

Санкт–Петербургский государственный университет

Пак Алексей Станиславович

Выпускная квалификационная работа

*Анализ влияния вычислительной погрешности в
структурных методах решения обыкновенных
дифференциальных уравнений*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2016 «Прикладная
математика, фундаментальная информатика и программирование»

Профиль «Математическое и программное обеспечение
вычислительных машин»

Научный руководитель:

кандидат физико-математических наук

доцент кафедры информационных систем Еремин Алексей Сергеевич

Рецензент:

кандидат физико-математических наук

доцент кафедры моделирования электромеханических и компьютерных
систем Кривовичев Герасим Владимирович

Санкт-Петербург

2020 г.

Содержание

Введение	3
Постановка задачи	4
Глава 1. Компенсация вычислительной погрешности в явных методах Рунге — Кутты	5
1.1. Компенсация вычислительной погрешности в явном методе Эйлера	5
1.2. Общая формулировка явных методов Рунге — Кутты . . .	8
1.3. Компенсация вычислительной погрешности в явных методах Рунге — Кутты (ЯМРК)	11
Глава 2. Структурные методы решения обыкновенных дифференциальных уравнений	15
2.1. Методы класса \mathfrak{C}	15
2.2. Расчётные схемы методов класса \mathfrak{C}	17
2.2.1 Метод второго порядка класса \mathfrak{C}	17
2.2.2 Метод третьего порядка класса \mathfrak{C}	17
2.2.3 Метод четвёртого порядка класса \mathfrak{C}	18
Глава 3. Реализация структурных методов решения обыкновенных дифференциальных уравнений с компенсацией погрешности	21
Выводы	24
Заключение	25
Список литературы	26
Приложение	27

Введение

Двадцать первый век можно охарактеризовать как период, в течение которого классические методы решения обыкновенных дифференциальных уравнений (ОДУ), таких как методы Рунге — Кутты (РК), Адамса, экстраполяции, разработанные для ручного счёта, пересматривались и развивались исходя из требований и цифровых возможностей. Поскольку мощности ЭВМ растут изо дня в день, то представляются новые возможности для решения более трудоёмких и сложных задач. Однако в практических вычислениях природа компьютерной арифметики может играть значительную и, возможно, подавляющую роль.

Рассмотрим явный метод Эйлера для решения начальной задачи

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0 \quad (1)$$

в общем виде он выглядит следующим образом:

$$\begin{aligned} y_1(x_0 + h) &\approx y_0 + hf(x_0, y(x_0)) \\ y_2(x_1 + h) &\approx y_1 + hf(x_1, y(x_1)) \\ &\dots \\ y_n(x_{n-1} + h) &\approx y_{n-1} + hf(x_{n-1}, y(x_{n-1})) \end{aligned} \quad (2)$$

Чтобы получить приближенное значение решения начальной задачи (1) в конечной точке X , будут применены формулы (2) последовательно к интервалам $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$. Предполагается, что последовательность приближений y_1, y_2, y_3, \dots к решению дифференциального уравнения при x_1, x_2, x_3, \dots в конечном итоге приведет к приемлимым приближениям решения дифференциального уравнения.

После многих шагов метода Эйлера, ошибки округления и накопление погрешности при суммировании будут накладываться друг на друга сложным образом. Надо отметить, что данная проблема является актуальной для всех численных методов решения ОДУ, таким образом, возникает необходимость улучшить и оптимизировать уже известные методы интегрирования.

Постановка задачи

Существующие методы численного интегрирования систем обыкновенных дифференциальных уравнений (СОДУ) на данный момент имеют ряд проблем, связанных с компьютерными вычислениями, которые могут оказывать неблагоприятное влияние на конечный результат. Наиболее значимой проблемой является накопление ошибок округления и погрешности, которые накладываются друг на друга сложным образом.

Главной задачей данной работы является распространение алгоритма действий для преодоления последствий ошибок округления на явные методы Рунге—Кутты, а также на структурные методы решения систем обыкновенных дифференциальных уравнений.

Глава 1. Компенсация вычислительной погрешности в явных методах Рунге — Кутты

В книге Бутчера [1] представлен анализ влияния ошибки округления в явном методе Эйлера, а также приведён алгоритм Гилла — Мёллера (Гилл [3]; Мёллер [7], [8]), который называют «компенсированным суммированием». В данной главе будет проведён аналогичный численный эксперимент для преодоления последствий накопления ошибок округления в методе Эйлера, а также распространение алгоритма Гилла — Мёллера на явные методы Рунге — Кутты.

1.1 Компенсация вычислительной погрешности в явном методе Эйлера

Пусть α_n обозначает общую погрешность, вычисленную на шаге n , и β_n соответствующую погрешность вычисления производной на этом шаге. Таким образом,

$$\begin{aligned}\alpha_n &= y(x_n) - y_n, \\ \beta_n &= f(x_n, (y(x_n))) - f(x_n, y_n),\end{aligned}$$

последовательности точных и приближенных значений связаны между собой следующим образом

$$\begin{aligned}y_n &= y_{n-1} + hf(x_{n-1}, y_{n-1}) - r_n, \\ y(x_n) &= y(x_{n-1}) + hf(x_{n-1}, y(x_{n-1})) + l_n,\end{aligned}$$

где r_n — ошибка округления, совершенная на этом шаге, а l_n — методическая ошибка.

Данная операция приводит к следующему разностному уравнению

$$\alpha_n = \alpha_{n-1} + h\beta_{n-1} + l_n + r_n.$$

Так как, детальный анализ ошибки округления в расчётах данной задачи

достаточно подробно представлен в работе Хенричи [4], то вместо того, чтобы пытаться провести анализ r_n и l_n , а также, поскольку шаги вычисляются последовательно, заметим, что можно преодолеть наихудшие последствия накопления ошибок округления. Это делается путем оценки значения r_n на любом конкретном шаге, а затем добавления его к значению $hf(x_n, y_n)$, прежде чем оно будет добавлено на следующем шаге.

Этот усовершенствованный метод, который может быть использован для многих ситуаций, связанных с суммированием большого количества малых величин, и называют алгоритмом Гилла — Мёллера или «компенсированным суммированием». Анализ арифметики с плавающей точкой был проведен Каханом [6], а конкретные приложения к задачам начального приближения были рассмотрены Витасеком [9]. Более подробный компенсированного суммирования с дополнительными ссылками доступен в Хайме [5].

