ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

**Федеральное государственное образовательное бюджетное учреждение**

**высшего профессионального образования**

**«Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»**

Факультет Информационных технологий и программной инженерии

Кафедра Программной инженерии и вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине:

**«Программирование»**

тема: Анализ сигнала на выходе электрической цепи

Передаточная характеристика – 7 вариант

Входной сигнал –7 вариант

Выполнил студент группы ИКПИ-41:

Дымнич К.А. \_\_\_\_\_\_\_\_

Дата выполнения: «13» Май

Проверил:

Хазиев Н.Н. \_\_\_\_\_\_\_\_

Санкт-Петербург

2025

Оглавление

[Аннотация 4](#_Toc199042703)

[Задание к курсовой работе 5](#_Toc199042704)

[1 Общие сведения 6](#_Toc199042705)

[1.1 Обозначение и наименование программы 6](#_Toc199042706)

[1.2 Языки разработки: 6](#_Toc199042707)

[2 Функциональное назначение 6](#_Toc199042708)

[2.1 Решаемые задачи 6](#_Toc199042709)

[2.2 Назначение: 6](#_Toc199042710)

[2.3 Ограничения в функциональности: 7](#_Toc199042711)

[3 Таблица идентификаторов 8](#_Toc199042712)

[4 Описание логической структуры 9](#_Toc199042713)

[4.1 Алгоритм программы 9](#_Toc199042714)

[4.2 Описание функций 9](#_Toc199042715)

[4.3 Используемые методы 10](#_Toc199042716)

[4.4 Связи программы с другими программами 11](#_Toc199042717)

[5 Используемые технические средства 12](#_Toc199042718)

[6 Вызов и загрузка 13](#_Toc199042719)

[6.1 Способ вызова программы с соответствующего носителя данных 13](#_Toc199042720)

[6.2 Входные точки в программу 13](#_Toc199042721)

[7 Входные данные 14](#_Toc199042722)

[7.1 Характер и организация входных данных 14](#_Toc199042723)

[8 Выходные данные 16](#_Toc199042724)

[8.1 Характер и организация выходных данных 16](#_Toc199042725)

[8.2 Контрольный расчёт 16](#_Toc199042726)

[8.3 Формат и кодирование выходных данных 17](#_Toc199042727)

[9 Структура кода 19](#_Toc199042728)

[10 Заключение 20](#_Toc199042729)

[11 Список используемой литературы 21](#_Toc199042730)

[12 Сокращения 22](#_Toc199042731)

[13 Приложения 24](#_Toc199042732)

[13.1 Приложение 1 24](#_Toc199042733)

[a) main.c 24](#_Toc199042734)

[b) app.c 24](#_Toc199042735)

[c) funct.c 25](#_Toc199042736)

[d) Заголовочные файлы 26](#_Toc199042737)

[13.2 Приложение 2 27](#_Toc199042738)

[a) menu.sh 27](#_Toc199042739)

[b) variables.sh 29](#_Toc199042740)

[c) p1.sh 30](#_Toc199042741)

[d) p2.sh 30](#_Toc199042742)

[e) functions.sh 31](#_Toc199042743)

[13.3 Приложение 3 31](#_Toc199042744)

Аннотация

Программный продукт "Анализатор сигналов электрических цепей" представляет собой консольное приложение, разработанное для автоматизированного анализа характеристик линейных электрических цепей. Программа выполняет численное моделирование прохождения сигнала через электрическую цепь с заданными параметрами.

**Основные функциональные возможности:**

* расчет выходного сигнала по известному входному воздействию;
* определение временных параметров сигнала (длительность импульса, время нарастания);
* оценка точности вычислений с заданной погрешностью (до 1%).

**Технические особенности реализации:**

* язык разработки: С (ядро вычислений) + Bash (интерфейс);
* платформа: ОС Linux (Ubuntu);
* графическая подсистема: wxMaxima;
* способ взаимодействия: командная строка (Command line interface).

**Ключевые алгоритмы:**

* дискретизация временной оси;
* кусочно-линейная аппроксимация;
* итерационный метод уточнения параметров;
* автоматическое построение графиков.

Программа разработана в соответствии с требованиями ЕСПД (ГОСТ 19.ххх) и предназначена для использования в учебном процессе и инженерных расчетах. Особенностью решения является сочетание высокой точности вычислений (использование 32-битной арифметики с плавающей точкой IEEE 754) с простотой использования через командный интерфейс.

Объем исходного кода: ~500 строк (без учета зависимостей)  
Требования к аппаратному обеспечению: процессор x86-64, 512 МБ ОЗУ, 10 МБ дискового пространства.

