**Явные s - стадийные методы Рунге-Кутты**

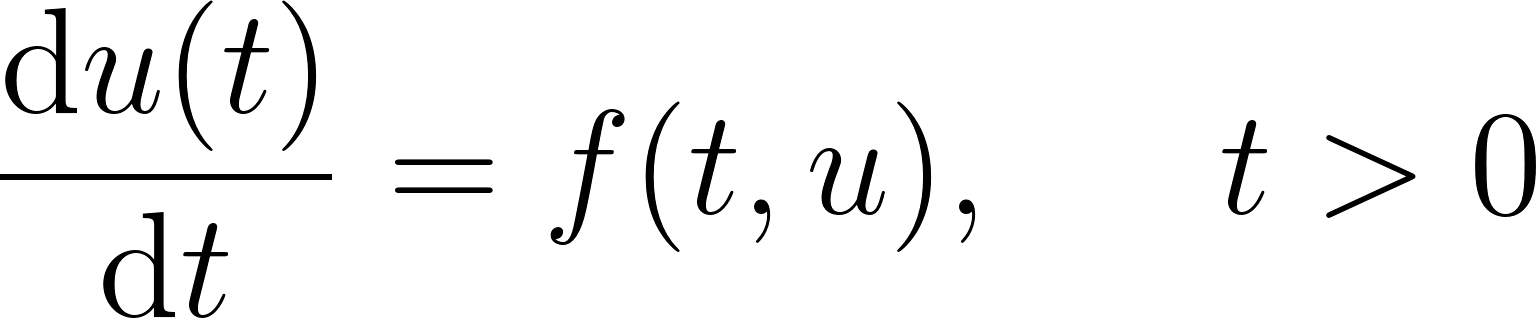
1 Свойства и структура алгоритма

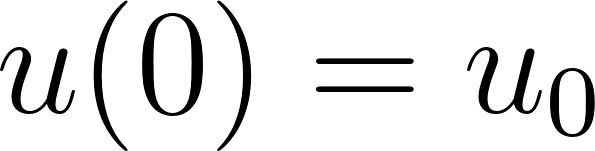
* 1.1 Общее описание алгоритма
* 1.2 Обобщение для ОДУ порядка выше первого
* 1.3 Математическое описание алгоритма
* 1.4 Вычислительное ядро алгоритма
* 1.5 Макроструктура алгоритма
* 1.6 Схема реализации последовательного алгоритма
* 1.7 Последовательная сложность алгоритма
* 1.8 Информационный граф
* 1.9 Описание ресурса параллелизма алгоритма
* 1.10 Входные и выходные данные алгоритма
* 1.11 Свойства алгоритма
* 1.12 Коэффициенты Бутчера в зависимости от степени аппроксимации
* 1.13 Пример

**1.1 Общее описание алгоритма**

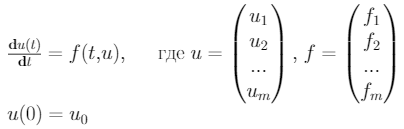
**Явные методы Рунге-Кутты** - класс численных методов решения задачи Коши для *обыкновенных дифференциальных уравнений (ОДУ)* и их систем.

Далее будут рассмотрены численные методы решения задачи Коши для обыкновенных дифференциальных уравнений (ОДУ) вида

[](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7B%5Ctext%7Bd%7Du(t)%7D%7B%5Ctext%7Bd%7Dt%7D%20%3D%20f(t%2Cu)%2C%20%5C%3B%5C%3B%5C%3B%5C%3B%5C%3B%20t%3E0%0)

