Parametric Curves

Computer Graphics and Visualization

Pedher Johansson
Department of Computer Science
Fall 2016

Parametric Polynomial Curves

- Parametric representation:

$$x = x(u), \quad y = y(u), \quad z = z(u), \qquad 0 \leq u \leq 1$$

- Polynomials define a curve segment.

$$\begin{aligned}
x(u) &= c_0^x + c_1^x u + \ldots + c_{n-1}^x u^{n-1} + c_n^x u^n, \\
y(u) &= c_0^y + c_1^y u + \ldots + c_{n-1}^y u^{n-1} + c_n^y u^n, \\
z(u) &= c_0^z + c_1^z u + \ldots + c_{n-1}^z u^{n-1} + c_n^z u^n,
\end{aligned}$$

Parametric Cubic Curves

Cubic are a good degree because:

- It is high enough to allow some flexibility in the curve design.
- It is not so high that wiggles creep into the curve.
- It is the lowest degree that can specify a non-planar space curve.
- A compromise between flexibility and speed of computation.

Parametric Cubic Curves

- Parametric representation:

$$x = x(u), \quad y = y(u), \quad z = z(u), \qquad 0 \leq u \leq 1$$

- Polynomials define a curve segment.

$$\mathbf{p}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}^T = \begin{bmatrix} c_0^x + c_1^x u + c_2^x u^2 + c_3^x u^3 \\ c_0^y + c_1^y u + c_2^y u^2 + c_3^y u^3 \\ c_0^z + c_1^z u + c_2^z u^2 + c_3^z u^3 \end{bmatrix}^T$$

Parametric Cubic Curves

- $\mathbf{p}(u) = \mathbf{c}_0 + \mathbf{c}_1 u + \mathbf{c}_2 u^2 + \mathbf{c}_3 u^3 = \mathbf{u}^T \cdot \mathbf{c}$, where
  $\mathbf{u} = \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}$, $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix}$ and $\mathbf{c_k} = \begin{bmatrix} c_k^x \\ c_k^y \\ c_k^z \end{bmatrix}$

Tangent Vector

- The parametric tangent vector of the curve is then calculated as
$$\frac{d}{du}p(u) = \frac{d}{du}\mathbf{u} \cdot \mathbf{c} = [0\ 1\ 2u\ 3u^2] \cdot \mathbf{c}$$
- Needed for continuity

Geometric Continuity

- G^0 when two curve segments join (same coordinate position).
- G^1 when two curve segments have tangent vectors with equal direction at the join point (1st derivative).
- G^2 when both first and second parametric derivative of the curve sections are proportional at their boundary.

Parametric Continuity

C^0 when two curve segments join (same coordinate position).

C^1 when the tangent vectors at the curves join point are equal (direction and magnitude) (1st derivative).

C^n when direction and magnitude of through the nth derivative are equal at the join point.

$$\frac{d^n}{du^n}[Q^u]$$

- In general, C^1 continuity implies G^1, but the converse is generally not true.
- C^n continuity is more restrictive than G^n continuity.

Interpolating Multiple Segments

- Easy to get C^0-continuity.

$$\mathbf{p}_3 = \mathbf{q}_0$$

- Difficult to get C^1 and C^2 continuity.

Basis Matrix

- The coefficient matrix $\mathbf{c}$ can be written as $\mathbf{c} = \mathbf{M} \cdot \mathbf{p}$, where $\mathbf{M}$ is a 4×4 basis matrix. $\mathbf{p}$ is a four element matrix of geometric constraints (geometry matrix).

$$p(u) = \mathbf{u}^T \cdot \mathbf{M} \cdot \mathbf{p}$$

$$\mathbf{p}(u) = \mathbf{u}^T \cdot \mathbf{c} = \mathbf{u}^T \cdot \begin{bmatrix} m_{11} & m_{21} & m_{31} & m_{41} \\ m_{12} & m_{22} & m_{32} & m_{42} \\ m_{13} & m_{23} & m_{33} & m_{43} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

Blending Functions

- The blending function $\mathbf{b}(u)$ is given by $\mathbf{b}(u) = \mathbf{u} \cdot \mathbf{M}$

$$\mathbf{p}(u) = \mathbf{b}(u)^T \cdot \mathbf{p} = \sum_{k=0}^{3} b_k(u) \cdot p_k$$

- A curve segment $\mathbf{p}(u)$ is defined by constraints on points, tangent vectors and continuity between curve segments etc.

