Curves and Surfaces

Intro. to Computer Graphics
Spring 2008, Y. G. Shin
Representation of Curves and Surfaces

- Key words: surface modeling, parametric surface, continuity, control points, basis functions, Bezier curve, B-spline curve
Why we need surface models?

- All shapes can be described in terms of points. But, it is impractical to enumerate the points that comprise a shape.
- We define shape indirectly through expressions that relate certain properties of points that comprise them.
Intrinsic and extrinsic properties

- Intrinsic properties
  - B has four sides
  - All four sides have equal length
  - All four angles are $90^o$, ......

- Extrinsic properties
  - two horizontal sides
  - two vertical sides vertices of B are at $P_0$, $P_1$, $P_2$ and $P_3$
Intrinsic and extrinsic properties

- Shape definitions that use extrinsic properties of the shape are dependent on the coordinate system used.
- a line: \( y = 3, \ 2 \leq x \leq 7 \)

\( \Leftarrow \) Axis dependency
**Axis Independence**

A mathematical representation of a line/curve is axis independent if its shape depends on only the *relative position of the points* defining its characteristic vectors and is independent of the coordinate system used.
Axis-independent shape definition

- Shape definitions that use intrinsic properties of the shape are axis-independent.

\[
x = (1-t)p_{1x} + tp_{2x} \\
y = (1-t)p_{1y} + tp_{2y} \\
0 \leq t \leq 1 \\
p_1 = (p_{1x}, p_{1y}) = (2,3) \\
p_2 = (p_{2x}, p_{2y}) = (7,3)
\]
Curve & Surface Models

- Explicit/implicit
- Parametric/non-parametric
- Approximation
  - polygon mesh: a collection of edges, vertices, and polygons
Nonparametric explicit representation

- \( x = x \)
- \( y = f(x) \)
- Successive values of \( y \) can be obtained by plugging in successive values of \( x \).
- Easy to generate polygons or line segments
- Single-valued function

\[
y = \sqrt{r^2 - x^2} \\
y = -\sqrt{r^2 - x^2}
\]
Nonparametric implicit representation

- $f(x, y, z) = 0$
- Define curves as solution of equation system
- E.g., a circle:

$x^2 + y^2 = r^2$
Nonparametric implicit representation

- algebraic quadric surfaces
  - $f$ is a polynomial of degree $\leq 2$

$$f(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k = 0$$

- sphere: $x^2 + y^2 + z^2 - 1 = 0$
- cylinder: $x^2 + y^2 - 1 = 0$
- corn: $x^2 + y^2 - z^2 = 0$
- paraboloid: $x^2 + y^2 + z = 0$
Nonparametric implicit representation

- Coefficients determine geometric properties
- Hard to render (have to solve non-linear equation system)
- Can represent closed or multi-valued curves
- Easy to classify point-membership

\[ \text{sphere}: x^2 + y^2 + z^2 - 1 = 0 \]
Parametric Curve

\[ x(t) = a_{13} t^3 + a_{12} t^2 + a_{11} t + a_{10} \]
\[ y(t) = a_{23} t^3 + a_{22} t^2 + a_{21} t + a_{20} \]
\[ z(t) = a_{33} t^3 + a_{32} t^2 + a_{31} t + a_{30} \]
Parametric Curve (Example)

line from $P_1 = (x_1, y_1)$ to $P_2 = (x_2, y_2)$

$$x = (1-t)x_1 + tx_2$$
$$y = (1-t)y_1 + ty_2 \quad 0 \leq t \leq 1$$

$P_1, P_2$ : control points

$t, 1-t$ : blending functions

unit circle

$$Q(u) = (\cos(u/2\pi), \sin(u/2\pi))$$
Parametric Curve Characteristics

- Simple to render
  - evaluate parameter function
- Hard to check whether a point lies on curve
  - have to compute the inverse mapping from \((x, y)\) to \(t\)
- Can represent closed or multi-valued curves
- Curve or surface can be easily translated or rotated
- Composite curves and surfaces can be formed by piecewise descriptions
Parametric Curve Characteristics

- No infinite slope problem

\[ Q'(u) = (-\sin(u/2\pi) / 2\pi, \cos(u/2\pi) / 2\pi, 0) \]

