Parametric Curves

CS 318

Interactive Computer Graphics

John C. Hart

Linear Interpolation

- Need to get from point $\mathbf{p}_0$ to point $\mathbf{p}_1$
- Define a parametric function $\mathbf{p}(t)$

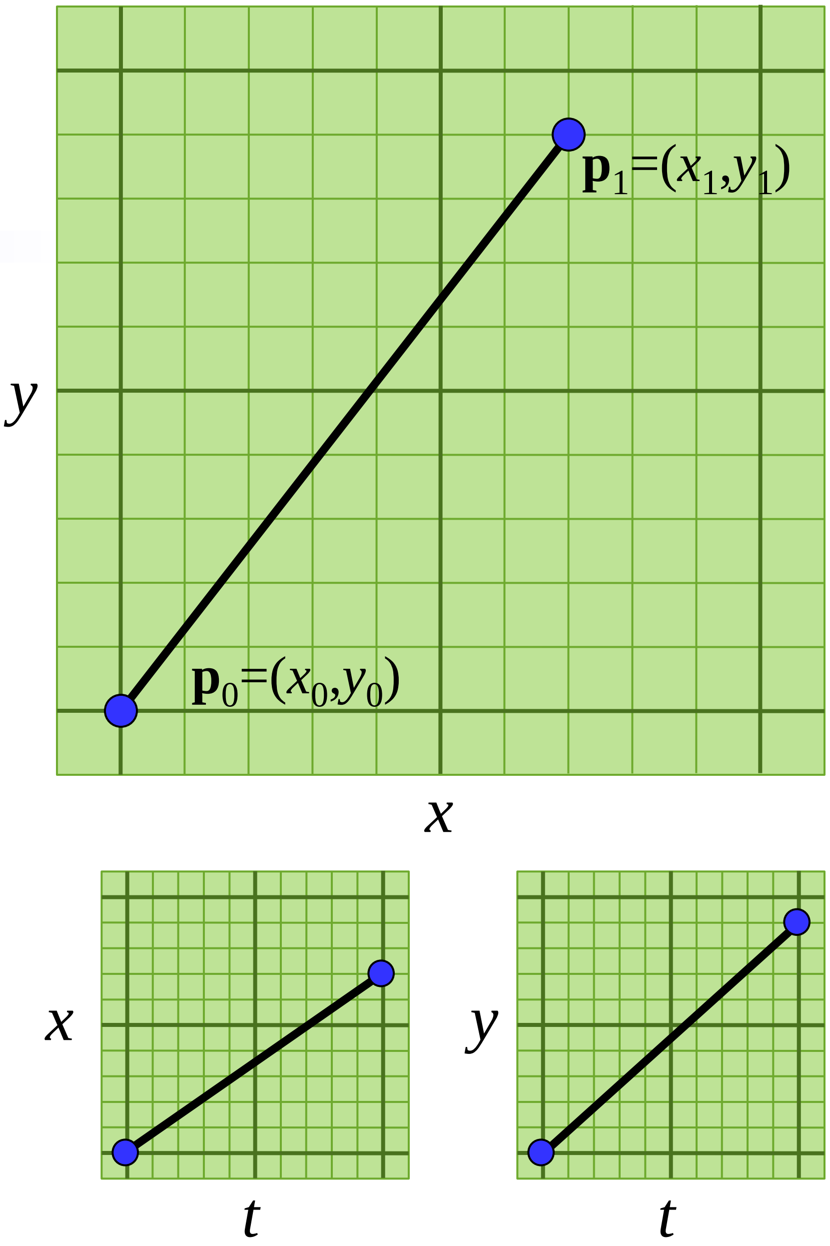
$$\mathbf{p}(0) = \mathbf{p}_0,\ \mathbf{p}(1) = \mathbf{p}_1$$

- Separate into coordinate functions

$$\mathbf{p}(t) = (x(t), y(t))$$
$$x(0) = x_0,\ x(1) = x_1$$
$$y(0) = y_0,\ y(1) = y_1$$

- Interpolate

$$\mathbf{p}(t) = \mathbf{p}_0 + t\,(\mathbf{p}_1 - \mathbf{p}_0) = (1-t)\mathbf{p}_0 + t\,\mathbf{p}_1$$
$$x(t) = x_0 + t(x_1 - x_0) = (1-t)x_0 + t\,x_1$$
$$y(t) = y_0 + t(y_1 - y_0) = (1-t)y_0 + t\,y_1$$



Hermite Interpolation

- From point $\mathbf{p}_0$ along $\mathbf{p}'_0$ to point $\mathbf{p}_1$ toward $\mathbf{p}'_1$
- Define a parametric function $\mathbf{p}(t)$

$$\mathbf{p}(0) = \mathbf{p}_0,\ \mathbf{p}(1) = \mathbf{p}_1$$

$$\mathbf{p}'(0) = \mathbf{p}'_0,\ \mathbf{p}'(1) = \mathbf{p}'_1$$

- Separate into coordinate functions

$$x(0) = x_0,\ x(1) = x_1$$

$$x'(0) = x'_0,\ x'(1) = x'_1$$

- Need cubic function

$$x(t) = At^3 + Bt^2 + Ct + D$$

$$x'(t) = 3At^2 + 2Bt + C$$

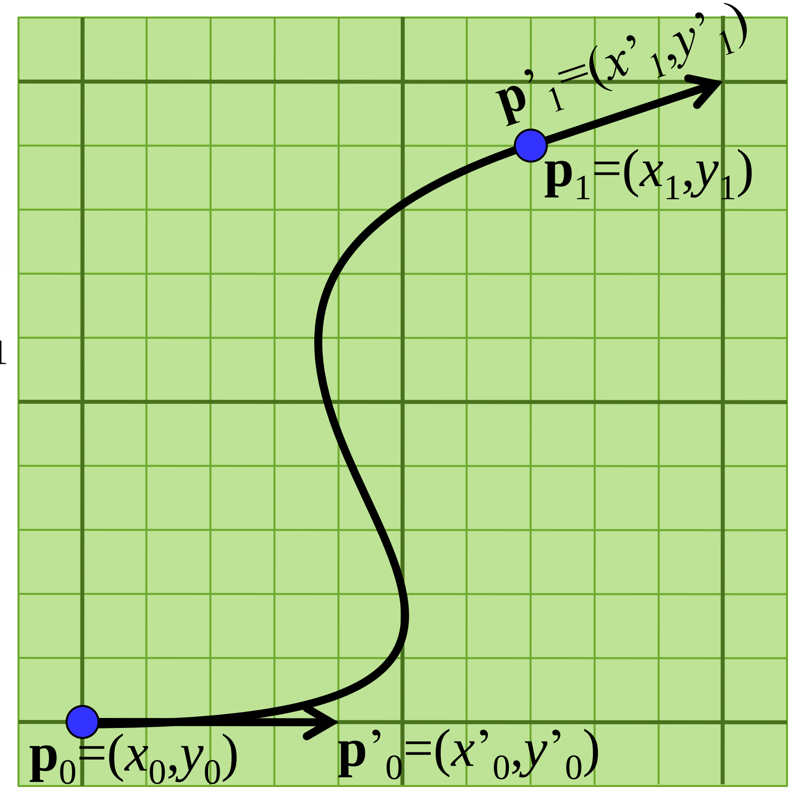
- Solve

$$A = 2x_0 - 2x_1 + x'_0 + x'_1$$

$$B = -3x_0 + 3x_1 - 2x'_0 - x'_1$$

$$C = x'_0, \qquad D = x_0$$

y



$x(0) = D = x_0, \qquad x'(0) = C = x'_0$

$x(1) = A + B + C + D = x_1$

$A + B = x_1 - x_0 - x'_0$

$x'(1) = 3A + 2B + C = x'_1$

$3A + 2B = x'_1 - x'_0$

$A \quad = x'_1 - x'_0 - 2x_1 + 2x_0 + 2x'_0$

$\quad = x'_1 + x'_0 - 2x_1 + 2x_0$

$B \quad = x_1 - x_0 - x'_0 - x'_1 - x'_0 + 2x_1 - 2x_0$

$\quad = 3x_1 - 3x_0 - x'_1 - 2x'_0$

Hermite Interpolation

- From point $\mathbf{p}_0$ along $\mathbf{p'}_0$ to point $\mathbf{p}_1$ toward $\mathbf{p'}_1$
- Define a parametric function $\mathbf{p}(t)$

