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

Lecture 6

N. M. Patrikalakis

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

Contents

Reading in the Textbook

- Chapter 1, pp. 6 - 33

Lecture 6

B-splines (Uniform and Non-uniform)

6.1 Introduction

The formulation of uniform B-splines can be generalized to accomplish certain objectives. These include

- Non-uniform parameterization.
- Greater general flexibility.
- Change of one polygon vertex in a Bézier curve or of one data point in a cardinal (or interpolatory) spline curve changes entire curve (global schemes).
- Remove necessity to increase degree of Bézier curves or construct composite Bézier curves using previous schemes in order to increase degrees of freedom.
- Obtain a "local" approximation scheme.

The development extends the Bézier curve formulation to a piecewise polynomial curve with easy continuity control.

6.2 Definition

A parametric non-uniform B-spline curve is defined by

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(u) \tag{6.1}$$

where, $\mathbf{P}_i$ are $n+1$ control points; $N_{i,k}(u)$ are piecewise polynomial B-spline basis functions of order k (or degree $k-1$) with $n \geq k-1$.

Therefore n, k are independent, unlike Bézier curves. The parameter u obeys the inequality

$$t_o \leq u \leq t_{n+k} \tag{6.2}$$

6.2.1 Knot vector

For open (non-periodic) curves, it is usual to define a set $\mathbf{T}$ of non-decreasing real numbers which is called the knot vector, as follows:

$$\mathbf{T} = \{\underbrace{t_o = t_1 = \cdots = t_{k-1}}_{k\ equal\ values} < \underbrace{t_k \leq t_{k+1} \leq \cdots \leq t_n}_{n-k+1\ internal\ knots} < \underbrace{t_{n+1} = \cdots = t_{n+k}}_{k\ equal\ values}\} \tag{6.3}$$

At each knot value, the curve $\mathbf{P}(u)$ has some degree of discontinuity in its derivatives above a certain order as we will see later. The total number of knots is $n+k+1$ which equals the number of control points in the curve plus the curve's order.

6.2.2 Properties and definition of basis function $N_{i,k}(u)$

1. $\sum_{i=0}^{n} N_{i,k}(u) = 1$ *(partition of unity).*

2. $N_{i,k}(u) \geq 0$ *(positivity).*

3. $N_{i,k}(u) = 0$ if $u \notin [t_i, t_{i+k}]$ (*local support).*

4. $N_{i,k}(u)$ is $(k-2)$ times continuously differentiable at simple knots.
   If a knot t_j is of multiplicity $\rho(\leq k)$, ie. if

   $$t_j = t_{j+1} = \cdots = t_{j+\rho-1} \tag{6.4}$$

   then $N_{i,k}(u)$ is $(k-\rho-1)$ times continuously differentiable, ie. it is of class $C^{k-\rho-1}$.

5. Recursive definition (Cox-de Boor algorithm)

$$N_{i,1}(u) = \begin{cases} 1 & u \in [t_i, t_{i+1}) \\ 0 & u \notin [t_i, t_{i+1}) \end{cases} \tag{6.5}$$

$$N_{i,k}(u) = \frac{u - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(u) + \frac{t_{i+k} - u}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(u) \tag{6.6}$$

(set $\frac{0}{0} = 0$ above when it occurs)

Properties 1-4 or property 5 by itself (given a known knot vector $\mathbf{T}$) define the B-spline basis.

6.2.3 ***Example:*** 2^{nd} order B-spline basis function

(Piecewise linear case– see Figure 6.1)
$k = 2, C^{k-2} = C^{2-2} = C^0$

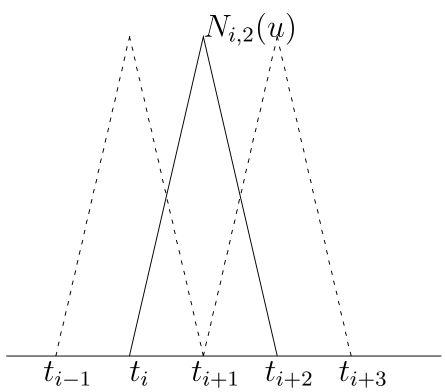

Figure 6.1: Plot of 2^{nd} order B-spline basis functions.

$N_{i,2}(u)$ consists of two piecewise linear polynomials:

$$N_{i,2}(u) = \begin{cases} \frac{u-t_i}{t_{i+1}-t_i} & t_i \leq u \leq t_{i+1} \\ \frac{t_{i+2}-u}{t_{i+2}-t_{i+1}} & t_{i+1} \leq u \leq t_{i+2} \end{cases}$$

6.2.4 ***Example:*** 3^{rd} order B-spline basis function

(Piecewise quadratic case– See Figure 6.2)
$k = 3, C^{k-2} = C^{3-2} = C^1$

$N_{i,3}(t_i)$ consists of three piecewise quadratic polynomials $y_1(u)$, $y_2(u)$ and $y_3(u)$. We want to find $y_1(u)$, $y_2(u)$ and $y_3(u)$ such that

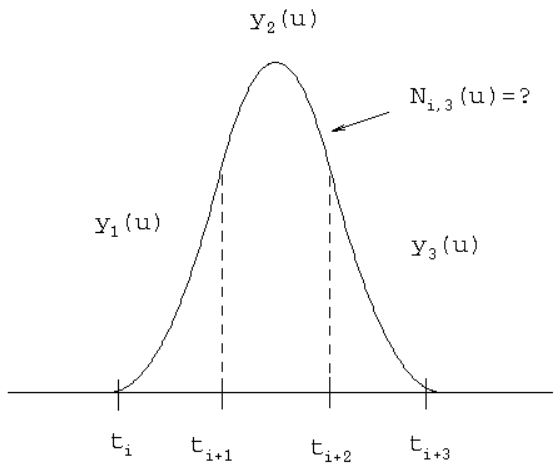

Figure 6.2: Plot of 3^{rd} order B-spline basis function.

$$\begin{aligned} N_{i,3}(t_i) &= N'_{i,3}(t_i) = 0. \\ N_{i,3}(t_{i+3}) &= N'_{i,3}(t_{i+3}) = 0. \end{aligned} \tag{6.7}$$

