МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и инворматики
Кафедра вычислительной математики

Отчет по лабораторной работе 5
"Методы решения ОДУ"
Вариант 5

Выполнил:
Карпович Артём Дмитриевич
студент 3 курса 7 группы

Преподаватель:
Репников Василий Иванович

Минск, 2024

Методы решения ОДУ

Постановка задач

1. Построить экстраполяционный метод Адамса пятого порядка (дать представление в двух формах).
2. Определить порядок точности метода
$$\begin{cases} y_{j+1} = y_j + \frac{\tau}{6}(f_j + 4f_{j+\frac{1}{2}} + f_{j+1}), \\ y_j = y_{j+1} - \tau f_{j+\frac{1}{2}}, \\ y_{j+\frac{1}{2}} = y_{j+1} - \frac{\tau}{2} f_{j+1}. \end{cases}$$
3. Найти интервал устойчивости метода из п. 1;
4. С заданной точность $\varepsilon = 10^{-4}$ найти решение задачи Коши с помощью: а) метода последовательного повышения порядка точности четвертого порядка; б) явного метода Рунге-Кутта третьего порядка; в) интерполяционого метода Адамса третьего порядка
$$\begin{cases} u' = \frac{1}{5}\begin{pmatrix} -104 & -198 \\ -198 & -401 \end{pmatrix} u, t \in [0;5], \\ u(0) = \begin{pmatrix} 5 \\ 10 \end{pmatrix}. \end{cases}$$

Задача 1

Экстраполяционным методом Адамса называется метод, получаемый по формуле
$$y_{j+1} = y_j + h\sum_{i=0}^{k} A_i f(x_{j-i}, y_{j-i}),$$
где коэффициенты вычисляются через интегралы
$$A_i = \frac{(-1)^i}{i!(k-i)!}\int\limits_0^1 \frac{\alpha(\alpha+1)\dots(\alpha+k)}{\alpha+i}d\alpha.$$
В случае метода пятого порядка, нам необходимо посчитать интегралы A_i для всех $i = \overline{0,k}$. Поскольку нас интересует метод пятого порядка, то выберем $k = 4$
$$i = 0 : A_0 = \frac{1}{24}\int_0^1 (\alpha+1)(\alpha+2)(\alpha+3)(\alpha+4)d\alpha = \frac{1901}{720} \approx 2.64;$$
$$i = 1 : A_1 = -\frac{1}{6}\int_0^1 \alpha(\alpha+2)(\alpha+3)(\alpha+4)d\alpha = -\frac{1387}{360} \approx -3.85;$$
$$i = 2 : A_2 = \frac{1}{4}\int_0^1 \alpha(\alpha+1)(\alpha+3)(\alpha+4)d\alpha = \frac{109}{30} \approx 3.63;$$
$$i = 3 : A_3 = -\frac{1}{6}\int_0^1 \alpha(\alpha+1)(\alpha+2)(\alpha+4)d\alpha = -\frac{637}{360} \approx -1.77;$$
$$i = 4 : A_4 = \frac{1}{24}\int_0^1 \alpha(\alpha+1)(\alpha+2)(\alpha+3)d\alpha = \frac{251}{720} \approx 0.35;$$
Проверим вычисленные значения с помощью инструментов Python

[6]:
```
from scipy import integrate
import math

k = 4

def f(x, k, n=4):
    res = 1

    for i in range(n + 1):
       res *= (x + i)

    return res / (x + k)

for i in range(k + 1):
    print(f'Интеграл {i} равен: {((-1)**i) / (math.factorial(i) * math.
 ↪factorial(k - i)) * integrate.quad(f, 0, 1, args=(i,))[0]}')
```

```
Интеграл 0 равен: 2.6402777777777775
Интеграл 1 равен: -3.8527777777777778
Интеграл 2 равен: 3.6333333333333337
Интеграл 3 равен: -1.7694444444444442
Интеграл 4 равен: 0.3486111111111112
```

Таким образом, получаем $A_0 \approx 2.64, A_1 \approx -3.85, A_2 \approx 3.63, A_3 \approx -1.77, A_4 \approx 0.35$. Тогда метод Адамса пятого порядка будет иметь вид

$$y_{j+1} = y_j + h(\frac{1901}{720} f_j - \frac{1387}{360} f_{j-1} + \frac{109}{30} f_{j-2} - \frac{637}{360} f_{j-3} + \frac{251}{720} f_{j-4}).$$