Types of Cubic Curves

Interpolation defined by four interpolated points.

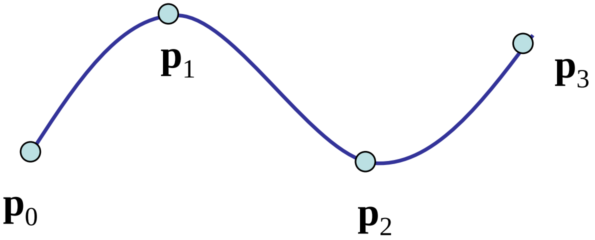
Hermite defined by two endpoints and two endpoint tangent vectors.

Bézier defined by two endpoints and two other points that control the endpoint tangent vector.

B-Spline defined by four control points and has C^1 and C^2 continuity at the join points. Does not generally interpolate the control points.

Interpolation

- Given four data (control) points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ determine cubic $\mathbf{p}(u)$ which passes through them



$$\mathbf{p}(u) = \mathbf{u}^T\mathbf{c} = \mathbf{u}^T \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix} = \mathbf{u}^T\mathbf{M}_I\mathbf{p}$$

- Must find $\mathbf{M}_I$ given the four input points in $\mathbf{p}$ and that $\mathbf{c} = \mathbf{M}_I\mathbf{p}$

Interpolation

- Recall that $u = [0, 1]$
- Apply the interpolating conditions at $u = 0,\ 1/3,\ 2/3,\ 1$

$$
\begin{aligned}
\mathbf{p}_0 &= \mathbf{p}(0) &= \mathbf{c}_0 \\
\mathbf{p}_1 &= \mathbf{p}(\tfrac{1}{3}) &= \mathbf{c}_0 + \tfrac{1}{3}\mathbf{c}_1 + (\tfrac{1}{3})^2\mathbf{c}_2 + (\tfrac{1}{3})^3\mathbf{c}_3 \\
\mathbf{p}_2 &= \mathbf{p}(\tfrac{2}{3}) &= \mathbf{c}_0 + \tfrac{2}{3}\mathbf{c}_1 + (\tfrac{2}{3})^2\mathbf{c}_2 + (\tfrac{2}{3})^3\mathbf{c}_3 \\
\mathbf{p}_3 &= \mathbf{p}(1) &= \mathbf{c}_0 + \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3
\end{aligned}
$$

or in matrix form with $\mathbf{p} = [\mathbf{p}_0\ \mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3]^T$

$$
\begin{bmatrix} \mathbf{p}(0) \\ \mathbf{p}(\frac{1}{3}) \\ \mathbf{p}(\frac{2}{3}) \\ \mathbf{p}(1) \end{bmatrix} = \mathbf{A}\mathbf{c} = \mathbf{p} \qquad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & (\frac{1}{3}) & (\frac{1}{3})^2 & (\frac{1}{3})^3 \\ 1 & (\frac{2}{3}) & (\frac{2}{3})^2 & (\frac{2}{3})^3 \\ 1 & 1 & 1 & 1 \end{bmatrix}
$$

