

**КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. А. Н. ТУПОЛЕВА – КАИ**

Институт компьютерных технологий и защиты информации

Кафедра: Автоматизированных систем обработки информации и управления

Г.М. НАБЕРЕЖНОВ, Н.Н. МАКСИМОВ

ЭЛЕМЕНТАРНОЕ ПРОГРАММИРОВАНИЕ ГРАФИКИ В OpenGL

**Методическое пособие
к лабораторным работам по курсу
«Компьютерная геометрия и графика»**

Студент, гр. 4210

Гауиш М.Г

Преподаватель

Гаптуллазянова Гульшат Ильдусовна

Казань – 2023

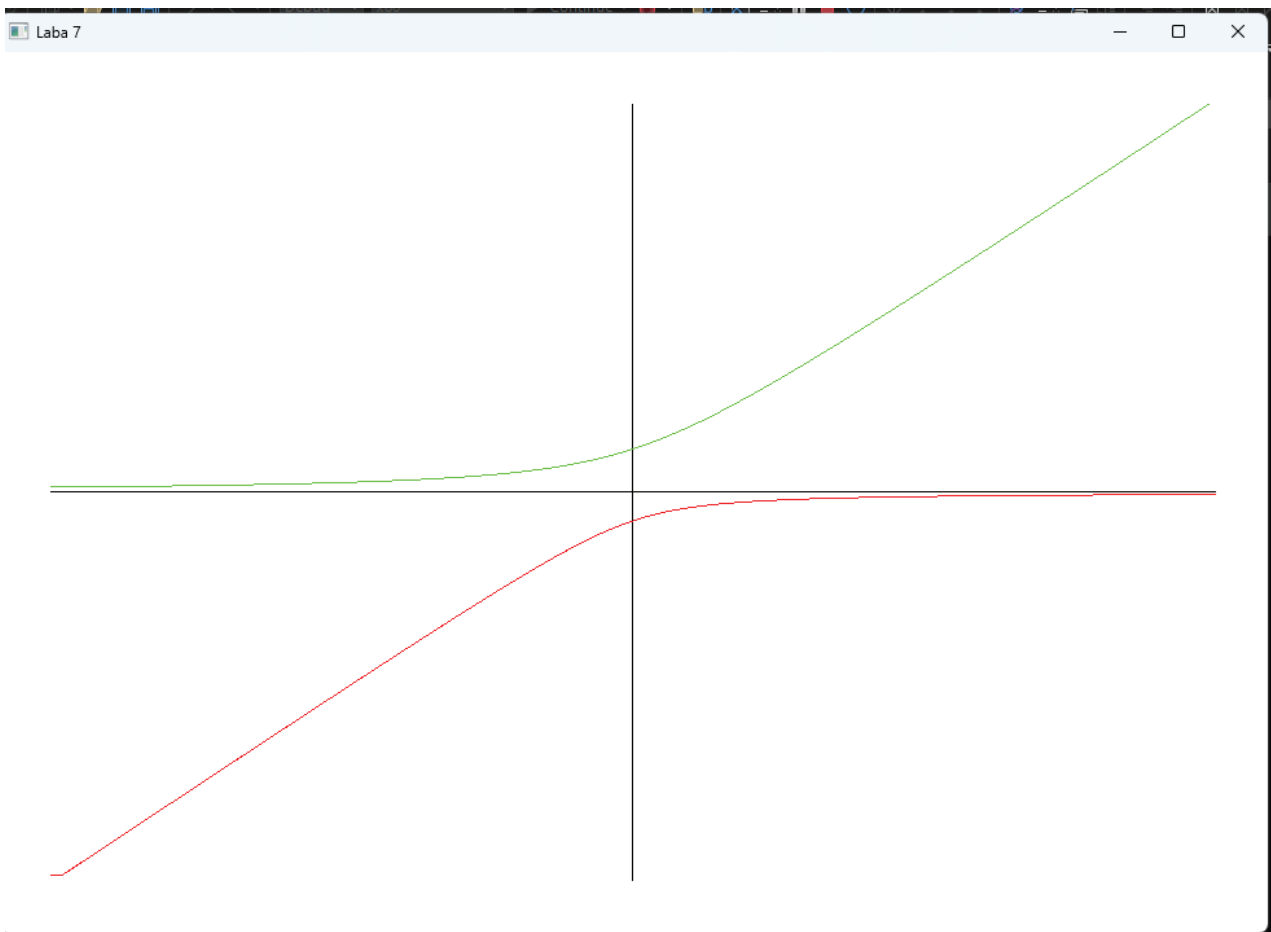
Лабораторная работа №7 (Отчет)

Задания для самостоятельного выполнения

Методами оценочной функции и ЦДА построить заданную линию.