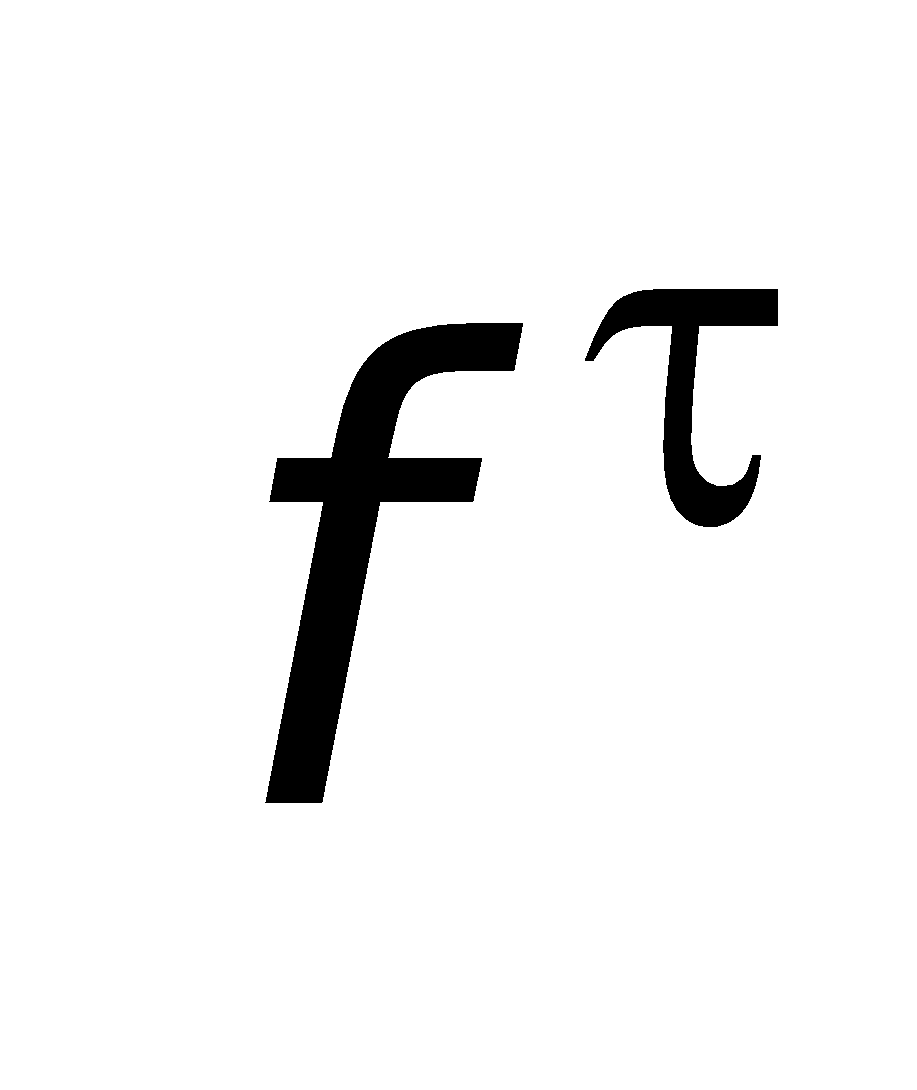
[](https://www.codecogs.com/eqnedit.php?latex=u(0)%3Du_0%0)

или систем ОДУ



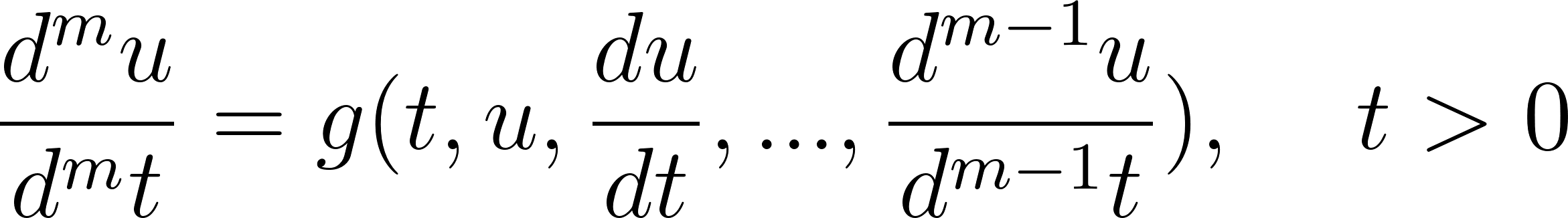
— векторы столбцы искомых функций и правых частей соответственно.

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.)