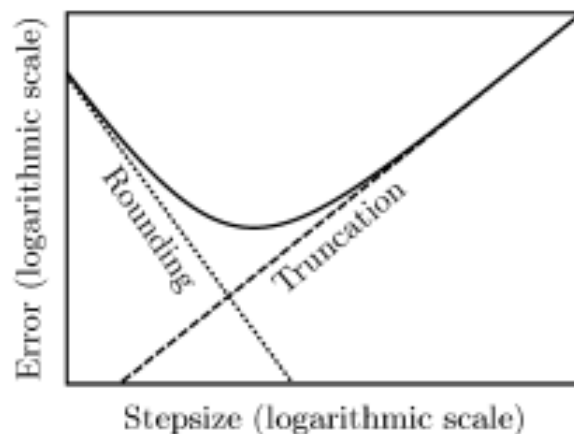


Рис. 1: Схема, показывающая влияние погрешности округления

На рисунке (1) пунктирная линия *Rounding* показывает влияние погрешности округления, *Truncation* - методическая погрешность метода Эйлера. Сплошная линия показывает совокупное влияние двух источников погрешностей. Предполагается рост погрешности, но последствия накопления погрешностей округления можно преодолеть, так как шаги вычисляются последовательно.

Покажем, как это делается, представив два фрагмента программного

кода [1] в системе научных вычислений MATLAB, из которых первый, наивно вычисляет решение, а второй делает улучшение, о котором говорилось ранее.

Простой метод Эйлера (Алгоритм 1 α)

```
for i = 1:n
    term = single(h * dydx(x(i), y(i)));
    y(i+1) = single(y(i) + term);
    x(i+1) = single(x(1) + i*h);
end
```

Метод Эйлера с компенсированным суммированием (Алгоритм 1 β)

```
z = single(0);
for i = 1:n
    term = single(h * dydx(x(i), y(i)) + z);
    newy = single(y(i) + term);
    z = single(term - (newy - y(i)));
    y(i+1) = single(newy);
    x(i+1) = single(x(1) + i*h);
end
```

В данном примере была взята последовательность шагов $h = 2^{-2}, h = 2^{-3}, \dots, h^{-23}$. Каждый из двух алгоритмов были использованы, а ошибки были нанесены на один и тот же график. Чтобы избежать необходимости использования слишком малых величин h , вычисления проводятся с точностью до девяти значимых десятичных разрядов. Видно, что наивная форма дает результаты, которые все сильнее ухудшаются при шаге менее 2^{-15} . С другой стороны, для «улучшенного» алгоритма Эйлера, нет никаких признаков накопленной ошибки округления.

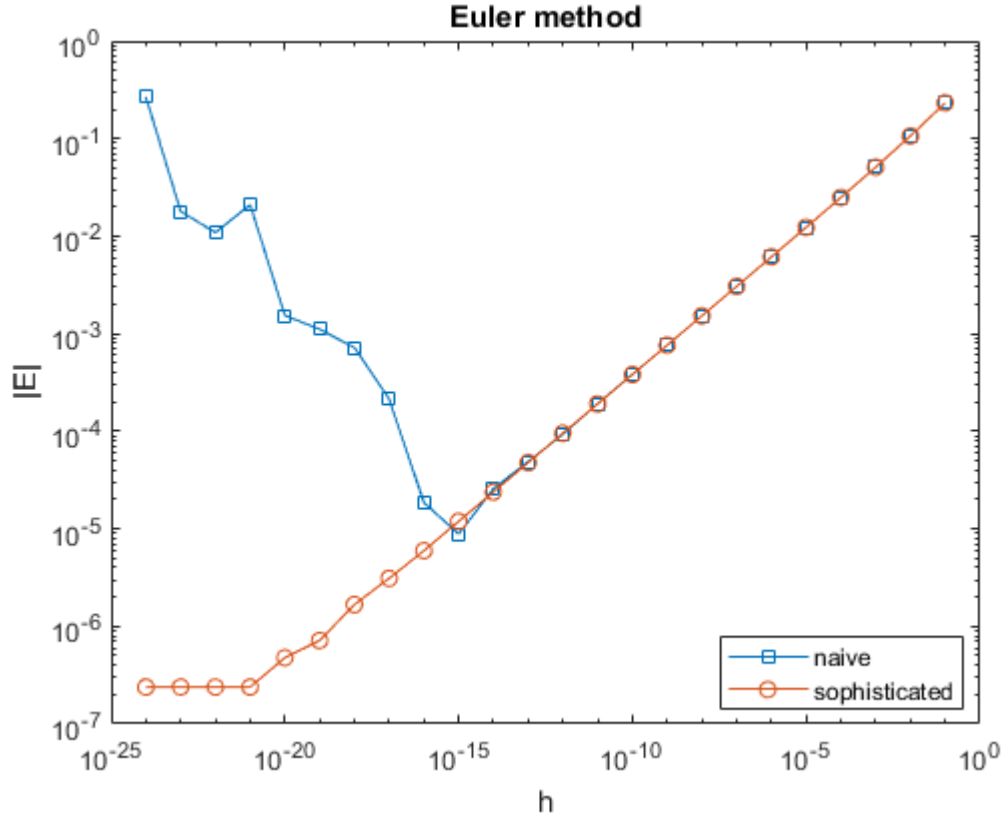


Рис. 2: Ошибки для наивной и компенсированной форм метода Эйлера

1.2 Общая формулировка явных методов Рунге — Кутты

Общая схема явного метода Рунге — Кутты численного интегрирования систем обыкновенных дифференциальных уравнений (СОДУ)

$$\frac{dy_s}{dx} = f_s(x, y_0, \dots, y_n), \quad s = 0, 1, \dots, n, \quad (3)$$

$$y_s(X_0) = y_{s0}, \quad s = 0, 1, \dots, n,$$

$$x \in [X_0, X_k] \subset R, \quad y_s : [X_0, X_k] \longrightarrow R,$$

$$f_s : [X_0, X_k] \times R^{n+1} \longrightarrow R, \quad s = 0, 1, \dots, n,$$

дана, например, в [10]:

$$y_s(x+h) \approx z_s = y_s(x) + \sum_{g=1}^{w-1} b_{sw} k_{sw}(h), \quad s = 0, 1, \dots, n, \quad (4)$$

где функции $k_{sw} = k_{sw}(h)$ вычисляются по схеме

$$k_{sw} = hf_s(x + c_{sw}h, y_0(x) + \sum_{g=1}^{w-1} a_{sw0g} k_{0g}, \dots, y_n(x) + \sum_{g=1}^{w-1} a_{swng} k_{ng}) \quad (5)$$

$$c_{s1} = 0, \quad a_{swt1} = 0, \quad t = 0, 1, \dots, n$$

Здесь $y_s(x+h)$, z_s — соответственно точное и приближенное значения s -й компоненты в точке $x+h \in [X_0, X_k]$, $y_s(x)$ — точное значение s -ой компоненты в точке $x \in [X_0, X_k]$, h — шаг метода. При построении расчётных схем в рамках (4), (5) параметры метода полагают

$$b_w \equiv b_{sw}, \quad c_w \equiv c_{sw}, \quad a_{wg} \equiv a_{swtg},$$

не зависящими от номера компоненты s и t .

