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 – 2 вариант

Входной сигнал – 14 вариант

Выполнил студент(ка):

Костров Д. Э. ИКПИ-44 \_\_\_\_\_\_\_\_\_\_\_\_\_

*(Ф.И.О., № группы) (подпись)*

Дата выполнения:

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

*(Ф.И.О.) (подпись)*

Санкт-Петербург

2025

## Аннотация

Программа анализирует сигналы в электрической цепи, вычисляя выходное напряжение по заданному алгоритму. Разработана на языке C с использованием математических функций. Позволяет проводить расчеты для произвольного количества точек, сохранять результаты в файлы и визуализировать данные. Результаты верифицированы в среде WxMaxima.

Оглавление

[Аннотация 2](#_Toc196825895)

[Задание 4](#_Toc196825896)

[Таблица идентификаторов 5](#_Toc196825897)

[Блок схемы 6](#_Toc196825898)

[Текст программы 11](#_Toc196825899)

[Описание программы 16](#_Toc196825900)

[Общие сведения 16](#_Toc196825901)

[Функциональное обозначение 16](#_Toc196825902)

[Логическая структура 16](#_Toc196825903)

[Используемые технические средства 16](#_Toc196825904)

[Вызов и загрузка 16](#_Toc196825905)

[Выходные данные 17](#_Toc196825906)

[Графики 18](#_Toc196825907)

[Интерфейс 19](#_Toc196825908)

[Заключение 20](#_Toc196825909)

[Список литературы 21](#_Toc196825910)

## Задание

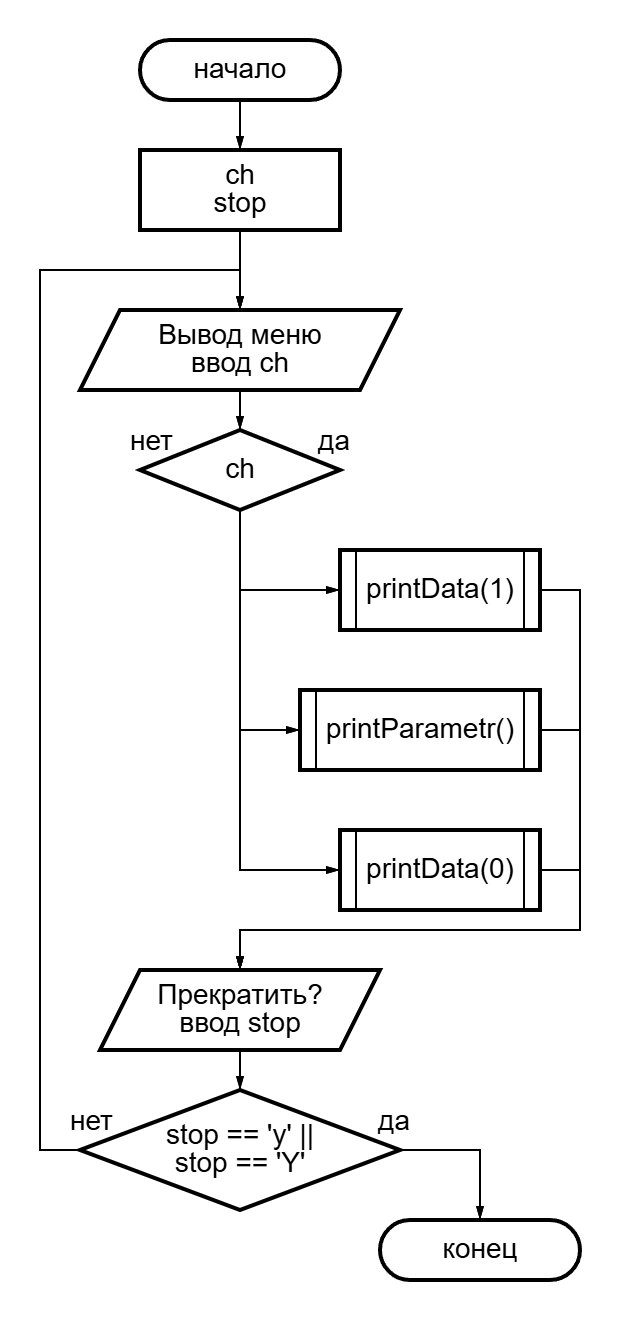
В курсовой работе необходимо для заданной электрической цепи по известному входному сигналу определить выходной сигнал для N равностоящих моментов времени, а затем определить некоторые его характеристики с погрешностью не более 1%.   
В ходе работы студентам необходимо:

* 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 или LibraOffice.Calc (электронные таблицы), построить графики зависимости напряжений входных и выходных сигналов от времени.
* Оформить пояснительную записку (doc-файл) по ГОСТ 19.402-78. ЕСПД. Описание программы. Плюс «Заключение» с личными выводами по работе.
* Объединить программу на Си и Wxmaxima (LibraOffice.Calc), вызов отчета с помощью скрипта на Bash.
* Защитить работу преподавателю.

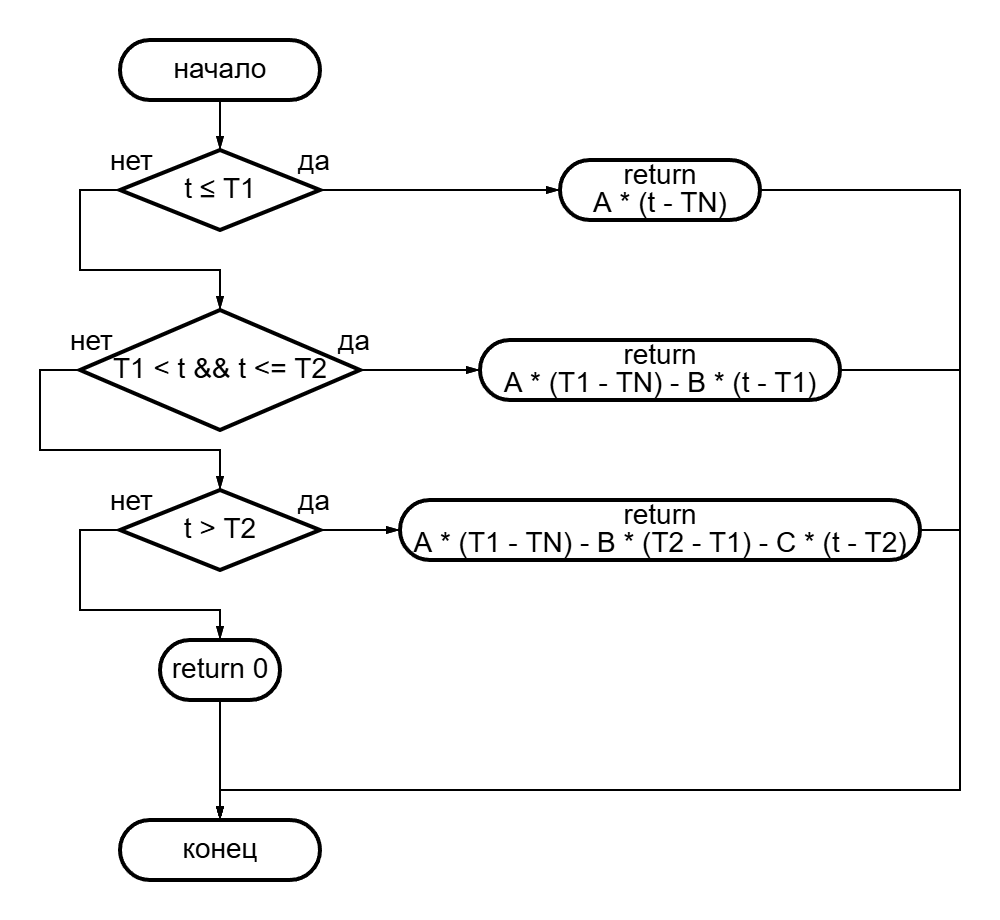
## Таблица идентификаторов

| **Идентификатор** | **Тип** | **Описание** |
| --- | --- | --- |
| TN | #define | Начальное время (5.0). |
| TK | #define | Конечное время (50.0). |
| T1 | #define | Первая граница времени (15.0), определяет изменение функции Uvx(t). |
| T2 | #define | Вторая граница времени (45.0), определяет изменение функции Uvx(t). |
| A, B, C | #define | Коэффициенты для расчета Uvx(t). |
| A1, B1 | #define | Коэффициенты для расчета Uvix(t). |
| Uvx(t) | double | Функция, вычисляющая входное напряжение в зависимости от времени t. |
| Uvix(t) | double | Функция, вычисляющая выходное напряжение в зависимости от Uvx(t). |
| getParam() | double | Функция, возвращающая параметр на основе заданных констант. |
| printData(print) | void | Функция для вывода данных в консоль и записи в файлы. |
| printParametr() | void | Функция для вычисления и вывода параметра с заданной точностью. |
| N | int | Количество точек для расчета (вводится пользователем). |
| DT | double | Шаг времени, вычисляется как (TK - TN) / (N - 1.0). |
| f1, f2, f3 | FILE\* | Указатели на файлы для записи данных (massiv\_t.txt, massiv\_Uvx.txt, massiv\_Uvix.txt). |
| ch | int | Переменная для выбора пункта меню. |
| stop | char | Переменная для проверки завершения программы (Y/N). |