Position continuity:

$$y_1(u) = A\left(\frac{u - t_i}{t_{i+1} - t_i}\right)^2 \tag{6.8}$$

$$y_3(u) = B\left(\frac{t_{i+3} - u}{t_{i+3} - t_{i+2}}\right)^2 \tag{6.9}$$

$$y_2(u) = A(1-s)^2 + Bs^2 + C2s(1-s) \tag{6.10}$$

$$s = \frac{u - t_{i+1}}{t_{i+2} - t_{i+1}} \tag{6.11}$$

Note that

$$y_1(t_{i+1}) = y_2(t_{i+1}) = A$$

$$y_3(t_{i+2}) = y_2(t_{i+2}) = B$$

at $s = 0,\ s = 1 (u = t_{i+1}, u = t_{i+2})$.
First derivative continuity:

$$2(C - A)\frac{1}{t_{i+2} - t_{i+1}} = 2A\frac{1}{t_{i+1} - t_i} \tag{6.12}$$

$$2(B - C)\frac{1}{t_{i+2} - t_{i+1}} = -2B\frac{1}{t_{i+3} - t_{i+2}} \tag{6.13}$$

hence,

$$A\left[\frac{1}{t_{i+1} - t_i} + \frac{1}{t_{i+2} - t_{i+1}}\right] = C\frac{1}{t_{i+2} - t_{i+1}} \tag{6.14}$$

$$B\left[\frac{1}{t_{i+2} - t_{i+1}} + \frac{1}{t_{i+3} - t_{i+2}}\right] = C\frac{1}{t_{i+2} - t_{i+1}} \tag{6.15}$$

where

$$A, B = \text{functions}(C)$$

Need one more condition (normalization):

$$\int_{t_i\ or\ -\infty}^{t_{i+k}\ or\ +\infty} N_{i,k}(u)du = \frac{1}{k}(t_{i+k} - t_i)$$

We obtain

$$A(t_{i+2} - t_i) + B(t_{i+3} - t_{i+1}) + C(t_{i+2} - t_{i+1}) = t_{i+3} - t_i \tag{6.16}$$

From Equation 6.14 to 6.16, we obtain

$$A = \frac{t_{i+1} - t_i}{t_{i+2} - t_i}, \quad B = \frac{t_{i+3} - t_{i+2}}{t_{i+3} - t_{i+1}}, \quad C = 1 \tag{6.17}$$

Finally,

$$\begin{aligned} N_{i,3}(u) &= y_1(u)N_{i,1}(u) + y_2(u)N_{i+1,1}(u) + y_3(u)N_{i+2,1}(u) \\ &= \frac{u - t_i}{t_{i+2} - t_i}N_{i,2}(u) + \frac{t_{i+3} - u}{t_{i+3} - t_{i+1}}N_{i+1,2}(u) \end{aligned} \tag{6.18}$$

6.2.5 ***Example:*** 4^{th} order basis function

(Cubic B-spline case– $K = 4$ see Figure 6.3)
$n = 6 \rightarrow 7$ control points
$n + k + 1 = 11$ knots

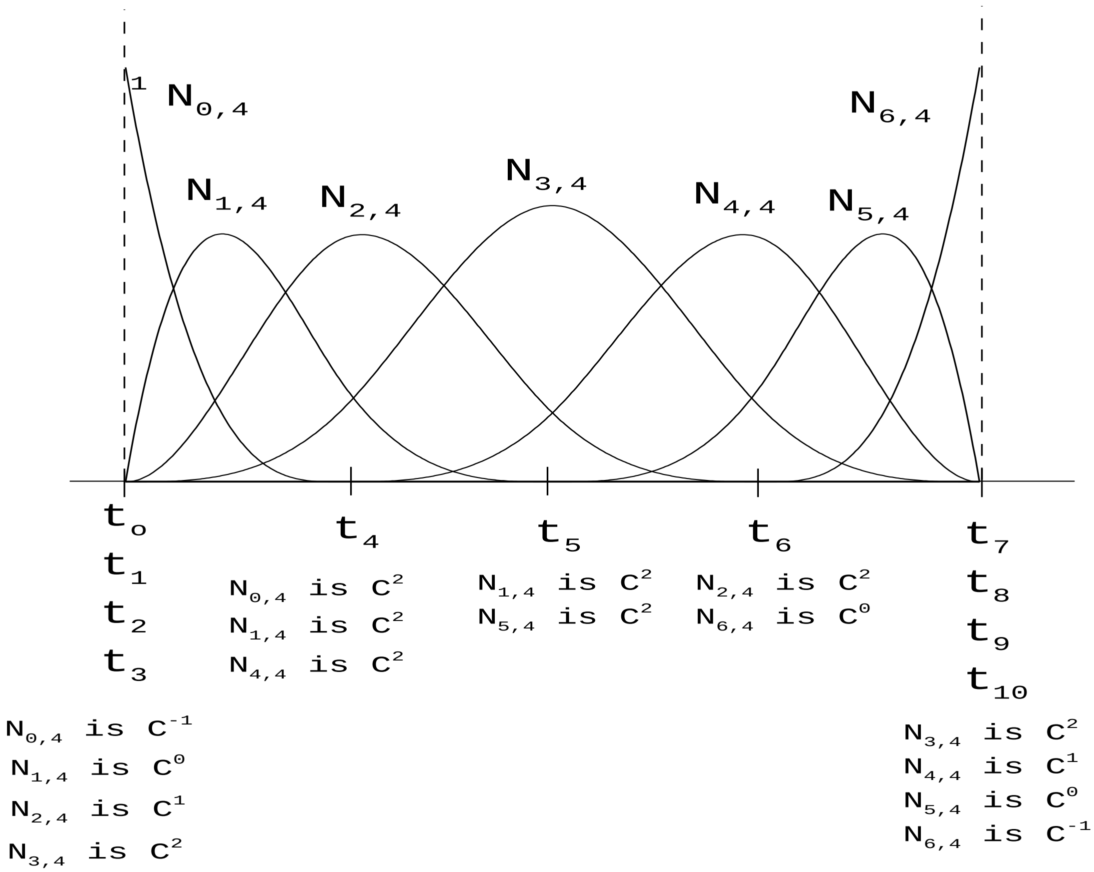

Figure 6.3: Cubic B-spline functions.