Получим представление метода Адамса пятого порядка через конечные разности, формула для этого метода имеет вид

$$y_{j+1} = y_j + h \sum_{i=0}^{k} C_i \Delta^i f_{j-i},$$

$$C_i = \frac{1}{i!} \int\limits_0^1 \alpha(\alpha+1)\dots(\alpha+i-1) d\alpha.$$

Аналогичным образом, найдем все C_i для $i = \overline{0, k}$, k выбираем то же

$$i = 0 : C_0 = 0;$$

$$i = 1 : C_1 = \int_0^1 \alpha d\alpha = 1;$$

$$i = 2 : C_2 = \frac{1}{2!} \int_0^1 \alpha(\alpha+1) d\alpha = \frac{1}{2} \int_0^1 (\alpha^2 + \alpha) d\alpha = \frac{5}{12};$$

$$i = 3 : C_3 = \frac{1}{3!} \int_0^1 \alpha(\alpha+1)(\alpha+2) d\alpha = \frac{1}{6} \int_0^1 (\alpha^3 + 3\alpha^2 + 2\alpha) d\alpha = \frac{3}{8};$$

$$i = 4 : C_4 = \frac{1}{4!}\int_0^1 \alpha(\alpha+1)(\alpha+2)(\alpha+3)d\alpha = \frac{1}{24}\int_0^1 (\alpha^4+6\alpha^3+11\alpha^2+6\alpha)d\alpha = \frac{1}{24}\Big(\frac{1}{5}+\frac{3}{2}+\frac{11}{3}+3\Big) = \frac{251}{720};$$

Таким образом, можем составить экстраполяционый метод Адамса пятого порядка через конечные разности

$$y_{j+1} = y_j + h\Big(\Delta^1 f_{j-1} + \frac{5}{12}\Delta^2 f_{j-2} + \frac{3}{8}\Delta^3 f_{j-3} + \frac{251}{720}\Delta^4 f_{j-4}\Big).$$

Задача 2

Для определения порядка точности нашего метода первым делом оценим локальные погрешности, определяемые следующим образом

$$r(t_j, \tau) = u(t_{j+1}) - F\big(u(t_{j-q}), \ldots, u(t_j), u(t_{j+1}), \ldots, u(t_{j+s})\big). \tag{2.1}$$

Для удобства переопределим $u(t_j) = u_j$. Тогда, переходя к нашему методу, будем поочередно рассматривать уравнения от меньшего порядка к большему. Приступим к рассмотрению третьего уравнения

$$y_{j+\frac{1}{2}} = y_{j+1} - \frac{\tau}{2} f_{j+1}$$

Тогда, применяя формулу (2.1), получим

$$r(t_j, \tau) = u_{j+\frac{1}{2}} - u_{j+1} + \frac{\tau}{2} f_{j+1}$$

Для оценки погрешности разложим функции по степеням τ

$$u_{j+\frac{1}{2}} \approx u(t_j + \frac{\tau}{2}) = u_j + \frac{\tau}{2\cdot 1!}u'_j + \frac{\tau^2}{4\cdot 2!}u''_j + O(\tau^3);$$

$$u_{j+1} \approx u(t_j + \tau) = u_j + \frac{\tau}{1!}u'_j + \frac{\tau^2}{2!}u''_j + O(\tau^3);$$

Для того, чтобы разложить f_{j+1} по степеням τ воспользуемся тем, что мы решаем задачу Коши вида

$$f(x, u(x)) = u'(x),\ \ u|_{x=x_0} = u_0.$$

Тогда

$$f_{j+1} \approx f(t_j + \tau, u(t_j)) = u'(t_j + \tau) = u'_j + \frac{\tau}{1!}u''_j + \frac{\tau^2}{2!}u'''_j + O(\tau^3).$$

Подставим полученные разложения в рассматриваемое уравнение

$$r(t_j, \tau) \approx u_j + \frac{\tau}{2}u'_j + \frac{\tau^2}{8}u''_j - u_j - \tau u'_j - \frac{\tau^2}{2}u''_j + \frac{\tau}{2}\Big(u'_j + \tau u''_j + \frac{\tau^2}{2}u'''_j\Big) + O(\tau^3) = \frac{\tau}{2}u'_j - \tau u'_j + \frac{\tau}{2}u'_j + \frac{\tau^2}{8}u''_j - \frac{\tau^2}{2}u''_j + \frac{\tau^2}{2}u''_j + \frac{\tau^3}{4}$$

