# Splines

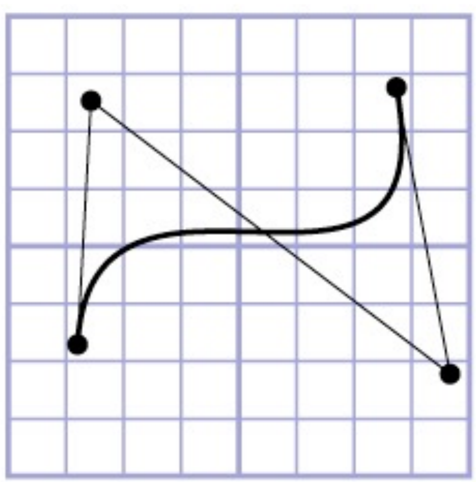
# Administration

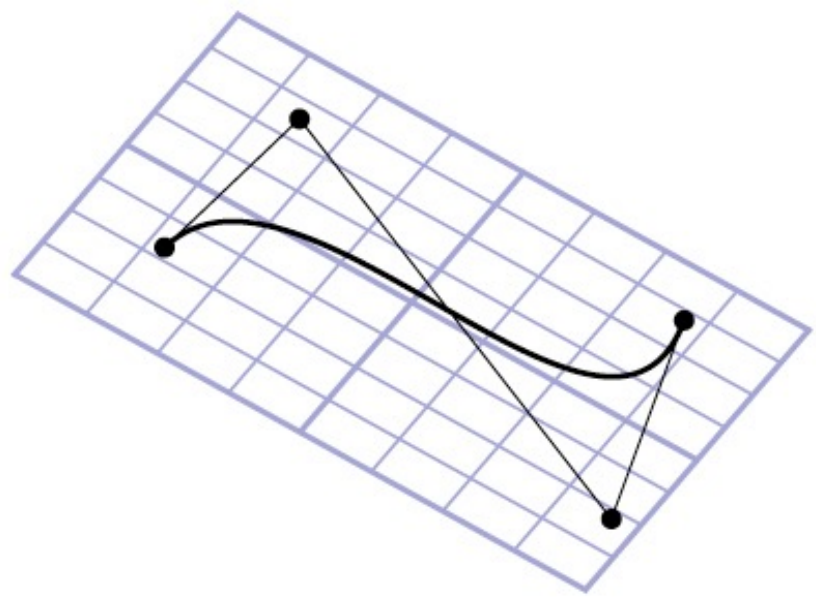
- A4 and PPA2 demos
  - Together on Monday
  - Please sign up

- 4621 lecture today
  - Particle Systems

# Affine invariance

- Transforming the control points is the same as transforming the curve
  - – true for all commonly used splines
  - – extremely convenient in practice…
- Basis functions associated with points should always sum to 1