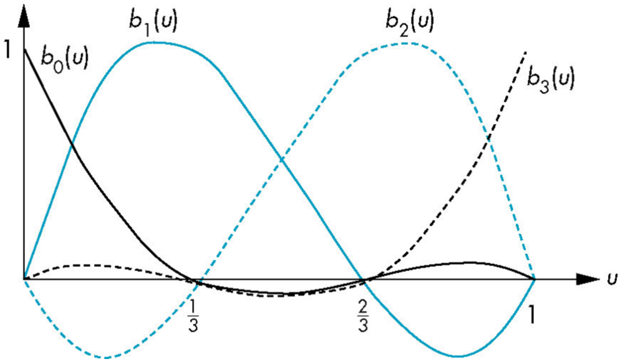
Interpolation

- Solving $\mathbf{Ac} = \mathbf{p}$ we find that $\mathbf{c} = \mathbf{A}^{-1}\mathbf{p} = \mathbf{M}_I\mathbf{p}$ where $\mathbf{M}_I$ is the interpolation matrix

$$\mathbf{M}_I = \mathbf{A}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5.5 & 9 & -4.5 & 1 \\ 9 & -22.5 & 18 & -4.5 \\ -4.5 & 13.5 & -13.5 & 4.5 \end{bmatrix}$$

Note that $\mathbf{M}_I$ does not depend on input data

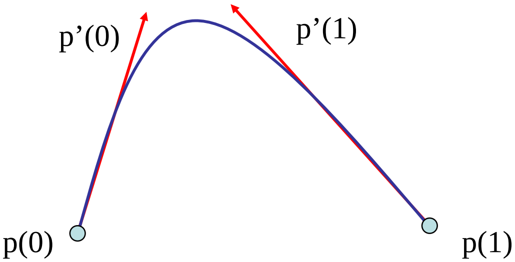
Blending Functions



- The sum of the blending functions are 1
- Blending functions are negative
- Derivatives changes between the interplation points

Hermite Curves

- Use two interpolating conditions and two derivative conditions per segment



Ensures continuity and first derivative continuity between segments

Hermite Equations

- Interpolating conditions are the same at the ends

$$
\begin{aligned}
\mathbf{p}_0 &= \mathbf{p}(0) = \mathbf{c}_0 \\
\mathbf{p}_3 &= \mathbf{p}(1) = \mathbf{c}_0 + \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3
\end{aligned}
$$

- Differentiating we find $\mathbf{p}'(u) = \mathbf{c}_1 + 2u\mathbf{c}_2 + 3u^2\mathbf{c}_3$

$$
\begin{aligned}
\mathbf{p}'(0) &= \mathbf{p}'_0 = \mathbf{c}_1 \\
\mathbf{p}'(1) &= \mathbf{p}'_3 = \mathbf{c}_1 + 2\mathbf{c}_2 + 3\mathbf{c}_3
\end{aligned}
$$

Hermite Matrix

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{p}'_0 \\ \mathbf{p}'_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \mathbf{c}$$

- Solving we find $\mathbf{c} = \mathbf{M}_H\mathbf{p}$ where $\mathbf{M}_H$ is the Hermite Matrix.

$$\mathbf{M}_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \mathbf{p}$$

Hermite Blending Functions

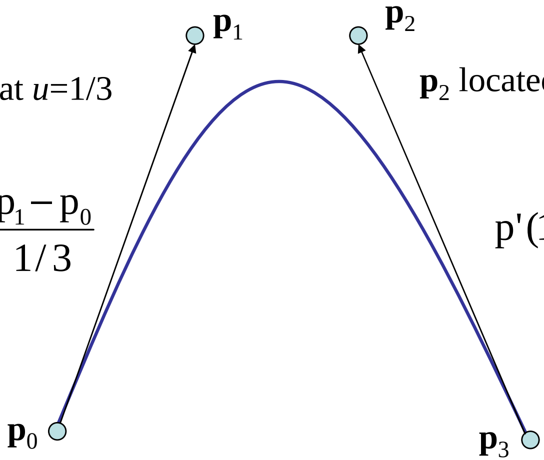
$$\mathbf{b}_H(u) = \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix}$$

- Although these functions are smooth, the Hermite form is not used directly in Computer Graphics and CAD because we usually have control points but not derivatives.

