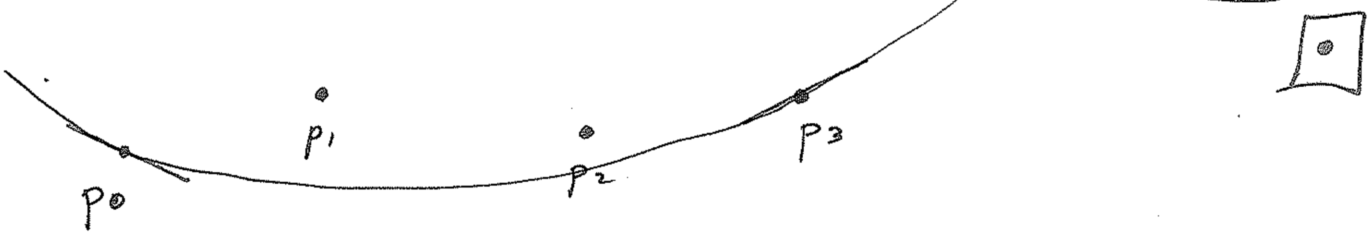
10.5 Hermite Curves & Surfaces

Lecture 11



$p(0) = p_0$

$p(1) = p_3$

$p(u) = c_0 + c_1 u + c_2 u^2 + c_3 u^3$ 4 unknowns

$$\boxed{\begin{aligned} p(0) &= c_0 \\ p(1) &= c_0 + c_1 + c_2 + c_3 \end{aligned}}$$ 2 equations.

Need <u>2 more</u> conditions

Assume we know

$p'(0) \rightsquigarrow p_0'$

$p'(1) \longrightarrow p_3'$

Recall

$p'(u) = c_1 + 2c_2 u + 3c_3 u^2$

$$\boxed{\begin{aligned} p'(0) &= c_1 \\ p'(1) &= c_1 + 2c_2 + 3c_3 \end{aligned}}$$ 2 equations

Our 4×4 system is then

$$\underbrace{\begin{pmatrix} p_0 \\ p_3 \\ p_0' \\ p_3' \end{pmatrix}}_{= q} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}}_{A_H} \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}}_{= c}.$$

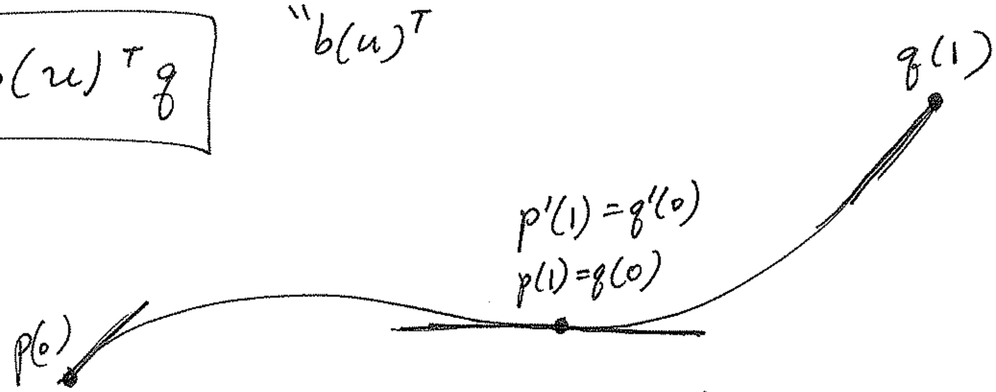
$$c = A_H^{-1} q. \qquad M_H \triangleq A_H^{-1}$$

$$\boxed{c = M_H q}$$ Hermite geometry matrix

$$p(u) = c^T u = q^T M_H^T u$$

$$= u^T c = \underbrace{u^T M_H}_{= b(u)^T} q$$

$$\boxed{p(u) = b(u)^T q}$$



Now have a C^1 curve

Hermite blending functions

$$b(u) = M_H^T u$$

$$\boxed{b(u) = \begin{pmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{pmatrix}}$$

These give less oscillatory results than the interpolating polynomial blending functions.

