

---

# INTRODUCTION TO POLYNOMIAL CURVES PART I

Peter Schröder

---

CS175 2000

1

## CURVES AND SURFACE

---

### Representations

- parameterized
- implicit
- piecewise linear
  - lines and polygons
- higher order (not good... Why?)
  - typically polynomials

---

CS175 2000

2

# PARAMETRIC DESC.

---

## Curves (piecewise)

- linear
  - quadratic? (why is this not enough?)
- cubic
  - Hermite, Bezier, B-Spline

## Surfaces (patches)

- tensor products (typically...)

---

CS175 2000

3

# PARAMETRIC CURVES

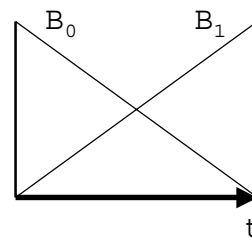
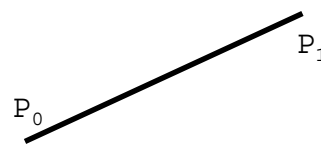
---

## Linear interpolation

$$C(t) = P_0 + t(P_1 - P_0)$$

$$C(t) = \begin{pmatrix} C_x(t) \\ C_y(t) \\ C_z(t) \end{pmatrix} = (P_0 \ P_1) \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}$$

$$C(t) = G^T B T(t) \quad C'(t) = G^T B T'(t)$$



---

CS175 2000

4

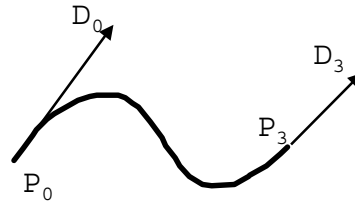
# HERMITE CURVES

---

Specify position and derivative at start and end point

■ 4DOF, need cubic polynomial

$$G^T = (P_0 \ P_3 \ D_0 \ D_3)$$
$$T^T(t) = (t^3 \ t^2 \ t \ 1)$$
$$T'^T(t) = (3t^2 \ 2t \ 1 \ 0)$$



---

CS175 2000

5

# HERMITE CURVES

---

Substitute constraints to find basis

$$G^T B \begin{pmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} = G^T \quad B = \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}$$

$$C(t) = B_0(t)P_0 + B_1(t)P_3 + B_2(t)D_0 + B_3(t)D_3$$

■ how to transform?

---

CS175 2000

6

## MULTIPLE SEGMENTS

---

What are the basis functions?

- why is it useful to think in these terms?
- what do they look like?
- does the number of DOFs seem to be right?

Need something better!

---

CS175 2000

7

## BEZIER CURVES

---

Bernstein blending polynomials

$$B_i^k(t) = \binom{k}{i} t^i (1-t)^{k-i}$$

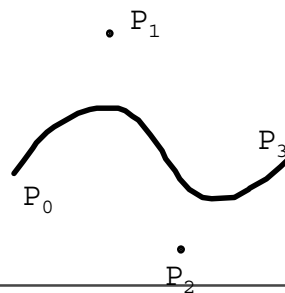
- cubic case

$$B_0^3(t) = (1-t)^3$$

$$B_1^3(t) = 3t(1-t)^2$$

$$B_2^3(t) = 3t^2(1-t)$$

$$B_3^3(t) = t^3$$



---

CS175 2000

8

# BEZIER CURVES

---

## Definition

$$C(t) = P_0B_0^3(t) + P_1B_1^3(t) + P_2B_2^3(t) + P_3B_3^3(t)$$

$$C'(0) = 3(P_1 - P_0)$$

$$C'(1) = 3(P_3 - P_2)$$

## Fun facts

- partition of unity

$$\sum_{i=0}^k B_i^k(t) = 1$$

- affine combinations...

- why do they matter?

---

CS175 2000

9

# BEZIER CURVES

---

## Properties

- curve is affine combination of control points
- endpoint interpolating
- convex hull property
- shape is invariant under affine transforms

---

CS175 2000

10

## CHANGE OF BASIS

---

### Example

- Hermite to Bezier

$$B(t) = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix} \quad H(t) = \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}$$

$$(P_0 \ P_1 \ P_2 \ P_3) = (P_0 \ P_3 \ D_0 \ D_3)HB^{-1}$$

---

CS175 2000

11

## EVALUATION

---

### Direct

$$C(t_0) = G^T B \begin{pmatrix} t_0^3 \\ t_0^2 \\ t_0 \\ 1 \end{pmatrix}$$

- not very efficient
- Horner's rule
- can be unstable for high orders

---

CS175 2000

12

# FORWARD DIFFERENCING

---

## Step along curve

$$\begin{aligned}C(t+\delta) &= C(t) + \Delta C(t) & C_{n+1} &= C_n + \Delta C_n \\ \Delta C(t+\delta) &= \Delta C(t) + \Delta^2 C(t) & \Delta C_{n+1} &= \Delta C_n + \Delta^2 C_n \\ \Delta^2 C(t+\delta) &= \Delta^2 C(t) + \Delta^3 C(t) & \Delta^2 C_{n+1} &= \Delta^2 C_n + \Delta^3 C_n\end{aligned}$$

■ choose step size

■ how many flops?