Определение 1. Метод Рунге — Кутты (4)–(5) имеет порядок p , если для достаточно гладких задач (3)

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

т. е. если ряды Тейлора для точного решения $y(x_0+h)$ и для y_1 совпадают до члена h^p включительно. После статьи Бутчера [2] традиционным стало символически представлять метод (4)–(5) посредством следующей таблицы

Таблица 1: Общий вид таблицы Бутчера

0					
c_2	a_{w1}				
c_3	a_{w1}	a_{w2}			
\cdot	\cdot	\cdot			
\cdot	\cdot	\cdot			
c_w	a_{w1}	a_{w2}	\dots	$a_{w,w-1}$	
<hr/>					
	b_1	b_2	\dots	b_{w-1}	b_w

Приведем примеры таблиц Бутчера для явных методов Рунге — Кутты второго, третьего и четвертого порядков

Таблица 2: ЯМРК второго порядка

0		
$\frac{1}{2}$	$\frac{1}{2}$	
<hr/>		
	0	1

Таблица 3: ЯМРК третьего порядка

0			
$\frac{1}{3}$	$\frac{1}{3}$		
$\frac{2}{3}$	0	$\frac{2}{3}$	
<hr/>			
	$\frac{1}{4}$	0	$\frac{3}{4}$

Таблица 4: ЯМРК четвёртого порядка

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
<hr/>				
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

1.3 Компенсация вычислительной погрешности в явных методах Рунге — Кутты (ЯМРК)

В данном параграфе будет показано как применить компенсированное суммирование для уменьшения ошибки округления в ЯМРК четвертого порядка на примере системы дифференциальных уравнений

$$\begin{aligned}y_1' &= 2xy_2^{1/5}y_4 \\y_2' &= 10x \exp(5(y_3 - 1))y_4 \\y_3' &= 2xy_4 \\y_4' &= -2x * \ln(y_1)\end{aligned}$$

с начальным приближением

$$y_i(0) = 1, \quad i = 1, 2, 3, 4,$$

которое имеет аналитическое решение вида

$$\begin{aligned}y_1 &= \exp(\sin(x^2)) \\y_2 &= \exp(5 \sin(x^2)) \\y_3 &= \sin(x^2) + 1 \\y_4 &= \cos(x^2),\end{aligned}$$

а также представлены результаты сравнения явных методов Рунге — Кутты второго, третьего и четвертого порядков с их компенсированными версиями.

В данном примере f — массив указателей на функции правых частей, $xStart$ — начальная абсцисса расчёта, $y0$ — вектор начальных переменных, n — количество шагов, N — количество уравнений.

ЯМРК4 (Алгоритм 2α)

```

for i = 1:n
    x0 = xStart + i*h;
    X = x0;
    Y = y0;
    for j=1:N
        K1 = f{i}(X,Y);
    end
    X = x0 + c2*h;
    Y = y0 + h*a21*K1;
    for j=1:N
        K2 = f{i}(X,Y);
    end
    X = x0 + c3*h;
    Y = y0 + h*(a31*K1 + a32*K2);
    for j=1:N
        K3 = f{i}(X,Y);
    end
    X = x0 + c4*h;
    Y = y0 + h*(a41*K1 + a42*K2 + a43*K3);
    for j=1:n
        K4 = f{i}(X,Y);
    end

    y1 = y0 + h*(b1*K1 + b2*K2 + b3*K3 + b4*K4);

    y0=y1; % for next step
end

```

ЯМРК4 с компенсированным суммированием (Алгоритм 2β)

```

z = zeros(N,1);
term = zeros(N,1);
newy = zeros(N,1);

```

```

for i = 1:n
    x0 = xStart + i*h;
    % the first stage
    X = x0;
    Y = y0;
    for j=1:N
        K1 = f{i}(X,Y);
    end
    % the second stage
    X = x0 + c2*h;
    Y = y0 + h*a21*K1;
    for j=1:N
        K2 = f{i}(X,Y);
    end
    % the third stage
    X = x0 + c3*h;
    Y = y0 + h*(a31*K1 + a32*K2);
    for j=1:N
        K3 = f{i}(X,Y);
    end
    % the fourth stage
    X = x0 + c4*h;
    Y = y0 + h*(a41*K1 + a42*K2 + a43*K3);
    for j=1:n
        K4 = f{i}(X,Y);
    end
    term = h*(b1*K1 + b2*K2 + b3*K3 + b4*K4) + z;
    newy = y0 + term;
    z = term - (newy - y0)
    y1 = newy;
    y0=y1; % for next step
end

```

В данном примере была взята последовательность шагов $h = 2^{-2}, h = 2^{-3}, \dots, h^{-24}$.

