

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: И. П. Моисеенков
Преподаватель: Г. С. Филиппов
Группа: М8О-308Б-19
Дата: 21.10.2021
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №1

Построение изображения 2D-кривых

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

Обеспечить автоматическое масштабирование и центрирование кривой при изменении размеров окна.

Вариант 11: $r = \frac{a}{\phi}$, $0 < A \leq \phi \leq B$, где a , A , B - параметры, задаваемые пользователем.

1 Описание

Для выполнения данного задания я использовал библиотеку matplotlib для python. Эта библиотека позволяет довольно просто работать с двумерными и трехмерными изображениями, но в то же время она предлагает довольно много различных настроек отображения графиков (в том числе и в полярных координатах).

Чтобы пользователь мог изменять параметры кривой во время работы программы, я использовал TextBox из matplotlib.widgets. Текстбокс представляет собой ячейку, куда пользователь может ввести желаемое значение. После того, как пользователь закончит ввод, текстбокс запустит функцию для того, чтобы обновить изображение кривой. В своей программе я использовал три текстбокса для трех изменяемых параметров (a , A , B).

Также matplotlib автоматически масштабирует графики при изменении размера окна и позволяет увеличивать/уменьшать изображение.

2 Исходный код

```
1 | import numpy as np
2 | import matplotlib.pyplot as plt
3 | from matplotlib.widgets import TextBox
4 |
5 |
6 | """
7 | function:  $r = a / \phi$ ;  $\phi$  in  $[A; B]$ 
8 |  $a$ ,  $A$ ,  $B$  are entered by user
9 | """
10 | eps = 1e-8
11 | a = 1
12 | A = 0.05
```

```

13 B = 40
14 phi = np.linspace(A+eps, B-eps, int(B-A) * 10)
15 r = a / (phi+eps)
16
17
18 def update_function(param_name, param_val):
19     """
20     Updates data after changing the parameters
21     """
22     global phi, r, a, A, B
23     if param_name == 'a':
24         a = param_val
25     elif param_name == 'A':
26         A = param_val
27     elif param_name == 'B':
28         B = param_val
29
30     phi = np.linspace(A + eps, B - eps, max(0, int(B-A) * 10))
31     r = a / (phi + eps)
32     p.set_xdata(phi)
33     p.set_ydata(r)
34
35
36 fig = plt.figure()
37 fig.subplots_adjust(bottom=0.2)
38 fig.canvas.mpl_disconnect(fig.canvas.manager.key_press_handler_id)
39
40 ax = fig.add_subplot(111, projection='polar')
41 p, = ax.plot(phi, r)
42 ax.set_title(r"$r = \frac{a}{\phi}; \phi \in [A; B]$")
43 ax.set_rmax(a+eps)
44 ax.set_rlabel_position(-22.5) # get radial labels away from plotted line
45 ax.grid(True)
46
47
48 def submit_fn_a(value):
49     """
50     Updates the plot after changing the parameter a in textbox
51     """
52     update_function('a', float(value))
53     ax.set_rmax(a+eps)
54     plt.draw()
55
56
57 axbox_a = fig.add_axes([0.1, 0.05, 0.2, 0.06])
58 text_box_a = TextBox(axbox_a, "a ")
59 text_box_a.on_submit(submit_fn_a)
60 text_box_a.set_val(a) # Trigger 'submit' with the initial value
61

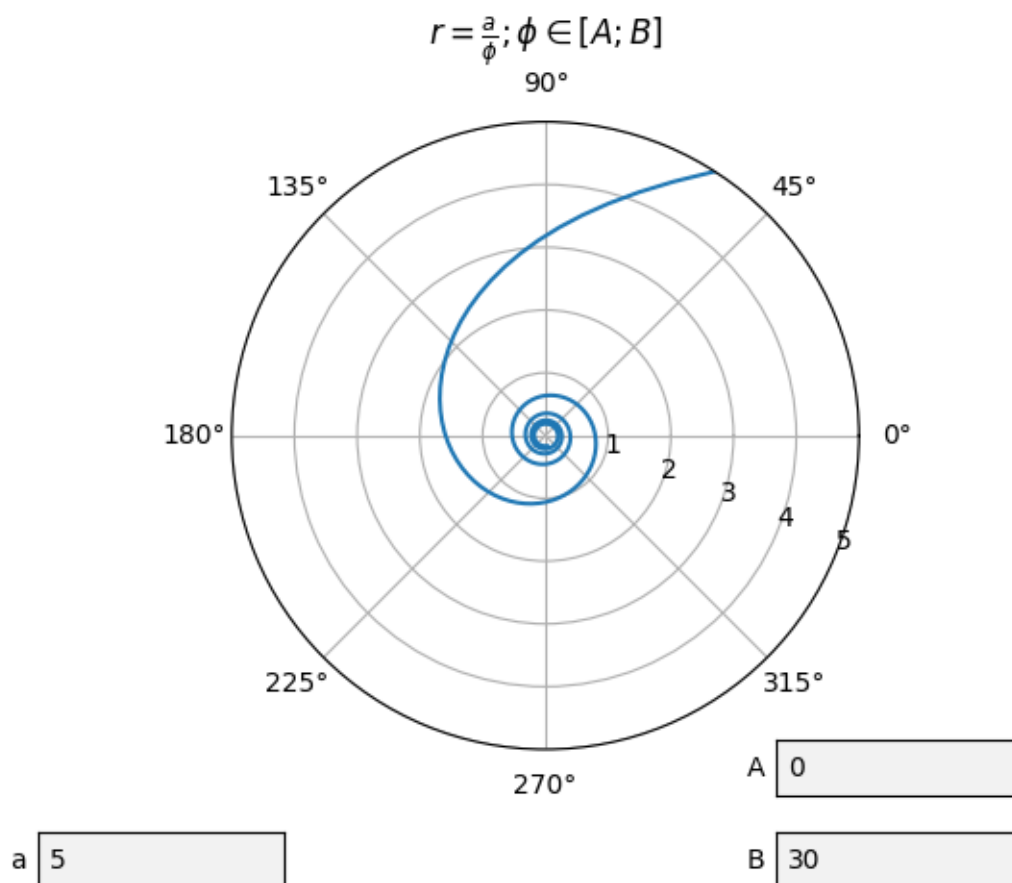
```

```

62
63 def submit_fn_A(value):
64     """
65     Updates the plot after changing the range of phi in textbox
66     """
67     update_function('A', float(value))
68     plt.draw()
69
70
71 axbox_A = fig.add_axes([0.7, 0.15, 0.2, 0.06])
72 text_box_A = TextBox(axbox_A, "A ")
73 text_box_A.on_submit(submit_fn_A)
74 text_box_A.set_val(A)
75
76 def submit_fn_B(value):
77     """
78     Updates the plot after changing the range of phi in textbox
79     """
80     update_function('B', float(value))
81     plt.draw()
82
83
84 axbox_B = fig.add_axes([0.7, 0.05, 0.2, 0.06])
85 text_box_B = TextBox(axbox_B, "B ")
86 text_box_B.on_submit(submit_fn_B)
87 text_box_B.set_val(B)
88
89 plt.show()

```

3 Результат работы



4 Выводы

Выполнив первую лабораторную работу по компьютерной графике, я попрактиковался в работе с библиотекой `matplotlib` и изучил различные виды виджетов, которые позволяет использовать эта библиотека.