1. Отрезок прямой в заданной четверти в указанном направлении.
2. Дуга окружности в заданной четверти в указанном направлении.
3. $y = 2px.$
4. $y = a\sqrt{2}x$
5. $x = -ay.$
6. $y^2 = ax + b.$

Скриншот работы программы:



Код:

```
<include <GL/glut.h#
define _USE_MATH_DEFINES#
<include <cmath#
<include <math.h#
;()void init
;()void draw

void main(int argc, char** argv)
{
    glutInit(&argc, argv); //инициализация библиотеки GLUT
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); // устанавливается режим
    дисплея: режим двойной буферизации (в одном буфере кадра выполняется рисование,
    другой отображается) и переключение буферов; цвета представлять как смесь RGB
    устанавливается размер экранного окна window// ;(680 ,980)glutInitWindowSize
```

```
устанавливается положение (позиция) экранного// ;(50 ,50)glutInitWindowPosition
окна window
инициализируется открытие экранного окна// ;("7 Laba")glutCreateWindow
window
;()init
glutDisplayFunc(draw); //функция draw( ) регистрируется как функция обратного
вызова для события открытия или обновления экранного окна
glutMainLoop(); //переводит программу в бесконечный цикл ожидания события
{
```

функция инициализации//

```
()void init
}
```

```
установка цвета фона// ;(1.0 ,1.0 ,1.0 ,1.0)glClearColor
glMatrixMode(GL_PROJECTION); //в качестве текущей устанавливается матрица
проецирования
glLoadIdentity(); //текущая матрица устанавливается в единицу
устанавливается мировое окно// ;(75.0 ,75.0- ,75.0 ,75.0-)gluOrtho2D
{
```

функция рисования и обновления окна//

```
()void draw
}
```

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //очистка буфферов до
предустановленных значений
установка положения и размеров порта просмотра// ;(600 ,900 ,40 ,40)glViewport
установка цвета рисования// ;(0.0 ,0.0 ,0.0)glColor3f
отрисовка осей//
;glBegin(GL_LINES)
;(0.0 ,75.0)glVertex2f ;(0.0 ,75.0-)glVertex2f
;(75.0- ,0.0)glVertex2f ;(75.0 ,0.0)glVertex2f
;()glEnd
```

переменные, хранящие константы// ;3 = c ,10 = b ,1 = GLfloat a

начальные значения x и y для метода оценочной// ;74.0- = Y1 ,75.0- = GLfloat X1
функции

начальные значения x и y для метода ЦДА// ,1.0 = Y2 ,75.0- = GLfloat X2

шаг по x и y для метода ЦДА// ;0.0 = dY ,0.0 = dX

```
; (0.0 ,0.0 ,1.0)glColor3f
```

МЕТОД ОЦЕНОЧНОЙ ФУНКЦИИ//

```
glBegin(GL_LINE_STRIP); //рисует связанную группу линейных сегментов от
первой вершины до последней
```

цикл по четвертой четверти// (75 > X1 && 75- < Y1) while

```
{
(0 > b * c - Y1 * a * X1 - (2 ,Y1)pow) if
{
;(Y1 ,X1)glVertex2d
;0.01 += X1
{
(0 < b * c - Y1 * a * X1 - (2 ,Y1)pow) else if
{
;(Y1 ,X1)glVertex2d
;0.01 += Y1
{
} else
} (Y1 => X1) if
;(Y1 ,X1)glVertex2d
```

```

;0.01 += X1
{
} else
;(Y1 ,X1)glVertex2d
;0.01 += Y1
{
{
{
;()glEnd
МЕТОД ЦДА//
установка цвета рисования// ;(0.0 ,1.0 ,0.0)glColor3f
glBegin(GL_LINE_STRIP); //рисует связанную группу линейных сегментов от
первой вершины до последней
цикл по первой четверти// (75 > X2) while
}
;dX += X2
;dY += Y2
расчет шага по x и y//
;((2 ,a * Y2-)pow + (2 ,a * X2 - Y2 * 2)pow)sqrt / (a * X2 - Y2 * 2) = dX
;((2 ,a * Y2-)pow + (2 ,a * X2 - Y2 * 2)pow)sqrt / (a * Y2-) = dY
;(Y2 ,X2)glVertex2f
{
;()glEnd
glFlush(); //принудительное выполнение накопленных команд OpenGL
{

```