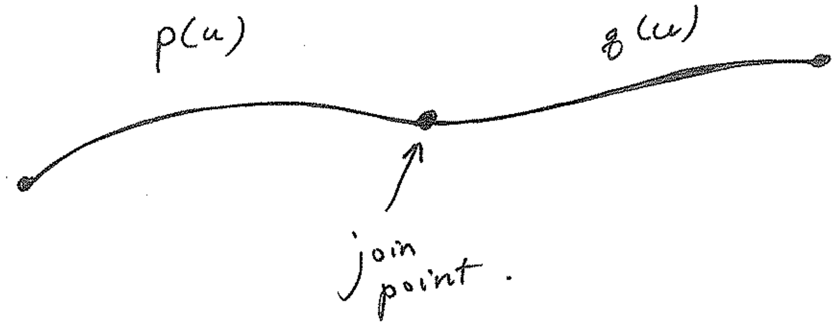
Bicubic Hermite Surface Patch.

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

$Q = [q_{ij}]$ is the surface data.

- However, not yet clear what the elements of Q are
- also user won't usually give derivative data, only point data.

$\rightsquigarrow$ Bezier Forms.



<u>Continuity</u> (C^0 parametric continuity)

$$p(\ ) = q(0)$$

<u>C^1 parametric continuity</u>

$$p'(1) = q'(0)$$

<u>G^1 geometric continuity</u>

a slightly weaker condition:

$$\underset{\sim}{p}'(1) = \alpha\, \underset{\sim}{q}'(0)$$

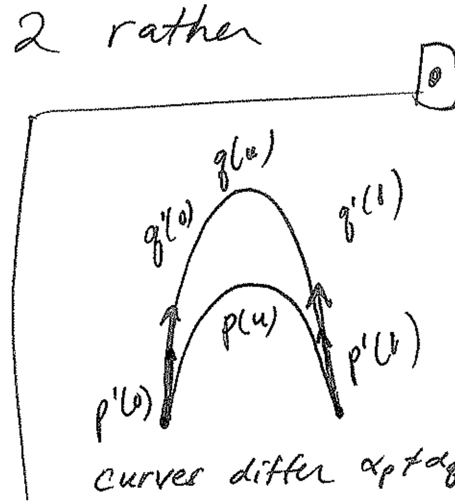
$\alpha > 0$

derivatives should be proportional.

tangent vectors still point in the same direction

this frees up one condition (2 rather than 3).

C^n and G^n <u>continuity</u>



10.6. Bezier ~~Curves~~ + Surfaces
(Bezier 1970 - Renault engineer)

Idea: use the same control point data to approximate the derivatives at the endpoints

+ good approximations to Hermite curves
+ comparable to the Interpolating curves because they use the same data.
+ do not need derivative information

$$\boxed{1} \quad \begin{cases} p_0 = p(0) & = c_0 \\ p_3 = p(1) & = c_0 + c_1 + c_2 + c_3 \end{cases}$$

Bezier: use the other 2 control pts to approximate tangents. at $u = 0, 1$

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

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

$$\boxed{2}\quad \begin{cases} p'(0) = c_1 = 3(p_1 - p_0) \\ p'(1) = c_1 + 2c_2 + 3c_3 = 3(p_3 - p_2) \end{cases}$$

4 equations & 4 unknowns (3 such sets)
(one for each x, y, z)

$$c = M_B\, p$$

$$\begin{pmatrix} p(0) \\ p(1) \\ p'(0) \\ p'(1) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}}_{A} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

or

$$\begin{pmatrix} p_0 \\ p_3 \\ 3p_1 - 3p_0 \\ 3p_3 - 3p_2 \end{pmatrix} = \begin{pmatrix} \quad " \quad \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