Задание к курсовой работе

Работа посвящена решению задач машинного анализа электрических цепей.

В курсовой работе необходимо для заданной электрической цепи по известному входному сигналу определить выходной сигнал для N равностоящих моментов времени, а затем определить некоторые его характеристики с погрешностью не более 1%. Варианты параметров входного сигнала (код А) и передаточной характеристики (код Б) aэлектрической цепи приведены в приложении. Номер варианта определяется преподавателем индивидуально для каждого студента.

|  |  |
| --- | --- |
| **Входной сигнал** | **Рабочий набор** |
|  |  |

|  |  |
| --- | --- |
| **Выходной сигнал** | **Рабочий набор** |
|  |  |

* + - 1. в ходе работы необходимо:
* произвести расчет входного и выходного сигнала в контрольных точках, используя при этом математический пакет wxmaxima;
* написать текст программы на языке Си;
* произвести запись полученных результатов в файлы данных;
* используя математический пакет Wxmaxima (электронные таблицы), построить графики зависимости напряжений входных и выходных сигналов от времени;
* объединить программу на Си и Wxmaxima (LibraOffice.Calc), вызов отчета с помощью скрипта на Bash.

# Общие сведения

## Обозначение и наименование программы

Для корректной работы программа требует установленную руси-фицированную версию операционной системы Ubuntu Linux.

Также необходима установка стороннего ПО wxMaxima. Установить его можно командой в терминале:

* + - 1. **sudo apt-get install wxmaxima;**

Компилятор gcc обычно является встроенным в ОС Linux, однако при его отсутствии его можно установить последовательностью команд:

* + - 1. **sudo apt update;**
      2. **sudo apt install build-essential;**

## Языки разработки:

Программа написана на языке программирования Си — на нём реализо-вана основная функциональность.

Меню реализовано с использованием Bash-скриптов, которые также запускают скрипт для wxMaxima, оформленный в виде текстового файла с расширением .mac.

# Функциональное назначение

## Решаемые задачи

Программа предназначена для численного и графического анализа сигналов в электрических цепях. Она решает следующие задачи:

**Моделирование сигналов:**

* + - 1. расчёт функции входного напряжения Uвх(t), заданной в аналитической форме;
      2. вычисление выходного напряжения Uвых(t), используя кусочно-линейную передаточную характеристику.

**Графическая визуализация:**

* + - 1. построение графиков Uвх(t) и Uвых(t) с помощью wxMaxima;
      2. экспорт полученных данных в форматы, совместимые с другими пакетами (например, GNU Plot или LibreOffice Calc).

## Назначение:

Программа **АСВЭЦ** предназначена для работы в среде **Ubuntu Linux**. Программа АСВЭЦ предназначена для работы в среде Ubuntu Linux.

Основное применение — образовательное: визуализация работы электрических цепей и сравнение различных численных методов анализа сигналов.

Также программа пригодна для инженерных целей — быстрой оценки параметров цепей с нелинейными элементами.

Проверка аналитических решений гарантирует точность вычислений. Дополнительно, программа поддерживает автоматизацию обработки результатов для различных наборов параметров.

## Ограничения в функциональности:

* + - 1. **совместимость с ОС:** программа работает только в Ubuntu Linux и не поддерживает Windows.
      2. ограничения по входным данным:
* временной диапазон жёстко зафиксирован: t ∈ [1, 10] (можно изменить вручную в коде);
* параметры цепи заданы для варианта №7;
* максимальное количество точек **nmax = 10 000** (определено размером массива).
  + - 1. численные ограничения:
* используется тип данных float, что ограничивает точность;
* итерационные методы могут не сойтись при слишком малом значении **eps**.
  + - 1. системные требования:
* только ОС Ubuntu Linux;
* наличие wxMaxima (для визуализации) и gcc (для компиляции) обязательно.
  + - 1. ограничения пользовательского интерфейса:
* отсутствие графического интерфейса — работа осуществляется через консоль;
* низкая устойчивость к некорректному вводу.
  + - 1. **Примечание:** при необходимости использовать программу для других параметров, требуется вручную изменить настройки в исходном файле funct.c.