Implicit form: \( x^2 + y^2 - 1 = 0, \quad z = 0 \)

At \( x = 1, y = 0, \) the parametric derivative is \( (0, 1/2\pi, 0) \)

Implicit form \( f''(x, y, z) = -x / y \Rightarrow \infty \)
Parametric Curve Characteristics

- Not unit form
  (e.g.) a circle with radius 1 centered at the origin

\[
\begin{align*}
  &\begin{aligned}
    x &= \cos \theta \\
    y &= \sin \theta
  \end{aligned} \\
  \Rightarrow &\begin{aligned}
    x &= t \\
    y &= \sqrt{1 - t^2}
  \end{aligned}
\end{align*}
\]
Tangent line to a curve

- The straight line that gives the curve's slope at a point
- Deduced from the derivative of the curve at the point

$X(t), y(t)$

Derivative vector $(x'(t), y'(t))$
Cut curve into segments and represent each segment as a polynomial curve.

But how do we guarantee smoothness at the joints? (continuity problem)
Continuity

- Implies a notion of smoothness at the connection points
- **Parametric continuity**
  - We view the curve or surface as a function rather than a shape.
  - Matching the parametric derivatives of adjoining curve sections at their common boundary
  - You need a parameterization
Parametric Continuity

\( C^0 \) : a curve is continuous if it can be drawn without lifting the pencil from the paper. \((x, y, z)\) - values of the two curves agree.

\( C^1 \) : the derivative curve is also continuous, i.e., \((dx/\,dt, dy/\,dt, dz/\,dt)\) agree at their junction.

\( C^2 \) : the direction and magnitude of \(d^2/\,dt^2[Q(t)]\) are equal at the join point
Geometric Continuity

- Geometric continuity is defined using only the shape of the curve
- Geometric smoothness independent of parametrization

\( G^0 \): joining two segments at a common end point (\( = C^0 \))

\( G^1 \): a curve's tangent direction changes continuously
(difference equal, but necessarily the magnitude)
The order of polynomial curves

A polynomial of order \( k + 1 \) (\( \equiv \text{degree } k \)):

\[
P(u) = c_0 + c_1 u + c_2 u^2 + \cdots + c_k u^k
\]

- In computer graphics, usually degree = 3
  - Sufficient flexibility w/o much cost
  - The cubic is the lowest degree polynomial that gives \( C^1 \) and \( C^2 \) continuity
Curve models

- Curve fitting techniques (interpolation techniques)
  - pass through each and every data point
  - linear approximation, natural cubic spline

- Curve fairing techniques (approximation techniques)
  - few if any points on the curve pass through each and every data points
  - Hermite curve, Bezier curve, B-spline curve
Natural Spline Curves

- Motivated by loftman’s spline
  - Long narrow strip of wood or plastic
  - Shaped by lead weights (called ducks)
- a cubic spline curve, $q(u)$, composed of cubic polynomials that interpolate the points $P_0$, $P_1$, ..., $P_n$
- $C^{n-1}$ continuity can be achieved from splines of degree $n$
Natural Cubic Splines

- divide the interval \([a,b]\) into \(n\) intervals \([u_i, u_{i+1}]\), for \(i = 0\) to \(n-1\). The numbers \(u_i\) are called knots.

- The vector \([u_0, u_1, \ldots, u_{i-1}]\) is called a knot vector for the spline. If the knots are equidistantly distributed in the interval \([a, b]\), we say the spline is uniform, otherwise we say it is non-uniform.
Natural Cubic Splines

- Each cubic spline curve is determined by the position vectors, tangent vectors and parameter values

\[ q_i(u) = c_{i0} + c_{i1}u + c_{i2}u^2 + c_{i3}u^3 \]

\[ q_i(u_{i-1}) = p_{i-1} \quad \text{and} \quad q_i(u_i) = p_i \quad \text{for} \quad i = 1 \text{ to } n \]

\[ q_i'(u_i) = q'_{i+1}(u_i) \quad \text{for} \quad i = 1 \text{ to } n-1 \]

\[ q_i''(u_i) = q''_{i+1}(u_i) \quad \text{for} \quad i = 1 \text{ to } n-1 \]
Cubic Splines

- The polynomial coefficients of a cubic spline are dependent on all \( n \) control points.
  - A change in any one segment affects the entire curve.