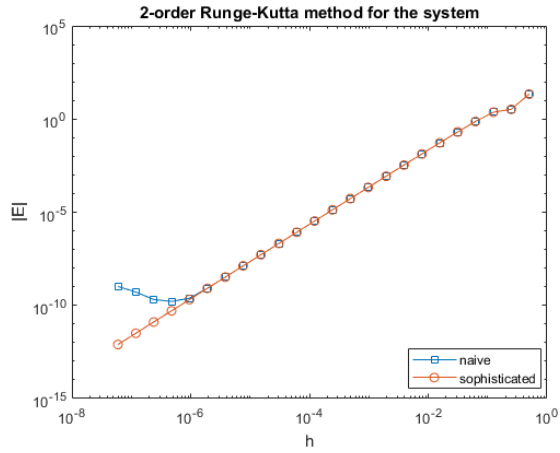


Рис. 3: Ошибки для явной и компенсированной форм ЯМРК2

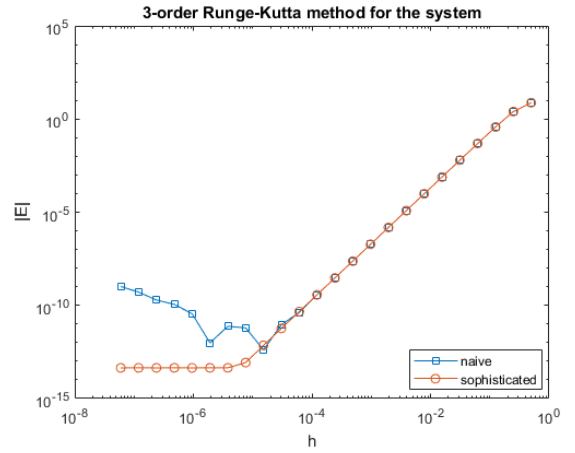


Рис. 4: Ошибки для явной и компенсированной форм ЯМРК3

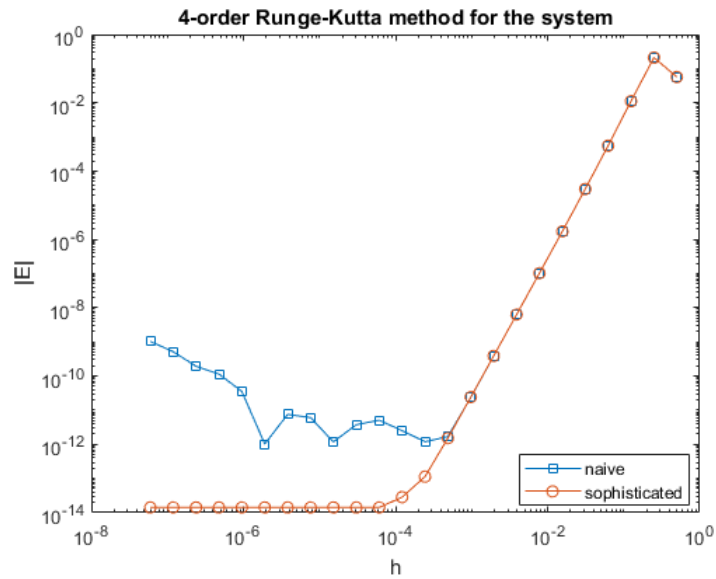


Рис. 5: Ошибки для явной и компенсированной форм ЯМРК4

Видно, что явная форма данных методов даёт результаты, которые сильнее ухудшаются при определенном шаге, в то время как их компенсированные версии значительно улучшают результаты.

Глава 2. Структурные методы решения обыкновенных дифференциальных уравнений

2.1 Методы класса \mathfrak{C}

В данном разделе будет рассматриваться явный одношаговый метод типа Рунге—Кутты интегрирования СОДУ, разработанный Игорем Владимировичем Олемским [11], для систем вида

$$\frac{dy_0}{dx} = f_0(x, y_0, \dots, y_n), \quad (6)$$

$$\frac{dy_i}{dx} = f_i(x, y_0, \dots, y_{i-1}, y_{l+1}, \dots, y_n), \quad i = 1, \dots, l \quad (7)$$

$$\frac{dy_j}{dx} = f_j(x, y_0, \dots, y_{j-1}), \quad j = l + 1, \dots, n, \quad (8)$$

$$\begin{aligned} l &\in \{0\} \cup N, \quad n \in \{0\} \cup N, \quad l \leq n, \\ x &\in [X_0, X_k] \subset R, \quad y_s : [X_0, X_k] \longrightarrow R^{r_s}, \quad s = 0, 1, \dots, n, \\ f_0 &: [X_0, X_k] \times R^r \longrightarrow R^{r_0}, \quad \sum_{s=1}^n r_s = r, \\ f_i &: [X_0, X_k] \times R^{r - \sum_{g=i}^l r_g} \longrightarrow R^{r_i}, \quad i = 1, \dots, l, \\ f_j &: [X_0, X_k] \times R^{r - \sum_{g=j}^l r_g} \longrightarrow R^{r_j}, \quad j = l + 1, \dots, n, \end{aligned}$$

где y_s, f_s — функции размерности r_s . Две группы уравнений (7), (8) структурно тождественны. Каждое уравнение одной из групп (7), (8) занимает определенное место в последовательности уравнений своей группы. Его правая часть не зависит от искомых функций, поведение которых описывается этим и всеми последующими уравнениями этой же группы. Уравнение (6) — представитель общей группы, в которую входят все уравнения, не имеющие структурных особенностей. Также, в данной системе может отсутствовать любая из групп уравнений (6), (7), (8).

Для численного интегрирования систем (6)–(8) был разработан яв-

ный одношаговый метод, который в дальнейшем будем называть структурным или методом класса \mathfrak{E} . В отличие от метода (4)–(5) вычислительная схема метода класса \mathfrak{E} использует структурные особенности системы на уровне алгоритма. Приближение z_s к точному решению $y_s(x+h)$, $s = 0, 1, \dots, n$ в точке $x+h \in [X_0, X_k]$ вычисляется в виде

$$y_s(x+h) \approx z_s = y_s(x) + \sum_{w=1}^{m_s} b_{sw} k_{sw}(h), \quad s = 0, 1, \dots, n, \quad (9)$$

причём $k_{sw} \equiv k_{sw}(h)$ вычисляются в строгой последовательности

$$k_{01}, k_{11}, \dots, k_{n1}, k_{02}, k_{12}, \dots, k_{n2}, k_{03}, k_{13}, \dots \quad (10)$$

по схеме

$$\begin{aligned} k_{0w} &= hf_0(T_{0,w}, Y_{0,w,0}, Y_{0,w,1}, \dots, Y_{0,w,n}), \\ k_{iw} &= hf_i(T_{i,w}, Y_{i,w,0}, \dots, Y_{i,w,i-1}, Y_{i,w,l+1}, \dots, Y_{i,w,n}), \quad i = 1, \dots, l, \\ k_{jw} &= hf_j(T_{j,w}, Y_{j,w,0}, \dots, Y_{j,w,j-1}), \quad j = l+1, \dots, n, \end{aligned}$$

где

$$T_{s,w} = \begin{cases} x, & \text{если } \{(w = 1 \wedge s \leq l)\}, \\ x + c_{sw}h, & \text{если } \{(w = 1 \wedge s > l) \vee (w > l)\}, \end{cases} \quad (11)$$

$$Y_{s,w,\nu} = \begin{cases} y_\nu(x), & \text{если } \{(w = 1 \wedge s \leq l)\}, \\ y_\nu(x) + \sum_{\mu=1}^{w-1} a_{sw\nu\mu} k_{\nu\mu}, & \text{если } \{(w > 1 \wedge s \leq \nu)\}, \\ y_\nu(x) + \sum_{\mu=1}^w a_{sw\nu\mu} k_{\nu\mu}, & \text{если } \{(w > 1 \wedge s \geq \nu)\} \vee \{(w = 1 \wedge s > l)\}. \end{cases} \quad (12)$$

Здесь $b_{sw}, c_{sw}, a_{sw\nu\mu}$ — параметры метода (9), (10), (11), (12); h — шаг метода. Далее рассмотрим методы различных порядков класса \mathfrak{E} , а также приведём расчётные схемы для интегрирования СОДУ.

2.2 Расчётные схемы методов класса \mathfrak{C}

2.2.1 Метод второго порядка класса \mathfrak{C} .

