САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Коврижных Николай Александрович

# Разработка экономичных схем интегрирования структурно разделённых систем обыкновенных дифференциальных уравнений

Образовательная программа «Математическая кибернетика»
Направление подготовки 02.06.01 —
«Компьютерные и информационные науки»

Выпускная квалификационная работа

Научный руководитель:
д. ф.-м. н., проф.
Олемской Игорь Владимирович

Рецензент:
д. т. н., проф.
Голоскоков Дмитрий Петрович

Санкт-Петербург — 2018

## Оглавление

Стр.

# Введение

Системы дифференциальных уравнений являются одним из основных способов математического моделирования. Математическое описание сложных динамических процессов, протекающих в окружающем мире, зачастую приводит к системам, не имеющим аналитического решения. Кроме того, в некоторых задачах (например, в линейных СОДУ большого размера) вывод точного решения может оказаться сопряжён с трудозатратными вычислениями, и получение приближённого решения с некоторой желаемой точностью будет более оправданным. Поэтому численные методы решения систем дифференциальных уравнений всегда будут необходимым инструментом математического моделирования.

В начале XX века немецкий математик Карл Рунге разработал теорию графических численных методов, показав, как с помощью простейших инженерных инструментов на бумаге производить построения, аналогичные сложным математическим операциям вплоть до приближённого интегрирования скалярных дифференциальных уравнений [1]. Развитие вычислительной техники привело к тому, что теория Рунге оказалась не нужна, однако предложенные им идеи привели к созданию явных одношаговых методов Рунге—Кутты, позволявших получать приближённое решение вплоть до четвёртого порядка точности при сравнительно небольших трудозатратах: для точности порядка $p$ требовалось $p$ вычислений правой части СОДУ ($p$-этапные методы одношаговые методы, $p \leqslant 4$).

Технологический рывок середины XX века принёс новые возможности ЭВМ и вместе с тем предъявил новые требования к точности численного интегрирования. Безрезультатные годы поиска пятиэтапных методов пятого порядка завершились разработкой теории Джона Бутчера, систематизировавшего процесс нахождения методов Рунге—Кутты. Бутчер показал, что для высоких порядков существуют ограничения («барьеры Бутчера»): при $p \geqslant 5$ методы $p$-ого порядка требуют не менее $p + 1$ вычисление правой части СОДУ [2], а при $p \geqslant 7$ — не менее $p + 2$ [3; 4]. Эти ограничения а также трудоёмкость получения новых методов высокого порядка стали основными аргументами для поиска альтернатив классическим методам типа Рунге—Кутты.

Начиная с 1970-х лет наступает время бурного развития новых методов интегрирования и способов сравнения одних методов с другими. Важными свойствами становятся не только порядок точности, но также возможность автоматического управления длиной шага и устойчивость методов. Эрвин Фельберг одним из первых предложил идею вложенных методов Рунге—Кутты, позволяющих существенно экономить вычисления с помощью автоматического управления длиной шага интегрирования [5]. Благодаря этому методы Рунге—Кутты обрели новую популярность, и вскоре на их основе были созданы схемы Дж. Дорманда и П. Принса [6], одна из которых в качестве функции $ode45$ сейчас является основным интегратором среды MATLAB.

## Постановка задачи

Барьеры Бутчера являются одним из основных недостатков методов Рунге—Кутты. Способ их преодоления был предложен И. В. Олемским [7]. Он заключается в анализе структуры СОДУ, разбиении переменных на группы и модификации классической схемы Рунге—Кутты. Для многих задач, особенно в области механики, возможно сократить количество вычислений правой части системы, сохранив порядок точности. На основе структурного подхода было построено несколько вложенных схем, превосходящих по своим характеристикам известные аналоги. К примеру, метод пятого порядка точности требует в полтора раза меньше обращений к правой части (4 вместо 6).

Целью работы является построение экономичного метода интегрирования, преодолевающего барьер Бутчера для $p = 6$ (требующего менее семи вычислений правой части СОДУ). Кроме того, метод должен сопровождаться алгоритмом оценки локальной погрешности на шаге. Подтверждение соответствия заявленному порядку производится путём сравнения с уже существующими методами того же порядка на известных моделях.

# Обзор литературы

Несмотря на долгую историю, методы Рунге—Кутты по-прежнему привлекают много внимания. Популярна идея разбиения переменных на несколько групп с последующим применением к каждой группе отдельной вычислительной схемы. К примеру, ранние работы были связаны с разбиением жёстких задач на «жёсткую» и «нежёсткую» компоненты и построению явно-неявных методов [8].

Существуют работы, посвящённые разбиению СОДУ на «быстрые» и «медленные» компоненты для лучшего контроля динамических процессов в больших системах [9], а также получению схем с определёнными численными свойствами [10; 11].

Схожие идеи встречаются при решении систем уравнений в частных производных [12]. В работе [13] предложена идея непрерывных расширений структурных методов, дающих приближение к решению на всём интервале интегрирования, и продемонстрировано их применение для интегрирования систем с запаздыванием.

# Глава 1. Структурные методы интегрирования

## 1.1 Классы структурных особенностей систем ОДУ

Исторически первым использованием структуры системы обыкновенных дифференциальных уравнений для построения экономичных схем численного интегрирования можно считать методы Нюстрёма. В случае системы уравнений второго порядка $y'' = f(x,y,y')$ алгоритм, предложенный Нюстрёмом, позволяет сократить машинную память, используемую при вычислениях. Однако настоящее преимущество возникает в том случае, когда правая часть не зависит от первой производной $y'$. Для систем вида

$$y'' = f(x,y) \tag{1.1}$$

удаётся сократить также и количество вычислений функции $f$, которые, как правило, занимают основную часть времени вычислительного процесса. Например, в то время, когда классические методы Рунге—Кутты для достижения пятого порядка точности требуют минимум шести вычислений правой части, метод Нюстрёма требует только четырёх, что даёт экономию в полтора раза.

Под термином «система ОДУ со структурными особенностями» обычно полагают то, что рассматриваемая система принадлежит к одному из трёх классов $\mathfrak{A}$, $\mathfrak{B}$ или $\mathfrak{C}$.

### 1.1.1 Класс $\mathfrak{A}$

К классу $\mathfrak{A}$ относятся системы с перекрёстной структурой зависимостей:

$$\begin{cases} y_1' = f_1(x,y_2), \\ y_2' = f_2(x,y_1), \end{cases} \tag{1.2}$$

$$
\begin{aligned}
x &\in [X_0,X_k] \subset \mathbb{R}, \\
y_s &: [X_0,X_k] \longrightarrow \mathbb{R}^{r_s}, \quad s = 1,2, \\
f_1 : [X_0,X_k] \times \mathbb{R}^{r_2} &\longrightarrow \mathbb{R}^{r_1}, \\
f_2 : [X_0,X_k] \times \mathbb{R}^{r_1} &\longrightarrow \mathbb{R}^{r_2}.
\end{aligned}
$$

Можно видеть, что этот класс является обобщением систем второго порядка, не зависящих от первой производной. Каждая система вида $y'' = f(x,y)$ с помощью введения обозначений $y_1 = y$, $y_2 = y'$ может быть записана в виде системы первого порядка с перекрёстной системой связей

$$
\begin{cases}
y_1' = y_2, \\
y_2' = f_2(x,y_1).
\end{cases} \tag{1.3}
$$

Важным качеством при этом является то, что, в отличие от систем второго порядка, две группы переменных могут иметь разную размерность.

### 1.1.2 Класс $\mathfrak{B}$

К классу $\mathfrak{B}$ относятся системы со следующей структурой зависимостей:

$$
\begin{cases}
y_i' = f_i(x,y_1,\ldots,y_{i-1},y_{l+1},\ldots,y_n), & i = 1,2,\ldots,l, \\
y_j' = f_j(x,y_1,\ldots,y_{j-1}), & j = l+1,\ldots,n,
\end{cases} \tag{1.4}
$$

$$
\rho = \sum_{\nu=1}^{n} r_\nu, \quad x \in [X_0,X_k] \subset \mathbb{R}, \qquad y_s : [X_0,X_k] \longrightarrow \mathbb{R}^{r_s}, \quad s = 1,\ldots,n,
$$

$$
f_i : [X_0,X_k] \times \mathbb{R}^{\rho - \hat{r}^i} \longrightarrow \mathbb{R}^{r_i}, \qquad \hat{r}^i = \sum_{\nu=i}^{l} r_\nu, \qquad i = 1,2,\ldots,l,
$$

$$
f_j : [X_0,X_k] \times \mathbb{R}^{\rho - \bar{r}^j} \longrightarrow \mathbb{R}^{r_j}, \qquad \bar{r}^j = \sum_{\nu=j}^{n} r_\nu, \qquad j = l+1,\ldots,n.
$$

Этот класс является обобщением класса $\mathfrak{A}$: в системах такого вида неизвестные двух групп могут зависеть не только от противоположной группы, но и от неизвестных своей группы, имеющих меньший индекс.

### 1.1.3 Класс $\mathfrak{C}$

Следующее обобщение — класс $\mathfrak{C}$, к нему относятся системы с т.н. «общей» группой $y_0$:

$$\begin{cases} y_0' = f_0(x,y_0,\ldots,y_n), & & (1.5)\\ y_i' = f_i(x,y_0,\ldots,y_{i-1},y_{l+1},\ldots,y_n), & i=1,2,\ldots,l, & (1.6)\\ y_j' = f_j(x,y_0,\ldots,y_{j-1}), & j=l+1,\ldots,n, & (1.7)\end{cases}$$

$$\rho=\sum_{\nu=0}^{n} r_\nu,\quad x\in[X_0,X_k]\subset\mathbb{R},\qquad y_s:[X_0,X_k]\longrightarrow\mathbb{R}^{r_s},\quad s=0,\ldots,n,$$

$$f_0:[X_0,X_k]\times\mathbb{R}^{\rho}\longrightarrow\mathbb{R}^{r_0},$$

$$f_i:[X_0,X_k]\times\mathbb{R}^{\rho-\hat{r}^i}\longrightarrow\mathbb{R}^{r_i},\qquad \hat{r}^i=\sum_{\nu=i}^{l} r_\nu,\qquad i=1,2,\ldots,l,$$

$$f_j:[X_0,X_k]\times\mathbb{R}^{\rho-\bar{r}^j}\longrightarrow\mathbb{R}^{r_j},\qquad \bar{r}^j=\sum_{\nu=j}^{n} r_\nu,\qquad j=l+1,\ldots,n.$$

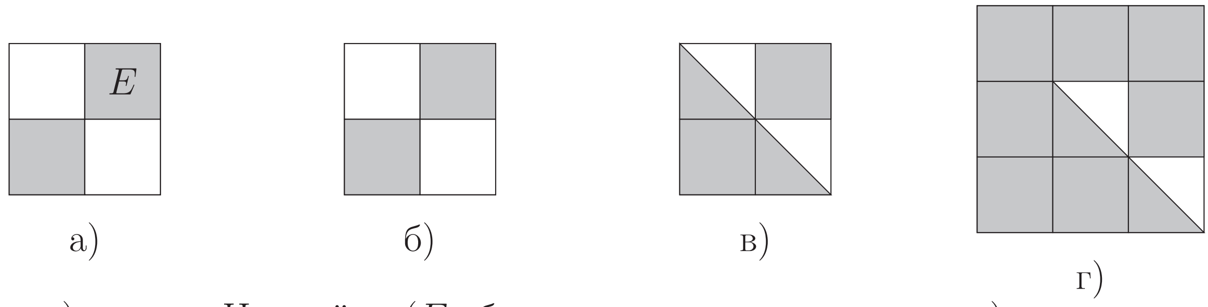
Последовательная вложенность классов $\mathfrak{A}$, $\mathfrak{B}$ и $\mathfrak{C}$ позволяет использовать одни и те же названия для групп переменных $y_k$: здесь и далее мы будем называть их нулевой (1.5), первой (1.6) и второй (1.7). В литературе также часто называют нулевую группу общей, а первую и вторую обозначают как $i$-ю и $j$-ю.

Нулевая группа — единственная, в которой производная какого-либо параметра системы может зависеть от него самого.

В случаях классов $\mathfrak{B}$ и $\mathfrak{C}$ любая группа может быть вырожденной: например, система вида

$$\begin{cases} y_0' = f_0(t,y_0,y_1),\\ y_1' = f_0(t,y_0)\end{cases}$$

принадлежит классу $\mathfrak{C}$ (группа 2 отсутствует).



