ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент		А. В. Аграновский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 5

ИНТЕРПОЛЯЦИОННАЯ КРИВАЯ CATMULL-ROM

по курсу:

КОМПЬЮТЕРНАЯ ГРАФИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4326		Г. С. Томчук
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

1 Цель работы

Цель работы — исследование интерполяционного сплайна Catmull-Rom и построение этой кривой с использованием математического пакета или языка программирования высокого уровня.

2 Формулировка задания

1. Реализовать алгоритм построения интерполяционной кривой Catmull-Rom на выбранном языке программирования высокого уровня или с использованием математического пакета.
2. Сгенерировать график гармонических колебаний на заданном интервале, выбрав 4+20 (вариант 20) опорные точки на одном периоде колебаний.
3. Построить интерполяционную кривую Catmull-Rom по этим опорным точкам на том же графике и визуально оценить точность восстановления гармонических колебаний.
4. Повторить построение с уменьшенным и увеличенным в 2 раза числом опорных точек, рассчитать и сравнить ошибку восстановления кривой для каждого случая.
5. Построить интерполяционную кривую Catmull-Rom с использованием полинома 24-го порядка и рассчитать ошибку восстановления.
6. Оформить результаты работы в виде графиков и таблиц, проанализировать полученные данные и сделать выводы о точности восстановления кривых при разном количестве опорных точек.

3 Теоретические сведения

1. Интерполяция и задачи Интерполяция — это метод нахождения функции, которая точно проходит через заданные опорные точки. В твоей лабораторной работе используется метод Catmull-Rom, который является частным случаем сплайновой интерполяции. Catmull-Rom сплайн — это кубический сплайн, который используется для построения гладких кривых, проходящих через заданные точки.

2. Интерполяционная кривая Catmull-Rom Катмулл-Ром сплайн часто

применяется для создания кривых в компьютерной графике и анимации, а также в других областях, где нужно сгладить данные или построить путь через точки.

Математически Catmull-Rom сплайн для четырёх опорных точек P0,P1,P2,P3P0,P1,P2,P3 определяется следующим образом:

$$P(t) = \frac{1}{2}\left[(2P_1) + (-P_0 + P_2)t + (2P_0 - 5P_1 + 4P_2 - P_3)t^2 + (-P_0 + 3P_1 - 3P_2 + P_3)t^3\right]$$

где t — это параметр, который изменяется от 0 до 1 для каждой промежуточной точки, и таким образом строится кривая через все точки.

3. Ошибка восстановления Ошибка восстановления интерполяционной кривой Catmull-Rom — это разница между исходными значениями функции (в данном случае гармоническими колебаниями) и значениями, полученными на интерполированной кривой. Ошибка восстановления может быть измерена как среднеквадратичное отклонение (RMSE) между истинными значениями и значениями, полученными с помощью интерполяции.

$$\text{Ошибка} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{\text{истинное}}(x_i) - y_{\text{интерполированное}}(x_i))^2}$$

где x_i — это координаты точек, $y_{\text{истинное}}$ — значения гармонических колебаний, а $y_{\text{интерполированное}}$ — значения на интерполированной кривой.

4 Скриншоты, иллюстрирующие результаты работы программы

На рис. 1, 2 и 3 представлены скриншоты окна программы, демонстрирующие соответственно кривые гармонических колебаний, Catmull-Rom и полиноминальной интерполяции.