- It is inconvenient to represent the curve directly using the coefficients \( c_i \).
  - The relationship between the shape of the curve and the coefficients is not clear or intuitive.
  - Rearrange the polynomial form into \textit{control points} and \textit{basis functions} (GEOMETRIC FORM).
Specifying Curves

- **Control Points**
  - A set of points that influence the curve's shape

- **Knots**
  - Points that lie on the curve
  - Subinterval endpoints
Hermite Curves

- Parametric curves
- Defined by two end points with the derivative of the curve at these points
Hermite Curves (cubic polynomial)

\[ x(t) = a_{13} t^3 + a_{12} t^2 + a_{11} t + a_{10} \]
\[ y(t) = a_{23} t^3 + a_{22} t^2 + a_{21} t + a_{20} \]
\[ z(t) = a_{33} t^3 + a_{32} t^2 + a_{31} t + a_{30} \]

\[ P(t) = [x(t) \quad y(t) \quad z(t)] = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \]

where \( a_i = (a_{i1}, a_{i2}, a_{i3}) \)

\[ P(0) = a_0 \]
\[ a_0 = P(0) \]
\[ P(1) = a_3 + a_2 + a_1 + a_0 \]
\[ a_1 = P(0) \]
\[ P'(0) = a_1 \]
\[ a_2 = -3P(0)^2 + 3P(1) - 2P'(0) - P'(1) \]
\[ P'(1) = 3a_3 + 2a_2 + a_1 \]
\[ a_3 = 2P(0) - 2P(1) + P'(0) + P(1) \]
Hermite Curves

\[ P(t) = B_1(t)P(0) + B_2(t)P(1) + B_3P'(0) + B_4P'(1) \]

\[
B_1(t) = 2t^3 - 3t^2 + 1 \quad B_2(t) = -2t^3 + 3t^2 \\
B_3(t) = t^3 - 2t^2 + t \quad B_4(t) = t^3 - t^2
\]

\( B_i(t) \): blending functions

\( P(0), P(1), P'(0), P'(1) \)

: geometric coefficients
Hermite Curves

\[ P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P'_0 \\ P'_1 \end{bmatrix} = T \cdot M_H \cdot G_H (= B \cdot G_H) \]

\[ M_H \text{: Hermite basis matrix} \]
\[ G_H \text{: Hermite geometry vector} \]
Represent Polynomials with basis functions

- Polynomials including degree $k$ forms a vector space $P^{k+1}$
- Specify a curve $P(u)$ as a position in the vector space $P^{k+1}$ via the coordinate $(p_0, \cdots, p_k)$

and the basis $(1, t, t^2, \cdots, t^k)$

$$b_i(t) = t^k, \quad 0 \leq i \leq k : \text{basis functions}$$

$$p_0, \cdots, p_k : \text{control points}$$

$$\Rightarrow Q(t) = \sum_{i=0}^{k} p_i b_i(t)$$
Properties shared by most useful bases

- **Convex hull property**
  
  if \( \sum_{i=0}^{k} b_i(t) = 1 \) and basis functions are not negative over the interval they are defined then any point on the curve is a weighted average of its control points.

  \[ P_0, P_1, P_2, P_3 \]

  \[ \Rightarrow \] no points on the curve lies outside the polygon formed by joining the control points together

  \[ \Rightarrow \] inexpensive means for calculating the bound of a curve or surface in space
Properties shared by most useful bases

- **Affine invariance** - any linear transformation or translation of the *control points* defines a new curve that is the just the transformation or translation of the original curve. (Perspective transform is not affine.)