# Таблица идентификаторов

Таблица 6.1:

**Таблица идентификаторов**

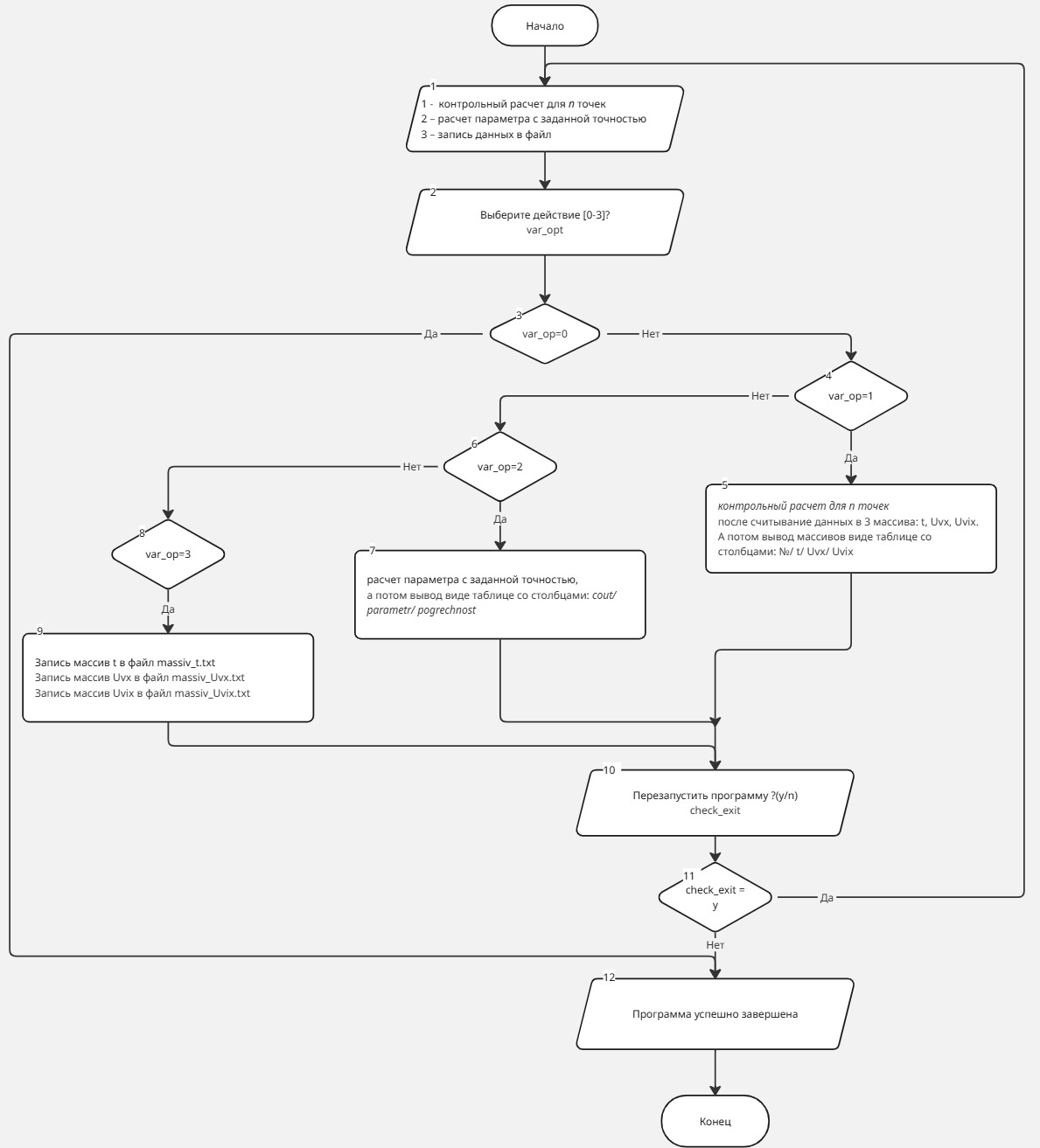
|  |  |
| --- | --- |
| **Идентификаторы** | **Пояснения** |
| **a0** | Постоянная составляющая напряжения |
| **a1** | Коэффициент линейного изменения напряжения во времени |
| **a2** | Коэффициент квадратичного изменения напряжения во времени |
| **tn** | Время, до которого рассматривается изменение напряжения (начальное время) |
| **tk** | Время, до которого рассматривается изменение напряжения (конечное время) |
| **a** | Постоянное значение выходного напряжения при малых входных напряжениях |
| **b** | Коэффициент квадратичной зависимости выходного напряжения |
| **Uvx1** | Пороговое значение входного напряжения, при котором меняется зависимость |
| **n** | Количество элементов в массиве |
| **eps** | Погрешность |
| **p** | Параметр |
| **t** | Массив времени |
| **Uvx** | Массив входного напряжение |

# Описание логической структуры

## Алгоритм программы

Рисунок 5.1:

**Рисунок блок схемы**



## Описание функций

Таблица 5.2:

**Таблица функций**

| **Функция** | **Описание** |
| --- | --- |
| **main** | Основная точка входа в программу на языке C. Запускает функцию run\_app. |
| **run\_app** | Управляющая функция приложения на C. Выбирает действие (прямой или приближённый расчёт). |
| **form\_time** | Формирует массив времён t по шагу времени. |
| **form\_Uvx** | Вычисляет массив напряжений Uvx на основе временных точек. |
| **form\_Uvix** | Вычисляет массив напряжений Uvix по кусочной линейной аппроксимации. |
| **parametr** | Вычисляет среднее значение сигнала U за промежуток времени. |
| **form\_tabl1** | Выводит таблицу значений времён и напряжений (t, Uvx, Uvix). |
| **control\_calc** | Основная функция расчёта (вариант 1). Формирует массивы и выводит таблицу. |
| **approx\_value** | Основная функция приближённого расчёта (вариант 2) с заданной точностью. |
| **menu.sh** | Скрипт bash. Отображает меню программы и управляет пользователем. |
| **out\_zast** | Выводит заставку программы. |
| **out\_menu** | Основное меню программы. Обрабатывает пользовательский ввод. |
| **start** | Запускает программу, очищает массивы данных. |
| **out\_file** | Записывает массивы t, Uvx, Uvix в текстовые файлы и запускает построение графиков. |
| **style** | Выводит текст с заданным стилем (цветом). |
| **clear\_line** | Очищает текущую строку терминала. |
| **is\_number** | Проверяет, является ли введённое значение числом. |
| **pg1** | Запускает бинарное приложение (вариант 1) и выводит результаты в таблице. |
| **pg2** | Запускает бинарное приложение (вариант 2) и выводит таблицу с погрешностями. |

## Используемые методы

Программа АСВЭЦ использует следующие методы:

* + - 1. **численные расчёты:**
* разбивает интервал t ∈ [0, 10] на N точек с шагом Δt = π / (N − 1));
* вычисляет значение сигнала по формуле Uвх(t) = 2+5\*t[i]+0.3\*pow(t[i],2) для каждой точки t[i];
* используются функции pow() из библиотеки math.h;
  + - 1. **применяет передаточную характеристику цепи через условные операторы:**
* итерационный метод уточнения (начинает с N = 11 точек, сравнивает параметр с предыдущим расчётом, если погрешность больше eps, удваивает N и повторяет расчёт);
* методы визуализации (построение графиков скриптов с wxMaxima);
* взаимодействие с ОС (осуществляется через Bash-скрипт).

## Связи программы с другими программами

Программа АСВЭЦ в ходе своей работы запускает следующие стандартные для Linux библиотеки (способ установки указан в п. 1.2):

* wxMaxima:
* используется для построения графиков;
* должна быть установлена в системе;
* gcc:
* обязателен для компиляции кода.

Все внешние вызовы выполняются через bash-скрипт menu.sh.

Программа работает в консоли, все вычисления выполняются последовательно, wxMaxima и gcc не являются частью программы. Программа также ориентирована на академические расчёты и локальное использование на Linux-системах.

# Используемые технические средства

Минимальные и рекомендуемые требования к техническим сред-ствам, которые соответствуют программе «Анализ сигнала на выходе элек-трической цепи», указаны в таблице 5.1. А также bash выше 4 версии.

Таблица 5.1:

**Требования к техническим средствам**

|  |  |  |
| --- | --- | --- |
| **Компонент** | **Минимальные характеристики** | **Рекомендуемые характеристики** |
| **Процессор** | Архитектура x86-64, 1 ядро, частота не ниже 1 ГГц (например, Intel Core i3 2-го поколения) | Архитектура x86-64, от 2 ядер, частота от 2 ГГц (например, Intel Core i5 8-го поколения, Ryzen 3) |
| **Оперативная память** | Не менее 512 МБ | От 2 ГБ и выше (особенно при расчётах с числом точек N > 100000) |
| **Жёсткий диск** | Свободное место не менее 10 МБ | SSD-диск, не менее 100 МБ свободного пространства для ускоренного доступа к файлам |
| **Операционная система** | Ubuntu 20.04+, Debian 10+ или совместимые дистрибутивы Linux | Astra Linux или дистрибутивы Linux с GUI (например, GNOME, KDE) |
| **Дополнительное ПО** | - GCC версии не ниже 9.3.0- wxMaxima версии не ниже 20.06 | - GCC версии 12 и выше- wxMaxima версии 23.04 и выше |
| **Монитор** | Разрешение экрана не менее 1280×720 | Разрешение экрана Full HD (1920×1080) |
| **Графическая карта** | Интегрированная, не ниже Intel HD Graphics 4000 | - |

# Вызов и загрузка

## Способ вызова программы с соответствующего носителя данных

* + - 1. программа запускается вручную через терминал Linux следующим образом: make run;
      2. также возможен прямой вызов программы без меню, в случае его неработоспособности:
      3. ./myapp pg n eps:
* Pg – это выбор контрольного расчёта/расчёт параметра;
* n – кол-во элементов в массиве;
* eps – предел точности погрешности.

