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 №6

на тему:

**«Интреполяционные многочлены»**

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) |

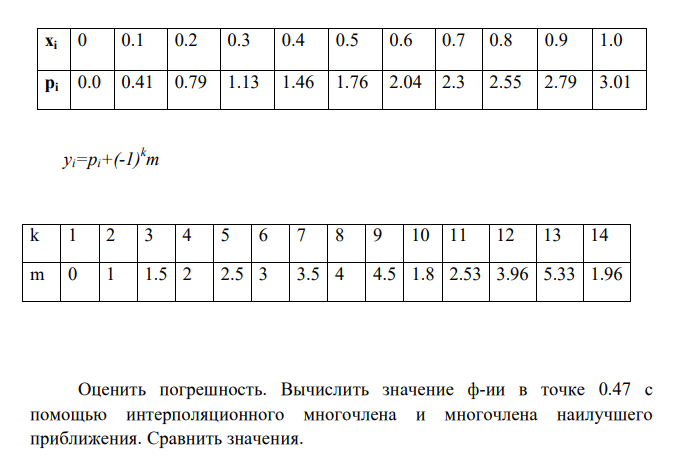
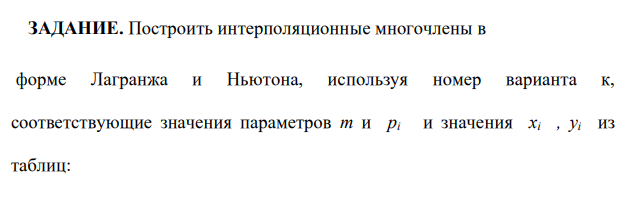
Минск 2024

**Содержание**

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

**Цель работы**

* изучить интерполяцию функций с помощью интерполяционных многочленов Лагранжа и Ньютона;
* составить программу нахождения интерполяционного многочлена методом Лагранжа и методом Ньютона;
* выполнить тестовые примеры и проверить правильность работы программы



**Вариант 3**

**Программная реализация**

*Исходные данные*



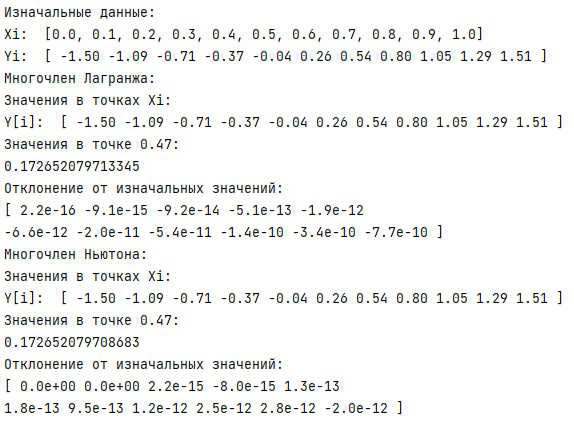
Код метода(многочлена) Лагранжа:

def lagrange\_method(X, Y):  
 f\_x = 1  
 L\_x = 0  
 for i in range(len(X)):  
 f\_x \*= (x - X[i])  
 for i in range(len(Y)):  
 f\_xj = f\_x / (x - X[i])  
 L\_x += (f\_xj / f\_xj.subs(x, X[i])) \* Y[i]  
 return L\_x

Код метода(многочлена) Ньютона:

def newton\_method(X, Y):  
 N\_x = Y[0]  
 for i in range(1, len(X)):  
 f\_i = add\_newton\_f(X[:i + 1:], Y[:i + 1:])  
 for j in range(i):  
 f\_i \*= (x - X[j])  
 N\_x += f\_i  
 return N\_x  
  
  
def add\_newton\_f(X, Y):  
 if(len(X) == 2):  
 return (Y[1] - Y[0]) / (X[1] - X[0])  
 a = add\_newton\_f(X[1::], Y[1::])  
 b = add\_newton\_f(X[:-1:], Y[:-1:])  
 return (a - b) / (X[-1] - X[0])

**Полученные результаты**

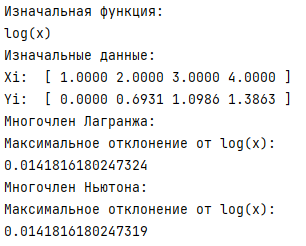


Для того чтобы проверить точность применим интерполяцию к функции log(x) с различным количеством узлов (т.е. точек Xi) и посмотрим на максимальную разницу между функциями на отрезке [1, 4].

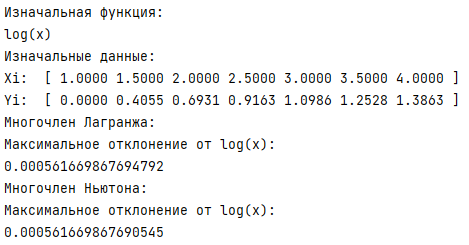
Исходя из формулы погрешности интерполяции, погрешность уменьшается при росте кол-ва узлов интерполяции:



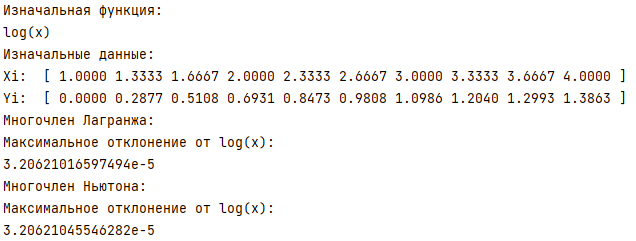
*Тестовый пример 1. Четыре узла*

**

*Тестовый пример 2. Семь узлов*



*Тестовый пример 3. Десять узлов*

****

**Вывод**

В ходе выполнения лабораторной работы я изучил метод Лагранжа и метод Ньютона построения интерполяционных многочленов, написал программу их реализации на языке Python, правильность работы программы проверил на тестовых примерах.

На основании тестов можно сделать следующие выводы:

* Достоинство интерполяционного многочлена Ньютона является удобство в расширении интерполяции и добавлении узлов
* Недостатком интерполяционного многочлена Ньютона является сложность в подсчете конечных разностей по сравнению с многочленом Лагранжа
* Точность интерполяции функции увеличивается с ростом количества узлов (точек Xi) отобранных для построения интерполяционного многочлена Лагранжа или Ньютона