Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Курский государственный университет

Кафедра Программного обеспечения

и администрирования

информационных систем

Отчёт по технологической

(проектно-технологической)

практике

Выполнил:

студент 113.1 группы Козявин М.С.

Проверил:

ст. преподаватель кафедры ПОиАИС Ураева Е.Е.

Курск, 2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc108370707)

[1 Задание на период практики 4](#_Toc108370708)

[2 Описание выполненной работы 4](#_Toc108370709)

[2.1 Лабораторная работа №1. Вычисление арифметических операций 10](#_Toc108370710)

[2.2 Лабораторная работа №2. Вектор-строки и вектор-столбцы 10](#_Toc108370711)

[2.3 Лабораторная работа №3. Матрицы 13](#_Toc108370712)

[2.4 Лабораторная работа №4. Графика и визуализация данных 15](#_Toc108370713)

[2.5 Лабораторная работа №5. Файл-функции и файл-программы, программирование 25](#_Toc108370714)

[2.6 Лабораторная работа №6. Работа со строками 30](#_Toc108370715)

[2.7 Программирование графики 31](#_Toc108370716)

[3 Рабочий график проведения практики 60](#_Toc108370717)

[Заключение 61](#_Toc108370718)

[Cписок использованных источников 62](#_Toc108370721)

# ВВЕДЕНИЕ

Целью технологической (проектно-технологической) практики в освоении приемов работы и программирования в среде Octave, получение навыков использования инструментов вычислений и визуализации результатов.

Octave является одновременно операционной средой   
и языком программирования. Его можно применять в различных сферах, таких как математика и вычисления, разработка алгоритмов, анализ данных, исследования и визуализация результатов, научная и инженерная графика   
и т.д. Octave актуален для студентов. Объединенный пакет программ может помочь при реализации поставленных перед учащимся задач.

Для достижения поставленной цели были поставлены следующие задачи:

− научиться записывать выражения на языке Octave и отображать результат их вычислений в различных форматах;

− изучить возможности Octave для создания векторов и выполнения операций над ними;

− изучить возможности Octave для создания матриц и выполнения операций над ними;

− научиться строить графики функций одной переменной,   
кусочно-заданной функции, функции, заданной параметрически, визуализировать функцию двух переменных, давать заголовки графическим окнам, размещать подписи к осям;

− научиться создавать и применять файл-функции в Octave;

– изучить особенности Octave при работе со строками;

− научиться создавать приложения на языке С++ для построения графиков функций.

# 1 Задание на период практики

# Лабораторная работа №1. Вычисление арифметических выражений

Занести в некоторые переменные значения выражений (1) при заданных   
x и y (например, x = -1.75\*10-3, y = 3.1π), отобразить результат в различных форматах и изучить информацию о переменных при помощи команды «whos».

**Лабораторная работа №2. Вектор-строки и вектор-столбцы**

*Задание 1.* Для заданных векторов a и b (2, 3) длины n:

1. Вычислить их сумму, разность и скалярное произведение;

2. Образовать вектор *c = [a1, a2, ..., an, b1, b2, ..., bn]*, определить его элементы вектора c в обратном порядке и записать результат в новый вектор;

3. Найти векторное произведение векторов *u = [a1, a3, a4]* и *v = [b2, b3, b4]*.

а = [0.6 5.4 -3.2 1.1 -2.4 5.8 9.7 3.5], (2)

b = [4.7 8.4 -2.0 4.4 -3.6 5.5 0.9 1.2], (3)

*Задание 2.* Вычислить значения функции (4) на отрезке (5) в заданном числе N (6) равномерно отстоящих друг от друга точек.

**Лабораторная работа №3. Матрицы**

*Задание 1.* Введите матрицы

,, *C =*  и найдите значение следующего выражения (7):

*Задание 2.* При помощи встроенных функций для заполнения стандартных матриц, индексации двоеточием и, возможно, поворотом, транспонированием или вычеркиванием получите следующую матрицу (8):

*Задание 3.* Вычислить значения функции (9) для всех элементов матрицы (10) и записать результат в матрицу того же размера, что и исходная.

*Задание 4.* Сконструировать блочные матрицы (11) (используя функции для заполнения стандартных матриц) и применить функции обработки данных   
и поэлементные операции для нахождения заданных величин (12).

**Лабораторная работа №4. Графика и визуализация данных**

*Задание 1*. Построить графики функций одной переменной на указанных интервалах (13, 14, 15, 16). Вывести графики различными способами:

* в отдельные графические окна;
* в одно окно на одни оси;
* в одно окно на отдельные оси.

Дать заголовки, разместить подписи к осям, легенду, использовать различные цвета, стили линий и типы маркеров, нанести сетку.

*Задание 2.* Построить график (17) функции в полярной системе координат (исп. функцию «polar»), подберите наиболее «интересный» диапазон значений для x.

*Задание 3.* Построить график кусочно-заданной функции (18), отобразить ветви разными цветами и маркерами.

Задание 4. Построить график параметрически заданной функции   
(19, 20), используя «plot» и «comet», используйте или подберите наиболее подходящий диапазон значений для t.

|  |  |
| --- | --- |
|  |  |

*Задание 5.* Визуализировать функцию двух переменных (21) на прямоугольной области определения различными способами:

* каркасной поверхностью;
* залитой цветом каркасной поверхностью;
* промаркированными линиями уровня (самостоятельно выбрать значения функции, отображаемые линиями уровня);
* освещенной поверхностью.

Расположить графики в отдельных графических окнах и в одном окне с соответствующим числом пар осей. Представить вид каркасной или освещенной поверхности с нескольких точек обзора. При необходимости, подберите более «интересные» диапазоны значений x и y.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Лабораторная работа №5. Файл-функции и файл-программы, программирование**

*Задание 1.* Написать файл-функции и построить графики на заданном отрезке при помощи «plot» (с шагом 0.05) и «fplot» для следующих функций (22):

|  |  |
| --- | --- |
|  |  |

*Задание 2.* Написать файл-функцию для решения поставленной задачи: «Написать файл-функцию для вычисления суммы отрицательных элементов матрицы, лежащих ниже главной диагонали.».

*Задание 3.* Написать файл-функцию для вычисления кусочно-заданной функции (см. индивидуальный вариант кусочно-заданной функции лабораторной №4).

*Задание 4.* Написать файл-функцию для решения поставленной задачи: «Имеются стол прямоугольной формы с размерами a×b (a и b — целые числа, a > b) и кости домино с размерами c×d×e (c, d и e — целые числа, c > d > e). Найти вариант размещения на столе наибольшего количества костей. Все размещаемые кости должны лежать на одной и той же грани в один ярус без свешивания со стола. Все ребра костей домино должны быть параллельны или перпендикулярны каждой стороне стола.».

**Лабораторная работа №6. Работа со строками**

Написать файл-функцию для решения поставленной задачи:  
«Дано предложение (слова разделяются пробелом). Во всех словах с нечетной длиной удалить средний символ.».

**Программирование графики**

Написать программу на языке С++, выполняющую построение графиков функций по точкам (путем отрисовки каждого элемента изображения в графическом окне при помощи функций графических примитивов). Недопустимо загружать в приложение готовые изображения. Для всех графиков: дать заголовки, построить оси, разместить подписи к осям, легенду, использовать различные цвета, стили линий.

Задание 1. Построить графики функций (23, 24) одной переменной на указанных интервалах. Вывести графики различными способами:

* в отдельные графические окна;
* в одно окно на одни оси;
* в одно окно на отдельные оси.

Задание 2. Построить график функции (25) в полярной системе координат, подберите наиболее «интересный» диапазон значений для x.

Задание 3. Построить график кусочно-заданной функции (26), отобразить ветви разными цветами.

Задание 4. Построить график параметрически заданной функции   
(27, 28), используйте или подберите наиболее подходящий диапазон значений для t.

**2 Описание выполненной работы**

## 2.1 Лабораторная работа №1. Вычисление арифметических операций

Текст программы

x = 1;

y = 2;

a = 1/(1+1/(1+1/x));

res = a + cos(y)^a;

format short

res

format long e

res

format hex

res

whos res

Тестирование программы

Тестирование задачи представлено на рисунке 1.

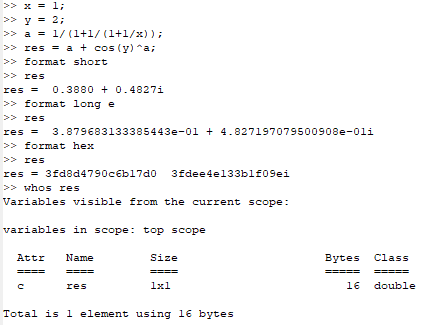


Рисунок 1 – Тестирование решения задачи лабораторной работы №1

## 2.2 Лабораторная работа №2. Вектор-строки и вектор-столбцы

Текст программы

Текст программы для решения задачи 1

a = [0.6 5.4 -3.2 1.1 -2.4 5.8 9.7 3.5];

b = [4.7 8.4 -2.0 4.4 -3.6 5.5 0.9 1.2];

a + b

a - b

dot(a, b)

c = [a, b];

ind = length(c):-1:1;

cr = [c(ind)]

cross([a(1), a(3), a(4)], b(2:4))

Текст программы для решения задачи 2

x = -4:1/3:4;

log(4-x.^2+e.^x)

Тестирование программы

Тестирование задачи 1 представлено на рисунке 2.