Таким образом, локальная погрешность равна $r(t_j, \tau) = O(\tau^2)$. Перейдем к следующему уравнению

$$y_j = y_{j+1} - \tau f_{j+\frac{1}{2}}.$$

Проделаем ту же процедуру

$$r(t_j, \tau) = u_j - u_{j+1} + \tau f_{j+\frac{1}{2}}$$

Приступим к разложению по степеням τ. Поскольку мы уже раскладывали u_{j+1}, а разложение u_j приведет нас к u_j, то сразу перейдем к разложению третьего слагаемого

$$f_{j+\frac{1}{2}} \approx u'(t_j + \frac{\tau}{2}) = u'_j + \frac{\tau}{2}u''_j + \frac{\tau^2}{8}u'''_j + O(\tau^3).$$

Подставим

$$r(t_j, \tau) \approx u_j - u_j - \tau u'_j - \frac{\tau^2}{2}u''_j + \tau\Big(u'_j + \frac{\tau}{2}u''_j + \frac{\tau^2}{8}u'''_j\Big) + O(\tau^3) = \frac{\tau^3}{8}u'''_j + O(\tau^3) = O(\tau^3).$$

Таким образом, локальная погрешность в этом случае равна $r(t_j, \tau) = O(\tau^3)$. Перейдем к последнему(первому) уравнению

$$r(t_j, \tau) = u_{j+1} - u_j - \frac{\tau}{6}(f_j + 4f_{j+\frac{1}{2}} + f_{j+1}).$$

Аналогично предыдущему уравнению, сразу перейдем к подстановке, учитывая, что $f_j \approx u'_j$

$$r(t_j, \tau) \approx u_j + \tau u'_j + \frac{\tau^2}{2}u''_j - u_j - \frac{\tau}{6}\Big(u'_j + 4\Big(u'_j + \frac{\tau}{2}u''_j + \frac{\tau^2}{8}u'''_j\Big) + u'_j + \tau u''_j + \frac{\tau^2}{2}u'''_j\Big) + O(\tau^3) = O(\tau^3).$$

Таким образом, мы получили, что у нас сократились все слагаемые, поэтому нам необходимо увеличить количество членов разложения, скажем, до третьей степени, сразу выполним подстановку с учетом полученных результатов

$$r(t_j, \tau) \approx \frac{\tau^3}{6}u'''_j - \frac{\tau}{6}\Big(\frac{4\tau^3}{12}u^{(4)}_j + \frac{\tau^3}{6}u^{(4)}_j\Big) + O(\tau^4) = O(\tau^3).$$

Таким образом, локальная погрешность равна $r(t_j, \tau) = O(\tau^3)$, найдем глобальную погрешность нашего метода

$$\psi(t_j, \tau) = \frac{O(\tau^3)}{\tau} = O(\tau^2).$$

Отсюда следует, что рассматриваемый метод является методом второго порядка точности.

Задача 3

Для нахождения интервала устойчивости для метода Адамса четвертого порядка

$$y_{j+1} = y_j + h(\frac{1901}{720}f_j - \frac{1387}{360}f_{j-1} + \frac{109}{30}f_{j-2} - \frac{637}{360}f_{j-3} + \frac{251}{720}f_{j-4})$$

применим модельное уравнение вида

$$u'(x) = \lambda u(x), \quad \lambda \in \mathbb{C},\ \operatorname{Re}\lambda < 0, \quad (1)$$

для которого известно, что задача Коши является устойчивой. Учтем обозначения

$$u(x_j) = y_j,\ u'(x_j) = f(x_j, u(x_j)) = f_j.$$

Тогда модельное уравнения (1) можно переписать в виде

$$f_j = \lambda y_j.$$

Тогда наш метод можно переписать в виде

$$y_{j+1} = y_j + h(\frac{1901}{720}\lambda y_j - \frac{1387}{360}\lambda y_{j-1} + \frac{109}{30}\lambda y_{j-2} - \frac{637}{360}\lambda y_{j-3} + \frac{251}{720}\lambda y_{j-4}).$$

Перенесём все слагаемые в левую сторону

$$y_{j+1} - y_j - h(\frac{1901}{720}\lambda y_j - \frac{1387}{360}\lambda y_{j-1} + \frac{109}{30}\lambda y_{j-2} - \frac{637}{360}\lambda y_{j-3} + \frac{251}{720}\lambda y_{j-4}) = 0.$$

Воспользуемся заменой $z = \lambda h$, тогда получим

$$y_{j+1} - y_j - \frac{1901}{720}zy_j + \frac{1387}{360}zy_{j-1} - \frac{109}{30}zy_{j-2} + \frac{637}{360}zy_{j-3} - \frac{251}{720}zy_{j-4} = 0.$$