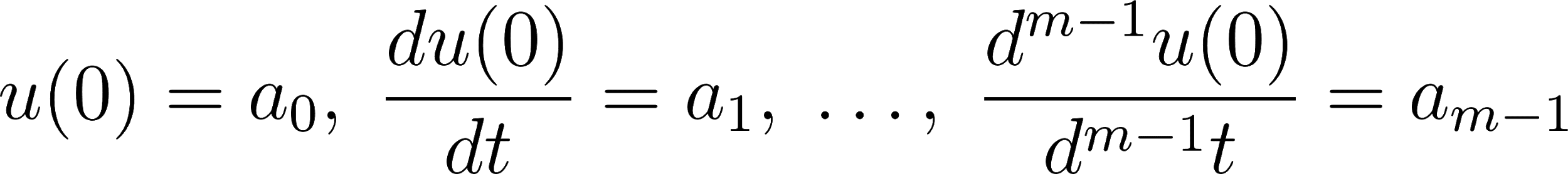
Введем сеточную функцию [](https://www.codecogs.com/eqnedit.php?latex=u_%5Ctau%0)определенную в узлах сетки и представляющую собой совокупность приближенных значений искомой функции, и — значения правой части в узлах сетки.

Задача состоит в построении численного алгоритма вычисления сеточной функции, приближающей решение ОДУ.

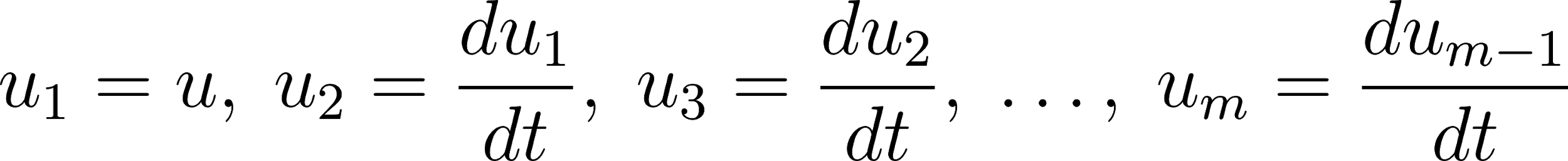
**1.2 Обобщение для ОДУ порядка выше первого**

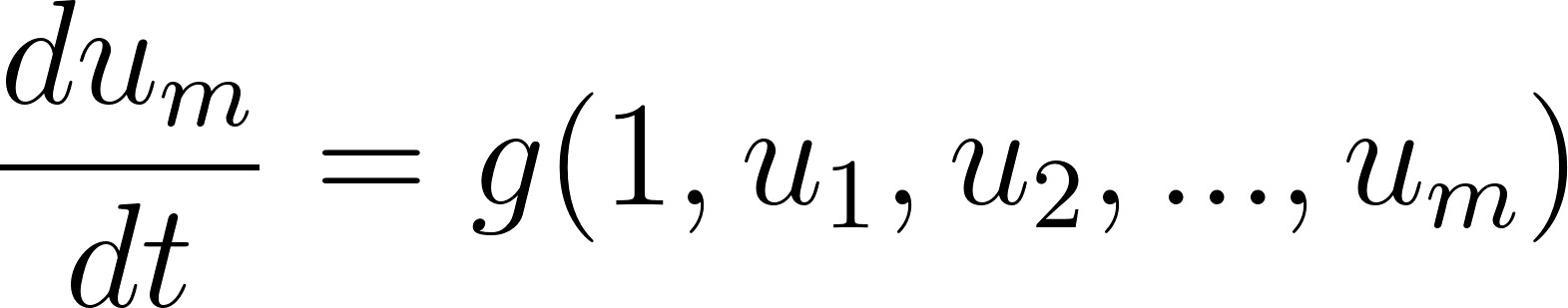
[](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7Bd%5E%7Bm%7Du%7D%7Bd%5E%7Bm%7Dt%7D%3Dg(t%2Cu%2C%5Cfrac%7Bdu%7D%7Bdt%7D%2C...%2C%5Cfrac%7Bd%5E%7Bm-1%7Du%7D%7Bd%5E%7Bm-1%7Dt%7D)%2C%5C%3B%5C%3B%5C%3B%5C%3Bt%3E0%0)

