примем прогноз в виде

$$\mathbf{F}^0 = \mathbf{f}^0_{n+1} = \mathbf{f}_n, \quad \mathbf{Y}^0 = \mathbf{y}_n, \quad \mathbf{y}^0_{n+1} = \mathbf{y}_n + h\mathbf{f}_n$$

и выполним одну итерацию относительно производных:

$$\begin{bmatrix} \mathbf{I} - 0.5h\mathbf{J} & 0.5h\mathbf{J} \\ -0.5h\mathbf{J} & \mathbf{I} - 0.5h\mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{F} - \mathbf{f}_n \\ \mathbf{f}_{n+1} - \mathbf{f}_n \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{f}(\mathbf{y}_n + h\mathbf{f}_n) - \mathbf{f}_n \end{bmatrix}, \quad \mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}(\mathbf{F} + \mathbf{f}_{n+1}).$$

Полученную схему можно записать в виде:

$$\Delta\mathbf{f} = (\mathbf{I} - h\alpha\mathbf{J})^{-1}[\mathbf{f}(\mathbf{y}_n + h\mathbf{f}_n) - \mathbf{f}_n], \quad \mathbf{F} = \mathbf{f}_n - \mathrm{Im}(\Delta\mathbf{f}), \quad \mathbf{f}_{n+1} = \mathbf{f}_n + \mathrm{Re}(\Delta\mathbf{f}),$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}[\mathbf{F} + \mathbf{f}_n], \quad \alpha = \frac{1+j}{2},$$

где j – мнимая единица. Эта схема является модификацией комплексной схемы CROS, поэтому назовем ее MCROS. В отличие от CROS, она является жесткоточной и сохраняет порядок при неточной матрице Якоби. В табл. 3.7 приведены ошибки решения задачи Капса схемами CROS и MCROS при вычислении якобиана на каждом шаге (NJ = 30) и одноразовом вычислении (NJ = 1).

3.8. Схемы решения дифференциально-алгебраических уравнений

Рассмотрим возможные схемы решения системы ДАУ

$$\begin{aligned} &\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \\ &\mathbf{0} = \mathbf{g}(\mathbf{y}, \mathbf{z}), \quad \mathbf{z}(t_0) = \mathbf{z}_0 \end{aligned} \tag{3.33}$$

на примере метода трапеций, один шаг которого сводится к решению алгебраической системы

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}\left(\mathbf{f}_n + \mathbf{f}(\mathbf{y}_{n+1}, \mathbf{z}_{n+1})\right),$$

$$\mathbf{0} = \mathbf{g}(\mathbf{y}_{n+1}, \mathbf{z}_{n+1}).$$

Обозначим через $\mathbf{f}_y$, $\mathbf{f}_z$, $\mathbf{g}_y$ и $\mathbf{g}_z$ соответствующие матрицы частных производных, вычисленные в некоторой точке численного решения (предполагаем, что эти матрицы могут сохраняться в течение нескольких шагов). Тогда итерации модифицированного метода Ньютона запишутся в виде:

$$\begin{bmatrix} \mathbf{I}-\frac{h}{2}\mathbf{f}_y & -\frac{h}{2}\mathbf{f}_z \\ -\mathbf{g}_y & -\mathbf{g}_z \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y}^i_{n+1}-\Delta\mathbf{y}^{i-1}_{n+1} \\ \Delta\mathbf{z}^i_{n+1}-\Delta\mathbf{z}^{i-1}_{n+1} \end{bmatrix} = \begin{bmatrix} \frac{h}{2}(\mathbf{f}_n+\mathbf{f}^{i-1}_{n+1})-\Delta\mathbf{y}^{i-1}_{n+1} \\ \mathbf{g}^{i-1}_{n+1} \end{bmatrix}, \tag{3.34}$$

$$\mathbf{y}^i_{n+1}=\mathbf{y}_n+\Delta\mathbf{y}^i_{n+1},\quad \mathbf{z}^i_{n+1}=\mathbf{z}_n+\Delta\mathbf{z}^i_{n+1},\quad \mathbf{f}^i_{n+1}=\mathbf{f}(\mathbf{y}^i_{n+1},\mathbf{z}^i_{n+1}),$$

$$\mathbf{g}^i_{n+1}=\mathbf{g}(\mathbf{y}^i_{n+1},\mathbf{z}^i_{n+1}),$$

а финальные значения после выполнения N итераций принимаем в виде:

$$\mathbf{y}_{n+1}=\mathbf{y}^N_{n+1},\quad \mathbf{z}_{n+1}=\mathbf{z}^N_{n+1},\quad \mathbf{f}_{n+1}=\frac{2}{h}\Delta\mathbf{y}^N_{n+1}-\mathbf{f}_n,\quad \mathbf{z}'_{n+1}=\frac{2}{h}\Delta\mathbf{z}^N_{n+1}-\mathbf{z}'_n$$

(значения $\mathbf{z}'$ применяются только в схеме с прогнозом для производных). Будем использовать двухшаговый прогноз, тогда в зависимости от схемы реализации начальные значения для итераций вычисляем по следующим формулам, где $w = h/(t_n - t_{n-1})$, а на первом шаге принимаем $w = 0$.

Стандартная схема:

$$\Delta\mathbf{y}^0_{n+1}=w\Delta\mathbf{y}_n,\quad \Delta\mathbf{z}^0_{n+1}=w\Delta\mathbf{z}_n,\quad \mathbf{f}^0_{n+1}=\mathbf{f}(\mathbf{y}_n+\Delta\mathbf{y}^0_{n+1},\mathbf{z}_n+\Delta\mathbf{z}^0_{n+1}),$$

$$\mathbf{g}^0_{n+1}=\mathbf{g}(\mathbf{y}_n+\Delta\mathbf{y}^0_{n+1},\mathbf{z}_n+\Delta\mathbf{z}^0_{n+1}).$$

Схема с прогнозом для производных:

$$\Delta\mathbf{y}^0_{n+1}=h\left[\mathbf{f}_n+\frac{w}{2}(\mathbf{f}_n-\mathbf{f}_{n-1})\right],\quad \Delta\mathbf{z}^0_{n+1}=h\left[\mathbf{z}'_n+\frac{w}{2}(\mathbf{z}'_n-\mathbf{z}'_{n-1})\right],$$

$$\mathbf{f}^0_{n+1}=\mathbf{f}(\mathbf{y}_n+\Delta\mathbf{y}^0_{n+1},\mathbf{z}_n+\Delta\mathbf{z}^0_{n+1}),\quad \mathbf{g}^0_{n+1}=\mathbf{g}(\mathbf{y}_n+\Delta\mathbf{y}^0_{n+1},\mathbf{z}_n+\Delta\mathbf{z}^0_{n+1})$$

(если значение $\mathbf{z}'_0$ неизвестно, то принимаем $\mathbf{z}'_0 = \mathbf{0}$).

Экономичная схема:

$$\Delta\mathbf{y}^0_{n+1}=w\Delta\mathbf{y}_n,\quad \Delta\mathbf{z}^0_{n+1}=w\Delta\mathbf{z}_n,\quad \mathbf{f}^0_{n+1}=\mathbf{f}_n+w(\mathbf{f}_n-\mathbf{f}_{n-1}),\quad \mathbf{g}^0_{n+1}=\mathbf{0}.$$

Рассмотренные здесь решатели TRE, TB2E, Lob2E и TB2R, а также решатели ode23t и ode23tb системы MATLAB являются жесткоточными, поэтому они без особых затруднений решают системы ДАУ индекса 1. Но при решении задач высших индексов могут возникнуть трудности, вызванные высоким индексом задачи.

Рассмотрим задачу индекса 2

$$y'_1=-3y_1+y_2^2,\quad y'_2=y_1-y_2(1+z),\quad 0=y_2^2-y_1,$$
$$y_1(0)=y_2(0)=z(0)=1,\quad 0\le t\le 1 \tag{3.35}$$

с точным решением $y_1(t) = \exp(-2t)$, $y_2(t) = z(t) = \exp(-t)$. Для решения этой задачи можно применять итерации вида (3.34) либо привести систему ДАУ (3.33) к системе ОДУ, заменив алгебраическое уравнение дифференциальным $z' = \varepsilon^{-1}(y_2^2 - y_1)$ с малым значением ε. Итерации (3.34) реализуют метод ε-вложения, поэтому при достаточно малом ε оба подхода должны давать одинаковый результат. При $\varepsilon \leq 10^{-10}$ и задаваемой нами точности это было действительно так. Однако в общем случае такой способ приведения ДАУ к системе ОДУ следует использовать с осторожностью, поскольку полученная система ОДУ может оказаться неустойчивой. Этот способ понадобился для того, чтобы сравнить наши решатели с решателями MATLAB, в которых реализован метод пространства состояний, не позволяющий решать системы ДАУ высших индексов. Для контроля корректности такого подхода мы сравнивали результаты, полученные при $\varepsilon = 10^{-10}$ и $\varepsilon = 10^{-20}$, и убеждались, что они одинаковы.

Результаты решения (максимальные значения евклидовой нормы ошибки на всем интервале и основные вычислительные затраты) при *Atol* = *Rtol* = *Tol* и $h_0 = Tol$ приведены в табл. 3.9. Лучшие результаты показывают схемы TB2E и TB2R, реализующие жесткоточный *L*-устойчивый метод 2-го стадийного порядка TR-BDF2. Решатель Lob2E при тех же затратах заметно менее точен. Это вызвано тем, что реализованный в нем метод имеет только 1-й стадийный порядок, что приводит к снижению реального порядка до 1-го при решении задач индекса 2. Решатели MATLAB оказались менее эффектными. Для ode23s это объясняется тем, что он имеет 1-й стадийный порядок и не является жесткоточным. Решатель ode23tb обеспечил заданную точность, но его затраты при $Tol = 10^{-3}$ на порядок больше, чем у TB2E. На рис. 3.3 приведены графики изменения размера шага этих решателей, из которых ясно, почему так происходит, – алгоритм управления шагом решателя ode23tb оказался неустойчивым. Из-за этого также не удалось получить результаты решателя ode23t при $Tol = 10^{-2}$.

Таблица 3.9. Результаты решения системы ДАУ (3.35)

Решатель	$Tol = 10^{-2}$				$Tol = 10^{-3}$			
	Ошибка	***Nf***	***NJ***	***NLU***	**Ошибка**	***Nf***	***NJ***	***NLU***
TRE	2.13×10^{-2}	23	5	8	2.76×10^{-3}	60	7	17
TB2E	7.80×10^{-3}	25	5	6	8.21×10^{-4}	61	7	12
Lob2E	5.83×10^{-2}	21	6	8	2.85×10^{-2}	90	9	33
TB2R	1.19×10^{-2}	17	8	8	2.55×10^{-4}	49	24	24
ode23t	–	–	–	–	1.09×10^{-3}	1779	2	5
ode23tb	1.34×10^{-3}	181	17	40	8.42×10^{-5}	678	52	177
ode23s	1.52×10^{-3}	828	272	275	1.53×10^{-4}	8143	2712	2715

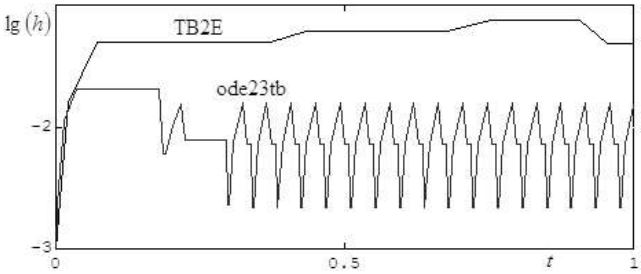


Рис. 3.3. Изменение размера шага при решении ДАУ (3.35)

Сходимость методов Рунге–Кутты при решении жестких и дифференциально-алгебраических задач

ГЛАВА 4

4.1. Сводка результатов о сходимости

Основные результаты, изложенные в этой главе, получены путем исследования простейших уравнений, моделирующих поведение различных составляющих ошибки при численном решении жестких ОДУ и ДАУ индексов 1, 2 и 3. Некоторые из этих результатов имеют характер предположений и нуждаются в строгих доказательствах. Поэтому приведем сначала строго доказанные теоретические результаты о сходимости методов Рунге–Кутты при решении нежестких, жестких и дифференциально-алгебраических задач.

Классические условия порядка были рассмотрены в главе 2 и представляют собой алгебраические уравнения относительно коэффициентов метода. При повышении порядка число таких условий быстро увеличивается, поэтому построение методов высоких порядков является сложной задачей. Чтобы упростить построение таких методов, используют упрощающие предположения:

$$B(\theta):\ \mathbf{b}^{\mathrm{T}}\mathbf{c}^{k-1} = 1/k, \qquad k = 1, \ldots, \theta;$$

$$C(\eta):\ \mathbf{A}\mathbf{c}^{k-1} = \mathbf{c}^{k}/k, \qquad k = 1, \ldots, \eta;$$

$$D(\zeta):\ (\mathbf{b}\mathbf{c}^{k-1})^{\mathrm{T}}\mathbf{A} = \frac{1}{k}(\mathbf{b} - \mathbf{b}\mathbf{c}^{k})^{\mathrm{T}}, \qquad k = 1, \ldots, \zeta.$$

Условие $B(\theta)$ означает, что квадратурная формула, задаваемая весами b_i и абсциссами c_i, имеет порядок θ. Условие $C(\eta)$ означает, что η – наименьший порядок на всех внутренних стадиях. Условие $D(\zeta)$ не имеет простой интерпретации, но оно используется во многих теоретических результатах о сходимости методов Рунге–Кутты.

Теорема 4.1 [88]. *Если коэффициенты метода Рунге–Кутты удовлетворяют условиям* $B(p)$, $C(\eta)$, $D(\zeta)$ *и при этом* $p \le \eta + \zeta + 1$ *и* $p \le 2\eta + 2$, *то метод имеет порядок* p.

Для методов решения жестких ОДУ и ДАУ важными понятиями являются жесткая точность и стадийный порядок. Неявный метод называется *жесткоточным*, если $b_i = a_{si}$, $i = 1, \ldots, s$ (при этом предполагается, что он не может быть приведен к методу с меньшим числом стадий). *Стадийный порядок* метода определяется как наибольшее целое число q, для которого выполняются равенства

$$\mathbf{A}\mathbf{c}^{k-1} = \mathbf{c}^k/k, \quad \mathbf{b}^{\mathrm{T}}\mathbf{c}^{k-1} = 1/k, \quad k = 1, \ldots, q, \tag{4.1}$$

т. е. условие стадийного порядка получим, объединив условия $B(q)$ и $C(q)$. В дальнейшем будем предполагать, что метод имеет порядок p, стадийный порядок q и удовлетворяет условиям $B(p)$, $C(q)$ и $D(\zeta)$.

Порядок метода можно оценить, последовательно уменьшая размер шага. При решении нежестких задач реальный порядок, полученный при умеренных размерах шага, обеспечивающих заданную точность, обычно мало отличается от порядка метода. Но при решении жестких задач реальный порядок может быть ниже порядка метода, причем его оценка практически не меняется при изменении размера шага в широких пределах, и только при очень малом шаге можно получить оценку, близкую к порядку метода.

Для исследования поведения ошибки при решении жестких задач Протеро и Робинсон [135] предложили рассмотреть уравнение

$$y' = \lambda(y - \varphi(t)) + \varphi'(t), \quad y(t_0) = \varphi(t_0), \tag{4.2}$$

решение которого $y(t) = \varphi(t)$. В [12, 75, 135] исследовалась погрешность численного решения этого уравнения при $h \to 0, z = h\lambda \to \infty$. При таких предположениях асимптотическое поведение локальной ошибки выражается формулой $\delta = O(z^{-r}h^{q+1})$, где q – стадийный порядок, r – порядок затухания ошибки при $z \to \infty$. Оценки глобальной ошибки дадим для случая переменного шага, тогда символы h и z нужно интерпретировать как $\max h_i$ и $z = \min(h_i\lambda)$. Если матрица $\mathbf{A}$ обратима и метод не является жесткоточным, то $r = 0$, а глобальная ошибка имеет оценку $\Delta = \varphi(t_n) - y_n = O(h^{q+1})$ при $|R(\infty)| < 1$ и $\Delta = O(h^q)$ при $|R(\infty)| = 1$. Для жесткоточных методов $r \geq 1$ (обычно $r = 1$), тогда $\delta \to 0$ при $z \to \infty$, а $\Delta = O(z^{-r}h^{q+1})$ при $|R(\infty)| < 1$ и $\Delta = O(z^{-r}h^q)$ при $|R(\infty)| = 1$.

Систему ДАУ индекса 1 можно представить в виде:

$$\begin{aligned} \mathbf{y}' &= \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \\ \mathbf{0} &= \mathbf{g}(\mathbf{y}, \mathbf{z}), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \end{aligned} \tag{4.3}$$

где матрица $\mathbf{g}_z = \partial\mathbf{g}(\mathbf{y}, \mathbf{z})/\partial\mathbf{z}$ обратима в окрестности решения. Для такой системы начальные условия согласованы, если выполняется соотношение $\mathbf{0} = \mathbf{g}(\mathbf{y}_0, \mathbf{z}_0)$. Для численного решения этой системы воспользуемся методом с обратимой матрицей $\mathbf{A}$. Тогда, согласно теореме VI.1.1 из [75, с. 423], глобальные ошибки соответствующих переменных имеют оценки $\mathbf{y}(t_n) - \mathbf{y}_n = O(h^p)$, $\mathbf{z}(t_n) - \mathbf{z}_n = O(h^r)$, где $r = p$, если метод жесткоточный; $r = \min(p, q+1)$, если

$-1 \leq R(\infty) < 1$, и $r = \min(p-1, q)$, если $R(\infty) = 1$.

Мы убедились в полезности жесткой точности и высокого стадийного порядка для эффективного решения жестких систем и ДАУ. Кроме методов с обратимой матрицей $\mathbf{A}$, для решения таких задач применяют жесткоточные методы с явной первой стадией, таблица Бутчера которых имеет вид:

$$\begin{array}{c|ccc} 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & \dots & a_{2s} \\ \vdots & \vdots & \cdots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & a_{s1} & \dots & a_{ss} \end{array} = \begin{array}{c|cc} 0 & 0 & \mathbf{0} \\ \tilde{\mathbf{c}} & \mathbf{a} & \tilde{\mathbf{A}} \\ \hline & a_{s1} & \tilde{\mathbf{b}}^{\mathrm{T}} \end{array}, \tag{4.4}$$

где матрица $\tilde{\mathbf{A}}$ обратима. К ним относятся методы Лобатто IIIA [75, 89], диагонально-неявные методы ESDIRK [56, 124], а также методы, рассмотренные в [67, 92, 106, 107].

Системы ДАУ индекса 2 принято представлять в виде:

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{4.5а}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{y}), \quad \mathbf{z}(t_0) = \mathbf{z}_0. \tag{4.5б}$$

Продифференцировав (4.5б), получим уравнение «скрытой связи» [75]:

$$\mathbf{0} = \mathbf{g}_{\mathbf{y}}\mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{z}(t_0) = \mathbf{z}_0. \tag{4.5в}$$

Если матрица $\mathbf{g}_{\mathbf{y}}\mathbf{f}_{\mathbf{z}}$ обратима в окрестности траектории решения, то уравнения (4.5а, в), представляют собой задачу индекса 1, а исходная система (4.5а, б) – задачу индекса 2. Начальные условия будут согласованными, если они удовлетворяют (4.5б) и (4.5в). Переменные, входящие в уравнения (4.5а, б), принято подразделять на переменные индекса 1 y_i (y-компонента) и переменные индекса 2 z_i (z-компонента). Для удобства дальнейшего изложения обозначим через p_y и p_z порядки глобальных ошибок соответствующих компонент, т. е. вместо $\mathbf{y}(t_n) - \mathbf{y}_n = O(h^r)$ будем записывать $p_y = r$.

Сходимость численных решений для ДАУ индекса 2 исследовалась в работах [75, 109, 116]. Рассмотрим два типа методов, наиболее предпочтительных для решения ДАУ: SI – жесткоточные методы с обратимой матрицей $\mathbf{A}$ (эти методы имеют $R(\infty) = 0$) и SE – жесткоточные методы с явной первой стадией и обратимой матрицей $\tilde{\mathbf{A}}$, имеющие $|R(\infty)| < 1$. Для всех таких методов $p_y = \min(p, 2q, q + \zeta + 1)$, $p_z = q$, а если $\mathbf{z}$ входит линейно в (4.5а), то $p_y = \min(p, 2q + 1, q + \zeta + 1)$.

Системы ДАУ индекса 3 принято записывать в виде:

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{4.6а}$$

$$\mathbf{z}' = \mathbf{k}(\mathbf{y}, \mathbf{z}, \mathbf{u}), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \tag{4.6б}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{y}), \quad \mathbf{u}(t_0) = \mathbf{u}_0. \tag{4.6в}$$

Такая система имеет индекс 3, если матрица $\mathbf{g}_{\mathbf{y}}\mathbf{f}_{\mathbf{z}}\mathbf{k}_{\mathbf{u}}$ обратима в окрестности решения. Начальные условия будут согласованными, если они удовлетворяют (4.6в) и уравнениям, полученным в результате однократного и двукратного дифференцирования (4.6в). Для таких задач известен результат, полученный в [117] для методов типа SI при $q \geq 2$. В этом случае $p_y = \min(p, 2q - 2, q + \zeta)$, $p_z = q$, $p_u = q - 1$, а если u входит линейно в (4.6б), то $p_y = \min(p, 2q - 1, q + \zeta)$. Для методов типа SE нам неизвестно об аналогичных теоретических оценках.

Результаты о сходимости методов типов SI и SE при решении ДАУ сведены в табл. 4.1. Приведенные порядки сходимости являются нижними оценками; при выполнении некоторых дополнительных условий они могут быть выше. В этой главе на основе исследования простых модельных уравнений получены некоторые условия, которые являются необходимыми для достижения более высоких порядков.

Таблица 4.1. Сходимость методов Рунге–Кутты при решении ДАУ

Индекс ДАУ	Методы	p_y	p_z	p_u
1	SI, SE	p	p	—
2	SI, SE	$\min(p, 2q, q+\zeta+1)$	q	—
3	SI ($q \geq 2$)	$\min(p, 2q-2, q+\zeta)$	q	$q-1$

4.2. Феномен снижения порядка

Согласно классическим представлениям, точность метода Рунге–Кутты определяется порядком сходимости и значениями коэффициентов погрешности. Поэтому параметры метода обычно выбирают исходя из условий обеспечения заданного порядка и минимизации коэффициентов погрешности. Такой подход вполне оправдан при построении методов решения нежестких задач, когда реальный порядок при размере шага, обеспечивающем приемлемую точность решения, практически совпадает с классическим порядком. Но для жестких задач традиционный подход, основанный на асимптотической оценке ошибки при $h \to 0$, не всегда позволяет правильно оценить точность решения. В этом случае реальный порядок может быть ниже классического порядка, сходимость с которым обеспечивается только при малых значениях $h\lambda_i$, где λ_i – собственные числа матрицы Якоби. Таким образом, в жестком случае поведение ошибки следует изучать при достаточно больших значениях $h\lambda_i$ для жесткого спектра.

Явление снижения реального порядка при решении жестких задач получило известность как *феномен снижения порядка* и впервые было исследовано в [135]. Продемонстрируем это явление на примере решения задачи Капса:

$$\begin{aligned} y_1' &= -(\mu+2)y_1 + \mu y_2^2, \quad y_1(0) = 1, \\ y_2' &= y_1 - y_2 - y_2^2, \quad y_2(0) = 1, \quad 0 \leq t \leq 1, \end{aligned} \tag{4.7}$$

имеющей точное решение $y_1(t) = \exp(-2t)$, $y_2(t) = \exp(-t)$, не зависящее от показателя жесткости μ.

Для решения этой задачи будем использовать неявные L-устойчивые методы Рунге–Кутты 3-го порядка с функцией устойчивости

$$R(z) = \frac{1 + (1-3\gamma)z + (1/2 - 3\gamma + 3\gamma^2)z^2}{(1-\gamma z)^3}, \quad \gamma = 0.43586652150846$$

(γ – один из корней уравнения $6x^3 - 18x^2 + 9x - 1 = 0$). Коэффициенты методов заданы следующими таблицами:

SDIRK31a:

$$\begin{array}{c|ccc} \gamma & \gamma & & \\ 1/2 & 1/2-\gamma & \gamma & \\ 1 & \dfrac{1-7\gamma+14\gamma^2}{1-2\gamma} & 4\gamma\dfrac{1-3\gamma}{1-2\gamma} & \gamma \\ \hline & \dfrac{1}{(1-\gamma)(6-12\gamma)} & \dfrac{2-6\gamma}{3-6\gamma} & \dfrac{1}{6-6\gamma} \end{array}$$

SDIRK31b:

$$\begin{array}{c|ccc} \gamma & \gamma & & \\ \dfrac{1+\gamma}{2} & \dfrac{1-\gamma}{2} & \gamma & \\ 1 & 1-b_2-\gamma & b_2 & \gamma \\ \hline & 1-b_2-\gamma & b_2 & \gamma \end{array}, \quad b_2 = \frac{5}{4} - 5\gamma + \frac{3}{2}\gamma^2$$

ESDIRK32:

$$\begin{array}{c|ccc} 0 & 0 & & \\ 2\gamma & \gamma & \gamma & \\ 1 & 1-a_{32}-\gamma & a_{32} & \gamma \\ 1 & 1-b_2-b_3-\gamma & b_2 & b_3 & \gamma \\ \hline & 1-b_2-b_3-\gamma & b_2 & b_3 & \gamma \end{array}, \quad a_{32} = \frac{1-2\gamma}{4\gamma}, \; b_2 = \frac{1}{12\gamma(1-2\gamma)}, \; b_3 = \frac{1-6\gamma+6\gamma^2}{3-6\gamma}$$

DESI33:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ (3-\sqrt{3})\gamma & \dfrac{\sqrt{3}}{3}\gamma & \dfrac{6-\sqrt{3}}{6}\gamma & \dfrac{12-7\sqrt{3}}{6}\gamma & 0 \\ (3+\sqrt{3})\gamma & -\dfrac{\sqrt{3}}{3}\gamma & \dfrac{12+7\sqrt{3}}{6}\gamma & \dfrac{6+\sqrt{3}}{6}\gamma & 0 \\ 1 & 1-b_2-b_3-\gamma & b_2 & b_3 & \gamma \\ \hline & 1-b_2-b_3-\gamma & b_2 & b_3 & \gamma \end{array},$$

$$b_2 = \frac{2-6\gamma+6c_3\gamma-3c_3}{6c_2(c_2-c_3)}, \quad b_3 = \frac{2-6\gamma+6c_2\gamma-3c_2}{6c_3(c_3-c_2)}.$$

В обозначении метода первая цифра – порядок, вторая – стадийный порядок. Метод SDIRK31b был предложен в [81], а ESDIRK32 – в [124]. Метод DESI33 принадлежит к классу диагонально-расширенных однократно неявных методов [67, 92]. При решении линейных автономных задач ошибка определяется только функцией устойчивости, поэтому все эти методы на одной и той же сетке показывают одинаковые результаты. Но при решении нелинейных либо неавтономных задач результаты существенно различаются.

Ошибку решения вычисляем в виде $e(h) = \max\left(\sqrt{e_1(t)^2 + e_2(t)^2}, 0 \le t \le 1\right)$, где $e_i(t)$ – ошибка по i-й компоненте при размере шага h. Порядок метода оцениваем по формуле $\tilde{p} = \lg\left(e(0.1)/e(0.01)\right)$. Результаты решения задачи Капса при $h = 0.1$ и $h = 0.01$ приведены на рис. 4.1 и характерны для многих жестких задач. Нежесткоточный метод SDIRK31a демонстрирует снижение точности и порядка при увеличении жесткости. Остальные методы являются жесткоточными и показывают даже некоторое уменьшение ошибки при большой жесткости. Это объясняется тем, что первое уравнение в (4.7) при $\mu \to \infty$ вырождается в алгебраическое соотношение $0 = -y_1 + y_2^2$, точное выполнение которого обеспечивают жесткоточные методы. Отметим, что преимущество методов более высокого стадийного порядка возрастает при уменьшении размера шага и сохраняется при малых значениях $h\mu$. Этот факт подтверждается графиками зависимости ошибки от размера шага, приведенными (при $\mu = 10^3$) на рис. 4.2.

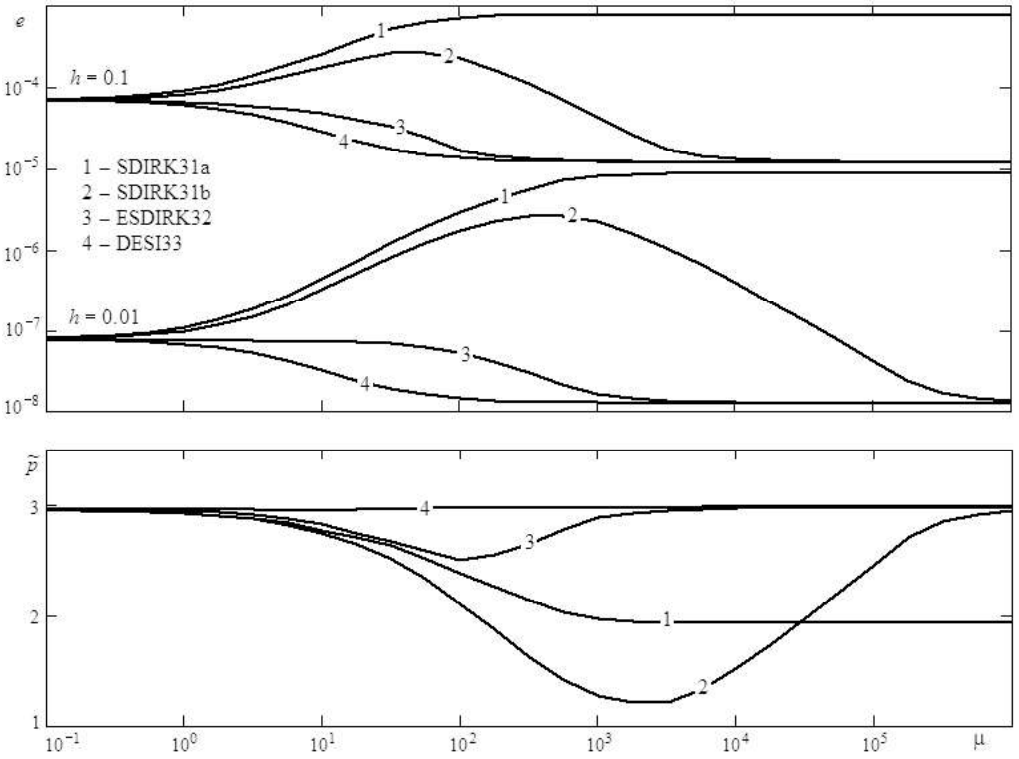


Рис. 4.1. Ошибки и оценки порядка в зависимости от жесткости задачи Капса

Феномен снижения порядка исследовался во многих работах, среди которых [12, 54, 75, 135, 137]. Впервые он был объяснен в [135] с помощью уравнения (4.2). Исследуем поведение ошибки при $h \to 0$ и конечных значениях $z = h\lambda$. Применяя один шаг метода Рунге–Кутты к уравнению (4.2), получаем

$y_1 = \varphi_0 + \mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(z(\mathbf{e}\varphi_0 - \mathbf{\Phi}) + h\mathbf{\Phi}')$,

где

$$\mathbf{\Phi} = \begin{bmatrix} \varphi(t_0 + c_1 h) \\ \cdots \\ \varphi(t_0 + c_s h) \end{bmatrix}, \quad \mathbf{\Phi}' = \begin{bmatrix} \varphi'(t_0 + c_1 h) \\ \cdots \\ \varphi'(t_0 + c_s h) \end{bmatrix}.$$

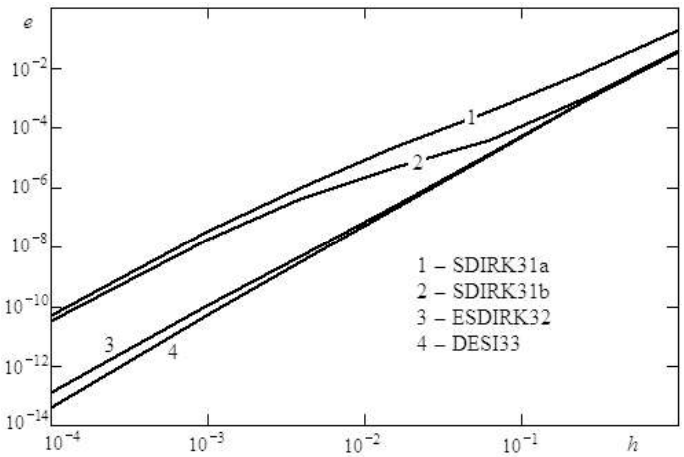


Рис. 4.2. Ошибки решения задачи Капса
в зависимости от размера шага при μ = 10^3

Используя разложение $\varphi(t)$ в ряд Тейлора, получаем локальную ошибку на первом шаге в виде

$$\delta_1 = \varphi(t_0 + h) - y_1 = \sum_{i=q+1}^{\infty} e_i(h\lambda)\frac{d^i\varphi(t_0)}{dt^i}\frac{h^i}{i!}, \tag{4.8}$$

где

$$e_i(z) = z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{c}^i - i\mathbf{A}\mathbf{c}^{i-1}) + (1 - i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1}) \tag{4.9}$$

– предложенные в [52, 53] функции погрешности. Глобальная ошибка выражается формулой

$$\Delta_{n+1} = \varphi(t_{n+1}) - y_{n+1} = R(h\lambda)\Delta_n + \delta_{n+1}, \tag{4.10}$$

где $R(z)$ – функция устойчивости метода, δ_{n+1} – локальная ошибка на $(n+1)$-м шаге.

Для метода стадийного порядка q выполняются равенства (4.1), поэтому $e_i(z) \equiv 0$ при $i \le q$. Это означает, что главный член ошибки в разложении (4.8) пропорционален $e_{q+1}(z)$ и соответствующему члену разложения $\varphi(t)$ в ряд Тейлора. Глобальная ошибка при $|R(z)| < 1$ асимптотически ведет себя так же, как и локальная, причем при $|R(z)| << 1$ она практически равна локальной. Таким

образом, глобальная ошибка может вести себя как $O(h^{q+1})$, а не как $O(h^p)$, что объясняет феномен снижения порядка. Для жесткоточного $L(\alpha)$-устойчивого метода глобальная ошибка при $z \to \infty, h \to 0$ имеет вид $O(z^{-r}h^{q+1})$, $r \geq 1$ и асимптотически равна локальной.

Понять поведение жесткой составляющей ошибки позволяет уравнение (4.2) при $\varphi(t) = t^i$. Если $i \leq q$, то оно решается точно. При $i = q + 1$ локальная ошибка на всех шагах одинакова (при постоянном размере шага) и выражается формулой $\delta = e_{q+1}(h\lambda)h^{q+1}$. В соответствии с (4.10) глобальная ошибка после выполнения большого числа шагов сходится к значению

$$\Delta(h) = E_{q+1}(h\lambda)h^{q+1}, \quad E_{q+1}(z) = \frac{e_{q+1}(z)}{1 - R(z)}.$$

Выражение для порядка метода при решении данной задачи получаем в виде

$$\tilde{p}(z) = \frac{h\left|\Delta(h)\right|'_h}{\left|\Delta(h)\right|} = q + 1 + \frac{z\left|E_{q+1}(z)\right|'_z}{\left|E_{q+1}(z)\right|}.$$

На рис. 4.3 приведены зависимости $|E_{q+1}(z)|$ и $\tilde{p}(z)$, которые объясняют поведение ошибок методов SDIRK31a и SDIRK31b на рис. 4.1. Ошибка метода SDIRK31a при больших значениях μ полностью определяется жесткой составляющей, поэтому его порядок снижается до 2-го. Жесткоточный метод SDIRK31b сохраняет порядок не только при малых, но и при очень больших μ, когда доминирует нежесткая составляющая ошибки. При умеренных значениях μ доминирует жесткая составляющая ошибки, а порядок снижается почти до 1-го. Приведенные результаты показали важность понятий стадийного порядка и жесткой точности для эффективного решения жестких задач.

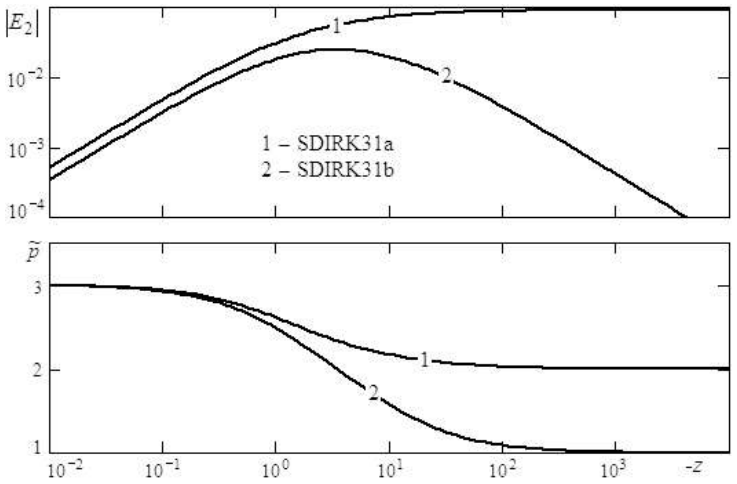


Рис. 4.3. Зависимости $|E_2(z)|$ и $\tilde{p}(z)$ для уравнения Протеро–Робинсона

4.3. Сходимость явных методов при решении жестких задач

Эффект снижения точности может проявляться при решении задач умеренной жесткости, которые можно эффективно решать явными методами. Поэтому имеет смысл исследовать влияние жесткости на точность явных методов. Явные методы Рунге–Кутты имеют только 1-й стадийный порядок, поэтому главный член ошибки при решении уравнения Протеро–Робинсона определяется функцией погрешности $e_2(z)$.

Рассмотрим методы Рунге–Кутты 3-го порядка. Трехстадийные методы рассмотрены в разделе 2.5 и образуют двухпараметрическое семейство, задаваемое абсциссами c_2 и c_3. Невязки условий 2-го и 3-го стадийного порядка этих методов имеют вид:

$$\mathbf{c}^2 - 2\mathbf{A}\mathbf{c} = \begin{bmatrix} 0 \\ c_2^2 \\ c_2 c_3 \dfrac{3c_3 - 2}{3c_2 - 2} \end{bmatrix}, \quad \mathbf{c}^3 - 3\mathbf{A}\mathbf{c}^2 = \begin{bmatrix} 0 \\ c_2^3 \\ c_3^3 + 3c_2 c_3 \dfrac{c_3 - c_2}{3c_2 - 2} \end{bmatrix},$$

а функции погрешности таких же порядков равны

$$e_2(z) = \frac{1}{6} c_2 z^2, \quad e_3(z) = \frac{1}{6}\left[c_2^2 z^2 + (2c_3 - c_2 - 3c_2 c_3) z\right].$$

При малом c_2 метод имеет «почти 2-й» стадийный порядок и малые значения функции $e_2(z)$, а если дополнительно задать малое значение c_3, то получим метод «почти 3-го» стадийного порядка с малыми значениями $e_2(z)$ и $e_3(z)$.

Обозначим через ERK31 «оптимальный» метод Ральстона, имеющий $c_2 = 1/2$, $c_3 = 3/4$, через ERK31b – метод, имеющий $c_2 = 0.01$, $c_3 = 3/4$, и через ERK31c – метод, имеющий $c_2 = 0.0001$, $c_3 = 0.01$. (2-я цифра в названии метода – стадийный порядок.) Для сравнения используем методы

RK32:

0	0	0	0
1/2	1/8	1/2	−1/8
1	−1/2	2	−1/2
	1/6	2/3	1/6

и RK33:

0	0	0	0	0
1/3	5/36	2/9	−1/36	0
2/3	1/3	−2/9	7/9	−2/9,
1	5/4	−3	15/4	−1
	5/4	−3	15/4	−1

которые формально являются неявными, но имеют такую же функцию устойчивости

$$R(z) = 1 + z + z^2/2 + z^3/6, \tag{4.11}$$

как и явные методы. Но при этом метод RK32 имеет 2-й, а метод RK33 – 3-й стадийный порядок. На рис. 4.4а приведены ошибки решения задачи Капса этими методами при $h = 1/30$. Видно, что методы ERK31b и RK32, а также ERK31c

и RK33 показывают очень близкие результаты, заметно лучшие, чем ERK31. Таким образом, при малых значениях c_2 или c_2 и c_3 явные методы приобретают свойства методов более высокого стадийного порядка.

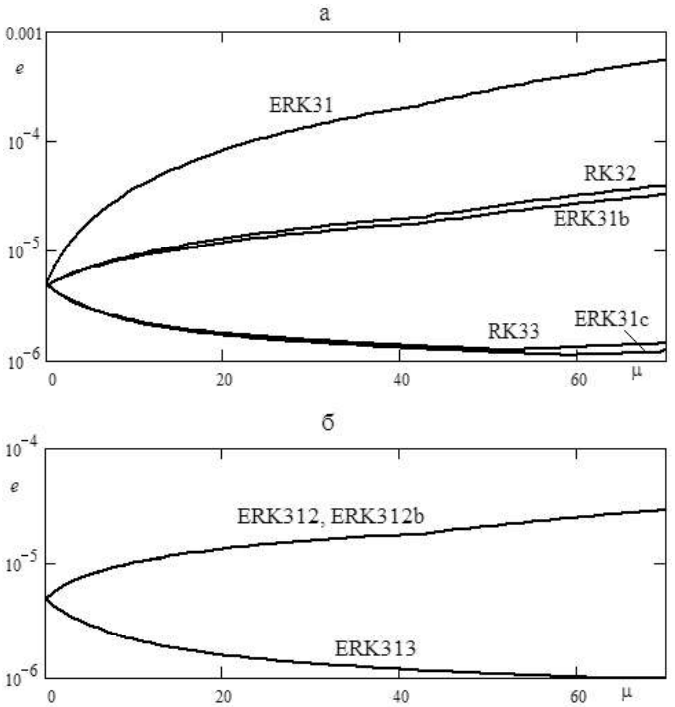


Рис. 4.4. Ошибки решения задачи Капса явными методами

Недостаток методов с малыми абсциссами – большие значения некоторых коэффициентов (метод ERK31b имеет $a_{32} = 28.17$, а у метода ERK31c $b_2 = -3.32 \times 10^5$). В [55] были предложены явные методы, свободные от этого недостатка и имеющие тождественно равные нулю функции погрешности. Метод 3-го порядка

	0				
	1/2	1/2			
ERK312:	1	1	0		
	1	−1/2	2	−1/2	
		1/6	2/3	−1/6	1/3

имеет функцию устойчивости (4.11) и $e_2(z) \equiv 0$, а методы 3-го порядка

ERK312b:

0					
1/3	1/3				
2/3	2/3	0			
1	1	0	0		
1	1/12	3/2	−3/4	1/6	
	5/4	−3	15/4	−2	1

и ERK313:

0					
1/3	1/3				
2/3	2/3	0			
1	1	0	0		
0	−11/12	3/2	−3/4	1/6	
	1/4	−3	15/4	−1	1

дополнительно имеют $e_3(z) \equiv 0$. Третья цифра в названиях методов – псевдостадийный порядок, который будет определен в разделе 4.8. Ошибки решения этими методами задачи Капса приведены на рис. 4.4б. Кривые ошибок методов ERK312 и ERK312b практически не различаются и близки к соответствующим кривым методов ERK31b и RK32 на рис. 4.4а, а ошибка метода ERK313b очень мало отличается от ошибок методов ERK31c и RK33. Для полноты картины приводим в табл. 4.2 показатели, характеризующие точность рассмотренных методов. Здесь $||e_2||$, $||e_3||$ и $||e_{32}||$ – нормы соответствующих функций погрешности, определенные как $||e_i|| = \max(|e_i(z)|, z^* \le z \le 0)$, где $z^* = -2.513$ – граница интервала устойчивости вдоль вещественной оси. Функция $e_{32}(z)$ учитывает нелинейные составляющие ошибки при решении жестких задач и будет определена ниже.

Таблица 4.2. Показатели точности методов 3-го порядка

Метод	q	$\|\|e_2\|\|$	$\|\|e_3\|\|$	$\|\|e_{32}\|\|$	$e(T_{41})$	$e(T_{42})$	$e(T_{43})$	$e(T_{44})$
ERK31	1	0.526	0.315	0	0.083	0	0	1
ERK31b	1	0.011	0.614	0.616	0.002	0	0.980	1
ERK31c	1	0.0001	0.008	0.008	0.987	0.987	1	1
RK32	2	0	1.154	1.154	0	0	1	1
RK33	3	0	0	0	1	1	1	1
ERK312	1	0	1.154	0.838	0	–0.333	1	1
ERK312b	1	0	0	0.838	1	–0.333	1	1
ERK313	1	0	0	0	1	1	1	1

На основании приведенных результатов можно сделать вывод, что минимизация функций погрешности может привести к эффекту, аналогичному повышению стадийного порядка. Пока нам не удалось объяснить, почему метод ERK312b показывает результаты как у метода 2-го стадийного порядка, а метод

ERK313 – как у метода 3-го стадийного порядка, хотя оба метода имеют $e_2(z) \equiv 0$ и $e_3(z) \equiv 0$. Можно предположить, что рассмотрения только уравнения Протеро–Робинсона недостаточно для объяснения поведения всех компонент жесткой составляющей ошибки. Ниже мы увидим, что это действительно так и что необходимо рассматривать также и другие простейшие уравнения, моделирующие поведение ошибки при решении жестких нелинейных задач.

4.4. Неявные методы, обратные к явным методам

Вернемся теперь к неявным методам и покажем, что и для них минимизация функций погрешности эквивалентна повышению стадийного порядка. Удобнее всего это сделать на примере методов, обратных к рассмотренным в предыдущем разделе явным методам Рунге–Кутты.

Пусть заданы коэффициенты a_{ij}, b_i, c_i метода Рунге–Кутты. Коэффициенты обратного к нему метода получаем в виде:

$$c_i^* = 1 - c_i, \quad a_{ij}^* = b_j - a_{ij}, \quad b_j^* = b_j. \tag{4.12}$$

Обратный метод обладает тем свойством, что если сделать из начальной точки один шаг прямого метода, а затем сделать шаг обратным методом в обратном направлении (поменяв h на $-h$), то получим исходный вектор $\mathbf{y}_0$. Отметим, что термин *обратный* (*inverse, backward*), используемый для обозначения таких методов в [20] и ряде других работ, не является общепринятым. Наряду с ним используют термины *присоединенный* (*adjoint*) [74] и *отраженный* (*reflected*) [90]. Обратный метод имеет тот же порядок, что и исходный, а его коэффициенты погрешности порядка $p + 1$ отличаются от коэффициентов погрешности исходного метода множителем $(-1)^p$ [74, теорема II.8.4]. Обратный метод имеет тот же стадийный порядок, что и исходный [90, теорема 343B]. Нетрудно также показать, что в результате двукратного обращения получим исходный метод.

Вместо принятого представления неявных методов Рунге–Кутты (1.24) иногда удобно использовать альтернативное представление [87, 131] в виде

$$\begin{aligned} &\mathbf{y}_1 = \mathbf{y}_0 + h\sum_{i=1}^{s} b_i \mathbf{F}_i, \quad \mathbf{F}_i = \mathbf{f}(t_0 + c_i h, \mathbf{Y}_i), \\ &\mathbf{Y}_i = (1 - v_i)\mathbf{y}_0 + v_i \mathbf{y}_1 + h\sum_{j=1}^{s} x_{ij} \mathbf{F}_j, \ i = 1, \ldots, s. \end{aligned} \tag{4.13}$$

Соответствующая модифицированная таблица имеет вид:

$$\begin{array}{c|c|ccc} c_1 & v_1 & x_{11} & \cdots & x_{1s} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_s & v_s & x_{s1} & \cdots & x_{ss} \\ \hline & & b_1 & \cdots & b_s \end{array} = \begin{array}{c|c|c} \mathbf{c} & \mathbf{v} & \mathbf{X} \\ \hline & & \mathbf{b}^{\mathrm{T}} \end{array}. \tag{4.14}$$

Особый интерес представляют методы, имеющие $x_{ij} = 0$ при $j \geq i$. В этом случае, зная $\mathbf{y}_1$, можно найти все стадийные значения $\mathbf{Y}_i$ непосредственно по фор-

мулам (4.13). Это позволяет свести систему алгебраических уравнений (4.13) относительно векторов $\mathbf{Y}_1, \ldots, \mathbf{Y}_s$ к уравнению относительно только вектора $\mathbf{y}_1$, что обеспечивает эффективную реализацию метода. В [96] такие методы были названы *мононеявными*, они рассматривались во многих работах, среди которых [31, 69, 87, 121, 131]. Отметим, что мононеявные методы все же уступают методам DIRK по простоте и эффективности реализации, поскольку матрицу Якоби алгебраической системы получаем в виде матричного многочлена от $h\mathbf{J}$, вычисление которого требует дополнительных затрат и в случае разреженной матрицы $\mathbf{J}$ приводит к потере разреженности.

Модифицированная таблица (4.14) удобна также и для представления обратных методов, коэффициенты которых находим по формулам $\mathbf{c}^* = \mathbf{e} - \mathbf{c}$, $\mathbf{v}^* = \mathbf{e} - \mathbf{v}$, $\mathbf{X}^* = -\mathbf{X}$, $\mathbf{b}^* = \mathbf{b}$. Модифицированные таблицы явного и обратного к нему неявного методов имеют вид:

Прямой метод

$$\begin{array}{c|c|ccccc}
0 & 0 & & & & & \\
c_2 & 0 & a_{21} & & & & \\
c_3 & 0 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & & \\
c_s & 0 & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Обратный метод

$$\begin{array}{c|c|ccccc}
1 & 1 & & & & & \\
1-c_2 & 1 & -a_{21} & & & & \\
1-c_3 & 1 & -a_{31} & -a_{32} & & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & & \\
1-c_s & 1 & -a_{s1} & -a_{s2} & \cdots & -a_{s,s-1} & \\
\hline
 & & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Из этих таблиц видно, что обратный к явному метод является мононеявным. В [20] было предложено использовать обратные к явным методы Рунге–Кутты для решения жестких задач. Такие методы обладают рядом полезных свойств. Они жесткоточные, имеют высокий порядок L-затухания и удобны для реализации, поскольку являются мононеявными. В то же время они имеют существенный недостаток – 1-й стадийный порядок, что приводит к снижению точности и реального порядка при решении жестких и дифференциально-алгебраических уравнений.

Непосредственно из свойств обратного метода получаем его функции устойчивости и погрешности в виде

$$R^*(z) = R^{-1}(-z), \quad e^*_{q+1}(z) = (-1)^q R^{-1}(-z) e_{q+1}(-z).$$

Функции погрешности метода, обратного к явному трехстадийному методу 3-го порядка, имеют вид:

$$e^*_2(z) = -c_2 z^2 / D(z), \quad e^*_3(z) = \left[(c_2 - 2c_3 + 3c_2c_3)z + (c_2^2 - 3c_2)z^2\right] \Big/ D(z),$$

$$D(z) = 6 - 6z + 3z^2 - z^3,$$

а если явный метод имеет $e_2(z) \equiv 0$ или $e_2(z) \equiv 0$, $e_3(z) \equiv 0$, то и обратный к нему метод обладает таким же свойством. Отметим, что согласно (4.12) малые значения абсцисс исходного явного метода преобразуются в значения, близкие к 1 обратного неявного метода.

Численные эксперименты подтвердили, что неявные методы, построенные на основе явных методов с минимизированными функциями погрешности, позволяют избежать снижения точности и порядка при решении жестких задач. На рис. 4.5 приведены ошибки и оценки порядка при решении задачи Капса с шагом $h = 1/30$. Использовались методы IERK31, IERK312, IERK312b и IERK313, полученные в результате обращения соответствующих методов ERK31, ERK312, ERK312b и ERK313. Порядок оцениваем по формуле

$$\tilde{p} = \frac{\lg\left(e(h_1)/e(h_2)\right)}{\lg(h_1/h_2)}, \tag{4.15}$$

где $e(h)$ – ошибка при размере шага h, $h_1 = 1/24$, $h_2 = 1/36$. Видно, что поведение ошибки метода IERK312b практически такое же, как у метода IERK312, и заметно хуже, чем у IERK313.

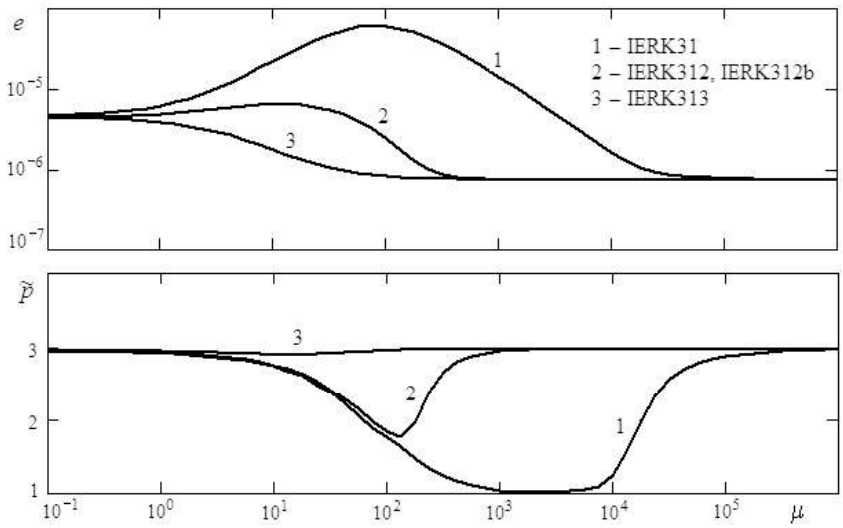


Рис. 4.5. Ошибки и оценки порядка при решении задачи Капса обратными методами

Но при решении линейных неавтономных задач метод IERK312b более точен, чем IERK312, и показывает такие же результаты, как метод IERK313. Например, при решении задачи PLATE получены значения *scd* (3.18), приведенные в табл. 4.3. Одинаковые ошибки методов IERK312b и IERK313 нетрудно объяснить. Эти методы имеют одинаковую функцию устойчивости, а их функции погрешности $e_i(z)$ также совпадают при всех i. Поэтому будут совпадать и численные решения уравнения Протеро–Робинсона, откуда следует также и совпадение решений векторного уравнения $\mathbf{y}' = \mathbf{J}\mathbf{y} + \mathbf{g}(t)$. Качественное отличие поведения ошибок этих методов при решении задачи Капса можно объяснить тем, что при решении нелинейных задач появляются составляющие ошибки, которых нет в решении линейного уравнения Протеро–Робинсона.

Таблица 4.3. Результаты решения задачи PLATE обратными методами

h	*scd*			
	IERK31	**IERK312**	**IERK312b**	**IERK313**
0.1	2.06	2.76	3.75	3.75
0.01	3.61	4.78	6.15	6.15

4.5. Модельные уравнения для нежестких задач

Исследование сходимости численного решения с помощью уравнения Протеро–Робинсона выполнялось во многих работах. Однако возникает вопрос: насколько правомерно использовать результаты, полученные для линейного неавтономного уравнения, в более общем случае нелинейных жестких дифференциальных уравнений? Для ответа на этот вопрос рассмотрим простейшие уравнения, моделирующие поведение различных составляющих ошибки при решении нежестких, жестких и дифференциально-алгебраических задач. Эти уравнения имеют известные точные решения, а их численные решения получены в виде выражений, содержащих коэффициенты метода. Подбором коэффициентов можно минимизировать ошибки решения модельных уравнений. Такой подход позволил построить явные и неявные методы повышенной точности для жестких и дифференциально-алгебраических задач [52–56, 60, 61, 63].

Начнем с нежестких задач. Определение порядка сходимости и коэффициентов погрешности метода Рунге–Кутты сводится к сравнению рядов Тейлора точного и численного решений. Для наглядного представления получаемых при разложении в ряд элементарных дифференциалов используют корневые деревья, при этом существует взаимно-однозначное соответствие между множеством элементарных дифференциалов и множеством деревьев. Эти же деревья будем использовать для формирования модельных уравнений, при этом каждой вершине дерева соответствует определенная переменная.

Принцип построения модельных уравнений изложен в [74, с. 164]. Дереву-точке поставим в соответствие переменную, описываемую уравнением $x'_1 = 1$. Корневой вершине дерева T_{ij} поставим в соответствие переменную x_{ij}. Уравнения зададим рекуррентно согласно формуле $x'_{\text{отца}} = \prod x_{\text{сыновей}}$, где «сыновья» – деревья, полученные в результате удаления корневой вершины дерева-«отца» вместе с инцидентными этой вершине ребрами. Начальные значения всех переменных зададим нулевыми. При таком подходе дерево можно рассматривать как сигнальный граф, ребра которого ориентированы по направлению к корневой вершине и осуществляют передачу сигналов, а вершины выполняют интегрирование произведения входных сигналов. Предположив, что вершины, не имеющие входов, вырабатывают сигнал t, получим на выходе корневой вершины дерева T_{ij} переменную x_{ij}.

Полученные уравнения и их решения ($x_{ij}(t)$ – точное, $\tilde{x}_{ij}(h)$ – численное на первом шаге) для деревьев до 4-го порядка включительно приведены

в табл. 4.4. При $t = 0$ разложение в ряд Тейлора переменной x_{ij} содержит только один ненулевой элементарный дифференциал, соответствующий дереву T_{ij}. Обозначим ошибку на первом шаге как $\delta_h(x_{ij}) = x_{ij}(h) - \tilde{x}_{ij}(h)$, тогда условия, обеспечивающие порядок p метода, запишутся в виде: $\delta_h(x_{ij}) = 0$ при $i \le p$. Коэффициенты погрешности получим как относительные ошибки соответствующих переменных на первом шаге: $e(T_{ij}) = \delta_h(x_{ij})/x_{ij}$.

Таблица 4.4. Нежесткие модельные уравнения

Дерево	Уравнение	$x_{ij}(t)$	$\tilde{x}_{ij}(h)$
•	$x_1' = 1$	t	$h\mathbf{b}^{\mathrm{T}}\mathbf{e}$
	$x_{21}' = x_1$	$t^2/2$	$h^2\mathbf{b}^{\mathrm{T}}\mathbf{c}$
	$x_{31}' = x_1^2$	$t^3/3$	$h^3\mathbf{b}^{\mathrm{T}}\mathbf{c}^2$
	$x_{32}' = x_{21}$	$t^3/6$	$h^3\mathbf{b}^{\mathrm{T}}\mathbf{Ac}$
	$x_{41}' = x_1^3$	$t^4/4$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{c}^3$
	$x_{42}' = x_1 x_{21}$	$t^4/8$	$h^4\mathbf{b}^{\mathrm{T}}(\mathbf{c}(\mathbf{Ac}))$
	$x_{43}' = x_{31}$	$t^4/12$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{Ac}^2$
	$x_{44}' = x_{32}$	$t^4/24$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{A}^2\mathbf{c}$

Применяя метод Рунге–Кутты для численного решения модельных уравнений, получаем:

$$\tilde{x}_1(t_{n+1}) = t_n + h, \quad \tilde{x}_{21}(t_{n+1}) = \tilde{x}_{21}(t_n) + h\mathbf{b}^{\mathrm{T}}\mathbf{X}_1,$$
$$\tilde{x}_{31}(t_{n+1}) = \tilde{x}_{31}(t_n) + h\mathbf{b}^{\mathrm{T}}\mathbf{X}_1^2, \quad \tilde{x}_{32}(t_{n+1}) = \tilde{x}_{32}(t_n) + h\mathbf{b}^{\mathrm{T}}\mathbf{X}_{21}, \ldots, \tag{4.16}$$

где векторы стадийных значений

$$\mathbf{X}_1 = \mathbf{e}t_n + h\mathbf{c}, \quad \mathbf{X}_{21} = \mathbf{e}\tilde{x}_{21}(t_n) + h\mathbf{A}\mathbf{X}_1,$$
$$\mathbf{X}_{31} = \mathbf{e}\tilde{x}_{31}(t_n) + h\mathbf{A}\mathbf{X}_1^2, \quad \mathbf{X}_{32} = \mathbf{e}\tilde{x}_{32}(t_n) + h\mathbf{A}\mathbf{X}_{21}, \ldots \tag{4.17}$$

Подставляя (4.17) в (4.16), имеем:

$$\tilde{x}_{21}(t_{n+1}) = \tilde{x}_{21}(t_n) + ht_n + h^2\mathbf{b}^{\mathrm{T}}\mathbf{c},$$
$$\tilde{x}_{31}(t_{n+1}) = \tilde{x}_{31}(t_n) + ht_n^2 + 2h^2t_n\mathbf{b}^{\mathrm{T}}\mathbf{c} + h^3\mathbf{b}^{\mathrm{T}}\mathbf{c}^2,$$
$$\tilde{x}_{32}(t_{n+1}) = \tilde{x}_{32}(t_n) + h\tilde{x}_{21}(t_n) + h^2t_n\mathbf{b}^{\mathrm{T}}\mathbf{c} + h^3\mathbf{b}^{\mathrm{T}}\mathbf{Ac},$$
$$\ldots$$

Эти выражения представляют собой разложения в ряд Тейлора численных решений. Аналогичные разложения точных решений запишутся в виде:

$x_{21}(t_{n+1}) = x_{21}(t_n) + ht_n + h^2/2,$
$x_{31}(t_{n+1}) = x_{31}(t_n) + ht_n^2 + h^2t_n + h^3/3,$
$x_{32}(t_{n+1}) = x_{32}(t_n) + ht_n^2/2 + h^2t_n/2 + h^3/6,$
...

Выражения для $\tilde{x}_{ij}(t_{n+1})$ и $x_{ij}(t_{n+1})$ являются многочленами степени i переменной h, при этом из условий порядка следует, что коэффициенты этих многочленов совпадают для степеней h до p-й включительно. Поэтому переменная x_{ij} интегрируется точно методом порядка p, если $i \le p$. Если $i = p + 1$, то глобальная ошибка выражается формулой

$$\Delta_{n+1}(x_{ij}) = x_{ij}(t_{n+1}) - \tilde{x}_{ij}(t_{n+1}) = \Delta_n(x_{ij}) + \delta_h(x_{ij}),$$

где локальная ошибка $\delta_h(x_{ij})$ пропорциональна h^{p+1}. При интегрировании на интервале $[0, T]$ с постоянным размером шага $h = T/N$ получим $\Delta_N(x_{ij}) = (T/h)\,\delta_h(x_{ij})$, т. е. ошибка возрастает линейно и пропорциональна h^p. Это полностью согласуется с классической теорией.

4.6. Модельные уравнения для ДАУ индекса 1

Рассмотрим систему ДАУ

$$\begin{aligned} &\mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \\ &\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0. \end{aligned} \tag{4.18}$$

Воспользуемся методом ε-вложения, тогда один шаг неявного метода Рунге–Кутты для решения системы (4.18) запишется в виде:

$$\begin{aligned} &\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{s} b_i \mathbf{X}_i', \quad \mathbf{y}_{n+1} = \mathbf{y}_n + h\sum_{i=1}^{s} b_i \mathbf{Y}_i', \\ &\mathbf{X}_i = \mathbf{x}_n + h\sum_{j=1}^{s} a_{ij} \mathbf{X}_j', \quad \mathbf{Y}_i = \mathbf{y}_n + h\sum_{j=1}^{s} a_{ij} \mathbf{Y}_j', \\ &\mathbf{X}_i' = \mathbf{f}(\mathbf{X}_i, \mathbf{Y}_i), \quad \mathbf{0} = \mathbf{g}(\mathbf{X}_i, \mathbf{Y}_i), \quad i = 1, ..., s. \end{aligned} \tag{4.19}$$

Рассмотрим простейшее алгебраическое уравнение

$$0 = y - \varphi(t), \quad y_0 = \varphi(t_0). \tag{4.20}$$

Используя формулы (4.19) для решения этого уравнения, получаем:

$y_{n+1} = y_n + h\mathbf{b}^{\mathrm{T}}\mathbf{Y}', \quad \mathbf{e}y_n + h\mathbf{A}\mathbf{Y}' = \mathbf{\Phi} = [\varphi(t_n + c_1 h), ..., \varphi(t_n + c_s h)]^{\mathrm{T}}.$

Предположим, что матрица $\mathbf{A}$ обратима, тогда $\mathbf{Y}' = h^{-1}\mathbf{A}^{-1}(\mathbf{\Phi} - \mathbf{e}y_n)$ и

$y_{n+1} = (1 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e})y_n + \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{\Phi}.$

Переменную y можно интерпретировать как результат последовательного выполнения численных операций дифференцирования и интегрирования

переменной $\varphi(t)$. При составлении модельных уравнений вместо $\varphi(t)$ будем использовать всевозможные произведения переменных x_{ij}.