$$\mathbf{p}(0) = \mathbf{p}_0,\ \mathbf{p}(1) = \mathbf{p}_1$$

$$\mathbf{p'}(0) = \mathbf{p'}_0,\ \mathbf{p'}(1) = \mathbf{p'}_1$$

- Separate into coordinate functions

$$x(0) = x_0,\ x(1) = x_1$$

$$x'(0) = x'_0,\ x'(1) = x'_1$$

- Need cubic function

$$x(t) = At^3 + Bt^2 + Ct + D$$

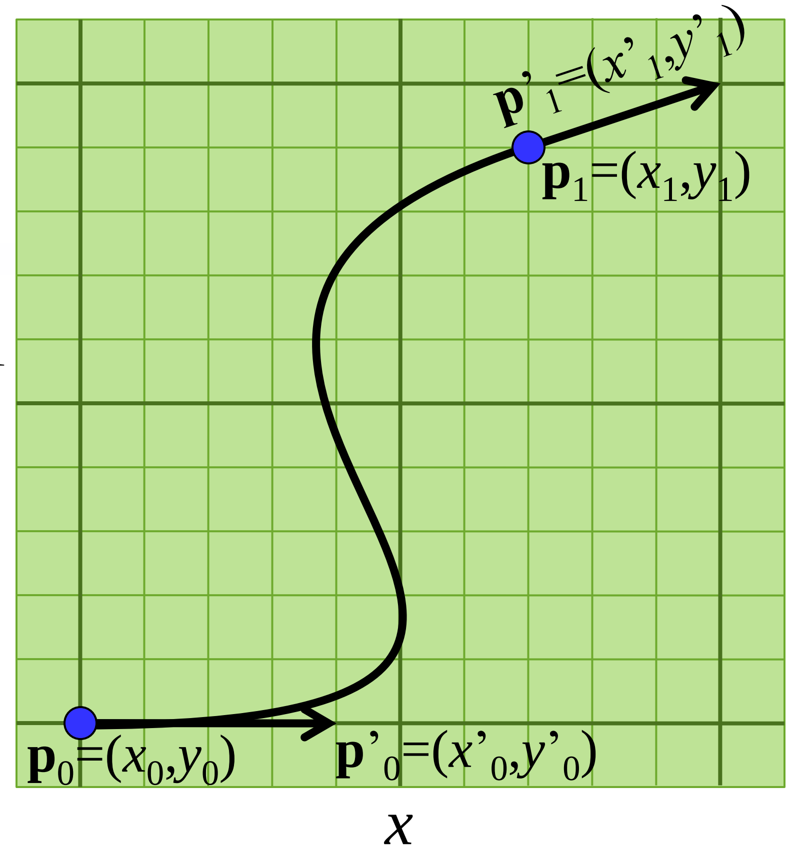
$$x'(t) = 3At^2 + 2Bt + C$$

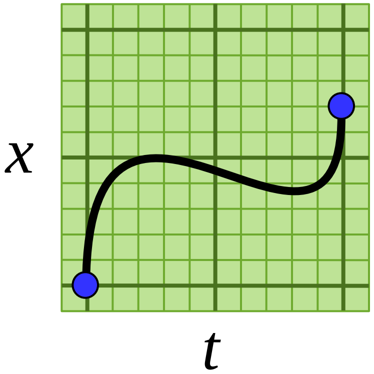
- Solve

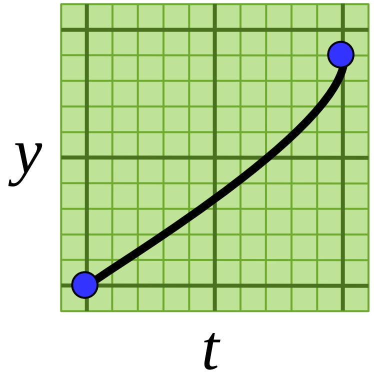
$$A = 2x_0 - 2x_1 + x'_0 + x'_1$$

$$B = -3x_0 + 3x_1 - 2x'_0 - x'_1$$

$$C = x'_0, \qquad D = x_0$$

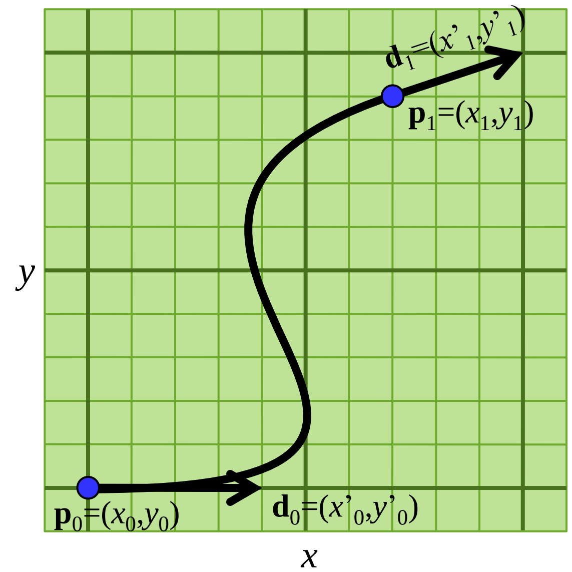






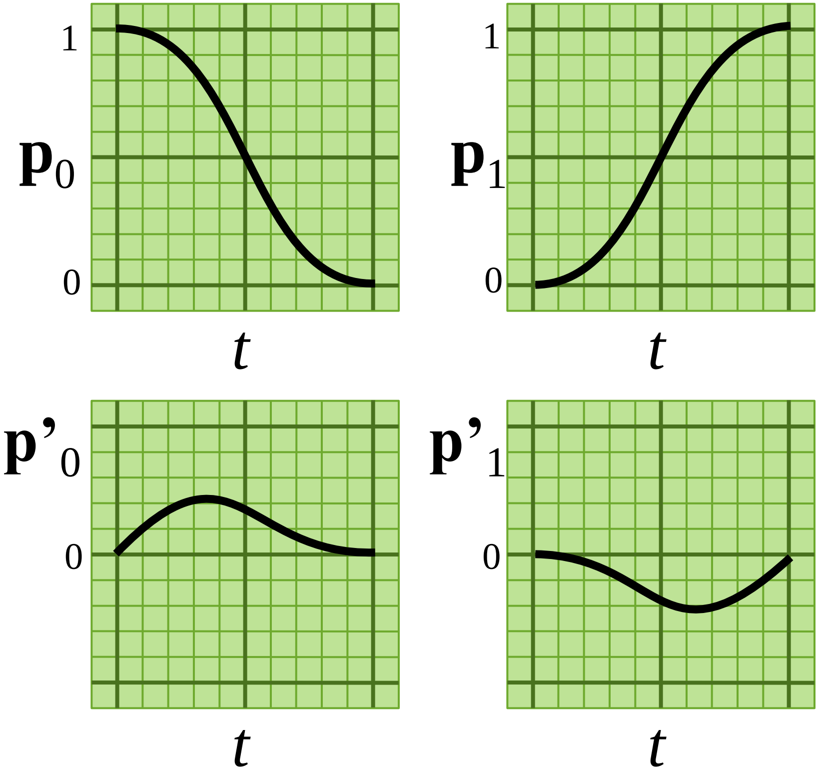
Hermite Matrix

$$\mathbf{p}(t) = (2\mathbf{p}_0 - 2\mathbf{p}_1 + \mathbf{p}'_0 + \mathbf{p}'_1)\,t^3 + (-3\mathbf{p}_0 + 3\mathbf{p}_1 - 2\mathbf{p}'_0 - \mathbf{p}'_1)\,t^2 + \mathbf{p}'_0\,t + \mathbf{p}_0\,(1)$$



$$\mathbf{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}'_0 \\ \mathbf{p}'_1 \end{bmatrix}$$

$$\mathbf{p}(\mathrm{t}) = (2t^3 - 3t^2 + 1)\,\mathbf{p}_0 + (-2t^3 + 3t^2)\,\mathbf{p}_1 + (t^3 - 2t^2 + 1)\,\mathbf{p}'_0 + (t^3 - t^2)\,\mathbf{p}'_1$$



Linear Interpolation

- Need to get from point $\mathbf{p}_0$ to point $\mathbf{p}_1$
- Define a parametric function $\mathbf{p}(t)$

