Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**ОТЧЁТ**

к лабораторной работе

на тему

Метод Адамса

Выполнил: студент группы 153503

Киселёва Елизавета Андреевна

Проверил: Анисимов Владимир Яковлевич

Минск 2022

**Оглавление**

Цели выполнения задания 3

Краткие теоретические сведения 4

Задание 6

Алгоритм задания 7

Программная реализация 8

Результат выполнения программы 11

Тестовые примеры 12

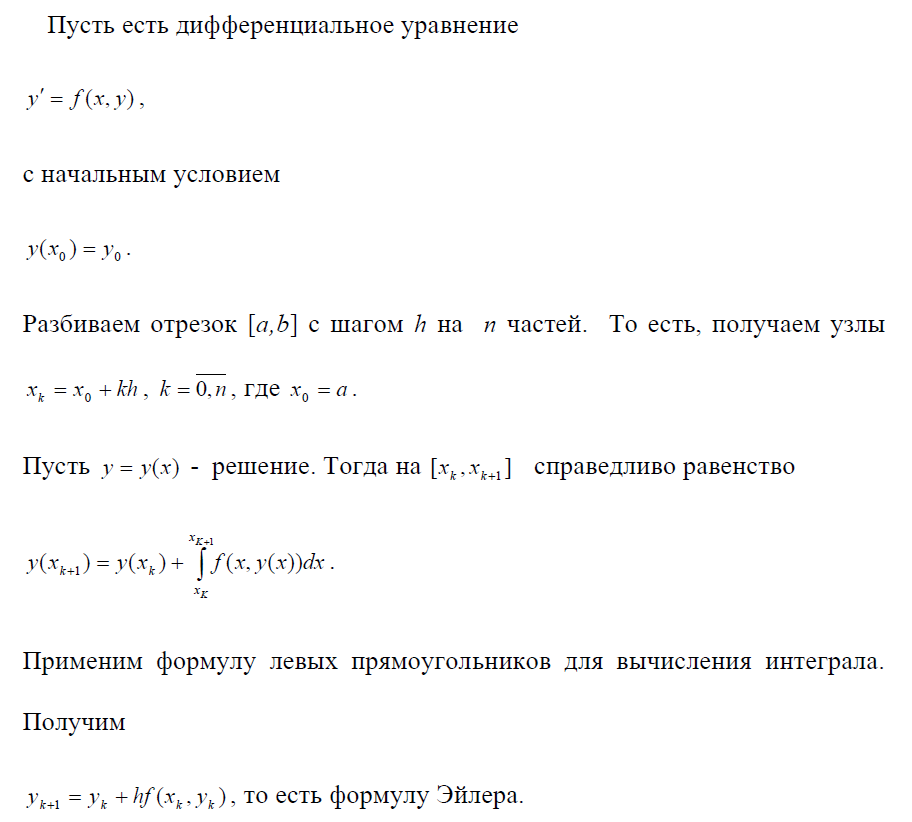
Выводы 14

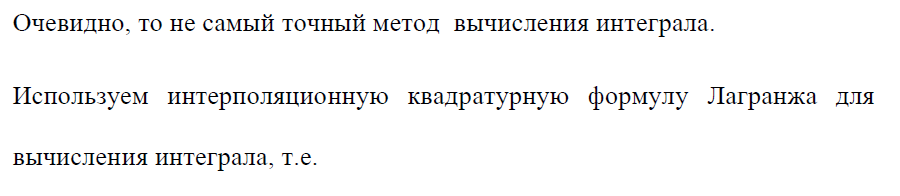
Вариант 9

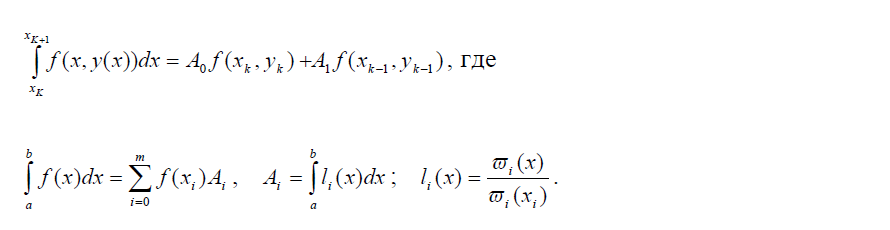
# **Цели выполнения задания**

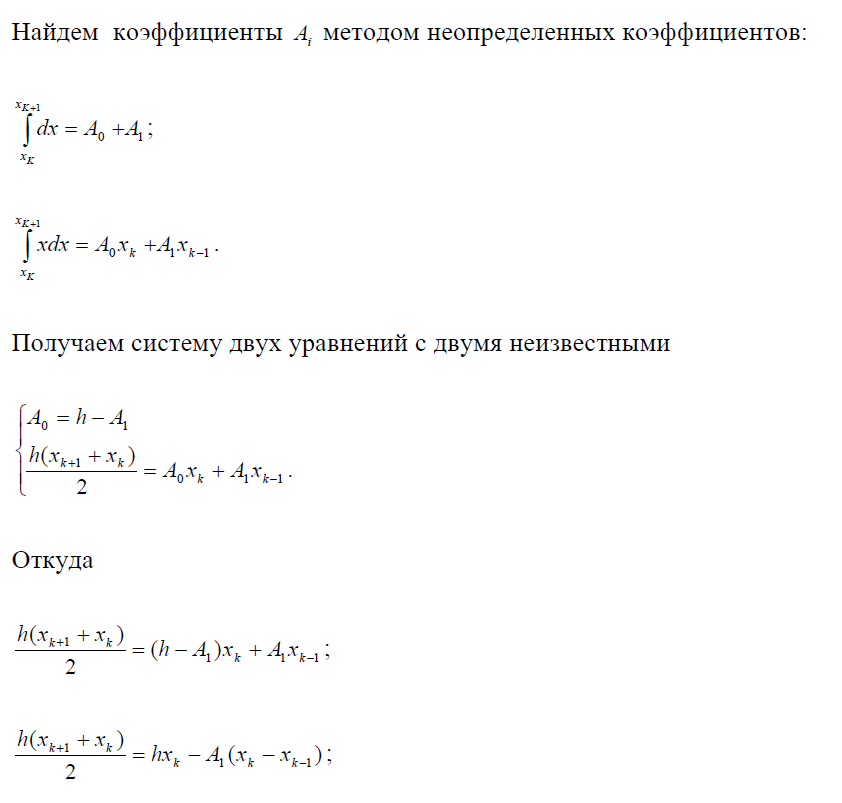
* Изучить решение задачи Коши для обыкновенных дифференциальных уравнений методом Адамса;
* С помощью метода Адамса найти решения уравнения;
* Сделать выводы на основе выполненного задания.

