**Лабораторная работа №2**

**Тема:** Численные методы решения дифференциальных уравнений

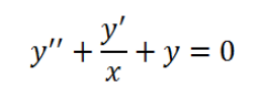
**Язык:** Python 3.6.5

**Постановка задачи:** : Написать программу, реализующую решение дифференциальных уравнений.

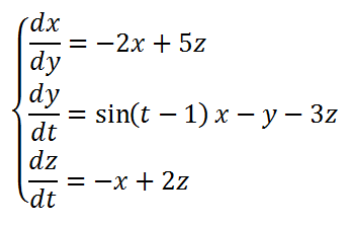
**Дифференциальное уравнение первого порядка:**



**Дифференциальное уравнение второго порядка:**



**Система дифференциальных уравнений:**



**Переменные:**

Модуль diffurs.py

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i | Счетчик цикла | int |
| a, b, c, d | Границы интегрирования | float |
| mainlist | Список с элементами главного меню | list |
| changelist | Список с элементами меню методов Эйлера и Рунге-Кутта | list |
| switch | Переключатель главного меню | int |
| switch1 | Переключатель меню методов Эйлера и Рунге-Кутта | int |
| printmain() | Печать главного меню | func |
| printchangelist() | Печать меню методов с постоянным шагом | func |

Модуль metods.py

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i, j | Счетчик цикла | int |
| x | x | int |
| h | Шаг вычисления | float |
| res | Результат | float |
| func() | Дифференциальное уравнение первого порядка | func |
| eler | Метод Эйлера | func |
| ruku | Метод Рунге-Кутта | func |
| ruchange | Вычисление F(i) | func |
| vtor | Вычисление ДУ второго порядка | func |
| sistema | Вычисление системы ДУ | func |

**Код программы:**

diffurs.py

import metods

def printmain():

for i in range(5):

print(mainlist[i])

print()

def printchangelist():

for i in range(4):

print(changelist[i])

print()

mainlist=["Главное меню", "1) Решение ДУ",

"2) Система ДУ", "3) ДУ 2ого порядка","4) Выход из программы"]

changelist=["Решение ДУ", "1) Метод Эйлера",

"2) Метод Рунге-Кутта", "3) Возврат в главное меню"]

while True:

printmain()

switch=int(input())

if switch==1:

while True:

printchangelist()

switch1=int(input())

if switch1==1:

metods.eler(0, 1, 10, 0, 1)

elif switch1==2:

metods.ruku(0, 1, 10, 0, 1)

elif switch1==3:

break

elif switch==2:

metods.sistema(1, 1.5, 0.1, 2, 1, 1)

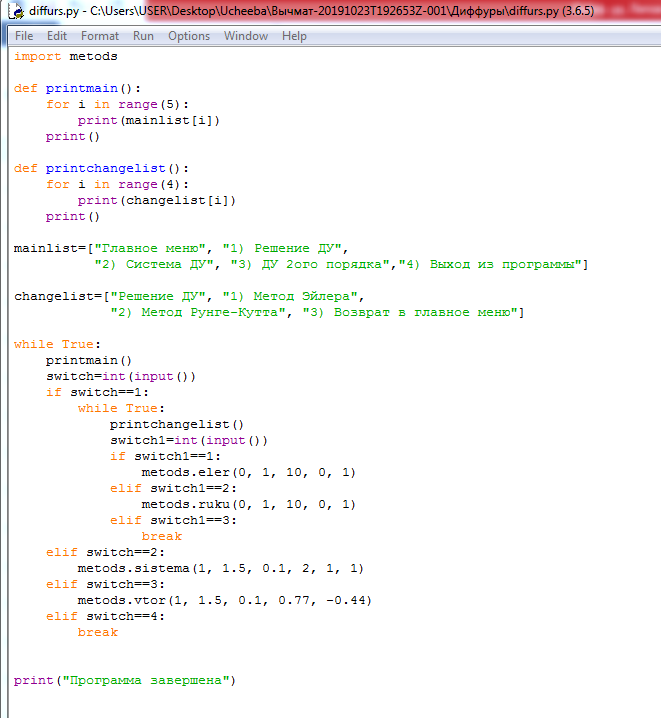
elif switch==3:

metods.vtor(1, 1.5, 0.1, 0.77, -0.44)

elif switch==4:

break

print("Программа завершена")



metods.py

from math import \*

def func(x, y):

return y\*(1-x)