From property 1:

$$\dot{N}_{0,4}(t_0) + \dot{N}_{1,4}(t_0) = 0 \tag{6.19}$$

Therefore,

$$\dot{\mathbf{P}}(t_0) = (\mathbf{P}_1 - \mathbf{P}_0)\dot{N}_{1,4}(t_0) \tag{6.20}$$

Similarly,

$$\dot{\mathbf{P}}(t_{10}) = (\mathbf{P}_6 - \mathbf{P}_5)\dot{N}_{6,4}(t_{10}) \tag{6.21}$$

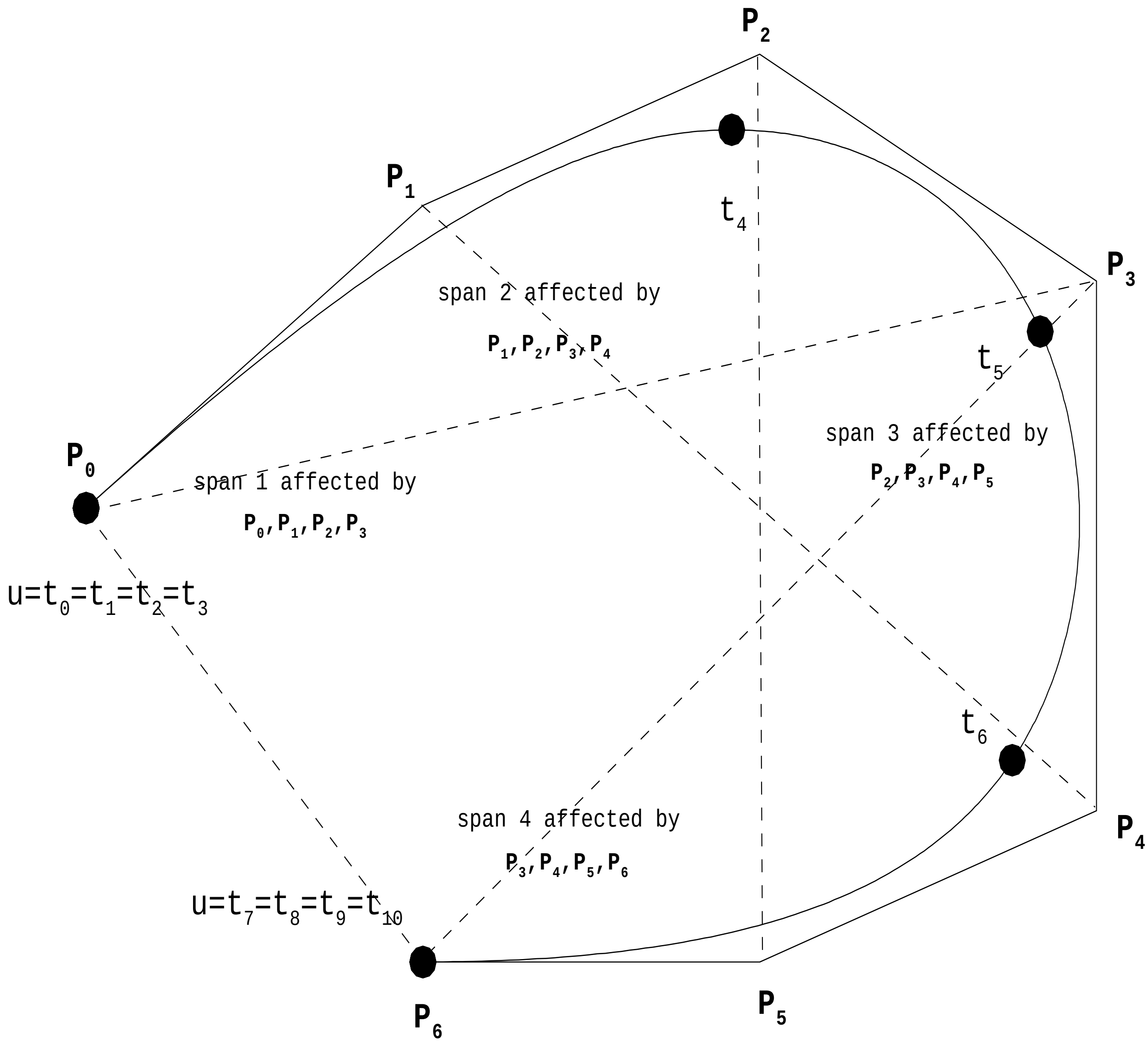

Figure 6.4: Example of local convex hull property of B-spline curve

Properties:

- Local support (eg. $\mathbf{P}_6$ affects span 4 only), see Figure 6.4
  i.e. $\mathbf{P}_i$ affects $[t_i, t_{i+k}]$ (k spans)
- Convex hull (stronger that Bézier) let $u \in [t_i, t_{i+1}]$, then $N_{j,k}(u) \neq 0$ for $j \in i-k+1, \cdots, i$ (k values)
$$\sum_{j=i-k+1}^{i} N_{j,k}(u) = 1, N_{j,k}(u) \geq 0$$
- Each span is in the convex hull of the k vertices contributing to its definition
- Consequence: k consecutive vertices are collinear $\rightarrow$ span is a straight line segment
- Variation diminishing property as for Bézier curves
- Exploit knot multiplicity to make complex curves

6.2.6 Special case $n = k - 1$

The B-spline curve is also a Bézier curve in this case.

$$\mathbf{T} = \{\underbrace{t_o = t_1 = \cdots = t_{k-1}}_{k\ equal\ knots} < \underbrace{t_k = t_{k+1} = \cdots = t_{2k-1}}_{k\ equal\ knots}\}$$

6.3 Derivatives

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(u) \tag{6.22}$$

$$\mathbf{P}'(u) = \sum_{i=1}^{n} \mathbf{d}_i N_{i,k-1}(u) \tag{6.23}$$

$$\mathbf{d}_i = (k-1)\frac{\mathbf{P}_i - \mathbf{P}_{i-1}}{t_{i+k-1} - t_i} \tag{6.24}$$

6.4 Approaches to design with B-spline curves

Design procedure A

1. Designer chooses knot vector and control points.
2. Designer displays curve and tweaks control points to improve curve.

Design procedure B

1. Designer starts with data points on or near curve.
2. Construct an interpolating/approximating B-spline curve.
3. Display curve and tweak control points to improve curve.

6.5 Interpolation of data points with cubic B-splines

Given N data points: $\mathbf{P}_i,\ i = 1, 2, \cdots, N$, and no other derivative data at the boundaries. The problem is to construct a cubic B-spline curve which interpolates (precisely matches) these data points.