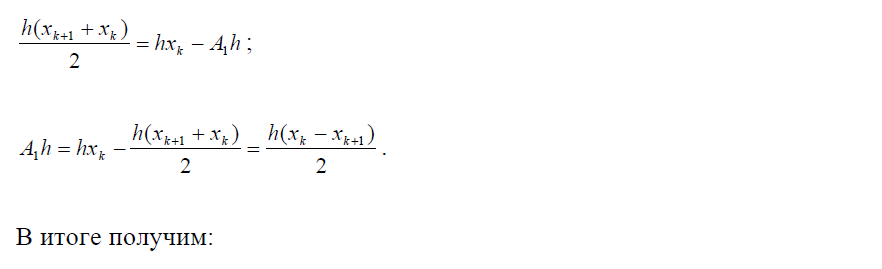
# **Краткие теоретические сведения**

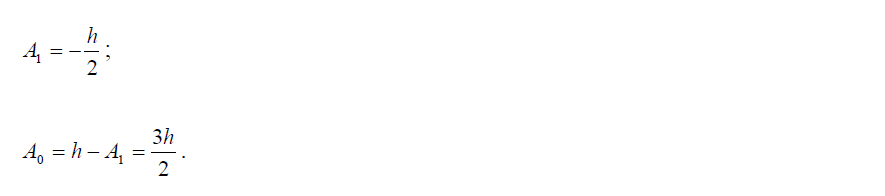


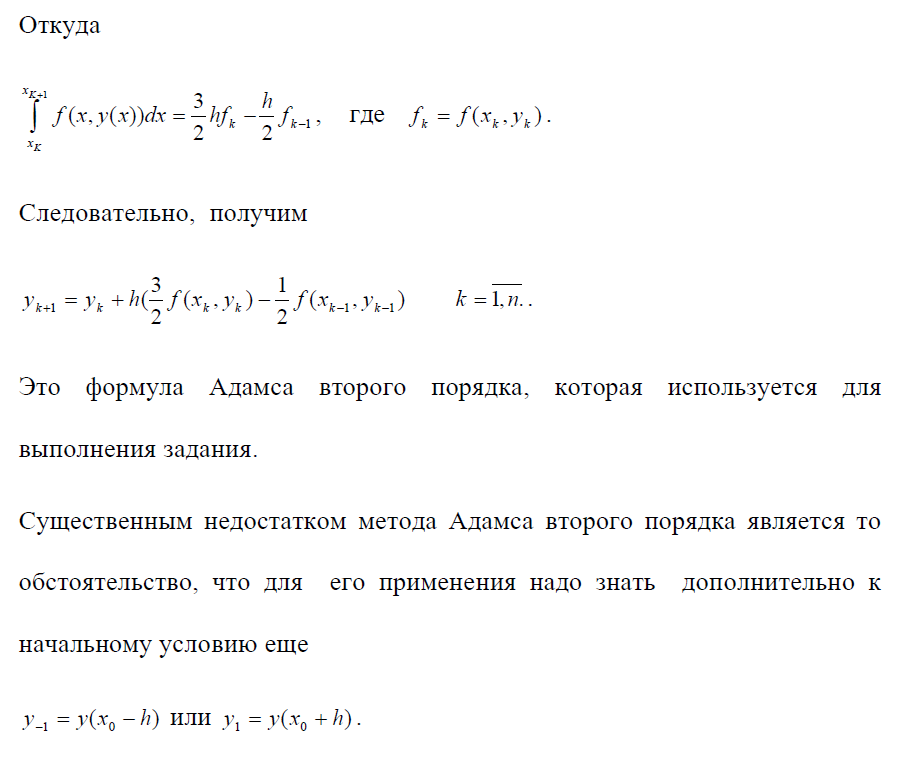


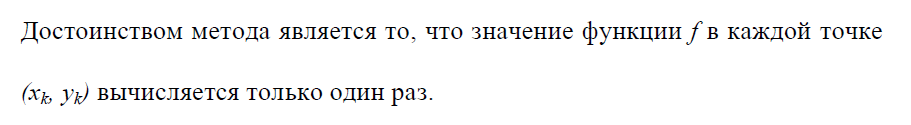






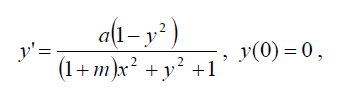






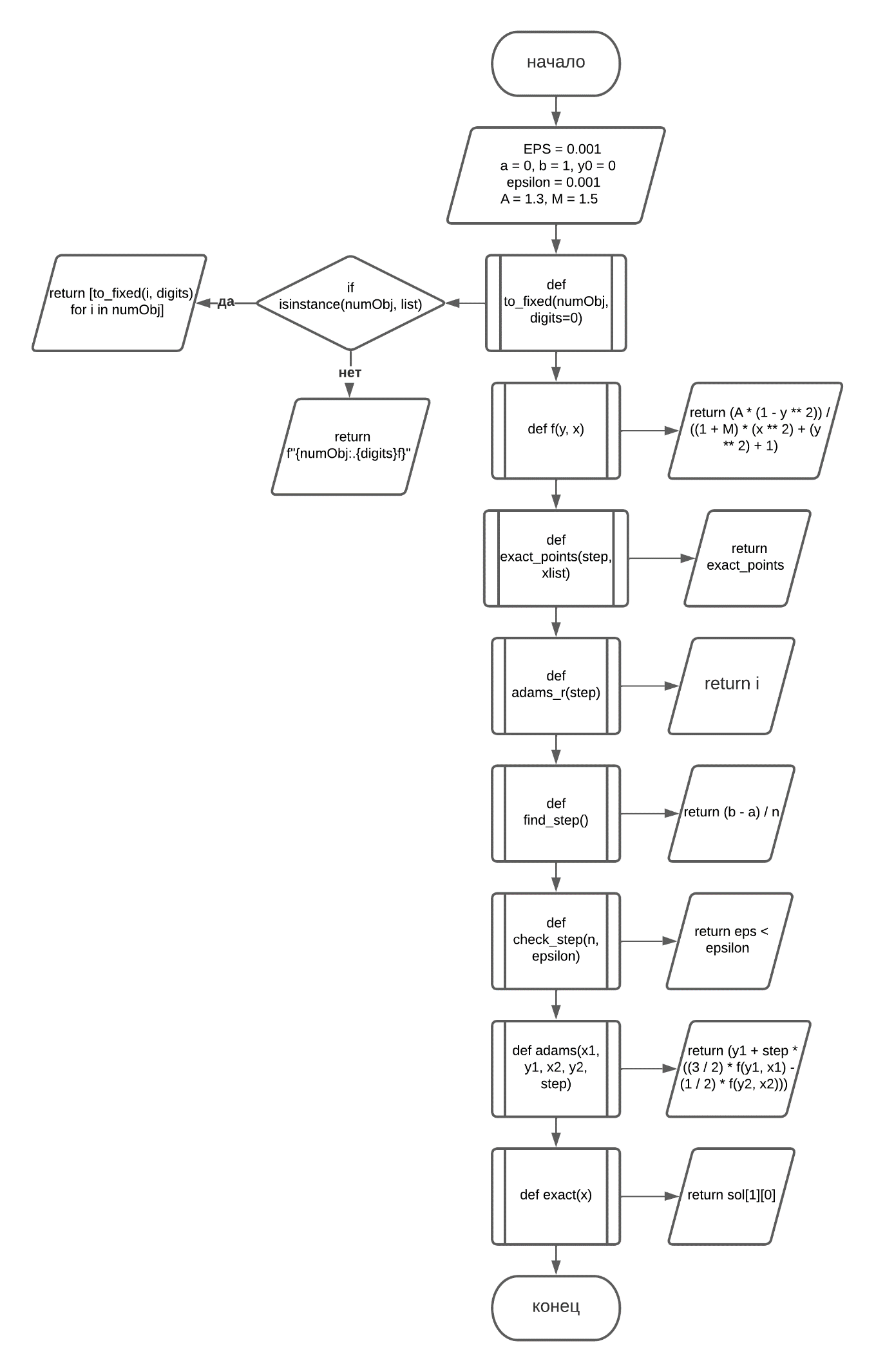
**Задание**

С помощью метода Адамса найти решения следующих уравнений на отрезке [0; 1].



где а = 1.1, m = 2.0.

Сравнить полученные результаты.

**Алгоритм задания**

**Программная реализация**

from matplotlib import pylab  
from scipy.integrate import odeint as od  
import numpy as np  
  
EPS = 0.001  
  
a = 0  
b = 1  
y0 = 0  
epsilon = 0.001  
  
A = 1.1  
M = 2.0  
  
  
def to\_fixed(numObj, digits=0):  
 if isinstance(numObj, list):  
 return [to\_fixed(i, digits) for i in numObj]  
 return f"{numObj:.{digits}f}"  
  
  
def f(y, x):  
 return (A \* (1 - y \*\* 2)) / ((1 + M) \* (x \*\* 2) + (y \*\* 2) + 1)  
  
  
def exact\_points(step, xlist):  
 exact\_points = []  
 for x in xlist:  
 r2 = exact(x)  
 exact\_points.append(r2)  