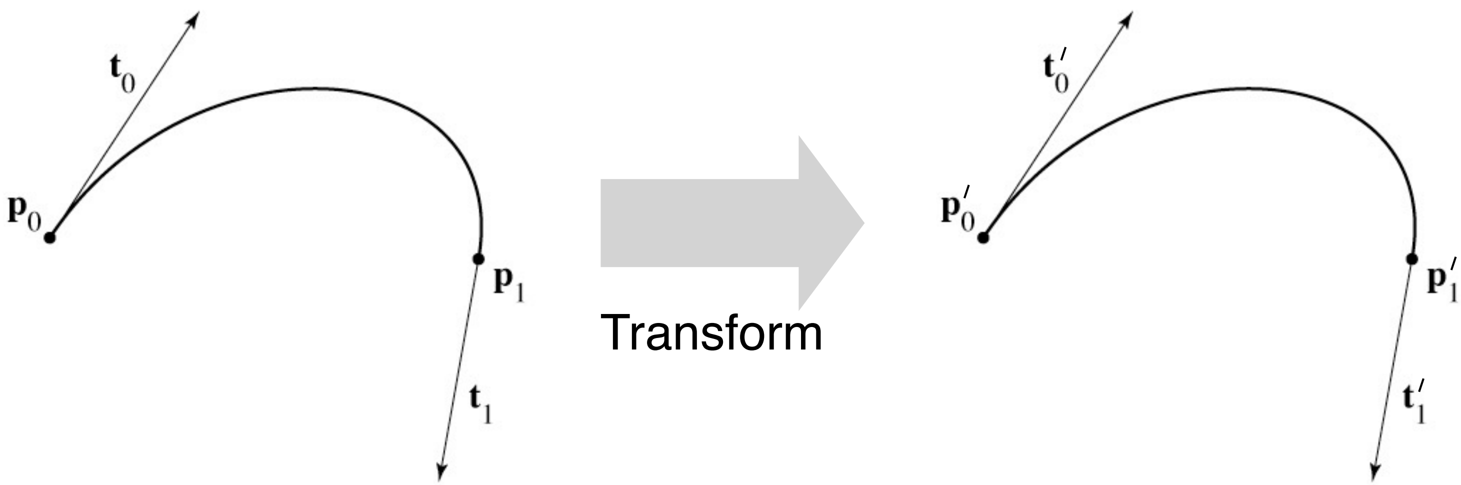




# Affine invariance

- Basis functions associated with points should always sum to 1




$$\mathbf{p}(t) = b_0\mathbf{p}_0 + b_1\mathbf{p}_1 + b_2\mathbf{v}_0 + b_3\mathbf{v}_1$$

Transformed curve $= \mathbf{p}(t) + \mathbf{u}$

# Bézier matrix

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

– note that these are the Bernstein polynomials

$$b_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k}$$

and that defines Bézier curves for any degree

# Cubic B-spline matrix

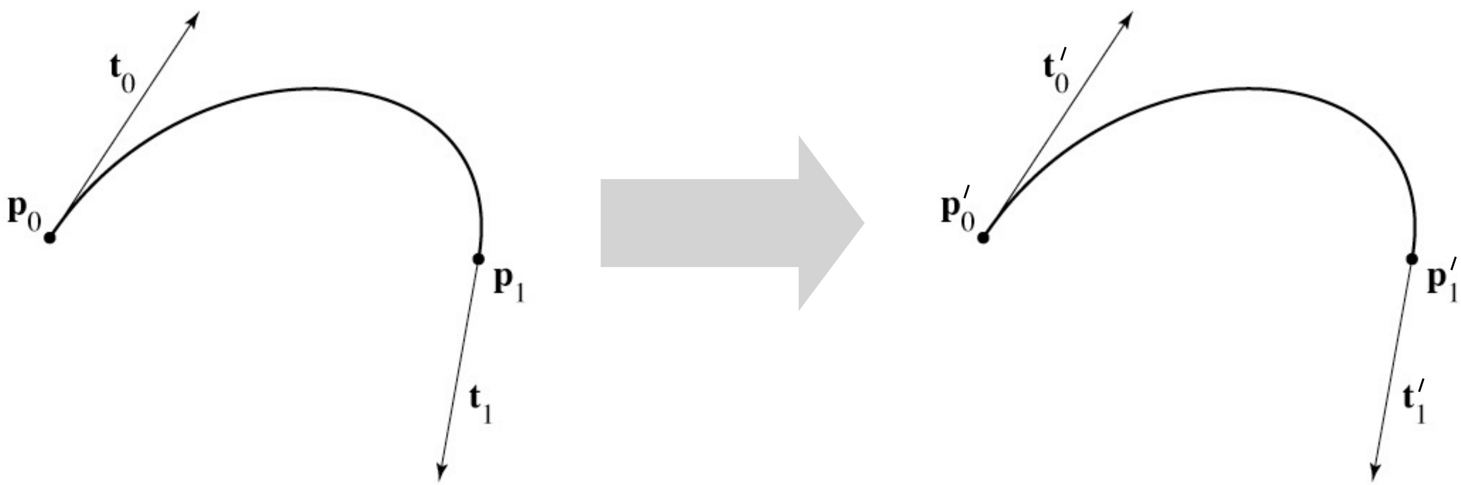
$$\mathbf{f}_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}$$



# Affine invariance

- Basis functions associated with points should always sum to 1




$$\begin{aligned}
\mathbf{p}(t) &= b_0\mathbf{p}_0 + b_1\mathbf{p}_1 + b_2\mathbf{v}_0 + b_3\mathbf{v}_1 \\
\mathbf{p}'(t) &= b_0(\mathbf{p}_0 + \mathbf{u}) + b_1(\mathbf{p}_1 + \mathbf{u}) + b_2\mathbf{v}_0 + b_3\mathbf{v}_1 \\
&= b_0\mathbf{p}_0 + b_1\mathbf{p}_1 + b_2\mathbf{v}_0 + b_3\mathbf{v}_1 + (b_0 + b_1)\mathbf{u} \\
&= \mathbf{p}(t) + \mathbf{u}
\end{aligned}$$



