Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Методы численного анализа»

**ОТЧЕТ**

к лабораторной работе №8

на тему:

**«Численное дифференцирование и интегрирование функций»**

БГУИР 6-05 0612 02 86

|  |
| --- |
| Выполнил студент группы 353505  МАРТЫНКЕВИЧ Евгений Дмитриевич |
|  |
| (дата, подпись студента) |
| Проверил доцент кафедры информатики  АНИСИМОВ Владимир Яковлевич |
|  |
| (дата, подпись преподавателя) |

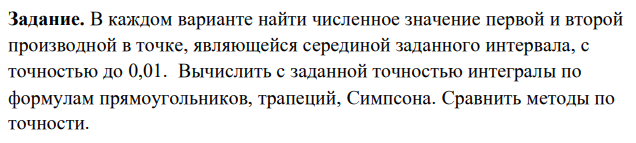
Минск 2024

**Содержание**

1. Цель работы
2. Задание
3. Программная реализация
4. Полученные результаты
5. Оценка полученных результатов
6. Вывод

**Цель работы**

* изучить методы численного вычисления производных и методы численного интегрирования;
* сравнить методы по трудоемкости, точности;
* составить программу методов численного вычисления;
* выполнить тестовые примеры и проверить правильность работы программы



**Вариант 3**

*Исходные данные*



**Программная реализация**

Код формул производной:

def diff\_method\_1(func, point, accuracy):  
 return (func.subs(x, point + accuracy) - func.subs(x, point)) / accuracy  
  
def diff\_method\_2(func, point, accuracy):  
 return (func.subs(x, point + accuracy) - func.subs(x, point - accuracy)) / (2 \* accuracy)  
  
def diff2\_method(func, point, accuracy):  
 return (func.subs(x, point + accuracy) - 2 \* func.subs(x, point) + func.subs(x, point - accuracy)) / (accuracy \*\* 2)

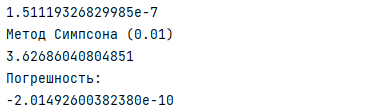
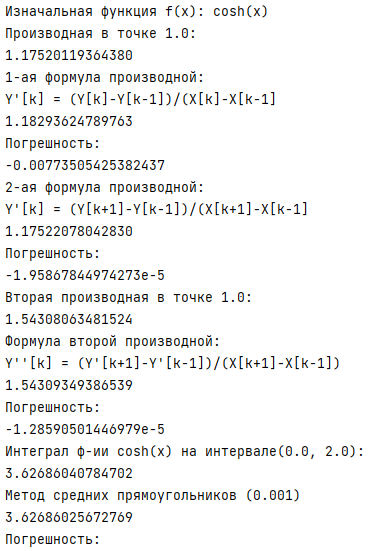
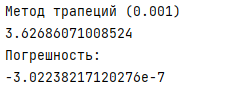
Код средних прямоугольников:

def integral\_middle\_rectangle(func, left, right, accuracy):  
 i = left  
 Integ = 0.  
 while(i < right - 0.5 \* accuracy):  
 Integ += accuracy \* func.subs(x, i + 0.5 \* accuracy)  
 i += accuracy  
 return Integ

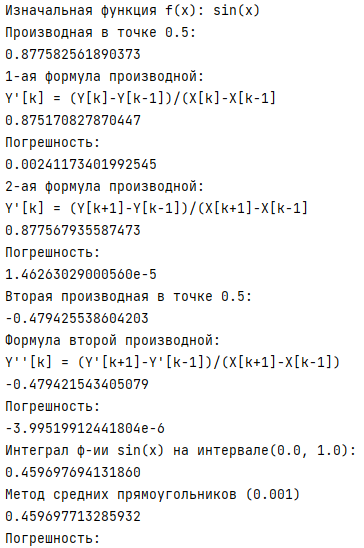
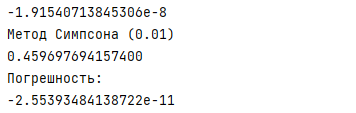
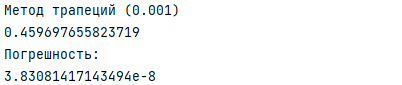
Код формулы Симпсона:

def simpthon\_method(func, left, right, accuracy):  
 X, Y = [], []  
 check = False  
 i = left  
 while i < right + 0.5 \* accuracy:  
 X.append(i)  
 i += accuracy  
 if(len(X) % 2 == 0):  
 X.append(right + accuracy)  
 check = True  
 for i in range(0, len(X)):  
 Y.append(func.subs(x, X[i]))  
 Integ = 0.  
 j = 0  
 while j < len(Y) - 2:  
 Integ += Y[j] + 4 \* Y[j + 1] + Y[j + 2]  
 j += 2  
 if(check):  
 Integ -= 0.5 \* (Y[len(Y)-3] + 4 \* Y[len(Y)-2] + Y[len(Y)-1])  
 return Integ \* accuracy / 3

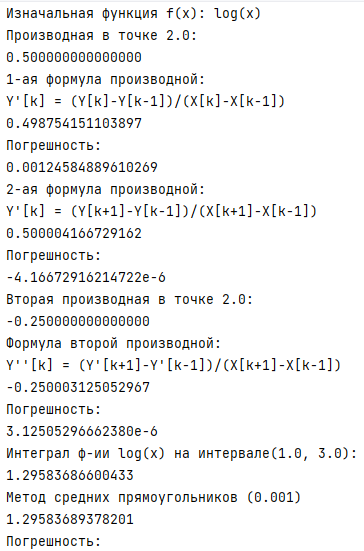
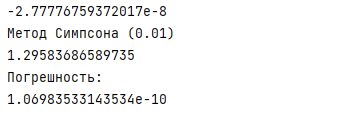
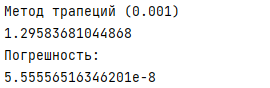
**Полученные результаты**

*Тестовый пример 1*

*  *

*Тестовый пример 2*

*  *

**Вывод**

В ходе выполнения лабораторной работы я изучил методы численного вычисления производных и методы численного интегрирования, написал программу их реализации на языке Python, правильность работы программы проверил на тестовых примерах.

На основании тестов можно сделать следующие выводы:

* Метод численного дифференцирования по формуле(2) более точный чем метод численного дифференцирования по формуле(1);
* Численное интегрирование по формуле Симпсона является более точным по сравнению с численным интегрированием методом средних прямоугольников и трапеций.
* Программа позволяет получить производную и интеграл с заданной точностью;