def eler(a, b, n, x, y):

h=(b-a)/n

print("Шаг: ", h)

while x<=b-h:

print("x =", round(x, 1), "y =", y)

y += h \* func(x, y)

x += h

def ruku(a, b, n, x, y):

def ruchange(x, y, h):

res=[]

res.append(h\*func(x, y))

res.append(2\*h\*func(x+h/2, y+res[0]/2))

res.append(2\*h\*func(x+h/2, y+res[1]/2))

res.append(h\*func(x+h, y+res[2]))

return sum(res)/6

h=(b-a)/n

while x<=b-h:

print("x =", round(x, 1), "y =", y)

y+=ruchange(x, y, h)

x+=h

def vtor(a, b, h, y, z):

x=a

while x<b:

y1=y

print("x =", round(x, 5), "y =", round(y, 5), "z =", round(z, 5))

y+=h\*z

z-=h\*(z/x+y1)

x+=h

def sistema(a, b, h, x, y, z):

t=a

while t<b:

print("x =", round(x, 5),

"y =", round(y, 5),

"z =", round(z, 5),

"t =", round(t, 5))

x0=x

y0=y

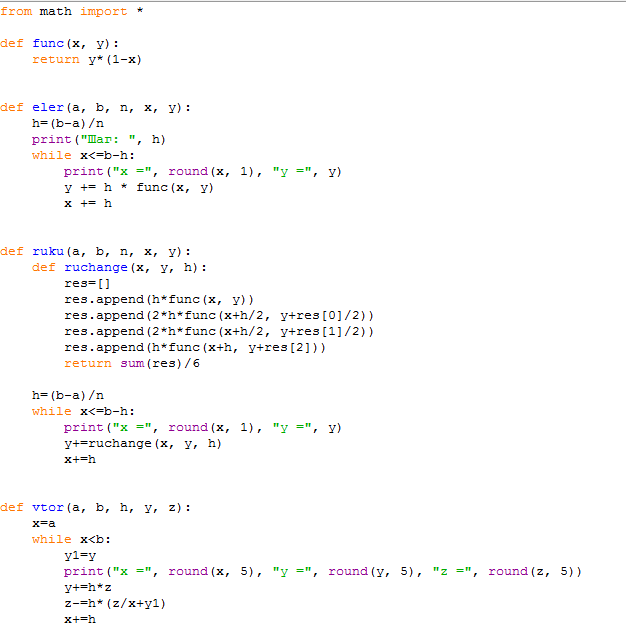
z0=z

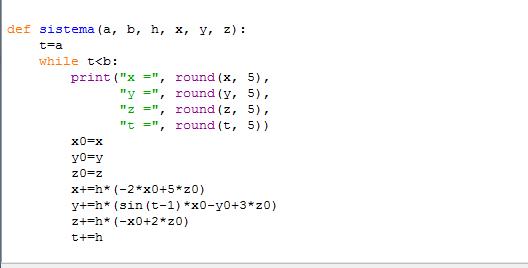
x+=h\*(-2\*x0+5\*z0)

y+=h\*(sin(t-1)\*x0-y0+3\*z0)

z+=h\*(-x0+2\*z0)

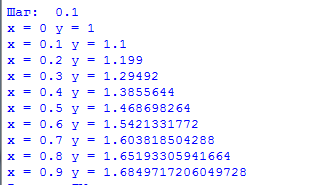
t+=h



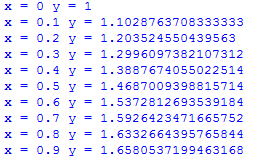


**Результат работы:**

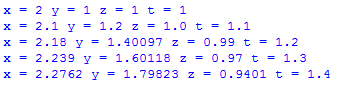
Метод Эйлера:



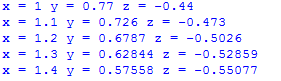
Метод Рунге-Кутта:



Система дифференциальных уравнений:



ДУ второго порядка:



**Вопросы:**

1) Какие методы целесообразно использовать для решения простых дифференциальных уравнений?

2) Каким выражением задается производная в методе Эйлера?

3) Каким образом задается интегральная кривая по методу Эйлера?

4) У какого метода выше точность при одинаковом шаге интегрирования, у метода Эйлера или Рунге-Кутта?

**Вывод:**

В ходе работы были реализованы различные методы решения дифференциальных уравнений. Полученные, с помощью таких методов, значения представлены в виде списка переменных.