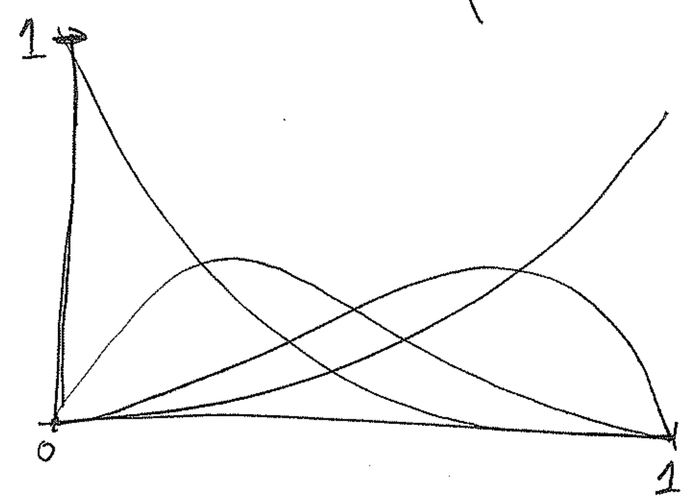
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & -3 & 3 & 0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

$$M_B = A^{-1} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & -3 & 3 & 0 \end{pmatrix}$$

$$p(u) = u^T c$$
$$= u^T (M_B p)$$
$$= (u^T M_B)\, p$$
$$= b(u)^T p$$

$$b(u) = M_B^T u = \begin{pmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 3u^2(1-u) \\ u^3 \end{pmatrix}$$

Bezier blending functions.



Blending polynomials for the Bézier cubic.

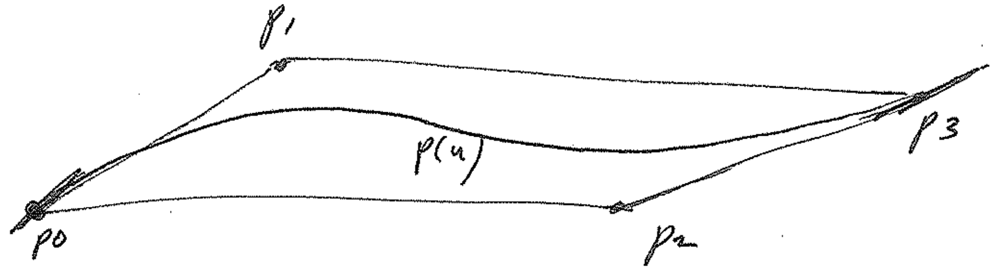
Bernstein polynomials

binomial coefficients.

$$b_{kd}(u) = \frac{d!}{k!(d-k)!} u^k (1-u)^{d-k} = \binom{d}{k} u^k (1-u)^{d-k}.$$

+ all zeros are at 0 or 1.
  - not very oscillatory.
+ $b_i(u) < 1$ for $0<u<1$
+ $\sum_{i=0}^{d} b_{id}(u) = 1$

$p(u) = \sum b_i(u)\, p_i$ convex sum (convex combination)



examples.

Since

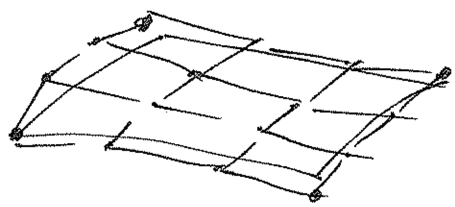
$$p(u) = \sum b_i(u)\, p_i$$

and $\sum_{i=0}^{3} b_{id}(u) = 1$

the Bezier polynomial lies in the <u>convex hull</u> of p_0, p_1, p_2, p_3.

* so although Bez. poly. doesn't interpolate all control points, it is not far from them.

10.6.2 Bezier Surface Patches.



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

\+ patch is contained in the convex hull of the control points.

\+ interpolates $p_{00}, p_{03}, p_{30}, p_{33}$

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

$$= (M_B^T u)^T P (M_B^T v)$$

$$= u^T M_B P M_B^T v$$

$\boxed{1}$ $p(0,0) = p_{00}$

$\boxed{2}$ $\frac{\partial p}{\partial u}(0,0) = 3(p_{10} - p_{00})$

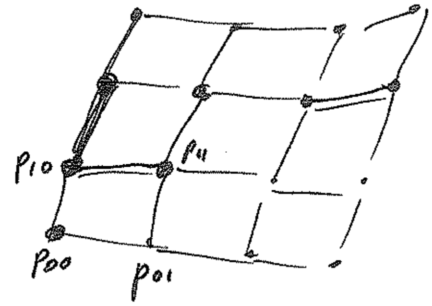
$\boxed{3}$ $\frac{\partial p}{\partial v}(0,0) = 3(p_{01} - p_{00})$