Construction of knot vector

Let

$$\hat{u}_1 = 0$$

$$\hat{u}_{i+1} = \hat{u}_i + d_{i+1}$$

$$d_{i+1} = |\mathbf{P}_{i+1} - \mathbf{P}_i| \qquad i = 1, 2, \cdots, N-1$$

$$d = \sum_{i=2}^{N} d_i \to u_i = \frac{\hat{u}_i}{d} \qquad i = 1, 2, \cdots, N-1$$

Remove two knot values u_2 and u_{N-1} from knot vector to obtain proper number of degrees of freedom (instead of having to prescribe boundary conditions, which could be done if needed)

$$T = \{\underbrace{u_1 = u_1 = u_1 = u_1}_{4\ times} < \underbrace{u_3 \leq u_4 \leq \cdots \leq u_{N-2}}_{internal\ \ knots=N-4} < \underbrace{u_N = u_N = u_N = u_N}_{4\ times}\}$$

knots $= n + 4 + 1 = (N-4) + 4 + 4 = N + 4 \to n = N - 1$

$$\mathbf{R}(u) = \sum_{i=0}^{n} \mathbf{R}_i B_{i,4}(u) \qquad 0 \leq u \leq 1 \tag{6.25}$$

$$n = N - 1$$

Require that

$$\mathbf{R}(u_j) = \sum_{i=0}^{N-1} \mathbf{R}_i B_{i,4}(u_j) = \mathbf{P}_j \qquad j = 1, 2, \cdots, N \tag{6.26}$$

Solve for $\mathbf{R}_i$ (system is banded).

There are also other more sophisticated ways to choose knot vector and parameterization attempting to make u proportional to arc length.

6.6 Evaluation and subdivision of B-splines

6.6.1 De Boor algorithm for B-spline curve evaluation

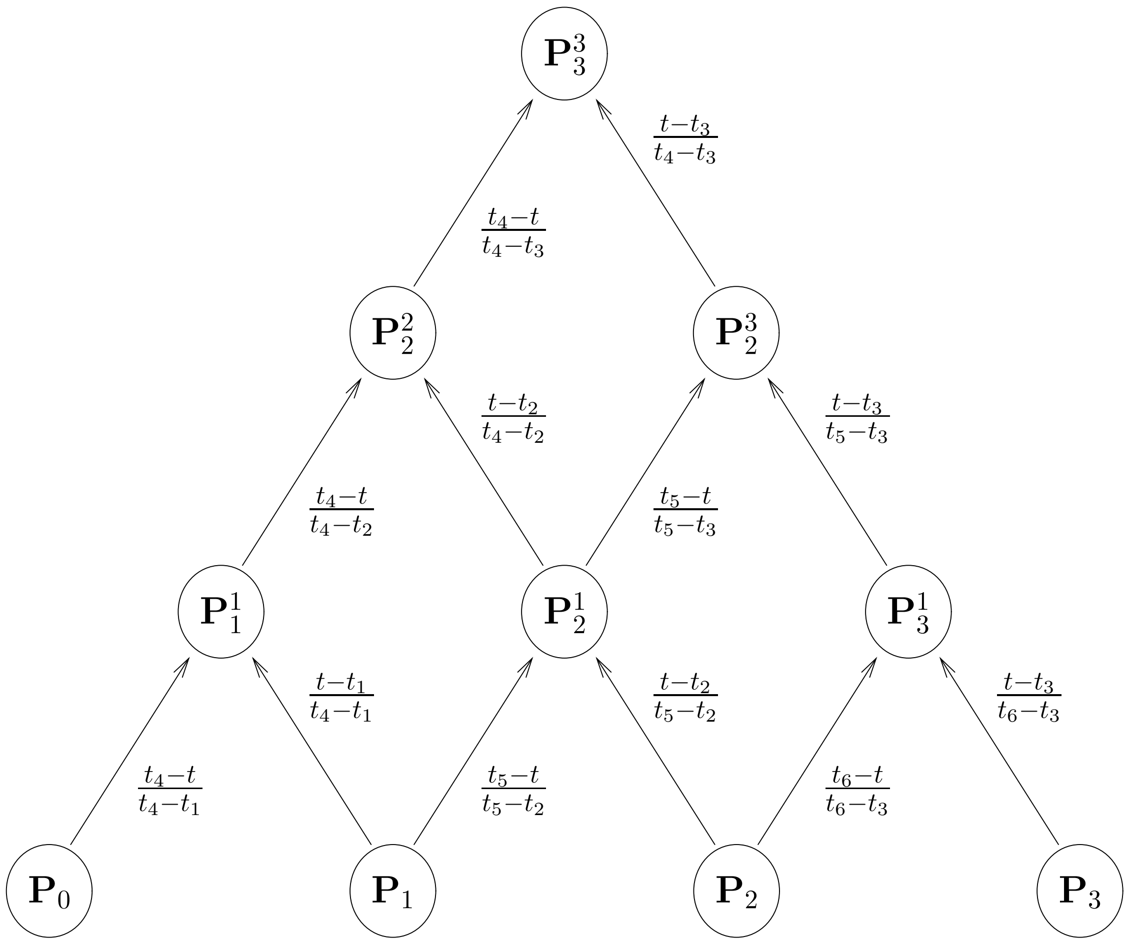

Figure 6.5: Diagram of de Boor subdivision algorithm over a cubic spline segment, where $t \in [t_3, t_4]$.

Evaluation of B-spline curves can be performed as follows (de Boor's algorithm):

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(u) \qquad t_0 \leq u \leq t_{n+k} \tag{6.27}$$

Let $u \in [t_l, t_{l+1})$ be a particular span. $N_{i,k}(u) \neq 0$ for $i = l-k+1, \cdots, l$

Let

$$\mathbf{P}_i^0 = \mathbf{P}_i \tag{6.28}$$

De Boor's recursive formula:

$$\mathbf{P}_i^j = (1-\alpha_i^j)\mathbf{P}_{i-1}^{j-1} + \alpha_i^j \mathbf{P}_i^{j-1}, \quad i \geq l-k+2 \tag{6.29}$$

where,

$$\alpha_i^j = \frac{u - t_i}{t_{i+k-j} - t_i} \tag{6.30}$$

$$\Rightarrow \boxed{\mathbf{P}_{k-1}^{k-1} = \mathbf{P}(t)}$$

This algorithm is shown graphically in Figure 6.5. An example of the evaluation of a point on an <u>cubic</u> ($k = 4$, $l = 3$) B-spline curve is shown in Figure 6.6.