■ accumulation of error

$$D = \begin{pmatrix} 0 & 0 & 0 & 1 \\ \delta^3 & \delta^2 & \delta & 0 \\ 6\delta^3 & 2\delta^2 & 0 & 0 \\ 6\delta^3 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

---

CS175 2000

13

# BERNSTEIN POLYNOMIALS

---

## Evaluation

■ recurrence

$$\begin{aligned}B_i^k(t) &= \binom{k}{i} t^i (1-t)^{k-i} \\ &= t \binom{k-1}{i-1} t^{i-1} (1-t)^{k-1-(i-1)} + (1-t) \binom{k-1}{i} t^i (1-t)^{k-1-i} \\ &= t B_{i-1}^{k-1}(t) + (1-t) B_i^{k-1}(t)\end{aligned}$$

■ affine combination!

---

CS175 2000

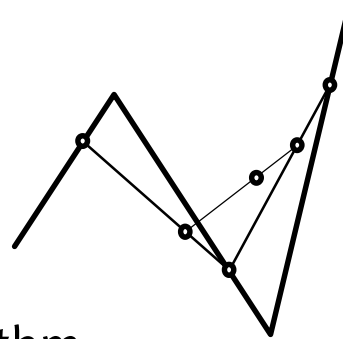
14

# BERNSTEIN POLYNOMIALS

---

## Evaluation

$$b_i^0(t) = P_i = 0 \dots 3$$
$$b_i^l(t) = (1-t)b_i^{l-1}(t) + tb_{i+1}^{l-1}(t)$$



## de Casteljau Algorithm

CS175 2000

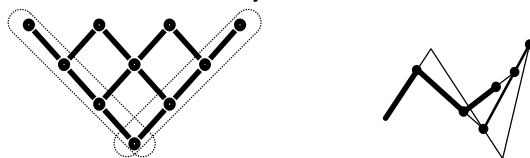
15

# DE CASTELJAU ALG.

---

## Properties

- very stable, uses only affine combinations
- use to split curve into pieces
- recursive, adaptive subdivision



CS175 2000

16

# AFFINE SPACE

---

## Affine Combination

$$u_k \in \mathbb{R}^n \quad \sum \alpha_k u_k \quad \sum \alpha_k = 1$$

## Affine Map

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m \quad f\left(\sum \alpha_k u_k\right) = \sum \alpha_k f(u_k)$$

- exactly the linear maps and translation

---

CS175 2000

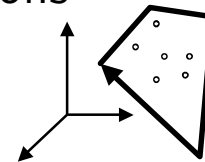
17

# AFFINE SPACE

---

## Definition

- subset of vector space closed under affine combinations



- exactly the linear subspaces plus translation

---

CS175 2000

18

# AFFINE FORMS

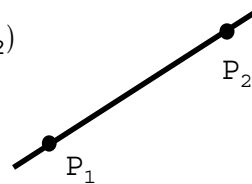
---

## Evaluation

$$f(P) = f(tP_1 + (1-t)P_2) = tf(P_1) + (1-t)f(P_2)$$

## Differentiation

$$\begin{aligned} Df|_P &= \lim_{\tau \rightarrow 0} \frac{f(P + \tau) - f(P)}{\tau} = \\ &= f(P+1) - f(P) \end{aligned}$$



$$\begin{aligned} f(t) &= at + b \\ Df(t) &= f(1) - f(0) = a \end{aligned}$$

# POLAR FORMS

---

## Blossom

- For every polynomial  $F(u)$  of degree  $n$  there exists a unique symmetric multi-affine map  $f(u_1, \dots, u_n)$  for which  $F(u) = f(u, \dots, u)$ . This map is called the polar form.  $F(u)$  is called its diagonal and  $f$  is called the blossom of  $F$

## EXAMPLE

---

$$F(u) = a_0 + a_1u + a_2u^2 + a_3u^3$$

$$\begin{aligned} f(u_1, u_2, u_3) &= a_0 \\ &+ \frac{a_1}{3}(u_1 + u_2 + u_3) \\ &+ \frac{a_3}{3}(u_1u_2 + u_2u_3 + u_3u_1) \end{aligned}$$

---

CS175 2000

21

## POLAR FORM

---

$$F(u) = \sum_{i=0}^n a_i u^i \quad f(u_1, \dots, u_n) = \sum_{i=0}^n a_i \binom{n}{i}^{-1} \sum_{\substack{S \subseteq \{1, \dots, n\} \\ |S|=i}} \prod_{j \in S} u_j$$