$\boxed{4}$ $\frac{\partial^2 p}{\partial u \partial v}(0,0) = 9(p_{00} - p_{01} \mp p_{10} + p_{11})$

$$= \left(\frac{1}{3}\right)^{-1}\left(\frac{p_{11} - p_{10}}{1/3} - \frac{p_{01} - p_{00}}{1/3}\right)$$

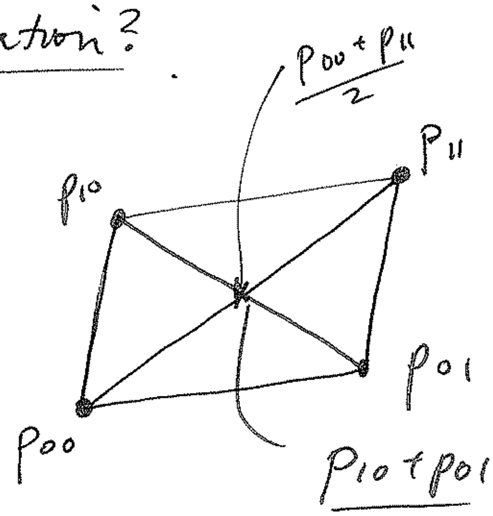
$$= 9(p_{00} - p_{01} - p_{10} + p_{11})$$

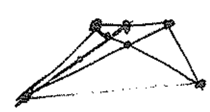
tendency of patch to divert from being flat, to $\underline{\text{twist}}$ at the corner



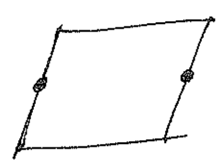
- 4 interpolation conditions at corners. $\boxed{1}$
- 4 $\frac{\partial}{\partial u}$ conditions. $\boxed{2}$
- 4 $\frac{\partial}{\partial v}$ conditions. $\boxed{3}$
- 4 more conditions of form $\boxed{4}$

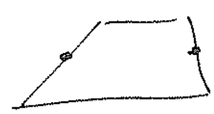
Interpretation?





$\boxed{4}$ will be zero if surface is locally flat there.

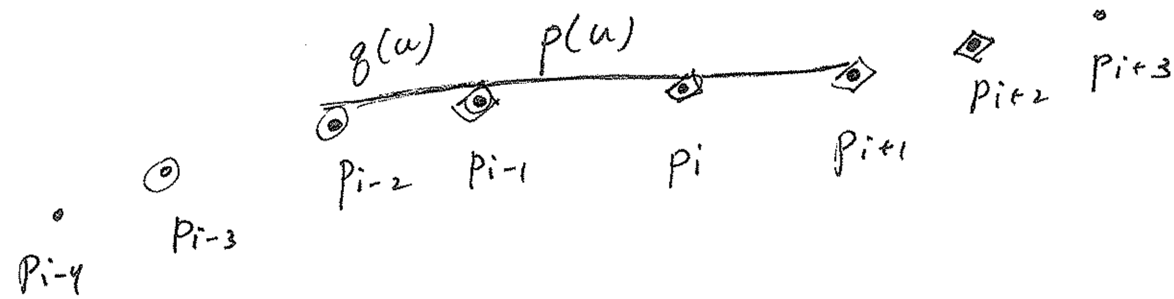




p.7.1 B-Spline Curve

Motivation: Bezier curves are only C^0. Relax interpolation conditions for smoothness

- C^2 continuity @ join points
- span only distance between middle 2 control points



Find M s.t.

$$p(u) = \underset{\sim}{u}^T M \underset{\sim}{p} \qquad p = \begin{pmatrix} p_{i-2} \\ p_{i-1} \\ p_i \\ p_{i+1} \end{pmatrix}$$

$$q(u) = \underset{\sim}{u}^T M \underset{\sim}{q} \qquad q = \begin{pmatrix} p_{i-3} \\ p_{i-2} \\ p_{i-1} \\ p_i \end{pmatrix}$$