## Входные точки в программу

Точкой входа в программу является главная функция – main() в файле main.c, которая вызывает функцию run\_app ().

Для работы программы необходимы установленные раннее пакеты (wxMaxima, gcc, eog), а также все обязательные файлы (Makefile, functions.sh, p1.sh, p2.sh, variables.sh, menu.sh, make\_graphs.mac, app.c, funct.c, app.h, funct.h, globals.h, main.c).

Программа не требует прав суперпользователя (root), а все файлы данных для графиков сохраняются в текущую директорию.

# Входные данные

## Характер и организация входных данных

Программа АСВЭЦ (Автоматизированная Система Визуализации Электрических Цепей) использует два типа входных данных

**Фиксированные параметры цепи:**

* + - 1. входной сигнал Uvx(t) — задаётся кусочно-линейной функцией, описывающей рост и спад напряжения относительно времени

Рисунок 7.1.1:

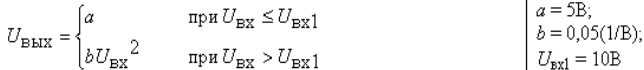
**Данные для Uvx**



* + - 1. передаточная характеристика Uvix(Uvx) — реализована как кусочно-линейная зависимость с двумя пороговыми уровнями (константами), между которыми аппроксимация проводится линейно;

Рисунок 7.1.2:

**Данные для Uvx**



**Пользовательские параметры (вводятся через консоль):**

* + - 1. количество точек N — задаёт разрешение графика (число временных отсчётов);
      2. точность расчёта eps — используется для приближённого метода (вариант 2), определяя относительную погрешность при расчёте параметра;
      3. подготовка входных данных не требуется — все вспомогательные параметры и данные генерируются внутри программы автоматически.

**Диапазоны допустимых значений:**

* + - 1. N ∈ [2, 10 000] — ограничение задано директивой #define N 10000.
      2. eps ∈ [0.001, 99.99] — значение вводится в процентах и преобразуется в доли (eps/100) внутри программы;
      3. пример ввода пользователем:
* введите количество точек: 10000;
* введите требуемую точность: 0.1.

**Кодировка:**

* + - 1. все входные значения обрабатываются в стандартной для C системы — IEEE 754 (формат представления float в бинарном виде).

# Выходные данные

## Характер и организация выходных данных

Программа АСВЭЦ генерирует два типа выходных данных:

* + - 1. текстовые файлы с результатами расчётов:
* massiv\_t.txt — массив значений времени t;
* massiv\_Uvx.txt — массив значений входного сигнала;
* massiv\_Uvix.txt — массив значений выходного сигнала.
  + - 1. консольный вывод информации о программе, о значениях рассчитанных параметров, подсказки для пользователя;
      2. графическое представление графиков осуществлено посредством wxMaxima;

Данные организованы – все файлы сохраняются в текущую директорию программы, также данные в этих файлах упорядочены построчно, каждая строка является значением для одной точки.

## Контрольный расчёт

Таблица 8.2.1:

**Таблица “Контрольный расчет для n точек”**

| **Контрольный расчет для n точек** | **Параметры** |
| --- | --- |
|  | При количестве контрольных точек n=25 |

Таблица 8.2.2:

**Таблица “Расчёт параметра с заданной точностью”**

| **Расчёт параметра с заданной точностью** | **Параметры** |
| --- | --- |
|  | n = 10, eps = 1% |

Таблица 8.2.3:

**Таблица “График Uvx и параметры”**

| **График Uvx** | **Параметры** |
| --- | --- |
|  | n = 25 |

Таблица 8.2.4:

**Таблица “График Uvix и параметры”**

| **График Uvix** | **Параметры** |
| --- | --- |
|  | n = 25 |

## Формат и кодирование выходных данных

Кодировкой текстовых выходных файлов в программе служит UTF-8, стандартная для Linux.

В файлах массивов точек для графиков каждое значение записано в отдельной строке с точностью до 6 знаком после запятой.

