

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Компьютерная графика»

Студент: Д. А. Ваньков  
Преподаватель: Г. С. Филиппов  
Группа: М8О-307Б  
Дата:  
Оценка:  
Подпись:

Москва, 2019

# Построение изображений 2D- кривых

**Задача:** Написать и отладить программу, строящую изображение заданной кривой. Обеспечить автоматическое масштабирование и центрирование кривой при изменении размеров окна.

**Вариант №18:**

Кривая:  $x = 3 * a * t / (1 + t^3)$ ,  $y = 3 * a * t^2 / (1 + t^3)$ ,  $-1 < A \leq t \leq B$

$a$  - константа, значение которой вводится с клавиатуры

## 1 Описание

Для реализации я использовал библиотеку GLUT и язык Python.

Идея заключается в последовательной отрисовке точек графика с заданной частотой аппроксимации, при малых значениях которой график на картинке достаточно приближен к идеальной математической модели.

Корректное отображение при изменении размеров окна также реализуется библиотекой, которую нужно инициализировать в начале программы.

## 2 Исходный код

В начале подключаются необходимые. Библиотека GLUT подключается как «GL/glut.h»

```
1 | from OpenGL.GL import *
2 | from OpenGL.GLU import *
3 | from OpenGL.GLUT import *
4 | from math import cos, sin, pi
5 | from sympy.solvers import solve
6 | from sympy import Symbol
```

Затем задаются глобальные переменные от которых зависит вид графика и его аппроксимация.

```
1 | global X_COORD
2 | global Y_COORD
3 | global ITERATIONS
4 | global x_center
```

```

5 || global y_center
6 || global a
7 || global A
8 || global B

```

Функция *drawgrid* отрисовывает координатную прямую с отметками на ней.

```

1 || def drawgrid(SERIF_OFFSET, SERIF_DISTANCE)

```

Самой отрисовкой графика занимается функция *drawfunc*, ставится точка удовлетворяющая функции, а затем производится перерасчет координат с шагом *ITERATIONS*.

```

1 || def drawfunc()

```

Отрисовка происходит в бесконечном «игровом» цикле, который выполняет в себе функцию *dispay*, а также обрабатывает обратные вызовы.

```

1 || def display()

```

Затем вызываются функции для отображения окна и инициализаций функций отрисовки.

### 3 Запуск программы

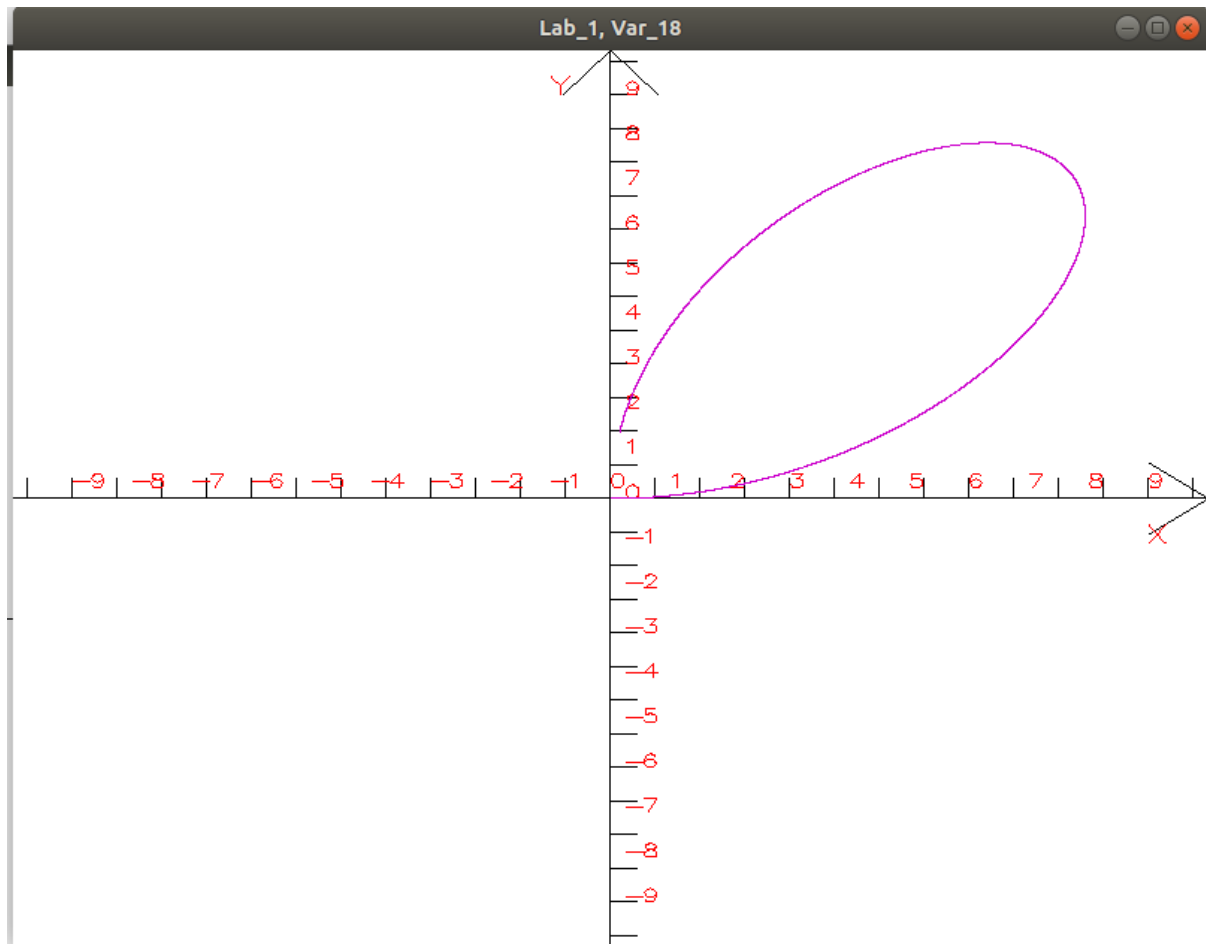
В консоли необходимо скомпилировать исходный код и запустить. В окне необходимо будет ввести параметры  $a$ ,  $A$ ,  $B$  от которых зависит вид графика. Если параметры введены некорректно выбрасывается исключение.

```

(base) chappybunny@chappybunny:~/CG$ python lab1.py
Enter a:
5
Enter A >-1:
0
Enter B >A:
10

```

После ввода параметров откроется изображение графика в окне следующего типа:



Это окно можно изменять по размерам и перемещать по экрану без всяких побочных эффектов, график подстраивается под изменение размеров экрана и масштабируется соответствующим образом.

## 4 Выводы

Выполнив данную лабораторную работу, я познакомился с библиотекой GLUT и как она реализуется на языке Python. В дальнейшем, в связи с отсутствием достаточного количества материалов для изучения этой библиотеки на Python, планирую перейти на язык C++.

В процессе решения задачи я столкнулся с такими трудностями, как изображение кривой. Однако эта проблема решилась поточечной отрисовкой всей кривой с изменением шага и пересчетом координат. А также проблема нумерования осей, так как все числа постоянно располагались не на осях, а где попало.