Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе №1**

**Дисциплина**: Вычислительная математика

Вариант: 26

Выполнил студент гр. 5130901/20003 Гусев М.М.

(подпись)

Преподаватель Гаранин В.А.

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Санкт-Петербург

2024

**ЗАДАНИЕ**

**Вариант 26:**

Для 1 ≤ x ≤ 4 с h = 0.375 вычислить значения , используя для вычисления интеграла программу **QUANC8**. По полученным точкам построить сплайн-функцию и полином Лагранжа 8-й степени. Сравнить значения обеих аппроксимаций в точках xi = 1.1875 + 0.375i (i = 0, 1, … , 7).

**ХОД РАБОТЫ**

Для выполнения работы был использован язык программирования Python версии 3.11 и следующие библиотеки:

* SciPy для вычисления интеграла и интерполирования;
* NumPy для работы с массивами точек;
* Matplotlib для представления данных в виде графиков;
* PrettyTable для представления данных в виде таблицы.

Во время работы были поставлены следующие цели:

1. С помощью QUANC8 (или её аналога) вычислить значения f(x) в каждой из точек xk = 1 + 0.375k (k = 0, 1, …, 8);
2. По полученным 9 точкам построить полином Лагранжа 8-й степени и сплайн-функцию;
3. С помощью QUANC8 вычислить значения f(x) в каждой из точек xi = 1.1875 + 0.375i (i = 0, 1, … , 7);
4. Сравнить значения функции, полинома Лагранжа и сплайна в точках xi следующим образом: вычислить значения погрешностей в точках xi для полинома Лагранжа и сплайн-функции и сравнить.

**Графики и таблицы**

При построении исходной функции, полинома Лагранжа и сплайн-функции графики получаются идентичные в данном масштабе, если построить их в одном окне, то они вовсе сливаются в одну линию. Выглядит график следующим образом:

Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

*Рис. 1 График функции по значениям, вычисленным с помощью quad()*

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

*Рис. 2 График функции, построенный с помощью полинома Лагранжа*

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

*Рис. 3 График функции, построенный с помощью кубического сплайна*

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Рис. 4 Таблицы значений функции и аппроксимаций и таблица погрешностей аппроксимаций*

Из рис. 4 видно, что полином Лагранжа 8-й степени в большинстве точек приближает функцию лучше, чем сплайн-интерполяция. Однако приближение в обоих случаях довольно хорошее и имеет место быть.

Также построены графики зависимости погрешностей от x они приведены на рис. 5.

Вид полинома Лагранжа:

*L(x) = 0.0001177x8 – 0.002675x7 + 0.0266x6 – 0.1519x5 + 0.5504x4 – 1.318x3 + 2.11x2 – 2.252x + 1.634*

Изображение выглядит как текст, диаграмма, снимок экрана, График

Автоматически созданное описание

*Рис. 5 Графики погрешностей*

**Код**

Ниже приведён код программы:

from scipy.integrate import quad  
from scipy.interpolate import lagrange, CubicSpline  
from numpy import arange, e  
from prettytable import PrettyTable  
from matplotlib import pyplot  
  
  
def func(x: float):  
 def u(z: float):  
 return 1 / (e \*\* z \* (z + x))  
  
 return u  
  
  
def integral(f):  
 return quad(f, 0, 20, limit=30)  
  
  
def print\_values\_table():  
 table = PrettyTable()  
 table.add\_column(«x», x\_i)  
 table.add\_column(«f(x)», G\_i)  
 table.add\_column(«L(x)», values\_l)  
 table.add\_column(«S(x)», values\_s)  
 print(table)  
  
  
def print\_errs\_table():  
 table = PrettyTable()  
 table.add\_column(«x», x\_i)  
 table.add\_column(«errL», errs\_l)  
 table.add\_column(«errS», errs\_s)  
 print(table)  
  
  
def draw\_function\_plots():  
 pyplot.title(«Function»)  
 pyplot.plot(x\_i, G\_i, color=»Red»)  
  
 pyplot.figure()  
 pyplot.title(«Lagrange»)  
 pyplot.plot(x\_i, values\_l, color=»Blue»)  
  
 pyplot.figure()  
 pyplot.title(«Spline»)  
 pyplot.plot(x\_i, values\_s, color=»Green»)  
  
 pyplot.show()  
  
  
def draw\_errs\_plots():  
 pyplot.title(«Errors»)  
 pyplot.plot(x\_i, errs\_l, label=»e\_l(x)», color=»Blue»)  
 pyplot.plot(x\_i, errs\_s, label=»e\_s(x)», color=»Green»)  
 pyplot.show()  
  
  
if \_\_name\_\_ == «\_\_main\_\_»:  
  
 a = 1  
 b = 4  
 h = 0.375  
  
 # Задание узлов  
 x\_k = arange(a, b + h, h)  
  
 # Задание функции внутри интеграла  
 inner\_values = []  
 for x\_h in x\_k:  
 inner\_values.append(func(x\_h))  
  
 # Вычисление значения функции в узлах с помощью аналога quanc8  
 G\_k = []  
 for f in inner\_values:  
 G\_k.append(integral(f)[0])  
  
 # Построение полинома Лагранжа  
 poly = lagrange(x\_k, G\_k)  
  
 # Построене сплайн-функции  
 spline = CubicSpline(x\_k, G\_k)  
  
 # Промежуток для сравнения аппроксимаций  
 x\_i = 1.1875 + 0.375 \* arange(8)  
  
 # Вычисление значений полинома Лагранжа в точках x\_i  
 values\_l = poly(x\_i)  
 # Вычислений значений сплайна в точках x\_i  
 values\_s = spline(x\_i)  
  
 inner\_values2 = []  
 for x\_h in x\_i:  
 inner\_values2.append(func(x\_h))  
  
 # Вычисление значения функции в точках x\_i  
 G\_i = []  
 for f in inner\_values2:  
 G\_i.append(integral(f)[0])  
  
 # Вывод таблицы значений  
 print\_values\_table()  
  
 # Вычисление погрешностей для значений полинома Лагранжа и сплайна  
 errs\_l = [abs(x – y) for x, y in zip(G\_i, values\_l)]  
 errs\_s = [abs(x – y) for x, y in zip(G\_i, values\_s)]  
  
 # Вывод таблицы погрешностей  
 print\_errs\_table()  
  
 # Построение графиков функции и графиков погрешностей  
 draw\_function\_plots()  
 draw\_errs\_plots()

*Листинг 1*

**ВЫВОД**

В ходе работы получены навыки взаимодействия с библиотекой SciPy, в частности с программами lagrange(), quad() (аналог QUANC8), классом CubicSpline(). С их помощью построены аппроксимации заданной аналитически функцией, вычислены погрешности обеих аппроксимаций и выполнено их сравнение – таблично и графически. Также в процессе работы опробованы возможности библиотек Matplotlib и PrettyTable для построения графиков и таблиц. Все поставленные цели достигнуты, результаты соответствуют ожидаемым.