13.472J/1.128J/2.158J/16.940J
COMPUTATIONAL GEOMETRY

Lectures 4 and 5

N. M. Patrikalakis

Massachusetts Institute of Technology
Cambridge, MA 02139-4307, USA

Contents

Reading in the Textbook

- Chapter 1, pp.6 - pp.33

Lecture 4

Introduction to Spline Curves

4.1 Introduction to parametric spline curves

Parametric formulation

$$x = x(u),\ y = y(u),\ z = z(u) \quad or \quad \mathbf{R} = \mathbf{R}(u) \quad \text{(vector notation)}$$

Usually applications need a finite range for u (e.g. $0 \leq u \leq 1$).

For free-form shape creation, representation, and manipulation the *parametric representation* is preferrable, see Table 1.2 in textbook. Furthermore, we will use *polynomials* for the following reasons:

- Cubic polynomials are good approximations of *physical splines*. (Historical note: Shape of a long flexible beam constrained to pass through a set of points → *Draftsman's Splines*).
- Parametric polynomial *cubic* spline curves are the "smoothest" curves passing through a set of points; (i.e. they minimize the bending strain energy of the beam $\propto \int_0^L \kappa^2 ds$).

4.2 Elastic deformation of a beam in bending

Within the Euler beam theory:

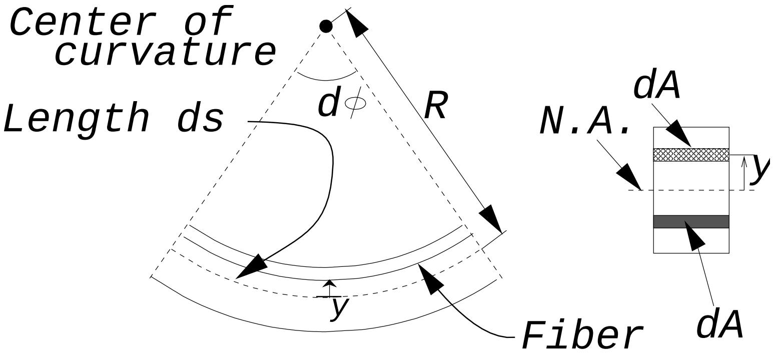

Figure 4.1: Differential segment of an Euler beam.

- Plane sections of the beam normal to the neutral axis (N.A.) remain plane and normal to the neutral axis after deformation.
- Elongation strain of a fiber at distance y above the N.A.:

$$\epsilon = \frac{(-y+R)d\phi - Rd\phi}{Rd\phi} = -\frac{y}{R}$$

- The radius of curvature can be related to the differential angle and arc length:

$$Rd\phi = ds \Rightarrow \kappa = \frac{1}{R} = \frac{d\phi}{ds}$$

- The stress experienced by a fiber at distance y from the N.A. can be found from Hooke's law relating stress and strain.

$$\sigma = E\epsilon = -E\frac{y}{R}$$

- The bending moment about the N.A. can be calculated by integrating the stress due to bending times the moment arm over the surface of the cross-section:

$$M = \int_A \sigma y dA = -\frac{E}{R}\int_A y^2 dA = -EI\frac{1}{R} = -EI\frac{d\phi}{ds} = -EI\kappa.$$

- For small deflections, the displacement $Y = Y(x)$ of the N.A., the following approximations can be made:

$$\phi \approx \frac{dY}{dx}, \quad ds \approx dx, \quad \frac{d\phi}{ds} \approx \frac{d^2Y}{dx^2}$$

- Therefore,

$$-\frac{M}{EI} \approx \frac{d^2Y}{dx^2} \tag{4.1}$$

- Consider a beam with simple supports and no distributed loads, as shown in Figure 4.2.

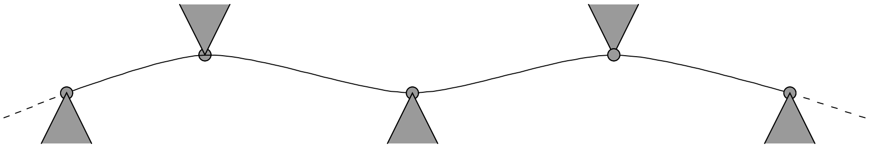

Figure 4.2: Simply supported beam.

Looking at a section of the beam between two supports (Figure 4.3), the bending moment can be expressed as a *linear* equation:

$$M = A_0 + A_1 x \tag{4.2}$$

where A_0 and A_1 are the bending moment and shearing force at $x = 0$ in Figure 4.3.

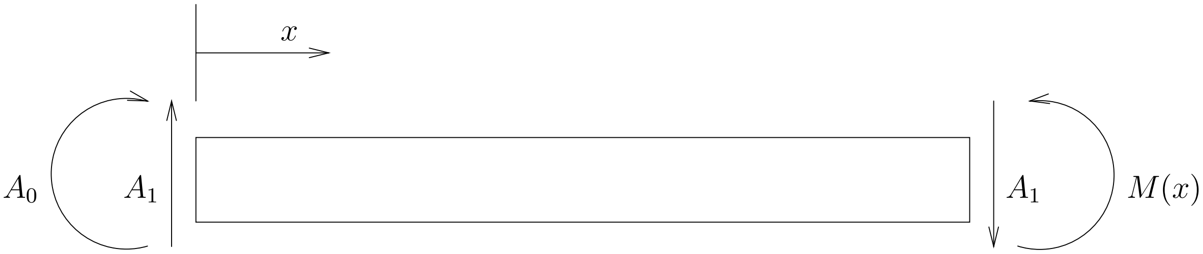

Figure 4.3: Section of simply supported beam between pins. The shear force and bending moment at the ends of the section are illustrated.

- Upon substituting Equation 4.2 into Equation 4.1 and integrating, we get:

$$\phi \cong \frac{dY}{dx} \cong -\frac{1}{EI}[A_0 x + A_1 \frac{x^2}{2}] + A_2$$
$$Y(x) \cong C_0 + C_1 x + C_2 x^2 + C_3 x^3 \quad (cubic)$$

Therefore, in order to replicate the shape of *physical splines*, the CAD community developed shape representation methods based on *cubic polynomials*.

Generically, polynomials have the following additional advantages:

- easy to store as sequences of coefficients;
- efficient to compute and trace efficiently;
- easy to differentiate, integrate, and adapt to matrix and vector algebra; and
- easy to piece together to construct composite curves with a certain order of continuity, a feature important in increasing complexity of a curve or surface.

4.3 Parametric polynomial curves

4.3.1 Ferguson representation

In 1963, Ferguson at Boeing developed a polynomial representation of space curves:

$$\mathbf{R}(u) = \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2 + \mathbf{a}_3 u^3 \tag{4.3}$$

where $0 \leq u \leq 1$ by convention. Note there are 12 coefficients, $\mathbf{a}_i$, defining the curve $\mathbf{R}(u)$. This representation is also known as the *power* basis or *monomial* form.

The coefficients, $\mathbf{a}_i$, are difficult to interpret geometrically, so we can express $\mathbf{a}_i$ in terms of $\mathbf{R}(0)$, $\mathbf{R}(1)$, $\dot{\mathbf{R}}(0)$, and $\dot{\mathbf{R}}(1)$ (where $\dot{\mathbf{R}}$ denotes derivative with respect to u):

$$\begin{aligned}
\mathbf{R}(0) &= \mathbf{a}_0 \\
\mathbf{R}(1) &= \mathbf{a}_0 + \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3 \\
\dot{\mathbf{R}}(0) &= \mathbf{a}_1 \\
\dot{\mathbf{R}}(1) &= \mathbf{a}_1 + 2\mathbf{a}_2 + 3\mathbf{a}_3
\end{aligned}$$

Solving the above equations yields expression for the coefficients, $\mathbf{a}_i$'s, in terms of the geometric end conditions of the curve.

$$\begin{aligned}
\mathbf{a}_0 &= \mathbf{R}(0) \\
\mathbf{a}_1 &= \dot{\mathbf{R}}(0) \\
\mathbf{a}_2 &= 3[\mathbf{R}(1) - \mathbf{R}(0)] - 2\dot{\mathbf{R}}(0) - \dot{\mathbf{R}}(1) \\
\mathbf{a}_3 &= 2[\mathbf{R}(0) - \mathbf{R}(1)] + \dot{\mathbf{R}}(0) + \dot{\mathbf{R}}(1)
\end{aligned}$$

4.3.2 Hermite-Coons curves

By substituting the coefficients into the Ferguson representation, we can rewrite Equation 4.3 as:

$$\mathbf{R}(u) = \mathbf{R}(0)\eta_0(u) + \mathbf{R}(1)\eta_1(u) + \dot{\mathbf{R}}(0)\eta_2(u) + \dot{\mathbf{R}}(1)\eta_3(u) \tag{4.4}$$

where

$$\begin{aligned}
\eta_0(u) &= 1 - 3u^2 + 2u^3 \\
\eta_1(u) &= 3u^2 - 2u^3 \\
\eta_2(u) &= u - 2u^2 + u^3 \\
\eta_3(u) &= u^3 - u^2
\end{aligned}$$

The new basis functions, $\eta_i(u)$, are known as Hermite polynomials or blending functions, see Figure 4.4. They were first used for 3D curve representation in a computer environment in the 60's by the late *Steven Coons*, an MIT professor, participant in the famous ARPA project MAC.

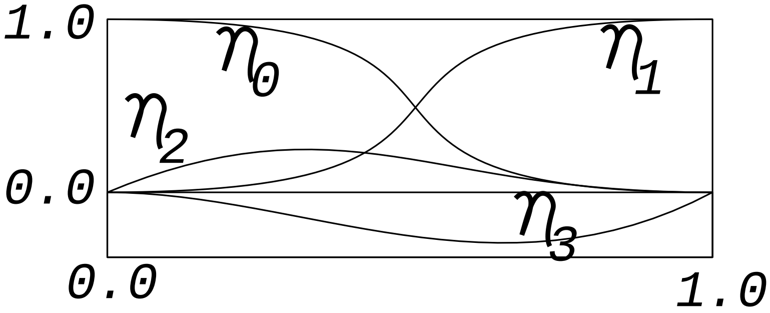

Figure 4.4: Plot of Hermite basis functions.

