ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича»

Факультет Инфокоммуникационных сетей и систем

Кафедра Программной инженерии и вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине:

**«Программирование»**

тема: Анализ сигнала на выходе электрической цепи

Передаточная характеристика – 9 вариант

Входной сигнал – 9 вариант

Выполнил студент:

Ильницкий И.Н., ИКПИ-15 \_\_\_\_\_\_\_\_\_\_\_\_\_

*(Ф.И.О., № группы) (подпись)*

Дата выполнения:

«30» апреля 2022г

Проверил:

Хазиев Н.Н \_\_\_\_\_\_\_\_\_\_\_\_\_

*(Ф.И.О.) (подпись)*

Санкт-Петербург

2022

Содержание

1. Введение
2. Постановка задачи
3. План работы
4. Распечатка контрольного расчета
5. Блок-схема функции main()
6. Текст программы
7. Графики (обработка полученных результатов)
8. Список используемой литературы

1.Введение

Работа посвящена решению задач машинного анализа электрических цепей.

Целью настоящей курсовой работы является получение навыков в разработке программного обеспечения на языке Си, а также анализ и решение поставленной задачи с обоснованием выбранных методов реализации программы.

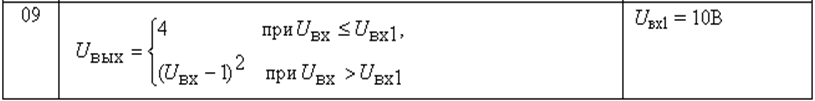
2.Постановка задачи

Имеется электрическая цепь, на вход которой подаётся определённый сигнал Uvx(t) . Необходимо создать программу, решающую задачу табулирования входного и выходного сигнала в «n» равностоящих промежутках времени, а так же вычисляющую момент времени, при котором входное напряжение Uvx(t) достигает максимума.

Напряжение входного сигнала описано аналитическим выражением



Передаточная характеристика описана аналитическим выражением



Нужно вывести формулу зависимости Uвых от t. Рассчитать значения Uвх и Uвых, записать значения в файл и построить графики зависимости Uвх и Uвых от t. Также необходимо рассчитать момент времени, при котором Uvx достигает максимума.