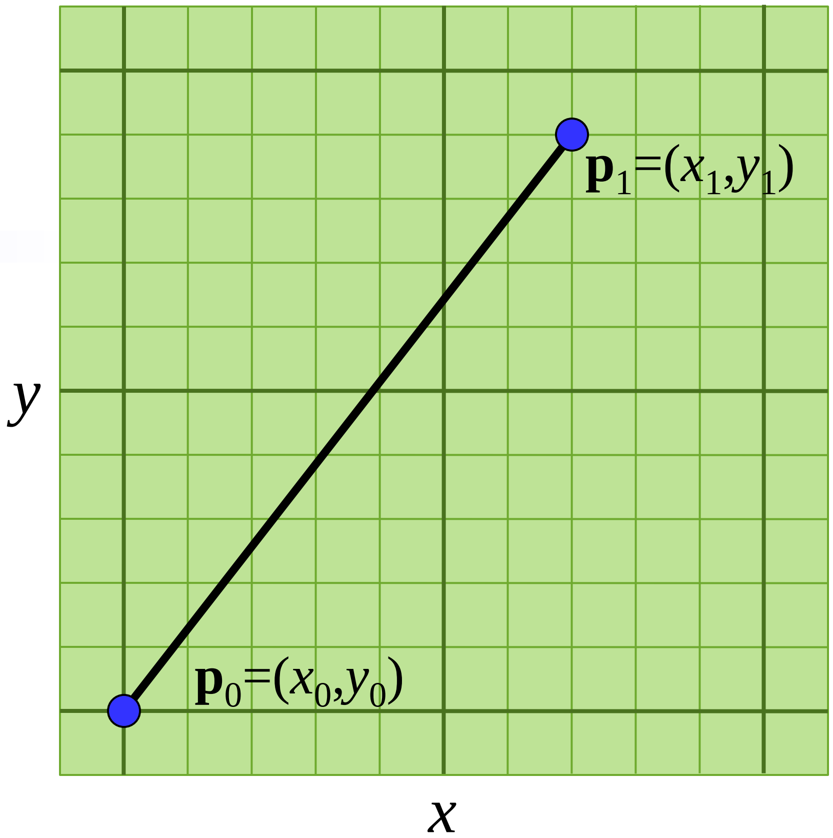
$$\mathbf{p}(0) = \mathbf{p}_0,\ \mathbf{p}(1) = \mathbf{p}_1$$

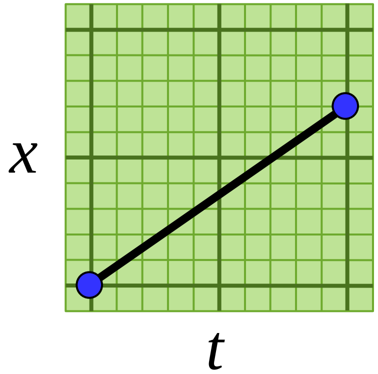
- Separate into coordinate functions

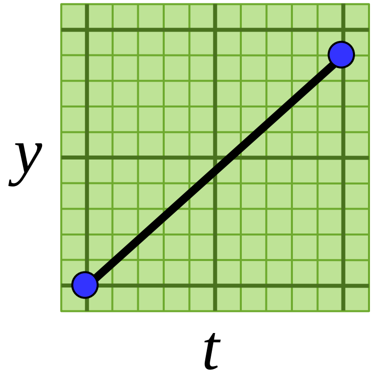
$$\mathbf{p}(t) = (x(t), y(t))$$
$$x(0) = x_0,\ x(1) = x_1$$
$$y(0) = y_0,\ y(1) = y_1$$

- Interpolate

$$\mathbf{p}(t) = \mathbf{p}_0 + t\,(\mathbf{p}_1 - \mathbf{p}_0) = (1-t)\mathbf{p}_0 + t\,\mathbf{p}_1$$
$$x(t) = x_0 + t(x_1 - x_0) = (1-t)x_0 + t\,x_1$$
$$y(t) = y_0 + t(y_1 - y_0) = (1-t)y_0 + t\,y_1$$

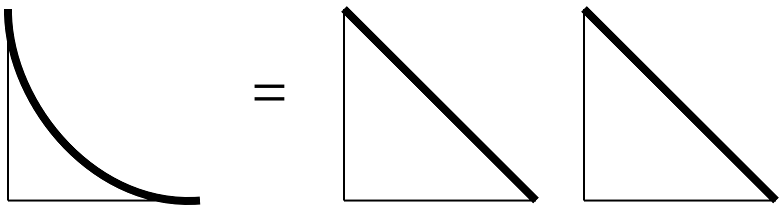




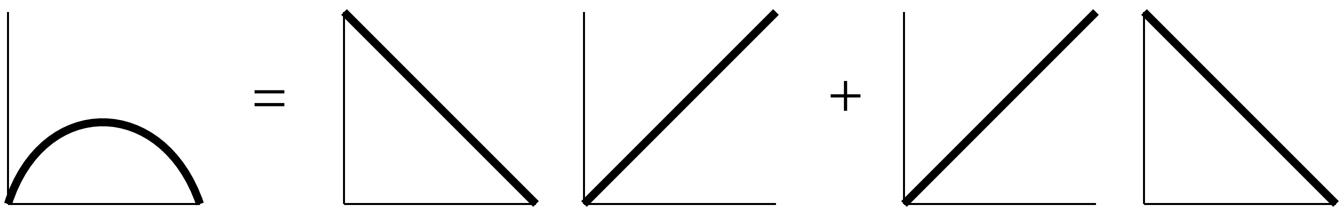


Interpolating Interpolations

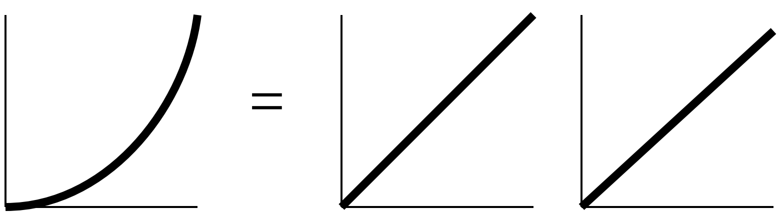
$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$$



$$B_0^2(t) = (1\text{-t}) \quad B_0^1(t)$$

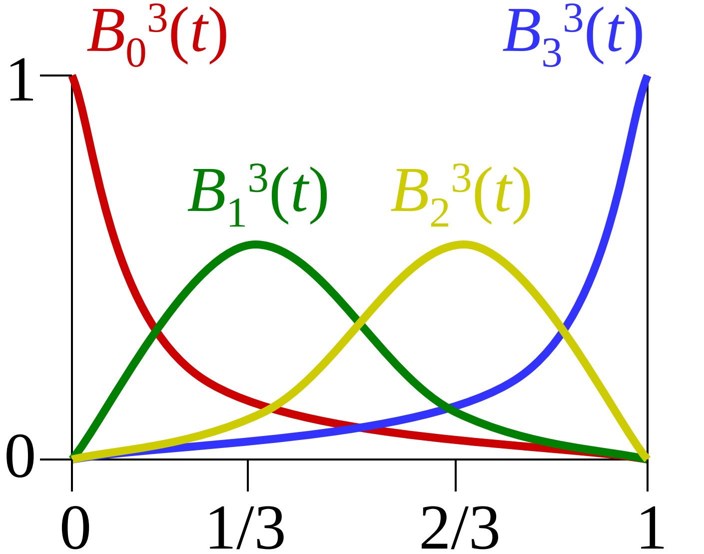


$$B_1^2(t) = (1-t) \quad B_1^1(t) + t \quad B_0^1(t)$$



$$B_2^2(t) = t \quad B_1^1(t)$$

Bernstein Polynomials



- Defined for any degree

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

- n choose i

$$\binom{n}{i} = n!/(i!(n-i)!) = \binom{n-1}{i} + \binom{n-1}{i-1}$$

- Partition of unity
  - Sum to one for any t in [0,1]

