Оглавление

Глава 1.

Введение

1.1. Уравнения Бутчера

1.1.1.Метод Эйлера. Простейший метод решения начальной задачи

$$y' = f(x,y), \quad y(x_0) = y_0$$

был описан Эйлером (1768) в его "Интегральном исчислении" и является, фактически, методом Рунге-Кутта порядка 1. Глобальная погрешность метода имеет вид $c \cdot h$, где с – постоянная, зависящая от задачи, и h – длина шага. Если желательно, скажем, получить 6 точных десятичных знаков, то требуется, следовательно, порядка миллиона шагов.

Если в исходном дифференциальном уравнении функция $f(x,y)$ не зависит от y, то решение дифференциального уравнения сводится к нахождению определенного интеграла и метод Эйлера переходит в простейший метод нахождения определенных интегралов – метод прямоугольников. Для нахождения определенных интегралов уже давно были известны гораздо более точные методы. Естественно, возникало желание найти аналогичные методы и для решения дифференциальных уравнений.

1.1.2.Такую попытку произвел Рунге (1895). Для перехода от метода прямоугольников к первой квадратурной формуле Гаусса он рассуждал следующим образом. Первый шаг длины h должен иметь вид

$$y(x_0 + h) \approx y_0 + hf(x_0 + h/2, y(x_0 + h/2)).$$

Но какое значение взять для $y(x_0 + h/2)$? За неимением лучшего естественно использовать один малый шаг метод Эйлера длины $h/2$. Расчетные формулы выглядят так:

$$\begin{array}{rcl} k_1 & = & f(x_0, y_0), \\ k_2 & = & f(x_0 + h/2, y_0 + h/2k_1), \ . \\ y_1 & = & y_0 + hk_2 \end{array} \qquad (*)$$

Может показаться странным, что для вычисления k_2 мы предлагаем сделать шаг методам Эйлера, о неэффективности которого говорилось выше.

Однако, в результате умножения k_2 в третьем выражении на h, влияние погрешности становится менее существенным. Точнее говоря, вычислим для y_1 в разложение Тейлора по степеням h:

$$\begin{aligned} y_1 &= y_0 + hf(x_0 + h/2, y_0 + h/2 f_0) = \\ &= y_0 + hf(x_0, y_0) + h^2/2(f_x + f_y f)(x_0, y_0) + \\ &+ h^3/8(f_{xx} + 2f_{xy}f + f_{yy}f^2)(x_0, y_0) + \ldots . \end{aligned}$$

Его можно сравнить с рядом Тейлора для точного решения:

$$\begin{aligned} y(x_0 + h) &= y_0 + hf(x_0, y_0) + h^2/2(f_x + f_y f)(x_0, y_0) + \\ &+ h^3/6(f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y f_x + {f^2}_y f)(x_0, y_0) + \ldots . \end{aligned}$$

Вычитая из последнего равенства предыдущее, получим для погрешности первого шага выражение

$$y(x_0 + h) - y_1 = h^3/24(f_{xx} + 2f_{xy}f + f_{yy}f^2 + 4(f_y f_x + f_y^2 f))(x_0, y_0) + \ldots$$

Таким образом, если все частные производные f второго порядка ограничены, то $\|y(x_0 + h) - y_1\| \leq ch^3$.

Чтобы получить приближенное значение решения задачи Коши в конечной точке X, будем применять формулы (*) последовательно к интервалам $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, X)$, подобно тому как применялся метод Эйлера. Погрешность численного решения ограничена величиной вида ch^2 (h – максимальная длина шага). Таким образом, (*) является усовершенствованием метода Эйлера. Рунге показал, что для вычислений с высокой точностью можно найти еще лучшие методы.

1.2. Общая формулировка методов Рунге-Кутта

Рунге и Хойн построили новые методы, включив в формулы расчета один или два добавочных шага. Но именно Кутта сформулировал общую схему того, что теперь называется методами Рунге-Кутта.

1.2.1. Определение. Пусть n – целое положительное число (количество "стадий" или "этапов") и

$$\begin{aligned} &c_2, a_{21}, \\ &c_3, a_{31}, a_{32} \\ &\ldots, \\ &c_n, a_{n1}, a_{n2}, \ldots a_{n,n-1}, \\ &b_1, \ldots, b_n, \end{aligned}$$

— вещественные коэффициенты. Тогда метод

$$\begin{aligned} k_1 &= f(x_0, y_0), \\ k_2 &= f(x_0 + c_2 h, y_0 + h a_{21} k_1), \\ k_3 &= f(x_0 + c_3 h, y_0 + h(a_{31}k_1 + a_{32}k_2)), \\ &\ldots \\ k_n &= f(x_0 + c_n h, y_0 + h(a_{n1}k_1 + \ldots + a_{n,n-1}k_{n-1})), \\ y_1 &= y_0 + h(b_1 k_1 + \ldots + b_n k_n) \end{aligned}$$

называется n-стадийным (n-шаговым) явным (explicit) методом Рунге-Кутта для задачи Коши (1.1).

Обычно коэффициенты c_i удовлетворяют условиям

$$\begin{aligned} &c_2 = a_{21}, \\ &c_3 = a_{31} + a_{32}, \\ &\ldots, \\ &c_n = a_{n1} + \ldots + a_{n,n-1} \end{aligned}$$

Эти условия были приняты Куттой без каких-либо комментариев. Смысл их в том, что все точки, в которых вычисляется f, являются приближениями первого порядка к решению. Эти условия сильно упрощают вывод условий, определяющих порядок аппроксимации для методов высокого порядка. Однако для методов низких порядков эти предположения не являются необходимыми.

1.2.2. Определение. Метод Рунге-Кутта имеет порядок p, если для достаточно гладких задач вида (1.1)

$$\|y(x_0 + h) - y_1\| \le kh^{p+1},$$

т.е. если ряды Тейлора для точного решения $y(x_0 + h)$ и для y_1 совпадают до члена h^p включительно.

1.2.3.Уравнения для методов порядка 4. Классическими принято называть 4-х стадийные методы 4 порядка. Именно они и были найдены в оригинальных работах Хойна, Рунге и Кутта. Для того чтобы зафиксировать обозначения и показать в простейших случаях примеры уравнений, которые будут рассматриваться в дальнейшем, рассмотрим их более подробно.

Для вывода уравнений на коэффициенты матрицы требуется просто вычислить производные порядков 1, 2, 3 и 4 от $y_1 = y_1(h)$ при $h = 0$ и сравнить их с производными точного решения. Получаются следующие условия (уравнения Рунге-Кутта):

$$\begin{array}{lll}
\sum_i b_i & = b_1 + b_2 + b_3 + b_4 & = 1, \\
\sum_i b_i c_i & = b_2 c_2 + b_3 c_3 + b_4 c_4 & = 1/2, \\
\sum_i b_i c_i^2 & = b_2 {c_2}^2 + b_3 {c_3}^2 + b_4 {c_4}^2 & = 1/3, \\
\sum_{i,j} b_i a_{ij} c_j & = b_3 a_{32} c_2 + b_4 (a_{42} c_2 + a_{43} c_3) & = 1/6, \\
\sum_i b_i c_i^3 & = b_2 c_2^3 + b_3 c_3^3 + b_4 c_4^3 & = 1/4, \\
\sum_{i,j} b_i c_i a_{ij} c_j & = b_3 c_3 a_{32} c_2 + b_4 c_4 (a_{42} c_2 + a_{43} c_3) & = 1/8, \\
\sum_{i,j} b_i a_{i,j} c_j^2 & = b_3 a_{32} c_2^2 + b_4 (a_{42} c_2^2 + a_{43} c_3^2) & = 1/12, \\
\sum_{i,j,k} b_i a_{ij} a_{jk} c_k & = b_4 a_{43} a_{32} c_2 & = 1/24.
\end{array}$$

Громоздкость нахождения этих уравнений очень быстро растет с ростом порядка метода.

1.2.4. Как уже было сказано ранее, методы Рунге-Кутта используемые для численного решения обыкновенных дифференциальных уравнение, характеризуются двумя основными параметрами (p, n) - порядком метода и количеством шагов соответственно. Естественно, что при фиксированном порядке (точности) метода, количество шагов хотелось бы минимизировать. Классические методы имеют параметры (4,4). К настоящему времени([22, 23, 24, 8]) известны методы с параметрами (5,6), (6,7), (7,9) (8,13) и некоторые другие.

Каждый n-шаговый метод РК порядка p задается нижнетреугольной $n \times n$-матрицей $\tilde{A}$ вида:

$$\tilde{A} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ & & \dots & \\ a_{n1} & a_{n2} & \dots & 0 \end{pmatrix},$$

и вектором $b = (b_1, \dots b_n)$ длины n. Их можно объединить в одну расширенную матрицу A размера $(n+1) \times (n+1)$:

$$A = \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ & & \dots & \\ b_1 & b_2 & \dots & 0 \end{pmatrix}.$$

1.2.5. Примеры методов (указываем расширенную матрицу).

Метод Эйлера (1 порядка):

$$A_0 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Методы 3 порядка:

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 \\ 1/6 & 4/6 & 1/6 & 0 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 \\ 1/4 & 0 & 3/4 & 0 \end{pmatrix}.$$

Методы 4 порядка:

$$A_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1/6 & 2/6 & 2/6 & 1/6 & 0 \end{pmatrix}, \qquad A_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 \\ -1/3 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 & 0 \end{pmatrix}.$$

1.2.6. Для нахождения n-шагового метода Рунге-Кутта требуется найти $\mathbf{R}$-матрицу A, коэффициенты которой удовлетворят некоторой системе нелинейных (полиномиальных) уравнений. В практически важных случаях ($p = 7..10$) мы получаем систему от нескольких десятков (или даже сотен) переменных и еще большего количества уравнений. В настоящее время не существует общих способов ее решения. Нахождение каждого отдельного решения является большой проблемой.

1.3. Обозначения

1.3.1. Для того, чтобы дальнейшие рассуждения были более компактны и эффективны, нам будут удобны следующие обозначения.

Матрицу $\tilde{A}$ будем отождествлять с соответствующими линейными операторами в пространствах $\mathbf{R}^n$. Вектора обычно будем представлять в виде матрицы из одного столбца.

Для двух векторов

$$x = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} \quad \text{и} \quad y = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix}$$

через $x * y$ обозначим вектор

$$x = \begin{pmatrix} x_1 \cdot y_1 \\ \dots \\ x_n \cdot y_n \end{pmatrix},$$

скалярное произведение векторов-столбцов обозначим, как обычно, $(x, y) = x_1 \cdot y_1 + \ldots + x_y \cdot y_n$.

Вектор, состоящий из одних единиц обозначим символом e, вектор, составленный из b_i обозначим B:

$$e = \begin{pmatrix} 1 \\ \dots \\ 1 \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ \dots \\ b_n \end{pmatrix},$$

1.3.2. Деревья. Как уже говорилось ранее, система уравнений, которой должны удовлетворять коэффициенты метода РК весьма громоздка. Будем рассматривать общую структуру условий, определяющих порядок метода, или условий порядка, как их называют для краткости. В первональных работах условия порядка находились, в основном, путем разложения функции $f(x, y)$ в степенной ряд и последовательного нахождения приближенных значений величин k_i. Со временем способ вывода прошел большую эволюцию. Он совершенствовался главным образом под влиянием работ Дж.Бутчера. Исследованию этих уравнений с разных точек зрения и их решению посвящена настоящая работа.

1.3.3. Определение. а) Деревом будем называть граф без циклов с отмеченной вершиной (корнем). Будем считать, что корень дерева расположен внизу.

б) Пусть a - некоторая вершина в дереве t, через t_a обозначим дерево, состоящее из всех вершин дерева t, лежащих над а.

в) Через t_0 обозначим дерево, состоящее из одного корня.

г) Для двух деревьев t_1 и t_2 через $t_1 t_2$ обозначим дерево, получающееся объединением корней деревьев t_1 и t_2. Относительно этой операции множество всех деревьев является свободной полугруппой бесконечного ранга, нейтральный элемент - t_0, образующими являются все одноногие деревья (у которых из корня выходит лишь одно ребро).

д) Для любого дерева t через αt обозначим дерево, полученное из t добавлением еще одной вершины "под" корнем. Дерево αt - всегда одноногое. Любое дерево может быть получено из t_0 комбинацией операций α и умножения.

е) Для любого дерева t через $w(t)$ обозначим количество вершин дерева минус 1 (то есть количество ребер). Число $w(t)$ будем называть "весом" дерева.

ж) Для любого дерева t через $\delta(t)$ обозначим произведение по всем вершинам a, кроме корня $(w(t_a)+1)$; через $\gamma(t)$ - произведение $\delta(t)(w(t)+1)$.

1.3.4. Предложение. а) $w(t_1t_2) = w(t_1) + w(t_2)$,

б) $w(\alpha t) = w(t) + 1$,

в) $\delta(t_1t_2) = \delta(t_1)\delta(t_2)$,

г) $\delta(\alpha t) = \delta(t)(w(t) + 1)$.

1.3.5. Пусть r_n, s_n - количество различных деревьев веса n ($\leq n$ соответственно). Тогда (см.[1]):

n	0	1	2	3	4	5	6	7	8	9
r_n	1	1	2	4	9	20	48	115	286	719
s_n	1	2	4	8	17	37	85	200	486	1205

1.3.6. Определение. Для любого дерева t и квадратной матрицы A определим вектор $\Phi_t(A)$ следующим образом. Для дерева t_0 положим $\Phi_{t_0}(A) = e$, где $e = (1, \ldots, 1)$. Для дерева αt положим $\Phi_{\alpha t}(A) = A \cdot \Phi_t(A)$, где $A \cdot v$ - произведение матрицы на вектор. Для двух деревьев t_1 и t_2 положим

$$\Phi_{t_1t_2}(A) = \Phi_{t_1}(A) * \Phi_{t_2}(A).$$

Так как с помощью операции α и умножения деревьев из дерева t_0 можно получить все деревья, то этих правил достаточно для полного задания оператора Φ.

N	w	дерево	$\Phi_t(A)$	$\delta(t)$	$\gamma(t)$
1	0	t_0	e	1	1
2	1	$t_1 = \alpha t_0$	Ae	1	2
3	2	$\alpha^2 t_0$	$A^2 e$	2	6
4	2	$t_1 \cdot t_1$	$Ae * Ae$	1	3
5	3	$\alpha^3 t_0$	$A^3 e$	6	24
6	3	$t_1 \cdot t_2$	$Ae * A^2 e$	2	8
7	3	$t_1 \cdot t_1 \cdot t_1$	$Ae * Ae * Ae$	1	4
8	3	$\alpha(t_1 \cdot t_1)$	$A(Ae * Ae)$	2	8

1.3.7. Теорема Бутчера. Чтобы метод Рунге-Кутта имел порядок p, необходимо и достаточно выполнения равенств

$$(B, \Phi_t(\tilde{A})) = 1/\gamma(t)$$

для всех деревьев t порядка, меньшего или равного p. Эта система уравнений и называется *системой Бутчера*.

Эта теорема позволяет легко сосчитать число условий порядка для любого p. Результаты подсчета, приведены в следующей таблице.

Порядок p	1	2	3	4	5	6	7	8	9	10
Число условий	1	2	4	8	17	37	85	200	486	1205

Для случая $p = 4$ все решения системы уравнений Бутчера (8 уравнений от 10 переменных), найдены полностью еще в начале XX века: система имеет 2-параметрическое семейство решений.

Для случая $p = 5$ в работах Бутчера [24] найдены все решения соответствующей системы 17 уравнений от 21 переменной, получено 5-параметрическое семейство решений.

Для случая $p = 6$ столь же полного решения получить уже не удается - система оказывается через чур сложна. В работе Бутчера [25] найдены решения соответствующей системы уравнений при условии выполнения упрощающих предположений. Характерным свойством таких решений является условие $b_2 = 0$.

Помимо этого имеются и другие семейство решений, например найденное Г.М.Хаммудом [19].

1.3.8. Поскольку аналитическое решение находится с трудом и не в полном объеме, Хашиным С.И. была написана программа [18], позваляющая решать уравнения Бутчера численно, а так же находить локальную размерность пространства решений. С ее помощью можно сравнительно быстро найти различные 7-стадийные методы Рунге-Кутта порядка 6. Вот одно из найденных с ее помощью решений.

$$
\begin{array}{lll@{\qquad}lll}
a_{21} & = & 0.2860825812544133 20, & a_{73} & = & 2.931386025829144800, \\
a_{31} & = & -0.006036490739423531, & a_{74} & = & 0.855972483839928045, \\
a_{32} & = & 0.292119071993836860, & a_{75} & = & -0.571264301554931399, \\
a_{41} & = & -0.283144705065800811, & a_{76} & = & -0.048728113298360533, \\
a_{42} & = & -0.733966353930938461, & b_1 & = & 0.086057647702322081, \\
a_{43} & = & 1.797004109444218350, & b_2 & = & 0.196591037531591140, \\
a_{51} & = & -0.702054950734009209, & b_3 & = & 0.243180094157103719, \\
a_{52} & = & -2.089412504022115860, & b_4 & = & 0.778821981768729596, \\
a_{53} & = & 3.811076756075609560, & b_5 & = & -0.400415137801391986, \\
a_{54} & = & -0.185521600243482257, & b_6 & = & -0.007015172007017765, \\
a_{61} & = & -1.162652454555768030, & b_7 & = & 0.102779548648663215, \\
a_{62} & = & 1.481947930561845810, & c_2 & = & 0.286082581254413320, \\
a_{63} & = & 0.172412714294101567, & c_3 & = & 0.286082581254413330, \\
a_{64} & = & -1.306027560126403490, & c_4 & = & 0.779893050447479079, \\
a_{65} & = & 1.100401951080637470, & c_5 & = & 0.834087701076002234, \\
a_{71} & = & -0.364527635518304012, & c_6 & = & 0.286082581254413333, \\
a_{72} & = & -1.802838459297476900, & c_7 & = & 1.
\end{array}
$$

Эта матрица дает пример 7-ми стадийного метода Рунге-Кутта порядка 6, у которого коэффициент $b_2 \neq 0$. Следовательно, такой метод не входит в семейства методов, полученных Д.Бутчером (а так же и в семейство, найденное Г.М.Хаммудом [16].) Значит, обнаружено новое семейство подобных методов. Программа показывает и размерность такого семейства - она равна 3. Все найденные численно решения из этого семейства обладают свойством $c_2 = c_3 = c_6$.

Таким образом, мы знаем о существовании некоторого 3-х мерного семейства решений, обладающего свойством $c_2 = c_3 = c_6$. Но на сегодняшний день известны лишь отдельные решения из этого семейства, найденные численно. Хотелось бы построить все семейство аналитически.

Мы видим, что решить полную систему Бутчера для методов Рунге-Кутта порядка больше или равного 6 на сегодняшний день не удается. Численное решение также сталкивается со многими проблемами [25]. Для численного нахождения методов порядка 8 требуется несколько тысяч итераций, причем процесс сходится к решению далеко не всегда.

Для методов порядка больше или равного 9 не известно даже минимальное количество шагов, при котором такой метод существует. Известно лишь, что для метода 9-го порядка минимальное количество шагов лежит в пределах от 11 до 17. Объясняется такая ситуация большой сложностью системы Бутчера [8], которой должны удовлетворять коэффициенты метода. Если для методов 4-го порядка (классический случай) количество уравнений равно 8 и их максимальный порядок равен 3, то для $p = 9$, система уравнений состоит из 486 полиномиальных уравнений до девятой степени включительно от 80...120 переменных (в зависимости от количества стадий). Даже методам современной компьютерной алгебры эта задача оказывается не под силу. Учитывая быстрый рост объема промежуточных вычислений при увеличении размера системы, не приходится надеяться, что увеличение мощности компьютеров, даже на несколько порядков, поможет справиться с проблемой.

Еще одной особенностью рассматриваемой системы уравнений является большое количество компонент связности многообразия решений. Для методов до 5-го порядка включительно, многообразие решений уравнений Бутчера неприводимо, то есть имеется лишь одно семейство решений. Начиная с порядка 6 количество компонент связности больше 1 и, как можно предположить, быстро растет. Сложность состоит в том, что для каждой неприводимой компоненты многообразия, процедуру решения приходится проводить отдельно. Учитывая описанные проблемы, необходимо искать какие-то новые подходы к разрешению системы уравнений Бутчера. В работах [18, 16, 15, 9] и некоторых других, предложено искать решения системы уравнений численными методами. Такой подход принес свои плоды: удалось получить большое количество неизвестных ранее методов высокого порядка (6, 7, 8). Детальное изучение найденных методов позволило аналитически построить несколько неизвестных ранее семейств 7-шаговых методов порядка 6 [16, 15]. Дальнейшее продвижение в этом направлении затормозилось из-за сильной вырожденности системы Бутчера для методов высокого порядка. В результате, в типичном случае, для нахождения метода 8 порядка требуется несколько тысяч шагов. Методы же порядка 9 и выше найти вообще не удалось.

В работе [20] предложено рассмотреть вместо полной системы уравнений, некоторую ее часть ("каркас"), которую можно надеяться решить в общем виде.

Этой задаче для 7-стадийных методов 6-го порядка и посвящена настоящая работа.

Глава 2.

Каркас

2.1. Каркас метода.

Коэффициенты a_{ij}, b_i метода Рунге-Кутта записываются в виде нижнетреугольной матрицы A с нулевой диагональю:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & \dots & 0 & 0 & 0 \\ & & & & \dots & & & \\ a_{n,1} & a_{n,2} & a_{n,3} & & \dots & a_{n,n-1} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & \dots & b_{n-1} & b_n & 0 \end{pmatrix}.$$

размера $(n+1) \cdot (n+1)$, количество неизвестных (без упрощающих предположений) равно $n(n+1)/2$, где n - количество стадий.

Коэффициенты c_i, которые уже упоминались ранее (п. 1.2.1) будем рассматривать как элементы вектора-столбца Ae, где e - матрица из одного столбца, составленного из единиц.

2.1.1. Определение. Полным каркасом метода Рунге-Кутта называется набор переменных

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ c_{21} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ c_{31} & c_{32} & 0 & 0 & \dots & 0 & 0 & 0 \\ c_{41} & c_{42} & c_{43} & 0 & \dots & 0 & 0 & 0 \\ & & & & \dots & & & \\ c_{n,1} & c_{n,2} & c_{n,3} & & \dots & c_{n,n-1} & 0 & 0 \end{pmatrix}.$$

где первый столбец c_{k1} совпадает с переменными c_k, описанными выше: $c_{k1} = c_k = A \cdot e$, второй столбец $c_{k2} = A^2 \cdot e$, третий $c_{k3} = A^3 \cdot e$ и так далее.

2.1.2. Теорема 1. Коэффициенты a_{ij} выражаются через c_{ij} в полиномиальном виде (то есть в виде полиномиальных функций).

Из теоремы следует, что набор переменных (c_{ij}, b_j) эквивалентен набору (a_{ij}, b_j), то есть переменные c_{ij} являются просто новой системой координат на том же пространстве. А значит, всю систему уравнений Бутчера можно переписать в новых координатах. Пытаясь довести этот процесс до конца, мы обнаружим, что в новой системе координат полная система уравнений Бутчера получаются еще более громоздкой, чем первоначальная. Порядок уравнений в новой системе также очень сильно возрастает. Поэтому попытка просто перейти к новой системе переменных оказывается неудачной — система уравнений от такой замены не упрощается, а становится лишь сложнее. Тем не менее, из этой идеи можно извлечь рациональное зерно.

2.1.3.Определение 2. Каркасом k-го порядка метода Рунге-Кутта называется набор из первых k столбцов матрицы полного каркаса C. Таким образом, каркас 1-го порядка состоит из коэффициентов $c_{k1} = c_k = Ae$, каркас 2-го порядка - из двух векторов $c_{k1} = Ae$ и $c_{k2} = A^2e$ и так далее.

Оказывается, что значительную часть уравнений Бутчера можно выписать, используя лишь каркас небольшого порядка и вектор $b = \{b_i\}$. Например, уравнения для каркаса 1-го порядка выглядят так:

$$\sum_{i=1}^{n} b_i c_{i1}^k = \frac{1}{k+1}, k = 0, \dots, p-1,$$

или, учитывая, что $c_{11} = 0$:

$$\begin{array}{lll} b_1 & + \ldots + b_n & = 1, \\ b_2 c_{21} & + \ldots + b_n c_{n1} & = 1/2, \\ & \ldots & \\ b_2 c_{21}^{p-1} & + \ldots + b_n c_{n1}^{p-1} & = 1/p, \end{array}$$

Таким образом, для каркаса первого порядка получаем систему p уравнений от $2n-1$ переменной $b_1, \ldots, b_n, c_{21}, \ldots, c_{n1}$. Если количество стадий совпадает с порядком метода, то из этой системы уравнений можно выразить все переменные b_i через c_i. К сожалению, из-за так называемых "барьеров Бутчера" это возможно лишь для методов 4-го порядка.

Рассмотрим сначала случай $p = n = 4$ (четырехстадийный метод 4-го порядка). Система уравнений для переменных $b_i, c_{i1}, c_{i2}, a_{ij}$ будет выглядеть так:

$$\begin{array}{llr} b_1 + b_2 + b_3 + b_4 & = 1; & (1) \\ b_2c_{21} + b_3c_{31} + b_4c_{41} & = 1/2; & (2) \\ b_3c_{32} + b_4c_{42} & = 1/6; & (3) \\ b_2c_{21}^2 + b_3c_{31}^2 + b_4c_{41}^2 & = 1/3; & (4) \\ b_4a_{43}c_{32} & = 1/24; & (5) \\ b_3c_{31}c_{32} + b_4c_{41}c_{42} & = 1/8; & (6) \\ b_2c_{21}^3 + b_3c_{31}^3 + b_4c_{41}^3 & = 1/4; & (7) \\ b_3c_{21}c_{32} + b_4c_{21}c_{42} + b_4c_{21}a_{43}c_{31} + b_4a_{43}c_{31}^2 & = 1/12; & (8) \end{array}$$

Это полная система из 8 уравнений, из них в каркас входят 6 - все, кроме 5 и 8. В оставшиеся два уравнения входят переменные a_{43} и a_{42}, не входящие в каркас.

Приведем решение каркасной системы:

$$
\begin{array}{lll}
b_1+b_2+b_3+b_4 & =1; & (1)\\
b_2c_{21}+b_3c_{31}+b_4c_{41} & =1/2; & (2)\\
b_3c_{32}+b_4c_{42} & =1/6; & (3)\\
b_2c_{21}^2+b_3c_{31}^2+b_4c_{41}^2 & =1/3; & (4)\\
b_3c_{31}c_{32}+b_4c_{41}c_{42} & =1/8; & (6)\\
b_2c_{21}^3+b_3c_{31}^3+b_4c_{41}^3 & =1/4; & (7)
\end{array}.
$$

Заметим, что уравнения 3 и 6 линейны относительно переменных c_{32}, c_{42}. Поэтому из них можно выразить переменные c_{32}, c_{42} через b_i, c_{i1}.

Затем, аналогичным образом из уравнений 2, 4 и 7 мы сможем выразить переменые b_2, b_3, b_4 через c_{21}, c_{31}, c_{41}. Наконец, переменная b_1 находится из 1-го уравнения.

$$
\begin{aligned}
b_1 &= \frac{1}{12}\frac{12c_{21}c_{31}c_{41}-6c_{21}c_{31}+4c_{21}-6c_{21}c_{41}+4c_{31}-6c_{31}c_{41}-3+4c_{41}}{c_{21}c_{31}c_{41}};\\
b_2 &= \frac{1}{12}\frac{-4c_{31}+6c_{31}c_{41}+3-4c_{41}}{c_{21}(c_{21}-c_{41})(c_{21}-c_{31})};\\
b_3 &= -\frac{1}{12}\frac{-4c_{21}+6c_{21}c_{41}+3-4c_{41}}{(c_{21}-c_{31})c_{31}(c_{31}-c_{41})};\\
b_4 &= \frac{1}{12}\frac{-4c_{21}+6c_{21}c_{31}+3-4c_{31}}{c_{41}(-c_{21}c_{41}+c_{21}c_{31}+c_{41}^2-c_{31}c_{41})};\\
c_{32} &= -\frac{1}{24}\frac{-3+4c_{41}}{(c_{31}-c_{41})b_3};\\
c_{42} &= \frac{1}{24}\frac{4c_{31}-3}{b_4(c_{31}-c_{41})};
\end{aligned}
$$

На этом решение каркасной системы завершено. Перейдем теперь, от решения каркасной системы к решению полной. Для этого нам остается выразить переменные c_{41} и a_{43} через свободные (c_{21}, c_{31}).