- **Variation diminishing** - no straight line intersects a curve more times than it intersects the curve's control polyline. It implies that the complexity (i.e., turning and twisting) of the curve is no more complex than the control polyline.
Bezíer Curves

- Developed by Pierre Bézier in the 1970's for CAD/CAM operations. (PostScript drawing model)
- Represent a polynomial segment as

\[ P(t) = \sum_{i=0}^{n} p_i J_{n,i}(t), \quad 0 \leq t \leq 1 \]

\[ J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \]

- \( J_{n,i}(t) \) are the Bernstein functions
  - basis or blending function of degree \( n \)
  - used to scale or blend the control points
Beziers blending functions

Note that $\sum_{i=0}^{n} J_{n,i}(t) = 1 \rightarrow$ convex hull property
Beziers Curves (example)

Given \( p_0 (1,1), p_1 (2,3), p_2 (4,3) \) and \( p_3 (3,1) \), find the Bezier curve.

\[
P(t) = \sum_{i=0}^{n} J_{n,i}(t), \quad 0 \leq t \leq 1
\]

→ Since there are four vertices, \( n = 3 \).

\[
J_{3,0}(t) = (1-t)^3 \quad J_{3,1}(t) = 3t (1-t)^2
\]
\[
J_{3,2}(t) = 3t^2 (1-t) \quad J_{3,3}(t) = t^3
\]

Thus, \( P(t) = p_0 J_{3,0} + p_1 J_{3,1} + p_2 J_{3,2} + p_3 J_{3,3} \)

\[
= (1-t)^3 \ p_0 + 3t(1-t)^2 \ p_1 + 3t^2(1-t) \ p_2 + t^3 \ p_3
\]
Beziers Curves (Matrix Form)

\[ P(t) = T \cdot M_B \cdot G = B \cdot G \]

where \( G = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}^T \)

\( B = \begin{bmatrix} J_{n,0} & J_{n,1} & \cdots & J_{n,n} \end{bmatrix} \)

\[ P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \]
Beziers Curves

- The Bezier curve of order \( n+1 \) (degree \( n \)) has \( n+1 \) control points.
- We can think a Bezier curve as a weighted average of all of its control points

Linear (n=1) : \[ P(t) = (1 - t)P_0 + tP_1 \]

Quadratic (n=2) : \[ P(t) = (1 - t)[(1 - t)P_0 + tP_1] + t[(1 - t)P_1 + tP_2] \]
\[ P(t) = (1 - t)^2 P_0 + 2(1 - t)tP_1 + t^2P_2 \]

Cubic (n=3) : \[ P(t) = (1 - t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2 (1-t)p_2 + t^3 p_3 \]
Bezier Curves
Beziers Curves

- A curve that is made of several Bézier curves is called a composite Bézier curve or a Bézier spline curve.
- Tangential continuity between Bézier segments:
  \[(Q_3 - Q_2) = k(R_1 - R_0)\]
- Continuity conditions create restrictions on control points

**C^1 continuity**

\[
Q'(1) = R'(0) \\
\Rightarrow (Q_3 - Q_2) = (R_1 - R_0) \\
\Rightarrow R_1 = Q_3 + R_0 - Q_2 \\
= Q_3 + (Q_3 - Q_2)
\]
Bezier Spline Curves

\( C^2 \) continuous two cubic Bezier segments \( V(t) \) and \( W(t) \) with the control points \( (V_0, V_1, V_2, V_3) \) and \( (W_0, W_1, W_2, W_3) \).

- For cubic Bezier spline:
  \[ V'(0) = 3(V_1 - V_0), \quad V'(1) = 3(V_3 - V_2), \]
  \[ V''(0) = 6(V_0 - 2V_1 + V_2), \quad V''(1) = 6(V_1 - 2V_2 + V_3) \]
- Continuity at the junction point \( \Rightarrow W_0 = V_3 \).
- Continuity of the first derivative \( W'(0) = V'(1) \)
  \( \Rightarrow W_1 = 2V_3 - V_2 \)
  \( i.e. W_1 \text{ depends on } V_2 \text{ & } V_3 \)
- Continuity of the second derivative \( W''(0) = V''(1) \)
  \( \Rightarrow W_0 - 2W_1 + W_2 = V_1 - 2V_2 + V_3 \)
  \( \Rightarrow W_2 = 2W_1 - (2V_2 - V_1) \)

Only one control point \( W_3 \) of the Bezier curve \( W(t) \) is really free.
Characteristics of Bezier Curves

- Convex hull
- Affine invariance
- Variation diminishing
- The degree of the polynomial defining the curve segment is one less than the number of defining control points.
- In CAGD applications, a curve may have a so complicated shape that it cannot be represented by a single Bézier cubic curve
- Global control (disadv.) : change a control point affects the continuity of the curve.
The de Casteljau Algorithm