$$\Sigma_{i=0..n} B_i^n(t) = 1$$

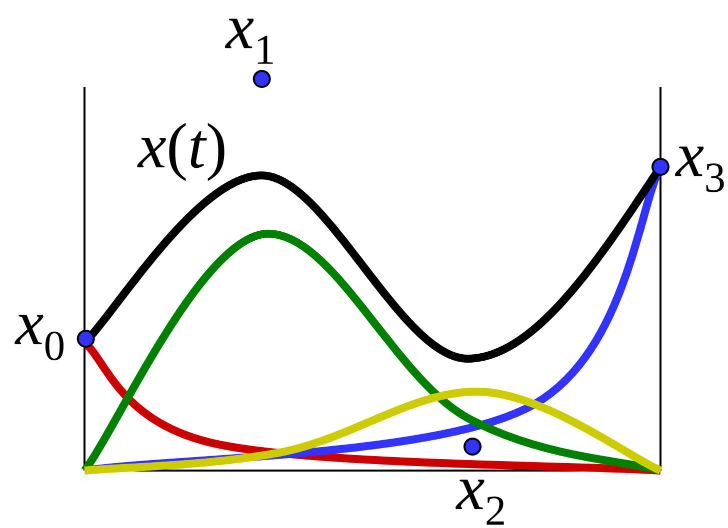
- Higher degrees lerps of lower degrees

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

$$= \binom{n-1}{i}\, t^i\, (1-t)^{n-i} + \binom{n-1}{i-1}\, t^i\, (1-t)^{n-i}$$

$$= (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$$

$$x(t) = aB_0^3(t) + bB_1^3(t) + cB_2^3(t) + dB_3^3(t)$$

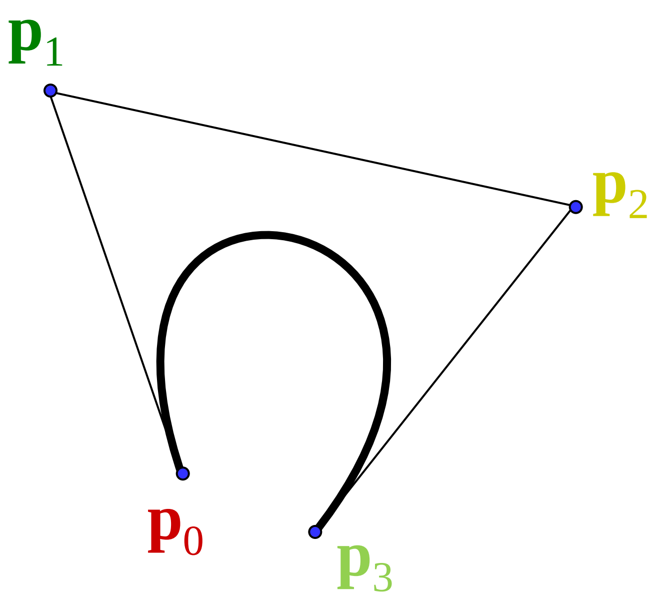


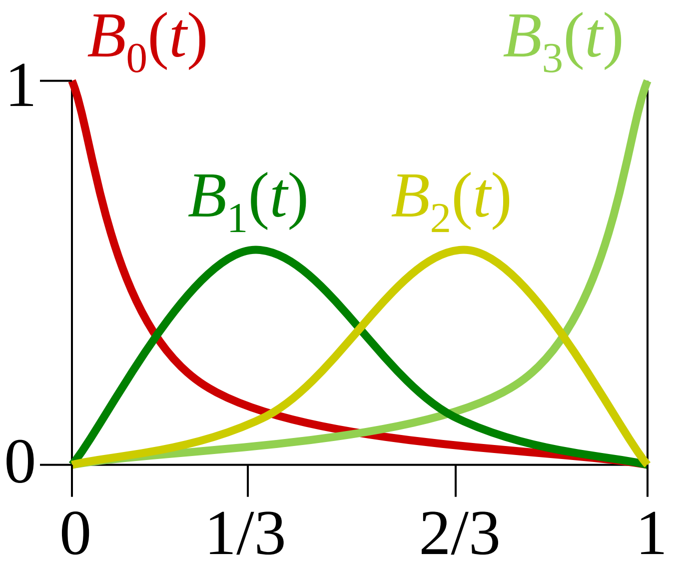
Cubic Bezier Curves

- Bernstein basis applied to points

$$\mathbf{p}(t) = \Sigma_i \binom{3}{i} t^i (1-t)^{3-i} \mathbf{p}_i$$

- Bezier curve specified by four *control* points including two *endpoints*
- Affine invariance:
  - Let M be a 4x4 transformation
  - Then $M\,\mathbf{p}(t) = \Sigma_i B_i(t)\, M\mathbf{p}_i$
- Curve entirely contained in the convex hull of the control points



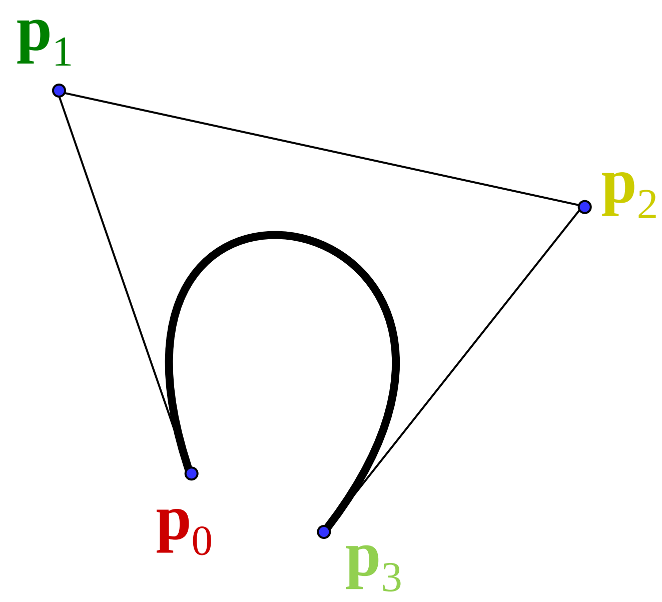


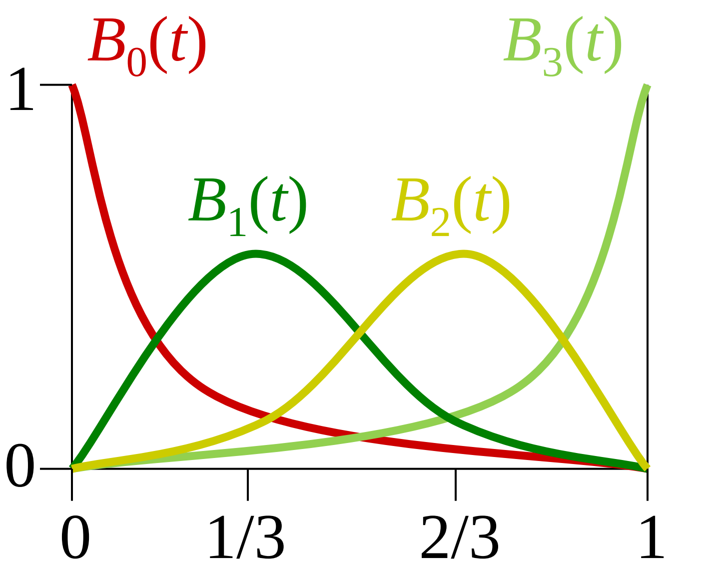
Cubic Bezier Matrix

$$\mathbf{p}(t) = (1\text{-}t)^3\mathbf{p}_0 + 3(1\text{-}t)^2t\mathbf{p}_1 + 3(1\text{-}t)t^2\mathbf{p}_2 + t^3\mathbf{p}_3$$

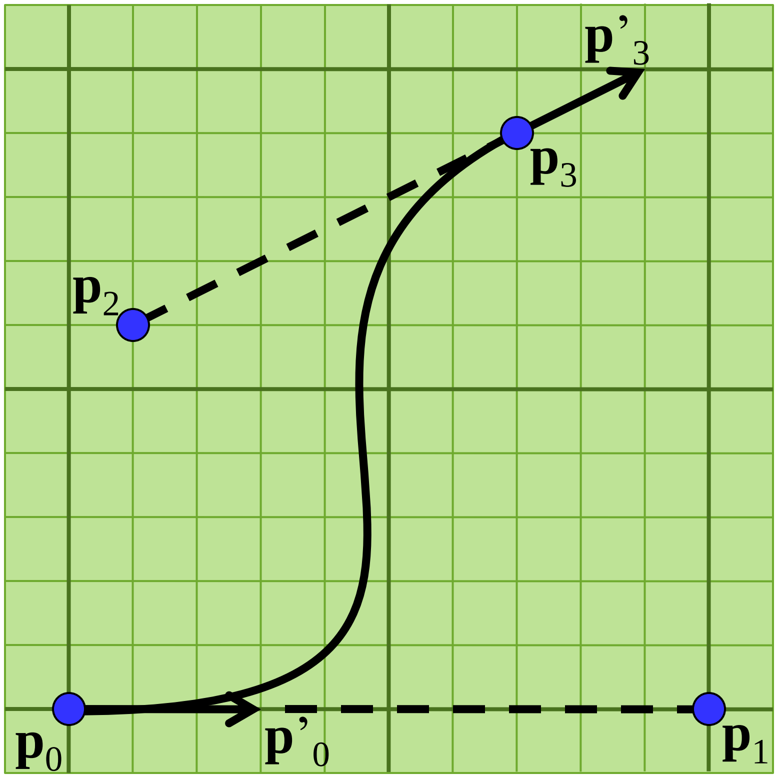
$$\begin{aligned} = \; & (1 - 3t + 3t^2 - t^3) \; \mathbf{p}_0 \; + \\ & (3t - 6t^2 + 3t^3) \; \mathbf{p}_1 \; + \\ & (3t^2 - 3t^3) \; \mathbf{p}_2 \; + \\ & t^3 \; \mathbf{p}_3 \end{aligned}$$

$$\mathbf{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$





Bezier v. Hermite



$\mathbf{p}_1 = \mathbf{p}_0 + 3\,\mathbf{p}'_0$

$\mathbf{p}_2 = \mathbf{p}_3 - 3\,\mathbf{p}'_3$

- Bezier

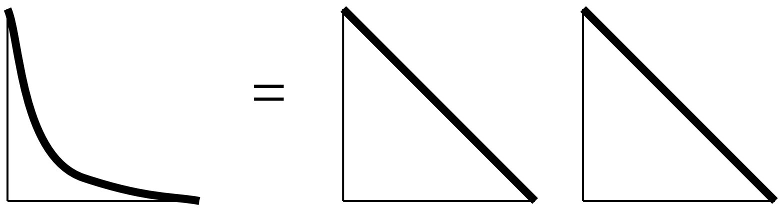
$$\mathbf{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -3 & 1 \\ 2 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