При выводе условий порядка для ДАУ используются деревья с вершинами двух видов (точки и кружки) [75]. *Порядком* такого дерева называется число вершин-точек минус число вершин-кружков. При формировании модельных уравнений будем рассматривать дерево как сигнальный граф, вершины которого отождествляются с интегрированием (точка) либо дифференцированием (кружок) произведения входных переменных. Наряду с определенными в табл. 4.4 дифференциальными переменными x_{ij} введем алгебраические переменные индекса 1 y_{ij}, которые определим как всевозможные произведения дифференциальных переменных. Полученные уравнения и их решения при нулевых начальных условиях для деревьев до 4-го порядка приведены в табл. 4.5. Эти уравнения имеют индекс 1, поскольку, продифференцировав любое из них, получим дифференциальное уравнение относительно алгебраической переменной.

Таблица 4.5. Модельные уравнения индекса 1

Дерево	Уравнение	$y_{ij}(t)$	$\tilde{y}_{ij}(h)$
	$0 = y_{21} - x_1^2$	t^2	$h^2\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{c}^2$
	$0 = y_{31} - x_1^3$	t^3	$h^3\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{c}^3$
	$0 = y_{32} - x_1x_{21}$	$t^3/2$	$h^3\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{c}(\mathbf{Ac}))$
	$0 = y_{41} - x_1^4$	t^4	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{c}^4$
	$0 = y_{42} - x_1^2x_{21}$	$t^4/2$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{c}^2(\mathbf{Ac}))$
	$0 = y_{43} - x_1x_{31}$	$t^4/3$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{c}(\mathbf{Ac}^2))$
	$0 = y_{44} - x_1x_{32}$	$t^4/6$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{c}(\mathbf{A}^2\mathbf{c}))$
	$0 = y_{45} - x_{21}^2$	$t^4/4$	$h^4\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{Ac})^2$

Исследуем точность решения алгебраической части ДАУ. Начнем с уравнения (4.20). Используя разложение $\varphi(t)$ в ряд Тейлора, получаем:

$$y_{n+1} = (1 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e})y_n + \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e}\varphi_n + \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\sum_{k=1}^{\infty}\frac{\mathbf{c}^k h^k}{k!}\frac{d^k\varphi(t_n)}{dt^k}.$$

Обозначим $\alpha_0 = 1 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e} = R(\infty)$, где $R(z) = 1 + z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}$ – функция устойчивости. Тогда выражение для глобальной ошибки запишется в виде:

$$\Delta_{n+1}(y) = \varphi_{n+1} - y_{n+1} = \alpha_0\Delta_n(y) + \sum_{k=q+1}^{\infty}(1 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{c}^k)\frac{h^k}{k!}\frac{d^k\varphi(t_n)}{dt^k},$$

где q – стадийный порядок.

Аналогичные выражения можно получить для ошибок алгебраических переменных, определенных в табл. 4.5. Переменная y_{ij} вычисляется точно, если $i \le q$. Если $i = q + 1$, то глобальная ошибка определяется только невязкой алгебраического уравнения и выражается формулой

$$\Delta_{n+1}(y_{ij}) = \alpha_0\Delta_n(y_{ij}) + \delta_h(y_{ij}), \quad \delta_h(y_{ij}) = y_{ij}(h) - \tilde{y}_{ij}(h).$$

Эта ошибка имеет оценку $O(h^{q+1})$ при $-1 \le \alpha_0 < 1$ и $O(h^q)$ при $\alpha_0 = 1$ (при $|\alpha_0| > 1$ численное решение расходится). Если $i > q + 1$, то в ошибку переменной y_{ij} могут войти составляющие, обусловленные неточным вычислением дифференциальных переменных. Если метод жесткоточный, то все алгебраические уравнения решаются точно, а ошибки алгебраических переменных определяются только ошибками дифференциальных переменных, т. е. имеют оценку $O(h^p)$. Полученные для модельных уравнений результаты согласуются с утверждениями теоремы VI.1.1 из [75, с. 423].

4.7. Жесткие модельные уравнения

На основе алгебраического уравнения (4.20) можно построить жесткое дифференциальное уравнение $y' = \lambda(y - \varphi(t)) + \varphi'(t)$, которое предложили Протеро и Робинсон для исследования феномена снижения порядка. Аналогичным образом сформируем жесткие модельные уравнения на основе алгебраических уравнений из табл. 4.5. Полученные уравнения приведены в табл. 4.6. Функции погрешности определим по аналогии с коэффициентами погрешности в виде

$$e_{ij}(z) = \delta_h(v_{ij}) / v_{ij}(h), \quad \delta_h(v_{ij}) = v_{ij}(h) - \tilde{v}_{ij}(h), \quad z = h\lambda.$$

Заметим, что $e_{i1}(z) = e_i(z)$, где $e_i(z)$ – функции погрешности (4.9), полученные при рассмотрении уравнения Протеро–Робинсона. Пусть q – стадийный порядок метода, тогда $e_{ij}(z) \equiv 0$ при $i \le q$. Поэтому методы высоких стадийных порядков имеют преимущество при решении жестких задач. Если $i = q + 1$, то все функции $e_{ij}(z)$ равны между собой. При повышении стадийного порядка уменьшается число различных функций погрешности. Например, при $q = 2$ имеем:

$$e_{31}(z) = e_{32}(z), \quad e_{41}(z) = e_{42}(z) = e_{45}(z), \quad e_{43}(z) = e_{44}(z).$$

Таблица 4.6. Жесткие модельные уравнения

Уравнение	Функция погрешности
$v'_{21} = \lambda(v_{21} - x_1^2) + 2t$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{c}^2 - 2\mathbf{Ac}) + (1 - 2\mathbf{b}^{\mathrm{T}}\mathbf{c})$
$v'_{31} = \lambda(v_{31} - x_1^3) + 3t^2$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{c}^3 - 3\mathbf{Ac}^2) + (1 - 3\mathbf{b}^{\mathrm{T}}\mathbf{c}^2)$
$v'_{32} = \lambda(v_{32} - x_1 x_{21}) + 3t^2/2$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(2\mathbf{c}(\mathbf{Ac}) - 3\mathbf{Ac}^2) + (1 - 3\mathbf{b}^{\mathrm{T}}\mathbf{c}^2)$
$v'_{41} = \lambda(v_{41} - x_1^4) + 4t^3$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{c}^4 - 4\mathbf{Ac}^3) + (1 - 4\mathbf{b}^{\mathrm{T}}\mathbf{c}^3)$
$v'_{42} = \lambda(v_{42} - x_1^2 x_{21}) + 2t^3$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(2\mathbf{c}^2(\mathbf{Ac}) - 4\mathbf{Ac}^3) + (1 - 4\mathbf{b}^{\mathrm{T}}\mathbf{c}^3)$
$v'_{43} = \lambda(v_{43} - x_1 x_{31}) + 4t^3/3$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(3\mathbf{c}(\mathbf{Ac}^2) - 4\mathbf{Ac}^3) + (1 - 4\mathbf{b}^{\mathrm{T}}\mathbf{c}^3)$
$v'_{44} = \lambda(v_{44} - x_1 x_{32}) + 2t^3/3$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(6\mathbf{c}(\mathbf{A}^2\mathbf{c}) - 4\mathbf{Ac}^3) + (1 - 4\mathbf{b}^{\mathrm{T}}\mathbf{c}^3)$
$v'_{45} = \lambda(v_{45} - x_{21}^2) + t^3$	$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(4(\mathbf{Ac})^2 - 4\mathbf{Ac}^3) + (1 - 4\mathbf{b}^{\mathrm{T}}\mathbf{c}^3)$

Теперь можно объяснить различие в поведении ошибок методов ERK312b и ERK313 на рис. 4.4. Оба метода имеют $e_3(z) \equiv 0$, но метод ERK313 имеет также и $e_{32}(z) \equiv 0$, тогда как у метода ERK312b $e_{32}(z) = z/3$. Поэтому при решении задачи Капса метод ERK313 имеет преимущество. Аналогично объясняется преимущество метода IERK313 по сравнению с IERK312b при решении жестких нелинейных задач. Но если задача линейная и неавтономная, то ошибка не содержит составляющей, соответствующей функции погрешности $e_{32}(z)$, поскольку она может быть описана только нелинейными уравнениями. Поэтому при решении линейных неавтономных задач методы IERK312b и IERK313 показывают одинаковые результаты.

Если $i = q + 1$, то глобальная ошибка переменной v_{ij} выражается формулой

$$\Delta_{n+1}(v_{ij}) = R(z)\Delta_n(v_{ij}) + e_{ij}(z)v_{ij}(h), \quad z = h\lambda, \tag{4.21}$$

и при $|R(z)| << 1$ она примерно равна локальной ошибке. При значениях $R(z)$, близких к 1, глобальная ошибка накапливается и может заметно превышать локальную ошибку. В общем случае формула для глобальной ошибки переменной v_{ij} содержит, кроме $e_{ij}(z)$, также и функции погрешности более низких порядков. Например:

$$\Delta_{n+1}(v_{31}) = R(z)\Delta_n(v_{31}) + 3t_n h^2 e_{21}(z) + h^3 e_{31}(z),$$
$$\Delta_{n+1}(v_{41}) = R(z)\Delta_n(v_{41}) + 6t_n^2 h^2 e_{21}(z) + 4t_n h^3 e_{31}(z) + h^4 e_{41}(z).$$

При $e_{21}(z) \equiv 0$ ошибки переменных v_{31}, v_{32}, а при $e_{ij}(z) \equiv 0$ для $i \leq 3$ ошибки переменных $v_{41}, \ldots, v_{45}$ выражаются формулой (4.21). В этом случае повышается порядок главного члена погрешности, что приводит к повышению точности решения жестких задач.

4.8. Функции погрешности и псевдостадийный порядок

Функции погрешности удобно определить через величины $\gamma(T_{ij})$ и $\mathbf{\Phi}(T_{ij})$, используемые в условиях порядка (2.4) и приведенные (для $i \leq 5$) в табл. 2.1.

Определение 4.1. *Функциями погрешности метода Рунге–Кутты* называются функции

$$e_{ij}(z) = z\mathbf{b}^{\mathrm{T}}(\mathbf{I}-z\mathbf{A})^{-1}\left(\frac{\gamma(T_{i+1,j})}{i+1}\boldsymbol{\Phi}(T_{i+1,j})-i\mathbf{A}\mathbf{c}^{i-1}\right)+1-i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1}, \quad (4.22)$$

$i \geq 1,\ \ j = 1, ..., n_{i+1},$

где $T_{i+1,j}$ – деревья $(i+1)$-го порядка, из корневой вершины которых выходят более одной ветви и дерево T_{21} при $i = 1$ (в табл. 2.1 это деревья T_{21}, T_{31}, T_{41}, T_{42}, T_{51}, ..., T_{55}), а n_{i+1} – число таких деревьев порядка $i + 1$.

Определение 4.2. *Псевдостадийным порядком метода Рунге–Кутты* называется наибольшее целое число $\bar{q}$, для которого выполняются условия

$$\mathbf{b}^{\mathrm{T}}\mathbf{A}^{k}\left(\frac{\gamma(T_{i+1,j})}{i+1}\boldsymbol{\Phi}(T_{i+1,j})-i\mathbf{A}\mathbf{c}^{i-1}\right)=0, \quad 1-i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1}=0, \quad (4.23)$$

$i = 1, ..., \bar{q},\ \ j = 1, ..., n_{i+1},\ \ k = 0, ..., s-1.$

Заметим, что при выполнении условия (4.23) для $k = 0, ..., s - 1$ оно выполняется также и для $k \geq s$ (это следует из того, что матрица $\mathbf{A}$ удовлетворяет своему характеристическому уравнению). Псевдостадийный порядок имеет такое же значение, как и стадийный порядок: его повышение позволяет избежать снижения реального порядка при решении жестких задач. В то же время он не имеет таких ограничений, как стадийный порядок; псевдостадийный порядок явных и обратных к ним методов может быть выше 1-го, диагонально-неявных методов – выше 2-го, а мононеявных методов – выше 3-го. В [69, 70] было дано определение псевдостадийного порядка через функции погрешности $e_{ij}(z)$: метод имеет псевдостадийный порядок $\bar{q}$, если все функции $e_{ij}(z)$ порядков $i \leqslant \bar{q}$ тождественно равны нулю.

Теорема 4.2. *Порядок p, псевдостадийный порядок $\bar{q}$ и стадийный порядок q метода Рунге–Кутты удовлетворяют неравенству $p \geqslant \bar{q} \geqslant q$. А если выполняется условие* $1 - (\bar{q} + 1)\mathbf{b}^{\mathrm{T}}\mathbf{c}^{\bar{q}} = 0$, *то* $p \geqslant \bar{q} + 1$.

Доказательство. При принятом в [74] упорядочении деревьев T_{i1} – «куст», имеющий $i - 1$ ветвей, выходящих из корневой вершины, $\boldsymbol{\Phi}(T_{i1}) = \mathbf{c}^{i-1}$, $\gamma(T_{i1}) = i$. Из (4.23) видно, что для деревьев T_{i1}, $i = 1, \ldots. \bar{q}$ условия порядка (2.4) выполняются. При $i > 2$, $m = n_i + 1$ имеем $\boldsymbol{\Phi}(T_{im}) = \mathbf{A}\mathbf{c}^{i-2}$, $\gamma(T_{im}) = i(i - 1)$. Из (4.23) при $k = 0$ получаем $\gamma(T_{i1})\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(T_{i1}) = \gamma(T_{im})\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(T_{im}) = 1$, $\gamma(T_{ij})\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(T_{ij}) = \gamma(T_{im})\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(T_{im})$, $i = 3, ..., \bar{q}$, $j = 2, ..., n_i$, $m = n_i + 1$, откуда следует выполнение условий порядка для деревьев T_{ij}, $i = 3, ..., \bar{q}$, $j = 2, ..., n_i + 1$. При $\bar{q} \geqslant 3$ выполняются все условия 3-го порядка. Из приведенного в [74] правила вычисления $\gamma(T)$ следует, что если T_i – некоторое дерево порядка i, а T_{i+1} – дерево, для которого $\boldsymbol{\Phi}(T_{i+1}) = \mathbf{A}\boldsymbol{\Phi}(T_i)$, то $\gamma(T_{i+1}) = (i + 1)\gamma(T_i)$. Поэтому из выполнения условий i-го порядка, а также условий $1 - (i + 1)\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i} = 0$ и $\mathbf{b}^{\mathrm{T}}(\mathbf{c}^{i} - i\mathbf{A}\mathbf{c}^{i-1}) = 0$ (последнее – условие (4.23) при $j = 1, k = 0$) следует выполнение условий порядка для всех деревьев $(i + 1)$-го по-

рядка, у которых из корневой вершины выходит только одна ветвь. При $i \geq 2$ это деревья $T_{i+1,j}$, $j = n_{i+1} + 1, \ldots, N_{i+1} = n_{i+1} + n_i$. Поскольку при $\bar{q} \geqslant 3$все условия 3-го порядка выполняются, то, применяя математическую индукцию, получаем также и выполнение всех условий порядка $\bar{q}$. Аналогично доказывается выполнение всех условий порядка $\bar{q} + 1$ при выполнении дополнительного условия $1 - (\bar{q} + 1)\mathbf{b}^{\mathrm{T}}\mathbf{c}^{\bar{q}} = 0$.

Докажем теперь, что $\bar{q} \geqslant q$. Для этого достаточно показать, что при $\bar{q} = q$ из (2.4) следует (4.23). При выполнении условий (2.4) будут выполняться и условия (4.23) для $i = 1, \ldots, q$, $j = 1$. А из неравенства $p \geq q$ (оно следует из теоремы 4.1) следует, что будут выполняться все условия (4.23) и для $i = 1, \ldots, q$, $j > 1$ (доказательство аналогично приведенному выше).

Теорема 4.3. *Условия* (4.23) *и условия*

$$e_{ij}(z) \equiv 0, \quad i = 1, \ldots, \bar{q}, \quad j = 1, \ldots, n_{i+1} \tag{4.24}$$

эквивалентны.

Доказательство. $(\mathbf{I} - z\mathbf{A})^{-1}$, как функцию от матрицы $z\mathbf{A}$, можно представить в виде матричного многочлена от $z\mathbf{A}$, степень которого меньше s [11]. Из этого следует, что при выполнении условий (4.23) будут выполняться и условия (4.24). Покажем теперь, что из (4.24) следует (4.23). Пусть $e_{ij}(z) \equiv 0$. Запишем функцию (4.22) в виде $e_{ij}(z) = z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{v} + (1 - i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1})$, тогда $1 - i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1} = 0$, $\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{v} \equiv 0$. Используя равенство $(\mathbf{I} - \mathbf{B})^{-1} = \mathbf{I} + \mathbf{B}(\mathbf{I} - \mathbf{B})^{-1}$, $\mathbf{B} = z\mathbf{A}$ (его справедливость легко проверить, умножив обе части на $\mathbf{I} - \mathbf{B}$), получаем $\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{v} = \mathbf{b}^{\mathrm{T}}\mathbf{v} + z\mathbf{b}^{\mathrm{T}}\mathbf{A}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{v}$, откуда $\mathbf{b}^{\mathrm{T}}\mathbf{v} = 0$, $\mathbf{b}^{\mathrm{T}}\mathbf{A}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{v} \equiv \mathbf{0}$. Продолжая аналогично, получаем $\mathbf{b}^{\mathrm{T}}\mathbf{A}^k\mathbf{v} = 0$, $k = 0, \ldots, s-1$, т. е. все условия (4.23) выполняются.

Для жесткоточных методов формулы (4.22), (4.23) можно упростить. Запишем условие жесткой точности в виде $\mathbf{b}^{\mathrm{T}} = \mathbf{e}_s^{\mathrm{T}}\mathbf{A}$, $\mathbf{e}_s = [0, \ldots, 0, 1]^{\mathrm{T}}$, $c_s = 1$, тогда из (4.23) получаем:

$$\mathbf{e}_s^{\mathrm{T}}\mathbf{A}^k\left(\frac{\gamma(T_{i+1,j})}{i+1}\mathbf{\Phi}(T_{i+1,j}) - i\mathbf{A}\mathbf{c}^{i-1}\right) = 0, \quad 1 - i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1} = 0,$$

$i = 1, \ldots, \bar{q}$, $j = 1, \ldots, n_{i+1}$, $k = 1, \ldots, s$.

Поскольку матрица $\mathbf{A}$ удовлетворяет своему характеристическому уравнению, значения k можно сдвинуть, задав $k = 0, \ldots, s-1$. В этом случае при $j = 1$, $k = 0$ условие $1 - i\mathbf{b}^{\mathrm{T}}\mathbf{c}^{i-1} = 0$ дублируется и его можно исключить, в результате вместо (4.23) получаем условия

$$\mathbf{e}_s^{\mathrm{T}}\mathbf{A}^k\left(\frac{\gamma(T_{i+1,j})}{i+1}\mathbf{\Phi}(T_{i+1,j}) - i\mathbf{A}\mathbf{c}^{i-1}\right) = 0, \quad i = 1, \ldots, \bar{q}, \quad j = 1, \ldots, n_{i+1}, \quad k = 0, \ldots, s-1. \tag{4.25}$$

Чтобы получить аналогичные формулы для $e_{ij}(z)$, последовательно подставляем в (4.22) соотношения

$$z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1} = \mathbf{e}_s^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1} - \mathbf{e}_s^{\mathrm{T}}, \quad \gamma(T_{i+1,j}) = (i+1)\prod_{l=1}^{k}\gamma(\tau_l), \quad \mathbf{e}_s^{\mathrm{T}}\mathbf{\Phi}(T_{i+1,j}) = \mathbf{b}^{\mathrm{T}}\prod_{l=1}^{k}\mathbf{\Phi}(\tau_l),$$

где $\tau_1, \ldots, \tau_k$ – деревья, полученные из дерева $T_{i+1,j}$ после удаления корня вместе с инцидентными ему ребрами (например, для дерева T_{53} из табл. 2.1 это $\tau_1 = T_1$ и $\tau_2 = T_{31}$). Пусть $i \le p$, тогда $\gamma(\tau_i)\mathbf{b}^{\mathrm{T}}\boldsymbol{\Phi}(\tau_i) = 1$, $i\mathbf{e}_s^{\mathrm{T}}\mathbf{A}\mathbf{c}^{i-1} = 1$. В результате выполнения соответствующих подстановок получаем:

$$e_{ij}(z) = \mathbf{e}_s^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\left(\frac{\gamma(T_{i+1,j})}{i+1}\boldsymbol{\Phi}(T_{i+1,j}) - i\mathbf{A}\mathbf{c}^{i-1}\right), \quad 1 \le i \le p, \quad j = 1, \ldots, n_{i+1}. \tag{4.26}$$

Рассмотрим теперь жесткоточные методы с явной первой стадией, т. е. методы вида (4.4), в которых $\tilde{\mathbf{b}}^{\mathrm{T}} = \mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}$. Для таких методов при $\bar{q} > 1$ условия псевдостадийного порядка (4.25) запишутся в виде:

$$\mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}^k\left(\frac{\gamma(T_{i+1,j})}{i+1}\tilde{\boldsymbol{\Phi}}(T_{i+1,j}) - i\tilde{\mathbf{A}}\tilde{\mathbf{c}}^{i-1}\right) = 0, \quad i = 2, \ldots, \bar{q}, \quad j = 1, \ldots, n_{i+1}, \quad k = 0, \ldots, s-2, \tag{4.27}$$

а функции погрешности (4.26) – в виде:

$$e_{ij}(z) = \mathbf{e}_{s-1}^{\mathrm{T}}(\mathbf{I} - z\tilde{\mathbf{A}})^{-1}\left(\frac{\gamma(T_{i+1,j})}{i+1}\tilde{\boldsymbol{\Phi}}(T_{i+1,j}) - i\tilde{\mathbf{A}}\tilde{\mathbf{c}}^{i-1}\right), \quad 2 \le i \le p, \quad j = 1, \ldots, n_{i+1}, \tag{4.28}$$

где выражение для $\tilde{\Phi}(T_{i+1,j})$ получено путем замены $\mathbf{A}$ на $\tilde{\mathbf{A}}$ и $\mathbf{c}$ на $\tilde{\mathbf{c}}$. По сравнению с (4.23), (4.22), выражения (4.27), (4.28) проще и удобнее для построения методов с минимизированными либо нулевыми функциями погрешности.

4.9. Модельные уравнения для ДАУ индекса 2

Дополним модельные уравнения, приведенные в табл. 4.4 и 4.5, уравнениями индекса 2. Алгебраические переменные индекса 2 z_{ij} определим с помощью уравнений, представленных в табл. 4.7 (ограничимся деревьями не выше 2-го порядка). Эти переменные получены двумя разными способами: z_{11}, z_{21} и z_{22} – путем дифференцирования переменных индекса 1, а z_{23} и z_{24} – в виде произведения определенных ранее переменных, среди которых должна быть хотя бы одна переменная индекса 2. Интегрируя переменные индекса 2, получаем дифференциальные переменные индекса 1, среди которых есть уже определенные в табл. 4.5 переменные, а также новые переменные, уравнения относительно которых представлены в табл. 4.8.

Таблица 4.7. Модельные уравнения индекса 2 (алгебраические переменные)

Дерево	Уравнение	$z_{ij}(t)$	$\tilde{z}_{ij}(h)$
	$y'_{21} = z_{11}$	$2t$	$h\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{c}^2$
	$y'_{31} = z_{21}$	$3t^2$	$h^2\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{c}^3$
	$y'_{32} = z_{22}$	$3t^2/2$	$h^2\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}(\mathbf{c}(\mathbf{A}\mathbf{c}))$

Окончание табл. 4.7

Дерево	Уравнение	$z_{ij}(t)$	$\tilde{z}_{ij}(h)$
	$0 = z_{23} - x_1 z_{11}$	$2t^2$	$h^2\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{c}(\mathbf{A}^{-1}\mathbf{c}^2))$
	$0 = z_{24} - z_{11}^2$	$4t^2$	$h^2\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{A}^{-1}\mathbf{c}^2)^2$

Таблица 4.8. Модельные уравнения индекса 2 (дифференциальные переменные)

Дерево	Уравнение	$y_{ij}(t)$	$\tilde{y}_{ij}(h)$
	$y'_{33} = z_{23}$	$2t^3/3$	$h^3\mathbf{b}^{\mathrm{T}}(\mathbf{c}(\mathbf{A}^{-1}\mathbf{c}^2))$
	$y'_{34} = z_{24}$	$4t^3/3$	$h^3\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{-1}\mathbf{c}^2)^2$

ДАУ индекса 2 обычно представляют в виде (4.5а, б), при котором алгебраическое уравнение не содержит алгебраических переменных. Модельные уравнения имеют более общий вид (4.3), однако, введя дополнительные переменные, можно избавиться от переменных z_{ij} в алгебраических уравнениях. Например, задав $y'_{33a} = x_1 z_{11}$, можно вместо уравнения $0 = z_{23} - x_1 z_{11}$ записать $0 = y_{33a} - y_{33}$. Численные решения при этом не изменятся. Аналогичное замечание справедливо и для уравнений индекса 3.

Для дальнейшего изложения нам нужно определить операцию численного дифференцирования с помощью неявного метода Рунге–Кутты. Система ДАУ индекса 2, осуществляющая дифференцирование переменной $\varphi(t)$, запишется в виде:

$$y' = z, \quad y_0 = \varphi(t_0),$$
$$0 = y - \varphi(t), \quad z_0 = \varphi'(t_0).$$

В эту систему входит алгебраическое уравнение, которое уже рассматривалось, а также дифференциальное уравнение, задающее искомую переменную z. Используя неявный метод Рунге–Кутты, получаем:

$$\begin{aligned} &y_{n+1} = y_n + h\mathbf{b}^{\mathrm{T}}\mathbf{Z}, \quad z_{n+1} = z_n + h\mathbf{b}^{\mathrm{T}}\mathbf{Z}', \\ &\mathbf{Y} = \mathbf{e}y_n + h\mathbf{A}\mathbf{Z}, \quad \mathbf{Z} = \mathbf{e}z_n + h\mathbf{A}\mathbf{Z}', \\ &\mathbf{Y} = \mathbf{\Phi} = [\varphi(t_n + c_1 h), \ldots, \varphi(t_n + c_s h)]^{\mathrm{T}}. \end{aligned} \tag{4.29}$$

Последовательно исключим векторы $\mathbf{Z}'$, $\mathbf{Z}$, $\mathbf{Y}$. Тогда для методов с обратимой матрицей $\mathbf{A}$ получим:

$$\begin{aligned} &y_{n+1} = \alpha_0 y_n + \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{\Phi}, \quad z_{n+1} = \alpha_0 z_n + \alpha_1 h^{-1} y_n + h^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{\Phi}, \\ &\alpha_0 = 1 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e} = R(\infty), \quad \alpha_1 = -\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{e} = \lim_{z\to\infty} z(R(z) - \alpha_0). \end{aligned} \tag{4.30}$$

Формулы (4.29), (4.30) использовались при выводе выражений для глобальных ошибок решения модельных уравнений индекса 2, при этом вместо переменной $\varphi(t)$ задавались произведения переменных, входящих в модельные уравнения.

В общем случае имеем:

$$\begin{aligned}\Delta_{n+1}(y_{21}) &= \alpha_0\Delta_n(y_{21}) + \delta_h(y_{21}),\\ \Delta_{n+1}(z_{11}) &= \alpha_0\Delta_n(z_{11}) + \alpha_1 h^{-1}\Delta_n(y_{21}) + \delta_h(z_{11}).\end{aligned} \tag{4.31}$$

Если выполняются условия $\mathbf{b}^{\mathrm{T}}\mathbf{c} = 1/2$, $\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{c}^2 = 1$, то

$$\Delta_{n+1}(y_{3j}) = \begin{cases}\alpha_0\Delta_n(y_{3j}) + \delta_h(y_{3j}), & j = 1, 2,\\ \Delta_n(y_{3j}) + \delta_h(y_{3j}), & j = 3, 4,\end{cases} \tag{4.32}$$

а если к перечисленным условиям добавить $\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{c}^2 = 2$, то

$$\Delta_{n+1}(z_{2j}) = \begin{cases}\alpha_0\Delta_n(z_{2j}) + \alpha_1 h^{-1}\Delta_n(y_{3j}) + \delta_h(z_{2j}), & j = 1, 2,\\ \alpha_0\Delta_n(z_{2j}) + \delta_h(z_{2j}), & j = 3, 4.\end{cases} \tag{4.33}$$

Согласно приведенным формулам, при $|\alpha_0| < 1$ глобальные ошибки переменных y_{21}, y_{31}, y_{32} и всех z_{ij} асимптотически ведут себя как локальные ошибки и имеют такие же порядки. Если $\alpha_0 = \alpha_1 = 0$ (таким свойством обладают $L2$-устойчивые схемы [19]), то глобальные ошибки этих переменных совпадают с локальными ошибками. Напротив, глобальные ошибки переменных y_{33} и y_{34} накапливаются, а их порядок на единицу меньше порядка локальных ошибок.

Сходимость методов Рунге–Кутты с обратимой матрицей $\mathbf{A}$ при решении ДАУ индекса 2 исследовалась в [109]. Как следствие из теорем 4.4 и 4.6 этой работы (в [75] они приведены под номерами VII.4.5 и VII.4.6) методы SDIRK при $|R(\infty)| < 1$ и $p \geq 2$ имеют 2-й порядок сходимости дифференциальных переменных (y-компонента) и 1-й порядок сходимости алгебраических переменных (z-компонента). Это нижние оценки; при выполнении некоторых дополнительных условий соответствующие порядки могут быть выше. Анализ модельных уравнений подтверждает эти оценки. Из (4.31)–(4.33) следует, что методы 1-го стадийного порядка (например, SDIRK) могут обеспечить 2-й порядок сходимости переменных y_{21}, y_{33}, y_{34} и 1-й порядок сходимости переменной z_{11}.

Попробуем найти жесткоточные методы SDIRK, имеющие $p_y = 3$ и $p_z = 2$, введя дополнительные условия. Чтобы обеспечить требуемые порядки для ДАУ индекса 2 общего вида, необходимо, чтобы эти порядки обеспечивались при решении модельных уравнений из табл. 4.4, 4.5, 4.7, 4.8. Прежде всего необходимо выполнение условий 3-го классического порядка

$$\mathbf{b}^{\mathrm{T}}\mathbf{e} = 1, \quad \mathbf{b}^{\mathrm{T}}\mathbf{c} = 1/2, \quad \mathbf{b}^{\mathrm{T}}\mathbf{c}^2 = 1/3, \quad \mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{c} = 1/6, \tag{4.34}$$

что обеспечивает точное вычисление переменных x_{ij} при $i \leq 3$ и y_{ij} из табл. 4.4 и 4.5. Для жесткоточного метода SDIRK имеем $\alpha_0 = 0$, поэтому глобальная ошибка переменной z_{11} совпадает с локальной ошибкой и равна $h(2 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{c}^2)$, откуда получаем необходимое условие 2-го порядка z-компоненты в виде

$$\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{c}^2 = 2. \tag{4.35}$$

При выполнении (4.34) и (4.35) глобальные ошибки переменных z_{2j} совпадают с локальными ошибками и пропорциональны h^2. Ошибки переменных y_{33} и y_{34} накапливаются согласно формуле (4.32), откуда следует, что необходимым условием 3-го порядка y-компоненты является их равенство нулю, т. е. выполнение условий

$$\mathbf{b}^{\mathrm{T}}(\mathbf{c}(\mathbf{A}^{-1}\mathbf{c}^2)) = 2/3, \quad \mathbf{b}^{\mathrm{T}}(\mathbf{A}^{-1}\mathbf{c}^2)^2 = 4/3. \tag{4.36}$$

Таким образом, условия (4.34)–(4.36) являются необходимыми для достижения требуемых порядков.

Примем $s = 4$, тогда для определения семи параметров матрицы $\mathbf{A}$ (a_{21}, a_{31}, a_{32}, b_1, b_2, b_3 и γ) нужно решить систему из семи алгебраических уравнений (4.34)–(4.36). Были найдены семь решений этих уравнений, которые можно записать следующим образом. Принимаем γ равным 1/4 либо одному из корней уравнения

$$36\gamma^6 - 252\gamma^5 + 540\gamma^4 - 432\gamma^3 + 147\gamma^2 - 21\gamma + 1 = 0.$$

Остальные параметры находим по формулам:

$$c_2 = \gamma\frac{6\gamma^3 - 24\gamma^2 + 15\gamma - 2}{6\gamma^3 - 18\gamma^2 + 9\gamma - 1}, \quad c_3 = \frac{10\gamma^2 - 6\gamma + 1}{6\gamma^2 - 4\gamma + 1}, \quad a_{21} = c_2 - \gamma,$$

$$a_{32} = \frac{(c_3 - \gamma)(c_3 - c_2)(6\gamma^3 - 18\gamma^2 + 9\gamma - 1)}{(c_2 - \gamma)\left[(6\gamma^2 - 12\gamma + 3)c_2 - 6\gamma^2 + 9\gamma - 2\right]}, a_{31} = c_3 - a_{32} - \gamma,$$

$$b_1 = 0, \quad b_2 = \frac{6\gamma(1 - c_3) + 3c_3 - 2}{6c_2(c_3 - c_2)}, \quad b_3 = 1 - b_2 - \gamma.$$

Метод, имеющий $\gamma = 1/4$, был построен в [95] на основе условий 2-го квазистадийного порядка, которые можно записать в виде:

$$\mathbf{b}(\mathbf{c}^2 - 2\mathbf{A}\mathbf{c}) = \mathbf{0}, \tag{4.37а}$$

$$\mathbf{b}(\mathbf{A}^{-1}\mathbf{c}^2 - 2\mathbf{c}) = \mathbf{0}. \tag{4.37б}$$

Таблица Бутчера этого метода:

1/4	1/4			
11/28	1/7	1/4		
1/3	61/144	−49/144	1/4	
1	0	0	3/4	1/4
	0	0	3/4	1/4

. (4.38)

Остальные 6 методов удовлетворяют условию (4.37б), но не удовлетворяют условию (4.37а). Среди полученных методов два являются L-устойчивыми ($\gamma = 1/4$ и $\gamma = 0.340399$), а остальные, за исключением метода, имеющего $\gamma = 0.069132$, являются $L(\alpha)$-устойчивыми.

Мы рассмотрели неявные методы с обратимой матрицей **A**. Но при решении систем ДАУ используют также неявные методы вида (4.4) с вырожденной матрицей **A**. Для таких методов приведенные выше формулы будут справедливы, если заменить **A**, **b** и **c** на $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$ и $\tilde{\mathbf{c}}$, а α_0 и α_1 вычислять по формулам

$$\alpha_0 = \lim_{z\to\infty} R(z) = 1 - \tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}, \quad \alpha_1 = \lim_{z\to\infty} z(R(z)-\alpha_0) = -\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-3}\tilde{\mathbf{c}}.$$

Кроме этого, для выполнения соотношений (4.32) при $j = 3, 4$ дополнительно должно выполняться равенство $\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^2 = 2$.

Метод (4.38), а также методы 3-го порядка, определенные в разделах 4.2 и 4.4, использовались для решения системы индекса 2:

$$y_1' = -(y_1y_2z)^{1/4}, \quad y_2' = -y_1(y_1^2 + y_2)/z, \quad 0 = y_1^2 - y_2,$$
$$y_1(0) = y_2(0) = z(0) = 1, \quad 0 \leqslant t \leqslant 1, \tag{4.39}$$

точное решение которой $y_1(t) = z(t) = \exp(-t)$, $y_2(t) = \exp(-2t)$. В табл. 4.9 приведены ошибки и оценки порядка по соответствующим компонентам при $h = 1/30$. Для методов SDIRK31b, ESDIRK32 и DESI33 полученные оценки соответствуют известным теоретическим результатам, приведенным в табл. 4.1. Более высокие порядки сходимости метода (4.38), по сравнению с SDIRK31b, объясняются выполнением дополнительных условий (4.35) и (4.36). Метод IERK313 демонстрирует такую же сходимость, как и метод более высокого стадийного порядка DESI33, что можно объяснить одинаковыми значениями $\bar{q}$ (хотя теоретического обоснования для этого случая у нас нет).

Таблица 4.9. Результаты решения системы ДАУ индекса 2

Метод	q	$\bar{q}$	e_y	e_z	$\tilde{p}_y$	$\tilde{p}_z$
SDIRK31b	1	1	1.10×10^{-5}	5.72×10^{-3}	2.07	1.02
ESDIRK32	2	2	7.38×10^{-7}	5.92×10^{-5}	2.98	2.03
DESI33	3	3	1.11×10^{-6}	3.85×10^{-6}	2.99	2.95
(4.38)	1	1	1.29×10^{-7}	1.24×10^{-4}	2.98	1.98
IERK313	1	3	1.73×10^{-6}	1.30×10^{-6}	2.97	2.97

4.10. Модельные уравнения для ДАУ индекса 3

Модельные уравнения индекса 3 строятся аналогично уравнениям индекса 2. Сначала определяем алгебраические переменные индекса 3 путем дифференцирования переменных индекса 2 либо в виде произведений определенных ранее переменных, среди которых должна быть хотя бы одна переменная индекса 3. Интегрируя эти произведения, определяем дополнительные дифференциальные переменные. Деревья самых низких порядков изображены на рис. 4.6, а соответствующие им уравнения приведены в табл. 4.10. На этот раз имеется бесконечное число деревьев заданного порядка. Отметим также, что

появились деревья нулевого порядка, причем локальные ошибки соответствующих переменных $\delta_h(u_{0j}) = u_{0j}(h) - \tilde{u}_{0j}(h)$ не зависят от размера шага.

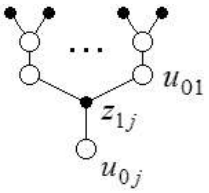


Рис. 4.6. Деревья низких порядков для ДАУ индекса 3

Таблица 4.10. Модельные уравнения индекса 3

Переменная	Уравнение	$u(t)$, $z(t)$	$u(h)$, $\tilde{z}(h)$
u_{01}	$z'_{11} = u_{01}$	2	$\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-3}\mathbf{c}^2 + 2\alpha_0$
$u_{0j}(j \geq 2)$	$0 = u_{0j} - u_{01}{}^j$	2^j	$\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{A}^{-2}\mathbf{c}^2)^j + 2^j\alpha_0$
$z_{1j}(j \geq 2)$	$z'_{1j} = u_{0j}$	$2^j t$	$h\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{-2}\mathbf{c}^2)^j$

Методы стадийного порядка 2 или выше точно решают эти уравнения. Проанализируем ошибки, предположив, что метод имеет 1-й стадийный порядок и обратимую матрицу **A**. Локальные ошибки переменных u_{0j} не зависят от величины шага. Глобальные ошибки переменных z_{1j} при $j \geq 2$ могут накапливаться, тогда они имеют нулевой порядок. Поэтому методы, имеющие $q = 1$, в общем случае не обеспечивают сходимости и непригодны для решения систем ДАУ индекса 3. С помощью модельных уравнений попробуем вывести условия, обеспечивающие 1-й порядок сходимости дифференциальных и алгебраических компонент. На основе этих условий построим методы 1-го стадийного порядка, пригодные для решения ДАУ индекса 3.

Необходимым условием 1-го порядка u-компоненты является точное вычисление всех переменных u_{0j}. Выражение для глобальной ошибки переменной u_{01} имеет вид

$$\Delta_{n+1}(u_{01}) = \alpha_0\Delta_n(u_{01}) + \alpha_1 h^{-1}\Delta_n(z_{11}) + \alpha_2 h^{-2}\Delta_n(y_{21}) + \delta_h(u_{01}),$$

где $\alpha_2 = -\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-3}\mathbf{e}$. Предположим, что метод жесткоточный, тогда $\alpha_0 = 0$, $\Delta_n(y_{21}) = 0$ и

$$\Delta_{n+1}(u_{01}) = \alpha_1 h^{-1}\Delta_n(z_{11}) + \delta_h(u_{01}),$$

где $\Delta_n(z_{11}) = \bar{h}(2 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-2}\mathbf{c}^2)$, $\delta_h(u_{01}) = 2 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-3}\mathbf{c}^2$, $\bar{h} = t_n - t_{n-1}$ – размер предыдущего шага. Возможны 2 варианта, при которых ошибки всех переменных u_{0j} равны 0: 1) если $\alpha_1 = 0$, $\delta_h(u_{01}) = 0$; 2) если $\Delta_n(z_{11}) = 0$, $\delta_h(u_{01}) = 0$.

Простейший метод, реализующий 1-й вариант, имеет таблицу Бутчера

$$\begin{array}{c|cc} 1/2 & 1/2 & \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}.$$

Один шаг этого метода можно рассматривать как 2 шага неявного метода Эйлера размером $h/2$. При использовании этого метода глобальные ошибки всех переменных z_{1j} равны локальным ошибкам и имеют 1-й порядок. Неявный метод Эйлера обладает таким же свойством, но он не обеспечивает 1-го порядка сходимости переменных u_{0j}, поскольку при любом изменении размера шага (а также на 1-м шаге) возникает ошибка, зависящая только от соотношения шагов и не зависящая от величины шага.

Методы, построенные согласно 2-му варианту, обладают тем свойством, что глобальные ошибки переменных z_{1j} при $j \geq 2$ накапливаются и выражаются формулой $\Delta_{n+1}(z_{1j}) = \Delta_n(z_{1j}) + \delta_h(z_{1j})$. В этом случае необходимым условием 1-го порядка сходимости дифференциальных переменных является равенство нулю ошибок переменных z_{1j}. Потребовав точного вычисления всех переменных u_{0j} и z_{1j}, получим условия

$$\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-3}\mathbf{c}^2 = 2, \quad \mathbf{b}^{\mathrm{T}}(\mathbf{A}^{-2}\mathbf{c}^2)^j = 2^j, \quad j \geq 1, \tag{4.40}$$

которые будут выполняться только в том случае, когда все компоненты вектора $\mathbf{b}(\mathbf{A}^{-2}\mathbf{c}^2 - 2\mathbf{e})$ равны 0. При $s = 3$ этим условиям удовлетворяют 4 метода SDIRK 1-го порядка, имеющие $\gamma = (3 \pm \sqrt{5})/2$ и $\gamma = (5 \pm \sqrt{13})/6$. Единственный среди них L-устойчивый метод задается таблицей

$$\begin{array}{c|ccc} \gamma & \gamma & & \\ 0 & -\gamma & \gamma & \\ 1 & 0 & 1-\gamma & \gamma \\ \hline & 0 & 1-\gamma & \gamma \end{array}, \quad \gamma = \frac{3-\sqrt{5}}{2} = 0.381966\ldots$$

Заметим, что число γ делит интервал [0, 1] в пропорции золотого сечения.

Попробуем теперь построить метод SDIRK, обеспечивающий 2-й порядок сходимости дифференциальных и 1-й порядок алгебраических переменных. Для этого к условиям (4.40) необходимо добавить условие 2-го классического порядка $\mathbf{b}^{\mathrm{T}}\mathbf{c} = 1/2$. Приняв $s = 4$, мы нашли три однопараметрических семейства жесткоточных методов SDIRK, удовлетворяющих всем этим условиям. Свободным параметром в них является диагональный элемент γ. Первое семейство получаем при $b_1 = c_2 = 0$, второе – при $b_1 = 0$, $c_2 = 2\gamma$, а третье – при $b_1 = b_2 = 0$. Коэффициенты методов первого семейства вычисляем по формулам:

$$c_1 = \gamma, \quad c_2 = 0, \quad c_3 = \gamma\frac{2\gamma^2 - 8\gamma + 3}{1-2\gamma}, \quad a_{21} = -\gamma, \quad a_{32} = c_3\frac{c_3 - 2\gamma}{\gamma},$$

$$a_{31} = c_3 - a_{32} - \gamma, \quad b_1 = 0, \quad b_3 = \frac{1-2\gamma}{2c_3}, \quad b_2 = 1 - b_3 - \gamma.$$

Потребовав дополнительно $L2$-устойчивости (т. е. $\alpha_1 = 0$), получим метод с таблицей коэффициентов

$$\begin{array}{c|cccc} 1/4 & 1/4 & & & \\ 0 & -1/4 & 1/4 & & \\ 9/16 & 11/64 & 9/64 & 1/4 & \\ 1 & 0 & 11/36 & 4/9 & 1/4 \\ \hline & 0 & 11/36 & 4/9 & 1/4 \end{array} . \tag{4.41}$$

Численные эксперименты показали, что построенные методы SDIRK, в отличие от известных методов SDIRK, приведенных, например, в [75, 81], действительно обеспечивают сходимость при решении ДАУ индекса 3. Для примера возьмем систему индекса 3

$$\begin{aligned} &y_1' = -(y_1 y_2 z_1 z_2)^{1/6}, \quad y_2' = y_1(y_2 - 3z_2)/z_1, \\ &z_1' = -z_1 z_2 u/(y_1 y_2), \quad z_2' = -(y_1 y_2 + z_1 z_2)/u, \\ &0 = y_1^2 - y_2, \quad y_1(0) = y_2(0) = z_1(0) = z_2(0) = u(0) = 1, \quad 0 \le t \le 1, \end{aligned} \tag{4.42}$$

с точным решением $y_1(t) = z_1(t) = u(t) = \exp(-t)$, $y_2(t) = z_2(t) = \exp(-2t)$. Для ее решения использовались известные методы, приведенные в разделе 4.2, а также методы, полученные в результате исследования модельных уравнений. Ошибки и оценки порядков при $h = 1/30$ приведены в табл. 4.11. Метод SDIRK31b, предложенный в [81], не обеспечивает сходимости. В то же время методы (4.41) и IERK313 обеспечивают сходимость как у методов более высокого стадийного порядка ESDIRK32 и DESI33. Отметим, что известные теоретические результаты о сходимости при решении ДАУ индекса 3 [117] получены при предположении, что матрица **A** обратима и $q \ge 2$. Ни один метод из табл. 4.11 этим условиям не удовлетворяет.

Таблица 4.11. Результаты решения системы ДАУ индекса 3

Метод	q	$\bar{q}$	e_y	e_z	e_u	$\tilde{p}_y$	$\tilde{p}_z$	$\tilde{p}_u$
SDIRK31b	1	1	1.70×10^{-3}	7.81×10^{-3}	3.78×10^{-1}	0.08	0.09	0.00
ESDIRK32	2	2	1.62×10^{-6}	4.15×10^{-5}	1.18×10^{-3}	1.86	2.00	1.07
DESI33	3	3	2.04×10^{-7}	2.87×10^{-6}	9.57×10^{-5}	3.00	2.97	2.03
(4.41)	1	1	6.29×10^{-6}	1.07×10^{-4}	7.04×10^{-4}	2.02	1.98	1.23
IERK313	1	3	3.85×10^{-7}	9.27×10^{-7}	2.25×10^{-5}	2.88	2.96	1.95

Диагонально-неявные методы Рунге–Кутты

ГЛАВА 5

5.1. Функция устойчивости

Диагонально-неявные методы Рунге–Кутты (DIRK) наиболее просты в реализации, но их точность при решении жестких задач ограничена невысоким стадийным порядком. Поэтому они наиболее эффективны при низкой и умеренной точности ($Rtol = 10^{-2}...10^{-6}$). В практических вычислениях применяют методы DIRK двух типов: однократно диагонально-неявные (SDIRK) с таблицей Бутчера вида

$$\begin{array}{c|ccccc} \gamma & \gamma & & & & \\ c_2 & a_{21} & \gamma & & & \\ c_3 & a_{31} & a_{32} & \gamma & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & \gamma \\ \hline & b_1 & b_2 & b_3 & \cdots & b_s \end{array}$$

и аналогичные методы с явной 1-й стадией (ESDIRK) и таблицей вида

$$\begin{array}{c|ccccc} 0 & 0 & & & & \\ c_2 & a_{21} & \gamma & & & \\ c_3 & a_{31} & a_{32} & \gamma & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & \gamma \\ \hline & b_1 & b_2 & b_3 & \cdots & b_s \end{array}.$$

Жесткоточные методы, для которых $b_i = a_{si}$, имеют ощутимое преимущество при решении жестких задач и ДАУ, поэтому будем рассматривать только такие методы, а таблицу Бутчера будем задавать в виде

$$
\begin{array}{c|ccccc}
c_1 & c_1 & & & & \\
c_2 & a_{21} & \gamma & & & \\
c_3 & a_{31} & a_{32} & \gamma & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
b_i & b_1 & b_2 & b_3 & \cdots & \gamma \\
\hline
\hat{b}_i & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \cdots & \hat{b}_s
\end{array}
$$

Поскольку коэффициенты b_i совпадают с коэффициентами последней стадии, под чертой приводим только коэффициенты $\hat{b}_i$ вложенной формулы, порядок которой на 1 меньше порядка метода. В приведенных ниже методах мы принимаем $\hat{b}_s = 0$, что позволяет использовать эту же формулу в качестве начального приближения для итераций на последней стадии. Вектор $K\hat{\mathbf{b}} + (1 - K)\mathbf{b}$ также задает вложенную формулу, где K – масштабный коэффициент для оценки ошибки. Если исходная формула (при $\hat{b}_s = 0$) переоценивает ошибку, то следует задать $K < 1$, а если недооценивает, то $K > 1$.

Сравним потенциальные возможности методов этих типов. Стадийный порядок методов SDIRK ограничен порядком 1-й стадии и не может быть выше 1-го. В то же время стадийный порядок методов ESDIRK ограничен порядком 2-й стадии и может быть равен двум. Явная стадия методов ESDIRK не требует вычислений, поскольку она совпадает с последней стадией предыдущего шага. Поэтому вычислительные затраты методов ESDIRK и SDIRK с одинаковым числом неявных стадий примерно одинаковы. При том же числе неявных стадий методы ESDIRK могут иметь такую же функцию устойчивости, но они имеют больше параметров, что позволяет строить методы более высокого порядка, обладающие некоторыми дополнительными свойствами. Таким образом, методы ESDIRK более перспективны, поэтому основное внимание будем уделять методам ESDIRK 2-го стадийного порядка. Для удобства дальнейшего изложения обозначим число неявных стадий через r, тогда $r = s$ для методов SDIRK и $r = s - 1$ для методов ESDIRK.

Функция устойчивости метода Рунге–Кутты задается формулой

$$R(z) = 1 + z\mathbf{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}.$$

Жесткоточные методы SDIRK удовлетворяют условию $R(\infty) = 0$. Потребуем выполнения этого условия также и для методов ESDIRK. Строгое требование L-устойчивости часто является завышенным, поэтому будем рассматривать также и $L(\alpha)$-устойчивые методы при значении α, близком к 90°.

Согласно [17], метод называется $L\mu(\alpha)$-*устойчивым*, если он $A(\alpha)$-устойчив и $R(z) = O(z^{-\mu})$ при $z \to \infty$, где μ – порядок L-затухания функции устойчивости. На основании численных экспериментов мы убедились, что повышенный порядок L-затухания ($\mu > 1$) не дает преимущества при решении жестких задач. Действительно, любой $L(\alpha)$-устойчивый метод обеспечит порядок L-затухания μ, если принять μ шагов за один шаг. Однако $L2(\alpha)$-устойчивый метод позволил реализовать эффективный контроль ошибки при решении ДАУ ин-

декса 3. Поэтому наряду с $L(\alpha)$-устойчивыми методами рассмотрим также и $L2(\alpha)$-устойчивый метод.

В случае обратимой матрицы **A** функцию $R(z)$ можно представить в виде

$$R(z) = 1 - \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e} + \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}. \tag{5.1}$$

Для метода SDIRK примем $\bar{\mathbf{A}} = \mathbf{A} - \gamma\mathbf{I}$, т. е. $\bar{\mathbf{A}}$ – матрица, полученная из **A** путем замены всех диагональных элементов нулями. Тогда

$$(\mathbf{I} - z\mathbf{A})^{-1} = \frac{1}{(1-\gamma z)}\left[\mathbf{I} - \frac{z}{(1-\gamma z)}\bar{\mathbf{A}}\right]^{-1} = \sum_{i=0}^{r-1}\frac{z^i}{(1-\gamma z)^{i+1}}\bar{\mathbf{A}}^i. \tag{5.2}$$

Подставляя (5.2) в (5.1), получаем

$$R(z) = 1 - d_0 + \sum_{i=0}^{r-1} d_i \frac{z^i}{(1-\gamma z)^{i+1}}, \tag{5.3}$$

где $d_i = \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\bar{\mathbf{A}}^i\mathbf{e}$. Из условия жесткой точности имеем $\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1} = \mathbf{e}_r^{\mathrm{T}} = (0, \ldots, 0, 1)$, тогда $d_0 = 1$, $d_i = \mathbf{e}_r^{\mathrm{T}}\bar{\mathbf{A}}^i\mathbf{e}$.

Матрицу и векторы коэффициентов метода ESDIRK представим в виде

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{a} & \tilde{\mathbf{A}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \tilde{\mathbf{b}} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}} \end{bmatrix},$$

где из условия 1-го стадийного порядка $\mathbf{a} = \tilde{\mathbf{c}} - \tilde{\mathbf{A}}\mathbf{e}$, $b_1 = 1 - \tilde{\mathbf{b}}^{\mathrm{T}}\mathbf{e}$. Функция устойчивости такого метода $R(z) = 1 + z(1 - \tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}) + z\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}(\mathbf{I} - z\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{c}}$. Используя условие жесткой точности $\tilde{\mathbf{b}}^{\mathrm{T}} = \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}$, получаем $R(z) = 1 - \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}} + \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}(\mathbf{I} - z\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{c}}$. Обозначив $\bar{\mathbf{A}} = \tilde{\mathbf{A}} - \gamma\mathbf{I}$, получим функцию устойчивости метода ESDIRK в виде (5.3), где

$$d_i = \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\bar{\mathbf{A}}^i\tilde{\mathbf{c}}, \quad i = 0, \ldots, r-1. \tag{5.4}$$

Будем рассматривать методы, имеющие $d_0 = 1$, что является необходимым условием $L(\alpha)$-устойчивости. Для выполнения этого условия достаточно, чтобы метод SDIRK был жесткоточным, а для жесткоточного метода ESDIRK дополнительно должно выполняться равенство

$$d_0 = \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}} = 1. \tag{5.5}$$

При $d_0 = 1$ получаем

$$R(z) = \frac{1}{(1-\gamma z)} + \sum_{i=1}^{r-1} d_i \frac{z^i}{(1-\gamma z)^{i+1}}. \tag{5.6}$$

Коэффициенты d_i метода порядка p удовлетворяют условиям

$$d_i = D_i(\gamma) = \sum_{j=0}^{i}\frac{(-\gamma)^j}{(i-j)!}\binom{i}{j}, \quad i = 1, \ldots, p. \tag{5.7}$$

где $\binom{i}{j} = \frac{i!}{j!(i-j)!}$ – биномиальные коэффициенты.

Рассмотрим методы, имеющие $p \geq r - 1$, тогда при $R(\infty) = 0$ функция устойчивости зависит только от γ и выражается формулами (5.6), (5.7). Максимальный порядок таких методов SDIRK – 4-й, а ESDIRK – 5-й. Разложив выражение $e^z - R(z)$ в ряд Тейлора, получаем

$$e^z - R(z) = \frac{C_r}{r!} z^r + \frac{C_{r+1}}{(r+1)!} z^{r+1} + O(z^{r+2}),$$

где $C_r = r!D_r(\gamma)$. Коэффициенты C_r и C_{r+1} совпадают с одними из коэффициентов погрешности метода. При $p = r$ значение γ должно быть равно одному из корней уравнения

$$D_r(\gamma) = C_r/r! = 0, \tag{5.8}$$

а при $p = r - 1$ функция устойчивости однозначно определяется значением γ. Заметим, что при $\gamma = 0$, $r \leq 5$ получаем функцию устойчивости явного p-стадийного метода порядка $p = r - 1$, тогда $C_r = C_{r+1} = 1$.

При выборе подходящих значений γ исходим из того, что метод должен иметь достаточно большой сектор устойчивости (например, $\alpha > 80°$). Это требование задает ограничение величины γ снизу. Кроме этого, должна обеспечиваться достаточно высокая точность решения уравнения Далквиста (зададим ограничения $|C_r| < 1$, $|C_{r+1}| < 1$, т. е. точность должна быть заведомо выше, чем у явного метода порядка $r - 1$, полученного при $\gamma = 0$). Из условия $0 \leq c_i \leq 1$ получаем также $\gamma \leq 0.5$. Эти требования ограничивают величину γ сверху.

Исследуем зависимости угла α и коэффициентов C_r, C_{r+1} от γ при $r = 4, 5, 6$. Полученные графики приведены на рис. 5.1. В общем случае при выборе соответствующих коэффициентов такие методы имеют порядок $p = r - 1$. А если задать γ равным одному из r значений, удовлетворяющих условию $C_r = 0$, то метод может иметь порядок r. Наилучшая точность обеспечивается при наименьшем из этих значений, но такие методы не являются даже $L(0)$-устойчивыми. Таким образом, выбор γ сводится к компромиссу между точностью и устойчивостью.

В табл. 5.1 приведены характеристики функции устойчивости для некоторых значений γ, пригодных для построения $L(\alpha)$-устойчивых методов.

При $r = 4$ значение $\gamma = 0.125$ примерно соответствует локальному максимуму зависимости $\alpha(\gamma)$. Близкое к этому значение $\gamma = 0.1288\ldots$ (один из корней многочлена $1 - 12\gamma + 36\gamma^2 - 24\gamma^3$) обеспечивает 2-й порядок L-затухания. При $\gamma = 0.2164\ldots$ в разделе 5.4 построен метод, удовлетворяющий дополнительным условиям порядка для ДАУ индексов 2 и 3. Значение $\gamma = 0.2204\ldots$ (корень многочлена $D_4(\gamma)$) является наиболее подходящим для построения метода ESDIRK 4-го порядка, обеспечивая малое значение $C_5 = e(T_{58}) = e(T_{59}) = 0.135$. Альтернативное значение $\gamma = 0.5728\ldots$ совпадает с правой границей интервала L-устойчивости $0.2236\ldots \leq \gamma \leq 0.5728\ldots$ (см. [75]), но тогда $C_5 = -3.27$. Близкое к левой границе этого интервала значение $\gamma = 0.225$ использовалось в двух методах из [119].

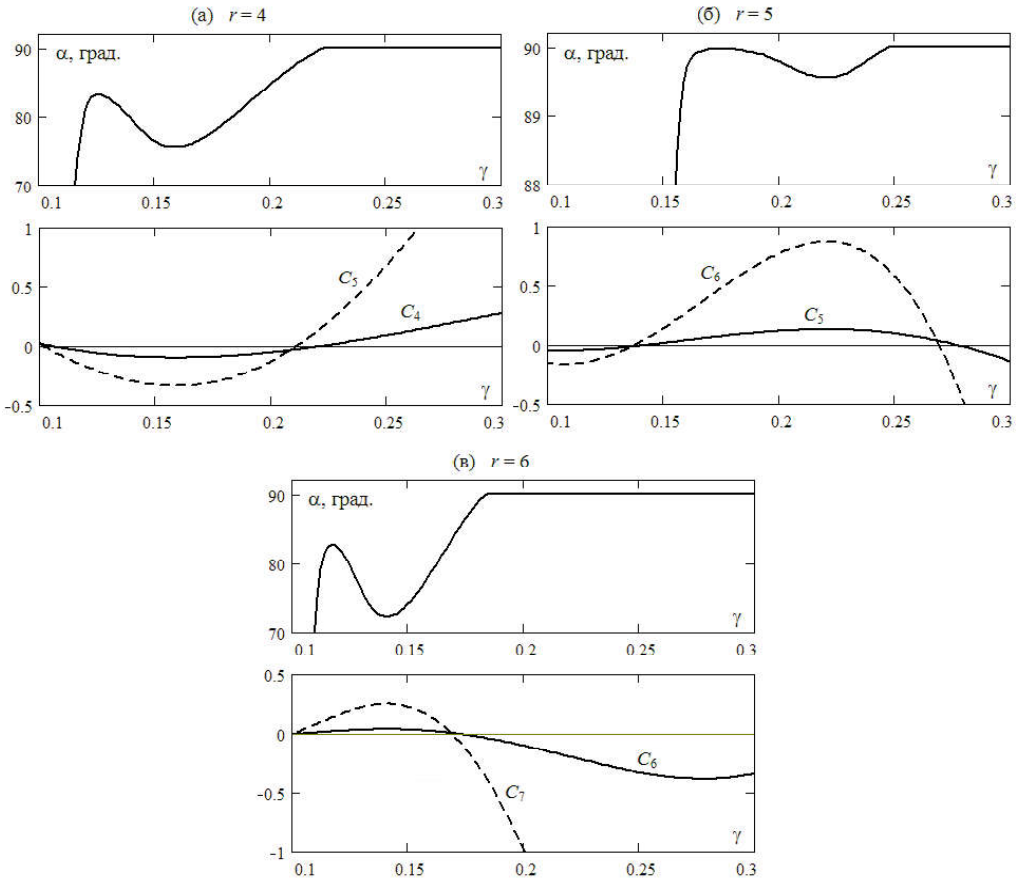


Рис. 5.1. Угол сектора устойчивости
и коэффициенты C_r, C_{r+1} методов DIRK при $p \geqslant r - 1$

При $r = 5$ в позициях 6 и 7 табл. 5.1 приведены значения γ, расположенные близко к локальному максимуму функции $\alpha(\gamma)$, при этом значение $\gamma = 0.1744\ldots$ является корнем многочлена $1 - 20\gamma + 120\gamma^2 - 240\gamma^3 + 120\gamma^4$. Значение $\gamma = 0.25$ близко к левой границе интервала L-устойчивости $0.2479\ldots \leq \gamma \leq 0.6760\ldots$ и применялось во многих методах 4-го порядка. В [75] рекомендовалось также значение $\gamma = 4/15 = 0.2666\ldots$, при котором C_5 и C_6 малы. Значение $\gamma = 0.2780\ldots$ (корень многочлена $D_5(\gamma)$) позволяет построить L-устойчивый метод ESDIRK 5-го порядка.

При $r = 6$ близкие к локальному максимуму функции $\alpha(\gamma)$ значения γ приведены в позициях 11 и 12 табл. 5.1, при этом значение $\gamma = 0.1190\ldots$ является корнем многочлена $1 - 30\gamma + 300\gamma^2 - 1200\gamma^3 + 1800\gamma^4 - 720\gamma^5$. Значение $\gamma = 0.1731\ldots$ (корень многочлена $D_6(\gamma)$) обеспечивает $C_6 = 0$ и малое значение C_7, но построить $L(\alpha)$-устойчивый жесткоточный метод DIRK 6-го порядка с таким значе-

нием γ невозможно [119]. Значения $\gamma = 0.184$ и $\gamma = 0.2$ принадлежат интервалу L-устойчивости $0.1839\ldots \le \gamma \le 0.3341\ldots$ и использовались в методах ESDIRK 5-го порядка, предложенных в [53, 119].

Таблица 5.1. Характеристики функций устойчивости при $p \geqslant r - 1$

№	r	γ	Устойчивость	C_r	C_{r+1}
1	4	0.125	$L(83.12°)$	−0.057	−0.206
2		0.1288...	$L2(82.90°)$	−0.069	−0.233
3		0.2164...	$L(88.81°)$	−0.011	0.076
4		0.2204...	$L(89.55°)$	0	0.135
5		0.225	$L(90°)$	0.013	0.207
6	5	1/6	$L(89.95°)$	0.059	0.367
7		0.1744...	$L2(89.97°)$	0.076	0.476
8		0.25	$L(90°)$	0.102	0.590
9		4/15	$L(90°)$	0.050	0.109
10		0.2780...	$L(90°)$	0	−0.382
11	6	0.1190...	$L2(82.51°)$	0.026	0.156
12		0.12	$L(82.26°)$	0.027	0.163
13		0.1731...	$L(85.57°)$	0	−0.079
14		0.184	$L(90°)$	−0.033	−0.370
15		0.2	$L(90°)$	−0.096	−0.965

5.2. Функции погрешности

Методы SDIRK имеют 1-й стадийный порядок, поэтому жесткая составляющая ошибки определяется прежде всего функцией $e_2(z)$, которую для жесткоточного метода порядка $p \geq 2$ можно представить в виде

$$e_2(z) = \mathbf{e}_r^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{c}^2 - 2\mathbf{A}\mathbf{c}) = \sum_{i=p-1}^{r-1} \mathbf{e}_r^{\mathrm{T}}\bar{\mathbf{A}}^i(\mathbf{c}^2 - 2\mathbf{A}\mathbf{c})\frac{z^i}{(1-\gamma z)^{i+1}}. \tag{5.9}$$

Рассмотрим жесткоточные методы SDIRK порядка $p = r - 1$. В этом случае, учитывая, что $c_1 = \gamma$, $a_{21} = c_2 - \gamma$, получим

$$\mathbf{e}_r^{\mathrm{T}}\bar{\mathbf{A}}^{r-2}(\mathbf{c}^2 - \mathbf{A}\mathbf{c}) = b_{r-1}a_{r-1,r-2}\ldots a_{32}a_{21}(c_2 - \gamma) = d_{r-1}(c_2 - \gamma),$$
$$\mathbf{e}_r^{\mathrm{T}}\bar{\mathbf{A}}^{r-2}\mathbf{A}\mathbf{c} = \mathbf{e}_r{}^{\mathrm{T}}\bar{\mathbf{A}}^{r-2}(\bar{\mathbf{A}} + \gamma\mathbf{I})^2\mathbf{e} = 2d_{r-1}\gamma + d_{r-2}\gamma^2,$$
$$\mathbf{e}_r^{\mathrm{T}}\bar{\mathbf{A}}^{r-1}(\mathbf{c}^2 - \mathbf{A}\mathbf{c}) = b_{r-1}a_{r-1,r-2}\ldots a_{32}a_{21}(-\gamma^2) = -d_{r-1}\gamma^2.$$

Подставляя эти выражения в (5.9), имеем:

$$e_2(z) = \left[d_{r-1}(c_2 - 3\gamma) - d_{r-2}\gamma^2\right]\frac{z^{r-2}}{(1-\gamma z)^{r-1}} - d_{r-1}\gamma^2\frac{z^{r-1}}{(1-\gamma z)^r}.$$

Учитывая (5.7) и предполагая, что $D_{r-1}(\gamma) \neq 0$, получаем:

$$e_2(z) = \frac{z^{r-2}}{(1-\gamma z)^r} D_{r-1}(\gamma)\left[c_2 - (c_2^* + \gamma) - \gamma z(c_2 - c_2^*)\right], \quad c_2^* = 2\gamma + \gamma^2 \frac{D_{r-2}(\gamma)}{D_{r-1}(\gamma)}. \tag{5.10}$$

Вычисленное согласно (5.10) значение c_2^* обеспечивает выполнение равенства $\mathbf{b}^{\mathrm{T}}\mathrm{A}^{-2}\mathbf{c}^2 = 2$, которое является одним из условий порядка для ДАУ индекса 2.

