**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №10**

**по дисциплине «Вычислительная математика»**

**Тема: Интерполяция функций для равноотстоящих узлов и узлов по многочлену Чебышёва**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3341 |  | Ягудин Д.Р. |
| Преподаватель |  | Пуеров Г. Ю. |

Санкт-Петербург

2025

**Цель работы.**

Целью работы является изучение и применение на практике интерполяции функции для равноотстоящих узлов и узлов по многочлену Чебышёва.

Вариант 26

f(x) = 1/(3+ 2cos(x)) - x3

**Выполнение работы.**

Равноотстоящие узлы:

Дан интервал с левой границей *a* и правой границей *b*. Случайным образом выбирается количество узлов n, после чего формируется вектор x, содержащий равномерно распределённые узлы. Чтобы между соседними узлами было одинаковое расстояние, используется следующая формула:

*i* = 0, 1, …, n – 1

Узлы Чебышёва:

Узлы Чебышёва первого рода задаются на интервале [-1, 1] и определяются по формуле:

Для произвольного интервала [a,b], их можно перенести с помощью линейного преобразования:

Для сравнения интерполяции при различных типах узлов построим графики с помощью библиотеки matplotlib. В качестве метода интерполяции используется полином Ньютона.

**Тестирование.**

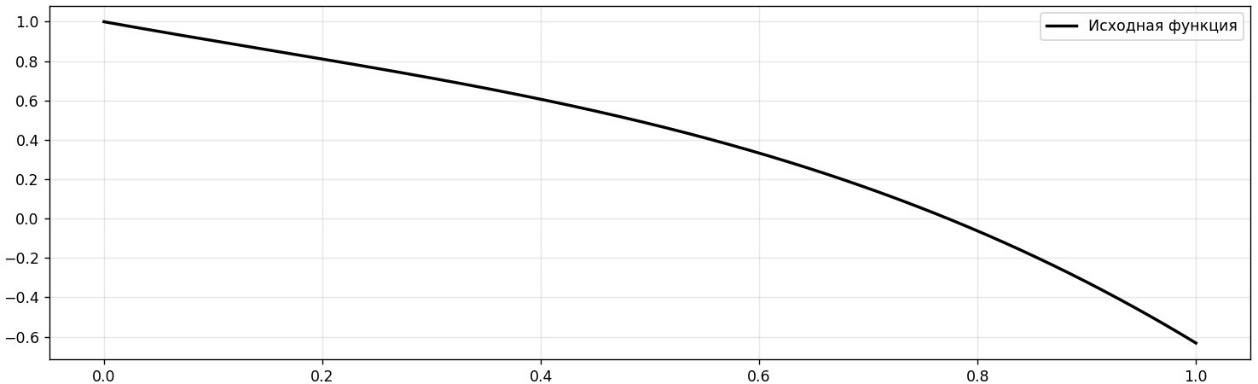