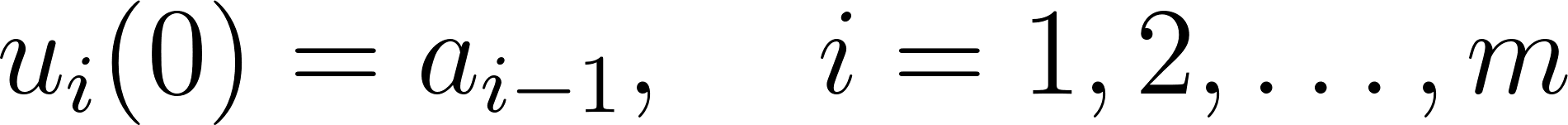
Н.У.:

[](https://www.codecogs.com/eqnedit.php?latex=u(0)%3Da_0%2C%20%5C%3B%20%5Cfrac%7Bdu(0)%7D%7Bdt%7D%3Da_1%2C%5C%3B%5Cldots%2C%20%5C%3B%5Cfrac%7Bd%5E%7Bm-1%7Du(0)%7D%7Bd%5E%7Bm-1%7Dt%7D%3Da_%7Bm-1%7D%0)

положим замену

[](https://www.codecogs.com/eqnedit.php?latex=u_1%3Du%2C%5C%3Bu_2%3D%5Cfrac%7Bdu_1%7D%7Bdt%7D%2C%5C%3Bu_3%3D%5Cfrac%7Bdu_2%7D%7Bdt%7D%2C%5C%3B%5Cldots%2C%20%5C%3Bu_m%3D%5Cfrac%7Bdu_%7Bm-1%7D%7D%7Bdt%7D%0)

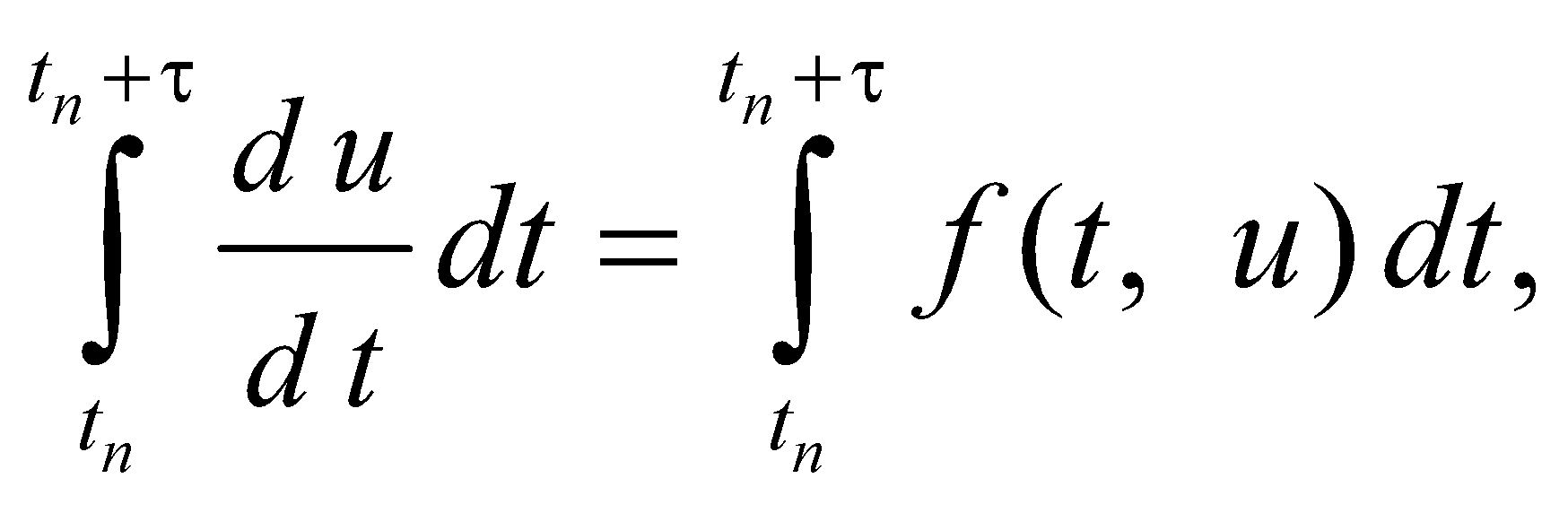
[](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7Bdu_%7Bm%7D%7D%7Bdt%7D%3Dg(1%2C%20u_1%2C%20u_2%2C%20...%2C%20u_m)%0)