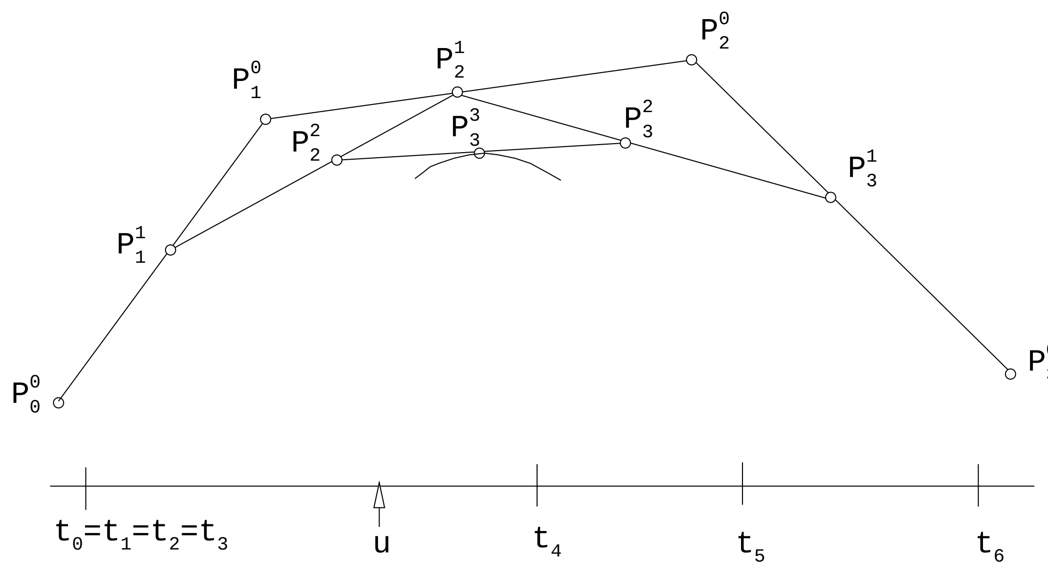

Figure 6.6: Evaluation of a point on a B-spline curve with the de Boor algorithm.

The algorithm shown in Figure 6.5 also permits the <u>splitting</u> of the segment into two smaller segments of the same order:

left polygon:	$P_0^0 P_1^1 P_2^2 P_3^3$
right polygon:	$P_3^3 P_3^2 P_3^1 P_3^0$

6.6.2 Knot insertion: Boehm's algorithm

$$\underbrace{\sum_{i=0}^{n} P_i N_{i,k}(t)}_{over\ [t_i]=[t_0,\cdots,t_l,t_{l+1},\cdots]} = \underbrace{\sum_{i=0}^{n+1} \hat{P}_i \hat{N}_{i,k}(t)}_{over\ [t_i]=[t_0,\cdots,t_l,\hat{t},t_{l+1},\cdots]} \tag{6.31}$$

$$\hat{P}_i = a_i P_i + (1 - a_i) P_{i-1} \tag{6.32}$$

$$a_i = \begin{cases} 1 & i \le l - k + 1 \\ 0 & l + 1 \le i \\ \frac{t - t_i}{t_{i+k-1} - t_i} & l - k + 2 \le i \le l \end{cases}$$

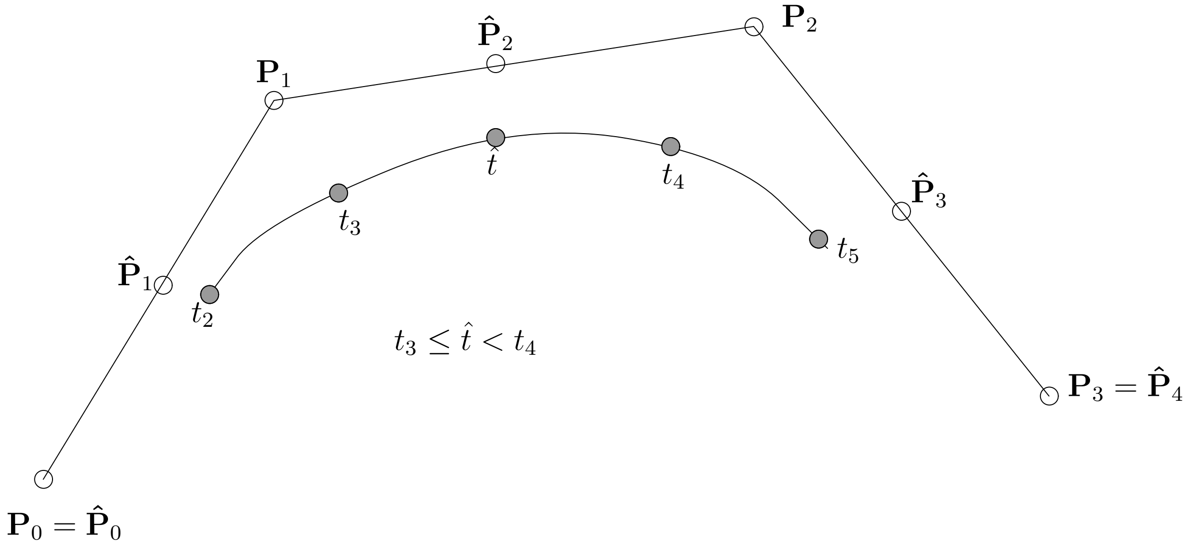

Figure 6.7: Control polygon of B-spline curve before and after knot insertion in interval $[t_3, t_4]$.

By repeatedly applying knot insertion algorithm, multiple knots within the knot vector can be created.

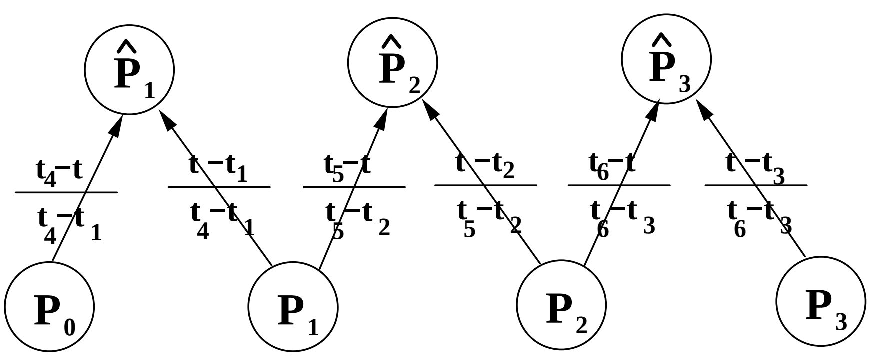

Figure 6.8: Boehm's algorithm diagram with added knot t in interval $[t_3, t_4]$.