Подставим все найденные переменные в уравнение 5. Оно оказывается линейным относительно переменной a_{43}, найдем из него a_{43}:

$$
a_{43} = \frac{(2c_{21}-1)(c_{31}-1)(c_{21}-1)}{c_{31}(-4c_{21}^2+6c_{21}^2c_{31}+3c_{21}-6c_{21}c_{31}^2-3c_{31}+4c_{31}^2)};
$$

Подставив найденное значение в уравнение 8, получим, что значение переменной c_{41} обязательно должно равняться 1.

Для случая уравнений 4 и 5-го порядков решение полной систему уравнений Бутчера в значительной степени сводится к решению карскасной.

Поэтому можно надеятся, что и для случая уравнений большего порядка каркасная система отражает решения полной системы.

Для случая $p = 6$, $n = 7$ каркасная система все равно оказывается достоточно сложной. Но тем не менее, она все-таки поддается решению. Настоящая статья и посвящена этой задаче.

2.2. Каркас методов порядка 6

Однако и для более высоких порядков выписанные уравнения все-же позволяют упростить решение общей системы.

Рассмотрим теперь более подробно каркас 2-го порядка на примере 7-стадийных методов порядка 6. При исходном наборе переменных a_{ij}, b_j вводим новые переменные $c_{i1} = a_{i1} + \ldots + a_{i,i-1}$, которые заменят a_{i1}:

$$\begin{aligned}
c_{21} &= a_{21},\\
c_{31} &= a_{31} + a_{32},\\
c_{41} &= a_{41} + a_{42} + a_{43},\\
c_{51} &= a_{51} + a_{52} + a_{53} + a_{54},\\
c_{61} &= a_{61} + a_{62} + a_{63} + a_{64} + a_{65},\\
c_{71} &= a_{71} + a_{72} + a_{73} + a_{74} + a_{75} + a_{76},
\end{aligned}$$

Обратно: переменные a_{i1} выражаем через c_{i1} и остальные a_{ij} по формулам:

$$\begin{aligned}
a_{21} &= c_{21},\\
a_{31} &= c_{31} - a_{32},\\
a_{41} &= c_{41} - a_{42} - a_{43},\\
a_{51} &= c_{51} - a_{52} - a_{53} - a_{54},\\
a_{61} &= c_{61} - a_{62} - a_{63} - a_{64} - a_{65},\\
a_{71} &= c_{71} - a_{72} - a_{73} - a_{74} - a_{75} - a_{76},
\end{aligned}$$

Таким образом, новая система переменных однозначно выражается через старую и наоборот.

Затем сделаем еще одну аналогичную замену: вводим переменные c_{i2} вместо переменных a_{i2}.

$$c_{i2} = a_{i2} \cdot c_{21} + \ldots + a_{i,i-1} \cdot c_{i-1,2}$$

или

$$\begin{aligned}
c_{32} &= a_{32}c_{21},\\
c_{42} &= a_{42}c_{21} + a_{43}c_{31},\\
c_{52} &= a_{52}c_{21} + a_{53}c_{31} + a_{54}c_{41},\\
c_{62} &= a_{62}c_{21} + a_{63}c_{31} + a_{64}c_{41} + a_{65}c_{51},\\
c_{72} &= a_{72}c_{21} + a_{73}c_{31} + a_{74}c_{41} + a_{75}c_{51} + a_{76}c_{6},
\end{aligned}$$

Обратная замена переменных осуществляется по следующим формулам:

$$\begin{aligned}
a_{32} &= c_{32}/c_{21},\\
a_{42} &= (c_{42} - a_{43}c_{31})/c_{21},\\
a_{52} &= (c_{52} - a_{53}c_{31} - a_{54}c_{41})/c_{21},\\
a_{62} &= (c_{62} - a_{63}c_{31} - a_{64}c_{41} - a_{65}c_{51})/c_{21},\\
a_{72} &= (c_{72} - a_{73}c_{31} - a_{74}c_{41} - a_{75}c_{51} - a_{76}c_{61})/c_{21},
\end{aligned}$$

Таким образом, мы заменили часть переменных (a_{i1}, a_{i2}) на другие (c_{i1}, c_{i2}).

Если записать исходную систему уравнений Бутчера в новых переменных, то окажется, что многие уравнения зависят лишь от переменных (b_i, c_{i1}, c_{i2}).

Пусть A - нижнетреугольная матрица коэффициентов метода РК. Для ее нахождения мы имеем систему 37 уравнений Бутчера от 28 переменных (a_{ij},b_j). Для использования в "каркасе" выберем новые переменные c_{i1}, c_{i2}, через которые выражается значительное количество уравнвнений.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & 0 \end{pmatrix}.$$

Для записи уравнений Бутчера примем следующие обозначения.
Обозначем через c_{i1} сумма чисел в i-й строке матрицы A то есть
$a_{i1} = c_{i1} - \ldots - a_{i,i-1}$

$$\tilde{A} \cdot e = \begin{pmatrix} 0 \\ a_{21} \\ a_{31} + a_{32} \\ a_{41} + a_{42} + a_{43} \\ a_{51} + a_{52} + a_{53} + a_{54} \\ a_{61} + a_{62} + a_{63} + a_{64} + a_{65} \\ a_{71} + a_{72} + a_{73} + a_{74} + a_{75} + a_{76} \end{pmatrix}.$$

Где e-вектор состоящий из одних единиц:

$$e = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Обозначим через d вектор, состоящий из элементов

$$c_{i2} = a_{i2} \cdot c_{21} + \ldots + a_{i,i-1} \cdot c_{i-1,1}$$

$$d = \tilde{A}^2 \cdot e = \begin{pmatrix} 0 \\ 0 \\ a_{32} \cdot c_{21} \\ a_{42} \cdot c_{21} + a_{43}c_{31} \\ a_{52} \cdot c_{21} + a_{53}c_{31} + a_{54}c_{41} \\ a_{62} \cdot c_{21} + a_{63}c_{31} + a_{64}c_{41} + a_{65}c_{51} \\ a_{72} \cdot c_{21} + a_{73}c_{31} + a_{74}c_{41} + a_{75}c_{51} + a_{76}c_{161} \end{pmatrix}.$$

Из всей системы 37 уравнений Бутчера, часть содержит только переменных b_k и c_k:

$$\sum_{i=2}^{n} b_i \cdot c_{i1}^k = \frac{1}{1+k}$$

при k=0,1,... p-1. (1)

Из этой подсистемы уравнения найти значения b_i, c_{i1} обычно не удается - это оказывается возможным лишь в отдельных, весьма редких случаях. Например, в работе [16] изучались 7-шаговые методы порядка 6 ($n = 7$, $p = 6$), у которых $b_2 = b_3 = b_4 = 0$. Используя этот факт, удалось найти и общее решение всей системы уравнений Бутчера для рассматриваемого случая.

Таким образом, всего у нас имеется 18 "каркасных" переменных:

$b_1, b_2, b_3, b_4, b_5, b_6, b_7, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}, c_{32}, c_{42}, c_{52}, c_{62}, c_{72}$.

Детальное изучение системы уравнений показывает, что будет удобно, вместо переменных c_{i2} ввести новые переменные e_i по следующим формулам: $e_i = c_{i2} - c_{i1}^2/2$. Например, второе упрощающее предположение тогда будет выглядеть так: $b_i \cdot e_i = 0$. Таким образом, для решения в этих переменных мы сможем сразу сказать, выполнено второе упрощающее предположение, или нет.

Выпишем уравнения, содержащие каркас 2-го порядка:

$$\sum_{i=2}^{n} b_i \cdot c_{il}^k \cdot c_{i2}^l = \frac{1}{2^l \cdot (1+k+2 \cdot l)}$$

при $k, l >= 0$, $k + 2 \cdot l < p$. (2)

Учитывая соотношения : $b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 = 1$, $c_7 = 1$ и $d_7 = 1/2(e_7 = 0)$, в окончательную систему уравнений, переменные $b_1, c_7, d_7(e_7)$ можно не включать.

Мы видим, что от системы из 37 уравнений Бутчера от 28 переменных, в каркас попадают 11 уравнений от 15 переменных.

Итак, каркас 2-го порядка в этом случае состоит из 11 переменных : $c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}$, $c_{32}, c_{42}, c_{52}, c_{62}, c_{72}$. Дополнительно, из первого упрощающего предположения [8] мы получаем: $c_{71} = 1$, из второго: $c_{72} = 1/2$. Вместе с 6 переменными $(b_2, b_3, b_4, b_5, b_6, b_7)$, мы получаем систему от 15 переменных. Они должны удовлетворять 5 уравнениям вида (1), при $k = 1, 2, 3, 4, 5$. Уравнение вида (1) при $k = 0$ выглядит так: $b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 = 1$. Это единственное уравнение в системе Бутчера, содержащее b_1, поэтому оно используется лишь в самом конце для нахождения b_1 и в дальнейшем не рассматривается. Помимо этого, мы имеем еще 6 уравнений вида (2) при следующих значениях пары (k, l) при $k + 2 \cdot l < 6$:

$(k, l) = (0, 1), (0, 2), (1, 1), (1, 2), (2, 1), (3, 1)$.

В результате, мы получаем систему 11 уравнений от 15 переменных.

Рассмотрим теперь эти уравнения более подробно.

2.3. Решение каркасной системы

В матричных обозначениях эта система выглядит следующим образом:

$$
\begin{array}{lll}
1) & \{b*\tilde{A}e\} & = 1/2, \\
2) & \{b*\tilde{A}e*\tilde{A}\} & = 1/3, \\
3) & \{b*\tilde{A}^2e*\tilde{A}e\} & = 1/4, \\
4) & \{b*\tilde{A}^3e*\tilde{A}e\} & = 1/5, \\
5) & \{b*\tilde{A}^4e*\tilde{A}e\} & = 1/6, \\
6) & \{b*\tilde{A}e*\tilde{A}e*\tilde{A}e*\tilde{A}e*\tilde{A}e\} & = 1/6. \\
7) & \{b*\tilde{A}^2e*\tilde{A}^2e\} & = 1/20, \\
8) & \{b*\tilde{A}e*\tilde{A}e*\tilde{A}e\} & = 1/8, \\
9) & \{b*\tilde{A}^2e*\tilde{A}e*\tilde{A}e\} & = 1/10, \\
10) & \{b*\tilde{A}^2e*\tilde{A}e*\tilde{A}e*\tilde{A}e\} & = 1/12, \\
11) & \{b*\tilde{A}(\tilde{A}e*\tilde{A}e)*\tilde{A}e\} & = 1/24,
\end{array}
$$

Для удобства введем упрощенные обозначения: вместо c_{i1} будем писать просто c_i, вместо c_{i2} будем писать d_i.

Раскрыв эту систему, то есть перейдя от матричных обозначений к обычным, получим:

$$
\begin{array}{ll}
b_2c_2 + b_3c_3 + b_4c_4 + b_5c_5 + b_6c_6 + b_7c_7 & = 1/2; \\
b_2c_2^2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2 + b_6c_6^2 + b_7c_7^2 & = 1/3; \\
b_2c_2^3 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3 + b_6c_6^3 + b_7c_7^3 & = 1/4; \\
b_2c_2^4 + b_3c_3^4 + b_4c_4^4 + b_5c_5^4 + b_6c_6^4 + b_7c_7^4 & = 1/5; \\
b_2c_2^5 + b_3c_3^5 + b_4c_4^5 + b_5c_5^5 + b_6c_6^5 + b_7c_7^5 & = 1/6; \\
b_3d_3 + b_4d_4 + b_5d_5 + b_6d_6 + b_7d_7 & = 1/6; \\
b_3d_3^2 + b_4d_4^2 + b_5d_5^2 + b_6d_6^2 + b_7d_7^2 & = 1/20; \\
b_3c_3d_3 + b_4c_4d_4 + b_5c_5d_5 + b_6c_6d_6 + b_7c_7d_7 & = 1/8; \\
b_3c_3^2d_3 + b_4c_4^2d_4 + b_5c_5^2d_5 + b_6c_6^2d_6 + b_7c_7^2d_7 & = 1/10; \\
b_3c_3^3d_3 + b_4c_4^3d_4 + b_5c_5^3d_5 + b_6c_6^3d_6 + b_7c_7^3d_7 & = 1/12; \\
b_3c_3d_3^2 + b_4c_4d_4^2 + b_5c_5d_5^2 + b_6c_6d_6^2 + b_7c_7d_7^2 & = 1/24;
\end{array}
$$

Это и есть та самая "каркасная" система уравнений, которую мы будем решать. Если в тех же уравнениях вместо переменных d_i использовать $e_i = d_i - c_i^2/2$, получим:

$$
\begin{array}{ll}
b_2c_2 + b_3c_3 + b_4c_4 + b_5c_5 + b_6c_6 + b_7c_7 & = 1/2; \\
b_2c_2^2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2 + b_6c_6^2 + b_7c_7^2 & = 1/3; \\
b_2c_2^3 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3 + b_6c_6^3 + b_7c_7^3 & = 1/4; \\
b_2c_2^4 + b_3c_3^4 + b_4c_4^4 + b_5c_5^4 + b_6c_6^4 + b_7c_7^4 & = 1/5; \\
b_2c_2^5 + b_3c_3^5 + b_4c_4^5 + b_5c_5^5 + b_6c_6^5 + b_7c_7^5 & = 1/6; \\
b_3e_3 + b_4e_4 + b_5e_5 + b_6e_6 - \frac{1}{2}b_2c_2^2 & = 0; \\
b_3e_3^2 + b_4e_4^2 + b_5e_5^2 + b_6e_6^2 + \frac{1}{4}b_2c_2^4 & = 0; \\
b_3c_3e_3 + b_4c_4e_4 + b_5c_5e_5 + b_6c_6e_6 - \frac{1}{2}b_2c_2^3 & = 0; \\
b_3c_3^2e_3 + b_4c_4^2e_4 + b_5c_5^2e_5 + b_6c_6^2e_6 - \frac{1}{2}b_2c_2^4 & = 0; \\
b_3c_3^3e_3 + b_4c_4^3e_4 + b_5c_5^3e_5 + b_6c_6^3e_6 - \frac{1}{2}b_2c_2^4 & = 0; \\
b_3c_3e_3^2 + b_4c_4e_4^2 + b_5c_5e_5^2 + b_6c_6e_6^2 + \frac{1}{4}b_2c_2^5 & = 0;
\end{array}
$$

Мы видим, что уравнения (6)..(11) стали однородными, что несколько упростит их решение.

Рассмотрим отдельно два случая:

$b_2 = 0$;

$b_2 \neq 0$.

2.3.1. Теорема 2. Для каркаса 2-го порядка для случая $b_2 \neq 0$, существует решение системы уравнений (1.2) со свободными переменными (c_3, c_4, c_5, c_6).

2.4. Решение в случае $b_2 = 0$

Если $b_2 = 0$,то система уравнений $(eq_6, eq_7, eq_8, eq_9, eq_{10}, eq_{11})$ будет выглядеть так:

$$
\begin{aligned}
&b_3 \cdot e_3 + b_4 \cdot e_4 + b_5 \cdot e_5 + b_6 \cdot e_6 &&= 0; &&(6)\\
&b_3 \cdot e_3^2 + b_4 \cdot e_4^2 + b_5 \cdot e_5^2 + b_6 \cdot e_6^2 &&= 0; &&(7)\\
&\tfrac{1}{5} \cdot b_3 \cdot e_3 + \tfrac{1}{3} \cdot b_4 \cdot e_4 + \tfrac{2}{5} \cdot b_5 \cdot e_5 + \tfrac{2}{3} \cdot b_6 \cdot e_6 &&= 0; &&(8)\\
&\tfrac{1}{25} \cdot b_3 \cdot e_3 + \tfrac{1}{9} \cdot b_4 \cdot e_4 + \tfrac{4}{25} \cdot b_5 \cdot e_5 + \tfrac{4}{9} \cdot b_6 \cdot e_6 &&= 0; &&(9)\\
&\tfrac{1}{125} \cdot b_3 \cdot e_3 + \tfrac{1}{27} \cdot b_4 \cdot e_4 + \tfrac{8}{125} \cdot b_5 \cdot e_5 + \tfrac{8}{27} \cdot b_6 \cdot e_6 &&= 0; &&(10)\\
&\tfrac{1}{5} \cdot b_3 \cdot e_3^2 + \tfrac{1}{3} \cdot b_4 \cdot e_4^2 + \tfrac{2}{5} \cdot b_5 \cdot e_5^2 + \tfrac{2}{3} \cdot b_6 \cdot e_6^2 &&= 0; &&(11)
\end{aligned}
$$

В качестве свободных выберем переменные c_3, c_4, c_5, c_6. Решив эту систему относительно переменных e_i, обнаружим, что все e_i оказываются равными 0:

$$e_3 = 0,\ e_4 = 0,\ e_5 = 0,\ e_6 = 0,\ e_7 = 0.$$

В результате, если $b_2 = 0$, то значения всех d_i однозначно находятся: $d_i = c_i^2/2$.

Таким образом, мы получаем решение удовлетворяющее "2-му упрощающему предположений" [8]. Дж. Бутчером были найдены его решения имено при этом предположении. Для наглядности параллельно с общими рассуждениями проведем вычисления и для некоторых конкретных значений свободных переменных:

$$c_3 = 1/5,\ c_4 = 1/3,\ c_5 = 2/5,\ c_6 = 2/3,\ c_7 = 1.$$

Переменные d_i найдем из соотношений: $d_i = c_i^2/2$:

$$d_3 = 1/50,\ d_4 = 1/18,\ d_5 = 2/25,\ d_6 = 2/9,\ d_7 = 1/2.$$

Оставшаяся система уравнений имеет следующий вид (5 линейных уравнений от 5 неизвестных $b2, \ldots, b_7$) которая решается очень просто.

$$
\begin{aligned}
1/5 \cdot b_3 + 1/3 \cdot b_4 + 2/5 \cdot b_5 + 2/3 \cdot b_6 + b_7 &= 1/2;\\
1/25 \cdot b_3 + 1/9 \cdot b_4 + 4/25 \cdot b_5 + 4/9 \cdot b_6 + b_7 &= 1/3;\\
1/125 \cdot b_3 + 1/27 \cdot b_4 + 8/125 \cdot b_5 + 8/27 \cdot b_6 + b_7 &= 1/4;\\
1/625 \cdot b_3 + 1/81 \cdot b_4 + 16/625 \cdot b_5 + 16/81 \cdot b_6 + b_7 &= 1/5;\\
1/3125 \cdot b_3 + 1/243 \cdot b_4 + 32/3125 \cdot b_5 + 32/243 \cdot b_6 + b_7 &= 1/6.
\end{aligned}
$$

Из этих уравнений найдем решение :

$$\{b_2, b_3, b_4, b_5, b_6, b_7\} = \{0, -125/672, 27/16, -125/96, 135/224, 3/32\}$$

Аналогично, поступим и в общем случае. Итак:

$$\begin{aligned}
b_2 &= 0;\\
b_3 &= \frac{1/60\cdot(10\cdot c_5\cdot c_4\cdot c_6-5\cdot c_4\cdot c_5-5\cdot c_4\cdot c_6+3\cdot c_4-5\cdot c_5\cdot c_6+3\cdot c_5+3\cdot c_6-2)}{(c_3-1)\cdot c_3\cdot(-c_6+c_3)\cdot(-c_5+c_3)\cdot(-c_4+c_3)};\\
b_4 &= \frac{-1/60\cdot(10\cdot c_5\cdot c_3\cdot c_6-5\cdot c_3\cdot c_5-5\cdot c_3\cdot c_6+3\cdot c_3-5\cdot c_5\cdot+3\cdot c_5+3\cdot c_6-2)}{(c_4-1)\cdot(-c_4+c_3)\cdot c_4\cdot(-c_6+c_4)\cdot(-c_5+c_4)};\\
b_5 &= \frac{1/60\cdot(10\cdot c_4\cdot c_3\cdot c_6-5\cdot c_4\cdot c_3-5\cdot c_3\cdot c_6+3\cdot c_3-5\cdot c_4\cdot c_6+3\cdot c_4+3\cdot c_6-2)}{(c_5-1)\cdot(-c_5+c_4)\cdot(-c_5+c_3)\cdot c_5\cdot(-c_6+c_5)};\\
b_6 &= \frac{-1/60\cdot(3\cdot c_3+3\cdot c_4+3\cdot c_5-5\cdot c_4\cdot c_3-5\cdot c_4\cdot c_5+10\cdot c_4\cdot c_3\cdot c_5-5\cdot c_3\cdot c_5-2)}{c_6\cdot(c_6-1)\cdot(-c_6+c_5)\cdot(-c_6+c_4)\cdot(-c_6+c_3)};\\
b_7 &= \frac{1/60\cdot(30\cdot c_5\cdot c_4\cdot c_3\cdot c_6-20\cdot c_4\cdot c_3\cdot c_5+15\cdot c_4\cdot c_3-20\cdot c_4\cdot c_3\cdot c_6+15\cdot c_3\cdot c_5-20\cdot c_5\cdot c_3\cdot c_6-12\cdot c_3+15\cdot c_3\cdot c_6)}{(c_6-1)\cdot(c_5-1)\cdot(c_4-1)\cdot(c_3-1)}+\\
&\frac{1/60\cdot(15\cdot c_4\cdot c_5-20\cdot c_5\cdot c_4\cdot c_6-12\cdot c_4+15\cdot c_4\cdot c_6-12\cdot c_5+15\cdot c_5\cdot c_6+10-12\cdot c_6)}{(c_6-1)\cdot(c_5-1)\cdot(c_4-1)\cdot(c_3-1)};
\end{aligned}$$

Где c_3,c_4,c_5 , c_6 отличные от 1, $c_3 \neq 0$,$c_4 \neq$,$c_5 \neq 0$,$c_6 \neq 0$ и c_3,c_4,c_5,c_6 различны.

Таким образом, в случае $b_2 = 0$ мы получили полное решение рассматриваемой системы уравнений.

2.5. Решение в случае $b_2 \neq 0$

Этот случай значительно более сложный, чем предыдущий. Учитывая упрощающие предположения[15], мы можем считать, что $c_7 = 1$ и $d_7 = 1/2$ (или $e_7 = 0$). В качестве свободных выберем переменные c_3, c_4, c_5, c_6. В этом случае коэффициенты e_i оказываются не обязательно равными 0 (кроме e_7).

Процесс решения каркасной системы начнем с рассмотрения системы уравнений (6,8,9,10). Заметим, что она линейна относительно переменных e_3, e_4, e_5, e_6.

Решив эту систему из 4-х линейных уравнений от 4-х переменных получим выражение e_i через остальные переменные.

$$\begin{aligned}
e_3 &= -\frac{1}{2}\cdot\frac{(-c_4+c_2)\cdot(c_2-c_5)\cdot(c_2-c_6)\cdot b_2\cdot c_2^2}{b_3\cdot(-c_3+c_6)\cdot(-c_3+c_5)\cdot(-c_3+c_4)};\\
e_4 &= \frac{1}{2}\cdot\frac{(-c_3+c_2)\cdot(c_2-c_5)\cdot(c_2-c_6)\cdot b_2\cdot c_2^2}{(-c_3+c_4)\cdot b_4\cdot(c_6-c_4)\cdot(c_5-c_4)};\\
e_5 &= -\frac{1}{2}\cdot\frac{(-c_3+c_2)\cdot(-c_4+c_2)\cdot(c_2-c_6)\cdot b_2\cdot c_2^2}{(-c_3+c_5)\cdot(c_5-c_4)\cdot b_5\cdot(c_6-c_5)};\\
e_6 &= \frac{1}{2}\cdot\frac{b_2\cdot c_2^2\cdot(-c_3+c_2)\cdot(-c_4+c2)\cdot(c_2-c_5)}{b_6\cdot(-c_3+c_6)\cdot(c_6-c_4)\cdot(c_6-c_5)};
\end{aligned}$$

Поскольку d_i однозначно выражаются через e_i, мы можем найти, также и переменные d_3, d_4, d_5, d_6 и d_7.

Теперь выразим переменные b_3,b_4,b_5,b_6 и b_7 через b_2 и c_2 из системы $(eq_1, eq_2, eq_3, eq_4, eq_5)$, которая линейна относительно них.

Хотя процедура решения квадратной системы линейных уравнений хорошо известна, полученные выражения в общем виде оказываются чересчур громоздкими. Чтобы продолжить вычисления, придадим свободным переменным c_3, c_4, c_5 и c_6 некоторые конкретные рациональные значения. Это позволит продемонстрировать процесс нахождения значения остальных переменных.

Итак, в качестве значений свободных переменных выберем: $c_3 = 1/5$; $c_4 = 1/3$; $c_5 = 2/5$; $c_6 = 2/3$;

Тогда переменные $b_3 \dots b_7$ будут выражаться через свободные и через пока еще на найденные переменные c_2, b_2 следующим образом:

$$\begin{aligned}
b_3 &= -\tfrac{625}{14}b_2c_2 + \tfrac{2500}{7}b_2c_2^2 - \tfrac{8125}{8}b_2c_2^3 + \tfrac{16875}{14}b_2c_2^4 - \tfrac{125}{672} - \tfrac{28125}{56}b_2c_2^5;\\
b_4 &= -810b_2c_2^2 + 81b_2c_2 + \tfrac{10611}{4}b_2c_2^3 - \tfrac{6885}{2}b_2c_2^4 + \tfrac{6075}{4}b_2c_2^5 + \tfrac{27}{16};\\
b_5 &= -\tfrac{45625}{24}b_2c_2^3 - \tfrac{625}{12}b_2c_2 + \tfrac{4375}{8}b_2c_2^2 + \tfrac{20625}{8}b_2c_2^4 - \tfrac{9375}{8}b_2c_2^5 - \tfrac{125}{96};\\
b_6 &= -\tfrac{11745}{56}b_2c_2^4 + \tfrac{81}{28}b_2c_2 - \tfrac{1863}{56}b_2c_2^2 + \tfrac{1053}{8}b_2c_2^3 + \tfrac{6075}{56}b_2c_2^5 + \tfrac{135}{224};\\
b_7 &= -\tfrac{75}{8}b_2c_2^5 - \tfrac{1}{6}b_2c_2 + 2b_2c_2^2 - \tfrac{203}{24}b_2c_2^3 + 15b_2c_2^4 + \tfrac{3}{32};
\end{aligned}$$

Подставив эти значения в выражения для e_i получим:

$$\begin{aligned}
e_3 &= \frac{24}{5}\cdot\frac{b_2\cdot c_2^2\cdot(3\cdot c_2-1)\cdot(5\cdot c_2-2)\cdot(3\cdot c_2-2)}{1-1920\cdot b_2\cdot c_2^2+5460\cdot b_2\cdot c_2^3-6480\cdot b_2\cdot c_2^4+2700\cdot b_2\cdot c_2^5+240\cdot b_2\cdot c_2};\\
e_4 &= \frac{4}{3}\cdot\frac{b_2\cdot c_2^2\cdot(5\cdot c_2-2)\cdot(3\cdot c_2-2)\cdot(5\cdot c_2-1)}{-480\cdot b_2\cdot c_2^2+48\cdot b_2\cdot c_2+1572\cdot b_2\cdot c_2^3-2040\cdot b_2\cdot c_2^4+900\cdot b_2\cdot c_2^5+1};\\
e_5 &= \frac{12}{5}\cdot\frac{b_2\cdot c_2^2\cdot(3\cdot c_2-1)\cdot(3\cdot c_2-2)\cdot(5\cdot c_2-1)}{1460\cdot b_2\cdot c_2^3+40\cdot b_2\cdot c_2-420\cdot b_2\cdot c_2^2-1980\cdot b_2\cdot c_2^4+900\cdot b_2\cdot c_2^5+1};\\
e_6 &= \frac{4}{3}\cdot\frac{b_2\cdot c_2^2\cdot(3\cdot c_2-1)\cdot(5\cdot c_2-2)\cdot(5\cdot c_2-1)}{-1740\cdot b_2\cdot c_2^4+24\cdot b_2\cdot c_2-276\cdot b_2\cdot c_2^2+1092\cdot b_2\cdot c_2^3+900\cdot b_2\cdot c_2^5+5};
\end{aligned}$$

У нас остаются два уравнения — 7-е и 11-е и две переменные c_2 и b_2, кроме свободных. Уравнения рациональные, степени 8 и весьма громоздкие. Решить их в общем виде весьма непросто. Напрямую системы компьютерной алгебры справиться с ними не в состоянии - общее решение получить не удается.

2.6. Выражение c_2 и b_2 через свободные переменные

Хотя найти общее решение оставшихся двух уравнений не удается, для каждого конкретного набора свободных переменных оставшаяся пара уравнений оказывается разрешимой, причем решение единственно.

Для нахождения решения в явном виде применим следующий подход. Заметим, что переменные $c_3, c_4, c5, c_6$ входят во все исходные каркасные уравнения симметрично. Поэтому c_2 должно быть симметричной функцией от свободных переменных.