Principle: conditions at join points on

symmetry: $p(0)$, $p'(0)$ must be same as those on $q(1)$, $q'(1)$

& hence can not depend on p_{i+1} or p_{i-3}

Example:

4 equations

$$\begin{cases} p(0) = \frac{1}{6}\left(p_{i-2} + 4p_{i-1} + p_i\right) = c_0 \\ p'(0) = \frac{1}{2}\left(p_i - p_{i-2}\right) = c_1 \end{cases}$$

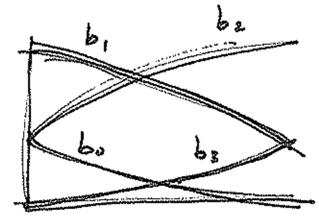
$$\begin{cases} p(1) = \frac{1}{6}\left(p_{i-1} + 4p_i + p_{i+1}\right) = c_0 + c_1 + c_2 + c_3 \\ p'(1) = \frac{1}{2}\left(p_{i+1} - p_{i-1}\right) = c_1 + 2c_2 + 3c_3 \end{cases}$$

$$\underbrace{\begin{pmatrix} \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}}_{A} \begin{pmatrix} p_{i-2} \\ p_{i-1} \\ p_i \\ p_{i+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix}}_{B} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

$$M_s = B^{-1}A = \frac{1}{6}\begin{pmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix}$$

B-spline geometry matrix

Blending Polynomials



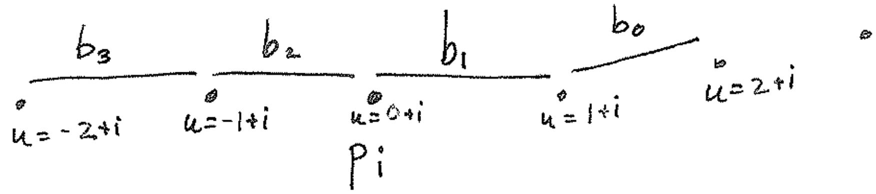
$$\underset{\sim}{b}(u) = M_s^T \underset{\sim}{u} = \frac{1}{6}\begin{pmatrix} (1-u)^3 \\ 4-6u^2+3u^3 \\ 1+3u+3u^2-3u^3 \\ u^3 \end{pmatrix}$$

properties:

- $\sum_i b_i(u) = 1$,
- $0 < b_i(u) < 1$ for $u \in (0,1)$
- curve lies in convex hull of control points
- C^1 by construction, & furthermore, C^2
  $\rightsquigarrow$ visually smooth appearance.
- 3x work as for Bezier (hop 3 pts for next curve, vs. 1 pt. for the B-spline).

10.7.2. B-Splines and Basis

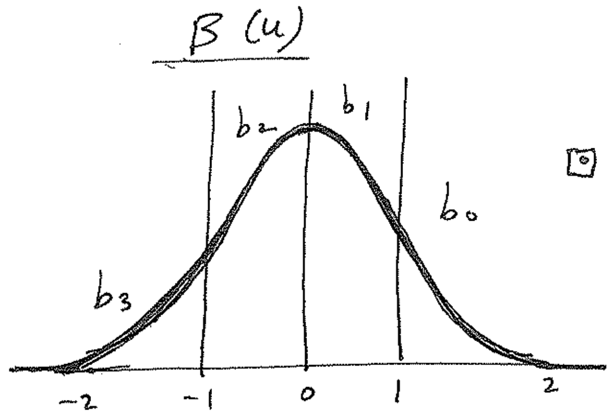
Consider the influence of the control point p_i:



The segments drawn will depend on p_i. For each segment p_i is multiplied by a different blending polynomial.