3.Контрольный расчет к курсовой работе

tn=0, tk=2, a1=1.5, a2=3, U=25, n=40

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | time | Uvx | Uvix | Uvx in wxMaxima | Uvix in wxMaxima |
| 1 | 0.00000 | 0.00000 | 4.00000 | Uvx(0)=0 | Uvix(0)=4 |
| 2 | 0.06897 | 2.21547 | 4.00000 | Uvx(0.0689655)=2.21547 | Uvix(0.0689655)=4 |
| 3 | 0.13793 | 3.79915 | 4.00000 | Uvx(0.137931)=3.79915 | Uvix(0.137931)=4 |
| 4 | 0.20690 | 4.89051 | 4.00000 | Uvx(0.206897)=4.89051 | Uvix(0.206897)=4 |
| 5 | 0.27586 | 5.60087 | 4.00000 | Uvx(0.275862)=5.60087 | Uvix(0.275862)=4 |
| 6 | 0.34483 | 6.01882 | 4.00000 | Uvx(0.344828)=6.01882 | Uvix(0.344828)=4 |
| 7 | 0.41379 | 6.21471 | 4.00000 | Uvx(0.413793)=6.21471 | Uvix(0.413793)=4 |
| 8 | 0.48276 | 6.24418 | 4.00000 | Uvx(0.482759)=6.24418 | Uvix(0.482759)=4 |
| 9 | 0.55172 | 6.15110 | 4.00000 | Uvx(0.551724)=6.1511 | Uvix(0.551724)=4 |
| 10 | 0.62069 | 5.96987 | 4.00000 | Uvx(0.62069)=5.96987 | Uvix(0.62069)=4 |
| 11 | 0.68966 | 5.72734 | 4.00000 | Uvx(0.689655)=5.72734 | Uvix(0.689655)=4 |
| 12 | 0.75862 | 5.44433 | 4.00000 | Uvx(0.758621)=5.44433 | Uvix(0.758621)=4 |
| 13 | 0.82759 | 5.13682 | 4.00000 | Uvx(0.827586)=5.13682 | Uvix(0.827586)=4 |
| 14 | 0.89655 | 4.81701 | 4.00000 | Uvx(0.896552)=4.81701 | Uvix(0.896552)=4 |
| 15 | 0.96552 | 4.49404 | 4.00000 | Uvx(0.965517)=4.49404 | Uvix(0.965517)=4 |
| 16 | 1.03448 | 4.17471 | 4.00000 | Uvx(1.03448)=4.17471 | Uvix(1.03448)=4 |
| 17 | 1.10345 | 3.86389 | 4.00000 | Uvx(1.10345)=3.86389 | Uvix(1.10345)=4 |
| 18 | 1.17241 | 3.56503 | 4.00000 | Uvx(1.17241)=3.56503 | Uvix(1.17241)=4 |
| 19 | 1.24138 | 3.28042 | 4.00000 | Uvx(1.24138)=3.28042 | Uvix(1.24138)=4 |
| 20 | 1.31034 | 3.01150 | 4.00000 | Uvx(1.31034)=3.0115 | Uvix(1.31034)=4 |
| 21 | 1.37931 | 2.75901 | 4.00000 | Uvx(1.37931)=2.75901 | Uvix(1.37931)=4 |
| 22 | 1.44828 | 2.52322 | 4.00000 | Uvx(1.44828)=2.52322 | Uvix(1.44828)=4 |
| 23 | 1.51724 | 2.30398 | 4.00000 | Uvx(1.51724)=2.30398 | Uvix(1.51724)=4 |
| 24 | 1.58621 | 2.10092 | 4.00000 | Uvx(1.58621)=2.10092 | Uvix(1.58621)=4 |
| 25 | 1.65517 | 1.91345 | 4.00000 | Uvx(1.65517)=1.91345 | Uvix(1.65517)=4 |
| 26 | 1.72414 | 1.74086 | 4.00000 | Uvx(1.72414)=1.74086 | Uvix(1.72414)=4 |
| 27 | 1.79310 | 1.58233 | 4.00000 | Uvx(1.7931)=1.58233 | Uvix(1.7931)=4 |
| 28 | 1.86207 | 1.43704 | 4.00000 | Uvx(1.86207)=1.43704 | Uvix(1.86207)=4 |
| 29 | 1.93103 | 1.30412 | 4.00000 | Uvx(1.93103)=1.30412 | Uvix(1.93103)=4 |
| 30 | 2.00000 | 1.18271 | 4.00000 | Uvx(2.0)=1.18271 | Uvix(2.0)=4 |

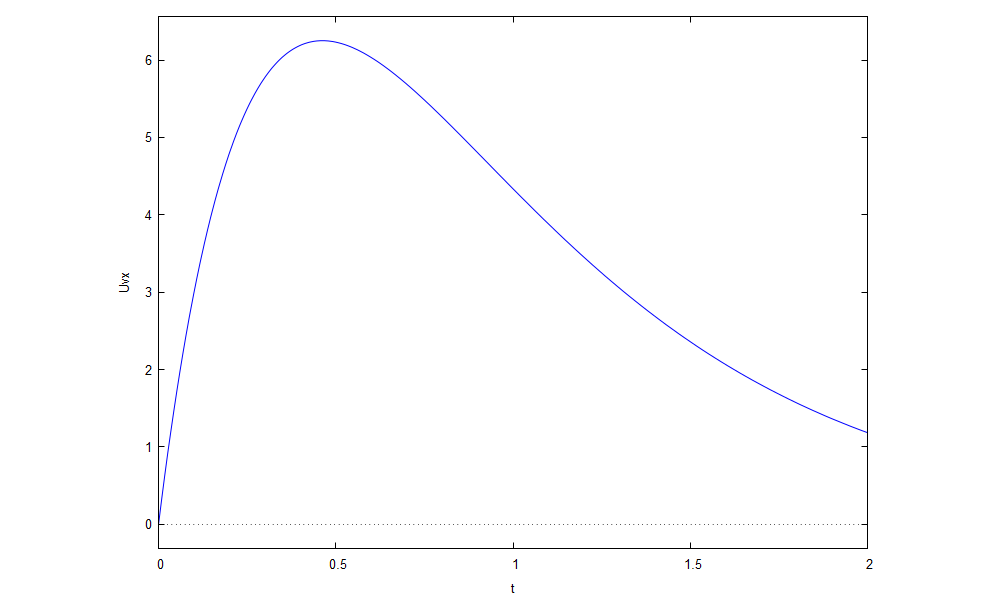
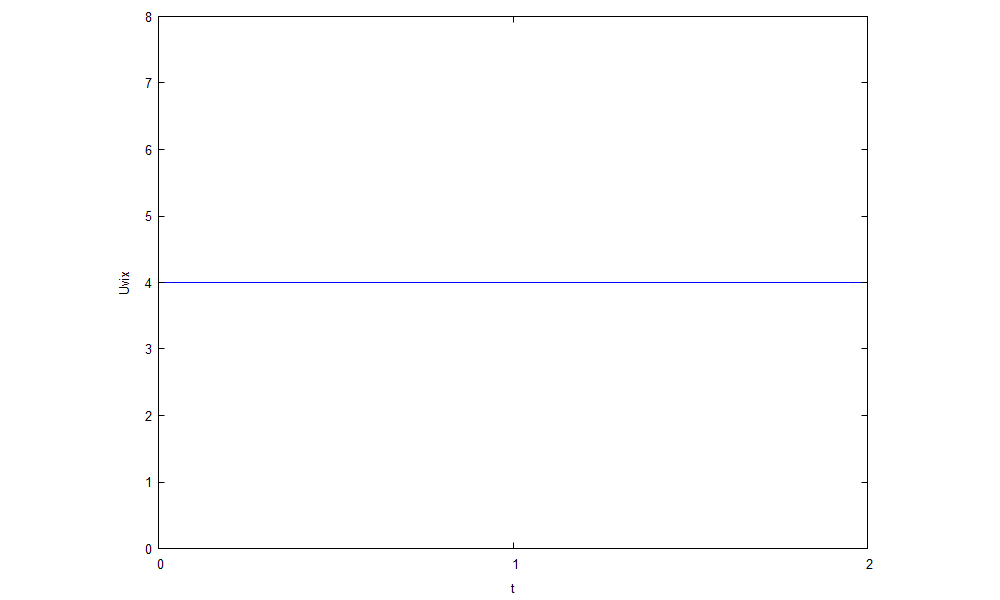
График зависимости Uvx(t) для рабочего набора:

График зависимости Uvx(t) для рабочего набора:



|  |  |  |
| --- | --- | --- |
| Обозначение в задаче | Идентификатор | Назначение |
| tнач | tn | начальный момент времени наблюдения входного напряжения |
| tкон | tk | конечный момент времени наблюдения входного напряжения |
| t | t[N] | массив времени |
| Uвх | Uvx[N] | массив входного напряжения |
| Uвых | Uvix[N] | массив выходного напряжения |
| - | dt | шаг между точками разбиения |
| U | U | константа входного сигнала |
| a1 | a1 | константа входного сигнала |
| a2 | a2 | константа входного сигнала |
| Uвх1 | Uvx1 | константа выходного сигнала |
| - | vrem | момент времени макс. значения вх.сигнала |

4.Таблица идентификаторов

5.Разработка структуры программы

Так как условия задачи подразумевают использование для ее решения математического пакета Wxmaxima, а также языка Си, то целесообразно будет использование bash-скрипта, на котором будет реализовано меню, позволяющее пользователю запускать как исполняемые файлы Wxmaxima, так и исполняемые Си-файлы.

Программа на языке Си разбита на функции, выполняющие свои задачи. Функции будут сгруппированы в модули, решающие схожие задачи. Возможны различные варианты организации структуры – программа может состоять из единственного файла, но гораздо целесообразнее разбить ее на модули.

Программа разделена на следующие составляющие модули: .c - основной модуль, содержащий в себе главную функцию программы - функцию main, в которой происходит вызов функций, выполняющих стоящие перед программой задачи, объявление переменных, а также реализация меню, которое делает доступ к использованию функций программы простым, быстрым и доступным обычному пользователю (далее клиент). Функция main разрабатывалась так, чтобы в ней содержалось как можно меньше кода. Это нужно для того, чтобы любой не знакомый с программой программист, открывший код данной программы мог легко и быстро в нем разобраться.

main.c - модуль, содержащий реализацию функции menu, которая выполняет две задачи: 1) выводит на консоль список команд, выполняемых программой (каждой команде соответствует определенный номер в списке, который клиент вводит с клавиатуры при работе с программой. 2) возвращает значение, введенное с клавиатуры клиентом. Далее это значение передается оператору switch, осуществляющему работу меню, то есть возможность запуска выбранной команды. Запуск выбранной команды осуществляется путем вызова соответствующей выбранной команде функции программы. Приведем список команд, доступных в меню:

«1» - Расчет параметра

«2» - Расчет параметра с заданной точностью

«3» - Запись данных t, Uvx, Uvix в файлы

Последняя команда «Запись данных t, Uvx, Uvix в файлы» сохраняет расчеты данных в три текстовых файла. Первый файл содержит столбец табуляции времени, второй файл столбец табуляции входного сигнала, третий файл - столбец выходного сигнала. Из сохраненных программой текстовых файлов, содержащих в себе лишь столбцы отдельных данных, с помощью программы wxMaxima можно построить график проводимых вычислений.