Note that these basis functions, $\eta_i(u)$, satisfy the following boundary conditions, see Figure 4.4:

$$\begin{gathered}
\eta_0(0) = 1, \eta_0(1) = \eta_0'(0) = \eta_0'(1) = 0; \\
\eta_1(1) = 1, \eta_1(0) = \eta_1'(0) = \eta_1'(1) = 0; \\
\eta_2'(0) = 1, \eta_2(0) = \eta_2'(1) = \eta_2'(1) = 0; \\
\eta_3'(1) = 1, \eta_3(0) = \eta_3'(0) = \eta_3'(1) = 0.
\end{gathered}$$

These boundary conditions also allow the computation of the cubic Hermite polynomials, $\eta_i(u)$, by setting up and solving a system of 16 linear equations in the coefficients of these polynomials.

Geometric Interpretation of Hermite-Coons curves.

If we make the following substitutions:

$$\begin{aligned}\dot{\mathbf{R}}(0) &= \alpha_0 \mathbf{t}(0)\\ \dot{\mathbf{R}}(1) &= \alpha_1 \mathbf{t}(1)\end{aligned}$$

where $\mathbf{t}(u)$ is the unit tangent of the curve, the following observations can be made from Figure 4.5, relating the coefficients α_0 and α_1 to the shape of the curve:

- $\alpha_0 \uparrow, \ \alpha_1 = \text{const} \Rightarrow$ bias towards $\mathbf{t}(0)$
- $\alpha_1 \uparrow, \ \alpha_0 = \text{const} \Rightarrow$ bias towards $\mathbf{t}(1)$
- $\alpha_0, \ \alpha_1 \uparrow \Rightarrow$ increase fullness
- $\alpha_0, \ \alpha_1$ both LARGE $\Rightarrow$ cusp forms ($\dot{\mathbf{R}}(u^*) = 0$) or self-intersection occurs

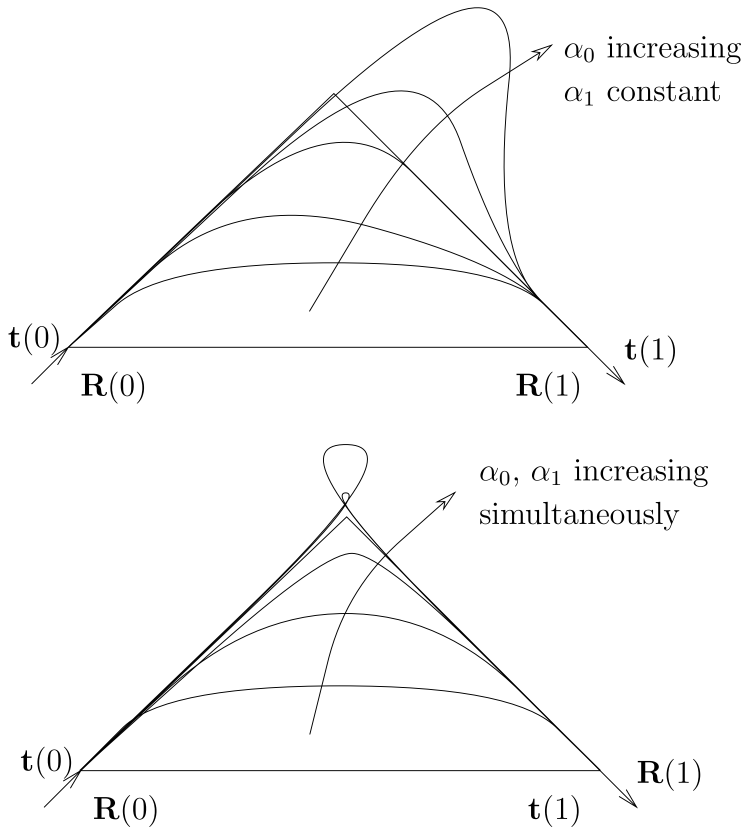

Figure 4.5: Fullness of a Hermite-Coons curve.

Due to the importance of choosing α_0 and α_1 appropriately, the Hermite-Coons representation can be difficult for designers to use efficiently, but is much easier to understand than the Ferguson or monomial form.

4.3.3 Matrix forms and change of basis

- Monomial or Power or Ferguson form

$$\mathbf{R}(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_3 \\ \mathbf{a}_2 \\ \mathbf{a}_1 \\ \mathbf{a}_0 \end{bmatrix} = \mathbf{U} \cdot \mathbf{F}_M$$

- Hermite-Coons

$$\mathbf{R}(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}(0) \\ \mathbf{R}(1) \\ \dot{\mathbf{R}}(0) \\ \dot{\mathbf{R}}(1) \end{bmatrix} = \mathbf{U} \cdot \mathbf{M}_H \cdot \mathbf{F}_H$$

Conversion between the two representations is simply a matter of matrix manipulation:

$$\mathbf{U} \cdot \mathbf{F}_M = \mathbf{U} \cdot \mathbf{M}_H \cdot \mathbf{F}_H$$

Hence,

$$\begin{aligned} \mathbf{F}_M &= \mathbf{M}_H \cdot \mathbf{F}_H \\ \mathbf{F}_H &= \mathbf{M}_H^{-1} \cdot \mathbf{F}_M \end{aligned}$$

4.3.4 Bézier curves

Bézier developed a reformulation of Ferguson curves in terms of *Bernstein polynomials* for the UNISURF System at Renault in France in 1970. This formulation is expressed mathematically as follows:

$$\mathbf{R}(u) = \sum_{i=0}^{n} \mathbf{R}_i B_{i,n}(u) \qquad 0 \leq u \leq 1$$

$\mathbf{R}_i$ and $B_{i,n}(u)$ represent polygon vertices and the Bernstein polynomial basis functions respectively. The definition of a Bernstein polynomial is:

$$\begin{aligned} B_{i,n}(u) &= \begin{pmatrix} n \\ i \end{pmatrix} u^i (1-u)^{n-i} \quad i = 0, 1, 2, ...n \\ &\text{where } \begin{pmatrix} n \\ i \end{pmatrix} = \frac{n!}{i!(n-i)!} \end{aligned}$$

The polygon joining $\mathbf{R}_0, \mathbf{R}_1, ..., \mathbf{R}_n$ is called the *control polygon.*

Examples of Bézier curves:

- n=2: Quadratic Bézier Curves (Parabola), see Figures 4.6 and Figure 4.7

$$\begin{aligned} \mathbf{R}(u) &= \mathbf{R}_0 \, (1-u)^2 + \mathbf{R}_1 \, 2u(1-u) + \mathbf{R}_2 \, u^2 \\ &= \begin{bmatrix} u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \end{aligned}$$

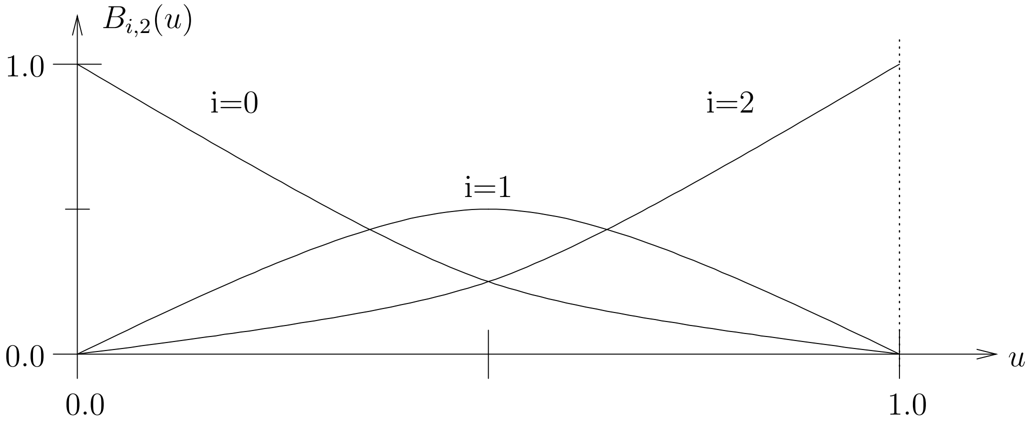

Figure 4.6: Plot of the quadratic Bernstein basis functions.

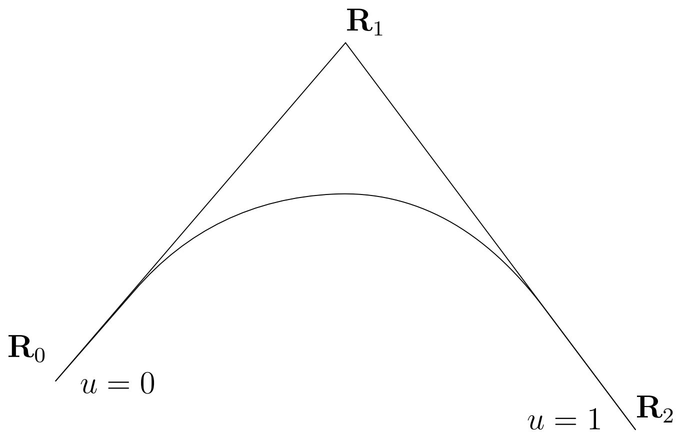

Figure 4.7: Illustration of end conditions for a quadratic Bézier curve.

- n=3: Cubic Bézier Curves, see Figures 4.8 and 4.9

$$\begin{aligned}\mathbf{R}(u) &= \mathbf{R}_0\ (1-u)^3 + \mathbf{R}_1\ 3u(1-u)^2 + \mathbf{R}_2\ 3u^2(1-u) + \mathbf{R}_3\ u^3 \\ &= \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3 \end{bmatrix}\end{aligned}$$

- Interpolation of $\mathbf{R}_0$, $\mathbf{R}_n$ and tangency of curve with polygon at $u = 0, 1$:
- The Bézier curve approximates and smooths control polygon.