- Hermite

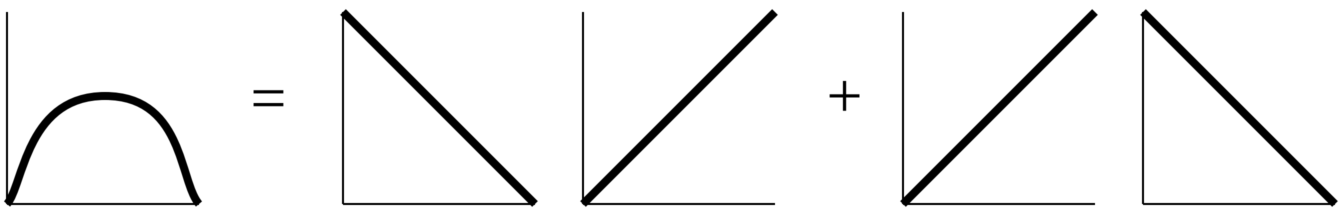
$$\mathbf{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{p}'_0 \\ \mathbf{p}'_3 \end{bmatrix}$$

Building Bernsteins

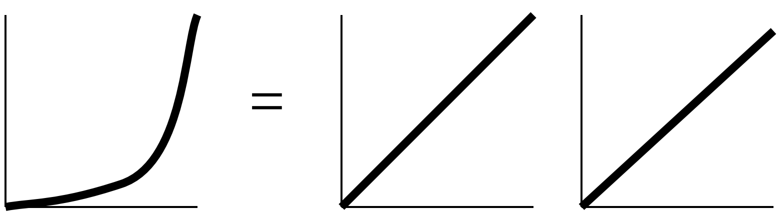
$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$$



$$B_0^2(t) \quad = \quad (1\text{-t}) \quad B_0^1(t)$$

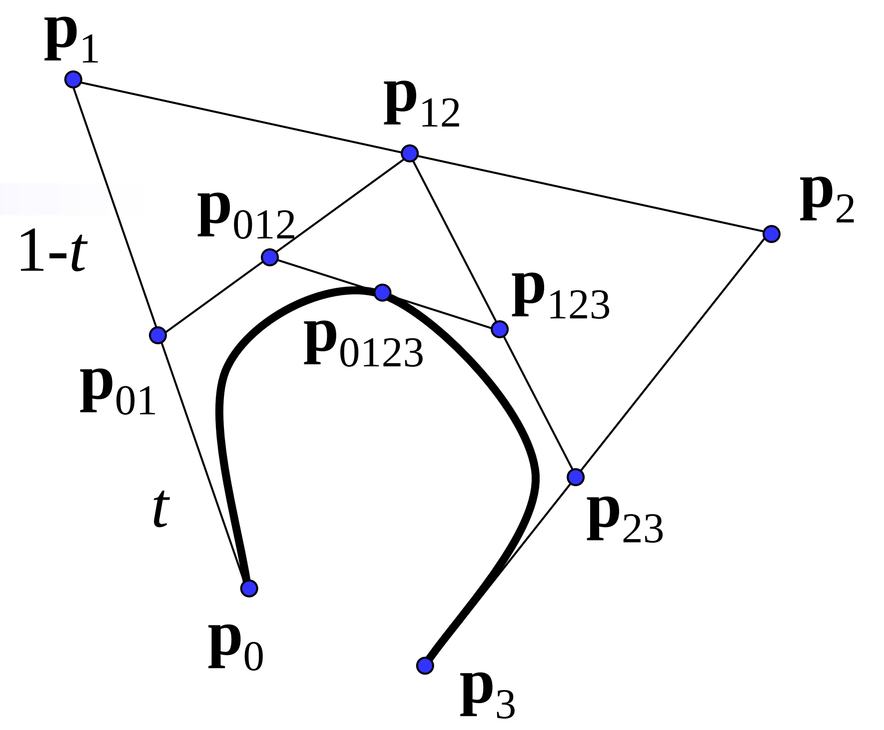


$$B_1^2(t) \quad = \quad (1-t) \quad B_1^1(t) \quad + \quad t \quad B_0^1(t)$$



$$B_2^2(t) \quad = \quad t \quad B_1^1(t)$$

de Casteljau Algorithm



- Cascading lerps

  $\mathbf{p}_{01} = (1\text{-}t)\,\mathbf{p}_0 + t\,\mathbf{p}_1$

  $\mathbf{p}_{12} = (1\text{-}t)\,\mathbf{p}_1 + t\,\mathbf{p}_2$

  $\mathbf{p}_{23} = (1\text{-}t)\,\mathbf{p}_2 + t\,\mathbf{p}_3$

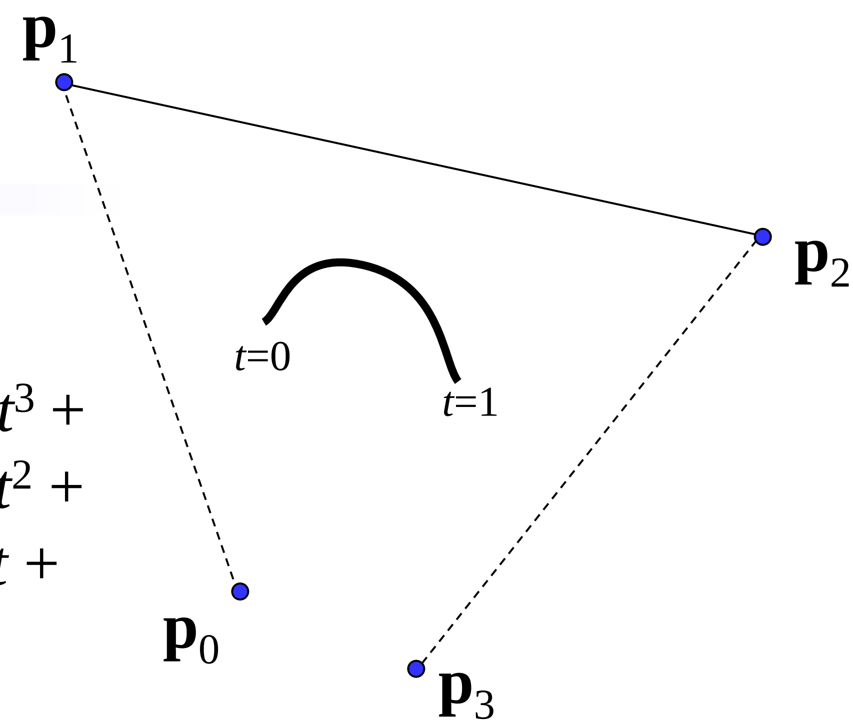
  $\mathbf{p}_{012} = (1\text{-}t)\,\mathbf{p}_{01} + t\,\mathbf{p}_{12}$

  $\mathbf{p}_{123} = (1\text{-}t)\,\mathbf{p}_{12} + t\,\mathbf{p}_{23}$

  $\mathbf{p}_{0123} = (1\text{-}t)\,\mathbf{p}_{012} + t\,\mathbf{p}_{123}$

- Subdivides curve at $\mathbf{p}_{0123}$
  - $\mathbf{p}_0\ \mathbf{p}_{01}\ \mathbf{p}_{012}\ \mathbf{p}_{0123}$
  - $\mathbf{p}_{0123}\ \mathbf{p}_{123}\ \mathbf{p}_{23}\ \mathbf{p}_3$
- Repeated subdivision converges to curve