6.7 Tensor product piecewise polynomial surface patches

Let

$$\mathbf{R}(u) = \sum_{i=0}^{n} \mathbf{R}_i F_i(u) \qquad A \leq u \leq B \tag{6.33}$$

be 3-D (or 2-D) curve expressed as linear combinations of basis functions $F_i(u)$.

Let this curve sweep a surface by moving and possibly deforming. This can be described by letting each $\mathbf{R}_i$ trace a curve

$$\mathbf{R}_i(v) = \sum_{k=0}^{m} \mathbf{a}_{ik} G_k(v) \qquad C \leq v \leq D \tag{6.34}$$

The resulting surface is a tensor product surface (see Figure 6.9).

$$\mathbf{R}(u, v) = \sum_{i=0}^{n} \sum_{k=0}^{m} \mathbf{a}_{ik} F_i(u) G_k(v) \qquad (u, v) \in [A, B] \times [C, D] \tag{6.35}$$

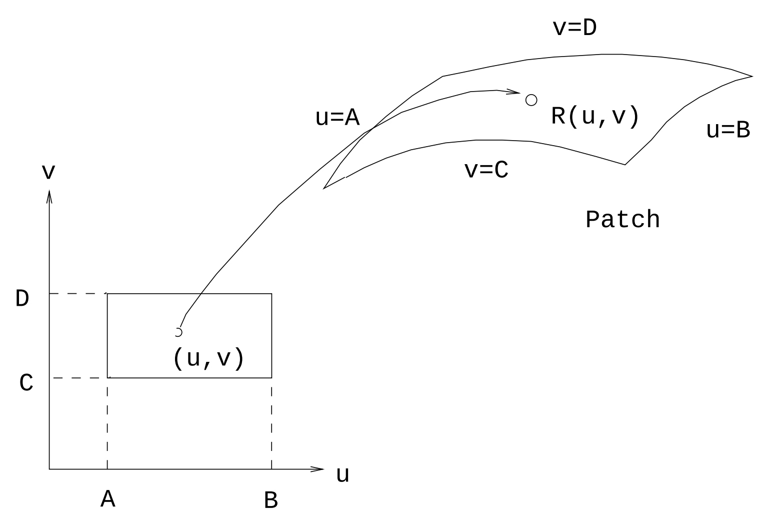

Figure 6.9: A tensor product patch.

Matrix (tensor) form:

$$\mathbf{R}(u,v) = [F_0(u) \cdots F_n(u)][\mathbf{a}_{ik}][G_0(v) \cdots G_m(v)]^T \tag{6.36}$$

The basis functions can be:

- Monomials → Ferguson patch
- Hermite → Hermite-Coons patch
- Bernstein → Bézier patch
- Lagrange → Lagrange patch
- Uniform B-splines → Uniform B-spline patch
- Non-uniform B-splines → Non-uniform B-spline patch.

6.7.1 *Example:* Bézier patch

$$\mathbf{R}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{R}_{ij} B_{i,n}(u) B_{j,m}(v) \qquad 0 \le u, v \le 1 \tag{6.37}$$

where $\mathbf{R}_{ij}$ are the control points creating a control polyhedron (net), see Figure 6.10.

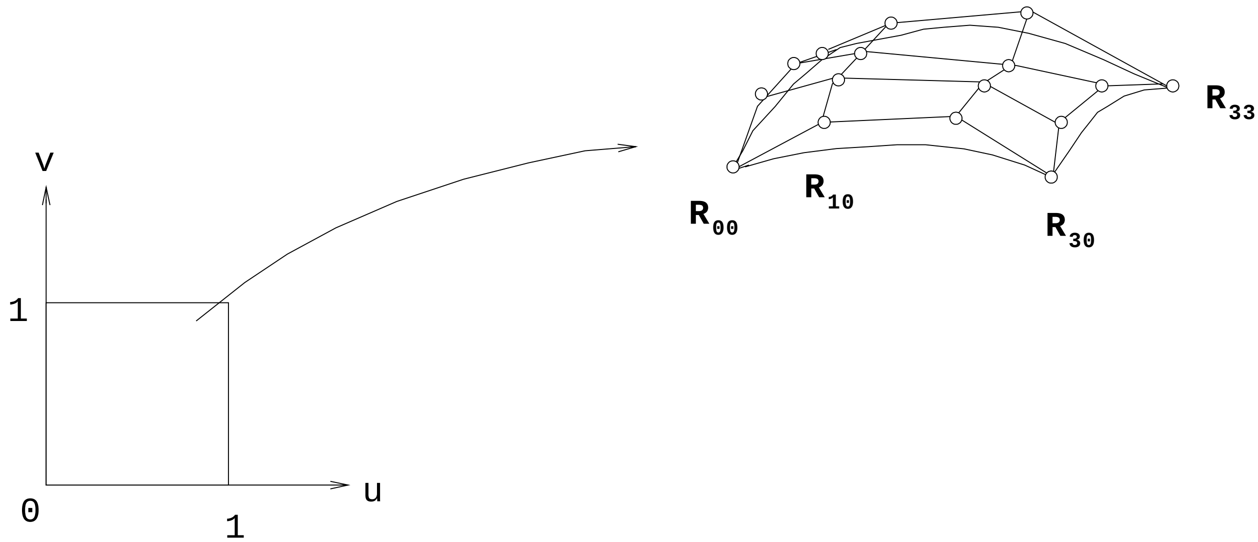

Figure 6.10: A Bézier patch.

Properties:

- Lines of $v = \bar{v}$ = constant (isoparameter lines) are Bézier curves of degree n with control points

$$\mathbf{Q}_i = \sum_{j=0}^{m} \mathbf{R}_{ij} B_{j,m}(\bar{v}). \tag{6.38}$$

- The boundary isoparameter lines have the same control points as the corresponding polyhedron points.
- The relation between the patch and Bézier net is affinely invariant (translation, rotation, scaling).
- Convex hull.
- No known variation diminishing property.
- The procedure to create piecewise continuous surfaces with Bézier patches is complex.