Properties of Bernstein polynomials

1. *Positivity* $B_{i,n}(u) \geq 0$ in $0 \leq u \leq 1$

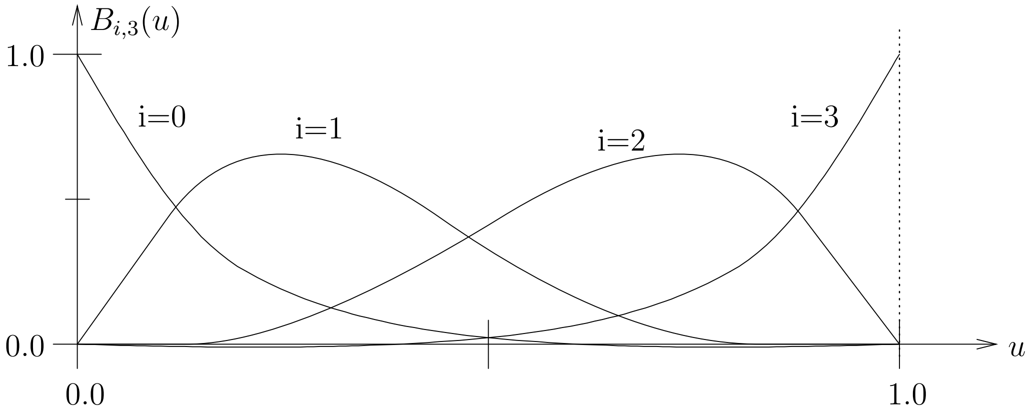

Figure 4.8: Plot of the cubic Bernstein basis functions.

2. *Partition of unity* $\sum_{i=0}^{n} B_{i,n}(u) = [1-u+u]^n = 1$ (by the *binomial* theorem)

3. *Recursion*

$$\begin{aligned} B_{i,n}(u) &= (1-u)B_{i,n-1}(u) + uB_{i-1,n-1}(u) \\ \textit{with} &\ \ B_{i,n}(u) = 0 \textbf{ for } i<0,\ i>n \\ &\ \ \textbf{and } \ B_{0,0}(u) = 1 \end{aligned}$$

4. *Linear Precision Property*

$$u = \sum_{i=0}^{n} \frac{i}{n} B_{i,n}(u)$$

   - Conversion of explicit curves to parametric Bézier curves.
   - Parametrization of straight lines

$$y = y(x) \rightarrow x = u; y = y(u)$$

5. *Degree elevation:* The basis functions of degree n can be expressed in terms of those of degree $n+1$ as:

$$B_{i,n}(u) = \left(1 - \frac{i}{n+1}\right) B_{i,n+1}(u) + \frac{i+1}{n+1} B_{i+1,n+1}(u), \quad i = 0, 1, \cdots, n$$

6. *Symmetry* $B_{i,n}(u) = B_{n-i,n}(1-u)$

7. *Derivative* $B'_{i,n}(u) = n[B_{i-1,n-1}(u) - B_{i,n-1}(u)]$ where $B_{-1,n-1}(u) = B_{n,n-1}(u) = 0$

8. *Basis conversion*
(for each of the x, y, z, coordinates)

   - Ferguson or monomial form:

$$f(u) = \sum_{i=0}^{3} f_i u^i = \left[\begin{array}{cccc} u^3 & u^2 & u & 1 \end{array}\right] \left[\begin{array}{c} f_3 \\ f_2 \\ f_1 \\ f_0 \end{array}\right] = \mathbf{U} \cdot \mathbf{F}_M$$

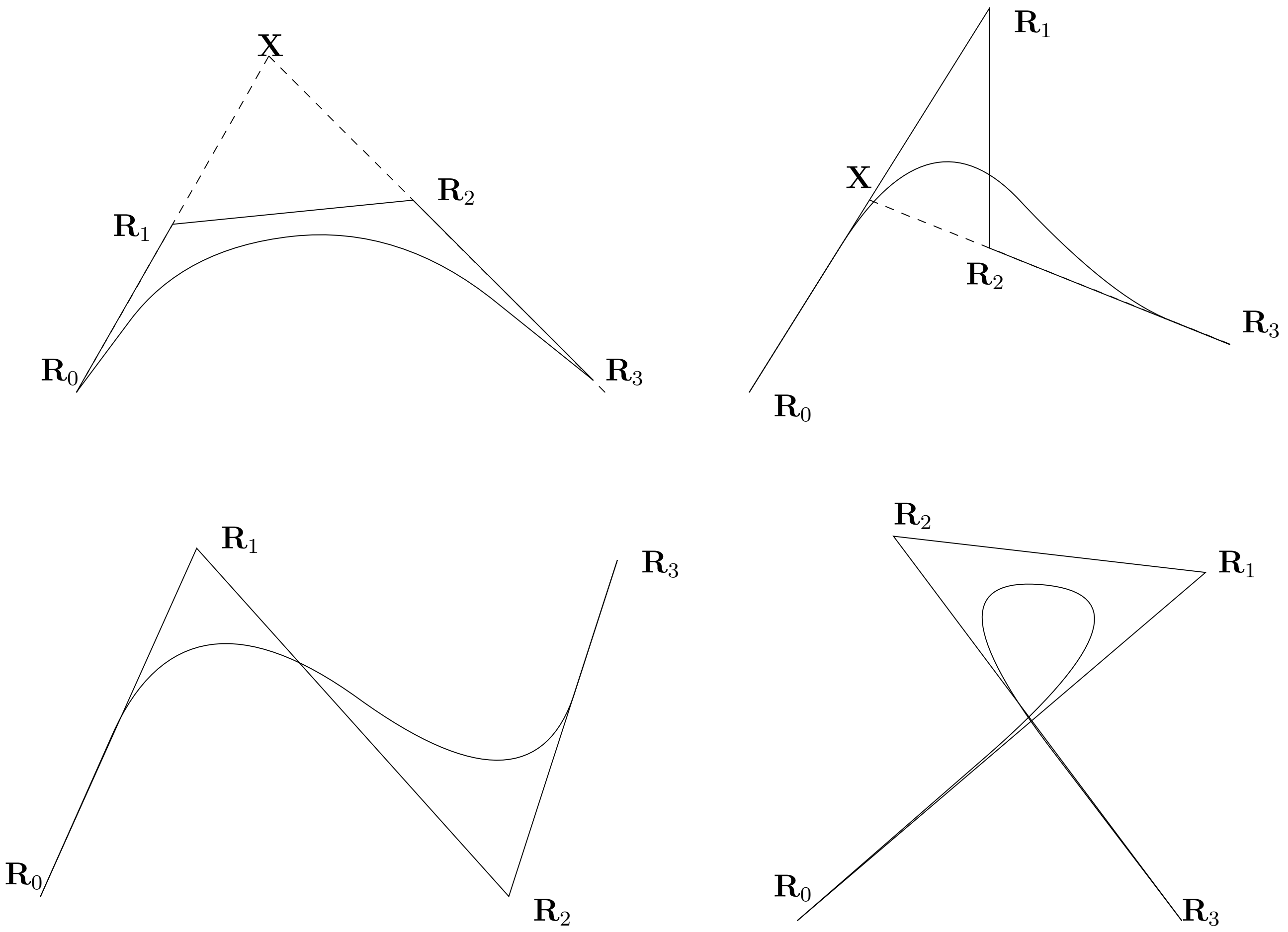

Figure 4.9: Examples of cubic Bézier curves approximating their control polygons.

- Hermite form:

$$f(u) = \left[\begin{array}{cccc} u^3 & u^2 & u & 1 \end{array}\right] \cdot \mathbf{M}_H \cdot \left[\begin{array}{c} f(0) \\ f(1) \\ \dot{f}(0) \\ \dot{f}(1) \end{array}\right] = \mathbf{U} \cdot \mathbf{M}_H \cdot \mathbf{F}_H$$

- Bézier form:

$$f(u) = \left[\begin{array}{cccc} u^3 & u^2 & u & 1 \end{array}\right] \cdot \mathbf{M}_B \cdot \left[\begin{array}{c} F_0 \\ F_1 \\ F_2 \\ F_3 \end{array}\right] = \mathbf{U} \cdot \mathbf{M}_B \cdot \mathbf{F}_B$$

- Conversion between forms:

$$\begin{aligned} \mathbf{U} \cdot \mathbf{F}_M = \mathbf{U} \cdot \mathbf{M}_H \cdot \mathbf{F}_H = \mathbf{U} \cdot \mathbf{M}_B \cdot \mathbf{F}_B \\ \Rightarrow \mathbf{F}_M = \mathbf{M}_H \cdot \mathbf{F}_H = \mathbf{M}_B \cdot \mathbf{F}_B \\ \text{or } \Rightarrow \mathbf{F}_H = \mathbf{M}_H^{-1} \cdot \mathbf{F}_M = \mathbf{M}_H^{-1} \cdot \mathbf{M}_B \cdot \mathbf{F}_B \\ \text{etc.} \end{aligned}$$

Properties of Bézier curve:

1. *Geometric conditions at ends*

   - $\mathbf{R}(0) = \mathbf{R}_0$; $\mathbf{R}(1) = \mathbf{R}_n$
   - $\dot{\mathbf{R}}(0) = n(\mathbf{R}_1 - \mathbf{R}_0)$; $\dot{\mathbf{R}}(1) = n(\mathbf{R}_n - \mathbf{R}_{n-1})$
   - $\ddot{\mathbf{R}}(0) = n(n-1)[(\mathbf{R}_2 - \mathbf{R}_1) - (\mathbf{R}_1 - \mathbf{R}_0)]$
   - $\ddot{\mathbf{R}}(1) = n(n-1)[(\mathbf{R}_n - \mathbf{R}_{n-1}) - (\mathbf{R}_{n-1} - \mathbf{R}_{n-2})]$
   - $$\kappa = \frac{1}{\rho} = \frac{|\dot{\mathbf{R}} \times \ddot{\mathbf{R}}|}{|\dot{\mathbf{R}}|^3}$$