Для методов ESDIRK потребуем, чтобы они имели второй стадийный порядок, т. е. удовлетворяли условию

$$\tilde{\mathbf{c}}^2 = 2\tilde{\mathbf{A}}\tilde{\mathbf{c}}. \tag{5.11}$$

В этом случае $e_2(z) \equiv 0$ и жесткая составляющая ошибки определяется функцией $e_3(z) = e_{31}(z) = e_{32}(z)$, которую для жесткоточного метода порядка $p \geq 3$ можно представить в виде:

$$e_3(z) = \mathbf{e}_r^{\mathrm{T}}(\mathbf{I} - z\tilde{\mathbf{A}})^{-1}(\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = \sum_{i=p-2}^{r-1} \mathbf{e}_r^{\mathrm{T}}\bar{\mathbf{A}}^i(\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2)\frac{z^i}{(1-\gamma z)^{i+1}}. \tag{5.12}$$

Рассмотрим сначала случай, когда $p = r$. Действуя по аналогии с методами SDIRK, из (5.11) и (5.12) получаем:

$$e_3(z) = \left[2d_{r-1}\gamma(c_3 - 5\gamma) - 2d_{r-2}\gamma^3\right]\frac{z^{r-2}}{(1-\gamma z)^{r-1}} - 2d_{r-1}\gamma^3\frac{z^{r-1}}{(1-\gamma z)^r},$$

а при выполнении (5.8) получим

$$e_3(z) = \frac{z^{r-2}}{(1-\gamma z)^r} 2\gamma D_{r-1}(\gamma)\left[c_3 - (c_3^* + \gamma) - \gamma z(c_3 - c_3^*)\right], \quad c_3^* = 4\gamma + \gamma^2 \frac{D_{r-2}(\gamma)}{D_{r-1}(\gamma)}. \tag{5.13}$$

Значение c_3^* из (5.13) обеспечивает выполнение равенства $\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^3 = 3$, которое является одним из условий порядка для ДАУ индекса 2.

Найдем теперь функцию $e_3(z)$ при $p = r - 1$, потребовав дополнительно выполнения условия

$$\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^3 = \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3 = 3, \tag{5.14}$$

которое гарантирует затухание $e_3(z)$ при $z \to \infty$ как $O(z^{-2})$, а также является одним из условий порядка для ДАУ индекса 2. Опуская промежуточные преобразования, приводим окончательную формулу

$$e_3(z) = \left[2d_{r-1}(c_3 - 4\gamma) - 2d_{r-2}\gamma^2\right]\frac{z^{r-3}}{(1-\gamma z)^{r-1}} - 2d_{r-1}\gamma^2\frac{z^{r-2}}{(1-\gamma z)^r},$$

которая для $L(\alpha)$-устойчивого метода при $D_{r-1}(\gamma) \neq 0$ запишется в виде:

$$e_3(z) = \frac{z^{r-3}}{(1-\gamma z)^r} 2D_{r-1}(\gamma)\left[c_3 - (c_3^* + \gamma) - \gamma z(c_3 - c_3^*)\right], \quad c_3^* = 3\gamma + \gamma^2 \frac{D_{r-2}(\gamma)}{D_{r-1}(\gamma)}. \tag{5.15}$$

На этот раз значение c_3^* обеспечивает также и выполнение равенства $\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-3}\tilde{\mathbf{c}}^3 = 6$, которое является одним из условий порядка для ДАУ индекса 3.

Нами получены выражения (5.10), (5.13), (5.15), в которых функция погрешности зависит только от двух параметров: γ и c_2 (или c_3). Обозначим через $\varepsilon(z, c_2)$ модуль функции погрешности (5.10) при заданном положительном γ. Тогда справедливы следующие утверждения:

1) если $c_2 < c_2^*$ и $\mathrm{Re}\, z < 0$, то $\varepsilon(z, c_2) > \varepsilon(z, c_2^*)$;

2) если $c_2 > c_2^* + \gamma$ и $\mathrm{Re}\, z < 0$, то $\varepsilon(z, c_2) > \varepsilon(z, c_2^* + \gamma)$;

3) если $c_2^* \leqslant c_2 < c_2' \leqslant c_2^* + \gamma$, то существует такое z, что $\mathrm{Re}\, z < 0$, $\varepsilon(z, c_2) < \varepsilon(z, c_2')$, и такое z', что $\mathrm{Re} z' < 0$, $\varepsilon(z', c_2) > \varepsilon(z', c_2')$.

Таким образом, неравенство

$$c_2^* \leqslant c_2 \leqslant c_2^* + \gamma \tag{5.16}$$

задает множество всех значений c_2, изменяя которые, невозможно уменьшить модуль функции (5.10) сразу во всех точках левой полуплоскости. Оптимальное значение c_2 следует искать на интервале (5.16), при этом для очень жестких задач оно смещается к левой границе интервала, а для задач малой жесткости – к правой границе. Аналогичное неравенство для функций (5.13) и (5.15) запишется в виде:

$$c_3^* \leqslant c_3 \leqslant c_3^* + \gamma. \tag{5.17}$$

5.3. Условия порядка

Основная трудность при построении методов высоких порядков заключается в необходимости обеспечить выполнение большого числа алгебраических условий. Учет диагональной формы матрицы $\mathbf{A}$ позволяет упростить эти условия, в результате они становятся ненамного сложнее, чем для явных методов. В [75] были выведены упрощенные условия порядка для методов SDIRK, а в [56] аналогичные условия получены для жесткоточных методов ESDIRK 2-го стадийного порядка, которые мы и рассмотрим.

Из условия 1-го стадийного порядка имеем:

$$a_{i1} = c_i - \sum_{j=2}^{i-1} a_{ij} - \gamma, \quad i = 2, \ldots, s. \tag{5.18}$$

Определяемые согласно (5.18) коэффициенты не входят во все остальные условия порядка, поэтому их вычисляем в последнюю очередь. Из условия 2-го стадийного порядка (5.11) получаем:

$$c_2 = 2\gamma, \quad a_{i2} = \frac{1}{4\gamma}\left[c_i^2 - 2\left(\sum_{j=3}^{i-1} a_{ij} c_j + \gamma c_i\right)\right], \quad i = 3, \ldots, s. \tag{5.19}$$

Обозначим $\bar{\mathbf{b}} = \tilde{\mathbf{b}} - \gamma \mathbf{e}_r = (\mathbf{e}_r^{\mathrm{T}} \bar{\mathbf{A}})^{\mathrm{T}}$, тогда квадратурные условия, обеспечивающие порядок p при решении уравнения $y' = f(t)$, запишутся в виде:

$$\bar{\mathbf{b}}^{\mathrm{T}} \tilde{\mathbf{c}}^{i-1} = \sum_{j=2}^{s-1} b_j c_j^{i-1} = \frac{1}{i} - \gamma, \quad i = 2, \ldots, p. \tag{5.20}$$

Одно из условий 3-го порядка входит в (5.20), а второе запишется как

$$\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}\tilde{\mathbf{c}} = (\bar{\mathbf{b}} + \gamma \mathbf{e}_r)^{\mathrm{T}}(\bar{\mathbf{A}} + \gamma \mathbf{I})\tilde{\mathbf{c}} = \frac{1}{6},$$

откуда, учитывая (5.20), получим

$$\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}\tilde{\mathbf{c}} = \frac{1}{6} - \gamma + \gamma^2. \tag{5.21}$$

При построении метода 3-го порядка достаточно обеспечить выполнение условий (5.18)–(5.20), откуда следует также и (5.21), но соотношение (5.21) понадобится нам при выводе условий более высоких порядков.

Метод 4-го порядка должен дополнительно удовлетворять четырем условиям. Учет второго стадийного порядка сокращает число дополнительных условий до двух, одно из которых входит в (5.20), а второе запишется в виде

$$\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^2\tilde{\mathbf{c}} = (\bar{\mathbf{b}} + \gamma \mathbf{e}_r)^{\mathrm{T}}(\bar{\mathbf{A}} + \gamma \mathbf{I})^2\tilde{\mathbf{c}} = \bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^2\tilde{\mathbf{c}} + 3\gamma\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}\tilde{\mathbf{c}} + 3\gamma^2\bar{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{c}} + \gamma^3 = \frac{1}{24},$$

откуда, используя (5.20), (5.21), получим:

$$\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^2\tilde{\mathbf{c}} = \frac{1}{24} - \frac{1}{2}\gamma + \frac{3}{2}\gamma^2 - \gamma^3. \tag{5.22}$$

При увеличении порядка число дополнительных условий быстро возрастает и для метода 5-го порядка равно девяти, но при выполнении условия 2-го стадийного порядка сокращается до четырех. Одно из них входит в (5.20), а три остальных, действуя аналогично, получим в виде:

$$\begin{aligned}
&\bar{\mathbf{b}}^{\mathrm{T}}\left(\tilde{\mathbf{c}}(\bar{\mathbf{A}}^2\tilde{\mathbf{c}})\right) = \frac{1}{30} - \frac{5}{12}\gamma + \frac{4}{3}\gamma^2 - \gamma^3,\\
&\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}\left(\tilde{\mathbf{c}}(\bar{\mathbf{A}}\tilde{\mathbf{c}})\right) = \frac{1}{40} - \frac{1}{3}\gamma + \frac{7}{6}\gamma^2 - \gamma^3,\\
&\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^3\tilde{\mathbf{c}} = \frac{1}{120} - \frac{1}{6}\gamma + \gamma^2 - 2\gamma^3 + \gamma^4.
\end{aligned} \tag{5.23}$$

Кроме условий порядка, потребуем выполнения равенства (5.5), обеспечивающего $L(\alpha)$-устойчивость. При $p = r$ для этого достаточно, чтобы γ было корнем уравнения (5.8), а при $p = r - 1$ запишем (5.5) в виде:

$$\mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}} = \frac{1}{\gamma}\mathbf{e}_r^{\mathrm{T}}\left(\mathbf{I} + \frac{1}{\gamma}\bar{\mathbf{A}}\right)^{-1}\tilde{\mathbf{c}} = \frac{1}{\gamma} + \sum_{i=0}^{r-2}\frac{(-1)^{i+1}}{\gamma^{i+2}}\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^i\tilde{\mathbf{c}} = 1. \tag{5.24}$$

Подставив в (5.24) значения $\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^i\tilde{\mathbf{c}}$, $i = 0,\ldots, r-1$, входящие в условия порядка (5.20)–(5.23), при $r = 5$, $p = 4$ получим

$$\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^3\tilde{\mathbf{c}} = b_5 a_{54} a_{43} a_{32} c_2 = \gamma\left(\frac{1}{24} - \frac{2}{3}\gamma + 3\gamma^2 - 4\gamma^3 + \gamma^4\right), \tag{5.25}$$

а при $r = 6$, $p = 5$ получим

$$\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^{4}\tilde{\mathbf{c}} = b_6 a_{65} a_{54} a_{43} a_{32} c_2 = \gamma\left(\frac{1}{120} - \frac{5}{24}\gamma + \frac{5}{3}\gamma^2 - 5\gamma^3 + 5\gamma^4 - \gamma^5\right). \tag{5.26}$$

Можно обобщить формулы (5.25), (5.26). Действительно, из (5.4), (5.7) при $i = p = r - 1$ получим

$$d_p = \mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\bar{\mathbf{A}}^{p}\tilde{\mathbf{c}} = \frac{1}{\gamma}\mathbf{e}_r^{\mathrm{T}}\left(\mathbf{I} + \frac{1}{\gamma}\bar{\mathbf{A}}\right)^{-1}\bar{\mathbf{A}}^{p}\tilde{\mathbf{c}} = \frac{1}{\gamma}\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^{p-1}\tilde{\mathbf{c}} = D_p(\gamma),$$

откуда $\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^{p-1}\tilde{\mathbf{c}} = \gamma D_p(\gamma)$.

При $p = r - 1$ потребуем также выполнения условия (5.14), которое можно записать в виде:

$$\mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3 = \frac{1}{\gamma} + \sum_{i=0}^{r-2}\frac{(-1)^{i+1}}{\gamma^{i+2}}\bar{\mathbf{b}}^{\mathrm{T}}\bar{\mathbf{A}}^{i}\tilde{\mathbf{c}}^3 = 3. \tag{5.27}$$

В разделах 4.9, 4.10 были рассмотрены методы SDIRK, обладающие повышенной точностью при решении ДАУ индексов 2 и 3. Обсудим теперь точность решения ДАУ методами ESDIRK. Системы ДАУ индекса 1 имеют вид

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{0} = \mathbf{g}(\mathbf{y}, \mathbf{z}),$$

где матрица $\mathbf{g}_{\mathbf{z}} = \partial\mathbf{g}(\mathbf{y}, \mathbf{z})/\partial\mathbf{z}$ обратима в окрестности решения. Жесткоточные методы, к которым относятся ESDIRK, обеспечивают точное выполнение алгебраического соотношения, поэтому порядки сходимости дифференциальных и алгебраических компонент совпадают с порядком метода: $p_y = p_z = p$.

Систему ДАУ индекса 2 можно привести к виду

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{0} = \mathbf{g}(\mathbf{y}),$$

где матрица $\mathbf{g}_{\mathbf{y}}\mathbf{f}_{\mathbf{z}}$ обратима в окрестности решения. Как следствие теоремы 5.2 из [116], порядки сходимости соответствующих компонент при решении таких задач методами ESDIRK (при $p \geq 3$, $q = 2$) следующие: $p_y = \min(p, q + 1) = 3$, $p_y = q = 2$.

Систему ДАУ индекса 3 можно представить в виде

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}), \quad \mathbf{z}' = \mathbf{k}(\mathbf{y}, \mathbf{z}, \mathbf{u}), \quad \mathbf{0} = \mathbf{g}(\mathbf{y}),$$

где матрица $\mathbf{g}_{\mathbf{y}}\mathbf{f}_{\mathbf{z}}\mathbf{k}_{\mathbf{u}}$ обратима в окрестности решения. В [117] получены порядки сходимости компонент $\mathbf{y}, \mathbf{z}, \mathbf{u}$ при решении таких задач методами с обратимой матрицей $\mathbf{A}$. Для методов с явной 1-й стадией аналогичных результатов мы не нашли, но численные эксперименты с методами ESDIRK давали оценки порядков $\tilde{p}_y = \tilde{p}_z = 2$, $\tilde{p}_u = 1$.

В [60], [61] были получены дополнительные условия, необходимые для повышения порядков сходимости различных компонент ДАУ. Для методов ESDIRK эти условия имеют вид

$$\mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3 = 3; \tag{5.28а}$$
$$\mathbf{e}_r^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^3 = 6; \tag{5.28б}$$
$$\tilde{\mathbf{b}}^{\mathrm{T}}(\tilde{\mathbf{c}}\cdot(\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^3)) = 2; \tag{5.28в}$$
$$\tilde{\mathbf{b}}^{\mathrm{T}}(\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^3)^2 = 12; \tag{5.28г}$$
$$\tilde{\mathbf{b}}^{\mathrm{T}}(\tilde{\mathbf{c}}\cdot(\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3)) = 3/4. \tag{5.28д}$$

Для получения этих условий были рассмотрены модельные уравнения, соответствующие деревьям 4-го порядка для переменных индекса 1, 3-го порядка для переменных индекса 2, 2-го порядка для переменных индекса 3

Ряд деревьев мы исключили из рассмотрения, поскольку для методов 2-го стадийного порядка некоторые условия порядка дублируются. Система модельных уравнений, которая была рассмотрена:

$$x_1' = 1, \quad x_{31}' = x_1^2, \quad 0 = y_{31} - x_1^3, \quad 0 = y_{41} - x_1^4, \quad 0 = y_{42} - x_1 x_{31},$$
$$y_{31}' = z_{21}, \quad y_{41}' = z_{31}, \quad y_{42}' = z_{32}, \quad 0 = z_{33} - x_1 z_{21}, \quad y_{43}' = z_{33},$$
$$z_{21}' = u_{11}, \quad z_{31}' = u_{21}, \quad z_{32}' = u_{22}, \quad z_{33}' = u_{23},$$
$$0 = u_{24} - x_1 u_{11}, \quad 0 = u_{25} - u_{11}^2, \quad z_{34}' = u_{24}, \quad z_{35}' = u_{25}.$$

Эти уравнения образуют систему индекса 3, а исключив уравнения, содержащие переменные u_{ij}, получим систему индекса 2. Соответствующие этим уравнениям деревья показаны на рис. 5.2. Анализ численных решений позволил вывести аналитические выражения для глобальных ошибок, на основе которых были получены условия (5.28).

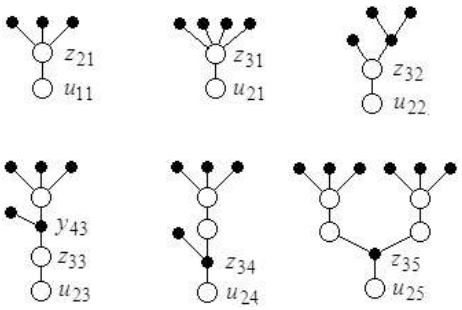


Рис. 5.2. Деревья для ДАУ индексов 2 и 3

Поскольку мы рассматриваем жесткоточные методы, переменные y_{31}, y_{41}, y_{42} вычисляются без ошибок. Для точного вычисления переменных z_{21} и u_{11} примем (5.28а, б), что обеспечивает также и точное вычисление переменных z_{33}, u_{24}, u_{25}. В этом случае при $R(\infty) = 0$ глобальные ошибки переменных z_{31}, z_{32} равны локальным ошибкам и пропорциональны h^3, а глобальные ошибки перемен-

ных u_{21}, u_{22}, u_{23} пропорциональны h^2. При принятых условиях ошибки переменных y_{43}, z_{34}, z_{35} накапливаются. Это не дает обеспечить 4-й порядок дифференциальных переменных для ДАУ индекса 2 (переменная y_{43}) и 3-й порядок для ДАУ индекса 3 (переменные z_{34}, z_{35}). Чтобы эти переменные вычислялись точно, должны выполняться условия (5.28в, г, д).

В табл. 5.2 приведены необходимые условия сходимости с заданным порядком для компонент ДАУ, где необходимое условие для каждой компоненты отмечено знаком +. Эти условия, вместе с соответствующими классическими условиями порядка, являются необходимыми для рассмотренных модельных уравнений, а значит, и для уравнений более общего вида. На ряде тестовых задач мы убедились, что выполнение этих условий действительно обеспечивает указанные порядки, но у нас нет доказательства, что эти условия являются также и достаточными.

Таблица 5.2. Необходимые условия для порядков сходимости компонент ДАУ

Условие	ДАУ индекса 2		ДАУ индекса 3		
	$p_y = 4$	$p_z = 3$	$p_y = 3$	$p_z = 3$	$p_u = 2$
(5.28а)	+	+	+	+	+
(5.28б)	–	–	+	+	+
(5.28в)	–	–	+	+	–
(5.28г)	–	–	+	+	–
(5.28д)	+	–	–	–	–

5.4. Методы 3-го порядка

Жесткоточные методы SDIRK при $p = r = 3$ рассматривались в [56, 81] и задаются таблицей

$$\begin{array}{c|ccc} \gamma & \gamma & & \\ \frac{1+\gamma}{2} & \frac{1-\gamma}{2} & \gamma & \\ b_i & 1-b_2-\gamma & b_2 & \gamma \\ \hline \hat{b}_i & 1-\hat{b}_2 & \hat{b}_2 & 0 \end{array}, \quad b_2 = \frac{5}{4} - 5\gamma + \frac{3}{2}\gamma^2, \quad \hat{b}_2 = \frac{1-2\gamma}{1-\gamma},$$

где γ – корень уравнения

$$D_3(\gamma) = \frac{1}{6} - \frac{3}{2}\gamma + 3\gamma^2 - \gamma^3 = 0.$$

Метод со значением $\gamma = \gamma_1 = 0.43586652150846$ является L-устойчивым и имеет коэффициенты погрешности $e(T_{4i}) = (-0.190, 0.069, 0.207, 0.622)$, а метод со значением $\gamma = \gamma_2 = 0.15898389998968$ является $L(75.6°)$-устойчивым и имеет коэффициенты погрешности $e(T_{4i}) = (-0.0253, 0.0105, 0.0314, 0.0934)$.

Аналогичные методы ESDIRK второго стадийного порядка при $p = r = 3$ рассматривались в [53, 56, 82, 118, 124, 150] и задаются таблицей

$$\begin{array}{c|ccccc} 0 & 0 & & & & \\ 2\gamma & \gamma & \gamma & & & \\ c_3 & c_3 - a_{32} - \gamma & a_{32} & \gamma & & \\ b_i & 1 - b_2 - b_3 - \gamma & b_2 & b_3 & \gamma & \\ \hline \hat{b}_i & 1 - \hat{b}_2 - \hat{b}_3 - \gamma & \hat{b}_2 & \hat{b}_3 & 0 & \end{array},$$

где из (5.19), (5.20) имеем

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma}, \quad b_2 = \frac{2 - 6\gamma - 3c_3 + 6\gamma c_3}{12\gamma(2\gamma - c_3)}, \quad b_3 = \frac{1 - 6\gamma + 6\gamma^2}{3c_3(c_3 - 2\gamma)}.$$

Коэффициенты вложенной формулы находим из условия 2-го порядка и условия ограниченности $\hat{R}(\infty)$. Последнее условие можно записать в виде $(\hat{b}_2, \hat{b}_3, 0)\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}} = 1$. В результате получаем

$$\hat{b}_3 = \frac{\gamma(1 - 2\gamma)}{c_3(c_3 - 2\gamma)}, \quad \hat{b}_2 = \frac{1 - 2\hat{b}_3 c_3}{4\gamma}.$$

Пусть $R(\infty) = 0$ (это необходимое условие $L(\alpha)$-устойчивости). Тогда диагональный элемент должен удовлетворять уравнению $D_3(\gamma) = 0$, а c_3 является свободным параметром. Условие равенства нулю коэффициентов погрешности $e(T_{41})$ и $e(T_{42})$ приводит к уравнению $1 - 4(b_2 c_2^3 + b_3 c_3^3 + \gamma) = 0$, откуда

$$c_3 = 1/2 + \gamma/4 \tag{5.29}$$

(тогда $c_3 = 0.609$ при $\gamma = \gamma_1$ и $c_3 = 0.5397$ при $\gamma = \gamma_2$). Такой метод при $\gamma = \gamma_1$ рассматривался в [82]. При $\gamma = \gamma_2$ близкое к (5.29) значение $c_3 = (2 + \sqrt{2})\gamma = 0.5428$ принято в [53]. В этом случае, кроме малых значений $e(T_{41})$ и $e(T_{42})$, обеспечиваются также L-устойчивость 3-й стадии и удобная реализация, поскольку на всех стадиях $a_{i1} = a_{i2}$, а также $\hat{b}_1 = \hat{b}_2$.

Другой подход может заключаться в выборе c_3 из условия минимизации функции погрешности $e_3(z)$. Тогда c_3 следует задавать на интервале $c_3^* \leqslant c_3 \leqslant c_3^* + \gamma$, где $c_3^* = \dfrac{(1 - \gamma)(1 - 4\gamma)}{1 - 4\gamma + 2\gamma^2}$. Для некоторых задач такой выбор дает небольшое преимущество, однако оно не столь ощутимо, как преимущество методов ESDIRK по сравнению с SDIRK.

Методы SDIRK и ESDIRK со значением $\gamma = \gamma_2$ более точны, но они обладают только $L(75.6°)$-устойчивостью, что может привести к заметному снижению их эффективности при решении задач с жестким спектром, содержащим собственные значения вблизи мнимой оси. К таким задачам принадлежит, например, тест BEAM.

Рассмотрим теперь построение методов ESDIRK 3-го порядка, обладающих повышенной точностью при решении ДАУ индексов 2 и 3. Такие методы

должны удовлетворять дополнительным условиям (5.28а–г), тогда число стадий должно быть не менее пяти. При $s = 5$ число алгебраических условий совпадает с числом свободных коэффициентов. Мы нашли 12 методов, которые удовлетворяют этим условиям. Они разбиваются на 3 группы, в зависимости от значений b_2 и b_3: 1) $b_2 = 0, b_3 \neq 0$ – 6 методов; 2) $b_2 = b_3 = 0$ – 3 метода; 3) $b_2 \neq 0$, $b_3 \neq 0$ – 3 метода.

При выборе подходящего метода мы потребовали, чтобы он был L-устойчивым либо $L(\alpha)$-устойчивым при α, близком к 90°, и все c_i не выходили за пределы интервала [0, 1]. Из найденных методов только один удовлетворяет этим требованиям. Этот метод является $L(88.81°)$-устойчивым и принадлежит 2-й группе, в которой γ – один из трех вещественных корней многочлена $2 - 36\gamma + 201\gamma^2 - 432\gamma^3 + 360\gamma^4 - 72\gamma^5$. Его коэффициенты находим по формулам

$$\gamma = 0.21646827973787, \quad c_2 = 2\gamma, \quad c_3 = \frac{(1-6\gamma+2\gamma^2)(3\gamma-6\gamma^2)}{1-9\gamma+18\gamma^2-6\gamma^3},$$
$$c_4 = \frac{2(1-3\gamma)}{3(1-2\gamma)}, \quad a_{43} = \frac{\gamma(1-9\gamma+18\gamma^2-6\lambda^3)}{6b_4a_{32}c_2}, \quad b_2 = b_3 = 0, \quad b_4 = \frac{3(1-2\gamma)^2}{4(1-3\gamma)} \tag{5.30}$$

(коэффициенты a_{i1}, a_{i2} находим по формулам (5.18), (5.19)). Коэффициенты погрешности этого метода: $e(T_{4i}) = (-0.059, -0.059, -0.011 -0.011)$.

При решении ДАУ индексов 2 и 3 усложняется контроль ошибки для компонент высших индексов. В оценках ошибки этих компонент появляются составляющие, пропорциональные отрицательным степеням h [109, глава 8]. В результате размер шага может уменьшаться до нуля, что приводит к аварийной остановке численного решения. Чтобы этого не происходило, в [109] предлагалось специальным образом масштабировать либо игнорировать оценки ошибки для переменных высших индексов. Но это не совсем удобно, поскольку индекс переменной может быть неизвестен, к тому же будет ослаблен контроль ошибки некоторых переменных. Опишем другой способ решения этой проблемы, позволяющий использовать обычный контроль ошибки для всех переменных.

В [155] показано, что если основной и вложенный методы имеют 2-й порядок L-затухания, т. е. если

$$R(\infty) = \hat{R}(\infty) = 0, \quad \lim_{z\to\infty} zR(z) = \lim_{z\to\infty} z\hat{R}(z) = 0,$$

то оценки ошибок переменных индексов 2 и 3 пропорциональны только локальным ошибкам этих переменных. В этом случае можно применять обычный контроль ошибки для всех переменных, не опасаясь аварийной остановки. Вложенная пара такого типа была построена в [155] на основе шестистадийного метода ESDIRK 3-го порядка, в который добавлена еще одна стадия специально для оценивания ошибки. Эта же стадия используется как прогноз $\mathbf{Y}_7^0 = \mathbf{Y}_6$ для заключительной стадии. Полученный метод имеет таблицу Бутчера:

$$
\begin{array}{c|ccccccc}
0 & 0 & & & & & & \\
1/3 & 1/6 & 1/6 & & & & & \\
2/3 & 1/6 & 1/3 & 1/6 & & & & \\
1 & 1/3 & 0 & 1/2 & 1/6 & & & \\
1 & 7/16 & 0 & 3/16 & 5/24 & 1/6 & & \\
1 & 7/48 & 17/48 & 17/48 & 1/80 & -1/30 & 1/6 & \\
\hline
b_i & 1/8 & 3/8 & 3/8 & 1/360 & -2/45 & 0 & 1/6 \\
\hat{b}_i & 7/48 & 17/48 & 17/48 & 1/80 & -1/30 & 1/6 & 0
\end{array}
\qquad (5.31)
$$

Здесь основной шестистадийный метод (стадии 1–5, 7) и вложенная формула (стадия 6) образуют семистадийный метод с контролем ошибки. Коэффициенты погрешности этого метода: $e(T_{4i}) = (0, 0, -0.0185\ -0.0185)$.

5.5. Методы 4-го порядка

Жесткоточных методов SDIRK при $p = r = 4$ не существует [81]. В [75] рассматривались пятистадийные методы SDIRK 4-го порядка. При заданном γ они образуют двухпараметрическое семейство со свободными параметрами c_2 и c_3. В результате минимизации коэффициентов погрешности в [75] был построен метод SDIRK4, имеющий $\gamma = 0.25$, $c_2 = 0.75$, $c_3 = 0.55$. Альтернативный подход к выбору параметров, основанный на минимизации функции погрешности $e_2(z)$, использован в [56]. Согласно (5.10), (5.16), оптимальное значение c_2 при $\gamma = 0.25$ принадлежит интервалу $[-1/12, 1/6]$. Приняв $c_2 = 0$, $c_3 = 0.5$, получим таблицу коэффициентов

$$
\begin{array}{c|ccccc}
1/4 & 1/4 & & & & \\
0 & -1/4 & 1/4 & & & \\
1/2 & 1/8 & 1/8 & 1/4 & & \\
1 & -3/2 & 3/4 & 3/2 & 1/4 & \\
\hline
b_i & 0 & 1/6 & 2/3 & -1/12 & 1/4 \\
\hat{b}_i & -2/3 & 5/12 & 7/6 & 1/12 & 0
\end{array}
$$

При решении тестовых задач этот метод оказался более точным, по сравнению с SDIRK4 из [75].

Более эффективными являются методы ESDIRK (результаты сравнения методов SDIRK и ESDIRK приведены в [53, 56]). Можно построить $L(\alpha)$-устойчивый метод ESDIRK при $p = r = 4$, если задать диагональный элемент равным корню уравнения

$$D_4(\gamma) = \frac{1}{24} - \frac{2}{3}\gamma + 3\gamma^2 - 4\gamma^3 + \gamma^4 = 0,$$

отличному от 0.106439. Такие методы рассматривались в [53, 56, 124]. Если абсциссы c_2, c_3, c_4 все различны, то при заданном γ эти методы образуют двухпараметрическое семейство, определяемое параметрами c_3, c_4. Веса b_2, b_3, b_4 находим из уравнений (5.20). Из (5.22) получаем

$$a_{43} = \frac{1-12\gamma+36\gamma^2-24\gamma^3}{12b_4c_3(c_3-2\gamma)},$$

а остальные коэффициенты находим из (5.19), (5.18).

Вложенную формулу находим из условий 3-го порядка и ограниченности $\hat{R}(\infty)$. Принимаем $\hat{b}_5 = 0$, тогда получаем:

$$\hat{b}_4 = \frac{6c_2c_3(c_2-c_3)+2c_2-3c_2^2-4c_3+6c_3^2}{6(c_3-c_4)(2c_3c_4+c_2(c_2-c_3-c_4))},$$

$$\hat{b}_3 = \frac{2-3c_2-6\hat{b}_4c_4(c_4-c_2)}{6c_3(c_3-c_2)}, \quad \hat{b}_2 = \frac{1-2(\hat{b}_3c_3+\hat{b}_4c_4)}{2c_2}.$$

Наиболее удобен для реализации метод, предложенный в [53], который оказался также и самым эффективным среди методов DIRK. Мы выбрали $\gamma = 0.220428410259$, $c_3 = (2+\sqrt{2})\gamma$, $c_4 = 0.610097451414$. Эти значения обеспечивают L-устойчивость внутренних стадий (кроме второй), а также выполнение неравенства (5.17), поскольку $c_3^* = 0.701$. Метод является $L(\alpha)$-устойчивым при $\alpha = 89.55°$ и имеет коэффициенты

$$\begin{aligned}
&\gamma = a_{21} = a_{22} = 0.22042841025921, \quad c_2 = 2\gamma, \quad c_3 = 0.75258966783935,\\
&c_4 = 0.61009745141424, \quad a_{31} = a_{32} = 0.26608062879007,\\
&a_{41} = a_{42} = 0.22703104746508, \quad a_{43} = -0.06439305377513,\\
&b_1 = b_2 = 0.17557544188348, \quad b_3 = -0.41553443172057,\\
&b_4 = 0.84395513769440, \quad \hat{b}_1 = \hat{b}_2 = 0.21711358669749,\\
&\hat{b}_3 = 0.41481167441242, \quad \hat{b}_4 = 0.15096115219260, \quad \hat{b}_5 = 0.
\end{aligned} \tag{5.32}$$

Метод с такими коэффициентами реализован в решателе DIRK4 ПО SimInTech.

Рассмотрим теперь методы ESDIRK 4-го порядка с пятью неявными стадиями, удовлетворяющие условиям (5.25), (5.27). Принимая γ, c_3, c_4, c_5, b_5 в качестве свободных параметров, из уравнений (5.20) находим b_2, b_3, b_4. Из (5.22), (5.25), (5.27), учитывая (5.19), получаем:

$$\begin{aligned}
a_{43} &= \frac{(1-16\gamma+72\gamma^2-96\gamma^3+24\gamma^4)c_4(c_4-c_3)(c_4-2\gamma)}{c_3(c_3-2\gamma)\left[3-32\gamma+84\gamma^2-48\gamma^3-c_3(4-36\gamma+72\gamma^2-24\gamma^3)\right]},\\
a_{54} &= \gamma\frac{1-16\gamma+72\gamma^2-96\gamma^3+24\gamma^4}{12b_5a_{43}c_3(c_3-2\gamma)},\\
a_{53} &= \frac{1-12\gamma+36\gamma^2-24\gamma^3-12b_4a_{43}c_3(c_3-2\gamma)-12b_5a_{54}c_4(c_4-2\gamma)}{12b_5c_3(c_3-2\gamma)}.
\end{aligned} \tag{5.33}$$

Остальные коэффициенты находим из (5.19), (5.18).

Остановимся на выборе γ и c_3. В [53, 75, 92 118] принимали $\gamma = 1/4$, что обеспечивает L-устойчивость и малую константу погрешности. Допустив также и $L(\alpha)$-устойчивые методы, отметим, что при $0.164 \le \gamma \le 0.191$ константа погрешности мала, а угол α сектора устойчивости больше 89.9° (см. рис. 5.16). Значение $\gamma = 1/6$ – наиболее удобное из этого интервала. Оптимальные значения c_3 задаются формулами (5.15), (5.17), откуда при $\gamma = 1/4$ получим $1/6 \le c_3 \le 5/12$, а при $\gamma = 1/6$ получим $8/15 \le c_3 \le 7/10$.

Коэффициенты вложенной формулы рассчитываем из условий 3-го порядка и ограниченности $\hat{R}(\infty)$, откуда при $\hat{b}_6 = 0$ получаем:

$$\hat{b}_4 = \frac{\left[3c_3(a_{54}a_{53} - \gamma a_{53})(c_3 - c_2) - 3\gamma a_{54}c_4(c_4 - c_2)\right]\hat{b}_5 + 6\gamma^4 - 6\gamma^3 + \gamma^2}{3\gamma a_{43}c_3(c_3 - c_2)},$$

$$\hat{b}_3 = \frac{6\hat{b}_4c_4(c_2 - c_4) + 6\hat{b}_5c_5(c_2 - c_5) - 3c_2 + 2}{6c_3(c_3 - c_2)}, \quad \hat{b}_2 = \frac{1 - 2(\hat{b}_3c_3 + \hat{b}_4c_4 + \hat{b}_5c_5)}{2c_2}.$$

Коэффициент $\hat{b}_5$ остается свободным, его выбираем из условий минимизации коэффициентов погрешности и значения $\hat{R}(\infty)$.

При построении методов мы старались минимизировать его коэффициенты, а также коэффициенты и функции погрешности $e(T_{4i})$ и $e_3(z)$. В результате при $\gamma = 1/4$ получен метод

0	0					
1/2	1/4	1/4				
1/4	1/16	−1/16	1/4			
3/4	1/16	−1/16	1/2	1/4		
1	−1/8	−5/8	9/8	3/8	1/4	
b_i	1/12	1/6	1/3	1/3	−1/6	1/4
$\hat{b}_i$	1/3	2/3	−1/3	1/3	0	0

, (5.34)

а при $\gamma = 1/6$ – метод

0	0					
1/3	1/6	1/6				
2/3	1/6	1/3	1/6			
1	11/24	−1/4	5/8	1/6		
1	11/36	−1/6	11/12	−2/9	1/6	
b_i	1/8	3/8	3/8	−1/12	1/24	1/6
$\hat{b}_i$	341/1800	109/600	341/600	47/900	1/120	0

. (5.35)

Оценим возможности методов, задаваемых формулами (5.18)–(5.20), (5.33), при решении ДАУ индексов 2 и 3. Как следствие теоремы 5.2 из [116] при реше-

нии ДАУ индекса 2 эти методы обеспечивают порядки сходимости $p_y = 3$ и $p_z = 2$ соответствующих компонент. Для ДАУ индекса 3 аналогичных результатов мы не нашли, но численные эксперименты давали оценки порядков $\tilde{p}_y = \tilde{p}_z = 2$ и $\tilde{p}_u = 1$. Попробуем улучшить эти показатели. Поставим задачу построить метод ESDIRK, обеспечивающий: а) 4-й порядок дифференциальных (y-компонента) и 3-й порядок алгебраических (z-компонента) переменных при решении ДАУ индекса 2; б) 3-й порядок дифференциальных (y- и z-компоненты) и 2-й порядок алгебраических (u-компонента) переменных при решении ДАУ индекса 3. Такой метод должен удовлетворять условиям (5.28).

Для методов, задаваемых формулами (5.18)–(5.20), (5.33), выполняется условие (5.28а). При заданном γ выполнение остальных четырех условий (5.28б–д) можно обеспечить путем подбора параметров c_3, c_4, c_5, b_5. При $\gamma = 1/4$ был найден один метод, удовлетворяющий этим условиям, который имеет $\gamma = 1/4$, $c_3 = 1/6$, $c_4 = 37/40$, $c_5 = 1/2$, $b_5 = 843750/1140071$ (все его коэффициенты приведены в [61, формула (4.3)]). При $\gamma = 1/6$ получено однопараметрическое семейство таких методов с коэффициентами

$$\gamma = b_1 = 1/6, \quad c_2 = 1/3, \quad c_3 = 8/15, \quad c_5 = 1/2, \quad b_2 = b_3 = b_4 = 0, \quad b_5 = 2/3,$$

$$a_{32} = \frac{4}{25}, \quad a_{42} = \frac{c_4}{132}(1-3c_4)(375c_4-266), \quad a_{43} = \frac{125c_4}{1056}(3c_4-1)(15c_4-8),$$

$$a_{52} = \frac{72c_4-35}{72(3c_4-1)}, \quad a_{53} = \frac{125(7-9c_4)}{576(15c_4-8)}, \quad a_{54} = \frac{11}{72c_4(1-3c_4)(15c_4-8)},$$

$$a_{i1} = c_i - \sum_{j=2}^{i-1} a_{ij} - \gamma, \quad i = 3, 4, 5$$

и свободным параметром c_4. Из этого семейства мы выбрали метод, имеющий $c_4 = 2/3$ и таблицу Бутчера

0	0					
1/3	1/6	1/6				
8/15	31/150	4/25	1/6			
2/3	23/88	8/99	125/792	1/6		
1/2	61/384	13/72	125/1152	−11/96	1/6	
b_i	1/6	0	0	0	2/3	1/6
$\hat{b}_i$	719/2400	−62/225	79/288	341/600	2/15	0

(5.36)

Решение ряда тестовых примеров показало, что методы ESDIRK, удовлетворяющие условиям (5.28), действительно имеют повышенные порядки сходимости при решении ДАУ индексов 2 и 3, по сравнению с методами, не удовлетворяющими этим условиям.

Характеристики методов 4-го порядка, в том числе и метода DESI (5.61), приведены в табл. 5.3. Все эти методы являются $L(\alpha)$-устойчивыми, т. е. имеют $R(\infty) = 0$, а значения $\hat{R}(\infty)$ приведены для вложенной формулы. Точность

оцениваем нормой функции погрешности $\|e_3(z)\| = \max(|e_3(z)|, \mathrm{Re}\, z < 0)$, а также значениями коэффициентов погрешности $e(T_{5i})$ – основной и $\hat{e}(T_{4i})$ – вложенной формул. Отметим, что при $q = 2$ имеем $e(T_{41}) = e(T_{42})$, $e(T_{43}) = e(T_{44})$, $e(T_{51}) = e(T_{52}) = e(T_{55})$, $e(T_{53}) = e(T_{54})$, $e(T_{56}) = e(T_{57})$, $e(T_{58}) = e(T_{59})$.

Таблица 5.3. Характеристики методов 4-го порядка

Метод	s	γ	α, град.	$\hat{R}(\infty)$	$\|e_3(z)\|$	$e(T_{5i})$, $i = 1, 3, 6, 8$	$\hat{e}(T_{4i})$, $i = 1, 3$
(5.32)	5	0.2204	89.55	2.70	0.0116	–0.053; –0.016; 0.092; 0.135	0.081; –0.153
(5.34)	6	1/4	90	2.33	0.0042	–0.003; 0.023; 0.128; 0.102	0.125; –0.125
(5.35)	6	1/6	89.95	0	0.0070	–0.019; 0.028; 0.053; 0.059	0.057; –0.093
(5.36)	6	1/6	89.95	0	0.0086	–0.042; –0.042; 0.033; 0.059	0.134; –0.093
(5.61)	6	1/6	89.95	1	0	0.005; 0.005; 0.059; 0.059	–0.111; –0.111

5.6. Методы 5-го порядка

Можно построить $L(\alpha)$-устойчивые методы ESDIRK при $p = r = 5$, если задать γ равным корню уравнения

$$D_5(\gamma) = \frac{1}{120} - \frac{5}{24}\gamma + \frac{5}{3}\gamma^2 - 5\gamma^3 + 5\gamma^4 - \gamma^5 = 0,$$

отличному от 0.0791. При заданном γ существует двухпараметрическое семейство таких методов, определяемое уравнениями (5.18)–(5.20), (5.22), (5.23). Приняв c_3 и c_4 в качестве свободных параметров, получим:

$$c_5 = \frac{4\gamma - 64\gamma^2 + 368\gamma^3 - 848\gamma^4 + 720\gamma^5 - c_3(1 - 18\gamma + 120\gamma^2 - 336\gamma^3 + 360\gamma^4)}{4\gamma - 56\gamma^2 + 288\gamma^3 - 600\gamma^4 + 480\gamma^5 - c_3(1 - 16\gamma + 96\gamma^2 - 240\gamma^3 + 240\gamma^4)}.$$

Веса b_2, b_3, b_4, b_5 находим из уравнений (5.20), после чего определяем

$$a_{43} = \frac{4 - 50\gamma + 160\gamma^2 - 120\gamma^3 - c_5(5 - 60\gamma + 180\gamma^2 - 120\gamma^3)}{60 b_4 c_3 (c_4 - c_5)(c_3 - 2\gamma)},$$

$$a_{54} = \frac{1 - 20\gamma + 120\gamma^2 - 240\gamma^3 + 120\gamma^4}{60 b_5 a_{43} c_3 (c_3 - 2\gamma)},$$

$$a_{53} = \frac{1 - 12\gamma + 36\gamma^2 - 24\gamma^3}{12 b_5 c_3 (c_3 - 2\gamma)} - \frac{b_4}{b_5} a_{43} - \frac{c_4(c_4 - 2\gamma)}{c_3(c_3 - 2\gamma)} a_{54}.$$

Остальные коэффициенты находим из (5.19), (5.18). Такие методы были рассмотрены в [53, 56], но нам не удалось добиться для них приемлемых результатов.

Задав $r = 6$, получим больше возможностей для построения эффективных методов. В этом случае методы, удовлетворяющие условиям (5.26), (5.27), образуют пятипараметрическое семейство. Рассмотрим последовательность построения таких методов. Значение γ однозначно определяет функцию устойчивости и может быть выбрано с помощью рис. 5.1в. Принимаем c_3, c_4, c_5, c_6, b_6 в качестве свободных параметров, задав которые, из (5.20) находим b_2, b_3, b_4, b_5.

Подставляя в (5.22), (5.23), (5.26), (5.27) выражения для a_{i2} (5.19), получаем систему уравнений, из которой находим a_{43}, a_{53}, a_{54}, a_{63}, a_{64}, a_{65}. Наконец, из (5.19), (5.18) определяем остальные коэффициенты. Мы выбрали $\gamma = 1/5$, $c_3 = 3/5$, что обеспечивает L-устойчивость и выполнение неравенства (5.17), поскольку $c_3^* = 0.518$. В результате подбора остальных параметров из условий минимизации коэффициентов погрешности и удобной реализации получен метод

$$
\begin{array}{c|cccccc|c}
0 & 0 & & & & & & \\
\frac{2}{5} & \frac{1}{5} & \frac{1}{5} & & & & & \\
\frac{3}{5} & \frac{1}{4} & \frac{3}{20} & \frac{1}{5} & & & & \\
1 & \frac{87}{140} & -\frac{27}{28} & \frac{8}{7} & \frac{1}{5} & & & \\
\frac{4}{5} & \frac{156973}{590625} & \frac{58202}{118125} & \frac{-35852}{118125} & \frac{12236}{84375} & \frac{1}{5} & & \\
\frac{1}{5} & \frac{2436319}{54337500} & \frac{-2573719}{10867500} & \frac{690136}{2716875} & \frac{-272248}{1940625} & \frac{9}{115} & \frac{1}{5} & \\
b_i & \frac{19}{288} & \frac{25}{144} & \frac{25}{144} & \frac{-193}{1440} & \frac{25}{96} & \frac{25}{96} & \frac{1}{5} \\
\hline
\hat{b}_i & \frac{-40133}{131744} & \frac{-98125}{197616} & \frac{-133225}{197616} & \frac{-42473}{131744} & \frac{570425}{395232} & \frac{23275}{17184} & 0
\end{array}
.
$$

В [56] приведен также жесткоточный метод ESDIRK 6-го порядка с семью неявными стадиями, но из-за большой разницы между классическим и стадийным порядком такие методы порядка 6 и выше вряд ли найдут практическое применение. Методы ESDIRK порядков от 2-го до 6-го вместе с обширной библиографией по методам DIRK приведены в обзоре [119].

5.7. Методы ESDIRK 3-го псевдостадийного порядка

Стадийный порядок методов ESDIRK не может быть выше 2-го, но можно построить такие методы 3-го псевдостадийного порядка. Согласно формуле (4.27), эти методы должны удовлетворять условиям

$$\mathbf{e}_{s-1}^{\mathrm{T}} \tilde{\mathbf{A}}^i (\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = 0, \quad i = 0, \ldots, s-2. \tag{5.37}$$

Условия 2-го стадийного порядка вместе с условием (5.37) при $i = 0$ обеспечивают 3-й порядок метода. Обозначим $\bar{\mathbf{A}} = \tilde{\mathbf{A}} - \gamma \mathbf{I}$. Тогда условия (5.37) эквивалентны следующим:

$$\mathbf{e}_{s-1}^{\mathrm{T}} \bar{\mathbf{A}}^i (\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = 0, \quad i = 0, \ldots, s-2. \tag{5.38}$$

При $i = s - 2$ получаем

$$\mathbf{e}_{s-1}^{\mathrm{T}} \bar{\mathbf{A}}^{s-2} (\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = a_{s,s-1} a_{s-1,s-2} \cdots a_{32} c_2^3 = 0, \tag{5.39}$$

откуда следует, что один из сомножителей в (5.39) должен быть равен 0. Это не может быть c_2, поскольку $\gamma > 0$, и не может быть a_{32} либо $a_{s,s-1}$, поскольку в этих случаях путем исключения одной из стадий метод приводится к методу с меньшим числом стадий.

Таким образом, минимальное число стадий, при котором метод, имеющий $q = 2$, удовлетворяет условию (5.39) и неприводим, равно 5. Приняв $s = 5$, получим:

$$\gamma = \frac{1}{2} \pm \frac{\sqrt{3}}{6}, \quad a_{43} = 0, \quad a_{53} = \frac{\gamma(1-3\gamma)}{3c_3(c_3-c_2)(c_3-c_4)}, \quad a_{54} = \frac{\gamma(1-3\gamma)}{3c_4(c_4-c_2)(c_4-c_3)}, \qquad (5.40)$$

c_3 и c_4 – свободные коэффициенты, а остальные коэффициенты находим из (5.19), (5.18). Чтобы метод был A-устойчивым, следует задать большее из значений γ, приведенных в (5.40). Функция устойчивости полученного метода:

$$R(z) = \frac{1+(1-2\gamma)z+(1/2-2\gamma+\gamma^2)z^2}{(1-\gamma z)^2}$$

(мы сократили сомножитель $(1 - \gamma z)^2$ в числителе и знаменателе). Такую же функцию устойчивости и 2-й стадийный порядок имеет метод

$$\begin{array}{c|ccc} 0 & 0 & & \\ 2\gamma & \gamma & \gamma & \\ b_i & -\dfrac{1-6\gamma+4\gamma^2}{4\gamma} & \dfrac{1-2\gamma}{4\gamma} & \gamma \end{array}. \qquad (5.41)$$

При решении тестовых задач метод, задаваемый формулами (5.40), практически не снижал порядка, в отличие от метода (5.41). Например, ошибка решения задачи Капса при $\mu = 100$ и $h = 0.01$ равна 4.99×10^{-8} у метода (5.40) и 1.55×10^{-7} у метода (5.41), а соответствующие оценки порядка равны 2.98 и 2.58.

Увеличим число стадий до 6 и потребуем, чтобы метод имел $\bar{q} = p = 3$ и был не только A-устойчивым, но и L-устойчивым. В этом случае должны выполняться условия 2-го стадийного порядка, $R(\infty) = 0$ и (5.38). При выполнении этих условий метод будет L-устойчивым, если γ – корень уравнения $\gamma^3 - 3\gamma^2 + 1.5\gamma - 1/6 = 0$, равный 0.435866... Такие методы образуют четырехпараметрическое семейство с коэффициентами

$$\begin{gathered} a_{43} = 0, \quad a_{53} = \frac{\alpha\gamma^2}{3a_{65}c_3(c_3-c_4)(c_3-c_2)}, \quad a_{54} = \frac{\alpha\gamma^2}{3a_{65}c_4(c_4-c_3)(c_4-c_2)}, \\ a_{63} = \frac{\beta-\alpha c_4-3a_{65}c_5(c_5-c_4)(c_5-c_2)}{3c_3(c_3-c_4)(c_3-c_2)}, \quad a_{64} = \frac{\beta-\alpha c_3-3a_{65}c_5(c_5-c_3)(c_5-c_2)}{3c_4(c_4-c_3)(c_4-c_2)}, \\ \alpha = 6\gamma^2 - 6\gamma + 1, \quad \beta = 33\gamma^2 - 23\gamma + 3, \end{gathered} \qquad (5.42)$$

где a_{65}, c_3, c_4 и c_5 – свободные коэффициенты, а остальные коэффициенты находим из (5.19) и (5.18).

Полученные методы удовлетворяют условиям (5.34), входящим в число условий порядка для ДАУ индексов 2 и 3. Дополнительно потребуем выполнения условий

$$\tilde{\mathbf{b}}^{\mathrm{T}}\left(\tilde{\mathbf{c}}(\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^{3})\right)=2,\quad \tilde{\mathbf{b}}^{\mathrm{T}}(\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^{3})^{2}=12, \tag{5.43}$$

которые вместе с (5.34) являются необходимыми для обеспечения 3-го порядка при решении ДАУ индекса 2, а также 3-го порядка дифференциальных и 2-го порядка алгебраических переменных при решении ДАУ индекса 3. При выбранном γ формулы (5.42), (5.43) задают двухпараметрическое семейство. Задаем γ = 0.43586652150846, c_4 = 1, c_5 = 0, тогда из (5.43) получаем c_3 = 0.07120169028617, a_{65} = –1.80278903126746, а остальные коэффициенты находим по формулам (5.42), (5.19), (5.18). Полученный метод имеет вложенную формулу 2-го порядка, совпадающую с 4-й стадией и позволяющую получить оценку ошибки в виде нормы вектора $\mathbf{y}_{n+1} - \mathbf{Y}_4$. Обозначим этот метод через ESDIRK323 (p = 3, q = 2, $\overline{q}$ = 3). При решении задачи Капса его результаты практически совпадают с результатами метода более высокого стадийного порядка DESI33, приведенными на рис. 4.1 и 4.2.

Попробуем построить аналогичный L-устойчивый метод 4-го порядка, для чего увеличим число стадий до 7. Чтобы метод имел $\overline{q}$ = 4, наряду с условиями q = 2 и (5.38) должны выполняться также условия, обеспечивающие $e_{4j}(z) \equiv 0$ при j = 1,..., 5. Для этого при q = 2 достаточно обеспечить выполнение условий $e_{41}(z) \equiv 0$ и $e_{43}(z) \equiv 0$, откуда получаем:

$$\mathbf{e}_{s-1}^{\mathrm{T}}\bar{\mathbf{A}}^{i}(\tilde{\mathbf{c}}^{4}-4\tilde{\mathbf{A}}\tilde{\mathbf{c}}^{3})=0,\quad i=0,\ldots,s-2. \tag{5.44}$$

$$\mathbf{e}_{s-1}^{\mathrm{T}}\bar{\mathbf{A}}^{i}(3\tilde{\mathbf{c}}(\tilde{\mathbf{A}}\tilde{\mathbf{c}}^{2})-4\tilde{\mathbf{A}}\tilde{\mathbf{c}}^{3})=0,\quad i=0,\ldots,s-2. \tag{5.45}$$

При q = 2 условие (5.38) и любое из условий (5.44) или (5.45) обеспечивают 4-й порядок метода.

Нахождение коэффициентов, удовлетворяющих условиям (5.38), (5.44) и (5.45), удобно производить в порядке уменьшения i. Выполнение каждого из этих условий при $i = s - 2, s - 3$ обеспечивают соотношения

a_{43} = 0, $a_{53}c_3(c_3 - c_2) + a_{54}(c_4 - c_2)$ = 0.

При меньших значениях i и s = 7 условия вступают в противоречие, поэтому мы отказались от выполнения (5.45). В результате найдено четырехпараметрическое семейство методов 4-го порядка (при заданном γ), удовлетворяющих условиям (5.38), (5.44). При построении метода свободные коэффициенты c_4, c_5, c_6 и a_{76} были выбраны, исходя из условий удобной и эффективной реализации, а γ – из условий точности и $L(\alpha)$-устойчивости.

Построенный метод ESDIRK423 имеет коэффициенты:

γ = 0.22042841025921, $c_2 = 2\gamma$, $c_3 = 2.5\gamma$, $c_4 = 3.5\gamma$, $c_5 = c_6 = c_7$ = 1,
a_{32} = 0.068883878206, a_{42} = 0.28931228846522, a_{43} = 0, a_{52} = 1.04217927873049,
a_{53} = –0.48962904540024, a_{54} = 0.11657834414291, a_{62} = 1.72653078404202,

$a_{63} = -1.36993741849107$, $a_{64} = 0.45371449007852$, $a_{65} = -0.07668893711619$,
$a_{72} = 4.1884035378415$, $a_{73} = -5.43022820816131$, $a_{74} = 2.70925969569028$,
$a_{75} = -1.89698055635424$, $a_{76} = 1.23231498057714$, $a_{i1} = c_i - \sum_{j=2}^{i} a_{ij}$, $i = 2,\ldots, 7$.

Метод имеет жесткоточную вложенную формулу 3-го порядка, совпадающую с 6-й стадией, и является $L(89.55°)$-устойчивым.

Методы ESDIRK323 и ESDIRK423, а также методы ESDIRK63PR и ESDIRK74PR, построенные в [137] на основе анализа ошибки уравнения Протеро–Робинсона, использовались для решения задачи PLATE. Результаты приведены в табл. 5.3, где $scd\,(h)$ – значение scd, вычисленное по формуле (3.18) при размере шага h, а оценка порядка получена в виде $\tilde{p} = scd(0.01) - scd(0.1)$. Методы, имеющие $\bar{q} > q$, обеспечивают сходимость с более высоким реальным порядком.

Таблица 5.4. Результаты решения задачи PLATE

Метод	s	p	q	$\bar{q}$	$scd(0.1)$	$scd(0.01)$	$\tilde{p}$
ESDIRK63PR	6	3	2	2	3.77	5.96	2.19
ESDIRK323	6	3	2	3	3.65	6.35	2.70
ESDIRK74PR	7	4	2	2	4.46	6.87	2.41
ESDIRK423	7	4	2	3	4.52	8.46	3.94

Построение восьмистадийного метода ESDIRK 5-го порядка, имеющего $\bar{q} = 3$, рассмотрено в [56].

5.8. Двухшаговые диагонально-неявные методы

Двухшаговые методы Рунге–Кутты (TSRK) [94, 113, 114, 146] обобщают обычные одношаговые методы, используя на очередном шаге информацию, полученную не только на текущем, но и на предыдущем шаге. Благодаря этому двухшаговые методы могут иметь более высокий стадийный порядок, что позволяет повысить их точность при решении жестких и дифференциально-алгебраических задач. Общий класс двухшаговых методов Рунге–Кутты предложен в [113]. Рассмотрим один частный подкласс этого класса, предложенный в [59] и задаваемый формулами

$$\mathbf{Y}_n^1 = \mathbf{y}_n, \quad \mathbf{F}_n^1 = \mathbf{f}(t_n, \mathbf{y}_n), \tag{5.46а}$$

$$\mathbf{Y}_n^i = \mathbf{y}_n + h_n \sum_{j=1}^{s} (b_{ij}\mathbf{F}_{n-1}^j + a_{ij}\mathbf{F}_n^j), \quad \mathbf{F}_n^i = \mathbf{f}(t_n + c_i h_n, \mathbf{Y}_n^i), \quad i = 2, \ldots, s-1, \tag{5.46б}$$

$$\mathbf{Y}_n^s = \mathbf{y}_n + h_n \sum_{j=1}^{s} a_{s,j}\mathbf{F}_n^j, \quad \mathbf{F}_n^s = \mathbf{f}(t_n + h_n, \mathbf{Y}_n^s), \quad \mathbf{y}_{n+1} = \mathbf{Y}_n^s, \tag{5.46в}$$

где h_n – размер очередного шага, а $\mathbf{Y}_n^i$ и $\mathbf{F}_n^i$, $i = 1,\ldots, s$ – стадийные значения и их производные на этом шаге. Будем называть стадии (5.46б) *внутренними*, а стадию (5.46в) *заключительной*. Как и в методах ESDIRK, первая стадия (5.46а) не

требует вычислений, поскольку результат ее выполнения совпадает с результатом выполнения заключительной стадии предыдущего шага. Представим коэффициенты метода (5.46) в виде двух матриц и вектора

$\mathbf{A} = [a_{ij}]$, $\mathbf{B} = [b_{ij}]$, $\mathbf{c} = [c_i]$, $i, j = 1,\ldots, s$,

при этом примем $c_1 = 0$, $c_s = 1$, $a_{1i} = b_{1i} = b_{si} = 0$, $i = 1,\ldots, s$.

В предлагаемом способе построения двухшаговых методов вида (5.46) за основу берется обычный одношаговый метод Рунге–Кутты, задаваемый матрицей $\mathbf{A}$, при этом $\mathbf{c} = \mathbf{A}\mathbf{e}$ и $\mathbf{b} = (a_{s1},\ldots, a_{ss})^{\mathrm{T}}$. Назовем его *исходным методом*. Матрицу $\mathbf{B}$ примем в виде

$$\mathbf{B} = \mathbf{d}\mathbf{g}^{\mathrm{T}}, \tag{5.47}$$

где векторы $\mathbf{d}$ и $\mathbf{g}$ определяем из условий повышения стадийного порядка исходного метода и обеспечения необходимых свойств устойчивости. В общем случае все коэффициенты двухшаговых методов зависят от соотношения размеров шагов $w = h_n/h_{n-1}$. В наших методах от w зависят только коэффициенты вектора $\mathbf{g}$, что упрощает их реализацию с переменным шагом. Первый шаг выполняем исходным одношаговым методом, поэтому предложенные методы не нуждаются в специальной стартовой процедуре.

С учетом (5.47) формулы (5.46) можно записать в виде, более удобном для реализации:

$$\begin{gathered}
\mathbf{Y}_n^1 = \mathbf{y}_n, \quad \mathbf{F}_n^1 = \mathbf{f}(t_n, \mathbf{y}_n), \quad \mathbf{u} = \sum_{j=1}^{s} g_j \mathbf{F}_{n-1}^j, \\
\mathbf{Y}_n^i = \mathbf{y}_n + h_n\left(d_i\mathbf{u} + \sum_{j=1}^{s} a_{ij}\mathbf{F}_n^j\right), \quad \mathbf{F}_n^i = \mathbf{f}(t_n + c_i h_n, \mathbf{Y}_n^i), \quad i = 2,\ldots, s-1, \\
\mathbf{Y}_n^s = \mathbf{y}_n + h_n\sum_{j=1}^{s} a_{sj}\mathbf{F}_n^j, \quad \mathbf{F}_n^s = \mathbf{f}(t_n + h_n, \mathbf{Y}_n^s), \quad \mathbf{y}_{n+1} = \mathbf{Y}_n^s.
\end{gathered} \tag{5.48}$$

Чтобы двухшаговый метод (5.46) имел стадийный порядок q, должны выполняться равенства

$$k\left[\mathbf{B}\left(\frac{\mathbf{c}-\mathbf{e}}{w}\right)^{k-1} + \mathbf{A}\mathbf{c}^{k-1}\right] = \mathbf{c}^k, \quad k = 1, \ldots, q.$$

Пусть исходный метод имеет стадийный порядок $\tilde{q}$. Стадийный порядок двухшагового метода будет $q = \tilde{q} + 1$, если задать векторы $\mathbf{d}$ и $\mathbf{g}$, исходя из условий

$$\mathbf{d} = \mathbf{c}^{\tilde{q}+1} - (\tilde{q}+1)\mathbf{A}\mathbf{c}^{\tilde{q}}, \quad \mathbf{g}^{\mathrm{T}}\mathbf{c}^k = 0, \ k = 0,\ldots, \tilde{q}-1, \quad \mathbf{g}^{\mathrm{T}}\mathbf{c}^{\tilde{q}} = \frac{w^{\tilde{q}}}{\tilde{q}+1}, \tag{5.49}$$

а чтобы было $q = \tilde{q} + 2$, дополнительно должны выполняться условия

$$\mathbf{c}^{\tilde{q}+2} - (\tilde{q}+2)\mathbf{A}\mathbf{c}^{\tilde{q}+1} = \alpha(\mathbf{c}^{\tilde{q}+1} - (\tilde{q}+1)\mathbf{A}\mathbf{c}^{\tilde{q}}), \quad \mathbf{g}^{\mathrm{T}}\mathbf{c}^{\tilde{q}+1} = \alpha\frac{w^{\tilde{q}+1}}{\tilde{q}+2} + w^{\tilde{q}},$$

где α – некоторая константа. При этом двухшаговый метод (5.48) имеет порядок $p = q + 1$, если $(q + 1)\mathbf{b}^{\mathrm{T}}\mathbf{c}^{q} = 1$, и порядок $p = q + 2$, если дополнительно выполняются равенства $(q + 2)\mathbf{b}^{\mathrm{T}}\mathbf{c}^{q+1} = 1$, $(q + 2)(q + 1)\mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{c}^{q} = 1$.