6.7.2 ***Example:*** B-Spline patch

$$\mathbf{R}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}\mathbf{R}_{ij}N_{i,k}(u)\mathbf{Q}_{j,l}(v) \qquad n\geq k-1 \; and \; m\geq l-1$$

$$U=\{\underbrace{u_0=u_1=\cdots=u_{k-1}}_{k\ equal\ knots}<u_k\leq u_{k+1}\leq\cdots\leq u_n<\underbrace{u_{n+1}=\cdots=u_{n+k}}_{k\ equal\ knots}\}$$

$$V=\{\underbrace{v_0=v_1=\cdots=v_{l-1}}_{l\ equal\ knots}<v_l\leq v_{l+1}\leq\cdots\leq v_m<\underbrace{v_{m+1}=\cdots=v_{m+l}}_{l\ equal\ knots}\}$$

Properties:

- Obeys same properties as a Bézier patch to which it reduces for $n=k-1$ and $m=l-1$ (isoparameter lines, boundaries, affine invariance, convex hull).
- Easy construction of complex piecewise continuous geometries.
- Local control:
  1. $\mathbf{R}_{ij}$ affects $[u_i,u_{i+k}]\times[v_j,v_{j+l}]$;
  2. Subpatch $[u_i,u_{i+1}]\times[v_j,v_{j+1}]$ affected by $\mathbf{R}_{p,q}$ where $(p,q)\ \in\ [i-k+1,\ \cdots\ i]\times[j-l+1,\ \cdots\ j]$.
- Strong convex hull – each subpatch lies in the convex hull of the vertices contributing to its definition.

6.7.3 *Example:* Composite Bézier patches

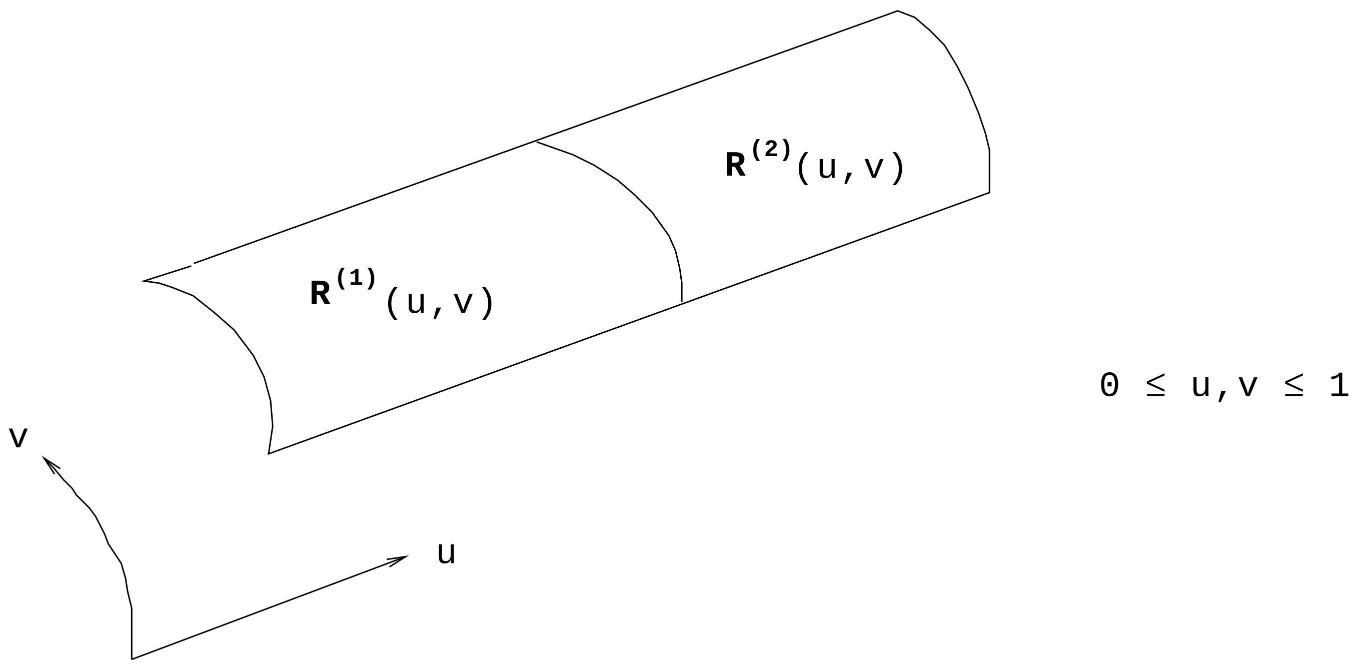

Figure 6.11: Composite Bézier patches.

1. Positional continuity

$$\mathbf{R}^{(1)}(1,v) = \mathbf{R}^{(2)}(0,v) \tag{6.39}$$

$$\mathbf{R}^{(1)}_{3i} = \mathbf{R}^{(2)}_{0i} \qquad i = 0,1,2,3 \tag{6.40}$$

2. Tangent plane (or normal) continuity

$$\mathbf{R}^{(2)}_u(0,v) \times \mathbf{R}^{(2)}_v(0,v) = \lambda(v)\mathbf{R}^{(1)}_u(1,v) \times \mathbf{R}^{(1)}_v(1,v) \tag{6.41}$$

(direction of surface normal continuous for $\lambda(v) > 0$)

Since

$$\mathbf{R}^{(2)}_v(0,v) = \mathbf{R}^{(1)}_v(1,v), \tag{6.42}$$

use

$$\underbrace{\mathbf{R}^{(2)}_u(0,v)}_{cubic\ in\ v} = \underbrace{\lambda(v)}_{constant}\ \underbrace{\mathbf{R}^{(1)}_u(1,v)}_{cubic\ in\ v} \tag{6.43}$$

to show

$$\mathbf{R}^{(2)}_{1i} - \mathbf{R}^{(2)}_{0i} = \lambda(\mathbf{R}^{(1)}_{3i} - \mathbf{R}^{(1)}_{2i}). \tag{6.44}$$

Therefore, collinearity of above polyhedron edges is required

6.8 Generalization of B-splines to NURBS

The acronym NURBS stands for non-uniform rational B-splines. These functions have the

- Same properties as B-splines, and
- Are capable of representing a wider class of geometries.