Let us take u to have values indicated above. Then we can summarize the influence of p_i through $B(u-i)\,p_i$, where

$$B(u) = \begin{cases} 0 & u < -2 \\ b_3(u+2) & -2 \le u < -1 \\ b_2(u+1) & -1 \le u < 0 \\ b_1(u) & 0 \le u < 1 \\ b_0(u-1) & 1 \le u < 2 \\ 0 & 2 \le u \end{cases}$$



Given control points $p_0, \ldots, p_m$, the spline can then be written,

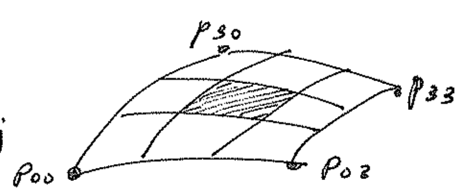
$$p(u) = \sum_{i=0}^{m} B\underbrace{(u-i)}\, p_i \qquad 0 \le u \le m$$

(shift B to be centered over the i^{th} control point)

The set $\{B(u-i)\}$ forms a basis for the cubic B-spline curves

10.7.3 Spline Surfaces

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



10.8 general B-Splines

knots

$$u_{min} = u_0 \le u_1 \le \cdots \le u_n = u_{max}$$

$$p(u) = \underbrace{\sum_{i=0}^{m} B_{id}(u)\, p_i}_{\text{convex combination of control points}}$$

$B_{id}(u)$ piecewise degree d polynomial.
0 outside (u_{imin}, u_{imax})

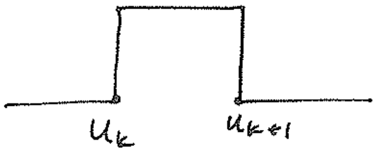
$\{B_{id}\}$ basis splines.

Cox-de Boor recursion:

$$B_{k0}(u) = \begin{cases} 1 & u_k \le u \le u_{k+1} \\ 0 & \text{otherwise.} \end{cases}$$

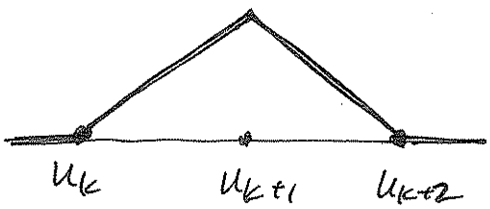
$$B_{kd}(u) = \frac{u - u_k}{u_{k+d} - u_k} B_{k,d-1}(u) + \frac{u_{k+d+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}(u)$$

$$B_{k0}(u) = \begin{cases} 1 & u \in [u_k, u_{k+1}] \\ 0 & \text{otherwise.} \end{cases} = \chi_{[u_k, u_{k+1}]} = \chi_{I_k}$$

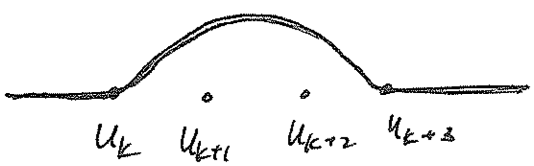


$$B_{k1}(u) = \frac{u-u_k}{u_{k+1}-u_k} B_{k0}(u) + \frac{u_{k+2}-u}{u_{k+2}-u_{k+1}} B_{k+1,0}(u)$$

$$= \frac{u-u_k}{u_{k+1}-u_k} \chi_{I_k} + \frac{u_{k+2}-u}{u_{k+2}-u_{k+1}} \chi_{I_{k+1}}$$



$B_{k2}(u)$



- each B_{id} is non-zero in $d+1$ intervals.

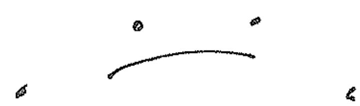
Set of spline basis functions is defined by the desired degree and the knot array

10.8.2 Uniform Splines.

Assume a <u>uniform</u> knot sequence:

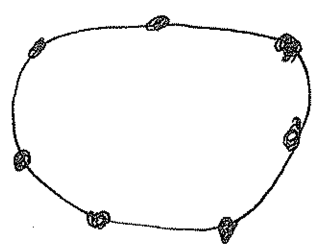
e.g. $\{0, 1, 2, \ldots, n\}$

e.g. cubic B-spline defined earlier



doesn't span data

In case of periodic (closed) curve,



Note: repeated knots allowed with convention $\frac{0}{0} = 1$

10.8.3 Nonuniform B-Splines.

Repeating knots pulls the spline closer to that control pt.