Func.h - этот модуль содержит в себе функции, отвечающие за ввод рабочих данных. Также данный модуль позволяет выводить расчетные данные на консоль и производить запись данных в текстовый файл, осуществлять решение поставленной перед программой задачи, заполнения массива времени, массивов входного и выходного напряжения, а так же расчета момента времени t, при котором входное напряжение Uvx достигает максимума.

6. Реализация программы

6.1 Описание меню bash-скрипта “script.sh”

Для удобства взаимодействия пользователя с программой имеется меню, в котором при выборе номера пункта выполняется одна из следующих функций:

* 1. Контрольный расчет в среде Wxmaxima

Открывается файл raschet.wxmx, выполняющий контрольный расчет для 30 точек с помощью среды Wxmaxima.

* 1. Расчет параметра на языке Си

Компилируется, а затем запускается исполняемый файл main, который выполняет контрольный расчет для введенного с клавиатуры числа точек, расчет параметра с заданной точностью и записывает получившиеся данные в файлы massiv\_t.txt, massiv\_Uvx.txt и

massiv\_Uvix.txt

* 1. Построение графиков в Wxmaxima

Открывается файл graph.wxmx, считывающий значения параметров t, Uvx и Uvix из файлов massiv\_t.txt, massiv\_Uvx.txt и

massiv\_Uvix.txt и строящий графики входного и выходного напряжения.

6.2 Описание меню Си-файла

1. Расчет параметра

Формируются массивы t, Uvx, Uvix для заданного числа точек и рассчитывается параметр (момент времени t, при котором входное напряжение Uvx достигает максимума

2. Расчет параметра с заданной точностью

Выполняет аналогичные пункту 1 действия и помимо этого вычисляет точность расчета

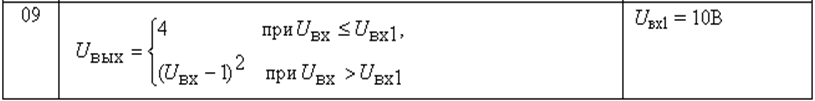
3. Формирует массивы времени, входного и выходного напряжения и записывает их в файлы massiv\_t.txt, massiv\_Uvx.txt и massiv\_Uvix.txt соответсвенно

5.3 Описание функций

6.3.1 void form\_t(int n, float \*t) – принимает на вход n - количество точек, а также указатель на массив времени t, объявленный вызывающей функции main. Расчет шага dt по формуле dt=(tk-tn)/(n-1). Создание массива времени Timeс tn по tk с шагом dt.

6.3.2 void form\_Uvx(int n,float \*t,float \*Uvx) – принимает на вход количество точек и указатели на массив времени t и на массив входного напряжения Uvx. Заполняется массив Uvx в соответствии с системой:

6.3.3 void form\_Uvix(int n,float \*Uvix,float \*Uvx) – принимает на вход число точек, указатели на массив времени t, на массив входного напряжения Uvx и на массив выходного напряжения Uvix. Формируется массив в соответствии с системой:



6.3.4 void print\_mas(int n,float \*t, float \*Uvx,float \*Uvix) - принимает на вход число точек, указатели на массив времени t, на массив входного напряжения Uvx и на массив выходного напряжения Uvix. Печатает таблицу зависимости сигналов входного (Uvx) и выходного (Uvix) напряжения от времени (t)

6.3.5 void write\_to\_file(n) – принимает на вход число точек n, формирует массивы t, Uvx, Uvix и записывает их в файлы massiv\_t.txt, massiv\_Uvx.txt и massiv\_Uvix.txt соответсвенно.