Выпишем элементарные симметрические функции от переменных (c_3, c_4, c_5, c_6):

$$\begin{aligned}
\sigma_1 &= c_3 + c_4 + c_5 + c_6;\\
\sigma_2 &= c_3c_4 + c_3c_5 + c_3c_6 + c_4c_5 + c_4c_6 + c_5c_6;\\
\sigma_3 &= c_3c_4c_5 + c_3c_5c_6 + c_3c_4c_6 + c_4c_5c_6;\\
\sigma_4 &= c_3c_4c_5c_6;
\end{aligned}$$

В силу симметричности c_2, ее можно выражать через свободные переменные в виде некоторой рациональной функции от σ_i. Подставляя конкретные значения переменных и доводя вычисления до конца, можно убедиться, что числитель и знаменатель этого выражения будет многочленом степени 4, причем линейным по каждой переменной. Это означает, что такую функцию можно преставить в виде:

$$c_2 = \frac{\alpha_0 + \alpha_1\sigma_1 + \alpha_2\sigma_2 + \alpha_3\sigma_3 + \alpha_4\sigma_4}{\beta_0 + \beta_1\sigma_1 + \beta_2\sigma_2 + \beta_3\sigma_3 + \beta_4\sigma_4},$$

для некоторых коэффициентов α_i, β_i. Не ограничивая общности, положим $\beta_0 = 1$. Таким образом, для нахождения c_2 нам надо найти значения девяти коэффициентов α_i, β_i. Найдем их методом неопределенных коэффициентов. Для каждого конкретного набора значений c_3, c_4, c_5, c_6 определим значение c_2. Это даст нам некоторое соотношения (линейное) на α_i, β_i. Взяв достаточное количество таких частных значение, можно получить такой набор соотношений, который позволит полностью найти α_i, β_i.

Более детально, была взята система уравнений при следующих значениях переменных c_3, c_4, c_5, c_6:

$$\begin{array}{rcccc}
 & c_3, & c_4, & c_5, & c_6 \\
U_1 = eq(& 1/5, & 1/7, & 1/11, & 1/13); \\
U_2 = eq(& 2/5, & 3/7, & 4/11, & 5/13); \\
U_3 = eq(& 3/5, & 1/7, & 2/11, & 1/13); \\
U_4 = eq(& 2/5, & 4/7, & 7/11, & 3/13); \\
U_5 = eq(& 7/5, & 9/7, & 9/11, & 2/13); \\
U_6 = eq(& 3/5, & 6/7, & 2/11, & 4/13); \\
U_7 = eq(& 1/5, & 5/7, & 3/11, & 9/13); \\
U_8 = eq(& 2/5, & 4/7, & 4/11, & 4/13); \\
U_9 = eq(& 4/5, & 3/7, & 5/11, & 2/13); \\
U_{10} = eq(& 3/5, & 2/7, & 6/11, & 7/13); \\
U_{11} = eq(& 3/7, & 2/7, & 6/5, & 7/9); \\
U_{12} = eq(& 13/17, & 12/19, & 11/23, & 19/29);
\end{array}$$

Конечно, для нахождения 9 переменных достаточно было бы взять 9 уравнений, тем не менее возьмем лишние уравнения для контроля правильности решения. Из указанных наборов свободных переменных получена систему 12 линейных уравнений (от 9 переменных). Несмотря на то, что уравнений больше, чем переменных, эта система имеет единственной решение.

$$\{\alpha_0 = \frac{5}{7}, \quad \alpha_1 = -1, \quad \alpha_2 = \frac{10}{7}, \quad \alpha_3 = \frac{-15}{7}, \quad \alpha_4 = \frac{25}{7},$$

$$\beta_1 = \frac{-10}{7}, \quad \beta_2 = \frac{15}{7}, \quad \beta_3 = \frac{-25}{7}, \quad \beta_4 = \frac{50}{7}, \quad \}$$

Теперь можно выписать полную формулу для c_2:

$$c_2 = \frac{5 - 7\sigma_1 + 10\sigma_2 - 15\sigma_3 + 25\sigma_4}{7 - 10\sigma_1 + 15\sigma_2 - 25\sigma_3 + 50\sigma_4},$$

где σ_i - элементарные симметричные функции от переменных (c_3, c_4, c_5, c_6).

Аналогичный подход можно применить и для нахождения коэффициента b_2. Опуская промежуточные выкладки, сразу выпишем полученный результат:

$$b_2 = \frac{(5 - 7\sigma_1 + 10\sigma_2 - 15\sigma_3 + 25\sigma_4)^5}{37500 R(c_3, c_4, c_5, c_6) R(c_6, c_3, c_4, c_5) R(c_5, c_6, c_3, c_4) R(c_4, c_5, c_6, c_3)},$$

где

$$R(x, x_1, x_2, x_3) = (1 - 7/5\rho_1 + 2\rho_2 - 3\rho_3) +$$
$$x \cdot (-14/5 + 4\rho_1 - 6\rho_2 + 10\rho_3) +$$
$$x^2 \cdot (2 - 3\rho_1 + 5\rho_2 - 10\rho_3)$$

и

$$\rho_1 = \rho_1(x_1, x_2, x_3) = x_1 + x_2 + x_3,$$
$$\rho_2 = \rho_2(x_1, x_2, x_3) = x_1 x_2 + x_1 x_2 + x_2 x_3,$$
$$\rho_3 = \rho_3(x_1, x_2, x_3) = x_1 x_2 x_3$$

- элементарные симметричные функции от переменных x_1, x_2, x_3.

В целом, процесс нахождения метода состоит из следующих шагов.

1. Переменные c_3, c_4, c_5, c_6 являются свободными, и принимают любые значения.

2. Находим значения c_2 и b_2 по формулам (7) и (8) соответственно.

3. Находим значения e_i по формулам (5)

4. Находим значения $b_i, (i = 3 \ldots 7)$ из системы линейных уравнений $eq_1, \ldots eq_5$.

2.7. Решение в полном виде.

При подстановке найденных решений каркасной системы в полную систему уравнений Бутчера, оказывается, что не при каждом наборе описанном выше свободных переменных, существует решение полной системы. Для этого переменные c_3, c_4, c_5, c_6 должны удовлетворять некоторому соотношению. Если оно выполняется, то для такого решения каркасной системы, остальные переменные a_{ij} находятся путем решения некоторой системы линейных уравнений.

Опишем теперь решение задачи в общем виде. Изначально мы имеем систему 37 уравнении от 28 переменных. Для ее решения воспользуемся пакетом компьютеной алгебры "MAPLE V Release 4". Эта версия пакета распространяется свободно и для наших целей ее вполне достаточно.

В качестве свободных выберем переменные c_3, c_4, c_5, c_6. Все остальных мы должны через них выразить.

Часть переменных выражается через свободные с помощью каркаса, а именно, мы нашли значения $c_2, b_2, b_3, b_4, b_5, b_6, b_7, e_3, e_4, e_5, e_6, e_7$ и все значения d_i, где $d_i = c_i^2/2$. Кроме того, через указанные переменные мы можем выразить:

$$a_{i1} = c_i - \ldots - a_{i,i-1},$$

$$a_{i2} = (d_i - \ldots - a_{i,i-1} \cdot c_{i-1})/c_2$$

а так же

$$a_{76} = (1 - c_6) \cdot b_6/b_7;$$

$$a_{75} = ((1 - c_5) \cdot b_5 - b_6 \cdot a_{65})/b_7;$$

$$a_{74} = ((1 - c_4) \cdot b_4 - b_6 \cdot a_{64} - b_5 \cdot a_{54})/b_7;$$

$$a_{73} = ((1 - c_3) \cdot b_3 - b_6 \cdot a_{63} - b_5 \cdot a_{53} - b_4 \cdot a_{43})/b_7;$$

Для получения решения, зададим переменным c_3, c_4, c_5 производные значения, а переменную c_6 оставим свободной. После этого, мы сможем выразить из оставшихся уравнений все остальные переменные, 34 уравнения (из 37) при этом обратятся в 0. Но три уравнения $eq_{21}, eq_{30}, eq_{32}$ окажутся не равными нулю тождественно. Это означает, что переменная c_6 может принимать не произовольное значение, а лишь одно из трех значений (в численном примере – 105/151 или 173/240 или 971/1345). Лишь при этих значениях переменной все уравнения обращаются в 0. Но при одном из найденных значений ($c_6 = 971/1345$) переменная c_2 оказывается равной 0, что невозможно. Таким образом, у нас остается 2 случая (в численном примере 105/151 или 173/240). Рассмотрим их более подробно.

2.7.1. Случае $c_6 = 105/151$.

Пусть:

$c_3 = 1/5; \quad c_4 = 1/3; \quad c_5 = 1/11; \quad c_6 = 105/151;$

С помощью каркаса, мы решили систему из 11 уравнении от 15 переменных. Сначала из системы $(u_6, u_7.u_8, u_9, u_{10})$, найдены переменные e_3, e_4, e_5, e_6 $(e_7 = 0)$ и далее из системы $(u_1, u_2.u_3, u_4, u_5)$ находим значения b_3, b_4, b_5, b_6, b_7. Зная b_3, b_4, b_5, b_6, b_7, автоматически находятся e_3, e_4, e_5, e_6.

Таким обрзом, с помощью каркаса, нам известные значения следующих переменных:

$$
\begin{aligned}
c_2 &= \tfrac{4}{5}, \\
b_2 &= \tfrac{168625}{2551068}, \\
a_{21} &= \tfrac{4}{5}, \\
a_{31} &= \tfrac{23}{40}, \\
a_{41} &= \tfrac{1}{3} - a_{42} - a_{43}, \\
a_{51} &= \tfrac{1}{11} - a_{52} - a_{53} - a_{54}, \\
a_{61} &= \tfrac{1}{13} - a_{62} - a_{63} - a_{64} - a_{65}, \\
a_{71} &= 1 - a_{72} - a_{73} - a_{74} - a_{75} - a_{76}, \\
a_{32} &= -\tfrac{3}{8}, \\
a_{42} &= -\tfrac{299}{392} - \tfrac{1}{4} \cdot a_{43}, \\
a_{52} &= -\tfrac{1131}{968} - \tfrac{1}{4} \cdot a_{53} - \tfrac{5}{28} \cdot a_{54}, \\
a_{62} &= \tfrac{76709}{182408} - \tfrac{1}{4} \cdot a_{63} - \tfrac{5}{28} \cdot a_{64} - \tfrac{5}{44} \cdot a_{65}, \\
a_{72} &= \tfrac{5}{8} - \tfrac{1}{4} \cdot a_{73} - \tfrac{5}{28} \cdot a_{74} - \tfrac{5}{44} \cdot a_{75} - \tfrac{525}{604} \cdot a_{76}, \\
a_{76} &= \tfrac{11957368823}{8661941728}, \\
a_{75} &= \tfrac{1390895}{469872} - \tfrac{78502725751}{17323883456} \cdot a_{65}, \\
a_{74} &= -\tfrac{511413}{53728} - \tfrac{78502725751}{17323883456} \cdot a_{64} - \tfrac{3059969}{939744} \cdot a_{54}.
\end{aligned}
$$

Таким образом, значения переменных $a_{21}, a_{31}, a_{32}, a_{41}, a_{51}, a_{61}, a_{71}, a_{76}$ и $b_1, b_2, b_3, b_4, b_5, b_6, b_7$ наидены.

Далее, из полной системы 37 уравнений от 28 переменных, уравнения

$$eq1, eq_2, eq_3, eq_4, eq_6, eq_7, eq_{15}, eq_{16}, eq_{17}, eq_{35}, eq_{36}, eq_{37}$$

оказываются равными 0 сразу.

Мы заметили, что уравнеия $eq_5, eq_8, eq_9, eq_{10}, eq_{11}, eq_{12}, eq_{13}, eq_{14}, eq_{19}, eq_{20}, eq_{22}, eq_{23}, eq_{24}, eq_{25}, eq_{26}, eq_{28}, eq_{29}, eq_{31}, eq_{32}, eq_{33}, eq_{34}$ линейные, а уравнении $eq_{18}, eq_{21}, eq_{27}, eq_{30}$ нелинейные.

Из системы уравнений $(eq_5, eq_9, eq_{12}, eq_{19}, eq_{22}, eq_{23}, eq_{31})$, было найдены значения $a_{43}, a_{53}, a_{54}, a_{63}, a_{64}, a_{65}, a_{73}$.

Выпишем теперь полностью полученную матрицу метода Рунге-Кутта:

$$
\begin{array}{llllll}
a_{21} & = & \frac{4}{5}, & a_{73} & = & \frac{-613735}{77418}, \\
a_{31} & = & \frac{23}{40}, & a_{74} & = & \frac{52463467}{3707232}, \\
a_{32} & = & \frac{-3}{8}, & a_{75} & = & \frac{-68901877}{10807056}, \\
a_{41} & = & \frac{2577}{2744}, & a_{76} & = & \frac{11957368823}{8661941728}, \\
a_{42} & = & \frac{-6187}{8232}, & b_1 & = & \frac{1}{12}, \\
a_{43} & = & \frac{-46}{1029}, & b_2 & = & \frac{168625}{2551068}, \\
a_{51} & = & \frac{13565}{10648}, & b_3 & = & \frac{168625}{161568}, \\
a_{52} & = & \frac{-2379}{2024}, & b_4 & = & \frac{-1193297}{1289472}, \\
a_{53} & = & \frac{156}{1331}, & b_5 & = & \frac{3059969}{11276928}, \\
a_{54} & = & \frac{-3822}{30613}, & b_6 & = & \frac{78502725751}{207886601472}, \\
a_{61} & = & \frac{-13159501}{82630824}, & b_7 & = & \frac{1}{12}, \\
a_{62} & = & \frac{41570650152101}{257418236021544}, & c_2 & = & \frac{4}{5}, \\
a_{63} & = & \frac{405360394802}{107616319407}, & c_3 & = & \frac{1}{5}, \\
a_{64} & = & \frac{-4235685191276}{825058448787}, & c_4 & = & \frac{1}{7}, \\
a_{65} & = & \frac{2882261109992}{1399012152291}, & c_5 & = & \frac{1}{11}, \\
a_{71} & = & \frac{1}{6}, & c_6 & = & \frac{105}{151}, \\
a_{72} & = & \frac{-3868445}{9779094}, & c_7 & = & 1,
\end{array}
$$

2.7.2.Случай $c_6 = 173/240$.
Пусть: $c_3 = 1/5$; $c_4 = 1/3$; $c_5 = 1/11$; $c_6 = 173/240$;
Аналогично, мы получили ответ и при $c_6 = 173/240$ в следующем виде:

$$
\begin{array}{llllll}
a_{21} & = & \frac{-1}{2}, & a_{73} & = & \frac{-179664}{2345}, \\
a_{31} & = & \frac{109}{475}, & a_{74} & = & \frac{368348827}{2342052}, \\
a_{32} & = & \frac{-14}{475}, & a_{75} & = & \frac{-1420811887}{17381676}, \\
a_{41} & = & \frac{106/343}{}, & a_{76} & = & \frac{3378806784}{2365642445}, \\
a_{42} & = & \frac{-114}{2401}, & b_1 & = & \frac{1}{12}, \\
a_{43} & = & \frac{-285}{2401}, & b_2 & = & \frac{-284}{719901}, \\
a_{51} & = & \frac{9488}{25289}, & b_3 & = & \frac{1349}{1008}, \\
a_{52} & = & \frac{-1742}{27951}, & b_4 & = & \frac{-1193297}{838944}, \\
a_{53} & = & \frac{-1755}{9317}, & b_5 & = & \frac{3059969}{6226272}, \\
a_{54} & = & \frac{-2548}{75867}, & b_6 & = & \frac{201719808}{473128489}, \\
a_{61} & = & \frac{1601021}{39398400}, & b_7 & = & \frac{1}{12}, \\
a_{62} & = & \frac{-49718600701}{549056102400}, & c_2 & = & \frac{-1}{2}, \\
a_{63} & = & \frac{24622320925}{1422655488}, & c_3 & = & \frac{1}{5}, \\
a_{64} & = & \frac{-201595696267}{6006767616}, & c_4 & = & \frac{1}{7}, \\
a_{65} & = & \frac{629734018859}{36989042688}, & c_5 & = & \frac{1}{11}, \\
a_{71} & = & \frac{1}{6}, & c_6 & = & \frac{173}{240}, \\
a_{72} & = & \frac{7824374}{16077789}, & c_7 & = & 1,
\end{array}
$$

Таким образом построено новое семейство 7-шаговых методов 6-го порядка.

2.7.3. Замечание. Для всех методов из найденного семейства выполняется равенство $b_1 = b_7 = 1/12$, при этом все c_i различны. Семистадийный метод 6-го порядка, найденных численно и приведенный в п.1.3.8

не удовлетворяет этим условиям. Следовательно, существуют и другие семейства решений, пока еще не обнаруженные.

Глава 3.

Методы численного нахождения методов Рунге-Кутта

3.1. Введение

Для нахождения n-шагового метода РК требуется найти $\mathbf{R}$ -матрицу A, коэффициенты которой удовлетворят некоторой системе нелинейных (полиномиальных) уравнений. В практически важных случаях ($p = 7..10$) мы получаем систему от нескольких десятков (или даже сотен) переменных и еще большего количества уравнений. В настоящее время не существует общих способов ее решения. Нахождение каждого отдельного решения является большой проблемой.

Даже просто выписать получающуюся систему уравнений довольно трудно. Эффективный способ описания этих уравнений был предложен Джоном Бутчером в ([22, 23]). Он сопоставляет каждому отмеченному дереву одно и только одно уравнение на коэффициенты матрицы A.

Аналитическое решение уравнений Бутчера оказывается чрезвычайно сложным делом и в общем виде уравнения до сих пор не решены, известны лишь отдельные решения. Поэтому имеет смысл попытаться решить их численным методом (Ньютона).

Итак, мы имеем систему нелинейных (полиномиальных) уравнений

$$\begin{gathered} f_1(x_1, \ldots, x_p) = 0, \\ \ldots \\ f_q(x_1, \ldots, x_p) = 0, \end{gathered}$$

Уравнения f_i можно выписать явно, но они оказываются черезчур громоздкими и сколько-нибудь эффективная работа с ними не возможна. Поэтому придется считать, что уравнения f_i заданы не аналитически, а лишь численно, то есть у нас нет явной формулы, а лишь некоторая процедура (рекурсивная) нахождения значений функций f_i в заданной точке. Это означает, что частные производные (df_i/dx_j) придется находить численно, что так же вносит дополнительное усложнение в расчеты.

На каждом шаге метода Ньютона вектор смещения находится из некоторой системы линейных уравнений $Dx = r$ где $D = (df_i/dx_j)$. Количество уравнений может быть как больше, так и меньше количества переменных, эти числа почти никогда не совпадают.

Такие задачи принято называть "некорректно поставленными". Для их решения, точнее говоря для нахождения "псеводорешений" применяются несколько методов, наиболее эффективным, в общем случае, является "метод регуляризации Тихонова": будем искать такой вектор x, что бы отклонение $\|Dx - r\|$ было минимальным. Это сводится к решению другой системы линейных уравнений

$$(D^t D)(x) = D^t(r),$$

где символом D^t обозначена транспонированная матрица. Так как решений может быть бесконечно много, проще всего было бы потребовать в этом случае взять наименьшее по модулю решение. Но это оказывается вычислительно довольно сложно. Эффективной заменой этому требованию оказывается следующее. Выберем некоторое достаточно малое число ε (например, 10^{-12}) и потребуем минимизации величины

$$\|Dx - r\| + \varepsilon\|x\|.$$

Эта задача приводит к решению системы линейных уравнений

$$(D^t D + \varepsilon E)(x) = D^t(r).$$

3.2. Используемые переменные и матрицы

Ввиду большого размера системы и плохой сходимости метода Ньютона, для достижения цели приходится использовать все возможности как-то сократить уравнения. Большую роль играют упрощающие предположения нескольких видов. Поэтому рассмотрим их более подробно. Для удобства выпишем матрицы и вектора, используемые в них.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & \dots & 0 & 0 & 0 \\ & & & & \dots & & & \\ a_{n,1} & a_{n,2} & a_{n,3} & & \dots & a_{n,n-1} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & \dots & b_{n-1} & b_n & 0 \end{pmatrix}.$$

$$B = \begin{pmatrix} 0 & a_{21} & a_{31} & a_{41} & \ldots & a_{n1} & b_1 \\ 0 & 0 & a_{32} & a_{42} & \ldots & a_{n2} & b_2 \\ 0 & 0 & 0 & a_{43} & \ldots & a_{n3} & b_3 \\ & & & & \ldots & & \\ 0 & 0 & 0 & & \ldots & a_{n,n-1} & b_{n-1} \\ 0 & 0 & 0 & & \ldots & 0 & b_n \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix},$$

$$Bd = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \\ 0 \end{pmatrix},$$

$$B^2 d = \begin{pmatrix} b_n a_{n1} + b_{n-1} a_{n-1,1} + \ldots + b_2 a_{21} \\ b_n a_{n2} + b_{n-1} a_{n-1,2} + \ldots + b_3 a_{32} \\ \ldots \\ b_n a_{n,n-2} + b_{n-1} a_{n-1,n-2} \\ b_n a_{n,n-1} \\ 0 \\ 0 \end{pmatrix},$$

$$c = Ae = \begin{pmatrix} 0 \\ c_2 \\ c_3 \\ c_4 \\ \ldots \\ c_n \\ c_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ a_{21} \\ a_{31} + a_{32} \\ a_{41} + a_{42} + a_{43} \\ \ldots \\ a_{n,1} + a_{n,2} + \ldots + a_{n,n-1} \\ b_1 + b_2 + \ldots + b_n \end{pmatrix},$$

$$A^2 e = Ac = \begin{pmatrix} 0 \\ 0 \\ a_{32} c_2 \\ a_{42} c_2 + a_{43} c_3 \\ a_{52} c_2 + a_{53} c_3 + a_{54} c_4 \\ \ldots \\ a_{n2} c_2 + a_{n3} c_3 + \ldots + a_{n,n-1} c_{n-1} \\ b_2 c_2 + b_2 c_3 + \ldots + b_n c_n \end{pmatrix},$$

$$Ae * Ae - 2A^2e = \begin{pmatrix} 0 \\ c_2^2 \\ c_3^2 - 2a_{32}c_2 \\ c_4^2 - 2(a_{42}c_2 + a_{43}c_3) \\ c_5^2 - 2(a_{52}c_2 + a_{53}c_3 + a_{54}c_4) \\ \ldots \\ c_n^2 - 2(a_{n2}c_2 + a_{n3}c_3 + \ldots + a_{n,n-1}c_{n-1}) \\ c_{n+1}^2 - 2(b_2c_2 + b_2c_3 + \ldots + b_nc_n) \end{pmatrix}.$$

Классическое упрощающее предположения (п.3.1): $B^2d = (e - Ae) * Bd$:

$$\begin{array}{cccc} (1) & b_na_{n1} + b_{n-1}a_{n-1,1} + \ldots + b_2a_{21} & = & b_1 \\ (2) & b_na_{n2} + b_{n-1}a_{n-1,2} + \ldots + b_3a_{32} & = & (1-c_2)b_2 \\ (3) & b_na_{n3} + b_{n-1}a_{n-1,3} + \ldots + b_4a_{43} & = & (1-c_3)b_3 \\ (4) & b_na_{n4} + b_{n-1}a_{n-1,4} + \ldots + b_5a_{54} & = & (1-c_4)b_4 \\ & & \ldots & \\ (n-2) & b_na_{n,n-2} + b_{n-1}a_{n-1,n-2} & = & (1-c_{n-2})b_{n-2} \\ (n-1) & b_na_{n,n-1} & = & (1-c_{n-1})b_{n-1} \\ (n) & 0 & = & (1-c_n)b_n \\ (n+1) & 0 & = & 0. \end{array}$$

Упрощающее предположение уровня 3: $(Ae * Ae - 2A^2e) * Bd = 0$. Это означает, что для каждого индекса соответствующая координата у одного из векторов $(Ae * Ae - 2A^2e)$ и Bd должна быть равна 0. Первая координата у первого вектора и последняя у второго всегда равны 0. Как показывают численные эксперименты, чаще всего у вектора Bd равны нулю координаты b_2, b_3 а все остальные координаты равны нулю у вектора $(Ae * Ae - 2A^2e)$. В программе, рассмотренной ниже, этот случай носит название "режим 3". С учетом этого упрощающее предположение уровня 3 выглядят так:

$$\begin{array}{cccc} (1) & 0 & = & 0 \\ (2) & b_2 & = & 0 \\ (3) & b_3 & = & 0 \\ (4) & c_4^2/2 & = & a_{42}c_2 + a_{43}c_3 \\ (5) & c_5^2/2 & = & a_{52}c_2 + a_{53}c_3 + a_{54}c_4 \\ & & \ldots & \\ (n-1) & c_{n-1}^2/2 & = & a_{n-1,2}c_2 + a_{n-1,3}c_3 + \ldots + a_{n-1,n-2}c_{n-2} \\ (n) & c_n^2/2 & = & a_{n2}c_2 + a_{n3}c_3 + \ldots + a_{n,n-1}c_{n-1} \\ (n+1) & c_{n+1}^2/2 & = & b_2c_2 + b_3c_3 + \ldots + b_nc_n \end{array}.$$

Выпишем теперь их все вместе, учитывая, что:

$$\begin{array}{c} c_n = c_{n+1} = 1 \\ b_2 = b_3 = 0 \end{array}.$$

Упрощающее предположение 1: $B^2d = (e - Ae) * Bd$:

$$\begin{array}{ccccc}
(1) & b_n a_{n1} + b_{n-1}a_{n-1,1} + \ldots + b_4 a_{41} & = & b_1 \\
(2) & b_n a_{n2} + b_{n-1}a_{n-1,2} + \ldots + b_4 a_{42} & = & 0 \\
(3) & b_n a_{n3} + b_{n-1}a_{n-1,3} + \ldots + b_4 a_{43} & = & 0 \\
(4) & b_n a_{n4} + b_{n-1}a_{n-1,4} + \ldots + b_5 a_{54} & = & (1-c_4)b_4 \\
 & & \ldots & \\
(n-2) & b_n a_{n,n-2} + b_{n-1}a_{n-1,n-2} & = & (1-c_{n-2})b_{n-2} \\
(n-1) & b_n a_{n,n-1} & = & (1-c_{n-1})b_{n-1} \\
(n) & 0 & = & 0 \\
(n+1) & 0 & = & 0
\end{array}.$$

Упрощающее предположение 2: $(Ae * Ae - 2A^2e) * Bd = 0$.

$$\begin{array}{cccc}
(4) & c_4^2/2 & = & a_{42}c_2 + a_{43}c_3 \\
(5) & c_5^2/2 & = & a_{52}c_2 + a_{53}c_3 + a_{54}c_4 \\
 & & \ldots & \\
(n-1) & c_{n-1}^2/2 & = & a_{n-1,2}c_2 + a_{n-1,3}c_3 + \ldots + a_{n-1,n-2}c_{n-2} \\
(n) & 1/2 & = & a_{n2}c_2 + a_{n3}c_3 + \ldots + a_{n,n-1}c_{n-1} \\
(n+1) & 1/2 & = & b_4c_4 + \ldots + b_nc_n
\end{array}.$$

3.3. Применение упрощающих предположений для сокращения системы уравнений

В п.3.2 было показано, как с помощью одного упрощающего предположения сократить количество переменных и уравнений. В программе, рассмотренной ниже, этот случай носит название "режим 2". Посмотрим теперь, что нам может дать использование сразу двух упрощающих предположений.

В общем случае неизвестными являются $n(n+1)/2$ ненулевых коэффициентов матрицы A. Используя упрощающие предположения, их количество можно сократить. В качестве неизвестных выберем следующие коэффициенты матрицы $\tilde{A}$:

$$\begin{pmatrix}
a_{21} & & & & & & \\
a_{31} & a_{32} & & & & & \\
 & a_{42} & a_{43} & & & & \\
 & a_{52} & a_{53} & a_{54} & & & \\
 & \ldots & & \ldots & & & \\
 & a_{n-1,2} & a_{n-1,3} & & \ldots & a_{n-1,n-2} & \\
 & & & a_{n,4} & \ldots & a_{n,n-2} & a_{n,n-1}
\end{pmatrix},$$

то есть из первого столбца матрицы $\tilde{A}$ мы оставляем только коэффициенты a_{21}, a_{31}, а из последней строки выбрасываем a_{n1}, a_{n2}, a_{n3}.

Используя упрощающие предположения, все остальные коэффициенты матрицы A можно выразить через выбранные.

Шаг 1. Из упрощающего предположения 2, строки с 4 по $(n-1)$, последовательно найдем $c_4, \ldots, c_{n-1}$. Так как уже известно, что $c_n = 1$, то в строке n упрощающего предположения 2

$$1/2 = a_{n2}c_2 + a_{n3}c_3 + \ldots + a_{n,n-1}c_{n-1}$$

не найденными остались a_{n2} и a_{n3}.