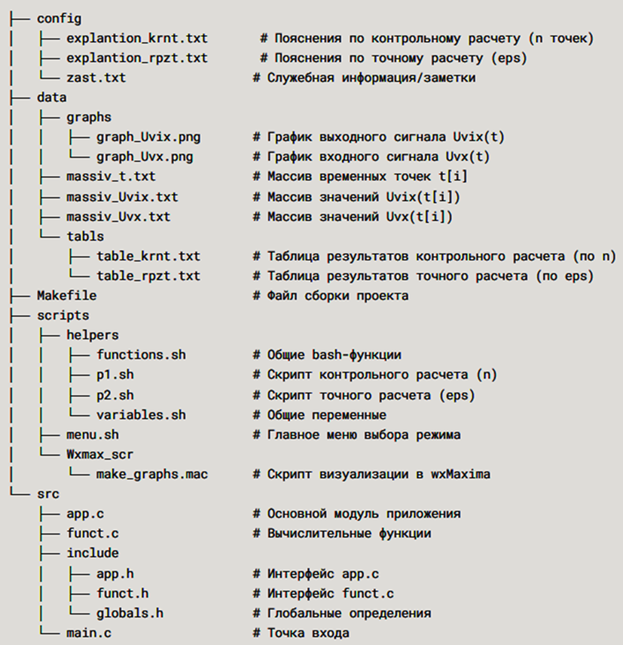
В консоли выводятся вещественные числа с фиксированной точностью (6 знаков).

Пример выходных данных:

* + - 1. (файл massic\_Uvx.txt):
* 0.000000
* 0.109134
* 0.216760

# Структура кода

* Рисунок 9:
* Характеристика фалов



Код Cи находится в приложение 1.

Код Bash находится в приложение 2.

Код Wxmaxima находится в приложение 3.

# Заключение

В ходе проекта я приобрела ценные практические навыки в различных областях. На техническом уровне я научилась реализовывать математические модели электрических цепей на языке C, освоила методы численного анализа с заданной точностью до 1% и получила опыт работы с математическими библиотеками для научных вычислений. В программировании я разработала программу для преобразования сигналов, создала систему хранения и обработки результатов расчетов, а также написала скрипты для автоматизации всего вычислительного цикла.

Особое внимание я уделила анализу данных: освоила методы визуализации результатов через построение графиков, научилась проводить сравнительный анализ сигналов и интерпретировать полученные данные. В работе с инструментами я приобрела опыт использования систем построения графиков, оптимизации вычислительных процессов и автоматической обработки данных.

Важными стали и организационные навыки: я научилась структурировать вычислительный процесс, документировать результаты, а также проверять и верифицировать расчеты. Полученные практические данные полностью соответствовали теоретическим расчетам, что подтвердило корректность реализованного алгоритма и достоверность результатов. Все эти навыки я последовательно применяла на каждом этапе проекта - от первоначального моделирования до финального анализа, уделяя особое внимание точности вычислений и наглядности представления данных.

# Список используемой литературы

1. ГОСТ 19.402-78. Единая система программной документации. Пояснительная записка.
2. . ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем.
3. Брауде Э.Я. Основы программирования на языке C. — М.: Финансы и статистика.
4. документация GNU Bash. URL: <https://www.gnu.org/software/bash/>
5. документация wxMaxima. URL: <https://wxmaxima-developers.github.io/wxmaxima/>
6. документация Си. URL: <https://c-language-documentation.vercel.app/>

# Сокращения

* **ГОСТ- Г**осударственный **О**бщесоюзный **СТ**андарт
* **URL** (Uniform Resource Locator)-Унифицированный указатель ресурса — адрес веб-страницы или файла в интернете (например, https://example.com).
* **ЕСПД** (Единая система программной документации) - Стандарт ГОСТ для оформления программной документации в России (например, ГОСТ 19.ххх).
* **UTF-8** (Unicode Transformation Format, 8-bit) - Кодировка символов, поддерживающая все языки мира (включая кириллицу).
* **ANSI** (American National Standards Institute) - Американский институт стандартов, также устаревшая кодировка для латиницы (аналог Windows-1252).
* **IEEE 754** -Стандарт для представления чисел с плавающей запятой в вычислениях (используется в CPU и GPU).
* **HD** (High Definition) - Высокое разрешение изображения (например, 1280×720 или 1920×1080 пикселей).
* **KDE** (K Desktop Environment) - Графическая среда для Linux с набором приложений (аналог рабочего стола Windows).
* **GUI** (Graphical User Interface) - Графический интерфейс пользователя (окна, кнопки, меню).
* **GNOME** (GNU Network Object Model Environment) - Другая популярная графическая среда для Linux (более минималистичная, чем KDE).
* **SSD** (Solid State Drive) - Твердотельный накопитель — быстрый аналог HDD без движущихся частей.
* **МБ** (Мегабайт) - 1 МБ = 1 048 576 байт (или 10⁶ байт в маркетинге).
* **ГБ** (Гигабайт) - 1 ГБ = 1024 МБ (объём памяти или хранилища).
* **ГГц** (Гигагерц) - Единица частоты процессора (1 ГГц = 1 млрд тактов в секунду).
* **АСВЭЦ** - «Анализ сигнала на выходе электрической цепи».
* **ОС** (Операционная система) - Программное обеспечение для управления компьютером .
* **ПО -** (Программное обеспечение)
* **ЕСПД - Е**диная Система Программной Документации