6.3.6 - float param(int n,float \*Uvx,float \*t,int pr) - принимает на вход число точек n, указатель на массив входного напряжения Uvx и на массив времени t и параметр pr. Выполняется расчет максимального значения Uvx, а затем вычисляется момент времени t, при котором Uvx принимает это максимальное значение. От параметра pr зависит, будет ли выведен результат на экран. После чего функция возвращает значение времени максимального значения Uvx

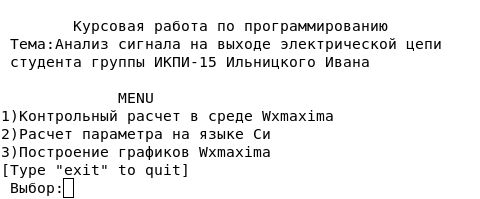
6.3.7 void param\_w\_accuracy(int n,float \*t,float \*Uvx ) - принимает на вход число точек n, указатели на массив времени t и на массив входного напряжения Uvx. Переменно ourParamValue присваивается значение функции param(n,Uvx,t,1). Переменной bigParamValue присваивается значение функции param для 500 точек. Рассчитывается погрешность времени, после чего на экран выводится максимальное значение Uvx, момент времени t и погрешность.

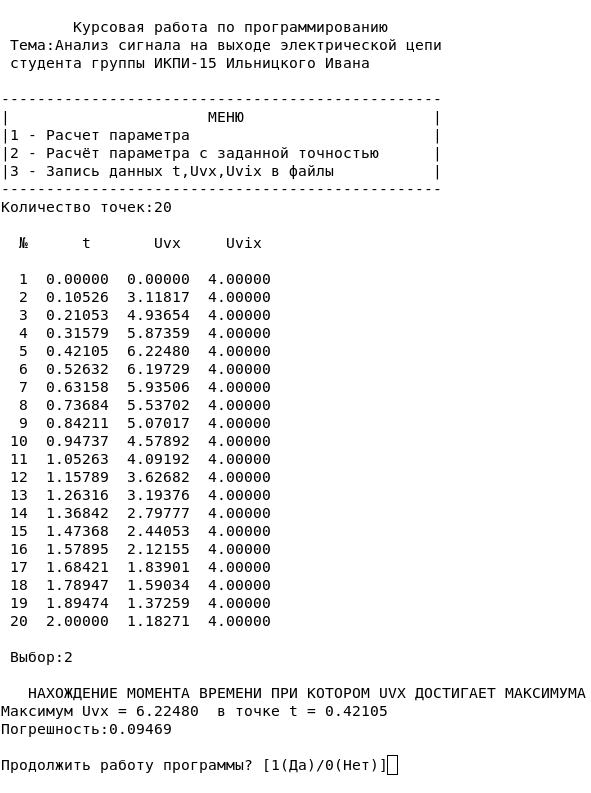
6.3.8 - void zastavka() – считывает из файла заставку к курсовой работе и выводит ее на экран.

6.3.9 Функция main() описана в модуле main.c. В функции реализован вызов функции ввода количества точек, формирования массивов t, Uvx, Uvix и вывода их на экран, а затем функций, соответствующих выбранному пункту меню, посредством переключателя switch, заключенный в цикл while (выводить меню пока не будет введено значение, равное 0 , по которому происходит выход из программы).