Рассмотрим теперь упрощающее предположение 1. Из строки $(n-1)$ получим:

$$b_{n-1} = \alpha_{n-1}b_n$$

где

$$\alpha_{n-1} = \frac{a_{n,n-1}}{1 - c_{n-1}}.$$

Из строки $(n-2)$ получаем:

$$b_{n-2} = \alpha_{n-2}b_n$$

где

$$\alpha_{n-2} = \frac{a_{n,n-2} + \alpha_{n-1}a_{n-1,n-2}}{1 - c_{n-2}}.$$

Далее, аналогично $b_k = \alpha_k b_n$ при $k = n-2, \ldots, 4$ для некоторых α_k, явно выраженных через уже найденные коэффициенты матрицы A. Теперь воспользуемся строкой $(n+1)$ упрощающего предположения 2:

$$1/2 = b_4c_4 + \ldots + b_nc_n$$

или

$$1/2 = (\alpha_4c_4 + \ldots + \alpha_{n-1}c_{n-1} + 1)b_n.$$

Отсюда мы можем найти b_n и, следовательно, $b_{n-1}, \ldots, b_4$. Как уже известно, $b_2 = b_3 = 0$. Коэффициент же b_1 можно найти из уравнения $b_1 + \ldots + b_n = 1$. Таким образом, у нас остались не найдены три коэффициента:

$$a_{n1}, a_{n2}, a_{n3}$$

Они получаются из строк 1,2,3 упрощающего предположения 1:

$$\begin{array}{lrcl} (1) & b_na_{n1} + b_{n-1}a_{n-1,1} + \ldots + b_4a_{41} & = & b_1 \\ (2) & b_na_{n2} + b_{n-1}a_{n-1,2} + \ldots + b_4a_{42} & = & 0 \\ (3) & b_na_{n3} + b_{n-1}a_{n-1,3} + \ldots + b_4a_{43} & = & 0 \end{array}.$$

Итак, все коэффициенты матрицы A выражены через выбранные. Помимо того, осталось еще два соотношения, которые не будут выполняться автоматически:

$$1/2 = a_{n2}c_2 + a_{n3}c_3 + \ldots + a_{n,n-1}c_{n-1}$$

и

$$1 = a_{n1} + a_{n2} + a_{n3} + \ldots + a_{n,n-1}$$

3.4. Описание программы

Все эти идеи были реализованы с помощью программы, описанной ниже. Она была написана на языке Паскаль для IBM PC (Turbo Pascal, TMT Pascal, Delphi). По смыслу задачи, стандартной памяти 640К байт должно быть вполне достаточно. Поэтому ее можно написать так, что бы она работала на любом IBM-совместимом компьютере, вплоть до XT. Тем не менее, ввиду большого объема вычислений, все реальные результаты были получены на процессоре класса P333 Mhz.

Очень просто было определится с типом действительных чисел. В Паскале их имеется несколько длиной от 4 до 10 байтов. Программе приходится иметь дело с сильно вырожденными матрицами. Точнее говоря, надо обращать матрицы близкие к сильно вырожденным. При этом следует ожидать значительной потери точности. Поэтому действительные числа должны иметь максимальную длину, то есть быть типа extended, длина 10 байтов, 64 двоичных знака после запятой, около 19 десятичных знаков после запятой. Кроме того, этот тип естественен для данного процессора, и обрабатывается им с максимальной скоростью. Таким образом, мы будем вести вычисления с 19 знаками после запятой. Метод будем считать найденным, если суммарная невязка для системы уравнений окажется меньше 10^{-18}.

Программу можно разбить на несколько частей.

3.4.1. Вектора и матрицы, unit Linear3

Этот модуль должен уметь обрабатывать в стандартной памяти 640К байт матрицы размером до 10 - 30 тысяч десятибайтовых чисел. Если хранить все элементы матрицы в одном массиве, то ее размер ограничен 6400 числами (матрица 80*80), что нежелательно. Поэтому $n*m$-матрица должна быть массивом из n указателей на вектора длины m.

На сегодняшний день имеется большое количество готовых модулей реализующих задачи линейной алгебры. Тем не менее было решено написать свой модуль. Это было сделано по следующим соображениям.

- Во-первых, задача эта не очень трудоемкая. Время на написание модуля сопоставимо со временем на изучение уже готового.
- Во-вторых, предъявленные выше требования значительно сузили круг поисков.
- В-третьих, некоторые специфические (и наиболее тяжелые) алгоритмы все равно пришлось бы писать заново.

Вместе с тем следует отметить, что при нахождении размерности многообразия решения для приведения квадратичной формы к диагональному виду ортогональными преобразованиями был применен готовый алгоритм.

3.4.2. Нахождение уравнений Бутчера, unit FUNCT_RK

Это наиболее сложная часть программы. В полном виде, без упрощающих предположений, каждому помеченному дереву веса $\leq p$ соответствует одно уравнение. Следовательно, программа должна иметь дело с помеченными деревьями. Для их описания и хранения был применен такой метод.

Все деревья пронумеровываем $t_1, t_2, \ldots$. Дереву из одной вершины (веса 0) присваиваем номер 1. Дереву αt_1 присваиваем номер 2. Любое дерево t можно представить в виде

$$t = \alpha(t_{i_1}) \cdot \ldots \cdot \alpha(t_{i_s}), \quad s \geq 1,$$

где деревья $t_{i_1}), \ldots, t_{i_s}$ имеют меньший вес, точнее говоря выполнено равенство

$$w(t_{i_1}) + \ldots + w(t_{i_s}) + s = w(t)$$

Поэтому дерево t однозначно описывается номерами деревьев $i_1, \ldots, i_s$, то есть для записи дерева t требуется массив из чисел $i_1, \ldots, i_s$.

Исходя из этого, а так же из удобства записи, было принято решение, хранить деревья в виде строки символов $string[8]$. Такой подход позволяет хранить деревья до веса 8 включительно, что позволяет искать методы Рунге-Кутта максимум порядка 9. Для поиска методов более высоких порядков способ хранения деревьев придется изменить.

В программе деревья описаны так:

```
const trees: array[3..486] of string[8] =
   {  3 }  ( #2,
   {  4 }    #1#1,
        ...
   {486 }    #1#1#1#1#1#1#1#1 );
```

Главная процедура этого модуля - $RK\_Eq(x, y)$, которая по заданному вектору переменных x возвращает вектор y со значениями уравнений Бутчера.

Предлагается три режима работы - полный ($regim = 1$), с использованием упрощающего предположения ($regim = 2$) и двух упрощающих предположений ($regim = 3$). Допустимый порядок для искомого метода Рунге-Кутта - от 4 до 9. Количество уравнений в зависимости от режима и порядка указано в следующей таблице.

p	$r = 1$	$r = 2$	$r = 3$
4	8	3	–
5	17	8	6
6	37	19	13
7	85	47	32
8	200	114	79
9	486	285	202

Допустимое количество шагов - от 4 до 17. Количество переменных в зависимости от режима и порядка указано в следующей таблице.

n	$r = 1$	$r = 2$	$r = 3$
4	10	5	–
5	15	9	6
6	21	14	10
7	28	20	15
8	36	27	21
9	45	35	28
10	55	44	36
11	66	54	45
12	78	65	55
13	91	77	66
14	105	90	78
15	120	104	91
16	136	119	105
17	153	135	120

3.4.3. Реализация метода Ньютона, unit Newton

Это сравнительно несложный модуль, наименее очевидная часть которого - определение момента завершения работы. Поскольку поведение метода Ньютона плохо предсказуемо, приходится отслеживать различные возможные причины завершения работы:

- исчерпан максимум шагов,
- одна из переменных стала черезчур велика,
- нажата клавиша Esc,
- черезчур большая производная
- черезчур большая невязка
- за много шагов невязка почти не уменьшилась.

Здесь стоит пояснить последнюю причину завершения работы. При поиске методов порядка 7 и выше, типичная ситуация такова: невязка быстро (за несколько десятков шагов) убывает до некоторого небольшого значения (порядка $10^{-7} - 10^{-9}$), после этого практически застывает на месте. Наиболее приемлемый выход из такой ситуации - начать вычисления заново, с новыми начальными (случайными) данными. Более точно, если за 50 шагов невязка не уменьшилась хотя бы на 5%, метод Ньютона завершается.

3.4.4. Основная программа, program rk

Начальные данные могут быть как случайными, так и взятыми из файла. Результаты работы также записываются в текстовый файл в виде, пригодном для дальнейшей обработки: их можно использовать в качестве начальных данных для дальнейших расчетов.

Для 16-шаговых методов 9 порядка (119 переменных, 285 уравнений) время расчета на процессоре P150 составляет около 5 секунд на составление матрицы частных производных и около 15 секунд на решение системы линейных уравнений. Всего на один шаг метода Ньютона уходит около 20 секунд.

Основная проблема состоит в том, что система уравнений сильно вырождена. Результат этого можно увидеть уже на примере поведения метода Ньютона для простой системы уравнений

$$\begin{aligned} f_1(x,y) &= x^2 - y^3 = 0, \\ f_2(x,y) &= x^3 - y^4 = 0, \end{aligned}$$

в окрестности решения $(x,y) = (0,0)$. Нетрудно проверить, что сходимость метода к 0 будет крайне медленной.

Аналогичные проблемы возникают и при рассмотрении системы уравнений Бутчера. Для нахождения метода четвертого порядка требуется 4-8 ньютоновских итераций, для метода 5 порядка - от 10 до 100. Для методов 7, 8 и 9 порядка требуемое количество итераций оказывается равным многим тысячам. При этом, далеко не каждое начальное случайное приближение приводит к результату. Тем не менее, на процессоре P333 в течении суток можно найти несколько методов 8 порядка.

Следует отметить, что основным параметром компьютера, влияющим на время расчета, является скорость работы процессора. Обращения к диску происходят достаточно редко. Для работы программы памяти 640К байт оказывается вполне достаточно, по крайней мере для методов порядка не более 9.

Множество методов Рунге-Кутта некоторого порядка с фиксированным количеством шагов образует некоторое алгебраическое многообразие в пространстве нижнетреугольных матриц. Большой интерес представляет нахождение размерности этого многообразия. Первое, что приходит в голову - это найти ранг матрицы из частных производных (df_i/dx_j) уравнений Бутчера. Но, к сожалению, система уравнений сильно вырождена, по крайней мере в окрестностях решений и ранг этой матрицы связан с размерностью многообразия довольно непонятным образом. Поэтому для нахождения размерности многообразия можно поступить так.

К некоторому найденному решению прибавим маленький случайный вектор, длины, например, 10^{-10}. Применим к полученному вектору метод Ньютона. Как показывают численные эксперименты, метод Ньютона в этом случае сходится очень быстро, за 3-6 шагов. Таким образом, мы можем найти в окрестностях выбранного решения несколько сотен других решений v_i за приемлемое время (несколько часов). Имея эти точки мы уже сможем определить размерность полученного многообразия. Для этого рассмотрим неотрицательную квадратичную форму

$$f(x) = \sum_i (x, v_i)^2,$$

где (x, y) - скалярное произведение векторов. Если вектора v_i содержатся в k-мерном подпространстве, то ранг квадратичной формы f не превышает k. В нашем случае, если вектора v_i содержатся в k-мерном алгебраическом многообразии, то они "почти" содержатся в соответствующем касательном k-мерном пространстве.

Пусть $\lambda_1 \geq \ldots \geq \lambda_s$ - последовательность собственных значений квадратичной формы $f(x)$ и $\mu_i = \lambda_i/\lambda_1$ - та же последовательность в нормализованном виде ($\mu_1 = 1$). Если первые k чисел из последовательности μ_i имеют порядок 1, а все последующие значительно (на несколько порядков) меньше, то можно считать, что вектора x_i принадлежат k-мерному многообразию. Если же такого номера k нет, то есть числа μ_i более-менее равномерно убывают, то вопрос о размерности многообразия, на которое ложатся точки x_i требует дополнительного изучения([6, 7]).

Глава 4.

Методика проверки точности и сравнения методов

4.1. Тестовая задача

С помощью программы RK.EXE мы можем получить большое количество матриц, удовлетворяющих уравнениям Бутчера, то есть новых методов Рунге-Кутта различных порядков и с различным количеством стадий. Однако не все эти методы одинаково пригодны для решения практических задач. Сравним их эффективность путем решения одной конкретной задачи .

Для иллюстрации работы методов, мы выбрали пример из небесной механики — ограниченнную задачу трех тел. Рассмотриваются два тела с массами $\mu' = 1-\mu$ и μ, участвующие в совместном круговом движении в некоторой плоскости, и движущееся вблизи них в той же плоскости третье тело пренебрежимо малой массы. Уравнения имеют вид:

$$\begin{aligned} y_1'' &= y_1 + 2 \cdot y_2' - \mu \cdot \tfrac{y_1 - \mu'}{D_2}, \\ y_2'' &= y_2 - 2 \cdot y_1' - \mu' \cdot \tfrac{y_2}{D_2}, \end{aligned}$$

где

$$\begin{aligned} D_1 &= ((y_1+\mu)^2 + y_2^2)^3/2, \\ D_2 &= ((y_1-\mu')^2 + y_2^2)^3/2, \\ \mu &= 0.012277471, \\ \mu' &= 1-\mu. \end{aligned}$$

Как известно, аналитического решения такая задача в общем случае не имеет, однако ее можно решить численно. Можно подобрать константы и начальные условия так, чтобы решение получилось периодическим :

$$\begin{aligned} y_1(0) &= 0.994, \\ y_1'(0) &= 0, \\ y_2(0) &= 0, \\ y_2'(0) &= -2.00158510637908252240537862224. \end{aligned}$$

В данной задаче период оказывается равным

$$X_{end} = 17.0652165601579625588917206249$$

Такие решения очень удобны для проверки различных методов численного решения и сравнения получаемой погрешности.

4.2. Алгоритм проверки

Большое количество различных методов Рунге-Кутты можно найти в литературе ([22, 23, 24, 25, 8]) их можно также получать с помощь программы RK.EXE, описанной в предыдущей главе. Каждый метод Рунге-Кутты описывается своей матрицей. В рассматриваемой программе эта матрица записывается в текстовый файл в следующем формате (пример для для 4-х стадийного метода):

```
Matrix; Time: 2004-03-14 23:17:57 steps=    11
 4 ; #шагов
 4 ; порядок
3.105E-0020 ; отклонение
 0.49401424249884009; a(2,1); c[2]=0.49401424249884009
 0.00023465349947433; a(3,1); c[3]=0.50583695332842301
 0.50560229982894868; a(3,2)
 0.02351896073182491; a(4,1); c[4]=1.00000000000000000
-0.02470061954949237; a(4,2)
 1.00118165881766746; a(4,3)
 0.16662006150041289; b(  1)
 0.32918507822890896; b(  2)
 0.33757477103447063; b(  3)
 0.16662008923620752; b(  4)
```

В первой строке записывается ключевое слово Matrix, служащее признаком файла нужного типа, затем в той же строке дата и время получения матрицы и количество шагов метода Ньютона для системы полиномиальных уравннеий, за которое метод был получен.

В следующих строках указывается количество шагов (стадий) найденной матрицы и порядок метода.

Затем идет погрешность с которой найденн метод: длина вектора в правой части уравнний Бутчера. Для точного решения это должен быть 0. Значение $1e-18$ и меньше можно считать практически точными.

Далее записаны коэффициенты матрицы метода, лежащие ниже главной диагонали (остальные коэффициенты всегда равны 0) - по одному коэффициенту в строке. Кроме того, в некоторых строках записаны дополнительно коэффициенты c_i. На самом деле они избыточны, $c_i = a_{i1}+\ldots+a_{i,i-1}$ и нужны лишь для удобства чтения человеком.

Замечание. Метод 4-го порядка может быть получен в теченни нескольких секунд, для полечения методов же высоких порядков (7-8) может потребоваться несколько часов. Обычной практикой является запуск компьютера на ночь, на выходные, на каникулы. Поэтому время получения матрицы - существенный реквизит.

На выходе из программы RK.EXE мы получаем один текстовый файл, содержащий все полученные матрицы – иногда по нескольку тысяч или даже десятков тысяч.

Для сравнения эффективности различных методов была написана небольшая программа (RK_USE.EXE), которая по данной матрице Бутчера, записанной в текстовый файл, строит соответствующий метод Рунге-Кутта, затем решает описанную выше задачу с таким количеством шагов на интервале $0 \ldots X_{end}$, чтобы количество вычислений правой части системы дифференциальных уравнений равнялось заданному. Например, если мы хотим получить 1000000 вычислений функции для 5-стадийного метода, то весь промежуток интегрирования надо разбить на $1000000/5 = 200000$ шагов, величина шага будет равна $X_{end}/200000$. При таком подходе к выбору величины шага мы получим более объективную оценку качества метода Рунге-Кутта.

Конечно, одного примера еще не достаточно, что бы полностью оценить качество метода. Тем не менее, некоторую их сравнительную характеристику мы получаем. Как показывают дополнительные эксперименты, соотношение между достигамой разными методами точности оказывается примерно таким же и для многих других задач. То есть найденная нами оценка качества является вполне надежной.

Так как количество полученных методов исчисляется в некоторых случаях тысячами (и даже десятками тысяч), полученные данные нуждаются в дополнительной обработке и сортировке.

Для разделения исходного файла на отдельные файлы, каждый из которых содержит лишь одну матрицу метода Рунге-Кутта был написан скрипт (на языке JavaScript).

```
// файл %1 со многими таблицами Бутчера режем на кусочки
// с именами _0001.res, _0002.res, ...
// Пример запуска:
// Cscript /nologo disting.js RK_01.RES
//
  var args =WScript.Arguments;
  if( args.length != 1 ) {
    WScript.Echo("Должен быть один аргумент" );
    Exit;
  }
  var infile = args(0);
  WScript.Echo("infile = " + infile);
```

```
  var FSO = new ActiveXObject("Scripting.FileSystemObject");

//----------------------------------------------
function outname(k) { // имя выходного файла
  var s = "000"+k;
  s ="_"+s.substr( s.length-4 )+".res";
  return s;
}

//----------------------------------------------
function showErrorInfo(e) {
  WScript.Echo(e);
  WScript.Echo("Источник ошибки: ", (e.number >> 16) & 0x1FFF);
  WScript.Echo("Номер ошибки: ", e.number & 0xFFFF);
  WScript.Echo("Описание ошибки: ", e.description);
}
//----------------------------------------------

  var f, fo;
  var ind;
  try{
    f = FSO.OpenTextFile(infile);
  } catch(e) {
    WScript.Echo( "File "+infile+" not open" );
  }
  ind = 1; // номер выходного файла
  try{
    fo= FSO.CreateTextFile(outname(ind));
    while (!f.AtEndOfStream) {
      s = f.ReadLine();
      if( s.substr(0,5)=="-----" ) { // файл закончен
        fo.close();
        ind++;
        fo= FSO.CreateTextFile(outname(ind));
        continue;
      }
      fo.WriteLine(s);
    }
    fo.Close();
    f.Close();
  } catch(e) {
    WScript.Echo( "Error3" );
    showErrorInfo(e);
  }
```

После его применения к файлу, содержащему различные матрицы методов Рунге-Кутта, в текущем каталоге оказываются файлы с именами _00000.*res*, _00001.*res*, ..., _*nnnnn.res*, в каждом из которых записана лишь одна матрица Рунге-Кутта.

Для применения прогаммы RK_use.exe ко всем матрицам с расширением *.res* в текущем каталоге, можно воспользоватся командой *for*:

```
for %%a in (_????.res) do RK_use.exe %%a 1000000 prot.txt
```

Выполнение этой команды может занять довольно много времени – вплоть до нескольких часов. После ее выполнения в файле протокола *prot.txt* окажется для каждого из обработанных файлов – по одной строке, состоящей из имени обработанного файла и величины погрешности тестовой задачи, полученной на заданной матрице, например:

```
RK:_00000.res;N=13;p=7;steps= 76923;cnt= 999999;err=5.650E-0006
RK:_00001.res;N= 6;p=5;steps=166666;cnt= 999996;err=1.826E+0000
RK:_00002.res;N= 4;p=4;steps=250000;cnt=1000000;err=4.134E-0003
RK:_00003.res;N= 6;p=4;steps=166666;cnt= 999996;err=2.238E+0000
RK:_00004.res;N= 5;p=4;steps=200000;cnt=1000000;err=1.546E+0000
RK:_00005.res;N= 5;p=5;steps=200000;cnt=1000000;err=2.050E-0004
RK:_00006.res;N= 5;p=4;steps=200000;cnt=1000000;err=1.599E-0005
RK:_00007.res;N=13;p=7;steps= 76923;cnt= 999999;err=4.703E-0008
RK:_00008.res;N=13;p=7;steps= 76923;cnt= 999999;err=4.703E-0008
RK:_00009.res;N= 4;p=4;steps=250000;cnt=1000000;err=1.249E-0001
RK:_00010.res;N= 7;p=6;steps=142857;cnt= 999999;err=1.265E-0002
RK:_00011.res;N= 5;p=5;steps=200000;cnt=1000000;err=1.162E-0003
RK:_00012.res;N= 5;p=5;steps=200000;cnt=1000000;err=2.603E-0003
RK:_00013.res;N= 7;p=5;steps=142857;cnt= 999999;err=3.980E-0008
```

Если методов было несколько десятков, то среди них легко выбрать методы с наимельншей погрешностью. Если же методов было несколько тысяч, то прямой их отбор будет уже не совсем простой задачей. Простая сотрировка файла не даст требуемого результата. Поэтому приходится сортировать в два этапа:

погрешность и выбрать соответствующая метода. Мы сортируем с таким образом : Сначала сортируем файл по 78 позиции - то есть по порядку величины err.

```
sort/+78 prot.txt> prot1.txt
```

Результат сортировки в данном случае будет таким:

```
RK:_00001.res;N= 6;p=5;steps=166666;cnt= 999996;err=1.826E+0000
RK:_00003.res;N= 6;p=4;steps=166666;cnt= 999996;err=2.238E+0000
RK:_00004.res;N= 5;p=4;steps=200000;cnt=1000000;err=1.546E+0000
RK:_00009.res;N= 4;p=4;steps=250000;cnt=1000000;err=1.249E-0001
```

```
RK:_00010.res;N= 7;p=6;steps=142857;cnt= 999999;err=1.265E-0002
RK:_00011.res;N= 5;p=5;steps=200000;cnt=1000000;err=1.162E-0003
RK:_00012.res;N= 5;p=5;steps=200000;cnt=1000000;err=2.603E-0003
RK:_00002.res;N= 4;p=4;steps=250000;cnt=1000000;err=4.134E-0003
RK:_00005.res;N= 5;p=5;steps=200000;cnt=1000000;err=2.050E-0004
RK:_00006.res;N= 5;p=4;steps=200000;cnt=1000000;err=1.599E-0005
RK:_00000.res;N=13;p=7;steps= 76923;cnt= 999999;err=5.650E-0006
RK:_00007.res;N=13;p=7;steps= 76923;cnt= 999999;err=4.703E-0008
RK:_00008.res;N=13;p=7;steps= 76923;cnt= 999999;err=5.703E-0008
RK:_00013.res;N= 7;p=5;steps=142857;cnt= 999999;err=3.980E-0008
```

После этого надо оставить только методы с найменьшим порядком погрешности:

```
RK:_00007.res;N=13;p=7;steps= 76923;cnt=999999;err=4.703E-0008
RK:_00008.res;N=13;p=7;steps= 76923;cnt=999999;err=5.703E-0008
RK:_00013.res;N= 7;p=5;steps=142857;cnt=999999;err=3.980E-0008
```

и еще раз отсортировать, но теперь еже по 72 позиции – то есть по величине погрешности:

```
sort/+72 prot1.txt> prot2.txt$
```

Полученный в файле *prot2.txt* результат уже удовлетворяет всем требованиям:

```
RK:_00013.res;N= 7;p=5;steps=142857;cnt=999999;err=3.980E-0008
RK:_00007.res;N=13;p=7;steps= 76923;cnt=999999;err=4.703E-0008
RK:_00008.res;N=13;p=7;steps= 76923;cnt=999999;err=5.703E-0008
```

4.3. Описание программы RK_USE.

Программа RK_USE.EXE реализована с помощью системы программирования "Delphi", является так называемым "консольным приложением", то есть не имеет графического интерфейса, запускается из командной строки, полученные результаты записывает в текстовый файл. На экран при этом не выводится ничего существенного, лишь индикатор выполнения, что бы можно было оценить требуемое время для обработки. Опишем основные функции, реализованные в программе и их назначение.

4.3.1.Procedure ErrProc . Процедура служит для сообщения об ошибке. Завершает работу программы.

4.3.2.Procedure Help. Выводит на экран краткое сообщение о программе и способе ее запуска. Завершает работу программы.

4.3.3.procedure read_RK. Чтение матрицы метода из заданного файла. При чтении контролируется допустимость и осмысленность параметров, чтобы предотвратить исползование ошибочного файла с матрицей метода.

4.3.4.procedure Power32. Функция вычисляет $x^{3/2}$.

4.3.5.procedure F_01. Нахождение значений функции, задающей дифференциальное уравнение.

$$\begin{aligned} y_1'' &= y_1 + 2 \cdot y_2' - \mu \cdot \tfrac{y_1 - \mu'}{D_2}, \\ y_2'' &= y_2 - 2 \cdot y_1' - \mu' \cdot \tfrac{y_2}{D_2}, \end{aligned}$$

где

$$\begin{aligned} D_1 &= ((y_1 + \mu)^2 + y_2^2)^3/2, \\ D_2 &= ((y_1 - \mu')^2 + y_2^2)^3/2, \\ \mu &= 0.012277471, \\ \mu' &= 1 - \mu. \end{aligned}$$

```
r[0]:=y[2];
r[1]:=y[3];
r[2]:=y[0]+2*y[3]-mu1*(y[0]+mu)/d1-mu*(y[0]-mu1)/d2;
r[3]:=y[1]-2*y[2]-mu1*y[1]/d1-mu*y[1]/d2;
```

4.3.6.procedure cop_vect - кописрование векторов.

4.3.7.procedure add_vect - сложение векторов: к первому вектору прибавляет второй, умноженный на константу.

4.3.8.procedure RK_step - один шаг метода Рунге-Кутта с данной матрицей.

```
// один шаг метода РК с данной матрицей RK
// f  - решаемое уравнение,
// x0 - начальное значение x,
// h  - шаг
// y  - вектор значений, в нем же возвращается следующий вектор
procedure RK_step
    ( f:VECTOR_FUNC;
      x0, h: extended;
      var y: array of extended );
```

Глава 5.

Сравнение различных методов Рунге-Кутта

5.0.9.Метод Эйлера. Простейший метод решения начальной задачи

$$y' = f(x, y), \quad y(x_0) = y_0$$

был описан Эйлером (1768) в его "Интегральном исчислении" и является, фактически, методом Рунге-Кутта порядка 1. Глобальная погрешность метода имеет вид $c \cdot h$, где с – постоянная, зависящая от задачи, и h – длина шага.

5.0.10.Определение. Метод Рунге-Кутта имеет порядок p если для достаточно гладких задач вида (1.1)

$$\|y(x_0 + h) - y_1\| \leq kh^{p+1}, \|y(x_0 + h) - y_1\| \leq kh^{p+1},$$

т.е. если ряды Тейлора для точного решения $y(x_0 + h)$ и для y_1 совпадают до члена h^p включительно.

Чтобы найти различные методы РК с помощью программы RK.EXE были использованы два режима работы - полный т.е без упрощаюшего предположения (regim=1) и сокращенный, с использованием основного упрощающего предположения (regim=2).

5.1. Методы порядка 4.

5.1.1.Обозначения. Классическими принято называть 4-х стадийные методы 4 порядка. Именно они и были найдены в оригинальных работах Хойна, Рунге и Кутта. Эти методы лежат в основе большинства стандартных программ численного решения задачи Коши на ЭВМ. Для того чтобы зафиксировать обозначения рассмотрим их более подробно.

Матрица 4-х стадийного метода выглядит следующем образом:

$$A_{44} = \begin{pmatrix} 0 & 0 & 0 & 0 & \\ a_{21} & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & 0 \end{pmatrix}.$$

Она содержит 10 неизвестных.