а) методы Нюстрёма ($E$ обозначает единичную матрицу);
б) класс $\mathfrak{A}$; в) класс $\mathfrak{B}$; г) класс $\mathfrak{C}$

Рисунок 1.1 — Структуры допустимых зависимостей в СОДУ

### 1.1.4 Обзор моделей

Поведение многих математических моделей описывается дифференциальными уравнениями второго порядка вида вида 1.1. Как уже было показано, каждая такая система может быть приведена к перекрёстному виду 1.2 (класс $\mathfrak{A}$) с помощью введения новых переменных. Существуют также и модели, описываемые системами класса $\mathfrak{A}$, но не являющиеся системами второго порядка. Примером такой задачи может являться движение неуправляемого космического аппарата (КА) вокруг точек либрации $L_1$ и $L_2$ [14].

Более сложные задачи относительного движения нескольких тел также часто могут быть приведены к виду 1.4 (класс $\mathfrak{B}$). Например:

1. Модель Хилла—Клохесси—Уилтшира: движение объектов в окрестности искусственного спутника Земли [15].
2. Модель Чаунера—Хемпеля: движение двух спутников по эллиптической орбите [16].
3. Модель Швайгарта—Седвика: движение КА в геопотенциальном поле [17].
4. Орбита Аренсторфа: периодическое движение КА в гравитационном поле Земли и Луны [18].

Все эти модели описываются системами второго порядка, однако при соответствующем переопределении переменных (менее очевидном, чем описанное выше для класса $\mathfrak{A}$) могут быть отнесены к классу $\mathfrak{B}$. Пример такого переопределения для орбиты Аренсторфа будет рассмотрен ниже при проведении численных экспериментов.

Очевидно также, что любая система ОДУ, описывающая какую-либо модель, может быть представлена как система класса $\mathfrak{C}$, содержащая лишь нулевую группу. Однако представление системы как имеющей только нулевую группу не принесёт результата по сравнению с классическими методами Рунге—Кутты, поскольку преимущество структурных методов заключается именно в экономии на вычислениях правых частей первой и второй группы.

Многие встречающиеся в реальной жизни системы имеют структуру, не позволяющую сразу выделить первую или вторую группу уравнений, однако с помощью простого переопределения переменных удаётся привести их к необходимому виду. К примеру, если зависимости производных представляют собой двудольный граф, она является системой класса $\mathfrak{A}$. Существует также алгоритм, позволяющий переопределять параметры системы таким способом, чтобы первая и вторая группа имели максимальный возможный размер [19].

## 1.2 Метод интегрирования

**Метод интегрирования систем класса $\mathfrak{C}$.** Будем считать, что нам известно точное решение $y_s(x)$, $s = 0, \ldots, n$, системы 1.5—1.7 в точке $x \in [X_0, X_k]$. Не умаляя общности рассуждений, для простоты вывода будем считать $r_s = 1$, $s = 0, \ldots, n$.

Для численного интегрирования систем 1.5—1.7 рассматривается явный одношаговый метод [20]. В предположении достаточной гладкости правой части рассматриваемой системы приближение $\tilde{y}_s$ к точному решению $y_s(x+h)$, $s = 1, \ldots, n$ в точке $x + h \in [X_0, X_k]$ ищется в виде:

$$y_0(x+h) \approx \tilde{y}(x+h) = y_0(x) + h \sum_{\nu=1}^{m_0} b_{0\nu} k_{0,\nu}(h), \tag{1.8}$$

$$y_i(x+h) \approx \tilde{y}_i(x+h) = y_i(x) + h \sum_{\nu=1}^{m_1} b_{1\nu} k_{i,\nu}(h), \qquad i = 1, \ldots, l, \tag{1.9}$$

$$y_j(x+h) \approx \tilde{y}_j(x+h) = y_j(x) + h \sum_{\nu=1}^{m_1} b_{2\nu} k_{j,\nu}(h), \qquad j = l+1, \ldots, n, \tag{1.10}$$

причём $k_{s,w} \equiv k_{s,w}(h)$ вычисляются в строгой последовательности

$$k_{0,1}, \ldots,\ k_{n,1},\ k_{0,2}, \ldots, k_{n,2}, k_{0,3}, k_{1,3}, \ldots \tag{1.11}$$

по формулам

$$\begin{aligned} k_{0,\nu} = f_0\Big(&x + c_{0\nu}h,\\ &y_0 + h\sum_{\mu=1}^{\nu-1} a_{00\nu\mu}k_{0,\mu}(h),\\ &y_1 + h\sum_{\mu=1}^{\nu-1} a_{01\nu\mu}k_{1,\mu}(h), \ldots, y_{i-1} + h\sum_{\mu=1}^{\nu-1} a_{01\nu\mu}k_{l,\mu}(h),\\ &y_{l+1} + h\sum_{\mu=1}^{\nu-1} a_{02\nu\mu}k_{l+1,\mu}(h), \ldots, y_n + h\sum_{\mu=1}^{\nu-1} a_{02\nu\mu}k_{n,\mu}(h)\Big), \end{aligned} \tag{1.12}$$

$$\begin{aligned} k_{i,\nu} = f_i\Big(&x + c_{1\nu}h,\\ &y_0 + h\sum_{\mu=1}^{\nu} a_{10\nu\mu}k_{0,\mu}(h),\\ &y_1 + h\sum_{\mu=1}^{\nu} a_{11\nu\mu}k_{1,\mu}(h), \ldots, y_{i-1} + h\sum_{\mu=1}^{\nu} a_{11\nu\mu}k_{i-1,\mu}(h),\\ &y_{l+1} + h\sum_{\mu=1}^{\nu-1} a_{12\nu\mu}k_{l+1,\mu}(h), \ldots, y_n + h\sum_{\mu=1}^{\nu-1} a_{12\nu\mu}k_{n,\mu}(h)\Big), \quad i = 1, \ldots, l, \end{aligned} \tag{1.13}$$

$$\begin{aligned} k_{j,\nu} = f_j\Big(&x + c_{2\nu}h,\\ &y_0 + h\sum_{\mu=1}^{\nu} a_{20\nu\mu}k_{0,\mu}(h),\\ &y_1 + h\sum_{\mu=1}^{\nu} a_{21\nu\mu}k_{1,\mu}(h), \ldots, y_{i-1} + h\sum_{\mu=1}^{\nu} a_{21\nu\mu}k_{l,\mu}(h),\\ &y_{l+1} + h\sum_{\mu=1}^{\nu} a_{22\nu\mu}k_{l+1,\mu}(h), \ldots, y_n + h\sum_{\mu=1}^{\nu} a_{22\nu\mu}k_{j-1,\mu}(h)\Big), \quad j = l+1, \ldots, n. \end{aligned} \tag{1.14}$$

Параметрами метода являются коэффициенты $a_{w_1 w_2 \nu\mu}$ (матрицы этих коэффициентов будем обозначать как $A_{w_1 w_2}$), $b_{w_1\nu}$, $c_{w_1\nu}$, $w_1, w_2 \in \{0,1,2\}$. Говорят, что метод имеет порядок $p$, если

$$|y_s(x+h) - \tilde{y}_s| \approx O(h^{p+1}), \qquad s = 0, \ldots, n.$$

Строгий порядок вычисления значений $k$ связывает количество этапов по каждой группе: $m_1 \leqslant m_0 \leqslant m_1 + 1$. Основная идея структурного подхода заключается в алгоритмическом использовании структурных особенностей математической модели для сокращения числа обращений к правым частям первой и второй групп уравнений, поэтому мы будем считать далее $m = m_0 = m_1 + 1$.

Для того, чтобы метод был явным, необходимо, чтобы матрицы $A_{10}$, $A_{20}$, $A_{11}$, $A_{21}$ и $A_{22}$ были *нижнетреугольными*, а матрицы $A_{00}$, $A_{01}$, $A_{02}$ и $A_{12}$ — *строго нижнетреугольными*. В этом случае к моменту вычисления каждого из выражений 1.12—1.14 аргументы в их правых частях будут уже известны. В табл. 1 приведён компактный общий вид коэффициентов явного метода при $m = m_0 = m_1 + 1$ (для удобства индексы отделены запятыми). Дополнительное требование $a_{1111} = a_{1011} = 0$ обусловлено часто используемым предположением $c_{w_1\nu} = \sum\limits_{\mu} a_{w_1 w_2 \nu\mu}, \quad w_1, w_2 \in \{0,1,2\}$ и тем, что $a_{1211} = 0$.

**Замечание 1.** В случае, когда система состоит только из нулевой группы, метод вырождается в классический метод Рунге—Кутты с коэффициентами $A = A_{00}$, $b = b_0$, $c = c_0$ и количеством этапов $s = m_0$:

$$y(x+h) \approx \tilde{y}(x+h) = y(x) + h\sum_{\nu=1}^{s} b_\nu k_\nu(h), \tag{1.15}$$

$$k_\nu = f\left(x + c_\nu h, y + h\sum_{\mu=1}^{s} a_{\nu\mu} k_\mu(h)\right),$$

**Замечание 2.** В силу того, что системы класса $\mathfrak{B}$ отличаются от систем класса $\mathfrak{C}$ лишь отсутствием нулевой группы, рассмотренный метод можно применить и к ним, просто исключив из всех формул параметры с индексом 0. То же касается и систем класса $\mathfrak{A}$: достаточно вдобавок исключить все параметры, использующие зависимости внутри первой группы и внутри второй. Табличные записи коэффициентов для обеих схем будут содержать коэффициенты $b_{1\nu}$, $b_{2\nu}$, $c_{1\nu}$, $c_{2\nu}$, $a_{12\nu\mu}$, $a_{21\nu\mu}$, а для схемы класса $\mathfrak{B}$ дополнительно $a_{11\nu\mu}$ и $a_{22\nu\mu}$. Требования к треугольности матриц для явных методов останутся теми же.

Таблица 1 — Коэффициенты явного метода класса $\mathfrak{C}$ при $m = m_0 = m_1 + 1$

| $c_{0,\nu}$ | $a_{0,0,\nu,\mu}$ | | | | $a_{0,1,\nu,\mu}$ | | | | $a_{0,2,\nu,\mu}$ | | | | $b_{0,\nu}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $0$ | $0$ | $0$ | $\dots$ | $0$ | $0$ | $0$ | $\dots$ | $0$ | $0$ | $0$ | $\dots$ | $0$ | $b_{0,1}$ |
| $c_{0,2}$ | $a_{0,0,2,1}$ | $0$ | $\dots$ | $0$ | $a_{0,1,2,1}$ | $0$ | $\dots$ | $0$ | $a_{0,2,2,1}$ | $0$ | $\dots$ | $0$ | $b_{0,2}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $c_{0,m-1}$ | $a_{0,0,m-1,1}$ | $a_{0,0,m-1,2}$ | $\dots$ | $0$ | $a_{0,1,m-1,1}$ | $a_{0,1,m-1,2}$ | $\dots$ | $0$ | $a_{0,2,m-1,1}$ | $a_{0,2,m-1,2}$ | $\dots$ | $0$ | $b_{0,m-1}$ |
| $c_{0,m}$ | $a_{0,0,m,1}$ | $a_{0,0,m,2}$ | $\dots$ | $a_{0,0,m,m-1}$ | $a_{0,1,m,1}$ | $a_{0,1,m,2}$ | $\dots$ | $a_{0,1,m,m-1}$ | $a_{0,2,m,1}$ | $a_{0,2,m,2}$ | $\dots$ | $a_{0,2,m,m-1}$ | $b_{0,m}$ |
| $c_{1,\nu}$ | $a_{1,0,\nu,\mu}$ | | | | $a_{1,1,\nu,\mu}$ | | | | $a_{1,2,\nu,\mu}$ | | | | $b_{1,\nu}$ |
| $0$ | $0$ | $0$ | $\dots$ | $0$ | $0$ | $0$ | $\dots$ | $0$ | $0$ | $0$ | $\dots$ | $0$ | $b_{1,1}$ |
| $c_{1,2}$ | $a_{1,0,2,1}$ | $a_{1,0,2,2}$ | $\dots$ | $0$ | $a_{1,1,2,1}$ | $a_{1,1,2,2}$ | $\dots$ | $0$ | $a_{1,2,2,1}$ | $0$ | $\dots$ | $0$ | $b_{1,2}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $c_{1,m-2}$ | $a_{1,0,m-2,1}$ | $a_{1,0,m-2,2}$ | $\dots$ | $0$ | $a_{1,1,m-2,1}$ | $a_{1,1,m-2,2}$ | $\dots$ | $0$ | $a_{1,2,m-2,1}$ | $a_{1,2,m-2,2}$ | $\dots$ | $0$ | $b_{1,m-2}$ |
| $c_{1,m-1}$ | $a_{1,0,m-1,1}$ | $a_{1,0,m-1,2}$ | $\dots$ | $a_{1,0,m-1,m-1}$ | $a_{1,1,m-1,1}$ | $a_{1,1,m-1,2}$ | $\dots$ | $a_{1,1,m-1,m-1}$ | $a_{1,2,m-1,1}$ | $a_{1,2,m-1,2}$ | $\dots$ | $0$ | $b_{1,m-1}$ |
| $c_{2,\nu}$ | $a_{2,0,\nu,\mu}$ | | | | $a_{2,1,\nu,\mu}$ | | | | $a_{2,2,\nu,\mu}$ | | | | $b_{2,\nu}$ |
| $c_{2,1}$ | $a_{2,0,1,1}$ | $0$ | $\dots$ | $0$ | $a_{2,1,1,1}$ | $0$ | $\dots$ | $0$ | $a_{2,2,1,1}$ | $0$ | $\dots$ | $0$ | $b_{2,1}$ |
| $c_{2,2}$ | $a_{2,0,2,1}$ | $a_{2,0,2,2}$ | $\dots$ | $0$ | $a_{2,1,2,1}$ | $a_{2,1,2,2}$ | $\dots$ | $0$ | $a_{2,2,2,1}$ | $a_{2,2,2,2}$ | $\dots$ | $0$ | $b_{2,2}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $c_{2,m-2}$ | $a_{2,0,m-2,1}$ | $a_{2,0,m-2,2}$ | $\dots$ | $0$ | $a_{2,1,m-2,1}$ | $a_{2,1,m-2,2}$ | $\dots$ | $0$ | $a_{2,2,m-2,1}$ | $a_{2,2,m-2,2}$ | $\dots$ | $0$ | $b_{2,m-2}$ |
| $c_{2,m-1}$ | $a_{2,0,m-1,1}$ | $a_{2,0,m-1,2}$ | $\dots$ | $a_{2,0,m-1,m-1}$ | $a_{2,1,m-1,1}$ | $a_{2,1,m-1,2}$ | $\dots$ | $a_{2,1,m-1,m-1}$ | $a_{2,2,m-1,1}$ | $a_{2,2,m-1,2}$ | $\dots$ | $a_{2,2,m-1,m-1}$ | $b_{2,m-1}$ |