Исследуем устойчивость двухшагового метода. Используя формулы (5.46), (5.47) для решения уравнения Далквиста $y' = \lambda y$, получаем:

$$\mathbf{Y}_n = \mathbf{H}(z)\mathbf{Y}_{n-1}, \quad \mathbf{H}(z) = (\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{e}\mathbf{e}_s^{\mathrm{T}} + z\mathbf{d}\mathbf{g}^{\mathrm{T}}), \quad z = h_n\lambda.$$

Устойчивость этой разностной схемы определяется спектром матрицы $\mathbf{H}(z)$, т. е. корнями характеристического многочлена $P(\eta, z) = |\eta\mathbf{I} - \mathbf{H}(z)|$. В общем случае $P(\eta, z)$, как многочлен от η, имеет s различных корней, зависящих от z, но при выборе матрицы $\mathbf{B}$ в виде (5.47) имеем

$$P(\eta, z) = \eta^{s-2}[\eta^2 - p_1(z)\eta + p_0(z)], \tag{5.50}$$

где

$$\begin{aligned} &p_1(z) = v_{11}(z) + v_{22}(z), \quad p_0(z) = v_{11}(z)v_{22}(z) - v_{12}(z)v_{21}(z), \\ &v_{11}(z) = \mathbf{e}_s^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}, \quad v_{12}(z) = \mathbf{e}_s^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}z\mathbf{d}, \\ &v_{21}(z) = \mathbf{g}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}, \quad v_{22}(z) = \mathbf{g}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}z\mathbf{d}. \end{aligned} \tag{5.51}$$

Заметим, что $v_{11}(z)$ является функцией устойчивости исходного одношагового метода.

Зададим в качестве исходного метод ESDIRK 2-го стадийного порядка. Тогда, чтобы получить двухшаговый метод 3-го стадийного порядка, в соответствии с (5.49) следует задать векторы $\mathbf{d}$ и $\mathbf{g}$, исходя из условий

$$\mathbf{d} = \mathbf{c}^3 - 3\mathbf{A}\mathbf{c}^2, \quad \mathbf{g}^{\mathrm{T}}\mathbf{e} = 0, \quad \mathbf{g}^{\mathrm{T}}\mathbf{c} = 0, \quad \mathbf{g}^{\mathrm{T}}\mathbf{c}^2 = w^2/3. \tag{5.52}$$

Дополнительно потребуем, чтобы метод обладал $L(\alpha)$-устойчивостью. Для этого необходимо, чтобы при любом w коэффициенты характеристического многочлена (5.50) удовлетворяли условию $p_1(\infty) = 0$, $p_0(\infty) = 0$. Из (5.51), (5.52), принимая $\tilde{\mathbf{g}} = (g_2, \ldots, g_s)^{\mathrm{T}}$, получаем:

$$\begin{aligned} &v_{11}(\infty) = 1 - \mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}, \quad v_{12}(\infty) = 3 - \mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3, \\ &v_{21}(\infty) = -\tilde{\mathbf{g}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}, \quad v_{22}(\infty) = w^2 - \tilde{\mathbf{g}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3. \end{aligned}$$

Из условия $p_1(\infty) = 0$ получаем

$$\mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}} = 1, \tag{5.53а}$$

$$\tilde{\mathbf{g}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3 = w^2, \tag{5.53б}$$

а чтобы было также и $p_0(\infty) = 0$, дополнительно должно выполняться одно из следующих равенств:

$$\mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^3 = 3, \tag{5.54а}$$

$$\mathbf{g}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}} = 0. \tag{5.54б}$$

Таким образом, необходимым условием $L(\alpha)$-устойчивости двухшагового метода является выполнение равенств (5.53) и одного из равенств (5.54).

Процедуры построения методов ESDIRK, удовлетворяющих условиям (5.53а) и (5.54а), рассматривались в разделах 5.5, 5.6. Такие методы целесообразно использовать в качестве исходных при построении двухшаговых диагонально-неявных методов (TSDIRK) 3-го стадийного порядка. В этом случае коэффициенты вектора **g** определяем из условий (5.52), (5.53б), а выполнение равенства (5.54б) является излишним.

С помощью полученных формул было построено несколько методов TSDIRK 4-го порядка. Приведем один из них, который имеет наименьшее число ненулевых коэффициентов векторов **d** и **g** и поэтому наиболее удобен для реализации. Он задается коэффициентами

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 55/196 & 2/49 & 1/4 & 0 & 0 & 0 \\ 17/96 & 7/12 & -49/96 & 1/4 & 0 & 0 \\ 5/48 & 13/24 & -49/48 & 9/8 & 1/4 & 0 \\ 1/6 & 0 & 0 & 2/3 & -1/12 & 1/4 \end{bmatrix},\quad \mathbf{c}=\begin{bmatrix} 0 \\ 1/2 \\ 4/7 \\ 1/2 \\ 1 \\ 1 \end{bmatrix},\quad \mathbf{d}=\begin{bmatrix} 0 \\ -1/16 \\ -61/686 \\ 0 \\ 0 \\ 0 \end{bmatrix},\quad \mathbf{g}=\frac{w^2}{6}\begin{bmatrix} 4 \\ 0 \\ 0 \\ -8 \\ 1 \\ 3 \end{bmatrix}$$

и является $L(89.58°)$-устойчивым. Предпоследнюю стадию можно использовать в качестве вложенной формулы. В [59] приведены результаты этого метода при решении жестких задач и ДАУ индекса 2 в сравнении с методом ESDIRK. Эти результаты показали преимущество двухшагового метода, объясняемое его более высоким стадийным порядком.

5.9. Диагонально расширенные однократно неявные методы

Рассмотрим сначала методы Рунге–Кутты с заполненной матрицей **A**, имеющей одно s-кратное собственное значение. Такие методы называют однократно неявными (SIRK), а их стадийный порядок ограничен только числом стадий s. Преобразование переменных, приводящее матрицу **A** к форме Жордана, позволяет сократить вычислительные затраты этих методов. Как и SDIRK, методы SIRK позволяют ограничиться на каждом шаге одним LU-разложением матрицы $n \times n$, где n – число переменных, а расчет стадийных значений $\mathbf{Y}_i$ сводится к решению s алгебраических систем с одной и той же матрицей порядка n. Преобразование переменных требует дополнительных вычислений, но их объем меньше, чем у полностью неявных методов. В то же время предложенные ранее методы SIRK не свободны от недостатков: они не являются жесткоточными, а отдельные значения c_i могут заметно превышать 1 [12, 75].

В [91, 92] были предложены диагонально расширенные однократно неявные методы (DESI), сочетающие достоинства методов SDIRK и SIRK и свободные от их недостатков. Первые несколько стадий этих методов выполняются аналогично методам SIRK, а последующие стадии аналогичны методам SDIRK.

Особый интерес представляют предложенные в [92] жесткоточные методы DESI с явной первой стадией. Таблица Бутчера таких методов имеет вид:

$$\begin{array}{c|cccccccc}
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
c_2 & a_{21} & a_{22} & \cdots & a_{2r} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
c_r & a_{r1} & a_{r2} & \cdots & a_{rr} & 0 & 0 & \cdots & 0 \\
c_{r+1} & a_{r+1,1} & a_{r+1,2} & \cdots & a_{r+1,r} & \gamma & 0 & \cdots & 0 \\
c_{r+2} & a_{r+2,1} & a_{r+2,2} & \cdots & a_{r+2,r} & a_{r+2,r+1} & \gamma & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
c_s = 1 & a_{s1} & a_{s2} & \cdots & a_{sr} & a_{s,r+1} & a_{s,r+2} & \cdots & \gamma \\
\hline
 & a_{s1} & a_{s2} & \cdots & a_{sr} & a_{s,r+1} & a_{s,r+2} & \cdots & \gamma
\end{array} \qquad (5.55)$$

Первая стадия – явная, а последующие $r-1$ стадии соответствуют методу SIRK, поэтому матрица

$$\bar{\mathbf{A}} = \begin{bmatrix} a_{22} & \cdots & a_{2r} \\ \vdots & \cdots & \vdots \\ a_{r2} & \cdots & a_{rr} \end{bmatrix} \qquad (5.56)$$

имеет одно $(r-1)$-кратное собственное значение, равное γ. Последние $s-r$ стадии аналогичны стадиям метода SDIRK с диагональным элементом γ.

Стадийный порядок q метода вида (5.55) ограничен неравенством $q \le r$. При $q = r = 2$ получаем методы ESDIRK. Наша задача – построить методы такого типа с более высоким стадийным порядком. Рассмотрим методы DESI вида (5.55) при $s = 6$ и $\gamma = 1/6$, которые являются $L(89.95°)$-устойчивыми, имеют малые члены погрешности и удобные для реализации коэффициенты.

При построении методов, имеющих $q \ge 3$, применяем следующие условия.

1. Стадийный порядок q обеспечивается путем выполнения равенств
$$\mathbf{A}\mathbf{c}^{k-1} = \mathbf{c}^k/k, \quad k = 1,\ldots,q. \qquad (5.57)$$
2. Для обеспечения 4-го порядка метода и $L(\alpha)$-устойчивости должны выполняться равенства
$$\mathbf{b}^{\mathrm{T}}\mathbf{c}^3 = 1/4, \quad \mathbf{b}^{\mathrm{T}}\mathbf{A}^3\mathbf{c} = \frac{1}{120} - D_5(\gamma) = \frac{5}{24}\gamma - \frac{5}{3}\gamma^2 + 5\gamma^3 - 5\gamma^4 + \gamma^5 \qquad (5.58)$$
(при $q = 4$ первое из условий (5.58) следует из (5.57), поэтому излишне).
3. Чтобы метод был однократно неявным, матрица (5.56) должна иметь одно $(r-1)$-кратное собственное значение, равное γ:
$$\left| z\mathbf{I}_{r-1} - \bar{\mathbf{A}} \right| = (z-\gamma)^{r-1}. \qquad (5.59)$$

При $q = r = 3$ абсциссы c_2 и c_3 однозначно определяются из условий (5.57), (5.59) и совпадают с гауссовыми узлами. Задав также $c_4 = c_5 = 1$, получим метод

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3-\sqrt{3}}{6} & \frac{\sqrt{3}}{18} & \frac{6-\sqrt{3}}{36} & \frac{12-7\sqrt{3}}{36} & 0 & 0 & 0 \\
\frac{3+\sqrt{3}}{6} & -\frac{\sqrt{3}}{18} & \frac{12+7\sqrt{3}}{36} & \frac{6+\sqrt{3}}{36} & 0 & 0 & 0 \\
1 & -1/6 & \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{6} & 1/6 & 0 & 0 \\
1 & -5/36 & \frac{18+5\sqrt{3}}{36} & \frac{18-5\sqrt{3}}{36} & -1/36 & 1/6 & 0 \\
b_i & 0 & 1/2 & 1/2 & -1 & 5/6 & 1/6 \\
\hline
\hat{b}_i & -5/36 & \frac{18+5\sqrt{3}}{36} & \frac{18-5\sqrt{3}}{36} & -1/36 & 1/6 & 0
\end{array}. \qquad (5.60)
$$

Вложенная формула совпадает с 5-й стадией ($\hat{\mathbf{y}}_1 = \mathbf{Y}_5$) и имеет $\hat{R}(\infty) = 0$.

При $r = 4$ также можно построить метод, имеющий $q = r$, но в этом случае абсцисса $c_4 = 1.293$ выходит за пределы интервала [0, 1], что нежелательно. Поэтому мы задали $q = r - 1 = 3$, а появившиеся свободные параметры использовали для удовлетворения дополнительных условий

$$\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^4 = 4, \quad \tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-3}\tilde{\mathbf{c}}^4 = 12, \quad \tilde{\mathbf{b}}^{\mathrm{T}}\left(\tilde{\mathbf{c}}(\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^4)\right) = 3,$$

где $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$ и $\tilde{\mathbf{c}}$ получаем из представления (4.4). Эти условия являются необходимыми для того, чтобы при решении ДАУ индекса 2 метод имел 4-й порядок, а при решении ДАУ индекса 3 обеспечивались 4-й порядок дифференциальных и 3-й порядок алгебраических переменных. Оставшиеся два свободных параметра c_2 и c_3 выбраны из условий жесткоточной вложенной формулы и удобной реализации. В результате получен метод

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 11/81 & 1/4 & -4/81 & -1/324 & 0 & 0 \\
1/2 & 1/8 & 3/8 & 0 & 0 & 0 & 0 \\
1 & 0 & 3/4 & 0 & 1/4 & 0 & 0 \\
3/4 & 131/1536 & 567/1024 & -7/128 & -1/1024 & 1/6 & 0 \\
b_i & 5/24 & 3/5 & -2/9 & -1/9 & 64/135 & 1/6 \\
\hline
\hat{b}_i & 0 & 3/4 & 0 & 1/4 & 0 & 0
\end{array}, \qquad (5.61)
$$

для которого $\hat{\mathbf{y}}_{n+1} = \mathbf{Y}_4$, $\hat{R}(\infty) = 1$. Характеристики этого метода в сравнении с методами ESDIRK приведены в табл. 5.3.

Наконец, был построен метод, имеющий $r = 5$ и $q = 4$:

0	0	0	0	0	0	0
1/6	473/6480	47/480	−1/648	−1/160	23/6480	0
1/2	1/80	57/160	1/8	3/160	−1/80	0
5/6	−43/1296	43/96	115/648	11/32	−133/1296	0
1	1/10	3/20	1/2	3/20	1/10	0
b_i	−11/90	13/20	−5/90	13/20	−13/45	1/6
$\hat{b}_i$	7/45	−1/10	19/18	−3/5	22/45	0

(5.62)

Коэффициенты вложенной формулы обеспечивают $\hat{p} = 3$ и $\hat{R}(\infty) = 0$.

В [67] приведены результаты решения тестовых задач методами DESI (5.60)–(5.62) в сравнении с методом ESDIRK, имеющим такую же функцию устойчивости. Методы DESI были более эффективными, а среди них лучшими оказались методы (5.61) и (5.62), которые показали примерно одинаковые результаты. Но при этом метод (5.61) более удобен для программной реализации.

5.10. Реализация методов ESDIRK

Рассмотрим реализацию методов ESDIRK применительно к системе ДАУ

$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0,$

$\mathbf{0} = \mathbf{g}(t, \mathbf{y}, \mathbf{z}), \quad \mathbf{z}(t_0) = \mathbf{z}_0.$

Формулы одного шага численного решения этой системы запишутся в виде:

$$\mathbf{F}_1 = \mathbf{f}_n, \tag{5.63а}$$

$$\left.\begin{aligned} \mathbf{Y}_i = \mathbf{y}_n + h\sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j + h\gamma\mathbf{F}_i, \quad \mathbf{F}_i = \mathbf{f}(t_0 + c_i h, \mathbf{Y}_i, \mathbf{Z}_i), \\ \mathbf{0} = \mathbf{g}(t_n + c_i h, \mathbf{Y}_i, \mathbf{Z}_i) \end{aligned}\right\} \quad i = 2, \ldots, s, \tag{5.63б}$$

$$\mathbf{y}_{n+1} = \mathbf{Y}_s, \quad \mathbf{z}_{n+1} = \mathbf{Z}_s, \quad \mathbf{f}_{n+1} = \mathbf{F}_s. \tag{5.63в}$$

При реализации неявных методов удобно иметь дело с приращениями, которые обозначим через $\Delta\mathbf{Y}_i = \mathbf{Y}_i - \mathbf{y}_n$ и $\Delta\mathbf{Z}_i = \mathbf{Z}_i - \mathbf{z}_n$. Обозначим через $\mathbf{f}_y, \mathbf{f}_z, \mathbf{g}_y\, \mathbf{g}_z$ соответствующие матрицы частных производных, вычисленные в некоторой точке численного решения (предполагается, что эти матрицы не изменяются в течение нескольких шагов). Упрощенные итерации метода Ньютона при реализации i-й стадии (5.63б) запишутся в виде:

$$\begin{bmatrix}\Delta\mathbf{Y}_i^k\\ \Delta\mathbf{Z}_i^k\end{bmatrix}=\begin{bmatrix}\Delta\mathbf{Y}_i^{k-1}\\ \Delta\mathbf{Z}_i^{k-1}\end{bmatrix}+\begin{bmatrix}\mathbf{I}-h\gamma\mathbf{f}_y & -h\gamma\mathbf{f}_z\\ -\mathbf{g}_y & -\mathbf{g}_z\end{bmatrix}^{-1}\begin{bmatrix}h\sum\limits_{j=1}^{i-1}a_{ij}\mathbf{F}_j+h\gamma\mathbf{F}_i^{k-1}-\Delta\mathbf{Y}_i^{k-1}\\ \mathbf{G}_i^{k-1}\end{bmatrix},$$

$$\begin{bmatrix}\mathbf{Y}_i^k\\ \mathbf{Z}_i^k\end{bmatrix}=\begin{bmatrix}\mathbf{y}_n+\Delta\mathbf{Y}_i^k\\ \mathbf{z}_n+\Delta\mathbf{Z}_i^k\end{bmatrix},\quad\begin{bmatrix}\mathbf{F}_i^k\\ \mathbf{G}_i^k\end{bmatrix}=\begin{bmatrix}\mathbf{f}(t_n+c_ih,\,\mathbf{Y}_i^k,\mathbf{Z}_i^k)\\ \mathbf{g}(t_n+c_ih,\,\mathbf{Y}_i^k,\mathbf{Z}_i^k)\end{bmatrix},\qquad(5.64)$$

$$\Delta\mathbf{Y}_i=\Delta\mathbf{Y}_i^N,\quad\Delta\mathbf{Z}_i=\Delta\mathbf{Z}_i^N,\quad\mathbf{Y}_i=\mathbf{Y}_i^N\quad\mathbf{Z}_i=\mathbf{Z}_i^N,$$

где $k = 1, \ldots, N$, N – число выполненных итераций.

Перед началом итераций нужно задать начальные значения $\mathbf{Y}_i^0$, $\mathbf{Z}_i^0$, $\mathbf{F}_i^0$ и $\mathbf{G}_i^0$, а после их окончания следует вычислить значение $\mathbf{F}_i$, которое будет использовано на последующих стадиях либо (при $i = s$) на следующем шаге. На первой стадии принимаем (5.63а), а после завершения последней стадии – (5.63в). Таким образом, схема реализации одного шага метода ESDIRK задается формулами (5.64) и формулами вычисления $\mathbf{Y}_i^0$, $\mathbf{Z}_i^0$, $\mathbf{F}_i^0$, $\mathbf{G}_i^0$ и $\mathbf{F}_i$. Рассмотрим конкретные схемы.

Тривиальная схема (T). Наиболее простой способ – использование тривиального прогноза, в котором в качестве начальных значений для итераций применяются значения, полученные в начальной точке шага интегрирования. Тривиальная схема задается значениями $\mathbf{Y}_i^0 = \mathbf{y}_n$, $\mathbf{Z}_i^0 = \mathbf{z}_n$ $\mathbf{F}_i^0 = \mathbf{f}_n$, $\mathbf{G}_i^0 = \mathbf{g}(t_n, \mathbf{y}_n, \mathbf{z}_n)$, $\mathbf{F}_i = \mathbf{f}(t_n + c_ih, \mathbf{Y}_i, \mathbf{Z}_i)$.

Стандартная схема (S). Для уменьшения числа итераций применяют нетривиальный прогноз, в котором начальные значения для итераций задают в виде линейной комбинации предыдущих стадийных значений. Такой прогноз используется во многих решателях, например RADAU5 [75]. Применительно к методам DIRK в [75, 112] предлагалось строить формулу прогноза по стадийным значениям только текущего шага. На первых стадиях целесообразно использовать также и значения предыдущего шага. Двухшаговый прогноз запишется в виде:

$$\mathbf{Y}_i^0=\sum_{j=1}^{s-1}\alpha_{ij}\bar{\mathbf{Y}}_j+\sum_{j=1}^{i-1}\beta_{ij}\mathbf{Y}_j,\quad\mathbf{Z}_i^0=\sum_{j=1}^{s-1}\alpha_{ij}\bar{\mathbf{Z}}_j+\sum_{j=1}^{i-1}\beta_{ij}\mathbf{Z}_j,\qquad(5.65)$$

где $\bar{\mathbf{Y}}_j$ и $\bar{\mathbf{Z}}_j$ – стадийные значения, полученные на предыдущем шаге.

Начальное значение $\mathbf{Y}_i^0$ имеет порядок r, если $\mathbf{Y}_i^0 - \mathbf{Y}_i^* = O(h^{r+1})$, где $\mathbf{Y}_i^*$ – точное решение уравнений (5.63б). Обычно порядок прогноза совпадает со стадийным порядком q, тогда формулу прогноза можно задать как значение интерполяционного многочлена, построенного по $q + 1$ стадийным значениям. После вычисления начальных значений (5.65) принимаем

$$\mathbf{F}_i^0=\mathbf{f}(t_n+c_ih,\,\mathbf{Y}_i^0,\mathbf{Z}_i^0),\quad\mathbf{G}_i^0=\mathbf{g}(t_n+c_ih,\,\mathbf{Y}_i^0,\mathbf{Z}_i^0),\qquad(5.66)$$

а после выполнения итераций (5.64) вычисляем $\mathbf{F}_i$ из формулы вычисления $\mathbf{Y}_i$ (5.63б), откуда

$$\mathbf{F}_i = \frac{1}{\gamma}\left(\frac{\Delta\mathbf{Y}_i}{h} - \sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j\right). \tag{5.67}$$

Использование (5.67) вместо $\mathbf{F}_i = \mathbf{f}(t_n + c_i h, \mathbf{Y}_i, \mathbf{Z}_i)$ позволяет сэкономить одно вычисление правой части, а для жестких задач дает более точный результат.

Схема с прогнозом для производных (D). Выполняем предварительный прогноз для производных:

$$\hat{\mathbf{F}}_i = \sum_{j=1}^{s-1} \alpha_{ij}\bar{\mathbf{F}}_j + \sum_{j=1}^{i-1} \beta_{ij}\mathbf{F}_j, \quad \hat{\mathbf{Z}}'_i = \sum_{j=1}^{s-1} \alpha_{ij}\bar{\mathbf{Z}}'_i + \sum_{j=1}^{i-1} \beta_{ij}\mathbf{Z}'_j.$$

Прогноз для переменных получаем, подставляя полученные значения в формулу i-й стадии:

$$\mathbf{Y}_i^0 = \mathbf{y}_n + h\sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j + h\gamma\hat{\mathbf{F}}_i, \quad \mathbf{Z}_i^0 = \mathbf{z}_n + h\sum_{j=1}^{i-1} a_{ij}\mathbf{Z}'_j + h\gamma\hat{\mathbf{Z}}'_i,$$

после чего вычисляем $\mathbf{F}_i^0$ и $\mathbf{G}_i^0$ по формулам (5.66). По окончании итераций вычисляем $\mathbf{F}_i$ по формуле (5.67), а также вычисляем

$$\mathbf{Z}'_i = \frac{1}{\gamma}\left(\frac{\Delta\mathbf{Z}_i}{h} - \sum_{j=1}^{i-1} a_{ij}\mathbf{Z}'_j\right).$$

Такая схема позволяет сформировать наиболее точный прогноз, но он наименее устойчив.

Экономичная схема (E). Использование нетривиального прогноза (5.65) позволяет уменьшить число итераций, но в этом случае добавляются вычисления правой части по формулам (5.66). В экономичной схеме вместо этого используется прогноз для вычисления начальных значений векторов $\mathbf{F}_i^0$ и $\mathbf{G}_i^0$. Предполагая также, что на всех стадиях алгебраическое соотношение $\mathbf{0} = \mathbf{G}_i$ выполняется достаточно точно, получаем:

$$\mathbf{F}_i^0 = \sum_{j=1}^{s-1} \alpha_{ij}\bar{\mathbf{F}}_j + \sum_{j=1}^{i-1} \beta_{ij}\mathbf{F}_j, \quad \mathbf{G}_i^0 = \mathbf{0}. \tag{5.68}$$

Таким образом, экономичная схема задается формулами (5.65), (5.68), (5.67).

Систему ДАУ часто задают в виде

$$\mathbf{M}\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{5.69}$$

где квадратная матрица $\mathbf{M}$ имеет неполный ранг. При решении таких задач итерации (5.64) запишутся в виде

$$\Delta\mathbf{Y}_i^k = \Delta\mathbf{Y}_i^{k-1} + (\mathbf{M} - h\gamma\mathbf{f}_\mathbf{y})^{-1}\left(h\sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j + h\gamma\mathbf{F}_i^{k-1} - \mathbf{M}\Delta\mathbf{Y}_i^{k-1}\right),$$

а вместо (5.67) используется формула

$$\mathbf{F}_i = \frac{1}{\gamma}\left(\frac{\mathbf{M}\triangle\mathbf{Y}_i}{h} - \sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j\right).$$

В (5.64) итерации выполняются относительно переменных. Но можно выполнять итерации и относительно производных, тогда они имеют вид:

$$\begin{bmatrix}\mathbf{F}_i^k\\(\mathbf{Z}_i')^k\end{bmatrix} = \begin{bmatrix}\mathbf{F}_i^{k-1}\\(\mathbf{Z}_i')^{k-1}\end{bmatrix} + \begin{bmatrix}\mathbf{I}-h\gamma\mathbf{f}_{\mathbf{y}} & -h\gamma\mathbf{f}_{\mathbf{z}}\\ -h\gamma\mathbf{g}_{\mathbf{y}} & -h\gamma\mathbf{g}_{\mathbf{z}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{f}(\mathbf{Y}_i^{k-1},\mathbf{Z}_i^{k-1}) - \mathbf{F}_i^{k-1}\\ \mathbf{g}(\mathbf{Y}_i^{k-1},\mathbf{Z}_i^{k-1})\end{bmatrix},$$

$$\mathbf{Y}_i^k = \mathbf{y}_n + h\sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j + h\gamma\mathbf{F}_i^k, \quad \mathbf{Z}_i^k = \mathbf{z}_n + h\sum_{j=1}^{i-1} a_{ij}\mathbf{Z}_j' + h\gamma(\mathbf{Z}_i')^k,$$

а финальные значения производных получим в виде $\mathbf{F}_i = \mathbf{F}_i^N$, $\mathbf{Z}_i' = (\mathbf{Z}_i')^N$. Такие итерации особенно удобны при использовании схемы D, а если при этом ограничиться одной итерацией, то получаем схемы типа Розенброка.

Сходимость итерационных схем решения задачи Коши исследовалась в [27, 115]. При определенных условиях каждая итерация с неточной матрицей Якоби повышает порядок схемы на 1 до тех пор, пока не будет достигнут порядок метода. Таким образом, при реализации метода порядка p с помощью тривиальной схемы необходимо выполнить не менее p итераций. Для достижения заданного порядка при использовании прогноза порядка r необходимо выполнить не менее $p - r$ итераций.

Остановимся на формулах прогноза. Для методов ESDIRK прогноз 2-го порядка формируется как значение интерполяционного многочлена, построенного по уже вычисленным трем стадийным значениям. Если использовать $\bar{\mathbf{Y}}_i$, $\bar{\mathbf{Y}}_j$ и $\mathbf{Y}_1$, то многочлен Лагранжа $\mathbf{L}(x)$ строим, исходя из условий:

$\mathbf{L}(c_i - 1) = \bar{\mathbf{Y}}_i$, $\mathbf{L}(c_j - 1) = \bar{\mathbf{Y}}_j$, $\mathbf{L}(0) = \mathbf{Y}_1$,

откуда получаем прогноз на 2-й стадии как $\mathbf{Y}_2^0 = \mathbf{L}(wc_2)$, $w = h/\bar{h}$, где $\bar{h}$ – размер предыдущего шага. В результате имеем:

$$\mathbf{Y}_2^0 = \alpha_{2i}\bar{\mathbf{Y}}_i + \alpha_{2j}\bar{\mathbf{Y}}_j + \beta_{21}\mathbf{Y}_1,$$

$$\alpha_{2i} = \frac{(wc_2 - c_j + 1)wc_2}{(c_i - c_j)(c_i - 1)}, \quad \alpha_{2j} = \frac{(wc_2 - c_i + 1)wc_2}{(c_j - c_i)(c_j - 1)}, \quad \beta_{21} = 1 - \alpha_{2i} - \alpha_{2j}.$$

На 3-й стадии, используя $\bar{\mathbf{Y}}_j$, $\mathbf{Y}_1$, $\mathbf{Y}_2$ и действуя аналогично, получаем:

$$\mathbf{Y}_3^0 = \alpha_{3j}\bar{\mathbf{Y}}_j + \beta_{31}\mathbf{Y}_1 + \beta_{32}\mathbf{Y}_2,$$

$$\beta_{31} = \frac{c_3 - c_2}{c_2}\left(\frac{wc_3}{c_j - 1} - 1\right), \quad \beta_{32} = \frac{c_3(wc_3 - c_j + 1)}{c_2(wc_2 - c_j + 1)}, \quad \alpha_{3j} = 1 - \beta_{31} - \beta_{32}.$$

На 4-й стадии можно использовать стадийные значения только текущего шага, тогда получаем:

$\mathbf{Y}_4^0 = \beta_{41}\mathbf{Y}_1 + \beta_{42}\mathbf{Y}_2 + \beta_{43}\mathbf{Y}_3$,

$$\beta_{42} = \frac{c_4(c_4 - c_3)}{c_2(c_2 - c_3)}, \quad \beta_{43} = \frac{c_4(c_4 - c_2)}{c_3(c_3 - c_2)}, \quad \beta_{41} = 1 - \beta_{42} - \beta_{43}.$$

На 5-й стадии имеет смысл формировать прогноз в виде:

$$\mathbf{Y}_5^0 = \beta_{51}\mathbf{Y}_1 + \beta_{52}\mathbf{Y}_2 + \beta_{53}\mathbf{Y}_3 + \beta_{54}\mathbf{Y}_4,$$

при этом принимаем $\beta_{51} = 1 - \beta_{52} - \beta_{53} - \beta_{54}$. Для обеспечения 3-го порядка прогноза должны выполняться условия:

$$\beta_{52}c_2 + \beta_{53}c_3 + \beta_{54}c_4 = c_5,$$
$$\beta_{52}c_2^2 + \beta_{53}c_3^2 + \beta_{54}c_4^2 = c_5^2,$$
$$\beta_{53}a_{32}c_2^2 + \beta_{54}(a_{42}c_2^2 + a_{43}c_3^2) = a_{52}c_2^2 + a_{53}c_3^2 + a_{54}c_4^2,$$

из которых нетрудно получить коэффициенты β_{52}, β_{53}, β_{54}.

Оценим эффективность рассмотренных схем на примере задачи

$$y_1' = -102y_1 + 100y_2^2, \quad y_2' = y_1 - y_2(1 + z), \quad 0 = y_2 - z + 0.1(y_1 - z^2),$$
$$y_1(0) = y_2(0) = z(0) = 1, \quad 0 \le t \le 1, \tag{5.70}$$

точное решение которой: $y_1(t) = \exp(-2t)$, $y_2(t) = z(t) = \exp(-t)$. Будем использовать метод 4-го порядка (5.32). Нетривиальный прогноз строим по трем предыдущим значениям, а на последней стадии – по четырем значениям. Обозначим через *nf* число вычислений правой части на каждой неявной стадии, а в качестве показателя точности примем евклидову норму ошибки в конце интервала. Ошибки рассмотренных схем в зависимости от *nf* при $h = 0.1$ и одном вычислении матрицы Якоби на всем интервале приведены в табл. 5.5.

Таблица 5.5. Ошибки решения задачи (5.70)

nf	Схема			
	T	S	D	E
1	1.09×10^{-2}	2.04×10^{-5}	5.17×10^{-3}	2.84×10^{-7}
2	2.18×10^{-4}	4.94×10^{-7}	9.57×10^{-7}	3.50×10^{-7}
3	2.15×10^{-5}	3.59×10^{-7}	3.27×10^{-7}	3.53×10^{-7}
4	2.33×10^{-6}	3.54×10^{-7}	3.52×10^{-7}	3.53×10^{-7}
5	4.50×10^{-7}	3.53×10^{-7}	3.53×10^{-7}	3.53×10^{-7}

Прерывать итерации следует в тот момент, когда следующая итерация уже не приводит к заметному уменьшению ошибки. Исходя из этого, схема E обеспечила приемлемую сходимость при одном вычислении правой части на каждой неявной стадии, тогда как схеме S понадобились два вычисления, а схеме D – три вычисления.

5.11. Реализация методов DESI

Для удобства вывода расчетных формул представим систему ДАУ в виде (5.69). Рассмотрим один шаг метода DESI (5.55). Первая стадия совпадает с последней

стадией предыдущего шага и задается формулами $\mathbf{Y}_1 = \mathbf{y}_n$, $\mathbf{F}_1 = \mathbf{f}_n$. Следующие $r - 1$ стадии выполняем путем итераций:

$$\begin{aligned}&(\mathbf{I}\otimes\mathbf{M} - h\bar{\mathbf{A}}\otimes\mathbf{J})\delta\bar{\mathbf{Y}}^k = \mathbf{V}^k,\\ &\Delta\bar{\mathbf{Y}}^k = \Delta\bar{\mathbf{Y}}^{k-1} + \delta\bar{\mathbf{Y}}^k, \quad \bar{\mathbf{Y}}^k = \mathbf{e}\otimes\mathbf{y}_n + \Delta\bar{\mathbf{Y}}^k,\end{aligned} \tag{5.71}$$

где $\bar{\mathbf{A}}$ – матрица (5.56), $\mathbf{J}$ – замороженная матрица Якоби $\partial\mathbf{f}(t, \mathbf{y})/\partial\mathbf{y}$,

$$\bar{\mathbf{Y}}^k = \begin{bmatrix}\mathbf{Y}_2^k\\ \vdots\\ \mathbf{Y}_r^k\end{bmatrix}, \quad \mathbf{V}^k = \begin{bmatrix}\mathbf{V}_2^k\\ \vdots\\ \mathbf{V}_r^k\end{bmatrix}, \quad \mathbf{V}_i^k = h\left[a_{i1}\mathbf{F}_1 + \sum_{j=2}^{r} a_{ij}\mathbf{f}(t_n + c_j h, \mathbf{Y}_j^{k-1})\right] - \mathbf{M}\Delta\mathbf{Y}_i^{k-1}.$$

Чтобы упростить решение линейной системы в (5.71), представим матрицу $\bar{\mathbf{A}}$ в виде $\bar{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{\Lambda}\mathbf{T}$, где матрица $\mathbf{\Lambda}$ имеет нижнюю двухдиагональную форму. Примем $\mathbf{X} = (\mathbf{T}\otimes\mathbf{I})\delta\mathbf{Y}^k$, $\mathbf{W} = (\mathbf{T}\otimes\mathbf{I})\,\mathbf{V}^k$, тогда эта система приводится к виду

$$(\mathbf{I}\otimes\mathbf{M} - h\mathbf{\Lambda}\otimes\mathbf{J})\mathbf{X} = \mathbf{W}. \tag{5.72}$$

Решение системы (5.72) сводится к последовательному решению $r - 1$ систем меньшей размерности, имеющих одинаковую матрицу коэффициентов. Приращение на k-й итерации получаем в виде $\delta\bar{\mathbf{Y}}^k = (\mathbf{T}^{-1}\otimes\mathbf{I})\mathbf{X}$.

Например, для метода (5.61) имеем:

$$\mathbf{T} = \begin{bmatrix}-27/8 & 1 & -1/8\\ 27/8 & 0 & 3/8\\ 0 & 0 & -3/4\end{bmatrix}, \quad \mathbf{T}^{-1} = \begin{bmatrix}0 & 8/27 & 4/27\\ 1 & 1 & 1/3\\ 0 & 0 & -4/3\end{bmatrix}, \quad \mathbf{\Lambda} = \gamma\begin{bmatrix}1 & 0 & 0\\ -1 & 1 & 0\\ 0 & -1 & 1\end{bmatrix}.$$

В этом случае система (5.72) запишется в виде

$$\begin{bmatrix}\mathbf{M} - h\gamma\mathbf{J} & 0 & 0\\ h\gamma\mathbf{J} & \mathbf{M} - h\gamma\mathbf{J} & 0\\ 0 & h\gamma\mathbf{J} & \mathbf{M} - h\gamma\mathbf{J}\end{bmatrix}\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\\ \mathbf{X}_3\end{bmatrix} = \begin{bmatrix}\mathbf{W}_1\\ \mathbf{W}_2\\ \mathbf{W}_3\end{bmatrix} = \mathbf{W}$$

и распадается на 3 системы:

$(\mathbf{M} - h\gamma\mathbf{J})\mathbf{X}_1 = \mathbf{W}_1$,

$(\mathbf{M} - h\gamma\mathbf{J})\mathbf{X}_2 = \mathbf{W}_1 + \mathbf{W}_2 - \mathbf{M}\mathbf{X}_1$,

$(\mathbf{M} - h\gamma\mathbf{J})\mathbf{X}_3 = \mathbf{W}_1 + \mathbf{W}_2 + \mathbf{W}_3 - \mathbf{M}\mathbf{X}_1 - \mathbf{M}\mathbf{X}_2$.

В результате выполнения итераций получим приращения стадийных значений $\Delta\mathbf{Y}_2, \ldots, \Delta\mathbf{Y}_r$. Значения $\mathbf{F}_2, \ldots, \mathbf{F}_r$ находим через эти приращения, т. е. решая уравнения

$$h\sum_{j=1}^{r} a_{ij}\mathbf{F}_j = \mathbf{M}\Delta\mathbf{Y}_i, \quad i = 2, \ldots, r.$$

Вычисление $\Delta\mathbf{Y}_i$ и $\mathbf{F}_i$ для $i > r$ выполняем точно так же, как и в методах ESDIRK.

5.12. Изменение размера шага и обновление матрицы Якоби

Управление размером шага осуществляется на основе оценки локальной ошибки, в качестве которой используется норма вектора $\delta\mathbf{y} = K(\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1})$. Мы принимаем $\hat{\mathbf{y}}_{n+1} = \mathbf{Y}_s^0$ (прогноз последней стадии), а коэффициент K настраиваем по результатам тестовых расчетов таким образом, чтобы ошибка решения примерно соответствовала задаваемому допуску. Например, задаем $K = 1$ для метода (5.31), $K = 1/2$ для методов (5.32), (5.34), (5.35) и $K = 1/4$ для методов (5.36), (5.61). В общем случае вектор $\delta\mathbf{y}$ содержит оценки ошибок всех переменных. Но для ДАУ высших индексов такая процедура может привести к аварийной остановке численного решения. Чтобы этого не происходило, для методов, имеющих $\hat{R}(\infty) \neq 0$, исключаем из $\delta\mathbf{y}$ оценки ошибок переменных индексов 2 и 3, а для методов, имеющих $\hat{R}(\infty) = 0$, исключаем оценки ошибок переменных индексов 3 (см. табл. 5.3). В отличие от других методов, метод (5.31) обеспечивает устойчивый контроль ошибок всех переменных при решении ДАУ индексов 2 и 3.

В наших программах нормированную оценку ошибки принимаем в виде $\delta = \mathrm{err}(\delta\mathbf{y})$, где

$$\mathrm{err}(\delta\mathbf{y}) = \max_j \left(\frac{\left|\delta y_j\right|}{Atol + Rtol \times \max\left(\left|y_{nj}\right|, \left|y_{n+1,j}\right|\right)} \right), \tag{5.73}$$

Atol – допустимая абсолютная ошибка, *Rtol* – допустимая относительная ошибка, а δy_j, y_{nj} и $y_{n+1,j}$ – j-е компоненты соответствующих векторов. Если $\delta \leq 1$, то шаг считается успешным, в противном случае он отбрасывается. Размер нового шага рассчитываем, используя стандартную процедуру, т. е. в виде

$$h_{\mathrm{new}} = wh, \quad w = \max(w_{\min}, \min(w_{\max}, fac \times \delta^{-1/p})),$$

где p – порядок метода. Мы задаем $w_{\min} = 1/8$, $w_{\max} = 8$, а множитель $fac \approx 0.3^{1/p}$ (для методов 4-го порядка принимаем $fac = 0.75$). Если $|1 - w| < 0.1$, то оставляем старое значение h, что позволяет сократить число LU-разложений матрицы.

Для оценивания ошибки итераций также используем формулу (5.73), в которой $y_{n+1,j}$ заменяем на соответствующие компоненты вектора $\mathbf{Y}_i$. В процессе итераций вычисляем значения $\delta_k = \mathrm{err}(\delta\mathbf{Y}_i^k)$, $\theta_k = \delta_k/\delta_{k-1}$, $\varepsilon_k = \delta_k\theta_k/(1 - \theta_k)$, где $\delta\mathbf{Y}_i^k$ – приращение на последней итерации. Величина θ_k оценивает скорость сходимости итераций, а нормированная ошибка последней итерации оценивается величиной ε_k. Итерации прекращаем, если они расходятся ($\theta_k \geq 1$) либо если $\varepsilon_k \leq \varepsilon_{\max}$, где $\varepsilon_{\max} = 0.01\ldots0.1$. Пересчет матрицы Якоби выполняется на основе оценки сходимости на последней стадии и возможен только после успешного шага. Матрицу Якоби пересчитываем, если $\theta_k > \theta_{\max}$, $\theta_{\max} = 0.03\ldots0.3$ либо если было выполнено максимально допустимое число итераций и при этом $\varepsilon_k > \varepsilon_{\max}$.

5.13. Численные эксперименты

Среди диагонально-неявных методов наиболее популярны методы 4-го порядка, сочетающие удобную реализацию и высокую эффективность при умеренных требованиях к точности. Такие методы представлены в табл. 5.3. Мы реализовали эти методы с автоматическим выбором шага и использовали их для решения ряда тестовых задач. При решении жестких задач все эти методы показывают примерно одинаковые результаты, но немного более эффективным почти для всех задач оказался метод (5.32). При решении ДАУ индексов 2 и 3 преимущество имеют методы 3-го порядка (5.30), (5.31) и методы 4-го порядка (5.36) и (5.61), параметры которых удовлетворяют дополнительным условиям порядка для таких задач. Ниже приведены результаты решения тестовых задач перечисленными методами. Чтобы удобнее было различать эти методы, примем для них следующие обозначения: (5.30) – ESDIRK53, (5.31) – ESDIRK73, (5.32) – ESDIRK54, (5.36) – ESDIRK64, (5.61) – DESI64 (1-я цифра в названии – число стадий, 2-я – порядок).

Нас интересует сходимость этих методов при решении ДАУ высших индексов, для чего были проведены расчеты с постоянным размером шага. Результаты решения ДАУ индекса 2 (4.39) и индекса 3 (4.42) приведены в табл. 5.6 и 5.7. Для сравнения приводим результаты метода Радо IIA 5-го порядка, обозначенного как Radau5. Ошибки по соответствующим компонентам получены при $h = 1/30$, а оценки порядка рассчитаны по формуле (4.15) при $h_1 = 1/30$, $h_2 = 1/60$. Использование дополнительных условий порядка для ДАУ индексов 2 и 3 в методах ESDIRK53, ESDIRK73, ESDIRK64 и DESI64 действительно позволило повысить порядки сходимости соответствующих компонент.

Таблица 5.6. Результаты решения системы ДАУ индекса 2

Метод	e_y	e_z	$\tilde{p}_y$	$\tilde{p}_z$
ESDIRK53	3.90×10^{-8}	3.62×10^{-6}	3.03	3.00
ESDIRK73	6.94×10^{-8}	1.48×10^{-6}	3.02	3.00
ESDIRK54	2.97×10^{-8}	2.51×10^{-5}	3.05	1.93
ESDIRK64	2.07×10^{-10}	3.18×10^{-6}	4.07	2.99
DESI64	8.13×10^{-10}	4.86×10^{-9}	4.00	4.01
Radau5	2.01×10^{-11}	2.57×10^{-6}	5.00	2.99

Таблица 5.7. Результаты решения системы ДАУ индекса 3

Метод	e_y	e_z	e_u	$\tilde{p}_y$	$\tilde{p}_z$	$\tilde{p}_u$
ESDIRK53	9.48×10^{-8}	2.25×10^{-6}	2.38×10^{-4}	2.98	2.98	2.05
ESDIRK73	2.75×10^{-7}	1.22×10^{-6}	2.19×10^{-4}	3.13	3.00	2.00
ESDIRK54	1.45×10^{-6}	1.80×10^{-5}	8.73×10^{-3}	2.05	1.92	0.99
ESDIRK64	2.80×10^{-8}	1.73×10^{-6}	4.19×10^{-4}	2.84	2.97	1.99
DESI64	2.07×10^{-10}	6.12×10^{-9}	1.56×10^{-6}	3.95	3.98	2.99
Radau5	3.83×10^{-10}	1.83×10^{-6}	2.25×10^{-4}	4.05	2.98	1.99

Для решения с переменным размером шага были выбраны 8 жестких тестов из [75]. Четыре из них приведены в разделе 3.6, а более подробные описания всех задач даны в [75, 128]. Некоторые характеристики всех восьми задач приведены в табл. 1.3. Использовались решатели, реализующие методы ESDIRK54, ESDIRK64 и DESI64, которые показали примерно одинаковые результаты. Поэтому приводим только результаты решателя ESDIRK54, который для большинства задач имел небольшое преимущество. Для сравнения приводим взятые из [85] результаты решателя RADAU5, признанного одним из лучших для решения жестких задач и ДАУ.

Для всех задач принимаем $Rtol = 10^{-4}$. Как и в [85], для задачи ROBER задаем $Atol = 10^{-4} \times Rtol$, а для остальных задач – $Atol = Rtol$. Начальный размер шага принимаем $h_0 = 10^{-2} \times Rtol$ для VDPOL и ROBER и $h_0 = Rtol$ для остальных задач. Точность решения оцениваем значениями

$$scd = -\lg\left(\max_i\left(\frac{|y_i - \tilde{y}_i|}{|y_i|}\right)\right), \quad mescd = -\lg\left(\max_i\left(\frac{|y_i - \tilde{y}_i|}{Atol/Rtol + |y_i|}\right)\right),$$

где y_i – точное, а $\tilde{y}_i$ – численное решение по i-й компоненте в конце интервала интегрирования. Отдельные задачи (например, ROBER) имеют очень малые значения некоторых переменных. В таком случае следовало бы также задавать и очень малое значение *Atol*. Поэтому значение *mescd* для таких задач лучше отражает соответствие полученной ошибки задаваемому допуску, чем *scd*. Вычислительные затраты оцениваем числом вычислений правой части *Nf*, числом вычислений матрицы Якоби *NJ* и числом LU-разложений *NLU*.

Полученные результаты приведены в табл. 5.8. При примерно равных затратах на вычисление правой части наш решатель требует заметно меньших вычислений матрицы Якоби. Заметим также, что одно LU-разложение в RADAU5 по вычислительным затратам соответствует примерно пяти LU-разложениям в методах DIRK и DESI [75].

Таблица 5.8. Результаты решения жестких задач

Задача	Решатель	*scd*	*mescd*	*Nf*	*NJ*	*NLU*
VDPOL	ESDIRK54	4.09	4.42	1766	26	222
	RADAU5	4.96	5.28	2253	162	252
ROBER	ESDIRK54	2.13	5.81	736	15	113
	RADAU5	3.06	6.74	811	108	113
OREGO	ESDIRK54	3.53	3.54	2216	60	287
	RADAU5	4.22	4.22	2702	238	290
HIRES	ESDIRK54	2.95	5.16	176	12	35
	RADAU5	0.75	2.96	295	20	36
PLATE	ESDIRK54	3.50	5.39	211	1	18
	RADAU5	1.62	3.77	74	4	15
BEAM	ESDIRK54	2.75	3.22	566	1	49
	RADAU5	2.49	3.57	406	43	60

Окончание табл. 5.8

Задача	Решатель	*scd*	*mescd*	*Nf*	*NJ*	*NLU*
CUSP	ESDIRK54	3.80	4.58	806	20	131
	RADAU5	3.10	3.94	881	70	101
BRUSS	ESDIRK54	4.30	4.40	246	3	40
	RADAU5	4.72	—	235	27	34

Рассмотрим также систему ДАУ индекса 3 «Pendulum», описывающую колебания маятника в декартовой системе координат [75]:

$$y_1' = z_1,\ \ y_2' = z_2,\ \ z'_1 = -y_1 u,\ \ z'_2 = -y_2 u - 1,\ \ 0 = y_1^2 + y_2^2 - 1,$$
$$y_1(0) = z_2(0) = u(0) = 1,\ \ y_2(0) = z_1(0) = 0,\ \ 0 \le t \le 1. \tag{5.74}$$

Принимаем $Tol = Rtol = Atol = h_0$. Результаты приведены в табл. 5.9. Некоторое преимущество метода DESI64, по сравнению с методами ESDIRK, объясняется более высоким стадийным порядком и выполнением дополнительных условий порядка для ДАУ индексов 2 и 3.

Чтобы получить решение, пришлось исключить контроль ошибок переменных z_i, u в методах ESDIRK54 и DESI64, имеющих $\hat{R}(\infty) \neq 0$, и переменной u в методе ESDIRK64. Как и ожидалось, только метод ESDIRK73 обеспечил решение при контроле ошибки по всем переменным. Метод ESDIRK73 показал также лучшее соответствие ошибки решения задаваемому допуску по сравнению с другими методами.

Таблица 5.9. Результаты решения задачи Pendulum

Решатель	*Tol*	*scd*	*mescd*	*Nf*	*NJ*	*NLU*
ESDIRK73	10^{-3}	1.85	2.33	97	6	10
	10^{-4}	2.81	3.29	206	12	20
	10^{-6}	4.41	5.06	1882	17	99
ESDIRK54	10^{-3}	0.79	1.90	43	5	7
	10^{-4}	1.19	2.40	66	7	12
	10^{-6}	2.10	3.13	217	9	20
ESDIRK64	10^{-3}	1.24	1.92	45	5	7
	10^{-4}	1.35	1.83	76	7	11
	10^{-6}	2.80	3.56	253	9	21
DESI64	10^{-3}	1.67	2.52	52	5	7
	10^{-4}	2.36	2.84	73	7	9
	10^{-6}	3.44	4.29	145	14	17
RADAU5	10^{-3}	2.04	2.94	82	7	11
	10^{-6}	3.21	3.69	166	16	24

Неявные методы повышенной точности для жестких задач и ДАУ

ГЛАВА 6

6.1. Коллокационные методы Рунге–Кутты для жестких задач

Зададим узлы $c_1, \ldots, c_s$, $c_i \neq c_j$ неявного s-стадийного метода Рунге–Кутты и потребуем, чтобы он имел стадийный порядок $q = s$. Тогда коэффициенты метода определяются из уравнений

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k}, \quad \sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k}, \quad i,k = 1, \ldots, s,$$

откуда получаем

$$\mathbf{A} = \begin{bmatrix} c_1 & c_1^2/2 & \cdots & c_1^s/s \\ c_2 & c_2^2/2 & \ldots & c_2^s/s \\ \vdots & \vdots & & \vdots \\ c_s & c_s^2/2 & \cdots & c_s^s/s \end{bmatrix} \mathbf{V}^{-1}, \quad \mathbf{b}^{\mathrm{T}} = \begin{bmatrix} 1 \\ 1/2 \\ \vdots \\ 1/s \end{bmatrix}^{\mathrm{T}} \mathbf{V}^{-1}, \quad \mathbf{V} = \begin{bmatrix} 1 & c_1 & \cdots & c_1^{s-1} \\ 1 & c_2 & \cdots & c_2^{s-1} \\ \vdots & \vdots & & \vdots \\ 1 & c_s & \cdots & c_s^{s-1} \end{bmatrix}.$$

Такие методы называют *коллокационными*, они имеют максимально возможный стадийный порядок при заданном числе стадий и однозначно определяются по своим узлам.

К наиболее известным и имеющим практическое значение коллокационным методам относятся A-устойчивые методы Гаусса и Лобатто IIIA, а также L-устойчивые методы Радо IIA [75]. Эти методы основаны на соответствующих квадратурных формулах, узлами которых являются нули многочленов:

- $\dfrac{d^s}{dx^s}\left(x^s(x-1)^s\right)$ для методов Гаусса;
- $\dfrac{d^{s-2}}{dx^{s-2}}\left(x^{s-1}(x-1)^{s-1}\right)$ для методов Лобатто IIIA;
- $\dfrac{d^{s-1}}{dx^{s-1}}\left(x^{s-1}(x-1)^{s}\right)$ для методов Радо IIA.

Методы Гаусса нежесткоточные, поэтому плохо подходят для решения жестких задач (но они являются симплектическими, что обеспечивает высо-

кую точность при интегрировании гамильтоновых систем [75]). Жесткоточные методы Лобатто IIIA и Радо IIA успешно применяются для эффективного решения жестких задач [36, 75, 104]. Методы Лобатто IIIA имеют явную первую стадию и более точные, поскольку при одинаковых вычислительных затратах, определяемых числом неявных стадий, их классический и стадийный порядки на 1 выше, чем у Радо IIA. Но методы Радо IIA L-устойчивые, тогда как Лобатто IIIA только A-устойчивые и имеют $|R(\infty)| = 1$, что не позволяет эффективно решать некоторые жесткие задачи и ДАУ высших индексов.

Рассмотрим жесткоточные коллокационные методы с явной первой стадией, тогда $c_1 = 0$, $c_s = 1$. Примем один из узлов в качестве свободного параметра, а остальные зададим из условий максимального порядка. В результате получим методы, имеющие $p = 2s - 1$, которые могут быть более устойчивыми, чем Лобатто IIIA, и более точными, чем Радо IIA. Такие методы исследовались в [106–108], где они получили название SAFERK (Stiffly Accurate First Explicit Runge–Kutta). При $c^*_{s-1} < c_{s-1} < 1$, где c^*_{s-1} – значение c_{s-1} для метода Лобатто IIIA, имеем $0 < |R(\infty)| < 1$. На ряде жестких задач в [106, 108] было показано, что эти методы могут быть более эффективными, чем методы Радо IIA и Лобатто IIIA. Основные показатели рассмотренных типов методов в зависимости от числа неявных стадий r приведены в табл. 6.1.

Таблица 6.1. Коллокационные методы (r – число неявных стадий)

Метод	$q = s$	p	$R(\infty)$
Лобатто IIIA	$r + 1$	$2r$	$(-1)^r$
Радо IIA	r	$2r - 1$	0
SAFERK	$r + 1$	$2r - 1$	$0 < \|R(\infty)\| < 1$

В качестве свободного параметра примем $\alpha = 1 - c_{s-1}$ (в [106, 108] свободный параметр $\beta = c_{s-1}$). Рассмотрим методы SAFERK 5-го порядка. Они имеют $s = 4$ и узлы

$$c_1 = 0, \quad c_2 = \frac{2 - 5\alpha}{5 - 10\alpha}, \quad c_3 = 1 - \alpha, \quad c_4 = 1, \quad 0 < \alpha < \alpha^* = \frac{5 - \sqrt{2}}{10} = 0.276...$$

Для этих методов $R(\infty) = \dfrac{\alpha(3 - 5\alpha)}{(1 - \alpha)(5\alpha - 2)}$, а коэффициенты погрешности 6-го порядка (всего их 20) выражаются формулами

$$e(T_{61,\,\ldots,\,611}) = \frac{(5\alpha^2 - 5\alpha + 1)}{50(1 - 2\alpha)}, \quad e(T_{612,\,\ldots,\,620}) = -5e(T_{61}).$$

При $\dfrac{4 - \sqrt{6}}{10} = 0.155... < \alpha < \alpha^*$ все коэффициенты погрешности метода SAFERK меньше соответствующих коэффициентов метода Радо IIA. В табл. 6.2

приведены характеристики методов Лобатто IIIA, Радо IIA и SAFERK, имеющих 3 неявные стадии. При $q > 1$ число различных значений коэффициентов погрешности 6-го порядка меньше 20, все они приведены в последней графе табл. 6.2.

Таблица 6.2. Характеристики методов при $r = 3$

Метод	α	p	q	$R(\infty)$	$e(T_{6i})$
Лобатто IIIA	0.2764	6	4	−1	0
SAFERK	0.25	5	4	−0.778	0.0025; −0.0125
SAFERK	0.2	5	4	−0.5	0.0067; −0.0333
SAFERK	0.1	5	4	−0.185	0.0138; −0.0688
Радо IIA	—	5	3	0	−0.01; 0.02; 0.05; −0.1

Методы из табл. 6.2 были реализованы с автоматическим выбором размера шага. Чтобы сравнение решателей было корректным, мы использовали одинаковые схемы реализации. Результаты решения трех задач приведены в табл. 6.3, где при задаваемом допуске на ошибку *Tol* точность оценивается значением *scd* (3.18). Из приведенных результатов видно, что методы SAFERK могут иметь преимущество при решении жестких задач, которое объясняется более высоким стадийным порядком, по сравнению с Радо IIA, и меньшим значением $|R(\infty)|$, по сравнению с Лобатто IIIA.

Таблица 6.3. Результаты решения жестких задач

Задача	Метод	$Tol = 10^{-4}$			$Tol = 10^{-7}$		
		scd	*Nf*	*NJ*	*scd*	*Nf*	*NJ*
VDPOL	Лобатто IIIA	3.70	1984	74	7.04	6988	124
	SAFERK(α=0.25)	3.85	2014	73	7.31	6811	118
	SAFERK(α=0.2)	5.11	2020	75	7.28	6568	120
	SAFERK(α=0.1)	4.52	2143	75	6.92	6046	119
	Радо IIA	4.26	2044	75	7.52	7270	124
ROBER	Лобатто IIIA	−0.18	3154	38	2.63	9964	37
	SAFERK(α=0.25)	2.41	577	44	5.78	2257	44
	SAFERK(α=0.2)	3.82	559	45	5.80	2209	43
	SAFERK(α=0.1)	3.07	592	43	5.74	2125	43
	Радо IIA	2.79	667	42	6.09	2614	45
PLATE	Лобатто IIIA	3.08	247	1	7.56	946	1
	SAFERK(α=0.25)	3.36	259	1	8.15	931	1
	SAFERK(α=0.2)	3.93	223	1	8.20	859	1
	SAFERK(α=0.1)	4.16	223	1	7.21	829	1
	Радо IIA	4.07	298	1	7.89	1540	1

6.2. Коллокационные методы Рунге–Кутты для ДАУ индексов 2 и 3

В [64] были исследованы методы SAFERK с целью найти оптимальное значение c_{s-1} для ДАУ высших индексов. Оказалось, что для таких уравнений преимущество имеют методы с близким к 1 значением c_{s-1} (для жестких задач в [106, 108] использовалось значение, близкое к c^*_{s-1}). В результате более детального исследования были выделены две группы коллокационных методов, имеющих преимущество при решении ДАУ индексов 2 и 3. При заданном s каждая из этих групп образует однопараметрическое семейство со свободным параметром α.

Методы первой группы имеют

$$c_1 = 0, \quad c_{s-1} = 1 - \alpha, \quad c_s = 1, \tag{6.1}$$

а остальные узлы задаем из условия обеспечения порядка $2s - 3$. Эти методы рекомендуется использовать для решения ДАУ индекса 2. Методы второй группы имеют

$$c_1 = 0, \quad c_{s-2} = 1 - \alpha, \quad c_{s-1} = 1 + \alpha, \quad c_s = 1, \tag{6.2}$$

а остальные узлы задаем из условия обеспечения порядка $2s - 4$. Эти методы рекомендуются для ДАУ индекса 3. Численные эксперименты показали, что при небольших значениях α предложенные методы превосходят известные методы Рунге–Кутты при решении ДАУ индексов 2 и 3.

Отметим, что при малом α расположение узлов (6.1) позволяет получить хорошую оценку второй производной решения на последней стадии, а расположение узлов (6.2) позволяет получить также и оценку третьей производной. В пределе при $\alpha \to 0$ получаем кратный узел, что соответствует использованию старших производных (коллокационные методы со старшими производными рассматривались в [5, 29, 74]). Ниже будет показано, что такие методы усиливают свойство жесткой точности применительно к уравнениям высших индексов. Практически это означает отсутствие снижения порядка при решении ДАУ индексов 2 и 3.

При численном решении ДАУ высших индексов самыми неточными являются алгебраические переменные: z-компонента в ДАУ индекса 2 (4.5а, б) и u-компонента в ДАУ индекса 3 (4.6). Исследуем точность этих компонент на примере системы

$$0 = y - \varphi(t), \quad y_0 = \varphi(t_0), \tag{6.3а}$$

$$y' = z, \quad z_0 = \varphi'(t_0), \tag{6.3б}$$

$$z' = u, \quad u_0 = \varphi''(t_0). \tag{6.3в}$$

Эти уравнения задают последовательное дифференцирование функции $\varphi(t)$. Уравнение (6.3а) имеет индекс 1 и содержит одну алгебраическую переменную y. Добавив уравнение (6.3б), получим систему индекса 2, в которой y становится дифференциальной переменной, а z является алгебраической переменной.

Добавив также уравнение (6.3в), получим систему индекса 3 с дифференциальными переменными y, z и алгебраической переменной u.

Глобальные ошибки решения уравнений (6.3) запишутся в виде:

$$\begin{aligned}
&\varphi_{n+1} - y_{n+1} = a_0(\varphi_n - y_n) + \delta y_{n+1},\\
&\varphi'_{n+1} - z_{n+1} = a_0(\varphi'_n - z_n) + h^{-1}a_1(\varphi_n - y_n) + \delta z_{n+1},\\
&\varphi''_{n+1} - u_{n+1} = a_0(\varphi''_n - u_n) + h^{-1}a_1(\varphi'_n - z_n) + h^{-2}a_2(\varphi_n - y_n) + \delta u_{n+1},
\end{aligned} \tag{6.4}$$

где для методов вида (4.4) с явной первой стадией имеем:

$$\begin{aligned}
&a_0 = R(\infty) = 1 - \tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}, \quad a_1 = \lim_{z\to\infty} z(R(z) - a_0) = -\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-3}\tilde{\mathbf{c}},\\
&a_2 = \lim_{z\to\infty} z\left[z(R(z) - a_0) - a_1\right] = -\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-4}\tilde{\mathbf{c}}.
\end{aligned} \tag{6.5}$$

Локальные ошибки можно представить в виде:

$$\begin{aligned}
&\delta y_{n+1} = \sum_{i=q+1}^{\infty} e_{yi}\varphi_n^{(i)}\frac{h^i}{i!}, \quad \delta z_{n+1} = \sum_{i=q+1}^{\infty} e_{zi}\varphi_n^{(i)}\frac{h^{i-1}}{(i-1)!},\\
&\delta u_{n+1} = \sum_{i=q+1}^{\infty} e_{ui}\varphi_n^{(i)}\frac{h^{i-2}}{(i-2)!}, \quad \varphi_n^{(i)} = \left.\frac{d^i\varphi(t)}{dt^i}\right|t = t_n,\\
&e_{yi} = 1 - \tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{c}}^i, \quad e_{zi} = 1 - \frac{1}{i}\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-2}\tilde{\mathbf{c}}^i, \quad e_{ui} = 1 - \frac{1}{i(i-1)}\tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}^{-3}\tilde{\mathbf{c}}^i.
\end{aligned} \tag{6.6}$$

(в выражении для δu_{n+1} предполагаем, что $q \geq 2$).

Из (6.6) видно, что в общем случае локальные ошибки при решении уравнений (6.3) имеют следующие оценки:

$\delta y_{n+1} = O(h^{q+1})$, $\delta z_{n+1} = O(h^q)$, $\delta u_{n+1} = O(h^{q-1})$.

Из (6.4), (6.5) следует, что при $|R(\infty)| < 1$ порядки глобальных ошибок совпадают с соответствующими порядками локальных ошибок, а при $|R(\infty)| = 1$ порядки глобальных ошибок будут ниже (за исключением переменной y при $|R(\infty)| = -1$). Оценки ошибок при решении уравнений (6.3) полностью согласуются с теоретическими результатами, полученными в [75, 109, 116, 117] для ДАУ более общего вида.

Уравнения (6.3) являются частным случаем ДАУ, поэтому полученные выражения для ошибок (6.4) и (6.6) позволяют вывести условия, которые вместе с классическими условиями порядка являются необходимыми (но не обязательно достаточными) для достижения заданных порядков при решении ДАУ более общего вида. Обозначим через p_y, p_z и p_u порядки сходимости соответствующих компонент и предположим, что $|R(\infty)| < 1$, $p_y > q + 1$, $p_z > q$ и $p_u > q - 1$. Тогда для достижения заданных порядков необходимо выполнение условий:

$e_{yi} = 0$, $i = q + 1,\ldots, p_y - 1$;

$e_{zi} = 0$, $i = q + 1,\ldots, p_z$;

$e_{ui} = 0$, $i = q + 1,\ldots, p_u + 1$.

Жесткоточные методы обеспечивают точное выполнение алгебраических соотношений в системе ДАУ, и для них все коэффициенты e_{yi} в (6.6) равны 0. Из этого следует, что при решении нежестких ДАУ индекса 1 порядок как дифференциальной, так и алгебраической компоненты совпадает с классическим порядком метода [75, раздел VI.1]. При решении ДАУ индексов 2 и 3 жесткоточным методом порядок y-компоненты может быть выше, чем $q + 1$, но порядки z- и u-компонент остаются низкими ($p_z = q$ и $p_u = q - 1$ при $|R(\infty)| < 1$, см. [116, теорема 5.2] и [117, теорема 6.1]). Попробуем построить методы, обеспечивающие более высокую точность этих компонент.

