B-Splines, Rational Curves & Surface Modelling

Comprehensive Lecture for B.Tech Mechanical/CS Engineering

Dinesh Kumar
Assistant Professor

School of Continuing Education
D.Y. Patil International University (DYPIU)

February 17, 2026

Lecture Roadmap

Part 1: B-Splines

- Limitations of Bézier
- Cox-de Boor Recursion
- Knot Vectors
- Numerical Examples

Part 2: Rational Curves

- The need for Weights
- Conic sections (Circles)

Part 3: Surface Modelling

- Tensor Products
- Control Nets
- Patches and Continuity

Part 4: Applications

- CAD/CAM Use Cases

1.1 Why move beyond Bézier Curves?

- **Global Control Issue:** In Bézier, moving one point P_i changes the whole curve. This makes detailed editing impossible.
- **Degree Issue:** If you have 10 control points, the Bézier degree is 9. High-degree polynomials "wiggle" too much (oscillations).
- **The B-Spline Fix:** We use "Basis Splines."
- **Local Control:** Moving a point only affects a small segment.

1.2 Mathematical Definition

The curve $\mathrm{C}(u)$ is defined as:

$$\mathrm{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathrm{P}_i$$

The Key Variables:

- $n + 1$: Number of control points.
- p: Degree of the curve (e.g., $p = 2$ for quadratic).
- $N_{i,p}(u)$: The **Basis Functions** (the "blending" weights).
- U: The **Knot Vector** (the "joints" of the curve).

1.3 Basis Functions (Cox-de Boor)

Basis functions are built from the bottom up (Degree 0 to Degree p):

Step 1: Degree 0 (Step Functions)

$N_{i,0}(u) = 1$ if $u_i \leq u < u_{i+1}$, else 0.

Step 2: Higher Degrees (Recursion)

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Interpretation: $N_{i,p}$ is only "active" over a small range of u. This is the secret to local control.

1.4 Understanding the Knot Vector

The Knot Vector $U = \{u_0, u_1, \ldots, u_m\}$ tells the curve where segments begin and end.

- **Relation:** $m = n + p + 1$.
- **Uniform Knots:** $\{0, 1, 2, 3, 4, 5\}$. Equal spacing.
- **Open/Clamped Knots:** $\{0, 0, 0, 1, 2, 3, 3, 3\}$.
- **B.Tech Point:** Clamping ensures the curve starts exactly at P_0 and ends at P_n.

1.5 Local Control Intuition

- Each $N_{i,p}$ basis function has a "support" of $p + 1$ knot spans.
- If you change control point P_i, the curve **only** changes between u_i and u_{i+p+1}.
- Everywhere else, the curve stays exactly the same.

Comparison:

- **Bézier:** Changing P_0 affects the curve at $t = 0.99$.
- **B-Spline:** Changing P_0 usually has **zero** effect at $u = 0.99$.

1.6 Numerical Example 1: Linear B-Spline ($p = 1$)

Given: $P_0(0,0), P_1(2,4), P_2(4,0)$. $U = \{0,0,1,2,2\}$. **Find** C(0.5):

1. At $u = 0.5$, we are in the interval $[u_1, u_2) = [0,1)$.
2. Use recursion: $N_{0,1}(0.5) = 0.5$ and $N_{1,1}(0.5) = 0.5$.
3. $C(0.5) = 0.5(0,0) + 0.5(2,4) = (1,2)$.

Result: At $u = 0.5$, the curve is exactly at the midpoint of the first leg of the control polygon.

1.7 Continuity and Smoothness

- At a "simple knot" (one that appears once), the curve is C^{p-1} continuous.
- Example: A quadratic B-spline ($p = 2$) is C^1 continuous (smooth slope).
- **Knot Multiplicity:** If we repeat a knot k times, continuity decreases to C^{p-k}.
- If $k = p$, we get a "Cusp" (sharp corner).