# Hermite splines

- Matrix form is much simpler

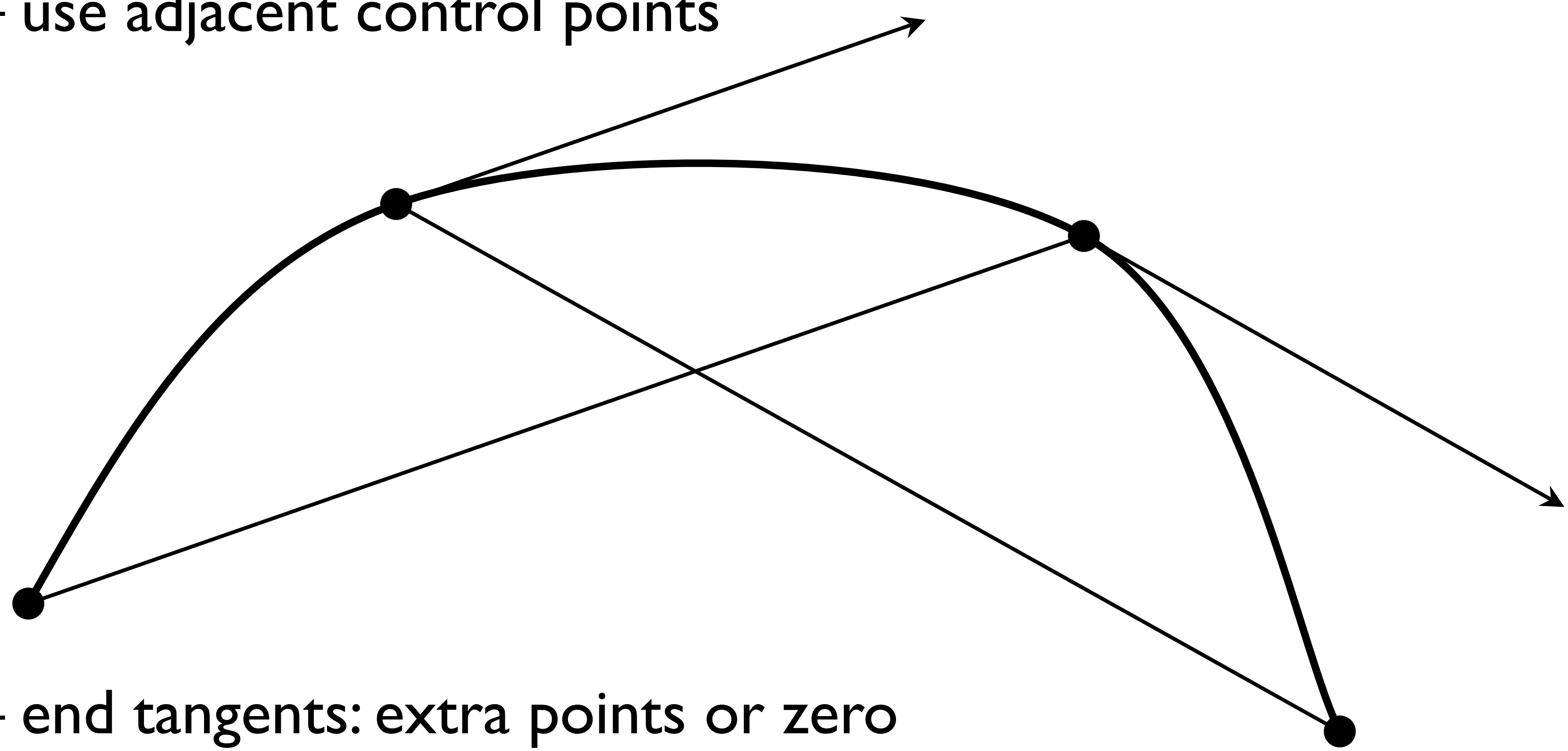
$$\mathbf{f}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix}$$

– coefficients = rows

– basis functions = columns

# Hermite to Catmull-Rom

- Have not yet seen any interpolating splines
- Would like to define tangents automatically
  – use adjacent control points



  – end tangents: extra points or zero

# Hermite to Catmull-Rom

- Tangents are $(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) / 2$
  - scaling based on same argument about collinear case

$$\mathbf{p}_0 = \mathbf{q}_k$$
$$\mathbf{p}_1 = \mathbf{q}_{k+1}$$
$$\mathbf{v}_0 = 0.5(\mathbf{q}_{k+1} - \mathbf{q}_{k-1})$$
$$\mathbf{v}_1 = 0.5(\mathbf{q}_{k+2} - \mathbf{q}_k)$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -.5 & 0 & .5 & 0 \\ 0 & -.5 & 0 & .5 \end{bmatrix} \begin{bmatrix} \mathbf{q}_{k-1} \\ \mathbf{q}_k \\ \mathbf{q}_{k+1} \\ \mathbf{q}_{k+2} \end{bmatrix}$$