и соответствующие ей начальные условия: [](https://www.codecogs.com/eqnedit.php?latex=u_i(0)%3Da_%7Bi-1%7D%2C%5C%3B%5C%3B%5C%3B%5C%3Bi%3D1%2C2%2C%5Cldots%2Cm%0).

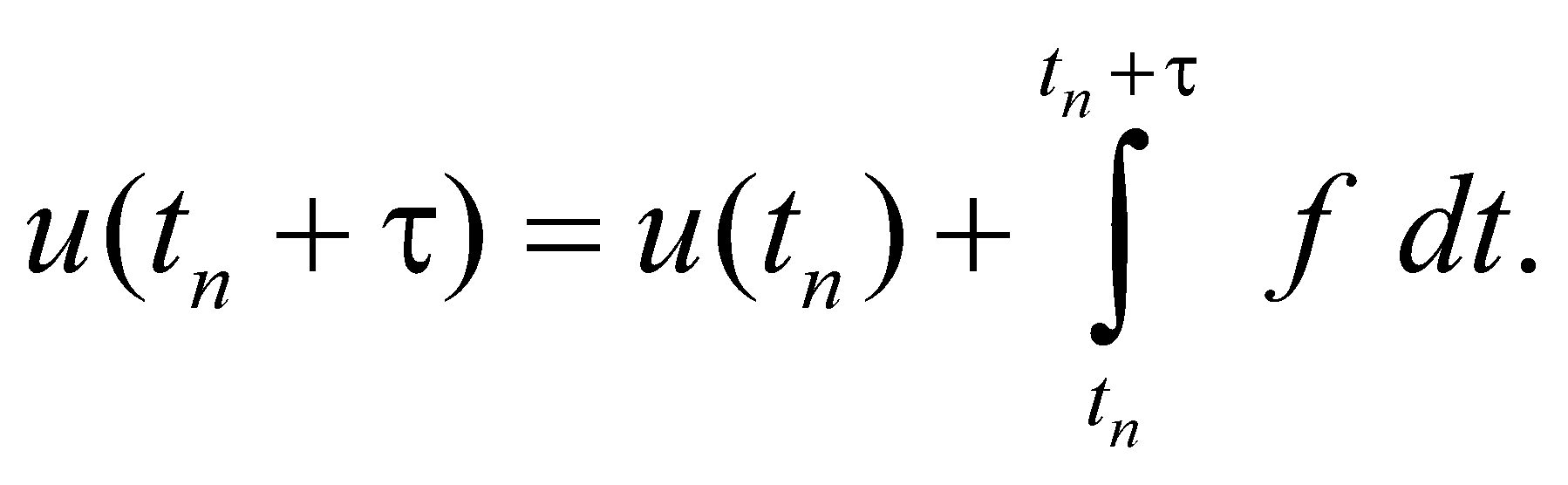
Такой заменой задача Коши для m-ой производной приводится к системе дифференциальных уравнений, указанной выше.