Вынесем общие множители за скобки

$$y_{j+1} - y_j(1 + \frac{1901}{720}z) + \frac{1387}{360}zy_{j-1} - \frac{109}{30}zy_{j-2} + \frac{637}{360}zy_{j-3} - \frac{251}{720}zy_{j-4} = 0.$$

Запишем для этого уравнения характеристического уравнение

$$q^5 - q^4(1 + \frac{1901}{720}z) + \frac{1387}{360}zq^3 - \frac{109}{30}zq^2 + \frac{637}{360}zq - \frac{251}{720}z = 0.$$

Попробуем выразить отсюда z.

$$q^5 - q^4 + z(-\frac{1901}{720}q^4 + \frac{1387}{360}q^3 - \frac{109}{30}q^2 + \frac{637}{360}q - \frac{251}{720}) = 0;$$

$$z(-\frac{1901}{720}q^4 + \frac{1387}{360}q^3 - \frac{109}{30}q^2 + \frac{637}{360}q - \frac{251}{720}) = q^4 - q^5;$$

$$z = \frac{q^4 - q^5}{(-\frac{1901}{720}q^4 + \frac{1387}{360}q^3 - \frac{109}{30}q^2 + \frac{637}{360}q - \frac{251}{720})}.$$

Введем обозначение $q = e^{i\varphi}$ и тогда получим

$$z = z(\varphi) = \frac{e^{4i\varphi} - e^{5i\varphi}}{(-\frac{1901}{720}e^{4i\varphi} + \frac{1387}{360}e^{3i\varphi} - \frac{109}{30}e^{2i\varphi} + \frac{637}{360}e^{i\varphi} - \frac{251}{720})}, \varphi \in [0, 2\pi].$$

Тогда

$$z(0) = 0,\ z(\pi) = \frac{\cos 4\pi - \cos 5\pi}{(-\frac{1901}{720}\cos 4\pi + \frac{1387}{360}\cos 3\pi - \frac{109}{30}\cos 2\pi + \frac{637}{360}\cos\pi - \frac{251}{720})} = -\frac{91}{551}.$$

Таким образом, мы получили следующий интервал устойчивости

$$\Big[-\frac{91}{551}; 0\Big].$$

Задача 4

Перепишем нашу задачу Коши

$$\begin{cases} u' = \frac{1}{5}\begin{pmatrix} -104 & -198 \\ -198 & -401 \end{pmatrix} u, t \in [0; 5], \\ u(0) = \begin{pmatrix} 5 \\ 10 \end{pmatrix}. \end{cases}$$

Зададим данную дифференциальную модель программно

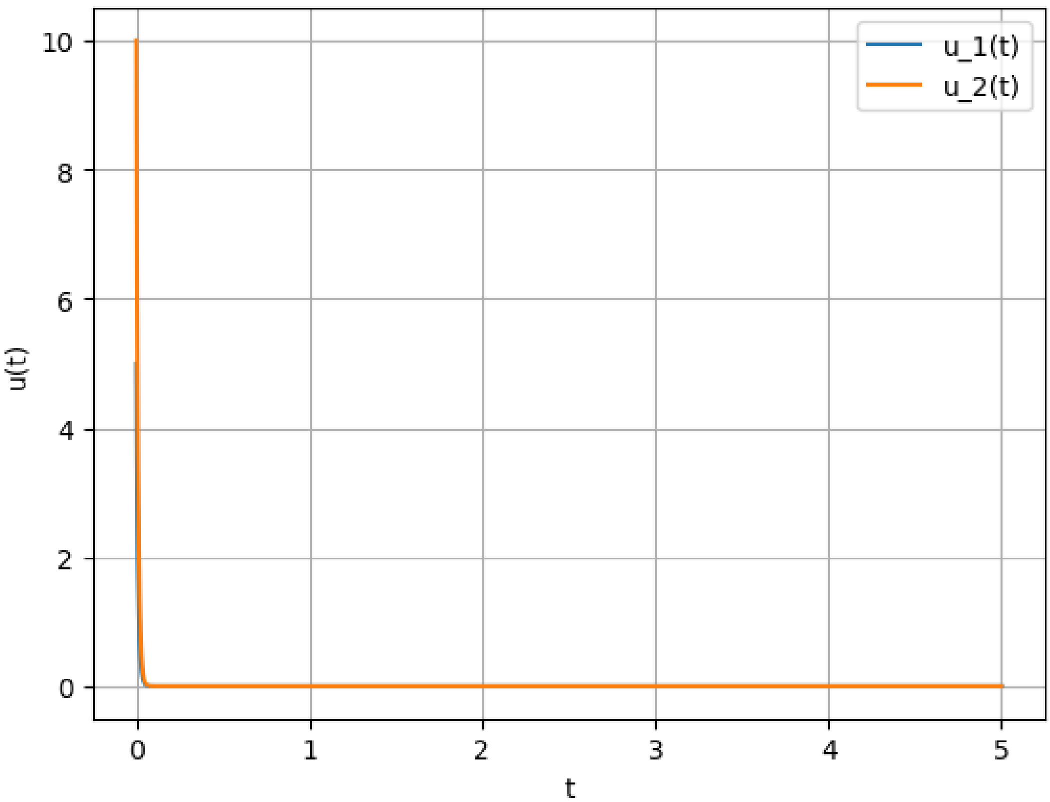
[16]:
```
import numpy as np
import matplotlib.pyplot as plt

def model(u, t):
    u_deriv = 1 / 5 * np.dot(np.array([[-104, -198], [-198, -401]]), u)

    return u_deriv

u_0 = [5, 10]

t_start = 0
t_end = 5
N = 1000

t = np.linspace(t_start, t_end, N)
```