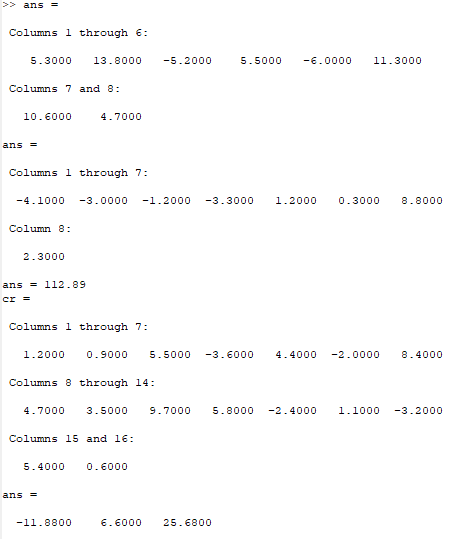


Рисунок 2 – Тестирование решения задачи 1 лабораторной работы №2

Тестирование задачи 2 представлено на рисунке 3.

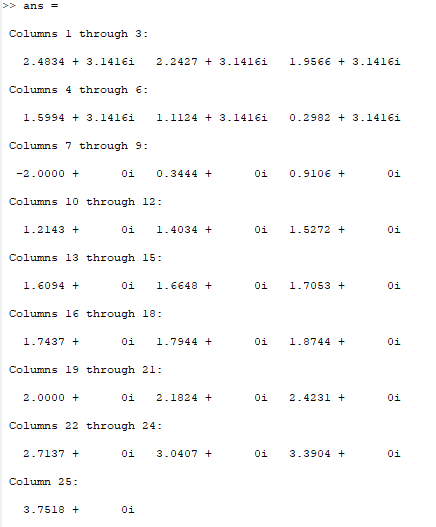


Рисунок 3 – Тестирование решения задачи 2 лабораторной работы №2

## 2.3 Лабораторная работа №3. Матрицы

Текст программы

Текст программы для решения задачи 1

A = [-9.8 4.4 1.3; -5.7 0.1 0.8; 2.4 4.4 8.6];

B = [1 0 2; 3 0 -1; 5 2 2; 8 9 0];

C = [0.1 0.2 -1.3 0.7; -0.2 0.3 2.2 0.8; 1.9 2.3 6.5 4.9];

(B'\*B-C\*C')\*(C'\*A)'

Текст программы для решения задачи 2

res = ones(6, 6)\*(-1) + diag(3:2:13) + diag(ones(1, 5)\*2, 1);

rot90(res)

Текст программы для решения задачи 3

x = [0.15 1.35 5.43 4.24; 0.01 3.37 5.81 9.09];

(1+x.^0.5)./(log(x.^2)-1)

Текст программы для решения задачи 4

A1 = rot90(diag(2:2:6));

A2 = ones(2, 3)\*3;

A3 = ones(5, 3)\*4;

A4 = ones(1, 6)\*(-5);

a = [[A1; A2] A3; A4];

max(sum(abs(a)))

Тестирование программы

Тестирование задачи 1 представлено на рисунке 4.

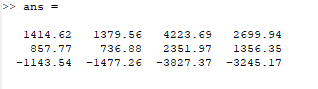


Рисунок 4 – Тестирование решения задачи 1 лабораторной работы №3

Тестирование задачи 2 представлено на рисунке 5.

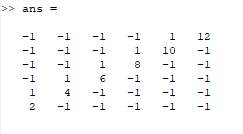


Рисунок 5 – Тестирование решения задачи 2 лабораторной работы №3

Тестирование задачи 3 представлено на рисунке 6.

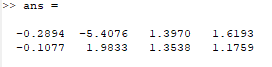


Рисунок 6 – Тестирование решения задачи 3 лабораторной работы №3

Тестирование задачи 4 представлено на рисунке 7.

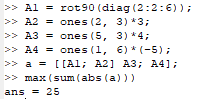


Рисунок 7 – Тестирование решения задачи 4 лабораторной работы №3

## 2.4 Лабораторная работа 4. Графика и визуализация данных

Текст программы

Текст программы для решения задачи 1

x1 = -pi:pi/30:pi;

x2 = -2\*pi:pi/10:2\*pi;

f = -x1.^2;

g = abs(sin(x1).\*cos(x1));

u = cos(x2./2);

v = sin(abs(x2));

figure

plot(x1, f, 'y:s')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 1')

legend('f(x)')

figure

plot(x1, g, 'r-\*')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 2')

legend('g(x)')

figure

plot(x2, u, 'b-.x')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 3')

legend('u(x)')

figure

plot(x2, v, 'g--+')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 4')

legend('v(x)')

figure

plot(x1, f, 'y:s')

hold on

plot(x1, g, 'r-\*')

plot(x2, u, 'b-.x')

plot(x2, v, 'g--+')

xlabel 'X'

ylabel 'Y'

title('Graphs 1-4')

legend('f(x)', 'g(x)', 'u(x)', 'v(x)')

hold off

subplot(2, 2, 1)

plot(x1, f, 'y:s')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 1')

legend('f(x)')

subplot(2, 2, 2)

plot(x1, g, 'r-\*')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 2')

legend('g(x)')

subplot(2, 2, 3)

plot(x2, u, 'b-.x')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 3')

legend('u(x)')

subplot(2, 2, 4)

plot(x2, v, 'g--+')

grid

xlabel 'X'

ylabel 'Y'

title('Graph 4')

legend('v(x)')

Текст программы для решения задачи 2

x = 0.1:0.01:0.9;

f = x.^exp(x);

polar(x, f)

Текст программы для решения задачи 3

x1 = -3:0.2:-2;

x2 = -2:0.2:1;

x3 = 1:0.2:3;

x4 = 3:0.2:4;

f1 = x1.\*0;;

f2 = (x2-1).^2;

f3 = cos(pi.\*x3./2);

f4 = 1-exp(3-x4);

figure

hold on

plot(x1, f1, 'r-\*')

plot(x2, f2, 'g--o')

plot(x3, f3, 'b-.^')

plot(x4, f4, 'm:d')

hold off

Текст программы для решения задачи 4

t = -1:0.1:1;

x = exp(-t).\*cos(t);

y = sin(t);

figure

plot(x, y);

figure

comet(x, y);

Текст программы для решения задачи 5

[x, y] = meshgrid(0:0.1:0.7, 0:0.1:0.7);

z = sqrt(1-x.^2-y.^2);

figure

mesh(x, y, z)

title("Mesh")

figure

surf(x, y, z)

title("Color")

figure

meshc(x, y, z)

title("Lines")

figure

surfl(x, y ,z)

shading interp

title("Light")

figure

subplot(2, 3, 1)

mesh(x, y, z)

title("Mesh")

subplot(2, 3, 2)

surf(x, y, z)

title("Color")

subplot(2, 3, 3)

meshc(x, y, z)

title("Lines")

subplot(2, 3, 4)

surfl(x, y ,z)

shading interp

title("Light")

subplot(2, 3, 5)

surfl(x, y ,z)

shading interp

title("Light rotate 1")

view(30, 60)

subplot(2, 3, 6)

surfl(x, y ,z)

shading interp

title("Light rotate 1")

view(90, 0)

Тестирование программы

Тестирование задачи 1 представлено на рисунках 8-10.