# Глава 2. Схемы шестого порядка класса $\mathfrak{B}$

## 2.1 Условия порядка

Коэффициенты схемы интегрирования типа Рунге—Кутты должны удовлетворять системе условий порядка. Эти условия представляют собой алгебраические уравнения, получающиеся при приравнивании соответствующих членов ряда Тейлора точного $y$ и приближённого $\tilde{y}$ решений вплоть до $p$-го, где $p$ — порядок метода. Размеры системы условий порядка для классических и структурных методов типа Рунге—Кутты приведены в таблице 2.

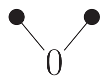
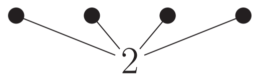
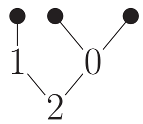
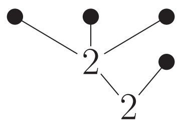
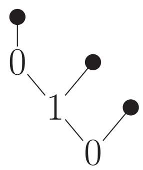
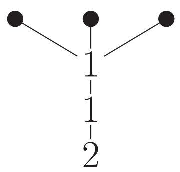
Таблица 2 — Размеры системы условий порядка

| класс метода | количество условий порядка | | | | | |
|---|---|---|---|---|---|---|
| $RK$ | 1 | 2 | 4 | 8 | 17 | 37 |
| $\mathfrak{A}$ | 2 | 4 | 8 | 16 | 34 | 74 |
| $\mathfrak{B}$ | 2 | 4 | 10 | 28 | 88 | 292 |
| $\mathfrak{C}$ | 3 | 6 | 18 | 66 | 276 | 1224 |
| порядок метода | 1 | 2 | 3 | 4 | 5 | 6 |

Быстрый рост количества уравнений с ростом порядка вынуждает автоматизировать процесс их получения. Джон Бутчер развил теорию *помеченных деревьев*, с помощью которой можно однозначно сопоставить каждый элементарный дифференциал $\Phi$ некоему древовидному графу $\tau(\Phi)$. В классической теории для получения полного набора условий требуется выписать все помеченные деревья порядка не выше $p$ (*порядком дерева* называется количество его вершин) и соответствующие им уравнения. Примеры соответствия деревьев и уравнений приведены в таблице 3.

В работе [21] предложена модификация этой теории, предназначенная для методов класса $\mathfrak{C}$ и представлены условия порядка до пятого включительно. В случае структурных методов класса $\mathfrak{C}$ помимо получения полного набора деревьев необходимо сопоставить индекс 0, 1 или 2 каждой вершине, имеющей потомков. Большим количеством возможных расстановок индексов и объясняется возрастание размера системы при переходе к структурным методам.

Таблица 3 — Примеры построения условий порядка

| $\tau$ | $\gamma(\tau)$ | условие порядка |
|---|---|---|
| 0 | 1 | $\sum\limits_{\nu} b_{0\nu} = 1$ |
|  | 2 | $\sum\limits_{\nu} b_{1\nu} c_{1\nu} = \frac{1}{2}$ |
|  | 3 | $\sum\limits_{\nu} b_{0\nu} c_{0\nu}^2 = \frac{1}{3}$ |
|  | 6 | $\sum\limits_{\nu} b_{1\nu} \sum\limits_{\mu} a_{11\nu\mu} c_{1\mu} = \frac{1}{6}$ |
|  | 5 | $\sum\limits_{\nu} b_{2\nu} c_{2\nu}^4 = \frac{1}{5}$ |
|  | 36 | $\sum\limits_{\nu} b_{2\nu} (\sum\limits_{\mu} a_{20\nu\mu} c_{0\mu}^2) \cdot (\sum\limits_{\mu} a_{21\nu\mu} c_{1\mu}) = \frac{1}{36}$ |
|  | 24 | $\sum\limits_{\nu} b_{2\nu} c_{2\nu}^1 \sum\limits_{\mu} a_{22\nu\mu} c_{2\mu}^3 = \frac{1}{24}$ |
|  | 48 | $\sum\limits_{\nu} b_{0\nu} c_{0\nu} \sum\limits_{\mu} a_{01\nu\mu} c_{1\mu} \sum\limits_{\xi} a_{10\mu\xi} c_{0\xi} = \frac{1}{48}$ |
|  | 120 | $\sum\limits_{\nu} b_{2\nu} \sum\limits_{\mu} a_{21\nu\mu} \sum\limits_{\xi} a_{11\mu\xi} c_{1\xi}^3 = \frac{1}{120}$ |

Отметим, что систему условий для методов класса $\mathfrak{B}$ того же порядка можно получить из неё, отбросив все уравнения, соответствующие деревьям, содержащим вершины с индексом 0 (поскольку используются только индексы 1 и 2).

Аналогично, систему условий порядка для методов класса $\mathfrak{A}$ можно получить из последней отбрасыванием деревьев, имеющих рёбра вида $1-1$ и $2-2$. Таким образом останутся только те деревья, рёбра которых соединяют вершины с разными индексами.

## 2.2 Семейство схем класса $\mathfrak{B}$

Система условий шестого порядка для методов класса $\mathfrak{B}$ состоит из 292 алгебраических уравнений. Было получено семипараметрическое семейство решений этой системы со свободными параметрами $c_{12} = \alpha_1$ $c_{13} = \alpha_2$ $c_{14} = \alpha_3$ $c_{23} = \alpha_4$ $c_{24} = \alpha_5$ $a_{1132} = \alpha_6$ и $a_{2243} = \alpha_7$. В прил. А представлены значения ненулевых параметров метода.

## 2.3 Интегрирование с переменным шагом. Оценка локальной погрешности

Регулирование длины шага интегрирования является основным инструментом при численном решении задачи Коши. Малая длина шага, как правило, даёт большую точность, однако приводит и к большим вычислительным затратам. Неоправданно высокие вычислительные затраты в свою очередь могут привести к тому, что основной составляющей глобальной погрешности станет накопившаяся ошибка округления.

Кроме того, во многих моделях интервал интегрирования можно разделить на «быстрые» и «медленные» участки: в первом случае переменные меняются «быстро», и для удержания локальной погрешности в заданных рамках приходится делать малые шаги, во втором — «медленно», и можно увеличить длину шага, чтобы ускорить вычислительный процесс.

**Вложенные методы.** Обозначим результат применения метода Рунге—Кутты 1.15 на одном шаге длины $h$ как

$$\tilde{y} = RK(x_0, y_0, f, A, b, h), \tag{2.1}$$

где $y_0$ и $f$ — параметры задачи Коши, $A$ и $b$ — коэффициенты конкретной схемы Рунге—Кутты, $h$ — длина шага итегрирования.

Главный параметр, в соответствии с которым изменяют длину шага — это локальная погрешность, которая на каждом шаге не должна превышать некоей заданной величины $tol$. Э. Фельберг предложил оценивать локальную погрешность, вычисляя одновременно два приближения: $\tilde{y}_1 = RK(x_0, y_0, f, A, b, h)$ порядка $p$ и $\tilde{y}_2 = RK(x_0, y_0, f, A, d, h)$ порядка $q < p$, то есть с помощью двух схем Рунге—Кутты, отличающихся лишь весовыми коэффициентами $b$ и $d$. Главные члены локальных погрешностей для этих приближений будут соответственны равны $\Delta_1 = C_1 h^{p+1}$ и $\Delta_2 = C_2 h^{q+1}$, где $C_1$ и $C_2$ — некоторые скаляры.

Матрицы коэффициентов $A$ для обеих схем одинаковы, поэтому оба приближения можно получить, не затрачивая лишнее время на вычисления правых частей СОДУ. Подобные пары получили название *вложенных методов*. Поскольку $\tilde{y}_1$ имеет больший порядок, главный член их разности $\Delta = \tilde{y}_1 - \tilde{y}_2$ будет равен $C_2 h^{q+1}$ и может быть использован в качестве оценки локальной погрешности приближённого решения $\tilde{y}_2$. В дальнейших работах (Дорманд, Принс) в качестве приближения стали использовать $\tilde{y}_1$, поскольку его порядок выше, а величина $\Delta$ приобрела значение уже не оценки локальной погрешности, а некоего контрольного члена, применяемого для регулирования длины шага.

Другим отличительным свойством вложенных методов типа Дорманда—Принса является их свойство **FSAL** (First Same As Last). $s$-этапную схему интегрирования дополняют ещё одним этапом $k_{s+1} = f(x + h, y_0 + h \sum_{\mu=1}^{s} b_\mu k_\mu(h))$. Фактически $k_{s+1}$ совпадает с результатом первого вычисления правой части, которое будет сделано на следующем шаге. А значит, на всех шагах начиная со второго можно по-прежнему совершать $s$ вычислений правой части, и вычислительные затраты почти не растут.

## 2.4 Схема $RKB6(4)\{7F\}$

На основе решения системы условий порядка, полученного при значениях свободных параметров $\alpha_1 = \frac{2}{9}$, $\alpha_2 = \frac{1}{6}$, $\alpha_3 = \frac{1}{2}$, $\alpha_4 = \frac{1}{6}$, $\alpha_5 = \frac{1}{2}$, $\alpha_6 = 0$, $\alpha_7 = \frac{2}{3}$ построена семиэтапная расчетная схема метода $RKB6(4)\{7F\}$. Это название означает:

- $RK$ — метод типа Рунге—Кутты;
- $B$ — метод класса $\mathfrak{B}$;
- 6 — порядок точности приближения на шаге;
- 4 — порядок точности вложенного метода—оценщика;
- 7 — количество используемых значений правой части СОДУ на шаге;
- $F$ — метод обладает свойством FSAL, фактически на каждом шаге начиная со второго требуется только шесть обращений к правой части СОДУ.

Значения параметров были выбраны из двух соображений: минимизация невязки по условиям седьмого порядка (это позволяет уменьшить норму главного члена локальной погрешности [22]) и краткость записи. Коэффициенты метода представлены в таблице 4.

**Замечание.** Благодаря тому, что все коэффициенты $a_{w_1 w_2 1 \mu}$ равны нулю, построенный метод избегает существенного недостатка, которым могут обладать другие вложенные структурные методы со свойством FSAL. В случае, когда среди коэффициентов $a_{w_1 w_2 1 \mu}$ есть ненулевые, нельзя утверждать, что в выражениях 1.12—1.14 обращения к правым частям СОДУ на смежных шагах происходят с одинаковыми значениями аргументов. Это приводит к необходимости модифицировать алгоритм адаптивного выбора шага интегрирования [23].

