Явные s - стадийные методы Рунге-Кутты

1.1 Общее описание алгоритма

Явные методы Рунге-Кутты - класс численных методов решения задачи Коши для *обыкновенных дифференциальных уравнений (ОДУ)* и их систем.
Далее будут рассмотрены численные методы решения задачи Коши для обыкновенных дифференциальных уравнений (ОДУ) вида

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = f(t,u), \qquad t > 0$$
$$u(0) = u_0$$

или систем ОДУ

$$\frac{\mathbf{d}u(t)}{\mathbf{d}t} = f(t,u), \qquad \text{где } u = \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_m \end{pmatrix}, f = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_m \end{pmatrix}$$
$$u(0) = u_0$$

— векторы столбцы искомых функций и правых частей соответственно.

Суть алгоритма в том, что в точках *расчетной области* вычисляется искомое решение. Для простоты будет рассмотрена равномерная сетка. (В реальных расчетах применяются и неравномерные сетки.)
Введем сеточную функцию u_τ определенную в узлах сетки и представляющую собой совокупность приближенных значений искомой функции, и f^τ — значения правой части в узлах сетки.
Задача состоит в построении численного алгоритма вычисления сеточной функции, приближающей решение ОДУ.

1.2 Обобщение для ОДУ порядка выше первого

$$\frac{d^m u}{d^m t} = g(t, u, \frac{du}{dt}, ..., \frac{d^{m-1}u}{d^{m-1}t}), \quad t > 0$$

Н.У.:

$$u(0) = a_0, \ \frac{du(0)}{dt} = a_1, \ \ldots, \ \frac{d^{m-1}u(0)}{d^{m-1}t} = a_{m-1}$$

положим замену

$$u_1 = u, \ u_2 = \frac{du_1}{dt}, \ u_3 = \frac{du_2}{dt}, \ \ldots, \ u_m = \frac{du_{m-1}}{dt}$$

$$\frac{du_m}{dt} = g(1, u_1, u_2, ..., u_m)$$

и соответствующие ей начальные условия: $u_i(0) = a_{i-1}, \quad i = 1, 2, \ldots, m$.

Такой заменой задача Коши для m-ой производной приводится к системе дифференциальных уравнений, указанной выше.

1.3 Математическое описание алгоритма

Распространенный способ получения простейших одношаговых расчетных схем для численного решения уравнения (1) заключается в следующем. Из формулировки задачи следует, что

$$\int_{t_n}^{t_n+\tau} \frac{du}{dt} dt = \int_{t_n}^{t_n+\tau} f(t, \ u) dt,$$

или

$$u(t_n + \tau) = u(t_n) + \int_{t_n}^{t_n+\tau} f \ dt.$$

Интеграл в правой части может быть вычислен с помощью явных s-стадийных методов Рунге-Кутты.

S-стадийный явный метод для численного решения задачи Коши для обыкновенного дифференциального уравнения (1):

$$\begin{aligned}
&k_1 = f(t_n, u_n), \\
&k_2 = f(t_n + c_2\tau, u_n + \tau a_{21}k_1), \\
&k_3 = f(t_n + c_3\tau, u_n + \tau(a_{31}k_1 + a_{32}k_2)), \\
&\ldots, \\
&k_s = f(t_n c_s\tau, u_n + \tau(a_{s1}k_1 + \ldots + a_{s,s-1}k_{s-1})), \\
&u_{n+1} = u_n + \tau(b_1k_1 + \ldots + b_sk_s)
\end{aligned}$$

где k_j— промежуточные вспомогательные величины (квадратурная формула).

Коэффициенты, определяющие конкретный метод, могут быть представлены в виде *таблицы Бутчера* (Таблица 1). Нулевые коэффициенты a_{jk}, при $j \geq i$, как правило, в таблице Бутчера для явного метода не указывают.

Таблица 1

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\cdots$	$\cdots$	$\cdots$	$\cdots$		
c_s	a_{s1}	a_{s2}	$\cdots$	a_{ss-1}	
	b_1	b_2	$\cdots$	b_{s-1}	b_s

Здесь коэффициенты c_j – узлы некой квадратурной формулы, b_j – веса соответствующей квадратурной формулы, набор коэффициентов a_{jk} необходим для экстраполяции значений функции правой части в точки, связанные с узлами квадратуры (j = 1,2,...,s, k = 1,2,3,...,s-1).