# Hermite splines

- Matrix form is much simpler

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

  – coefficients = rows

  – basis functions = columns



# Hermite to Catmull-Rom

- Tangents are $(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) / 2$
  – scaling based on same argument about collinear case

$$\mathbf{p}_0 = \mathbf{q}_k$$
$$\mathbf{p}_1 = \mathbf{q}_{k+1}$$
$$\mathbf{v}_0 = 0.5(\mathbf{q}_{k+1} - \mathbf{q}_{k-1})$$
$$\mathbf{v}_1 = 0.5(\mathbf{q}_{k+2} - \mathbf{q}_k)$$

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -.5 & 0 & .5 & 0 \\ 0 & -.5 & 0 & .5 \end{bmatrix} \begin{bmatrix} \mathbf{q}_{k-1} \\ \mathbf{q}_k \\ \mathbf{q}_{k+1} \\ \mathbf{q}_{k+2} \end{bmatrix}$$

# Hermite to Catmull-Rom

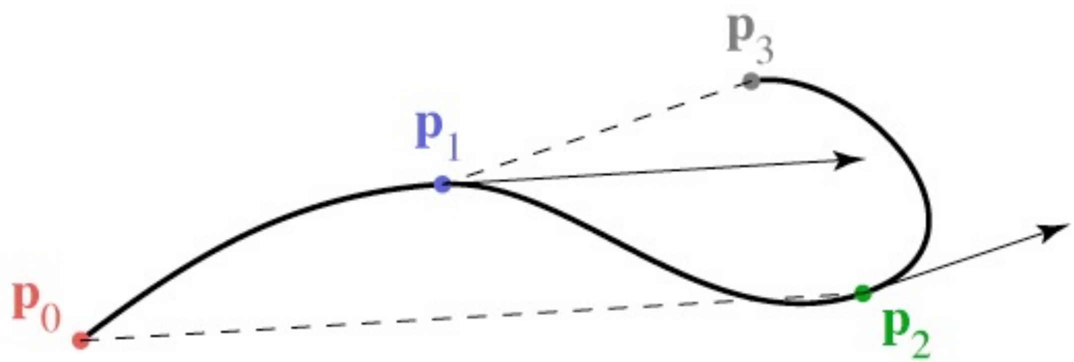
- Tangents are $(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) / 2$
  - scaling based on same argument about collinear case

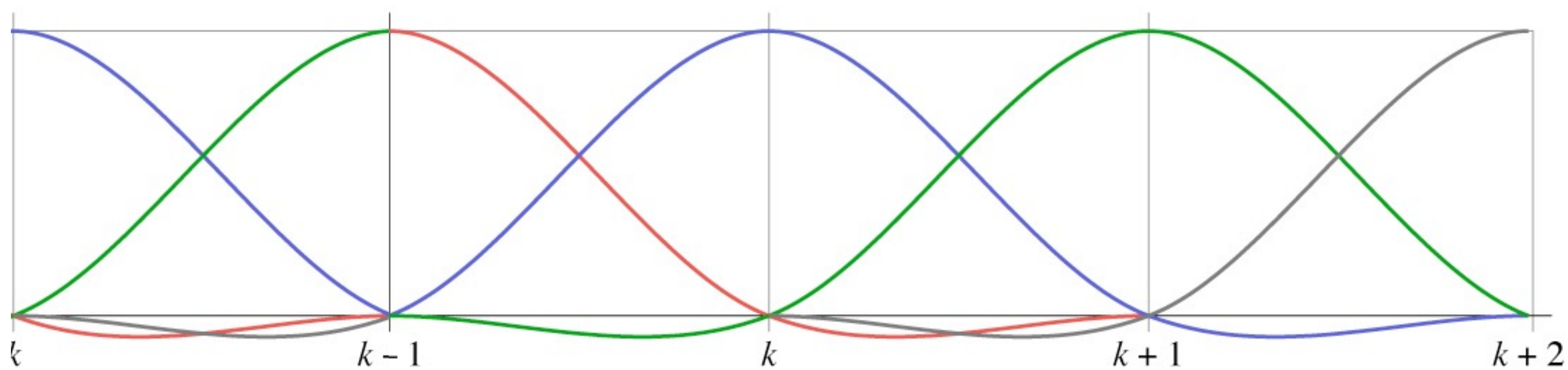
$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} -.5 & 1.5 & -1.5 & .5 \\ 1 & -2.5 & 2 & -.5 \\ -.5 & 0 & .5 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}_{k-1} \\ \mathbf{q}_k \\ \mathbf{q}_{k+1} \\ \mathbf{q}_{k+2} \end{bmatrix}$$