Bézier Curves

- In graphics and CAD, we do not usually have derivative data
- Bézier suggested using the same 4 data points as with the cubic interpolating curve to approximate the derivatives in the Hermite form

Approximating Derivatives



$\mathbf{p}_1$ located at u=1/3

$\mathbf{p}_2$ located at u=2/3

$$p'(0) \approx \frac{p_1 - p_0}{1/3}$$

$$p'(1) \approx \frac{p_3 - p_2}{1/3}$$

Bézier Equations

- Interpolating conditions are the same at the ends

$$
\begin{aligned}
\mathbf{p}_0 &= \mathbf{p}(0) = \mathbf{c}_0 \\
\mathbf{p}_3 &= \mathbf{p}(1) = \mathbf{c}_0 + \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3
\end{aligned}
$$

- Differentiating we find $\mathbf{p}'(u) = \mathbf{c}_1 + 2u\mathbf{c}_2 + 3u^2\mathbf{c}_3$

$$
\begin{aligned}
\mathbf{p}'(0) &= 3(\mathbf{p}_1 - \mathbf{p}_0) = \mathbf{c}_1 \\
\mathbf{p}'(1) &= 3(\mathbf{p}_3 - \mathbf{p}_2) = \mathbf{c}_1 + 2\mathbf{c}_2 + 3\mathbf{c}_3
\end{aligned}
$$

- Solve four linear equations for $\mathbf{c} = \mathbf{M}_B\mathbf{p}$.

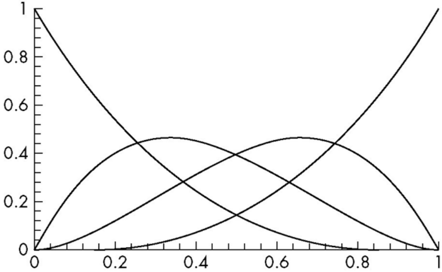
Bézier Matrix

$$\mathbf{M}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

$$\mathbf{p}(u) = \mathbf{u}^T \mathbf{M}_B \mathbf{p} = \mathbf{b}(u)^T \mathbf{p}$$

Bézier Blending Functions

$$\mathbf{b}_H(u) = \begin{bmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 3u^2(1-u) \\ u^3 \end{bmatrix}$$



Bernstein Polynomials

- The blending functions are a special case of the *Bernstein polynomials*

$$\mathbf{b}_{kd}(u) = \frac{d!}{k!(d-k)!} u^k (1-u)^{d-k}$$

- These polynomials give the blending polynomials for any degree Bezier form
  - All zeros at 0 and 1
  - For any degree they all sum to 1
  - They are all between 0 and 1 inside (0,1)

Convex Hull Property

- The properties of the Bernstein polynomials ensure that all Bézier curves lie in the convex hull of their control points

Achieving Continuity

- For Hermite curves, the user specifies the derivatives, so C^1 is achieved simply by sharing points and derivatives across the ”knot”.
- For Bézier curves:
  - They interpolate their endpoints, so C^0 is achieved by sharing control points
  - The parametric derivative is a constant multiple of the vector joining the first/last two control points
  - So C^1 is achieved by setting $\mathbf{p}_3 = \mathbf{q}_0 = \mathbf{r}$, and making $\mathbf{p}_2$, $\mathbf{r}$ and $\mathbf{q}_1$ collinear, with $\mathbf{r} - \mathbf{p}_2 = \mathbf{q}_1 - \mathbf{r}$

Higher Degree Curves

- A single cubic Bézier or Hermite curve can only capture a small class of curves.
- One solution is to raise the degree.
  - Allows more control, at the expense of more control points and higher degree polynomials.
  - Control is not local, one control point influences entire curve
- Alternate, most common solution is to join pieces of cubic curves together into piecewise cubic curves
  - Total curve can be broken into pieces, each of which is cubic.
  - Local control: Each control point only influences a limited part of the curve.
  - Interaction and design is much easier.