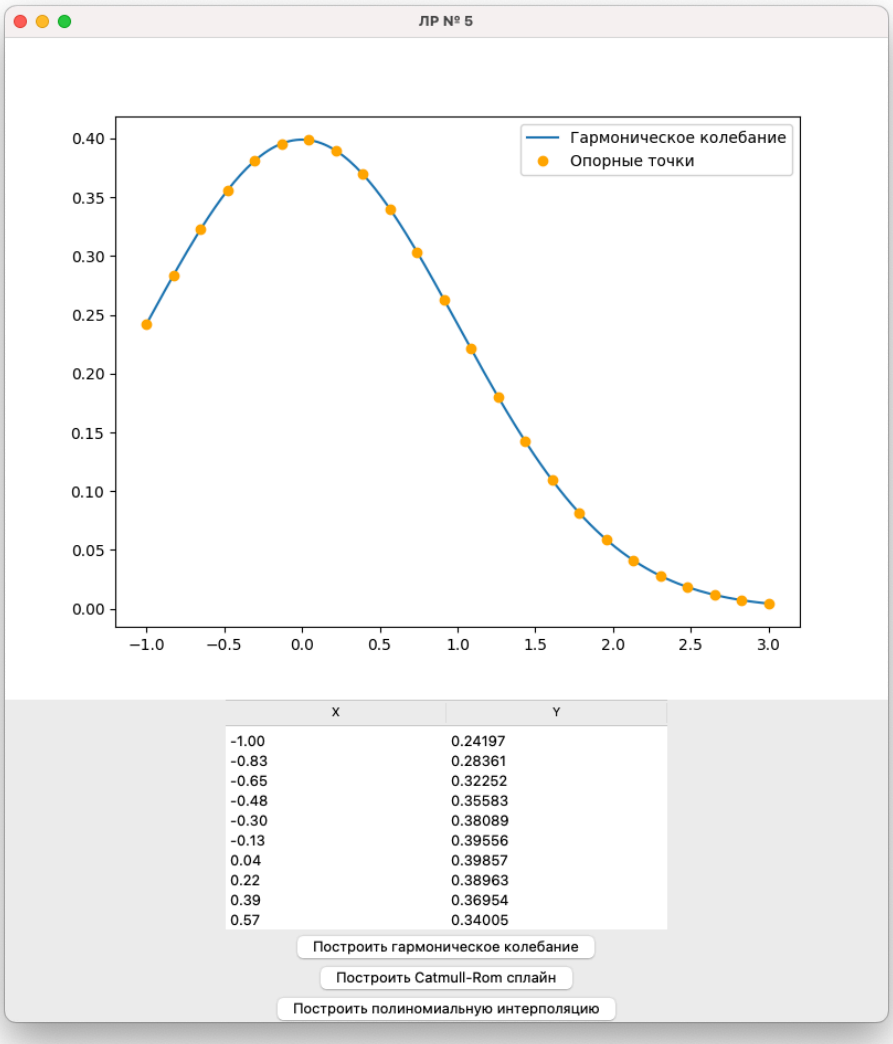

Рисунок 1

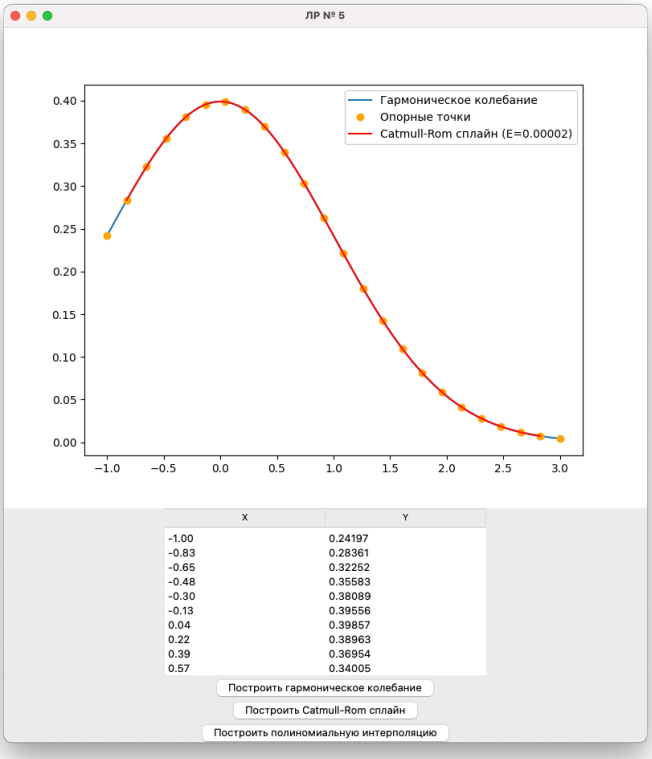

Рисунок 2

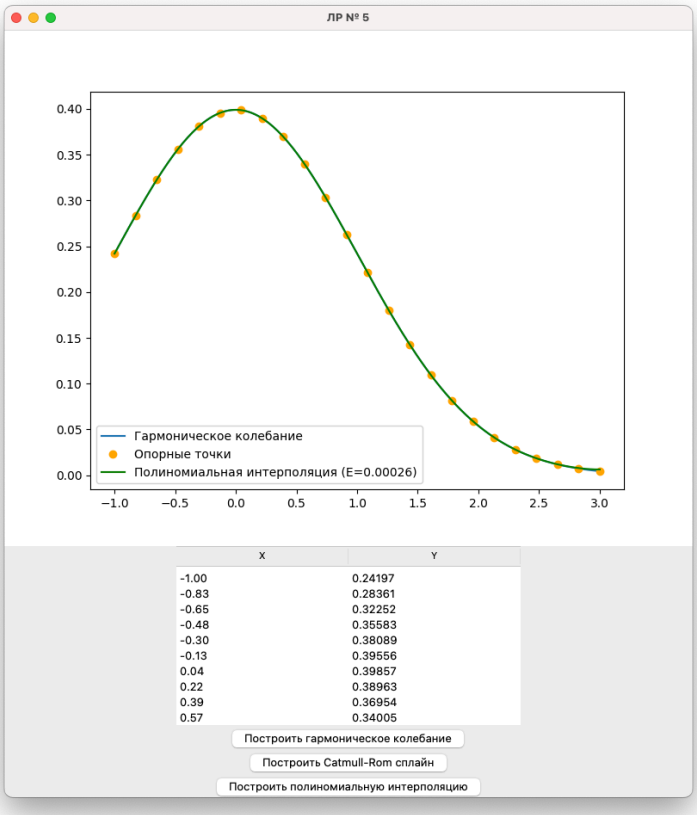

Рисунок 3

5 Вывод

В ходе выполнения работы был исследован метод интерполяции Catmull-Rom, который позволяет эффективно строить гладкие кривые, проходящие через заданные опорные точки. Этот метод широко используется в компьютерной графике и анимации для создания плавных движений и траекторий.

Было показано, что точность интерполяции Catmull-Rom зависит от количества выбранных точек. При увеличении числа точек на периоде гармонических колебаний точность интерполяции значительно улучшалась. Это подтверждается уменьшением ошибки восстановления, которая вычислялась как разница между истинными значениями функции и значениями, полученными на интерполированной кривой.

Ошибка восстановления, рассчитанная для кривой Catmull-Rom, уменьшалась при увеличении количества опорных точек и оставалась приемлемо малой даже при уменьшении числа точек. Это демонстрирует, что метод Catmull-Rom является достаточно устойчивым к изменениям количества опорных точек, в отличие от полиномиальной интерполяции высокого порядка, которая может приводить к осцилляциям при большом числе точек

В ходе работы была построена кривая на основе полинома N-го порядка, который также использовался для интерполяции опорных точек. Хотя полиномиальная интерполяция также показала хорошее приближение к исходным данным, она оказалась менее устойчивой при увеличении числа точек, что может привести к нестабильности и осцилляциям (эффект Рунге). В отличие от этого, интерполяция Catmull-Rom показала гораздо большую стабильность и плавность кривой.