Open splines: uniform w/ multiplicity $d+1$ at ends to force interpolation at ends.

e.g. cubic B-splines:

$$\{0,0,0,0,1,2,\ldots,n-1,n,n,n,n\}$$

cubic B-spline $\rightsquigarrow$ cubic Bezier curve

$$\{0,0,0,0,1,1,1,1\}$$

10.8.4 NURBS.

Nonuniform rational B-spline

$$q_i = w_i \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

homogeneous coordinate representation of the point

Use w_i to control importance or weight of the control point q_i.

$$p(u) = \frac{\sum_{i=0}^{n} B_{i,d}(u) w_i p_i}{\sum_{i=0}^{n} B_{id}(u) w_i}$$

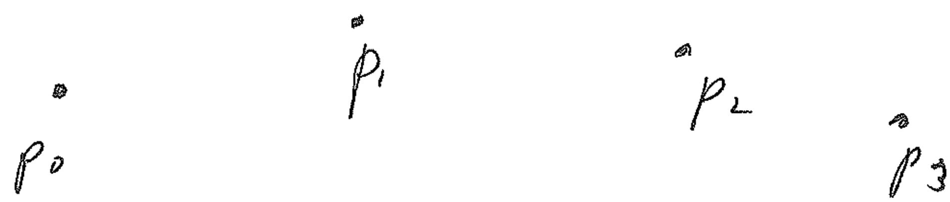
NURBs

properties:

- convex hull
- continuity
- handled correctly in perspective view unlike regular spline.
- quadratic NURBs can be used to represent quadrics.

10.8.5 Catmull-Rom Splines.

- Popular.
- relax requirement that curves lie in convex hull of data.



Conditions:

$$p(0) = p_1$$

$$p(1) = p_2$$

$$p'(0) = \frac{p_2 - p_0}{2}$$

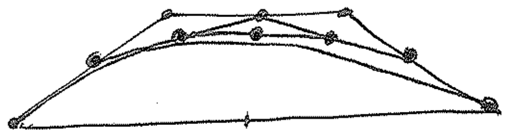
$$p'(1) = \frac{p_3 - p_1}{2}$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

$$\Rightarrow \quad p(u) = \underset{\sim}{u}^T M_R \, p$$

$$M_R = \frac{1}{2} \begin{pmatrix} -1 & 3 & -3 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$$

Foley et al 11.2.7 Subdividing Curves.

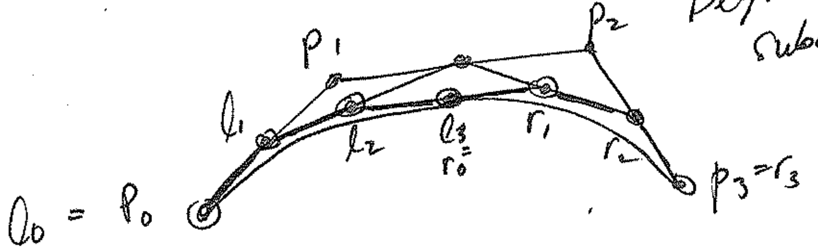


① increase polynomial degree

② increase curve segments

gain degrees of freedom:

Bezier Curve subdivision



The new curve exactly matches the old curve until a control point is moved.

<u>properties</u> :

- each new control point is convex comb. of old.
- new control points closer to the curve than old control points. "variation diminishing property"

This subdivision process could be described by the matrices

$$G_B^l = \frac{1}{8}\begin{pmatrix} 8 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 \\ 2 & 4 & 2 & 0 \\ 1 & 3 & 3 & 1 \end{pmatrix}\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

$$G_B^r = \frac{1}{8}\begin{pmatrix} 1 & 3 & 3 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 8 \end{pmatrix}\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

~~Similar ways to subdivide uniform B-spline curves~~

~~G_S^l, G_S^r~~

~~-disconnected spline~~

AE 10.9.3

convert to Bezier, & subdivide Bezier.

① $p(u) = u^T M_B P$

② $p(u) = u^T M q$

To get the same curve as ① but in representation ②,
choose $q = M^{-1} M_B P$