Для того, чтобы такой метод имел порядок 4, должны быть выполнены следующие уравнения:

$$\begin{array}{llcllcl}
1) & \{b\} & = & 1, & b_1+b_2+b_3+b_4 & = & 1, \\
2) & \{b*\tilde{A}e\} & = & 1/2, & b_2c_2+b_3c_3+b_4c_4 & = & 1/2, \\
3) & \{b*\tilde{A}^2e\} & = & 1/6, & b_3a_{32}c_2+b_4(a_{42}c_2+c_{43}c_3) & = & 1/3, \\
4) & \{b*\tilde{A}e*\tilde{A}e\} & = & 1/3, & b_2c_2^2+b_3c_3^2+b_4c_4^2 & = & 1/3, \\
5) & \{b*\tilde{A}^3e\} & = & 1/24, & b_4a_{21}a_{32}a_{43} & = & 1/24, \\
6) & \{b*\tilde{A}^2e*\tilde{A}e\} & = & 1/8, & b_3a_{32}c_2c_3+b_4(a_{42}c_2+a_{43}c_3c_4) & = & 1/8, \\
7) & \{b*\tilde{A}e*\tilde{A}e*\tilde{A}e\} & = & 1/4, & b_2c_2^3+b_3c_3^3+b_4c_4^3 & = & 1/4, \\
8) & \{b*\tilde{A}(\tilde{A}e*\tilde{A}e)\} & = & 1/12. & b_3a_{32}c_2^2+b_4(a_{42}c_2^2+c_{43}c_3^2) & = & 1/12,
\end{array}$$

где, как обычно, через c_i обозначена сумма чисел в i-й строке матрицы A. Как известно [...], эта система уравнений имеет двухпараметрическое семейство решений:

$$\begin{aligned}
b_2 &= \tfrac{2\cdot v-1}{12\cdot u\cdot(1-u)\cdot(v-u)}, \\
b_3 &= \tfrac{1-2\cdot u}{12\cdot v\cdot(1-v)\cdot(v-u)}, \\
b_4 &= \tfrac{6\cdot u\cdot v+3-4\cdot u-4\cdot v}{12\cdot(1-u)\cdot(1-v)}, \\
b_1 &= 1-b_2-b_3-b_4, \\
c_2 &= u, \\
c_3 &= v, \\
c_4 &= 1.
\end{aligned}$$

Где u и v свободные переменные. Остальных член через них находятся.

На практике чаще всего используются следующие два метода: "классический метод РК" и "правило 3/8" (см. ниже).

Для практических вычисленй не все получающиеся методы одинаково пригодны. Для сравнения их между собой, решим с помощью различных описанных выше методов одну и ту же задачу (...). Так как программа должна произвести 1000000 вычислений правой части системы уравнений, для четырехстадийных методов интервал интегрирования должен быть разбит на 1000000/4 = 250000 шагов, для пятистадийных методов интервал интегрирования должен быть разбит на 1000000/5 = 200000 шагов, для шестистадийных методов интервал интегрирования должен быть разбит на 1000000/6 = 90909 шагов, для семистадийных методов интервал интегрирования должен быть разбит на 1000000/7 = 142857 шагов, для восимистадийных методов интервал интегрирования должен быть разбит на $1000000/8 = 125000$ шагов, для девятистадийных методов интервал интегрирования должен быть разбит на $1000000/9 = 111111$ шагов, для десятистадийных методов интервал интегрирования должен быть разбит на $1000000/10 = 100000$ шагов, для одйннадцатистадийных методов интервал интегрирования должен быть разбит на $1000000/11 = 90909$ шагов, для двенадцатистадийных методов интервал интегрирования должен быть разбит на $1000000/12 = 83333$ шага и для тринадцатистадийных методов интервал интегрирования должен быть разбит на $1000000/13 = 76923$ шага. Сравним погрешность, получающуюся при использовании каждого из методов.

5.1.2."Классический метод"

$$\begin{array}{rcl} a(2,1) &=& 1/3, \\ a(3,1) &=& -1/3, \\ a(3,2) &=& 1, \\ a(4,1) &=& 1, \\ a(4,2) &=& -1, \\ a(4,3) &=& 1, \end{array} \qquad \begin{array}{rcl} b(1) &=& 1/8, \\ b(2) &=& 3/8, \\ b(3) &=& 3/8, \\ b(4) &=& 1/8, \\ c[2] &=& 1/3, \\ c[3] &=& 2/3, \\ c[4] &=& 1, \end{array}$$

После прохождения 250000 шагов, погрешность решения на последнем шаге составила $1.249 \cdot 10^{-1}$.

5.1.3."Правило 3/8"

$$\begin{array}{rcl} a(2,1) &=& 1/2, \\ a(3,1) &=& 0, \\ a(3,2) &=& 1/2, \\ a(4,1) &=& 0, \\ a(4,2) &=& 0, \\ a(4,3) &=& 1, \end{array} \qquad \begin{array}{rcl} b(1) &=& 1/6, \\ b(2) &=& 2/6, \\ b(3) &=& 2/6, \\ b(4) &=& 1/6, \\ c[2] &=& 1/2, \\ c[3] &=& 1/2, \\ c[4] &=& 1, \end{array}$$

После прохождения 250000 шагов, погрешность решения на последнем шаге составила $4.134 \cdot 10^{-3}$. Таким образом, для рассматриваемой задачи, "правило 3/8" дает погрешность на 2 порядка меньше, чем "классический метод". Одного примера для практических рекомендаций еще недостаточно, но тем не менее, можно ожидать, что и в других случаях "правило 3/8" окажется предпочтительнее.

5.1.4.Методы, полученные программно

Для поиска методов 4-го порядка программе RK.EXE приходится решать систему 8 уравнений от 10 переменный (режим без упрощающих предположений). При использовании упрощающего предоположения количество переменных сокращается до 5, а количество уравнений - до 3. Такая система решается на современных компьютерах очень быстро.

Для всех полученных методов с помощью той же программы RK_use.exe была найдена погрешность решения задачи за 250000 шагов. Погрешность оказалась очень различной.

Из общего количества методов, у более 100 погрешность больше 1, у 1378 методов - от 1 до 10^{-1}, у 2340 методов - от 10^{-1} до 10^{-2}, у 1776 методов - от 10^{-2} до 10^{-3}, и у 199 метода - от 10^{-3} до 10^{-4}. Из них, мы заметили что у следующего матрица самая маленкая погрешность (а именно, $1.913 \cdot 10^{-4}$) то есть лучший метод:

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.49616200439373783, & b(1) & = & 0.16664801680080635,\\
a(3,1) & = & 0.00900868259165983, & b(2) & = & 0.32465610548977035,\\
a(3,2) & = & 0.49463414528294008, & b(3) & = & 0.34204784635407608,\\
a(4,1) & = & 0.01490922072805321, & b(4) & = & 0.16664803135534722,\\
a(4,2) & = & -0.03369054305805685, & c[2] & = & 0.49616200439373783,\\
a(4,3) & = & 1.01878132233000364, & c[3] & = & 0.50364282787459991,\\
 & & & c[4] & = & 1,
\end{array}
$$

В режиме 2 после вычислении погрешности , очень быстро тоже получили более 5000 методов(3 уравнения от 5 переменных).

Из общего количества методов, у всех методов - от 10^{-1} до 10^{-2}. Мы выбрали самый хороший метод из них с погрешности $2.21 \cdot 10^{-2}$. Выпишем матрицу этого метода:

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.46428624941858638, & b(2) & = & 0.31626356648374693,\\
a(3,1) & = & -0.01051299167535561, & b(3) & = & 0.35345741978446946,\\
a(3,2) & = & 0.54243572258526219, & b(4) & = & 0.16515113889739116,\\
a(4,1) & = & 0.13325336284461003, & c[2] & = & 0.46428624941858638, .\\
a(4,2) & = & -0.13503503324009472, & c[3] & = & 0.53192273090990659,\\
a(4,3) & = & 1.00178167039548469, & c[4] & = & 1,\\
b(1) & = & 0.16512787483439246, & & &
\end{array}
$$

5.1.5. Пятистадийные методы 4-го порядка

Обычно считается, что чем меньше стадий у метода при фиксированном порядке – тем лучше. Однако это не значит, что метод с большим количеством стадий обязательно окажется хуже метода с меньшим количеством. Чтобы убедиться в этом, рассмотрим методы 4-го порядка, но с большим количеством стадий — 5 или даже 6. Начнем с 5-стадийный методов. Общей вид матрицы в этом случае будет такой:

$$
A_{45} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0\\
a_{21} & 0 & 0 & 0 & 0 & 0\\
a_{31} & a_{32} & 0 & 0 & 0 & 0\\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0\\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0\\
b_1 & b_2 & b_3 & b_4 & b_5 & 0
\end{pmatrix}.
$$

Каждый четырехстайдийный метод 4-го порядка обязательно удовлетворяет упрощающему предположению. Поэтому запуск программы RK.EXE в режиме 1 (без упрощающего предположения) и в режиме 2 (с использованием упрощающего предоположения) приводит к почти одинаковым результатам. При этом, конечно, вычисления с упрощающим предположением проходят быстрее из-за меньшего количества переменных и уравнений.

Для пятистадийных методов ситуация иная. Они могут как удовлетворять, так и не удовлетворять упрощающему предположению. Чтобы убедиться в этом, достаточно заметить, что у матрицы приведенной ниже, коэффициент c_5 отличен от 1, для методов удовлетворяющих упрощающему предположению, этот коэффициент обязательно должен быть равен 1.

Система уравнений Бутчера в этом случае состоит из 8 уравнений от 15 переменных. Уравнения остались те же, что и в предыдущем случае, количество же переменных увеличилось. Хотя система уравнений получается уже значительно более горомоздкой, тем не менее ее решения находятся довольно быстро - в среднем несколько секунд на нахождение одной матрицы. В результате, в течении небольшого времени было найдено более 10000 пятистадийных методов порядка 4. К каждому из них была применена программа RK_USE.EXE. С ее помощью были найдены погрешности каждого из методов на решении одной и той же задачи (см. п.....). Интервал интегрирования разбивался на 200 000 шагов, для того, чтобы общее количество вычислений функции было равно 1000000, то есть столько же, сколько и для четырехстадийных методов.

Из общего количества методов, у 9990 погрешность больше 0.1, и лишь у 10 методов - меньше 0.1. Для примера, выберем из них, мы выбрали наилучший метод (погрешность:($3.077 \cdot 10^{-1}$):

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.27662911231719602, & b(1) & = & 0.11891134444539452, \\
a(3,1) & = & -0.00627859286416447, & b(2) & = & 0.25928135058617516, \\
a(3,2) & = & 0.56432189179083180, & b(3) & = & 0.21907527176513450, \\
a(4,1) & = & 0.09712206747645369, & b(4) & = & 0.23280065054215895, \\
a(4,2) & = & 0.06704406174026453, & b(5) & = & 0.16993138266113686, \\
a(4,3) & = & 0.44377447299248985, & c[2] & = & 0.27662911231719602, \\
a(5,1) & = & 0.22449039373644900, & c[3] & = & 0.55804329892666733, \\
a(5,2) & = & 0.07386638534965839, & c[4] & = & 0.60794060220920808, \\
a(5,3) & = & 0.25400718087312985, & c[5] & = & 0.96799527266010208, \\
a(5,4) & & 0.41563131270086484, & & &
\end{array}
$$

В режиме 2, быстро(несколько минут)было найдено более 5500 пятистадийных методов порядка 4. Число уравнении сократилось до 3 а переменных – до 9. Из общего количества методов, у 4855 методов погрешность- от 10^{-1} до 10^{-2}, у 659 методов - от 10^{-2} до 10^{-3}, и у 5 методов - меньше 10^{-3}. Погрешность наилучшего метода оказалась $4.09 \cdot 10^{-4}$. Выпишем матрицу этого метода:

$$
\begin{array}{lcl@{\qquad}lcl}
a(2,1) & = & 0.10749255373503380, & b(1) & = & 0.10945402527513178,\\
a(3,1) & = & 0.25748729891822796, & b(2) & = & 0.10269805676472501,\\
a(3,2) & = & 0.26618254239527791, & b(3) & = & 0.29798522973467749,\\
a(4,1) & = & 0.04198971346121476, & b(4) & = & 0.33080280165565011,\\
a(4,2) & = & 0.03273581434278963, & b(5) & = & 0.15905988656981561,\\
a(4,3) & = & 0.45082914768803141, & c[2] & = & 0.10749255373503380,\\
a(5,1) & = & 0.04901941201150147, & c[3] & = & 0.52366984131350587,\\
a(5,2) & = & 0.00950091984822643, & c[4] & = & 0.52555467549203580,\\
a(5,3) & = & -0.04524203752855505, & c[5] & = & 1,\\
a(5,4) & & 0.98672170566882714, & & &
\end{array}
$$

В целом, можно сказать, что методы, не удовлетворяющие упрощающему предположению оказались заметно хуже тех, которые удовлетворяют. Судя по всему, упрощающее предположение не только сокращает количество уравнений и переменных, но и отражает какие-то внутренние свойста методов Рунге-Кутта.

5.1.6.Шестистадийные методы 4-го порядка

Для методов Рунге-Кутта шестистадийых 4-го порядка(21 перемменых и 8 уравнений), мы нашли более 1000 решений в режиме 1 очень быстро (несколько минут) с помошью программа RK.EXE. Погрешности методов, найденных в таком режиме оказались очень велики у многие - больше 1, кроме некоторые.Из них, у лучшего метода получили погрешность - $1.128 \cdot 10^{-1}$.

В режиме 2, быстро (за несколько минут) было найдено более 1000 шестистадийных методов порядка 4 . В этом режиме вычислений количество уравнений осталось прежним - 3, количество же переменных увеличилось до 14. У всех методов, погрешность оказалаь меньше 1. Из общего количества методов, у 1124 погрешность - от 10^{-1} до 10^{-2} у 63 метода - от 10^{-2} до 10^{-3}, и у 29 методов - от 10^{-3} до 10^{-4}. Из них, наилучший имеет погрешность $2.6 \cdot 10^{-4}$. Выпишем матрицу этого метода:

$$
\begin{array}{lcl@{\qquad}lcl}
a(2,1) & = & 0.18394776798534021, & a(6,4) & = & 0.05076623223613031,\\
a(3,1) & = & 0.13565175668714090, & a(6,5) & = & 0.77847243351140774,\\
a(3,2) & = & 0.23697261049125985, & b(1) & = & 0.11099151218156427,\\
a(4,1) & = & 0.23204272927694766, & b(2) & = & 0.12622275917088566,\\
a(4,2) & = & 0.14775224749388735, & b(3) & = & 0.16189804779241491,\\
a(4,3) & = & 0.26376019689663269, & b(4) & = & 0.21155643660214971,\\
a(5,1) & = & 0.02498376117491207, & b(5) & = & 0.24928128065481071,\\
a(5,2) & = & 0.10487170696332526, & b(6) & = & 0.14004996359817474,\\
a(5,3) & = & 0.15880582308604130, & c[2] & = & 0.18394776798534021,\\
a(5,4) & & 0.27398121570978080, & c[3] & & 0.37262436717840075,\\
a(6,1) & = & 0.07492501661372310, & c[4] & = & 0.64355517366746770,\\
a(6,2) & = & 0.05168490409873751, & c[5] & = & 0.56264250693405942,\\
a(6,3) & = & 0.04415141354000134, & c[6] & = & 1,
\end{array}
$$

Из всех методов 4-го порядка с помощью программы RK.EXE, наименьшая погрешность $(1.913 \cdot 10^{-4})$ оказалась у 4-х стадийного метода найденного в режиме 1.

5.2. Методы 5-го порядка

Как известно при $p \geq 5$ не существует явных методов Рунге-Кутта порядка p с числом стадий $n = p$ (так называемый "барьер Бутчера"). Поэтому рассмотрение методов 5-го порядка начинаем с 6-стадийных.

5.2.1.Шестистадийные методы 5-го порядка

В этом случае общий вид матрицы следующий:

$$A_{56} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & 0 \end{pmatrix}.$$

В режиме 1 (то есть без упрощающих предположений), в течении примерно часа было найдено более 4900 шестистадийных методов порядка 5 с помощью программы RK.EXE. В этом режиме решаемая система состоит из 17 полиномиальных уравнений от 21 переменной.

Из общего количества методов, у 1247 погрешность больше 1, у 615 методов - от 1 до 10^{-1}, у 1250 методов - от 10^{-1} до 10^{-2}, у 1284 метода - от 10^{-2} до 10^{-3}, у 474 метода - от 10^{-3} до 10^{-4}, и у 37 методов - от 10^{-4} до 10^{-5}. Особенности: $b(2) = 0$ и $c[3] < 0$ Наименьшая погрешность $(1.222 \cdot 10^{-5})$ оказалась у метода со следующей матрицей: .

$$\begin{array}{lcllcl}
a(2,1) & = & 0.00727831696573130, & a(6,4) & = & -0.16789684826231472, \\
a(3,1) & = & 2.24833475451107214, & a(6,5) & = & 1.84681489362229149, \\
a(3,2) & = & -2.33483638737250763, & b(1) & = & -3.56863305401435313, \\
a(4,1) & = & 3.14816373394171417, & b(2) & = & 3.87107828267588623, \\
a(4,2) & = & 4.34512696297589083, & b(3) & = & -0.00398769540983424, \\
a(4,3) & = & -6.01457046846297192, & b(4) & = & 0.00000583529338506, \\
a(5,1) & = & -4.61387806368814166, & b(5) & = & 0.57696262140087176, \\
a(5,2) & = & 6.13327390503349091, & b(6) & = & 0.12457401005404431, \\
a(5,3) & = & -0.95439443225436590, & c[2] & = & 0.00727831696573130, \\
a(5,4) & & 0.03624635186888374, & c[3] & & -0.08650163286143549, \\
a(6,1) & = & -7.43193144526986008, & c[4] & = & 1.47872022845463308, \\
a(6,2) & = & 2.36724808292298008, & c[5] & = & 0.60124776095986709, \\
a(6,3) & = & 4.38576531698690324, & c[6] & = & 1,
\end{array}$$

В режиме 2 (с упрощающим предположением), система состоит из 8 уравнений от 14 переменных. Для нее в течении несколько часов было найдено через более 500 (шестистадийных методов порядка 5).

Из общего количества методов, у более 40 погрешность - больше 1, у 393 метода - от 1 до 10^{-1}, у 139 методов - от 10^{-1} до 10^{-2}, у 375 методов - от 10^{-2} до 10^{-3}, и у 29 методов - от 10^{-3} до 10^{-4}.

Наименьшая погрешность ($1.697 \cdot 10^{-4}$) оказалась у метода со следующей матрицей:

$$
\begin{array}{lcl@{\qquad}lcl}
a(2,1) & = & 0.10465726413494475, & a(6,4) & = & -0.65216110451581539, \\
a(3,1) & = & -0.51005387224535585, & a(6,5) & = & 1.44242855738732231, \\
a(3,2) & = & 0.98718639429146485, & b(1) & = & 0.06237299485982668, \\
a(4,1) & = & 0.34091917373992318, & b(2) & = & 0.07121901500777854, \\
a(4,2) & = & -0.17208101026500930, & b(3) & = & -0.06712847088665127, \\
a(4,3) & = & 0.13992772213123003, & b(4) & = & 0.41165323113101695, \\
a(5,1) & = & -0.31849131896558775, & b(5) & = & 0.43563139136645310, \\
a(5,2) & = & 0.29147809117130860, & b(6) & = & 0.08625183852157600, \\
a(5,3) & = & -0.04088723843799547, & c[2] = & = & 0.10465726413494475, \\
a(5,4) & & 0.78231013120055126, & c[3] = & & 0.47713252204610900, \\
a(6,1) & = & 0.22126610171755046, & c[4] = & = & 0.3087658856061439, \\
a(6,2) & = & 0.85672884914995368, & c[5] = & = & 0.71440966496827664, \\
a(6,3) & = & -0.86826240373901106, & c[6] = & = & 1,
\end{array}
$$

5.2.2. Семистадийных методы 5-го порядка

$$
A_{57} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 \\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 \\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 \\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & 0
\end{pmatrix}.
$$

а) Было найдено более 2300 семистадийных методов порядка 5 с помощью программы RK_EXE в режиме 1 (то есть без упрощающих предположений, 17 уравнений от 28 переменных). С помощью программы RK_USE.EXE была найдена погрешность решения задачи за 142857 шагов, то есть за 1000000 вычислений правой части системы уравнений. Из всех рассмотренныз методов, наименьшая погрешность а именно $8.725 \cdot 10^{-5}$, достигается для метода со следующей матрицей:

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.04185977601021258, & a(7,3) & = & 0.08367217272126381,\\
a(3,1) & = & -0.56900098785532786, & a(7,4) & = & 0.02729570416598036,\\
a(3,2) & = & 0.84196583674461851, & a(7,5) & = & -0.31506332951252581,\\
a(4,1) & = & 0.18369078860192582, & a(7,6) & = & 0.94407012776454778,\\
a(4,2) & = & -0.08308950689655647, & b(1) & = & -0.02914875320814301,\\
a(4,3) & = & 0.06117105500462851, & b(2) & = & 0.05962171794877373,\\
a(5,1) & = & -0.08251685587616795, & b(3) & = & -0.01810106570627492,\\
a(5,2) & = & -0.02837693605319256, & b(4) & = & 0.35259618419042294,\\
a(5,3) & = & 0.51506399650554954, & b(5) & = & 0.20670517239151820,\\
a(5,4) & = & 0.17482466345826965, & b(6) & = & 0.31964511035883712,\\
a(6,1) & = & -0.25237190941061218, & b(7) & = & 0.10868163402486595,\\
a(6,2) & = & 0.21987690590624930, & c[2] & = & 0.04185977601021258,\\
a(6,3) & = & -0.47017397542526596, & c[3] & = & 0.27296484888929066,\\
a(6,4) & = & 0.80230281433670058, & c[4] & = & 0.16177233670999786,\\
a(6,5) & = & 0.37937553844403570, & c[5] & = & 0.57899486803445868,\\
a(7,1) & = & -0.08268731597028387, & c[6] & = & 0.67900937385110745,\\
a(7,2) & = & 0.34271264083101779, & c[7] & = & 1.00000000000000006,
\end{array}
$$

В режиме 2 (с упрощающим предположением), система состоит из 8 уравнений от 20 переменных. Для нее в течении несколько часов было найдено несколько семистадийных методов порядка 5 Рунге-Кутта с помощью программы RK_EXE. Из общего количества методов, вычисление погрешности дает хорошие результаты. Выберем наилучший метод ($8.797 \cdot 10^{-6}$) и выпишем соответствующую матрицу:

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.07031631863188226, & a(7,3) & = & 0.43694774787958571,\\
a(3,1) & = & -0.12486355592851731, & a(7,4) & = & -0.14055413393716607,\\
a(3,2) & = & 0.30946502082985555, & a(7,5) & = & -0.47615019010709204,\\
a(4,1) & = & -0.14477122523875633, & a(7,6) & = & 1.04679215568578060,\\
a(4,2) & = & 0.01712944770613268, & b(1) & = & -0.07103951918967855,\\
a(4,3) & = & 0.53877148975661113, & b(2) & = & 0.22471135287074367,\\
a(5,1) & = & -0.00374800917933312, & b(3) & = & 0.21287019578268091,\\
a(5,2) & = & 0.10077969357667145, & b(4) & = & -0.02736877751957849,\\
a(5,3) & = & 0.19618050437431184, & b(5) & = & 0.19837772515167296,\\
a(5,4) & = & 0.25251929107093959, & b(6) & = & 0.35301478685720315,\\
a(6,1) & = & -0.09006230167190893, & b(7) & = & 0.10943423604695635,\\
a(6,2) & = & 0.21889755508132761, & c[2] & = & 0.07031631863188226,\\
a(6,3) & = & 0.28776323674784540, & c[3] & = & 0.18460146490133825,\\
a(6,4) & = & -0.14398668281140161, & c[4] & = & 0.41112971222398748,\\
a(6,5) & = & 0.40288365595392781, & c[5] & = & 0.54573147984258976,\\
a(7,1) & = & -0.28954403592210860, & c[6] & = & 0.67549546329979026,\\
a(7,2) & = & 0.42250845640100040, & c[7] & = & 1,
\end{array}
$$

b)Метод Дормана и Принса. Метод в котором к тому же $a_{ni} = b_i$,так что результат последнего вычисления производной может снова использоваться на следующем шаге,построили Дорман и Принс (1980).Он дает великолепные результаты - погрешность при тех же условиях, что и раньше (1000000 вычислений значения функции) оказалась равна $3.98 \cdot 10^{-10}$, что значительно меньше, чем для всех остальных методов. Отметим также, что $b(2) = b(7) = a(7,2) = 0$. Коэффициенты этого метода приведены в следующей матрице:

$$
\begin{array}{lcllcl}
a(2,1) &=& 1/5, & a(7,3) &=& 500/1113,\\
a(3,1) &=& 3/40, & a(7,4) &=& 125/192,\\
a(3,2) &=& 9/40, & a(7,5) &=& -2187/6784,\\
a(4,1) &=& 44/45, & a(7,6) &=& 11/84,\\
a(4,2) &=& -56/15, & b(1) &=& 35/384,\\
a(4,3) &=& 32/9, & b(2) &=& 0,\\
a(5,1) &=& 19372/6561, & b(3) &=& 500/1113,\\
a(5,2) &=& -25360/2187, & b(4) &=& 125/192,\\
a(5,3) &=& 64448/6561, & b(5) &=& -2187/6784,\\
a(5,4) &=& -212/729, & b(6) &=& 11/84,\\
a(6,1) &=& 9017/3168, & b(7) &=& 0,\\
a(6,2) &=& -355/33, & c[2] &=& 1/5,\\
a(6,3) &=& 46732/5247, & c[3] &=& 3/10,\\
a(6,4) &=& 49/176, & c[4] &=& 4/5,\\
a(6,5) &=& -5103/18656, & c[5] &=& 8/9,\\
a(7,1) &=& 35/384, & c[6] &=& 1,\\
a(7,2) &=& 0, & c[7] &=& 1,
\end{array}
$$

Мы можем сказать, что метод Дорман-Принса $(5,7)$, пока является найлучшим среди предыдущими рассмотревающими методами.

5.3. Методы 6-го порядка

5.3.1. Семистадийных методы 6-го порядка

c_2, c_3, c_5, c_6 являются свободными параметрами. Среди множества возможных примеров мы выбрали метод(Бутчера (6,7)), потому что он обладает положительными $b_1, b_2, \ldots, b_7$.

При $p \geq 7$ не существует явных методов p,имеющих $n = p + 1$ стадию(Бутчер(1965b)).