Таблица 4 — Коэффициенты метода $RKB6(4)\{7F\}$

| $c_{1\nu}$ | $a_{11\nu\mu}$ | | | | | | $a_{12\nu\mu}$ | | | | | | $b_{1\nu}$ | $d_{1\nu}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $0$ | | | | | | | | | | | | | $\frac{7}{150}$ | $\frac{13}{200}$ |
| $\frac{2}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ | | | | | $\frac{2}{9}$ | | | | | | $0$ | $0$ |
| $\frac{1}{6}$ | $\frac{1}{12}$ | $0$ | $\frac{1}{12}$ | | | | $\frac{5}{48}$ | $\frac{1}{16}$ | | | | | $\frac{27}{100}$ | $\frac{183}{800}$ |
| $\frac{1}{2}$ | $\frac{-1}{44}$ | $0$ | $\frac{9}{22}$ | $\frac{5}{44}$ | | | $\frac{37}{176}$ | $\frac{243}{176}$ | $\frac{-12}{11}$ | | | | $\frac{11}{30}$ | $\frac{33}{80}$ |
| $\frac{5}{6}$ | $\frac{7}{36}$ | $0$ | $0$ | $\frac{5}{9}$ | $\frac{1}{12}$ | | $\frac{-635}{432}$ | $\frac{-167}{16}$ | $\frac{100}{9}$ | $\frac{44}{27}$ | | | $\frac{27}{100}$ | $\frac{183}{800}$ |
| $1$ | $\frac{-3}{7}$ | $0$ | $\frac{9}{8}$ | $\frac{-5}{28}$ | $\frac{27}{56}$ | | $\frac{29}{4}$ | $\frac{1377}{28}$ | $\frac{-1425}{28}$ | $\frac{-11}{2}$ | $\frac{27}{28}$ | | $\frac{7}{150}$ | $\frac{7}{300}$ |
| $1$ | $\frac{7}{150}$ | $0$ | $\frac{27}{100}$ | $\frac{11}{30}$ | $\frac{27}{100}$ | $\frac{7}{150}$ | $\frac{7}{150}$ | $0$ | $\frac{27}{100}$ | $\frac{11}{30}$ | $\frac{27}{100}$ | $\frac{7}{150}$ | $0$ | $\frac{1}{24}$ |

| $c_{2\nu}$ | $a_{21\nu\mu}$ | | | | | | $a_{22\nu\mu}$ | | | | | | $b_{2\nu}$ | $d_{2\nu}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $0$ | | | | | | | | | | | | | $\frac{7}{150}$ | $\frac{13}{200}$ |
| $\frac{2}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ | | | | | $\frac{1}{9}$ | $\frac{1}{9}$ | | | | | $0$ | $0$ |
| $\frac{1}{6}$ | $\frac{7}{48}$ | $\frac{3}{16}$ | $\frac{-1}{6}$ | | | | $\frac{7}{48}$ | $\frac{3}{16}$ | $\frac{-1}{6}$ | | | | $\frac{27}{100}$ | $\frac{183}{800}$ |
| $\frac{1}{2}$ | $\frac{-31}{176}$ | $\frac{-81}{176}$ | $\frac{45}{44}$ | $\frac{5}{44}$ | | | $\frac{-185}{1584}$ | $\frac{-123}{880}$ | $\frac{2}{3}$ | $\frac{89}{990}$ | | | $\frac{11}{30}$ | $\frac{33}{80}$ |
| $\frac{5}{6}$ | $\frac{73}{144}$ | $\frac{15}{16}$ | $\frac{-5}{4}$ | $\frac{5}{9}$ | $\frac{1}{12}$ | | $\frac{1031}{3888}$ | $\frac{-53}{144}$ | $\frac{65}{324}$ | $\frac{317}{486}$ | $\frac{1}{12}$ | | $\frac{27}{100}$ | $\frac{183}{800}$ |
| $1$ | $\frac{-39}{28}$ | $\frac{-81}{28}$ | $\frac{279}{56}$ | $\frac{-5}{28}$ | $\frac{27}{56}$ | | $\frac{-29}{63}$ | $\frac{15}{7}$ | $\frac{-103}{168}$ | $\frac{-139}{252}$ | $\frac{27}{56}$ | | $\frac{7}{150}$ | $\frac{7}{300}$ |
| $1$ | $\frac{7}{150}$ | $0$ | $\frac{27}{100}$ | $\frac{11}{30}$ | $\frac{27}{100}$ | $\frac{7}{150}$ | $\frac{7}{150}$ | $0$ | $\frac{27}{100}$ | $\frac{11}{30}$ | $\frac{27}{100}$ | $\frac{7}{150}$ | $0$ | $\frac{1}{24}$ |

# Глава 3. Численное исследование моделей

Для проверки качества построенного метода было произведено сравнение с явными одношаговыми методами шестого порядка Цитураса [24], Вернера [25] и Эль-Миккави [26] а также методом Дорманда—Принса пятого порядка [27].

Помимо набора коэффициентов схема интегрирования должна также обладать алгоритмом выбора длины шага. Для объективной оценки качества работы конкретных методов за основу был взят алгоритм из реализации метода Дорманда—Принса в среде MATLAB — функции $ode45$, основного интегратора в этой среде [27]. Таким образом, различия между методами заключались лишь в затратах на вычисления правых частей СОДУ и достигаемой точности на шаге. На основе метода $RKB6(4)\{7F\}$ реализована модификация $ode46b$ той же функции, позволяющая обращаться к группам системы независимо. Аналогичная модификация функций MATLAB и их внедрение для методов класса $\mathfrak{A}$ были представлены в работе [28]. Текст программы $ode46b$ приведён в прил. Б.

Интерфейс функции $ode46b$ аналогичен интерфейсу $ode45$. Дополнительно от функции вычисления значений правой части СОДУ (аргумент под именем $ode$) требуется возможность вернуть информацию о структуре СОДУ (количество уравнений первой и второй групп) и выдавать значение правой части для конкретных уравнений СОДУ.

Для исследования были выбраны модели, описывающие движение космического аппарата в окрестности точки либрации $L_1$ системы Солнце—Земля и движение КА по орбите Аренсторфа.

## 3.1 Модель 1: точка либрации $L_1$

Плоское движение неуправляемого КА во вращающейся системе координат в окрестности точки либрации $L_1 = (1{,}1)$ системы Солнце–Земля описывается [14] системой обыкновенных дифференциальных уравнений:

$$\begin{cases} \dot{x}_1 = x_2 + y_1, \\ \dot{x}_2 = -x_1 + y_2, \\ \dot{y}_1 = 8(x_1 - 1) + (y_2 - 1), \\ \dot{y}_2 = -4x_2 - y_1. \end{cases} \tag{3.1}$$

При начальных условиях

$$x(0) = (1 + \frac{\sqrt{7} - 3}{2}\varepsilon, 0),$$
$$y(0) = (0, 1 + \varepsilon)$$

КА движется по периодической орбите вокруг $L_1$. При переопределении параметров: $\tilde{x}_1 = x_1, \tilde{x}_2 = y_2, \tilde{x}_3 = x_2, \tilde{x}_4 = y_1$ СОДУ приобретает перекрестную структуру. Для тестирования было выбрано значение $\varepsilon = \frac{1}{100}$.

Поскольку полученная система является линейной, существует возможность на каждом шаге сравнивать численное решение с точным.

## 3.2 Модель 2: орбита Аренсторфа

Плоское движение космического аппарата с координатами $(x_1, x_2)$ в гравитационном поле, создаваемом Землей (0,0) и Луной (1,0) описывается [18] системой обыкновенных дифференциальных уравнений:

$$\begin{cases} \ddot{x}_1 = x_1 + 2\dot{x}_2 - \mu' \dfrac{x_1 + \mu}{D_1} - \mu \dfrac{x_1 - \mu'}{D_2}, \\ \ddot{x}_2 = x_2 - 2\dot{x}_1 - \mu' \dfrac{x_2}{D_1} - \mu \dfrac{x_2}{D_2}, \end{cases} \tag{3.2}$$

где

$$D_1 = \left((x_1 + \mu)^2 + x_2^2\right)^{3/2}, D_2 = \left((x_1 - \mu')^2 + x_2^2\right)^{3/2},$$
$$\mu = 0.012277471, \mu' = 1 - \mu.$$

При начальных условиях

$$x(0) = (0.994, 0),$$
$$\dot{x}(0) = (0, 2.00158510637908252240537862224)$$

КА движется по орбите с периодом $T_{per} = 17.0652165601579625588917206249$. При переопределении параметров: $\tilde{x}_1 = x_1, \tilde{x}_2 = \dot{x}_2, \tilde{x}_3 = x_2, \tilde{x}_4 = \dot{x}_1$ система приобретает структуру, описываемую классом $\mathfrak{B}$.

Поскольку орбита периодична, глобальной погрешностью численного приближения считается отклонение от исходного положения.

## 3.3 Результаты моделирования

На рис. 3.1 и 3.2 приведён график зависимости глобальной погрешности $Err$ от трудоёмкости $N_{fevals}$ (количества обращений к правой части СОДУ).

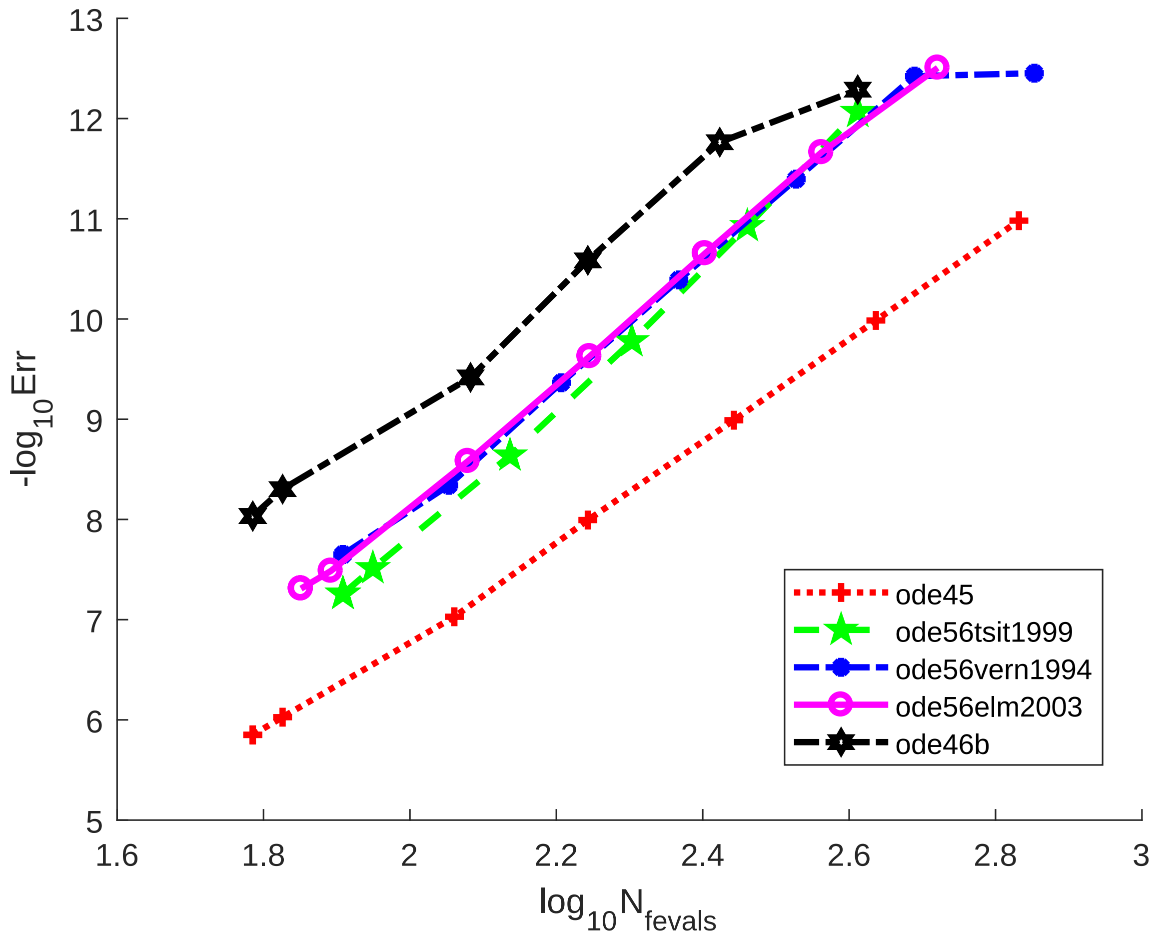



Рисунок 3.1 — Зависимость глобальной погрешности от количества вычислений для модели 1