2. *Curve symmetry*
   $\mathbf{R}^*_i = \mathbf{R}_{n-i} \iff \mathbf{R}^*(u) = \mathbf{R}(1-u)$, see Figure 4.10

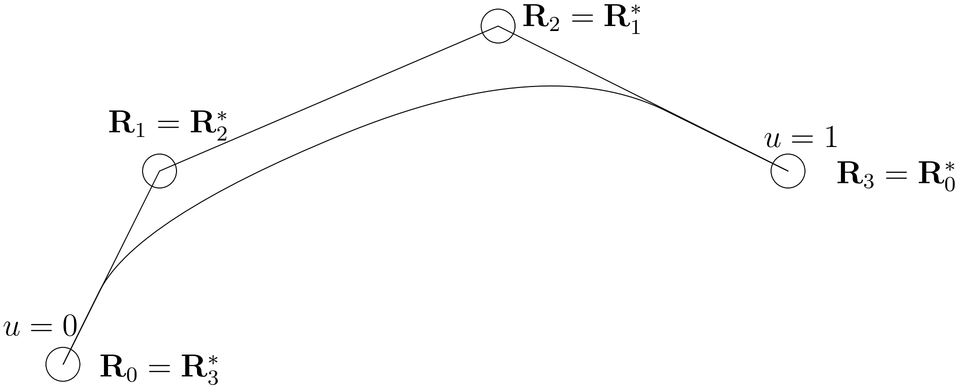
$\mathbf{R}_2 = \mathbf{R}^*_1$
$\mathbf{R}_1 = \mathbf{R}^*_2$
$u = 1$
$\mathbf{R}_3 = \mathbf{R}^*_0$
$u = 0$
$\mathbf{R}_0 = \mathbf{R}^*_3$

Figure 4.10: Bézier curve symmetry

3. *Bounding box* $\min(x_i) \leq \sum_{i=0}^{n} x_i B_{i,n}(u) \leq \max(x_i)$, etc, see Figure 4.11. A tighter bounding box can usually be obtained by axis reorientation. Bounding boxes are useful in interference and intersection problems in CAD.

4. *Geometry invariance properties*

   (a) Translation:
   Recalling that $\sum_{i=0}^{n} B_{i,n}(u) = 1$, we can translate the whole curve by $\mathbf{R}_0$ and translate each control vertex back by $-\mathbf{R}_0$ and show that our resulting curve is the same as the original, see Figure 4.12

   $$\mathbf{R}(u) = \mathbf{R}_0 + \sum_{i=0}^{n} (\mathbf{R}_i - \mathbf{R}_0) B_{i,n}(u) = \sum_{i=0}^{n} \mathbf{R}_i B_{i,n}(u)$$

   - *Absolute* position of origin is unimportant ⇒ "invariance under *translation* ".
   - *Relative* position of control vertices is important.

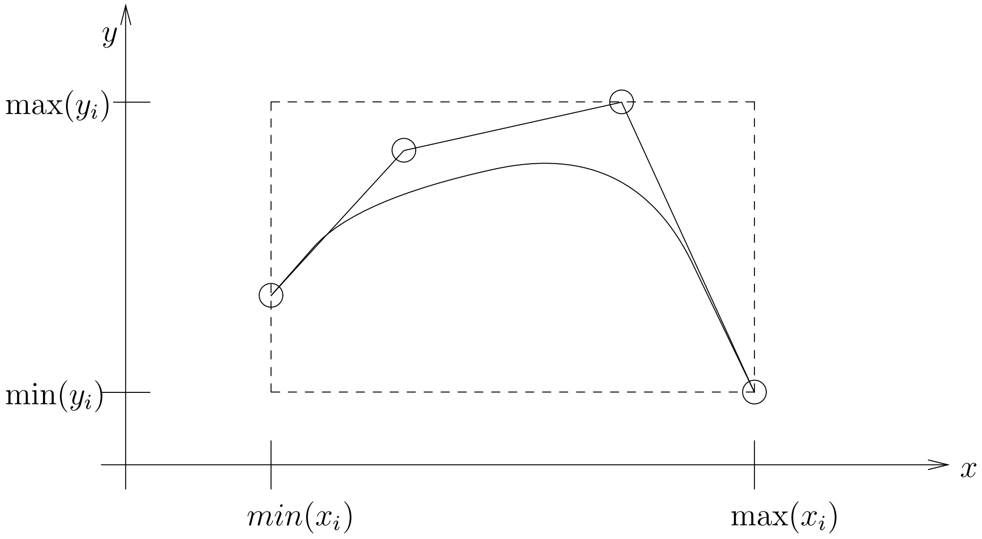

Figure 4.11: Bézier curve bounding box

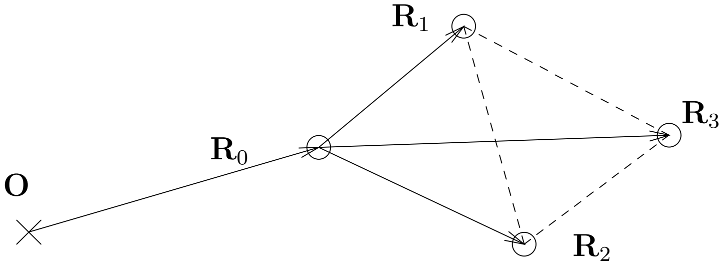

Figure 4.12: Geometry of Bézier curves is invariant under translation.

(b) Rotation (using a rotation matrix $\mathbf{A}$):
Let $\mathbf{A}$ be a matrix of rotation about an arbitrary point. We can then demonstrate that the Bézier curve is invariant under rotation:

$$\begin{aligned}
\mathbf{R}(u) &= [x(u), y(u), z(u)] \\
\mathbf{R}_i &= [x_i, y_i, z_i] \\
[x'(u), y'(u), z'(u)] &= [x(u), y(u), z(u)] \cdot \mathbf{A} \\
&= \sum_{i=0}^{n} [x_i, y_i, z_i] \mathbf{A} B_{i,n}(u) \\
&= \sum_{i=0}^{n} [x'_i, y'_i, z'_i] B_{i,n}(u)
\end{aligned}$$

- Form of Bézier curve is invariant under rotations
- Curve is easily computable

Therefore, Bézier curves are invariant under translations and rotations and they are easily computable under such transformations.

5. *End points geometric property*:

- The first and last control points are the endpoints of the curve. In other words, $\mathbf{b}_0 = \mathbf{R}(0)$ and $\mathbf{b}_n = \mathbf{R}(1)$.

- The curve is tangent to the control polygon at the endpoints. This can be easily observed by taking the first derivative of a Bézier curve

  $$\dot{\mathbf{R}}(u) = \frac{d\mathbf{R}(u)}{du} = n\sum_{i=0}^{n-1}(\mathbf{b}_{i+1} - \mathbf{b}_i)B_{i,n-1}(u), \qquad 0 \le u \le 1.$$

  In particular we have $\dot{\mathbf{R}}(0) = n(\mathbf{b}_1 - \mathbf{b}_0)$ and $\dot{\mathbf{R}}(1) = n(\mathbf{b}_n - \mathbf{b}_{n-1})$. Equation (4.5) can be simplified by setting $\Delta\mathbf{b}_i = \mathbf{b}_{i+1} - \mathbf{b}_i$:

  $$\dot{\mathbf{R}}(u) = n\sum_{i=0}^{n-1}\Delta\mathbf{b}_i B_{i,n-1}(u), \qquad 0 \le u \le 1$$

  The first derivative of a Bézier curve, which is called *hodograph*, is an another Bézier curve whose degree is lower than the original curve by one and has control points $\Delta\mathbf{b}_i$, $i = 0, \cdots, n-1$. Hodograph is useful in the study of intersection and other interrogation problems such as singularities and inflection points.

6. *Convex hull property*

$$\mathbf{R}(u) = \sum_{i=0}^{n}\mathbf{R}_i B_{i,n}(u); \quad 0 \le u \le 1$$

Recall: $0 \le B_{i,n}(u) \le 1$ and $\sum_{i=0}^{n} B_{i,n}(u) = 1$.

n=1 The convex hull is the line segment connecting the two vertices, see Figure 4.13

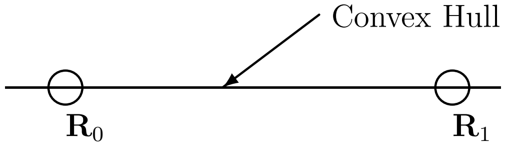

Figure 4.13: Convex hull of a linear Bézier curve.

$$\begin{aligned}\mathbf{R}(u) &= (1-u)\mathbf{R}_0 + u\mathbf{R}_1; \quad 0 \le u \le 1\\ &= \mathbf{R}_0 + (\mathbf{R}_1 - \mathbf{R}_0)(u); \quad 0 \le u \le 1\end{aligned}$$

n=2

$$\begin{aligned}\mathbf{R}(u) &= \mathbf{R}_0(1-u)^2 + 2u(1-u)\mathbf{R}_1 + u^2\mathbf{R}_2\ 0 \le u \le 1\\ &= \mathbf{R}_0 + 2u(1-u)(\mathbf{R}_1 - \mathbf{R}_0) + u^2(\mathbf{R}_2 - \mathbf{R}_0)\\ &= \mathbf{R}_0 + \alpha(\mathbf{R}_1 - \mathbf{R}_0) + \beta(\mathbf{R}_2 - \mathbf{R}_0)\end{aligned}$$

Here we have made the substitution $\alpha = 2u(1-u)$ and $\beta = u^2$ to define an oblique coordinate system (α, β). With the restriction $0 \le u \le 1 \Rightarrow 0 \le \alpha, \beta \le 1$, and $\alpha + \beta = 1 - (1-u)^2 \le 1$, any point on the curve must also be contained in the triangle defined by the polygon vertices, as shown in Figure 4.14.

