

**Московский Авиационный Институт
(национальный исследовательский университет)
Факультет прикладной математики**

**Компьютерная графика
Отчет по лабораторной работе №1**

Бондарева Елена
Группа М8О-305Б-21

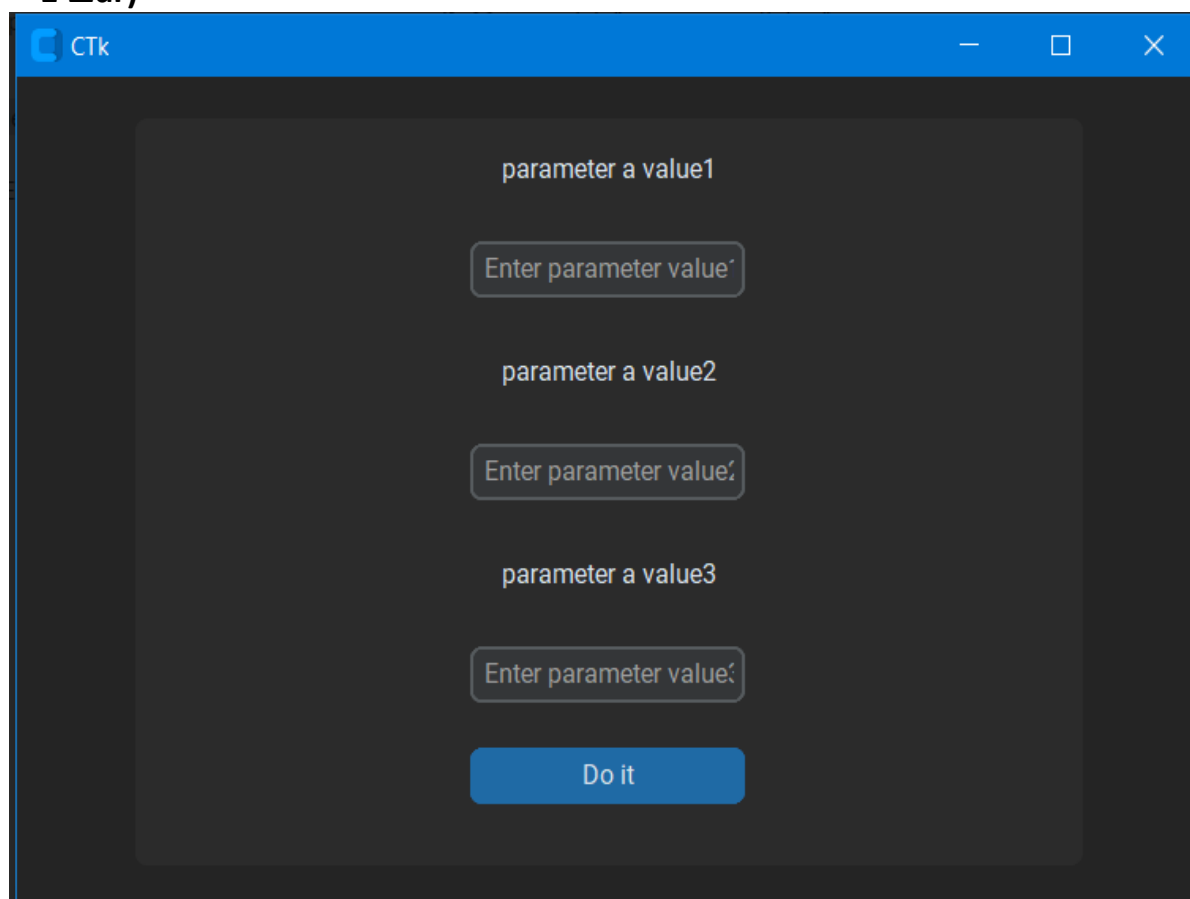
Преподаватель: Симкин О.В.

Москва, 2023

1. **Тема:** Построение изображений 2D- кривых.
2. **Цель работы:** Написать и отладить программу, строящую изображение заданной замечательной кривой.
3. **Задание (вариант №19):** $x = 3at/(1+t^3)$, $y = 3at^2/(1+t^3)$, $A \leq t \leq B$, -1
4. **Идея, метод, алгоритм** решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Для получения изображения сначала пользователю будет представлено окно для ввода параметров, после введения которых непосредственно будет выведен сам график.

1 шаг)



The screenshot shows a Qt window titled "CTk" with a dark gray background. Inside the window, there is a light gray rectangular area containing three input fields and a button. Each input field is preceded by the text "parameter a value" followed by a subscript (1, 2, or 3). The input fields themselves contain the placeholder text "Enter parameter value:". Below the input fields is a blue button with the text "Do it".

