Parametric Polynomial Curves

- We'll use polynomial parametric curves, where the functions are all **polynomials** in the parameter.

$$P_t = \sum_{i=0}^{n} A_i t^i$$

- $P_t = A_0 + A_1 t + A_2 t^2 + A_3 t^3 \;\; \ldots$
- Advantages
  - easy (and efficient) to compute
  - infinitely differentiable

- We'll also assume that t varies from 0 to 1

Bezier Curves

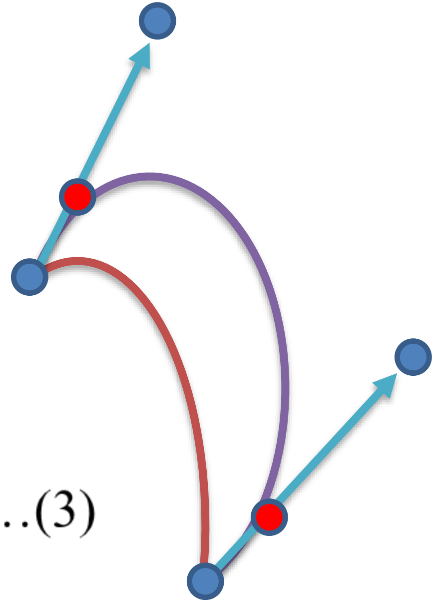
- $P_t = T \,.\, M \,.\, G$ ... (1)
- For the case of Bezier,
- $P_t = T \,.\, M_B \,.\, G_B$ ... (2)
- Where $G_B = \begin{vmatrix} P_o \\ P_1 \\ P_2 \\ P_3 \end{vmatrix}$, $P0$ .. $P3$ are 4 control points
- We know that $G_H = \begin{vmatrix} P_o \\ P_3 \\ G_0 \\ G_3 \end{vmatrix} = \begin{vmatrix} P_o \\ P_3 \\ P_1 - P_0 \\ P_3 - P_2 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{vmatrix} . \begin{vmatrix} P_o \\ P_1 \\ P_2 \\ P_3 \end{vmatrix}$

Bezier Curves

- Relationship between G_H and G_B

- $G_H = \begin{vmatrix} P_o \\ P_3 \\ G_0 \\ G_3 \end{vmatrix} = \begin{vmatrix} P_o \\ P_3 \\ P_1 - P_0 \\ P_3 - P_2 \end{vmatrix}$

- $G_{HB} = \begin{vmatrix} P_o \\ P_3 \\ 3G_0 \\ 3G_3 \end{vmatrix} = \begin{vmatrix} P_o \\ P_3 \\ 3(P_1 - P_0) \\ 3(P_3 - P_2) \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix} . \begin{vmatrix} P_o \\ P_1 \\ P_2 \\ P_3 \end{vmatrix} \quad \ldots(3)$



Bezier Curves

- 4 point based Hermite curve is,
- $$P_t = T \,.\, M_H \,.\, G_H = T \,.\, M_H \,.\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{vmatrix} . \begin{vmatrix} P_o \\ P_1 \\ P_2 \\ P_3 \end{vmatrix}$$
- And the Bezier curve in Hermite form,
- $$P_t = T \,.\, M_H \,.\, G_{HB} = T \,.\, M_H \,.\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix} . \begin{vmatrix} P_o \\ P_1 \\ P_2 \\ P_3 \end{vmatrix}, \text{ where } G_{BH} = M_{HB} \,.\, G_B$$
- $$P_t = T \,.\, M_H \,.\, G_{HB} = T \,.\, M_H \,.\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix} . \, G_B = T \,.\, M_B \,.\, G_B \quad \ldots (4)$$

Bezier Curves

- Therefore M_B can be written as,
- $$M_B = M_H \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix} = \begin{vmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix}$$
- $$M_B = \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