Расчётная схема второго порядка точности имеет вид

$$\begin{aligned}y_0(x+h) &= y_0(x) + k_{02} + O(h^3), \\y_i(x+h) &= y + 0(x) + k_{i2} + O(h^3), \\y_j(x+h) &= y + 0(x) + k_{j1} + O(h^3),\end{aligned}$$

где

$$\begin{aligned}k_{d1} &= hf_d(x, y_0(x), \dots, y_n(x)), \quad d = 1, \dots, l, \\k_{j1} &= hf_j(x + \frac{h}{2}, y_0(x) + \frac{1}{2}k_{01}, \dots, y_{j-1}(x) + \frac{1}{2}k_{j-1,1}), \\k_{02} &= hf_0(x + \frac{h}{2}, y_0(x) + \frac{1}{2}k_{01}, \dots, y_n(x) + \frac{1}{2}k_{n1}), \\k_{i2} &= hf_i(x + \frac{h}{2}, y_0(x) + \frac{1}{2}k_{02}, \dots, y_{i-1}(x) + \frac{1}{2}k_{i-1,2}, \\&\quad y_{l+1}(x) + \frac{1}{2}k_{l+1,1}, \dots, y_n(x) + \frac{1}{2}k_{n1}).\end{aligned}$$

2.2.2 Метод третьего порядка класса \mathfrak{C} .

Расчётная схема третьего порядка точности имеет вид

$$\begin{aligned}y_0(x+h) &= y_0(x) + \frac{1}{6}[k_{01} + 4k_{02} + k_{03}] + O(h^4), \\y_i(x+h) &= y_0(x) + \frac{1}{4}[k_{i1} + 3k_{i2}] + O(h^4), \\y_j(x+h) &= y_0(x) + \frac{1}{4}[3k_{j1} + k_{j2}] + O(h^4),\end{aligned}$$

где

$$\begin{aligned}
k_{d1} &= hf_d(x, y_0(x), \dots, y_n(x)), \quad d = 1, \dots, l, \\
k_{j1} &= hf_j(x + \frac{h}{3}, y_0(x) + \frac{1}{3}k_{01}, \dots, y_{j-1}(x) + \frac{1}{3}k_{j-1,1}), \\
k_{02} &= hf_0(x + \frac{h}{2}, y_0(x) + \frac{1}{2}k_{01}, \dots, y_n(x) + \frac{1}{2}k_{n1}), \\
k_{i2} &= hf_i(x + \frac{2h}{3}, y_0(x) + \frac{2}{9}k_{01} + \frac{4}{9}k_{02}, y_1(x) + \frac{1}{3}k_{11} + \frac{1}{3}k_{12}, \dots, y_{i-1}(x) + \\
&+ \frac{1}{3}k_{i-1,1} + \frac{1}{3}k_{i-1,2}, y_{l+1}(x) + \frac{2}{3}k_{l+1,1}, \dots, y_n(x) + \frac{2}{3}k_{n1}), \\
k_{j2} &= hf_j(x + h, y_0(x) - \frac{1}{3}k_{01} + \frac{4}{3}k_{02}, y_1(x) + k_{12}, \dots \\
&\dots, y_l(x) + k_{l2}, \dots, y_{l+1}(x) + k_{l+1,1}, \dots, y_{j-1}(x) + k_{j-1,1}), \\
k_{03} &= hf_0(x + h, y_0(x) - k_{01} + 2k_{02}, y_1(x) - \frac{1}{2}k_{11} + \\
&+ \frac{3}{2}k_{12}, \dots, y_l(x) + \frac{1}{2}k_{l1} + \frac{3}{2}k_{l2}, y_{l+1}(x) + k_{l+1,1}, \dots, y_n(x) + k_{n1}).
\end{aligned}$$

2.2.3 Метод четвёртого порядка класса \mathfrak{C} .

Расчётная схема четвёртого порядка точности имеет вид

$$\begin{aligned}
y_0(x + h) &= y_0(x) + \frac{1}{8}[k_{01} + 3k_{02} + 3k_{03} + k_{04}] + O(h^5), \\
y_i(x + h) &= y_0(x) + \frac{1}{10}[k_{i1}5 + k_{i2} + 4k_{i3}] + O(h^5), \\
y_j(x + h) &= y_0(x) + \frac{1}{10}[4k_{j1} + 5k_{j2} + k_{j3}] + O(h^5),
\end{aligned}$$

где

$$\begin{aligned}
k_{d1} &= hf_d(x, y_0(x), \dots, y_n(x)), \quad d = 1, \dots, l, \\
k_{j1} &= hf_j(x + \frac{h}{6}, y_0(x) + \frac{1}{6}k_{01}, \dots, y_{j-1}(x) + \frac{1}{6}k_{j-1,1}), \\
k_{02} &= hf_0(x + \frac{h}{3}, y_0(x) + \frac{1}{3}k_{01}, \dots, y_n(x) + \frac{1}{3}k_{n1}), \\
k_{i2} &= hf_i(x + \frac{h}{3}, y_0(x) + \frac{1}{6}k_{01} + \frac{1}{6}k_{02}, \dots, y_{i-1}(x) + \\
&+ \frac{1}{6}k_{i-1,1} + \frac{1}{6}k_{i-1,2}, y_{l+1}(x) + \frac{1}{3}k_{l+1,1}, \dots, y_n(x) + \frac{1}{3}k_{n1}), \\
k_{j2} &= hf_j(x + \frac{2h}{3}, y_0(x) - \frac{1}{12}k_{01} + \frac{3}{4}k_{02}, \dots, y_l(x) - \\
&- \frac{1}{12}k_{l1} + \frac{3}{4}k_{l2}, y_{l+1}(x) + \frac{1}{2}k_{l+1,1} + \frac{1}{6}k_{l+1,2}, \dots \\
&\dots, y_{j-1}(x) + \frac{1}{2}k_{j-1,1} + \frac{1}{6}k_{j-1,2}), \\
k_{03} &= hf_0(x + \frac{2h}{3}, y_0(x) - \frac{1}{3}k_{01} + k_{02}, \dots, y_l(x) - \frac{1}{3}k_{l1} + k_{l2}, \\
&y_{l+1}(x) + \frac{4}{9}k_{l+1,1} + \frac{2}{9}k_{l+1,2}, \dots, y_n(x) + \frac{4}{9}k_{n1} + \frac{2}{9}k_{n2}), \\
k_{i3} &= hf_i(x + \frac{5h}{6}, y_0(x) + \frac{5}{48}k_{01} + \frac{5}{12}k_{02} + \frac{5}{16}k_{03}, \\
&y_1(x) + \frac{1}{24}k_{11} + \frac{5}{8}k_{12} + \frac{1}{6}k_{13}, \dots, y_{i-1}(x) + \frac{1}{24}k_{i-1,1} + \frac{5}{8}k_{i-1,2} + \frac{1}{6}k_{i-1,3}, \\
&y_{l+1}(x) + \frac{5}{12}k_{l+1,1} + \frac{5}{12}k_{l+1,2}, \dots, y_n(x) + \frac{5}{12}k_{n1} + \frac{5}{12}k_{n2}), \\
k_{j3} &= hf_j(x + h, y_0(x) + k_{01} - \frac{5}{4}k_{02} + \frac{5}{4}k_{03}, \\
&y_1(x) + \frac{3}{4}k_{11} - \frac{5}{12}k_{12} + \frac{2}{3}k_{13}, \dots, y_l(x) + \frac{3}{4}k_{l1} - \frac{5}{12}k_{l2} + \frac{2}{3}k_{l3}, \\
&y_{l+1}(x) + \frac{1}{6}k_{l+1,1} + \frac{5}{6}k_{l+1,2}, \dots, y_{j-1}(x) + \frac{1}{6}k_{j-1,1} + \frac{5}{6}k_{j-1,2}), \\
k_{04} &= hf_0(x + h, y_0(x) + k_{01} - k_{02} + k_{03}, \\
&y_1(x) + \frac{4}{5}k_{11} - \frac{1}{3}k_{12} + \frac{8}{15}k_{13}, \dots, y_l(x) + \frac{4}{5}k_{l1} - \frac{1}{3}k_{l2} + \frac{8}{15}k_{l3}, \\
&y_{l+1}(x) + \frac{1}{3}k_{l+1,1} + \frac{2}{3}k_{l+1,2}, \dots, y_n(x) + \frac{1}{3}k_{n1} + \frac{2}{3}k_{n2}).
\end{aligned}$$