Методы для ДАУ индекса 2. Примем $c_{s-1} = 1 - \alpha$, а $c_2,\ldots, c_{s-2}$ зададим из условия обеспечения максимального порядка метода, равного в этом случае $2s - 3$. В результате при $s = 3, 4, 5$ получим методы, задаваемые следующими узлами:

$$s = 3,\quad c_1 = 0,\quad c_2 = 1 - \alpha,\quad c_3 = 1; \tag{6.7}$$

$$s = 4,\quad c_1 = 0,\quad c_2 = \frac{2 - 5\alpha}{5 - 10\alpha},\quad c_3 = 1 - \alpha,\quad c_4 = 1; \tag{6.8}$$

$$\left.\begin{aligned} &s = 5,\quad c_1 = 0,\quad c_4 = 1 - \alpha,\quad c_5 = 1,\\ &c_{2,3} = \frac{35\alpha^2 - 33\alpha + 6 \mp \sqrt{245\alpha^4 - 490\alpha^3 + 333\alpha^2 - 88\alpha + 8}}{70\alpha^2 - 70\alpha + 14} \end{aligned}\right\}. \tag{6.9}$$

Эти методы совпадают с рассмотренными выше методами SAFERK, отличаясь от них только рекомендуемыми значениями свободного параметра.

Метод (6.7) имеет функцию устойчивости

$$R(z) = \frac{6 + 2(1 + \alpha)z + \alpha z^2}{6 - (4 - 2\alpha)z + (1 - \alpha)z^2},$$

которая в пределе при $\alpha \to 0$ совпадает с (1, 2)-аппроксимацией Паде экспоненциальной функции (как у двухстадийного метода Радо IIA). Кроме этого, для всех коэффициентов e_{zi} из (6.6) имеем $e_{zi} \to 0$ при $\alpha \to 0$.

Аналогичными свойствами обладают и методы с узлами (6.8) и (6.9). При уменьшении α их функция устойчивости приближается к $(s - 2, s - 1)$-аппроксимации Паде, а коэффициенты e_{zi} устремляются к 0. Таким образом, минимизация ошибки решения системы (6.3а, б) приводит к «почти L-устойчивым» методам с малым значением α.

В пределе при $\alpha \to 0$ рассмотренные методы порождают методы, использующие вторую производную, вычисленную в конечной точке интервала $[t_n, t_{n+1}]$. Например, на основе узлов (6.7) получим метод, расчетные формулы которого применительно к системе ДАУ индекса 2 (4.5а, б) имеют вид:

$$\begin{aligned} &\mathbf{y}_{n+1} = \mathbf{y}_n + h\left(\frac{1}{3}\mathbf{y}'_n + \frac{2}{3}\mathbf{y}'_{n+1}\right) - \frac{h^2}{6}\mathbf{y}''_{n+1},\\ &\mathbf{z}_{n+1} = \mathbf{z}_n + h\left(\frac{1}{3}\mathbf{z}'_n + \frac{2}{3}\mathbf{z}'_{n+1}\right) - \frac{h^2}{6}\mathbf{z}''_{n+1},\\ &\mathbf{0} = \mathbf{g}(\mathbf{y}_{n+1}),\quad \mathbf{0} = \mathbf{g}_{\mathrm{y}}(\mathbf{y}_{n+1})\mathbf{y}'_{n+1}. \end{aligned} \tag{6.10}$$

Этот метод обеспечивает не только точное выполнение соотношения $\mathbf{0} = \mathbf{g}(\mathbf{y})$, но и выполнение соотношения, полученного в результате дифференцирования алгебраического уравнения. В частности, это означает, что уравнения (6.3а, б) будут решены без ошибок. Таким образом, данный метод усиливает свойство жесткой точности применительно к ДАУ индекса 2.

Методы для ДАУ индекса 3. В рассмотренных выше методах было принято расположение узлов, позволяющее получить хорошие оценки 1-й и 2-й производных решения при $t = t_{n+1}$. Это позволило повысить точность z-компоненты. Действуя по аналогии, для решения ДАУ индекса 3 примем расположение узлов (6.2), которое позволяет получить также и хорошую оценку 3-й производной. В этом случае максимально возможный порядок метода равен $2s - 4$, и при $s = 4, 5$ получаем методы, задаваемые узлами:

$$s = 4, \quad c_1 = 0, \quad c_2 = 1 - \alpha, \quad c_3 = 1 + \alpha, \quad c_4 = 1; \tag{6.11}$$

$$s = 5, \quad c_1 = 0, \quad c_2 = \frac{1 - 5\alpha^2}{3 - 10\alpha^2}, \quad c_3 = 1 - \alpha, \quad c_4 = 1 + \alpha, \quad c_5 = 1. \tag{6.12}$$

Для этих методов при $\alpha \to 0$ имеем

$$a_0 = R(\infty) \to 0, \quad a_1 = \lim_{z \to \infty} z(R(z) - a_0) \to 0,$$

а функция устойчивости $R(z)$ приближается к $(s - 3, s - 1)$-аппроксимации Паде экспоненциальной функции. Кроме этого, при $\alpha \to 0$ все коэффициенты e_{zi}, e_{ui} из (6.6) приближаются к 0.

В пределе при $\alpha \to 0$ получаем методы, использующие вторую и третью производные, вычисленные в конечной точке шага интегрирования. Например, на основе узлов (6.11) получим метод

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left(\frac{1}{4}\mathbf{y}'_n + \frac{3}{4}\mathbf{y}'_{n+1}\right) - \frac{h^2}{4}\mathbf{y}''_{n+1} + \frac{h^3}{24}\mathbf{y}'''_{n+1}. \tag{6.13}$$

Такие методы обеспечивают точное выполнение соотношений, полученных в результате однократного и двукратного дифференцирований алгебраического уравнения в системе индекса 3 (4.6). В частности, они точно решают все уравнения системы (6.3). Отметим, что методы (6.10) и (6.13) относятся к методам Обрешкова [74].

Численные эксперименты. Обозначим методы с узлами (6.7)–(6.9) через CRKDAE II (Collocation Runge–Kutta for DAEs of index II), а методы с узлами (6.11), (6.12) – через CRKDAE III. Эти методы применялись для решения нескольких тестовых задач индексов 2 и 3 при различных значениях α. Уменьшение α приводит к увеличению коэффициентов последних двух столбцов матрицы $\mathbf{A}$ в методах CRKDAE II и последних трех столбцов в методах CRKDAE III. Это объясняется тем, что последние стадии фактически осуществляют численное дифференцирование правой части. При достаточно малых α эти коэффициенты растут пропорционально α^{-1} в методах CRKDAE II и пропорционально α^{-2} в методах CRKDAE III. Увеличение коэффициентов приводит к росту ошибок округления, поэтому следует ограничить снизу значение α.

Приведем результаты решения задачи индекса 2 (4.39) и индекса 3 (4.42) на интервале $0 \le t \le 2$. Размер шага выбираем из условия выполнения 24 неявных стадий на всем интервале, тогда $h = r/12$, где r – число неявных стадий. Вычисляем максимальные относительные ошибки на всем интервале по соответствующим компонентам. Чтобы посмотреть влияние α на ошибку решения, для каждого метода задаем два различных значения этого параметра: α_1 и α_2. Выбранные значения α, а также соответствующие им значения максимального коэффициента матрицы **A** приведены в табл. 6.4. Ошибки решения задачи индекса 2 приведены в табл. 6.5, а задачи индекса 3 – в табл. 6.6.

Таблица 6.4. Значения α коллокационных методов

s	CRKDAE II		CRKDAE III	
	α_1, $\max\|a_{ij}\|$	α_2, $\max\|a_{ij}\|$	α_1, $\max\|a_{ij}\|$	α_2, $\max\|a_{ij}\|$
3	0.2; 1.04	0.002; 83.5	–	–
4	0.03; 1.06	0.0003; 92.7	0.3; 1.06	0.03; 92.1
5	0.01; 0.91	0.0001; 83.2	0.1; 0.86	0.01; 83.2

Таблица 6.5. Ошибки решения системы ДАУ индекса 2

Метод	$r = 2$		$r = 3$		$r = 4$	
	e_y	e_z	e_y	e_z	e_y	e_z
Радо IIA	6.12×10^{-4}	8.27×10^{-3}	3.78×10^{-6}	1.32×10^{-3}	3.72×10^{-8}	1.84×10^{-4}
Лобатто IIIA	7.44×10^{-5}	6.08×10^{-2}	1.57×10^{-7}	1.36×10^{-3}	1.60×10^{-9}	5.95×10^{-4}
CRKDAE II (α_1)	3.73×10^{-4}	5.56×10^{-4}	3.57×10^{-6}	9.79×10^{-6}	3.42×10^{-8}	5.72×10^{-7}
CRKDAE II (α_2)	6.04×10^{-4}	3.04×10^{-4}	3.96×10^{-6}	2.06×10^{-6}	3.52×10^{-8}	2.30×10^{-8}
CRKDAE III (α_1)	–	–	8.39×10^{-5}	8.91×10^{-5}	8.60×10^{-7}	1.56×10^{-6}
CRKDAE III (α_2)	–	–	1.21×10^{-4}	6.09×10^{-5}	9.48×10^{-7}	4.85×10^{-7}

Таблица 6.6. Ошибки решения системы ДАУ индекса 3

Метод	$r = 3$			$r = 4$		
	e_y	e_z	e_u	e_y	e_z	e_u
Радо IIA	2.87×10^{-5}	8.91×10^{-4}	1.50×10^{-2}	1.00×10^{-6}	1.26×10^{-4}	4.16×10^{-3}
Лобатто IIIA	7.95×10^{-4}	7.53×10^{-3}	5.11×10^{-1}	4.19×10^{-5}	6.68×10^{-4}	1.22×10^{-1}
CRKDAE II (α_1)	1.50×10^{-6}	8.16×10^{-6}	8.29×10^{-4}	3.01×10^{-8}	4.77×10^{-7}	1.24×10^{-4}
CRKDAE II (α_2)	1.53×10^{-6}	5.51×10^{-6}	7.68×10^{-4}	2.63×10^{-8}	8.57×10^{-8}	1.08×10^{-4}
CRKDAE III (α_1)	1.39×10^{-5}	3.47×10^{-5}	1.79×10^{-4}	2.23×10^{-7}	8.33×10^{-7}	4.93×10^{-6}
CRKDAE III (α_2)	7.78×10^{-6}	1.89×10^{-5}	1.24×10^{-5}	2.23×10^{-7}	1.73×10^{-7}	1.84×10^{-7}

Были проведены также эксперименты по определению реальных порядков сходимости отдельных компонент решения при $h \to 0$. Порядки методов CRKDAE оценивались при достаточно большом α, а также при $\alpha \to 0$ (значения h и α уменьшали до тех пор, пока оценки порядков практически не переставали изменяться). Полученные оценки для ДАУ индекса 2 приведены в табл. 6.7,

а для ДАУ индекса 3 – в табл. 6.8. Для методов Лобатто IIIA, имеющих $R(\infty) = -1$, порядки отдельных компонент при переменном шаге ниже, чем при постоянном шаге (см. примечание 2 к теореме 5.2 из [116]). Поэтому были получены также оценки порядков для переменного шага, которые в случае их отличия приведены в табл. 6.7 и 6.8 в скобках (мы чередовали размер шага h и $h/2$). Оценки методов Радо IIA полностью совпадают с теоретическими оценками, полученными для ДАУ индекса 2 в [75, табл. VII.4.1] и для ДАУ индекса 3 в [117]. А оценки методов Лобатто IIIA и CRKDAE для ДАУ индекса 2 совпадают с оценками, полученными в [116]. О теоретических результатах по другим позициям табл. 6.7 и 6.8 (в том числе и для $\alpha \to 0$, т. е. для кратных узлов) нам неизвестно.

Таблица 6.7. Оценки порядков сходимости для ДАУ индекса 2

Метод	$r = 2$		$r = 3$		$r = 4$	
	p_y	p_z	p_y	p_z	p_y	p_z
Радо IIA	3	2	5	3	7	4
Лобатто IIIA	4	2	6	4(3)	8	4
CRKDAE II	3	3	5	4	7	5
CRKDAE II ($\alpha \to 0$)	3	3	5	5	7	7
CRKDAE III	–	–	4	4	6	5
CRKDAE III ($\alpha \to 0$)	–	–	4	4	6	6

Таблица 6.8. Оценки порядков сходимости для ДАУ индекса 3

Метод	$r = 3$			$r = 4$		
	p_y	p_z	p_u	p_y	p_z	p_u
Радо IIA	4	3	2	6	4	3
Лобатто IIIA	4(2)	4(2)	2(1)	4	4	2
CRKDAE II	5	4	3	7	5	4
CRKDAE II ($\alpha \to 0$)	5	5	3	7	7	4
CRKDAE III	4	4	3	6	5	4
CRKDAE III ($\alpha \to 0$)	4	4	4	6	6	6

Результаты численных экспериментов показали, что методы CRKDAE оказались более точными, по сравнению с другими методами, и позволяют в значительной степени уменьшить эффект снижения порядка. Недостатком методов является возможный рост ошибок при малых значениях α. Однако методы CRKDAE имеют ощутимое преимущество даже при относительно больших α, когда ошибки округления практически не сказываются.

6.3. Неявные методы Рунге–Кутты с явными внутренними стадиями

На примере диагонально-неявных и коллокационных методов мы убедились, что введение явной первой стадии в неявный метод позволяет повысить его точ-

ность без увеличения вычислительных затрат. Рассмотрим теперь возможность получения аналогичного эффекта путем введения явных внутренних стадий.

Обозначим через r степень знаменателя функции устойчивости, вычисленной по формуле

$R(z) = |\mathbf{I} - z\mathbf{A} + z\mathbf{e}\mathbf{b}^{\mathrm{T}}|/|\mathbf{I} - z\mathbf{A}|$.

Предположим также, что ранг матрицы $\mathbf{A}$ (либо $\tilde{\mathbf{A}}$ для метода вида (4.4) с явной первой стадией) равен r. Тогда среди строк матрицы $\mathbf{A}$ (либо $\tilde{\mathbf{A}}$) можно выделить r базисных строк, через которые линейно выражаются остальные строки. Соответствующие базисным строкам стадии будем считать неявными, а остальные стадии – явными. Разделение стадий на явные и неявные достаточно условно, поскольку выбор базисных строк неоднозначен. Явные стадийные значения являются линейными комбинациями неявных значений, поэтому систему алгебраических уравнений можно формировать и решать относительно только неявных стадий. Введение явных стадий позволяет при том же числе неявных стадий повысить стадийный (либо псевдостадийный) порядок метода.

До недавнего времени эффективные неявные методы с явными внутренними стадиями не были известны. Методы Лобатто IIIB содержат явную внутреннюю стадию, но для решения жестких задач они не подходят [75]. Впервые эффективные неявные методы с явной первой и явными внутренними стадиями, получившие название «неявные гнездовые методы типа Гаусса», были предложены в [30, 31, 121]. Неявные методы с явными внутренними стадиями, пригодные для решения жестких задач и ДАУ индексов 2 и 3, предложены также в [69–71].

Методы 4-го порядка. Рассмотрим построение четырехстадийных методов с явной первой и одной явной внутренней стадиями, имеющих функцию устойчивости

$$R(z) = \frac{1+\beta_1 z+\beta_2 z^2}{1+\alpha_1 z+\alpha_2 z^2}, \quad \alpha_1 = -\frac{2}{3}+2\beta_2, \quad \alpha_2 = \frac{1}{6}-\beta_2, \quad \beta_1 = \frac{1}{3}+2\beta_2. \tag{6.14}$$

При $\beta_2 = 0$ эта функция задает аппроксимацию Паде 3-го порядка, а при $\beta_2 = 1/12$ – аппроксимацию Паде 4-го порядка. Предположим, что $q \geq 2$, тогда

$$a_{i1} = c_i - a_{i2} - a_{i3} - a_{i4}, \quad a_{i2} = \frac{c_i^2 - 2(a_{i3}c_3 + a_{i4}c_4)}{2c_2}, \quad i = 2, 3, 4. \tag{6.15}$$

Для обеспечения заданной функции устойчивости (6.14) коэффициенты многочлена $|\mathbf{I} - z\tilde{\mathbf{A}}| = 1 + \alpha_1 z + \alpha_2 z^2 + \alpha_3 z^3$ должны удовлетворять условиям

$$\alpha_1 = -\frac{2}{3}+2\beta_2, \ \alpha_2 = \frac{1}{6}-\beta_2, \quad \alpha_3 = -\left|\tilde{\mathbf{A}}\right| = 0. \tag{6.16}$$

При $p = 3$ условия (6.16) однозначно задают функцию устойчивости (6.14). Потребуем, чтобы построенные методы имели псевдостадийный порядок $\overline{q} = 3$, тогда они должны удовлетворять равенствам

$$\mathbf{e}_{s-1}^{\mathrm{T}}(\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = 0, \quad \tilde{\mathbf{b}}^{\mathrm{T}}(\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = 0, \quad \tilde{\mathbf{b}}^{\mathrm{T}}\tilde{\mathbf{A}}(\tilde{\mathbf{c}}^3 - 3\tilde{\mathbf{A}}\tilde{\mathbf{c}}^2) = 0. \tag{6.17}$$

На основе условий (6.15)–(6.17) в [71] построено четырехпараметрическое семейство методов, имеющих $q = 2$, $\bar{q} = 3$, функцию устойчивости (6.14) и свободные параметры β_2, c_2, c_3 и a_{33}.

Рассмотрим методы 4-го порядка, которые получим, задав в (6.16) $\beta_2 = 1/12$. Они имеют функцию устойчивости

$$R(z) = \frac{1 + z/2 + z^2/12}{1 - z/2 + z^2/12}$$

(такую же, как у методов Гаусса и Лобатто IIIA 4-го порядка). Потребовав, чтобы метод был симметричным (условия симметричности см. в [74]), получим двухпараметрическое семейство со свободными параметрами $\alpha = c_2$, $\beta = a_{33}$ и таблицей Бутчера

$$\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\alpha & \beta - \dfrac{1 - 12\alpha^2 + 18\alpha^3 - 6\alpha^4}{12\alpha(1-\alpha)} & \dfrac{1}{12\alpha(1-\alpha)} - \beta & \dfrac{1}{12\alpha(1-\alpha)} - \beta & \beta - \dfrac{1 - 6\alpha^3 + 6\alpha^4}{12\alpha(1-\alpha)} \\
1-\alpha & \dfrac{1-\alpha^2}{2} - \beta & \beta & \beta & \dfrac{(1-\alpha)^2}{2} - \beta \\
1 & -\dfrac{1 - 6\alpha + 6\alpha^2}{12\alpha(1-\alpha)} & \dfrac{1}{12\alpha(1-\alpha)} & \dfrac{1}{12\alpha(1-\alpha)} & -\dfrac{1 - 6\alpha + 6\alpha^2}{12\alpha(1-\alpha)}
\end{array} . \qquad (6.18)$$

При $\alpha = (3 - \sqrt{3})/6$ получаем гауссовы узлы, а таблица (6.18) задает гнездовой метод типа Гаусса, предложенный в [121].

Обсудим выбор коэффициента β. В общем случае методы (6.18) имеют $q = 2$ и $\bar{q} = 3$. Если задать

$$\beta = \frac{(1-\alpha)(1+2\alpha)}{12\alpha}, \qquad (6.19)$$

то получим $q = \bar{q} = 3$. Наличие явных стадий может повлиять на устойчивость внутренних стадий. Рассмотрим вектор стадийных функций устойчивости $\mathbf{R}(z) = (\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}$. При больших значениях z 2-я и 3-я компоненты этого вектора пропорциональны z и выражаются формулой $R_{2,3}(z) \approx (1 - \alpha)(1 - 12\alpha\beta)z$. Можно избежать неустойчивости внутренних стадий, если задать

$$\beta = \frac{1}{12\alpha}. \qquad (6.20)$$

Такой метод имеет $q = 2$ и $R_{2,3}(\infty) = 1 - 6\alpha + 6\alpha^2$.

Поскольку в методах (6.18) $\bar{q} = 3 < p$, то эффект снижения порядка хоть и в малой степени, но может проявляться. Поэтому имеет смысл проанализировать функции погрешности 4-го порядка. Всего их пять, но для наших методов при любом значении β имеем:

$$e_{41}(z) = e_{42}(z) = e_{45}(z) = \frac{2\alpha(1-\alpha)z}{12 - 6z + z^2}. \qquad (6.21)$$

Такие же значения имеют функции $e_{43}(z)$ и $e_{44}(z)$ при $q = 3$, т. е. при β в виде (6.19), а при β в виде (6.20) получаем

$$e_{43}(z) = e_{44}(z) = \frac{z}{2(12 - 6z + z^2)}. \tag{6.22}$$

Задание малого α обеспечивает малые значения всех функций погрешности 4-го порядка при $q = 3$ и функций (6.21) при $q = 2$. Но очень малым задавать α не следует, поскольку в этом случае коэффициенты метода становятся большими, что может привести к росту вычислительных ошибок. Мы выбрали $\alpha = 1/30$, тогда наибольший по модулю коэффициент $a_{42} = a_{43} = -75/29$. Примем обозначение новых методов в виде IESRK (Implicit with Explicit Stages Runge–Kutta). Метод, полученный при $\alpha = 1/30$ и значении β (6.19), обозначим через IESRK43, а метод при $\alpha = 1/30$ и значении β (6.20) – через IESRK423. Как и в 4-й главе, 1-я цифра в названии – порядок p, 2-я цифра – q, а если $\overline{q} > q$, то 3-я цифра – $\overline{q}$. При двух разных значениях параметра β можно оценить влияние выбора этого параметра на эффективность метода. Для сравнения используем метод Лобатто IIIA 4-го порядка, который обозначим через Lobatto43. Заметим, что этот метод можно получить из семейства (6.18), задав $\alpha = 1/2$, $\beta = 1/6$. Тогда 2-я и 3-я стадии совпадут, а исключив одну из них, получим метод Lobatto43.

Методы 5-го и 6-го порядков. Методы 5-го порядка построим на основе функции устойчивости в виде аппроксимации Паде

$$R(z) = \frac{60 + 24z + 3z^2}{60 - 36z + 9z^2 - z^3}.$$

Примем $s = 5$, $q = 3$ и $\overline{q} = 4$, тогда наряду с условиями стадийного порядка должны выполняться условия:

$$\left|\mathbf{I} - z\tilde{\mathbf{A}}\right| = 1 - \frac{3}{5}z + \frac{3}{20}z^2 - \frac{1}{60}z^3, \quad \mathbf{e}_{s-1}^{\mathrm{T}}\tilde{\mathbf{A}}^i(\tilde{\mathbf{c}}^4 - 4\tilde{\mathbf{A}}\tilde{\mathbf{c}}^3) = 0, \quad i = 0, \ldots, 3.$$

Мы приняли $c_i = (i - 1)/4$ и построили два метода: IESRK54 ($q = 4$, неустойчивые внутренние стадии) и IESRK534 ($q = 3$, устойчивые внутренние стадии). Коэффициенты этих методов:

IESRK54

0	0	0	0	0	0
1/4	413/2880	−1/1440	59/240	−271/1440	143/2880
1/2	53/360	7/90	7/15	−23/90	23/360
3/4	33/320	39/160	27/80	9/160	3/320
1	7/90	16/45	2/15	16/45	7/90

IESRK534

0	0	0	0	0	0
1/4	31/225	−31/900	457/1200	−301/900	361/3600
1/2	271/1800	22/225	29/75	−38/225	61/1800
3/4	43/400	27/100	93/400	17/100	−3/100
1	7/90	16/45	2/15	16/45	7/90

Для сравнения используем метод Радо IIA 5-го порядка, который обозначим через Radau53.

Был построен также метод 6-го порядка с функцией устойчивости

$$R(z)=\frac{1+z/2+z^2/10+z^3/120}{1-z/2+z^2/10-z^3/120}.$$

Чтобы получить $\bar{q}=5$, должно быть не менее двух явных внутренних стадий, т. е. $s\geq 6$, но в этом случае не удалось построить метод с устойчивыми внутренними стадиями. Мы задали $s=6$ и $q=\bar{q}=5$, а коэффициенты матрицы $\mathbf{A}$ определили из условий 5-го стадийного порядка и $|\mathbf{I}-z\tilde{\mathbf{A}}|=1-z/2+z^2/10-z^3/120$. Исходя из условия симметричности, вектор абсцисс должен иметь вид $\mathbf{c}=[0,\beta,\alpha,1-\alpha,1-\beta,1]^{\mathrm{T}}$. Для минимизации функций погрешности 6-го порядка следует задать $\beta=-\alpha$, тогда для всех j

$$e_{6j}(z)=\frac{4\alpha^2(5\alpha^2-3)}{120-60z+12z^2-z^3}z,$$

и при малом значении α эти функции тоже будут малы. Задав $\alpha=0.1$, получим метод

$$\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.1 & \frac{-131}{1350} & \frac{-1439}{48000} & \frac{24797}{864000} & \frac{-15133}{864000} & \frac{-449}{48000} & \frac{17}{675} \\
0.1 & \frac{533}{7425} & \frac{-5261}{528000} & \frac{32533}{864000} & \frac{5803}{864000} & \frac{2029}{528000} & \frac{-149}{14850} \\
0.9 & \frac{-369}{275} & \frac{72657}{176000} & \frac{45711}{32000} & \frac{44721}{32000} & \frac{75087}{176000} & \frac{-783}{550} \\
1.1 & \frac{-1859}{1350} & \frac{20449}{48000} & \frac{1255133}{864000} & \frac{1215203}{864000} & \frac{21439}{48000} & \frac{-847}{675} \\
1 & \frac{-73}{54} & \frac{5}{12} & \frac{155}{108} & \frac{155}{108} & \frac{5}{12} & \frac{-73}{54}
\end{array}.$$

Обозначим построенный метод через IESRK65, а для сравнения используем метод Лобатто IIIA 6-го порядка (Lobatto64).

Численные эксперименты. На рис. 6.1 приведены зависимости ошибки решения задачи Капса от жесткости при $h=0.1$. Среди методов 4-го порядка лучшим оказался метод с минимизированными функциями погрешности IESRK43. Благодаря более высокому псевдостадийному порядку ошибки методов 5-го порядка IESRK54 и IESRK534 значительно меньше, чем у Radau53. Среди методов 6-го порядка более точен IESRK65, который имеет $q=5$, тогда как у Lobatto64 $q=4$.

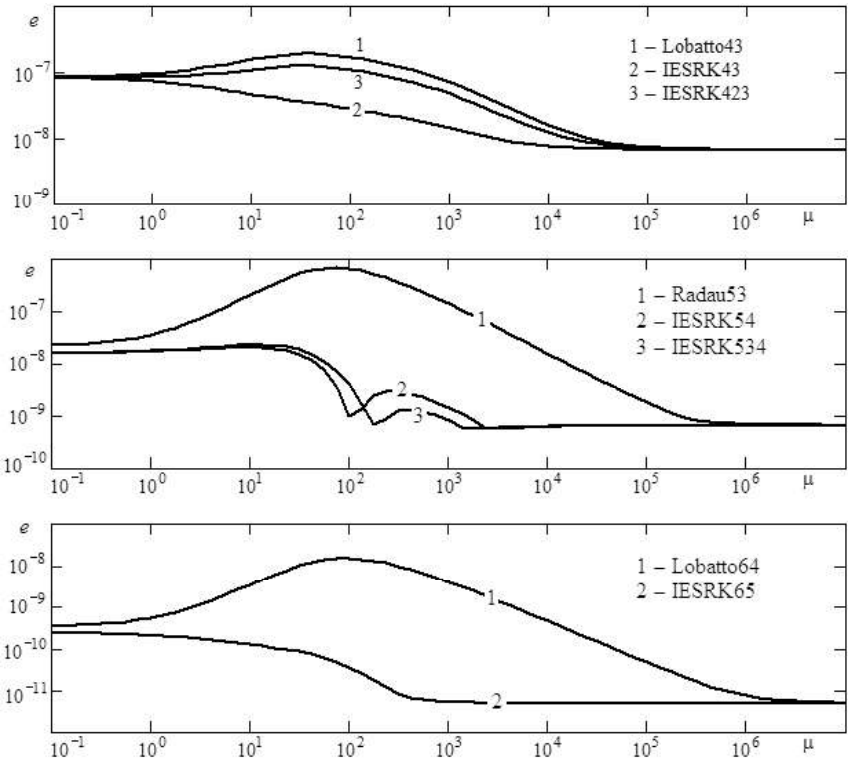


Рис. 6.1. Ошибки решения задачи Капса

Линейная жесткая задача взята из [93] и получена путем дискретизации методом прямых уравнения диффузии $\partial u/\partial t = \partial^2 u/\partial x^2$, $0 \le x \le 1$. Полученная в результате система ОДУ имеет вид:

$$y_i' = (N+1)^2(y_{i-1} - 2y_i + y_{i+1}), \quad i = 1, \ldots, N,$$
$$y_0 = 0, \; y_{N+1} = \phi(t), \; 0 \le t \le T,$$
$$\phi(t) = ae^{-\nu t}\sin(\sqrt{2}) - e^{-\mu t}\sin(1), \quad a = \cos(\sqrt{2})\Big/\left(\sqrt{2}\cos(2^{-1/2})\right), \tag{6.23}$$
$$\mu = 2(N+1)^2\left(1 - \cos\left(\frac{1}{N+1}\right)\right), \quad \nu = 2(N+1)^2\left(1 - \cos\left(\frac{\sqrt{2}}{N+1}\right)\right),$$

а ее точное решение:

$$y_i = ae^{-\nu t}\sin\left(\frac{\sqrt{2}\,i}{N+1}\right) - e^{-\mu t}\sin\left(\frac{i}{N+1}\right), \quad i = 1, \ldots, N.$$

Как и в [93], принимаем $T = 1$, $N = 10$, тогда собственные значения матрицы Якоби расположены на интервале от –9.8 до –474.2.

Результаты решения этой задачи (максимальные ошибки на всем интервале среди всех компонент и оценки порядка) приведены в табл. 6.9. Они согласуются с результатами решения задачи Капса, за исключением результатов методов IESRK43 и IESRK423, которые заслуживают отдельного комментария. При решении задачи (6.23) эти методы показали одинаковые результаты, хотя при решении задачи Капса метод IESRK43 имел ощутимое преимущество. Объясняется это тем, что задача (6.23) – линейная, поэтому в разложении ее решения в ряд Тейлора отсутствуют элементарные дифференциалы, которые имеются у нелинейных задач. Методы IESRK43 и IESRK423 имеют одинаковую функцию погрешности $e_{41}(z)$, которая выражается формулой (6.21). Поэтому при решении линейной задачи (6.23) они показывают одинаковые результаты. Но для метода IESRK423 имеем $e_{43}(z) = (225/29)e_{41}(z)$, тогда как у IESRK43 $e_{43}(z) = e_{41}(z)$, что объясняет ощутимое преимущество метода IESRK43 при решении задачи Капса.

Таблица 6.9. Результаты решения задачи (6.23)

Метод	Ошибка		$\tilde{p}$
	$h = 0.25$	$h = 0.025$	
Lobatto43	5.60×10^{-6}	7.24×10^{-10}	3.89
IESRK43	1.67×10^{-6}	1.56×10^{-10}	4.03
IESRK423	1.67×10^{-6}	1.56×10^{-10}	4.03
Radau53	1.57×10^{-6}	1.56×10^{-10}	4.00
IESRK54	1.73×10^{-7}	2.95×10^{-12}	4.77
IESRK534	1.73×10^{-7}	2.95×10^{-12}	4.77
Lobatto64	1.10×10^{-7}	9.10×10^{-13}	5.08
IESRK65	3.31×10^{-9}	3.16×10^{-15}	6.02

Таким образом, приведенные результаты решения жестких задач полностью объясняются с помощью псевдостадийного порядка и функций погрешности. Псевдостадийный порядок оказывает решающее влияние на точность решения. При одинаковых значениях q более точен метод, имеющий большее значение $\bar{q}$, а при одинаковых $\bar{q}$ более точен метод с меньшими значениями функций погрешности порядка $\bar{q} + 1$.

Исследуем теперь сходимость методов при решении ДАУ высших индексов на примере задачи индекса 2 (4.39) и задачи индекса 3 (4.42). Ошибки при $h = 1/20$ и оценки порядка (4.15) при $h_1 = h, h_2 = h/2$ вычисляем для соответствующих компонент. Результаты решения задачи индекса 2 приведены в табл. 6.10, а задачи индекса 3 – в табл. 6.11. Если отсутствует сходимость метода (одна из оценок порядка около нуля или отрицательная), то его результаты не приводятся. Похожие результаты были получены также при решении других задач индексов 2 и 3. Все методы 4-го порядка оказались непригодными для решения задач индекса 3. Скорее всего, это вызвано тем, что они имеют $R(\infty) = 1$,

что приводит к накоплению ошибок. Среди методов 5-го порядка лучшим был метод с устойчивой внутренней стадией IESRK534. Из методов 6-го порядка только Lobatto64 обеспечил сходимость, но он имеет $R(\infty) = -1$, что приводит к ухудшению сходимости при интегрировании с переменным шагом.

Таблица 6.10. Результаты решения системы ДАУ индекса 2

Метод	e_y	e_z	$\tilde{p}_y$	$\tilde{p}_z$
Lobatto43	2.10×10^{-8}	3.01×10^{-4}	4.00	2.00
IESRK43	3.50×10^{-5}	2.26×10^{-5}	2.00	2.00
IESRK423	4.87×10^{-8}	1.89×10^{-4}	4.00	2.00
Radau53	1.52×10^{-10}	8.64×10^{-6}	5.00	2.99
IESRK54	3.91×10^{-7}	2.28×10^{-7}	2.97	3.15
IESRK534	2.14×10^{-10}	3.87×10^{-8}	5.00	3.99
Lobatto64	1.04×10^{-12}	1.58×10^{-7}	5.99	3.98

Таблица 6.11. Результаты решения системы ДАУ индекса 3

Метод	e_y	e_z	e_u	$\tilde{p}_y$	$\tilde{p}_z$	$\tilde{p}_u$
Radau53	2.01×10^{-9}	6.10×10^{-6}	5.03×10^{-4}	4.08	2.97	1.99
IESRK54	8.04×10^{-8}	7.40×10^{-7}	7.94×10^{-6}	2.99	2.98	2.99
IESRK534	1.60×10^{-10}	2.71×10^{-8}	6.56×10^{-6}	5.00	3.96	2.99
Lobatto64	1.21×10^{-9}	8.77×10^{-8}	4.63×10^{-4}	3.98	3.97	2.00

Результаты экспериментов показали, что методы с явными внутренними стадиями могут быть более точными, чем аналогичные методы (такого же порядка и с той же функцией устойчивости), не имеющие таких стадий. Эти методы могут иметь преимущество при решении больших задач с достаточно простыми правыми частями, когда основные вычислительные затраты связаны с решением на каждой итерации системы линейных алгебраических уравнений.

6.4. Неявный двухшаговый метод пятого порядка для жестких задач и ДАУ

Стадийный порядок одношаговых методов Рунге–Кутты не может быть больше числа стадий. Оптимальные (имеющие наивысший порядок при заданном числе стадий) методы Рунге–Кутты высоких порядков имеют большую разность между классическим и стадийным порядками, что приводит к снижению реального порядка при решении жестких задач и ДАУ. Среди жесткоточных L-устойчивых методов оптимальными являются методы Радо IIA, которые имеют классический порядок $2s - 1$ при стадийном порядке, равном числу стадий s. Тем не менее решатели RADAU5 и RADAU, реализующие эти мето-

ды, показывают превосходные результаты при решении многих задач. Можно ожидать, что методы, стадийный порядок которых превышает число стадий и равен классическому порядку, окажутся еще более эффективными.

Стадийный порядок может быть больше числа стадий в многошаговых методах либо в методах со старшими производными, к которым относятся методы Обрешкова [74]. В предлагаемом методе сочетаются оба этих подхода. За основу взят метод Обрешкова 5-го порядка

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h(b_1\mathbf{y}'_n + a_1\mathbf{y}'_{n+1}) + h^2(b_2\mathbf{y}''_n + a_2\mathbf{y}''_{n+1}) + h^3 a_3\mathbf{y}'''_{n+1},$$
$$b_1 = 2/5, \quad b_2 = 1/20, \quad a_1 = 3/5, \quad a_2 = -3/20, \quad a_3 = 1/60. \tag{6.24}$$

Его функция устойчивости

$$R(z) = \frac{60 + 24z + 3z^2}{60 - 36z + 9z^2 - z^3} \tag{6.25}$$

является аппроксимацией Паде (2, 3) экспоненты и совпадает с функцией устойчивости трехстадийного метода Радо IIA 5-го порядка. Недостатком метода (6.24) является необходимость вычисления старших производных, поэтому методы такого типа редко применяют на практике. Для устранения этого недостатка аппроксимируем старшие производные через конечные разности. В результате получим двухшаговый трехстадийный метод Рунге–Кутты 5-го порядка, стадийный порядок которого совпадает с классическим порядком.

Построение метода. Для аппроксимации 2-й и 3-й производных в начале и в конце интервала $[t_n, t_{n+1}]$ выберем абсциссы метода в виде $c_1 = 1 - \alpha$, $c_2 = 1 + \alpha$, $c_3 = 1$, где значение α достаточно мало (аналогично методам, рассмотренным в разделе 6.2). Формулы шага интегрирования при решении задачи $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ запишем в виде:

$$\mathbf{Y}_i = \mathbf{y}_n + h\sum_{j=1}^{3}(a_{ij}\mathbf{Y}'_j + b_{ij}\bar{\mathbf{Y}}'_j), \quad \mathbf{Y}'_i = \mathbf{f}\left(t_n + c_i h, \mathbf{Y}_i\right), \quad i = 1, 2, 3, \quad \mathbf{y}_{n+1} = \mathbf{Y}_3, \tag{6.26}$$

где $\bar{\mathbf{Y}}_j$ – векторы стадийных значений, полученные на предыдущем шаге. Такой метод относится к классу двухшаговых методов Рунге–Кутты [113].

Чтобы ограничиться оценкой 2-й производной при $t = t_n$, примем

$$b_{i1} = -b_{i2}, \quad i = 1, 2, 3. \tag{6.27}$$

Из условий 5-го стадийного порядка получим:

$$\sum_{j=1}^{3} a_{ij}c_j^{k-1} + \sum_{j=1}^{3} b_{ij}\bar{c}_j^{k-1} = \frac{c_i^k}{k}, \quad i = 1, 2, 3, \quad k = 1, \ldots, 5,$$
$$\bar{c}_1 = -\alpha/w, \quad \bar{c}_2 = \alpha/w, \quad \bar{c}_3 = 0, \tag{6.28}$$

где w – отношение размера текущего шага к размеру предыдущего шага. Из уравнений (6.27), (6.28) находим коэффициенты a_{ij}, b_{ij} (явные формулы для этих коэффициентов довольно громоздки, поэтому здесь не приводятся).

Поскольку метод двухшаговый, то на первом шаге следует использовать подходящий одношаговый метод, содержащий абсциссы $1-\alpha$, $1+\alpha$ и 1. Такие методы рассмотрены в разделе 6.2.

Свойства метода. Исследуем устойчивость метода. Обозначим через $\mathbf{A}$ и $\mathbf{B}$ матрицы коэффициентов a_{ij} и b_{ij}. Решая уравнение Далквиста $y' = \lambda y$, получим $\mathbf{Y} = \mathbf{H}(z)\bar{\mathbf{Y}}$, где $z = h\lambda$, $\mathbf{H}(z) = (\mathbf{I} - z\mathbf{A})^{-1}(\mathbf{E} + z\mathbf{B})$, $\mathbf{E} = (1,1,1)^{\mathrm{T}}(0,0,1)$. Устойчивость метода определяется корнями характеристического многочлена $P(\eta, z) = |\eta\mathbf{I} + \mathbf{H}(z)|$. Метод устойчив, если абсолютные значения этих корней не превышают 1, а среди корней, равных по модулю 1, нет кратных (см. определение V.1.1 из [75]). В нашем случае матрица $\mathbf{H}(z)$ имеет ранг 2, поэтому характеристический многочлен может иметь не более двух ненулевых корней и запишется в виде $P(\eta, z) = \eta(\eta^2 - p_1(z)\eta + p_0(z))$.

При постоянном размере шага ($w = 1$) имеем:

$$p_0(z) = N_0(z)/D(z), \quad p_1(z) = N_1(z)/D(z), \quad N_0(z) = \alpha^4(8z + 7z^2 + z^3),$$
$$N_1(z) = 60 + 120\alpha^2 + (24 + 24\alpha^2 + 16\alpha^4)z + (3 - 6\alpha^2 - 8\alpha^4)z^2 - (2\alpha^2 + 8\alpha^4)z^3,$$
$$D(z) = 60 + 120\alpha^2 - (36 + 96\alpha^2 - 8\alpha^4)z + (9 + 30\alpha^2 - 15\alpha^4)z^2 - (1 + 4\alpha^2 - 5\alpha^4)z^3.$$

Такой метод A-устойчив при $|\alpha| \le 0.498$ и $A(89°)$-устойчив при $|\alpha| \le 0.553$. При $\alpha = 0.5$ он имеет 6-й порядок. При $w \ne 1$ (т. е. при изменении размера шага) A-устойчивость может нарушаться, но если значение w не очень велико, то сохраняется $A(\varphi)$-устойчивость с углом φ, близким к 90°.

Если $\alpha \to 0$, то $p_0(z) \to 0$, $p_1(z) \to R(z)$, где $R(z)$ – функция устойчивости (6.25). Это означает, что при решении линейной автономной системы ОДУ новый метод с малым значением α практически равноценен по точности и устойчивости методам 5-го порядка (6.24) и Радо IIA.

Формулу интегрирования можно записать в виде:

$$\mathbf{y}_{n+1} = \mathbf{Y}_3 = \mathbf{y}_n + h\left(\alpha_1\mathbf{Y}_3' + \alpha_2\frac{\mathbf{Y}_2' - \mathbf{Y}_1'}{2\alpha} + \alpha_3\frac{\mathbf{Y}_2' - 2\mathbf{Y}_3' + \mathbf{Y}_1'}{\alpha^2} + \beta_1\bar{\mathbf{Y}}_3' + \beta_2\frac{\bar{\mathbf{Y}}_2' - \bar{\mathbf{Y}}_1'}{2\alpha/w}\right),$$

где $\alpha_1 = a_{31} + a_{32} + a_{33}$, $\alpha_2 = \alpha(a_{32} - a_{31})$, $\alpha_3 = \alpha^2(a_{31} + a_{32})/2$, $\beta_1 = b_{33}$, $\beta_2 = -2\alpha b_{31}/w$. При $\alpha \to 0$ имеем:

$$\alpha_i \to a_i, \quad \beta_i \to b_i, \quad \frac{\mathbf{Y}_2' - \mathbf{Y}_1'}{2\alpha} \to h\mathbf{y}''_{n+1},$$
$$\frac{\mathbf{Y}_2' - 2\mathbf{Y}_3' + \mathbf{Y}_1'}{\alpha^2} \to h^2\mathbf{y}'''_{n+1}, \quad \frac{\bar{\mathbf{Y}}_2' - \bar{\mathbf{Y}}_1'}{2\alpha/w} \to h\mathbf{y}''_n,$$

где a_i, b_i – коэффициенты из (6.24). Таким образом, при $\alpha \to 0$ абсциссы c_1, c_2 и c_3 стягиваются в один кратный узел, что эквивалентно использованию старших производных, а в пределе получаем метод Обрешкова (6.24).

Перейдем теперь к решению системы ДАУ

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}, \mathbf{z}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{6.29а}$$
$$\mathbf{0} = \mathbf{g}(t, \mathbf{y}, \mathbf{z}), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \tag{6.29б}$$

где $\mathbf{f}$ и $\mathbf{g}$ – достаточное число раз дифференцируемые векторные функции, размерности которых совпадают с размерностями векторов $\mathbf{y}$ и $\mathbf{z}$. Предположим, что начальные условия согласованы. Воспользуемся методом ε-вложения и рассмотрим решение этой системы с помощью метода Обрешкова (6.24). Обозначив через $\mathbf{g}_n$, $\mathbf{g}'_n$ и $\mathbf{g}''_n$ значения функции $\mathbf{g}$ и ее производных при $t = t_n$, получим:

$$\varepsilon(\mathbf{z}_{n+1} - \mathbf{z}_n) = h(b_1\mathbf{g}_n + a_1\mathbf{g}_{n+1}) + h^2(b_2\mathbf{g}'_n + a_2\mathbf{g}'_{n+1}) + h^3 a_3\mathbf{g}''_{n+1},$$

тогда $\mathbf{z}'_{n+1} = \varepsilon^{-1}\mathbf{g}_{n+1}$, $\mathbf{z}''_{n+1} = \varepsilon^{-1}\mathbf{g}'_{n+1}$, $\mathbf{z}'''_{n+1} = \varepsilon^{-1}\mathbf{g}''_{n+1}$. Если численное решение системы (6.29) существует, то значения $\mathbf{z}'_{n+1}$, $\mathbf{z}''_{n+1}$ и $\mathbf{z}'''_{n+1}$ должны иметь конечные пределы при $\varepsilon \to 0$, а это возможно, только когда $\mathbf{g}_{n+1} = \mathbf{g}'_{n+1} = \mathbf{g}''_{n+1} = \mathbf{0}$. Таким образом, один шаг численного решения уравнений (6.29) методом (6.24) запишется в виде:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h(b_1\mathbf{f}_n + a_1\mathbf{f}_{n+1}) + h^2(b_2\mathbf{f}'_n + a_2\mathbf{f}'_{n+1}) + h^3 a_3\mathbf{f}''_{n+1}, \tag{6.30а}$$

$$\mathbf{0} = \mathbf{g}_{n+1}, \quad \mathbf{0} = \mathbf{g}'_{n+1}, \quad \mathbf{0} = \mathbf{g}''_{n+1}. \tag{6.30б}$$

Размерность полученной системы нелинейных алгебраических уравнений (6.30) больше размерности исходной системы (6.29), но дополнительные уравнения в (6.30б) необходимы для нахождения векторов $\mathbf{z}'_{n+1}$ и $\mathbf{z}''_{n+1}$, которые входят в выражения для $\mathbf{f}'_{n+1}$, $\mathbf{f}''_{n+1}$, $\mathbf{g}'_{n+1}$ и $\mathbf{g}''_{n+1}$. Например, $\mathbf{f}' = \mathbf{f}_t + \mathbf{f}_\mathbf{y}\mathbf{f} + \mathbf{f}_\mathbf{z}\mathbf{z}'$, где $\mathbf{f}_t$, $\mathbf{f}_\mathbf{y}$ и $\mathbf{f}_\mathbf{z}$ – вектор и матрицы соответствующих частных производных.

Обсудим формулы (6.30). Известно, что жесткоточные методы обеспечивают точное выполнение алгебраического соотношения (6.29б). Благодаря этому они позволяют решать ДАУ индекса 1 без снижения порядка (теорема VI.3.3 из [75]). Наше предположение состоит в том, что выполнение соотношений (6.30б) в методе со старшими производными позволяет решать без снижения порядка также и ДАУ индексов 2 и 3. Это предположение подтверждается при решении тестовых задач, но строгого доказательства у нас нет.

Рассмотрим теперь решение ДАУ (6.29) предложенным методом. В этом случае формулы шага интегрирования запишутся в виде:

$$\begin{gathered}\mathbf{Y}_i = \mathbf{y}_n + h\sum_{j=1}^{3}(a_{ij}\mathbf{Y}'_j + b_{ij}\bar{\mathbf{Y}}'_j), \quad \mathbf{Y}'_i = \mathbf{f}(t_n + c_i h, \mathbf{Y}_i, \mathbf{Z}_i), \\ \mathbf{0} = \mathbf{g}(t_n + c_i h, \mathbf{Y}_i, \mathbf{Z}_i), \quad i = 1, 2, 3, \quad \mathbf{y}_{n+1} = \mathbf{Y}_3, \quad \mathbf{z}_{n+1} = \mathbf{Z}_3.\end{gathered} \tag{6.31}$$

Преимущество этих формул по сравнению с (6.30) – отсутствие старших производных, которые фактически заменяются конечными разностями. Численные эксперименты показали, что при малых значениях α методы (6.30) и (6.31) показывают близкие результаты без какого-либо снижения порядка.

При уменьшении α увеличиваются абсолютные значения коэффициентов (наибольшие по модулю коэффициенты $a_{i3} \approx -\alpha^{-2}/30$). Как следствие возрастают вычислительные ошибки, что может потребовать повышенной разрядности вычислений. Поэтому был проведен ряд экспериментов с целью нахождения подходящего значения α. Расчеты выполнялись с двойной точностью (64-битовые числа). В результате выбрано значение $\alpha = 0.1$, которое не очень

мало и в то же время обеспечивает решение задач практически без снижения порядка.

Численные эксперименты. Приведем сначала результаты расчетов с постоянным размером шага. Новый метод со значением $\alpha = 0.1$ обозначим через TSRK5 (Two Step Runge Kutta of order 5). Для сравнения используем метод Обрешкова (6.24), который обозначим через Obres5, и метод Радо IIA 5-го порядка, который обозначим через Radau5.

Для исследования влияния жесткости задачи на точность ее решения выберем систему ОДУ

$$y_1' = y_2 + \beta_1 e_1(t) + (\beta_2 - 1)e_2(t),\quad y_2' = -y_1 + (\beta_2 + 1)e_1(t) + \beta_1 e_2(t),$$
$$\beta_1 = -(\mu + 1)/2,\quad \beta_2 = -(\mu - 1)/2,\quad e_1(t) = y_1 - \sin t,\quad e_2(t) = y_2 - \cos t, \qquad (6.32)$$
$$y_1(0) = 0,\quad y_2(0) = 1,\quad 0 \le t \le 2\pi$$

с точным решением $y_1 = \sin t$, $y_2 = \cos t$. Собственные значения матрицы Якоби этой системы равны $-\mu$ и -1. Таким образом, значение μ может служить мерой жесткости задачи. Ошибку численного решения оцениванием величиной $e_y = \max\limits_t \sqrt{e_1(t)^2 + e_2(t)^2}$. Полученные ошибки в зависимости от жесткости задачи μ приведены в табл. 6.12. Видно, что ошибки методов Obres5 и TSRK5 почти не зависят от жесткости, а их поведение соответствует порядку метода (т. е. при уменьшении размера шага h в 10 раз ошибка уменьшается примерно в 10^p раз, где p – порядок метода). В то же время точность метода Radau5 заметно снижается при умеренной жесткости ($\mu = 10^1, 10^2$), а его реальный порядок при $\mu = 10^2$ ненамного больше 3-го. Такое поведение ошибки характерно для жесткоточных методов и вполне объясняется функциями погрешности.

Таблица 6.12. Ошибки решения жесткой задачи (6.32)

Метод	h	e_y				
		$\mu = 10^0$	$\mu = 10^1$	$\mu = 10^2$	$\mu = 10^3$	$\mu = 10^4$
TSRK5	$\pi/4$	2.98×10^{-5}	2.93×10^{-5}	2.93×10^{-5}	2.93×10^{-5}	2.93×10^{-5}
	$\pi/40$	3.22×10^{-10}	3.15×10^{-10}	3.14×10^{-10}	3.14×10^{-10}	3.14×10^{-10}
Obres5	$\pi/4$	2.93×10^{-5}	2.86×10^{-5}	2.86×10^{-5}	2.86×10^{-5}	2.86×10^{-5}
	$\pi/40$	3.11×10^{-10}	3.05×10^{-10}	3.04×10^{-10}	3.04×10^{-10}	3.04×10^{-10}
Radau5	$\pi/4$	4.50×10^{-5}	2.05×10^{-4}	5.56×10^{-5}	4.27×10^{-5}	4.27×10^{-5}
	$\pi/40$	4.75×10^{-10}	5.69×10^{-9}	2.21×10^{-8}	5.57×10^{-9}	6.37×10^{-10}

Рассмотрим теперь систему ДАУ индекса 3

$$y_1' = z_1,\quad y_2' = z_2,$$
$$z'_1 = y_2 \sin t - y_1 u,\quad z'_2 = -y_1 \sin t - y_2 u,\quad 0 = y_1^2 + y_2^2 - 1, \qquad (6.33)$$
$$y_1(0) = z_1(0) = z_2(0) = u(0) = 0,\quad y_2(0) = 1,\quad 0 \le t \le 2\pi,$$

точное решение которой: $y_1(t) = \sin(t - \sin t)$, $y_2(t) = \cos(t - \sin t)$, $z_1(t) = y_2(t)(1 - \cos t)$, $z_2(t) = -y_1(t)(1 - \cos t)$, $u(t) = (1 - \cos t)^2$. Обозначим через e_y, e_z и e_u ошибки соответ-

ствующих компонент, которые вычисляем аналогично предыдущему примеру. Для всех компонент приводим также оценки порядков сходимости, полученные путем сравнения ошибок при размерах шага h и $h/2$. Результаты приведены в табл. 6.13. Методы TSRK5 и Obres5 показывают близкие результаты, которые соответствуют порядку метода. А метод Radau5 демонстрирует снижение порядка, наиболее значительное для u-компоненты. При этом поведение ошибок метода Radau5 полностью соответствует теоретическим оценкам, полученным в теореме 6.1 из [117]. Аналогичные результаты были получены и при решении других задач индексов 2 и 3 с постоянным размером шага. Во всех случаях методы TSRK5 и Obres5 показывали близкие результаты и были заметно точнее, чем Radau5.

Таблица 6.13. Результаты решения системы ДАУ индекса 3 (6.33)

Метод	h	e_y	e_z	e_u	$\tilde{p}_y$	$\tilde{p}_z$	$\tilde{p}_u$
TSRK5	$\pi/16$	7.43×10^{-5}	4.48×10^{-5}	1.28×10^{-4}	4.86	4.82	4.90
	$\pi/160$	8.77×10^{-10}	5.60×10^{-10}	1.43×10^{-9}	4.99	4.98	4.99
Obres5	$\pi/16$	8.39×10^{-5}	5.04×10^{-5}	1.33×10^{-4}	4.88	4.84	4.91
	$\pi/160$	9.64×10^{-10}	6.09×10^{-10}	1.48×10^{-9}	4.99	4.98	4.99
Radau5	$\pi/16$	9.21×10^{-5}	2.24×10^{-3}	7.14×10^{-2}	4.96	2.99	2.03
	$\pi/160$	9.70×10^{-10}	2.27×10^{-6}	6.94×10^{-4}	4.99	3.00	2.00

При реализации метода TSRK5 с переменным шагом оценивание ошибки выполняем с помощью двухшаговой вложенной формулы 4-го порядка, построенной в виде интерполяционного многочлена по значениям $\bar{\mathbf{Y}}_1$, $\bar{\mathbf{Y}}_2$, $\bar{\mathbf{Y}}_3$, $\mathbf{Y}_1$ и $\mathbf{Y}_2$. Добавив к этим значениям $\mathbf{Y}_3$, можно получить прогноз 5-го порядка для стадийных значений следующего шага. При автоматическом выборе размера шага (в отличие от постоянного шага) результаты зависят от конкретной реализации метода. Поэтому мы различаем понятия «метод» и «решатель» и для сравнения используем решатель RADAU5 – наиболее эффективную реализацию метода Radau5.

Результаты решения трех тестовых задач приведены в табл. 6.14, где результаты для RADAU5 взяты из [85]. Как и в [85], принимаем допуск на ошибку в виде $Tol = Atol = Rtol$, где $Atol$ и $Rtol$ – задаваемые допуски на абсолютную и относительную ошибки, а фактическую точность оцениваем значениями *scd* и *mescd*.

Таблица 6.14. Результаты решения тестовых задач

Задача	Решатель	*Tol*	*scd*	*mescd*	*Nstp*(*Nbad*)	*Nf*	*NJ*	*NLU*
VDPOL	TSRK5	10^{-4}	4.37	4.70	234(39)	1834	68	246
		10^{-7}	7.19	7.52	839(6)	4273	71	673
		10^{-10}	10.36	10.68	3301(0)	14 002	77	1449
	RADAU5	10^{-4}	4.96	5.28	245(36)	2253	162	252
		10^{-7}	8.15	8.48	712(8)	5731	432	587
		10^{-10}	10.18	10.50	2270(2)	17 414	616	1711

Окончание табл. 6.14

Задача	Решатель	*Tol*	*scd*	*mescd*	*Nstp*(*Nbad*)	*Nf*	*NJ*	*NLU*
PLATE	TSRK5	10^{-3}	2.47	4.41	14(1)	52	1	15
		10^{-5}	4.78	7.01	34(2)	121	1	35
		10^{-7}	6.60	8.84	77(3)	271	1	67
	RADAU5	10^{-4}	1.62	3.77	14(3)	74	4	15
		10^{-7}	4.03	6.11	57(3)	267	4	29
		10^{-10}	6.00	8.04	190(3)	937	4	62
Pendulum	TSRK5	10^{-3}	2.57	3.88	9(2)	88	5	11
		10^{-6}	4.91	6.24	26(0)	154	9	26
		10^{-9}	6.25	7.57	45(0)	247	10	42
	RADAU5	10^{-3}	2.04	2.94	9(2)	82	7	11
		10^{-6}	3.21	3.69	21(3)	166	16	24
		10^{-9}	5.43	5.91	59(13)	553	40	71

Обсудим полученные результаты. При решении задачи VDPOL оба решателя показывают примерно одинаковые результаты по всем позициям, за исключением числа вычислений якобиана *NJ*, которое у TSRK5 в несколько раз меньше, чем у RADAU5. При уменьшении *Tol* значение *NJ* возрастает у RADAU5, оставаясь почти неизменным у TSRK5.

Решатель TSRK5 оказался особенно эффективным при решении задачи PLATE. В этом случае RADAU5 недооценивает ошибку (особенно при $Tol = 10^{-10}$), тогда как TSRK5 ее переоценивает. Если сравнивать оба решателя при примерно одинаковой фактической точности, то затраты решателя TSRK5 оказываются в несколько раз меньше, чем у RADAU5. Преимущество решателя TSRK5 можно объяснить тем, что задача имеет распределенный спектр матрицы Якоби, т. е. содержит не только жесткие, но и умеренно жесткие компоненты. А именно при умеренной жесткости проявляется феномен снижения порядка у жесткоточных методов, к которым относится Radau5.

При решении задачи индекса 3 Pendulum решатель TSRK5 также имеет преимущество, которое возрастает при уменьшении *Tol*.

Результаты экспериментов (в том числе и приведенные в разделе 6.2) позволяют сделать вывод о том, что использование старших производных или их оценок в конечной точке шага интегрирования усиливает свойство жесткой точности. Это проявляется в отсутствии снижения порядка не только при решении ДАУ индекса 1 (как у жесткоточных методов), но и при решении ДАУ индексов 2 и 3. Особенно перспективны двухшаговые методы такого типа, стадийный порядок которых совпадает с классическим порядком. Такие методы эффективны также и при решении жестких систем с распределенным спектром матрицы Якоби.

Явные методы с расширенными областями устойчивости

ГЛАВА 7

7.1. Явные стабилизированные методы Рунге–Кутты

Явные методы Рунге–Кутты с расширенными вдоль вещественной оси областями устойчивости могут быть эффективны при решении задач, имеющих вещественный жесткий спектр матрицы Якоби [32–34, 39, 63, 75, 79, 80, 127, 129, 130, 145, 148]. Многочлены устойчивости таких методов обычно рассчитывают, исходя из условия чебышёвского альтернанса, поэтому их называют также методами Рунге–Кутты–Чебышёва. При конструировании методов такого типа решаются две основные задачи: 1) расчет многочленов устойчивости порядка p и степеней $s = s_1, \ldots, s_m$; 2) построение методов Рунге–Кутты заданного порядка, имеющих заданные многочлены устойчивости. Под порядком многочлена будем понимать порядок аппроксимации экспоненты в окрестности нуля. Обычно порядок многочлена устойчивости совпадает с порядком метода, но бывают и исключения (например, в [127] построены методы 2-го порядка, многочлены устойчивости которых имеют 6-й порядок).

Первую из перечисленных задач можно сформулировать следующим образом: для заданных p, s и η найти многочлен

$$R(z) = 1 + \sum_{i=1}^{p} z^i / i! + \sum_{i=p+1}^{s} a_i z^i, \tag{7.1}$$

удовлетворяющий условиям

$$-l \le z \le 0 \Rightarrow |R(z)| \le 1; \quad -l < z^* < 0, R'(z^*) = 0 \Rightarrow |R(z^*)| \le \eta \tag{7.2}$$

при максимально возможной длине интервала устойчивости l. Многочлен, полученный в результате решения этой задачи, будем называть *оптимальным* (при заданном η). Часто оптимальным называют многочлен, полученный при $\eta = 1$ [7, 78, 138], но на практике задают $\eta = 0.9\ldots0.98$.

При $p \le 3$ достаточно просто построить s-стадийный метод Рунге–Кутты порядка p, имеющий заданный многочлен устойчивости (7.1). Задача усложняется дополнительными требованиями, к которым относятся: 1) устойчивость (либо ограниченность функций устойчивости) внутренних стадий; 2) минимизация вычислительных ошибок; 3) наличие стабилизированной вложенной формулы для получения оценки ошибки на одном шаге. Первые два из этих

требований удовлетворяются путем специального упорядочения внутренних шагов [34, 35, 75]. При $p \leq 3$ несложно также построить стабилизированную вложенную формулу порядка $p - 1$.

Однако даже методы, построенные с учетом перечисленных выше требований, не всегда хороши для решения жестких задач. При малой допустимой ошибке методы 3-го и 4-го порядков могут оказаться менее эффективными, чем методы 2-го порядка [127]. Объяснить снижение точности при повышении порядка явных стабилизированных методов позволяет анализ ошибки решения модельного уравнения Протеро–Робинсона. Применительно к явным методам Рунге–Кутты анализ уравнения Протеро–Робинсона и других простейших модельных уравнений был выполнен в [52, 54, 55], где были также предложены приемы, позволяющие повысить точность решения нелинейных жестких задач явными методами Рунге–Кутты.

7.2. Многочлены устойчивости

Нахождение оптимального многочлена устойчивости (7.1) сводится к определению $s - p$ параметров, обеспечивающих максимальную длину интервала устойчивости l при ограничениях (7.2). Обычно в качестве настраиваемых параметров используют вещественные корни многочлена. При $p = 1$ решение задачи выражается через многочлены Чебышёва [34, 75, 148]. При $p = 2$ также известно аналитическое решение, которое выражается через эллиптические функции, но на практике его вычисляют с помощью итерационной процедуры [32, 33]. При $p > 2$ аналитическое решение неизвестно, поэтому для нахождения многочленов применяют численные методы. Например, в [127] при $p = 6$ и $s \leq 320$ применялся алгоритм Ремеза, реализованный в системе Mathematica, причем вычисления приходилось выполнять с повышенной точностью (150 разрядов при $s = 40$, 350 при $s = 160$ и 600 разрядов при $s = 320$). О трудоемкости задачи свидетельствует также приведенный в [7] пример, когда нахождение оптимального многочлена при $p = 3$ и $s = 576$ итерационным методом потребовало четырех суток вычислений многопроцессорной рабочей станции. В [7] предложен более эффективный метод, который был реализован при $p = 3$, но он очень сложен.

В [62, 63] предложен простой метод, позволяющий найти близкие к оптимальным многочлены для всех значений p и s, используемых при построении стабилизированных схем Рунге–Кутты (он испытывался для p до 8 и s до 2500). Метод не требует повышенной точности и может быть без труда реализован в любой системе математических вычислений. Опишем этот метод.

Свойства оптимальных многочленов устойчивости исследовались в [7, 33, 78, 130, 138, 148]. Пусть r – число комплексных корней. Тогда $r = p$ для четных p и $r = p - 1$ для нечетных p. Все остальные корни вещественные и находятся внутри интервала устойчивости левее комплексных корней. Известно также, что при увеличении s длина интервала устойчивости растет пропорционально s^2. Упорядочим корни так, чтобы выполнялись неравенства

$0 > \mathrm{Re}\, z_1, \ldots, \mathrm{Re}\, z_p > z_{p+1} > z_{p+2} > \cdots > z_s > -l$.

Рассмотрим многочлен как функцию вещественной переменной. Оптимальный многочлен имеет $s-p$ экстремальных точек z_i^* — таких, что $0 > z_1^* > z_2^* > \ldots > z_{s-p}^* > -l$ и

$$R'(z_i^*) = 0, \quad R(z_i^*) = (-1)^{p+i-1}\eta, \quad i = 1, \ldots, s-p. \tag{7.3}$$

При нечетных p число локальных экстремумов исчерпывается этими точками. При четных p имеется дополнительная точка локального минимума z_0^*, удовлетворяющая неравенствам $0 > z_0^* > z_1^*$ и $0 < R(z_0^*) < \eta$. Таким образом, исходную задачу можно сформулировать как задачу нахождения многочлена (7.1), удовлетворяющего условиям (7.3).