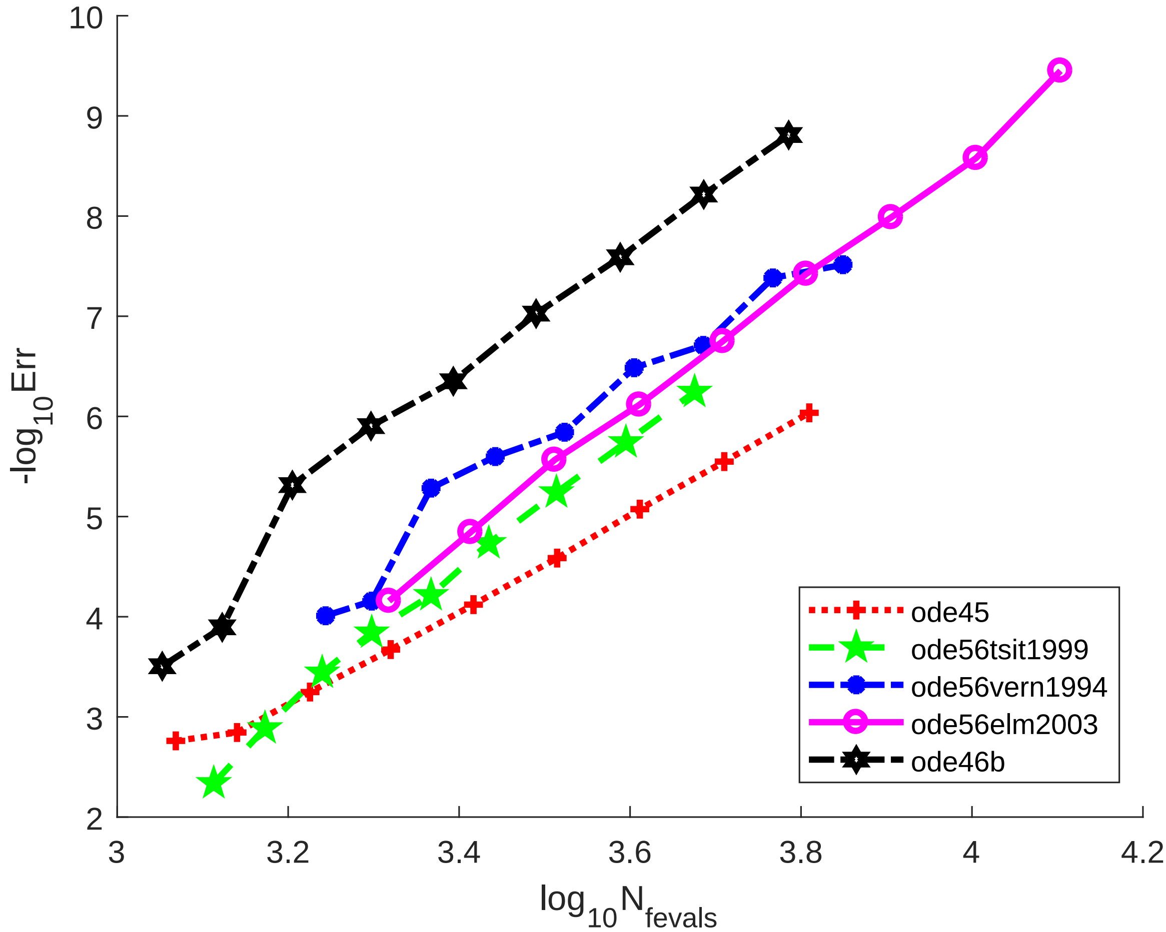



Рисунок 3.2 — Зависимость глобальной погрешности от количества вычислений для модели 2

На рис. 3.3 и 3.4 приведён график изменения длины шага $T_{step}$ во время интегрирования для конкретных значений глобальной погрешности. Поскольку её нельзя выбрать заранее, для каждого метода были подобраны такие значения $tol$, при которых глобальная погрешность примерно равна $10^{-9}$ для модели 1 и $10^{-5}$ для модели 2.

В табл. 5 и 6 приведены значения глобальных погрешностей $Err$, достигаемых при конкретных количествах шагов интегрирования. Помимо некоторого выигрыша в точности метод $ode46b$ требует и меньших вычислительных затрат на каждом шаге по сравнению с методами шестого порядка. В сравнении с $ode45$ метод требует таких же вычислительных затрат, однако имеет больший порядок точности и, как следствие, существенно меньшую глобальную погрешность.

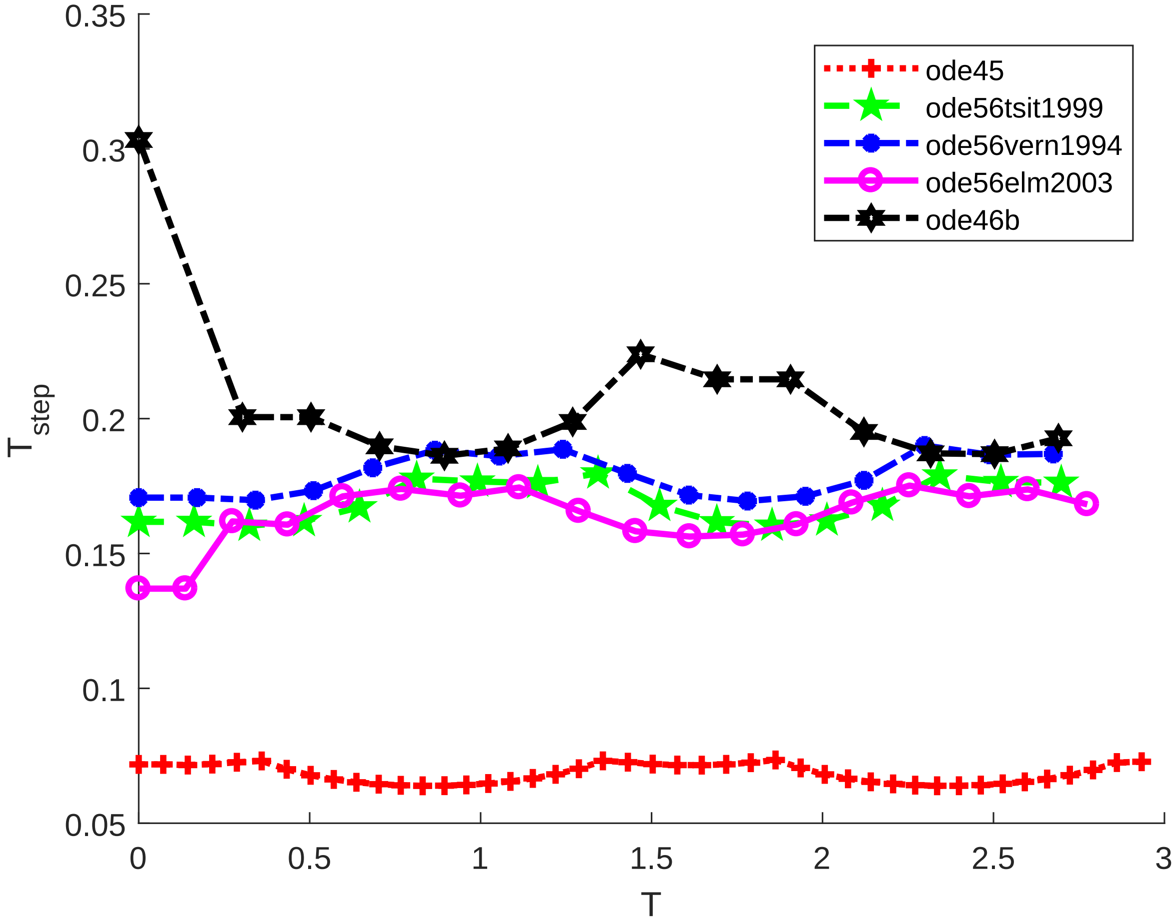



Рисунок 3.3 — Изменение длины шага для модели 1

Таблица 5 — Точность при фиксированном количестве шагов для модели 1

| Кол-во шагов | $-log_{10}(Err)$ | | | | |
|---|---|---|---|---|---|
| | $ode45$ | $ode56tsit1999$ | $ode56vern1994$ | $ode56elm2003$ | $ode46b$ |
| 20 | 7,2431 | 9,3053 | 9,4704 | 9,1986 | 9,7187 |
| 30 | 8,1387 | 10,4688 | 10,5491 | 10,2473 | 10,7620 |
| 40 | 8,7382 | 11,3921 | 11,3264 | 11,0563 | 11,6226 |

Таблица 6 — Точность при фиксированном количестве шагов для модели 2

| Кол-во шагов | $-log_{10}(Err)$ | | | | |
|---|---|---|---|---|---|
| | $ode45$ | $ode56tsit1999$ | $ode56vern1994$ | $ode56elm2003$ | $ode46b$ |
| 400 | 4,0095 | 5,2410 | 5,8257 | 5,1803 | 6,4222 |
| 500 | 4,4443 | 5,8332 | 6,5114 | 5,7427 | 6,9794 |
| 600 | 4,8206 | 6,2936 | 6,6131 | 6,2454 | 7,4493 |

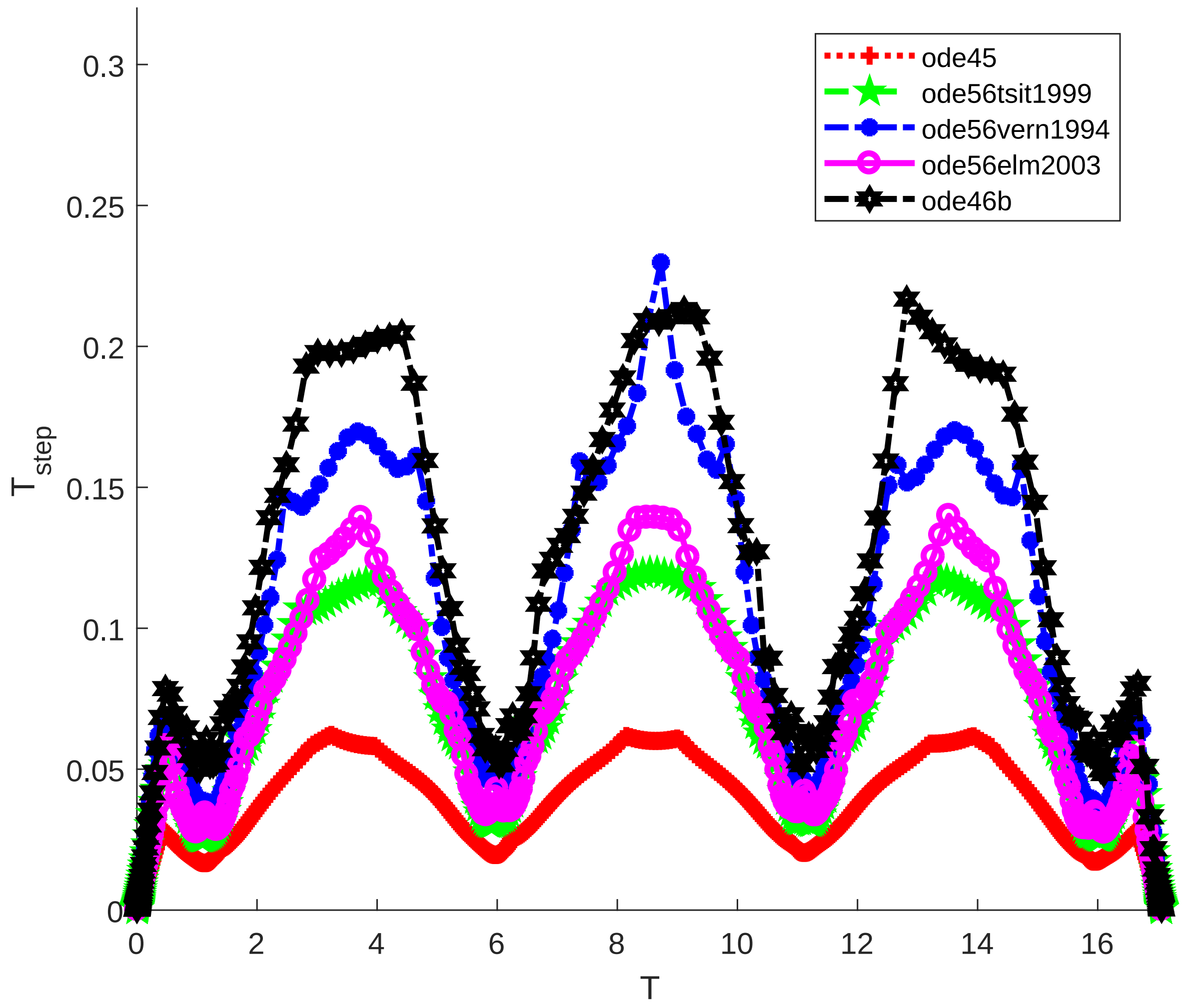



Рисунок 3.4 — Изменение длины шага для модели 2

## Выводы

Для того, чтобы метод имел $p$-й порядок точности, график зависимости логарифма погрешности от логарифма длины шага должен вести себя примерно так же, как прямая линия с углом наклона $\operatorname{arctg} p$. Видно, что на рис. 3.1 и 3.2 наклоны графиков для схем шестого порядка примерно равны. Это значит, что построенная схема действительно имеет шестой порядок.