# Catmull-Rom basis





# Catmull-Rom splines

- Our first example of an interpolating spline
- Like Bézier, equivalent to Hermite

- First example of a spline based on just a control point sequence
- Does not have convex hull property

# Converting spline representations

- All the splines we have seen are equivalent
  - all represented by geometry matrices

$$\mathbf{p}_S(t) = T(t) M_S P_S$$

    - where *S* represents the type of spline
  - therefore the control points may be transformed from one type to another using matrix multiplication

$$P_1 = M_1^{-1} M_2 P_2$$

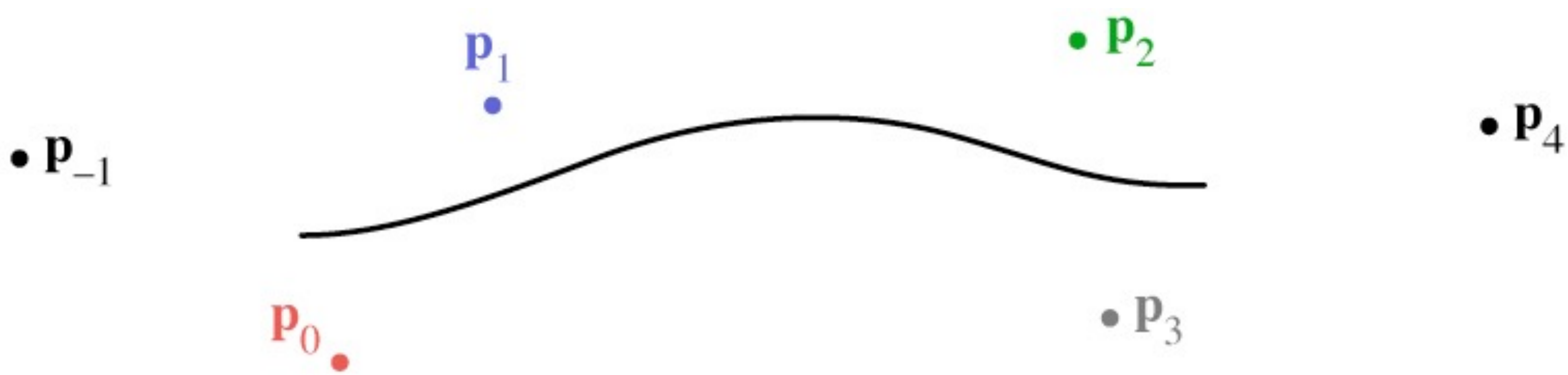
$$\begin{aligned}\mathbf{p}_1(t) &= T(t) M_1 (M_1^{-1} M_2 P_2) \\ &= T(t) M_2 P_2 = \mathbf{p}_2(t)\end{aligned}$$