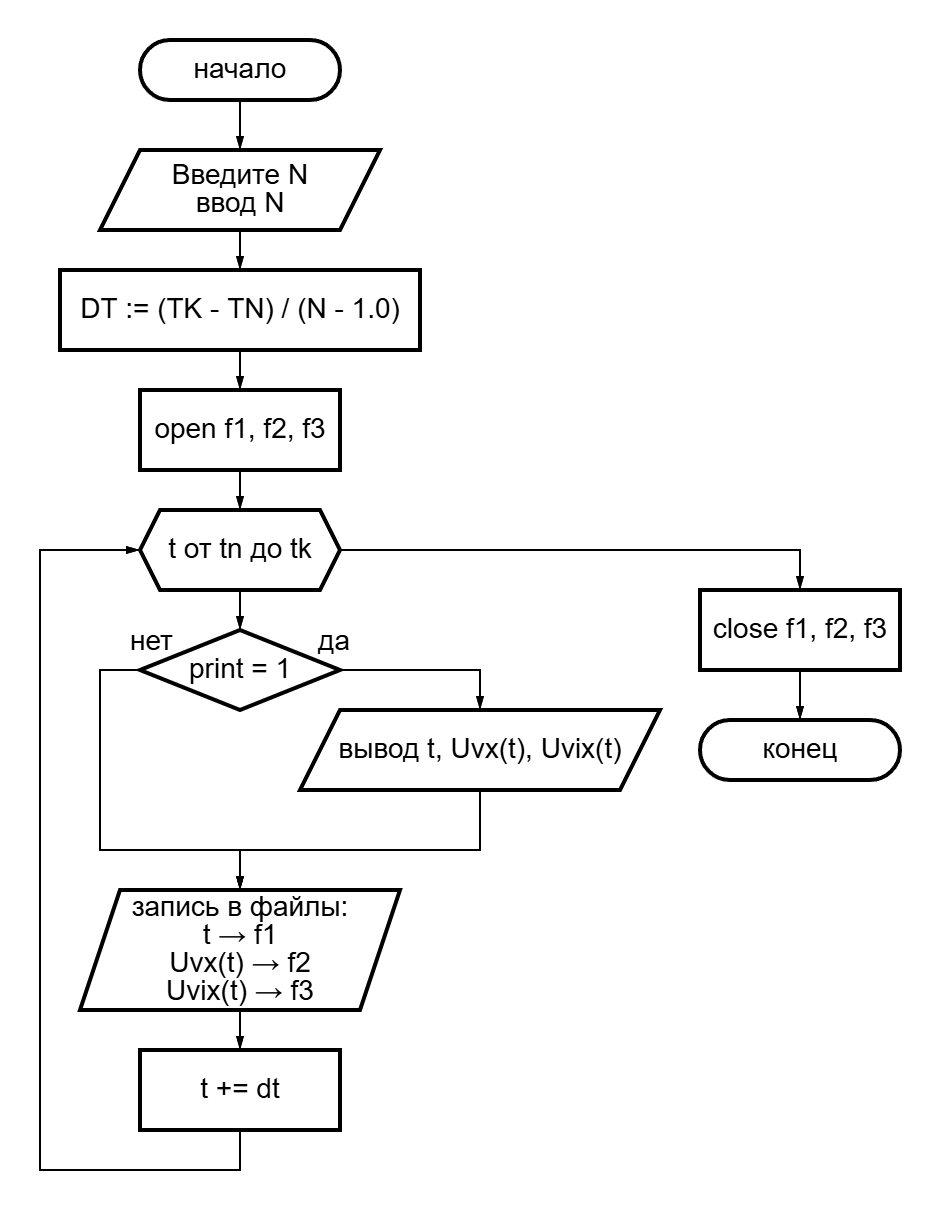
## Блок схемы



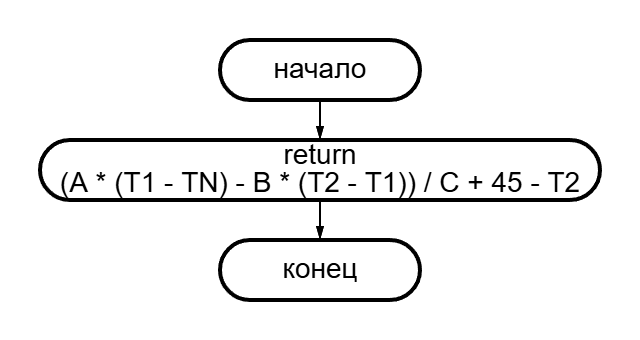
. main()



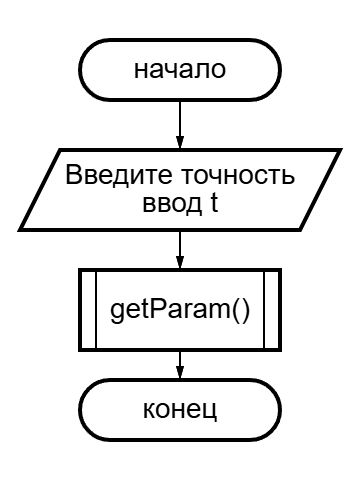
. Uvx(t)



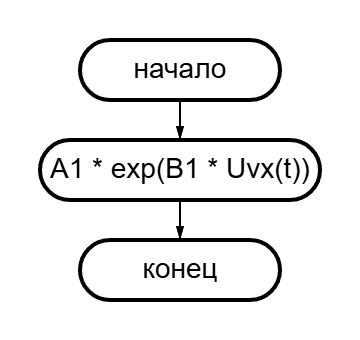
. PrintData(print)



.getParam(t)



. printParametr()



6. Uvix(t)

## Текст программы

/\* Код на C \*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define TN 5.0

#define TK 50.0

#define T1 15.0

#define T2 45.0

#define A 20.0

#define B 0.5

#define C 17.0

#define A1 5.0

#define B1 0.5

double Uvx(double t)

{

if (t <= T1)

{

return A \* (t - TN);

}

if (T1 < t && t <= T2)

{

return A \* (T1 - TN) - B \* (t - T1);

}

if (t > T2)

{

return A \* (T1 - TN) - B \* (T2 - T1) - C \* (t - T2);

}

printf("\tt = %F\tSomething went wrong...\n", t);

return 0;

