

# Computer Graphics

## Homework 3

<b>Name</b>	Ranime Ahmed Elsayed Shehata
<b>ID</b>	21010531

## **Problem Statement:**

### Drawing Free Curves (Bezier Technique)

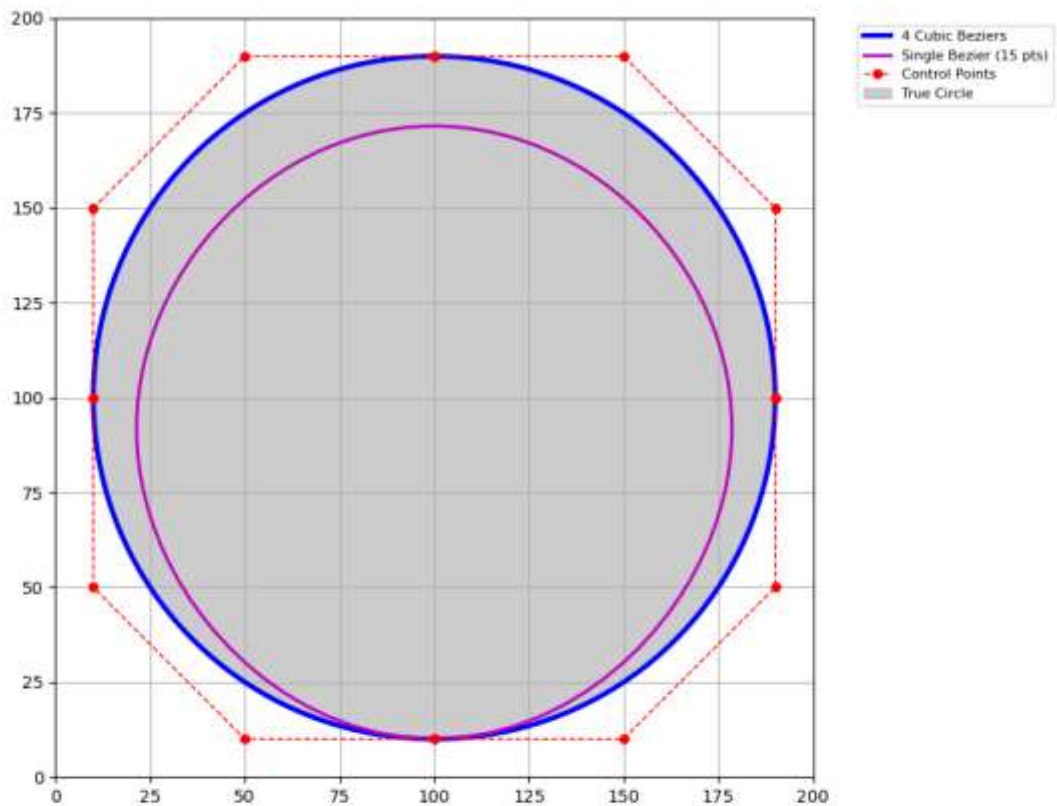
1. Using 4 Bezier Curves, as indicated below, Write an OpenGL program to draw a circle with center at (100,100) and radius=90, where  $P_0 = (100, 10)$ ,  $P_1 = (150, 10)$ ,  $P_2 = (190, 50)$ ,  $P_3 = P_4 = (190, 100)$ ,  $P_5 = (190, 150)$ ,  $P_6 = (150, 190)$ ,  $P_7 = P_8 = (100, 190)$ ,  $P_9 = (50, 190)$ ,  $P_{10} = (10, 150)$ ,  $P_{11} = P_{12} = (10, 100)$ ,  $P_{13} = (10, 50)$ ,  $P_{14} = (50, 10)$ ,  $P_{15} = P_0$ .

