МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК**  **Кафедра информатики, математического**  **и компьютерного моделирования** |

**Курсовой проект**

по дисциплине «Численные методы решения дифференциальных уравнений»

на тему «**Методы Рунге-Кутта 3-го, 4-го, 5-го порядков погрешности решения задачи Коши для обыкновенного дифференциального уравнения 1-го порядка**»

Направление подготовки   
02.03.01 «Математика и компьютерные науки»

Выполнил(а) студент(ка) гр. Б8117-02.03.01

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.С.Михайлов

(*подпись*) *(Ф.И.О.)*

Проверил доцент, к.ф.-м.н.

Колобов А.Г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

**г. Владивосток**

**2020**

## Заключение.

[Введение. 3](#_Toc42193570)

[Основная часть. 4](#_Toc42193571)

[1. Постановка задачи. 4](#_Toc42193572)

[2. Описание алгоритмов решения задачи. 4](#_Toc42193573)

[3. Описание тестов. 7](#_Toc42193574)

[4. Вычислительные эксперименты. 7](#_Toc42193575)

[Заключение. 9](#_Toc42193576)

[Литература. 10](#_Toc42193577)

[Код программы. 11](#_Toc42193578)

## Введение.

Объектом исследования являются численные методы решения задач математической физики, а также программное обеспечение, реализующее эти методы.

Цель работы – ознакомиться с численными методами решения обыкновенных дифференциальных уравнений, уравнений в частных производных, интегральных уравнений, решить предложенные типовые задачи, сформулировать выводы по полученным решениям, отметить достоинства и недостатки методов, приобрести практические навыки и компетенции, а также опыт самостоятельной профессиональной деятельности, а именно:

• создать алгоритм решения поставленной задачи и реализовать его, протестировать программы;

• освоить теорию вычислительного эксперимента; современных компьютерных технологий;

• приобрести навыки представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

Работа над курсовым проектом предполагает выполнение следующих задач:

• дальнейшее углубление теоретических знаний обучающихся и их систематизацию;

• получение и развитие прикладных умений и практических навыков по направлению подготовки;

• овладение методикой решения конкретных задач;

• развитие навыков самостоятельной работы;

• развитие навыков обработки полученных результатов, анализа и осмысления их с учетом имеющихся литературных данных;

• приобретение навыков оформления описаний программного продукта;

• повышение общей и профессиональной эрудиции.

## Основная часть.

# 1. Постановка задачи.

Рассмотрим задачу Коши для обыкновенного дифференциального уравнения (ОДУ) первого порядка на некотором отрезке :

Решение которого удовлетворяет начальному условию:

Разобьем отрезок на равных частей точками .

Требуется найти решение задачи Коши для ОДУ первого порядка методами Рунге-Кутта 3-го, 4-го и 5-го порядков точности, а также провести сравнение методов. Для реализации методов будем использовать ЯП Python версии 3.8.3.

# 2. Описание алгоритмов решения задачи.

Идея методов Рунге-Кутта состоит в том, что они используют для вычисления значения значение , а также значения функции при некоторых специальным образом выбираемых значениях и . На их основе могут быть построены разностные схемы разного порядка точности.

2.1. Метод Рунге-Кутта 3-го рода точности.

Решение задачи методом Рунге-Кутта 3-го рода точности имеет следующий вид:

Где:

Суммарная погрешность этого метода есть величина

Программная реализация данного метода на ЯП Python выглядит следующим образом:

def Runge\_Kutta\_3(y0, h): #первый этап - инициализация

y = [0 for i in range(int(1/h)+1)]

x = [h\*i for i in range(int(1/h)+1)]

y[0] = y0

#второй этап – определение коэффициентов

k0 = lambda x, y: h\*func(x, y)

k1 = lambda x, y: h\*func(x + h/2, y + k0(x, y)/2)

k2 = lambda x, y: h\*func(x + h, y + 2\*k1(x, y) - k0(x, y))

Y = lambda x, y: y + (k0(x, y) + 4\*k1(x, y) + 2\*k2(x, y))/6

for i in range(1, int(1/h)+1):

y[i] = Y(x[i-1], y[i-1]) #Третий этап – находим решение

return x, y

2.2 Метод Рунге-Кутта 4-го рода точности.