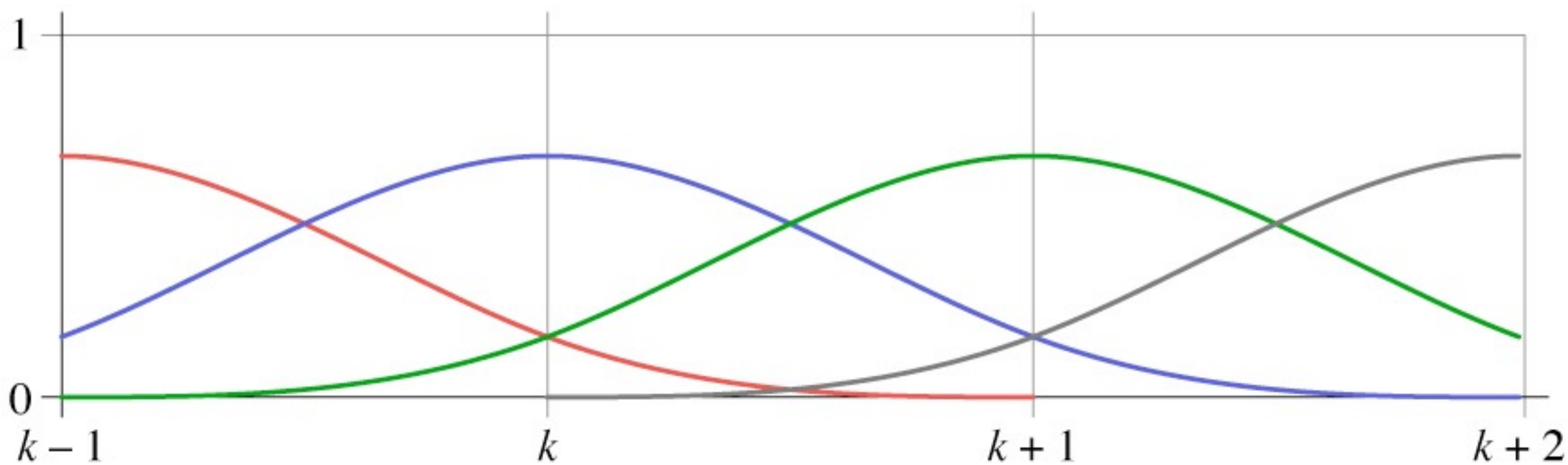

 

# B-splines

- We may want more continuity than $C^1$
- We may not need an interpolating spline
- B-splines are a clean, flexible way of making long splines with arbitrary order of continuity

# Cubic B-spline basis





# Deriving the B-Spline

- Approached from a different tack than Hermite-style constraints
  - Want a cubic spline; therefore 4 active control points
  - Want $C^2$ continuity
  - Turns out that is enough to determine everything



# Efficient construction of any B-spline

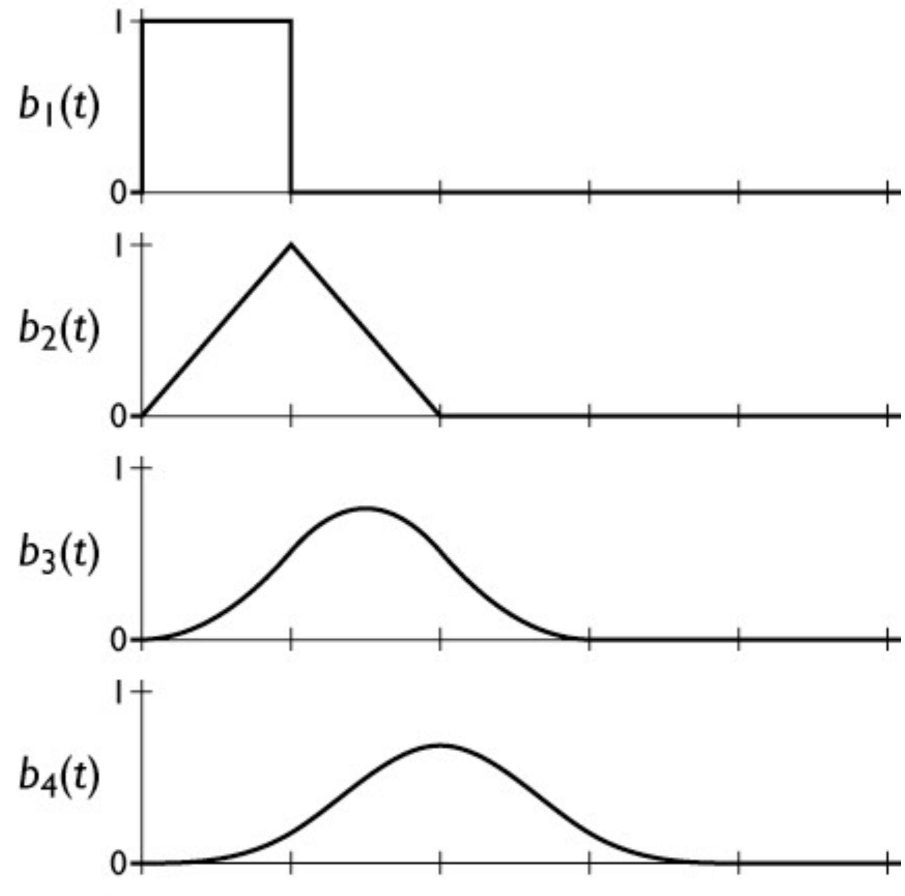
- B-splines defined for all orders
  - order *d*: degree *d* – 1
  - order *d*: *d* points contribute to value
- One definition: Cox-deBoor recurrence

$$b_1 = \begin{cases} 1 & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_d = \frac{t}{d-1} b_{d-1}(t) + \frac{d-t}{d-1} b_{d-1}(t-1)$$

# B-spline construction, alternate view

- Recurrence
  - – ramp up/down
- Convolution
  - – smoothing of basis fn
  - – smoothing of curve