- Evaluation of the Bezier curve function
- Repeated linear interpolation
- Example of a quadratic (degree 2) Bezier curve

3 control points

interpolate \( t = 0.2 \)
The de Casteljau Algorithm

The point on the curve

Degree=3 and $t=0.25$

repeating the procedure
Parametric Surface

- Extend 2D parametric representation
  - increase the number of parameters from one to two, \((s,t)\) in order to address each point in the 2D spaces.
  - express the 3D structure of the curved 2D surface by introducing a parameter \(z\) coordinate, \(z(s,t)\), i.e., a patch

\[
x = f_x(s,t), \quad y = f_y(s,t), \quad z = f_z(s,t).
\]
\[
0 \leq s,t \leq 1
\]
Bicubic Bezier Surface

- Bezier patch: 16 control points define one patch
- ease of interactivity & representation

\[ P(s, t) = \sum_{i=0}^{n} \binom{n}{i} (1 - s)^{n-i} s^i \sum_{j=0}^{n} \binom{n}{j} (1 - t)^{n-j} t^j P_{i,j} \]

\[ P(u, v) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix} \]

where \( B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \)

\[ P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix} \]
B-Splines Curves

\[
Q(u) = \sum_{k=0}^{n} P_k B_{k,d}(u)
\]

- \(P_k\): an input set of \(n+1\) control points
- \(B_{k,d}\): blending function of degree \(d-1\)

- The polynomial curve has degree \(d-1\) and \(C^{d-2}\) continuity over the range of \(u\)
- For \(n+1\) control points, the curve is described with \(n+1\) blending functions
- The range of \(u\) is divided into \(n+d\) subintervals by the \(n+d+1\) knot values
B-Splines Curves

A cubic b-spline which consists of three curve segments

\[ n=5, \ d=4 \]

5+1 control points

5+1 blending functions with degree 4-1

5+4+1 knots
Cubic B-Splines

Each control point is associated with a unique blending function.

⇒ (Local control) Each control point affects the shape of a curve only over a range of a parameter values, a curve sections, where its associated basis function is nonzero.
B-Splines Curves

- Knot vector: a set of subinterval endpoints in non-decreasing sequence

\[ U = \{ u_0, u_1, \ldots, u_{n+d} \} \]

- uniform, open uniform, nonuniform B-splines.
B-Splines Basis Functions

Cox-deBoor Algorithm

generate the basis functions recursively

\[
B_{k,1}(u) = 1, \quad \text{if} \quad u_k \leq u \leq u_{k+1}
\]

\[
= 0, \quad \text{otherwise}
\]

\[
B_{k,d}(u) = \frac{u - u_k}{u_{k+d+1} - u_k} B_{k,d-1}(u) + \frac{u_{k+d} - u}{u_{k+d} - u_{k+1}} B_{k+1,d-1}(u)
\]
Uniform cubic B-spline basis functions

- Knots are spaced at equal intervals of parameter. e.g., \{0,1,2,3,4,5,6,7,8,9\}
- Bell-shaped basis function
- Each blending function $B_{k,4}$ is defined over four subintervals starting at knot value $u_k$
Basis functions of Uniform Cubic B-splines

In $u_i \leq u \leq u_{i+1}$, we get basis functions by substituting $0 \leq u \leq 1$.

\[
\begin{align*}
B_0(u) &= \frac{1}{6} (1 - u)^3 \\
B_1(u) &= \frac{1}{6} (3u^3 - 6u^2 + 4) \\
B_2(u) &= \frac{1}{6} (-u^3 + 3u^2 + 3u + 1) \\
B_3(u) &= \frac{1}{6} u^3
\end{align*}
\]
Uniform Cubic B-splines

ith cubic segment

\[ Q_i(u) = \sum_{k=0}^{3} p_{i-3+k} B_{i-3+k}(u) \]

\( k \): local control point index
\( u \): local control parameter, \( 0 \leq u \leq 1 \)

A cubic B-spline is a series of \( m-2 \) curve segments, \( Q_3, Q_4, \ldots, Q_m \), that approximate a series of \( m+1 \) control points \( P_0, P_1, \ldots, P_m \), \( m \geq 3 \)
Uniform Cubic B-splines