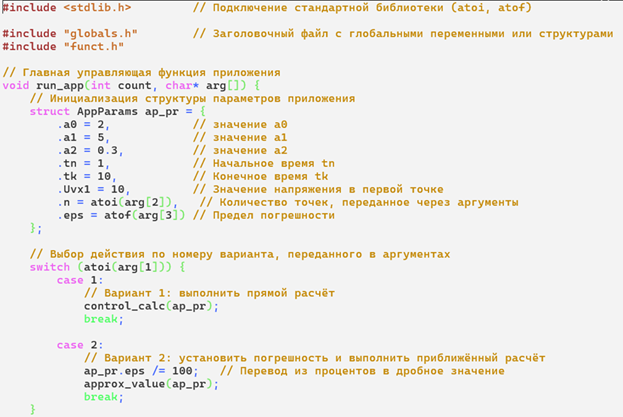
# Приложения

## Приложение 1

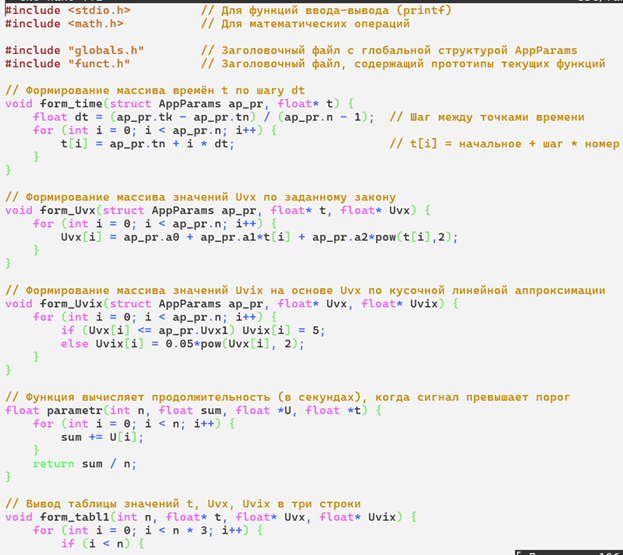
### main.c

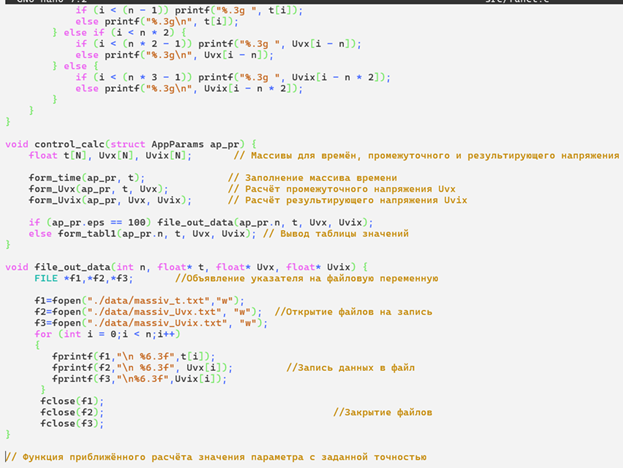


### app.c



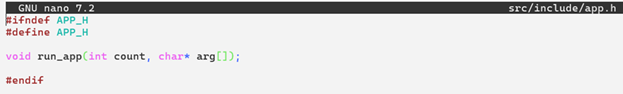
### funct.c

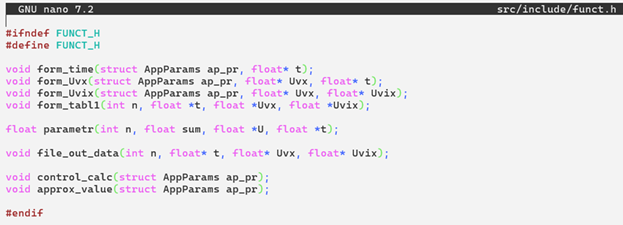


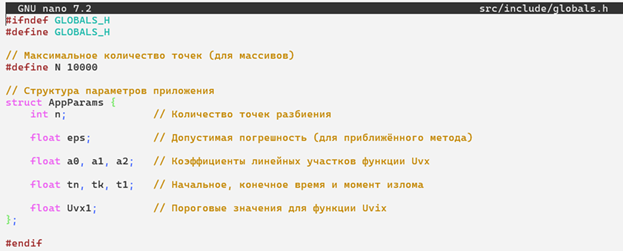