Рисунок № 1 – График исходной функции

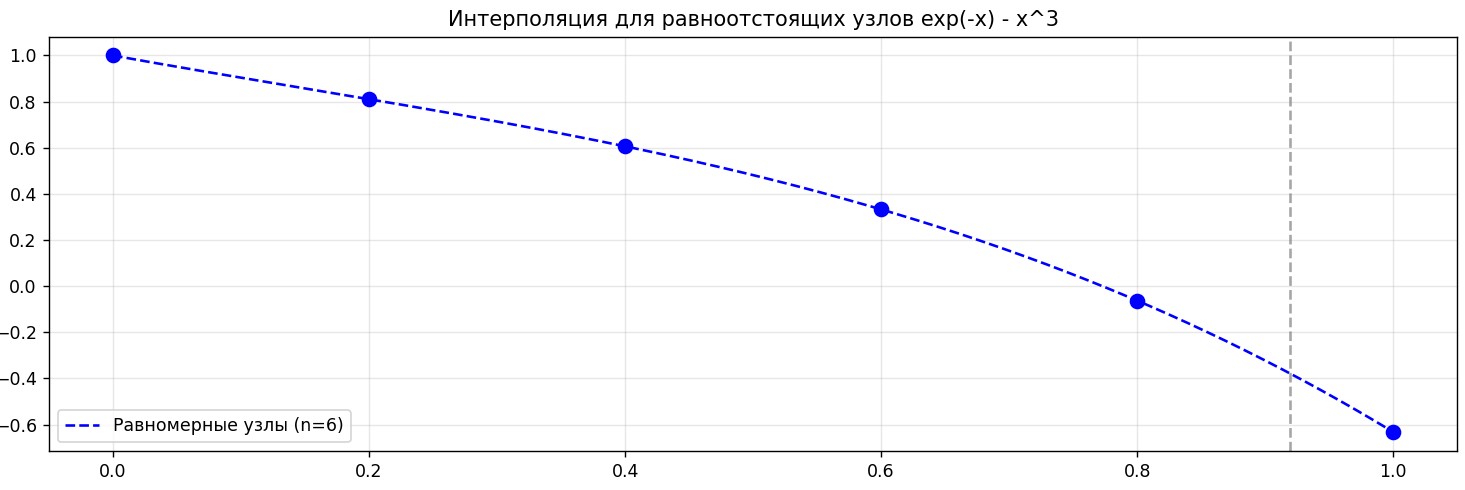


Рисунок № 2 – График, полученный интерполяцией для равноотстоящих узлов

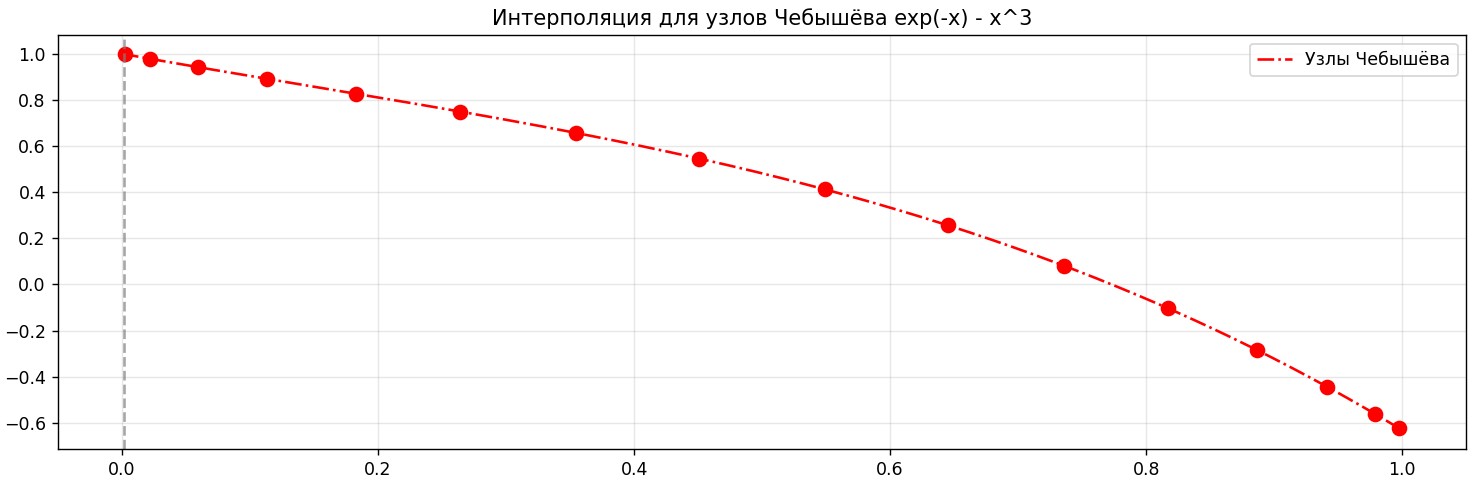


Рисунок № 3 – График, полученный интерполяцией для узлов по многочлену Чебышёва

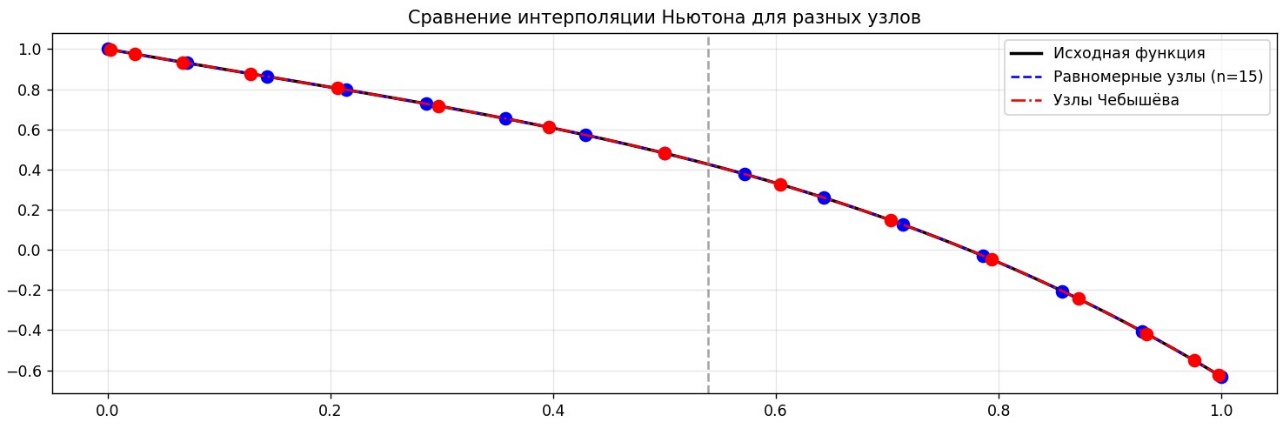


Рисунок № 4 – График, показывающий интерполяции для разных узлов и исходную функцию.

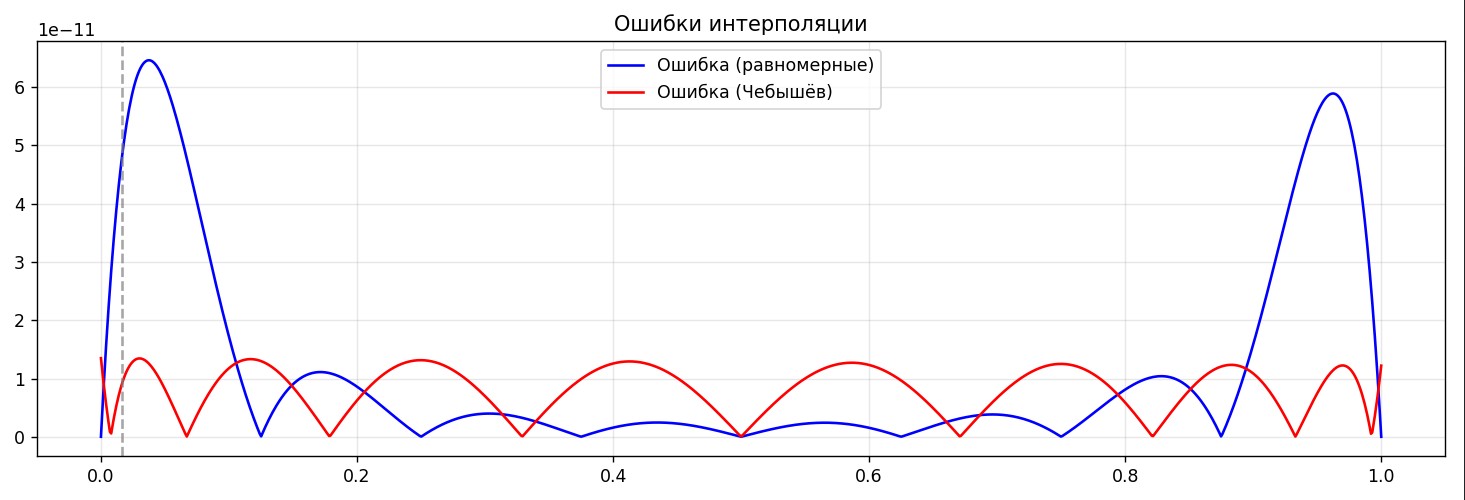


Рисунок № 5 – График, показывающий ошибки интерполяции для равноотстоящих и Чебышёва узлов.

Видно, что график Чебышёва перекрывает остальные, так как они слились в один (минимальная погрешность интерполяции).