### Differentiation

$$a_q = \frac{F^{(q)}(0)}{q!} = \binom{n}{q} f\left(\underbrace{0, \dots, 0}_{n-q}, \underbrace{\hat{1}, \dots, \hat{1}}_q\right)$$

$$F^{(q)}(u) = \frac{n!}{(n-q)!} f(u, \dots, u, \hat{1}, \dots, \hat{1})$$

---

CS175 2000

22

# CONTINUITY

---

## Two Polynomials

- Let  $F$  and  $G$  be two polynomials of degree  $n$  and let  $u$  be a point on the real line. Then  $F$  and  $G$  are  $C^q$  continuous at  $u$  iff

$$f(u^{n-q}, u_1, \dots, u_q) = g(u^{n-q}, u_1, \dots, u_q) \\ u_1, \dots, u_q \in \mathbb{R}$$

---

CS175 2000

23

# BEZIER POINTS

---

## Finding Control Points

- The Bezier control points of  $F$  w.r.t  $b_i = \binom{n}{i} s^{n-i} t^i = 0, \dots, n$   $u = \frac{t-u}{t-s}s + \frac{u-s}{t-s}t$

Proof

$$f(u^n) = \left(\frac{t-u}{t-s}\right) f(su^{n-1}) + \left(\frac{u-s}{t-s}\right) f(u^{n-1}t) \\ = \dots = \sum_{i=0}^n B_i^{\Delta n}(u) f(s^{n-i}t^i)$$

---

CS175 2000

24

## DE CASTELJAU ALGORITHM

---

### Evaluate Bezier form

$$b_i^0(u) = b_i, i = 0, \dots, n$$

$$b_i^l(u) = \left(\frac{t-u}{t-s}\right) b_i^{l-1}(u) + \left(\frac{u-s}{t-s}\right) b_{i+1}^{l-1}(u)$$

$$l = 1, \dots, n, i = 0, \dots, n-l$$

- Very stable, uses convex combinations only

---

CS175 2000

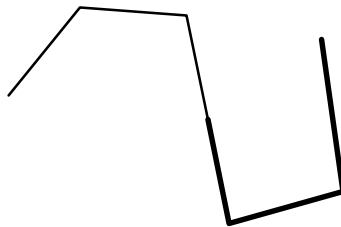
25

## MULTIPLE SEGMENTS

---

How to stick them together?

- can match first derivative easily
- second derivative?



---

CS175 2000

26

## MODELING

---

### Bezier curves

- requires to keep track of continuity conditions explicitly... Possible, but very messy
- better: use basis which has continuity conditions built in!

### B-Splines

---

CS175 2000

27

## B - SPLINES

---

### Definition

- knotvector

$$T = (t_i)_{i \in \mathbb{Z}} \quad t_i < t_{i+n+1}$$

- non-decreasing sequence, can have multiple entries (multiplicity)

- curve:  $C(u) = \sum N_i(u) d_i$

---

CS175 2000

28

## B - SPLINES

---

### Definition

- recursive

$$N_i^0(u) = \begin{cases} 1 & t_i \leq u < t_{i+1} \\ 0 & \text{else} \end{cases}$$

$$N_i^l(u) = \frac{u - t_i}{t_{i+l} - t_i} N_i^{l-1}(u) + \frac{t_{i+l+1} - u}{t_{i+l+1} - t_{i+1}} N_{i+1}^{l-1}(u)$$

- support  $N_i^l(u) = 0$   $u \notin [t_i, t_{i+l+1}]$

CS175 2000

29

## B - SPLINES

---

### DeBoor Algorithm $t_0 \leq \dots \leq r_1 < s_1 \leq \dots \leq s_n$

$$d_j^l(u) = f(r_1, \dots, r_{n-l-j}, u, \dots, u, s_1, \dots, s_j)$$

$$= \left( \frac{s_{j+1} - u}{s_{j+1} - r_{n-l-j+1}} \right) f(r_1, \dots, r_{n-l-j+1}, u, \dots, u, s_1, \dots, s_j)$$

$$+ \left( \frac{u - r_{n-l-j+1}}{s_{j+1} - r_{n-l-j+1}} \right) f(r_1, \dots, r_{n-l-j}, u, \dots, u, s_j, \dots, s_{j+1})$$

$$= \left( \frac{s_{j+1} - u}{s_{j+1} - r_{n-l-j+1}} \right) d_j^{l-1}(u) + \left( \frac{u - r_{n-l-j+1}}{s_{j+1} - r_{n-l-j+1}} \right) d_{j+1}^{l-1}(u)$$

