

**Московский авиационный институт  
(национальный исследовательский университет)**

**Институт №8 «Информационные технологии и прикладная  
математика»**

**Кафедра 806 «Вычислительная математика и  
программирование»**

**Лабораторная работа №1 по курсу «Компьютерная графика»**

Студент: М. А. Инютин  
Преподаватель: А. В. Морозов  
Группа: М8О-307Б-19  
Дата:  
Оценка:  
Подпись:

**Москва, 2021**

## Построение изображений 2D- кривых

**Задача:** Написать и отладить программу, строящую изображение заданной замечательной кривой.

**Вариант задания:**  $\rho = a \cdot e^{k\phi}$ ,  $0 \leq \phi \leq B$

# 1 Описание

Для изображения кривой используется техника расчёта координат кривой при каких-то близких значениях  $\phi$ . Под шагом подразумевается модуль разности  $|\phi_i - \phi_{i-1}|$ . Чем больше степень аппроксимации, тем меньше шаг.

Полученные координаты точек нужно перевести в экранные координаты, так как их начало находится в левом верхнем углу экрана и ось  $Oy$  направлена вниз, а не вверх. С помощью библиотеки `Caio` рисуем линии между соседними точками кривой, получая ломаную. При достаточной степени аппроксимации она визуально не отличима от кривой.

## 2 Исходный код

Почти код весь описан в *MainWindow.cs*.

Метод обновления картинки на экране:

```
1 private void UpdateScreen(Context ct)
2 {
3     GetData();
4     points = Misc.GenFunctionValues(a, k, B, step);
5     RotateAndRescale();
6     center = new Vector2D(width / 2.0, height / 2.0);
7     center = center + shift;
8     for (int i = 0; i < points.Count; ++i)
9     {
10         points[i].X = scale.X * points[i].X;
11         points[i].Y = scale.Y * points[i].Y;
12         points[i] = points[i] + center;
13         points[i].Y = 2 * center.Y - points[i].Y;
14     }
15     DrawAxes(ct);
16     DrawScale(ct);
17     DrawScale(ct);
18     DrawPlot(ct);
19     DrawRotationPoint(ct);
20 }
```

Методы для отображения координатных осей и кривой.

```
21 private void DrawScaleOX(Context ct)
22 {
23     double scaleDiv = 1e-3;
24     for (int degree = -3; degree <= 5; ++degree) {
25         if (scale.X * scaleDiv > DIVISION_SCALE_PIXELS)
26         {
27             for (int i = 1; center.X + i * scale.X * scaleDiv < width; ++i)
28             {
29                 DrawLine(ct, new Vector2D(center.X + i * scale.X * scaleDiv, OXdown),
30                     new Vector2D(center.X + i * scale.X * scaleDiv, OXup));
31                 PrintText(ct, new Vector2D(center.X + i * scale.X * scaleDiv - 10,
32                     OXdown + 10), Misc.NumToString(i, degree));
33             }
34             for (int i = 1; center.X - i * scale.X * scaleDiv > 0; ++i)
35             {
36                 DrawLine(ct, new Vector2D(center.X - i * scale.X * scaleDiv, OXdown),
37                     new Vector2D(center.X - i * scale.X * scaleDiv, OXup));
38                 PrintText(ct, new Vector2D(center.X - i * scale.X * scaleDiv - 10,
39                     OXdown + 10), Misc.NumToString(-i, degree));
40             }
41             break;
42         }
43     }
```