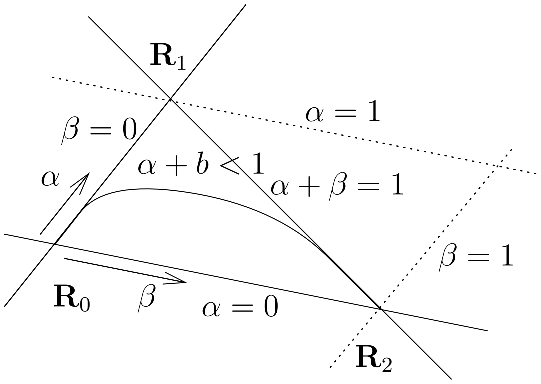

Figure 4.14: Convex hull of a quadratic Bézier curve.

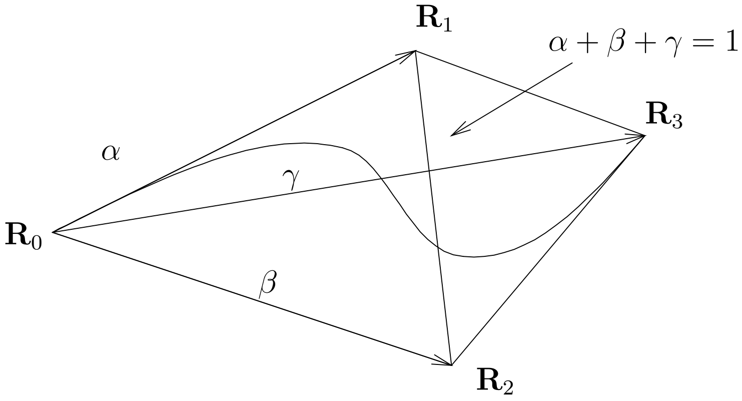

Figure 4.15: Convex hull of a 3D space cubic Bézier curve is a tetrahedron.

n=3

$$\mathbf{R}(u) = \mathbf{R}_0 + \alpha(\mathbf{R}_1 - \mathbf{R}_0) + \beta(\mathbf{R}_2 - \mathbf{R}_0) + \gamma(\mathbf{R}_3 - \mathbf{R}_0)\ 0 \leq \alpha, \beta, \gamma \leq 1\ for\ 0 \leq u \leq 1$$

α, β, and γ are the oblique coordinates of a point on the curve. With $0 \leq \alpha, \beta, \gamma \leq 1$, and $\alpha+\beta+\gamma = 1-(1-u)^3 \leq 1$ the curve is contained in the tetrahedron $\mathbf{R}_0\mathbf{R}_1\mathbf{R}_2\mathbf{R}_3$, see Figure 4.15 or the quadrilateral in Figure 4.16.

In general, the convex hull is the intersection of all the convex sets containing all vertices or the intersection of the half spaces generated by taking 3 vertices at a time to construct a plane and having all other vertices on one side. The convex hull can also be conceptualized at the shape of a rubber band/sheet stretched taut over the polygon vertices. This is illustrated for the 2D case in Figure 4.16.

The convex hull property is useful in

- Intersection problems:
  - Subdivide curves into two curve segments at $u = 0.5$.
  - Compare the convex hulls of the subdivided curves (see Figure 4.17).
  - Continue subdivision of intersecting subcurves until the size of their convex hulls are less than a given tolerance.

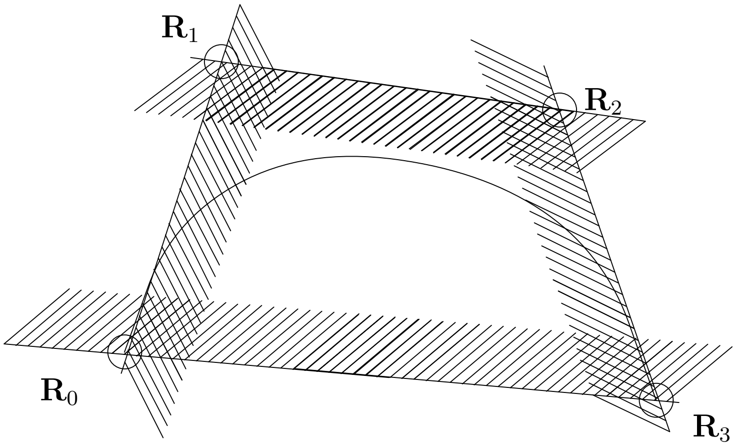

Figure 4.16: Convex hull of a planar cubic Bézier curve

- Detection of absence of interference.
- Providing the approximate position of curve through simple bounds.

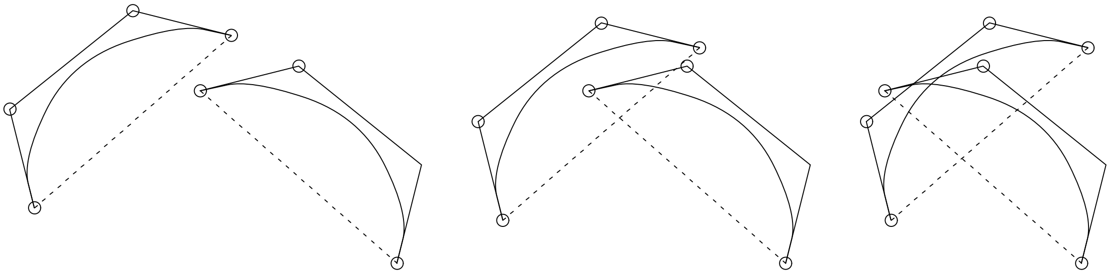

Figure 4.17: Comparison of convex hulls of Bézier curves as a means to detect intersection

7. *Variation diminishing property*
   A polygon can be created by the segments connecting the ordered vertices of a Bézier curve.

   In 2D: The number of intersections of a straight line with a planar Bézier curve is no greater than the number of intersections of the line with the control polygon. A line intersecting the convex hull of a planar Bézier curve may intersect the curve, be tangent to the curve, or not intersect the curve at all. It may not, however, intersect the curve more times than it intersects the control polygon. This property is illustrated in Figure 4.18.
   In 3D: The same relation holds true for a plane.
   Results : (a rough interpretation)

   - A Bézier curve oscillates less than its polygon.
   - The polygon's segments exaggerate the oscillation of the curve.

   This property is important in intersection algorithms and in detecting the "fairness" of Bézier curves.

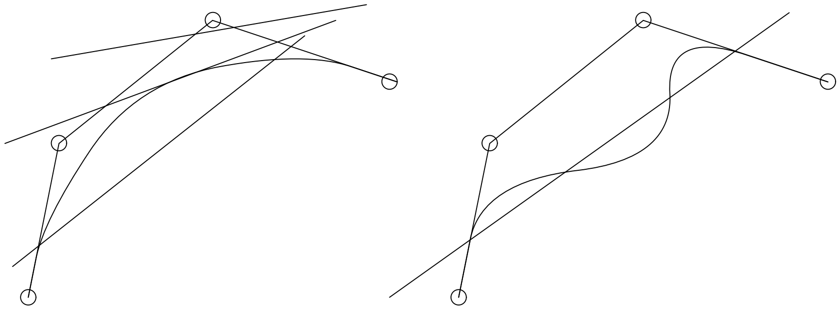

Figure 4.18: Variation diminishing property of a cubic Bézier curve

Algorithms for Bézier curves

- *Evaluation and subdivision algorithm*: A Bézier curve can be evaluated at a specific parameter value t_0 and the curve can be split at that value using the *de Casteljau algorithm*, where

  $$\mathbf{b}_i^k(t_0) = (1 - t_0)\mathbf{b}_{i-1}^{k-1} + t_0\mathbf{b}_i^{k-1}, \quad k = 1, 2, \ldots, n, \quad i = k, \ldots, n \tag{4.5}$$

  is applied recursively to obtain the new control points. The algorithm is illustrated in Figure 4.19, and has the following properties:

  – The values $\mathbf{b}_i^0$ are the original control points of the curve.

  – The value of the curve at parameter value t_0 is $\mathbf{b}_n^n$.

  – The curve can be represented as two curves, with control points $(\mathbf{b}_0^0,\ \mathbf{b}_1^1, \ldots,\ \mathbf{b}_n^n)$ and $(\mathbf{b}_n^n,\ \mathbf{b}_n^{n-1}, \ldots,\ \mathbf{b}_n^0)$.

- *Continuity algorithm*: Bézier curves can represent complex curves by increasing the degree and thus the number of control points. Alternatively, complex curves can be represented using composite curves, which can be formed by joining several Bézier curves end to end. If this method is adopted, the continuity between consecutive curves must be addressed.

  One set of continuity conditions are the *geometric* continuity conditions, designated by the letter G with an integer exponent. *Position continuity*, or G^0 continuity, requires the endpoints of the two curves to coincide,

  $$\mathbf{R}^a(1) = \mathbf{R}^b(0).$$

  The superscripts denote the first and second curves. *Tangent continuity*, or G^1 continuity, requires G^0 continuity and in addition the tangents of the curves to be in the same direction,

  $$\dot{\mathbf{R}}^a(1) = \alpha_1\mathbf{t}$$
  $$\dot{\mathbf{R}}^b(0) = \alpha_2\mathbf{t}$$

where $\mathbf{t}$ is the common unit tangent vector and α_1, α_2 are the magnitude of $\dot{\mathbf{r}}^a(1)$ and $\dot{\mathbf{r}}^b(0)$. G^1 continuity is important in minimizing stress concentrations in physical solids and preventing flow separation in fluids.

Curvature continuity, or G^2 continuity, requires G^1 continuity and in addition the center of curvature to move continuously past the connection point ,

$$\ddot{\mathbf{R}}^b(0) = \left(\frac{\alpha_2}{\alpha_1}\right)^2 \ddot{\mathbf{R}}^a(1) + \mu\dot{\mathbf{R}}^a(1).$$

where μ is an arbitrary constant. G^2 continuity is important for aesthetic reasons and also for preventing fluid flow separation.