 return exact\_points  
  
  
def adams\_r(step):  
 y1 = y0  
 xlist = np.arange(a, b + step, step)  
 exact\_point = exact\_points(step, xlist)  
 adams\_points = []  
 adams\_points.append(y1)  
 adams\_points.append(exact\_point[1])  
 for i in range(1, len(xlist) - 1, 1):  
 y1 = adams(xlist[i], adams\_points[i], xlist[i - 1], adams\_points[i - 1], step)  
 adams\_points.append(y1)  
 sum = 0  
 for i in adams\_points:  
 sum += i  
 return i  
  
  
def find\_step():  
 h0 = epsilon \*\* (1 / 4)  
 n = int((b - a) // h0)  
 if n % 2 != 0:  
 n += 1  
 while check\_step(n, epsilon):  
 n = n // 4 \* 2  
 while not check\_step(n, epsilon):  
 n \*= 2  
 return (b - a) / n  
  
  
def check\_step(n, epsilon):  
 h = (b - a) / n  
 y2 = adams\_r(h)  
 y2e = adams\_r(h \* 2)  
 eps = (1 / 15) \* abs(y2 - y2e)  
 return eps < epsilon  
  
  
def adams(x1, y1, x2, y2, step):  
 return (y1 + step \* ((3 / 2) \* f(y1, x1) - (1 / 2) \* f(y2, x2)))  
  
  
def exact(x):  
 sol = od(f, y0, [a, x])  
 return sol[1][0]  
  