Аналитическое решение найдем с использованием инструментов WolframMathematica

$$u_1(t) = 5e^{-100t},\ u_2(t) = 10e^{-100t}.$$

Зададим и изобразим ее программно, чтобы в дальнейшем сравнивать полученные решения.

[18]:
```
def solution(nodes):
    t = np.array(nodes)
    return np.array([5 * np.e**(-100 * t), 10 * np.e**(-100 * t)]).T

plt.plot(t, solution(t)[:, 0], label='u_1(t)')
plt.plot(t, solution(t)[:, 1], label='u_2(t)')
plt.grid(True)
plt.xlabel('t')
plt.ylabel('u(t)')
plt.legend()
plt.show()
```



Метод последовательного повышения порядка точности четвертого порядка

Построение метода Первая проблема, с которой мы сталкиваемся, необходимость построения такого метода. Поскольку нам нужен метод четвертого порядка, то выбираем $k = 4$ и получаем систему уравнения для коэффициентов

$$\sum_{i=0}^{q} A_i = 1, \quad \sum_{i=0}^{q} A_i\alpha_i = \frac{1}{2}, \quad \sum_{i=0}^{q} A_i\alpha_i^2 = \frac{1}{3}, \quad \sum_{i=0}^{q} A_i\alpha_i^3 = \frac{1}{4}.$$

Выберем $q = 2$, чтобы получить систему из четырех уравнений с четырьмя неизвестными

$$\begin{cases} A_0 + A_1 = 1, \\ A_0\alpha_0 + A_1\alpha_1 = \frac{1}{2}, \\ A_0\alpha_0^2 + A_1\alpha_1^2 = \frac{1}{3}, \\ A_0\alpha_0^3 + A_1\alpha_1^3 = \frac{1}{4}. \end{cases}$$

Для решения данной системы домножим поочередно первое, второе, третье уравнения на $-\alpha_0$ и соответственно сложим полученное со вторым, третьим и четвертым

$$\begin{cases} A_1\alpha_1 - A_1\alpha_0 = \dfrac{1}{2} - \alpha_0, \\ A_1\alpha_1^2 - A_1\alpha_0\alpha_1 = \dfrac{1}{3} - \frac{1}{2}\alpha_0, \\ A_1\alpha_1^3 - A_1\alpha_0\alpha_1^2 = \dfrac{1}{4} - \frac{1}{3}\alpha_0. \end{cases}$$

Получили систему из трех уравнений, с которой проделаем аналогичную операцию, только домножать будем на $-\alpha_1$, тогда получим следующую систему

$$\begin{cases} \alpha_0\alpha_1 - \frac{1}{2}\alpha_0 - \frac{1}{2}\alpha_1 + \frac{1}{3} = 0, \\ \frac{1}{2}\alpha_0\alpha_1 - \frac{1}{3}\alpha_0 - \frac{1}{3}\alpha_1 + \frac{1}{4} = 0. \end{cases}$$

Получили систему из двух уравнений с двумя неизвестными, для решения которой домножим первое уравнение на $-\frac{1}{2}$ и прибавим его ко второму

$$(\alpha_0 + \alpha_1)(\frac{1}{4} - \frac{1}{3}) - \frac{1}{6} + \frac{1}{4} = 0 \Rightarrow -(\alpha_0 + \alpha_1) + \frac{1}{12} = 0 \Rightarrow \alpha_0 + \alpha_1 = \frac{1}{12}.$$