Finding *P(t)* in Bezier Curve

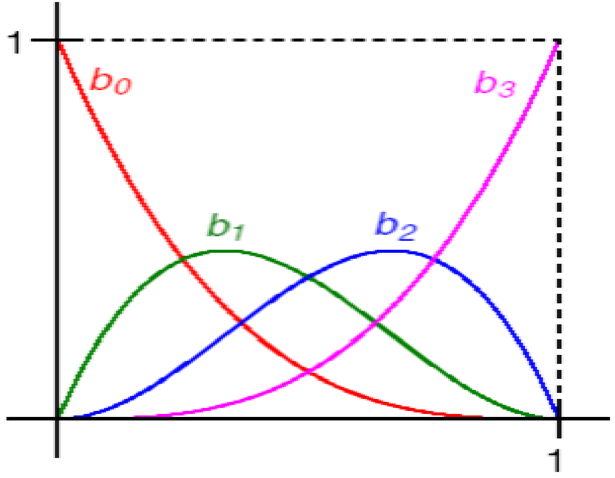
- Bernstein polynomials
- In general
- $P_{(t)} = \sum_{i=0}^{n} \binom{n}{i} t^{i} (1-t)^{n-i} P_i$

where “n choose i” is $\binom{n}{i} = \frac{n!}{(n-i)!i!}$

- This defines **Bezier curves**
- What is the relationship between the number of control points and the degree of the polynomials?

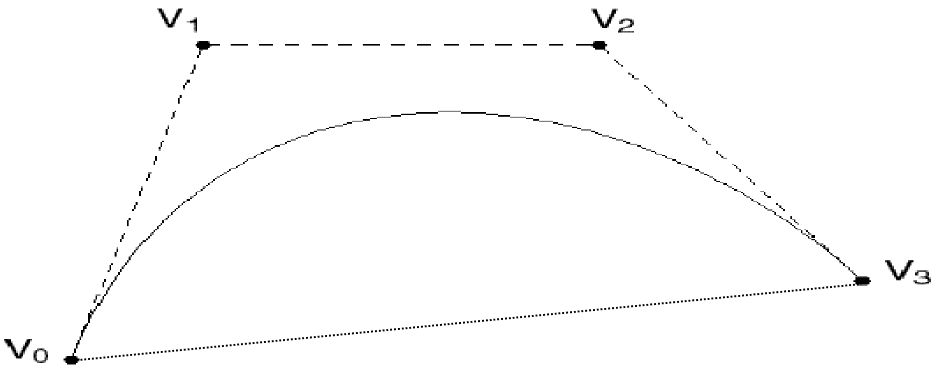
Finding *P(t)* in Bezier Curve

- The coefficients of the control points are a set of functions called the **Bernstein polynomials** also the blending functions.
- For degree 3, we have:
- $(1\text{-}t)^3 P_0 + 3t(1\text{-}t)^2 P_1 + 3t^2(1\text{-}t) p_2 + t^3 P_3$
- $B_{B0} = (1 - t)^3$
- $B_{B1} = 3(1\text{-}t)^2 t$
- $B_{B2} = 3(1\text{-}t)\, t^2$
- $B_{B3} = t^3$