- I. Using one single Bezier Curve defined by 13 control points (note that First Control Point = Last Control Point).
  - II. Compare the above circles with a circle (of different color) drawn using high level commands (in the programming language you are using).
  - III. Move point  $P_1$  vertically (up and down) and observe the effect on the drawings in cases a) and b) above. What do you conclude?
- 
-

## Code Snippet:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib.widgets import Slider
4
5 # -----
6 # Slider class
7 # -----
8
9 class Slider:
10     @staticmethod
11     def twoPoints(x1, x2):
12         return (x1 - x2) * x1 + x1 * x2
13
14     @staticmethod
15     def points(x, points):
16         return [Slider.twoPoints(x, points[i]) * points[i] for i in range(len(points) - 1)]
17
18     @staticmethod
19     def points(x, points):
20         while len(points) > 1:
21             points = Slider.twoPoints(x, points)
22         return points[0]
23
24     @staticmethod
25     def Curve(x, x1, x2, points):
26         return np.array([Slider.twoPoints(x, points[i]) for i in range(1, len(points))])
27
28 # -----
29 # Initial points function
30 # -----
31 def create_control_points(x):
32     return np.array([
33         100, 101,
34         120, 11, # 0 is important
35         180, 91,
36         190, 100,
37         190, 100,
38         180, 190,
39         190, 190,
40         190, 100,
41         180, 190,