Обычно также используют условие, предложенное Куттой без объяснений и не являющееся обязательным, но упрощающее вывод условий порядка аппроксимации для многостадийных методов:

$$c_n = \sum_{j}^{s} a_{nj}$$

1.4 Вычислительное ядро алгоритма

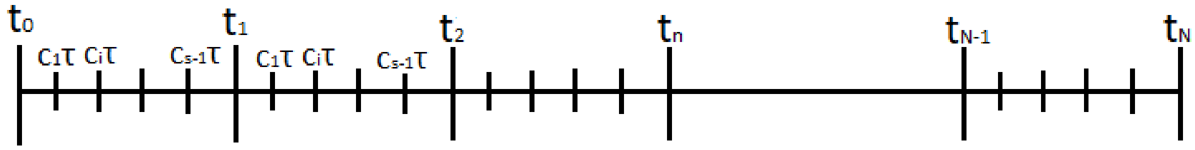
Вычислительное ядро алгоритма можно считать состоящим из двух частей - вычисление промежуточных значений k_i для каждого шага на заданной сетке c_i , где i = 1, 2, …, s и вычисление линейной комбинации $\sum_{i=1}^{s} k_i \cdot b_i$. В первой части вычислительное ядро составляют последовательности операций вычисления функции f в точке, а также умножение и сложение. Во второй части в вычислительном ядре остаются только последовательности умножения и сложения.

Сюда картинку

1.5 Макроструктура алгоритма

Макроструктура алгоритма может быть представлена как совокупность последовательных вычислений значений функции в точках и их линейной комбинации.

1.6 Схема реализации последовательного алгоритма



1) $t_0 = c_1\tau = 0$

$u(0) = u_0$ (начальные условия)

2) Последовательность исполнения метода следующая (для n-го шага):

1. Вычисляем величину k_1, пользуясь начальными условиями

и таблицей Бутчера ($c_1 = 0$):

$k_1 = f(t_n, u_n)$

2. Вычисляем величину k_2, пользуясь предыдущими расчетами

и таблицей Бутчера ($c_2,\ a_{21}$):

$k_2 = f(t_n + c_2\tau, u_n + \tau a_{21}k_1)$

3. Вычисляем величину k_i, пользуясь предыдущими расчетами

и таблицей Бутчера ($c_i,\ a_{i1},\ a_{i2}...a_{i,i-1}$):

$k_i = f(t_n + c_i\tau, u_n + \tau a_{i1}k_1 + ... + \tau a_{i,i-1}k_{i-1})$

4. Вычисляем величину k_s, пользуясь предыдущими расчетами

и таблицей Бутчера ($c_s,\ a_{s1},\ a_{s2}...a_{s,s-1}$):

$k_s = f(t_n + c_s\tau, u_n + \tau a_{s1}k_1 + ... + \tau a_{s,s-1}k_{s-1})$

5. Вычисляем величину u_{n+1}, t_{n+1} , пользуясь предыдущими расчетами:

$u_{n+1} = u_n + \tau(b_1k_1 + ... + b_sk_s)$

$t_{n+1} = t_n + c_s\tau$

3) Далее считаем тоже самое для шагов n+1, n+2, …, N.

1.7 Последовательная сложность алгоритма

Для вычисления значений функции U(t) в расчетной области за N шагов S - стадийным методом Рунге - Кутты требуется:

- N * S вычислений значений функции в точке,
- N * [(S*S/2 + 3/2*S - 1) + S * P(f(t,u))] сложений,
- N * [(S*S/2 + 5/2*S - 1) * S * M(f(t,u))] умножений.

Таким образом, при классификации по последовательной сложности, явный метод Рунге - Кутты относится к алгоритмам *с линейной сложностью по числу шагов N и квадратичная по числу стадий S*.

1.8 Информационный граф

Информационный граф одного шага явного метода Рунге - Кутты для S = 3 представлен на рис.1, последующие шаги имеют тот же вид. Как следует из анализа графа, алгоритм является практически последовательным как при выполнении шагов, так и при выполнению стадий.