CS175 2000

30

## DEBOOR ALGORITHM

---

### Recursion

$$N_i^0(u) = \begin{cases} 1 & t_i \leq u < t_{i+1} \\ 0 & \text{else} \end{cases}$$

$$N_i^l(u) = \frac{u - t_i}{t_{i+1} - t_i} N_i^{l-1}(u) + \frac{t_{i+2} - u}{t_{i+2} - t_{i+1}} N_{i+1}^{l-1}(u)$$

$$d_i^0(u) = d_i = j - n \dots j$$

$$d_i^l(u) = \frac{t_{i+n+1} - u}{t_{i+n+1} - t_{i+1}} d_i^{l-1}(u) + \frac{u - t_{i+1}}{t_{i+n+1} - t_{i+1}} d_{i+1}^{l-1}(u)$$

$$l = 1 \dots n_i = j - n, \dots, j - l$$

---

CS175 2000

31

## B - SPLINES

---

### Terms

- degree: highest exponent in polynomial
- order: 1+degree
- multiplicity of a knot: how many times repeated
- continuity:  $C^{n-m}$  with multiplicity  $m$

---

CS175 2000

32

## B - SPLINES

---

### Knot insertion

- create new degree of freedom along curve

$$d_i^* = \alpha_i d_i + (1 - \alpha_i) d_{i-1}$$

$$\alpha_i = \begin{cases} 1 & i \leq j - n \\ \frac{t - t_i}{t_{i+n} - t_i} & j - n + 1 \leq i \leq j \\ 0 & j + 1 \leq i \end{cases}$$

---

CS175 2000

33

## SURFACES

---

### Tensor product case

- nothing special happens

$$F(u, v) = f(u^n; v^n)$$

- treat each parameter direction separately

---

CS175 2000

34

# SURFACES

---

## Non-tensor product

$$F(u,v) = \sum_{i+j+k=n} p_{ijk} \binom{n}{i!j!k!} u^i v^j (1-u-v)^k$$

- example: quadratic in two variables  
 $F(u,v) = a_{00} + a_{10}u + a_{01}v + a_{20}u^2 + a_{11}uv + a_{02}v^2$

- Bezier functions for triangles (deCasteljau, etc.)

---

CS175 2000 ■ B-patches? very hard.

35

# SURFACES

---

## Tensor product

- 2 parameter directions which decouple  
 $F(u,v) = F_u(u)F_v(v)$

- Example: B<sup>3</sup><sub>u</sub>(u)B<sup>3</sup><sub>v</sub>(v) patches

$$= (a_0 + a_1u + a_2u^2 + a_3u^3)(b_0 + b_1v + b_2v^2 + b_3v^3)$$

---

CS175 2000

36

# TENSOR PRODUCT SURFACES

## Polarform

- reduces to 1D setting

$$F(u,v) = F_u(u)F_v(v) = f_u(u)u f_v(v)v = f(uu,vv)$$

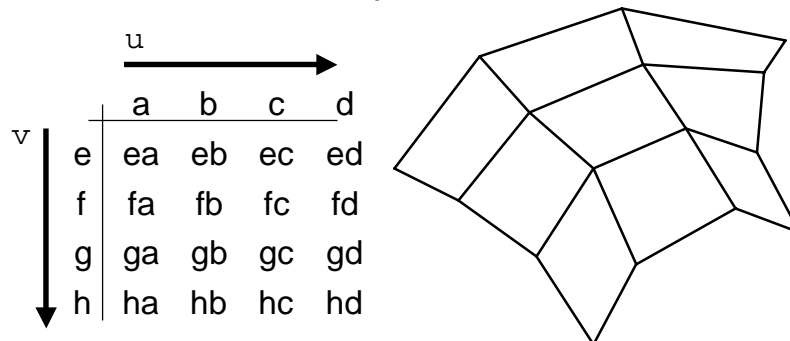
$$F(u,v) = \sum_{i=0}^n \sum_{j=0}^m f(u_1^{n-i}, u_2^i; v_1^{m-j}, v_2^j) B_i^n(u) B_j^m(v)$$

- control points of tensor product surface are outerproduct of control points for 1D components

# TENSOR PRODUCT SURFACES

## Control points

$$F(u,v) = f_u(u)u f_v(v)v$$



# SURFACES OF REVOLUTION

---

## Profiles and axes

- modulate diameter of circle

$$S(u) = \begin{pmatrix} B_x^s(u)/B_w^s(u) \\ B_y^s(u)/B_w^s(u) \end{pmatrix}$$

$$T(u,v) = \begin{pmatrix} R(v)S(u) \\ A(v) \end{pmatrix} = \begin{pmatrix} B_x^r(v)/B_w^r(v)S(u) \\ B_z^r(v)/B_w^r(v) \end{pmatrix}$$