42         18, 190,
43         18, 100,
44         18, 100,
45         18, 91,
46         18, 91,
47         100, 101
48     ])
49
50 # -----
51 # Drawing function
52 # -----
53 def draw(x):
54     plt.clf()
55     points = create_control_points(x)
56     t_points = np.linspace(0, 1, 100)
57
58     # Create curve
59     full_curve = Slider.Curve(x, points, points)
60     curve1 = Slider.Curve(x, points, points[0:1])
61     curve2 = Slider.Curve(x, points, points[0:2])
62     curve3 = Slider.Curve(x, points, points[0:3])
63     curve4 = Slider.Curve(x, points, points[0:4])
64
65     # Plot Slider curve
66     plt.plot(curve1, 0, curve2, 1, 'r', linestyle='solid', label="2 Point Slider")
67     plt.plot(curve2, 0, curve3, 1, 'r', linestyle='solid')
68     plt.plot(curve3, 0, curve4, 1, 'r', linestyle='solid')
69     plt.plot(curve4, 0, curve1, 1, 'r', linestyle='solid')
70
71     plt.plot(full_curve, 0, full_curve, 1, 'r', linestyle='solid', label="Single slider (100%)")
72
73     # Plot control points
74     plt.plot(points, 0, points, 1, 'r', linestyle='solid', label="Control Points")
75
76     # Draw reference circle
77     circle = plt.Circle((100, 100), 50, color='gray', linestyle='solid', label="Reference Circle")
78     plt.gca().add_artist(circle)
79
80     # Text area
81     ax.set_xlabel('x')
82     ax.set_ylabel('y')
83     ax.set_aspect('equal')
84     ax.grid(True)
85
86     # Add legend box
