

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: Я. С. Поскряков  
Преподаватель: Г. С. Филиппов  
Группа: М8О-306Б  
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:  $p = a * (1 - \cos(\phi))$

$a$  - константа, значение которой вводится с клавиатуры

## 1 Описание

Для реализации я использовал библиотеку Qt 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 QT подключается как «PyQt5»

```
1 | import sys
2 | from PyQt5.QtWidgets import QWidget, QApplication, QLineEdit
3 | from PyQt5.QtGui import QPainter, QColor, QPen
4 | from PyQt5.QtCore import Qt
```

Класс *Example()* отрисовывает координатную прямую с отметками на ней.

```
1 | ex = Example()
```

Самой отрисовкой графика занимается функция *def drawFigure(self, qp)*, которая является методом класса *Example()*, 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 *display*, а также обрабатывает обратные вызовы.

```
1 || def display()
```

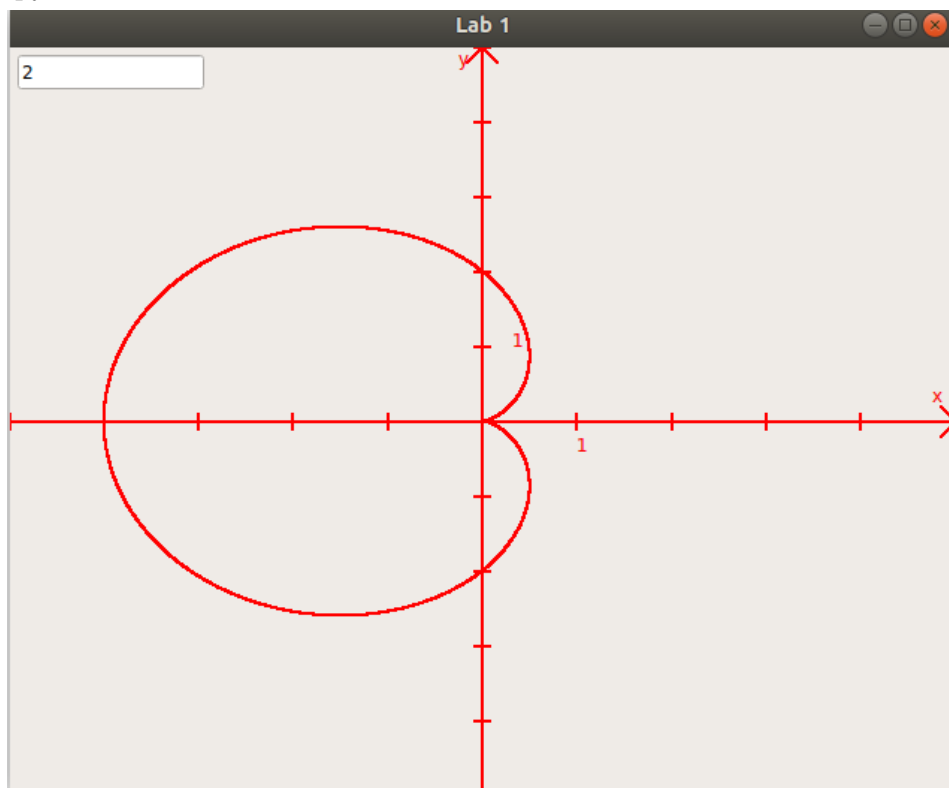
Затем вызываются функции для отображения окна и инициализаций функций отрисовки.

### 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) yar@yarmachine:~/CG/already/LAB1/My python main.py
```

Непосредственно после запуска программы откроется изображение графика в окне следующего типа, где можно самому задавать параметр, определяющий отрисовку функции:



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.