Решение задачи методом Рунге-Кутта 4-го рода точности имеет следующий вид:

Где:

Суммарная погрешность этого метода есть величина

Программная реализация данного метода на ЯП Python выглядит следующим образом:

def Runge\_Kutta\_4(y0, h): #первый этап - инициализация

y = [0 for i in range(int(1/h)+1)]

x = [h\*i for i in range(int(1/h)+1)]

y[0] = y0

#второй этап – определение коэффициентов

k0 = lambda x, y: h\*func(x, y)

k1 = lambda x, y: h\*func(x + h/2, y + k0(x, y)/2)

k2 = lambda x, y: h\*func(x + h/2, y + k1(x, y)/2)

k3 = lambda x, y: h\*func(x + h, y + k2(x, y))

Y = lambda x, y: y + (k0(x, y) + 2\*k1(x, y) + 2\*k2(x, y) + k3(x, y))/6

for i in range(1, int(1/h)+1):

y[i] = Y(x[i-1], y[i-1]) #Третий этап – находим решение

return x, y

2.3 Метод Рунге-Кутта 5-го рода точности.

Решение задачи методом Рунге-Кутта 5-го рода точности имеет следующий вид:

Где:



Суммарная погрешность этого метода есть величина .

Программная реализация данного метода на ЯП Python выглядит следующим образом:

def Runge\_Kutta\_5(y0, h): #первый этап - инициализация

y = [0 for i in range(int(1/h)+1)]

x = [h\*i for i in range(int(1/h)+1)]

y[0] = y0

#второй этап – определение коэффициентов

k0 = lambda x, y: h\*func(x, y)

k1 = lambda x, y: h\*func(x + h/3, y + k0(x, y)/3)

k2 = lambda x, y: h\*func(x + h/3, y + k1(x, y)/6 + k0(x, y)/6)

k3 = lambda x, y: h\*func(x + h/2, y + k0(x, y)/8 + 3\*k2(x, y)/8)

k4 = lambda x, y: h\*func(x + h, y + k0(x, y) - 3\*k2(x, y)/2 + 2\*k3(x, y))

Y = lambda x, y: y + (k0(x, y) + k3(x, y) + k4(x, y))/6

for i in range(1, int(1/h)+1):

y[i] = Y(x[i-1], y[i-1]) #Третий этап – находим решение

return x, y

# 3. Описание тестов.

В качестве тестов возьмем следующие ОДУ с задачей Коши и точным решением соответственно:

# 4. Вычислительные эксперименты.

Далее, я буду указывать таблицу решений задачи для точного решения и решения с помощью вышеуказанных методов при на сетке при и шаге . Ответ будем записывать до 5-го знака.

Пример 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Точное решение** | **Р-Г 3п.** | **Р-Г 4п.** | **Р-Г 5п.** |
| 1 | 1 | 1 | 1 |
| 1.10551 | 1.12418 | 1.10551 | 1.10552 |
| 1.22421 | 1.26636 | 1.22421 | 1.22423 |
| 1.35958 | 1.43123 | 1.35958 | 1.35962 |
| 1.51547 | 1.62407 | 1.51548 | 1.51555 |
| 1.69616 | 1.85082 | 1.69617 | 1.69628 |
| 1.90636 | 2.11814 | 1.90636 | 1.90653 |
| 2.15126 | 2.43352 | 2.15126 | 2.15151 |
| 2.43662 | 2.80538 | 2.43662 | 2.43697 |
| 2.76881 | 3.24318 | 2.76881 | 2.76928 |
| 3.15485 | 3.75754 | 3.15485 | 3.15547 |

Пример 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Точное решение** | **Р-Г 3п.** | **Р-Г 4п.** | **Р-Г 5п.** |
| 1 | 1 | 1 | 1 |
| 0.99548 | 0.99414 | 0.99547 | 0.99547 |
| 0.98361 | 0.98004 | 0.9836 | 0.9836 |
| 0.96654 | 0.9603 | 0.96654 | 0.96654 |
| 0.94595 | 0.93688 | 0.94594 | 0.94594 |
| 0.92308 | 0.91125 | 0.92307 | 0.92307 |
| 0.89888 | 0.88447 | 0.89887 | 0.89887 |
| 0.87404 | 0.85729 | 0.87403 | 0.87403 |
| 0.84906 | 0.83024 | 0.84905 | 0.84905 |
| 0.8243 | 0.80368 | 0.82429 | 0.82429 |
| 0.8 | 0.77785 | 0.8 | 0.79999 |