Выигрыш, достигаемый функцией $ode46b$ (табл. 5, табл. 6) при равном количестве шагов, объясняется тем, что свободные параметры $\alpha_1, \ldots, \alpha_7$ были выбраны из соображений минимизации членов седьмого порядка в разложении методической погрешности на шаге. Это приводит к тому, что на большинстве шагов главный член локальной погрешности по модулю меньше, чем у методов-конкурентов.

Алгоритмическое же преимущество функции $ode46b$ достигается тем, что на каждом шаге совершается лишь шесть вычислений правой части СОДУ, тогда как у методов-конкурентов того же порядка — семь или восемь.

На рис. 3.4 отчётливо видны «быстрые» и «медленные» области (более «медленные» участки соответствуют более «прямым» участкам траектории космического аппарата). Также видно, что метод $ode46b$ позволяет совершать более длинные шаги, ускоряя процесс интегрирования.

## Заключение

1. В работе в явном виде представлено семипараметрическое семейство шестиэтапных методов шестого порядка класса $\mathfrak{B}$.
2. При фиксированных значениях параметров представлена расчётная схема численного интегрированияа $RKB6(4)\{7F\}$. Найденная схемы экономичны в плане меньшего количества вычислений правой части СОДУ по сравнению с уже существующими методами.
3. На базе расчётной схемы создана функция, аналогичная встроенному интегратору MATLAB $ode45$ в части автоматического выбора шага. При том же количестве этапов представленная функция $ode46b$ имеет на один порядок точности больше.
4. Проведено численное исследование различных моделей механики с помощью полученных методов численного интегрирования. Продемонстрирована эффективность в сравнении с известными классическими методами, в том числе с $ode45$.

## Список литературы


1. *Runge, C.* Graphical Methods: A Course of Lectures Delivered in Columbia University, New York, October, 1909, to January, 1910 / C. Runge. — Columbia University Press, 1912. — (... Graphical Methods).
2. *Butcher, J. C.* On Runge–Kutta Processes of High Order / J. C. Butcher. — 1964. — May.
3. *Butcher, J. C.* On the Attainable Order of Runge–Kutta Methods / J. C. Butcher // Math. Comput. — 1965. — Vol. 19. — P. 408—417.
4. *Butcher, J. C.* The non-existence of ten Stage eight Order Explicit Runge–Kutta Methods / J. C. Butcher // BIT. — 1985. — Vol. 25. — P. 521—540.
5. *Fehlberg, E.* Classical fifth-, sixth-, seventh-, and eighth-order Runge–Kutta formulas with stepsize control / E. Fehlberg // NASA Technical Report 287. — 1968.
6. *Dormand, J. R.* A Family of Embedded Runge–Kutta Formulae / J. R. Dormand, P. J. Prince // Journal of Computational and Applied Mathematics. — 1980. — Mar. — Vol. 6. — P. 19—26.
7. *Олемской, И. В.* Численный метод интегрирования систем обыкновенных дифференциальных уравнений / И. В. Олемской // Математические методы анализа управляемых процессов. — 1986. — С. 157—160.
8. *Hofer, E.* A partially implicit method for large stiff systems of ODEs with only few equations introducing small time-constants / E. Hofer // SIAM J. Numer. Anal. — 1976. — Vol. 13, no. 5. — P. 645—663.
9. *Sandu, A.* Multirate generalized additive Runge-Kutta methods / A. Sandu, M. Günther // Numer. Math. — 2016. — Vol. 133, no. 3. — P. 497—524.
10. *McLachlan, R.* High order multisymplectic Runge-Kutta methods / R. McLachlan, B. Ryland, Y. Sun // SIAM J. Sci. Comput. — 2014. — Vol. 36, no. 5. — A2199—A2226.

11. *Wang, D.* Parametric symplectic partitioned Runge-Kutta methods with energy-preserving properties for Hamiltonian systems / D. Wang, A. Xiao, X. Li // Comput. Phys. Comm. — 2013. — Vol. 184, no. 2. — P. 303—310.

12. *Ketcheson, D.* Spatially partitioned embedded Runge-Kutta methods / D. Ketcheson, C. MacDonald, S. Ruuth // SIAM J. Numer. Anal. — 2013. — Vol. 51, no. 5. — P. 2887—2910.

13. *Eremin, A. S.* Continuous Extensions for Structural Runge—Kutta Methods / A. S. Eremin, N. A. Kovrizhnykh // Computational Science and Its Applications – ICCSA 2017. — Cham : Springer International Publishing, 2017. — P. 363—378. — (Lecture Notes in Computer Science ; 10405).

14. *Шиманчук, Д. В.* Построение траектории возвращения в окрестность коллинеарной точки либрации системы Солнце–Земля / Д. В. Шиманчук, А. С. Шмыров // Вестн. С.-Петерб. ун-та, Сер. 10. — 2013. — Т. 2. — С. 76—85.

15. *Clohessy, W. H.* Terminal Guidance System for Satellite Rendezvous / W. H. Clohessy, R. S. Wiltshire // J. Astronaut. Sci. — 1960. — Vol. 27, no. 9. — P. 653—678.

16. *Tschauner, J.* Rendezvous Zu Einem In Elliptischer Bahn Umlaufenden Ziel / J. Tschauner, P. Hempel // Astronautica Acta. — 1965. — Т. 11, № 2. — С. 104—109.

17. *Schweighart, S. A.* High-Fidelity Linearized J2 Model for Satellite Formation Flight / S. A. Schweighart, R. J. Sedwick // Journal of Guidance, Control, and Dynamics. — 2002. — Vol. 25, no. 6. — P. 1073—1080.

18. *Хайрер, Э.* Решение обыкновенных дифференциальных уравнений. Нежёсткие задачи / Пер. с англ. И. А. Кульчицкой, С. С. Филиппова под ред. С. С. Филиппова. / Э. Хайрер, С. Нёрсетт, Г. Ваннер. — М.: Мир, 1990. — 512 с.

19. *Олемской, И. В.* Модификация алгоритма выделения структурных особенностей / И. В. Олемской // Вестн. С.-Петерб. ун-та, Сер. 10. — 2006. — Т. 2. — С. 46—54.

20. *Олемской, И. В.* Структурный подход в задаче конструирования явных одношаговых методов / И. В. Олемской // Журнал вычислительной математики и математической физики. — 2003. — Т. 43, № 7. — С. 961—974.

21. *Еремин, А. С.* Разработка явного одношагового вложенного метода для систем структурно разделенных обыкновенных дифференциальных уравнений : дис. ... кандидата физико-математических наук: 01.01.07 / А. С. Еремин. — С.-Петерб. гос. ун-т, Санкт-Петербург, 2009. — 91 с.

22. *Kovrizhnykh, N. A.* On a Two Families of Efficient Fifth Order Schemes for Solving ODE Systems / N. A. Kovrizhnykh, A. S. Eremin // AIP Conference Proceedings. — 2018. — Vol. 1959, no. 1. — P. 030014.

23. *Олемской, И. В.* Методы интегрирования систем структурно разделенных дифференциальных уравнений / И. В. Олемской. — СПб.: Изд-во С.-Петерб. ун-та, 2009. — 180 с.

24. *Tsitouras, C.* Cheap Error Estimation for Runge–Kutta methods / C. Tsitouras, S. N. Papakostas // Siam Journal on Scientific Computing. — 1999. — Vol. 20, issue 6. — P. 2067—2088.

25. *Verner, J.* Strategies for Deriving New Explicit Runge—Kutta Pairs / J. Verner // Annals of Numerical Mathematics. — 1994. — Vol. 1. — P. 225—244.

26. *El-Mikkawy, M. E. A.* A General Four-Parameter Non-FSAL Embedded Runge–Kutta Algorithm of Orders 6 and 4 in Seven Stages / M. E. A. El-Mikkawy, M. M. M. Eisa // Applied Mathematics and Computation. — 2003. — Vol. 143, no. 2. — P. 259—267.

27. Choose an ODE Solver [Электронный ресурс]. — URL: https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html (дата обр. 19.01.2018).

28. Сравнительное исследование преимуществ структурных методов численного решения обыкновенных дифференциальных уравнений / В. Бубнов [и др.] // Труды СПИИРАН. — 2017. — Т. 53, № 4. — С. 51—72.

## Список рисунков

## Список таблиц

# Приложение A

## Семейство методов шестого порядка класса $\mathfrak{B}$

Параметры метода $c_{12} = \alpha_1$, $c_{13} = \alpha_2$, $c_{14} = \alpha_3$, $c_{23} = \alpha_4$, $c_{24} = \alpha_5$, $a_{1132} = \alpha_6$ и $a_{2243} = \alpha_7$ могут быть выбраны в качестве свободных параметров, подчиненных лишь некоторым очевидным ограничениям. В силу громоздкого вида общего решения ограничимся последовательным выражением параметров метода друг через друга.

$$c_{16} = c_{26} = 1$$

$$a_{1166} = a_{2166} = a_{2266} = 0$$

$$c_{22} = \frac{3c_{13} - 2c_{23}}{15c_{13}c_{23} - 10c_{23} + 2}$$

Введём вспомогательные переменные $\xi_\mu \equiv \sum_{\nu=\mu+1}^{6} b_{1\nu}c_{1\nu}a_{12\nu\mu}$, $\quad (\mu = 2{,}3{,}4)$.

$$\xi_2 = \sum_{\nu=3}^{6} b_{1\nu}c_{1\nu}a_{12\nu2} = \frac{5 + 15c_{24}c_{23} - 8c_{23} - 8c_{24}}{120c_{22}(c_{24} - c_{22})(c_{23} - c_{22})} +$$
$$+ \frac{(c_{25}^4 + c_{23}c_{25}^2 + c_{24}c_{25}^2 - c_{24}c_{25}^3 - c_{23}c_{24}c_{25} + c_{23}c_{24}c_{25}^2 - c_{23}c_{25}^3 - c_{25}^3)}{c_{22}(c_{24} - c_{22})(c_{23} - c_{22})} b_{25}$$

$$\xi_3 = \sum_{\nu=4}^{6} b_{1\nu}c_{1\nu}a_{12\nu3} = -\frac{5 + 15c_{24}c_{22} - 8c_{24} - 8c_{22}}{120(c_{23} - c_{22})c_{23}(c_{24} - c_{23})} -$$
$$- \frac{c_{24}c_{25}^2c_{22} + c_{24}c_{25}^2 - c_{24}c_{25}^3 + c_{25}^4 - c_{25}^3 - c_{24}c_{25}c_{22} + c_{25}^2c_{22} - c_{25}^3c_{22}}{(c_{23} - c_{22})c_{23}(c_{24} - c_{23})} b_{25} -$$

$$\xi_4 = \sum_{\nu=5}^{6} b_{1\nu}c_{1\nu}a_{12\nu4} = \frac{5 + 15c_{23}c_{22} - 8c_{23} - 8c_{22}}{120(c_{24} - c_{23})(c_{24} - c_{22})c_{24}} +$$
$$+ \frac{c_{25}^2c_{23}c_{22} + c_{23}c_{25}^2 - c_{23}c_{25}^3 + c_{25}^4 - c_{25}^3 - c_{25}c_{23}c_{22} + c_{25}^2c_{22} - c_{25}^3c_{22}}{(c_{24} - c_{23})(c_{24} - c_{22})c_{24}} b_{25}$$