### Заголовочные файлы



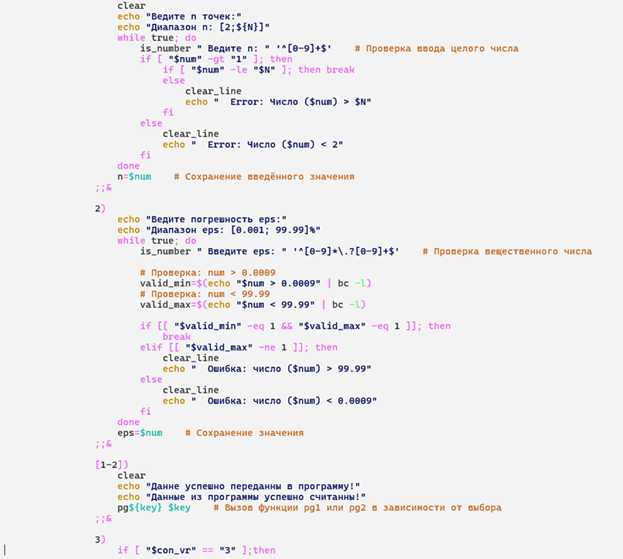




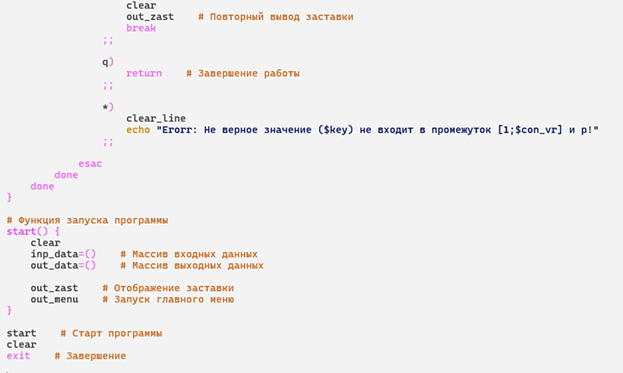
## Приложение 2

### menu.sh

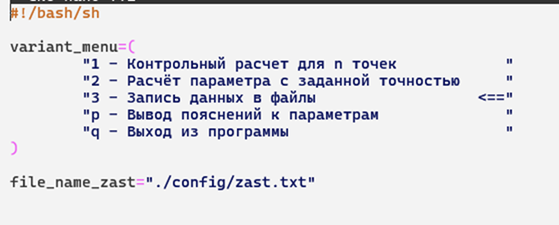




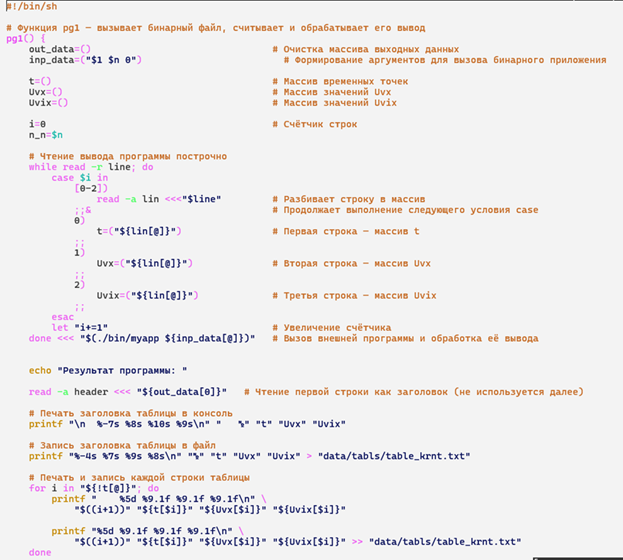




### variables.sh



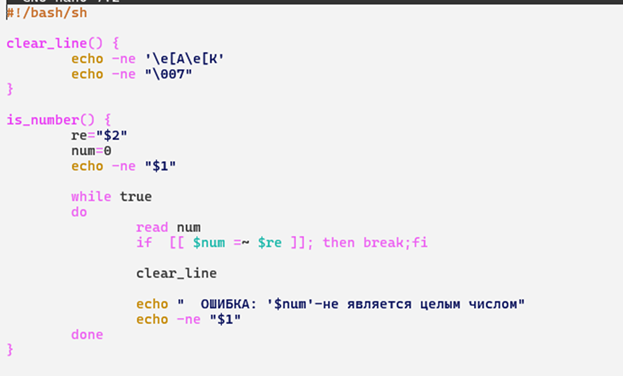
### p1.sh



### p2.sh



### functions.sh



## Приложение 3