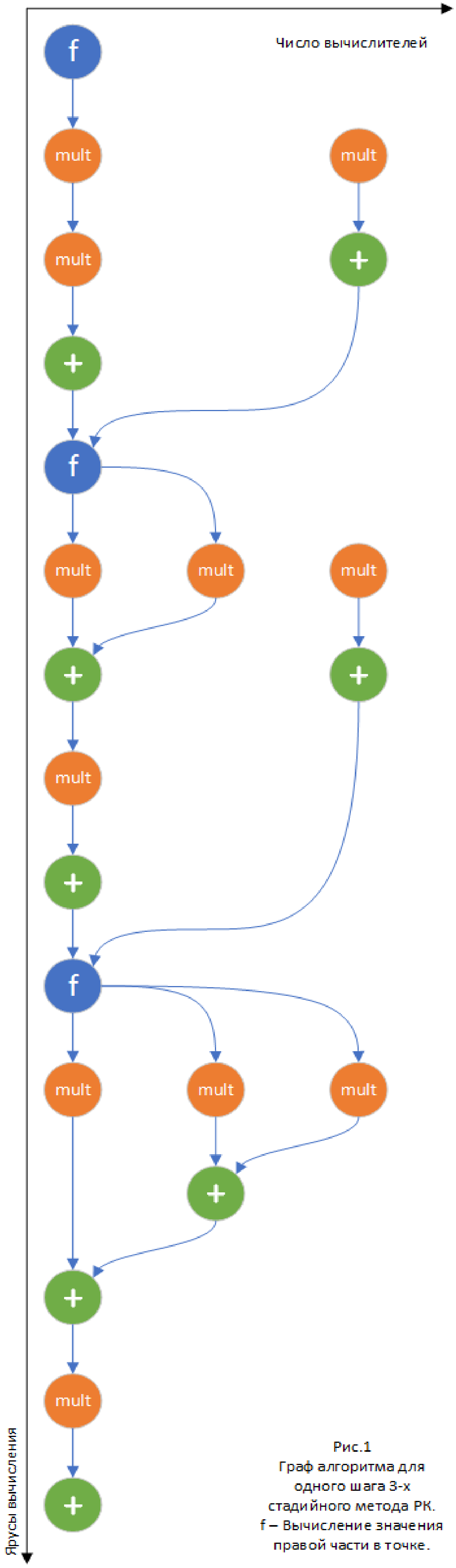


Рис.1
Граф алгоритма для одного шага 3-х стадийного метода РК.
f – Вычисление значения правой части в точке.

1.9 Описание ресурса параллелизма

Для решения дифференциального уравнения методом Рунге - Кутты за N шагов с S стадиями в параллельном варианте:

ЯПФ - Ярусно-параллельная форма графа алгоритма

Высота ЯПФ = N * (3 + $log_2 S$ + 3 * (S-2) + $\sum_{s=2}^{S} log_2(s-1)$)

Ширина ЯПф = S

Общее число операций ~ N * (3 + 2 * S + $S * log_2 S + \sum_{s=2}^{S} (s-1) * log_2(s-1)$)

Доля последовательных операций ~ 1/S

Максимальное ускорение = S

На рис.2 представлен параллельный граф алгоритма S-й стадии, но при высоком теоретическом ускорении требуется очень частая пересылка данных между процессами, что значительно снижает ускорение. На практике скорость выполнения алгоритма на параллельной машине не превышает последовательную реализацию.