**Пример 3.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Точное решение** | **Р-Г 3п.** | **Р-Г 4п.** | **Р-Г 5п.** |
| 0 | 0 | 0 | 0 |
| -0.2214 | -0.26267 | -0.2214 | -0.2216 |
| -0.49182 | -0.59433 | -0.49182 | -0.49231 |
| -0.82212 | -1.01311 | -0.82212 | -0.82301 |
| -1.22554 | -1.54189 | -1.22553 | -1.227 |
| -1.71828 | -2.20956 | -1.71826 | -1.72051 |
| -2.32012 | -3.0526 | -2.32008 | -2.32339 |
| -3.0552 | -4.11708 | -3.05515 | -3.05987 |
| -3.95303 | -5.46117 | -3.95296 | -3.95956 |
| -5.04965 | -7.1583 | -5.04955 | -5.05862 |
| -6.38906 | -9.30121 | -6.38892 | -6.40124 |

Из вычислительных экспериментов следует очевидный вывод: метод Рунге-Кутта 4-го порядка точности обладает более высокой точностью, чем методы Рунге-Кутта 3-его и 5-го порядка точности.

## Заключение.

В результате работы над курсовым проектом приобрел практические навыки владения:

* современными численными методами решения задач математической физики;
* основами алгоритмизации для численного решения задач математической физики на одном из языков программирования;
* инструментальными средствами, поддерживающими разработку программного обеспечения для численного решения задач математической физики;

а также навыками представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

## Литература.

1. Бахвалов Н.С. Численные методы. – М.: Наука, 1975.- 631 с.
2. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М.: Наука, 2002 г. – 630 с.
3. Вержбицкий В. М. Основы численных методов. 2-e изд., перераб. / В. М. Вержбицкий. – М.: Высшая школа, 2005. – 267 с.

## Код программы.

from math import exp

def Runge\_Kutta\_3(y0, h):

y = [0 for i in range(int(1/h)+1)]

x = [h\*i for i in range(int(1/h)+1)]

y[0] = y0

k0 = lambda x, y: h\*func(x, y)

k1 = lambda x, y: h\*func(x + h/2, y + k0(x, y)/2)

k2 = lambda x, y: h\*func(x + h, y + 2\*k1(x, y) - k0(x, y)) Y = lambda x, y: y + (k0(x, y) + 4\*k1(x, y) + 2\*k2(x, y))/6 for i in range(1, int(1/h)+1):

y[i] = round(Y(x[i-1], y[i-1]), 5)

return x, y

def Runge\_Kutta\_4(y0, h):

y = [0 for i in range(int(1/h)+1)]

x = [h\*i for i in range(int(1/h)+1)]

y[0] = y0

k0 = lambda x, y: h\*func(x, y)

k1 = lambda x, y: h\*func(x + h/2, y + k0(x, y)/2)

k2 = lambda x, y: h\*func(x + h/2, y + k1(x, y)/2)

k3 = lambda x, y: h\*func(x + h, y + k2(x, y))

Y = lambda x, y: y + (k0(x, y) + 2\*k1(x, y) + 2\*k2(x, y) + k3(x, y))/6 for i in range(1, int(1/h)+1):

y[i] = round(Y(x[i-1], y[i-1]), 5)

return x, y

def Runge\_Kutta\_5(y0, h):

y = [0 for i in range(int(1/h)+1)]

x = [h\*i for i in range(int(1/h)+1)]

y[0] = y0

k0 = lambda x, y: h\*func(x, y)

k1 = lambda x, y: h\*func(x + h/2, y + k0(x, y)/2)

k2 = lambda x, y: h\*func(x + h/2, y + k1(x, y)/2)

k3 = lambda x, y: h\*func(x + h/2, y + k2(x, y)/2)

k4 = lambda x, y: h\*func(x + h, y + k3(x, y))

Y = lambda x, y: y + (k0(x, y) + 2\*k1(x, y) + 2\*k2(x, y) + 2\*k3(x, y) + k4(x,

y))/8

for i in range(1, int(1/h)+1):

y[i] = round(Y(x[i-1], y[i-1]), 5)

return x, y

def func(x, y):

return 2\*y - 2