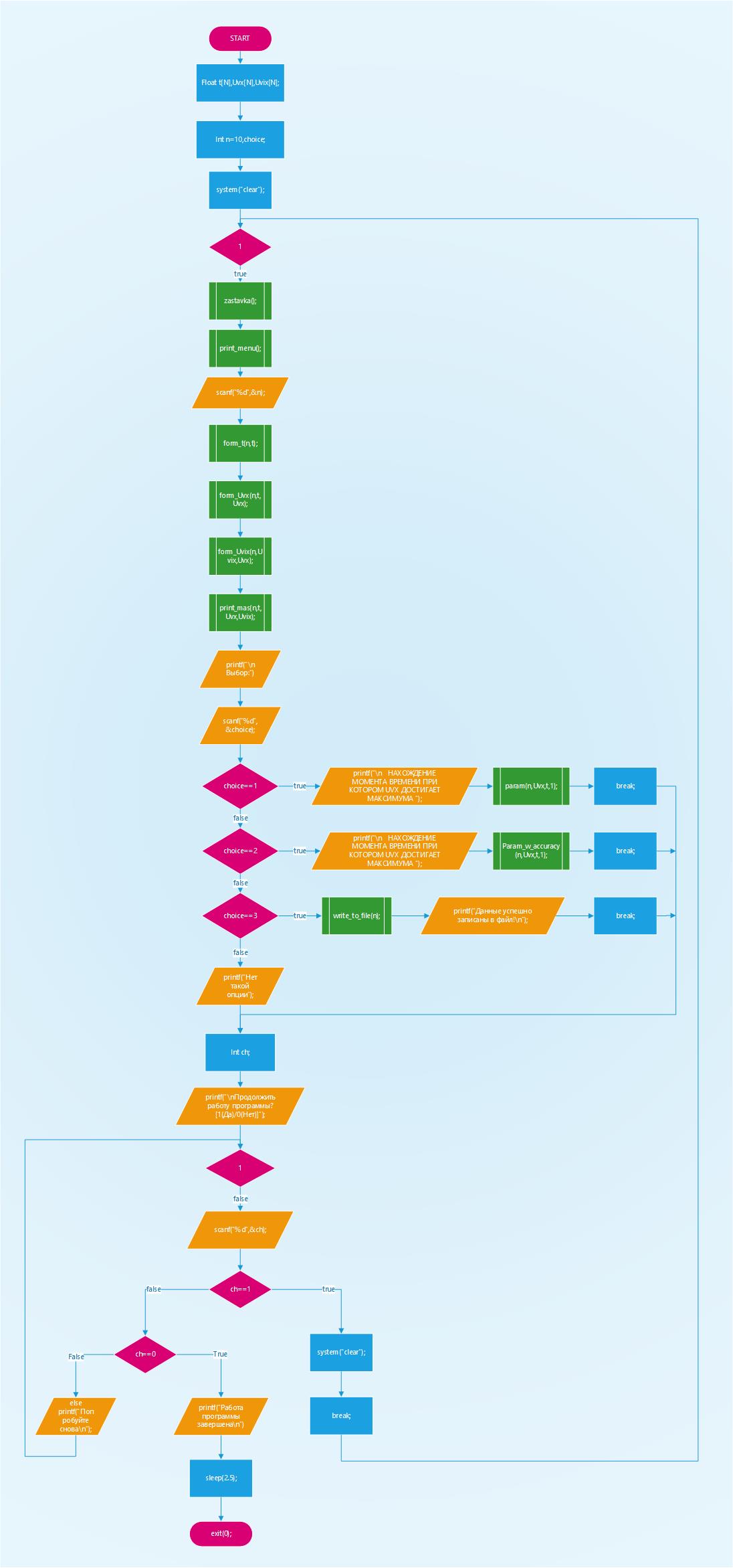
7.Интерфейс программы

BASH:

**

*C language:*

# 8. Блок-схема функции main



# 9.Текст программы

Func.h:

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#define N 1000

//Формирование массива t

void form\_t(int n, float \*t)

{

float tn=0,tk=2;

float dt=(tk-tn)/(n-1);

for (int i=0;i<n;i++)

{

t[i]=tn+i\*dt;

}

}

//Формирование массива Uvx

void form\_Uvx(int n,float \*t,float \*Uvx)

{

float a1=1.5,a2=3,U=25;

for (int i=0;i<n;i++)

{

Uvx[i]=U\*(exp(-a1\*t[i])-exp(-a2\*t[i]));

}

}

//Формирование массива Uvix

void form\_Uvix(int n,float \*Uvix,float \*Uvx)

{

float Uvx1=10;

for (int i=0;i<n;i++)

{

if (Uvx[i]<=Uvx1){

Uvix[i]=4;}

else {Uvix[i]=(Uvx[i]-1)\*(Uvx[i]-1);}

}

}

//Вывод таблицы t,Uvx,Uvix

void print\_mas(int n,float \*t, float \*Uvx,float \*Uvix)

{

printf("\n № t Uvx Uvix \n\n");

for (int i=0;i<n;i++)

{

printf("%3d %8.5f %8.5f %8.5f\n",i+1,t[i],Uvx[i],Uvix[i]);

}

}

//Запись данных в файлы

void write\_to\_file(int n)

{

//Формирование массивов для n точек и вывод в файл

float t[N],Uvx[N],Uvix[N];

//int n;printf("Количество точек:");scanf("%d",&n);

form\_t(n,t);

form\_Uvx(n,t,Uvx);

form\_Uvix(n,Uvix,Uvx);

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

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

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

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

for (int i=0;i<n;i++)

{

fprintf(f1,"%7.5f\n",t[i]);

fprintf(f2,"%7.5f\n",Uvx[i]);

fprintf(f3,"%7.5f\n",Uvix[i]);