Общей вид матриций в следующем образом:

$$
A_{67} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0\\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0\\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0\\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0\\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0\\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & 0
\end{pmatrix}.
$$

а) Было найдено более 40 семистадийных методов порядка 6 Рунге-Кутта с помощью программы RK_EXE в режиме 1 (то есть без упрощающих предположений). Мы получаем 37 уравнений от 28 переменных. Мы вычислели их погрешности с помощью программы RK_USE.EXE. Мы заметили после вычислении погрешности , что метод хороший. Из них мы выбрали лучший метод с погрешности $2.501 \cdot 10^{-6}$.Особенности метода : $b(2) = b(3) = 0$. Матрица данного метода в следующем виде:

$$
\begin{array}{lcl@{\qquad}lcl}
a(2,1) & = & 0.27570243018011160, & a(7,3) & = & 0.41167425753033579, \\
a(3,1) & = & 3.57774883785972516, & a(7,4) & = & -0.47216749844484514, \\
a(3,2) & = & -3.31800225865506774, & a(7,5) & = & -7.40055902508654856, \\
a(4,1) & = & -0.29350710595779466, & a(7,6) & = & 1.07336729029111384, \\
a(4,2) & = & 1.14446208909865877, & b(1) & = & 0.08333333333333333, \\
a(4,3) & = & -0.12772759878587911, & b(2) & = & 0.73817572713716671, \\
a(5,1) & = & 0.49375307202644892, & b(3) & = & 0.00180158328007867, \\
a(5,2) & = & -0.24216903382867028, & b(4) & = & 0.09291722845258618, \\
a(5,3) & = & 0.00921799413915299, & b(5) & = & -0.32331148428152149, \\
a(5,4) & = & 0.01392195139637985, & b(6) & = & 0.32375027874502327, \\
a(6,1) & = & 0.14328912910327743, & b(7) & = & 0.08333333333333333, \\
a(6,2) & = & -0.75907659202817770, & c[2] & = & 0.27570243018011160, \\
a(6,3) & = & -0.05598198144737488, & c[3] & = & 0.25974657920465742, \\
a(6,4) & = & 0.21487352206799066, & c[4] & = & 0.72322738435498500, \\
a(6,5) & = & 1.18061114337634929, & c[5] & = & 0.27472398373331148, \\
a(7,1) & = & 0.16666666666666665, & c[6] & = & 0.72371522107206480, \\
a(7,2) & = & 7.22101830904327743, & c[7] & = & 1,
\end{array}
$$

В режиме 2 (с упрощающим предположением), система состоит из 19 уравнений от 20 переменных. Для нее в течении несколько часов было найдено несколько семистадийных методов порядка 6 Рунге-Кутта с помощью программы RK_EXE. После прохождения 142857 шагов, погрешность решения на последнем шаге составила $(1.19 \cdot 10^{-6})$ и выпишим соответствующая матрица:

$$
\begin{aligned}
a(2,1) &= 0.27402869584764873, & a(7,3) &= 0.05554670724019128,\\
a(3,1) &= -2.94803488094554378, & a(7,4) &= 0.40934819629583718,\\
a(3,2) &= 3.28640828765406540, & a(7,5) &= -1.23527952634360115,\\
a(4,1) &= 0.12797933517337341, & a(7,6) &= 1.40462242753410118,\\
a(4,2) &= 0.46698249517459300, & b(1) &= 0.08333333333333334,\\
a(4,3) &= 0.13442955575330779, & b(2) &= 0.39193607813788643,\\
a(5,1) &= -1.93693506045360869, & b(3) &= 0.03021796385542233,\\
a(5,2) &= 1.74975177972694356, & b(4) &= 0.35377855580630075,\\
a(5,3) &= -0.89034363204961303, & b(5) &= -0.26754475309618831,\\
a(5,4) &= 1.70133534102773174, & b(6) &= 0.32494548862991212,\\
a(6,1) &= -1.57677780193166558, & b(7) &= 0.08333333333333333,\\
a(6,2) &= 1.45120648399924713, & c[2] &= 0.27402869584764873,\\
a(6,3) &= -0.83214248096756546, & c[3] &= 0.33837340670852161,\\
a(6,4) &= 1.59044066391148741, & c[4] &= 0.72939138610127420,\\
a(6,5) &= 0.00705311945694326, & c[5] &= 0.62380842825145358,\\
a(7,1) &= 0.16666666666666660, & c[6] &= 0.63977998446844677,\\
a(7,2) &= 0.19909552860680491, & c[7] &= 1,
\end{aligned}
$$

b) После вычислении погрешности метода Бутчера ($b(2) = 0$), мы получили $1.265 \cdot 10^{-2}$ и выпишим данного матрица:

$$
\begin{aligned}
a(2,1) &= 1/2, & a(7,3) &= 43/156,\\
a(3,1) &= 2/9, & a(7,4) &= -118/39,\\
a(3,2) &= 4/9, & a(7,5) &= 32/195,\\
a(4,1) &= 7/36, & a(7,6) &= 80/39,\\
a(4,2) &= 2/9, & b(1) &= 13/200,\\
a(4,3) &= -1/12, & b(2) &= 0,\\
a(5,1) &= -35/144, & b(3) &= 11/40,\\
a(5,2) &= -55/36, & b(4) &= 11/40,\\
a(5,3) &= 35/48, & b(5) &= 4/25,\\
a(5,4) &= 15/8, & b(6) &= 4/25,\\
a(6,1) &= -1/360, & b(7) &= 13/200,\\
a(6,2) &= -11/36, & c[2] &= 1/2,\\
a(6,3) &= -1/8, & c[3] &= 2/3,\\
a(6,4) &= 1/2, & c[4] &= 1/3,\\
a(6,5) &= 1/10, & c[5] &= 5/6,\\
a(7,1) &= -41/260, & c[6] &= 1/6,\\
a(7,2) &= 22/13, & c[7] &= 1,
\end{aligned}
$$

5.3.2. Восимистадийных методы 6-го порядка

Общей вид матриций в следующем образом:

$$A_{68} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & 0 \end{pmatrix}.$$

Было найдено многие восимистадийных методов порядка 6. В режиме 1 мы получили 37 уравнений от 36 переменных. Итерации метода Ньютона в этом случае сходятся очень медленно. Тем не менее, удалось найти несколько сот таких методов. Из них выберем метод, который имеет наименьшую погрешность ($9.966 \cdot 10^{-6}$). Оказывается,что $b(2) = b(3) = b(4) = b(6) = 0$ Выпишем матрицу этого метода:

$$\begin{array}{lcllcl} a(2,1) & = & 0.06633985175729792, & a(8,2) & = & -3.18886018769169785, \\ a(3,1) & = & -0.10763143145240780, & a(8,3) & = & 1.60802574180509715, \\ a(3,2) & = & 0.56994254133228203, & a(8,4) & = & -0.52688047360428619, \\ a(4,1) & = & -0.13178936041171854, & a(8,5) & = & 2.94933122820289908, \\ a(4,2) & = & 0.34699988123115100, & a(8,6) & = & -3.08334297895167737, \\ a(4,3) & = & 0.27554493406377064, & a(8,7) & = & 1.38196601125010316, \\ a(5,1) & = & -0.16200453623297531, & b(1) & = & 0.08333333333333256, \\ a(5,2) & = & 0.41711422953946341, & b(2) & = & 0.00000000000000170, \\ a(5,3) & = & -0.00282329786111343, & b(3) & = & -0.00000000000000019, \\ a(5,4) & = & 0.02410680680464688, & b(4) & = & -0.00000000000000019, \\ a(6,1) & = & 0.71990909191786172, & b(5) & = & 0.41666666666666596, \\ a(6,2) & = & -0.84487123444111388, & b(6) & = & 0.00000000000000074, \\ a(6,3) & = & -0.06200052292499902, & b(7) & = & 0.41666666666666614, \\ a(6,4) & = & -0.07236984048035502, & b(8) & = & 0.08333333333333327, \\ a(6,5) & = & 0.77450882601136761, & c[2] & = & 0.06633985175729792, \\ a(7,1) & = & -0.00994759556494060, & c[3] & = & 0.46231110987987423, \\ a(7,2) & = & 0.22065780799888239, & c[4] & = & 0.49075545488320311, \\ a(7,3) & = & -0.31878185049990617, & c[5] & = & 0.27639320225002156, \\ a(7,4) & = & 0.08126928791621032, & c[6] & = & 0.51517632008276142, \\ a(7,5) & = & 0.13374055210939664, & c[7] & = & 0.72360679774997923, \\ a(7,6) & = & 0.61666859579033665, & c[8] & = & 0.99999999999999999, \\ a(8,1) & = & 1.85976065898956200, & & & \end{array}$$

В режиме 2 , было найдено несколько восимистадийных методов порядка 6 (18 уравнений от 27 переменных). Процесс – медленно. Из общего количества методов, у 8 погрешность - от 1 до 10^{-1}, у 74 метода - от 10^{-1} до 10^{-2}, у 173 метода - от 10^{-2} до 10^{-3}, у 162 метода - от 10^{-3} до 10^{-4}, у 43 метода - от 10^{-4} до 10^{-5}, у 5 методов - от 10^{-5} до 10^{-6} и у 1 метода - меньше 10^{-6} . Самая лучшая из них $4.413 \cdot 10^{-7}$. Мы получили $b(2) = b(3) = b(5) = b(6) = 0$. Выпишем матрицу этого метода:

$$
\begin{array}{lcl@{\qquad}lcl}
a(2,1) & = & 0.17091491992865984, & a(8,2) & = & -0.08972385883693802, \\
a(3,1) & = & -0.32252167298789009, & a(8,3) & = & 0.73566965926212073, \\
a(3,2) & = & 0.55286665305501682, & a(8,4) & = & 0.15504160218712476, \\
a(4,1) & = & 0.06585968177082702, & a(8,5) & = & -0.55419833707414830, \\
a(4,2) & = & 0.17329173871330902, & a(8,6) & = & -0.60118852357155465, \\
a(4,3) & = & 0.03724178176588499, & a(8,7) & = & 1.38196601125010514, \\
a(5,1) & = & 0.02780408941107911, & b(1) & = & 0.08333333333333333, \\
a(5,2) & = & 0.44954863834724432, & b(2) & = & 0, \\
a(5,3) & = & -0.40468521706156374, & b(3) & = & 0, \\
a(5,4) & = & 0.66088771792872327, & b(4) & = & 0.41666666666666667, \\
a(6,1) & = & -0.33287030556431405, & b(5) & = & 0, \\
a(6,2) & = & -0.24267939390184179, & b(6) & = & 0, \\
a(6,3) & = & 0.56047945491940119, & b(7) & = & 0.41666666666666666, \\
a(6,4) & = & 0.34186529027361703, & b(8) & = & 0.08333333333333333, \\
a(6,5) & = & 0.15631468194832447, & c[2] & = & 0.17091491992865984, \\
a(7,1) & = & 0.13965362887251491, & c[3] & = & 0.23034498006712673, \\
a(7,2) & = & -0.15534696694592141, & c[4] & = & 0.27639320225002103, \\
a(7,3) & = & -0.18437571361830913, & c[5] & = & 0.73355522862548297, \\
a(7,4) & = & 0.69259847731255402, & c[6] & = & 0.48310972767518683, \\
a(7,5) & = & 0.11083966741482966, & c[7] & = & 0.72360679774997897, \\
a(7,6) & = & 0.12023770471431093, & c[8] & = & 1, \\
a(8,1) & = & -0.02756655321670965, & & &
\end{array}
$$

5.3.3. Девятистадийных методы 6-го порядка

Общей вид матриций в следующем образом:

$$
A_{69} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 \\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 \\
a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 \\
a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & 0
\end{pmatrix}.
$$

RK_EXE (37 уравнений от 45 переменных) в режиме 1 . Процесс было медленно (несколько дней). После вычислении погрешности, мы заметили, что найдены методы хорошие и лучший из них $(1.091 \cdot 10^{-3})$ в матричным виде:

$$
\begin{aligned}
a(2,1) &= 0.06377594525174173, \\
a(3,1) &= -0.24166534652061487, \\
a(3,2) &= 0.48969840212127384, \\
a(4,1) &= 0.38135275726882226, \\
a(4,2) &= -0.56382992709692760, \\
a(4,3) &= 0.73759138023773998, \\
a(5,1) &= 0.24392634224013229, \\
a(5,2) &= -0.26417643070091878, \\
a(5,3) &= 0.43769835140597853, \\
a(5,4) &= 0.30725113561709386, \\
a(6,1) &= 0.09133556184679123, \\
a(6,2) &= -0.11159100010237438, \\
a(6,3) &= 0.77093517367968121, \\
a(6,4) &= -0.64222425680989822, \\
a(6,5) &= 0.74176791849857039, \\
a(7,1) &= -0.24798837395524381, \\
a(7,2) &= 0.51205201856764170, \\
a(7,3) &= 0.33581612517503676, \\
a(7,4) &= -0.13857084623747194, \\
a(7,5) &= -0.08249781901314753, \\
a(7,6) &= 0.60758264113857792, \\
a(8,1) &= 0.38579120092732301, \\
a(8,2) &= -0.46173163121712612, \\
a(8,3) &= 0.66764545740249403, \\
a(8,4) &= -0.47912827310874253, \\
a(8,5) &= 0.08283787436004992, \\
a(8,6) &= 0.42722599013821137,
\end{aligned}
$$

$$
\begin{aligned}
a(8,7) &= -0.21418917998224918, \\
a(9,1) &= -0.02024765702171238, \\
a(9,2) &= 0.37546152226287739, \\
a(9,3) &= -0.57677061993949847, \\
a(9,4) &= -0.37757065637150326, \\
a(9,5) &= 0.82242136093264956, \\
a(9,6) &= -0.70838312024672048, \\
a(9,7) &= 0.39457396864782068, \\
a(9,8) &= 1.09051520173608686, \\
b(1) &= 0.06975662887936167, \\
b(2) &= 0.01859459071759125, \\
b(3) &= 0.30285254673254277, \\
b(4) &= -0.01490523436377146, \\
b(5) &= 0.36940474777545290, \\
b(6) &= 0.03461000189162999, \\
b(7) &= -0.00161376442806634, \\
b(8) &= 0.14347347937532207, \\
b(9) &= 0.07782700341993714, \\
c[2] &= 0.06377594525174173, \\
c[3] &= 0.24803305560065897, \\
c[4] &= 0.55511421040963464, \\
c[5] &= 0.72469939856228591, \\
c[6] &= 0.85022339711277024, \\
c[7] &= 0.98639374567539311, \\
c[8] &= 0.40845143851996050, \\
c[9] &= 0.99999999999999991,
\end{aligned}
$$

В режиме 2 , было найдено несколько восимистадийных методов порядка 6 (19 уравнений от 35 переменных). Процесс сходится очень медленно. Из всех найденных методов, выпишем метод с наименьшей погрешностью $(1.246 \cdot 10^{-6})$. Его характереная особенность: $b(2) = 0$. Коэффициенты этого метода приведены в следующей матрице:

$$
\begin{aligned}
a(2,1) &= 0.11787424399872871, \\
a(3,1) &= 0.03563046096796495, \\
a(3,2) &= 0.15636874575197961, \\
a(4,1) &= 0.11872725438158715, \\
a(4,2) &= -0.08201433020662056, \\
a(4,3) &= 0.21386303906859307, \\
a(5,1) &= 0.09366668564286509, \\
a(5,2) &= -0.16046475736599736, \\
a(5,3) &= 0.18413186856694493, \\
a(5,4) &= 0.32524028225672374, \\
a(6,1) &= 0.18764442210235394, \\
a(6,2) &= -0.09575795247902063, \\
a(6,3) &= 0.10082408972253844, \\
a(6,4) &= -0.02458432043995927, \\
a(6,5) &= 0.50000537877025639, \\
a(7,1) &= -0.08126630212910641, \\
a(7,2) &= 0.24450563843791397, \\
a(7,3) &= -0.06174469490959869, \\
a(7,4) &= 0.50497093488110497, \\
a(7,5) &= -0.17860643836058194, \\
a(7,6) &= 0.30005031168275789, \\
a(8,1) &= 0.16404563732509593, \\
a(8,2) &= -0.15739158749456607, \\
a(8,3) &= 0.36851522918886277, \\
a(8,4) &= -0.13028648839672466, \\
a(8,5) &= 0.31114034387375244, \\
a(8,6) &= -0.06894164818297322,
\end{aligned}
$$

$$
\begin{aligned}
a(8,7) &= 0.44853438102659407, \\
a(9,1) &= 0.00020938899361035, \\
a(9,2) &= 0.03161395517137291, \\
a(9,3) &= -0.00697533110471281, \\
a(9,4) &= 0.41406136255423328, \\
a(9,5) &= 0.24810617191179654, \\
a(9,6) &= -0.06957650967153484, \\
a(9,7) &= 0.13186996863703688, \\
a(9,8) &= 0.25069099350819768, \\
b(1) &= 0.06912252613439566, \\
b(2) &= 0.00000000000000022, \\
b(3) &= 0.13322001126891937, \\
b(4) &= 0.18261985766946323, \\
b(5) &= 0.14162099285019795, \\
b(6) &= 0.14527800665335876, \\
b(7) &= 0.19201841790093901, \\
b(8) &= 0.10830466200744192, \\
b(9) &= 0.02781552551528386, \\
c[2] &= 0.11787424399872871, \\
c[3] &= 0.19199920671994456, \\
c[4] &= 0.25057596324355966, \\
c[5] &= 0.44257407910053641, \\
c[6] &= 0.66813161767616888, \\
c[7] &= 0.72790944960248979, \\
c[8] &= 0.93561586734004126, \\
c[9] &= 1,
\end{aligned}
$$

5.3.4. Десятистадийных методы 6-го порядка

Общей вид матриций в следующем образом:

$$
A_{610} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0 \\
a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0 \\
a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0 \\
a_{10,1} & a_{10,2} & a_{10,3} & a_{10,4} & a_{10,5} & a_{10,6} & a_{10,7} & a_{10,8} & a_{10,9} & 0 & 0 \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & 0
\end{pmatrix}.
$$

RK_EXE (37 уравнений от 55 переменных). Процесс требовал несколько дней. После вычислении погрешности, мы выбрали самый лучший метод $(7.097 \cdot 10^{-5})$. Особенность–b(2) почти равно нуль. Выпишим матрицу данного метода :

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.10099316708803768, & a(9,5) & = & -0.17197079931753203,\\
a(3,1) & = & 0.02524338949190244, & a(9,6) & = & 0.24596519838259258,\\
a(3,2) & = & 0.14719336893388977, & a(9,7) & = & -0.10295234484913842,\\
a(4,1) & = & 0.10307924084557942, & a(9,8) & = & 0.17765252698282128,\\
a(4,2) & = & -0.11188397970023132, & a(10,1) & = & 0.11266572037672758,\\
a(4,3) & = & 0.28209218255612704, & a(10,2) & = & -0.13193291368996510,\\
a(5,1) & = & 0.09742989653190980, & a(10,3) & = & 0.24237703113616906,\\
a(5,2) & = & -0.02360955397525642, & a(10,4) & = & 0.15087180644308101,\\
a(5,3) & = & 0.09277134857987836, & a(10,5) & = & 0.21568690431519618,\\
a(5,4) & = & 0.21083852954585032, & a(10,6) & = & -0.23759856524270026,\\
a(6,1) & = & 0.17309518099200092, & a(10,7) & = & 0.13688033750230977,\\
a(6,2) & = & -0.08534751974784361, & a(10,8) & = & 0.11395326738909358,\\
a(6,3) & = & -0.12427899503548107, & a(10,9) & = & 0.39709641177008819,\\
a(6,4) & = & 0.35047112420360625, & b(1) & = & 0.06918248406780436,\\
a(6,5) & = & 0.25569949156788550, & b(2) & = & -0.00000062554437788,\\
a(7,1) & = & -0.00939711212620068, & b(3) & = & 0.12263703030237127,\\
a(7,2) & = & 0.07493563001107430, & b(4) & = & 0.18764547080236246,\\
a(7,3) & = & 0.06395667698521343, & b(5) & = & 0.10467837026675515,\\
a(7,4) & = & 0.15143160087373827, & b(6) & = & 0.12124383810478960,\\
a(7,5) & = & 0.19719855863977564, & b(7) & = & 0.09545970944236365,\\
a(7,6) & = & 0.31192446653008609, & b(8) & = & 0.12257372810275386,\\
a(8,1) & = & 0.07153392965075404, & b(9) & = & 0.10398183450362891,\\
a(8,2) & = & 0.03310156248853542, & b(10) & = & 0.07259815995154863,\\
a(8,3) & = & 0.19316210515636480, & c[2] & = & 0.10099316708803768,\\
a(8,4) & = & 0.05608308535489480, & c[3] & = & 0.17243675842579220,\\
a(8,5) & = & 0.14331335184159084, & c[4] & = & 0.27328744370147514,\\
a(8,6) & = & 0.11483433349933327, & c[5] & = & 0.37743022068238206,\\
a(8,7) & = & 0.16977289067324301, & c[6] & = & 0.56963928198016800,\\
a(9,1) & = & -0.00472741689327975, & c[7] & = & 0.79004982091368706,\\
a(9,2) & = & 0.13588184879963044, & c[8] & = & 0.78180125866471618,\\
a(9,3) & = & 0.06285208472099829, & c[9] & = & 0.72275475850673914,\\
a(9,4) & = & 0.38005366068064676, & c[10] & = & 1,
\end{array}
$$

В режиме 2 , было найдено несколько десятистадийных методов порядка 6 (19 уравнений от 44 переменных). Процесс – медленно(несколько дней). После вычислении погрешности , мы выбрали найлучшего метода$(1.873 \cdot 10^{-5})$ соотствествующая матрица:

$$
\begin{aligned}
a(2,1) &= 0.07133703472816074, & a(9,5) &= 0.29823782500673930,\\
a(3,1) &= 0.01278722958793688, & a(9,6) &= 0.13606995122709567,\\
a(3,2) &= 0.12183659454346904, & a(9,7) &= -0.37293700791677206,\\
a(4,1) &= -0.04729803371904548, & a(9,8) &= 0.48671970679507131,\\
a(4,2) &= 0.06377014174341405, & a(10,1) &= 0.07962580960318516,\\
a(4,3) &= 0.26477308186611828, & a(10,2) &= -0.00787577318542346,\\
a(5,1) &= 0.18941382907475159, & a(10,3) &= 0.07014398172682900,\\
a(5,2) &= -0.18341913405685989, & a(10,4) &= 0.19736064785758253,\\
a(5,3) &= 0.16653118326166899, & a(10,5) &= 0.21705720875877946,\\
a(5,4) &= 0.14349336926872339, & a(10,6) &= -0.11501477240865577,\\
a(6,1) &= 0.06169149972181072, & a(10,7) &= 0.15079362422444505,\\
a(6,2) &= 0.01150420956781215, & a(10,8) &= 0.09897985689846510,\\
a(6,3) &= 0.01523569110593313, & a(10,9) &= 0.30892941652479294,\\
a(6,4) &= 0.12941441135838184, & b(1) &= 0.06374550994603844,\\
a(6,5) &= 0.30042677607629164, & b(2) &= -0.00804872042082124,\\
a(7,1) &= 0.01381774506319728, & b(3) &= 0.11757288423676071,\\
a(7,2) &= -0.02843766067927550, & b(4) &= 0.11168944363025561,\\
a(7,3) &= 0.12249960997939839, & b(5) &= 0.17835592827150361,\\
a(7,4) &= 0.14440105855515820, & b(6) &= 0.11418085653880962,\\
a(7,5) &= 0.17649632730516631, & b(7) &= -0.00070201256106632,\\
a(7,6) &= 0.27469762001398952, & b(8) &= 0.23693933323022889,\\
a(8,1) &= 0.10091271575645934, & b(9) &= 0.11342519829525935,\\
a(8,2) &= -0.01110944122744073, & b(10) &= 0.07284157883303134,\\
a(8,3) &= 0.12935828700726935, & c[2] &= 0.07133703472816074,\\
a(8,4) &= 0.02182633301174140, & c[3] &= 0.13462382413140591,\\
a(8,5) &= 0.16111492886587776, & c[4] &= 0.28124518989048686,\\
a(8,6) &= 0.20317853297976709, & c[5] &= 0.31601924754828408,\\
a(8,7) &= 0.13129202103731256, & c[6] &= 0.51827258783022947,\\
a(9,1) &= -0.02141223704680623, & c[7] &= 0.70347470023763419,\\
a(9,2) &= 0.04994157564142407, & c[8] &= 0.73657337743098677,\\
a(9,3) &= 0.04458884271352545, & c[9] &= 0.80160575616491035,\\
a(9,4) &= 0.18039709974463286, & c[10] &= 1,
\end{aligned}
$$

5.3.5. Одйннадцатистадийных методы 6-го порядка

Общей вид матриций в следующем образом:

$$A_{611} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0 & 0 \\ a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0 & 0 \\ a_{101} & a_{102} & a_{103} & a_{104} & a_{105} & a_{106} & a_{107} & a_{108} & a_{109} & 0 & 0 & 0 \\ a_{111} & a_{112} & a_{113} & a_{114} & a_{115} & a_{116} & a_{117} & a_{118} & a_{119} & a_{1110} & 0 & \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & 0 \end{pmatrix}.$$

Для методов Рунге-Кутта одйннадцатистадийных 6-го порядка(37 уравнений от 66 перемменых), мы нашли более 60 решений в режиме 1 медленно с помошью программы RK.EXE. Самая нижая погрешность $(1.99 \cdot 10^{-4})$ методов, найденных в таком режиме оказалась матрица в следующем виде:

$$
\begin{aligned}
a(2,1) &= 0.04332870053302830, \\
a(3,1) &= -0.16712700134062018, \\
a(3,2) &= 0.33846590437862169, \\
a(4,1) &= 0.28179955192595694, \\
a(4,2) &= -0.30150504825631392, \\
a(4,3) &= 0.29115426152303984, \\
a(5,1) &= 0.16261484948686949, \\
a(5,2) &= -0.09123922619703079, \\
a(5,3) &= 0.06043100786671784, \\
a(5,4) &= 0.01775156753377500, \\
a(6,1) &= 0.09520311442570742, \\
a(6,2) &= 0.03607017400740377, \\
a(6,3) &= -0.06109965118457564, \\
a(6,4) &= 0.39205100925039848, \\
a(6,5) &= -0.03154961349754980, \\
a(7,1) &= 0.06449110508257072, \\
a(7,2) &= 0.06145694851640118, \\
a(7,3) &= 0.04379842250006209, \\
a(7,4) &= 0.36637647377739214, \\
a(7,5) &= -0.13789873824930269, \\
a(7,6) &= 0.17564135965259331, \\
a(8,1) &= -0.04049755671416830, \\
a(8,2) &= -0.02657233610571366, \\
a(8,3) &= 0.04112729217440444, \\
a(8,4) &= -0.01411957858259914, \\
a(8,5) &= 0.31722158585766282, \\
a(8,6) &= 0.19566482654684684, \\
a(8,7) &= 0.33375367816660201, \\
a(9,1) &= 0.00640873585095069, \\
a(9,2) &= 0.05759357848764258, \\
a(9,3) &= 0.26858302084955899, \\
a(9,4) &= 0.10193565442217711, \\
a(9,5) &= 0.13055839313615658, \\
a(9,6) &= -0.11398379800560440, \\
a(9,7) &= 0.03308078932620940, \\
a(9,8) &= 0.29563455851558488, \\
a(10,1) &= 0.06291732666818249, \\
a(10,2) &= 0.13136016442127034,
\end{aligned}
$$

$$
\begin{aligned}
a(10,3) &= -0.04715194563587637, \\
a(10,4) &= 0.23783045084263615, \\
a(10,5) &= -0.10028650091290415, \\
a(10,6) &= 0.22107529900952992, \\
a(10,7) &= 0.22495224990255355, \\
a(10,8) &= -0.20458393523923796, \\
a(10,9) &= 0.27792205084756133, \\
a(11,1) &= 0.10155961941429899, \\
a(11,2) &= -0.09639266973094318, \\
a(11,3) &= 0.15071090503968705, \\
a(11,4) &= -0.05235427502207069, \\
a(11,5) &= 0.23333438678293527, \\
a(11,6) &= 0.04711031690847987, \\
a(11,7) &= 0.24365430341149816, \\
a(11,8) &= 0.12327722476295498, \\
a(11,9) &= 0.08929684023695497, \\
a(11,10) &= 0.15980334123505945, \\
b(1) &= 0.06245417575732734, \\
b(2) &= -0.00044985915762939, \\
b(3) &= 0.10908640981760272, \\
b(4) &= 0.16797650451735258, \\
b(5) &= 0.04194165614198907, \\
b(6) &= 0.10495504326856984, \\
b(7) &= 0.17333979679731839, \\
b(8) &= 0.12848315939924069, \\
b(9) &= 0.09368050303365002, \\
b(10) &= 0.05324227679134172, \\
b(11) &= 0.06529033363369834, \\
c[2] &= 0.04332870053302830, \\
c[3] &= 0.17133890303800150, \\
c[4] &= 0.27144876519268286, \\
c[5] &= 0.14955819869033154, \\
c[6] &= 0.43067503300138423, \\
c[7] &= 0.57386557127971675, \\
c[8] &= 0.80657791134303501, \\
c[9] &= 0.77981093258267583, \\
c[10] &= 0.80403515990371532, \\
c[11] &= 0.99999999303885488,
\end{aligned}
$$

В режиме 2 (с упрощающим предположением), система состоит из 19 уравнений от 54 переменных. Для нее в течении несколько часов было найдено более 80 методов (одйннадцатистадийных методов порядка 6). Особенность: $b(2)$ почти равно нуль . После вычислении погрешности методов, мы заметили , найлучший из них $(4.232 \cdot 10^{-5})$ имеет матрицйы в следующем виде:

$$
\begin{aligned}
a(2,1) &= 0.09014195559593239, & a(10,3) &= 0.20504099972273892,\\
a(3,1) &= -0.01139476185772238, & a(10,4) &= -0.12146244440719172,\\
a(3,2) &= 0.20244059524053983, & a(10,5) &= 0.32195628048992380,\\
a(4,1) &= 0.11644563825354430, & a(10,6) &= 0.18650286419986464,\\
a(4,2) &= -0.10122826304306933, & a(10,7) &= -0.24245784017310910,\\
a(4,3) &= 0.06436937873639827, & a(10,8) &= 0.15325100305906427,\\
a(5,1) &= -0.00567147901469362, & a(10,9) &= 0.24262378483127705,\\
a(5,2) &= -0.03126144607261765, & a(11,1) &= 0.07966502081315342,\\
a(5,3) &= 0.15035962158451534, & a(11,2) &= -0.02983144372465646,\\
a(5,4) &= 0.16649493718305642, & a(11,3) &= 0.05133037224950475,\\
a(6,1) &= 0.02813946441252450, & a(11,4) &= 0.08567900416856000,\\
a(6,2) &= -0.00963232687884552, & a(11,5) &= 0.14295159044950844,\\
a(6,3) &= 0.04926751938623165, & a(11,6) &= 0.12606430058207663,\\
a(6,4) &= 0.12957253836982637, & a(11,7) &= 0.19223208176070539,\\
a(6,5) &= 0.15700563148635182, & a(11,8) &= -0.04870211419120333,\\
a(7,1) &= -0.01236921160237027, & a(11,9) &= 0.04212694774825574,\\
a(7,2) &= -0.04382582552853110, & a(11,10) &= 0.35848424014409542,\\
a(7,3) &= 0.01910890570394344, & b(1) &= 0.04450519722392218,\\
a(7,4) &= 0.17490844953178922, & b(2) &= -0.00000731315924828,\\
a(7,5) &= 0.13265904888988556, & b(3) &= 0.07852539726325724,\\
a(7,6) &= 0.27195471716666654, & b(4) &= 0.07573654323801357,\\
a(8,1) &= 0.09312373664702384, & b(5) &= 0.14542767851790367,\\
a(8,2) &= -0.03626143583794692, & b(6) &= 0.12495148612302253,\\
a(8,3) &= -0.00938233908385715, & b(7) &= 0.05555387130939724,\\
a(8,4) &= 0.12483256234729837, & b(8) &= 0.13135285088327981,\\
a(8,5) &= 0.11791821078281690, & b(9) &= 0.14935614713291244,\\
a(8,6) &= 0.14065935132405036, & b(10) &= 0.12911785060454295,\\
a(8,7) &= 0.15879109038617302, & b(11) &= 0.06548029086299665,\\
a(9,1) &= 0.07964415711310319, & c[2] &= 0.09014195559593239,\\
a(9,2) &= 0.01677189011785914, & c[3] &= 0.19104583338281745,\\
a(9,3) &= 0.00643478713433633, & c[4] &= 0.07958675394687324,\\
a(9,4) &= 0.08881049508330123, & c[5] &= 0.27992163368026048,\\
a(9,5) &= 0.07573713478285111, & c[6] &= 0.35435282677608881,\\
a(9,6) &= 0.09878973114127274, & c[7] &= 0.54243608416138340,\\
a(9,7) &= 0.15586909199658420, & c[8] &= 0.58968117656555842,\\
a(9,8) &= 0.24972615074826192, & c[9] &= 0.77178343811756986,\\
a(10,1) &= 0.04053213982727640, & c[10] &= 0.81819979031148928,\\
a(10,2) &= 0.03221300276164501, & c[11] &= 1,
\end{aligned}
$$