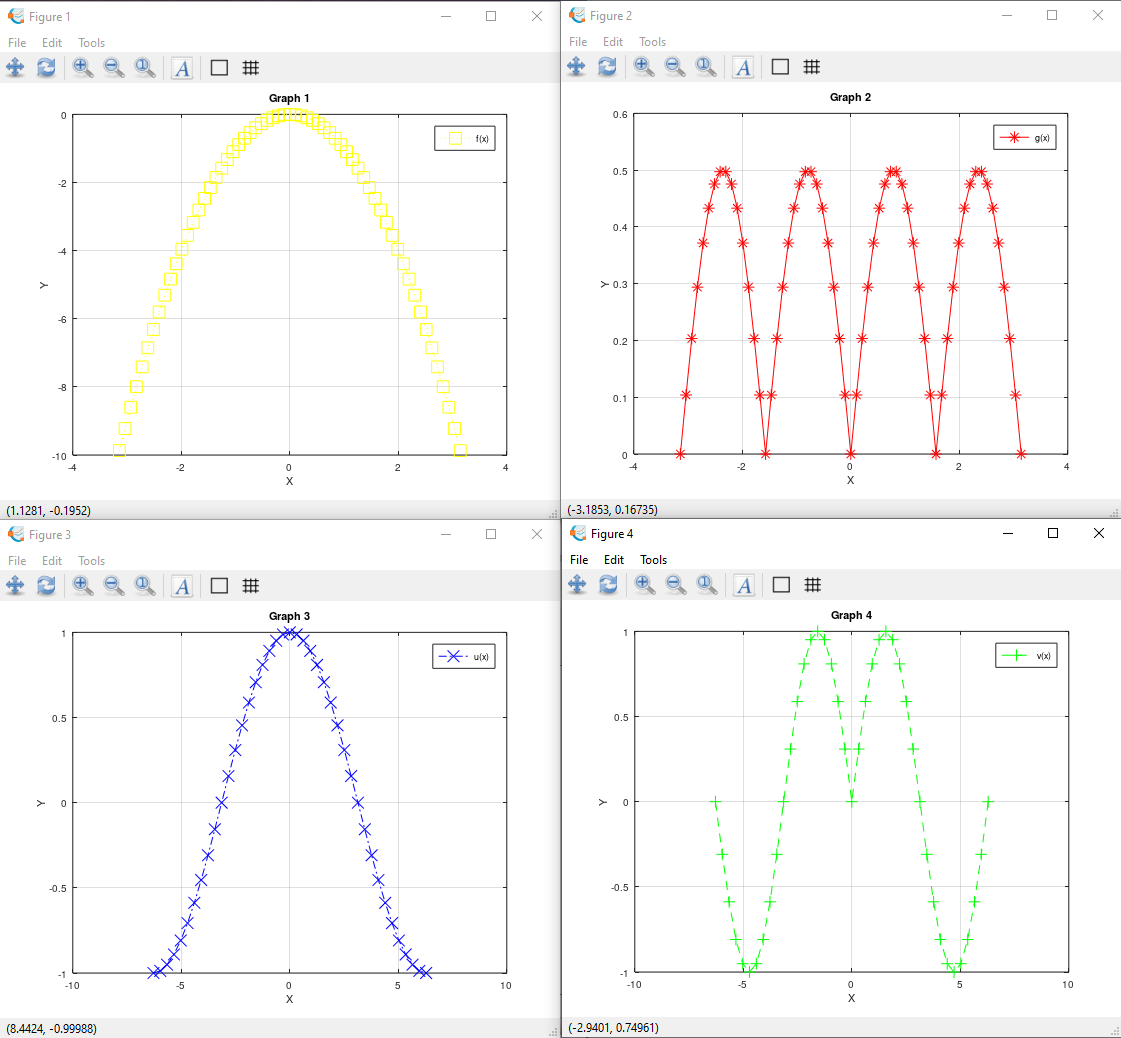


Рисунок 8 – Тестирование решения задачи 1 лабораторной работы №4

(разные окна)

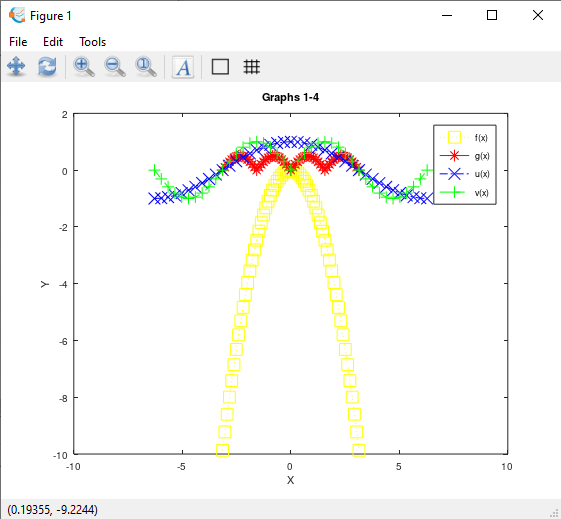


Рисунок 9 – Тестирование решения задачи 1 лабораторной работы №4

(одна ось)

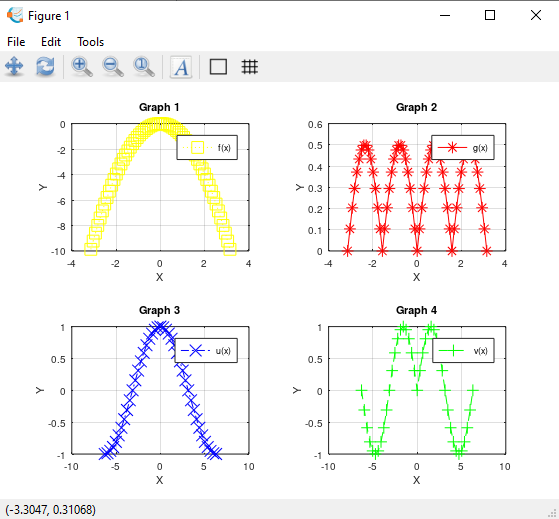


Рисунок 10 – Тестирование решения задачи 1 лабораторной работы №4

(отдельные оси)

Тестирование задачи 2 представлено на рисунке 11.

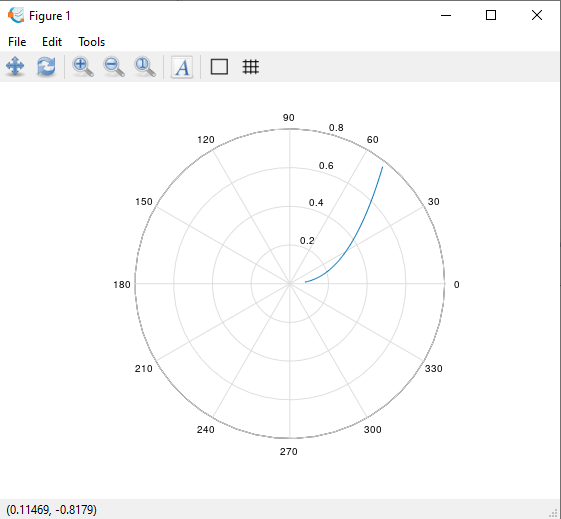


Рисунок 11 – Тестирование решения задачи 2 лабораторной работы №4

Тестирование задачи 3 представлено на рисунке 12.

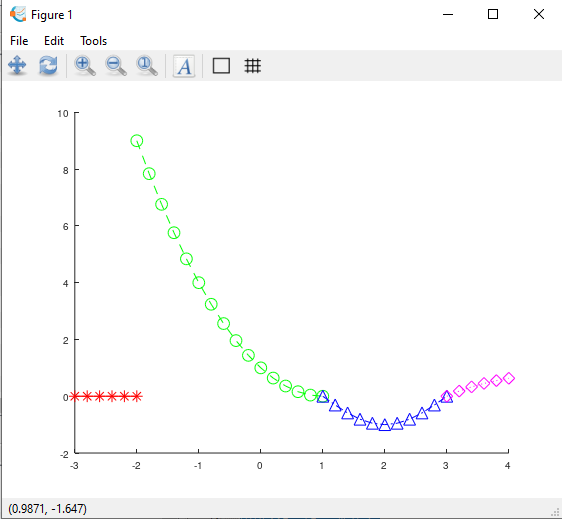


Рисунок 12 – Тестирование решения задачи 3 лабораторной работы №4

Тестирование задачи 4 представлено на рисунке 13.

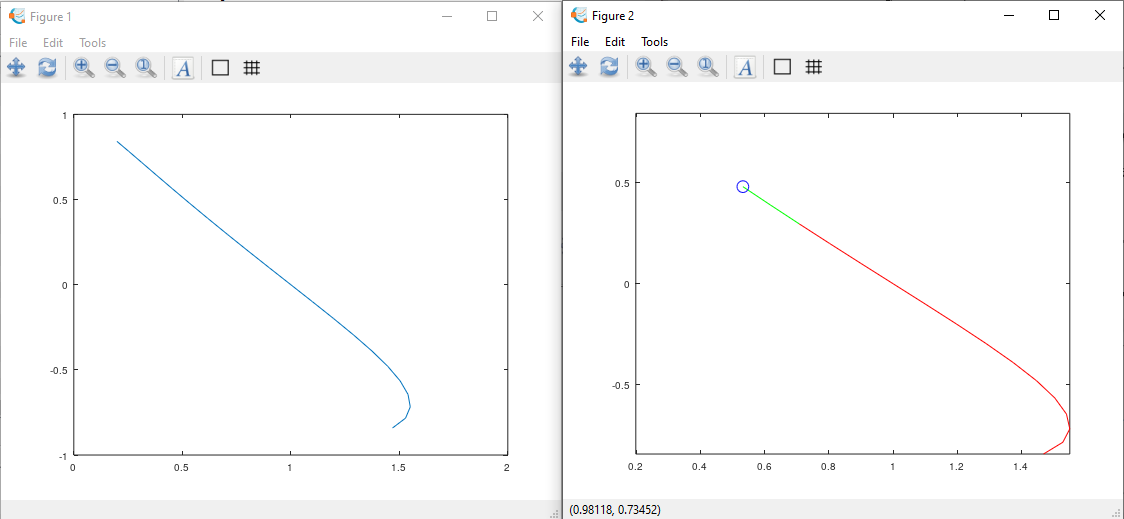


Рисунок 13 – Тестирование решения задачи 4 лабораторной работы №4

Тестирование задачи 5 представлено на рисунках 14-15.

