**МИНОБРНАУКИ РОССИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»**

**(НГТУ)**

Кафедра «Вычислительные системы и технологии»

Дисциплина «Методы и средства обработки сигналов»

**Отчёт по 1 лабораторной работе**

Выполнила:

Студентка группы 23 – ВМз

Бутыкова Юлия Олеговна

Проверил:

доцент, кандидат технических наук

Авербух Михаил Леонидович

Нижний Новгород, 2025

**Постановка задачи**

Имеем функцию y(x) = a1 \* sin(b1 \* x) + a2 \* sin(b2 \* x) + a3 \* sin(b3 \* x)

Пользователем задаются: a1, b1, a2, b2, a3, b3, x0 (начальное значение), xк (конечное значение), Δx (шаг). Расчет y(x) по заданным значениям a1, b1, a2, b2, a3, b3, x0 (начальное значение), xк (конечное значение), Δx (шаг). Отображение векторов x и y (в виде таблицы).

Построение графика y(x) по указанным векторам.

**Ход выполнения – демонстрация работы программы**

Код программы на языке С++. Файл source.cpp

#include <iostream> // для ввода/вывода

#include <fstream> // для работы с файлами

#include <cmath> // для математических функций

#include <iomanip> // для форматирования вывода

// Функция для расчета y(x)

double calculate\_y(double a1, double b1, double a2, double b2, double a3, double b3, double x) {

return a1 \* sin(b1 \* x) + a2 \* sin(b2 \* x) + a3 \* sin(b3 \* x);

}

int main() {

// Ввод параметров

double a1, b1, a2, b2, a3, b3;

double x0, xk, dx;

std::cout << "Введите параметры функции:\n";

std::cout << "a1 = "; std::cin >> a1;

std::cout << "b1 = "; std::cin >> b1;

std::cout << "a2 = "; std::cin >> a2;

std::cout << "b2 = "; std::cin >> b2;

std::cout << "a3 = "; std::cin >> a3;

std::cout << "b3 = "; std::cin >> b3;

std::cout << "Начальное значение x0 = "; std::cin >> x0;

std::cout << "Конечное значение xk = "; std::cin >> xk;

std::cout << "Шаг Δx = "; std::cin >> dx;

// Создаем файл с данными

std::ofstream dataFile("data.dat");

// Вывод заголовка таблицы

std::cout << "\nТаблица значений:\n";

std::cout << std::setw(15) << "x" << std::setw(15) << "y\n";

std::cout << std::string(30, '-') << "\n";

// Запись заголовка в файл

dataFile <<"# x y\n";

// Генерируем данные

for (double x = x0; x <= xk; x += dx) {

double y = calculate\_y(a1, b1, a2, b2, a3, b3, x);

// Запись в файл

dataFile << x << " " << y << "\n";

// Вывод в консоль

std::cout << std::fixed << std::setprecision(4);

std::cout << std::setw(15) << x << std::setw(15) << y << "\n";

}

dataFile.close();

// Сообщение о завершении

std::cout << "\nРезультаты сохранены в файл data.dat\n";

return 0;

}

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1. Скриншот из терминала. Создание программы

Изображение выглядит как текст, снимок экрана, дисплей, компьютер

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2. Скриншот из файла data.dat

Для построения графика будем использовать внешнюю программу - Gnuplot

Файл plot.gp

# Настройки графика

set title "График функции y(x)"

set xlabel "x"

set ylabel "y"

set grid

# Построение графика из файла

plot 'data.dat' with lines lw 2 title 'y(x)'

# Сохранение в файл

set term png

set output 'graph.png'

replot

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3. Скриншот из терминала. Построение графика

Изображение выглядит как линия, График, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4. График функции. Файл graph.png

**Выводы**

Разработали программу на языке программирования С++ для расчёта значений y(x) = a1 \* sin(b1 \* x) + a2 \* sin(b2 \* x) + a3 \* sin(b3 \* x). Реализовали ввод параметров функции пользователем.

Для отрисовки графика использовали программу Gnuplot.

Использование Gnuplot имеет ряд преимуществ:

1.Высокое качество графиков

2.Простота настройки внешнего вида

3.Возможность сохранения в различных форматах (pdf, svg, png)

4.Автоматическая обработка данных

Разработанная программа может применяться для:

1.Исследования поведения сложных тригонометрических функций

2.Демонстрации влияния параметров на форму графика

3.Обучения работе с математическими функциями

4.Создания наглядных материалов