У равноотстоящих узлов есть свойство, что на концах интервала демонстрируются заметные отклонения. Это связано с тем, что при использовании равноотстоящих узлов наблюдается эффект Рунге — резкие колебания полинома на границах интервала, особенно при увеличении числа узлов.

Узлы Чебышёва, в отличие от равномерных, распределены с сгущением к концам интервала, что позволяет минимизировать максимальную погрешность интерполяции и избежать описанных колебаний

Для наблюдения данного явления рассмотрим поведении интерполяции на функции Рунге ( ) на отрезку [-1, 1]

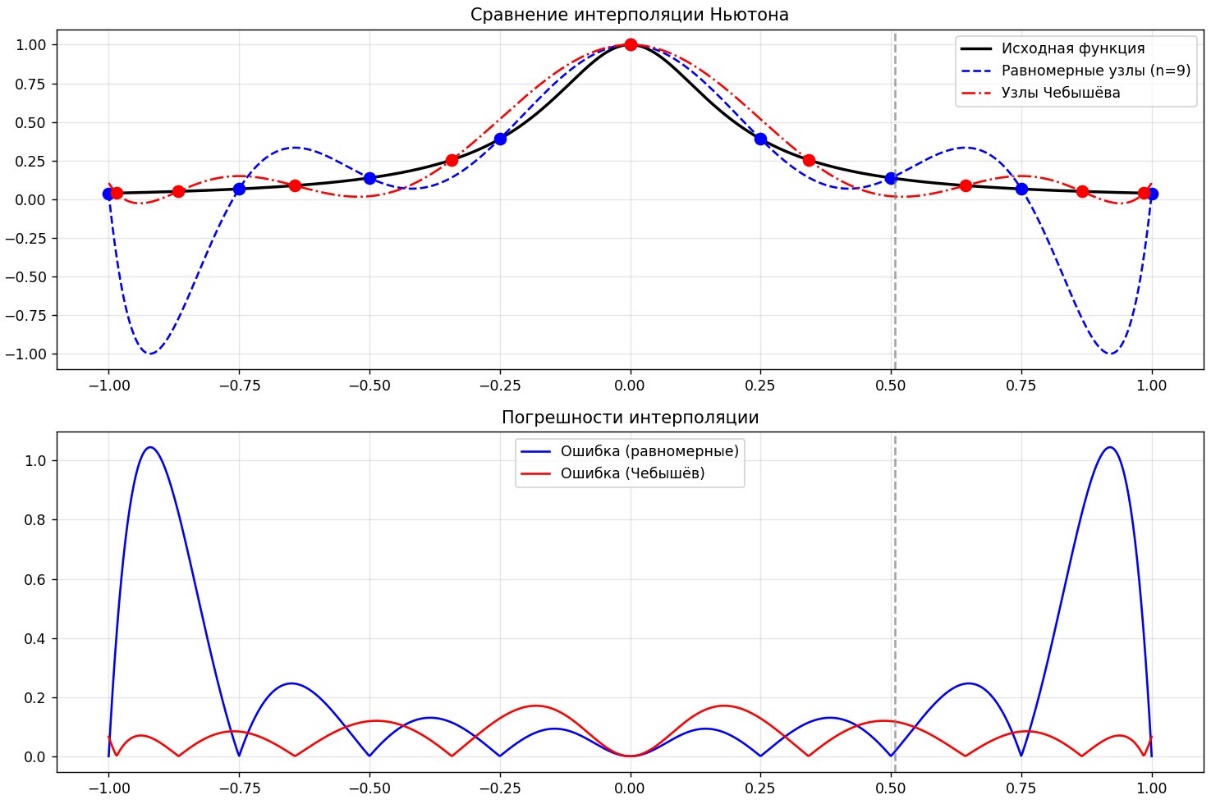


Рисунок № 6 – Поведение различных узлов на концах интервала для функции Рунге

**Выводы.**

В ходе выполнения работы были изучена и применена на практике интерполяция функции для равноотстоящих узлов и узлов по многочлену Чебышёва.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.py

import numpy as np

import matplotlib.pyplot as plt

import random

def divided\_difference(x, y):