Выразим отсюда $\alpha_0 = \frac{1}{12} - \alpha_1$ и подставим в первое уравнение

$$\alpha_1^2 + \frac{\alpha_1}{12} - \frac{7}{24} = 0.$$

Получили квадратное уравнение, корнями которого являются

$$\alpha_1 = \frac{1}{2}, \quad \alpha_1 = -\frac{7}{12}.$$

Отсюда найдем пару α_0

$$\alpha_0 = -\frac{5}{12}, \quad \alpha_0 = \frac{2}{3}.$$

Выберем любую пару α_0, α_1

$$\alpha_0 = -\frac{5}{12}, \quad \alpha_1 = \frac{1}{2}.$$

Вернемся к нашей исходной системе и подставим эти значения в первые два уравнения

$$\begin{cases} A_0 + A_1 = 1, \\ -A_0\frac{5}{12} + \frac{1}{2}A_1 = \frac{1}{2}. \end{cases} \Rightarrow \begin{cases} A_0 = 1 - A_1, \\ -(1 - A_1)\frac{5}{12} + \frac{1}{2}A_1 = \frac{1}{2}. \end{cases} \Rightarrow A_1 = 1, \quad A_0 = 0.$$

Таким образом, имеем

$$A_0 = 0, \quad A_1 = 1, \quad \alpha_0 = -\frac{5}{12}, \quad \alpha_1 = \frac{1}{2}.$$

Подставим данные коэффициенты в формулу метода

$$y_{j+1} = y_j + \tau \sum_{i=0}^{q} A_i f(t_j + \alpha_i\tau, y(t_j + \alpha_i\tau)),$$

получим неявный метод четвертого порядка

$$\overset{[5]}{y}_{j+1} = \overset{[5]}{y}_j + h\overset{[5]}{f}_{j+\frac{1}{2}}.$$

Чтобы сделать метод явным, необходимо дополнить его методами более низкого порядка, например методом "3/4"

$$\begin{cases} \overset{[4]}{y}_{j+1} = \overset{[4]}{y}_j + \dfrac{h}{4}\left(\overset{[4]}{f}_j + 3\overset{[3]}{f}_{j+\frac{2}{3}}\right), \\ \overset{[3]}{y}_{j+\frac{2}{3}} = \overset{[4]}{y}_j + \dfrac{2}{3}h\overset{[2]}{f}_{j+\frac{1}{3}}, \\ \overset{[2]}{y}_{j+\frac{1}{3}} = \overset{[4]}{y}_j + \dfrac{1}{3}h\overset{[4]}{f}_j. \end{cases}$$

Для понижения порядка $\overset{[5]}{f}_{j+\frac{1}{2}}$ заменим в методе "3/4" $h = \frac{1}{2}h$

$$\begin{cases} \overset{[4]}{y}_{j+\frac{1}{2}} = \overset{[5]}{y}_j + \dfrac{h}{8}\left(\overset{[5]}{f}_j + 3\overset{[3]}{f}_{j+\frac{1}{3}}\right), \\ \overset{[3]}{y}_{j+\frac{1}{3}} = \overset{[5]}{y}_j + \dfrac{1}{3}h\overset{[2]}{f}_{j+\frac{1}{6}}, \\ \overset{[2]}{y}_{j+\frac{1}{6}} = \overset{[5]}{y}_j + \dfrac{1}{6}h\overset{[4]}{f}_j. \end{cases}$$

Соберем построенные методы и получим явный метод четвертого порядка

$$\begin{cases} \overset{[5]}{y}_{j+1} = \overset{[5]}{y}_j + h\overset{[4]}{f}_{j+\frac{1}{2}}, \\ \overset{[4]}{y}_{j+\frac{1}{2}} = \overset{[5]}{y}_j + \dfrac{h}{8}\left(\overset{[5]}{f}_j + 3\overset{[3]}{f}_{j+\frac{1}{3}}\right), \\ \overset{[3]}{y}_{j+\frac{1}{3}} = \overset{[5]}{y}_j + \dfrac{1}{3}h\overset{[2]}{f}_{j+\frac{1}{6}}, \\ \overset{[2]}{y}_{j+\frac{1}{6}} = \overset{[5]}{y}_j + \dfrac{1}{6}h\overset{[5]}{f}_j. \end{cases}$$

Программную реализацию я не осилил.