B-Spline Segment



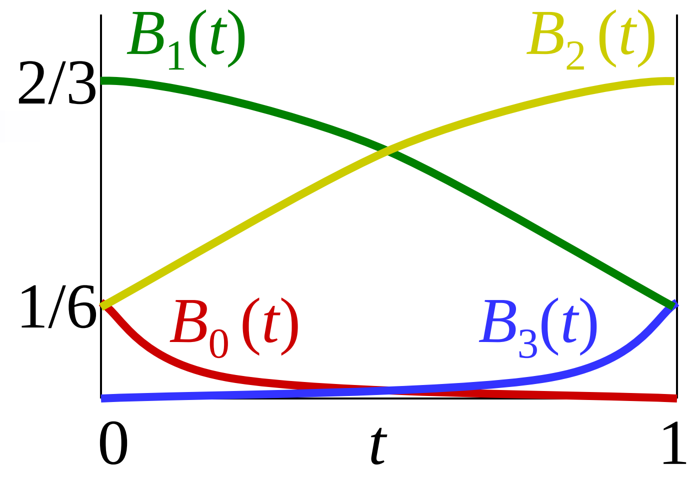
$$\begin{aligned}\mathbf{p}(t) = {} & (-1/6\mathbf{p}_0 + 1/2\mathbf{p}_1 - 1/2\mathbf{p}_2 + 1/6\mathbf{p}_3)t^3 + \\ & (\ 1/2\mathbf{p}_0 - \mathbf{p}_1 + 1/2\mathbf{p}_2)t^2 + \\ & (-1/2\mathbf{p}_0 + 1/2\mathbf{p}_2)t + \\ & \ 1/6\mathbf{p}_0 + 2/3\mathbf{p}_1 + 1/6\mathbf{p}_2\end{aligned}$$

but makes more sense as…

$$\begin{aligned}\mathbf{p}(t) = {} & (-1/6t^3 + 1/2t^2 - 1/2t + 1/6)\mathbf{p}_0 + \\ & (\ 1/2t^3 - t^2 + 2/3)\mathbf{p}_1 + \\ & (-1/2t^3 + 1/2t^2 + 1/2t + 1/6)\mathbf{p}_2 + \\ & (\ 1/6t^3)\mathbf{p}_3\end{aligned}$$

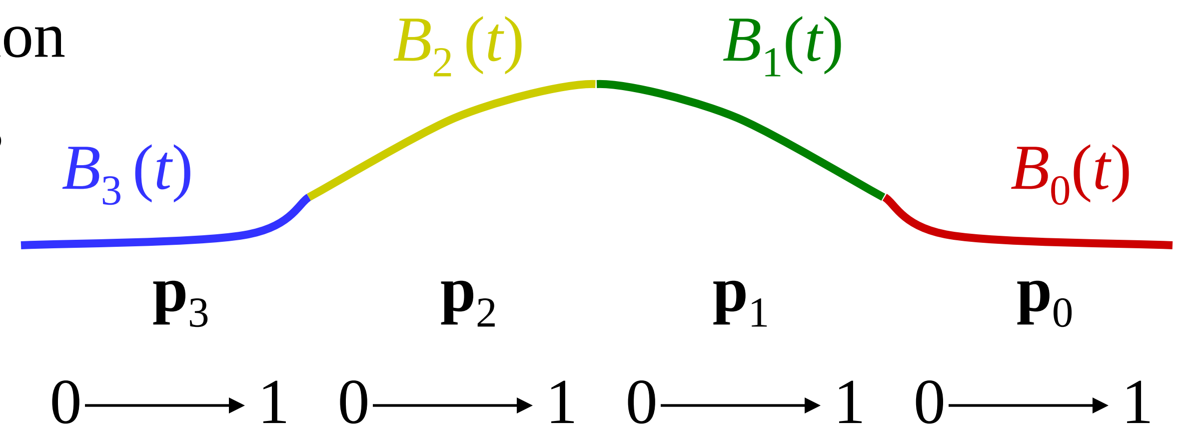
B-Spline Basis

$$
\begin{aligned}
\mathbf{p}(t) = {} & (-1/6t^3 + 1/2t^2 - 1/2t + 1/6)\mathbf{p}_0 + \\
& (\ 1/2t^3 - t^2 + 2/3)\mathbf{p}_1 + \\
& (-1/2t^3 + 1/2t^2 + 1/2t + 1/6)\mathbf{p}_2 + \\
& (\ 1/6t^3)\mathbf{p}_3
\end{aligned}
$$



$$= B_0(t)\mathbf{p}_0 + B_1(t)\mathbf{p}_1 + B_2(t)\mathbf{p}_2 + B_3(t)\mathbf{p}_3$$

- Piecewise cubic approximation of a Gaussian bump function
- Progressively weights points along spline

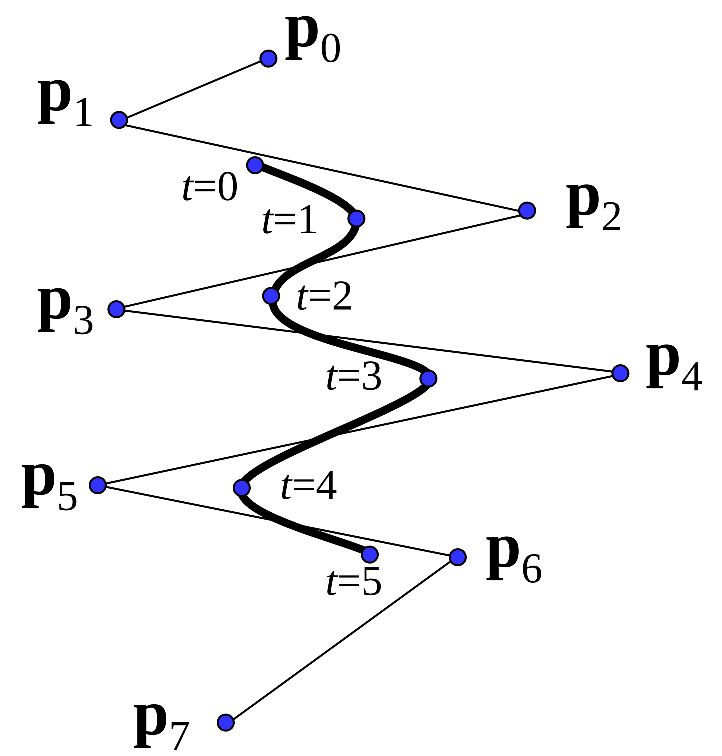


Uniform B-Splines

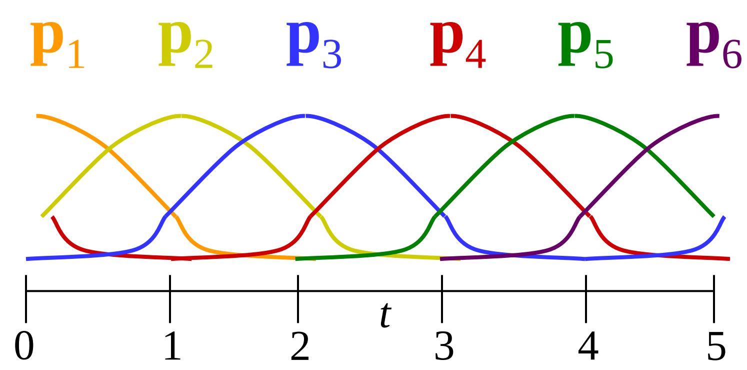
- Notation
  - d = degree of polynomial
  - k = order of polynomal = d + 1
  - E.g. cubic: d = 3, k = 4
- Segment $i \le t < i+1$ uses $k = d + 1$ control points $\mathbf{p}_i$ to $\mathbf{p}_{i+d}$

$$\mathbf{p}(\mathrm{t}) = \Sigma_{j=0}^{3} B_j(t \bmod 1)\mathbf{p}_{i+j}$$

- Normalized basis function $N_{i,d}(t)$
- $N_{i,d}(t) = B_{\mathrm{floor}(t-i)}(t \bmod 1)$ if $i \le t < i+d+1$
  - Otherwise its zero
- Knot vector
  - e.g. [0,1,2,3,4,5,6,7]
  - in general $[t_0, t_1, \ldots, t_{n+k}]$



$$\mathbf{p}(t) = \sum_{i=0}^{n} N_{i,k}(t)\mathbf{p}_i$$



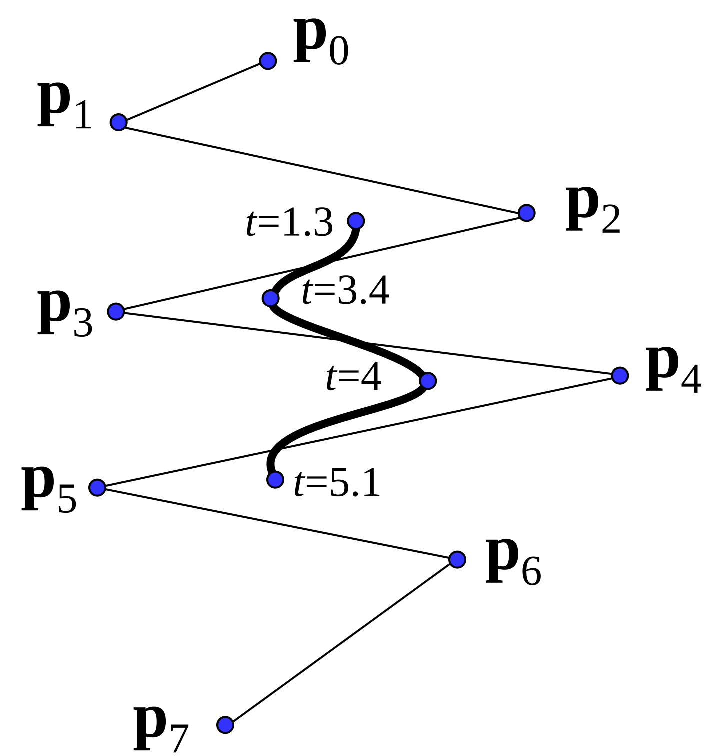
Non-Uniform B-Splines

knot vector: [0, .5, 1.3, 3.4, 4, 5.1, 6, 7]

