

Instituto Tecnológico de Morelia

práctica 5: Curvas de Beizer

Graficación

Profesor:

Martinez Guzman Bryan Eduardo

Santiago Gonzalez Lara 22121360

Mayavi Sarael Gonzalez Ornelas 22120661

11 de febrero del 2025

Las **curvas de Bézier** son una herramienta matemática usada para dibujar formas suaves y curvas en gráficos por computadora, diseño, animación, y más. Fueron desarrolladas por Pierre Bézier, un ingeniero de Renault, para diseñar formas de carros. Estas curvas son populares porque son fáciles de calcular y manipular.

¿Cómo funcionan?

Las curvas de Bézier se construyen usando **puntos de control**, que son como "anclas" que definen la forma de la curva. Dependiendo de cuántos puntos de control tengas, obtendrás curvas más o menos complejas:

1. **Curva lineal (grado 1):** Es solo una línea recta entre dos puntos.
2. **Curva cuadrática (grado 2):** Tiene tres puntos de control. La curva se dobla suavemente hacia el punto medio, creando una curva más flexible.
3. **Curva cúbica (grado 3):** Usa cuatro puntos de control y permite crear formas aún más complejas.

El movimiento de la curva entre los puntos se calcula con un proceso llamado **interpolación**, usando una fórmula basada en el **teorema del binomio**.

Aplicaciones prácticas

1. **Gráficos y diseño digital:**
 - Se usan en programas como Adobe Illustrator o Inkscape para dibujar formas vectoriales.
 - Permiten diseñar logos, tipografía y gráficos escalables sin pérdida de calidad.
2. **Animación:**
 - En animación 2D/3D, las curvas de Bézier ayudan a mover personajes o cámaras de forma suave.
 - Son esenciales en herramientas como After Effects para crear trayectorias de movimiento.
3. **Diseño de interfaces (UI/UX):**
 - Usadas para definir transiciones y animaciones, como deslizamientos o desvanecimientos suaves en aplicaciones y sitios web.
4. **Edición de video y audio:**
 - En edición de video, controlan cómo cambian las velocidades o efectos visuales a lo largo del tiempo.
 - En audio, permiten ajustar volúmenes o mezclas de manera precisa.

5. Videojuegos:

- Crean trayectorias para personajes o cámaras en mundos 2D/3D.

6. Modelado 3D:

- Sirven para modelar superficies suaves, como carrocerías de autos o diseños industriales.

```
import matplotlib.pyplot as plt
import numpy as np
import math # Importar el módulo math para usar math.factorial

def bezier_curve(points, num_points=100):
    n = len(points) - 1
    t = np.linspace(0, 1, num_points)
    curve = np.zeros((num_points, 2))

    for i in range(n + 1):
        curve += np.outer(bernstein_poly(n, i, t), points[i])

    return curve

def bernstein_poly(n, i, t):
    return comb(n, i) * (t ** i) * (1 - t) ** (n - i)

def comb(n, k):
    return math.factorial(n) / (math.factorial(k) * math.factorial(n - k)) # Usar math.factorial

def plot_bezier_curve(points):
    curve = bezier_curve(points)

    plt.figure()
    plt.plot(curve[:, 0], curve[:, 1], 'b-', label='Curva de Bézier')
    plt.plot([p[0] for p in points], [p[1] for p in points], 'ro-',
label='Puntos de Control')
    plt.legend()
    plt.title('Curva de Bézier')
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.grid(True)
    plt.show()

# Ejemplo de uso
points = np.array([[0, 0], [1, 3], [4, 4], [5, 0]])
```

```
plot_bezier_curve(points)
```