"""Вычисление разделенных разностей для полинома Ньютона"""

coefficients = y.copy()

for j in range(1, len(x)):

for i in range(len(x) - 1, j - 1, -1):

coefficients[i] = (coefficients[i] - coefficients[i - 1]) / (x[i] - x[i - j])

return coefficients

def newton\_interpolation(coefficients, x\_nodes, x\_target):

"""Интерполяция полиномом Ньютона в точке x\_target"""

result = coefficients[0]

term = 1

for i in range(1, len(coefficients)):

term \*= (x\_target - x\_nodes[i - 1])

result += coefficients[i] \* term

return result

def chebyshev\_nodes(a, b, n\_nodes):

j = np.arange(1, n\_nodes + 1)

nodes = np.cos((2 \* j - 1) \* np.pi / (2 \* n\_nodes))

return 0.5 \* (a + b) + 0.5 \* (b - a) \* nodes

def experiment(f, a, b):

"""Проводит эксперимент по интерполяции функции f на отрезке [a, b]"""

x\_plot = np.linspace(a, b, 1000)

n\_nodes = random.randint(5, 20)

# Узлы и значения функции

uniform\_x = np.linspace(a, b, n\_nodes)

cheb\_x = chebyshev\_nodes(a, b, n\_nodes)

uniform\_y = f(uniform\_x)

cheb\_y = f(cheb\_x)

# Коэффициенты Ньютона

uniform\_coeffs = divided\_difference(uniform\_x, uniform\_y)

cheb\_coeffs = divided\_difference(cheb\_x, cheb\_y)

# Интерполяции

uniform\_interp = [newton\_interpolation(uniform\_coeffs, uniform\_x, x) for x in x\_plot]

cheb\_interp = [newton\_interpolation(cheb\_coeffs, cheb\_x, x) for x in x\_plot]

# Проверка в случайной точке

x\_target = random.uniform(a, b)

exact\_target = f(x\_target)

uniform\_target = newton\_interpolation(uniform\_coeffs, uniform\_x, x\_target)

cheb\_target = newton\_interpolation(cheb\_coeffs, cheb\_x, x\_target)

# График

plt.figure(figsize=(12, 8))

plt.subplot(2, 1, 1)

plt.plot(x\_plot, f(x\_plot), 'k-', label='Исходная функция', linewidth=2)

plt.plot(x\_plot, uniform\_interp, 'b--', label=f'Равномерные узлы (n={n\_nodes})')

plt.plot(uniform\_x, uniform\_y, 'bo', markersize=8)

plt.plot(x\_plot, cheb\_interp, 'r-.', label='Узлы Чебышёва')

plt.plot(cheb\_x, cheb\_y, 'ro', markersize=8)

plt.axvline(x=x\_target, color='gray', linestyle='--', alpha=0.7)

plt.legend()

plt.title('Сравнение интерполяции Ньютона')

plt.grid(True, alpha=0.3)

plt.subplot(2, 1, 2)

plt.plot(x\_plot, np.abs(f(x\_plot) - uniform\_interp), 'b-', label='Ошибка (равномерные)')

plt.plot(x\_plot, np.abs(f(x\_plot) - cheb\_interp), 'r-', label='Ошибка (Чебышёв)')

plt.axvline(x=x\_target, color='gray', linestyle='--', alpha=0.7)

plt.legend()

plt.title('Погрешности интерполяции')

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.show()

# Вывод результатов

print(f"\nРезультаты для {n\_nodes} узлов:")

print(f"Точка интерполяции: x = {x\_target:.4f}")

print(f"Точное значение: f(x) = {exact\_target:.6f}")

print(f"Равномерные узлы: интерполяция = {uniform\_target:.6f}, ошибка = {abs(exact\_target - uniform\_target):.2e}")

print(f"Узлы Чебышёва: интерполяция = {cheb\_target:.6f}, ошибка = {abs(exact\_target - cheb\_target):.2e}")

print("\nРавномерные узлы:", np.round(uniform\_x, 4))

print("Узлы Чебышёва:", np.round(cheb\_x, 4))

if \_\_name\_\_ == "\_\_main\_\_":

func1 = lambda x: np.exp(-x) - x\*\*3

func2 = lambda x: 1 / (25 \* x\*\*2 + 1)

experiment(func1, 0, 1)

experiment(func2, -1, 1)