5.3.6. Двенадцатистадийных методы 6-го порядка

Общей вид матриций в следующем образом:

$$A_{612} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0 & 0 & 0 \\ a_{101} & a_{102} & a_{103} & a_{104} & a_{105} & a_{106} & a_{107} & a_{108} & a_{109} & 0 & 0 & 0 & 0 \\ a_{111} & a_{112} & a_{113} & a_{114} & a_{115} & a_{116} & a_{117} & a_{118} & a_{119} & a_{1110} & 0 & 0 & 0 \\ a_{121} & a_{122} & a_{123} & a_{124} & a_{125} & a_{126} & a_{127} & a_{128} & a_{129} & a_{1210} & a_{1211} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & b_{12} & 0 \end{pmatrix}.$$

Рунге-Кутта с помощью программы RK_EXE. Процесс медленно. Мы вычислели их погрешности с помощью программы RK_USE.EXE. Мы заметили после вычислении погрешности , что метод хороший. Из них мы выбрали лучший метод с погрешности $2.692 \cdot 10^{-5}$. Матрица данного метода в следующем виде:

$$
\begin{aligned}
a(2,1) &= 0.04330691166697415,\\
a(3,1) &= -0.04873549520792616,\\
a(3,2) &= 0.17058448646865899,\\
a(4,1) &= 0.01873770377152976,\\
a(4,2) &= -0.00097883046025127,\\
a(4,3) &= 0.20566570422818963,\\
a(5,1) &= 0.17073761872840157,\\
a(5,2) &= -0.05136409212609066,\\
a(5,3) &= -0.05497716187786078,\\
a(5,4) &= 0.10132295225398727,\\
a(6,1) &= 0.09918905350073983,\\
a(6,2) &= -0.06383983778694886,\\
a(6,3) &= 0.01190830681698989,\\
a(6,4) &= 0.10246400656306661,\\
a(6,5) &= 0.16101939790916064,\\
a(7,1) &= 0.04288783973546667,\\
a(7,2) &= -0.01661916609638866,\\
a(7,3) &= 0.02329969658951096,\\
a(7,4) &= 0.07032726060453202,\\
a(7,5) &= 0.13586310794703413,\\
a(7,6) &= 0.19033170171353499,\\
a(8,1) &= 0.08096082352721342,\\
a(8,2) &= 0.01008136538418191,\\
a(8,3) &= -0.00994493150077134,\\
a(8,4) &= 0.03582899059574243,\\
a(8,5) &= 0.11879376656654394,\\
a(8,6) &= 0.09293161293858524,\\
a(8,7) &= 0.24927319583868736,\\
a(9,1) &= 0.08664449417394166,\\
a(9,2) &= 0.01600564420315607,\\
a(9,3) &= 0.05146732694217347,\\
a(9,4) &= 0.07261464976698643,\\
a(9,5) &= 0.04075929473959460,\\
a(9,6) &= 0.12970379423588520,\\
a(9,7) &= 0.04098386924392486,\\
a(9,8) &= 0.20742844808718787,\\
a(10,1) &= 0.08525871518799866,\\
a(10,2) &= 0.01924556749528086,\\
a(10,3) &= 0.00263049252781957,\\
a(10,4) &= 0.05925986416933974,\\
a(10,5) &= 0.09047650418418458,\\
a(10,6) &= 0.06794446782967186,\\
a(10,7) &= 0.11253967204300437,\\
a(10,8) &= 0.17272101089146837,\\
a(10,9) &= 0.20906324490650304,
\end{aligned}
$$

$$
\begin{aligned}
a(11,1) &= -0.02338966858762020,\\
a(11,2) &= 0.03689714135696114,\\
a(11,3) &= 0.13907053999033206,\\
a(11,4) &= 0.13341935183157350,\\
a(11,5) &= -0.03138856727473449,\\
a(11,6) &= 0.23073135341960603,\\
a(11,7) &= 0.10698738724682360,\\
a(11,8) &= -0.02341763493344499,\\
a(11,9) &= 0.10649766423334796,\\
a(11,10) &= 0.13547088046845060,\\
a(12,1) &= 0.11982112139561423,\\
a(12,2) &= -0.07205765706208924,\\
a(12,3) &= 0.00873929698088060,\\
a(12,4) &= 0.12916518464965203,\\
a(12,5) &= 0.11725103070463138,\\
a(12,6) &= 0.10248667238852565,\\
a(12,7) &= 0.14324804075749441,\\
a(12,8) &= -0.00074953596664264,\\
a(12,9) &= 0.04864359575934454,\\
a(12,10) &= -0.01324133648994775,\\
a(12,11) &= 0.41669358688253679,\\
b(1) &= 0.05807104239262850,\\
b(2) &= -0.00031559990511021,\\
b(3) &= 0.04948149133504334,\\
b(4) &= 0.09173887411307136,\\
b(5) &= 0.07392955922061548,\\
b(6) &= 0.12684501691151995,\\
b(7) &= 0.10905009373513154,\\
b(8) &= 0.08547203164740304,\\
b(9) &= 0.10780154310729406,\\
b(10) &= 0.09829028574948127,\\
b(11) &= 0.13731399987756144,\\
b(12) &= 0.06232166181536022,\\
c[2] &= 0.04330691166697415,\\
c[3] &= 0.12184899126073283,\\
c[4] &= 0.22342457753946812,\\
c[5] &= 0.1657193169784374,\\
c[6] &= 0.31074092700300811,\\
c[7] &= 0.44609044049369011,\\
c[8] &= 0.57792482335018296,\\
c[9] &= 0.64560752139285017,\\
c[10] &= 0.81913953923527105,\\
c[11] &= 0.81087844775129522,\\
c[12] &= 1,
\end{aligned}
$$

Из всех методов 6-го порядка, восимистадийный метод с погрешности

$4.413 \cdot 10^{-7}$ лучший, но метод Дорман-Принса (5,7)

остает наилучшим ($3.98 \cdot 10^{-10}$).

5.4. Методы 7-го порядка

5.4.1. Девятистадийных методы7-го порядка

Общей вид матриций в следующем образом:

$$A_{79} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 \\ a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & 0 \end{pmatrix}.$$

RK_EXE . Процесс нахождения очень медленно и долго. Все методы хорошие, но самая лучшая со следующем матрицей с погрешности $1.766 \cdot 10^{-7}$. В этом методе, $b(2) = b(3) = 0$. Выпишем матрицу этого метода:

$$
\begin{aligned}
a(2,1) &= 0.22985026943498231, & a(8,7) &= 0.02142189809806867,\\
a(3,1) &= 0.18243102770123480, & a(9,1) &= 1.43734132820065691,\\
a(3,2) &= 0.05076337397718929, & a(9,2) &= -0.00028973237831293,\\
a(4,1) &= 0.08935637712508556, & a(9,3) &= 0.00049307602619974,\\
a(4,2) &= -0.38091551394657206, & a(9,4) &= 0.55222592819067030,\\
a(4,3) &= 0.64825446513173052, & a(9,5) &= -2.46720863018497831,\\
a(5,1) &= 0.06704270415147081, & a(9,6) &= -4.53755768879579501,\\
a(5,2) &= -0.06743284161761258, & a(9,7) &= 0.96881187590935665,\\
a(5,3) &= 0.11475941271656763, & a(9,8) &= 5.04618384303220266,\\
a(5,4) &= -0.01875856236573176, & b(1) &= 0.00512996183005472,\\
a(6,1) &= -0.34840491857714152, & b(2) &= 0,\\
a(6,2) &= 1.22342368691420827, & b(3) &= 0,\\
a(6,3) &= -2.08206239639087028, & b(4) &= -0.09928823843774115,\\
a(6,4) &= 0.64879692534185479, & b(5) &= 0.22587713004637444,\\
a(6,5) &= 1.06628164136259889, & b(6) &= -0.07542645448767416,\\
a(7,1) &= -0.44206307950702273, & b(7) &= 0.33749139676525464,\\
a(7,2) &= 0.04847240217159485, & b(8) &= 0.54305144498407074,\\
a(7,3) &= -0.08249191748017051, & b(9) &= 0.06316475929966077,\\
a(7,4) &= -0.84524006988306451, & c[2] &= 0.22985026943498231,\\
a(7,5) &= 1.09742248918109395, & c[3] &= 0.23319440167842409,\\
a(7,6) &= 1.00810812166684153, & c[4] &= 0.35669532831024401,\\
a(8,1) &= 0.05705316722653135, & c[5] &= 0.09561071288469410,\\
a(8,2) &= 0.09823912643546461, & c[6] &= 0.50803493865065016,\\
a(8,3) &= -0.16718655457903713, & c[7] &= 0.78420794614927258,\\
a(8,4) &= 0.44135971007644487, & c[8] &= 0.41305563078592701,\\
a(8,5) &= 0.12922659816507596, & c[9] &= 1,\\
a(8,6) &= -0.16705831463662131, & &
\end{aligned}
$$

5.4.2. Десятистадийных методы 7-го порядка

Общей вид матриций в следующем образом:

$$
A_{710} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0\\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0\\
a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0\\
a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0\\
a_{10,1} & a_{10,2} & a_{10,3} & a_{10,4} & a_{10,5} & a_{10,6} & a_{10,7} & a_{10,8} & a_{10,9} & 0 & 0\\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & 0
\end{pmatrix}.
$$

Было найдено более 280 десятистадийных методов порядка 7 с помощью программы RK_EXE. Процесс требовал несколько дней. После вычислении погрешности, мы выбрали наилучший метод ($1.391 \cdot 10^{-6}$). Оказывается,что $b(2) = b(3) = 0$. Выпишем матрицу этого метода:

$$
\begin{array}{lcllcl}
a(2,1) & = & 0.02363631962989042, & a(9,5) & = & 0.00374848876974525,\\
a(3,1) & = & -1.01256498059558173, & a(9,6) & = & 0.07860455823523571,\\
a(3,2) & = & 1.25839514976378262, & a(9,7) & = & 0.27425876037782656,\\
a(4,1) & = & 0.74888257751564213, & a(9,8) & = & 0.35318898480071248,\\
a(4,2) & = & -0.77177975310119136, & a(10,1) & = & -0.48341313746505995,\\
a(4,3) & = & 0.45636006927761901, & a(10,2) & = & -0.04634574445731584,\\
a(5,1) & = & 0.17113192816340021, & a(10,3) & = & 0.02740464111198193,\\
a(5,2) & = & -0.11419876849729158, & a(10,4) & = & 0.72235152208713641,\\
a(5,3) & = & 0.06752672338635635, & a(10,5) & = & 1.98019320566702176,\\
a(5,4) & = & -0.01931746172650786, & a(10,6) & = & 1.63281846977659204,\\
a(6,1) & = & -0.36216326359292175, & a(10,7) & = & -1.04497060248723293,\\
a(6,2) & = & 0.27113594199560580, & a(10,8) & = & -2.73609254228233719,\\
a(6,3) & = & -0.16032503674212798, & a(10,9) & = & 0.94805418804921378,\\
a(6,4) & = & 0.11487696899760063, & b(1) & = & 0.01069679634744849,\\
a(6,5) & = & 0.51242270557247175, & b(2) & = & 0,\\
a(7,1) & = & 0.88058032316516293, & b(3) & = & 0,\\
a(7,2) & = & -0.61457934313490503, & b(4) & = & 0.11747032518071467,\\
a(7,3) & = & 0.36340610191267755, & b(5) & = & 0.24432089477304292,\\
a(7,4) & = & 0.58274815322920809, & b(6) & = & 0.31207598142752791,\\
a(7,5) & = & -0.39020551595745454, & b(7) & = & 0.05437649479318292,\\
a(7,6) & = & -0.13603227854352214, & b(8) & = & -0.08735384878776016,\\
a(8,1) & = & 0.20547629008766217, & b(9) & = & 0.28734239085411220,\\
a(8,2) & = & -0.41405780745329551, & b(10) & = & 0.06107096541173106,\\
a(8,3) & = & 0.24483597676028096, & c[2] & = & 0.02363631962989042,\\
a(8,4) & = & 0.66897927359209932, & c[3] & = & 0.24583016916820089,\\
a(8,5) & = & 0.48164811631499533, & c[4] & = & 0.43346289369206977,\\
a(8,6) & = & -0.91403631731241855, & c[5] & = & 0.10514242132595711,\\
a(8,7) & = & -0.02392458125314472, & c[6] & = & 0.37594731623062845,\\
a(9,1) & = & -0.02253253692460101, & c[7] & = & 0.68591744067116687,\\
a(9,2) & = & 0.11841919764524535, & c[8] & = & 0.24892095073617900,\\
a(9,3) & = & -0.07002229978697533, & c[9] & = & 0.79850315731452156,\\
a(9,4) & = & 0.06283800419733254, & c[10] & = & 1,
\end{array}
$$

5.4.3. Одйннадцатистадийных методы 7-го порядка

$$A_{711} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0 & 0 \\ a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0 & 0 \\ a_{10,1} & a_{10,2} & a_{10,3} & a_{10,4} & a_{10,5} & a_{10,6} & a_{10,7} & a_{10,8} & a_{10,9} & 0 & 0 & 0 \\ a_{11,1} & a_{11,2} & a_{11,3} & a_{11,4} & a_{11,5} & a_{11,6} & a_{11,7} & a_{11,8} & a_{11,9} & a_{11,10} & 0 & \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & 0 \end{pmatrix}.$$

программы RK_EXE. Процесс требовал несколько дней . Оказывается,что $b(2) = b(3) = 0$. После вычислении погрешности, мы выбрали наилучший метод $(8.196 \cdot 10^{-6})$. Выпишем матрицу этого метода:

$$
\begin{aligned}
a(2,1) &= 0.16221743059490618, \\
a(3,1) &= 0.03691774285116069, \\
a(3,2) &= 0.24503728422645302, \\
a(4,1) &= 0.02791795080089209, \\
a(4,2) &= 0.00995275764961362, \\
a(4,3) &= -0.00364938030709960, \\
a(5,1) &= -0.53607074337137527, \\
a(5,2) &= -0.34248566923255612, \\
a(5,3) &= 0.32019999529559934, \\
a(5,4) &= 0.92248106732945037, \\
a(6,1) &= -0.20522706490784358, \\
a(6,2) &= 0.18165676172812478, \\
a(6,3) &= -0.11097818010602297, \\
a(6,4) &= 0.33190671382150506, \\
a(6,5) &= 0.25582493599527398, \\
a(7,1) &= -0.52902404725512037, \\
a(7,2) &= -0.34239555450677751, \\
a(7,3) &= -0.13523880199139245, \\
a(7,4) &= 0.97536133158922076, \\
a(7,5) &= 0.50611283612453406, \\
a(7,6) &= 0.13003488173603661, \\
a(8,1) &= -0.39440678971702382, \\
a(8,2) &= 0.45241934690116848, \\
a(8,3) &= 0.08301210028562699, \\
a(8,4) &= 0.35361866284296812, \\
a(8,5) &= 0.23376391998840009, \\
a(8,6) &= -0.42586751043562042, \\
a(8,7) &= 0.34592978288384969, \\
a(9,1) &= -0.39456507859837419, \\
a(9,2) &= -0.08410065386181999, \\
a(9,3) &= -0.10507642265211110, \\
a(9,4) &= 0.69263118002606597, \\
a(9,5) &= 0.32138192867400315, \\
a(9,6) &= 0.09864870643606423, \\
a(9,7) &= 0.14092744181452953, \\
a(9,8) &= 0.05965748593923754, \\
a(10,1) &= -0.07101901171453739, \\
a(10,2) &= 0.56615227386101877,
\end{aligned}
$$

$$
\begin{aligned}
a(10,3) &= 0.42532474306397981, \\
a(10,4) &= -0.19912897358937518, \\
a(10,5) &= -0.05742464819988089, \\
a(10,6) &= 0.09601611575082472, \\
a(10,7) &= -0.18775169309444067, \\
a(10,8) &= -0.46207938862000497, \\
a(10,9) &= 0.63506918198514470, \\
a(11,1) &= 0.00637168104840252, \\
a(11,2) &= 0.00764310997485652, \\
a(11,3) &= -0.14401977673173250, \\
a(11,4) &= 0.17141531778645347, \\
a(11,5) &= 0.06763422144064619, \\
a(11,6) &= 1.08516159438253351, \\
a(11,7) &= -0.65056557051834659, \\
a(11,8) &= -0.87900861192291810, \\
a(11,9) &= 0.92364457111181919, \\
a(11,10) &= 0.41172346342828578, \\
b(1) &= -0.16714450360219938, \\
b(2) &= 0, \\
b(3) &= 0, \\
b(4) &= 0.35690820311090211, \\
b(5) &= 0.16500285565605127, \\
b(6) &= 0.39120398151613437, \\
b(7) &= -0.19895615380085314, \\
b(8) &= -0.23987944198719855, \\
b(9) &= 0.50838023383254862, \\
b(10) &= 0.11395249782273605, \\
b(11) &= 0.07053232745187866, \\
c[2] &= 0.16221743059490618, \\
c[3] &= 0.28195502707761371, \\
c[4] &= 0.03422132814340611, \\
c[5] &= 0.36412465002111831, \\
c[6] &= 0.45318316653103727, \\
c[7] &= 0.60485064569650109, \\
c[8] &= 0.64846951274936912, \\
c[9] &= 0.72950458777759513, \\
c[10] &= 0.74515859944272891, \\
c[11] &= 1,
\end{aligned}
$$

5.4.4. Двенадцатистадийных методы 7-го порядка

RK_EXE. Процесс требовал несколько дней. После вычислении погрешности, мы выбрали лучший метод ($2.318{\cdot}10^{-5}$). Особенности : $b(2) = b(3) = 0$.

Матрица данного метода в следующем виде:

$$
\begin{aligned}
a(2,1) &= 0.07820841416585512,\\
a(3,1) &= 0.06778432013295680,\\
a(3,2) &= 0.17867569897258279,\\
a(4,1) &= -0.01155828957601195,\\
a(4,2) &= 0.17611970399072784,\\
a(4,3) &= -0.00280681332408988,\\
a(5,1) &= 0.18139844629872272,\\
a(5,2) &= -0.28311944608303641,\\
a(5,3) &= 0.00941816502202859,\\
a(5,4) &= 0.38205401344075025,\\
a(6,1) &= 0.26733278335077176,\\
a(6,2) &= -0.16870335508854032,\\
a(6,3) &= -0.08401830323896520,\\
a(6,4) &= -0.12518619846691805,\\
a(6,5) &= 0.59965769747404004,\\
a(7,1) &= -0.07390024480397082,\\
a(7,2) &= 0.17096031093001813,\\
a(7,3) &= -0.02180862988931352,\\
a(7,4) &= 0.31824263884974967,\\
a(7,5) &= -0.11082872457065881,\\
a(7,6) &= 0.33843418301344933,\\
a(8,1) &= -0.10486290394564656,\\
a(8,2) &= 0.18803078742230275,\\
a(8,3) &= 0.10987805843078142,\\
a(8,4) &= 0.13185309225820543,\\
a(8,5) &= 0.18900582583024771,\\
a(8,6) &= -0.08145681036238983,\\
a(8,7) &= 0.25689466737857346,\\
a(9,1) &= 0.19491823825959864,\\
a(9,2) &= -0.05073306990715174,\\
a(9,3) &= -0.03402764849658864,\\
a(9,4) &= 0.10324258865035797,\\
a(9,5) &= 0.08871655182727337,\\
a(9,6) &= 0.11018353869429062,\\
a(9,7) &= 0.10708577787119760,\\
a(9,8) &= 0.41495883206095946,\\
a(10,1) &= 0.03875860099622808,\\
a(10,2) &= 0.15353925885337271,\\
a(10,3) &= 0.19552439431622689,\\
a(10,4) &= -0.04114707263646592,\\
a(10,5) &= 0.02425886518836054,\\
a(10,6) &= 0.01912608352807640,\\
a(10,7) &= 0.25440163827547680,\\
a(10,8) &= 0.24848185514964763,\\
a(10,9) &= -0.01161422850800933,
\end{aligned}
$$

$$
\begin{aligned}
a(11,1) &= 0.03789325042346607,\\
a(11,2) &= 0.09888591894947300,\\
a(11,3) &= 0.11721693871179105,\\
a(11,4) &= 0.07622925568065288,\\
a(11,5) &= -0.02731083634298177,\\
a(11,6) &= 0.33980941677452689,\\
a(11,7) &= -0.03406867008376817,\\
a(11,8) &= 0.16394732845478946,\\
a(11,9) &= -0.00643412410722392,\\
a(11,10) &= 0.22654393039483279,\\
a(12,1) &= 0.30158560647415514,\\
a(12,2) &= -0.04928776836682296,\\
a(12,3) &= -0.12936482399221231,\\
a(12,4) &= -0.01348361542271086,\\
a(12,5) &= 0.15511055900430953,\\
a(12,6) &= 0.00425395495531349,\\
a(12,7) &= 0.33770398249772813,\\
a(12,8) &= 0.28557488745875383,\\
a(12,9) &= -0.19440744891898313,\\
a(12,10) &= 0.30686613985252987,\\
a(12,11) &= -0.00455147354206072,\\
b(1) &= 0.05546662889997590,\\
b(2) &= 0,\\
b(3) &= 0,\\
b(4) &= 0.18314048466761082,\\
b(5) &= 0.16634700484468125,\\
b(6) &= 0.11522335342208416,\\
b(7) &= 0.07340667760237407,\\
b(8) &= 0.26198728144070246,\\
b(9) &= 0.32993364985024285,\\
b(10) &= -0.14613468031887455,\\
b(11) &= 0.06549182522163546,\\
b(12) &= -0.10486222563043242,\\
c[2] &= 0.07820841416585512,\\
c[3] &= 0.24646001910553958,\\
c[4] &= 0.16175460109062601,\\
c[5] &= 0.28975117867846514,\\
c[6] &= 0.48908262403038821,\\
c[7] &= 0.62109953352927398,\\
c[8] &= 0.68934271701207438,\\
c[9] &= 0.93434480895993728,\\
c[10] &= 0.88132939516291381,\\
c[11] &= 0.99271240885555829,\\
c[12] &= 1,
\end{aligned}
$$

5.4.5. Тринадцатистадийных методы 7-го порядка

Общей вид матриций в следующем образом:

$$A_{713} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{10,1} & a_{10,2} & a_{10,3} & a_{10,4} & a_{10,5} & a_{10,6} & a_{10,7} & a_{10,8} & a_{10,9} & 0 & 0 & 0 & 0 & 0 \\
a_{11,1} & a_{11,2} & a_{11,3} & a_{11,4} & a_{11,5} & a_{11,6} & a_{11,7} & a_{11,8} & a_{11,9} & a_{11,10} & 0 & 0 & 0 & 0 \\
a_{12,1} & a_{12,2} & a_{12,3} & a_{12,4} & a_{12,5} & a_{12,6} & a_{12,7} & a_{12,8} & a_{12,9} & a_{12,10} & a_{12,11} & 0 & 0 & 0 \\
a_{13,1} & a_{13,2} & a_{13,3} & a_{13,4} & a_{13,5} & a_{13,6} & a_{13,7} & a_{13,8} & a_{13,9} & a_{13,10} & a_{13,11} & a_{13,12} & 0 & 0 \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & 0
\end{pmatrix}$$

а) Методы Дорман-Принса.

Прекрасные численые результаты дает метод $(7, 13)$, который предложил Приис и Дорман (1981), приложивщие много усилий , чтобы минимизировать коэффициенты погрешности для аппроксимации 8-го порядка .После вычислении погрешности с помощью программы RK_use.exe получили: $4.703 \cdot 10^{-8}$. Этот метод очень хороший.

Заметим, что:

$$\begin{aligned}
&a(4,2) = a(5,2) = a(6,2) = a(6,3) = a(7,2) = \\
&a(7,3) = a(8,2) = a(8,3) = a(9,2) = a(9,3) = \\
&a(10,2) = a(10,3) = a(11,2) = a(11,3) = \\
&a(12,2) = a(12,3) = a(13,2) = a(13,3) = \\
&a(13,12) = b(2) = b(3) = b(4) = b(5) = 0
\end{aligned}$$

Коэффициенты не являются точными рационнальными числами. Матрица метода Дорман-Принса выписывает в следующем виде:

$$
\begin{aligned}
a(2,1) &= 1/18,\\
a(3,1) &= 1/48,\\
a(3,2) &= 1/16,\\
a(4,1) &= 1/32,\\
a(4,2) &= 0,\\
a(4,3) &= 3/32,\\
a(5,1) &= 5/16,\\
a(5,2) &= 0,\\
a(5,3) &= -75/64,\\
a(5,4) &= 75/64,\\
a(6,1) &= 3/80,\\
a(6,2) &= 0,\\
a(6,3) &= 0,\\
a(6,4) &= 3/16,\\
a(6,5) &= 3/20,\\
a(7,1) &= 29443841/614563906,\\
a(7,2) &= 0,\\
a(7,3) &= 0,\\
a(7,4) &= 77736538/692538347,\\
a(7,5) &= -28693883/1125000000,\\
a(7,6) &= 23124283/1800000000,\\
a(8,1) &= 16016141/946692911,\\
a(8,2) &= 0,\\
a(8,3) &= 0,\\
a(8,4) &= 61564180/158732637,\\
a(8,5) &= 22789713/633445777,\\
a(8,6) &= 545815736/2771057229,\\
a(8,7) &= -180193667/1043307555,\\
a(9,1) &= 39632708/573591083,\\
a(9,2) &= 0,\\
a(9,3) &= 0,\\
a(9,4) &= -433636366/683701615,\\
a(9,5) &= -421739975/2616292301,\\
a(9,6) &= 100302831/723423059,\\
a(9,7) &= 790204164/839813087,\\
a(9,8) &= 800635310/3783071287,\\
a(10,1) &= 246121993/1340847787,\\
a(10,2) &= 0,\\
a(10,3) &= 0,\\
a(10,4) &= -37695042795/15268766246,\\
a(10,5) &= -309121744/1061227803,\\
a(10,6) &= -12992083/490766935,\\
a(10,7) &= 6005943493/2108947869,\\
a(10,8) &= 393006217/1396673457,\\
a(10,9) &= 123872331/1001029789,\\
a(11,1) &= -1028468189/846180014,\\
a(11,2) &= 0,
\end{aligned}
$$

$$
\begin{aligned}
a(11,8) &= 15336726248/1032824649,\\
a(11,9) &= -45442868181/3398467676,\\
a(11,10) &= 3065993473/597172653,\\
a(12,1) &= 185892177/718116043,\\
a(12,2) &= 0,\\
a(12,3) &= 0,\\
a(12,4) &= -3185094517/667107341,\\
a(12,5) &= -477755414/1098053517,\\
a(12,6) &= -703635378/230739211,\\
a(12,7) &= 5731566787/1027545527,\\
a(12,8) &= 5232866602/830066563,\\
a(12,9) &= -4093664535/808688257,\\
a(12,10) &= 3962137247/1805957418,\\
a(12,11) &= 65686358/487910083,\\
a(13,1) &= 403863854/491063109,\\
a(13,2) &= 0,\\
a(13,3) &= 0,\\
a(13,4) &= -5068492393/434740067,\\
a(13,5) &= -411421997/543043805,\\
a(13,6) &= 652783627/914296604,\\
a(13,7) &= 11173962825/925320556,\\
a(13,8) &= -13158990841/6184727034,\\
a(13,9) &= 3936647629/1978049680,\\
a(13,10) &= -160528059/685178525,\\
a(13,11) &= 248638103/1413531061,\\
a(13,12) &= 0,\\
b(1) &= 14005451/335480064,\\
b(2) &= 0,\\
b(3) &= 0,\\
b(4) &= 0,\\
b(5) &= 0,\\
b(6) &= -59238493/1068277825,\\
b(7) &= 181606767/758867731,\\
b(8) &= 561292985/797845752,\\
b(9) &= -1041891430/1371343529,\\
b(10) &= 760417239/1151165299,\\
b(11) &= 118820643/751138087,\\
b(12) &= -528747749/2220607170,\\
b(13) &= 1/4,\\
c[2] &= 1/18,\\
c[3] &= 1/12,\\
c[4] &= 1/8,\\
c[5] &= 5/16,\\
c[6] &= 3/8,\\
c[7] &= 59/400,\\
c[8] &= 93/200,\\
c[9] &= 54900023248/9719169821,
\end{aligned}
$$

$$
\begin{aligned}
a(11,3) &= 0, & c[10] &= 13/20,\\
a(11,4) &= 8478235783/508512852, & c[11] &= 1201146811/1299019798,\\
a(11,5) &= 1311729495/1432422823, & c[12] &= 1,\\
a(11,6) &= -10304129995/1701304382, & c[13] &= 1,\\
a(11,7) &= -48777925059/3047939560,
\end{aligned}
$$

b) Метод Фельберга .