2.1 The Need for Rational Curves

- **Polynomial Limitation:** Standard B-splines can only represent parabolic shapes. They **cannot** draw a circle.
- **The Fix:** We use a ratio of polynomials.
- **NURBS:** Non-Uniform Rational B-Splines.
- This is the industry standard in CAD (SolidWorks, CATIA, Rhino).

2.2 The Role of Weights

NURBS formula:

$$C(u) = \frac{\sum w_i N_{i,p}(u) P_i}{\sum w_j N_{j,p}(u)}$$

- w_i is the **Weight** of control point P_i.
- $w_i > 1$: Stronger pull.
- $w_i < 1$: Weaker pull.
- $w_i = 1$: Standard B-spline.

2.3 Representing a Circle (Numerical)

How to represent a 90° arc?

1. **Points:** $P_0(1,0), P_1(1,1), P_2(0,1)$.
2. **Weights:** $w_0 = 1, w_2 = 1$.
3. **Magic Weight:** $w_1 = \cos(45^\circ) = 1/\sqrt{2} \approx 0.707$.

By changing w_1, we change the conic:

- $w_1 < 0.707$: Ellipse
- $w_1 = 0.707$: **Circle**
- $w_1 > 0.707$: Hyperbola

2.4 Homogeneous Coordinates

- In 2D, we see (x, y).
- In Computer Graphics, we use (wx, wy, w).
- This makes projections and rotations much easier mathematically.
- A NURBS curve in 3D is actually a non-rational B-spline in 4D!

3.1 From Curves to Surfaces

- Curve: 1 Parameter (u).
- Surface: 2 Parameters (u, v).
- Think of a surface as a "family of curves."
- As you change v, you get a different curve in u.

3.2 Tensor Product Surface

We combine two basis functions (one for u, one for v):

$$\mathrm{S}(u,v) = \sum_i \sum_j N_{i,p}(u) M_{j,q}(v) \mathrm{P}_{i,j}$$

- $\mathrm{P}_{i,j}$ is a **Control Net** (a 2D grid of points).
- The surface "follows" the shape of the net.

3.3 Surface Patches

- **Bézier Patch:** Controlled by a fixed grid (usually 4×4).
- **B-Spline Patch:** Uses knots in both directions.
- **Local Control:** Moving one point in the 10×10 grid only dents a small part of the sheet.

3.4 Joining Patches (Continuity)

Engineering parts are complex. We join patches at their edges.

- G^0: Edge meets edge (no gap).
- G^1: The surface looks smooth (tangents are aligned).
- G^2: The surface reflects light perfectly (curvature is continuous). *Essential for car design!*

3.5 Surface Evaluation Process

How does a GPU draw a point $S(0.5, 0.5)$?

1. Calculate all u-basis functions at 0.5.
2. Calculate all v-basis functions at 0.5.
3. Multiply them with the (i, j) control points.
4. Sum them up!

4.1 Comparison Table

Feature	**Bézier**	**B-Spline**	**NURBS**
Local Control	No	Yes	Yes
Exact Circle	No	No	**Yes**
Degree Control	No	Yes	Yes
Industry Use	Graphic Art	CAD/CAM	**Advanced CAD**

4.2 Engineering Use Cases

- **Aerospace:** Modelling wing airfoils (NURBS surfaces).
- **Automotive:** Designing car hoods for better aerodynamics and aesthetics.
- **Bio-medical:** Fitting surfaces to MRI/CT scan data of bones.
- **Robotics:** Generating smooth paths for robot arms to avoid jerky movements.

Summary & Key Takeaways

1. **B-Splines** are piecewise polynomials $\rightarrow$ better control.
2. **Knots** act as the "bones" of the curve.
3. **Weights** in NURBS allow us to draw perfect circles/conics.
4. **Surfaces** are created using a 2D control net grid.

Questions?

`dinesh.kumar@dypiu.edu.in`