$Q_3$ is defined $P_0 P_1 P_2 P_3$ which are scaled by $B_0 B_1 B_2 B_3$

$Q_4$ is defined $P_1 P_2 P_3 P_4$ which are scaled by $B_1 B_2 B_3 B_4$

$Q_5$ is defined $P_2 P_3 P_4 P_5$ which are scaled by $B_2 B_3 B_4 B_5$
Uniform Quadratic B-splines

- Let \( d=n=3 \), we need \( n+d+1 = 7 \) knot values: \{0,1,2,3,4,5,6\}.
- Get blending functions using Cox-deBoor Algorithm

\[
B_{0,3}(u) = \frac{u}{2} B_{0,2}(u) + \frac{3-u}{2} B_{1,2}(u)
\]

Read text book!!

\[
Q_i(u) = \sum_{k=0}^{2} p_{i-2+k} B_{i-2+k,3}(u)
\]
Uniform B-splines (Example)

- The curve is defined from \( u_{d-1}=2 \) to \( u_{n+1}=4 \)
- We can get starting and ending positions (boundary condition) of the curve:

\[
Q_{begin} = \frac{1}{2} (p_0 + p_1), \quad Q_{end} = \frac{1}{2} (p_2 + p_3)
\]

by applying \( u=2 \) and \( u=4 \) to the \( Q(u) \).
In general, weighted average of \( d-1 \) control points.

- Derivatives at the starting and ending position

\[
Q'_{begin} = p_1 - p_0, \quad Q'_{end} = p_3 - p_2
\]
Uniform Cubic B-splines

- Using a general cubic polynomial expression and the following boundary conditions:

\[ Q(0) = \frac{1}{6}(p_0 + 4p_1 + p_2) \quad Q(1) = \frac{1}{6}(p_1 + 4p_2 + p_3) \]

\[ Q'(0) = \frac{1}{2}(p_2 - p_0) \quad Q'(1) = \frac{1}{2}(p_3 - p_1) \]

We can get a matrix formulation:

\[ Q_i(u) = \left[u^3 \quad u^2 \quad u \quad 1\right] \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \]
Convex Hull Property of B-Splines Curves

- B-spline curve of degree $d-1$ must lie within the union of all such convex hulls formed by taking $d$ successive defining polygon vertices.
Uniform Cubic B-splines

- The effect of multiple control points
  ⇒ interpolate control points but the loss of continuity.

   multiplicity
   1   $G_2$ continuous
   2   $G_1$ continuous
   3   $G_0$ continuous
Non-uniform B-splines

- Non-uniform interval of knot values
- To permit the spline to interpolate control points by inserting multiple knots
- Knot vector is any non-decreasing sequence of knot values.
Non-uniform B-splines

- knot vector: \([0,0,0,0,1,2,3,3,3,3]\)
  - nine segment: \(Q_0, Q_1, \ldots, Q_8\)
  - \(Q_0, Q_1, Q_2, Q_6, Q_7,\) and \(Q_8\) are reduced to a single point
  - \(Q_3, Q_4,\) and \(Q_5\) are defined over the range \(0 \leq u \leq 3\)
- knot vector \([0,0,0,0,0,1,1,1,1,1]\) \(\equiv\) Bezier curve
  - \(P_0, \ldots, P_3\) control points
Given the following information:

- a set of $m+1$ rows and $n+1$ column control points $p_{i,j}$, where $1 \leq i \leq m$, $1 \leq j \leq n$;
- a knot vector of $h + 1$ knots in the $u$-direction, $U = (u_0, u_1, u_2, \ldots, u_h)$
- a knot vector of $k + 1$ knots in the $v$-direction, $V = (v_0, v_1, v_2, \ldots, v_k)$
- the degree $p$ in the $u$-direction; and the degree $q$ in the $v$-direction;
B-Spline Surfaces

The B-spline surface defined by these information is the following:

\[
Q(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_{i,p}(u) B_{j,q}(v) p_{ij}
\]
B-Spline Surfaces

The coefficient of control point $p_{i,j}$ is the product of two one-dimensional B-spline basis functions, one in the $u$-direction, $B_{i,p}(u)$, and the other in the $v$-direction, $B_{j,q}(v)$. All of these products are two-dimensional B-spline functions. The following figures show the basis functions of control points $p_{2,0}$, $p_{2,1}$, $p_{2,2}$, $p_{2,3}$, $p_{2,4}$ and $p_{2,5}$.
NURBS