2 шаг) Ввод параметров a ; а также A и B , удовлетворяющих следующему неравенству: $A \leq t \leq B$, -1

CTk

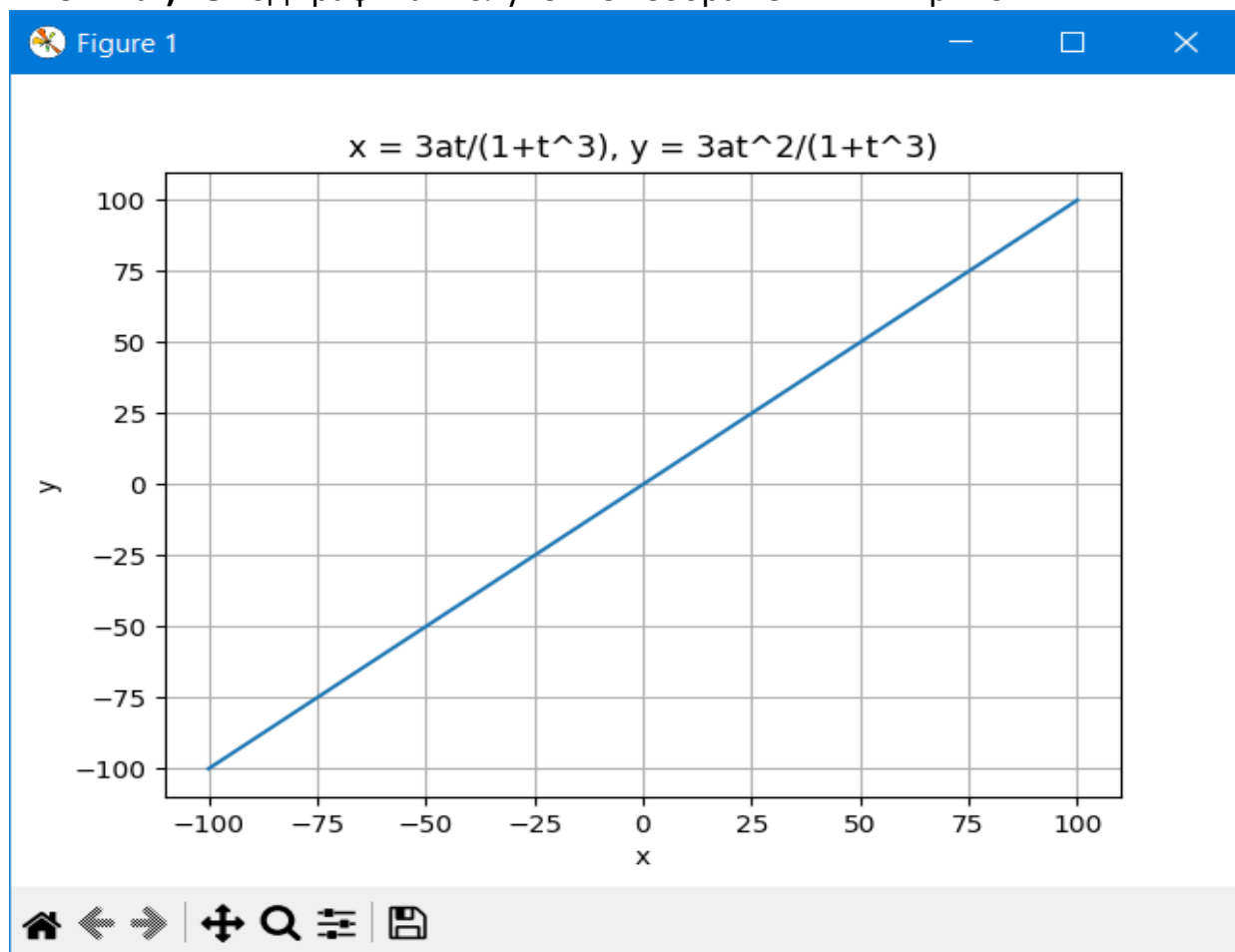
parameter a value1

parameter a value2

parameter a value3

Do it

3 шаг) Вывод графика. Получение изображения 2D- кривой.