При решении поставленной задачи будем исходить из предположения, что распределение корней $z_{p+1}, \ldots, z_s$ оптимального многочлена примерно такое же, как у соответствующих корней многочлена Чебышёва. Пусть $T_s(x) = \cos(s \arccos x)$ – многочлен Чебышёва степени s, корни которого

$$x_i = \cos\left(\frac{2i-1}{2s}\pi\right), \quad i = 1, \ldots, s. \tag{7.4}$$

Тогда наше предположение запишется в виде:

$$z_i - z_{p+1} \approx c(x_i - x_{p+1}), \quad i = p+2, \ldots, s,$$

где c – масштабный коэффициент. Заменив приближенное равенство точным, получим:

$$z_i = z_{p+1} + c(x_i - x_{p+1}), \quad c = \frac{z_s - z_{p+1}}{x_s - x_{p+1}}, \quad i = p+2, \ldots, s-1. \tag{7.5}$$

При заданном порядке p и известных корнях $z_{p+1}, \ldots, z_s$ многочлен устойчивости можно записать в виде:

$$R(z) = \left(1 + \sum_{i=1}^{p} d_i z^i\right) \prod_{i=1}^{s-p} (1 + \gamma_i z), \tag{7.6}$$

где

$$d_{i0} = 1/i!, \quad d_{0j} = 1, \quad \gamma_j = -1/z_{p+j}, \quad d_{ij} = d_{i,j-1} - \gamma_j d_{i-1,j}, \tag{7.7}$$
$$d_i = d_{i,s-p}, \quad i = 1, \ldots, p, \quad j = 1, \ldots, s-p.$$

Многочлен, задаваемый формулами (7.4)–(7.7), имеет два неизвестных параметра: z_{p+1} и z_s. Если эти параметры получены путем максимизации l при ограничениях (7.2), то будем называть такой многочлен *квазиоптимальным*. Первоначально мы решали задачу именно в такой постановке, однако вскоре убедились, что экстремальные точки полученных таким образом многочленов удовлетворяют условиям

$$R(z_1^*) = (-1)^p\eta, \quad R(z_{s-p}^*) = (-1)^{s-1}\eta, \quad |R(z_i^*)| < \eta, \quad i = 2, \ldots, s-p-1 \tag{7.8}$$

(у нас нет доказательства справедливости этих соотношений, но на практике они всегда выполнялись). Используя (7.8), можно сформулировать исходную задачу как задачу нахождения параметров z_{p+1} и z_s, обеспечивающих выполнение равенств

$$R(z_1^*) = (-1)^p\eta, \quad R(z_{s-p}^*) = (-1)^{s-1}\eta. \tag{7.9}$$

Для нахождения экстремальных точек z_1^* и z_{s-p}^* нет необходимости сканировать весь интервал устойчивости $[-l, 0]$, поскольку они находятся внутри интервалов, задаваемых неравенствами $0 > z_1^* > z_{p+1}$ и $z_{s-1} > z_{s-p}^* > z_s$. Более того, для них можно найти хорошие начальные приближения, используя тот факт, что экстремальные точки многочлена Чебышёва степени s задаются формулами

$$x_i^* = \cos(i\pi/s), \quad i = 1, \ldots, s-1. \tag{7.10}$$

В этом случае точкам z_1^* и z_{s-p}^* квазиоптимального многочлена соответствуют точки x_p^* и x_{s-1}^* многочлена Чебышёва. Поэтому начальные приближения для нахождения этих точек можно задать в виде:

$$z_{1,0}^* = z_{p+1} + c(x_p^* - x_{p+1}), \quad z_{s-p,0}^* = z_{p+1} + c(x_{s-1}^* - x_{p+1}), \quad c = \frac{z_s - z_{p+1}}{x_s - x_{p+1}},$$

где x_i, x_i^* находим из (7.4), (7.10).

Процедура нахождения параметров квазиоптимальных многочленов реализована с использованием режима «Оптимизация» ПО SimInTech. Задача решается путем минимизации невязки уравнений (7.9). Для очередного s находим оптимальные значения $\alpha = -z_{p+1}$ и $\beta = -z_s/s^2$, которые в дальнейшем используются как начальные приближения для следующего значения s. Весь расчет для заданных p, s и η занимает не более нескольких секунд работы персонального компьютера.

Для ряда значений p, s и η мы определили длину области устойчивости квазиоптимальных многочленов. Полученные результаты сравнивались с аналогичными результатами, приведенными в [7, 33, 129, 130] для многочленов, рассчитанных исходя из условия чебышёвского альтернанса. Сравнение показало, что сокращение длины области устойчивости квазиоптимальных многочленов (по сравнению с оптимальными) не превышает 0.5% при $p = 2$ и 1% при $p = 3$. Некоторые результаты приведены в табл. 7.1, где l – длина области устойчивости квазиоптимального, а l^* – оптимального многочлена при $\eta = 0.98$. Примеры оптимальных и квазиоптимальных многочленов приведены на рис. 7.1.

Таблица 7.1. Интервалы устойчивости квазиоптимальных и оптимальных многочленов при $\eta = 0.98$

p	s	l/s^2	l/l^*	p	s	l/s^2	l/l^*
2	5	0.771971	0.99535	2	9	0.800301	0.99662
2	45	0.812231	0.99923	3	6	0.442844	0.99842
3	15	0.48672	0.99589	3	48	0.494248	0.99384
3	243	0.495037	0.99506	4	8	0.309021	0.99621
4	16	0.340831	0.99190	4	50	0.350438	0.99029
6	20	0.208853	0.98851	8	28	0.151214	0.98603

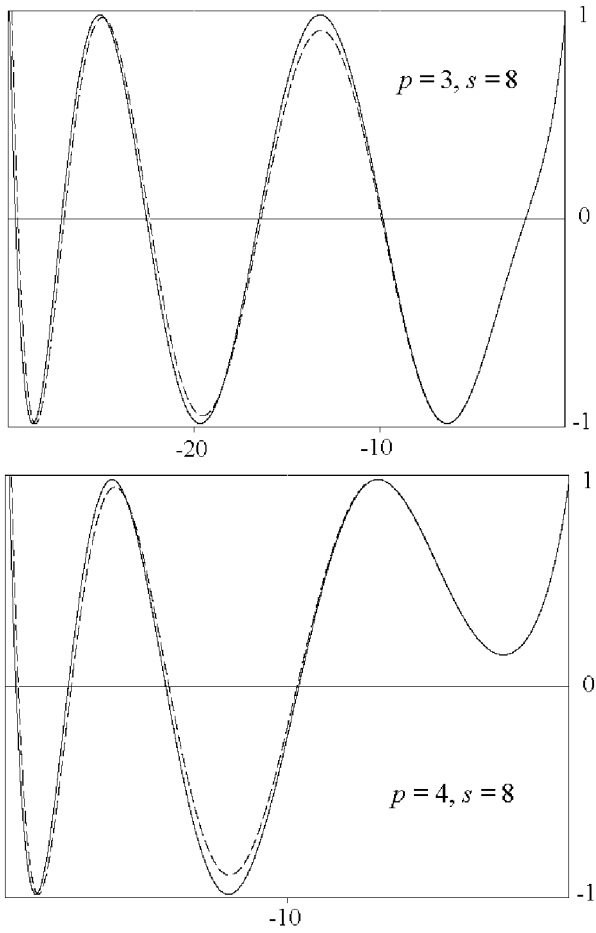


Рис. 7.1. Примеры оптимальных (сплошная линия) и квазиоптимальных (пунктир) многочленов устойчивости

Число различных значений s, для которых требуется рассчитать многочлены устойчивости, может быть велико, поэтому задача продолжает оставаться трудоемкой. Решение задачи можно еще более упростить, если найти простые приближенные зависимости z_{p+1} и z_s от s. В результате ряда вычислительных экспериментов были найдены простые и достаточно точные аппроксимации, которые задаются выражениями:

$$z_{p+1}(s) = \begin{cases} -\tilde{\alpha}, & s = p+1, \\ -(\alpha_0 + \alpha_{-2}s^{-2} + \alpha_{-4}s^{-4}), & s > p+1, \end{cases} \qquad (7.11)$$

$$z_s(s) = -(\beta_2 s^2 + \beta_0 + \beta_{-2}s^{-2}).$$

Для определения параметров, входящих в (7.11), достаточно найти z_{p+1} и z_s для небольшого числа различных значений s. Параметр $\tilde{\alpha}$ находим при $s = p + 1$.

Параметры α_0 и β_2 вычисляем при большом s (например, $s = 1000$) по формулам $\alpha_0 = -z_{p+1}$, $\beta_2 = -z_s/s^2$. Далее для нескольких значений s находим z_{p+1} и z_s (использовались значения $s = p + i$ при $i = 2, 3, 4, 6, 9, 14, 20$). На основе полученных результатов вычисляем α_{-2}, α_{-4}, β_0 и β_{-2}, применяя метод наименьших квадратов.

Данная процедура реализована в ПО SimInTech. С ее помощью нетрудно получить в компактном виде (7.11) параметры всех многочленов устойчивости для явного стабилизированного метода Рунге–Кутты заданного порядка. Параметры многочленов устойчивости, рассчитанные с помощью этой процедуры при $\eta = 0.9$ и $p \le 4$, приведены в табл. 7.2. Зная эти параметры, коэффициенты многочлена (7.6) находим по формулам (7.11), (7.4), (7.5), (7.7).

Таблица 7.2. Параметры формулы (7.11) для вычисления z_{p+1} и z_s при $\eta = 0.9$

p	$\tilde{\alpha}$	α_0	α_{-2}	α_{-4}	β_2	β_0	β_{-2}
2	5.4622	10.502	−55.907	99.904	0.78955	−1.4703	−1.0809
3	5.4766	11.515	−121.32	422.09	0.48567	−2.1508	−1.5194
4	5.6108	12.73	−224.9	1253	0.34653	−2.922	−1.4366

Выбор степени многочлена устойчивости осуществляем по длине интервала устойчивости $l = h\bar{\lambda}$, где $\bar{\lambda}$ – оценка сверху величины $\max\limits_i(-\lambda_i)$, λ_i – собственные числа матрицы Якоби. Воспользуемся приближенным равенством

$$l \approx \begin{cases} l_p, & s = p, \\ \beta_2 s^2 - \Delta l_p, & s > p, \end{cases}$$

откуда получаем

$$s = \begin{cases} p, & l \le l_p, \\ \left\lceil \sqrt{(l + \Delta l_p)/\beta_2} \right\rceil, & l > l_p, \end{cases}$$

где $\lceil x \rceil$ – минимальное целое, которое больше или равно x. Для методов от 2-го до 4-го порядка включительно мы задавали

$l_2 = 1.9$, $\Delta l_2 = 1.08$; $l_3 = 2.45$, $\Delta l_3 = 1.91$; $l_4 = 2.7$, $\Delta l_4 = 2.85$,

что при заданных в табл. 7.2 параметрах гарантирует устойчивость метода на интервале $-l \le z \le 0$.

7.3. Построение стабилизированных методов Рунге–Кутты 2-го порядка

Рассмотрим три способа построения методов, реализующих заданные многочлены устойчивости (7.6). Во всех них наряду с основной формулой строится вложенная формула порядка $p - 1$ с многочленом устойчивости

$$\hat{R}(z) = \left(1 + \sum_{i=1}^{p-1} d_i z^i\right) \prod_{i=1}^{s-p} (1 + \gamma_i z). \tag{7.12}$$

В результате выполнения одного шага будет получен вектор $\mathbf{y}_1$, аппроксимирующий точное решение $\mathbf{y}(t_0 + h)$ с порядком p, а также вектор $\hat{\mathbf{y}}_1$, порядок которого $p - 1$. Пусть $\mathbf{Y}_0 = \mathbf{y}_0$, $\mathbf{F}_0 = \mathbf{f}(t_0, \mathbf{y}_0)$, а через $\mathbf{Y}_i$, $i = 1, \ldots, s$ обозначим результаты выполнения всех стадий. Кроме этого, через $\hat{\mathbf{Y}}_i$, $i = 1, \ldots, s$ обозначим стадийные значения вложенного метода. Тогда $\mathbf{y}_1 = \mathbf{Y}_s$, $\hat{\mathbf{y}}_1 = \hat{\mathbf{Y}}_s$.

Способ (a). В соответствии с многочленом устойчивости (7.6) шаг численного решения можно представить в виде композиции $s - p + 1$ внутренних шагов: $s - p$ шагов метода Эйлера, размеры которых $h\gamma_I$, $i = 1, \ldots, s - p$, и одного шага размером hd_i с многочленом устойчивости

$$Q(z) = 1 + \sum_{i=1}^{p} d_i z^i. \tag{7.13}$$

Эйлеровы шаги обеспечивают стабилизацию метода, поэтому назовем их *стабилизирующими*. Шаг с многочленом устойчивости (7.13) обеспечивает порядок p метода, поэтому назовем его *порядковым*. В способе (a) сначала выполняются стабилизирующие шаги, а затем порядковый шаг. Аналогичный способ используется в программе DUMKA [32, 33, 75] с той разницей, что там эйлеровы шаги сгруппированы попарно. Такой порядок расчета позволяет обеспечить устойчивость всех внутренних стадий и простоту построения вложенной формулы.

Способ (b). Он отличается от предыдущего последовательностью выполнения внутренних шагов: сначала выполняем порядковый шаг, а затем стабилизирующие.

Способ (c). Этот способ предложен в [54] и основан на следующих соотношениях. Обозначим

$$R_p(z) = 1 + \sum_{i=1}^{p} z^i/i!, \quad \hat{R}_p(z) = 1 + \sum_{i=1}^{p-1} z^i/i!. \tag{7.14}$$

Многочлены (7.6) и (7.12) можно получить по рекуррентным формулам

$$\begin{aligned} &R_{p+i}(z) = R_{p+i-1}(z) + \alpha_i z\left(R_{p+i-1}(z) - \hat{R}_{p+i-1}(z)\right), \\ &\hat{R}_{p+i}(z) = \hat{R}_{p+i-1}(z) + \beta_i\left(R_{p+i-1}(z) - \hat{R}_{p+i-1}(z)\right), \\ &\beta_i = d_{p-1,i}\gamma_i/d_{p,i-1}, \quad \alpha_i = (1-\beta_i)\gamma_i, \quad i = 1, \ldots, s-p, \end{aligned} \tag{7.15}$$

где $d_{p-1,i}$, $d_{p,i-1}$ определяем согласно (7.7). Выполним предварительный шаг размером h таким образом, чтобы получить векторы переменных $\mathbf{Y}_p$, $\hat{\mathbf{Y}}_p$, соответствующие многочленам устойчивости (7.14), и векторы производных $\mathbf{F}_p$, $\hat{\mathbf{F}}_p$. Тогда формулам (7.15) будут соответствовать стадии

$$\begin{aligned} &\mathbf{Y}_{p+i} = \mathbf{Y}_{p+i-1} + \alpha_i h(\mathbf{F}_{p+i-1} - \hat{\mathbf{F}}_{p+i-1}), \quad \mathbf{F}_{p+i} = \mathbf{f}(t_0 + h, \mathbf{Y}_{p+i}), \\ &\hat{\mathbf{Y}}_{p+i} = \hat{\mathbf{Y}}_{p+i-1} + \beta_i(\mathbf{Y}_{p+i-1} - \hat{\mathbf{Y}}_{p+i-1}), \quad \hat{\mathbf{F}}_{p+i} = \hat{\mathbf{F}}_{p+i-1} + \beta_i(\mathbf{F}_{p+i-1} - \hat{\mathbf{F}}_{p+i-1}), \\ &i = 1, \ldots, s-p, \end{aligned} \tag{7.16}$$

в результате выполнения которых получим векторы $\mathbf{Y}_s$ и $\hat{\mathbf{Y}}_s$.

Во всех методах следует применять упорядочение корней [34, 35, 75], позволяющее уменьшить ошибки округления и ограничить рост функций устойчивости внутренних стадий. Рассмотрим методы 2-го порядка, которые в соответствии со способами их построения обозначим как SRK2a, SRK2b и SRK2c (SRK – Stabilized Runge–Kutta). Свободным параметром является абсцисса порядкового шага c_2. Для методов SRK2a и SRK2b принимаем

$$\bar{c}_2 = c_2 d_1, \quad g_i = \sum_{j=1}^{i} \gamma_j, \quad i = 1, \ldots, s-p,$$

а порядковый шаг имеет таблицу Бутчера

$$\begin{array}{c|cc} 0 & 0 & \\ \bar{c}_2 & \bar{c}_2 & \\ \hline \mathbf{Y} & \bar{b}_1 & \bar{b}_2 \\ \hat{\mathbf{Y}} & d_1 & \end{array}, \quad \bar{b}_2 = \frac{d_2}{\bar{c}_2}, \quad \bar{b}_1 = d_1 - \bar{b}_2,$$

что обеспечивает выполнение условий 2-го порядка основного и 1-го порядка вложенного метода, а также требуемые многочлены устойчивости.

Шаг метода SRK2a выполняем согласно формулам:

$$\mathbf{Y}_i = \mathbf{Y}_{i-1} + h\gamma_i \mathbf{F}_{i-1}, \quad \mathbf{F}_i = \mathbf{f}(t_0 + hg_i, \mathbf{Y}_i), \quad i = 1, \ldots, s-2,$$
$$\mathbf{Y}_{s-1} = \mathbf{Y}_{s-2} + h\bar{c}_2 \mathbf{F}_{s-2}, \quad \mathbf{F}_{s-1} = \mathbf{f}(t_0 + h(g_{s-2} + \bar{c}_2), \mathbf{Y}_{s-1}),$$
$$\mathbf{Y}_s = \mathbf{Y}_{s-2} + h(\bar{b}_1 \mathbf{F}_{s-2} + \bar{b}_2 \mathbf{F}_{s-1}), \quad \hat{\mathbf{Y}}_s = \mathbf{Y}_{s-2} + hd_1 \mathbf{F}_{s-2}.$$

При построении метода SRK2b вложенный метод реализуется на основе соотношения $\hat{R}_{p+i} = (1 + \gamma_i z)\hat{R}_{p+i-1}(z)$. Расчетные формулы одного шага запишутся в виде:

$$\mathbf{Y}_1 = \mathbf{Y}_0 + h\bar{c}_2 \mathbf{F}_0, \quad \mathbf{Y}_2 = \mathbf{Y}_0 + h(\bar{b}_1 \mathbf{F}_0 + \bar{b}_2 \mathbf{F}_1), \quad \hat{\mathbf{Y}}_2 = \mathbf{Y}_0 + hd_1 \mathbf{F}_0,$$
$$\mathbf{F}_1 = \mathbf{f}(t_0 + h\bar{c}_2, \mathbf{Y}_1), \quad \mathbf{F}_2 = \mathbf{f}(t_0 + hd_1, \mathbf{Y}_2), \quad \hat{\mathbf{F}}_2 = \mathbf{F}_0 + \frac{1}{c_2}(\mathbf{F}_1 - \mathbf{F}_0),$$
$$\mathbf{Y}_{i+2} = \mathbf{Y}_{i+1} + h\gamma_i \mathbf{F}_{i+1}, \quad \mathbf{F}_{i+2} = \mathbf{f}(t_0 + h(d_1 + g_i), \mathbf{Y}_{i+2}),$$
$$\hat{\mathbf{Y}}_{i+2} = \hat{\mathbf{Y}}_{i+1} + h\gamma_i \hat{\mathbf{F}}_{i+1}, \quad \hat{\mathbf{F}}_{i+2} = \hat{\mathbf{F}}_{i+1} + \frac{\gamma_i}{d_2}\left[\frac{1}{h}(\mathbf{Y}_{i+1} - \hat{\mathbf{Y}}_{i+1}) + d_1(\mathbf{F}_{i+1} - \hat{\mathbf{F}}_{i+1})\right],$$
$$i = 1, \ldots, s-2.$$

Первые стадии метода SRK2c выполняем двухстадийным методом 2-го порядка с вложенным методом 1-го порядка, что соответствует формулам

$$\mathbf{Y}_1 = \mathbf{Y}_0 + hc_2 \mathbf{F}_0, \quad \mathbf{Y}_2 = \mathbf{Y}_0 + h(b_1 \mathbf{F}_0 + b_2 \mathbf{F}_1), \quad \hat{\mathbf{Y}}_2 = \mathbf{Y}_0 + h\mathbf{F}_0,$$
$$\mathbf{F}_1 = \mathbf{f}(t_0 + hc_2, \mathbf{Y}_1), \quad \mathbf{F}_2 = \mathbf{f}(t_0 + h, Y_2), \quad \hat{\mathbf{F}}_2 = \mathbf{F}_0 + \frac{1}{c_2}(\mathbf{F}_1 - \mathbf{F}_0),$$
$$b_2 = \frac{1}{2c_2}, \quad b_1 = 1 - b_2.$$

Последующие стадии выполняются согласно формулам (7.16) при $p = 2$.

7.4. Упорядочение внутренних шагов (стадий)

Стабилизированные методы SRK2a и SRK2b построены в виде композиции нескольких методов, тогда основной шаг состоит из последовательности внутренних шагов, выполненных в общем случае разными методами. Подобным образом построены также и некоторые другие стабилизированные методы до 4-го порядка включительно [32, 33, 80, 129, 130]. При реализации такого подхода возникает задача выбора наилучшего порядка следования шагов. Если порядковый шаг – последний, то можно обеспечить устойчивость внутренних стадий и несложно построить стабилизированный вложенный метод. При упорядочении остальных шагов принимают во внимание их устойчивость и распространение ошибок округления. Такой способ реализован в программе DUMKA.

Исследуем точность стабилизированного композиционного метода, основной шаг которого состоит из m внутренних шагов длиной $h_i = \gamma_i h$, $i = 1, \ldots, m$ (h – длина основного шага). При решении уравнения Протеро–Робинсона главный член ошибки определяется функцией погрешности $e_2(z)$. Пусть $R_i(z)$ и $e_{2i}(z)$ – функция устойчивости и функция погрешности $e_2(z)$ i-го внутреннего метода. Тогда, используя (4.21), получим функцию погрешности композиционного метода в виде:

$$e_2(z) = \sum_{i=1}^{m-1} \gamma_i^2 e_{2i}(\gamma_j z) \prod_{j=i+1}^{m} R_j(\gamma_j z) + \gamma_m^2 e_{2m}(\gamma_m z). \tag{7.17}$$

В методах, построенных согласно способу (a), первые $m - 1$ шагов – стабилизирующие, последний шаг – порядковый. А в методах, построенных по способу (b), первый шаг – порядковый, последующие шаги – стабилизирующие. Поэтому в методах (a) погрешность порядкового шага входит без изменений в функцию погрешности композиционного метода. В методах (b) функция погрешности порядкового шага будет обнулена во всех точках стабилизации. А поскольку порядковый шаг – самый большой и имеет наибольшую ошибку, то способ (b) более предпочтителен с точки зрения точности численного решения. Приведем примеры, подтверждающие это.

Рассмотрим сначала стабилизацию в одной вещественной точке жесткого спектра $z_1 = h\lambda_1$. При $p \le 3$ нетрудно построить композиционный метод порядка p, шаг которого состоит из стабилизирующего эйлерова шага и порядкового p-стадийного шага. Для метода Эйлера $e_2(z) \equiv 1$. Используя также (7.17), получим формулы, отражающие поведение функции погрешности при $z = z_1 \to \infty$ для обоих вариантов:

(a) $$e_2(z_1) = \begin{cases} 1 + O(z_1^{-1}), & p = 1, \\ \dfrac{c_2}{p!} z_1^{p-1} + O(z_1^{p-2}), & p > 1, \end{cases}$$

(b) $$e_2(z_1) = z_1^{-2},$$

где c_2 – абсцисса порядкового шага. Полученные формулы показывают, что вариант (b) обладает полезным свойством: ошибка решения уравнения Протеро–Робинсона уменьшается при увеличении жесткости. Аналогичным свойством обладает метод SRK2c, который имеет $e_2(z_1) = c_2(z_1 + 2)z_1^{-2}$, а также другие методы, построенные по способу (c). Напротив, в варианте (a) при $p > 1$ ошибка растет с увеличением жесткости.

На рис. 7.2 приведены графики квазиоптимального многочлена устойчивости и функций погрешности методов SRK2a, SRK2b и SRK2c при $s = 10$, $\eta = 0.9$, $c_2 = 1/2$. Такие же зависимости для методов 3-го порядка приведены в [63] и показывают аналогичную качественную картину.

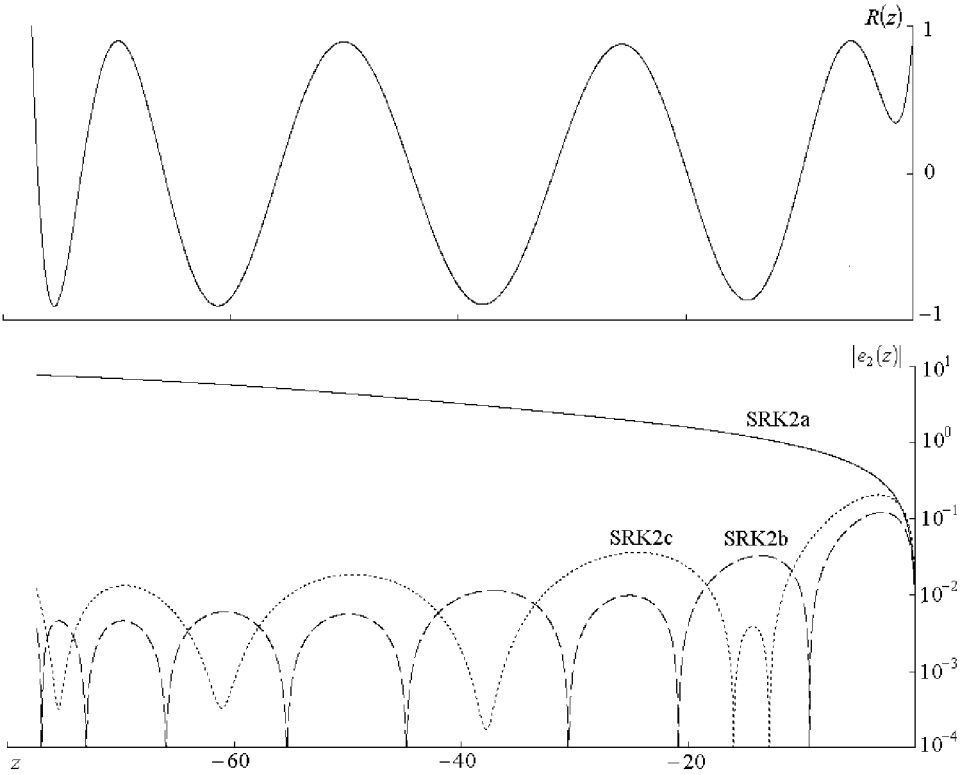


Рис. 7.2. Многочлен устойчивости и функции погрешности стабилизированных методов

Стабилизация в нескольких точках позволяет построить методы с расширенными областями устойчивости. При этом возникает задача наилучшего упорядочения корней многочлена устойчивости, определяющих стабилизирующие шаги (стадии). Если исходить из условия минимизации функции погрешности (7.17), то наилучшим является расположение корней в порядке возрастания их абсолютных значений (или уменьшения γ_i). Но в этом случае следование подряд больших по размеру шагов приводит к сильному расхож-

дению численного решения на первых стадиях, что неизбежно сказывается на окончательном результате. Поэтому следует определенным образом упорядочить шаги, чтобы избежать следования подряд больших шагов.

Рекуррентные алгоритмы такого упорядочения изложены в [34, 35]. Приведем один из них. Пусть оптимальное упорядочение N корней, первоначально расположенных в порядке возрастания абсолютных значений, задается списком $\mathbf{U}_N = (j_1, \ldots, j_N)$. Тогда упорядочение $2N$ корней получаем заменой числа j_i в этом списке парой чисел $2N + 1 - j_i, j_i$. А упорядочение $2N - 1$ корней получаем заменой j_i парой $2N - j_i, j_i$, но если $j_i = N$, то такой замены не производим. Зададим $N_1 = 1, \mathbf{U}_1 = (1)$. Чтобы упорядочить r корней, нужно последовательно упорядочить $N_2 = 2, N_3, \ldots, N_k = r$ корней, где $N_{i-1} = N_i/2$, если N_i четное, и $N_{i-1} = (N_i + 1)/2$, если N_i нечетное. Например, при $r = 11$ получаем

$N_1 = 1, \quad \mathbf{U}_1 = (1),$

$N_2 = 2, \quad \mathbf{U}_2 = (2, 1),$

$N_3 = 3, \quad \mathbf{U}_3 = (2, 3, 1),$

$N_4 = 6, \quad \mathbf{U}_6 = (5, 2, 4, 3, 6, 1),$

$N_5 = 11, \quad \mathbf{U}_{11} = (7, 5, 10, 2, 8, 4, 9, 3, 6, 11, 1).$

Поскольку для минимизации функции погрешности (7.17) выгодно, чтобы последние значения γ_i были малыми, мы испытали также варианты, в которых несколько последних корней сохраняло первоначальный порядок. Эксперименты показали, что преимущество имеет вариант, когда рассмотренным способом упорядочены $r = s - 2p$ корней, а порядок следования остальных стабилизирующих корней задается цифрами $j_i = i, i = r + 1, \ldots, r + p$. Такой алгоритм упорядочения применялся для всех рассмотренных методов.

В разделе 4.3 было показано, что в ряде случаев уменьшение c_2 позволяет уменьшить значения $e_2(z)$. Посмотрим, справедливо ли это для стабилизированных методов. В табл. 7.3 приведены значения

$$||e_2(z)|| = \max(|e_2(z)|, -l \le z \le 0)$$

при различных c_2 и s. Уменьшение c_2 позволяет уменьшить норму функции $e_2(z)$ примерно на порядок в методах SRK2a (при $s = 10$) и SRK2b. В методе SRK2c значение $||e_2(z)||$ пропорционально c_2. Во всех дальнейших экспериментах принимаем $c_2 = 1/20$ (задание меньших значений нежелательно, поскольку тогда некоторые коэффициенты метода принимают большие значения).

Таблица 7.3. Зависимости функций погрешности от c_2

c_2	$\|\|e_2(z)\|\|$ ($s = 10$)			$\|\|e_2(z)\|\|$ ($s = 100$)		
	SRK2a	**SRK2b**	**SRK2c**	**SRK2a**	**SRK2b**	**SRK2c**
1/2	7.7×10^{0}	1.2×10^{-1}	2.1×10^{-1}	9.2×10^{3}	1.1×10^{-1}	2.0×10^{-1}
1/20	5.4×10^{-1}	9.6×10^{-3}	2.1×10^{-2}	9.6×10^{3}	1.2×10^{-2}	2.0×10^{-2}
1/200	8.3×10^{-1}	8.0×10^{-3}	2.1×10^{-3}	9.6×10^{3}	9.2×10^{-3}	2.0×10^{-3}

Посмотрим, как влияет жесткость задачи на точность численного решения на примере линейной неавтономной задачи (6.23) при $T = 0.5$. Для определения числа стадий s необходимо иметь оценку $\bar{\lambda}$ величины $\max_i(-\lambda_i)$, где λ_i – собственные числа матрицы Якоби. Матрица Якоби задачи (6.23) имеет спектр собственных значений [10, §19]

$$\lambda_i = -4(N+1)^2 \sin^2\left(\frac{\pi i}{2(N+1)}\right), \quad i = 1, \ldots, N, \tag{7.18}$$

поэтому можно задать $\bar{\lambda} = 4(N+1)^2$. Для анализа влияния жесткости на точность решения мы решали эту задачу с постоянным размером шага $h = 0.05$ при различных значениях N. В табл. 7.4 приведены ошибки, вычисленные по формуле

$$err = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2}, \tag{7.19}$$

где y_i – точное, а $\tilde{y}_i$ – численное решение по i-й компоненте в конце интервала интегрирования. Видно, что ошибка метода SRK2a возрастает с увеличением жесткости, а ошибки двух других методов изменяются очень мало.

Таблица 7.4. Ошибки решения задачи (6.23)

N	s	Ошибка		
		SRK2a	SRK2b	SRK2c
5	4	9.71×10^{-6}	1.89×10^{-6}	6.80×10^{-6}
50	26	2.58×10^{-3}	2.05×10^{-6}	6.24×10^{-6}
500	253	7.24×10^{0}	2.27×10^{-6}	6.16×10^{-6}
1000	504	8.38×10^{1}	2.29×10^{-6}	6.16×10^{-6}

Вторая задача получена путем дискретизации уравнения Бюргерса $\partial u/\partial t = \nu\,\partial^2 u/\partial x^2 - u\,\partial u/\partial t$, $0 \le x \le 1$ (задача 4 из [32]). Выберем второе из приведенных в [32] решений этого уравнения: $u(t) = x/(1+t)$. Применяя метод прямых, получаем систему ОДУ

$$y_i' = \nu\frac{y_{i-1} - 2y_i + y_{i+1}}{\Delta x^2} - y_i\frac{y_{i+1} - y_{i-1}}{2\Delta x}, \quad y_i(0) = i\Delta x, \quad i = 1, \ldots, N,$$

$\Delta x = 1/(N+1)$, $y_0 = 0$, $y_{N+1} = 1/(1+t)$, $0 \le t \le 1$

с известным точным решением $y_i(t) = x_i/(1+t)$, $x_i = i/(N+1)$. При получении оценки $\bar{\lambda}$ для достаточно больших значений N и ν можно пренебречь нелинейным членом, тогда $\bar{\lambda} = 4\nu(N+1)^2$ (такая оценка успешно использовалась при $N \ge 50$ и $\nu \ge 0.005$). Ошибки при $N = 1000$ и $\nu = 0.01$ приведены в табл. 7.5 (прочерк означает, что решение получить не удалось). При $h = 0.001$ все три метода показывают очень близкие результаты. При увеличении размера шага ошибка метода SRK2a растет заметно быстрее, чем у других методов. Отметим, что такой быстрый рост ошибки обычно приводит к неустойчивости управления размером шага.

Таблица 7.5. Ошибки решения уравнения Бюргерса

h	s	Ошибка		
		SRK2a	SRK2b	SRK2c
0.001	8	2.87×10^{-8}	2.98×10^{-8}	2.93×10^{-8}
0.01	23	1.04×10^{-5}	2.86×10^{-6}	2.81×10^{-6}
0.1	72	1.43×10^{-1}	3.13×10^{-4}	3.17×10^{-4}
0.2	101	–	–	1.46×10^{-3}

Приведенные результаты показали преимущество способов (b) и (c) построения стабилизированных методов при решении жестких задач. Это преимущество объясняется тем, что стабилизирующие стадии выполняются в конечной точке шага интегрирования $t = t_n + h$ для метода SRK2c либо вблизи этой точки для метода SRK2b. То есть фактически эти методы обладают свойством жесткой точности.

7.5. Стабилизированные методы порядков 3 и 4

Стабилизированные методы 3-го порядка, построенные согласно способам (a), (b) и (c), приведены в [63]. Из них метод 3a оказался малоэффективным, а метод 3b – излишне сложным, что вызвано сложностью построения вложенной формулы. Поэтому остановимся на способе (c), который позволяет без особого труда строить эффективные стабилизированные методы повышенного порядка на основе вложенных пар, задающих формулы вычисления $\hat{\mathbf{Y}}_p$ и $\mathbf{Y}_p$. Пусть $\hat{\mathbf{F}}_p = \mathbf{f}(t_0 + h, \hat{\mathbf{Y}}_p)$, $\mathbf{F}_p = \mathbf{f}(t_0 + h, \mathbf{Y}_p)$ и выполняются условия

$$\hat{p} = p-1, \quad \hat{R}(z) = 1+\sum_{i=1}^{p-1} z^i/i!, \quad R(z) = \hat{R}(z)+1/p!.$$

Тогда стабилизирующие стадии выполняются по формулам (7.16). Подобные вложенные пары рассматривались в [55]. При $s = p \geq 3$ такие пары построить невозможно, а если задать $s > p$, то их можно построить на основе методов повышенного псевдостадийного порядка ($\overline{q} > 1$, см. раздел 4.3).

Для построения стабилизированных методов 3-го и 4-го порядков были выбраны пары

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1 & 1 & 0 & & \\ \hat{\mathbf{Y}}_3 & -1/2 & 2 & -1/2 & \\ \hline \mathbf{Y}_3 & 1/6 & 2/3 & -1/6 & 1/3 \end{array} \qquad (7.20)$$

и

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
1/3 & 1/3 & & & & & \\
2/3 & 2/3 & 0 & & & & \\
1 & 1 & 0 & 0 & & & \\
0 & -11/12 & 3/2 & -3/4 & 1/6 & & \\
\hat{\mathbf{Y}}_4 & 1/4 & -3 & 15/4 & -1 & 1 & \\
\hline
\mathbf{Y}_4 & -1/8 & 3/8 & 3/8 & -1/8 & 1/4 & 1/4
\end{array}
\quad . \tag{7.21}
$$

Оба метода (вложенный и основной) пары (7.20) имеют $e_2(z) \equiv 0$, а у пары (7.21) также и $e_3(z) \equiv e_{32}(z) \equiv 0$, что обеспечивает повышенную точность при решении жестких задач. После выполнения порядковых стадий, задаваемых таблицей (7.20) или (7.21), стабилизирующие стадии выполняем согласно (7.16). Стабилизированный метод 3-го порядка, построенный на основе пары (7.20), обозначим через SRK3c, а метод 4-го порядка на основе пары (7.21) – через SRK4c.

7.6. Двухшаговые стабилизированные методы 1-го порядка

Одношаговые стабилизированные методы подобны бегуну на длинную дистанцию, который, достигнув оптимальной скорости, останавливается, затем снова стартует и т. д. Очевидно, что такая тактика не принесет ему успеха, и бегущий с постоянной скоростью будет быстрее. В нашем случае «бегун с постоянной скоростью» – это многошаговый метод, стартующий с малым размером шага и наращивающий шаг до оптимального размера, который в дальнейшем поддерживается на всем интервале интегрирования. Другой недостаток рассмотренных одношаговых методов состоит в том, что наиболее эффективные из них, т. е. построенные по схемам (b) и (c), содержат неустойчивые внутренние стадии. В связи с этим вызывает интерес построение явных стабилизированных многошаговых методов.

Рассмотрим сначала двухшаговый одностадийный метод

$$\mathbf{y}_{n+1} = \mathbf{y}_n + b(\mathbf{y}_n - \mathbf{y}_{n-1}) + h_n(1 - b/w)\mathbf{f}(t_n, y_n), \quad w = h_n/h_{n-1}, \tag{7.22}$$

где h_n и h_{n-1} – величины текущего и предыдущего шагов. Исследуем устойчивость этого метода. Применяя его для решения модельного уравнения Далквиста $y' = \lambda y$, получаем разностную схему

$$y_{n+1} = R(z)y_n - by_{n-1}, \quad R(z) = 1 + b + (1 - b/w)z, \quad z = h_n\lambda,$$

характеристическое уравнение которой:

$$\mu^2 - R(z)\mu + b = 0. \tag{7.23}$$

Согласно определению V.1.1 из [75], точка z принадлежит области устойчивости многошагового метода, если все корни $\mu_i(z)$ его характеристического уравнения удовлетворяют неравенству $|\mu_i(z)| \le 1$ и при этом среди корней нет кратных, равных по модулю 1. Характеристическое уравнение можно использовать для анализа устойчивости, если его коэффициенты не изменяются от шага к шагу. Для линейных многошаговых методов это выполняется лишь в случае постоянного шага интегрирования. Поэтому при исследовании устойчивости в случае переменного шага (например, при возрастании величины шага) прибегают к предположению, что коэффициенты характеристического уравнения изменяются достаточно медленно.

Будем также исходить из этого предположения. Заметим, что коэффициенты уравнения (7.23) не изменяются, если на всех шагах

$b = \text{const}, \quad h_n = bh_{n-1} + c, \quad c = \text{const}.$

Для нас более интересен случай, когда $b = 1 - \varepsilon$, где ε – малое положительное число (не обязательно константа), а величина шага возрастает линейно. Именно в этом случае после выполнения достаточно большого числа шагов длина интервала устойчивости может быть сколь угодно большой. Устойчивость многошаговых методов исследовалась также путем проведения вычислительных экспериментов. В частности, режим линейного возрастания величины шага исследовался на модельной задаче

$y_i' = -i/N, \quad y_i(0) = 1, \quad i = 1, \ldots, N$

при числе переменных и числе шагов до 1000.

При принятом нами предположении устойчивость метода (7.22) определяется двумя корнями

$$\mu_{1,2}(z) = \frac{1}{2}\left(R(z) \pm \sqrt{R^2(z) - 4b}\right)$$

уравнения (7.23). Область устойчивости задается неравенствами

$$|\mu_1(z)| \le 1, \quad |\mu_2(z)| \le 1. \tag{7.24}$$

Воспользовавшись алгебраическим критерием устойчивости дискретных систем (см., например, [76]), запишем условие (7.24) в более удобном виде:

$$1 - R(z) + b \ge 0, \quad 1 - b \ge 0, \quad 1 + R(z) + b \ge 0, \tag{7.25}$$

откуда

$$-(1 - b/w)z \ge 0, \quad 1 - b \ge 0, \quad 2(1 + b) + (1 - b/w)z \ge 0. \tag{7.26}$$

Пусть l_n – длина интервала устойчивости $[-l_n, 0]$ на очередном шаге интегрирования. Чтобы было $l_n \ge 2$, потребуем выполнения неравенства

$$0 \le b < \min(w, 1). \tag{7.27}$$

В этом случае первые два неравенства (7.26) всегда выполняются при $z \le 0$, а из третьего неравенства получим

$$l_n = 2\frac{w(1+b)}{w-b}. \tag{7.28}$$

Оценим, насколько быстро может возрастать величина шага без потери устойчивости. При $w > 1$ из (7.27), (7.28) получим:

$$l_n < 4\frac{w}{w-1}. \tag{7.29}$$

Пусть предыдущий шаг был максимального (исходя из условия устойчивости) размера, т. е. выполнялось соотношение $l_{n-1} = |h_{n-1}\lambda|$. Чтобы сохранилась устойчивость на очередном шаге, должно выполняться неравенство

$$w \le l_n/l_{n-1}, \tag{7.30}$$

т. е. размер шага должен расти не быстрее, чем длина интервала устойчивости. Из (7.29), (7.30) получим

$$w < \frac{l_{n-1}+4}{l_{n-1}} = \frac{|z_{n-1}|+4}{|z_{n-1}|}, \quad z_{n-1} = h_{n-1}\lambda. \tag{7.31}$$

Максимальный прирост величины шага обеспечивается при $w = l_n/l_{n-1}$. Подставив это выражение в (7.31), получим

$$l_n < l_{n-1} + 4 = |z_{n-1}| + 4. \tag{7.32}$$

Неравенство (7.32) накладывает ограничение на скорость увеличения длины интервала устойчивости. Выполним первый шаг методом Эйлера и зададим последующие шаги такими, чтобы длина интервала возрастала наиболее быстро. В предельном случае получим $b = 1$, $h_n = h_{n-1} + 2h_0$, что соответствует последовательности шагов

$$\mathbf{y}_1 = \mathbf{y}_0 + h_0\mathbf{f}(t_0, \mathbf{y}_0), \quad \mathbf{y}_{n+1} = 2\mathbf{y}_n - \mathbf{y}_{n-1} + 2h_0\mathbf{f}(t_0 + n^2h_0, \mathbf{y}_0), \quad n = 1, 2, \ldots \tag{7.33}$$

Рассмотрим s шагов (7.33) как один шаг s-стадийного одношагового метода. Можно показать, что функцией устойчивости такого метода является смещенный многочлен Чебышёва, экстремальные значения которого на интервале $[-2s^2, 0]$ равны 1 и -1.

Обычно значение w, т. е. фактически величина нового шага $h_n = wh_{n-1}$, определяется на основе оценки ошибки, полученной на предыдущем шаге. Однако при решении жестких задач такая процедура становится неустойчивой, что проявляется в резких колебаниях величины шага, которые могут достигать нескольких порядков. Чтобы избавиться от таких колебаний, следует контролировать не только точность, но и устойчивость разностной схемы, как предлагалось в [39]. Для этого следует принять

$$w = \min(w_{er}, w_{st}), \tag{7.34}$$

где значение w_{er} получено исходя из оценки ошибки, а w_{st} – исходя из устойчивости. Пусть $\hat{z}_{n-1} = h_{n-1}\hat{\lambda}$, где $\hat{\lambda}$ – оценка наибольшего по модулю отрицательного собственного значения матрицы Якоби. Тогда из (7.31) получим

$$w_{\text{st}} = \frac{\left|\hat{z}_{n-1}\right| + \Delta l}{\left|\hat{z}_{n-1}\right|}, \quad \Delta l < 4, \tag{7.35}$$

где Δl – максимальное приращение длины интервала устойчивости на одном шаге интегрирования.

Параметр b метода найдем из (7.28):

$$b = w\frac{l_n - 2}{l_n + 2w}, \quad l_n = \max\left(2, w\left|\hat{z}_{n-1}\right|\right). \tag{7.36}$$

Соотношения (7.34)–(7.36) гарантируют выполнение неравенств (7.26) при $-l_n \le z \le 0$.

На примере простейшего двухшагового одностадийного метода, задаваемого формулами (7.22), (7.34)–(7.36), мы рассмотрели основной принцип построения методов такого типа. Однако для практической реализации более удобен двухстадийный метод, который позволяет оценивать наибольшее по модулю собственное значение и переходить к схеме второго порядка в случае нежесткой задачи.

Двухшаговый двухстадийный метод первого порядка зададим в виде:

$$\begin{aligned}
&\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + h_n\mathbf{f}_n, \quad \mathbf{f}_n = \mathbf{f}(t_n, \mathbf{y}_n),\\
&\mathbf{y}_{n+1} = \mathbf{y}_n + b_0(\mathbf{y}_n - \mathbf{y}_{n-1}) + h_n\left[b_1\mathbf{f}_n + b_2(\hat{\mathbf{f}}_{n+1} - \mathbf{f}_n)\right],\\
&\hat{\mathbf{f}}_{n+1} = \mathbf{f}(t_n + h_n, \hat{\mathbf{y}}_{n+1}).
\end{aligned} \tag{7.37}$$

Характеристическое уравнение этого метода:

$\mu^2 - R(z)\mu + b_0 = 0$, $R(z) = 1 + b_0 + b_1 z + b_2 z^2$.

Из условия первого порядка имеем $b_1 = 1 - b_0/w$, тогда условие устойчивости (7.25) запишется в виде:

$S_1(z) = -(1 - b_0/w)z - b_2 z^2 \ge 0$,

$S_2(z) = 1 - b_0 \ge 0$,

$S_3(z) = 2(1 + b_0) + (1 - b_0/w)z + b_2 z^2 \ge 0$.

Чтобы было $l_n \ge 2$, потребуем выполнения неравенства

$$0 \le b_0 < \min(1, w). \tag{7.38}$$

Приняв $S_1(-l_n) = 0$, получим длину области устойчивости

$$l_n = (1 - b_0/w)/b_2. \tag{7.39}$$

Чтобы обеспечить положительность $S_3(z)$ на интервале $[-l_n, 0]$, примем

$$\min S_3(z) = S_3(-l_n/2) = \frac{8w^2(1+b_0)b_2 - (w-b_0)^2}{4w^2 b_2} = \varepsilon > 0. \tag{7.40}$$

Удобно задать $\varepsilon = 3(1 - b_0)/2$, тогда из (7.39), (7.40) получим параметры метода (7.37) в виде:

$$b_0 = w\frac{l_n - 2}{l_n + 14w}, \quad b_1 = 1 - \frac{b_0}{w}, \quad b_2 = \frac{b_1}{l_n}, \tag{7.41}$$

а при $l_n = 2$ имеем одношаговый метод второго порядка

$$\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + h_n\mathbf{f}_n, \quad \mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h_n}{2}(\mathbf{f}_n + \hat{\mathbf{f}}_{n+1}). \tag{7.42}$$

Из (7.38), (7.41), (7.30) получим $w < (l_{n-1} + 16)/l_{n-1}$, а задав $w = l_n/l_{n-1}$, получим ограничение на увеличение длины интервала устойчивости на одном шаге в виде $l_n < l_{n-1} + 16$.

Опишем алгоритм одного шага двухстадийного метода. На основе оценки ошибки предыдущего шага $\delta\mathbf{y}_n = \mathbf{y}_n - \hat{\mathbf{y}}_n$ определяем w_{er}. Вычисляем:

$$\hat{z}_{n-1} = h_{n-1}\hat{\lambda}_n, \quad w_{\text{st}} = \frac{\left|\hat{z}_{n-1}\right| + \Delta l}{\left|\hat{z}_{n-1}\right|},$$
$$w = \min(w_{\text{er}}, w_{\text{st}}), \quad h_n = wh_{n-1}, \quad l_n = \max\left(2, w\left|\hat{z}_{n-1}\right|\right), \tag{7.43}$$

где $\hat{\lambda}_n$ – текущая оценка наибольшего по модулю отрицательного собственного значения матрицы Якоби, $\Delta l < 16$ (в [63] рекомендуется задать $\Delta l = 8$). Определяем параметры метода по формулам (7.41) и выполняем шаг интегрирования согласно (7.37).

7.7. Трехшаговый стабилизированный метод 2-го порядка

Анализ двухшаговых схем первого порядка показывает, что при возрастании длины интервала устойчивости l_n формула шага интегрирования приближается к формуле линейной экстраполяции $\mathbf{y}_{n+1} = \mathbf{y}_n + w(\mathbf{y}_n - \mathbf{y}_{n-1})$, отличаясь от нее малыми по величине членами. Это объясняет наличие в формуле интегрирования разности назад первого порядка $\mathbf{y}_n - \mathbf{y}_{n-1}$. Можно предположить, что для построения аналогичных схем второго порядка в формуле интегрирования должна присутствовать разность назад второго порядка. Исходя из этих соображений, метод второго порядка должен быть, по крайней мере, трехшаговым.

Рассмотрим двухстадийный трехшаговый метод

$$\begin{aligned}
&\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + h_n\mathbf{f}_n, \quad \mathbf{f}_n = \mathbf{f}(t_n, \mathbf{y}_n),\\
&\mathbf{y}_{n+1} = \mathbf{y}_n + b_0(\mathbf{y}_n - \mathbf{y}_{n-1}) + c_0(\mathbf{y}_n - (1+w_2)\mathbf{y}_{n-1} + w_2\mathbf{y}_{n-2}) +\\
&\quad + h_n\left[b_1\mathbf{f}_n + b_2(\hat{\mathbf{f}}_{n+1} - \mathbf{f}_n) + c_1\mathbf{f}_{n-1} + c_2w_1(\hat{\mathbf{f}}_n - \mathbf{f}_{n-1})\right],\\
&\hat{\mathbf{f}}_{n+1} = \mathbf{f}(t_n + h_n, \hat{\mathbf{y}}_{n+1}), \quad w_1 = h_n/h_{n-1}, \quad w_2 = h_{n-1}/h_{n-2}.
\end{aligned} \tag{7.44}$$

Условия, обеспечивающие второй порядок этого метода, запишутся в виде:

$$\frac{1}{w_1}b_0 + b_1 + c_1 = 1,$$
$$\frac{1+w_2}{2w_1^2 w_2}c_0 - \frac{1}{2w_1^2}b_0 - \frac{1}{w_1}c_1 + b_2 + c_2 = \frac{1}{2}. \tag{7.45}$$

Исследуем устойчивость метода (7.44). Используя его для решения уравнения $y' = \lambda y$, получаем:

$$y_{n+1} = R(z)y_n - Q(z)y_{n-1} + Py_{n-2},$$
$$R(z) = 1 + b_0 + c_0 + b_1 z + b_2 z^2, \quad Q(z) = b_0 + (1 + w_2)c_0 - c_1 z - c_2 z^2, \tag{7.46}$$
$$P = w_2 c_0, \quad z = h_n \lambda.$$

Предполагая, что коэффициенты разностной схемы (7.46) изменяются достаточно медленно, получим характеристическое уравнение в виде:

$$\mu^3 - R(z)\mu^2 + Q(z)\mu - P = 0.$$

Область устойчивости задается неравенствами относительно корней:

$$|\mu_i(z)| \leqslant 1, \quad i = 1, 2, 3. \tag{7.47}$$

Воспользовавшись алгебраическим критерием устойчивости, получим условие (7.47) в виде:

$$S_1(z) = 1 - R(z) + Q(z) - P \geqslant 0, \tag{7.48а}$$
$$S_2(z) = 3 - R(z) - Q(z) + 3P \geqslant 0, \tag{7.48б}$$
$$S_3(z) = 3 + R(z) - Q(z) - 3P \geqslant 0, \tag{7.48в}$$
$$S_4(z) = 1 + R(z) + Q(z) + P \geqslant 0, \tag{7.48г}$$
$$S_5(z) = 1 - P^2 + R(z)P - Q(z) \geqslant 0. \tag{7.48д}$$

Подставляя в (7.48) значения $R(z)$, $Q(z)$, P, имеем:

$$S_1(z) = -(b_2 + c_2)z^2 - (b_1 + c_1)z,$$
$$S_2(z) = (c_2 - b_2)z^2 + (c_1 - b_1)z + 2(1 - b_0 - c_0 + w_2 c_0),$$
$$S_3(z) = (b_2 + c_2)z^2 + (b_1 + c_1)z + 4(1 - w_2 c_0),$$
$$S_4(z) = (b_2 - c_2)z^2 + (b_1 - c_1)z + 2(1 + b_0 + c_0 + w_2 c_0),$$
$$S_5(z) = (w_2 c_0 b_2 + c_2)z^2 + (w_2 c_0 b_1 + c_1)z + (1 - w_2 c_0)(1 - b_0 - c_0 + w_2 c_0).$$

Из соображений симметрии примем $S_i(-l_n) = S_i(0)$, $i = 1, \ldots, 5$, тогда

$$b_1 - l_n b_2 = 0, \quad c_1 - l_n c_2 = 0. \tag{7.49}$$

Из (7.45), (7.49) получаем:

$$c_1 = \frac{1}{2}\left(\frac{1+w_2}{w_1 w_2}c_0 - \frac{l_n + 2w_1}{w_1 l_n}b_0 - w_1\frac{l_n - 2}{l_n}\right),$$
$$b_1 = 1 - \frac{b_0}{w_1} - c_1, \quad b_2 = \frac{b_1}{l_n}, \quad c_2 = \frac{c_1}{l_n}. \tag{7.50}$$

Коэффициенты b_0, c_0 остаются свободными. Для их нахождения примем

$$S_3(0) = K_1 S_3(-l_n/2), \quad S_4(0) = K_2 S_4(-l_n/2), \quad K_1 > 1, \quad K_2 > 1, \tag{7.51}$$

тогда из (7.50), (7.51) получим:

$$c_0 = w_1 w_2 \frac{K_1 l_n (1+w_1)(K_2 l_n - 8K_2 + 8) + 32 w_1 (K_1 - 1)(3K_2 - 4)}{K_1 l_n (1+w_2)(K_2 l_n + 8 w_1 w_2 (K_2 - 1)) + 32 w_1^2 w_2^2 (K_1 - 1)(3K_2 - 4)},$$
$$b_0 = w_1 \left(1 - 16(1 - w_2 c_0)\frac{K_1 - 1}{K_1 l_n}\right). \tag{7.52}$$

Зададим $K_1 = K_2$, тогда

$$S_5(0) = K S_5(-l_n/2), \quad K = K_1 = K_2. \tag{7.53}$$

Пусть коэффициенты метода вычислены по формулам (7.50), (7.52), (7.53) и $w_1 > 0$, $w_2 > 0$, $l_n \geq 2$, $K > 1$. Покажем, что если выполняются неравенства

$$w_1 - b_0 > 0, \tag{7.54а}$$
$$1 - b_0 - c_0 + w_2 c_0 > 0, \tag{7.54б}$$
$$1 - w_2 c_0 > 0, \tag{7.54в}$$
$$1 + b_0 + c_0 + w_2 c_0 > 0, \tag{7.54г}$$

то будут выполняться и все условия (7.48) при $-l_n \leq z \leq 0$. Выполнение условия (7.48а) следует из (7.54а), а выполнение условий (7.48в, г, д) – из (7.54в, г), (7.51), (7.53). Из (7.51), (7.54г) имеем $b_1 - c_1 > 0$, откуда следует также, что если выполняется неравенство (7.54б), то выполняется и условие (7.48б). Таким образом, выполнение неравенств (7.54) гарантирует устойчивость разностной схемы (7.46) при $-l_n \leq z \leq 0$.

Анализ формул (7.52) показывает, что при уменьшении l_n коэффициент b_0 также уменьшается. При малых значениях l_n коэффициент b_0 может принимать большие по модулю отрицательные значения, вследствие чего нарушается неравенство (7.54г) и не выполняется условие (7.48г). Чтобы этого не происходило, поступим следующим образом. Найдем максимальное значение $l_n = l_{n0}$, при котором $b_0 = 0$:

$$l_{n0} = \frac{1}{2a}\left(-b + \sqrt{b^2 - 4ac}\right), \quad a = K^2(1 + w_2),$$
$$b = 8K(K-1)\left(w_1 w_2^2 (3 + 2w_1) - w_2(2 - w_1) - 2\right), \tag{7.55}$$
$$c = -32 w_1 w_2 (K-1)\left(w_1 w_2 K + 4(K-1)(1 + 2w_2)\right).$$

При $l_n \geq l_{n0}$ будем вычислять b_0 и c_0 по формулам (7.52), а в противном случае принимаем:

$$b_0 = 0, \quad c_0 = \frac{l_n - 2}{(l_{n0} - 2) w_2}\left(1 - \frac{K l_{n0}}{16(K-1)}\right) N. \tag{7.56}$$

Формула (7.56) построена так, что при $l_n = 2$ получаем метод (7.42).

В [49] приведена модификация, позволяющая избежать вычислений по формулам (7.55), (7.56). Для этого следует принять в (7.52)

$$K_1 = \frac{7K(l_n - 2) + 8}{7(l_n - 1)}, \quad K_2 = \frac{3K(l_n - 2) + 4}{3(l_n - 1)}. \tag{7.57}$$

При больших значениях l_n могут нарушаться неравенства (7.54а, б, в), что приводит к невыполнению условий (7.48а, б, в, д). Чтобы эти условия выполнялись, необходимо, как и в методах 1-го порядка, ограничивать рост величины шага. Примем:

$$w_1 = \frac{l_n}{l_n - \Delta l}, \quad w_2 = \frac{l_n - \Delta l}{l_n - 2\Delta l}. \tag{7.58}$$

Подставляя (7.52), (7.58) в (7.54а, б, в), находим, что приращение длины интервала устойчивости на одном шаге Δl должно удовлетворять неравенству

$$\Delta l < \frac{16(K-1)}{3K}.$$

Опишем алгоритм одного шага. На основе оценки ошибки предыдущего шага $\delta\mathbf{y}_n = \mathbf{y}_n - \hat{\mathbf{y}}_n$ определяем w_{er}. Вычисляем:

$$\begin{gathered} \hat{z}_{n-1} = h_{n-1}\hat{\lambda}_n, \quad w_{\text{st}} = \frac{\left|\hat{z}_{n-1}\right| + \Delta l}{\left|\hat{z}_{n-1}\right|}, \quad w_1 = \min(w_{\text{er}}, w_{\text{st}}), \\ h_n = w_1 h_{n-1}, \quad l_n = \max\left(2, w_1\left|\hat{z}_{n-1}\right|\right), \end{gathered} \tag{7.59}$$

где $\hat{\lambda}_n$ – оценка наибольшего по модулю отрицательного собственного значения. Если $l_n \geq l_{n0}$, где l_{n0} определяем из (7.55), то вычисляем b_0 и c_0 по формулам (7.52) при $K_1 = K_2 = K$, а иначе по формулам (7.56). Другой вариант: вычисляем b_0 и c_0 по формулам (7.57), (7.52). Вычисляем остальные коэффициенты по формулам (7.50) и выполняем шаг интегрирования согласно (7.44). В [63] рекомендуется задавать $K = 16$, $\Delta l = 2$. Обозначим этот метод через SEM2 (Stabilized Explicit Multistep of order 2).

7.8. Оценивание границы жесткого спектра

Для практического применения методов с расширенными областями устойчивости необходимо знать оценку границы жесткого спектра матрицы Якоби, которая используется для выбора числа стадий в методах Рунге–Кутты либо для определения размера шага по формулам (7.43), (7.59) в многошаговых методах. Как правило, получение такой оценки в методах чебышёвского типа осуществляется в самостоятельной процедуре, которая не входит в состав основного алгоритма. На наш взгляд, необходимость в такой процедуре является недостатком известных методов. Если жесткий спектр заметно изменяется на траектории решения, то вычисление матрицы Якоби и оценивание границы ее спектра может потребовать большого объема дополнительных вычислений.

В то же время известен способ оценивания наибольшего по модулю собственного значения по результатам шага интегрирования, не требующий дополнительных вычислений матрицы Якоби и правой части. Этот способ основан на степенном методе и используется в нелинейных (адаптивных) методах, рассмотренных в следующей главе, а также в методах с контролем устойчивости [39]. Рассмотрим его применительно к явным стабилизированным методам.

Пусть в результате выполнения стадий очередного шага получены значения переменных $\mathbf{Y}$ и $\hat{\mathbf{Y}}$, аппроксимирующие решение при $t = t_n$, и соответствующие значения производных $\mathbf{F}$ и $\hat{\mathbf{F}}$. Если правая часть достаточно гладкая, то справедливо приближенное равенство

$$\delta\mathbf{F}_n \approx \mathbf{J}_n \delta\mathbf{Y}_n, \quad \delta\mathbf{Y}_n = \mathbf{Y} - \hat{\mathbf{Y}}, \quad \delta\mathbf{F}_n = \mathbf{F} - \hat{\mathbf{F}}, \tag{7.60}$$

где $\mathbf{J}_n$ – матрица Якоби, вычисленная при $t = t_n$. Соотношение (7.60) позволяет воспользоваться степенным методом для получения грубой оценки наибольшего по модулю собственного значения матрицы Якоби. Покомпонентные оценки можно получить в виде

$$\hat{\lambda}_n^i = \delta F_n^i / \delta Y_n^i, \tag{7.61}$$

где δf_n^i, δy_n^i – i-е компоненты соответствующих векторов. Удобнее всего применять такие оценки в методах SRK, построенных по способу (c). В этом случае в качестве $\hat{\mathbf{Y}}$, $\mathbf{Y}$, $\hat{\mathbf{F}}$ и $\mathbf{F}$ в (7.60) используем векторы $\hat{\mathbf{Y}}_p$, $\mathbf{Y}_p$, $\hat{\mathbf{F}}_p$ и $\mathbf{F}_p$, полученные в результате выполнения порядковых стадий, а последующие стабилизирующие стадии (7.16) выполняем с использованием полученной на этом шаге оценки наибольшего собственного значения. В многошаговых методах в качестве $\hat{\mathbf{Y}}$, $\mathbf{Y}$, $\hat{\mathbf{F}}$ и $\mathbf{F}$ используем $\hat{\mathbf{y}}_n$, $\mathbf{y}_n$, $\hat{\mathbf{f}}_n$ и $\mathbf{f}_n$, а полученную оценку собственного значения применяем на следующем шаге.

Поскольку нам нужна только одна оценка, то можно усреднить все покомпонентные оценки (7.61), используя, например, метод наименьших квадратов. В этом случае получим:

$$\hat{\lambda}_n = \frac{(\mathbf{D}\delta\mathbf{F}_n)^{\mathrm{T}} \mathbf{D}\delta\mathbf{Y}_n}{(\mathbf{D}\delta\mathbf{Y}_n)^{\mathrm{T}} \mathbf{D}\delta\mathbf{Y}_n}, \tag{7.62}$$

где $\mathbf{D}$ – диагональная матрица масштабных коэффициентов, заданных, например, по формуле (2.10) (масштабирование необходимо, когда компоненты решения сильно различаются по величине). Другой способ – выбрать минимальное значение среди покомпонентных оценок:

$$\hat{\lambda}_n = \min_i (\delta F_n^i / \delta Y_n^i). \tag{7.63}$$

Применение формул (7.62), (7.63) в составе алгоритма интегрирования показало, что оценки по формуле (7.62) часто сильно смещены в сторону нуля, а формула (7.63) может давать большой разброс оценок. Поэтому был выбран комбинированный способ, в котором вычисляются сглаженные по независимой переменной покомпонентные оценки, среди которых затем выбирается

минимальная. Сглаживание производится экспоненциально взвешенным методом наименьших квадратов. Рекуррентные расчетные формулы для получения покомпонентных оценок запишутся в виде:

$$\hat{\lambda}^i_n = \hat{\lambda}^i_{n-1} + \frac{\delta Y^i_n}{d^i_n}(\delta F^i_n - \hat{\lambda}^i_{n-1}\delta Y^i_n), \quad d^i_n = \gamma d^i_{n-1} + (\delta Y^i_n)^2,$$
$$\hat{\lambda}^i_0 = 0, \quad d^i_0 = 0, \quad 0 < \gamma \le 1. \tag{7.64}$$

Значение γ следует выбирать как компромисс между точностью (чем ближе γ к 1, тем точнее оценка) и скоростью оценивания (чем меньше γ, тем быстрее может изменяться оценка при изменении собственного значения). Если максимальное по модулю собственное число мало изменяется на траектории решения, то следует задавать $\gamma = 0.9 \ldots 1$, а при быстром изменении можно задать $\gamma = 0.5$. Окончательную оценку получаем как

$$\hat{\lambda}_n = K_\lambda \min_i(\hat{\lambda}^i_n), \quad K_\lambda \ge 1$$

(в приведенных далее экспериментах задаем $K_\lambda = 1.1$).