- NURBS (non-uniform rational B-spline)
  - Adding some relative weight to the control point for extra control facility
  - Can represent more various curves such as circles and cylinders
  - More useful for interpolation
  - Invariant w.r.t a projective transformation
\[ P^w_i = (w_i x_i, w_i y_i, w_i z_i, w_i) \]

\[
\begin{align*}
P(u) &= \frac{\sum_{i=0}^{n} p_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)} \\
&= \sum_{i=0}^{n} p_i R_{i,k}(u)
\end{align*}
\]

\[
R_{i,k}(u) = \frac{B_{i,k}(u)w_i}{\sum_{j=0}^{n} B_{j,k}(u)w_i}
\]

\( w_i = \text{weight} \)

- \( w_i = 1 \) for all \( i \) \( \Rightarrow R_{i,k}(u) = B_{i,k}(u) \)
- extra shape parameter
- \( w_i \) increase \( \Rightarrow \) curve is pulled toward control point \( P_i \)
**Forward-differencing method**: to plot a curve or a surface, a polynomial must be evaluated at successive \( t \) values with fixed increments.

For \( P(t) = at^3 + bt^2 + ct + d, 0 \leq t \leq 1 \)

\[
P_i = P(i/n) = a(i/n)^3 + b(i/n)^2 + c(i/n) + d
\]

\[
P_{i+1} - P_i = a\{(i+1)/n)^3 - (i/n)^3\}
+ b\{(i+1)/n)^2 - (i/n)^2\} + c\{(i+1)/n) - (i/n)\}
\]

\[
\Delta_{1,i} = \frac{a}{n^3} (3i^2 + 3i + 1) + \frac{b}{n^2} (2i + 1) + \frac{c}{n}
\]

\[
\Delta_{2,i} = \Delta_{1,i+1} - \Delta_{1,i} = 6(i+1) \frac{c}{n^3} + \frac{2b}{n^2}
\]

\[
\Delta_{3,i} = \Delta_{2,i+1} - \Delta_{2,i} = \frac{6a}{n^3}
\]
Drawing Curves

- Recursive subdivision
  - stops when the control points get sufficiently close to the curve
  - need flatness test
  - Bezier curve - divide the control points
Drawing Bezier Curves

\[ R_0 = Q_0 \]
\[ R_1 = \frac{Q_0 + Q_1}{2} \]
\[ R_2 = \frac{R_1}{2} + \frac{Q_1 + Q_2}{4} \]
\[ R_3 = \frac{R_2 + S_1}{2} \]

\[ S_0 = R_3 \]
\[ S_1 = \frac{Q_1 + Q_2}{4} + \frac{S_2}{2} \]
\[ S_2 = \frac{Q_2 + Q_3}{2} \]
\[ S_3 = Q_3 \]
Comparison of Surface

<table>
<thead>
<tr>
<th>Comparison of Four Different Forms of Parametric Cubic Curves</th>
<th>Hermite</th>
<th>Bézier</th>
<th>Uniform B-Spline</th>
<th>Nonuniform B-spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convex hull defined by control points</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Interpolates some control points</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Interpolates all control points</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Ease of subdivision</td>
<td>Good</td>
<td>Best</td>
<td>Average</td>
<td>High</td>
</tr>
<tr>
<td>Continuities inherent in representation</td>
<td>$C^0$</td>
<td>$C^0$</td>
<td>$C^2$</td>
<td>$C^2$</td>
</tr>
<tr>
<td></td>
<td>$G^0$</td>
<td>$G^0$</td>
<td>$G^2$</td>
<td>$G^2$</td>
</tr>
<tr>
<td>Continuities achieved easily</td>
<td>$C^1$</td>
<td>$C^1$</td>
<td>$C^2$*</td>
<td>$C^2$*</td>
</tr>
<tr>
<td></td>
<td>$G^1$</td>
<td>$G^1$</td>
<td>$G^2*$</td>
<td>$G^2*$</td>
</tr>
<tr>
<td>Number of parameters controlling a curve segment</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

*Except for special case discussed in Section 9.2.