Introduction to Polynomial Curves Part I

Peter Schröder

Curves and Surface

Representations

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

Parametric Desc.

Curves (piecewise)

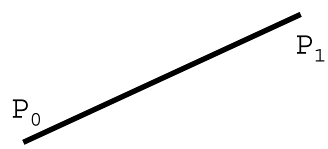
- linear
  - quadratic? (why is this not enough?)
- cubic
  - Hermite, Bezier, B-Spline

Surfaces (patches)

- tensor products (typically...)

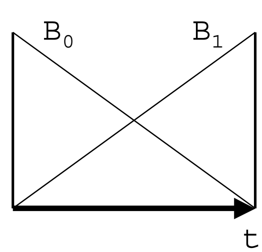
Parametric Curves

Linear interpolation



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

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



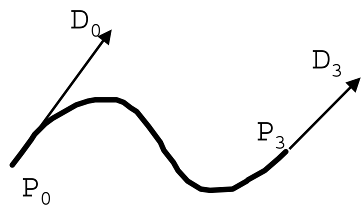
$$C(t) = G^T B T(t) \qquad C'(t) = G^T B T'(t)$$

Hermite Curves

Specify position and derivative at start and end point

- 4DOF, need cubic polynomial

$$G^T = \begin{pmatrix} P_0 & P_3 & D_0 & D_3 \end{pmatrix}$$
$$T^T(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix}$$
$$T'^T(t) = \begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix}$$



Hermite Curves

Substitute constraints to find basis

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

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

- how to transform?

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!

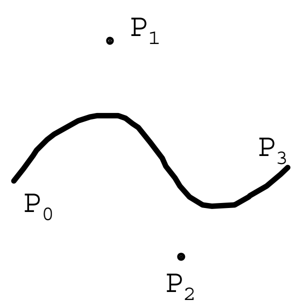
Bezier Curves

Bernstein blending polynomials

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

- cubic case

$$\begin{aligned} B_0^3(t) &= (1-t)^3 \\ B_1^3(t) &= 3t(1-t)^2 \\ B_2^3(t) &= 3t^2(1-t) \\ B_3^3(t) &= t^3 \end{aligned}$$



Bezier Curves

Definition

$$\begin{aligned} c(t) &= p_0 B_0^3(t) + p_1 B_1^3(t) + p_2 B_2^3(t) + p_3 B_3^3(t) \\ c'(0) &= 3(p_1 - p_0) = d_0 \\ c'(1) &= 3(p_3 - p_2) = d_3 \end{aligned}$$

Fun facts

- partition of unity $\sum_{i=0}^{k} B_i^k(t) = 1$
- affine combinations...
  - why do they matter?

Bezier Curves

Properties

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

Change of Basis

Example

- Hermite to Bezier

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

$$\begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} = \begin{pmatrix} p_0 & p_3 & d_0 & d_3 \end{pmatrix} H B^{-1}$$

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

Forward Differencing

Step along curve

$$C(t+\delta) = C(t) + \Delta C(t) \qquad C_{n+1} = C_n + \Delta C_n$$
$$\Delta C(t+\delta) = \Delta C(t) + \Delta^2 C(t) \qquad \Delta C_{n+1} = \Delta C_n + \Delta^2 C_n$$
$$\Delta^2 C(t+\delta) = \Delta^2 C(t) + \Delta^3 C(t) \qquad \Delta^2 C_{n+1} = \Delta^2 C_n + \Delta^3 C_n$$

- choose step size
- how many flops?
- accumulation of error

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

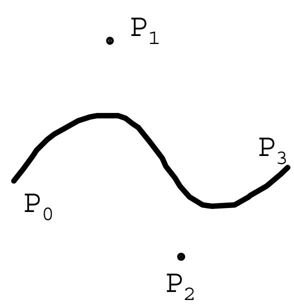
Bezier Curves

Bernstein blending polynomials

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

- cubic case

$$\begin{aligned} B_0^3(t) &= (1-t)^3 \\ B_1^3(t) &= 3t(1-t)^2 \\ B_2^3(t) &= 3t^2(1-t) \\ B_3^3(t) &= t^3 \end{aligned}$$



Bernstein Polynomials

Evaluation

- recurrence

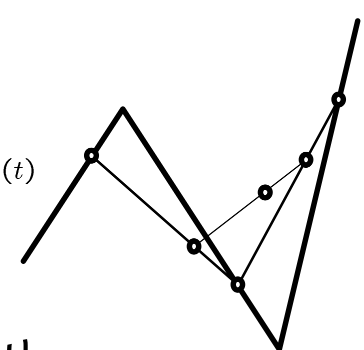
$$\begin{aligned} B_i^k(t) &= \binom{k}{i} t^i (1-t)^{k-i} \\ &= t\left(\binom{k-1}{i-1} t^{i-1}(1-t)^{k-1-(i-1)}\right) + \\ &\quad (1-t)\left(\binom{k-1}{i} t^i (1-t)^{k-1-i}\right) \\ &= tB_{i-1}^{k-1}(t) + (1-t)B_i^{k-1}(t) \end{aligned}$$

- affine (convex!) combination

Bernstein Polynomials

Evaluation

$$\begin{aligned} b_i^0(t) &= p_i \quad i = 0 \ldots 3 \\ b_i^l(t) &= (1-t)b_i^{l-1}(t) + tb_{i+1}^{l-1}(t) \end{aligned}$$

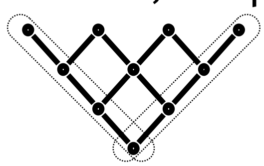


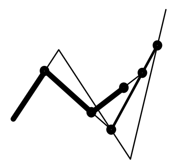
de Casteljau Algorithm

de Casteljau Alg.

Properties

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





Affine Space

Affine Combination

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

Affine Map

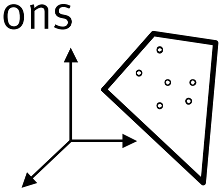
$$f : \mathbb{R}^n \mapsto \mathbb{R}^m \quad f(\sum \alpha_k u_k) = \sum \alpha_k f(u_k) \quad \sum \alpha_k = 1$$

- preserves affine combinations
- exactly the linear maps and translation

Affine Space

Definition

- subset of vector space closed under affine combinations

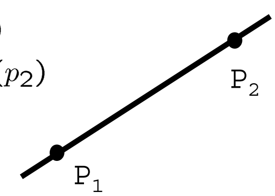


- exactly the linear subspaces plus translation (points, lines, planes...)

Affine Forms

Evaluation

$$\begin{aligned} f(p) &= f(tp_1 + (1-t)p_2) \\ &= tf(p_1) + (1-t)f(p_2) \end{aligned}$$



Differentiation

$$\begin{aligned} Df|_p &= \lim_{\tau \to 0} \frac{f(p + \tau(\bar{p} - p)) - f(p)}{\tau} \\ &= f(p + \bar{1}) - f(p) \end{aligned}$$

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