# Cubic B-spline matrix

$$\mathbf{f}_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}$$



# Bézier matrix

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

– note that these are the Bernstein polynomials

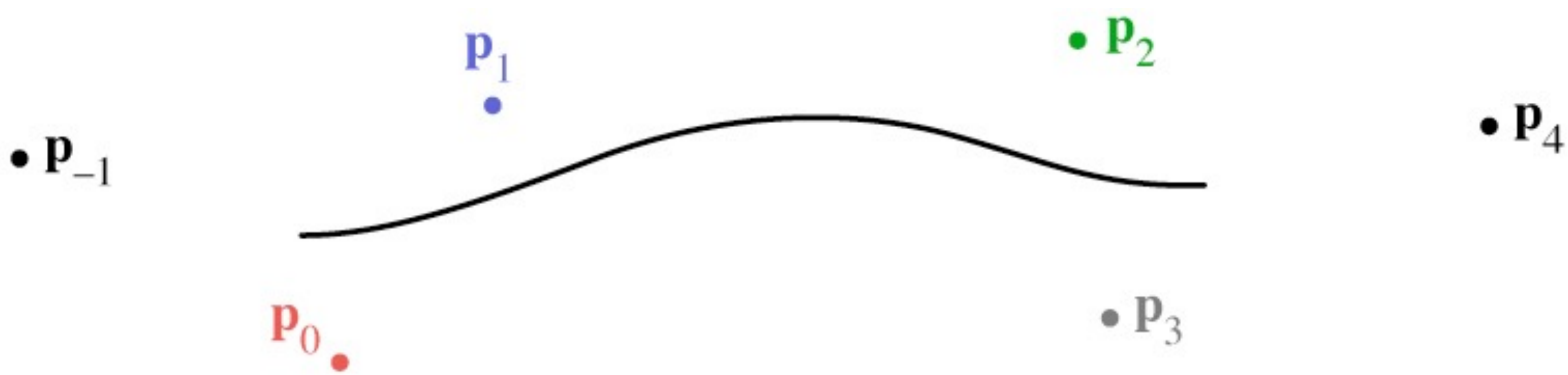
$$b_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k}$$

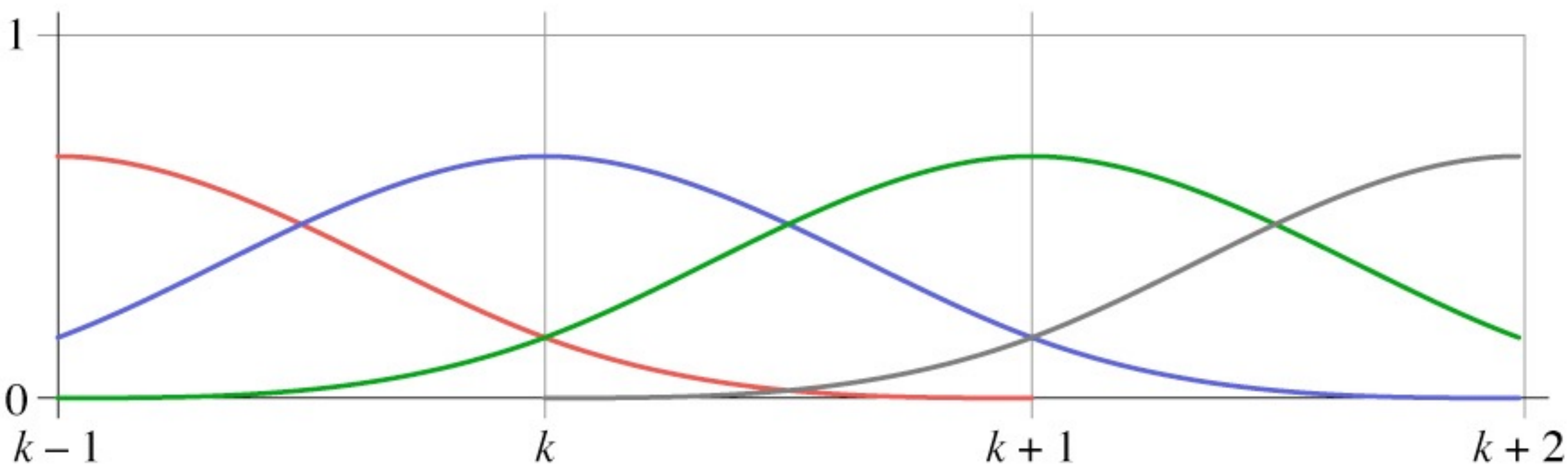
and that defines Bézier curves for any degree