Из представленных расчётных схем можно заметить, что при нали-

чии нескольких уравнений в структурных группах погрешность из этапа в этап переходит не параллельно по всем координатам, а из координаты в координату внутри одного этапа. Таким образом, накопление погрешности происходит быстрее и необходимость в реализации методов класса \mathfrak{C} с компенсированным суммированием возрастает.

Глава 3. Реализация структурных методов решения обыкновенных дифференциальных уравнений с компенсацией погрешности

Представим графики сравнения глобальной погрешности структурного метода 4-го порядка с компенсированной версией метода Рунге — Кутты 4-го порядка и компенсированной версией структурного метода 4-го порядка на примере системы размерностью $n = 2N + 2$, где N — число уравнений в группах 1 и 2. В группе 0 — 2 уравнения.

Решением системы являются функции вида

$$y_i(x) = \begin{cases} e^t, & i \bmod 2 = 1, \\ e^{-t}, & i \bmod 2 = 0, \end{cases} \quad i = 1, \dots, n,$$

группа 0 системы:

$$\begin{aligned} \frac{dy_1}{dx} &= y_1^2 y_2 + \sum_{i=3}^n (-1)^{\lceil i/2 \rceil \bmod 2} y_i^2, \\ \frac{dy_2}{dx} &= -y_1 y_2^2 + \sum_{i=3}^n (-1)^{\lceil (i+1)/2 \rceil \bmod 2} y_i^2, \end{aligned}$$

группа 1:

$$\frac{dy_i}{dx} = (-1)^{i+1 \bmod 2} y_{i-2}^2 y_{i-1} + \sum_{i=N+3}^n (-1)^{\lceil (i+1)/2 \rceil \bmod 2} y_i^2, \quad i = 3, \dots, N + 2,$$

группа 2:

$$\frac{dy_i}{dx} = (-1)^{i+1 \bmod 2} y_{i-2}^2 y_{i-1} + \sum_{i=3}^{N+2} (-1)^{\lceil (i+1)/2 \rceil \bmod 2} y_i^2, \quad i = N + 3, \dots, n.$$

Здесь $\lceil \cdot \rceil$ означает округление вверх. В данном примере была взята последовательность шагов $h^{-2}, h^{-3}, \dots, h^{-17}$.

На графиках (6), (7) представлено сравнение ошибок структурного

метода 4-го с методом Рунге—Кутты 4-го порядка и компенсированной версией структурного метода 4-го порядка соответственно при $N = 4$.

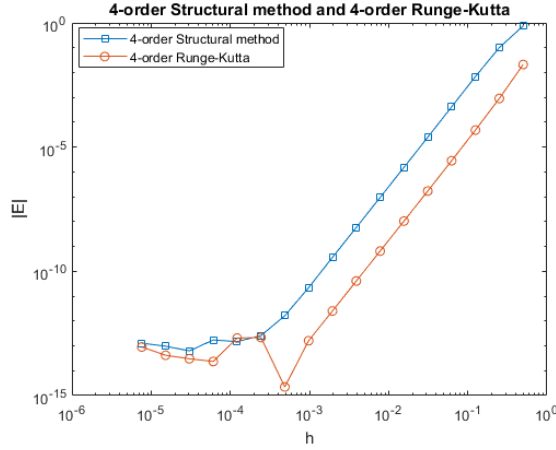


Рис. 6: Ошибки для структурного метода 4-го порядка и метода Рунге—Кутты 4-го порядка

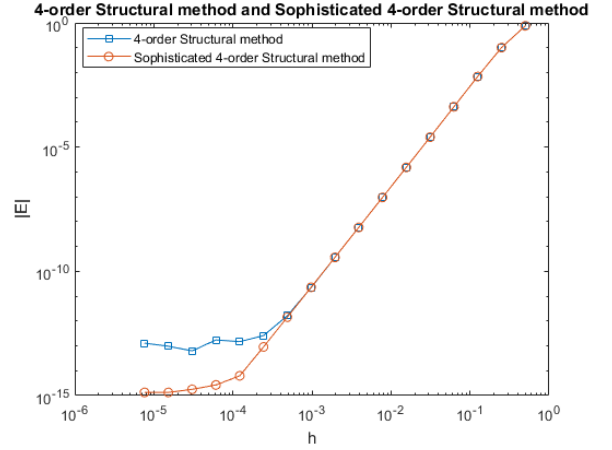


Рис. 7: Ошибки для структурного метода 4-го порядка и компенсированного структурного метода 4-го порядка

На графиках (8), (9) представлено аналогичное сравнение ошибок при $N = 8$.

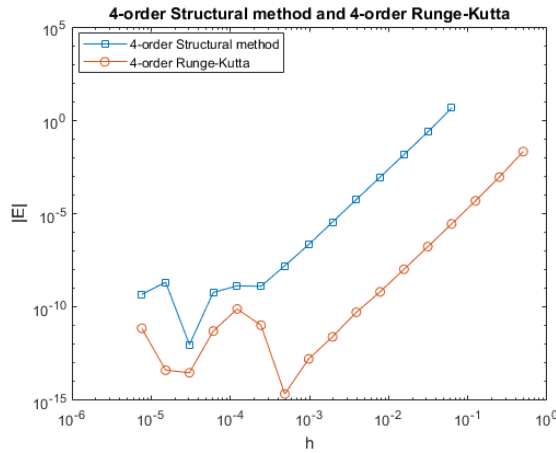


Рис. 8: Ошибки для структурного метода 4-го порядка и метода Рунге—Кутты 4-го порядка

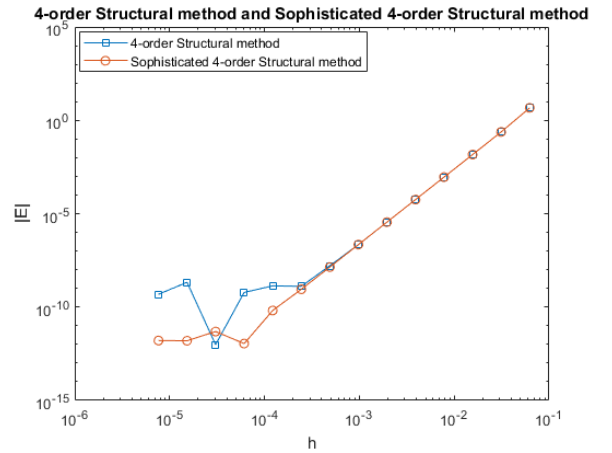


Рис. 9: Ошибки для структурного метода 4-го порядка и компенсированного структурного метода 4-го порядка

На графиках (10), (11) представлено аналогичное сравнение ошибок при $N = 12$.