Явный метод Рунге-Кутта третьего порядка

Запишем более экономичный с точки зрения количества операций метод Рунге-Кутта третьего порядка

$$\begin{cases} y_{j+1} = y_j + \dfrac{1}{4}(\varphi_0 + 3\varphi_2), \\ \varphi_0 = hf(x_j, y_j), \\ \varphi_1 = hf\left(x_j + \dfrac{1}{3}h, y_j + \dfrac{1}{3}\varphi_0\right), \\ \varphi_2 = hf\left(x_j + \dfrac{2}{3}h, y_j + \dfrac{2}{3}\varphi_1\right). \end{cases}$$

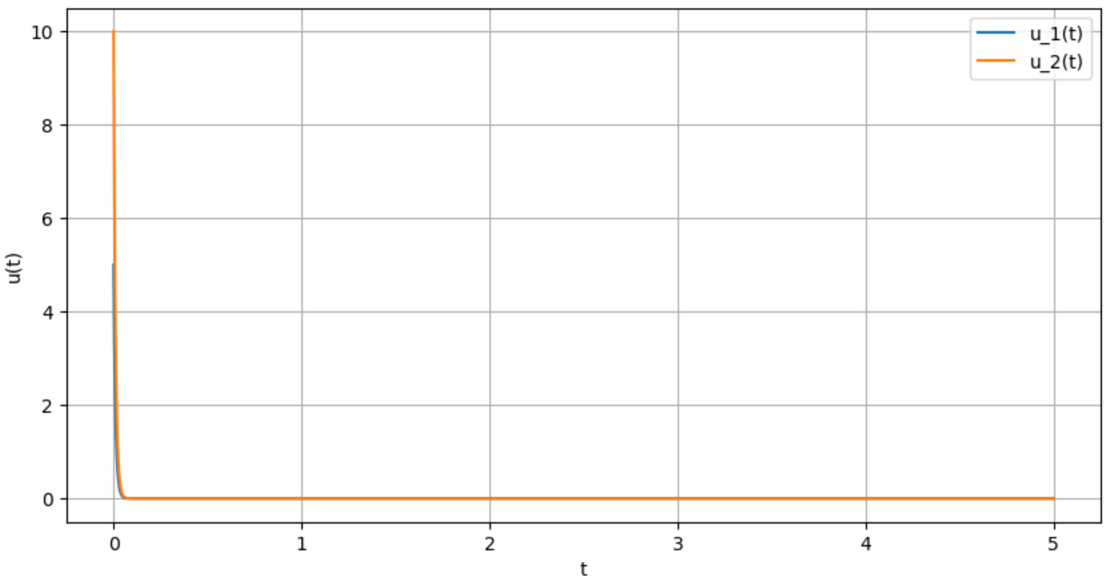
[24]:
```
def runge_kutta(f, y0, t_start, t_end, epsilon):
    y = [np.array(y0)]
    t = t_start
    nodes = [t_start]
    j = 0
    h1 = epsilon**(1/3)
```

```python
    k = 1.005

    while True:

        while True:
            phi_0_h1 = h1 * f(y[j], t)
            phi_1_h1 = h1 * f(y[j] + 1/3 * phi_0_h1, t + 1/3 * h1)
            phi_2_h1 = h1 * f(y[j] + 2/3 * phi_1_h1, t + 2/3 * h1)

            y_h1 = y[j] + 1/4 * (phi_0_h1 + 3 * phi_2_h1)

            h2 = h1 / 2
            phi_0_h2 = h2 * f(y[j], t)
            phi_1_h2 = h2 * f(y[j] + 1/3 * phi_0_h2, t + 1/3 * h2)
            phi_2_h2 = h2 * f(y[j] + 2/3 * phi_1_h2, t + 2/3 * h2)

            y_h2 = y[j] + 1/4 * (phi_0_h2 + 3 * phi_2_h2)

            if np.max(y_h2 - y_h1) / (1 - (h2 / h1)**3) < epsilon:
                y.append(y_h1)
                break
            else:
                h1 /= 2

        j += 1
        if t + h1 > t_end:
            nodes.append(t_end)
            break
        else:
            t += h1
            nodes.append(t)
            h1 *= k

    return np.stack(y, axis=0), nodes
```

[25]:
```python
sol, t_0 = runge_kutta(model, u_0, t_start, t_end, 1e-4)
```

Построим полученное решение

[27]:
```python
plt.figure(figsize=(10, 5))
plt.plot(t_0, sol[:, 0], label='u_1(t)')
plt.plot(t_0, sol[:, 1], label='u_2(t)')
plt.grid(True)
plt.xlabel('t')
plt.ylabel('u(t)')
plt.legend()
plt.show()
```



Как видим, полученное решение визуально совпадает с аналитическим решением.