Для $q = 1{,}2$:

$$c_{q5} = \frac{5c_{q4}c_{q3} - 3c_{q4} - 3c_{q3} + 2}{10c_{q4}c_{q3} - 5c_{q4} - 5c_{q3} + 3}$$

$$b_{q1} = \frac{50c_{q4}^2c_{q3}^2 - 40c_{q4}^2c_{q3} + 5c_{q4}^2 - 40c_{q4}c_{q3}^2 + 35c_{q4}c_{q3} - 5c_{q4} + 5c_{q3}^2 - 5c_{q3} + 1}{60c_{q4}c_{q3}(5c_{q4}c_{q3} - 3c_{q4} - 3c_{q3} + 2)}$$

$$b_{q3} = \frac{5c_{q4}^2 - 5c_{q4} + 1}{60c_{q3}(1 - c_{q3})(c_{q4} - c_{q3})(3c_{q4} + 6c_{q3} - 2 - 10c_{q4}c_{q3} + 10c_{q4}c_{q3}^2 - 5c_{q3}^2)}$$

$$b_{q4} = \frac{5c_{q3}^2 - 5c_{q3} + 1}{60c_{q4}(c_{q4} - 1)(c_{q4} - c_{q3})(6c_{q4} + 3c_{q3} - 2 - 10c_{q4}c_{q3} + 10c_{q4}^2c_{q3} - 5c_{q4}^2)}$$

$$b_{q5} = \frac{1}{60(5c_{q4}c_{q3} - 3c_{q4} - 3c_{q3} + 2)(3c_{q4} + 6c_{q3} - 2 - 10c_{q4}c_{q3} + 10c_{q4}c_{q3}^2 - 5c_{q3}^2)} \times$$

$$\times \frac{(10c_{q4}c_{q3} - 5c_{q4} - 5c_{q3} + 3)^5}{(5c_{q4}c_{q3} - 2c_{q4} - 2c_{q3} + 1)(6c_{q4} + 3c_{q3} - 2 - 10c_{q4}c_{q3} + 10c_{q4}^2c_{q3} - 5c_{q4}^2)}$$

$$b_{q6} = \frac{50c_{q4}^2c_{q3}^2 - 60c_{q4}^2c_{q3} + 15c_{q4}^2 + 75c_{q4}c_{q3} - 20c_{q4} - 60c_{q4}c_{q3}^2 + 15c_{q3}^2 - 20c_{q3} + 6}{60(5c_{q4}c_{q3} - 2c_{q4} - 2c_{q3} + 1)(1 - c_{q4})(1 - c_{q3})}$$

$$a_{1243} = \frac{c_{14}}{6c_{23}(5c_{13}^2 - 5c_{13} + 1)\Big\{\big(15c_{13} - 10\big)c_{23}^2 + 4c_{23} - 3c_{13}\Big\}(c_{13} - c_{14})}\Big\{\big(20c_{13} -$$

$$- 30c_{13}^2 + 150c_{14}c_{13}^2 - 100c_{14}c_{13} - 6 + 20c_{14}\big)c_{23} - 45c_{14}c_{13}^2 + 5c_{13} + 20c_{14}c_{13} - 4c_{14}\Big\}$$

$$a_{2132} = \frac{c_{23}(3c_{13} - 2c_{23})(15c_{13}c_{23} - 10c_{23} + 2)}{2c_{12}(45c_{13}^2 - 30c_{13} + 4)}$$

$$a_{2144} = \frac{c_{24}(15c_{13}c_{24}c_{23} + 2c_{23} - 10c_{24}c_{23} - 3c_{13} + 2c_{24})(c_{23} - c_{24})}{6c_{14}(c_{13} - c_{14})(5c_{23}^2 - 5c_{23} + 1)}$$

$$a_{2233} = \frac{15c_{13}^2c_{23} - 10c_{13}c_{23} + 2c_{23} - c_{13}}{45c_{13}^2 - 30c_{13} + 4}$$

$$\begin{aligned}
a_{2244} = \Big\{&\big(600c_{23}^3 - 270c_{13}^2 - 2025c_{23}c_{13}^3 - 6750c_{23}^3c_{13}^3 - 3900c_{23}^3c_{13} \\
&- 520c_{23}^2 - 9000c_{13}^2c_{23}^2 - 1200c_{13}c_{23} + 9000c_{23}^3c_{13}^2 + 6750c_{23}^2c_{13}^3 + 120c_{13} \\
&+ 160c_{23} + 3600c_{23}^2c_{13} + 3150c_{13}^2c_{23} - 16\big)c_{24}^3 + \\
&+ \big(630c_{13}^2c_{23} - 180c_{13}^2 - 120c_{13}c_{23} - 900c_{13}^2c_{23}^2 + 2025c_{23}^2c_{13}^3 - 120c_{23}^2c_{13} + \\
&+ 405c_{13}^3 - 1350c_{23}c_{13}^3 - 320c_{23}^3 - 1350c_{23}^3c_{13}^2 + 36c_{13} - \\
&- 24c_{23} + 160c_{23}^2 + 1200c_{23}^3c_{13}\big)c_{24}^2 + \\
&+ \big(90c_{13}^2c_{23} - 135c_{23}c_{13}^3 - 60c_{23}^3c_{13} + 24c_{13}c_{23} + 180c_{13}^2c_{23}^2 - 120c_{23}^2c_{13} + 40c_{23}^3 \\
&- 18c_{13}^2 - 8c_{23}^2\big)c_{24} + \big(5400c_{23}^3c_{13} - 720c_{23}^3 + 96c_{23}^2 + 4050c_{23}^2c_{13}^3 - 1620c_{13}^2c_{23}^2 \\
&+ 1680c_{23}^4 - 27000c_{23}^5c_{13}^2 - 1200c_{23}^5 - 360c_{23}^2c_{13} - 72c_{13}c_{23} - 810c_{23}c_{13}^3 \\
&+ 540c_{13}^2c_{23} - 20250c_{23}^4c_{13}^3 + 20250c_{23}^5c_{13}^3 - 8100c_{23}^3c_{13}^2 - 14400c_{23}^4c_{13} \\
&+ 32400c_{23}^4c_{13}^2 + 10800c_{23}^5c_{13}\big)a_{2243}\Big\} \times
\end{aligned}$$

$$\times \frac{-1}{6c_{24}((15c_{13}c_{23} + 2 - 10c_{23})c_{24} + 2c_{23} - 3c_{13})(5c_{23}^2 + 5c_{23} + 1)(45c_{13}^2 - 30c_{13} + 4)}$$

$$a_{1121} = a_{1122} = \frac{1}{2}c_{12}$$

$$a_{1155} = \frac{1}{2}(1 - c_{15})$$
$$a_{1165} = \frac{1}{2}\frac{b_{15}}{b_{16}}(1 - c_{15})$$
$$a_{1131} = \frac{1}{2}c_{13} - a_{1132}\frac{c_{13} - c_{12}}{c_{13}}$$
$$a_{1133} = \frac{1}{2}c_{13} - a_{1132}\frac{c_{12}}{c_{13}}$$
$$a_{1142} = \frac{b_{13}}{b_{14}} \cdot \frac{(c_{15} - c_{13})(1 - c_{13})}{(c_{15} - c_{14})(c_{14} - 1)}a_{1132}$$
$$a_{1152} = -\frac{b_{13}}{b_{15}} \cdot \frac{(c_{14} - c_{13})(1 - c_{13})}{(c_{15} - c_{14})(c_{15} - 1)}a_{1132}$$
$$a_{1162} = \frac{b_{13}}{b_{16}} \cdot \frac{(c_{14} - c_{13})(c_{13} - c_{15})}{(1 - c_{14})(1 - c_{15})}a_{1132}$$
$$a_{1141} = \frac{30c_{13}c_{14}^2 + 30c_{14}^3c_{12} - 30c_{14}^3c_{13} - 30c_{14}^2c_{12} - 6c_{14}c_{13} + 6c_{12}c_{14}}{6(5c_{13}^2 - 5c_{13} + 1)c_{13}^2}a_{1132} +$$
$$+ \frac{c_{13}^3 - c_{13}c_{14}^2 - 20c_{13}^3c_{14} + 5c_{14}^2c_{13}^2 + 15c_{13}^4c_{14} + 3c_{14}c_{13}^2}{6(5c_{13}^2 - 5c_{13} + 1)c_{13}^2}$$
$$a_{1144} = \frac{2c_{14} - 10c_{14}c_{13} - c_{13} + 15c_{13}^2c_{14}}{6(5c_{13}^2 - 5c_{13} + 1)}$$
$$a_{1154} = \frac{b_{14}(c_{14} - 1)(2a_{1144} + c_{14} - 1)}{2b_{15}(1 - c_{15})}$$
$$a_{1164} = \frac{b_{14}\big(2a_{1144}(c_{14} - c_{15}) + (2c_{15} - c_{14} - 1)(1 - c_{14})\big)}{2b_{16}(c_{15} - 1)}$$
$$a_{1153} = \frac{c_{15}^2}{2c_{13}} - \frac{(a_{1152}c_{12} + a_{1154}c_{14} + a_{1155}c_{15})}{c_{13}}$$
$$a_{1163} = \frac{1}{2c_{13}} - \frac{a_{1162}c_{12} + a_{1164}c_{14} + a_{1165}c_{15}}{c_{13}}$$
$$a_{11w1} = c_{1w} - \sum_{\nu=2}^{w} a_{11w\nu}, \quad w = 5{,}6$$
$$a_{1265} = \frac{b_{25}(1 - c_{25})}{b_{16}}$$
$$a_{1231} = \frac{c_{13}(2c_{22} - c_{13})}{2c_{22}}$$
$$a_{1232} = \frac{c_{13}^2}{2c_{22}}$$
$$a_{1254} = \frac{\xi_4 - b_{24}(1 - c_{24})}{b_{15}(c_{15} - 1)}$$

$$a_{1264} = \frac{b_{24}c_{15}(1-c_{24}) - \xi_4}{b_{15}(c_{15}-1)}$$

$$a_{1253} = \frac{(1-c_{14})a_{1243}b_{14} + \xi_3 + (c_{23}-1)b_{23}}{b_{15}(c_{15}-1)}$$

$$a_{1263} = \frac{(c_{15}-c_{14})a_{1243}b_{14} + c_{15}(c_{23}-1)b_{23} + \xi_3}{b_{16}(1-c_{15})}$$

$$a_{1252} = \frac{(c_{14}-1)b_{14}c_{23}a_{1243}}{b_{15}c_{22}(c_{15}-1)} + \frac{\xi_2}{b_{15}(c_{15}-1)} + \frac{(1-c_{13})c_{13}^2b_{13} + (1-c_{14})c_{14}^2b_{14}}{2b_{15}c_{22}(c_{15}-1)}$$

$$a_{1262} = \frac{(c_{15}-c_{14})b_{14}c_{23}a_{1243}}{b_{16}c_{22}(c_{15}-1)} - \frac{\xi_2}{b_{16}(c_{15}-1)} - \frac{(c_{15}-c_{13})c_{13}^2b_{13} + (c_{15}-c_{14})c_{14}^2b_{14}}{2b_{16}c_{22}(c_{15}-1)}$$

$$a_{1241} = \frac{(c_{23}-c_{22})a_{1243}}{c_{22}} - \frac{c_{14}^2}{2c_{22}} + c_{14}$$

$$a_{1242} = -\frac{c_{23}a_{1243}}{c_{22}} + \frac{c_{14}^2}{2c_{22}}$$

$$a_{12w1} = c_{1w} - \sum_{\nu=2}^{w-1} a_{12w\nu}, \quad w = 5{,}6$$

$$a_{2121} = \frac{c_{22}(2c_{12}-c_{22})}{c_{12}}$$

$$a_{2122} = \frac{c_{22}^2}{c_{12}}$$

$$a_{2131} = \frac{(c_{12}-c_{13})}{c_{13}}a_{2132} + \frac{c_{23}(2c_{13}-c_{23})}{2c_{13}}$$

$$a_{2133} = \frac{c_{23}^2}{2c_{13}} - \frac{c_{12}}{c_{13}}a_{2132}$$

$$a_{2155} = \frac{b_{15}(1-c_{15})^2}{2b_{25}(1-c_{25})}$$

$$a_{2165} = \frac{b_{15}(1-c_{15})(1+c_{15}-2c_{25})}{2b_{26}(1-c_{25})}$$

$$a_{2142} = -\frac{b_{23}a_{2132}(c_{23}-1)(c_{25}-c_{23})}{b_{24}(c_{24}-1)(c_{25}-c_{24})}$$

$$a_{2152} = \frac{b_{23}a_{2132}(c_{23}-1)(c_{24}-c_{23})}{b_{25}(c_{25}-1)(c_{25}-c_{24})}$$

$$a_{2162} = -\frac{b_{23}a_{2132}(c_{25}-c_{23})(c_{24}-c_{23})}{b_{26}(1-c_{25})(1-c_{24})}$$

$$a_{2154} = \frac{1}{b_{25}(c_{25}-1)}\left\{(1-c_{24})a_{2144}b_{24} - \frac{(1-c_{14})^2}{2}b_{14}\right\}$$

$$a_{2164} = \frac{1}{b_{26}(1-c_{25})}\left\{(c_{25}-c_{24})a_{2144}b_{24} + \frac{(1-c_{14})(1+c_{14}-2c_{25})}{2}b_{14}\right\}$$