Метод Catmull-Rom является полезным инструментом для построения траекторий и анимации в различных областях, таких как компьютерная графика, робототехника и визуализация данных. Результаты работы подтверждают эффективность использования этого метода для создания

плавных кривых, особенно при наличии ограничений на количество опорных точек.

В будущем можно расширить исследование, рассматривая другие методы интерполяции, такие как B-сплайны или интерполяцию по кривым Эрмита, а также анализируя влияние других параметров, таких как распределение точек по периоду и использование различных функций для моделирования данных.

Таким образом, выполненная лабораторная работа позволила глубже понять методы интерполяции, их применение и влияние на точность восстановления данных, а также продемонстрировала практическую ценность метода Catmull-Rom в области компьютерной графики.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики. - СПб.: БХВ-Петербург, 2003. – 560 с.
2. Петров, А. Н., Смирнов, И. В. Основы программирования на Python: Учебное пособие. – Санкт-Петербург: Издательство СПб ГУАП, 2022. – 320 с.
3. Бобылев, С. И. Основы компьютерной графики: учебное пособие / С. И. Бобылев. — М.: Издательство МГТУ им. Н. Э. Баумана, 2010. — 224 с.
4. Ласков, А. В. Математические основы компьютерной графики / А. В. Ласков, Ю. Г. Лоскутов. — СПб.: Питер, 2008. — 352 с.
5. NumPy. Documentation. URL: https://numpy.org/doc/stable/ (дата обращения: 27.10.2024).
6. Matplotlib. Documentation. URL: https://matplotlib.org/stable/contents.html (дата обращения: 27.10.2024).