Invariance

- Affine invariance means that applying an affine transformation on the control points and then evaluating the curve is the same as evaluating and then transform the curve.
- This property is essential for parametric curves used in graphics.
- Bézier curves, Hermite curves and B-splines are affine invariant.
- Some curves, rational splines (e.g., NURBS), are also perspective invariant
  - Can do perspective transform of control points and then evaluate the curve.

Limitations of Bézier Curves

- Bézier curves are widely used but has one fundamental limitation
  - At the join points we only have C^0 continuity.
  - C^1 continuity is easily archived, C^2 is not.
  - We could use higher order curves or shorten the the i intervall.
- We could use B-splines

Basis for a Knot Sequence

- A parametric curve can be seen as a sum of functions applied to the control points

$$\mathbf{p}(u) = \sum_{i=0}^{m} B_{id}(u)\mathbf{p}_i$$

- Assume the following
  - in the intervall $u = [u_{min}\ u_{max}]$ there are m values, called *knots*, dividing the intervall into $m-1$ pieces.
  - B_{id} is a polynomial of degree d except at the knots and is zero outside the interval $(u_{i_{min}}, u_{i_{max}})$.
  - The function values should always sum to 1.

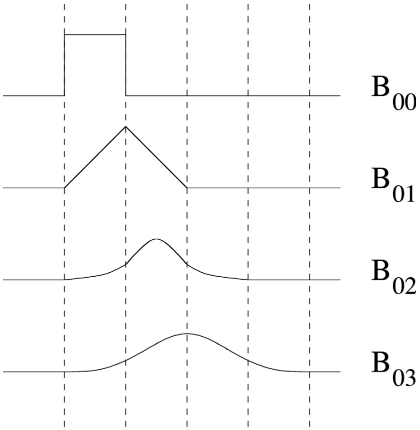
The set $\{B_{id}\}$ forms a *basis* for the given knot sequence and degree.

Cox-deBoor Basis Splines

- The basis polynomials can be defined in several ways.
- Of particular importance is the set of recursive splines defined by *Cox-deBoor*

$$
\begin{aligned}
B_{k0} &= \begin{cases} 1, & u_k \leq u \leq u_{k+1} \\ 0, & \text{otherwise} \end{cases} \\
B_{kd} &= \frac{u-u_k}{u_{k+d}-u_k} B_{k,d-1}(u) + \frac{u_{k+d}-u}{u_{k+d+1}-u_{k+1}} B_{k+1,d-1}(u)
\end{aligned}
$$

Cox-deBoor Polynomials



Cox-deBoor Basis Splines

Each of the

- first set of basis functions B_{k0} is constant over one intervall and then zero.
- second set of basis functions B_{k1} is linear over two intervalls and then zero.
- third set of basis functions B_{k2} is quadratic over tree intervall and then zero.

- At the knots there is C^{d-1} continuity

Cubic Uniform Spline

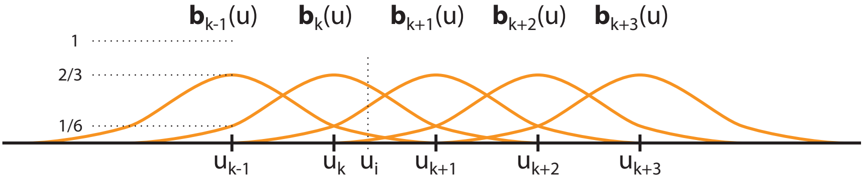
Lets assume

- Cox-deeBoor polynomials of degree 3 (Cubic)
- uniformly spaced knot values $\{0, 1, 2, \ldots, n\}$.