More stringent continuity conditions are the *parametric* continuity conditions, where C^k continuity requires the kth derivative (and all lower derivatives) of each curve to be equal at the joining point. In other words,

$$\frac{d^k\mathbf{R}^a(1)}{dt^k} = \frac{d^k\mathbf{R}^b(0)}{dt^k}.$$

The C^1 and C^2 continuity conditions for consecutive segments of a composite degree n Bézier curve can be stated as

$$\begin{aligned} h_{i+1}\left(\mathbf{b}_{ni} - \mathbf{b}_{ni-1}\right) &= h_i\left(\mathbf{b}_{ni+1} - \mathbf{b}_{ni}\right), \quad \text{and} && (4.6)\\ \mathbf{b}_{ni-1} + \frac{h_{i+1}}{h_i}(\mathbf{b}_{ni-1} - \mathbf{b}_{ni-2}) &= \mathbf{b}_{ni+1} + \frac{h_i}{h_{i+1}}(\mathbf{b}_{ni+1} - \mathbf{b}_{ni+2}) && (4.7)\end{aligned}$$

where, for the ith Bézier curve segment parameter t runs over the interval $[t_i,\ t_{i+1}]$, $h_i = t_{i+1} - t_i$ (see Figure 4.20 for the connection of cubic Bézier curve segments).

- *Degree elevation*: The degree elevation algorithm permits us to increase the degree and control points of a Bézier curve from n to $n+1$ without changing the shape of the curve. The new control points $\bar{\mathbf{b}}_i$ of the degree $n+1$ curve are given by

$$\bar{\mathbf{b}}_i = \frac{i}{n+1}\mathbf{b}_{i-1} + \left(1 - \frac{i}{n+1}\right)\mathbf{b}_i, \quad i = 0, \ldots, n+1 \qquad (4.8)$$

  where $\mathbf{b}_{-1} = \mathbf{b}_{n+1} = \mathbf{0}$

ls

4.3.5 Bézier surfaces

A tensor product surface is formed by moving a curve through space while allowing deformations in that curve. This can be thought of as allowing each control point $\mathbf{b}_i$ to sweep a curve in space. If this surface is represented using Bernstein polynomials, a Bézier surface (patch) is formed, with the following formula:

$$\mathbf{R}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{b}_{ij} B_{i,m}(u) B_{j,n}(v), \qquad 0 \le u, v \le 1.$$

Here, the set of straight lines drawn between consecutive control points $\mathbf{b}_{ij}$ is referred to as the *control net*. It is easy to see that boundary isoparametric curves ($u = 0$, $u = 1$, $v = 0$ and $v = 1$) have the same control points as the corresponding boundary points on the net. An example of a bi-quadratic Bézier surface with its control net can be seen in Figure 4.21. Since a Bézier surface is a direct extension of univariate Bézier curve to its bivariate form, it inherits many of the properties of the Bézier curve described before such as:

- Geometry invariance property
- End points geometric property
- Convex hull property

However, no variation diminishing property is known for Bézier surface patch.

4.4 Composite curves

4.4.1 Motivation

- Increase complexity without increasing degree of polynomial.
- Interpolation/ approximation of large arrays of points.
- Avoid problems of high degree polynomials such as oscillation, see Section 4.4.2.

4.4.2 Lagrange basis

The Lagrange form is another basis for curve representation. It is presented here as motivation for establishing a composite polynomial curve representation of lower order.

Given: $(u_i,\ \mathbf{R}_i)$ $i = 0, 1, 2, \cdots, N$, we create an interpolatory curve in terms of:

$$\begin{aligned}\mathbf{R}(u) &= \sum_{i=0}^{N} \mathbf{R}_i L_{i,N}(u) \quad 0 \leq u \leq 1 \\ &\text{where } L_{i,N}(u_j) = \delta_{ij} \\ &\quad \text{and } L_{i,N}(u) = \Pi_{j=0,j\neq i}^{N}\Big(\frac{u-u_j}{u_i-u_j}\Big).\end{aligned}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$ (δ_{ij} is called the Kronecker delta); u_i are parameter values (eq. accumulated length of the polygon $\mathbf{R}_0\mathbf{R}_1 ... \mathbf{R}_i$) and $\mathbf{R}_i$ are data points; $L_{i,N}$ is a polynomial of degree N. As with any polynomial of degree N, it has up to $N - 1$ real extrema. As can be seen in Figure 4.22, Lagrange interpolation may lead to highly oscillatory approximants.

The Lagrange basis is not frequently used because of oscillation and of the need to use high degree basis functions for interpolation. This leads to developing methods involving low degree composite curves (splines).

4.4.3 Continuity conditions

See Figure 4.23 for an illustration.

- Position:

$$\mathbf{R}^{(1)}(1) = \mathbf{R}^{(2)}(0)$$

- Tangent:

$$\begin{aligned} \dot{\mathbf{R}}^{(1)}(1) &= \alpha_1 \mathbf{t} \\ \dot{\mathbf{R}}^{(2)}(0) &= \alpha_2 \mathbf{t} \end{aligned}$$

  where $\mathbf{t}$ is a common unit tangent vector at the interface and α_1 and α_2 are real constants.

- Curvature:

$$\begin{aligned} \dot{\mathbf{R}} &= \dot{s}\mathbf{t} \quad \text{where } \dot{s} = |\dot{\mathbf{R}}| \\ \dot{\mathbf{t}} &= \mathbf{t}_s \dot{s} = \dot{s}\kappa \cdot \mathbf{n} \\ \ddot{\mathbf{R}} &= \ddot{s}\mathbf{t} + \dot{s}^2 \kappa \mathbf{n} \end{aligned}$$

  Using the above equations, taking $\dot{\mathbf{R}} \times \ddot{\mathbf{R}}$ and using $\mathbf{t} \times \mathbf{n} = \mathbf{b}$ implies

$$\kappa \mathbf{b} = \frac{\dot{\mathbf{R}} \times \ddot{\mathbf{R}}}{|\dot{\mathbf{R}}|^3}$$

  Therefore, if κ, $\mathbf{n}$ are continuous → $\mathbf{b} = \mathbf{t} \times \mathbf{n}$ is also continuous → $\kappa \cdot \mathbf{b}$ is continuous. Hence,

$$\begin{aligned} \frac{\dot{\mathbf{R}}^{(1)}(1) \times \ddot{\mathbf{R}}^{(1)}(1)}{|\dot{\mathbf{R}}^{(1)}(1)|^3} &= \frac{\dot{\mathbf{R}}^{(2)}(0) \times \ddot{\mathbf{R}}^{(2)}(0)}{|\dot{\mathbf{R}}^{(2)}(0)|^3} \\ \mathbf{t} \times \left[\ddot{\mathbf{R}}^{(2)}(0) - \left(\frac{\alpha_2}{\alpha_1}\right)^2 \ddot{\mathbf{R}}^{(1)}(1) \right] &= 0 \\ \ddot{\mathbf{R}}^{(2)}(0) &= \left(\frac{\alpha_2}{\alpha_1}\right)^2 \ddot{\mathbf{R}}^{(1)}(1) + \mu \dot{\mathbf{R}}^{(1)}(1) \end{aligned}$$

  where μ is another real constant.

Order of geometric continuity:

- G^0– position continuity,
- G^1– unit tangent continuity, and
- G^2– center of curvature continuity.

Order of parametric continuity (more restrictive than geometric continuity):

- C^0– position continuity,
- C^1– first derivative continuity, and
- C^2– second derivative continuity.

4.4.4 Bézier composite curves (splines)

- Cardinal or interpolatory splines are used for fitting data.
- Bézier composite curves usually used for preliminary design (without fitting)

$$\mathbf{R}(u) = \sum_{i=0}^{3} u^i (1-u)^{3-i} \mathbf{R}_i \quad 0 \leq u \leq 1$$

- Conditions for position and tangent continuity, see Figure 4.24
  - Position (G^0)

$$\mathbf{R}_3^{(1)} = \mathbf{R}_0^{(2)}$$

  - Tangent (G^1)

$$\mathbf{t} = \frac{3}{\alpha_1}\Big(\mathbf{R}_3^{(1)} - \mathbf{R}_2^{(1)}\Big) = \frac{3}{\alpha_2}\Big(\mathbf{R}_1^{(2)} - \mathbf{R}_0^{(2)}\Big)$$

    Hence, $\mathbf{R}_3^{(1)} = \mathbf{R}_0^{(2)}$ and $\mathbf{R}_2^{(1)}$, $\mathbf{R}_1^{(2)}$ are collinear.
  - Conditions for curvature continuity:

$$\begin{aligned} \ddot{\mathbf{R}}^{(2)}(0) &= \lambda^2 \ddot{\mathbf{R}}^{(1)}(1) + \mu \dot{\mathbf{R}}^{(1)}(1) \\ \lambda &= \frac{\alpha_2}{\alpha_1} \end{aligned}$$

    Using $\ddot{\mathbf{R}}^{(1)}(1) = 6\Big(\mathbf{R}_3^{(1)} - 2\mathbf{R}_2^{(1)} + \mathbf{R}_1^{(1)}\Big)$

$$\mathbf{R}_2^{(2)} - \mathbf{R}_3^{(1)} = \lambda^2\Big(\mathbf{R}_1^{(1)} - \mathbf{R}_2^{(1)}\Big) + \Big(\lambda^2 + 2\lambda + \frac{\mu}{2}\Big)\Big(\mathbf{R}_3^{(1)} - \mathbf{R}_2^{(1)}\Big)$$

    These conditions imply coplanarity of points $\mathbf{R}_1^{(1)}$, $\mathbf{R}_2^{(1)}$, $\mathbf{R}_3^{(1)}$, $\mathbf{R}_1^{(2)}$, $\mathbf{R}_2^{(2)}$.

The design process may involve the following steps:

- Start with one segment.
- For each new segment $\lambda, \mu, \mathbf{R}_3$ may be chosen freely for shape creation.

However, such construction may lead to

- not enough freedom for G^2 continuity.
- enough freedom for G^1 continuity.

If more degrees of freedom are needed, an increase of the degree of the curve is necessary.