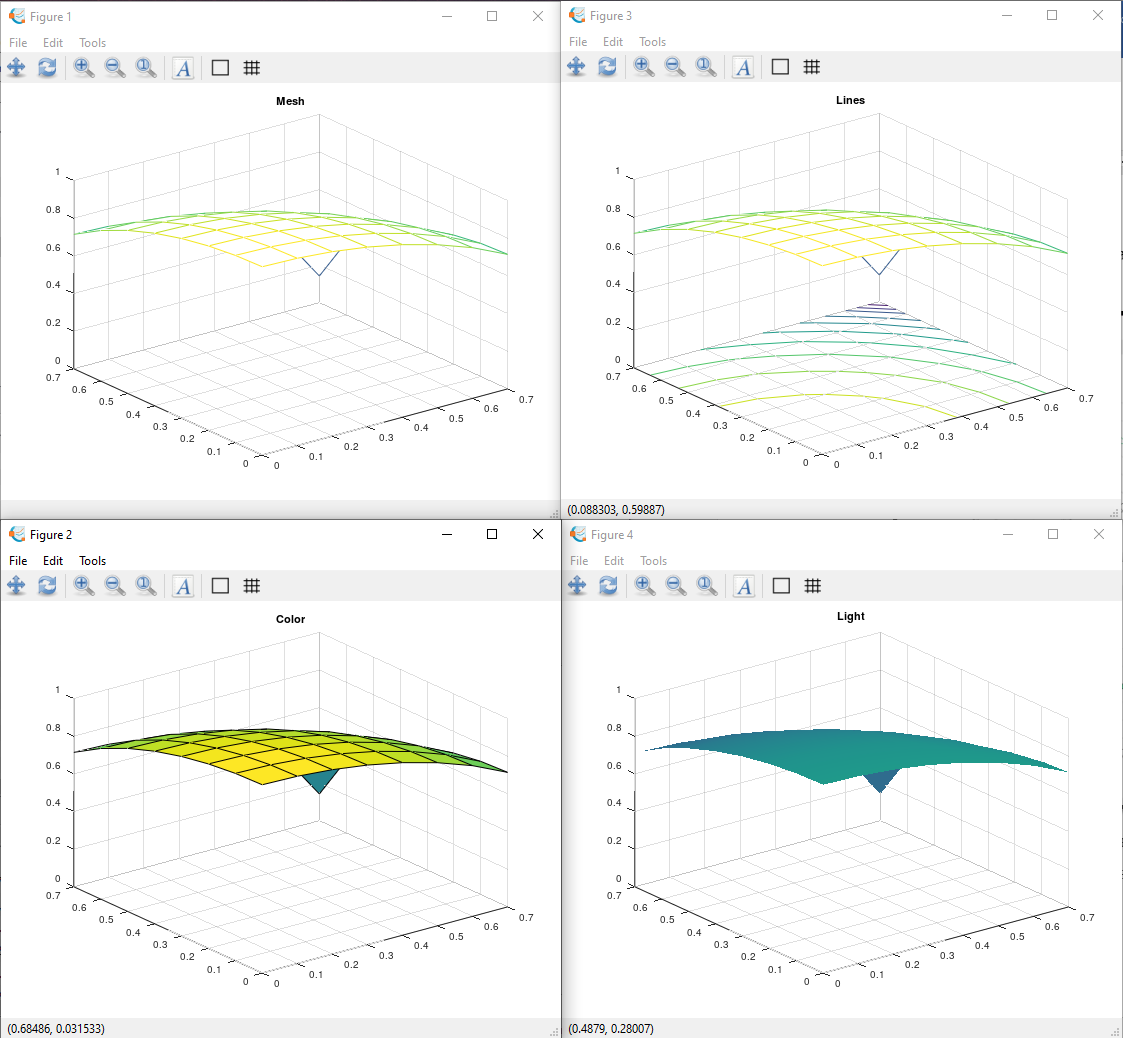


Рисунок 14 – Тестирование решения задачи 5 лабораторной работы №4  
(разные окна)

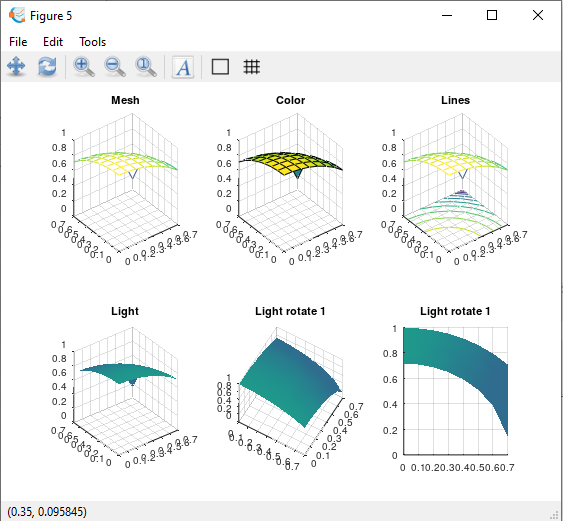


Рисунок 15 – Тестирование решения задачи 5 лабораторной работы №4

## 2.5 Лабораторная работа 5. Файл-функции и файл-программы, программирование

Текст программы

Текст программы для решения задачи 1

x = -1:0.05:1;

y = f1(x);

figure

plot(x, y);

figure

fplot('f1', [-1, 1])

Текст функции для решения задачи 1

function y = f1(x)

y = asin(x)+acos(exp(1-x));

Текст программы для решения задачи 2

x = [-1 -1 -5; 5 1 -1; -4 -9 0];

sum(x)

Текст функции для решения задачи 2

function res=sum(x)

res = 0;

[n, m] = size(x);

for i=1:n

for j=1:m

if j <= i && x(i, j) < 0

res += x(i, j);

endif

endfor

endfor

res;

end

Текст программы для решения задачи 3

x1 = -3:0.2:-2;

x2 = -2:0.2:1;

x3 = 1:0.2:3;

x4 = 3:0.2:4;

f1 = x1.\*0;

f2 = (x2-1).^2;

f3 = cos(pi.\*x3./2);

f4 = 1-exp(3-x4);

x = [x1, x2, x3, x4];

y = [f1, f2, f3, f4];

draw(x, y)

Текст функции для решения задачи 3

function draw(x, y)

if (length(x) == length(y))

plot(x, y);

endif

Текст программы для решения задачи 4

fill(9, 7, 6, 3, 2);

fill(9, 8, 6, 3, 2);

fill(2, 2, 6, 3, 2);

fill(10, 2, 6, 3, 2);

Текст функции для решения задачи 4

function [res, sm]=fill(a, b, c, d ,e)

sm = 0;

res = zeros(a, b);

arr = [c, d, e];

[n, ind] = min(arr);

arr(ind) = [];

m = min(arr);

k1 = 0;

k2 = 0;

if (n <= a && m <= b)

k1 = idivide(a, int32(n), "fix");

k2 = idivide(b, int32(m), "fix");

sm += k1\*k2;

res(1:n\*k1, 1:m\*k2) = 1;

endif

restB = b - m\*k2;

if (m <= a && n <= b)

begin = m\*k2+1;

k1 = idivide(restB, int32(n), "fix");

k2 = idivide(a, int32(m), "fix");

sm += k1\*k2;

res(1:m\*k2, begin:begin+n\*k1-1) = 1;

endif

res

sm

Тестирование программы

Тестирование задачи 1 представлено на рисунке 16.

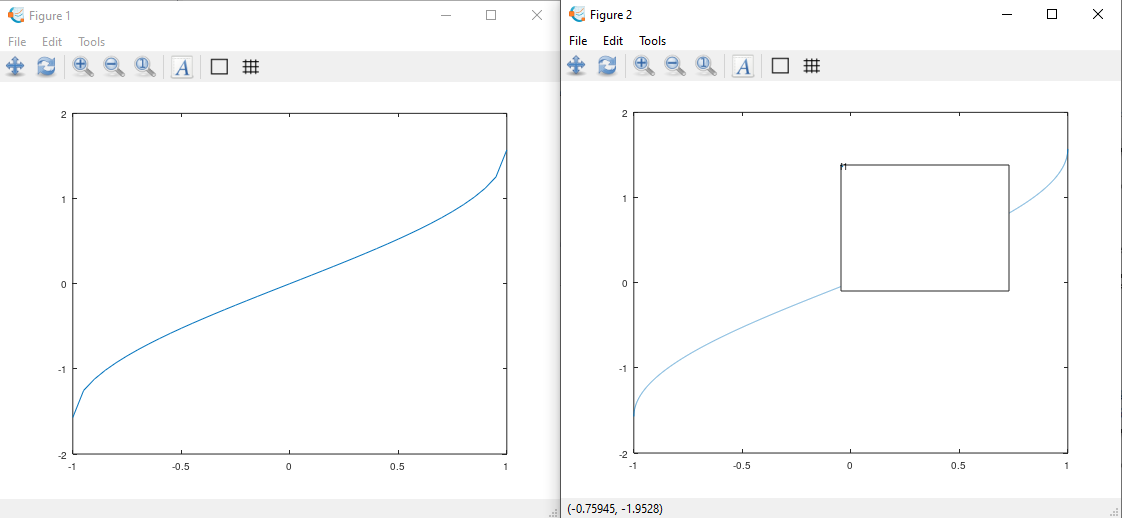


Рисунок 16 – Тестирование решения задачи 1 лабораторной работы №5

Тестирование задачи 2 представлено на рисунке 17.



Рисунок 17 – Тестирование решения задачи 2 лабораторной работы №5

Тестирование задачи 3 представлено на рисунке 18.

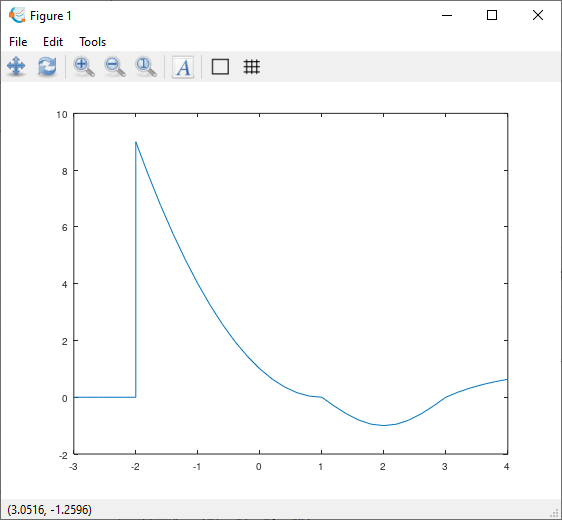


Рисунок 18 – Тестирование решения задачи 3 лабораторной работы №5

Тестирование задачи 4 представлено на рисунке 19.