$$a_{2143} = \frac{c_{24}^2}{2c_{13}} - \frac{c_{12}}{c_{13}} a_{2142} - \frac{c_{14}}{c_{13}} a_{2144}$$

$$a_{21w3} = \frac{1}{c_{13}} \{ \frac{c_{2w}^2}{2} - \sum_{\nu=2 \nu \neq 3}^{5} a_{21w\nu} c_{1\nu} \}, \quad w = 5{,}6$$

$$a_{21w1} = c_{2w} - \sum_{\nu=2}^{w} a_{21w\nu}, \quad w = 4{,}5{,}6$$

$$a_{2221} = a_{2222} = \frac{c_{22}}{2}$$

$$a_{2253} = \frac{b_{23}(c_{23}-1)(2a_{2233}+c_{23}-1)}{2b_{25}(1-c_{25})} + \frac{b_{24}(c_{24}-1)a_{2243}}{b_{25}(1-c_{25})}$$

$$a_{2263} = \frac{b_{24}(c_{25}-c_{24})}{b_{26}(1-c_{25})} a_{2243} + \frac{b_{23}(c_{25}-c_{23})}{b_{26}(1-c_{25})} a_{2233} + \frac{b_{23}(1-c_{23})(1+c_{23}-2c_{25})}{2b_{26}(1-c_{25})}$$

$$a_{2254} = \frac{b_{24}(c_{24}-1)}{2b_{25}(1-c_{25})} \Big\{ 2a_{2244} + c_{24} - 1 \Big\}$$

$$a_{2264} = \frac{b_{24}}{2b_{25}(1-c_{25})} \Big\{ 2(c_{25}-c_{24})a_{2244} + (1-c_{24})(1+c_{24}-2c_{25}) \Big\}$$

$$a_{2255} = \frac{1}{2}(1-c_{25})$$

$$a_{2265} = \frac{b_{25}}{2b_{26}}(1-c_{25})$$

$$a_{22\mu 1} = c_{2\mu} - \frac{1}{c_{22}} \{ \frac{c_{2\mu}^2}{2} + \sum_{\nu=3}^{\mu} (c_{22} - c_{2\nu}) a_{22\mu\nu} \}, \quad \mu = 3{,}4{,}5{,}6$$

$$a_{22\mu 2} = \frac{1}{c_{22}} \{ \frac{c_{2\mu}^2}{2} - \sum_{\nu=3}^{\mu} c_{2\nu} a_{22\mu\nu} \}, \quad \mu = 3{,}4{,}5{,}6$$

# Приложение Б

## Текст программы ode46b

Листинг Б.1 Функция численного интегрирования систем класса $\mathfrak{B}$

```
function varargout = ode46b(ode,tspan,y0,options,varargin)

solver_name = 'ode45';

% Stats
nsteps  = 0;
nfailed = 0;
nfevals = 0;

% Output
FcnHandlesUsed  = isa(ode,'function_handle');
output_sol = (FcnHandlesUsed && (nargout==1));

sol = []; f3d = [];
if output_sol
    sol.solver = solver_name;
    sol.extdata.odefun = ode;
    sol.extdata.options = options;
    sol.extdata.varargin = varargin;
end

% Handle solver arguments
for i=1:4
    [neq, tspan, ntspan, next, t0, tfinal, tdir, y0, f0, odeArgs
       , odeFcn, ...
        options, threshold, rtol, normcontrol, normy, hmax, htry
           , htspan, dataType] = ...
        odearguments(FcnHandlesUsed, solver_name, ode, tspan, y0
           , options, varargin);
end
nfevals = nfevals + 1;

% Handle the output
if nargout > 0
    outputFcn = odeget(options,'OutputFcn',[],'fast');
else
```

```
    outputFcn = odeget(options,'OutputFcn',@odeplot,'fast');
end
outputArgs = {};
if isempty(outputFcn)
    haveOutputFcn = false;
else
    haveOutputFcn = true;
    outputs = odeget(options1,'OutputSel',1:neq,'fast');
    if isa(outputFcn,'function_handle')
        % With MATLAB 6 syntax pass additional input arguments
           to outputFcn.
        outputArgs = varargin;
    end
end
refine = max(1,odeget(options,'Refine',4,'fast'));

% Handle the event function
[haveEventFcn,eventFcn,eventArgs,valt,teout,yeout,ieout] = ...
    odeevents(FcnHandlesUsed,odeFcn,t0,y0,options,varargin);

t = t0;
y = y0;

% Allocate memory if we're generating output.
nout = 0;
tout = []; yout = [];
if nargout > 0
    chunk = min(max(100,50*refine), refine+floor((2^11)/neq));
    tout = zeros(1,chunk,dataType);
    yout = zeros(neq,chunk,dataType);
    f3d = zeros(neq,7,chunk,dataType);

    nout = 1;
    tout(nout) = t;
    yout(:,nout) = y;
end

% Initialize method parameters.
pow = 1/5;
[B11 B12 B21 B22 A1 A2 E1 E2] = coeff;

f(neq,7) = zeros(1,1,dataType);

```

```
hmin = 16*eps(t);

% Compute an initial step size h using y'(t).
absh = min(hmax, htspan);

rh = norm(f0 ./ max(abs(y),threshold),inf) / (0.8 * rtol^pow);
if absh * rh > 1
    absh = 1 / rh;
end
absh = max(absh, hmin);

f(:,1) = f0;

structure = feval(odeFcn, 's');
ny1 = structure('1');
ny2 = structure('2')+ny1;

% Initialize the output function.
if haveOutputFcn
    feval(outputFcn,[t tfinal],y(outputs),'init',outputArgs{:});
end

% THE MAIN LOOP

done = false;
while ~done

    % By default, hmin is a small number such that t+hmin is only slightly
    % different than t.  It might be 0 if t is 0.
    hmin = 16*eps(t);
    absh = min(hmax, max(hmin, absh));    % couldn't limit absh until new hmin
    h = tdir * absh;

    % Stretch the step if within 10% of tfinal-t.
    if 1.1*absh >= abs(tfinal - t)
        h = tfinal - t;
        absh = abs(h);
        done = true;
    end

    % LOOP FOR ADVANCING ONE STEP.
```

```
    nofailed = true;                           % no failed attempts
    while true
        hA1 = h * A1;
        hA2 = h * A2;
        hB11 = h * B11;
        hB12 = h * B12;
        hB21 = h * B21;
        hB22 = h * B22;
        for i=2:6
            for j=1:ny1
                yt = [
                    y(     1:ny1)+f(     1:ny1,:)*hB11(:,i-1)
                    y(ny1+1:ny2)+f(ny1+1:ny2,:)*hB12(:,i-1)
                    ];
                f(j,i) = feval(odeFcn, t+hA1(i-1), yt, j,
                   odeArgs{:});
            end

            for j=ny1+1:ny2
                yt = [
                    y(     1:ny1)+f(     1:ny1,:)*hB21(:,i-1)
                    y(ny1+1:ny2)+f(ny1+1:ny2,:)*hB22(:,i-1)
                    ];
                f(j,i) = feval(odeFcn, t+hA2(i-1), yt, j,
                   odeArgs{:});
            end
        end

        tnew = t + hA1(6);
        if done
            tnew = tfinal;    % Hit end point exactly.
        end

        ynew = y + [
            f(     1:ny1,:)*hB11(:,6)
            f(ny1+1:ny2,:)*hB22(:,6)
            ];
        f(:,7) = feval(odeFcn,tnew,ynew,odeArgs{:});

        nfevals = nfevals + 6;

        % Estimate the error.
        NNrejectStep = false;
```

```
        err = absh * norm(([f(1:ny1,:)*E1; f(ny1+1:ny2,:)*E2])
           ...
            ./ max(max(abs(y),abs(ynew)),threshold),inf);

        % Accept the solution only if the weighted error is no
           more than the
        % tolerance rtol.  Estimate an h that will yield an
           error of rtol on
        % the next step or the next try at taking this step, as
           the case may be,
        % and use 0.8 of this value to avoid failures.
        if err > rtol                       % Failed step
            nfailed = nfailed + 1;
            if absh <= hmin
                warning(message('MATLAB:ode45:
                   IntegrationTolNotMet', sprintf( '%e', t ),
                   sprintf( '%e', hmin )));
                solver_output = odefinalize(solver_name, sol,...
                    outputFcn, outputArgs,...
                    0, [nsteps, nfailed, nfevals],...
                    nout, tout, yout,...
                    haveEventFcn, teout, yeout, ieout,...
                    {f3,idxNonNegative});
                if nargout > 0
                    varargout = solver_output;
                end
                return;
            end

            if nofailed
                nofailed = false;
                if NNrejectStep
                    absh = max(hmin, 0.5*absh);
                else
                    absh = max(hmin, absh * max(0.1, 0.8*(rtol/
                       err)^pow));
                end
            else
                absh = max(hmin, 0.5 * absh);
            end
            h = tdir * absh;
```

```
            done = false;

        else                                % Successful step

            NNreset_f7 = false;

            break;

        end
    end
    nsteps = nsteps + 1;

    if output_sol
        nout = nout + 1;
        if nout > length(tout)
            tout = [tout, zeros(1,chunk,dataType)];  % requires
               chunk >= refine
            yout = [yout, zeros(neq,chunk,dataType)];
            f3d  = cat(3,f3d,zeros(neq,7,chunk,dataType));
        end
        tout(nout) = tnew;
        yout(:,nout) = ynew;
        f3d(:,:,nout) = f;
    end

    if done
        break
    end

    % If there were no failures compute a new h.
    if nofailed
        % Note that absh may shrink by 0.8, and that err may be
          0.
        temp = 1.25*(err/rtol)^pow;
        if temp > 0.2
            absh = absh / temp;
        else
            absh = 5.0*absh;
        end
    end

    % Advance the integration one step.
    t = tnew;
```

```
    y = ynew;

    if NNreset_f7
        % Used f7 for unperturbed solution to interpolate.
        % Now reset f7 to move along constraint.
        f(:,7) = feval(odeFcn,tnew,ynew,odeArgs{:});
        nfevals = nfevals + 1;
    end
    f(:,1) = f(:,7);  % Already have f(tnew,ynew)

end

solver_output = odefinalize(solver_name, sol,...
    outputFcn, outputArgs,...
    0, [nsteps, nfailed, nfevals],...
    nout, tout, yout,...
    haveEventFcn, teout, yeout, ieout,...
    {f3d,false});


if nargout > 0
    varargout = solver_output;
end

end

function [B11, B12, B21, B22, A1, A2, E1, E2] = coeff
B11 = [
    [   1/9, 1/9,      0,     0,      0,     0, 0]
    [  1/12,   0,   1/12,     0,      0,     0, 0]
    [ -1/44,   0,   9/22,  5/44,      0,     0, 0]
    [  7/36,   0,      0,   5/9,   1/12,     0, 0]
    [  -3/7,   0,    9/8, -5/28,  27/56,     0, 0]
    [ 7/150,   0, 27/100, 11/30, 27/100, 7/150, 0]
    ]';
B12 = [
    [      2/9,        0,        0,     0,      0,     0, 0]
    [     5/48,     1/16,        0,     0,      0,     0, 0]
    [   37/176,  243/176,   -12/11,     0,      0,     0, 0]
    [ -635/432,  -167/16,    100/9, 44/27,      0,     0, 0]
    [     29/4,  1377/28, -1425/28, -11/2,  27/28,     0, 0]
    [    7/150,        0,   27/100, 11/30, 27/100, 7/150, 0]
    ]';
```

```
A1 = sum(B11);

B21 = [
    [     1/9,      1/9,       0,     0,       0,     0, 0]
    [    7/48,     3/16,    -1/6,     0,       0,     0, 0]
    [ -31/176,  -81/176,   45/44,  5/44,       0,     0, 0]
    [  73/144,    15/16,    -5/4,   5/9,    1/12,     0, 0]
    [  -39/28,   -81/28,  279/56, -5/28,   27/56,     0, 0]
    [   7/150,        0, 27/100, 11/30, 27/100, 7/150, 0]
    ]';
B22 = [
    [       1/9,       1/9,        0,         0,      0,     0, 0]
    [      7/48,      3/16,     -1/6,         0,      0,     0, 0]
    [ -185/1584,  -123/880,      2/3,   89/990,      0,     0, 0]
    [ 1031/3888,   -53/144,   65/324,  317/486,   1/12,     0, 0]
    [    -29/63,      15/7, -103/168, -139/252,  27/56,     0, 0]
    [     7/150,         0,  27/100,     11/30, 27/100, 7/150, 0]
    ]';
A2 = sum(B22);

E1 = [11/25, 0, -99/100, 11/10, -99/100, -14/25, 1]'/24;
E2 = E1;
end
```