Погрешность метода Фельберга $(7,13)$ после вычислении дает: $5.65 \cdot 10^{-6}$. Метод Фельберга недооценивает погрешность. Особенности этого метода :

$a(4,2) = a(5,2) = a(6,2) = a(6,3) = a(7,2) = a(7,3) =$
$a(8,2) = a(8,3) = a(9,2) = a(9,3) = a(10,2) = a(10,3) =$
$a(11,2) = a(11,3) = a(12,2) = a(12,3) = a(12,4) =$
$a(12,5) = a(12,11) = a(13,2) = a(13,3) = a(13,11) =$
$b(2) = b(3) = b(4) = b(5) = b(12) = b(13) = c[12] = 0.$

Мартица метода Фельберга в слеующем образом:

$$
\begin{aligned}
a(2,1) &= 2/27, & a(11,8) &= 45/82,\\
a(3,1) &= 1/36, & a(11,9) &= 45/164,\\
a(3,2) &= 1/12, & a(11,10) &= 18/41,\\
a(4,1) &= 1/24, & a(12,1) &= 3/205,\\
a(4,2) &= 0, & a(12,2) &= 0,\\
a(4,3) &= 1/8, & a(12,3) &= 0,\\
a(5,1) &= 5/12, & a(12,4) &= 0,\\
a(5,2) &= 0, & a(12,5) &= 0,\\
a(5,3) &= -25/16, & a(12,6) &= -6/41,\\
a(5,4) &= 25/16, & a(12,7) &= -3/205,\\
a(6,1) &= 1/20, & a(12,8) &= -3/41,\\
a(6,2) &= 0, & a(12,9) &= 3/41,\\
a(6,3) &= 0, & a(12,10) &= 6/41,\\
a(6,4) &= 1/4, & a(12,11) &= 0,\\
a(6,5) &= 1/5, & a(13,1) &= -1777/4100,\\
a(7,1) &= -25/108, & a(13,2) &= 0,\\
a(7,2) &= 0, & a(13,3) &= 0,\\
a(7,3) &= 0, & a(13,4) &= -341/164,\\
a(7,4) &= 125/108, & a(13,5) &= 4496/1025,\\
a(7,5) &= -65/27, & a(13,6) &= -289/82,\\
a(7,6) &= 125/54, & a(13,7) &= 2193/4100,\\
a(8,1) &= 31/300, & a(13,8) &= 51/82,\\
a(8,2) &= 0, & a(13,9) &= 33/164,\\
a(8,3) &= 0, & a(13,10) &= 12/41,\\
a(8,4) &= 0, & a(13,11) &= 0,\\
a(8,5) &= 61/225, & a(13,12) &= 1,\\
a(8,6) &= -2/9, & b(1) &= 41/840,\\
a(8,7) &= 13/900, & b(2) &= 0,\\
a(9,1) &= 2, & b(3) &= 0,
\end{aligned}
$$

$$
\begin{aligned}
a(9,2) &= 0, & b(4) &= 0,\\
a(9,3) &= 0, & b(5) &= 0,\\
a(9,4) &= -53/6, & b(6) &= 34/105,\\
a(9,5) &= 704/45, & b(7) &= 9/35,\\
a(9,6) &= -107/9, & b(8) &= 9/35,\\
a(9,7) &= 67/90, & b(9) &= 9/280,\\
a(9,8) &= 3, & b(10) &= 9/280,\\
a(10,1) &= -91/108, & b(11) &= 41/840,\\
a(10,2) &= 0, & b(12) &= 0,\\
a(10,3) &= 0, & b(13) &= 0,\\
a(10,4) &= 23/108, & c[2] &= 2/27,\\
a(10,5) &= -976/135, & c[3] &= 1/9,\\
a(10,6) &= 311/54, & c[4] &= 1/6,\\
a(10,7) &= -19/60, & c[5] &= 5/12,\\
a(10,8) &= 17/6, & c[6] &= 1/2,\\
a(10,9) &= -1/12, & c[7] &= 5/6,\\
a(11,1) &= 2383/4100, & c[8] &= 1/6,\\
a(11,2) &= 0, & c[9] &= 2/3,\\
a(11,3) &= 0, & c[10] &= 1/3,\\
a(11,4) &= -341/164, & c[11] &= 1,\\
a(11,5) &= 4496/1025, & c[12] &= 0,\\
a(11,6) &= -301/82, & c[13] &= 1,\\
a(11,7) &= 2133/4100, & &
\end{aligned}
$$

c) Было найдено несколько тринадцатистадийных методов порядка 7 с помощью программы RK_EXE.Процесс решения очень долго по времени.Мы получаем 85 уравнений от 91 переменных.С помощью программы RK_USE.EXE, мы вычисли погрешности все методы. Мы получили у найлучшего метода , погрешность $4.988 \cdot 10^{-6}$. $b(2) = b(8) = 0$.

$a(2,1) = 0.05411044038043938,$
$a(3,1) = -0.00463306858810390,$
$a(3,2) = 0.11730402912834176,$
$a(4,1) = 0.15556103614499486,$
$a(4,2) = -0.26685400658639877,$
$a(4,3) = 0.31827480778663624,$
$a(5,1) = 0.09578658836471725,$
$a(5,2) = -0.06583546684448628,$
$a(5,3) = 0.08015890533209387,$
$a(5,4) = 0.18741638054979606,$
$a(6,1) = 0.04823847663925234,$
$a(6,2) = 0.13669387873214139,$
$a(6,3) = -0.09910288342357586,$
$a(6,4) = 0.06121316534027382,$
$a(6,5) = 0.28768782459685284,$
$a(7,1) = -0.07706918644212207,$
$a(7,2) = 0.12397432763852204,$
$a(7,3) = 0.19982759267581115,$
$a(7,4) = 0.06101549606468176,$
$a(7,5) = 0.00288248848849239,$
$a(7,6) = 0.18989396430920858,$
$a(8,1) = -0.06388275764134100,$
$a(8,2) = 0.34155185625172280,$
$a(8,3) = 0.09437954676492724,$
$a(8,4) = -0.35611863531047524,$
$a(8,5) = 0.18727151187666936,$
$a(8,6) = 0.30355397827466319,$
$a(8,7) = 0.19837488436406779,$
$a(9,1) = -0.02001527326961771,$
$a(9,2) = -0.07620389740405288,$
$a(9,3) = 0.30535752772422328,$
$a(9,4) = 0.18480244651862567,$
$a(9,5) = -0.00222689881554682,$
$a(9,6) = -0.06118678265638402,$
$a(9,7) = 0.25190210802870647,$
$a(9,8) = 0.02004604988417446,$
$a(10,1) = -0.03174796358938113,$
$a(10,2) = -0.02018561025383715,$
$a(10,3) = 0.30478666524050755,$
$a(10,4) = 0.16456812644174138,$
$a(10,5) = -0.09510660893278333,$
$a(10,6) = 0.06350998446908944,$
$a(10,7) = 0.02374212173315502,$
$a(10,8) = -0.07456743107282076,$
$a(10,9) = 0.37318470058978368,$
$a(11,1) = 0.08821736407682579,$
$a(11,8) = 0.22342419038323077,$
$a(11,9) = 0.27801941675910048,$
$a(11,10) = 0.07498749342797758,$
$a(12,1) = 0.11364197440321181,$
$a(12,2) = -0.03389765313094119,$
$a(12,3) = 0.10978359347195030,$
$a(12,4) = 0.01816488842510696,$
$a(12,5) = 0.04131099376830319,$
$a(12,6) = 0.32639834337717126,$
$a(12,7) = 0.00109927086644571,$
$a(12,8) = 0.03674071859083082,$
$a(12,9) = 0.05199945912514759,$
$a(12,10) = 0.17357065556606699,$
$a(12,11) = 0.02737797637248145,$
$a(13,1) = -0.05543267984098631,$
$a(13,2) = 0.13943643542890233,$
$a(13,3) = 0.01168913166188073,$
$a(13,4) = 0.35370548819358674,$
$a(13,5) = -0.10737318568495895,$
$a(13,6) = 0.05508334985828302,$
$a(13,7) = 0.15116990398678664,$
$a(13,8) = 0.02610967165490209,$
$a(13,9) = 0.25441594391216541,$
$a(13,10) = -0.26799384766206285,$
$a(13,11) = -0.14731876087211435,$
$a(13,12) = 0.58650854936361549,$
$b(1) = 0.03786352133603152,$
$b(2) = 0.00000000000029210,$
$b(3) = 0.13395062932605527,$
$b(4) = 0.10417260712077066,$
$b(5) = 0.06016872985205511,$
$b(6) = 0.14687340224522467,$
$b(7) = 0.07541702024181464,$
$b(8) = -0.00000000000000547,$
$b(9) = 0.11673741844883964,$
$b(10) = 0.07871454965825385,$
$b(11) = -0.02664356255192346,$
$b(12) = 0.22207914101556409,$
$b(13) = 0.05066654330702740,$
$c[2] = 0.05411044038043938,$
$c[3] = 0.11267096054023787,$
$c[4] = 0.20698183734523233,$
$c[5] = 0.29752640740212090,$
$c[6] = 0.43473046188494453,$
$c[7] = 0.50052468273459385,$
$c[8] = 0.70513038458023414,$

$$\begin{aligned}
a(11,2) &= 0.09125494951247488, & c[9] &= 0.60247528001012845,\\
a(11,3) &= 0.15935576648399132, & c[10] &= 0.70818398462545470,\\
a(11,4) &= -0.04727698942661623, & c[11] &= 0.94805293400902251,\\
a(11,5) &= -0.14292458077073771, & c[12] &= 0.86619022083577488,\\
a(11,6) &= 0.16633271642670204, & c[13] &= 1,\\
a(11,7) &= 0.05666260713607359, & &
\end{aligned}$$

Из всех методов порядка 7, метод Дорман-Принса (7,13) с погрешности $4.703 \cdot 10^{-8}$ оказался лучшим, но их метод (5,7) остает наилучшим ($3.98 \cdot 10^{-10}$).

5.5. Методы 8-го порядка

5.5.1. Одйннадцатистадийых методы 8-го порядка

При $p \geq 8$ не существует явных методов РК порядка p , имеющих $n = p + 2$ стадии (*Бутчер*(1985b).

Общей вид матриций в следующем образом:

$$A_{811} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 & 0 & 0 & 0\\
a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 & 0 & 0 & 0\\
a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 & 0 & 0 & 0\\
a_{10,1} & a_{10,2} & a_{10,3} & a_{10,4} & a_{10,5} & a_{10,6} & a_{10,7} & a_{10,8} & a_{10,9} & 0 & 0 & 0\\
a_{11,1} & a_{11,2} & a_{11,3} & a_{11,4} & a_{11,5} & a_{11,6} & a_{11,7} & a_{11,8} & a_{11,9} & a_{11,10} & 0 & \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & 0
\end{pmatrix}.$$

помощью программы RK_EXE . Процесс решение очень долго по времени(несколько дней). С помощью программы RK_USE.EXE, мы вычисли погрешности все методы. Из них , лучший результат с погрешностью $2.58 \cdot 10^{-7}$.Заметно, что $b(2) = b(3) = 0$. Выпишем матрицу этого метода:

$$
\begin{aligned}
a(2,1) &= 0.21088921906082289,\\
a(3,1) &= -0.01431529935431976,\\
a(3,2) &= 0.24805190070891197,\\
a(4,1) &= 0.08267537338020832,\\
a(4,2) &= 0.13265534878462005,\\
a(4,3) &= 0.11801823936449757,\\
a(5,1) &= 0.08557136250103330,\\
a(5,2) &= 0.08590250733743863,\\
a(5,3) &= 0.07642407762555100,\\
a(5,4) &= -0.04799004841226748,\\
a(6,1) &= 0.12545005590960485,\\
a(6,2) &= 0.27251341861477857,\\
a(6,3) &= 0.24244445597390993,\\
a(6,4) &= 0.23557526503409315,\\
a(6,5) &= -0.40209471694586855,\\
a(7,1) &= -0.21132330894573849,\\
a(7,2) &= -1.22972306532575887,\\
a(7,3) &= -1.09403618026207638,\\
a(7,4) &= -0.70905891569982867,\\
a(7,5) &= 3.26079992321533518,\\
a(7,6) &= 0.35889519601998398,\\
a(8,1) &= -0.46847035127864415,\\
a(8,2) &= -0.04309698017453703,\\
a(8,3) &= -0.03834168594576267,\\
a(8,4) &= -6.19352437937338905,\\
a(8,5) &= 3.58045984730560648,\\
a(8,6) &= 2.24527614212142061,\\
a(8,7) &= 1.76713314807738507,\\
a(9,1) &= 0.50433439537904409,\\
a(9,2) &= -0.20037808921422209,\\
a(9,3) &= -0.17826849435736052,\\
a(9,4) &= 4.89782280817203483,\\
a(9,5) &= -1.98533264086160678,\\
a(9,6) &= -0.82784420279683260,\\
a(9,7) &= -1.60635179375227440,\\
a(9,8) &= 0.15573678625514174,\\
a(10,1) &= 0.41801574317494329,\\
a(10,2) &= -0.19368652827771579,\\
a(10,3) &= -0.17231527613011114,\\
a(10,4) &= 3.92487898122021427,\\
a(10,5) &= -1.48227705147617985,\\
a(10,6) &= -0.55162740905495275,\\
a(10,7) &= -1.31553818960698970,\\
a(10,8) &= 0.14271183494902223,\\
a(10,9) &= 0.00698328436938212,\\
a(11,1) &= 0.06330450192372935,\\
a(11,2) &= 0.28056482582041543,\\
a(11,3) &= 0.24960747587112240,\\
a(11,4) &= -0.42668617754374346,\\
a(11,5) &= -0.12695686532737581,\\
a(11,6) &= 0.21618467999922244,\\
a(11,7) &= 0.41342163498667997,\\
a(11,8) &= -0.20917985057516428,\\
a(11,9) &= -2.90496404699593976,\\
a(11,10) &= 3.44470382184105371,\\
b(1) &= 0.05394301970200152,\\
b(2) &= 0,\\
b(3) &= 0,\\
b(4) &= -0.29011614011909030,\\
b(5) &= 0.38117032457844453,\\
b(6) &= 0.45947246262035963,\\
b(7) &= 0.07865622040764991,\\
b(8) &= 0.06409137855003040,\\
b(9) &= -0.61092284884668996,\\
b(10) &= 0.81122359130523627,\\
b(11) &= 0.05248199180205801,\\
c[2] &= 0.21088921906082289,\\
c[3] &= 0.23373660135459222,\\
c[4] &= 0.33334896152932595,\\
c[5] &= 0.19990789905175546,\\
c[6] &= 0.47388847858651796,\\
c[7] &= 0.37555364900191676,\\
c[8] &= 0.84943574073207926,\\
c[9] &= 0.75971876882392428,\\
c[10] &= 0.77714538916761266,\\
c[11] &= 1,
\end{aligned}
$$

5.5.2. Двенадцатистадийных методы 8-го порядка

Процесс нахождения решении очень медленно(несколько дней). После вычислении погрешности,нашли очень хорошая($7.316 \cdot 10^{-6}$).Заметим, что $b(2) = b(3) = 0$. Выпишем матрицу этого метода:

$$
\begin{aligned}
a(2,1) &= 0.17621190923742057,\\
a(3,1) &= 0.61870389592821341,\\
a(3,2) &= -0.23650531341624547,\\
a(4,1) &= 0.07176034148319476,\\
a(4,2) &= 0.07421144627722090,\\
a(4,3) &= -0.01004382267606310,\\
a(5,1) &= 0.00152079674340406,\\
a(5,2) &= -0.21634650457771569,\\
a(5,3) &= 0.02928046868198058,\\
a(5,4) &= 0.44416888983409816,\\
a(6,1) &= 0.25513932774781371,\\
a(6,2) &= -0.91224869246478674,\\
a(6,3) &= 0.12346429780333155,\\
a(6,4) &= 0.11340304513170768,\\
a(6,5) &= 0.97544976900612506,\\
a(7,1) &= -0.00338460120482409,\\
a(7,2) &= 0.96300610761932561,\\
a(7,3) &= -0.13033383751561698,\\
a(7,4) &= -0.38703387369268386,\\
a(7,5) &= -0.38762081737097528,\\
a(7,6) &= 0.07438638878379881,\\
a(8,1) &= -0.34688272057927110,\\
a(8,2) &= -0.93041959747123893,\\
a(8,3) &= 0.12592355923675832,\\
a(8,4) &= 1.51105940982576962,\\
a(8,5) &= -0.81661432859575014,\\
a(8,6) &= 0.69468777687918786,\\
a(8,7) &= 0.58656433222445619,\\
a(9,1) &= 0.36230876838945222,\\
a(9,2) &= -1.53506946514817471,\\
a(9,3) &= 0.20775724334751066,\\
a(9,4) &= 0.31403956983032163,\\
a(9,5) &= 1.37610741878614276,\\
a(9,6) &= -0.06238986500150766,\\
a(9,7) &= -0.09344148324359761,\\
a(9,8) &= 0.00501225023850869,\\
a(10,1) &= -0.45232501264486841,\\
a(10,2) &= 0.26121420621098237,\\
a(10,3) &= -0.03535289095230847,\\
a(10,4) &= -0.07790032992614540,\\
a(10,5) &= -1.10981261745176303,\\
a(10,6) &= -0.22758179472733966,\\
a(10,7) &= 1.53226845684519658,\\
a(10,8) &= 0.00134599511617789,\\
a(10,9) &= 0.94470983387096405,
\end{aligned}
$$

$$
\begin{aligned}
&a(11,1) = 0.92978192438247650,\\
&a(11,2) = -0.44956145720877531,\\
&a(11,3) = 0.06084392347415334,\\
&a(11,4) = 0.36050185553285962,\\
&a(11,5) = 2.36792149044454641,\\
&a(11,6) = 1.91078806883411455,\\
&a(11,7) = -2.49197203253317762,\\
&a(11,8) = -0.54003782058034343,\\
&a(11,9) = -2.08822243848174019,\\
&a(11,10) = 0.76998059403332631,\\
&a(12,1) = 0.06819224287048619,\\
&a(12,2) = 0.82220048490261382,\\
&a(12,3) = -0.11127711813736400,\\
&a(12,4) = -0.94437678483826005,\\
&a(12,5) = 0.17958904435641741,\\
&a(12,6) = 2.74116070175902290,\\
&a(12,7) = 0.37080251680089533,\\
&a(12,8) = 1.49809234442720174,\\
&a(12,9) = -2.70377572786507846,\\
&a(12,10) = -1.24106977368453101,\\
&a(12,11) = 0.32046206940859614,\\
&b(1) = 0.04466284613064174,\\
&b(2) = 0,\\
&b(3) = 0,\\
&b(4) = 0.15244324052110214,\\
&b(5) = 0.15330386860405292,\\
&b(6) = 0.94038930735589507,\\
&b(7) = 0.01311512010387176,\\
&b(8) = 0.11679397193663187,\\
&b(9) = -0.62568879712246705,\\
&b(10) = 0.06328323793999999,\\
&b(11) = 0.09258781529323158,\\
&b(12) = 0.04910938923703997,\\
&c[2] = 0.17621190923742057,\\
&c[3] = 0.38219858251196794,\\
&c[4] = 0.13592796508435255,\\
&c[5] = 0.25862365068176710,\\
&c[6] = 0.55520774722419125,\\
&c[7] = 0.12901936661902420,\\
&c[8] = 0.82431843151991183,\\
&c[9] = 0.57432443719865599,\\
&c[10] = 0.83656584634089592,\\
&c[11] = 0.83002410789744019,\\
&c[12] = 1,
\end{aligned}
$$

Из всех методов 8-го порядка , метод (8,11) с погрешности $2.58 \cdot 10^{-7}$ лучшии, самы великолепный результат дает метод Дорман-Принса (5,7).

Глава 6.

Заключение

6.1. Заключение.

6.1.1. Нахождение методов Рунге-Кутта высокой точности остается проблемой, не поддающейся до конца усилиям математиков. Поэтому большой интерес представляют как отдельные найденные решения, их семейства, так и разработка новых подходов к системе уравнений Бутчера, служащей для нахождения методов Рунге-Кутта.

В диссертационной работе получены результаты по обоим этим направлениям.

- Исследованы общие подходы к решению уравнений Бутчера, предложен практический метод использования "каркаса" системы уравнений.
- Аналитически найдено новое семейство семистадийных методов Рунге-Кутта порядка 6.
- Численно найдено большое количество методов порядков 6, 7, и 8.
- Проведено сравнение эффективности уже известных и вновь найденных методов. Для проведения вычислений создан программный комплекс, позволяющий проводить сравнение методов с разным количеством этапов.

6.1.2. Некоторые нерешенные проблемы, для которых есть шанс получить решение в ближайшее время.

1. Найти методы Рунге-Кутта порядка 9. На сегодняшний день известно лишь, что минимальное количество шагов n для таких методов не меньше 12 и не больше 17. Сами методы не известны.

2. Оптимизировать алгоритм численного решения уравнений Бутчера. Версия, имеющаяся сегодня не позволяет находить методы порядка выше 8.

Литература

[1] А.А.Самарский,*Введение в численные методы,* стр.177-187 Москва., "Физматгиз", 1987.

[2] Бахвалов Н.С., Жидков Н.П., Кобельков Г.М., *Численные методы.* М.,"Наука", 1963.

[3] Березин И.С., Жидков Н.П., *Методы вычислений,* том 2. 640 с. Москва., "Физматгиз", 1962.

[4] Демидович Б.П., Марон И.А., *Основы вычислительной математики.* М.,"Наука", 1987.

[5] . Курош А.Г. *Курс высшей алгебры*, Москва 1968 "Наука".

[6] .Морозов В.А. *Алгоритмические основы методов решения некорректно поставленных задач."Вычислительные методы и программирование".* 2003, т.4, стр. 130-141.

[7] .Тихонов А.Н., Арсенин В.Я., *Методы решения некорректных задач.* М., Наука, 1979.

[8] Хайрер Э., Нерсетт С., Ваннер Г., *Решение обыкновенных дифференциальных уравнений. Нежесткие задачи.* 512 с., Москва., "Мир", 1990. Holland, NY, 1989.

[9] Ассюй Куасси Ришар, *Новый метод Рунге-Кутта порядка 8*, "Молодая наука - XXI веку", Тезисы докладов, Иваново, 2001 г., Часть. VI, с.51.

[10] Ассюй Куасси Ришар, *Новый метод Рунге-Кутта порядка 6*, "Молодая наука в классическом университете",Тезисы докпадов, Иваново, 2002 г, с.75

[11] Ассюй Куасси Ришар, *Нахождение каркаса метода Рунге-Кутта типа (6,7)*, "Молодая наука в классическом университете",Тезисы докпадов, Иваново, 2003 г, с.83

[12] Ассюй Куасси Ришар, *Семистадийные методы Рунге-Кутта порядка шесть*, ВЕСТНИК РУДН, Москва 2003 г, Т.2, №2, с.61-76.

[13] Хаммуд Г.М., Хашин С.И., *Упрощающие предположения высшего порядка для методов Рунге-Кутта,* Научные труды ИвГУ. Иваново, 2000 г. Вып.3. с.107-118.

[14] Хаммуд Г.М., Хашин С.И., *Шестимерное семейство 6-шаговых методов Рунге-Кутта порядка 5,* "Молодая наука - XXI веку", Тезисы докладов, Иваново, 2001 г., Часть. VI, с.70.

[15] Хаммуд Г.М., *Трехмерное семейство 7-шаговых методов Рунге-Кутта порядка 6*, "Молодая наука - XXI веку", Тезисы докладов, Иваново, 2001 г., Часть. VI, с.69.

[16] . Хаммуд Г.М., *3-х мерное семейство 7-шаговых методов Рунге-Кутты порядка 6.* Москва, МГУ, Вычислительные методы и программирование, 2001 г. т.2. с.159-166.

[17] . Хаммуд Г.М., Хашин С.И., *Шестимерное семейство 6-шаговых методов Рунге-Кутта порядка 5.* Научные труды ИвГУ. Математика. Иваново, 2001 г. Вып.4. с.114-122.

[18] Хашин С.И., *Численное решение уравнений Бутчера*, Вестник ИвГУ, Иваново, 2000, вып. 3. с.155-164.

[19] Alexander, Roger K.; Coyle, James J. *Runge-Kutta methods and differential-algebraic systems,* [J] SIAM J. Numer. Anal. 27, No.3, 736-752 (1990).

[20] Bartoszewski, Z.; Jackiewicz, Z., *Construction of two-step Runge-Kutta methods of high order of ordinary differential equations,* [J] Numer. Algorithms 18, No.1, 51-70 (1998). [ISSN 1017-1398].

[21] Butcher J.C., *Numerical analysis of ordinary differential equations,* John Wiley and Sons, Toronto, 1987.

[22] Butcher J.C., *Coefficients for the sttudy of Runge-Kutta Iteration Processes,* J. of the Australian Math.Soc. 3, 185-201.

[23] Butcher J.C., *On Runge-Kutta processes of high order,* J. Austral. Math. Soc. vol. IV, Part 2, p. 179-194. J. of the Australian Math.Soc. 3, 185-201.

[24] Butcher J.C., *Implicit Runge-Kutta Processes*, Math.Comp. **18** (1964), 50-64.

[25] Butcher J.C., *The Numerical Analysis of Ordinary Differential Equations.* Runge-Kutta and General Linear Methods, NY, 1987.

[26] Johnson L., Riess R.D., *Numerical analysis*, 1972.

[27] Stoer J., Bilirsch R., *Introduction to numerical analysis*, 1980.

[28] Bretan K.E., Campbell S.L., Petzold L.R., *Numerical Solution of Initial Problems in Differntial-algebraic Equations* Noth Holland, NY,1989.

[29] Konen, H.P.; Luther, H.A., *Some singular explicit fifth order Runge-Kutta solutions,* [J] SIAM J. Numer. Anal. 4, 607-619 (1967).

[30] Lawson, J.D., *An order six Runge-Kutta process with extended region of stability,* [J] SIAM J. Numer. Anal. 4, 620-625 (1967).

[31] Fehlberg, E., *New high-order Runge-Kutta formulas with an arbitrarily small truncation error,* [J] Z. Angew. Math. Mech. 46, 1-16 (1966).

[32] Fritsche, Michael, *Runge-Kutta methods of differential equations. I: Ordinary differential equations,* [J] Wiss. Beitr. Friedrich-Schiller-Univ. Jena 1983, 87-111 (1983).

[33] Hairer, E.; Wanner, G., *Algebraically stable and implementable Runge-Kutta methods of high order,* [J] SIAM J. Numer. Anal. 18, 1098-1108 (1981).

[34] Papageorgiou, G.S., *Comparison of Runge-Kutta type methods of order five,* [J] Bull. Greek Math. Soc. 28, 71-79 (1987).

[35] Sofroniou, M., *Symbolic derivation of Runge-Kutta methods,* [J] J. Symb. Comput. 18, No.3, 265-296 (1994).

[36] Hairer, E.; Wanner, G., *Symplectic Runge-Kutta methods with real eigenvalues,* [J] BIT 34, No.2, 310-312 (1994).