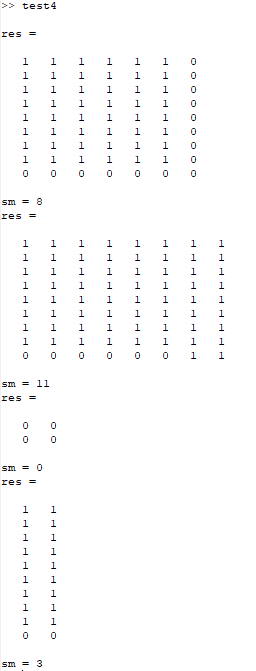


Рисунок 19 – Тестирование решения задачи 4 лабораторной работы №5

## 2.6 Лабораторная работа 6. Работа со строками

Текст программы

str = 'aabaa hjh hhhk ll lkl';

textFunc(str)

Текст функции для решения задачи

function res=textFunc(str)

ind = strfind(str, ' ');

ind(length(ind)+1) = length(str)+1;

last = 1;

for i=1:length(ind)

if rem(ind(i) - last, 2) ~= 0

delInd = last + idivide((ind(i) - last), int32(2), 'floor');

if (exist('res') == 0)

res = [str(last:delInd-1), str(delInd+1:ind(i)-1)];

else

res = [res, ' ', str(last:delInd-1), str(delInd+1:ind(i)-1)];

endif

else

if (exist('res') == 0)

res = str(last:ind(i)-1);

else

res = [res, ' ', str(last:ind(i)-1)];

endif

endif

last = ind(i)+1;

endfor

end

Тестирование программы

Тестирование задачи представлено на рисунке 20.



Рисунок 20 – Тестирование решения задачи лабораторной работы №5

## 2.7 Программирование графики

Текст программы для решения задачи 1

Заголовочные файлы

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

#include <oneaxis.h>

#include <ggraph.h>

#include <fgraph.h>

QT\_BEGIN\_NAMESPACE

namespace Ui { class Widget; }

QT\_END\_NAMESPACE

class Widget : public QWidget

{

Q\_OBJECT

public:

Widget(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

static void drawAxis(QPainter &canv, int centerX, int centerY, int width, int height, QString name, QStringList legends, QList<Qt::GlobalColor> colors);

static double f(double x);

static double g(double x);

~Widget();

private:

Ui::Widget \*ui;

};

#endif // WIDGET\_H

#ifndef ONEAXIS\_H

#define ONEAXIS\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

#include <widget.h>

namespace Ui {

class OneAxis;

}

class OneAxis : public QWidget

{

Q\_OBJECT

public:

explicit OneAxis(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

~OneAxis();

private:

Ui::OneAxis \*ui;

};

#endif // ONEAXIS\_H

#ifndef GGRAPH\_H

#define GGRAPH\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

#include <widget.h>

namespace Ui {

class GGraph;

}

class GGraph : public QWidget

{

Q\_OBJECT

public:

explicit GGraph(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

~GGraph();

private:

Ui::GGraph \*ui;

};

#endif // GGRAPH\_H

#ifndef FGRAPH\_H

#define FGRAPH\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

#include <widget.h>

namespace Ui {

class FGraph;

}

class FGraph : public QWidget

{

Q\_OBJECT

public:

explicit FGraph(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

~FGraph();

private:

Ui::FGraph \*ui;

};

#endif // FGRAPH\_H

Код программы

#include "widget.h"

#include "ui\_widget.h"

Widget::Widget(QWidget \*parent)

: QWidget(parent)

, ui(new Ui::Widget)

{

ui->setupUi(this);

OneAxis\* oneaxis = new OneAxis();

oneaxis->show();

FGraph\* fgraph = new FGraph();

GGraph\* ggraph = new GGraph();

fgraph->show();

ggraph->show();

}

Widget::~Widget()

{

delete ui;

}

void Widget::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int width = 300;

const int height = 300;

const double pi = 3.141592653;

int axisX1 = 350;

int axisY1 = 350;

int axisX2 = 1000;

int axisY2 = 350;

const int scale = 80;

const int pointsCount = 300;

const double step = pi\*2/ (pointsCount-1);

QPoint points1[pointsCount];

QPoint points2[pointsCount];

int i = 0;

for (double x = -pi; x < pi; x+=step, i++) {

double y1 = f(x);

double y2 = g(x);

points1[i] = QPoint(x\*scale+axisX1, axisY1-y1\*scale);

points2[i] = QPoint(x\*scale+axisX2, axisY2-y2\*scale);

}

drawAxis(canv, axisX1, axisY1, fmax(fmax(abs(axisX1-points1[pointsCount - 1].x()), abs(axisX1-points1[0].x())), width),

fmax(fmax(abs(axisY1-points1[pointsCount - 1].y()), abs(axisY1-points1[0].y())), height),

"График 1", QStringList({"f(x)"}), QList<Qt::GlobalColor>({Qt::red}));

drawAxis(canv, axisX2, axisY2, fmax(fmax(abs(axisX2-points2[pointsCount - 1].x()), abs(axisX2-points2[0].x())), width),

fmax(fmax(abs(axisY2-points2[pointsCount - 1].y()), abs(axisY2-points2[0].y())), height),

"График 2", QStringList({"g(x)"}), QList<Qt::GlobalColor>({Qt::blue}));

QPen pen;

pen.setColor(Qt::red);

pen.setWidth(2);

canv.setPen(pen);

canv.drawPolyline(points1, pointsCount);

pen.setColor(Qt::blue);

canv.setPen(pen);

canv.drawPolyline(points2, pointsCount);

canv.end();

}

void Widget::drawAxis(QPainter &canv, int centerX, int centerY, int width, int height, QString name, QStringList legends, QList<Qt::GlobalColor> colors) {

QPen pen;

pen.setColor(Qt::black);

canv.setPen(pen);

canv.drawLine(centerX-width, centerY, centerX+width, centerY);

canv.drawLine(centerX, centerY-height, centerX, centerY+height);

QFont font;

font.setFamily("Segoe UI");

font.setPixelSize(20);

canv.setFont(font);

canv.drawText(centerX-20, centerY+20, "0");

canv.drawText(centerX-20, centerY-height+20, "Y");

canv.drawText(centerX-width+20, centerY+20, "X");

font.setPixelSize(14);

canv.drawText(centerX-300, centerY-300, name);

for (int count = 1; count <= legends.size(); count++) {

pen.setColor(colors.at(count-1));

canv.setPen(pen);

canv.drawText(centerX-300, centerY-300+(18\*count), legends.at(count-1));

}

}

double Widget::f(double x) {

return -x\*x;

}

double Widget::g(double x) {

return abs(sin(x)\*cos(x));

}

#include "oneaxis.h"

#include "ui\_oneaxis.h"

OneAxis::OneAxis(QWidget \*parent) :

QWidget(parent),

ui(new Ui::OneAxis)

{

ui->setupUi(this);

}

OneAxis::~OneAxis()

{

delete ui;

}

void OneAxis::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int width = 300;

const int height = 300;

const double pi = 3.141592653;

int axisX = 350;

int axisY = 350;

const int scale = 80;

const int pointsCount = 300;

const double step = pi\*2/ (pointsCount-1);

QPoint points1[pointsCount];

QPoint points2[pointsCount];

int i = 0;

for (double x = -pi; x < pi; x+=step, i++) {

double y1 = Widget::f(x);

double y2 = Widget::g(x);

points1[i] = QPoint(x\*scale+axisX, axisY-y1\*scale);

points2[i] = QPoint(x\*scale+axisX, axisY-y2\*scale);

}

Widget::drawAxis(canv, axisX, axisY, fmax(fmax(abs(axisX-points1[pointsCount - 1].x()), abs(axisX-points1[0].x())), width),

fmax(fmax(abs(axisY-points1[pointsCount - 1].y()), abs(axisY-points1[0].y())), height),

"Одна ось", QStringList({"f(x)", "g(x)"}), QList<Qt::GlobalColor>({Qt::red, Qt::blue}));

QPen pen;

pen.setColor(Qt::red);

pen.setWidth(2);

canv.setPen(pen);

canv.drawPolyline(points1, pointsCount);

pen.setColor(Qt::blue);

canv.setPen(pen);

canv.drawPolyline(points2, pointsCount);

canv.end();

}

#include "ggraph.h"

#include "ui\_ggraph.h"

GGraph::GGraph(QWidget \*parent) :

QWidget(parent),

ui(new Ui::GGraph)

{

ui->setupUi(this);

}

GGraph::~GGraph()

{

delete ui;

}

void GGraph::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int width = 300;

const int height = 300;

const double pi = 3.141592653;

int axisX = 350;

int axisY = 350;

const int scale = 80;

const int pointsCount = 300;

const double step = pi\*2/ (pointsCount-1);