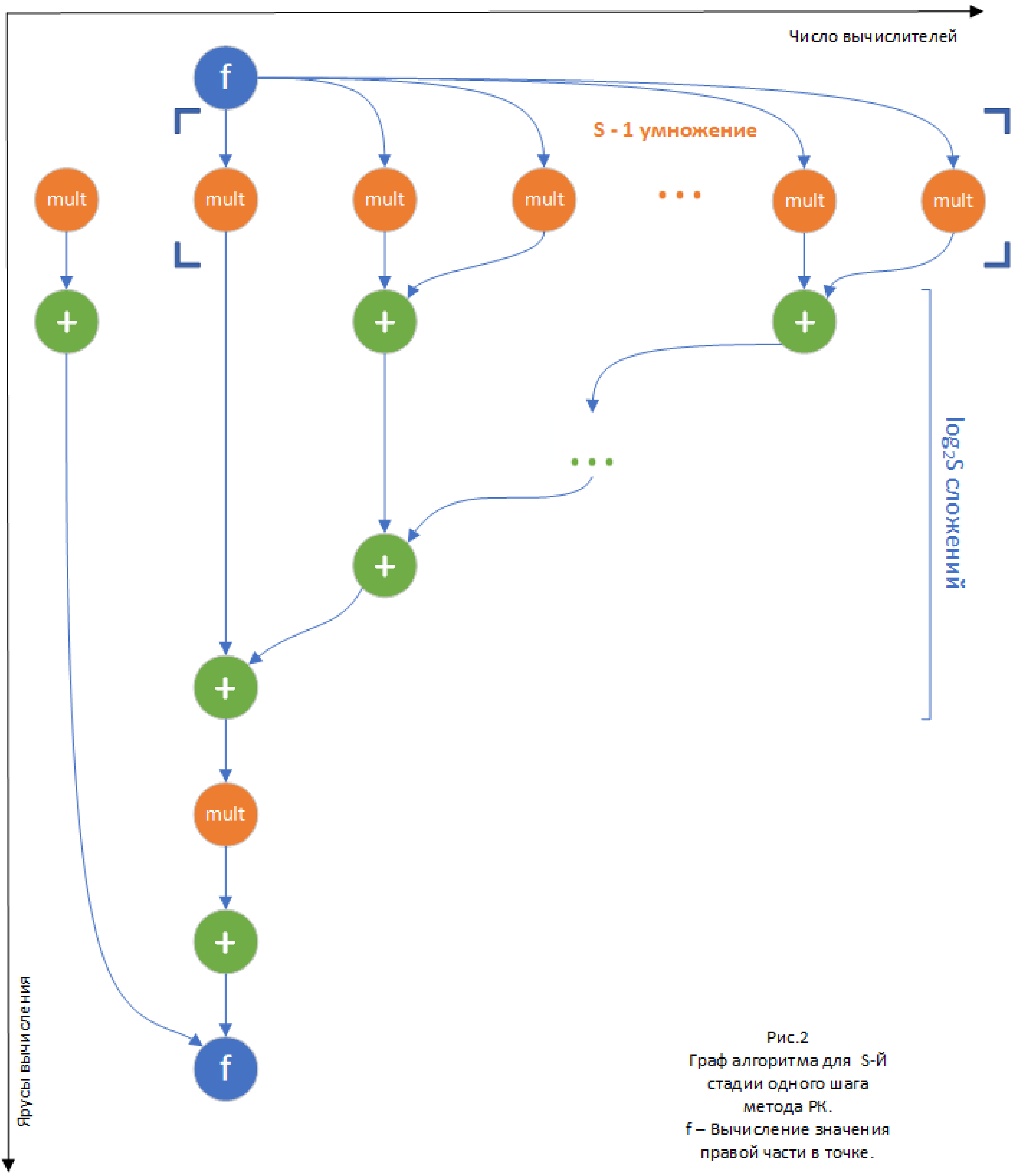


Рис.2
Граф алгоритма для S-й
стадии одного шага
метода РК.
f – Вычисление значения
правой части в точке.

1.10 Входные и выходные данные алгоритма

Входные данные: функция f(t,u) - правая часть дифференциального уравнения, u(0) - начальные условия.

Выходные данные: $((x_0, u_0),\ (x_1, u_1),\ \ldots,\ (x_n, u_n),\ (x_{n+1}, u_{n+1}),\ \ldots,\ (x_N, u_N))$ - вектор значений функций в точках расчётной сетки.

Объём выходных данных: 2N.

1.11 Свойства алгоритма

Алгоритм не параллелится.
Алгоритм в рамках выбранной версии полностью детерминирован.

1.12 Коэффициенты Бутчера в зависимости от степени аппроксимации

Первый порядок аппроксимации: метод Эйлера (таблица 2).

Таблица 2

0	0
	1

Второй порядок аппроксимации: метод Эйлера с пересчетом (таблица 3).

Таблица 3

0		
1/2	1/2	
	0	1

Третий порядок аппроксимации: метод Хойна (таблица 4).