ПРИЛОЖЕНИЕ А

```
import numpy as np
import tkinter as tk
from tkinter import ttk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
from scipy.interpolate import lagrange


# Исходная функция
def func(x):
    return (1 / np.sqrt(2 * np.pi)) * np.exp(-x ** 2 / 2)


# Опорные точки
x_values = np.linspace(-1, 3, 24)
y_values = func(x_values)


# Функция Catmull-Rom для генерации точек интерполяции между двумя
заданными точками
def catmull_rom_segment(p0, p1, p2, p3, num_points=50):
    t = np.linspace(0, 1, num_points)
    segment = []
    for tj in t:
        # Catmull-Rom формула для интерполяции
        x = 0.5 * (
                (2 * p1[0]) +
                (-p0[0] + p2[0]) * tj +
                (2 * p0[0] - 5 * p1[0] + 4 * p2[0] - p3[0]) * tj ** 2
+
                (-p0[0] + 3 * p1[0] - 3 * p2[0] + p3[0]) * tj ** 3
        )
        y = 0.5 * (
                (2 * p1[1]) +
                (-p0[1] + p2[1]) * tj +
                (2 * p0[1] - 5 * p1[1] + 4 * p2[1] - p3[1]) * tj ** 2
+
                (-p0[1] + 3 * p1[1] - 3 * p2[1] + p3[1]) * tj ** 3
        )
        segment.append([x, y])
    return np.array(segment)


# Основная функция для построения Catmull-Rom сплайна
def catmull_rom_spline(x_vals, y_vals, num_points=200):
    points = np.column_stack((x_vals, y_vals))
    spline_points = []

    # Генерация сегментов сплайна между всеми точками
    for i in range(1, len(points) - 2):
        p0, p1, p2, p3 = points[i - 1], points[i], points[i + 1],
points[i + 2]
```

```
        segment = catmull_rom_segment(p0, p1, p2, p3, num_points //
(len(points) - 3))
        spline_points.extend(segment)

    spline_points = np.array(spline_points)
    return spline_points[:, 0], spline_points[:, 1]


# Полиномиальная интерполяция Лагранжа
def lagrange_interpolation(x_vals, y_vals, x_dense):
    poly = lagrange(x_vals, y_vals)
    y_dense = poly(x_dense)
    return y_dense


# Функция для вычисления среднеквадратичной ошибки
def calculate_error(original_func, x_dense, y_dense):
    y_original = original_func(x_dense)
    return np.sqrt(np.mean((y_original - y_dense) ** 2))


# Создание интерфейса
class InterpolationApp:
    def __init__(self, root):
        self.root = root
        self.root.title("ЛР № 5")

        # Настройка matplotlib Figure
        self.figure, self.ax = plt.subplots(figsize=(8, 6))
        self.canvas = FigureCanvasTkAgg(self.figure, master=root)
        self.canvas.get_tk_widget().pack()

        # Таблица для координат
        self.table = ttk.Treeview(root, columns=("X", "Y"),
show="headings")
        self.table.heading("X", text="X")
        self.table.heading("Y", text="Y")
        self.table.pack()

        # Кнопки
        button1 = tk.Button(root, text="Построить гармоническое
колебание", command=self.plot_harmonic)
        button2 = tk.Button(root, text="Построить Catmull-Rom
сплайн", command=self.plot_catmull_rom)
        button3 = tk.Button(root, text="Построить полиномиальную
интерполяцию", command=self.plot_lagrange)
        button1.pack()
        button2.pack()
        button3.pack()

    # Функция для построения исходной функции
    def plot_harmonic(self):
        self.ax.clear()
```

```
        x_dense = np.linspace(-1, 3, 200)
        y_dense = func(x_dense)
        self.ax.plot(x_dense, y_dense, label="Гармоническое
колебание")
        self.ax.plot(x_values, y_values, 'o', color="orange",
label="Опорные точки")
        self.update_table(x_values, y_values)
        self.ax.legend()
        self.canvas.draw()

    # Функция для построения Catmull-Rom сплайна
    def plot_catmull_rom(self):
        self.ax.clear()
        x_dense = np.linspace(-1, 3, 200)
        y_dense = func(x_dense)
        self.ax.plot(x_dense, y_dense, label="Гармоническое
колебание")
        self.ax.plot(x_values, y_values, 'o', color="orange",
label="Опорные точки")

        x_spline, y_spline = catmull_rom_spline(x_values, y_values)
        error = calculate_error(func, x_spline, y_spline)
        self.ax.plot(x_spline, y_spline, 'r-', label=f"Catmull-Rom
сплайн (E={error:.5f})")
        self.ax.legend()
        self.canvas.draw()

    # Функция для построения полиномиальной интерполяции Лагранжа
    def plot_lagrange(self):
        self.ax.clear()
        x_dense = np.linspace(-1, 3, 200)
        y_dense = func(x_dense)
        self.ax.plot(x_dense, y_dense, label="Гармоническое
колебание")
        self.ax.plot(x_values, y_values, 'o', color="orange",
label="Опорные точки")

        y_lagrange_dense = lagrange_interpolation(x_values, y_values,
x_dense)
        error = calculate_error(func, x_dense, y_lagrange_dense)
        self.ax.plot(x_dense, y_lagrange_dense, 'g-',
label=f"Полиномиальная интерполяция (E={error:.5f})")
        self.ax.legend()
        self.canvas.draw()

    # Обновление таблицы
    def update_table(self, x_vals, y_vals):
        for i in self.table.get_children():
            self.table.delete(i)
        for x, y in zip(x_vals, y_vals):
            self.table.insert("", "end", values=(f"{x:.2f}",
f"{y:.5f}"))
```

```
# Запуск приложения
root = tk.Tk()
app = InterpolationApp(root)
root.mainloop()
```