  
def main():  
 print("Исходные данные:")  
 print(f"y = (0.5\*(1-(y\*\*2)))/((1+1.0)\*(x\*\*2)+(y\*\*2)+1)")  
 print(f"y(0) = {y0}")  
 print(f"\nИнтервал: [{a}, {b}]")  
 print(f"Погрешность: {EPS}")  
 print()  
 step = find\_step()  
 print("Шаг итерирования: ", round(step, 5))  
 xlist = np.arange(a, b + step, step)  
 adams\_points = []  
 exact\_point = exact\_points(step, xlist)  
 y1 = y0  
 adams\_points.append(y1)  
 adams\_points.append(exact\_point[1])  
 for i in range(1, len(xlist) - 1, 1):  
 y1 = adams(xlist[i], adams\_points[i], xlist[i - 1], adams\_points[i - 1], step)  
 adams\_points.append(y1)  
  
 print(f"Значения функции в точках методом Адамса:\n {to\_fixed(adams\_points, 3)}")  
 print(f"Точные значения функции:\n {to\_fixed(exact\_point, 3)}")  
 pylab.cla()  
 pylab.plot(xlist, exact\_point, label="точное решение", color=(0, 1, 0))  
 pylab.plot(xlist, adams\_points, label="кривая методом Адамса", color=(1, 0, 0))  
 pylab.grid(True)  
 pylab.legend()  
 pylab.savefig("lab10.png")  
 pylab.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

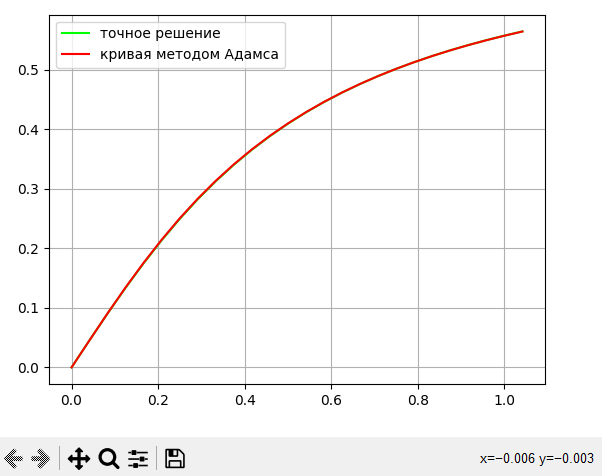
**Результат выполнения программы**

Шаг итерирования: 0.04167

Значения функции в точках методом Адамса:

['0.000', '0.046', '0.091', '0.134', '0.176', '0.214', '0.250', '0.283', '0.313', '0.341', '0.366', '0.389', '0.410', '0.428', '0.446', '0.461', '0.476', '0.489', '0.501', '0.512', '0.523', '0.532', '0.541', '0.549', '0.557', '0.564']

Точные значения функции:

['0.000', '0.046', '0.091', '0.134', '0.175', '0.213', '0.249', '0.282', '0.313', '0.340', '0.365', '0.388', '0.409', '0.428', '0.445', '0.461', '0.476', '0.489', '0.501', '0.512', '0.523', '0.532', '0.541', '0.549', '0.557', '0.564']****

**Тестовые примеры**

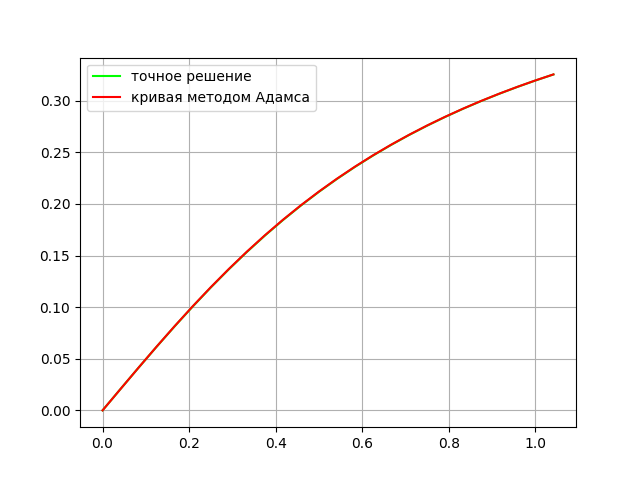
**a = 0.5, m = 1.0**

Значения функции в точках методом Адамса:

['0.000', '0.021', '0.042', '0.062', '0.082', '0.101', '0.119', '0.137', '0.154', '0.170', '0.185', '0.199', '0.212', '0.225', '0.236', '0.247', '0.257', '0.267', '0.276', '0.285', '0.292', '0.300', '0.307', '0.313', '0.320', '0.326']