}

fclose(f1);fclose(f2);fclose(f3);

}

//Расчет параметра 4

float param(int n,float \*Uvx,float \*t,int pr)

{

float max=0;

for (int i=0;i<n;i++)

{

if (Uvx[i]>max) {max=Uvx[i];}

}

float vrem;

for (int i=0;i<n;i++)

{

if (Uvx[i]==max) {vrem=t[i];break;}

}

if (pr){printf("\nМаксимум Uvx = %7.5f в точке t = %7.5f\n",max,vrem);}

return vrem;

}

//Расчет параметра с заданной точностью

void param\_w\_accuracy(int n,float \*t,float \*Uvx )

{

float ourParamValue,bigParamValue,pogreshnost;

ourParamValue=param(n,Uvx,t,1);

float t1[N],Uvx1[N],Uvix1[N];

int n1=500;

form\_t(n1,t1);

form\_Uvx(n1,t1,Uvx1);

form\_Uvix(n1,Uvix1,Uvx1);

bigParamValue=param(n1,Uvx1,t1,0);

pogreshnost=fabs(bigParamValue-ourParamValue)/ourParamValue;

printf("\rПогрешность:%.5f\n",pogreshnost);

}

//Чтение и вывод на экран заставки

void zastavka()

{

FILE \*zast = fopen("zastavka.txt","r");

char ch;

while (!feof(zast))

{

fscanf(zast,"%c",&ch);

printf("%c",ch);

}

fclose(zast);

}

//Вывод меню на экран

void print\_menu()

{

char str[]="-------------------------------------------------\n";

printf("%s| МЕНЮ |\n",str);

printf("|1 - Расчет параметра |\n");

printf("|2 - Расчёт параметра с заданной точностью |\n");

printf("|3 - Запись данных t,Uvx,Uvix в файлы |\n%s",str);

printf("Количество точек:");

}

Main.c:

#include "func.h"

#define N 1000

int main()

{

float t[N],Uvx[N],Uvix[N];

int n=10,choice;

system("clear");

while (1)

{

zastavka();

print\_menu();

scanf("%d",&n);

form\_t(n,t);

form\_Uvx(n,t,Uvx);

form\_Uvix(n,Uvix,Uvx);

print\_mas(n,t,Uvx,Uvix);

printf("\n Выбор:");scanf("%d",&choice);

switch (choice)

{

case 1:printf("\n НАХОЖДЕНИЕ МОМЕНТА ВРЕМЕНИ ПРИ КОТОРОМ UVX ДОСТИГАЕТ МАКСИМУМА ");param(n,Uvx,t,1);break;

case 2:printf("\n НАХОЖДЕНИЕ МОМЕНТА ВРЕМЕНИ ПРИ КОТОРОМ UVX ДОСТИГАЕТ МАКСИМУМА ");param\_w\_accuracy(n,t,Uvx);break;

case 3:write\_to\_file(n);printf("Данные успешно записаны в файл!\n");break;

default:printf("Нет такой опции\n");

}

int ch;

printf("\nПродолжить работу программы? [1(Да)/0(Нет)]");

while (1)

{

scanf("%d",&ch);

if (ch==1) {system("clear");break;}

else if (ch==0) {printf("Работа программы завершена\n");sleep(2.5);exit(0);}

else printf("Попробуйте снова\n");