4.4.5 Uniform Cubic B-splines

Motivation:

- Changing of one polygon vertex of a Bézier curve or one data point of a cardinal or interpolatory spline affects the curve GLOBALLY.
- Avoid increasing degree of Bézier curve to construct complex composite shapes.
- Develop a LOCAL curve scheme.

Define a *uniform*, cubic B-spline curve span by:

$$\mathbf{R}_i(u) = \mathbf{V}_i N_{0,4}(u) + \mathbf{V}_{i+1} N_{1,4}(u) + \mathbf{V}_{i+2} N_{2,4}(u) + \mathbf{V}_{i+3} N_{3,4}(u)$$

$\mathbf{R}_i(u)$ represents the i^{th} span of the composite curve and $N_{i,4}(u)$ represents the i^{th} blending function of order 4 (or degree 3). Each span is defined in terms of a local parameter: $0 \leq u \leq 1$. $N_{k,4}(u)$ are degree 3 polynomials defined from continuity conditions. Let us assume:

$$N_{j,4}(u) = a_j + b_j u + c_j u^2 + d_j u^3 \qquad j = 0, \cdots, 3$$

Consider two adjacent spans, as shown in Figure 4.25 with control points $\mathbf{V}_i$ shown in Figure 4.26.

For C^2 continuity, the following three vector conditions must hold:

- $\mathbf{R}_i(1) = \mathbf{R}_{i+1}(0) \Rightarrow$

$$\begin{bmatrix} N_{0,4}(1) & N_{1,4}(1) & N_{2,4}(1) & N_{3,4}(1) \end{bmatrix} \begin{bmatrix} \mathbf{V}_i \\ \mathbf{V}_{i+1} \\ \mathbf{V}_{i+2} \\ \mathbf{V}_{i+3} \end{bmatrix} =$$

$$\begin{bmatrix} N_{0,4}(0) & N_{1,4}(0) & N_{2,4}(0) & N_{3,4}(0) \end{bmatrix} \begin{bmatrix} \mathbf{V}_{i+1} \\ \mathbf{V}_{i+2} \\ \mathbf{V}_{i+3} \\ \mathbf{V}_{i+4} \end{bmatrix}$$

This is valid for every $\mathbf{V}_i \ \Rightarrow$

$$\begin{aligned} N_{0,4}(1) &= 0 && (4.9) \\ N_{1,4}(1) &= N_{0,4}(0) && (4.10) \\ N_{2,4}(1) &= N_{1,4}(0) && (4.11) \\ N_{3,4}(1) &= N_{2,4}(0) && (4.12) \\ N_{3,4}(0) &= 0 && (4.13) \end{aligned}$$

- $\mathbf{R}'_i(1) = \mathbf{R}'_{i+1}(0) \Rightarrow$

$$\begin{aligned} N'_{0,4}(1) &= 0 && (4.14) \\ N'_{1,4}(1) &= N'_{0,4}(0) && (4.15) \\ N'_{2,4}(1) &= N'_{1,4}(0) && (4.16) \\ N'_{3,4}(1) &= N'_{2,4}(0) && (4.17) \\ N'_{3,4}(0) &= 0 && (4.18) \end{aligned}$$

- $\mathbf{R}_i''(1) = \mathbf{R}_{i+1}''(0) \Rightarrow$

$$\begin{aligned} N_{0,4}''(1) &= 0 && (4.19) \\ N_{1,4}''(1) &= N_{0,4}''(0) && (4.20) \\ N_{2,4}''(1) &= N_{1,4}''(0) && (4.21) \\ N_{3,4}''(1) &= N_{2,4}''(0) && (4.22) \\ N_{3,4}''(0) &= 0 && (4.23) \end{aligned}$$

- Also if all four vertices coincide $\Rightarrow$ the i^{th} span reduces to a point $\Rightarrow \mathbf{R}_i(u) = \mathbf{V}_i$

$$\sum_{i=0}^{3} N_{i,4}(u) = 1 \qquad (4.24)$$

Solving the above 16 equations 4.9–4.24 yield $a_j, b_j, c_j, d_j;\ j = 0, 1, 2, 3$.

$$\begin{aligned} \begin{bmatrix} N_{0,4}(u) & N_{1,4}(u) & N_{2,4}(u) & N_{3,4}(u) \end{bmatrix} &= \frac{1}{3!}\begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \cdot \mathbf{M}_B \\ &= \frac{1}{3!}\begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \end{aligned}$$

Properties:

- Position, derivatives (see Figure 4.26):

$$\begin{aligned} \mathbf{R}_i(0) &= \frac{2}{3}\mathbf{V}_{i+1} + \frac{1}{3}\Big(\frac{\mathbf{V}_i + \mathbf{V}_{i+2}}{2}\Big) \\ \mathbf{R}_i(1) &= \frac{2}{3}\mathbf{V}_{i+2} + \frac{1}{3}\Big(\frac{\mathbf{V}_{i+1} + \mathbf{V}_{i+3}}{2}\Big) \\ \mathbf{R}_i'(0) &= \frac{1}{2}\Big(\mathbf{V}_{i+2} - \mathbf{V}_i\Big) \\ \mathbf{R}_i'(1) &= \frac{1}{2}\Big(\mathbf{V}_{i+3} - \mathbf{V}_{i+1}\Big) \\ \mathbf{R}_i''(0) &= \Big(\mathbf{V}_{i+2} - \mathbf{V}_{i+1}\Big) - \Big(\mathbf{V}_{i+1} - \mathbf{V}_i\Big) \\ \mathbf{R}_i''(1) &= \Big(\mathbf{V}_{i+3} - \mathbf{V}_{i+2}\Big) - \Big(\mathbf{V}_{i+2} - \mathbf{V}_{i+1}\Big) \end{aligned}$$

- Local control – one vertex affects only 4 spans.
- Convex hull property – local.
- Variation diminishing property.
- *Replication of polygon control of Bézier curves at end points* can be achieved by setting

$$\begin{aligned} &\mathbf{V}_{-1} = 2\mathbf{V}_0 - \mathbf{V}_1 \quad \text{(imaginary vertex)} \\ \text{Then}\quad &\mathbf{R}_{-1}(0) = \frac{2}{3}\mathbf{V}_0 + \frac{1}{6}(\mathbf{V}_{-1} + \mathbf{V}_1) = \mathbf{V}_0 \\ &\mathbf{R}_{-1}'(0) = \frac{1}{2}(\mathbf{V}_1 - \mathbf{V}_{-1}) = \mathbf{V}_1 - \mathbf{V}_0 \end{aligned}$$

- *Interpolation* can also be achieved, via the so called cardinal or interpolatory splines, which involve interpolation of N data points $\mathbf{R}_i$ $(i = 0, 1, ...N-1)$

$$\tfrac{2}{3}\mathbf{V}_{i+1} + \tfrac{1}{6}(\mathbf{V}_i + \mathbf{V}_{i+2}) = \mathbf{R}_i$$

or

$$\mathbf{V}_i + 4\mathbf{V}_{i+1} + \mathbf{V}_{i+2} = 6\mathbf{R}_i \quad \textbf{for } i = 0,\ 2,\ \cdots N-1$$

We have $N+2$ unknowns so we need two boundary conditions.

To replicate the *natural spline* boundary conditions, we set $\mathbf{R}''(0) = \mathbf{R}''(1) = 0$, which imply zero curvature at the ends. Natural splines minimize the integral of curvature square and are the smoothest curves that pass through a set of points. We then have $\mathbf{R}_0''(0) = \mathbf{V}_2 - 2\mathbf{V}_1 + \mathbf{V}_0 = 0$. Also we have $N-1$ spans, N data points and $N+2$ control points, $\mathbf{V}_0, \mathbf{V}_1, ...\mathbf{V}_{N+1}$ the condition will be

$$\mathbf{R}_{N-1}''(0) = \mathbf{V}_{N+1} - 2\mathbf{V}_N + \mathbf{V}_{N-1} = 0.$$

Then we can derive the banded matrix to solve the above linear system.

$$\begin{bmatrix} 1 & -2 & 1 & & & & & \\ 1 & 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & & \\ & & 1 & 4 & 1 & & & \\ & & & & \ddots & & & \\ & & & & & 1 & 4 & 1 \\ & & & & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \\ \vdots \\ \mathbf{V}_N \\ \mathbf{V}_{N+1} \end{bmatrix} = 6 \begin{bmatrix} 0 \\ \mathbf{R}_0 \\ \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_{N+1} \\ 0 \end{bmatrix}$$