Точные значения функции:

['0.000', '0.021', '0.041', '0.062', '0.081', '0.101', '0.119', '0.137', '0.153', '0.169', '0.184', '0.198', '0.212', '0.224', '0.236', '0.247', '0.257', '0.267', '0.276', '0.284', '0.292', '0.300', '0.307', '0.313', '0.320', '0.325']

****

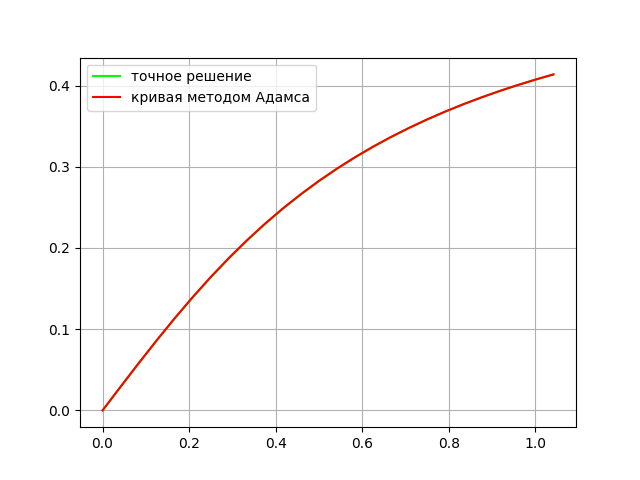
**a = 0.7, m = 1.5**

Значения функции в точках методом Адамса:

['0.000', '0.029', '0.058', '0.086', '0.113', '0.140', '0.164', '0.188', '0.209', '0.230', '0.249', '0.266', '0.283', '0.298', '0.312', '0.325', '0.337', '0.348', '0.358', '0.368', '0.377', '0.385', '0.393', '0.401', '0.407', '0.414']

Точные значения функции:

['0.000', '0.029', '0.058', '0.086', '0.113', '0.139', '0.164', '0.187', '0.209', '0.229', '0.248', '0.266', '0.282', '0.297', '0.311', '0.325', '0.337', '0.348', '0.358', '0.368', '0.377', '0.385', '0.393', '0.400', '0.407', '0.414']



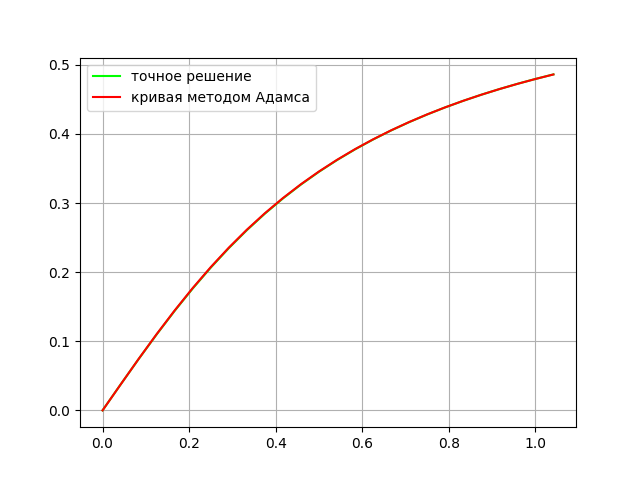
**a = 0.9, m = 2.0:**

Значения функции в точках методом Адамса:

['0.000', '0.037', '0.074', '0.110', '0.145', '0.177', '0.207', '0.235', '0.261', '0.285', '0.307', '0.327', '0.346', '0.363', '0.378', '0.392', '0.405', '0.417', '0.428', '0.438', '0.448', '0.457', '0.465', '0.472', '0.479', '0.486']

Точные значения функции:

['0.000', '0.037', '0.074', '0.110', '0.144', '0.176', '0.207', '0.235', '0.261', '0.285', '0.307', '0.327', '0.345', '0.362', '0.378', '0.392', '0.405', '0.417', '0.428', '0.438', '0.448', '0.457', '0.465', '0.472', '0.480', '0.486']

****

**Выводы**

Таким образом, в ходе выполнения лабораторной работы был освоен метод Адамса для решения задачи Коши для обыкновенных дифференциальных уравнений. Составлена компьютерная программа, на тестовых примерах проверена правильность её работы, с заданной точностью построен график решения дифференциального уравнения заданного варианта, по количеству необходимых для этого отрезков оценена трудоёмкость метода.