QPoint points2[pointsCount];

int i = 0;

for (double x = -pi; x < pi; x+=step, i++) {

double y2 = Widget::g(x);

points2[i] = QPoint(x\*scale+axisX, axisY-y2\*scale);

}

Widget::drawAxis(canv, axisX, axisY, fmax(fmax(abs(axisX-points2[pointsCount - 1].x()), abs(axisX-points2[0].x())), width),

fmax(fmax(abs(axisY-points2[pointsCount - 1].y()), abs(axisY-points2[0].y())), height),

"График G", QStringList({"g(x)"}), QList<Qt::GlobalColor>({Qt::blue}));

QPen pen;

pen.setWidth(2);

pen.setColor(Qt::blue);

canv.setPen(pen);

canv.drawPolyline(points2, pointsCount);

canv.end();

}

#include "fgraph.h"

#include "ui\_fgraph.h"

FGraph::FGraph(QWidget \*parent) :

QWidget(parent),

ui(new Ui::FGraph)

{

ui->setupUi(this);

}

FGraph::~FGraph()

{

delete ui;

}

void FGraph::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int width = 300;

const int height = 300;

const double pi = 3.141592653;

int axisX = 350;

int axisY = 350;

const int scale = 80;

const int pointsCount = 300;

const double step = pi\*2/ (pointsCount-1);

QPoint points1[pointsCount];

int i = 0;

for (double x = -pi; x < pi; x+=step, i++) {

double y1 = Widget::f(x);

points1[i] = QPoint(x\*scale+axisX, axisY-y1\*scale);

}

Widget::drawAxis(canv, axisX, axisY, fmax(fmax(abs(axisX-points1[pointsCount - 1].x()), abs(axisX-points1[0].x())), width),

fmax(fmax(abs(axisY-points1[pointsCount - 1].y()), abs(axisY-points1[0].y())), height),

"График F", QStringList({"f(x)"}), QList<Qt::GlobalColor>({Qt::red}));

QPen pen;

pen.setColor(Qt::red);

pen.setWidth(2);

canv.setPen(pen);

canv.drawPolyline(points1, pointsCount);

canv.end();

}

Текст программы для решения задачи 2

Заголовочные файлы

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

QT\_BEGIN\_NAMESPACE

namespace Ui { class Widget; }

QT\_END\_NAMESPACE

class Widget : public QWidget

{

Q\_OBJECT

public:

Widget(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

void drawAxis(QPainter &canv, int centerX, int centerY, int size, double maxR);

double f(double x);

double g(double x);

double u(double x);

double y(double x);

~Widget();

private:

Ui::Widget \*ui;

};

#endif // WIDGET\_H

Код программы

#include "widget.h"

#include "ui\_widget.h"

#include <QDebug>

Widget::Widget(QWidget \*parent)

: QWidget(parent)

, ui(new Ui::Widget)

{

ui->setupUi(this);

}

Widget::~Widget()

{

delete ui;

}

void Widget::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int size = 400;

int axisX = 450;

int axisY = 450;

double scale = 200.0;

const int pointsCount = 100;

const double left = 0.1;

const double right = 0.9;

const double step = (right-left)/(pointsCount-1);

const double offset = 0.001;

QPoint points1[pointsCount];

QPointF rawPoints[pointsCount];

int i = 0;

double maxR = 0;

for (double a = left; a <= right+offset; a+=step, i++) {

double r = f(a);

if (r > maxR) {

maxR = r;

}

double x = cos(a)\*r;

double y = sin(a)\*r;

qInfo() << a << r << x << y << x\*scale+axisX << axisY-y\*scale << i;

rawPoints[i] = QPointF(x, y); //QPoint(x\*scale+axisX, axisY-y\*scale);

}

qInfo() << maxR;

scale = size/maxR;

qInfo() << scale;

for (int n = 0; n < pointsCount; n++) {

points1[n] = QPoint(rawPoints[n].x()\*scale+axisX, axisY-rawPoints[n].y()\*scale);

qInfo() << rawPoints[n].x()\*scale+axisX << axisY-rawPoints[n].y()\*scale;

}

drawAxis(canv, axisX, axisY, size, maxR);

QPen pen;

pen.setColor(Qt::red);

pen.setWidth(2);

canv.setPen(pen);

canv.drawPolyline(points1, pointsCount);

canv.end();

}

void Widget::drawAxis(QPainter &canv, int centerX, int centerY, int size, double maxR) {

double coef = sqrt(2)/2;

QPen pen;

pen.setColor(Qt::black);

canv.setPen(pen);

canv.drawLine(centerX-size, centerY, centerX+size, centerY);

canv.drawLine(centerX, centerY-size, centerX, centerY+size);

canv.drawLine(centerX-size\*coef, centerY+size\*coef, centerX+size\*coef, centerY-size\*coef);

canv.drawLine(centerX-size\*coef, centerY-size\*coef, centerX+size\*coef, centerY+size\*coef);

canv.drawEllipse(centerX-size, centerY-size, size\*2, size\*2);

canv.drawEllipse(centerX-size\*0.75, centerY-size\*0.75, size\*1.5, size\*1.5);

canv.drawEllipse(centerX-size/2.0, centerY-size/2.0, size, size);

canv.drawEllipse(centerX-size/4.0, centerY-size/4.0, size/2.0, size/2.0);

QFont font;

font.setFamily("Helvetica");

font.setPixelSize(10);

canv.setFont(font);

canv.drawText(centerX+5, centerY-size-2, QString::number(maxR));

canv.drawText(centerX+5, centerY-size\*0.75-2, QString::number(maxR\*0.75));

canv.drawText(centerX+5, centerY-size\*0.5-2, QString::number(maxR\*0.5));

canv.drawText(centerX+5, centerY-size\*0.25-2, QString::number(maxR\*0.25));

canv.drawText(centerX+5, centerY-2, "0");

font.setPixelSize(14);

canv.drawText(centerX+size+20, centerY+3, "0°");

canv.drawText(centerX-8, centerY-size-20, "90°");

canv.drawText(centerX-size-20, centerY+3, "180°");

canv.drawText(centerX-10, centerY+size+20, "270°");

}

double Widget::f(double x) {

return pow(x, exp(x));

}

Текст программы для решения задачи 3

Заголовочные файлы

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

QT\_BEGIN\_NAMESPACE

namespace Ui { class Widget; }

QT\_END\_NAMESPACE

class Widget : public QWidget

{

Q\_OBJECT

public:

Widget(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

static void drawAxis(QPainter &canv, int centerX, int centerY, int width, int height, QString name, QStringList legends, QList<Qt::GlobalColor> colors);

double f(double x);

double g(double x);

double u(double x);

double v(double x);

const double pi = 3.141592653;

~Widget();

private:

Ui::Widget \*ui;

};

#endif // WIDGET\_H

Код программы

#include "widget.h"

#include "ui\_widget.h"

Widget::Widget(QWidget \*parent)

: QWidget(parent)

, ui(new Ui::Widget)

{

ui->setupUi(this);

}

Widget::~Widget()

{

delete ui;

}

void Widget::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int width = 300;

const int height = 300;

int axisX = 350;

int axisY = 350;

const int scale = 30;

const int pointsCount = 11;

QPoint points1[pointsCount];

QPoint points2[pointsCount];

QPoint points3[pointsCount];

QPoint points4[pointsCount];

int i = 0;

for (double x = -3; x < -1.9; x+=0.1, i++) {

double y1 = f(x);

points1[i] = QPoint(x\*scale+axisX, axisY-y1\*scale);

}

i = 0;

for (double x = -2; x < 1.1; x+=0.3, i++) {

double y2 = g(x);

points2[i] = QPoint(x\*scale+axisX, axisY-y2\*scale);

}

i = 0;

for (double x = 1; x < 3.1; x+=0.2, i++) {

double y3 = u(x);

points3[i] = QPoint(x\*scale+axisX, axisY-y3\*scale);

}

i = 0;

for (double x = 3; x < 4.1; x+=0.1, i++) {

double y4 = v(x);

points4[i] = QPoint(x\*scale+axisX, axisY-y4\*scale);

}

drawAxis(canv, axisX, axisY, fmax(fmax(abs(axisX-points1[pointsCount - 1].x()), abs(axisX-points1[0].x())), width),

fmax(fmax(abs(axisY-points1[pointsCount - 1].y()), abs(axisY-points1[0].y())), height),

"Кусочно заданная функция", QStringList({"f(x)", "g(x)", "u(x)", "v(x)"}), QList<Qt::GlobalColor>({Qt::red, Qt::blue, Qt::green, Qt::darkYellow}));

QPen pen;

pen.setColor(Qt::red);

pen.setWidth(3);

pen.setStyle(Qt::DotLine);

canv.setPen(pen);

canv.drawPolyline(points1, pointsCount);

pen.setColor(Qt::blue);

pen.setStyle(Qt::SolidLine);

canv.setPen(pen);

canv.drawPolyline(points2, pointsCount);

pen.setColor(Qt::green);

pen.setStyle(Qt::DashDotLine);

canv.setPen(pen);

canv.drawPolyline(points3, pointsCount);

pen.setColor(Qt::darkYellow);

pen.setStyle(Qt::DashLine);

canv.setPen(pen);

canv.drawPolyline(points4, pointsCount);

canv.end();

}

void Widget::drawAxis(QPainter &canv, int centerX, int centerY, int width, int height, QString name, QStringList legends, QList<Qt::GlobalColor> colors) {