N=3 curve segments

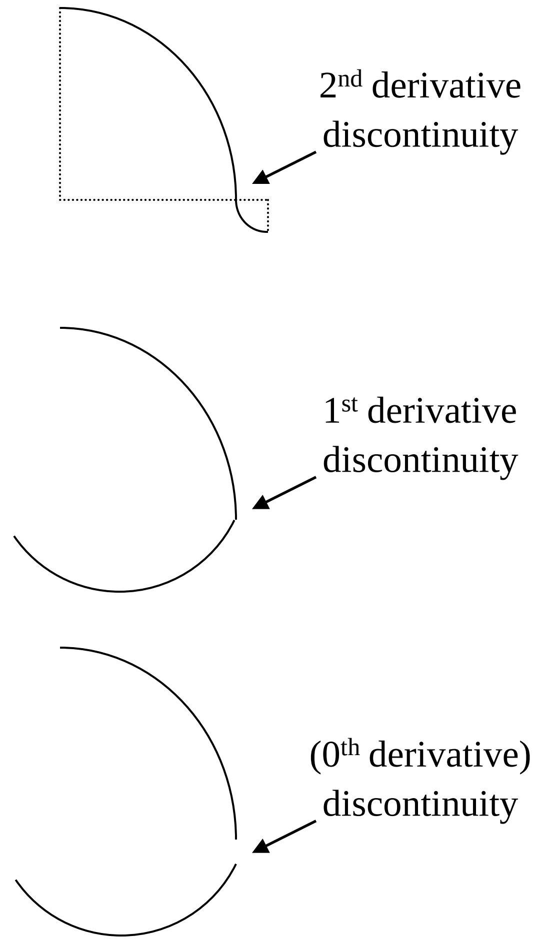
d=3 (cubic)

- Can specify an arbitrary parameter t_i at each control point $\mathbf{p}_i$
- Let N = # of polynomial curve segments
- Parameters contained in a *knot vector*
  - Length $(N+1) + 2d - 2$
  - $[t_0,t_1,t_2,t_3,\ldots,t_{N+2d-2}]$
  - Cubic: $[t_0,t_1,t_2,t_3,\ldots,t_{N+4}]$
- Domain of resulting curve is $[t_{d-1},t_{N+d-1}]$
  - Cubic: domain = $[t_2,t_{N+2}]$ (segments $[t_0,t_1]$, $[t_1,t_2]$, $[t_{N+2},t_{N+3}]$ and $[t_{N+3},t_{N+4}]$ aren't plotted)
  - Need d–1 "extra" knots at the beginning and end of the knot vector



Knot Multiplicity

- Knot multiplicity = # of times a given knot appears in the knot vector
- Continuity = *d* – multiplicity
- Cubic example
  - All knots unique – 2nd derivative continuity
  - Multiplicity two – 1st derivative cont.
  - Multiplicity three – 0th derivative cont.
  - Multiplicity four – discontinuous
- Endpoint interpolation
  - Knots of multiplicity *d*+1 at beginning and end of knot vector
  - e.g. [0, 0, 0, 0, 1, 2, 3, 3, 3, 3]



Recursion

knot vector: [0, .5, 1.3, 3.4, 4, 5.1, 6, 7]

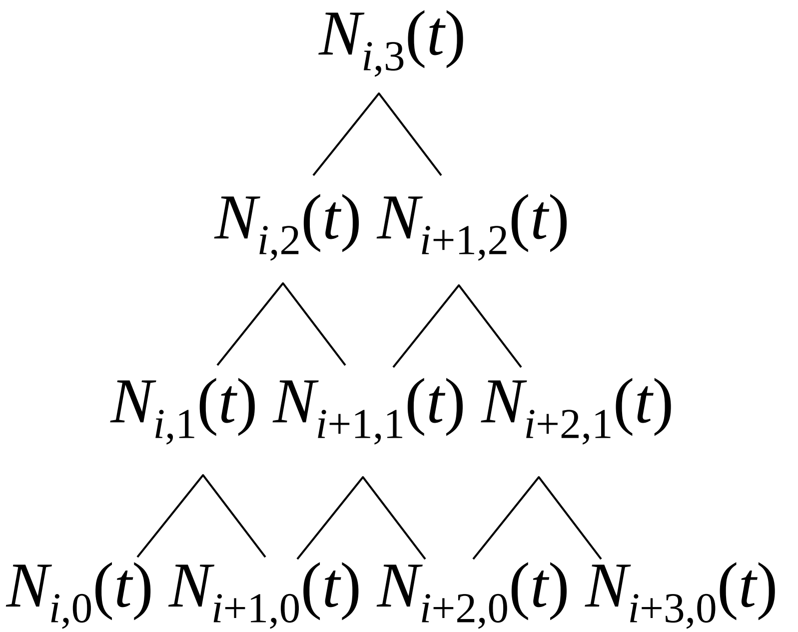
N=3 curve segments

d=3 (cubic)

- Higher degree basis can be constructed from lower degree bases

$$N_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t)$$

- $N_{i,0}(t) =$
  - 1 if $t_i \le t < t_{i+1}$
  - 0 otherwise
- Non-uniform B-splines constructed using a systolic array

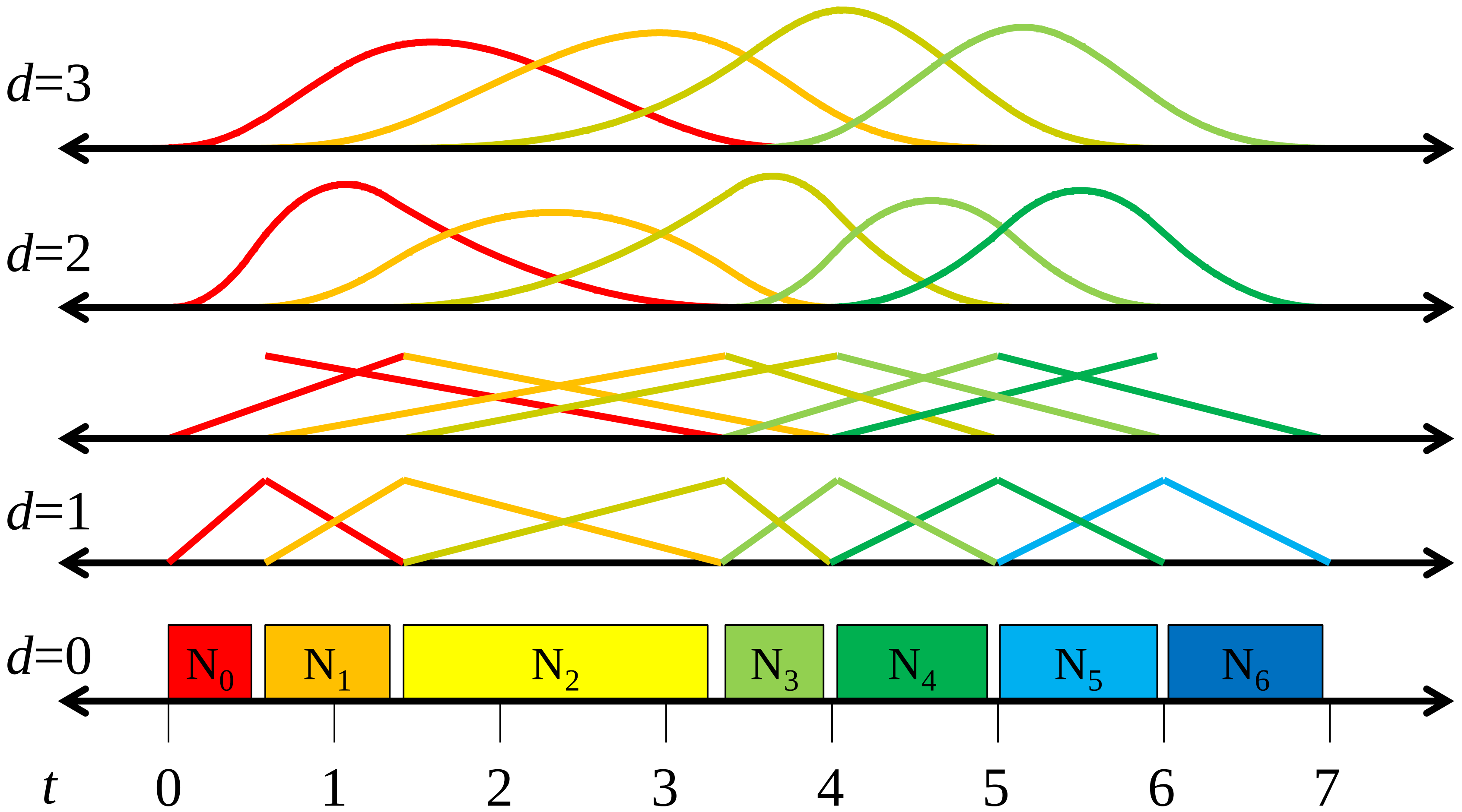


Example

knot vector: [0, .5, 1.3, 3.4, 4, 5.1, 6, 7]