7.9. Численные эксперименты

Приведем результаты решения жестких задач явными стабилизированными методами с автоматическим выбором шага. Линейную неавтономную задачу (6.23) мы решали при $T = 0.5$, $N = 1000$, $Atol = Rtol$ и $h_0 = Tol$ (для SEM2 задавали $h_0 = 1/\bar{\lambda}$, что обеспечивает устойчивость первого шага). Согласно (7.18), можно принять $\bar{\lambda} = \max(-\lambda_i) \approx 4(N+1)^2 = 4.008\times10^6$, поэтому нет необходимости использовать приведенную выше процедуру оценивания границы жесткого спектра. Ошибку вычисляем по формуле (7.19). Полученные результаты приведены в табл. 7.6, где *Nf* – число вычислений правой части, а результаты решателей RKC, DUMKA3 и ROCK4 при $Tol = 10^{-6}, 10^{-8}$ взяты из [127]. Низкая эффективность решателей SRK2a, DUMKA3 и ROCK4 объясняется тем, что в них порядковый шаг выполняется после стабилизирующих.

Таблица 7.6. Результаты решения задачи (6.23) с переменным шагом

Метод	$Tol = 10^{-3}$		$Tol = 10^{-6}$		$Tol = 10^{-8}$	
	Ошибка	***Nf***	**Ошибка**	***Nf***	**Ошибка**	***Nf***
SRK2a	9.39×10^{-4}	19313	7.50×10^{-7}	48550	5.13×10^{-9}	87991
SRK2c	3.46×10^{-5}	3582	3.64×10^{-8}	16962	2.75×10^{-10}	53672
SRK3c	5.79×10^{-6}	3988	7.86×10^{-8}	7020	1.22×10^{-9}	13531
SRK4c	2.97×10^{-6}	4956	5.38×10^{-8}	6962	1.54×10^{-9}	10775
SEM2	3.72×10^{-5}	2834	4.09×10^{-7}	4926	1.18×10^{-10}	29902
RKC	–	–	2.00×10^{-7}	7909	7.77×10^{-9}	15164
DUMKA3	–	–	2.10×10^{-10}	454036	3.90×10^{-11}	628339
ROCK4	–	–	5.04×10^{-7}	184002	6.43×10^{-9}	298927

Результаты решения задач BRUSS, HIRES и VDPOL приведены в табл. 7.7, 7.8 и 7.9. Задаем $Atol = Rtol = Tol$ для BRUSS и VDPOL и $Atol = 10^{-4} \times Tol$ для HIRES, а начальный шаг принимаем в виде $h_0 = 10^{-6} \times T \times Tol^{1/p}$. В тех случаях, когда применялась процедура оценивания наибольшего собственного числа, приводим значение γ в (7.64), в противном случае задаем предварительную оценку $\bar{\lambda}$, а в таблице ставим прочерк. В задаче BRUSS (см. раздел 3.6) линейные члены доминируют, что позволяет, используя (7.18), принять такую оценку в виде $\bar{\lambda} = 4\alpha(N + 1)^2$. А поскольку задача BRUSS автономная и «почти линейная», то и метод SRK2a не уступает методу SRK2c при решении этой задачи, хотя при решении других (нелинейных или неавтономных) задач он заметно хуже. Для HIRES задаем $\bar{\lambda} = 220$, а для VDPOL – $\bar{\lambda} = 3 \times 10^6$.

Таблица 7.7. Результаты решения задачи BRUSS

Метод	γ	*scd*	*Nf*	*scd*	*Nf*	*scd*	*Nf*
		$Tol = 10^{-2}$		$Tol = 10^{-3}$		$Tol = 10^{-4}$	
SRK2a	–	2.12	3510	2.99	5307	3.93	9359
SRK2c	–	2.09	3510	2.98	5304	3.93	9359
SRK2c	0.95	2.09	3793	2.98	5704	3.93	9584
SEM2	–	0.64	1482	1.14	1926	2.09	3150
SEM2	0.9	1.75	2298	2.96	3666	3.46	7434
		$Tol = 10^{-3}$		$Tol = 10^{-4}$		$Tol = 10^{-5}$	
SRK3c	–	2.99	4933	3.86	6136	4.86	8794
		$Tol = 10^{-5}$		$Tol = 10^{-6}$		$Tol = 10^{-7}$	
SRK4c	–	4.89	6975	5.91	9193	6.92	12265

Таблица 7.8. Результаты решения задачи HIRES

Метод	γ	*scd*	*Nf*	*scd*	*Nf*	*scd*	*Nf*
		$Tol = 10^{-2}$		$Tol = 10^{-3}$		$Tol = 10^{-4}$	
SRK2a	–	2.47	7560	3.32	10886	4.24	16860
SRK2c	0.5	1.92	1758	2.89	2257	3.90	4737
SRK2c	–	1.97	2372	2.89	3226	3.93	6718
SEM2	0.5	2.92	2024	4.25	4376	4.08	8474
SEM2	–	0.93	1246	1.70	2590	2.59	6744
		$Tol = 10^{-3}$		$Tol = 10^{-4}$		$Tol = 10^{-5}$	
SRK3c	–	3.97	5390	3.82	5412	4.48	6224
		$Tol = 10^{-5}$		$Tol = 10^{-6}$		$Tol = 10^{-7}$	
SRK4c	–	4.81	10367	6.60	9495	7.19	10211

Таблица 7.9. Результаты решения задачи VDPOL

Метод	γ	*scd*	*Nf*	*scd*	*Nf*	*scd*	*Nf*
		$Tol = 10^{-2}$		$Tol = 10^{-3}$		$Tol = 10^{-4}$	
SRK2a	–	1.94	239307	3.84	346114	3.80	517960
SRK2c	0.5	1.91	27993	3.23	25992	4.14	41583
SRK2c	–	2.13	32198	3.10	31567	4.19	58228
SEM2	0.5	3.09	16455	2.87	21717	3.17	38435
SEM2	–	1.10	13621	1.43	20357	1.97	35991
		$Tol = 10^{-5}$		$Tol = 10^{-6}$		$Tol = 10^{-7}$	
SRK3c	–	5.91	102062	6.50	90706	6.91	110871
		$Tol = 10^{-7}$		$Tol = 10^{-8}$		$Tol = 10^{-9}$	
SRK4c	–	7.42	384949	8.44	327827	9.50	333178

Для методов SRK3c и SRK4c приводим результаты при значениях *Tol*, обеспечивающих минимальные для данной задачи значения *Nf* (если *Tol* увеличить, то вычислительные затраты будут заметно больше либо решение вообще не будет получено). При низкой точности наименьшие затраты имеет метод SEM2, но из-за сложной зависимости ошибки от размера шага и жесткости задачи он не всегда обеспечивает требуемую точность. Многошаговые стабилизированные методы переменного порядка могли бы быть весьма эффективными, но построение таких методов встречает серьезные трудности.

Явные адаптивные методы для жестких и колебательных задач

ГЛАВА 8

Построение явных методов для жестких задач может быть основано на использовании оценок наибольших по модулю собственных значений матрицы Якоби для настройки формулы интегрирования на решаемую задачу. Такие оценки нетрудно получить по результатам стадий явного метода, что практически не требует дополнительных вычислений. Подобный подход был реализован в явных нелинейных методах [6, 83, 100, 125, 149, 151]. Среди них отметим метод Фаулера–Вартена [100], в котором впервые были четко выделены этапы получения оценок собственных значений и последующего использования этих оценок для стабилизации расчетной схемы. Критический анализ предлагавшихся ранее явных нелинейных методов выполнен в [14], где показано, что эти методы пригодны для решения с низкой точностью весьма узкого класса задач.

Более эффективные явные нелинейные методы были предложены в [44–51, 154]. Параметры этих методов настраиваются на решаемую задачу, используя для этого полученные в результате выполнения предварительных стадий оценки собственных значений. Таким образом, метод адаптируется к задаче, поэтому эти методы были названы адаптивными. Оценки собственных значений могут быть комплексными, что позволяет применять явные адаптивные методы для решения как жестких, так и колебательных задач. Явные адаптивные методы реализованы в SimInTech, эксплуатация которого показала, что при решении многих жестких задач эти методы не уступают неявным методам, а иногда и превосходят их.

8.1. Построение явных адаптивных методов Рунге–Кутты

Рассмотрим адаптивные методы, построенные на основе явных стадий Рунге–Кутты, выполняемых по формулам

$$\mathbf{Y}_i = \mathbf{y}_0 + h\sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j, \quad \mathbf{F}_i = \mathbf{f}(t_0 + c_i h, \mathbf{Y}_i), \quad i = 1, \ldots, s. \tag{8.1}$$

Пусть $a_{i,i-1} \neq 0$, $i = 2, \ldots, s$. Тогда можно задать векторы $\mathbf{u}_1, \ldots, \mathbf{u}_s$ в виде линейных комбинаций $\mathbf{F}_1, \ldots, \mathbf{F}_s$ таким образом, что при решении линейной системы $\mathbf{y}' = \mathbf{J}\mathbf{y}$ будут выполняться соотношения

$$\mathbf{u}_i = h^{-1}(h\mathbf{J})^i\mathbf{y}_0, \quad i = 1, \ldots, s. \tag{8.2}$$

Векторы $\mathbf{u}_1, \ldots, \mathbf{u}_s$ однозначно определяются из уравнений

$$\mathbf{u}_1 + \sum_{j=1}^{i-1} \beta_{ij}\mathbf{u}_{j+1} = \mathbf{F}_i, \quad i = 1, \ldots, s, \tag{8.3}$$

где β_{ij} – коэффициенты внутренних функций устойчивости

$$R_i(z) = 1 + \sum_{j=1}^{i-1} \beta_{ij} z^j. \tag{8.4}$$

Для явных методов эти функции запишутся в виде:

$$\left[R_1(z), \ldots, R_s(z)\right]^{\mathrm{T}} = \mathbf{e} + \sum_{i=1}^{s-1} z^i \mathbf{A}^i \mathbf{e}, \quad \mathbf{e} = (1, \ldots, 1)^{\mathrm{T}},$$

откуда $\beta_{21} = c_2$, $\beta_{31} = c_3$, $\beta_{32} = a_{32}c_2$, $\beta_{41} = c_4$, $\beta_{42} = a_{42}c_2 + a_{43}c_3$, $\beta_{43} = a_{43}a_{32}c_2$, ... Для нелинейной задачи с гладкой функцией $\mathbf{f}$ и матрицей Якоби $\mathbf{J}$ равенства (8.2) выполняются приближенно.

Соотношения (8.2) позволяют воспользоваться степенным методом для получения вектора покомпонентных оценок наибольшего по модулю собственного значения матрицы $h\mathbf{J}$ в виде

$$\mathbf{z}_1 = \mathbf{u}_s / \mathbf{u}_{s-1} \tag{8.5}$$

(предполагаем покомпонентное выполнение всех операций с векторами).

Шаг интегрирования выполняем согласно формуле

$$\mathbf{y}_1 = \mathbf{y}_0 + h\left(\sum_{i=1}^{s-1} \mathbf{u}_i / i! + \mathbf{d}_s \mathbf{u}_s\right), \tag{8.6}$$

где $\mathbf{d}_s$ – вектор настраиваемых параметров. Учитывая (8.5), эту же формулу можно записать в виде:

$$\mathbf{y}_1 = \mathbf{y}_0 + h\left(\sum_{i=1}^{s-2} \mathbf{u}_i / i! + \mathbf{d}_{s-1} \mathbf{u}_{s-1}\right), \tag{8.7}$$

где $\mathbf{d}_{s-1} = \mathbf{e} / (s-1)! + \mathbf{d}_s \mathbf{z}_1$. В дальнейшем будем использовать формулу (8.7), которая содержит меньше членов и менее чувствительна к вычислительным ошибкам. Компоненты вектора $\mathbf{d}_{s-1}$ задаются из условия обеспечения необходимых свойств устойчивости и точности и зависят от соответствующих компонент вектора $\mathbf{z}_1$. Таким образом, в адаптивных методах осуществляется покомпонентная настройка коэффициентов формулы интегрирования.

Рассмотрим скалярное уравнение

$$y' = \lambda y, \quad y(t_0) = y_0. \tag{8.8}$$

Используя адаптивный метод, получим точную оценку собственного значения $z_1 = h\lambda$ и численное решение в виде $y_1 = R(z_1)y_0$, где

$R(z) = 1 + z + \cdots + z^{s-2}/(s-2)! + d_{s-1}z^{s-1}$.

Выберем настраиваемый параметр d_{s-1} из условия $R(z_1) = Q(z_1)$, где $Q(z)$ – заданная функция, которую назовем *скалярной функцией устойчивости*. Тогда d_{s-1} можно вычислить по рекуррентным формулам

$$d_0 = Q(z_1), \quad d_{i+1} = (d_i - 1/i!)/z_1, \quad i = 0, 1, \ldots, s-2. \tag{8.9}$$

В случае системы уравнений эти действия будем выполнять покомпонентно, тогда для линейной автономной системы $\mathbf{y}' = \mathbf{J}\mathbf{y}$ получим $\mathbf{y}_1 = \mathbf{R}(h\mathbf{J})\mathbf{y}_0$, где

$$\mathbf{R}(\mathbf{Z}) = \mathbf{I} + \mathbf{Z} + \cdots + \mathbf{Z}^{s-2}/(s-2)! + \mathbf{D}_{s-1}\mathbf{Z}^{s-1}$$

– *матричная функция устойчивости*, $\mathbf{D}_{s-1} = \mathrm{diag}(\mathbf{d}_{s-1})$.

Для обычных методов Рунге–Кутты скалярная и матричная функции устойчивости совпадают, но в случае нелинейных (в том числе и адаптивных) методов они различаются. В качестве скалярной функции устойчивости адаптивного метода можно задать практически любую функцию, но функция устойчивости линейного метода может быть только дробно-рациональной, а если метод явный, то это может быть только многочлен.

Простейший двухстадийный адаптивный метод запишется в виде:

$$\mathbf{y}_1 = \mathbf{y}_0 + h\mathbf{d}_1\mathbf{F}_1, \quad \mathbf{d}_1 = (Q(\mathbf{z}_1) - \mathbf{e})/\mathbf{z}_1, \quad \mathbf{z}_1 = (\mathbf{F}_2 - \mathbf{F}_1)/(c\mathbf{F}_1), \tag{8.10}$$
$$\mathbf{F}_1 = \mathbf{f}(t_0, \mathbf{y}_0), \quad \mathbf{F}_2 = \mathbf{f}(t_0 + hc, \mathbf{y}_0 + hc\mathbf{F}_1),$$

где $Q(\mathbf{z})$ означает результат покомпонентного применения функции $Q(z)$ к вектору $\mathbf{z}$. Скалярную функцию устойчивости следует задавать так, чтобы обеспечить точность нежестких и устойчивость жестких компонент. Например, можно потребовать, чтобы метод (8.10) точно решал уравнение (8.8), откуда получим

$$Q(z) = \exp(z). \tag{8.11}$$

Формулы (8.10), (8.11) задают метод, который является стартовым в двухшаговом методе Фаулера–Вартена [100]. Методы такого типа рассматривались также в [14, 83], их принято называть *экспоненциальными*.

Можно задать в качестве $Q(z)$ функцию устойчивости некоторого неявного метода Рунге–Кутты. Например, задав $Q(z) = 1/(1-z)$, получим

$$\mathbf{y}_1 = \mathbf{y}_0 + h\frac{c\mathbf{F}_1^2}{(1+c)\mathbf{F}_1 - \mathbf{F}_2}. \tag{8.12}$$

Используя разные дробно-рациональные аппроксимации экспоненты, можно получить множество различных формул вида (8.12). Методы, основанные на таких формулах, называют *дробно-рациональными*, они рассматривались в [6, 125, 149, 151]. Как правило, в этих методах процедура оценивания наибольшего собственного значения замаскирована.

Остановимся на выборе функции $Q(z)$. Исходя из точности решения уравнения (8.8), лучший выбор определяется условием экспоненциальной подгонки. Однако в этом случае возникает опасность переполнения при больших значениях компонент вектора $\mathbf{z}_1$. Поэтому в методе Фаулера–Вартена формула (8.11)

используется только при $z < 0$, а при $z \geq 0$ принимается $Q(z) = 1 + z + z^2/2$.

Анализ результатов множества экспериментов позволил сформулировать следующие требования к скалярной функции устойчивости $Q(z)$.

1. Для нежестких компонент, соответствующих небольшим по модулю значениям z, функция $Q(z)$ должна аппроксимировать экспоненту с порядком не ниже желаемого порядка метода (обычно s или $s - 1$). Аппроксимации порядка $s + 1$ и выше не дают преимущества (за исключением некоторых скалярных уравнений).
2. Необходимо обеспечить быстрое затухание жестких компонент, соответствующих большим отрицательным значениям z, что приводит к условию $|Q(z)| << 1$ для таких значений. При этом порядок согласованности с экспонентой не имеет значения.
3. Следует обеспечить качественно правильное поведение неустойчивых компонент, соответствующих большим положительным значениям z. Для таких значений должно выполняться условие $Q(z) >> 1$. Этому условию не удовлетворяют дробно-рациональные методы, которые обеспечивают быстрое затухание не только жестких, но и неустойчивых компонент.
4. Для любых z_1 найденное в соответствии с (8.9) значение d_{s-1} должно быть ограничено, поскольку в противном случае возможны снижение точности и потеря устойчивости при неточном определении z_1. Такое ограничение можно принять в виде $0 \leq d_{s-1} \leq k/(s-1)!$, где k немного больше 1. По этой же причине желательно обеспечить выполнение условия $d_{s-1} \to 0$ при $z_1 \to +\infty$. Желательно также, чтобы функция $Q(z)$ была непрерывной и монотонной.

Таким образом, явный адаптивный метод Рунге–Кутты с постоянным размером шага задается коэффициентами c_i, a_{ij} и скалярной функцией устойчивости $Q(z)$. При интегрировании с автоматическим выбором размера шага, кроме основной формулы (8.7), используется вложенная формула вычисления $\hat{\mathbf{y}}_1$, позволяющая получить оценку ошибки как норму вектора $\mathbf{y}_1 - \hat{\mathbf{y}}_1$. Примем вложенную формулу в виде

$$\hat{\mathbf{y}}_1 = \mathbf{y}_0 + h\left(\sum_{i=1}^{s-3} \mathbf{u}_i/i! + \hat{\mathbf{d}}_{s-2}\mathbf{u}_{s-2}\right),$$

где компоненты вектора $\hat{\mathbf{d}}_{s-2}$ вычисляются по формулам (8.9) с использованием скалярной функции устойчивости вложенного метода $\hat{Q}(z)$. К функции $\hat{Q}(z)$ предъявляются такие же требования, как и к $Q(z)$. Дополнительно потребуем, чтобы $\hat{Q}(z)$ и $Q(z)$ различались при малых по модулю и положительных значениях z (иначе будет ослаблен контроль ошибки для нежестких и неустойчивых компонент).

Приведем функции $Q(z)$, которые можно использовать при построении адаптивных методов для заданных значений s:

$$s=2,\quad Q(z)=\begin{cases}1+z+z^2/2, & |z|\le 1,\\ 1/(1-z), & z<-1,\\ 5/2, & z>1.\end{cases} \tag{8.13а}$$

$$s=3,\quad Q(z)=\begin{cases}1+z+\dfrac{z^2}{2}+\dfrac{z^3}{6}, & |z|\le 1.6,\\ 0, & z<-1.6,\\ 1+\dfrac{167}{75}z, & z>1.6.\end{cases} \tag{8.13б}$$

$$s=4,\quad Q(z)=\begin{cases}1+z+\dfrac{z^2}{2}+\dfrac{z^3}{6}+\dfrac{z^4}{24}, & |z|\le 1.6,\\ -0.43264z^{-1}, & z<-1.6,\\ 1+z+\dfrac{131}{150}z^2, & z>1.6.\end{cases} \tag{8.13в}$$

8.2. Сходимость адаптивных методов

В [45, 46] были предложены явные адаптивные методы, стадии которых выполняются согласно формулам

$$\mathbf{F}_1=\mathbf{f}(t_0,\mathbf{y}_0),\quad \mathbf{Y}_2=\mathbf{y}_0+h\beta\mathbf{F}_1,\quad \mathbf{F}_2=\mathbf{f}(t_0+h\beta,\mathbf{Y}_2), \tag{8.14}$$
$$\mathbf{Y}_i=\mathbf{y}_0+h[(\beta-\alpha)\mathbf{F}_1+\alpha\mathbf{F}_{i-1}],\quad \mathbf{F}_i=\mathbf{f}(t_0+h\beta,\mathbf{Y}_i),\quad i=3,\ldots,s.$$

Тогда $\mathbf{u}_1=\mathbf{F}_1$, $\mathbf{u}_i=(\mathbf{F}_i-\mathbf{F}_{i-1})/(\beta\alpha^{i-2})$, $i=2,\ldots,s$. Исходя из классической теории, оптимальное значение β в этих методах равно 2/3. В этом случае при $s\ge 3$ метод может иметь третий порядок. Однако эксперименты показали, что для жестких задач значение $\beta=1$ более предпочтительно.

Исследуем точность методов со стадиями (8.14) на примере задачи

$$x'=10^6(e^{-t}-x)-e^{-t},\quad y'=-(x+y)/2,\quad x(0)=y(0)=1,\quad 0\le t\le 1, \tag{8.15}$$

решение которой $x(t)=y(t)=e^{-t}$. При $\alpha=10^{-3}$ и $h=1/40$ вычислим максимальные ошибки e_x и e_y каждой переменной. Определим также оценки порядков сходимости p_x и p_y, для чего используем ошибки, полученные при $h=1/40$ и $h=1/80$. Результаты для двух значений β приведены в табл. 8.1. При $s=2$ получаем нулевой порядок переменной y, поэтому двухстадийные методы непригодны для решения жестких задач.

Рассмотрим более подробно первое уравнение в (8.15). Его можно представить в более общем виде как уравнение Протеро–Робинсона

$$x'=\lambda(x-\varphi(t))+\varphi'(t),\quad x(t_0)=x_0, \tag{8.16}$$

которое при $x_0=\varphi(t_0)$ имеет решение $x(t)=\varphi(t)$. Будем решать это уравнение адаптивным методом. Обозначим $z=h\lambda$. При $s\ge 3$ оценка (8.5) будет точной, т. е. получим $z_1=z=h\lambda$. Из эквивалентности формул (8.6) и (8.7) следует, что при

заданных $Q(z)$ и β все методы с числом стадий не менее трех дадут одинаковый результат. Приведенные в табл. 8.1 результаты это подтверждают. Таким образом, достаточно рассмотреть трехстадийный метод.

Таблица 8.1. Результаты решения задачи (8.15)

β	s	e_x	p_x	e_y	p_y
2/3	2	2.46×10^{-2}	1.00	6.04×10^{-1}	−0.01
	3	1.03×10^{-4}	1.99	6.01×10^{-3}	1.01
	4	1.03×10^{-4}	1.99	2.51×10^{-5}	2.01
	5	1.03×10^{-4}	1.99	6.99×10^{-5}	3.01
1	2	2.46×10^{-2}	1.00	6.04×10^{-1}	−0.01
	3	1.23×10^{-8}	0.99	5.98×10^{-3}	1.03
	4	1.23×10^{-8}	0.99	1.27×10^{-5}	2.01
	5	1.23×10^{-8}	0.99	1.25×10^{-5}	2.00

Один шаг решения уравнения (8.16) трехстадийным адаптивным методом запишется в виде:

$$x_1 = x_0 + h(F_1 + d_2\,(F_2 - F_1)/\beta), \tag{8.17}$$
$$F_1 = \lambda(x_0 - \varphi_0) + \varphi_0', \quad F_2 = \lambda(x_0 + h\beta F_1 - \varphi_\beta) + \varphi_\beta',$$

где $\varphi_0 = \varphi(t_0)$, $\varphi_\beta = \varphi(t_0 + h\beta)$. Будем считать, что уравнение жесткое, т. е. $z << -1$. Для таких z примем $Q(z) = 0$, тогда $d_2 = -z^{-1} - z^{-2}$. Подставив это выражение в (8.17), получим:

$$x_1 = \varphi_0 + \frac{\varphi_\beta - \varphi_0}{\beta} + \left[\frac{(\varphi_\beta - \varphi_0) - h(\varphi_\beta' - \varphi_0')}{\beta} - h\varphi_0'\right] z^{-1} - h\frac{(\varphi_\beta' - \varphi_0')}{\beta} z^{-2}.$$

Численное решение не зависит от начального условия, поэтому глобальная ошибка не накапливается и равна локальной ошибке.

Выражение для ошибки численного решения получим в виде:

$$\varphi(t_0 + h) - x_1 = \left[(1-\beta) + (2-\beta)z^{-1} + 2z^{-2}\right]\frac{h^2}{2}\varphi_0'' + \\ + \left[(1-\beta^2) + \beta(3-\beta)z^{-1} + 3\beta z^{-2}\right]\frac{h^3}{6}\varphi_0''' + O(h^4). \tag{8.18}$$

Из (8.18) видно, что при $\beta = 2/3$ ошибка пропорциональна h^2, что объясняет второй порядок сходимости переменной x в табл. 8.1. Эту ошибку можно значительно уменьшить, если задать $\beta = 1$ (тогда она будет пропорциональна $h^2 z^{-1} = h/\lambda$). В этом случае ошибка переменной x очень мала, хотя и имеет первый порядок. В результате при $s \geq 3$ доминирует ошибка переменной y, которая и определяет реальный порядок метода. Таким образом, при любом β реальный порядок при решении жестких задач методами, основанными на стадиях (8.14), не может быть выше второго.

Найдем оптимальные параметры схем (8.14). Из приведенных в табл. 8.1 результатов видно, что значение $\beta = 1$ имеет некоторое преимущество, при этом практический интерес представляют методы с числом стадий 3 или 4. Значение коэффициента α практически не влияет на точность и устойчивость решения линейных задач. Однако при интегрировании жестких нелинейных систем большое отклонение от истинного решения на предварительных стадиях может привести к недостоверности полученных оценок собственных значений и неустойчивости численного решения. Поэтому α следует задавать достаточно малым, например 10^{-6}. Для многих жестких задач такое значение вполне приемлемо, однако для некоторых задач существенное преимущество дает более точная настройка.

Расхождение решения на предварительных стадиях характеризуется внутренними функциями устойчивости

$$R_1(z) = 1, \quad R_2(z) = 1 + \beta z, \quad R_3(z) = 1 + \beta z + \alpha\beta z^2, \qquad (8.19)$$
$$R_4(z) = 1 + \beta z + \alpha\beta z^2 + \alpha^2\beta z^3, \quad \ldots$$

Выбирем α из условия минимизации значений этих функций в точках жесткого спектра, задаваемых вектором $\mathbf{z}_1$. Для скалярного уравнения лучший выбор $\alpha = -1/z_1$, тогда функции (8.19) поочередно принимают значения 1 и $1 + \beta z_1$. Для системы уравнений будем ориентироваться на наихудший случай, т. е. наибольшее по модулю значение среди отрицательных компонент $\mathbf{z}_1$. Используя вектор $\mathbf{z}_1$ предыдущего шага (обозначим его $\mathbf{z}_{1\text{old}}$), можно задать α как минимальное значение среди положительных компонент вектора $-(\mathbf{z}_{1\text{old}} h/h_{\text{old}})^{-1}$. Следует ограничить α сверху и снизу, задав, например, $\alpha_{\max} = 0.5$, $\alpha_{\min} = 10^{-12}$.

Влияние выбора параметров α и β на точность и устойчивость численного решения продемонстрируем на примере задачи

$$\begin{aligned} y_1' &= y_2 - \mu y_1(y_1^2 + y_2^2 - 1), \quad y_1(0) = 0, \\ y_2' &= -y_1 - \mu y_2(y_1^2 + y_2^2 - 1), \quad y_2(0) = 1, \quad 0 \le t \le 2\pi, \end{aligned} \qquad (8.20)$$

которая имеет собственные значения $\lambda_{1,2} = -\mu \mp \sqrt{\mu^2 - 1}$ и решение $y_1(t) = \sin t$, $y_2(t) = \cos t$. При больших μ задача жесткая и существенно нелинейная. Ожидаемое оптимальное значение α для этой задачи:

$$\alpha^* = -(h\lambda_1)^{-1} = \left[h\left(\mu + \sqrt{\mu^2 - 1}\right)\right]^{-1}.$$

В табл. 8.2 приведена евклидова норма ошибки в конце интервала при $s = 3$, $h = 2\pi/400$ и различных μ, α и β (прочерк соответствует расхождению численного решения). Из приведенных результатов видно, что значения $\alpha = \alpha^*$ и $\beta = 1$ действительно являются оптимальными. Аналогичные результаты получены и при $s = 4$.

Таблица 8.2. Ошибки решения задачи (8.20)

μ	β = 2/3			β = 1		
	α = α*/2	α = α*	α = 2α*	α = α*/2	α = α*	α = 2α*
10^3	3.94×10^{-3}	3.88×10^{-3}	3.78×10^{-3}	1.10×10^{-3}	1.10×10^{-3}	1.12×10^{-3}
10^4	–	5.61×10^{-2}	–	1.19×10^{-3}	1.27×10^{-3}	1.51×10^{-3}
10^5	–	–	–	8.38×10^{-4}	1.29×10^{-3}	–
10^6	–	–	–	–	1.30×10^{-3}	–
10^7	–	–	–	–	1.37×10^{-3}	–

8.3. Адаптивный метод порядка 2 для нежестких и 1 для жестких задач

Рассмотрим метод, построенный на основе стадий (8.14) при $s = 3$. Примем $\beta = 1$. Это значение является оптимальным для жестких задач, поскольку обеспечивает свойство жесткой точности. Предварительные стадии такого метода имеют вид:

$$\mathbf{F}_1 = \mathbf{f}(t_0, \mathbf{y}_0),\quad \mathbf{Y}_2 = \mathbf{y}_0 + h\mathbf{F}_1,\quad \mathbf{F}_2 = \mathbf{f}(t_0 + h, \mathbf{Y}_2),$$
$$\mathbf{Y}_3 = \mathbf{Y}_2 + h\alpha(\mathbf{F}_2 - \mathbf{F}_1),\quad \mathbf{F}_3 = \mathbf{f}(t_0 + h, \mathbf{Y}_3),$$

а основную и вложенную формулы примем в виде

$$\mathbf{y}_1 = \mathbf{Y}_2 + h\mathbf{d}_2(\mathbf{F}_2 - \mathbf{F}_1),$$
$$\hat{\mathbf{y}}_1 = \mathbf{y}_0 + h\hat{\mathbf{d}}_1\mathbf{F}_1.$$

Компоненты векторов $\mathbf{d}_2$ и $\hat{\mathbf{d}}_1$ рассчитываем по формулам (8.9) с использованием функций (8.13б) для $\mathbf{d}_2$ и (8.13а) для $\hat{\mathbf{d}}_1$. Чтобы предотвратить переполнение или деление на 0, при вычислениях используем оценки z_{1i} для нежестких компонент и обратные значения $a_i = 1/z_{1i}$ для жестких компонент.

Приведем фрагмент программы на языке Паскаль, включающий вычисление настраиваемых параметров $\mathbf{d}_2$, $\hat{\mathbf{d}}_1$, α, формулу интегрирования, оценивание ошибки и изменение размера шага:

```
err:=1e-10; alfanew:=10;
for i:=1 to n do
begin
  u2:=F2[i]-F1[i];  u3:=(F3[i]-F2[i])/alfa;
  if abs(u3)<=1.6*abs(u2) then
  begin  //нежесткая компонента
    if u2<>0 then z:=u3/u2 else z:=0;
    d2:=0.5+z/6;
  end else //жесткая или неустойчивая компонента
  begin
    a:=u2/u3; if a<0 then d2:=-a*(a+1) else d2:=(92/75)*a;
    alfanew:=min(alfanew,abs(a));
  end;
```

```
  if abs(u3)<=abs(u2) then d1_:=1+z/2 else
  begin
    a:=u2/u3; if a<0 then d1_:=a/(a-1) else d1_:=1.5*a;
  end;
  y1[i]:=Y2[i]+d2*u2;
dy:=h*((1-d1_)*F1[i]+d2*u2); //оценка ошибки i-й компоненты
  a:=Rtol*max(abs(y1[i]),abs(y0[i]))+Atol;
  eri:=0.5*abs(dy)/a; //нормированная оценка ошибки
  if eri>err then err:=eri;
end;
w:=0.5*power(err,-0.5);
if w>4 then w:=4 else if w<0.25 then w:=0.25;
if err<=1 then //успешный шаг
begin t:=t+h; for i:=1 to n do y0[i]:=y1[i] end;
h:=w*h; //h и alfa следующего шага
alfa:=max(alfanew/w,0.5),1e-12);
```

В [50, 51] предложена модификация адаптивных методов, позволяющая повысить их устойчивость и сократить вычислительные затраты. Согласно (8.1) первая стадия следующего шага сводится к вычислению вектора производных по формуле

$$\mathbf{F}_1 = \mathbf{f}_1 = \mathbf{f}(t_1, \mathbf{y}_1). \tag{8.21}$$

Из (8.3), (8.4) следует

$$\mathbf{Y}_i - \mathbf{y}_0 = h\sum_{j=1}^{s-1}\beta_{ij}\mathbf{u}_j, \quad \mathbf{F}_i - \mathbf{f}_0 = \sum_{j=1}^{s-1}\beta_{ij}\mathbf{u}_{j+1}, \quad i = 2,\ldots,s. \tag{8.22}$$

Формула интегрирования (8.7) и соотношения (8.22) позволяют аппроксимировать вектор (8.21) в виде

$$\mathbf{F}_1 = \mathbf{f}_1 = \mathbf{f}_0 + \sum_{i=1}^{s-2}\mathbf{u}_{i+1}/i! + \mathbf{d}_{s-1}\mathbf{u}_s. \tag{8.23}$$

Посмотрим, в чем различие формул (8.21) и (8.23). При решении линейной системы $\mathbf{y}' = \mathbf{J}\mathbf{y}$ получаем $\mathbf{f}_1 = \mathbf{P}(h\mathbf{J})\mathbf{f}_0$, где в случае использования формулы (8.21)

$$\mathbf{P}(\mathbf{Z}) = \mathbf{I} + \mathbf{Z} + \cdots + \mathbf{Z}^{s-2}/(s-2)! + \mathbf{Z}\mathbf{D}_{s-1}\mathbf{Z}^{s-2},$$

а в случае использования формулы (8.23)

$$\mathbf{P}(\mathbf{Z}) = \mathbf{I} + \mathbf{Z} + \cdots + \mathbf{Z}^{s-2}/(s-2)! + \mathbf{D}_{s-1}\mathbf{Z}^{s-1}.$$

Таким образом, при использовании формулы (8.23) функция $\mathbf{P}(\mathbf{Z})$ совпадает с матричной функцией устойчивости $\mathbf{R}(\mathbf{Z})$, т. е. для линейной системы операторы перехода от $\mathbf{f}_0$ к $\mathbf{f}_1$ и от $\mathbf{y}_0$ к $\mathbf{y}_1$ совпадают. При использовании формулы (8.21) эти операторы различаются. Применение формулы (8.23) вместо (8.21) позволило значительно повысить устойчивость адаптивных методов при решении жестких задач и сэкономить одно вычисление правой части на шаге интегрирования. Поэтому формулу (8.23) назовем *стабилизированной первой стадией*. Из (8.5) и (8.9) следует, что эту же формулу можно записать в виде

$$\mathbf{F}_1 = \mathbf{f}_1 = \mathbf{f}_0 + \sum_{i=1}^{s-3} \mathbf{u}_{i+1}/i! + \mathbf{d}_{s-2}\mathbf{u}_{s-1}. \quad (8.24)$$

По сравнению с (8.23) формула (8.24) содержит меньше членов и менее чувствительна к вычислительным ошибкам.

Обозначим метод с первой стадией в виде (8.21) через ARK21 (Adaptive Runge–Kutta). В обозначении метода первая цифра – порядок для нежестких, а вторая – для жестких задач. Модификацию со стабилизированной первой стадией (8.24) обозначим через ARK21s. Результаты решения этими методами задачи Капса (4.7) (максимальные относительные ошибки на всем интервале и число вычислений функции *Nf* при $Tol = 10^{-2}$, $h_0 = 10^{-6}$) приведены в табл. 8.3. Метод ARK21 эффективно решил задачу для значений показателя жесткости вплоть до $\mu = 10^{18}$, а ARK21s – до $\mu = 10^{156}$. При этом ошибка и вычислительные затраты практически не изменялись при изменении μ в широких пределах. Значение μ примерно равно числу вычислений функции для явного метода 2-го порядка. Получить решение для таких значений μ другими известными явными методами практически невозможно.

Таблица 8.3. Результаты решения задачи Капса с переменным шагом

μ	ARK21		ARK21s		ARK21c	
	Ошибка	***Nf***	**Ошибка**	***Nf***	**Ошибка**	***Nf***
1	7.60×10^{-4}	51	9.20×10^{-4}	35	7.60×10^{-4}	51
10^3	1.54×10^{-3}	333	2.51×10^{-3}	221	5.50×10^{-4}	370
10^6	9.67×10^{-3}	372	1.98×10^{-3}	247	9.78×10^{-3}	494
10^{12}	9.89×10^{-3}	372	1.87×10^{-3}	247	9.89×10^{-3}	496
10^{18}	1.01×10^{-2}	429	1.88×10^{-3}	247	9.89×10^{-3}	496
10^{52}	–	–	1.88×10^{-3}	247	9.89×10^{-3}	496
10^{156}	–	–	1.88×10^{-3}	247	–	–

Другой способ повышения устойчивости заключается в выполнении дополнительной стадии, на которой осуществляется коррекция жестких компонент. Если *i*-я компонента жесткая (например, если $d_{1i} < 0.1$), то соответствующую переменную пересчитываем по формуле

$$y_{1i\ \mathrm{cor}} = y_{0i} + hd_{1i}f_{0i} + (1 - d_{1i})(y_{1i} - y_{0i}) + hd_{2i}(f_{1i} - f_{0i}).$$

В результате появляется дополнительный корень многочлена устойчивости по этой компоненте, близкий к z_{1i}, что приводит к расширению области устойчивости в окрестности z_{1i}. Модификацию с коррекцией жестких компонент обозначим через ARK21c, результаты этого метода при решении задачи Капса также приведены в табл. 8.3. За счет выполнения корректирующих стадий удалось решить задачу для значений μ вплоть до 10^{52}.

Из приведенных результатов очевидно преимущество метода ARK21s, который позволил при меньших затратах получить более точное решение для

более широкого диапазона изменения показателя жесткости. Этот метод имел преимущество и при решении многих других задач (результаты решения некоторых из них приведены в табл. 8.4). Однако для некоторых задач метод ARK21s оказался хуже. Рассмотрим линейную задачу из [14]:

$$y_1' = \frac{1}{1-a}(\lambda_1 - a\lambda_2)y_1 + \gamma y_2 + \frac{1}{1-a}(-\lambda_1 + \beta + a\gamma)y_3, \quad y_1(0) = 2.1,$$
$$y_2' = \frac{a}{1-a}(\lambda_1 - \lambda_2)y_1 + \beta y_2 + \frac{1}{1-a}(-a\lambda_1 + a\beta + \gamma)y_3, \quad y_2(0) = -0.9 + a, \qquad (8.25)$$
$$y_3' = \frac{a}{1-a}(\lambda_1 - \lambda_2)y_1 + \gamma y_2 + \frac{1}{1-a}(-a\lambda_1 + \beta + a\gamma)y_3, \quad y_3(0) = 1.1 + a,$$

$\beta = 0.5(\lambda_2 + \lambda_3)$, $\gamma = 0.5(\lambda_2 - \lambda_3)$, $a = 0.5$,

$\lambda_1 = -10^{-5}$, $\lambda_2 = -30$, $\lambda_3 = -1$, $0 \le t \le 3$,

решение которой:

$y_1 = \exp(\lambda_1 t) + 0.1\exp(\lambda_2 t) + \exp(\lambda_3 t)$,

$y_2 = a\exp(\lambda_1 t) + 0.1\exp(\lambda_2 t) - \exp(\lambda_3 t)$,

$y_3 = a\exp(\lambda_1 t) + 0.1\exp(\lambda_2 t) + \exp(\lambda_3 t)$.

При $Tol = 10^{-2}$ максимальная ошибка решения этой задачи методом ARK21 была 2.80×10^{-3} при затратах $Nf = 476$, а у метода ARK21s ошибка 8.76×10^{-2} при $Nf = 1512$. Поэтому в решателе, реализующем метод ARK21, имеет смысл сделать опцию, позволяющую выбрать способ (8.21) либо (8.24) реализации первой стадии.

Таблица 8.4. Результаты решения тестовых задач методом ARK21 и его модификациями

Задача	*Tol*	ARK21		ARK21s		ARK21c	
		scd	*Nf*	*scd*	*Nf*	*scd*	*Nf*
VDPOL	10^{-1}	1.02	697	1.43	491	1.20	749
	10^{-2}	2.01	3750	2.16	2515	2.02	4355
	10^{-3}	3.21	24033	3.07	16183	3.25	28319
ROBER	10^{-1}	1.35	3179	–	–	1.35	2136
	10^{-2}	2.30	13288	–	–	2.32	16937
OREGO	10^{-2}	1.54	7988	0.58	3036	1.81	6670
	10^{-3}	2.59	31996	1.40	12999	2.86	20488
HIRES	10^{-2}	0.82	2771	1.07	1877	1.16	3277
	10^{-3}	1.66	14070	2.21	9151	2.21	15353
CUSP	10^{-1}	2.32	2033	2.55	1141	3.01	2224
	10^{-2}	4.92	13321	4.09	7863	4.53	16918

8.4. Адаптивные методы Рунге–Кутты порядков 2 и 3

Адаптивный метод, обеспечивающий второй порядок при решении жестких задач, построим на основе стадий (8.14) при $s = 4$ и $\beta = 1$. В зависимости от модификации первую стадию выполняем по формуле (8.21) либо (8.24). Последующие стадии выполняем по формулам:

$$\begin{aligned}
&\mathbf{Y}_2 = \mathbf{y}_0 + h\mathbf{F}_1, \quad \mathbf{F}_2 = \mathbf{f}(t_0 + h, \mathbf{Y}_2),\\
&\mathbf{Y}_3 = \mathbf{Y}_2 + h\alpha(\mathbf{F}_2 - \mathbf{F}_1), \quad \mathbf{F}_3 = \mathbf{f}(t_0 + h, \mathbf{Y}_3),\\
&\mathbf{Y}_4 = \mathbf{Y}_3 + h\alpha(\mathbf{F}_3 - \mathbf{F}_2), \quad \mathbf{F}_4 = \mathbf{f}(t_0 + h, \mathbf{Y}_4).
\end{aligned} \tag{8.26}$$

Далее вычисляем:

$$\mathbf{u}_2 = \mathbf{F}_2 - \mathbf{F}_1, \quad \mathbf{u}_3 = (\mathbf{F}_3 - \mathbf{F}_2)/\alpha, \quad \mathbf{u}_4 = (\mathbf{F}_4 - \mathbf{F}_3)/\alpha^2. \tag{8.27}$$

Скалярную функцию устойчивости принимаем в виде (8.13в) для основной и (8.13б) для вложенной формулы. Алгоритм вычисления настраиваемых параметров и выполнения шага интегрирования запишется в виде:

```
err:=1e-10; alfanew:=10;
for i:=1 to n do
begin
  u2:=F2[i]-F1[i]; u3:=(F3[i]-F2[i])/alfa;
  u4:=(F4[i]-F3[i])/(alfa*alfa);
  if abs(u4)<=1.6*abs(u3) then
  begin  //нежесткая компонента
    if u3<>0 then z:=u4/u3 else z:=0;
    d3:=1/6+z/24; d2:=0.5+z*d3; d2_:=0.5+z/6;
  end else //жесткая или неустойчивая компонента
  begin
    a:=u3/u4; if a<0 then
begin d2:=-a*(a*(0.43264*a+1)+1); d2_:=-a*(a+1) end
else begin d2:=131/150; d2_:=(92/75)*a end;
d3:=(d2-0.5)*a; alfanew:=min(alfanew,abs(a));
  end;
  y1[i]:=Y2[i]+h*(0.5*u2+d3*u3);
  dy:=h*((0.5-d2_)*u2+d3*u3); //оценка ошибки i-й компоненты
  a:=Rtol*max(abs(y1[i]),abs(y0[i]))+Atol;
  eri:=abs(dy)/a; //нормированная оценка ошибки
  if eri>err then err:=eri;
end;
w:=0.7*power(err,-1/3);
if w>4 then w:=4 else if w<0.25 then w:=0.25;
if err<=1 then //успешный шаг
begin t:=t+h; for i:=1 to n do y0[i]:=y1[i] end;
h:=w*h; //h и alfa следующего шага
alfa:=max(min(alfanew/w,0.5),1e-8);
```

Для метода со стабилизированной первой стадией компоненты вектора $\mathbf{F}_1$ следующего шага вычисляем в виде `F1[i]:=F2[i]+d2*u3[i]`. Такой вариант оказался более эффективным для всех решаемых нами задач; обозначим его через ARK2s.

Построение адаптивных методов, реально имеющих третий порядок при решении жестких задач, затруднено низким стадийным порядком и неустойчивостью внутренних стадий явных схем Рунге–Кутты. Практически невозможно задать стадии таким образом, чтобы все они были устойчивыми, но в этом и нет необходимости. Достаточно, чтобы внутренние функции устойчивости имели вид (8.19) при $\beta = 1$ и подходящем значении α. Однако в методах порядка три и выше добиться этого непросто.

Преодолеть эти трудности удалось с помощью предложенных в [55] методов с нулевыми функциями погрешности, обладающих свойствами методов более высокого стадийного порядка (см. главу 4). За основу взят метод 3-го порядка, принадлежащий семейству [55, формула (4.2)] и имеющий таблицу Бутчера

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1 & 1 & 0 & & \\ 1 & 1-3\alpha & 4\alpha & -\alpha & \\ \hline & 1/6 & 2/3 & 1/6-1/(6\alpha) & 1/(6\alpha) \end{array} .$$

Этот метод имеет 2-й псевдостадийный порядок. Это свойство сохраняется и в построенном на его основе адаптивном методе. Для получения оценок собственных значений и стабилизации формулы интегрирования следует дополнить метод двумя стадиями. В результате получен описанный ниже метод. Мы убедились, что он более эффективен для жестких задач только при использовании стабилизированной первой стадии, поэтому рассмотрим именно такой метод, который обозначим через ARK3s.

На первом шаге принимаем $\mathbf{F}_1 = \mathbf{f}(t_0, \mathbf{y}_0)$, а на каждом последующем шаге вычисляем $\mathbf{F}_{1\text{new}} = \mathbf{F}_3 + (\mathbf{F}_4 - \mathbf{F}_3)/(2\alpha) + \mathbf{u}_4/6 + \mathbf{d}_4\mathbf{u}_5$. Остальные стадии выполняем по формулам:

$$\begin{aligned} &\mathbf{Y}_2 = \mathbf{y}_0 + (h/2)\mathbf{F}_1,\ \mathbf{F}_2 = \mathbf{f}(t_0 + h/2, \mathbf{Y}_2),\\ &\mathbf{Y}_3 = \mathbf{y}_0 + h\mathbf{F}_1,\quad \mathbf{Y}_4 = \mathbf{Y}_3 + h\alpha(-3\mathbf{F}_1 + 4\mathbf{F}_2 - \mathbf{F}_3),\\ &\mathbf{Y}_5 = \mathbf{Y}_4 + h\alpha[(\mathbf{F}_4 - \mathbf{F}_3) + 4\alpha(\mathbf{F}_1 - 2\mathbf{F}_2 + \mathbf{F}_3)],\\ &\mathbf{Y}_6 = \mathbf{Y}_5 + h\alpha(\mathbf{F}_5 - \mathbf{F}_4),\ \mathbf{F}_i = \mathbf{f}(t_0 + h, \mathbf{Y}_i),\ \ i = 3,\ldots, 6. \end{aligned} \tag{8.28}$$

Функции устойчивости этих стадий: $R_2(z) = 1 + z/2, R_3(z) = 1 + z, R_4(z) = 1 + z + \alpha z^2$, $R_5(z) = 1 + z + \alpha z^2 + \alpha^2 z^3, R_6(z) = 1 + z + \alpha z^2 + \alpha^2 z^3 + \alpha^3 z^4$. Далее принимаем:

$$\begin{aligned} &\mathbf{u}_2 = -3\mathbf{F}_1 + 4\mathbf{F}_2 - \mathbf{F}_3,\ \ \mathbf{u}_3 = (\mathbf{F}_4 - \mathbf{F}_3)/\alpha + 4(\mathbf{F}_1 - 2\mathbf{F}_2 + \mathbf{F}_3),\\ &\mathbf{u}_4 = (\mathbf{F}_5 - \mathbf{F}_4)/\alpha^2,\ \ \mathbf{u}_5 = (\mathbf{F}_6 - \mathbf{F}_5)/\alpha^3. \end{aligned} \tag{8.29}$$

Основная и вложенная формулы интегрирования запишутся в виде:

$$\mathbf{y}_{n+1} = \mathbf{Y}_3 + h(\mathbf{u}_2/2 + \mathbf{u}_3/6 + \mathbf{d}_4\mathbf{u}_4), \quad \hat{\mathbf{y}}_{n+1} = \mathbf{Y}_3 + h(\mathbf{u}_2/2 + \hat{\mathbf{d}}_3\mathbf{u}_3).$$

Компоненты векторов $\mathbf{d}_4$ и $\hat{\mathbf{d}}_3$ определяем согласно формулам (8.9) с использованием функций

$$Q(z) = \begin{cases} 1 + z + \dfrac{z^2}{2} + \dfrac{z^3}{6} + \dfrac{z^4}{24}, & |z| \le 2, \\ 1/(1-z), & z < -2, \\ 1 + z + \dfrac{z^2}{2} + \dfrac{z^3}{4}, & z > 2, \end{cases} \qquad \hat{Q}(z) = \begin{cases} 1 + z + \dfrac{z^2}{2} + \dfrac{5}{32}z^3, & z \ge -\dfrac{1}{4}, \\ Q(z), & z < -\dfrac{1}{4}. \end{cases}$$

Использование стадий (8.14) при $\beta = 2/3$ позволяет построить метод, имеющий 3-й порядок для нежестких задач, но для эффективного решения жестких задач значение β должно быть близко к 1. Поэтому имеет смысл настраивать в процессе решения не только заключительную формулу интегрирования, но и параметр β, что позволяет наиболее эффективно решать не только жесткие, но и нежесткие задачи.

Характерной особенностью многих жестких задач является наличие в решении коротких пограничных участков (слоев) с быстрым изменением переменных. При прохождении этих участков все значения $h\lambda_i$ невелики, поэтому задача внутри такого участка не является жесткой. Таким образом, все решение можно разбить на «медленные» жесткие и «быстрые» нежесткие участки. На нежестких участках изменения переменных наиболее значительны, поэтому они вносят заметный вклад в ошибку численного решения. Это является еще одной причиной для построения методов, имеющих повышенный порядок сходимости при решении нежестких задач.

Построим такой метод на основе стадий (8.14) при $s = 4$. Параметры α и β задаем исходя из полученных на предыдущем шаге оценок z_{1i}, при этом принимаем

$$\alpha = \min\left(1/3, \min_i |z_{1i}^{-1}| h_{\text{old}}/h\right), \quad \beta = 1 - \alpha.$$

Такие значения обеспечивают эффективное решение как жестких, так и нежестких задач.

Скалярную функцию устойчивости метода принимаем в виде

$$Q(z) = \begin{cases} 1 + z + \dfrac{z^2}{2} + \dfrac{z^3}{6} + \dfrac{z^4}{48}, & |z| \le 4.5, \\ 0, & z < -4.5, \\ 1 + z + \dfrac{107}{64}z^2, & z > 4.5, \end{cases}$$

а шаг интегрирования выполняем по формуле

$\mathbf{y}_1 = \mathbf{y}_0 + h(\mathbf{f}_0 + \mathbf{u}_2/2 + \mathbf{d}_3\mathbf{u}_3)$,

где $\mathbf{u}_i = (\mathbf{F}_i - \mathbf{F}_{i-1})/(\beta\alpha^{i-2})$, $i = 2, 3$. Вычисление компонент вектора $\mathbf{d}_3$ выполняем согласно (8.5), (8.9). Если i-я компонента нежесткая, т. е. если $|u_{4i}| \le 4.5|u_{3i}|$, то вычисляем $d_{3i} = 1/6 + z_{1i}/48$, $z_{1i} = u_{4i}/u_{3i}$ (при $u_{3i} = 0$ принимаем $d_{3i} = 1/6$), а иначе вычисляем

$$d_{3i} = \begin{cases} -a(0.5 + a(1+a)), & a < 0, \\ \dfrac{75}{64}a, & a > 0, \end{cases} \qquad a = u_{3i}/u_{4i}.$$

При $\beta < 1$ вложенная формула, построенная на основе только стадийных производных, может давать недостоверную оценку ошибки (см. пример в разделе 2.4). Поэтому строим вложенную FSAL-пару, в которой используем также и значение $\mathbf{f}_1 = \mathbf{f}(t_0 + h, \mathbf{y}_1)$, а вложенную формулу принимаем в виде

$\hat{\mathbf{y}}_1 = \mathbf{y}_0 + h(\mathbf{f}_0 + \hat{\mathbf{d}}_2\mathbf{u}_2 + \hat{\mathbf{d}}_3\mathbf{u}_3 + \hat{\mathbf{d}}_4\mathbf{v}_4)$, $\mathbf{v}_4 = \mathbf{f}_1 - \mathbf{f}_0 - \mathbf{u}_2 - \mathbf{u}_3/2$.

Вычисление очередной компоненты векторов $\hat{\mathbf{d}}_2, \hat{\mathbf{d}}_3$ и $\hat{\mathbf{d}}_4$ выполняем, используя оценку $a = z_1^{-1} = u_3/u_4$ по этой компоненте (индекс компоненты опускаем):

$$\hat{d}_2 = (1-\gamma-g)\gamma + c + g(1-g), \quad \hat{d}_3 = ((1-\gamma-g)\gamma + c)g + c\gamma,$$

$$\hat{d}_4 = cg(2 + 4\gamma(1+\gamma)), \quad c = g\left[g - \frac{7}{9}\right] + \frac{53}{162}, \quad \gamma = \min\left[\frac{2}{9}, |a|\right].$$

Эти значения выбраны с учетом условия 2-го порядка вложенной формулы для нежестких компонент и 1-го порядка для жестких компонент. Кроме этого, функция устойчивости $\hat{R}(z) = 1 + z + \hat{d}_2z^2 + \hat{d}_3z^3 + \hat{d}_4d_3z^4$ удовлетворяет условию $\hat{R}(-1/\gamma) \approx 0$ для жестких компонент, а параметр g остается свободным (мы принимаем $g = 1/8$). Полученный метод обозначим через ARK32.

Рассмотрим теперь метод с коррекцией жестких компонент, которую выполняем после вычисления соответствующих компонент векторов $\mathbf{y}_1, \mathbf{v}_4$. Если для i-й компоненты $-2/9 < a < 0$, то пересчитываем эту компоненту согласно формулам

$y_{1\text{cor}} = y_1 + h(\delta_3u_3 + \delta_4v_4)$, $\delta_3 = \gamma(1/2 - \gamma(2 - 3\gamma))$, $\delta_4 = \delta_3(2 + 4\gamma(1 + \gamma))$,

где y_1, u_3, v_4 – i-е компоненты векторов $\mathbf{y}_1, \mathbf{u}_3, \mathbf{v}_4$. Метод с коррекцией жестких компонент обозначим через ARK32c. В [154] было показано, что при решении жестких задач с умеренной точностью этот метод более эффектен, чем ARK32.

8.5. Методы с покомпонентным оцениванием двух собственных значений

Рассмотрим теперь адаптивные методы, в которых по каждой компоненте производится оценивание двух наибольших по модулю собственных значений

якобиана, причем полученные оценки могут быть комплексными числами. Использование таких оценок позволяет обеспечить быстрое и качественно правильное решение не только жестких, но и колебательных задач.

Пусть по результатам предварительных стадий получены векторы $\mathbf{u}_1,\ldots,\mathbf{u}_r$, которые сформированы через векторы $\mathbf{F}_1,\ldots,\mathbf{F}_r$ таким образом, что для линейной системы $\mathbf{y}' = \mathbf{J}\mathbf{y}$ имеем $\mathbf{u}_l = h^{-1}(h\mathbf{J})^l\mathbf{y}_0$. Рассмотрим i-ю компоненту численного решения, опуская для упрощения ее индекс. На основе полученной информации можно определить коэффициенты трехчлена

$$a_2z^2 + a_1z + a_0 = a_2(z - z_1)(z - z_2), \tag{8.30}$$

нулями которого являются оценки собственных значений матрицы $h\mathbf{J}$ по этой компоненте. Применяя степенной метод, получаем:

$$a_0 = u^2_{r-1} - u_{r-2}u_r, \quad a_1 = u_{r-3}u_r - u_{r-2}u_{r-1}, \quad a_2 = u^2_{r-2} - u_{r-3}u_{r-1}.$$

Формулу шага интегрирования принимаем в виде:

$$y_1 = y_0 + h\left(\sum_{l=1}^{r-4} u_l/l! + d_{r-3}u_{r-3} + d_{r-2}u_{r-2} + d_{r-1}u_{r-1}\right), \tag{8.31}$$

что соответствует функции устойчивости

$$R(z) = 1 + \sum_{l=1}^{r-4} z^l/l! + d_{r-3}z^{r-3} + d_{r-2}z^{r-2} + d_{r-1}z^{r-1}. \tag{8.32}$$

Предположим, что $a_2 \geq 0$ (в противном случае умножим a_0, a_1 и a_2 на -1). Компоненту будем считать нежесткой, если выполняется условие

$$|\mathrm{Re}\, z_1| \leq \zeta, \quad |\mathrm{Re}\, z_2| \leq \zeta, \tag{8.33}$$

где ζ – константа, определяемая размером области устойчивости для нежестких компонент. Используя критерий Рауса [76], получим условие (8.33) в виде:

$$|a_1| \leq 2a_2\zeta, \quad a_0 - |a_1|\zeta + a_2\zeta^2 \geq 0. \tag{8.34}$$

Среди нежестких компонент, удовлетворяющих (8.34), выделим колебательные, т. е. компоненты, для которых мнимая часть наибольшего собственного значения превышает действительную часть. Для колебательных компонент потребуем также, чтобы в течение периода колебаний было выполнено хотя бы несколько шагов интегрирования. Оба этих условия запишутся в виде:

$$2a_0a_2 > a_1^2, \quad a_0 \leqslant a_2\zeta^2. \tag{8.35}$$

Колебательные компоненты рассчитываем по формуле (8.31), в которой $d_{r-1} = 0$, а d_{r-2} и d_{r-3} определяются из условия

$$R(z_R + jz_I) = \exp(z_R)[\cos(z_I) + j\sin(z_I)], \quad z_R = -\frac{a_1}{2a_2}, \quad z_I = \frac{\sqrt{4a_0a_2 - a_1^2}}{2a_2},$$

где $R(z)$ имеет вид (8.32). Для нежестких компонент, удовлетворяющих (8.34) и не удовлетворяющих (8.35), принимаем $d_l = 1/l!$, $l = r-3, r-2, r-1$.

Рассмотрим теперь компоненты с двумя жесткими собственными значениями, для которых $\mathrm{Re}\, z_1 < -\zeta$ и $\mathrm{Re}\, z_2 < -\zeta$, что эквивалентно условию

$$a_1 - 2a_2\zeta > 0, \quad a_0 - a_1\zeta + a_2\zeta^2 > 0. \tag{8.36}$$

Для таких компонент принимаем $d_{r-1} = 0$, а d_{r-2} и d_{r-3} находим из условия делимости без остатка многочлена (8.32) на трехчлен (8.30). Например, при $r = 5$ получим:

$$d_2 = b_2 + b_1(1 - b_1), \quad d_3 = b_2(1 - b_1), \quad b_1 = a_1/a_0, \quad b_2 = a_2/a_0 .$$

Остались компоненты, которые не удовлетворяют ни (8.34), ни (8.36). Для них оцениваем одно наибольшее собственное значение по формуле $z_1 = u_r/u_{r-1}$. После этого находим d_{r-2} из соотношений (8.9) и применяем формулу интегрирования (8.31) при $d_{r-3} = 1/(r-3)!$, $d_{r-1} = 0$.

На основе изложенной методики были построены методы 2-го и 3-го порядков. Метод 2-го порядка имеет $r = s = 5$ и построен на основе стадий (8.14) при $\beta = 1$. На первом шаге принимаем $\mathbf{F}_1 = \mathbf{f}(t_0, \mathbf{y}_0)$, а на каждом последующем шаге вычисляем $\mathbf{F}_{1\mathrm{new}} = \mathbf{F}_2 + \mathbf{d}_2\mathbf{u}_3 + \mathbf{d}_3\mathbf{u}_4 + \mathbf{d}_4\mathbf{u}_5$. Дальнейшие вычисления производим по формулам (8.26), (8.27) и

$$\mathbf{Y}_5 = \mathbf{Y}_4 + h\alpha(\mathbf{F}_4 - \mathbf{F}_3), \quad \mathbf{F}_5 = \mathbf{f}(t_0 + h, \mathbf{Y}_5), \quad \mathbf{u}_5 = (\mathbf{F}_5 - \mathbf{F}_4)/\alpha^3,$$

после чего покомпонентно вычисляем настраиваемые параметры и выполняем шаг интегрирования согласно (8.31). Обозначим этот метод через ARK2os (os – oscillatory and stiff).

Метод 3-го порядка имеет $s = 7$ и $r = 6$. На всех шагах, кроме первого, принимаем $\mathbf{F}_{1\mathrm{new}} = \mathbf{F}_3 + (\mathbf{F}_4 - \mathbf{F}_3)/(2\alpha) + \mathbf{d}_3\mathbf{u}_4 + \mathbf{d}_4\mathbf{u}_5 + \mathbf{d}_5\mathbf{u}_6$. Последующие стадии выполняем по формулам (8.28), (8.29) и

$$\mathbf{Y}_7 = \mathbf{Y}_6 + h\alpha(\mathbf{F}_6 - \mathbf{F}_5), \quad \mathbf{F}_7 = \mathbf{f}(t_0 + h, \mathbf{Y}_7), \quad \mathbf{u}_6 = (\mathbf{F}_7 - \mathbf{F}_6)/\alpha^4,$$

после чего вычисляем настраиваемые параметры и выполняем шаг интегрирования согласно (8.31). Обозначим этот метод через ARK3os.

Покажем, что эти методы позволяют эффективно и точно решать колебательные задачи. Рассмотрим задачу из [99, 101]:

$$y'' = -\omega^2 y + (\omega^2 - 1)\sin(t), \quad y(0) = 1, \quad y'(0) = \omega + 1, \quad \omega = 10, \quad 0 \le t \le T. \tag{8.37}$$

Ее решение: $y(t) = \cos(\omega t) + \sin(\omega t) + \sin(t)$. Максимальные ошибки переменной y приведены в табл. 8.5. Для сравнения приведены результаты явного семистадийного метода 5-го порядка, предложенного в [143] для решения колебательных задач (обозначим его Simos5). Отметим, что ошибки методов ARK2os

Таблица 8.5. Ошибки решения колебательной задачи

h	$T = 100$			$T = 1000$		
	Simos5	ARK2os	ARK3os	Simos5	ARK2os	ARK3os
1/10	2.17×10^{-2}	9.17×10^{-4}	1.51×10^{-6}	2.20×10^{-1}	9.17×10^{-4}	1.52×10^{-6}
1/20	3.84×10^{-4}	2.27×10^{-4}	9.33×10^{-8}	3.84×10^{-3}	2.27×10^{-4}	9.42×10^{-8}
1/40	1.58×10^{-5}	5.67×10^{-5}	5.87×10^{-9}	1.58×10^{-4}	5.67×10^{-5}	6.78×10^{-9}

и ARK3os при решении этой задачи практически не зависят от T, тогда как ошибка метода Simos5 растет пропорционально T.