**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

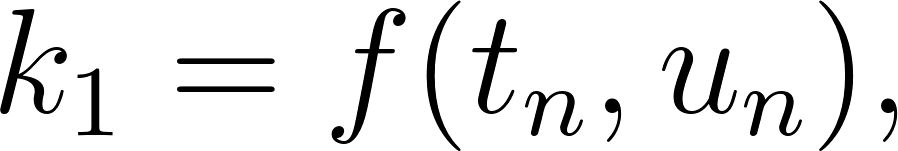


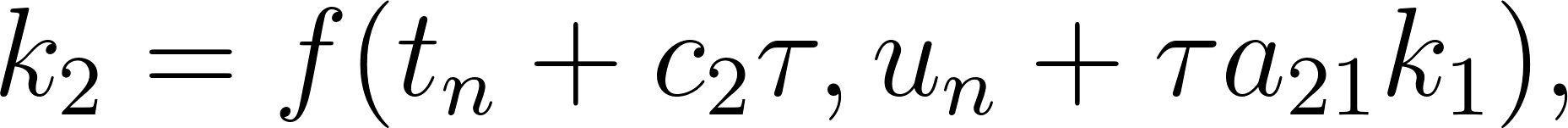
или

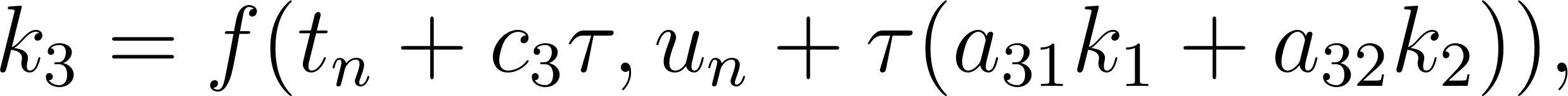


Интеграл в правой части может быть вычислен с помощью явных s-стадийных методов Рунге-Кутты.

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

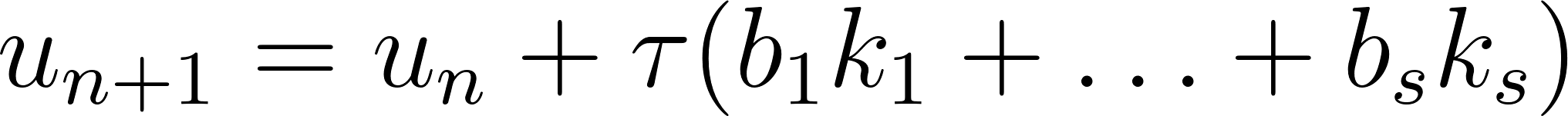
[](https://www.codecogs.com/eqnedit.php?latex=k_1%3Df(t_n%2Cu_n)%2C%0)

[](https://www.codecogs.com/eqnedit.php?latex=k_2%3Df(t_n%2Bc_2%5Ctau%2Cu_n%2B%5Ctau%20a_%7B21%7Dk_1)%2C%0)

[](https://www.codecogs.com/eqnedit.php?latex=k_3%3Df(t_n%2Bc_3%5Ctau%2Cu_n%2B%5Ctau(a_%7B31%7D%20k_1%20%2B%20a_%7B32%7D%20k_2))%2C%0)

[](https://www.codecogs.com/eqnedit.php?latex=%5Cldots%2C%0)

[](https://www.codecogs.com/eqnedit.php?latex=k_s%3Df(t_n%20c_s%20%5Ctau%2Cu_n%2B%5Ctau(a_%7Bs1%7D%20k_1%20%2B%20%5Cldots%20%2B%20a_%7Bs%2Cs-1%7D%20k_%7Bs-1%7D))%2C%0)

[](https://www.codecogs.com/eqnedit.php?latex=u_%7Bn%2B1%7D%3Du_n%2B%5Ctau(b_1%20k_1%20%2B%20%5Cldots%20%2B%20b_s%20k_s)%0)

где *k*ⱼ— промежуточные вспомогательные величины (квадратурная формула).

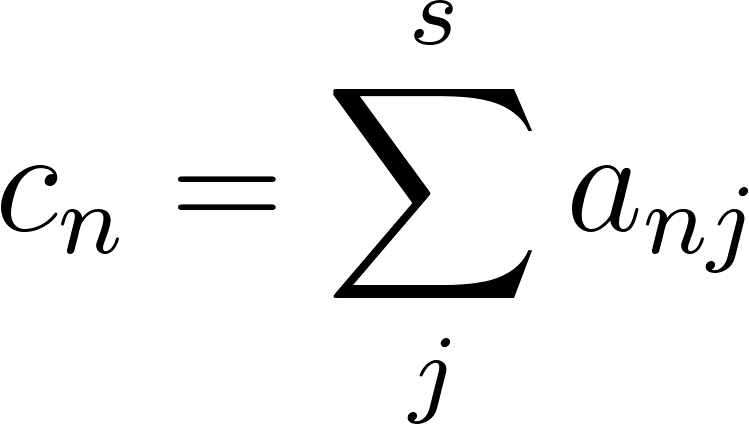
Коэффициенты, определяющие конкретный метод, могут быть представлены в виде *таблицы Бутчера* (Таблица 1). Нулевые коэффициенты *a*ⱼₖ, при *j*≥ *i*, как правило, в таблице Бутчера для явного метода не указывают.