$$N_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t)$$

$N_{i,0}(t) = 1$ when $t_i \le t < t_{i+1}$ else 0



de Boor Algorithm

knot vector: [0 0 0 0 1 4 5 5 5 5]

Cubic (d = 3, k = 4)

- Evaluate at t = 2

$\mathbf{p}_{4,3} = 1/3\ \mathbf{p}_{4,2} + 2/3\ \mathbf{p}_{3,2}$

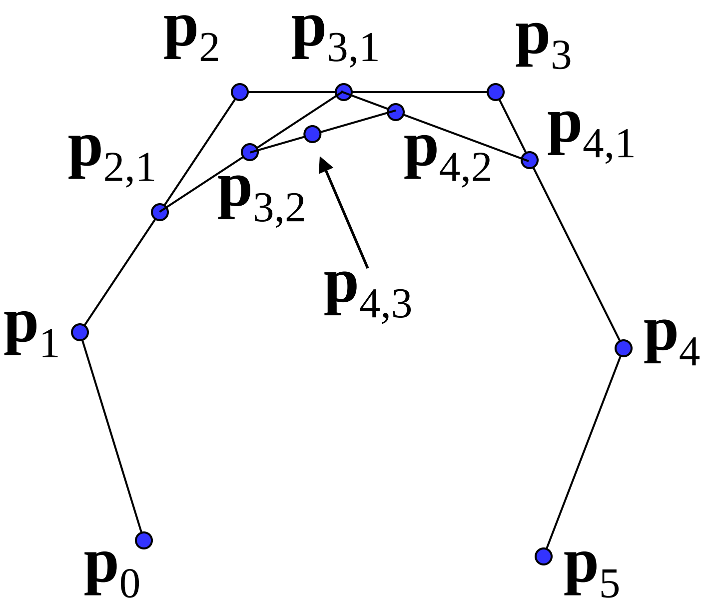
$\mathbf{p}_{4,2} = 1/4\ \mathbf{p}_{4,1} + 3/4\ \mathbf{p}_{3,1}$

$\mathbf{p}_{3,2} = 2/4\ \mathbf{p}_{3,1} + 2/4\ \mathbf{p}_{2,1}$

$\mathbf{p}_{4,1} = 1/4\ \mathbf{p}_{4,0} + 3/4\ \mathbf{p}_{3,0}$

$\mathbf{p}_{3,1} = 2/5\ \mathbf{p}_{3,0} + 3/5\ \mathbf{p}_{2,0}$

$\mathbf{p}_{2,1} = 2/4\ \mathbf{p}_{2,0} + 2/4\ \mathbf{p}_{1,0}$



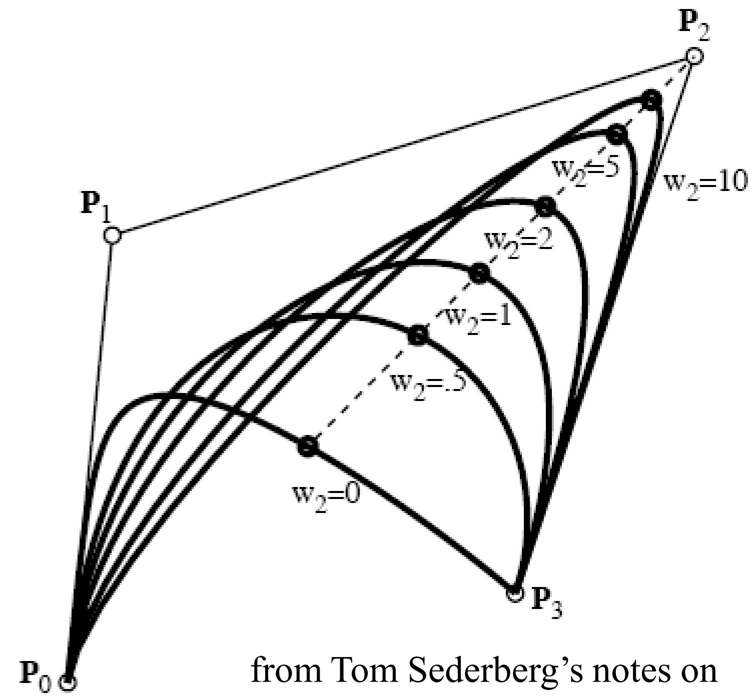
$$\mathbf{p}_{i,l} = \frac{t - t_i}{t_{k+i-l} - t_i}\mathbf{p}_{i,l-1} + \frac{t_{k+i-l} - t}{t_{k+i-l} - t_i}\mathbf{p}_{i-1,l-1}$$

Rational B-Splines

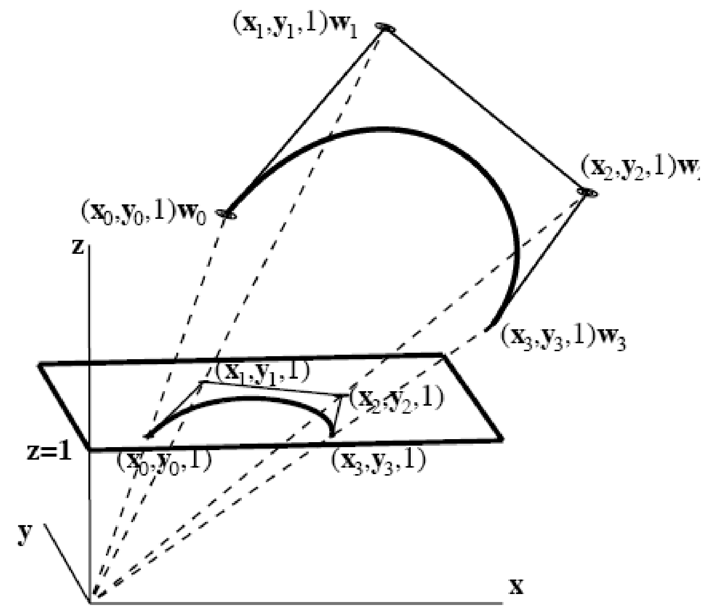
- Quotient of B-splines

  $$\mathbf{p}(t) = (\Sigma\, w_i\, \mathbf{p}_i\, N_i(t))/(\Sigma\, w_i\, N_i(t))$$

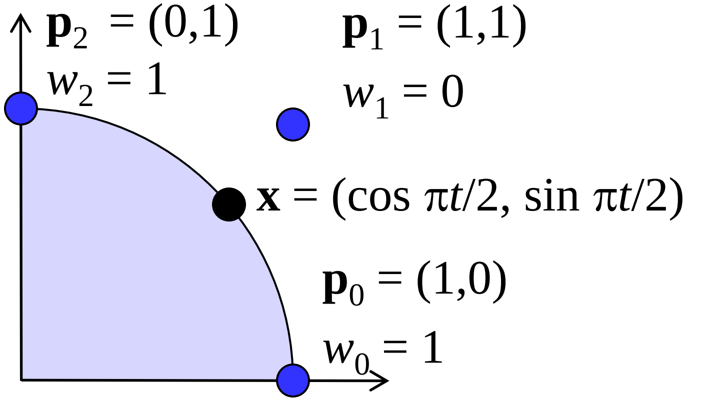
- B-spline in 4-D homogenous space
- Projected back into 3-D via homogenous division
- Weight values affect “tension” near control points
- Weights can also define control points at infinity



from Tom Sederberg’s notes on
Computer Aided Geometric Design



Conic Sections



- Circles, ellipses, arcs
- Only approximated by polynomial parametrics
- Modeled precisely by rational parametrics
- Can be rational Bezier, rational B-spline, etc.