that is a

B-Spline

B-Spline Blending Functions



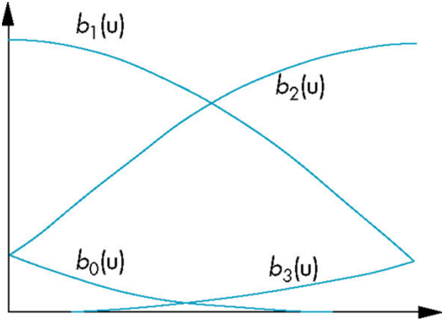
- For a point at $u_k \leq u_i \leq u_{k+1}$

$$\mathbf{p}(u_i) = \mathbf{b}_{k-1}\mathbf{p}_{k-1} + \mathbf{b}_k\mathbf{p}_k + \mathbf{b}_{k+1}\mathbf{p}_{k+1} + \mathbf{b}_{k+2}\mathbf{p}_{k+2}$$

- Given four control points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ the curve is only defined between $\mathbf{p}_1$ and $\mathbf{p}_2$

B-Spline Blending Functions

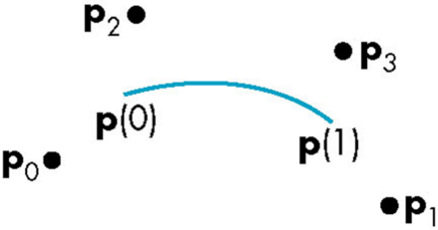
$$\mathbf{b}(u) = \begin{bmatrix} (1-u)^3 \\ 4-6u^2+3u \\ 1+3u+3u^2-3u^2 \\ u^3 \end{bmatrix}$$



B-Spline Matrix

$$\mathbf{p}(u) = \mathbf{u}^T \mathbf{M}_S \mathbf{p} = \mathbf{b}(u)^T \mathbf{p}$$

$$\mathbf{M}_S = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

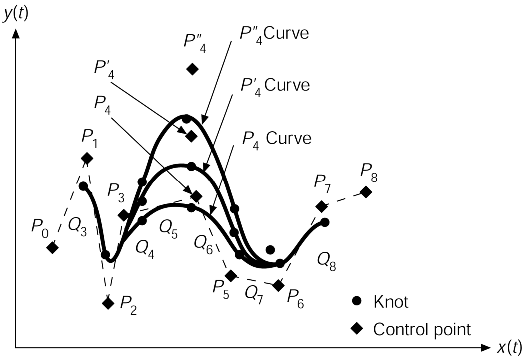


B-spline

- Use the data at $p = [p_{i-2}\ p_{i-1}\ p_i\ p_{i-1}]^T$ to define curve only between p_{i-1} and p_i
- For cubics, we can have continuity of function, first and second derivatives at join points
- Cost is 3 times as much work for curves
  - Add one new point each time rather than three

Local Control

- Polynomial coefficients depend on a few points
- Moving a control point (P_4) affects only local curve
- Based on curve def'n, affected region extends at most 2 knot points away



Nonuniform B-spline

Nonuniform spacing between the knot values

- Can interpolate control points, by repeating knots.
- d repeated knots interpolates a curve of degree $d-1$.

Give Control Points a Weight

Consider $\mathbf{p}_i = [x_i \ y_i \ z_i]$ then

- a weighted representation of that point in homogeneous coordinates would be

$$\mathbf{q}_i = w_i \begin{bmatrix} x_i & y_i & z_i & 1 \end{bmatrix}^T$$

- First three components are

$$\mathbf{q}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \sum_{i=0}^{n} B_{i,d}(u) w_i \mathbf{p}_i$$

  where

$$w(u) = \sum_{i=0}^{n} B_{i,d}(u) w_i$$

- In homogeneous coord. w may not be 1, so we must do a *perspective* division

$$\mathbf{p}(u) = \frac{1}{w(u)} \mathbf{q}(u) = \frac{\sum_{i=0}^{n} B_{i,d}(u) w_i \mathbf{p}_i}{\sum_{i=0}^{n} B_{i,d}(u) w_i}$$