8.6. Построение многошаговых адаптивных методов

Рассмотренные одношаговые адаптивные методы имеют невысокий порядок и эффективны при решении жестких задач с умеренной точностью, но при высокой задаваемой точности их эффективность заметно снижается. Построение эффективных одношаговых методов более высоких порядков проблематично из-за их невысокого стадийного порядка, неустойчивости промежуточных стадий и уменьшения областей устойчивости в окрестности оценок $\mathbf{z}_1$ при повышении порядка метода. Поэтому адаптивные методы повышенной точности построим на основе предложенных в [45, 46] многошаговых формул.

Рассмотрим сначала скалярное дифференциальное уравнение

$$y' = f(t, y), \quad y(t_0) = y_0, \tag{8.38}$$

а затем распространим полученные формулы на векторный случай. Пусть в процессе интегрирования получено численное решение $y_1,\ldots, y_n$. Выведем k-шаговую формулу интегрирования. Аппроксимируя правую часть исходного уравнения (8.38) в окрестности текущей точки решения t_n, y_n, получаем:

$$y' = f_n + \lambda(y - y_n) + \sum_{j=1}^{k-1} b_j \frac{(t-t_n)^j}{j!}, \quad f_n = f(t_n, y_n). \tag{8.39}$$

Точное решение уравнения (8.39) при $t_{n+1} = t_n + h$ запишется в виде:

$$y_{n+1} = y_n + c_1 h f_n + c_2 h^2 b_1 + \cdots + c_k h^k b_{k-1}, \tag{8.40}$$

где

$$c_0 = e^z, \quad z = h\lambda, \tag{8.41а}$$

$$c_{i+1} = \frac{c_i - 1/i!}{z}, \quad i = 0, 1, \ldots, k-1. \tag{1.41б}$$

Отметим, что коэффициенты c_i совпадают с коэффициентами d_i, полученными для одношаговых методов в (8.9) при $Q(z) = e^z$.

Для использования формул (8.40), (8.41) при численном решении уравнения (8.38) необходимо на каждом шаге интегрирования оценивать значения $b_1, \ldots, b_{k-1}$. Это можно сделать, используя информацию, полученную на предыдущих шагах. Пусть шаг интегрирования постоянен, тогда, применяя аппроксимацию (8.39), получим:

$$f_{n-i} = f_n + \lambda(y_{n-i} - y_n) + \sum_{j=1}^{k-1} b_j \frac{(-ih)^j}{j!}, \quad i = 1,\ldots,k-1. \tag{8.42}$$

Эти формулы представляют собой систему линейных алгебраических уравнений относительно оцениваемых величин. Используя разности назад, задаваемые рекуррентными соотношениями

$\nabla^0 y_n = y_n$, $\nabla^i y_n = \nabla^{i-1} y_n - \nabla^{i-1} y_{n-1}$,

запишем решение уравнений (8.42) в рекуррентной форме:

$$b_i h^i = \nabla^i f_n - \lambda \nabla^i y_n - \sum_{j=i+1}^{k-1} \left[\frac{b_j h^j}{j!} \sum_{l=1}^{i} l^j (-1)^{j+l} \binom{i}{l} \right], \quad i = k-1, k-2, \ldots, 1. \tag{8.43}$$

Подставляя (8.43) в (8.40), получаем явную k-шаговую формулу

$$y_{n+1} = y_n + hc_1 f_n + \sum_{i=1}^{k-1} (\alpha_i \nabla^i y_n + h\beta_i \nabla^i f_n), \tag{8.44}$$

где

$$\begin{gathered} \beta_1 = c_2, \quad \beta_2 = c_3 + \frac{1}{2} c_2, \quad \beta_3 = c_4 + c_3 + \frac{1}{3} c_2, \\ \beta_4 = c_5 + \frac{3}{2} c_4 + \frac{11}{12} c_3 + \frac{1}{4} c_2, \ldots, \alpha_i = -z\beta_i, i = 1, 2, \ldots \end{gathered} \tag{8.45}$$

Учитывая (8.41б), имеем также

$$\begin{gathered} \alpha_1 = 1 - c_1, \quad \alpha_2 = 1 - c_2 - \frac{1}{2} c_1, \quad \alpha_3 = 1 - c_3 - c_2 - \frac{1}{3} c_1, \\ \alpha_4 = 1 - c_4 - \frac{3}{2} c_3 - \frac{11}{12} c_2 - \frac{1}{4} c_1, \ldots \end{gathered} \tag{8.46}$$

При $z \to 0$ получим $c_i = 1/i!$, $\alpha_i = 0$, $i = 1, 2, \ldots$, тогда вычисленные согласно (8.45) коэффициенты β_i задают явные методы Адамса. При $z \to -\infty$ получим $c_i = \beta_i = 0$, $\alpha_i = 1$, $i = 1, 2, \ldots$, тогда формула (8.44) задает экстраполяцию по предыдущим значениям y.

Явная формула (8.44) может обеспечить устойчивость численного решения, но точность при этом будет неудовлетворительной, поскольку коэффициенты при f_n, $\nabla^i f_n$ близки к нулю, вследствие чего теряется связь с правой частью. Формулы такого типа можно использовать для контроля точности, но шаг интегрирования будем выполнять на основе формул типа «прогноз–коррекция».

Неявная k-шаговая формула строится аналогично явной формуле и отличается от нее двумя последними членами:

$$y_{n+1} = y_n + hc_1 f_n + \sum_{i=1}^{k-1} (\alpha_i \nabla^i y_n + h\beta_i \nabla^i f_n) + \alpha_k \nabla^k y_{n+1} + h\beta_k \nabla^k f_{n+1}. \tag{8.47}$$

При $z \to 0$ формула (8.47) задает неявные методы Адамса, а при $z \to -\infty$ получаем формулы дифференцирования назад (в последнем случае нужно сначала избавиться от y_{n+1} в правой части, иначе получим тривиальное равенство $y_{n+1} = y_{n+1}$).

Шаг интегрирования многошагового адаптивного метода выполняем в три этапа. На первом этапе делаем прогноз по явной формуле Адамса

$$\hat{y}_{n+1} = y_n + hf_n + h\sum_{i=1}^{k-1} \gamma_i \nabla^i f_n, \quad \hat{f}_{n+1} = f(t_{n+1}, \hat{y}_{n+1}),$$

где $\gamma_1 = 1/2, \gamma_2 = 5/12, \gamma_3 = 3/8, \gamma_4 = 251/720, \ldots$ На втором этапе оцениваем значение $z = h\lambda$:

$$\tilde{y}_{n+1} = \hat{y}_{n+1} + h\alpha\nabla^k \hat{f}_{n+1}, \quad \tilde{f}_{n+1} = f(t_{n+1}, \tilde{y}_{n+1}), \quad z = \frac{\tilde{f}_{n+1} - \hat{f}_{n+1}}{\alpha\nabla^k \hat{f}_{n+1}}, \tag{8.48}$$

где параметр α рекомендуется выбирать в зависимости от оценки жесткости (для нежесткого уравнения можно задать $\alpha = \gamma_k$, а для жесткого уравнения следует задать α достаточно малым). Наконец, на третьем этапе выполняем шаг интегрирования по формуле (8.47), в которую вместо y_{n+1}, f_{n+1} подставляем $\hat{y}_{n+1}, \hat{f}_{n+1}$, а коэффициенты находим из (8.41), (8.45), (8.46). В частном случае при $k = 1$ получим одношаговый метод, задаваемый формулами

$$\hat{y}_{n+1} = y_n + hf_n, \quad \hat{f}_{n+1} = f(t_{n+1}, \hat{y}_{n+1}),$$

$$\tilde{y}_{n+1} = \hat{y}_{n+1} + h\alpha(\hat{f}_{n+1} - f_n), \quad z = \frac{f(t_{n+1}, \tilde{y}_{n+1}) - \hat{f}_{n+1}}{\alpha(\hat{f}_{n+1} - f_n)},$$

$$y_{n+1} = y_n + hc_1 f_n + (1 - c_1)(\hat{y}_{n+1} - y_n) + hc_2(\hat{f}_{n+1} - f_n)$$

(фактически это формулы метода ARK21).

Формула интегрирования (8.47) однозначно определяется коэффициентами c_i (8.41) и построена таким образом, что при решении уравнения Далквиста $y' = \lambda y$ получаем $y_{n+1} = c_0 y_n$, где $c_0 = e^z$. Такой способ задания коэффициентов (8.41) обеспечивает точное решение уравнения (8.39), но его недостатком является неограниченный рост этих коэффициентов при увеличении z, а также возможность деления на ноль в формулах (8.48), (8.41б). Поэтому определим коэффициенты таким образом, чтобы при любых значениях z они были ограничены, а также обеспечивали точность и качественно правильное воспроизведение нежестких, жестких и неустойчивых компонент численного решения.

Если выполняется неравенство

$$\left|\tilde{f}_{n+1} - \hat{f}_{n+1}\right| \le z_k^* \left|\alpha\nabla^k \hat{f}_{n+1}\right|, \tag{8.49}$$

где z_k^* – некоторая константа, то будем считать дифференциальное уравнение нежестким на данном шаге интегрирования. В этом случае вычисляем z по формуле (8.48), а если $\alpha\nabla^k \hat{f}_{n+1} = 0$, то принимаем $z = 0$. Коэффициенты c_i вычисляем по формулам

$$c_{k+1} = \frac{1}{(k+1)!} + \frac{z}{(k+2)!}, \quad c_i = \frac{1}{i!} + zc_{i+1}, \quad i = k, k-1, \ldots, 1, \tag{8.50}$$

тогда

$$c_0 = 1 + z + \frac{z^2}{2} + \ldots + \frac{z^{k+2}}{(k+2)!},$$

что соответствует аппроксимации экспоненты порядка $k + 2$ (можно задать $c_{k+1} = 1/(k+1)!$, тогда получим аппроксимацию порядка $k + 1$). Аппроксимации более высоких порядков практически не дают преимущества.

Если неравенство (8.49) не выполняется, то вместо z вычисляем

$$a = |z|^{-1} = \left| \frac{\alpha \nabla^k \hat{f}_{n+1}}{\tilde{f}_{n+1} - \hat{f}_{n+1}} \right|$$

и принимаем

$$c_0 = 0, \quad c_{i+1} = a(1/i! - c_i), \quad i = 0, 1, \ldots, k,$$

что обеспечивает затухание жестких и расхождение неустойчивых компонент численного решения.

Мы получили формулы численного интегрирования скалярного уравнения, параметры которых настраиваются на жесткость решаемой задачи. Распространим теперь эти формулы на случай решения системы дифференциальных уравнений $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$. Это можно выполнить тремя способами, которые отличаются размерностью величин, аналогичных значению $z = h\lambda$ в скалярном случае.

Первый способ (матричная реализация) является наиболее точным и наиболее трудоемким. В этом случае вычисляется и используется матрица $\mathbf{Z} = h\mathbf{J}$, где $\mathbf{J}$ – матрица Якоби, тогда коэффициенты c_i, α_i, β_i в формулах (8.45)–(8.47) заменяются соответствующими матрицами. Методы, основанные на использовании матричной экспоненты, рассматривались, например, в [42, 126]. Эти методы способны обеспечить высокую точность решения, но по сложности реализации и вычислительным затратам они превосходят не только явные, но и многие неявные методы.

Второй способ (покомпонентная реализация) предполагает использование вектора покомпонентных оценок наибольшего собственного значения матрицы $\mathbf{Z} = h\mathbf{J}$. В этом случае коэффициенты метода являются векторами, операции с которыми выполняются покомпонентно. Такой подход применяется при построении многих явных нелинейных методов [6, 14, 44–51, 100, 125].

Третий способ (усредненная реализация) основан на использовании одной усредненной по всем компонентам оценки наибольшего собственного значения, которую можно получить, применяя, например, метод наименьших квадратов. Методы такого типа рассматривались в [151] и некоторых других работах.

Рассмотрим методы с покомпонентной реализацией. В таких методах по разным компонентам можно получить оценки разных собственных значений, поэтому при решении сложных задач они имеют преимущество перед методами с усредненной реализацией.

8.7. Двухшаговый адаптивный метод

Интегрирование с постоянным шагом неэффективно для большинства задач, поэтому при реализации многошагового метода необходимо предусмотреть возможность изменения размера шага. Проще всего это сделать в двухшаговом методе, стадии которого выполняются по формулам:

$$\mathbf{f}_n = \mathbf{f}(t_n, \mathbf{y}_n), \quad w = h_n / h_{n-1},$$

$$\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + h_n \mathbf{f}_n + \frac{h_n}{2} w(\mathbf{f}_n - \mathbf{f}_{n-1}), \quad \hat{\mathbf{f}}_{n+1} = \mathbf{f}(t_{n+1}, \hat{\mathbf{y}}_{n+1}),$$

$$\tilde{\mathbf{y}}_{n+1} = \hat{\mathbf{y}}_{n+1} + h_n \alpha \nabla^2 \mathbf{f}, \quad \tilde{\mathbf{f}}_{n+1} = \mathbf{f}(t_{n+1}, \tilde{\mathbf{y}}_{n+1}).$$

Используемые разности назад 2-го порядка учитывают соотношение шагов w и обозначены в виде:

$$\nabla^2 \mathbf{y} = (\hat{\mathbf{y}}_{n+1} - \mathbf{y}_n) - w(\mathbf{y}_n - \mathbf{y}_{n-1}), \quad \nabla^2 \mathbf{f} = (\hat{\mathbf{f}}_{n+1} - \mathbf{f}_n) - w(\mathbf{f}_n - \mathbf{f}_{n-1}).$$

На первом шаге принимаем $\mathbf{y}_{-1} = \mathbf{y}_0$, $\mathbf{f}_{-1} = \mathbf{f}_0$.

Покомпонентные оценки наибольшего по модулю собственного значения получаем в виде:

$$\mathbf{z}_1 = \frac{\tilde{\mathbf{f}}_{n+1} - \hat{\mathbf{f}}_{n+1}}{\alpha \nabla^2 \mathbf{f}}. \tag{8.51}$$

Эти оценки используются для расчета настраиваемых параметров согласно формулам

$$\mathbf{c}_1 = \frac{Q(\mathbf{z}_1) - \mathbf{e}}{\mathbf{z}_1}, \quad \mathbf{c}_2 = \frac{\mathbf{c}_1 - \mathbf{e}}{\mathbf{z}_1}, \quad \mathbf{c}_3 = \frac{\mathbf{c}_2 - 0.5\mathbf{e}}{\mathbf{z}_1}, \quad \mathbf{e} = (1, \ldots, 1)^{\mathrm{T}},$$

где $Q(\mathbf{z})$ – примененная покомпонентно к вектору $\mathbf{z}$ функция (8.13б). Алгоритм вычисления этих параметров для очередной компоненты запишется в виде:

```
if abs(b)<=1.6*abs(a) then
begin
  if a<>0 then b:=b/a;
  c3:=1/6; c2:=0.5+b*c3; c1:=1+b*c2;
end else
begin
  a:=a/b;
  if a<0 then c1:=-a else c1:=167/75;
  c2:=a*(c1-1); c3:=a*(c2-0.5);
end;
```

где `b` и `a` – соответствующие компоненты числителя и знаменателя в выражении (8.51).

Шаг интегрирования выполняем по формуле:

$$\delta\mathbf{y} = \frac{\mathbf{e}-\mathbf{c}_1+w(\mathbf{e}-2\mathbf{c}_2)}{1+w}\nabla^2\mathbf{y} + h_n\frac{\mathbf{c}_2+2w\mathbf{c}_3}{1+w}\nabla^2\mathbf{f},$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n\mathbf{c}_1\mathbf{f}_n + w(\mathbf{e}-\mathbf{c}_1)(\mathbf{y}_n-\mathbf{y}_{n-1}) + h_n w\mathbf{c}_2(\mathbf{f}_n-\mathbf{f}_{n-1}) + \delta\mathbf{y},$$

а $\delta\mathbf{y}$ используем для вычисления нормированной оценки ошибки и размера следующего шага. Метод имеет 3-й порядок для нежестких и 2-й порядок для жестких задач; обозначим его AM32 (Adaptive Multistep of orders 3 and 2).

8.8. Многошаговый адаптивный метод переменного порядка и шага

В общем случае построение многошаговых методов с переменным шагом приводит к громоздким формулам, но их можно упростить, если использовать разделенные разности [74]. Обычно разделенную разность порядка i для функции $\mathbf{f}(t)$, заданную в точках t_n, t_{n-1}, ..., t_{n-i}, обозначают в виде $\mathbf{f}[t_n, \ldots, t_{n-i}]$, но нам удобнее использовать более компактное обозначение в виде $\bar{\nabla}^i\mathbf{f}_n$.

Разделенные разности задаются рекуррентными соотношениями

$$\bar{\nabla}^0\mathbf{f}_n = \mathbf{f}_n, \quad \bar{\nabla}^i\mathbf{f}_n = \frac{\bar{\nabla}^{i-1}\mathbf{f}_n - \bar{\nabla}^{i-1}\mathbf{f}_{n-1}}{t_n - t_{n-i}} \tag{8.52}$$

(аналогично определяются $\bar{\nabla}^i\mathbf{y}_n$). Введем коэффициенты a_{ij}, g_{ij}, зависящие от размеров текущего и предыдущих шагов и задаваемые формулами

$$a_{11} = t_{n+1} - t_n = h, \quad a_{1i} = a_{1,i-1}(t_n - t_{n-i+1}), \quad a_{ii} = iha_{i-1,i-1},$$
$$a_{ji} = jha_{j-1,i-1} + a_{j,i-1}(t_n - t_{n-i+1}), \quad i = 2,\ldots,k, \quad j = 2,\ldots,i-1; \tag{8.53}$$
$$g_i = \sum_{j=1}^{i}\frac{a_{ji}}{(j+1)!}, \quad i = 1,\ldots,k.$$

Тогда k-шаговые формулы прогноза и коррекции при интегрировании с переменным шагом запишутся в виде:

$$\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + h\mathbf{f}_n + h\sum_{i=1}^{k-1} g_i\bar{\nabla}^i\mathbf{f}_n, \quad \hat{\mathbf{f}}_{n+1} = \mathbf{f}(t_{n+1}, \hat{\mathbf{y}}_{n+1}), \tag{8.54а}$$

$$\tilde{\mathbf{y}}_{n+1} = \hat{\mathbf{y}}_{n+1} + h\alpha\bar{\nabla}^k\hat{\mathbf{f}}_{n+1}, \quad \tilde{\mathbf{f}}_{n+1} = \mathbf{f}(t_{n+1}, \tilde{\mathbf{y}}_{n+1}), \tag{8.54б}$$

где $\bar{\nabla}^i\hat{\mathbf{f}}_{n+1}$ – разделенная разность с использованием $\hat{\mathbf{f}}_{n+1}$ вместо $\mathbf{f}_{n+1}$.

На основе формул (8.52)–(8.54) строят многошаговые методы переменного порядка и шага, при этом принимают $\alpha = g_k$, а оценку ошибки определяют как разность между прогнозом (8.54а) и скорректированным решением (8.54б). Для адаптивного метода эти формулы являются предварительными, на их основе получаем вектор оценок наибольшего собственного значения

$$\mathbf{z}_1 = \frac{\tilde{\mathbf{f}}_{n+1} - \hat{\mathbf{f}}_{n+1}}{\alpha\bar{\nabla}^k\hat{\mathbf{f}}_{n+1}}. \tag{8.55}$$

Для получения достоверных оценок в случае жесткой нелинейной задачи величина α должна быть достаточно малой. Практически оптимальное значение этого параметра задается формулой

$$\alpha = \min(g_k, \tilde{\alpha}), \quad \tilde{\alpha} = a_{\min} \prod_{i=0}^{k-1} (t_{n+1} - t_{n-i}),$$

где $a_{\min}$ – минимальное значение среди полученных на предыдущем шаге компонент вектора $\mathbf{a} = |\mathbf{z}_1|^{-1}$.

Начиная с нахождения оценок (8.55) все вычисления выполняем покомпонентно, при этом расчетные формулы построены так, чтобы исключить возможность деления на ноль или переполнения. Приведем формулы вычисления $\mathbf{y}_{n+1}$ для одной компоненты, опуская для упрощения индекс этой компоненты. Если выполняется неравенство

$$\left|\tilde{f}_{n+1} - \hat{f}_{n+1}\right| \le z_k^* \left|\alpha \bar{\nabla}^k \hat{f}_{n+1}\right|, \quad z_k^* = \begin{cases} 1.6, & k \le 3, \\ 4/k, & k > 3, \end{cases} \tag{8.56}$$

то соответствующая компонента считается нежесткой. В этом случае вычисляем оценку

$$z = \frac{\tilde{f}_{n+1} - \hat{f}_{n+1}}{\alpha \bar{\nabla}^k \hat{f}_{n+1}}$$

(если правая часть в (8.56) равна нулю, то принимаем $z = 0$), после чего находим настраиваемые коэффициенты по формулам (8.50). Шаг интегрирования выполняем по формуле

$$y_{n+1} = y_n + hc_1 f_n + \sum_{i=1}^{k-1} u_i (h\bar{\nabla}^i f_n - z\bar{\nabla}^i y_n) + u_k (h\bar{\nabla}^k \hat{f}_{n+1} - z\bar{\nabla}^k \hat{y}_{n+1}), \tag{8.57}$$

где

$$u_i = \sum_{j=1}^{i} c_{j+1} a_{ji}, \quad i = 1, \ldots, k.$$

Если для очередной компоненты неравенство (8.56) не выполняется (жесткая либо неустойчивая компонента), то вместо z вычисляем

$$a = |z|^{-1} = \left| \frac{\alpha \bar{\nabla}^k \hat{f}_{n+1}}{\tilde{f}_{n+1} - \hat{f}_{n+1}} \right|.$$

В этом случае коэффициенты метода находим по формулам

$$d_0 = 1, \quad d_i = \frac{1}{i!} - a d_{i-1}, \quad v_i = \sum_{j=1}^{i} d_j a_{ji}, \quad i = 1, \ldots, k,$$

а шаг интегрирования принимаем в виде:

$$y_{n+1} = y_n + haf_n + \sum_{i=1}^{k-1} v_i (\bar{\nabla}^i y_n + ha\bar{\nabla}^i f_n) + v_k (\bar{\nabla}^k \hat{y}_{n+1} + ha\bar{\nabla}^k \hat{f}_{n+1}). \tag{8.58}$$

Оценку ошибки получаем как последний член в формуле (8.57) или (8.58). Порядок метода определяется значением k, он равен $k + 1$ для нежестких задач и k для жестких задач, что подтверждается экспериментально [46]. В линейных многошаговых методах порядок обычно выбирается из условия, чтобы при заданной точности шаг был максимальным, т. е. алгоритмы изменения шага и порядка совмещены в одной процедуре. В нелинейных методах реализация такой процедуры затруднена, поскольку трудно получить достоверные оценки ошибки при значениях k, отличных от текущего значения. Поэтому мы разделяем процедуры изменения шага и изменения порядка. Размер нового шага получаем на основе оценки ошибки на предыдущем шаге. Новое значение k определяем на этапе прогноза, формулу которого (8.54а) можно записать в виде:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta_1 + \Delta_2 + \cdots + \Delta_k.$$

Для обеспечения наиболее точного прогноза мы стремимся минимизировать величину $||\Delta_k||$, при вычислении которой используем принятую при оценке ошибки норму. В соответствии с этим проверяем условие

$$||\Delta_{k-1}|| < ||\Delta_k||, \quad k > 1, \tag{8.59}$$

а если оно не выполняется, то проверяем условие

$$||\Delta_k|| > ||\Delta_{k+1}||, \quad k < k_{\max}. \tag{8.60}$$

Если условие (8.59) выполняется два шага подряд или если оно выполняется после неудачного шага, то уменьшаем k на 1, а если два шага подряд выполняется условие (8.60), то увеличиваем k на 1.

Многошаговый адаптивный метод реализован в ПО SimInTech и показал высокую эффективность при решении многих тестовых и прикладных задач. В этой реализации мы приняли $k_{\max} = 5$, тогда порядок метода может изменяться от 2-го до 6-го для нежестких и от 1-го до 5-го для жестких задач. Обозначим этот метод через AM61.

8.9. Численные эксперименты

Для тестирования адаптивных методов были выбраны 5 жестких задач из [75, 128]: VDPOL, ROBER, OREGO, HIRES и CUSP, характеристики которых приведены в табл. 1.3. Результаты решения этих задач методом ARK21 и его модификациями приведены в табл. 8.4. Из этих результатов видно, что метод 1-го порядка (для жестких задач) обеспечивает эффективное решение только с весьма низкой точностью. Для получения более точных результатов следует использовать методы более высоких порядков, результаты которых приведем ниже. Из всех методов, имеющих порядок не ниже 2-го для жестких задач, только многошаговые методы AM32 и AM61 смогли решить задачу ROBER на всем интервале ($T = 10^{11}$). Поэтому мы решали эту задачу также и на укороченном интервале ($T = 10^4$). Такая задача остается жесткой и имеет меру жесткости $M_ж = 10^8$, обозначим ее через ROBER*.

Как и ранее, вычислительные затраты оцениваем числом вычислений правой части *Nf*, а в качестве оценки точности решения используем значение *scd* (3.18). Для всех задач задаем допустимую относительную ошибку *Rtol* = *Tol*, а допустимую абсолютную ошибку принимаем в виде *Atol* = *Tol* для VDPOL и OREGO, $Atol = 10^{-12} \times Tol$ для ROBER, $Atol = 10^{-6} \times Tol$ для ROBER*, $Atol = 10^{-4} \times Tol$ для HIRES и $Atol = 10^{-2} \times Tol$ для CUSP. По сравнению с [75, 85, 128], мы задаем меньшие значения *Atol* для задач ROBER, HIRES и CUSP, что позволяет получить лучшее соответствие значения *scd* задаваемому допуску на ошибку.

Результаты тестирования, некоторые из которых приведены в [154], показали, что среди одношаговых методов наиболее эффективны ARK32c (при умеренной точности) и ARK3s (при повышенной точности). В табл. 8.6 приводим результаты этих двух методов, а также методов AM32 и AM61 при трех значениях *Tol*. Обратим внимание на результаты методов ARK3s, AM32 и AM61 при решении задачи ROBER*. При $Tol = 10^{-2}$ вычислительные затраты *Nf* намного больше, чем при меньших значениях *Tol*. Объясняется это тем, что при

Таблица 8.6. Результаты решения тестовых задач адаптивными методами

Задача	Метод	$Tol = 10^{-2}$		$Tol = 10^{-3}$		$Tol = 10^{-4}$	
		scd	*Nf*	*scd*	*Nf*	*scd*	*Nf*
VDPOL	ARK32c	2.44	1093	3.11	2029	4.13	4110
	ARK3s	2.31	1181	3.06	1276	3.56	2276
	AM32	1.91	736	3.02	2105	4.12	5592
	AM61	2.72	702	3.16	972	4.08	1398
ROBER	AM32	2.12	5815	3.22	4468	4.07	11620
	AM61	1.92	3873	1.98	3549	4.30	2546
ROBER*	ARK32c	3.84	925	4.17	1394	4.47	2330
	ARK3s	2.96	41 917	2.61	21 156	4.74	10 031
	AM32	2.31	4762	3.14	1267	4.10	2824
	AM61	2.60	832	3.49	465	4.49	612
OREGO	ARK32c	0.95	1870	1.67	3598	2.92	8883
	ARK3s	1.71	2451	2.18	3036	3.19	4971
	AM32	0.99	1807	2.23	3921	3.55	9071
	AM61	1.97	3808	2.59	2167	3.48	3060
HIRES	ARK32c	0.73	1344	1.29	1652	2.71	2293
	ARK3s	1.22	2216	3.06	2341	3.78	2666
	AM32	1.23	1269	1.80	1768	2.64	3124
	AM61	2.35	1197	3.59	1301	3.70	1760
CUSP	ARK32c	2.42	679	3.18	1185	3.91	2826
	ARK3s	3.05	7086	3.66	3611	4.53	3331
	AM32	2.94	5383	2.78	1318	4.35	3178
	AM61	2.81	1527	2.94	1166	4.32	1175

увеличении размера шага возникает неустойчивость численного решения, которая приводит к резким колебаниям величины шага и увеличению числа шагов. Аналогичное явление проявляется и при решении задачи CUSP (хотя эта задача значительно менее жесткая, чем задачи VDPOL и OREGO, при решении которых данный эффект отсутствует).

Для более наглядного сравнения методов мы объединили результаты решения пяти задач (VDPOL, ROBER*, OREGO, HIRES и CUSP) и приводим на рис. 8.1 зависимости суммарных затрат на их решение от усредненного по этим задачам значения *scd* при $Tol = 10^{-i}$, $i = 2, \ldots, 7$. При $Tol \le 10^{-3}$ метод AM61 имеет заметное преимущество, которое увеличивается с уменьшением *Tol*. Среди других методов отметим ARK32c, который показывает хорошие результаты при умеренных требованиях к точности.

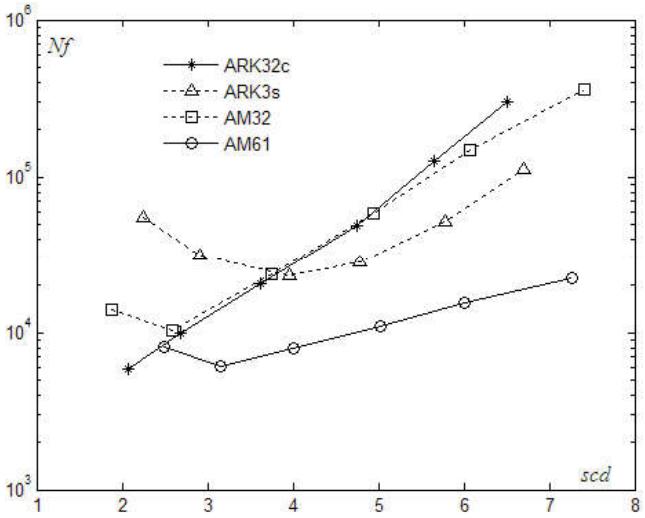


Рис. 8.1. Диаграмма «точность – объем вычислений» для пяти задач

Приведенные результаты показывают, что при решении некоторых жестких задач явные адаптивные методы не уступают неявным методам. Приведем теперь примеры жестких задач, при решении которых явные адаптивные методы заметно лучше неявных методов.

Одну из таких задач мы уже рассматривали – это уравнения (8.20). При решении этой задачи все используемые нами неявные решатели требовали больших вычислительных затрат, которые заметно увеличивались при увеличении μ. Низкую эффективность неявных методов можно объяснить быстрым изменением якобиана в процессе решения (хотя собственные числа не меняются). В то же время явные адаптивные методы эффективно решают эту задачу. В табл. 8.7 приведены результаты явного решателя AM61 и неявного решателя ode15s при $Tol = 10^{-4}$. Оба решателя реализуют многошаговые методы порядков

от 1-го до 5-го (AM61 – для жестких задач), поэтому такое сравнение вполне правомерно и показывает убедительное преимущество явного адаптивного метода.

Таблица 8.7. Результаты решения задачи (8.20)

Метод	μ	Ошибка	*Nf*	*NJ*	*NLU*
AM61	10^2	1.53×10^{-5}	203	0	0
	10^4	4.88×10^{-5}	246	0	0
	10^6	4.11×10^{-5}	243	0	0
ode15s	10^2	2.83×10^{-4}	330	33	94
	10^4	3.34×10^{-4}	1379	184	443
	10^6	3.60×10^{-3}	40284	4369	15831

Применение неявных методов для решения некоторых жестких задач может привести к качественно неверному результату. Предположим, что решение гладкое, а шаг интегрирования устанавливается большим. В этом случае неявные методы подавляют все составляющие решения, соответствующие большим по модулю собственным значениям якобиана, независимо от знака действительной части. Поэтому если одно из собственных значений быстро перемещается в правую полуплоскость и становится большим, то неявный метод вместо неустойчивого решения может дать неправильное устойчивое решение.

Рассмотрим, например, задачу

$$y_1' = y_2, \quad y_2' = \mu(1 - y_1^2)(y_1 + y_2),$$
$$y_1(0) = 2, \quad y_2(0) = -2, \quad 0 \le t \le 3, \tag{8.61}$$

построенную аналогично осциллятору Ван-дер-Поля, но имеющую более резкий переход от устойчивого состояния к неустойчивому. Задача имеет периодическое решение, показанное для $\mu = 10^8$ на рис. 8.2 сплошной линией. Однако при больших значениях μ неявные методы обычно дают неправильное затухающее решение, показанное на рисунке пунктиром, которое практически не изменяется при изменении допустимой точности в широких пределах. Этот факт может привести к ошибочному мнению, что данное решение является правильным. На приведенном примере видно, что при моделировании процессов, имеющих быстро нарастающий, катастрофический характер, неявные методы могут давать совершенно неверный результат, соответствующий устойчивому процессу. Явные адаптивные методы правильно решают эту и подобные локально-неустойчивые задачи при умеренных требованиях к точности. Например, при $\mu = 10^8$ и $Tol = 10^{-i}$, $i = 2,\ldots, 7$, неявный решатель ode15s не смог правильно решить задачу, и только при $Tol = 10^{-8}$ было получено верное решение. Но уже при $\mu = 10^{12}$ ни один решатель системы MATLAB не позволил получить правильное решение ни при каких значениях *Tol*. Явный адаптивный решатель ПО SimInTech «Адаптивный 1» правильно решает задачу для значений μ вплоть до 10^{48} при $Tol = 10^{-3}$.

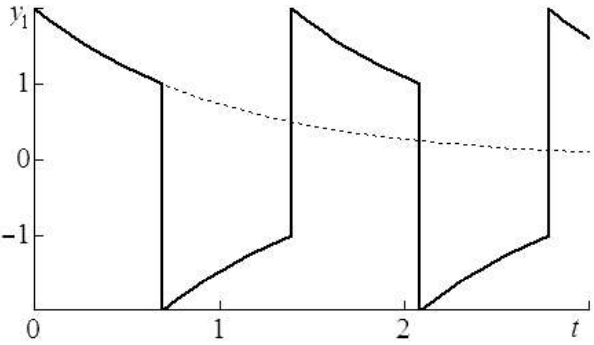


Рис. 8.2. Решение задачи (8.61): правильное (сплошная линия) и неправильное (пунктир)

Результаты тестовых испытаний показали, что при решении многих жестких задач явные адаптивные методы не уступают неявным методам, а иногда и превосходят их. Существуют, однако, жесткие задачи, для которых адаптивные методы не столь эффективны. К ним относятся, например, задачи, полученные путем дискретизации уравнений в частных производных (такие как PLATE, BEAM, BRUSS). При решении таких задач явные адаптивные методы могут быть эффективнее обычных явных методов, но уступают неявным методам, а если спектр якобиана вещественный, то и явным методам с расширенными областями устойчивости. Объясняется это тем, что в задачах такого типа жесткие составляющие решения, соответствующие разным собственным значениям, сильно связаны между собой. А для эффективного использования адаптивных методов связь между жесткими составляющими решения должна быть слабой, что обеспечивает доминирование не более одного жесткого собственного значения по каждой компоненте.

Литература

1. *Альшин А. Б., Альшина Е. А., Калиткин Н. Н., Корягина А. Б.* Схемы Розенброка с комплексными коэффициентами для жестких и дифференциально-алгебраических систем // Журнал вычисл. матем. и матем. физ. 2006. Т. 46. № 8. С. 1392–1414.
2. *Альшина Е. А., Закс Е. М., Калиткин Н. Н.* Оптимальные параметры явных схем Рунге–Кутты невысоких порядков // Математическое моделирование. 2006. Т. 18. № 2. С. 3–15.
3. *Альшина Е. А., Закс Е. М., Калиткин Н. Н.* Оптимальные схемы Рунге–Кутты с первого по шестой порядок точности // Журнал вычисл. матем. и матем. физ. 2008. Т. 48. № 3. С. 418–429.
4. *Арушанян О. Б., Залеткин С. Ф.* Численное решение обыкновенных дифференциальных уравнений на Фортране. М.: Изд-во МГУ, 1990. 336 с.
5. *Аульченко С. М., Латыпов А. Ф., Никуличев Ю. В.* Метод численного интегрирования систем обыкновенных дифференциальных уравнений с использованием интерполяционных полиномов Эрмита // Журнал вычисл. матем. и матем. физ. 1998. Т. 38. № 10. С. 1665–1670.
6. *Бобков В. В.* Новые явные A-устойчивые методы численного решения дифференциальных уравнений // Дифференциальные уравнения. 1978. Т. 14. № 12. С. 2249–2251.
7. *Богатырев А. Б.* Эффективное решение задачи о наилучшем многочлене устойчивости // Математический сборник. 2005. Т. 196. № 7. С. 27–50.
8. *Булатов М. В., Тыглиян А. В., Филиппов С. С.* Об одном классе одношаговых одностадийных методов для жестких систем обыкновенных дифференциальных уравнений // Журнал вычисл. матем. и матем. физ. 2011. Т. 51. № 7. С. 1251–1265.
9. *Вайнер Р., Куликов Г. Ю.* Эффективное управление точностью численного интегрирования обыкновенных дифференциальных уравнений и оптимальные интерполяционные равнозначные блочные методы с переменным шагом // Журнал вычисл. матем. и матем. физ. 2014. Т. 54 № 4. С. 591–607.
10. *Воеводин В. В., Кузнецов Ю. А.* Матрицы и вычисления. М.: Наука, 1984. 320 с.
11. *Гантмахер Ф. Р.* Теория матриц. 4-е изд. М.: Наука, 1988. 552 с.
12. *Деккер К., Вервер Я.* Устойчивость методов Рунге–Кутты для жестких нелинейных дифференциальных уравнений. М.: Мир, 1988. 334 с.

13. *Жук Д. М., Маничев В. Б., Папсуев А. Ю.* Обобщенный метод моделирования динамики технических систем // Информационные технологии. 2004. № 8. С. 6–14.
14. *Заворин А. Н.* Применение нелинейных методов для расчета переходных процессов в электрических цепях // Известия вузов. Радиоэлектроника. 1983. Т. 26. № 3. С. 35–41.
15. *Зубанов А. М., Ширков П. Д.* Численное исследование одношаговых явно-неявных методов, L-эквивалентных жесткоточным двухстадийным схемам Рунге–Кутты // Математическое моделирование. 2012. Т. 24. № 12. С. 129–136.
16. *Калиткин Н. Н.* Численные методы решения жестких систем // Математическое моделирование. 1995. Т. 7. № 5. С. 8–11.
17. *Калиткин Н. Н., Корякин П. В.* Численные методы: в 2 кн. Кн. 2: Методы математической физики. 2013. М.: Издательский центр «Академия», 2013. 304 с.
18. *Калиткин Н. Н., Альшин А. Б., Альшина Е. А., Рогов Б. В.* Вычисления на квазиравномерных сетках. М.: Физматлит, 2005. 224 с.
19. *Калиткин Н. Н., Панченко С. Л.* Оптимальные схемы для жестких неавтономных систем // Математическое моделирование. 1999. Т. 11. № 6. С. 52–81.
20. *Калиткин Н. Н., Пошивайло И. П.* Вычисления с использованием обратных схем Рунге–Кутты // Математическое моделирование. 2013. Т. 25. № 10. С. 79–96.
21. *Калиткин Н. Н., Пошивайло И. П.* Решение задачи Коши для жестких систем с гарантированной точностью методом длины дуги // Математическое моделирование. 2014. Т. 26. № 7. С. 3–18.
22. *Карташов Б. А., Шабаев Е. А., Козлов О. С., Щекатуров А. М.* Среда динамического моделирования технических систем SimInTech. М.: ДМК Пресс, 2017. 424 с.
23. *Козлов О. С., Скворцов Л. М.* Тестовое сравнение решателей ОДУ системы MATLAB // Всероссийская научная конференция «Проектирование научных и инженерных приложений в среде MATLAB». М.: Изд-во ИПУ РАН, 2002. С. 53–60. URL: http://matlab.exponenta.ru/conf2002/proceedings.php.
24. *Козлов О. С., Кондаков Д. Е., Скворцов Л. М. и др.* Программный комплекс «Моделирование в технических устройствах». 2005. URL: https://klinachevnv.ru/root/mvtu/20050615.html.
25. *Козлов О. С., Скворцов Л. М., Ходаковский В. В.* Решение дифференциальных и дифференциально-алгебраических уравнений в программном комплексе «МВТУ». 2005. URL: https://klinachevnv.ru/root/mvtu/20051121.html.
26. *Козлов О. С., Скворцов Л. М.* Программный комплекс «МВТУ» в научных исследованиях и прикладных разработках // Математическое моделирование. 2015. Т. 27. № 11. С. 32–46.
27. *Куликов Г. Ю.* Теоремы сходимости для итерационных методов Рунге–Кутты с постоянным шагом интегрирования // Журнал вычисл. матем. и матем. физ. 1996. Т. 36. № 8. С. 73–89.
28. *Куликов Г. Ю.* Численное решение задачи Коши для системы дифференциально-алгебраических уравнений с помощью неявных методов Рунге–Кутты с нетривиальным предиктором // Журнал вычисл. матем. и матем. физ. 1998. Т. 38. № 1. С. 68–84.
29. *Куликов Г. Ю., Меркулов А. И.* Об одношаговых коллокационных методах со старшими производными для решения обыкновенных дифференциальных уравнений // Журнал вычисл. матем. и матем. физ. 2004. Т. 44. № 10. С. 1782–1807.

30. *Куликов Г. Ю., Кузнецов Е. Б., Хрусталева Е. Ю.* О контроле глобальной ошибки в неявных гнездовых методах Рунге–Кутты гауссовского типа // Сибирский журнал вычисл. матем. 2011. Т. 14. № 3. С. 245–259.
31. *Куликов Г. Ю.* Вложенные симметричные неявные гнездовые методы Рунге–Кутты типов Гаусса и Лобатто для решения жестких обыкновенных дифференциальных уравнений и гамильтоновых систем // Журнал вычисл. матем. и матем. физ. 2015. Т. 55. № 6. С. 986–1007.
32. *Лебедев В. И.* Как решать явными методами жесткие системы дифференциальных уравнений // Вычислительные процессы и системы. М.: Наука, 1991. Вып. 8. С. 237–291.
33. *Лебедев В. И., Медовиков А. А.* Явный метод второго порядка точности для решения жестких систем обыкновенных дифференциальных уравнений // Известия вузов. Математика. 1998. № 9. С. 55–63.
34. *Лебедев В. И.* Функциональный анализ и вычислительная математика. М.: Физматлит, 2005. 296 с.
35. *Лебедев В. И., Финогенов С. А.* Об использовании упорядоченных чебышёвских параметров в итерационных методах // Журнал вычисл. матем. и матем. физ. 1976. Т. 16. № 4. С. 895–907.
36. *Маничев В. Б., Глазкова В. Н.* Методы интегрирования систем ОДУ для адаптируемых программных комплексов анализа РЭС // Радиотехника. 1988. № 4. С. 88–91.
37. *Маничев В. Б., Уваров М. Ю.* Базовые методы интегрирования обыкновенных дифференциальных уравнений для программ анализа радиоэлектронных схем // Известия вузов. Радиоэлектроника. 1989. Т. 32. № 6. С. 45–49.
38. *Новиков А. Е., Новиков Е. А.* Численное решение жестких задач с небольшой точностью // Математическое моделирование. 2010. Т. 22. № 1. С. 46–56.
39. *Новиков Е. А.* Явные методы для жестких систем. Новосибирск: Наука, 1997. 195 с.
40. *Новиков Е. А., Шитов Ю. А., Шокин Ю. И.* О классе (m, k)-методов решения жестких систем // Журнал вычисл. матем. и матем. физ. 1989. Т. 29. № 2. С. 194–201.
41. *Новиков Е. А., Шорников Ю. В.* Компьютерное моделирование жестких гибридных систем. Новосибирск: Изд-во НГТУ, 2012. 451 с.
42. *Ракитский Е. В., Устинов С. М., Черноруцкий И. Г.* Численные методы решения жестких систем. М.: Наука, 1979. 199 с.
43. *Сениченков Ю. Б.* Численное моделирование гибридных систем. СПб.: Изд-во Политехн. ун-та, 2004. 206 с.
44. *Скворцов Л. М.* Адаптивные методы цифрового моделирования динамических систем // Известия РАН. Теория и системы управления. 1995. № 4. С. 180–190.
45. *Скворцов Л. М.* Адаптивные методы численного интегрирования в задачах моделирования динамических систем // Известия РАН. Теория и системы управления. 1999. № 4. С. 72–78.
46. *Скворцов Л. М.* Явные адаптивные методы численного решения жестких систем // Математическое моделирование. 2000. Т. 12. № 12. С. 97–107.
47. *Скворцов Л. М.* Явный многошаговый метод численного решения жестких дифференциальных уравнений // Журнал вычисл. матем. и матем. физ. 2007. Т. 47. № 6. С. 959–967.

48. *Скворцов Л. М.* Простые явные методы численного решения жестких обыкновенных дифференциальных уравнений // Вычислительные методы и программирование. 2008. Т. 9. С. 154–162. URL: http://num-meth.srcc.msu.ru.
49. *Скворцов Л. М.* Явные многошаговые методы численного решения жестких обыкновенных дифференциальных уравнений // Вычислительные методы и программирование. 2008. Т. 9. С. 409–418. URL: http://num-meth.srcc.msu.ru.
50. *Скворцов Л. М.* Явные адаптивные методы Рунге–Кутты // Математическое моделирование. 2011. Т. 23. № 7. С. 73–87.
51. *Скворцов Л. М.* Явные адаптивные методы Рунге–Кутты для жестких и колебательных задач // Журнал вычисл. матем. и матем. физ. 2011. Т. 51. № 8. С. 1434–1448.
52. *Скворцов Л. М.* О повышении точности явных методов Рунге–Кутты при решении умеренно жестких задач // Доклады АН. 2001. Т. 378. № 5. С. 602–604.
53. *Скворцов Л. М.* Диагонально-неявные FSAL-методы Рунге–Кутты для жестких и дифференциально-алгебраических систем // Математическое моделирование. 2002. Т. 14. № 2. С. 3–17.
54. *Скворцов Л. М.* Точность методов Рунге–Кутты при решении жестких задач // Журнал вычисл. матем. и матем. физ. 2003. Т. 43. № 9. С. 1374–1384.
55. *Скворцов Л. М.* Явные методы Рунге–Кутты для умеренно жестких задач // Журнал вычисл. матем. и матем. физ. 2005. Т. 45. № 11. 2017–2030.
56. *Скворцов Л. М.* Диагонально-неявные методы Рунге–Кутты для жестких задач // Журнал вычисл. матем. и матем. физ. 2006. Т. 46. № 12. С. 2209–2222.
57. *Скворцов Л. М.* Свойство интерполяционности методов Рунге–Кутты // Математическое моделирование. 2008. Т. 20. № 12. С. 119–128.
58. *Скворцов Л. М.* Экономичная схема реализации неявных методов Рунге–Кутты // Журнал вычисл. матем. и матем. физ. 2008. Т. 48. № 11. С. 2008–2018.
59. *Скворцов Л. М.* Простой способ построения двухшаговых методов Рунге–Кутты // Журнал вычисл. матем. и матем. физ. 2009. Т. 49. № 11. С. 1920–1930.
60. *Скворцов Л. М.* Модельные уравнения для исследования точности методов Рунге–Кутты // Математическое моделирование. 2010. Т. 22. № 5. С. 146–160.
61. *Скворцов Л. М.* Диагонально-неявные методы Рунге–Кутты для дифференциально-алгебраических уравнений индексов 2 и 3 // Журнал вычисл. матем. и матем. физ. 2010. Т. 50. № 6. С. 1047–1059.
62. *Скворцов Л. М.* Простой способ построения многочленов устойчивости для явных стабилизированных методов Рунге–Кутты // Математическое моделирование. 2011. Т. 23. № 1. С. 81–86.
63. *Скворцов Л. М.* Явные стабилизированные методы Рунге–Кутты // Журнал вычисл. матем. и матем. физ. 2011. Т. 51. № 7. С. 1236–1250.
64. *Скворцов Л. М.* Коллокационные методы Рунге–Кутты для дифференциально-алгебраических уравнений индексов 2 и 3 // Журнал вычисл. матем. и матем. физ. 2012. Т. 52. № 10. С. 1801–1811.
65. *Скворцов Л. М.* Эффективная реализация неявных методов Рунге–Кутты второго порядка // Математическое моделирование. 2013. Т. 25. № 5. С. 15–28.
66. *Скворцов Л. М., Козлов О. С.* Эффективная реализация диагонально-неявных методов Рунге–Кутты // Математическое моделирование. 2014. Т. 26. № 1. С. 96–108.

67. *Скворцов Л. М.* Однократно неявные диагонально расширенные методы Рунге–Кутты четвертого порядка // Журнал вычисл. матем. и матем. физ. 2014. Т. 54. № 5. С. 755–765.
68. *Скворцов Л. М.* Неявный метод пятого порядка для численного решения дифференциально-алгебраических уравнений // Журнал вычисл. матем. и матем. физ. 2015. Т. 55. № 6. С. 978–984.
69. *Скворцов Л. М.* О неявных методах Рунге–Кутты, полученных в результате обращения явных методов // Математическое моделирование. 2017. Т. 29. № 1. С. 3–19.
70. *Скворцов Л. М.* Как избежать снижения точности и порядка методов Рунге–Кутты при решении жестких задач // Журнал вычисл. матем. и матем. физ. 2017. Т. 57. № 7. С. 1126–1141.
71. *Скворцов Л. М.* Неявные методы Рунге–Кутты с явными внутренними стадиями // Журнал вычисл. матем. и матем. физ. 2018. Т. 58. № 3.
72. Современные численные методы решения обыкновенных дифференциальных уравнений / под ред. Дж. Холла и Дж. Уатта. М.: Мир, 1979. 312 с.
73. *Федоренко Р. П.* Жесткие системы обыкновенных дифференциальных уравнений и их численное интегрирование // Вычислительные процессы и системы. Вып. 8. М.: Наука, 1991. С. 328–380.
74. *Хайрер Э., Нёрсетт С., Ваннер Г.* Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. М.: Мир, 1990. 512 с.
75. *Хайрер Э., Ваннер Г.* Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. М.: Мир, 1999. 685 с.
76. *Цыпкин Я. З.* Основы теории автоматических систем. М.: Наука, 1977. 560 с.
77. *Шампайн Л. Ф., Гладвел И., Томпсон С.* Решение обыкновенных дифференциальных уравнений с использованием MATLAB. СПб.: Лань, 2009. 304 с.
78. *Abdulle A.* On roots and error constants of optimal stability polynomials // BIT. 2000. V. 40. № 1. P. 177–182.
79. *Abdulle A., Medovikov A.A.* Second order Chebyshev methods based on orthogonal polynomials // Numer. Math. 2001. V. 90. № 1. P. 1–18.
80. *Abdulle A.* Fourth order Chebyshev methods with recurrence relation // SIAM J. Sci. Comput. 2002. V. 23. № 6. P. 2041–2054.
81. *Alexander R.* Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s // SIAM J. Numer. Anal. 1977. V. 14. № 6. P. 1006–1021.
82. *Alexander R.* Design and implementation of DIRK integrators for stiff systems. // Appl. Numer. Math. 2003. V. 46. № 1. P. 1–17.
83. *Ashour S. S., Hanna O. T.* Explicit exponential method for the integration of stiff ordinary differential equations // J. of Guidance, Control and Dynamics. 1991. V. 14. № 6. P. 1234–1239.
84. *Berzins M., Furzeland R.M.* An adaptive theta method for the solution of stiff and nonstiff differential equations // Appl. Numer. Math. 1992. V. 9. № 1. P. 1–19.
85. BIMD (release 1.1.2, November 2014). URL: http://web.math.unifi.it/users/brugnano/BiM/ BiMD/index_BiMD.htm.
86. *Bogacki P., Shampine L. F.* A 3(2) pair of Runge–Kutta formulas // Appl. Math. Lett. 1989. V. 2. № 4. P. 321–325.

87. *Burrage K., Chipman F. H., Muir P. H.* Order results for mono-implicit Runge-Kutta methods // SIAM J. Numer. Anal. 1994. V. 31. № 3. P. 876–891.
88. *Butcher J. C.* Implicit Runge–Kutta processes // Math. Comput. 1964. V. 18. № 85. P. 50–64.
89. *Butcher J. C.* Numerical methods for ordinary differential equations in the 20th century // J. Comput. Appl. Math. 2000. V. 125. P. 1–29.
90. *Butcher J. C.* Numerical methods for ordinary differential equations. 2th edition. Chichester: John Wiley & Sons, 2008. 463 p.
91. *Butcher J. C., Cash J. R., Diamantakis M. T.* DESI methods for stiff initial-value problems // ACM Trans. Math. Software. 1996. V. 22. № 4. P. 401–422.
92. *Butcher J. C., Chen D. J. L.* A new type of singly implicit Runge-Kutta methods // Appl. Numer. Math. 2000. V. 34. № 2–3. P. 179–188.
93. *Butcher J. C., Rattenbury N.* ARK methods for stiff problems // Appl. Numer. Math. 2005. V. 53. P. 165–181.
94. *Butcher J. C., Tracogna S.* Order conditions for two-step Runge–Kutta methods // Appl. Numer. Math. 1997. V. 24. № 2–3. P. 351–364.
95. *Cameron F., Palmroth M., Piche R.* Quasi stage order conditions for SDIRK methods // Appl. Numer. Math. 2002. V. 42. № 1–3. P. 61–75.
96. *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.
97. *Curtiss C. W., Hirschelder J. O.* Integration of stiff equations // Proc. Nat. Acad. Sci. USA. 1952. V. 38. P. 235–243.
98. *Dormand J. R., Prince P. J.* A family of embedded Runge–Kutta formulae // J. Comp. Appl. Math. 1980. V. 6. № 1. P. 19–26.
99. *Fang Y., Song Y., Wu X.* New embedded pairs of explicit Runge–Kutta methods with FSAL properties adapted to the numerical integration of oscillatory problems // Physics Letters A. 2008. V. 372. № 44. P. 6551–6559.
100. *Fowler M. E., Warten R. M.* A numerical integration technique for ordinary differential equations with widely separated eigenvalues // IBM J. Res. and Development. 1967. V. 11. № 5. P. 537–543.
101. *Franco J. M.* Runge–Kutta methods adapted to the numerical integration of oscillatory problems // Appl. Numer. Math. 2004. V. 50. № 3–4. P. 427–443.
102. *Gear C. W.* Algorithm 407: DIFSUB for solution of ordinary differential equations // Comm. ACM. 1971. V. 14. № 3. P. 185–190.
103. *Gear C. W.* Numerical solution of ordinary differential equations: is there anything left to do? // SIAM Review. 1981. V. 23. № 1. P. 10–24.
104. *Gonzalez-Pinto S., Perez-Rodriguez S., Montijano J. I.* On the numerical solution of stiff IVPs by Lobatto IIIA Runge–Kutta methods // J. Comput. Appl. Math. 1997. V. 82. P. 129–148.
105. *Gonzalez-Pinto S., Montijano J. I., Perez-Rodriguez S.* On the starting algorithms for fully implicit Runge-Kutta methods // BIT. 2000. V. 40. № 4. P. 685–714.
106. *Gonzalez-Pinto S., Hernandez-Abreu D., Montijano J. I.* An efficient family of strongly A-stable Runge–Kutta collocation methods for stiff systems and DAEs. Part I: Stability and order results // J. Comput. Appl. Math. 2010. V. 234. № 4. P. 1105–1116.

107. *Gonzalez-Pinto S., Hernandez-Abreu D., Montijano J. I.* An efficient family of strongly A-stable Runge–Kutta collocation methods for stiff systems and DAEs. Part II: Convergence results // Appl. Numer. Math. 2012. V. 62. P. 1349–1360.
108. *Gonzalez-Pinto S., Hernandez-Abreu D., Simeon B.* Strongly A-stable first stage explicit collocation methods with stepsize control for stiff and differential-algebraic equations // J. Comput. Appl. Math. 2014. V. 259. P. 138–152.
109. *Hairer E., Lubich Ch., Roche M.* The numerical solution of differential-algebraic systems by Runge–Kutta methods. Lecture Notes in Math. 1409. Berlin: Springer-Verlag, 1989. 139 p.
110. *Higham D. J., Hall G.* Embedded Runge–Kutta formulae with stable equilibrium states // J. Comput. Appl. Math. 1990. V. 29. P. 25–33.
111. *Higueras I., Roldan T.* Starting algorithms for some DIRK methods // Numer. Algorithms. 2000. V. 23. № 4. P. 357–369.
112. *Hosea M. E., Shampine L. F.* Analysis and implementation of TR-BDF2 // Appl. Numer. Math. 1996. V. 20. № 1–2. P. 21–37.
113. *Jackiewich Z., Tracogna S.* A general class of two-step Runge–Kutta methods for ordinary differential equations // SIAM J. Numer. Anal. 1995. V. 32. № 5. P. 1390–1427.
114. *Jackiewicz Z., Verner J. H.* Derivation and implementation of two-step Runge–Kutta pairs // Japan J. Ind. Appl. Math. 2002. V. 19. P. 227–248.
115. *Jackson K. R., Kværnø A., Nørsett S. P.* An analysis of the order of Runge–Kutta methods that use an iterative scheme to compute their internal stage values // BIT. 1996. V. 36. № 4. P. 713–765.
116. *Jay L.* Convergence of a class of Runge-Kutta methods for differential-algebraic systems of index 2 // BIT. 1993. V. 33. № 1. P. 137–150.
117. *Jay L.* Convergence of Runge-Kutta methods for differential-algebraic systems of index 3 // Appl. Numer. Math. 1995. V. 17. № 2. P. 97–118.
118. *Kennedy C. A., Carpenter M. H.* Additive Runge–Kutta schemes for convection–diffusion–reaction equations // Appl. Numer. Math. 2003. V. 44. № 1–2. P. 139–181.
119. *Kennedy C. A., Carpenter M. H.* Diagonally implicit Runge–Kutta methods for ordinary differential equations. A Review. NASA report NASA/TM-2016-219173. 2016. 156 c.
120. *Kosti A. A., Anastassi Z. A., Simos T. E.* An optimized explicit Runge–Kutta method with increased phase-lag order for the numerical solution of the Schrödinger equation and related problems // J. Math. Chem. 2010. V. 47. № 1. P. 315–330.
121. *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.
122. *Kulikov G. Yu., Weiner R.* Variable-stepsize interpolating explicit parallel peer methods with inherent global error control// SIAM J. Sci. Comput. 2010. V. 32. № 4. P. 1695–1723.
123. *Kulikov G. Yu.* Cheap global error estimation in some Runge–Kutta pairs // IMA J. Numer. Anal. 2013 V. 33. P. 136–163.
124. *Kværnø A.* Singly diagonally implicit Runge–Kutta methods with an explicit first stage // BIT. 2004. V. 44. № 3. P. 489–502.
125. *Lambert J. D.* Nonlinear methods for stiff systems of ordinary differential equations // Lect. Notes in Math. 1974. V. 363. P. 75–88.

126. *Martin-Vaquero J., Vigo-Aguiar J.* Exponential fitting BDF algorithms: Explicit and implicit 0-stable methods // J. Comput. Appl. Math. 2006. V. 192. № 1. P. 100–113.
127. *Martin-Vaquero J., Janssen B.* Second-order stabilized explicit Runge–Kutta methods for stiff problems // Computer Physics Communications. 2009. V. 180. № 10. P. 1802–1810.
128. *Mazzia F., Magherini C.* Test set for initial value problem solvers. Release 2.4. 2008. URL: http://pitagora.dm.uniba.it/~testset/report/testset.pdf.
129. *Medovikov A. A.* Third order explicit method for the stiff ordinary differential equations // Lect. Notes in Comput. Sci. 1196. Numerical Analysis and its Applications. Springer, 1997. P. 327–334.
130. *Medovikov A. A.* High order explicit methods for parabolic equations // BIT. 1998. V. 38. № 2. P. 372–390.
131. *Muir P., Owren B.* Order barriers and characterizations for continuous mono-implicit Runge–Kutta schemes // Math. Comput. 1993. V. 61. № 204. P. 675–699.
132. ODELab. Zuse Institute Berlin. 2017. URL: http://num-lab.zib.de/public/odelab.
133. *Olsson H., Söderlind G.* Stage value predictors and efficient Newton iterations in implicit Runge–Kutta methods // SIAM J. Sci. Comput. 1998. V. 20. № 1. P. 185–202.
134. *Papakostas S. N., Papageorgiou G.* A family of fifth-order Runge–Kutta pairs. // Math. Comput. 1996. V. 65. № 215. P. 1165–1181.
135. *Prothero A., Robinson A.* On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations // Math. Comput. 1974. V. 28. № 1. P. 145–162.
136. *Ralston A.* Runge–Kutta methods with minimum error bounds // Math. Comput. 1962. V. 16. P. 431–437.
137. *Rang J.* An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4 // Appl. Numer. Math. 2015. V. 94. P. 75–87.
138. *Riha W.* Optimal stability polynomials // Computing. 1972. V. 9. P. 37–43.
139. *Rosenbrock H. H.* Some general implicit processes for the numerical solution of differential equations // Comput. J. 1963. V. 5. № 4. P. 329–330.
140. *Sandu A.* Rosenbrock methods with an explicit first stage // Int. J. Comput. Math. 2016. V. 93. № 6. P. 995–1010.
141. *Shampine L. F., Reichelt M. W.* The MATLAB ODE suite // SIAM J. Sci. Comput. 1997. V. 18. № 1. P. 1–22.
142. *Shampine L. F., Thompson S.* Event location for ordinary differential equations // Int. J. Comput. Math. Appl. 2000. V. 39. P. 43–54.
143. *Simos T. E.* A modified Runge–Kutta method for the numerical solution of ODE's with oscillation solutions // Appl. Math. Lett. 1996. V. 9. № 6. P. 61–66.
144. *Söderlind G.* Time-step selection algorithms: Adaptivity, control, and signal processing // Appl. Numer. Math. 2006. V. 56. № 3–4. P. 488–502.
145. *Sommeijer B. P., Shampine L. F., Verwer J. D.* RKC: An explicit solver for parabolic PDEs // J. Comput. Appl. Math. 1997. V. 88. № 2. P. 315–326.
146. *Tracogna S., Welfert B.* Two-step Runge-Kutta: theory and practice // BIT. 2000. V. 40. № 4. P. 775–799.
147. *Van de Vyver H.* Stability and phase-lag analysis of explicit Runge–Kutta methods with variable coefficients for oscillatory problems. Comput. Phys. Commun. 2005.

V. 173. P. 115–130.
148. *Verwer J. G.* Explicit Runge–Kutta methods for parabolic partial differential equations // Appl. Numer. Math. 1996. V. 22. № 1–3. P. 359–379.
149. *Wambecq A.* Rational Runge–Kutta methods for solving systems of ordinary differential equations // Computing. 1978. V. 20. № 4. P. 333–342.
150. *Williams R., Burrage K., Cameron I., Kerr M.* A four-stage index 2 diagonally implicit Runge–Kutta method // Applied Numerical Mathematics. 2002. V. 40. № 3. P. 415–432.
151. *Wu X. Y., Xia J. L.* Two low accuracy methods for stiff systems // Appl. Math. Comput. 2001. V. 123. № 2. P. 141–153.
152. *Boom P. D., Zingg D. W.* Optimization of high-order diagonally-implicit Runge–Kutta methods. J. Comput. Phys. 2018. V. 371. P. 168–191.
153. *Kennedy C. A., Carpenter M. H.* Diagonally implicit Runge–Kutta methods for stiff ODEs // Appl. Numer. Math. 2019. V. 146. P. 221–244.
154. *Скворцов Л. М.* Построение и анализ явных адаптивных одношаговых методов численного решения жестких задач // Ж. вычисл. матем. и матем. физ. 2020. Т. 60. № 7. С. 1111–1125.
155. *Скворцов Л. М.* Методы ESDIRK третьего и четвертого порядков для жестких и дифференциально-алгебраических задач // Ж. вычисл. матем. и матем. физ. 2022. Т. 62. № 5. С. 790–808.

Книги издательства «ДМК ПРЕСС»
можно купить оптом и в розницу
в книготорговой компании «Галактика»
(представляет интересы издательств
«ДМК ПРЕСС», «СОЛОН ПРЕСС», «КТК Галактика»).
Адрес: г. Москва, пр. Андропова, 38, оф. 10;
тел.: **(499) 782-38-89**, электронная почта: **books@alians-kniga.ru**.
При оформлении заказа следует указать адрес (полностью),
по которому должны быть высланы книги;
фамилию, имя и отчество получателя.
Желательно также указать свой телефон и электронный адрес.
Эти книги вы можете заказать и в интернет-магазине: **http://www.galaktika-dmk.com/**.

Скворцов Леонид Маркович

Численное решение обыкновенных дифференциальных и дифференциально-алгебраических уравнений

Втрое издание

Главный редактор *Мовчан Д. А.*
dmkpress@gmail.com
Зам. главного редактора *Сенченкова Е. А.*
Корректор *Синяева Г. И.*
Верстка *Чаннова А. А.*
Дизайн обложки *Мовчан А. Г.*

Формат 70×100 1/16.
Гарнитура «PT Serif». Печать офсетная.
Усл. печ. л. 19,18. Тираж 50 экз.

Веб-сайт издательства: **www.dmkpress.com**