5. Распечатка протокола:

main.py

```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *
import numpy as np
from math import *

# Constants for the paraboloid
a = 0.5
b = 0.5
c = 1.0

dots = int(input("Please, input amount of dots for approximation: "))

global xrot          # Величина вращения по оси x
global zrot          # Величина вращения по оси z
global ambient       # Рассеянное освещение
global lightpos       # Положение источника освещения
global scale

scale = 75

def picture():
    global scale

    for i in np.linspace(-scale, scale, dots):
        # glBegin(GL_LINE_STRIP)
        glBegin(GL_LINES)
        for j in np.linspace(-scale, scale, dots):
            x = a * i / 50
            y = b * j / 50
            z = c * (x**2 + y**2)
            glColor3f(1.0, 0.0, 0.0)
            glVertex3f(x, y, z)
        glEnd()

# Function to draw the paraboloid
def draw_paraboloid():
    global xrot
    global zrot
    global lightpos

    print("draw_parabaloid invoked")

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glPushMatrix()
    gluLookAt(0, 0, 5, 0, 0, 0, 0, 1, 0)
    glRotatef(xrot, 1.0, 0.0, 0.0) # Вращаем по оси X на
    # величину xrot
    glRotatef(zrot, 0.0, 1.0, 0.0)
    glLightfv(GL_LIGHT0, GL_POSITION, lightpos) # Источник света
    # вращаем вместе с елкой

    picture()

    glPopMatrix()
    glutSwapBuffers()

# Function to handle window resizing
def reshape(width, height):
    glViewport(0, 0, width, height)
    glMatrixMode(GL_PROJECTION)
```

```

glLoadIdentity()
gluPerspective(45, float(width) / height, 0.1, 100)
glMatrixMode(GL_MODELVIEW)

# Function to initialize OpenGL
def init():

    global xrot          # Величина вращения по оси x
    global zrot          # Величина вращения по оси z
    global ambient       # Рассеянное освещение
    global lightpos      # Положение источника освещения

    xrot = 0.0           # Величина вращения по оси x = 0
    zrot = 0.0           # Величина вращения по оси z = 0
    ambient = (1.0, 1.0, 1.0, 1) # Первые три числа - цвет в формате RGB, а
последнее - яркость
    greencolor = (0.2, 0.8, 0.0, 0.8) # Зеленый цвет для иголок
    treecolor = (0.9, 0.6, 0.3, 0.8)  # Коричневый цвет для ствола
    lightpos = (1.0, 1.0, 1.0)         # Положение источника освещения по осям хуз

    gluOrtho2D(-1.0, 1.0, -1.0, 1.0) # Определяем границы рисования по
горизонтали и вертикали
    glRotatef(30, 1, 0, 0)
    glRotatef(30, 0, 1, 0)
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambient) # Определяем текущую модель
освещения
    glEnable(GL_DEPTH_TEST)
    glClearColor(0.5, 0.0, 0.5, 1.0) # Серый цвет для первоначальной
закраски
    gluOrtho2D(-1.0, 1.0, -1.0, 1.0) # Определяем границы рисования по
горизонтали и вертикали
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambient) # Определяем текущую модель
освещения
    glEnable(GL_LIGHTING) # Включаем освещение
    glEnable(GL_LIGHT0)   # Включаем один источник света
    glLightfv(GL_LIGHT0, GL_POSITION, lightpos) # Определяем положение источника
света

speeds = np.linspace(0, 2 * pi, 100)
global sinusoida
global index
index = 0
index_step = 1
sinusoida = np.sin(speeds)

# Процедура обработки специальных клавиш
def specialkeys(key, x, y):
    global xrot
    global scale
    global zrot
    global index
    global index_step

    # Обработчики для клавиш со стрелками
    if key == GLUT_KEY_UP: # Клавиша вверх
        xrot -= 4.0        # Уменьшаем угол вращения по оси X
    if key == GLUT_KEY_DOWN: # Клавиша вниз
        xrot += 4.0        # Увеличиваем угол вращения по оси X
    if key == GLUT_KEY_LEFT: # Клавиша влево
        zrot -= 4.0        # Уменьшаем угол вращения по оси Z
    if key == GLUT_KEY_RIGHT: # Клавиша вправо
        zrot += 20.0 * sinusoida[index]
        if index == len(sinusoida) - 1 or index < 0:
            index_step = -index_step
        index += index_step
    glutPostRedisplay() # Вызываем процедуру перерисовки

glutInit()

```

```
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH)
glutInitWindowSize(800, 600)
glutCreateWindow(b"3D Paraboloid")
glutDisplayFunc(draw_paraboloid)
glutReshapeFunc(reshape)
glutSpecialFunc(specialkeys)
init()
glutMainLoop()
```

module1.py

```
import numpy as np
import matplotlib.pyplot as plt

# Задаем функции x(t) и y(t)
def x(a, t):
    return 3 * a * t / (1 + t**3)

def y(a, t):
    return 3 * a * t**2 / (1 + t**3)

def doit(a, A, B):
    # Создаем массив значений t в заданном диапазоне
    t = np.linspace(A, B, 100)

    # Вычисляем значения x и y для каждого значения t
    x_value = x(a, t)
    y_value = y(a, t)

    # Строим график
    plt.plot(x_value, y_value)

    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('x = 3at/(1+t^3), y = 3at^2/(1+t^3)')
    plt.grid(True)
    plt.show()
```