# Cubic B-spline basis



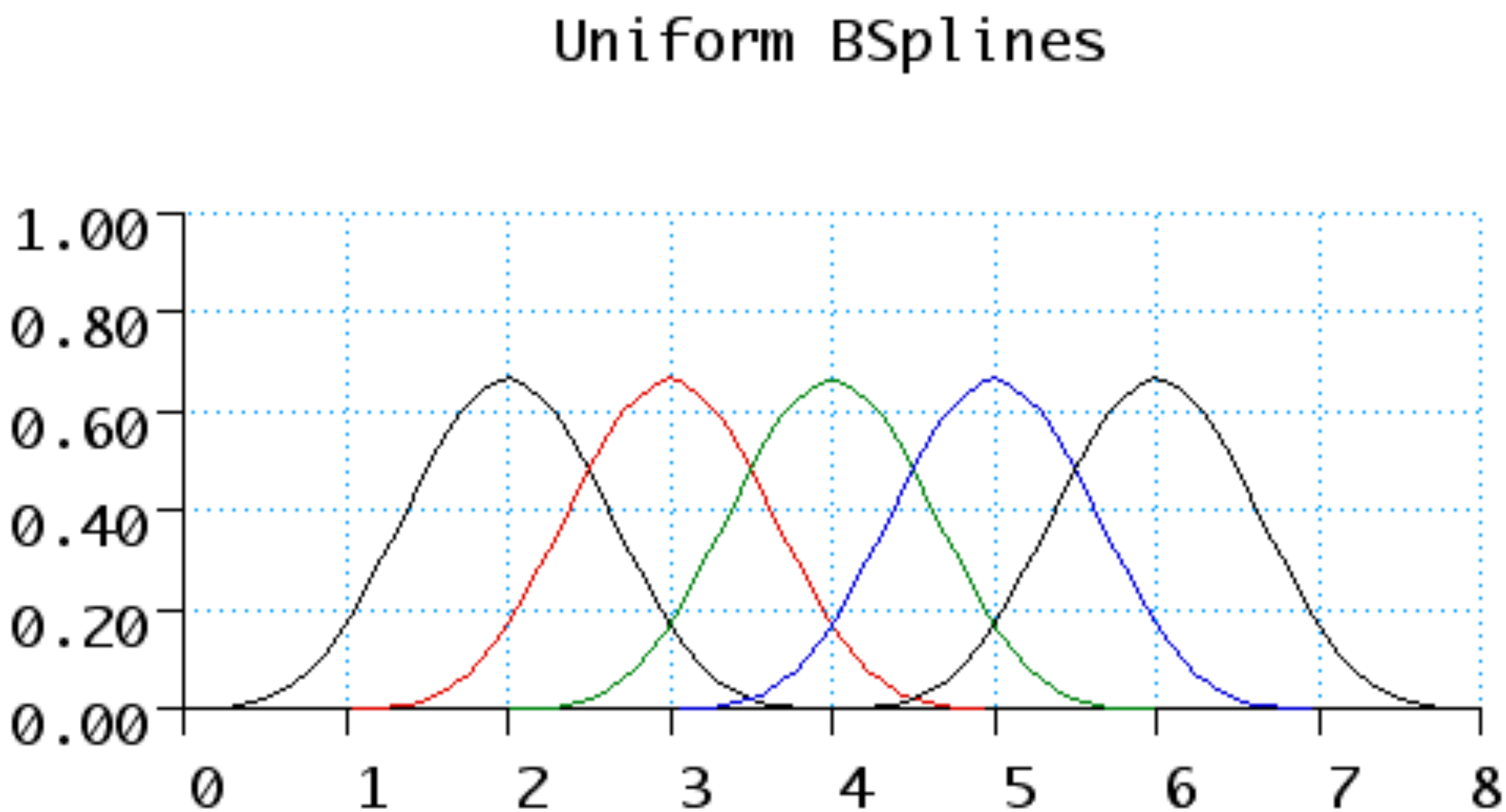





# B-spline

- All points are same, no special points
- Basis functions are the same over many segments



# Other types of B-splines

- Nonuniform B-splines
  – discontinuities not evenly spaced
  – allows control over continuity or interpolation at certain points
  – e.g. interpolate endpoints (commonly used case)
- Nonuniform Rational B-splines (NURBS)
  – ratios of nonuniform B-splines: $x(t)$ / $w(t)$; $y(t)$ / $w(t)$
  – key properties:
    - invariance under perspective as well as affine
    - ability to represent conic sections exactly