QPen pen;

pen.setColor(Qt::black);

canv.setPen(pen);

canv.drawLine(centerX-width, centerY, centerX+width, centerY);

canv.drawLine(centerX, centerY-height, centerX, centerY+height);

QFont font;

font.setFamily("Segoe UI");

font.setPixelSize(20);

canv.setFont(font);

canv.drawText(centerX-20, centerY+20, "0");

canv.drawText(centerX-20, centerY-height+20, "Y");

canv.drawText(centerX-width+20, centerY+20, "X");

font.setPixelSize(14);

canv.drawText(centerX-300, centerY-300, name);

for (int count = 1; count <= legends.size(); count++) {

pen.setColor(colors.at(count-1));

canv.setPen(pen);

canv.drawText(centerX-300, centerY-300+(18\*count), legends.at(count-1));

}

}

double Widget::f(double x) {

return 0;

}

double Widget::g(double x) {

return (x-1)\*(x-1);

}

double Widget::u(double x) {

return cos(pi/2)\*x;

}

double Widget::v(double x) {

return 1-exp(3-x);

}

Текст программы для решения задачи 4

Заголовочные файлы

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include <QPainter>

#include <cmath>

QT\_BEGIN\_NAMESPACE

namespace Ui { class Widget; }

QT\_END\_NAMESPACE

class Widget : public QWidget

{

Q\_OBJECT

public:

Widget(QWidget \*parent = nullptr);

void paintEvent(QPaintEvent \*event);

static void drawAxis(QPainter &canv, int centerX, int centerY, int width, int height, QString name, QStringList legends, QList<Qt::GlobalColor> colors);

double f(double x);

double g(double x);

~Widget();

private:

Ui::Widget \*ui;

};

#endif // WIDGET\_H

Код программы

#include "widget.h"

#include "ui\_widget.h"

Widget::Widget(QWidget \*parent)

: QWidget(parent)

, ui(new Ui::Widget)

{

ui->setupUi(this);

}

Widget::~Widget()

{

delete ui;

}

void Widget::paintEvent(QPaintEvent \*event) {

QPainter canv(this);

const int width = 300;

const int height = 300;

int axisX = 350;

int axisY = 350;

const int scale = 100;

const int pointsCount = 80;

const double step = 2.0/pointsCount;

QPoint points1[pointsCount];

int i = 0;

for (double t = -1; t < 1; t+=step, i++) {

double x = f(t);

double y = g(t);

points1[i] = QPoint(x\*scale+axisX, axisY-y\*scale);

}

drawAxis(canv, axisX, axisY, fmax(fmax(abs(axisX-points1[pointsCount - 1].x()), abs(axisX-points1[0].x())), width),

fmax(fmax(abs(axisY-points1[pointsCount - 1].y()), abs(axisY-points1[0].y())), height),

"Параметрически заданная функция", QStringList({"f(x)"}), QList<Qt::GlobalColor>({Qt::red}));

QPen pen;

pen.setColor(Qt::red);

pen.setWidth(2);

canv.setPen(pen);

canv.drawPolyline(points1, pointsCount);

canv.end();

}

void Widget::drawAxis(QPainter &canv, int centerX, int centerY, int width, int height, QString name, QStringList legends, QList<Qt::GlobalColor> colors) {

QPen pen;

pen.setColor(Qt::black);

canv.setPen(pen);

canv.drawLine(centerX-width, centerY, centerX+width, centerY);

canv.drawLine(centerX, centerY-height, centerX, centerY+height);

QFont font;

font.setFamily("Segoe UI");

font.setPixelSize(20);

canv.setFont(font);

canv.drawText(centerX-20, centerY+20, "0");

canv.drawText(centerX-20, centerY-height+20, "Y");

canv.drawText(centerX-width+20, centerY+20, "X");

font.setPixelSize(14);

canv.drawText(centerX-300, centerY-300, name);

for (int count = 1; count <= legends.size(); count++) {

pen.setColor(colors.at(count-1));

canv.setPen(pen);

canv.drawText(centerX-300, centerY-300+(18\*count), legends.at(count-1));

}

}

double Widget::f(double t) {

return exp(-t)\*cos(t);

}

double Widget::g(double t) {

return sin(t);

}

Тестирование программы

Тестирование задачи 1 представлено на рисунках 21 – 24. Тестирование задачи 2 представлено на рисунке 25. Тестирование задачи 3 представлено на рисунке 26. Тестирование задачи 4 представлено на рисунке 27.

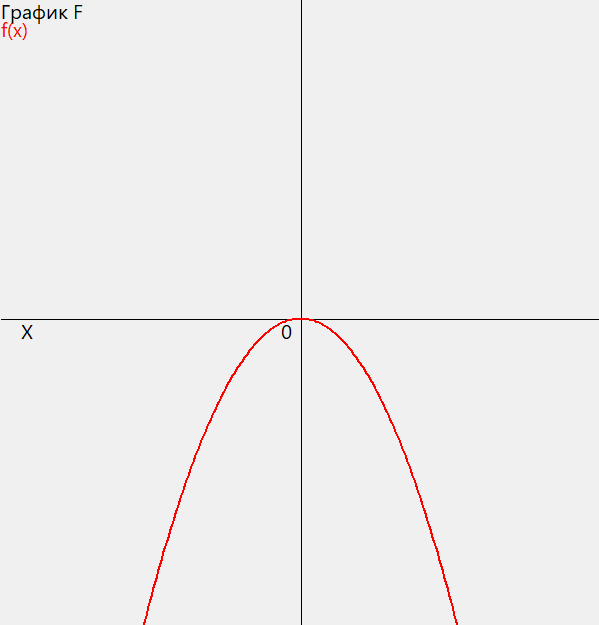


Рисунок 21 – Тестирование решения задачи 1. График 1

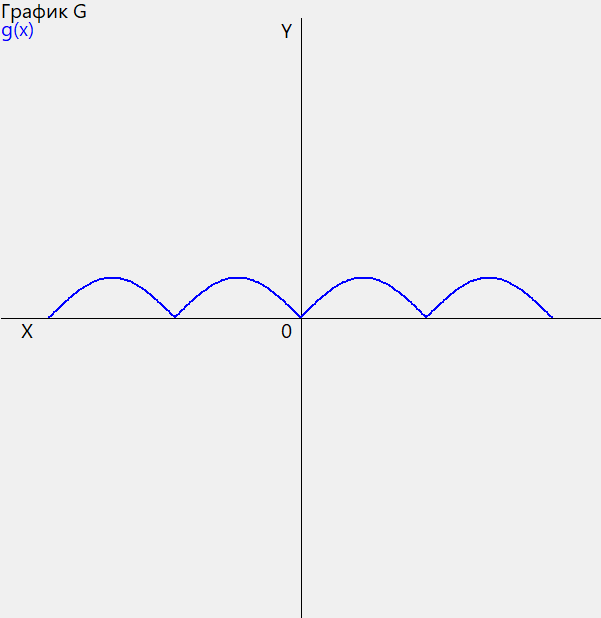


Рисунок 22 – Тестирование решения задачи 1. График 2

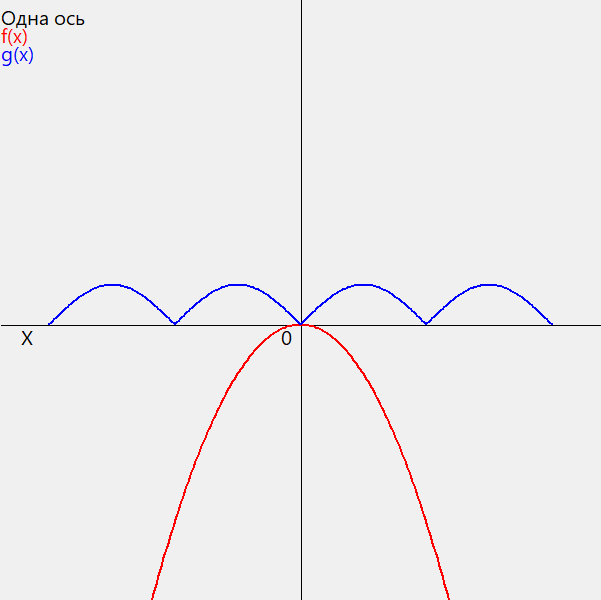


Рисунок 23 – Тестирование решения задачи 1

(отображение графиков на одной оси)