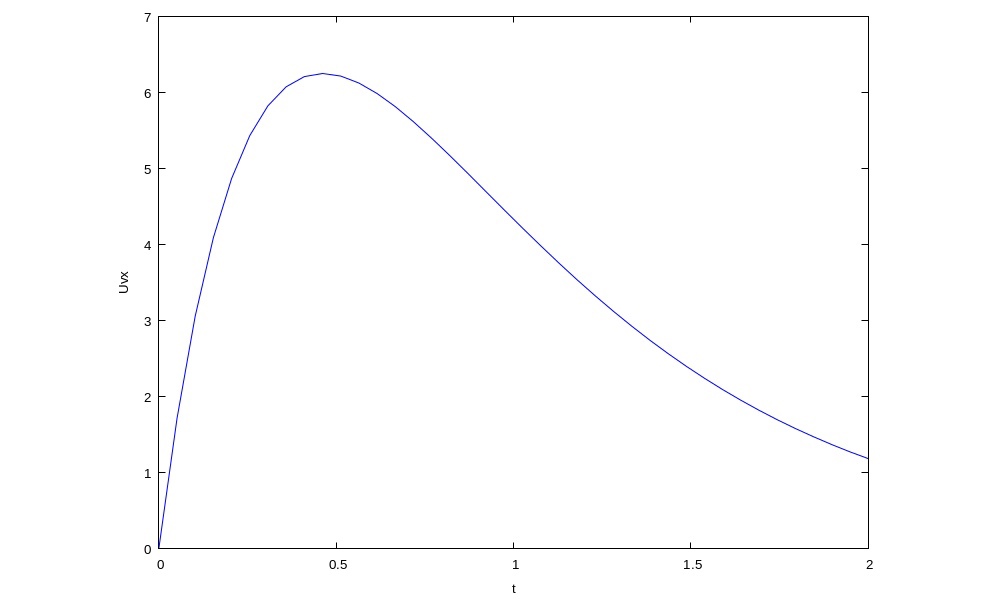
}

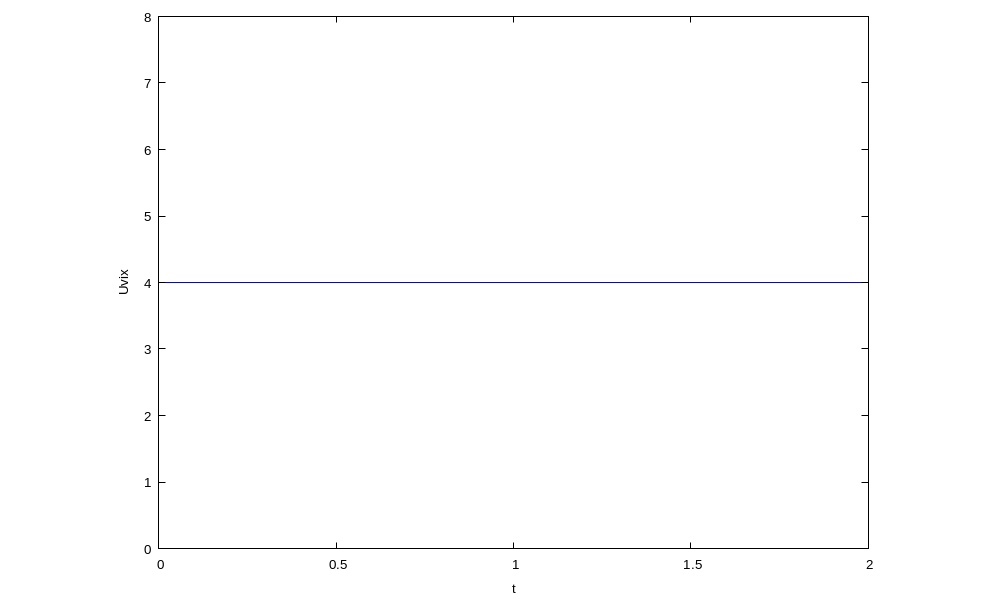
}

return 0;

}

# 10.Графики (обработка полученных результатов)

Uvx(t):

Uvix(t):

# 11.Заключение

По заданным начальным данным были получены следующие результаты:

• Расчет для n равностоящих точек значений времени t, напряжения входного сигнала Uвх(t) и напряжения выходного сигнала Uвых(t).

• Расчет параметров для Uвых(t).

• Сделана запись в файл.

• Построены графики зависимости напряжения входного и выходного сигнала в wxMaxima.

• Сделан контрольный расчёт в wxMaxima. Результаты вычислений в NetBeans сошлись с результатами в wxMaxima.

Поставленная задача успешно проанализирована и решена, получены навыки разработки программного обеспечения на языке C, а также навыки отладки и тестирования программы. Так как программа имеет модульную структуру, легко осуществить ее возможное расширение. Например, возможно добавить графический интерфейс переписав меню, но не затрагивая остального кода; также программа может быть адаптирована под решение более сложной задачи путем расширения соответствующих функций вычислений.

# 12.Использованная литература

1. Б. Керниган, Д. Ритчи – «Язык программирования С» 2011

2. С. В. Козин, Н. А. Матиясевич– «Практикум по программированию на языке Си. Часть 1» 2008

3. Л. Н. Бережной И.О. Воронцова Д.В. Окунева– «Информатика. Анализ сигнала на выходе электрической цепи. Методические указания к курсовой работе» 2012