Интерполяционный метод Адамса третьего порядка

Интерполяционный метод Адамса третьего порядка имеет следующий вид

$$y_{j+1} = y_j + \frac{h}{12}(5f_{j+1} + 8f_j - f_{j-1}).$$

Поскольку для реализации данного метода нам потребуются значения y_0, y_1, то обратимся к предыдущему пункту, методу Рунге-Кутта 3-его порядка.

[30]:
```
def adams(f, y0, t_0, t, epsilon):
    y = y0.tolist()
    t_new = t_0
    j = 2
    h_current = epsilon**(1/2)
    k = 1.005

    while True:
        while True:
            y_high_k = y[j]
            y_high = y[j]

            while True:
                y_high = y[j] + h_current / 12 * (5 * f(y_high_k,␣
↪t_new[j]+h_current) + 8 * f(y[j], t_new[j]) - f(y[j-1], t_new[j-1]))

                if np.max(np.absolute(y_high - y_high_k)) < epsilon * 1e-3:
                    break
```

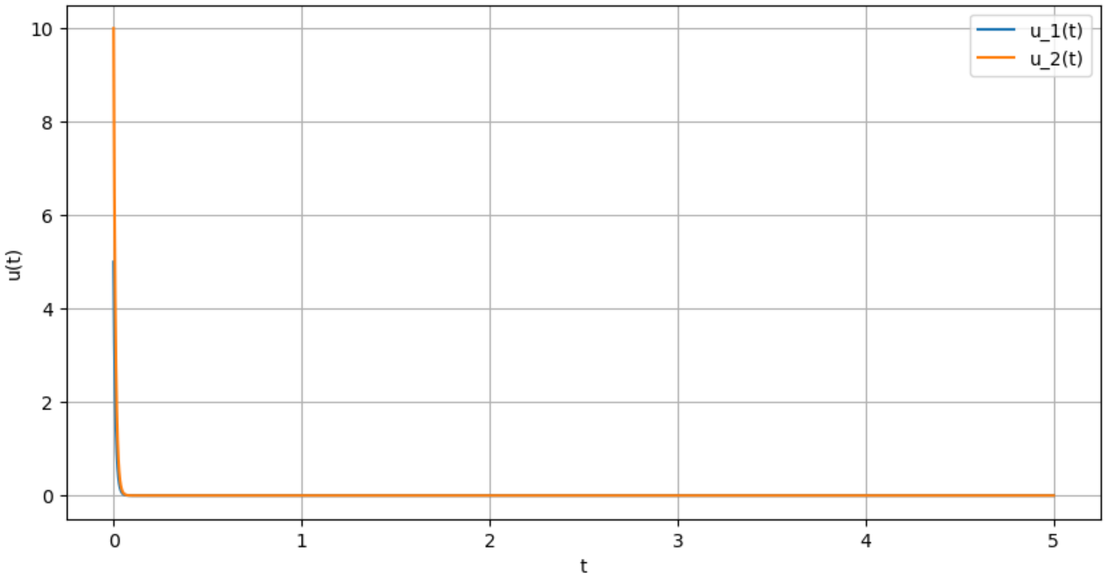
```
                y_high_k = y_high

            h_half = 0.5 * h_current

            y_low_k = y[j]
            y_low = y[j]

            while True:
                y_low = y[j] + h_half / 12 * (5 * f(y_low_k, t_new[j]+h_half) +␣
↪8 * f(y[j], t_new[j]) - f(y[j-1], t_new[j-1]))

                if np.max(np.absolute(y_low - y_low_k)) < epsilon * 1e-3:
                    break

                y_low_k = y_low

            error = np.max(y_low - y_high) / (1 - (h_half / h_current)**3)

            if error <= epsilon:
                y.append(y_high)
                break
            else:
                h_current *= 0.5

        j += 1

        if t_new[j-1] + h_current > t[-1]:
            t_new.append(t[-1])
            break

        else:
            t_current = t_new[j-1] + h_current
            t_new.append(t_current)
            h_current *= k

    return np.stack(y, axis=0), t_new
```

[31]:
```
solution, nodes = adams(model, sol, t_0, t, 1e-4)
```

Построим полученное решение

[67]:
```
plt.figure(figsize=(10, 5))
plt.plot(nodes, solution[:, 0], label='u_1(t)')
plt.plot(nodes, solution[:, 1], label='u_2(t)')
plt.grid(True)
plt.xlabel('t')
```

```
plt.ylabel('u(t)')
plt.legend()
plt.show()
```



Опять-таки, визуально график решения, полученного методом Адамса, совпадает с графиком аналитического решения.