}

double Uvix(double t)

{

return A1 \* exp(B1 \* Uvx(t));

}

double getParam()

{

return (A \* (T1 - TN) - B \* (T2 - T1)) / C + 45 - T2;

}

void printData(int print)

{

int N;

printf("Enter N: ");

scanf("%d", &N);

double DT = (TK - TN) / (N - 1.0);

FILE \*f1, \*f2, \*f3;

f1 = fopen("massiv\_t.txt", "w");

f2 = fopen("massiv\_Uvx.txt", "w");

f3 = fopen("massiv\_Uvix.txt", "w");

if (print == 1)

printf("%.\*s\n", 36, "------------------------------------");

for (double t = TN; t <= TK; t += DT)

{

if (t != TN && print == 1)

printf("%.\*s\n", 36, "|--------|----------|--------------|");

if (print == 1)

printf("| %4.1f | %6.2f | %10.4e |\n", t, Uvx(t), Uvix(t));

fprintf(f1, "%4.1f\n", t);

fprintf(f2, "%6.2f\n", Uvx(t));

fprintf(f3, "%10.4f\n", Uvix(t));

}

if (print == 1)

printf("%.\*s\n", 36, "------------------------------------");

fclose(f1);

fclose(f2);

fclose(f3);

}

void printParametr()

{

int t;

printf("Set the parametr's accuracy (from 0 to 15): ");

scanf("%d", &t);

printf("Parametr: %.\*f\n", t, getParam());

}

int main()

{

int ch;

char stop;

while (1 == 1)

{

printf("1 - control calculation for n points\n");

printf("2 - calculation of a parameter with a given accuracy\n");

printf("3 - writing data to a file\n");

scanf("%d", &ch);

switch(ch)

{

case 1:

printData(1);

break;

case 2:

printParametr();

break;

case 3:

printData(0);

printf("Done");

break;

default:

continue;

}

printf("\nTerminate?[Y/N] ");

getchar();

stop = getchar();

if (stop == 'y' || stop == 'Y')

{

break;

}

}

return 0;

}

/\* wxMaxima код \*/

fpprintprec:4$;

tn: 5$; tk: 50$;

t1: 15$; t2: 45$;

a: 20$; b: 0.5$; c: 17$;

a1: 5$; b1: 0.5$;

N: 51$; dt: (tk - tn) / (N - 1)$;

Uvx(t):=

if t < t1

then a \* (t - tn)

else if (t1 < t and t < t2)

then a \* (t1 - tn) - b \* (t - t1)

else a \* (t1 - tn) - b \* (t2 - t1) - c \* (t - t2)$;

Uvix(t):= a1 \* exp(b1 \* Uvx(t))$;

for t:tn thru tk step dt do display(Uvx(t), Uvix(t)), numer;

data\_t: read\_list("massiv\_t.txt");

data\_Uvx: read\_list("massiv\_Uvx.txt");

data\_Uvix: read\_list("massiv\_Uvix.txt");

wxplot2d([discrete, data\_t, data\_Uvx], [gnuplot\_preamble, "set grid;"]);

wxplot2d([Uvx(t)], [t, tn, tk], [gnuplot\_preamble, "set grid;"]);

wxplot2d([discrete, data\_t, data\_Uvix], [y, 0, 1.35 \* 10^44], [gnuplot\_preamble, "set grid;"]);

wxplot2d([Uvix(t)], [t, tn, tk], [y, 0, 1.35 \* 10^44], [gnuplot\_preamble, "set grid;"]);

# Bash-скрипт

#!/bin/bash

cat zastavka.txt;

./c.exe;

"D:\maxima-5.47.0\bin\wxmaxima.exe" --eval "kpm.mac";

start "" "Пояснительная записка.docx";

## Описание программы

### Общие сведения

Исполняемые файлы:

* c.exe (скомпилированная C-программа)
* run.sh/run.bat (Bash-скрипт для автоматизации)

Языки:

* C (основная программа)
* Bash (автоматизация вызова)
* Maxima (математические расчеты)

### Функциональное обозначение

Запуск производится через bash.sh для:

* + Запуска программы с интерактивным меню
  + Выполнения расчетов
  + Построения графиков в WxMaxima

### Логическая структура

Запуск bash.sh → Запуск c.exe, ожидание выхода из программы → Запуск wxMaxima и построение графиков, ожидание выхода из программы → Открытие Пояснительная записка.doc

### Используемые технические средства

Git Bash или WSL wxMaxima

### Вызов и загрузка

Для запуска программы необходимо из директории, в которой она хранится, вызвать bash.sh.

### Выходные данные

Автоматическое создание:

* massiv\_t.txt
* massiv\_Uvx.txt
* massiv\_Uvix.txt

## Графики

|  |  |  |
| --- | --- | --- |
|  | По программе на C | С помощью средств WxMaxima |
| Uvx(t) |  |  |
| Uvix(t) |  |  |

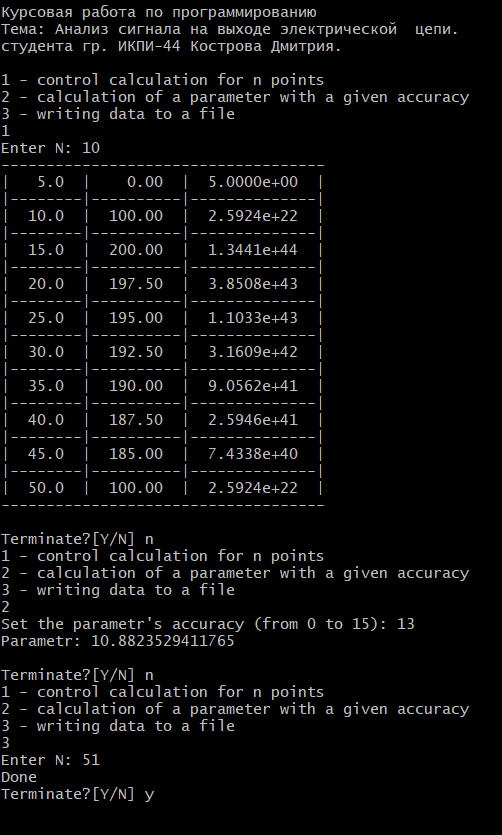
На основании проведенного анализа сигналов и построенных графиков можно сделать следующие выводы:  
Результаты, полученные при ручном расчете в среде WxMaxima, полностью совпадают с данными, вычисленными программой на языке C. Это подтверждает:

* + Корректность аналитических формул для Uvx(t) и Uvix(t)
  + Правильность алгоритмов, реализованных в программе
  + Точность численных методов

Графики входного (Uvx(t)) и выходного (Uvix(t)) сигналов демонстрируют ожидаемое поведение:

* + Выходной сигнал экспоненциально возрастает при увеличении входного напряжения
  + В точках t = 15.0 и t = 45.0 наблюдаются характерные изломы, соответствующие изменению режима работы цепи

## Интерфейс



## Заключение

В ходе выполнения курсовой работы были успешно решены поставленные задачи:

* Разработана программа на языке C для анализа сигналов в электрической цепи.
* Проведены контрольные расчеты в WxMaxima, подтвердившие корректность алгоритмов.
* Построены графики входного и выходного сигналов, наглядно демонстрирующие их зависимость от времени.
* Обеспечена запись результатов в файлы для последующей обработки.

Программа показала свою эффективность и точность расчетов, что позволяет рекомендовать ее для учебных целей. В дальнейшем функционал можно расширить, добавив поддержку других типов сигналов.

## Список литературы

1. ГОСТ 19.402-78. Описание программы.
2. Методические указания к курсовой работе «Анализ сигнала на выходе электрической цепи».
3. Керниган Б., Ритчи Д. «Язык программирования C».
4. Документация по WxMaxima.