6.8.1 Curves and Surface Patches

NURBS curves are defined by

$$\mathbf{R}(u) = \frac{\sum_{i=0}^{n} w_i \mathbf{R}_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)} \tag{6.45}$$

where weights $w_i > 0$; if all $w_i = 1$, the integral piecewise polynomial spline case is recovered. This formulation permits exact representation of conics, eq. circle, ellipse, hyperbola.

NURBS surface patches are defined by

$$\mathbf{R}(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{ij} \mathbf{R}_{ij} N_{i,k}(u) Q_{j,l}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{ij} N_{i,k}(u) Q_{j,l}(v)} \tag{6.46}$$

where weights $w_{ij} \geq 0$. This formulation allows for exact representation of quadrics, tori, surfaces of revolution and very general free-form surfaces. If all $w_{ij} = 1$, the integral piecewise polynomial case is recovered.

6.8.2 *Example:* Representation of a quarter circle as a rational polynomial

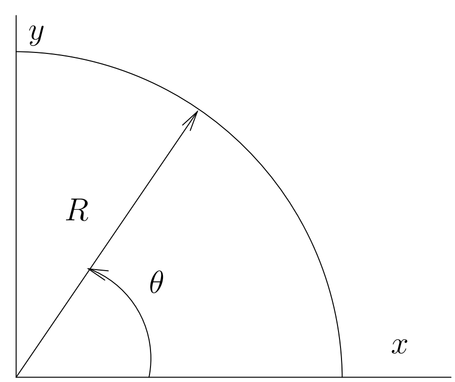

Figure 6.12: Quarter circle.

Consider a quarter circle (see Figure 6.12) described in terms of trigonometric functions by

$$\left.\begin{aligned} x &= R\, cos(\theta) \\ y &= R\, sin(\theta) \end{aligned}\right\} \qquad \text{for } 0 \leq \theta \leq \frac{\pi}{2}.$$

Setting $t = tan(\frac{\theta}{2})$ and using basic identities from trigonometry, we can express x and y as functions of t:

$$\left.\begin{array}{l} x(t) = R\frac{1-t^2}{1+t^2} \\ \\ y(t) = R\frac{2t}{1+t^2} \end{array}\right\} \qquad \text{for } 0 \ \leq \ t \ \leq \ 1. \tag{6.47}$$

For the conversion of Equation 6.47 to the Bézier representation, apply

$$[t^2 \ t \ 1] \begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix} = [t^2 \ t \ 1] \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

separately to numerators and denominators to obtain the Bézier form.

6.8.3 Trimmed patches

- $\mathbf{R}(u, v)$ is an untrimmed patch in the parametric domain $(u, v) \in [A, B] \times [C, D]$.
- Describe external loop as a set of edges (ie. curves in parameter space $\mathbf{r}_i(t) = [u_i(t_i)v_i(t_i)]$ – *eg.* external loop the Figure 6.13 if made up of $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5\}$, while the internal loop is made up of curve $\{\mathbf{r}_6\}$.

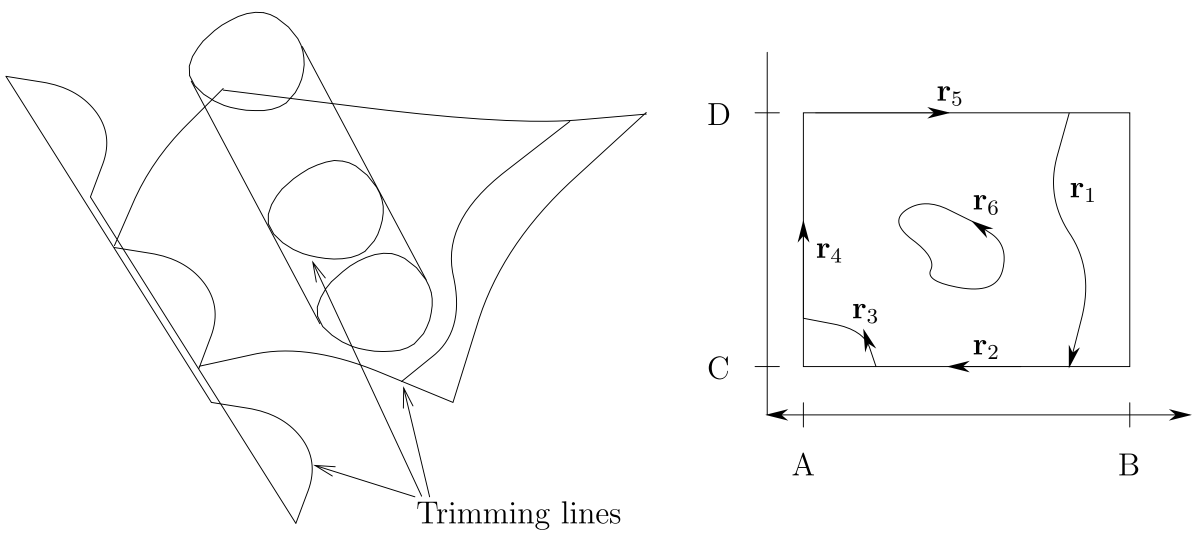

Figure 6.13: Trimmed surface patch.

6.9 Comparison of free-form curve/surface representation methods

Single span/patch	Composite
Ferguson (monomial or power basis) Hermite Bézier Lagrange	Bézier Cardinal or interpolatory spline B-spline

Table 6.1: Classification of free-form curve/surface representation.

Property	Ferguson	Hermite-Coons	Bézier	Lagrange	Composite Bézier	Cardinal Spline	B-Splines	NURBS
Easy geometric representation	low	med	high	med	high	Medium	high	high
Convex hull	no	no	yes	no	yes	no	yes	yes
Variation diminishing *	no	no	yes	no	yes	no	yes	yes
Easiness for creation	low	med	med	inappr.	med	high	high	high
Local control	no	no	no	no	yes but complex	no	yes	yes
Approximation ease	med	med	high	low	high	medium	high	high
Interpolation ease	med	med	med	high but inappr.	med	high	high	high
Generality	med	med	med	med	med	med	med	high
Popularity **	low	low	med	low	med	med	high	very high

Table 6.2: Comparison of curve/surface representation methods.

$*$ Variation diminishing property does not apply to surfaces.
$**$ Popularity in industry and STEP/PDES standards.

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] J. Owen. *STEP: An Introduction*. Information Geometers, Winchester, UK, 1993.

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

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