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| … | … | … | … |  |  |
|  |  |  | … |  |  |
|  |  |  | … |  |  |

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:

[](https://www.codecogs.com/eqnedit.php?latex=c_n%20%3D%5Csum_%7Bj%7D%5E%7Bs%7D%20a_%7Bnj%7D%0)

**1.4 Вычислительное ядро алгоритма**

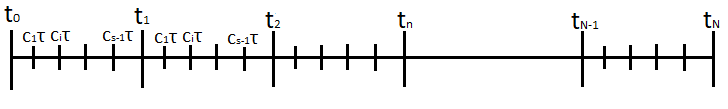
Вычислительное ядро алгоритма можно считать состоящим из двух частей - вычисление промежуточных значений kᵢ для каждого шага на заданной сетке cᵢ , где i = 1, 2, …, s и вычисление линейной комбинации . В первой части вычислительное ядро составляют последовательности операций вычисления функции f в точке, 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**

****

(начальные условия)

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

1. Вычисляем величину , пользуясь начальными условиями

и таблицей Бутчера ():

2. Вычисляем величину , пользуясь предыдущими расчетами

и таблицей Бутчера ():

3. Вычисляем величину , пользуясь предыдущими расчетами

и таблицей Бутчера ():

4. Вычисляем величину , пользуясь предыдущими расчетами

и таблицей Бутчера ():

5. Вычисляем величину , пользуясь предыдущими расчетами:

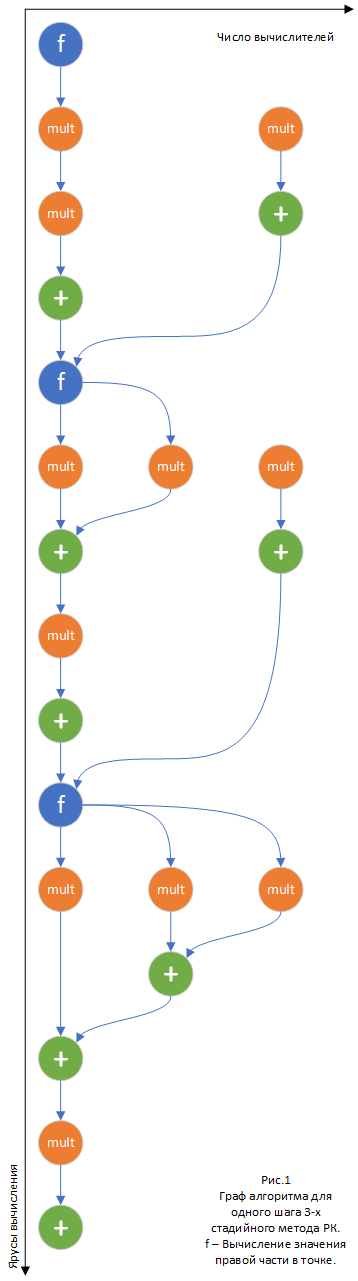
3) Далее считаем тоже самое для шагов n+1, n+2, …, N.

**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.

**1.9 Описание ресурса параллелизма**

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

ЯПФ - Ярусно-параллельная форма графа алгоритма

Высота ЯПФ = N \* (3 + + 3 \* (S-2) + )

Ширина ЯПф = S

Общее число операций ~ N \* (3 + 2 \* S + + )

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.

## 

## 

## **1.10 Входные и выходные данные алгоритма**

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

**Выходные данные**: () - 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 **** методом Рунге-Кутты 4 порядка (т.e s = 4) на заданном отрезке с шагом h = 0,1.

Вход:

n = 1

n = 2

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