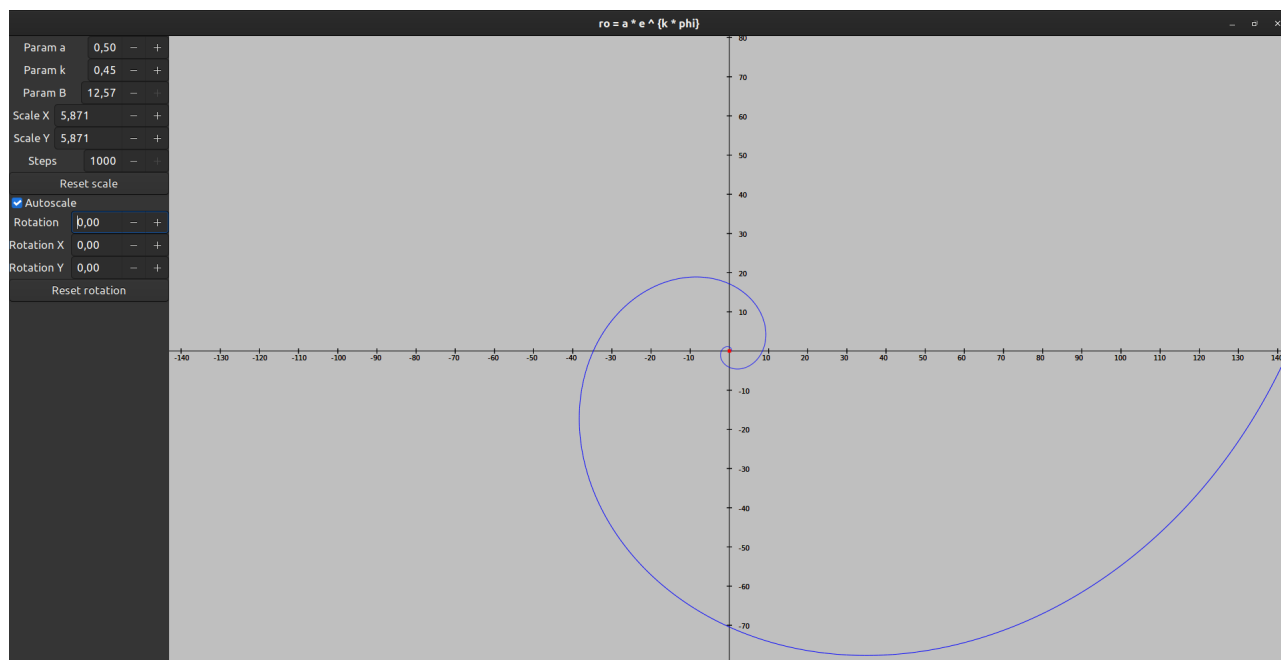
```

39     scaleDiv = scaleDiv * 10;
40 }
41 }
42
43 private void DrawScaleOY(Context ct)
44 {
45     double scaleDiv = 1e-3;
46     for (int degree = -3; degree <= 5; ++degree) {
47         if (scale.Y * scaleDiv > DIVISION_SCALE_PIXELS)
48             {
49                 for (int i = 1; center.Y + i * scale.Y * scaleDiv < height; ++i)
50                     {
51                         DrawLine(ct, new Vector2D(OYleft, center.Y + i * scale.Y * scaleDiv),
52                             new Vector2D(OYRight, center.Y + i * scale.Y * scaleDiv));
53                         PrintText(ct, new Vector2D(OYRight + 10, center.Y + i * scale.Y *
54                             scaleDiv + 5), Misc.NumToString(-i, degree));
55                     }
56                 for (int i = 1; center.Y - i * scale.Y * scaleDiv > 0; ++i)
57                     {
58                         DrawLine(ct, new Vector2D(OYleft, center.Y - i * scale.Y * scaleDiv),
59                             new Vector2D(OYRight, center.Y - i * scale.Y * scaleDiv));
60                         PrintText(ct, new Vector2D(OYRight + 10, center.Y - i * scale.Y *
61                             scaleDiv + 5), Misc.NumToString(i, degree));
62                     }
63                 break;
64             }
65         scaleDiv = scaleDiv * 10;
66     }
67 }
68
69 private void DrawPlot(Context ct)
70 {
71     ct.SetSourceRGB(0, 0, 1);
72     for (int i = 1; i < points.Count; ++i)
73     {
74         DrawLine(ct, points[i], points[i - 1]);
75     }
76 }

```

### 3 Демонстрация работы программы

Кривая с максимальным числом шагов.



Поворот кривой относительно точки.