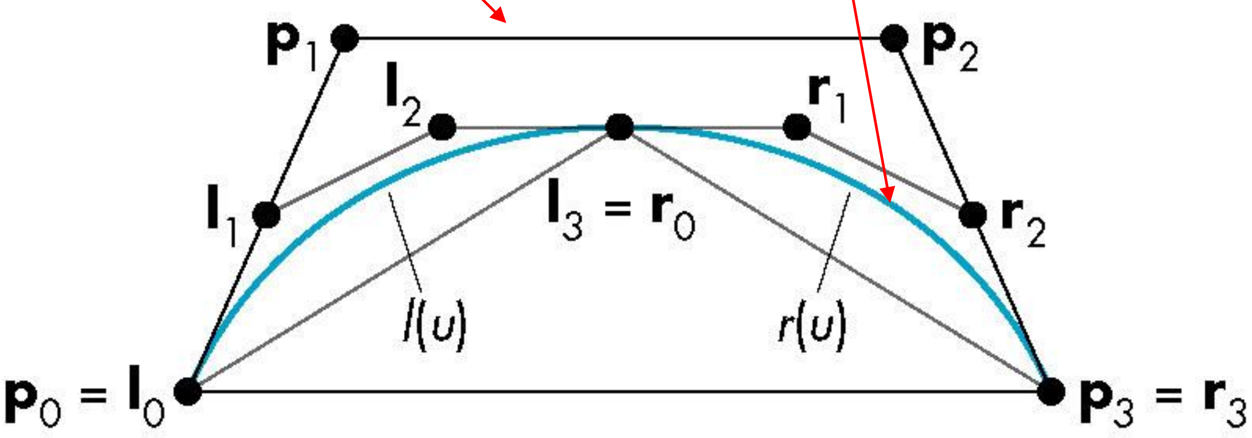
Useful properties of Bezier curve

- Bernstein polynomials has some useful properties in [0,1]:
  - each Bernstein coefficient is positive
  - sum of all four coefficients is always exactly 1
- These properties together imply that the curve lies within the **convex hull** of its control points. (convex hull is the smallest convex polygon that contains the control points)



Splitting a Cubic Bezier

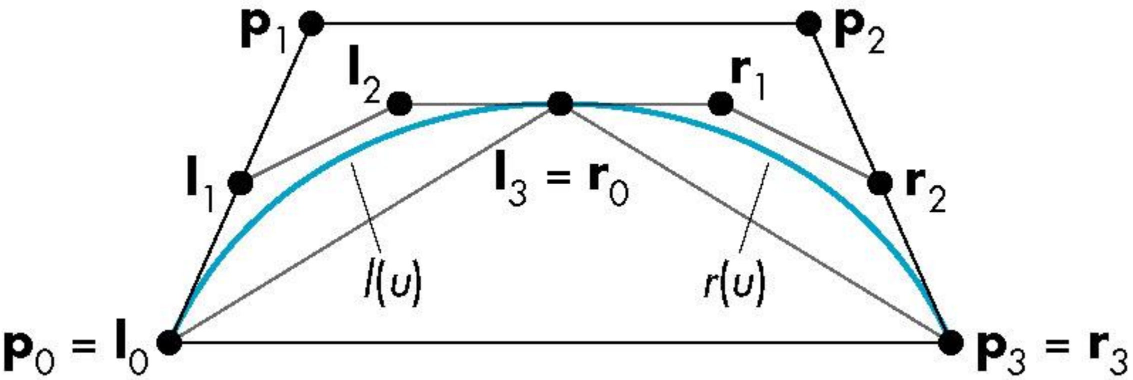
p_0, p_1 , p_2 , p_3 determine a cubic Bezier polynomial and its convex hull



Consider left half l(u) and right half r(u)

l(t) and r(t)

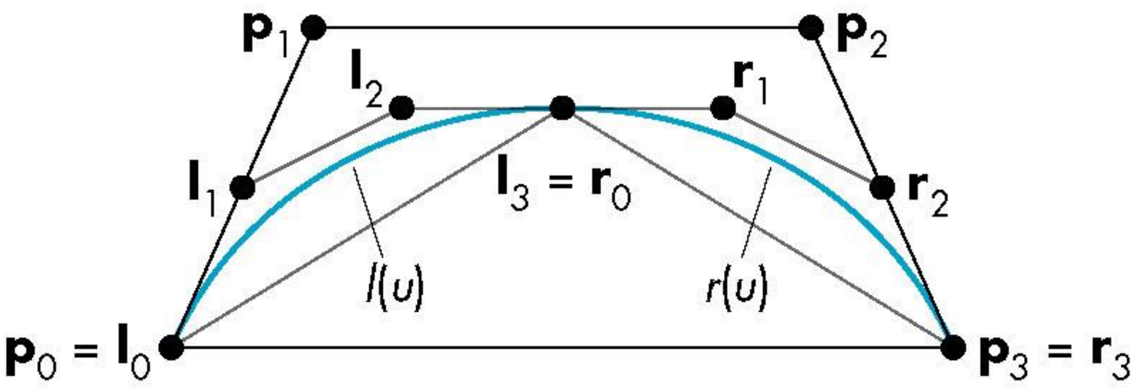
Since l(t) and r(t) are Bezier curves, we should be able to find two sets of control points $\{l_0, l_1, l_2, l_3\}$ and $\{r_0, r_1, r_2, r_3\}$ that determine them



Convex Hulls

$\{l_0, l_1, l_2, l_3\}$ and $\{r_0, r_1, r_2, r_3\}$each have a convex hull that that is closer to *p(t)* than the convex hull of $\{p_0, p_1, p_2, p_3\}$ This is known as the *variation diminishing property*.