87     ax.legend(loc='upper left', bbox_to_anchor=(0.4, 1), border=0)
88
89     fig.canvas.draw_idle()
90
91 # -----
92 # Plot setup
93 # -----
94 fig, ax = plt.subplots(figsize=(7, 7))
95 plt.subplots_adjust(bottom=0.15, top=0.75) # extra room for slider and legend
96 draw(0)
97
98 # -----
99 # Slider
100 # -----
101 ax_slider = plt.axes([0.25, 0.1, 0.45, 0.05])
102 slider = Slider(ax_slider, 'x', 0, 100, 0.01)
103
104 # Update plot
105 slider.on_changed(draw)
106
107 plt.show()
```

## **Solution:**

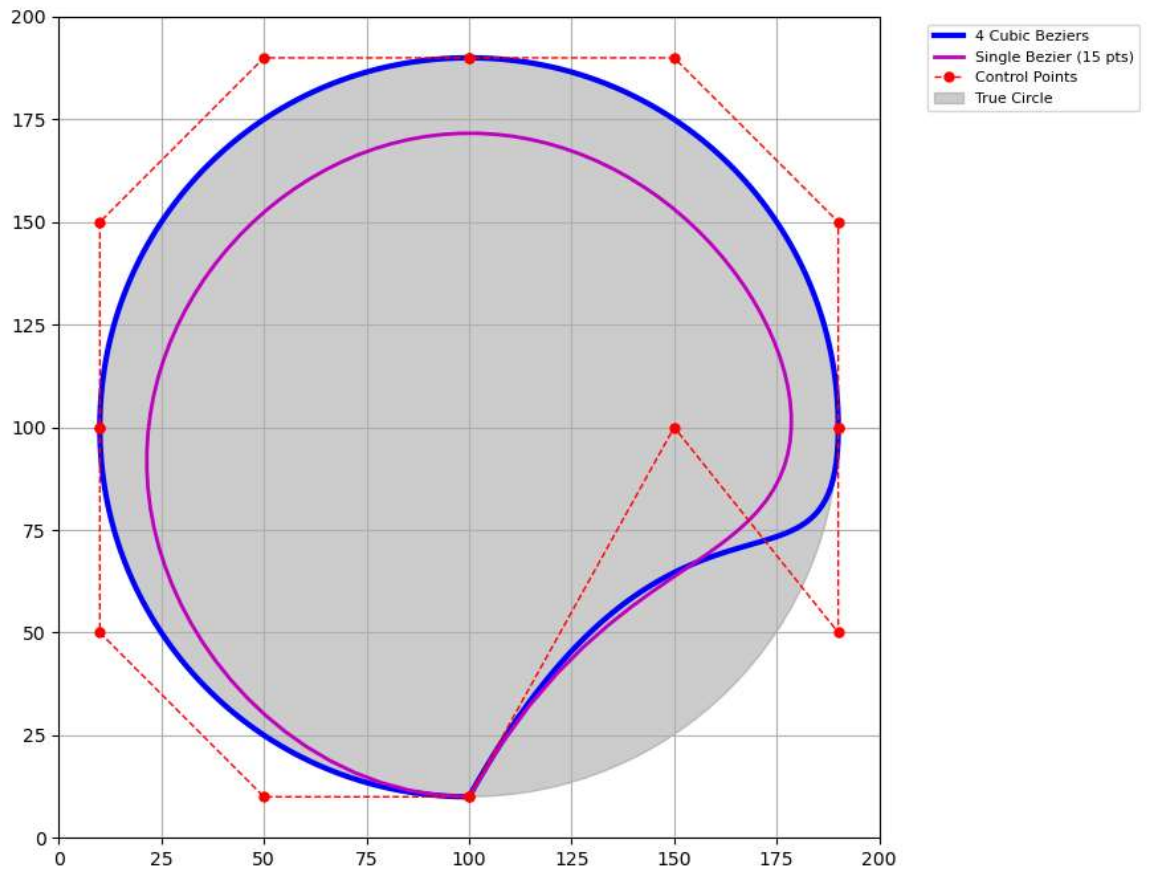


P1 Y  10

- ♦ Red circles are the control points.
  - ♦ Blue circle is the 4 piecewise Bezier curves.
  - ♦ Magenta circle is the 1 single Bezier curve.
  - ♦ Gray shading represents the circle drawn using high level commands using built in function.
-

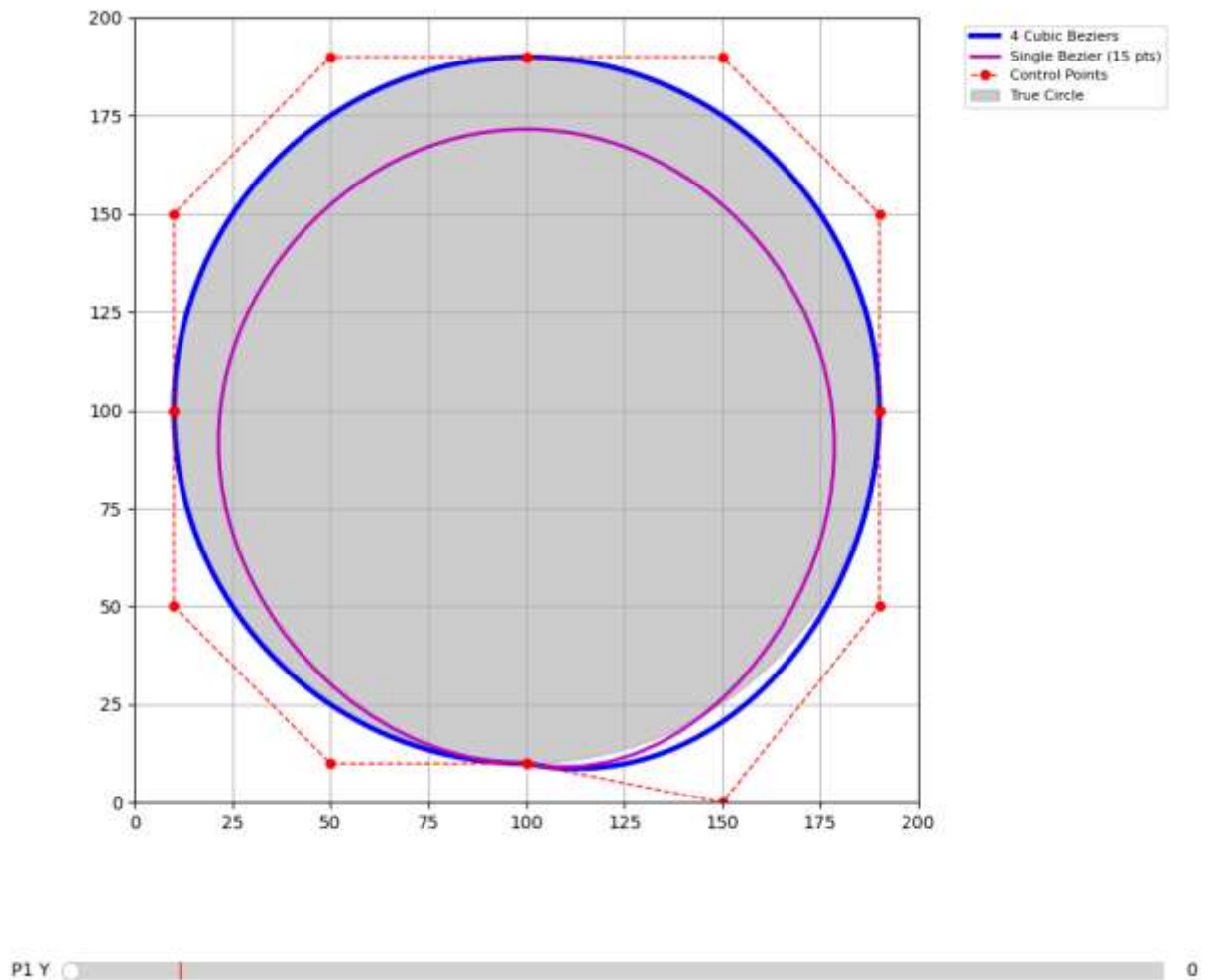
I. Move point P1 vertically (up and down) and observe the effect on the drawings in cases a) and b) above. What do you conclude?

→ When increasing the y coordinate of P1 (moving up):



Y  100

→ When decreasing the y coordinate of P1 (moving down):



## **Conclusion:**

- Higher order Bezier curve has no local control.
- To solve this → 4 piecewise Bezier Curves.

**Note that** → The circle with 4 piecewise Bezier curves is more accurate and closer to the original circle than the circle with one single Bezier Curve because it passes through the 4 main points that passes through the radius.