Rational Functions

A rational function can be written as the ratio of two polynomial functions

A rational parameterization in u of a unit circle in xy-plane

$$\begin{aligned} x(u) &= \frac{1-u}{1+u^2} \\ y(u) &= \frac{2u}{1+u^2} \\ z(u) &= 0 \end{aligned}$$

a unit circle in homogeneous coordinates

$$\begin{aligned} x(u) &= 1-u \\ y(u) &= 2u \\ z(u) &= 0 \\ w(u) &= 1+u^2 \end{aligned}$$

Nonuniform Rational B-Splines

NURBS

- Nonuniform B-Splines defined as a rational function are called NURBS.
- Since the perspective division is embedded in the definition, NURBS are *perspective invariant*
- NURBS are important in CAD software.

Parametric Surfaces Patches

- A *surface patch* is part of a surface that is bounded by a closed curve
- We have $\mathbf{p}(u) = \mathbf{u}^T \cdot \mathbf{M} \cdot \mathbf{p}$ from parametric curves.
- Let $\mathbf{p}_i$ vary in 3D along path parameterized on v, then
$$\mathbf{p}(u, v) = \mathbf{u}^T \cdot \mathbf{M} \cdot [\mathbf{p}_0(v)\ \mathbf{p}_1(v)\ \mathbf{p}_2(v)\ \mathbf{p}_3(v)]^T$$
- If $\mathbf{p}_k(v)$ are themselves cubic, the surface is said to be bicubic

Parametric Bicubic Surface Patch

- A parametric bicubic surface is given by

$$\mathbf{p}(u,v) = \mathbf{u}^T \cdot \mathbf{M} \cdot \mathbf{P} \cdot \mathbf{M}^T \cdot \mathbf{v}$$

  where P is a 4×4 array of control points

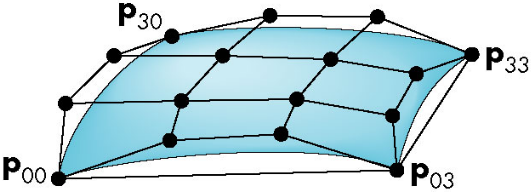
$$\mathbf{P} = [\mathbf{p}_{ij}]$$

- Expressed with blending functions

$$\mathbf{p}(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} b_i(u) b_j(u) \mathbf{p}_{ij}$$

Bézier Surface Patch

$$\mathbf{p}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} b_i(u) b_j(u) \mathbf{p}_{ij} = \mathbf{u}^T \cdot \mathbf{M}_B \cdot \mathbf{P} \cdot \mathbf{M}_B^T \cdot \mathbf{v}$$



Surface Normal

- Needed, e.g., for shading.
- Use the tangent vectors

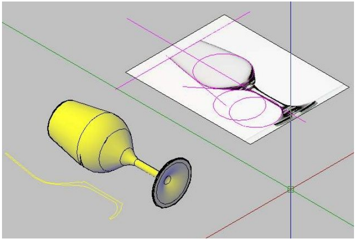
$$\begin{aligned}\frac{\partial}{\partial u} &= [3u^2\ 2u\ 1\ 0]^T \cdot \mathbf{M}_B \cdot \mathbf{P} \cdot \mathbf{M}_B^T \cdot \mathbf{v} \\ \frac{\partial}{\partial v} &= \mathbf{u}^T \cdot \mathbf{M}_B \cdot \mathbf{P} \cdot \mathbf{M}_B^T \cdot [3v^2\ 2v\ 1\ 0]\end{aligned}$$

- Both parallel to the surface at the point (u, v) and their cross product is perpendicular to the surface

$$\mathbf{n} = \frac{\partial}{\partial u}\mathbf{p}(u, v) \times \frac{\partial}{\partial v}\mathbf{p}(u, v)$$

Surface of Revolution

- A very simple way of defining objects is obtained by rotating a curve or a line around an axis.



Sweeps

- Specify a 2D shape and sweep the shape through a region.
- The sweep transformation can contain translation, scaling or rotation.