The polyline from l_0 to l_3 (= r_0) to r_3 is an approximation to *p(t)*. Repeating recursively we get better approximations.



Efficient Form

Assuming t = 0.5

$l_0 = p_0$

$l_1 = \frac{1}{2}(p_0 + p_1)$

$l_2 = \frac{1}{2}(l_1 + \frac{1}{2}(p_1 + p_2))$

$\quad = \frac{1}{4}(p_0 + 2p_1 + p_2)$

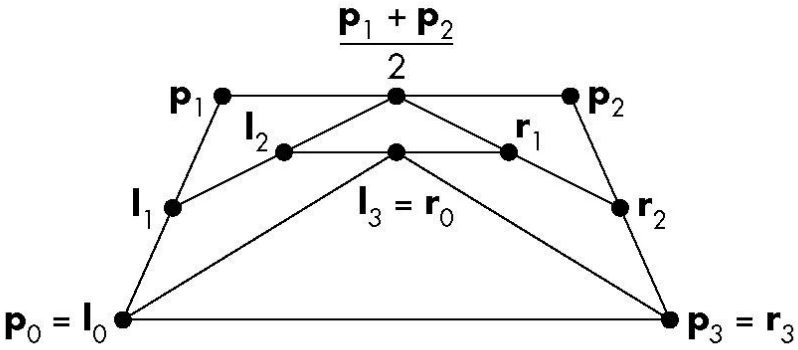
$r_3 = p_3$

$r_2 = \frac{1}{2}(p_2 + p_3)$

$r_1 = \frac{1}{4}(p_1 + 2p_2 + p_3)$

$l_3 = r_0 = \frac{1}{2}(l_2 + r_1)$

$\quad = \frac{1}{8}(p_0 + 3p_1 + 3p_2 + p_3)$



Requires only shifts and adds!

Left and Right Segments

The geometric constrain of the left segment assuming t=0.5 can be written as

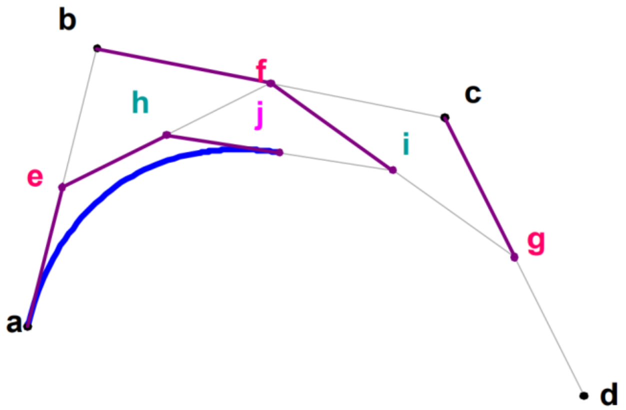
$$G_{BL} = \frac{1}{8}\begin{vmatrix} 8 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 \\ 2 & 4 & 2 & 0 \\ 1 & 3 & 3 & 1 \end{vmatrix} \bullet \begin{vmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{vmatrix}$$

And the right segment is

$$G_{BR} = \frac{1}{8}\begin{vmatrix} 1 & 3 & 3 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 8 \end{vmatrix} \bullet \begin{vmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{vmatrix}$$

Geometric Construction by De Casteljau

- Casteljau's algorithm provides a method for geometrically constructing the Bezier curve. In the following example construction of a cubic Bezier is demonstrated.



- For the case of a cubic Bezier, we consider the three limbs of the open control polygon *ab*, *bc*, and *cd*. Next create the intermediate points *e*, *f* and *g* in the ratios *ae/ab* = *bf/bc* = *cg/cd* = *t* (given value of the parameter). Continuing iteratively we obtain the point j on the curve. Similarly a series of values of *t* give rise to the corresponding ratios and hence the points on the Bezier curve.