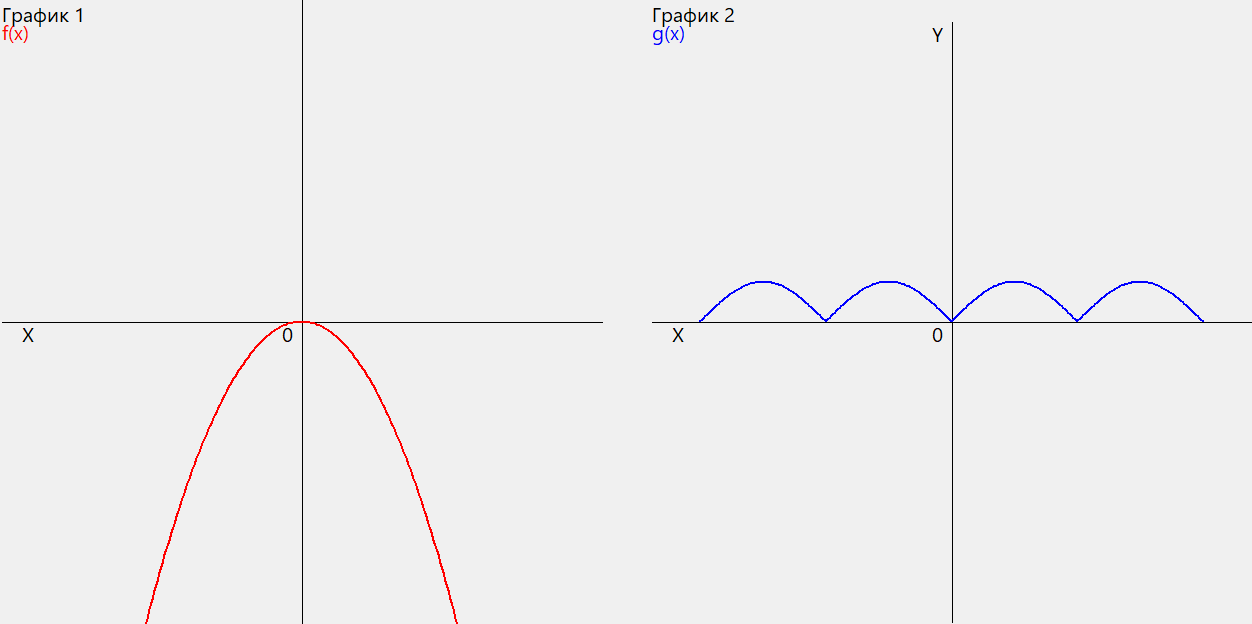


Рисунок 24 – Тестирование решения задачи 1

(отображение графиков на отдельные оси)

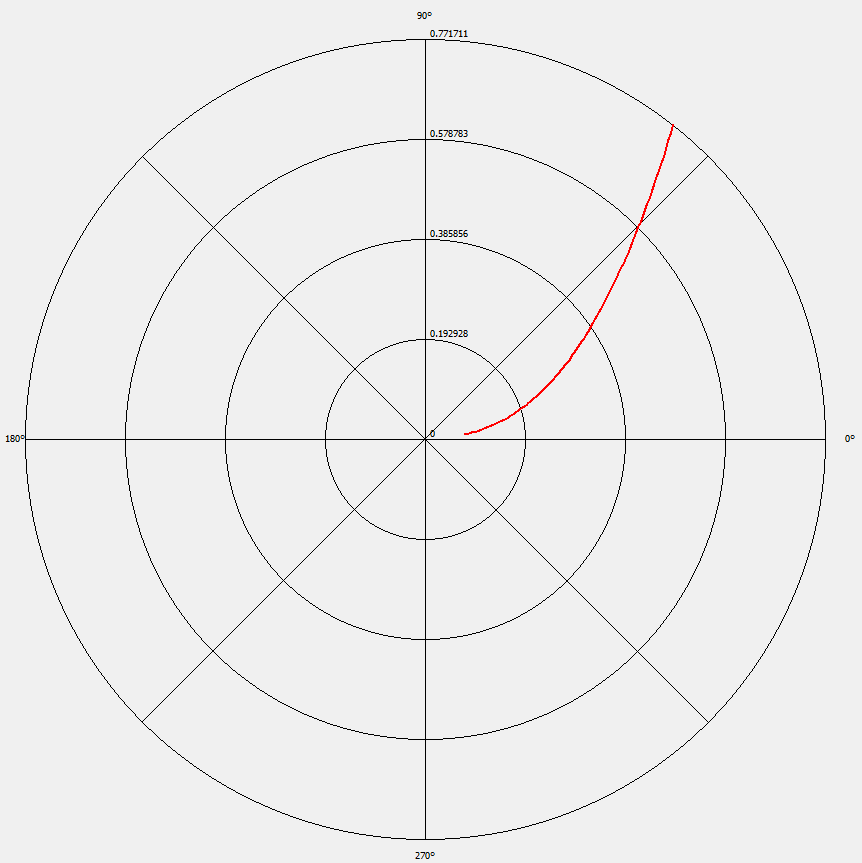


Рисунок 25 – Тестирование решения задачи 2

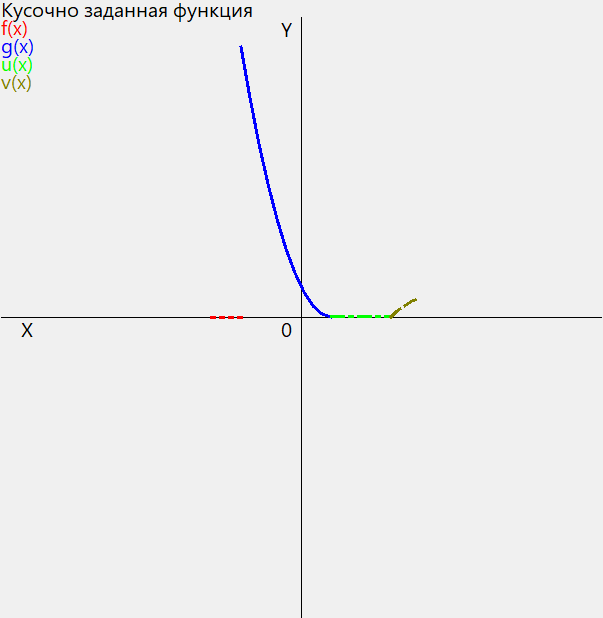


Рисунок 26 – Тестирование решения задачи 3

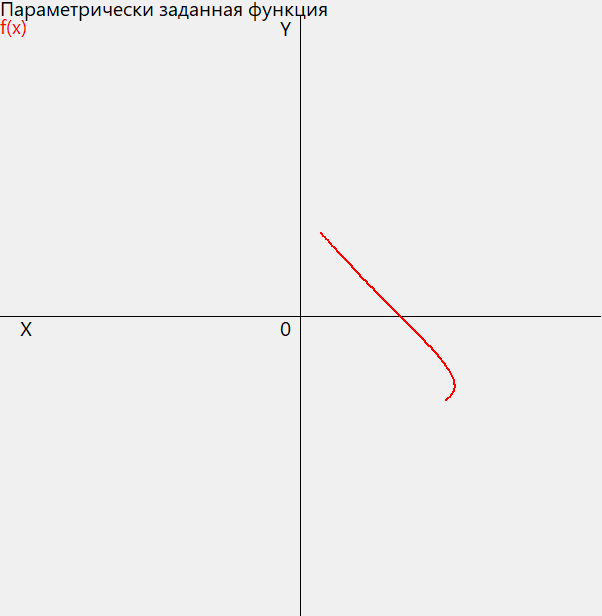


Рисунок 27 – Тестирование решения задачи 4

# 3 Рабочий график проведения практики

План проведения практики представлен в таблице 1.

Таблица 1 – План проведения практики

|  |  |  |
| --- | --- | --- |
| № п/п | Содержание  (типовые задания для текущего контроля) | Дата |
| 1 | Установочная конференция по практике | 29.06.2022 |
| 2 | Выполнение лабораторных работ 1-2 | 30.06.2022 |
| 3 | Защита лабораторных работ 1-2 | 01.07.2022 |
| 4 | Выполнение лабораторных работ 3-4 | 02.07.2022 |
| 5 | Защита лабораторных работ 3-4 | 04.07.2022 |
| 6 | Выполнение лабораторных работ 5-6 | 05.07.2022 |
| 7 | Защита лабораторных работ 5-6 | 06.07.2022 |
| 8 | Выполнение задания на программирование графики | 07.07.2022 |
| 9 | Защита задания на программирование графики | 08.07.2022 |
| 10 | Оформление отчёта и индивидуального плана | 09.07.2022 |
| 11 | Проверка отчёта и индивидуального плана | 11.07.2022 |
| 12 | Итоговая конференция по практике | 12.07.2022 |

# ЗАКЛЮЧЕНИЕ

# В ходе проектно-технологической практики были получены знания, а также приобретены умения и навыки, требуемые для работы с платформой Octave. В ходе выполнения лабораторных работ была осуществлена практическая отработка умений, связанных с арифметическими операциями на языке Octave, что также подразумевает работу с различными форматами данных. Также отрабатывались умения по работе с линейной алгеброй на платформе Octave (векторы, матрицы), проведена работа с применением Octave при решении геометрических задач (работа с графиками), а также вычислений, связанных с математическим анализом. Изучение языка платформы Octave подразумевало изучение способов работы с файлами-функциями.

# При выполнении практических заданий изучались и применялись синтаксические особенности работы со строками в данных средах.

На базе QT Creator разработаны приложения для отрисовки графиков. Реализованы данные приложения на базе языка программирования С++.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. OCTAVE [Сайт] URL: https://docs.exponenta.ru/Octave/index.html   
   (дата обращения : 30.06.2022).
2. Введение в Octave для инженеров и математиков: / Е. Р. Алексеев, О. В. Чеснокова – М.: ALT Linux, 2012. – 368 с.
3. Функция figure [Сайт]. URL: https://octave.sourceforge.io/octave/  
   function/figure.html (дата обращения: 03.07.2022).
4. Функция surfc [Сайт]. URL: [https://docs.exponenta.ru/Octave/ref/surfc.html](https://docs.exponenta.ru/matlab/ref/surfc.html) (дата обращения: 06.07.2022).