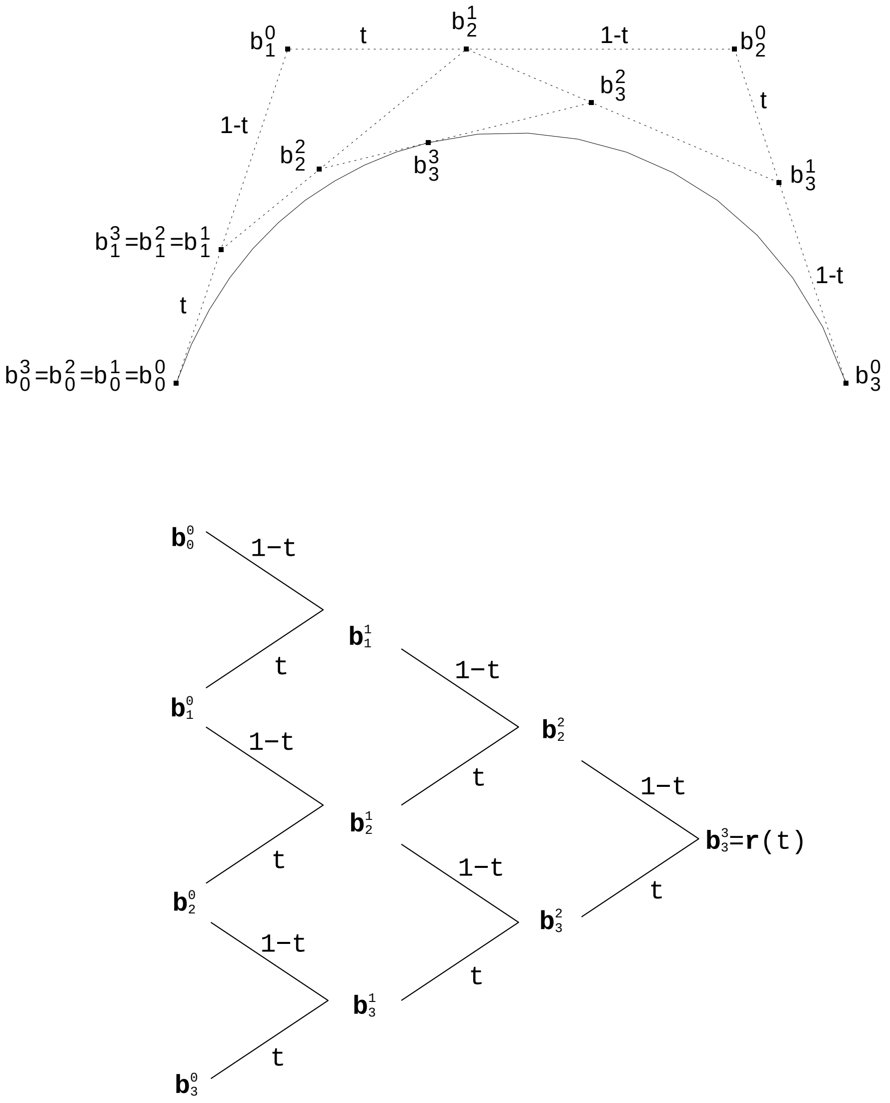

Figure 4.19: The de Casteljau algorithm.

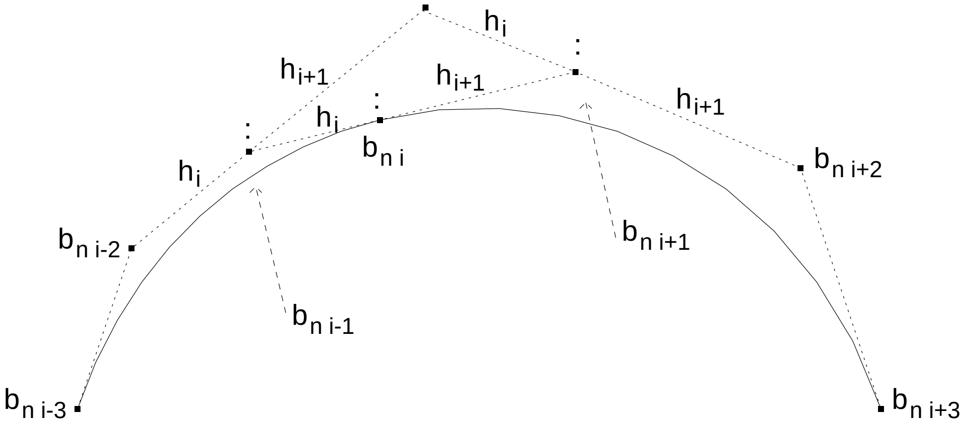

Figure 4.20: Continuity conditions.

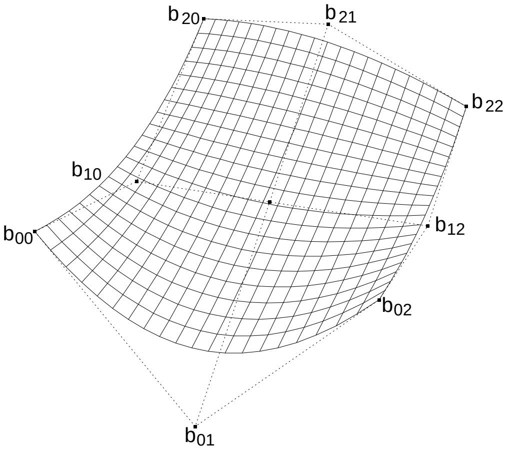

Figure 4.21: A Bézier Surface with Control Net.

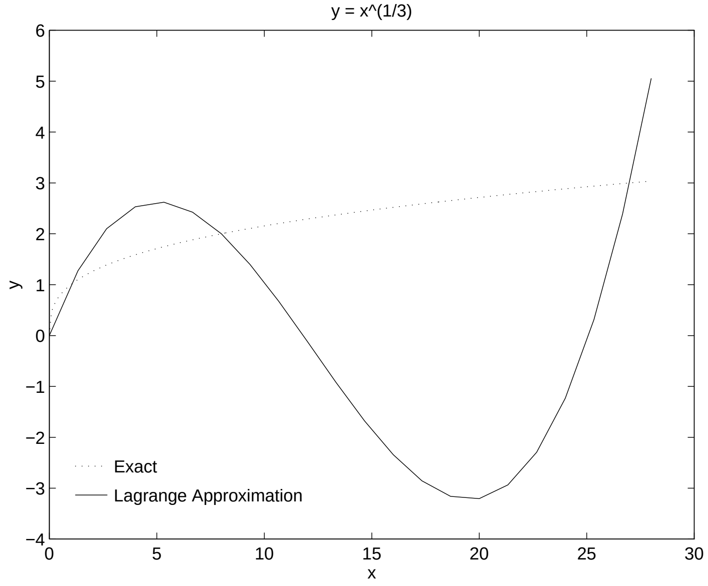

Figure 4.22: Approximation of the function $y = x^{\frac{1}{3}}$ using a cubic Lagrange curve.

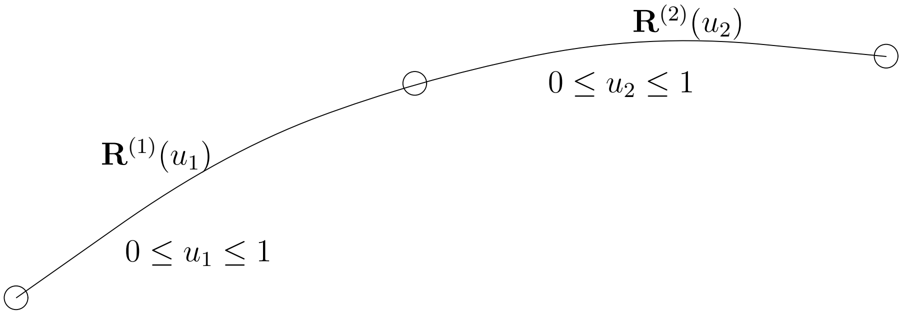

Figure 4.23: Composite curve consisting of two smaller curve segments.

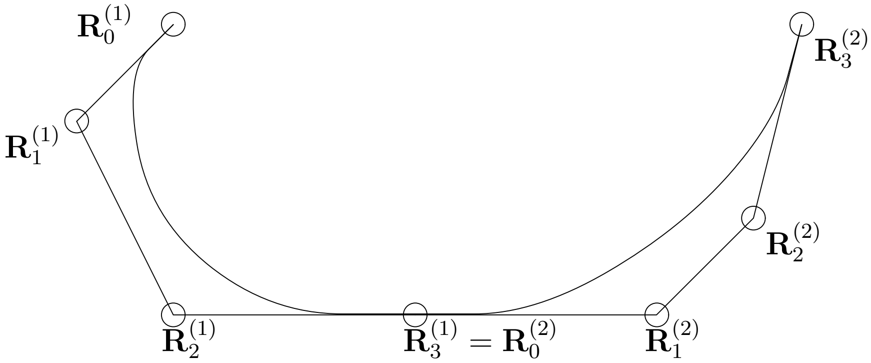

Figure 4.24: Continuity between Bézier curves restricts placement of control vertices.

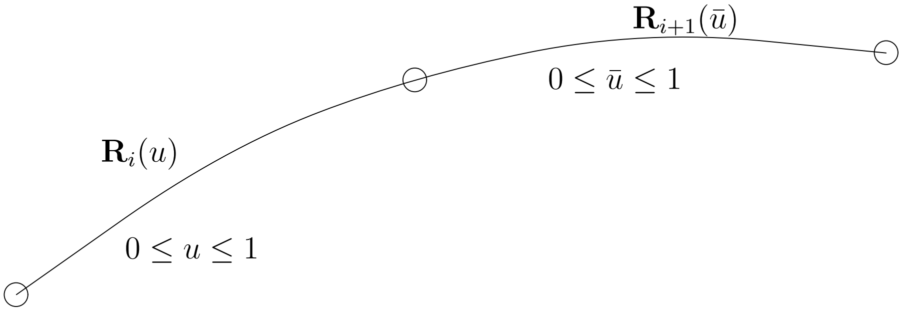

Figure 4.25: Two adjacent curve spans with position, first, and second derivative continuity.

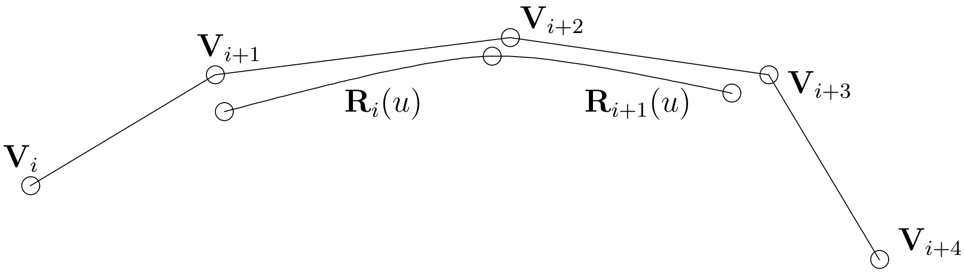

Figure 4.26: Two spans of a uniform cubic B-spline curve.

Bibliography

[1] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide.* Academic Press, Boston, MA, 3rd edition, 1993.

[2] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis Horwood, Chichester, England, 1981.

[3] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* A. K. Peters, Wellesley, MA, 1993. Translated by L. L. Schumaker.

[4] L. A. Piegl and W. Tiller. *The NURBS Book.* Springer, New York, 1995.

[5] F. Yamaguchi. *Curves and Surfaces in Computer Aided Geometric Design.* Springer-Verlag, NY, 1988.