Применяя сравнительный анализ для графиков (6)–(11), можно сделать вывод, что использование компенсированной версии структурного метода объективно обоснованно и приводит к уменьшению ошибок округления и погрешности. Также можно заметить на графиках (6), (8), (10), что

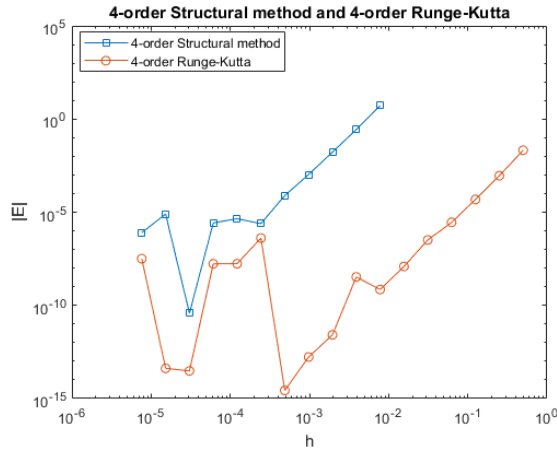


Рис. 10: Ошибки для структурного метода 4-го порядка и метода Рунге — Кутты 4-го порядка

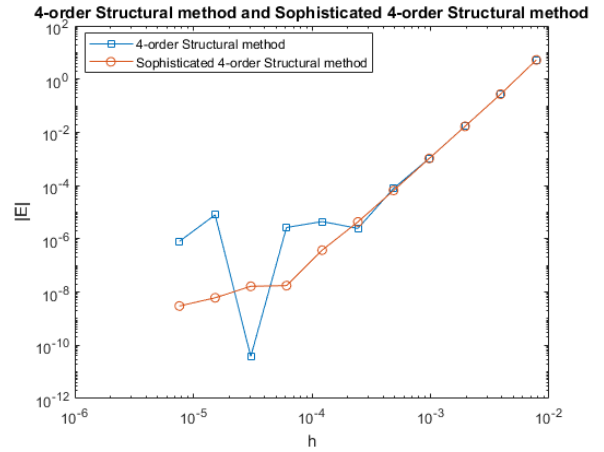


Рис. 11: Ошибки для структурного метода 4-го порядка и компенсированного структурного метода 4-го порядка

при увеличении количества уравнений в структурных группах 1 и 2, расхождение в глобальной погрешности значительно увеличивается. Таким образом, реализация структурных методов с компенсацией погрешности оправдана.

Программный код реализации компенсированной версии структурного метода 4-го порядка представлен в Приложении.

Выводы

Основываясь на приведённых результатах, отметим, что методическая погрешность методов класса \mathfrak{E} растёт при увеличении количества уравнений в первой и второй структурных группах, по сравнению с применением явных методов Рунге — Кутты к тем же системам обыкновенных дифференциальных уравнений. И в тоже время, выигрыш в вычислительных затратах для структурного метода оказывается незначительным по сравнению с ростом вычислительной погрешности.

Рассмотренная в главе 3 система была выбрана для сравнения влияния вычислительной погрешности на методы Рунге — Кутты и структурные. Мы предполагали, что в силу большего количества последовательных суммирований вычислительная погрешность проявит себя сильнее в структурных методах. В то же время обнаружилось, что для рассмотренной задачи последовательная природа вычислений в структурных методах оказывает куда большее влияние на методическую погрешность.

К сожалению, рассмотреть подробно методическую погрешность структурных методов в зависимости от сложности системы (6)–(8) в настоящей работе не удалось в силу сложности такого анализа.

Тем не менее, можно с уверенностью сказать, что реализация компенсации погрешности суммирования тем более необходима и в структурных методах.

Заключение

Целью данной работы являлась в нахождение приёмов и способов улучшения и оптимизации известных методов интегрирования систем обыкновенных дифференциальных уравнений. В ходе работы для решения СОДУ было предложено применить компенсированное суммирование в явных методах Рунге — Кутты второго, третьего и четвертого порядков, а также в структурных методах класса \mathfrak{C} .

Применение данного метода к известным алгоритмам показало, что возможно значительно уменьшить накопление ошибок округления и погрешностей на каждом этапе, что в свою очередь даёт положительный результат при достаточно малых шагах интегрирования.

К сожалению, строгий анализ того, как ведет себя методическая погрешность в структурных методах не был представлен, поэтому решение данной задачи планируется выполнить в будущем.

Список литературы

- [1] Butcher J. C. «Numerical Methods for ordinary Differential Equations, Second Edition». The University of Auckland, New Zealand 2008, 463 p.
- [2] Butcher J. C. «On Runge — Kutta processes of high order». J. Austral. Math. Soc. Vol. IV 1964, Part 2.
- [3] Gill S. «A process for the step-by step integration of differential equations in an automatic computing machine». Proc. Cambridge Philos. Soc 1951.
- [4] Henrici P. «Discrete Variable Methods in Ordinary Differential Equations». John Wiley & Sons Inc, New York 1962.
- [5] Higham N. J. «The accuracy of floating point summation». SIAM J. Sci. Comput. 1993.
- [6] Kahan W. «Further remarks on reducing truncation errors». Comm. ACM 1965.
- [7] Møller O. «Quasi double-precision in floating point addition». BIT 1965.
- [8] Møller O. «Note on quasi double-precision». BIT 1965.
- [9] Vitasek E. «The numerical stability in solution of differential equations. In J.L. Morris (ed.)». Conference on the Numerical Solution of Differential Equations 1969.
- [10] Крылов В. И., Бобков В. В., Монастырный П. И. «Начала теории вычислительных методов. Дифференциальные уравнения». Минск: Наука и техника 1982.
- [11] Олемской И.В. «Методы интегрирования систем структурно разделенных дифференциальных уравнений». СПбГУ 2009, 179 с.
- [12] Э.Хайрер, С.Нёрсетт, Г.Ваннер «Решение обыкновенных дифференциальных уравнений. Нежёсткие задачи». «Мир» 1990, 512 с.