Таблица 4

0			
1/3	1/3		
2/3	0	2/3	
	1/4	0	3/4

Третий порядок аппроксимации: метод Рунге–Кутты (таблица 5).

Таблица 5

0			
1/2	1/2		
1	0	1	
	1/6	2/3	1/6

Четвёртый порядок аппроксимации: классический метод Рунге–Кутты (таблица 6).

Таблица 6

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

Четвёртый порядок аппроксимации: правило трех восьмых (таблица 7). Этот метод имеет, по-видимому, наименьшую погрешность среди явных схем Рунге–Кутты четвертого порядка аппроксимации.

Таблица 7

0				
1/3	1/3			
2/3	–1/3	1		
1	1	–1	1	
	1/8	3/8	3/8	1/8

Шестой порядок аппроксимации: метод Бутчера (таблице 8).

Таблица 8

0							
1/2	1/2						
2/3	2/9	4/9					
1/3	7/36	2/9	–1/12				
5/6	–35/144	–55/36	35/48	15/8			
1/6	–1/360	–11/36	–1/8	1/2	1/10		
1	–41/260	22/13	43/156	–118/39	32/195	80/39	
	13/200	0	11/40	11/40	4/25	4/25	13/200

1.13 Пример (N = 2, s = 4)

Найти приближённое решение задачи Коши $y' - 3x^2 y - x^2 e^{x^3} = 0,\ y(0) = 0$ методом

Рунге-Кутты 4 порядка (т.е s = 4) на заданном отрезке $x \in [0;1]$ с шагом h = 0,1.

$$y(x_{n+1} + h) = y(x_n) + \int_{x_n}^{x_n+h} f(x, y)dx$$

Вход: $f(x, y) = 3x^2 y + x^2 e^{x^3}$, $y_1 = 0$

n = 1

$k_1^{(1)} = f(x_1 + c_1 h, y_1) = f(0, 0) = 0,$

$k_2^{(1)} = f(x_1 + c_2 h, y_1 + h a_{21} k_1^{(1)}) = f(h/2, 0) = f(0.05, 0) = 0.0025,$
$k_3^{(1)} = f(x_1 + c_3 h, y_1 + h(a_{31} k_1^{(1)} + a_{32} k_2^{(1)})) = f(h/2, k_2 h/2) = f(0.05, 0.0000625) = 0.002501,$
$k_4^{(1)} = f(x_1 + c_4 h, y_1 + h(a_{41} k_1^{(1)} + a_{42} k_2^{(1)} + a_{43} k_3^{(1)})) = f(h, hk_3^{(1)}) = f(0.1, 0.0002501) = 0.010018,$
$y_2 = y_1 + h(b_1 k_1^{(1)} + b_2 k_2^{(1)} + b_3 k_3^{(1)} + b_4 k_4^{(1)}) = 0 + 0.1(0 + 0.000833 + 0.000834 + 0.00167) = 0.000333$
$x_2 = x_1 + c_4 h = 0.1$

n = 2
$k_1^{(2)} = f(x_2 + c_1 h, y_2) = f(0.1, 0.000333) = 0.01002,$
$k_2^{(2)} = f(x_2 + c_2 h, y_2 + h a_{21} k_1^{(2)}) = f(x_2 + h/2, y_2) = f(0.15, 0.000333) = 0.022632,$
$k_3^{(2)} = f(x_2 + c_3 h, y_2 + h(a_{31} k_1^{(2)} + a_{32} k_2^{(2)})) = f(x_2 + h/2, y_2 + k_2^{(2)} h/2) = f(0.15, 0.0.001465) = 0.022675,$
$k_4^{(2)} = f(x_2 + c_4 h, y_2 + h(a_{41} k_2^{(2)} + a_{42} k_2^{(2)} + a_{43} k_3^{(2)})) = f(x_2 + h, y_2 + hk_3^{(2)}) = f(0.2, 0.002601) = 0,040634,$
$y_3 = y_2 + h(b_1 k_1^{(2)} + b_2 k_2^{(2)} + b_3 k_3^{(2)} + b_4 k_4^{(2)}) = 0.002687$
$x_3 = x_2 + c_4 h = 0.2$

Выход: ((0,0), (0.1,0.000333), (0.2,0.002687))