Приложение

```
function y_final = soph_SM4(f, x0, xend, y0, N, n0, n1, n2)
    %f - массив указателей на функции правых частей
    % f{i}(x,y) - i-й компонент f
    %x0 - начальная абсцисса расчёта
    %xend - конечная абсцисса расчёт
    %y0 - вектор начальных переменных
    %N - число шагов
    %[n0 n1 n2] - количество уравнений в общей,
        %первой и второй структурных
    %группах соответственно

    %constants
    h = (xend-x0)/N;
    neq = n0 + n1 + n2;
    x_start = x0;
    y1 = zeros(neq,1);

    %1 - первый индекс в нулевой группе
    % n0 - последний индекс во нулевой группе
    n11 = n0 + 1; % первый индекс в первой группе
    n12 = n0 + n1; % последний индекс в первой группе
    n21 = n12 + 1; % первый индекс во второй группе
    n22 = n12 + n2; % последний индекс во второй группе

    % переменные для компенсации
    z = zeros(neq,1);
    term = zeros(neq,1);
    newy = zeros(neq,1);

    % аргументы K - X, Y
    for j = 1:N
```

```

% Первый этап
K1 = zeros(neq,1);
% Группа 0
X = x0; % начальная точка
Y = y0; % начальная точка
for i = 1:n0
    K1(i) = h * f{i}(X,Y);
end

% Группа 1
X = x0; % начальная точка
Y = y0; % начальная точка
for i = n11:n12
    K1(i) = h * f{i}(X,Y);
end

% Группа 2
X = x0 + 1/6 * h;
Y = y0 + 1/6 * K1;
for i = n21:n22
    K1(i) = h * f{i}(X,Y);
    Y(i) = y0(i) + 1/6 * K1(i);
end

% Второй этап
K2 = zeros(neq,1);
% Группа 0
X = x0 + 1/3 * h;
Y = y0 + 1/3 * K1;
for i = 1:n0
    K2(i) = h * f{i}(X,Y);
end

```

```

% Грунна 1
X = x0 + 1/3 * h;
Y(1:n0) = y0(1:n0)
          + 1/6 * K1(1:n0)
          + 1/6 * K2(1:n0); % По грунне 0
Y(n21:n22) = y0(n21:n22)
            + 1/3 * K1(n21:n22); % По грунне 2

for i = n11:n12
    K2(i) = h * f{i}(X,Y);
    Y(i) = y0(i) + 1/6 * K1(i) + 1/6 * K2(i);
end

% Грунна 2
X = x0 + 2/3 * h;
Y(1:n0) = y0(1:n0)
          - 1/12 * K1(1:n0)
          + 3/4 * K2(1:n0);
Y(n11:n12) = y0(n11:n12)
            - 1/12 * K1(n11:n12)
            + 3/4 * K2(n11:n12);

for i=n21:n22
    K2(i) = h * f{i}(X,Y);
    Y(i) = y0(i) + 1/2 * K1(i) + 1/6 * K2(i);
end

% Третьй этап
K3 = zeros(neq,1);
%грунна 0
X = x0 + 2/3 * h;
Y(1:n0) = y0(1:n0)
          - 1/3 * K1(1:n0) + K2(1:n0);

```

```

Y(n11:n12) = y0(n11:n12)
            - 1/3 * K1(n11:n12) + K2(n11:n12);
Y(n21:n22) = y0(n21:n22)
            + 4/9 * K1(n21:n22)
            + 2/9 * K2(n21:n22);
for i = 1:n0
    K3(i) = h * f{i}(X,Y);
end

% Грунна 1
X = x0 + 5/6 * h;
Y(1:n0) = y0(1:n0)
        + 5/48 * K1(1:n0)
        + 5/12 * K2(1:n0)
        + 5/16 * K3(1:n0); % По групне 0
Y(n21:n22) = y0(n21:n22)
        + 5/12 * K1(n21:n22)
        + 5/12 * K2(n21:n22); % По групне 2

for i = n11:n12
    K3(i) = h * f{i}(X,Y);
    Y(i) = y0(i)
        + 1/24 * K1(i)
        + 5/8 * K2(i)
        + 1/6 * K3(i);
end

% Грунна 2
X = x0 + h;
Y(1:n0) = y0(1:n0)
        + K1(1:n0)
        - 5/4 * K2(1:n0)
        + 5/4 * K3(1:n0);

```

```

Y(n11:n12) = y0(n11:n12)
            + 3/4 * K1(n11:n12)
            - 5/12 * K2(n11:n12)
            + 2/3 * K3(n11:n12);

for i=n21:n22
    K3(i) = h * f{i}(X,Y);
    Y(i) = y0(i) + 1/6 * K1(i) + 5/6 * K2(i);
end

% Четвёртый этап
K4 = zeros(neq,1);
% группа 0
X = x0 + h;
Y(1:n0) = y0(1:n0)
          + K1(1:n0)
          - K2(1:n0)
          + K3(1:n0);
Y(n11:n12) = y0(n11:n12)
            + 4/5 * K1(n11:n12)
            - 1/3 * K2(n11:n12)
            + 8/15 * K3(n11:n12);
Y(n21:n22) = y0(n21:n22)
            + 1/3 * K1(n21:n22)
            + 2/3 * K2(n21:n22);
for i = 1:n0
    K4(i) = h * f{i}(X,Y);
end

% Итог
term(1:n0) = 1/8 * (K1(1:n0)
                  + 3 * K2(1:n0)
                  + 3 * K3(1:n0)

```

```

        + K4(1:n0)) + z(1:n0);
term(n11:n12) = 1/10 * (K1(n11:n12)
        + 5 * K2(n11:n12)
        + 4 * K3(n11:n12)) + z(n11:n12);
term(n21:n22) = 1/10 * (4 * K1(n21:n22)
        + 5 * K2(n21:n22)
        + K3(n21:n22)) + z(n21:n22);

newy(1:n0) = y0(1:n0) + term(1:n0);
newy(n11:n12) = y0(n11:n12) + term(n11:n12);
newy(n21:n22) = y0(n21:n22) + term(n21:n22);

z(1:n0) = term(1:n0)
        - (newy(1:n0) - y0(1:n0));
z(n11:n12) = term(n11:n12)
        - (newy(n11:n12) - y0(n11:n12));
z(n21:n22) = term(n21:n22)
        - (newy(n21:n22) - y0(n21:n22));

y1(1:n0) = newy(1:n0);
y1(n11:n12) = newy(n11:n12);
y1(n21:n22) = newy(n21:n22);

x0 = x_start + j*h;
y0 = y1; % Для следующего шага
end

y_final = y1;
end

```