Spline Representation

Xin (Shane) Li

Piecewise linear approximation

- Previous: polygonal representation (meshes) and polylines are first-degree, piecewise linear approximations to surfaces and curves
- When the object is not piecewise linear
  - To improve its approximation accuracy
    → more sample points
      → large number of coordinates to be created and stored
- Interactive manipulation is tedious
- Need a more compact and more manipulable representation
  - To use functions that are of a higher degree

Three general approaches

1) Explicit functions:

→ $y=f(x)$, $z=g(x)$

- Can't get multiple values of y for a single x → closed curves must be represented by multiple segments
- Not rotationally invariant
- Curves with vertical tangents is difficult (infinite slop)

2) Implicit functions:

→ $f(x,y,z)=0$

- A simple equation is usually not enough, need several for constraints
  - e.g. : a half circle
- Not easy to merge several simple sub-parts
  - e.g. : when merge two curve segments, difficult to determine whether their tangent directions agree

3) Parametric representation:

→ $x=x(t)$, $y=y(t)$, $z=z(t)$

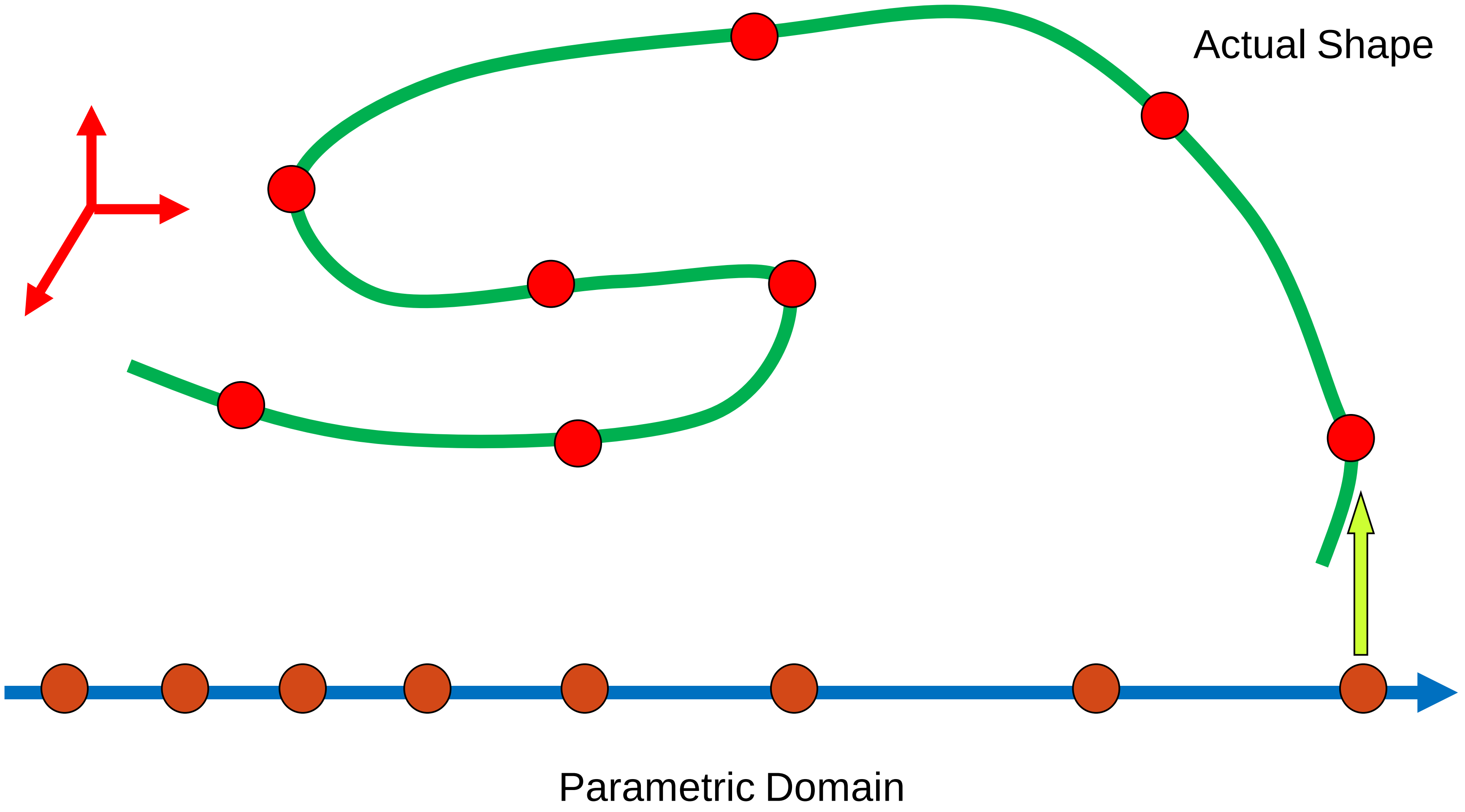
- Overcome above problems
- geometric slopes (may be infinite) → parametric tangent vectors (never infinite)
- Piecewise linear shapes → piecewise polynomial shapes

Spline

- Spline = long flexible strips of metal used by draftspersons to lay out the surfaces of airplanes, cars, and ships
- The metal splines, unless severely stressed, had second-order continuity

R. Bartels, J. Beatty, and B. Barsky, “An Introduction to Splines for Use in Computer Graphics and Geometric Modeling”, Morgan Kaufmann, 1987

Parametric Curve



Parametric Cubic Curves

A curve segment defined by the cubic polynomial Q(t)=[x(t) y(t) z(t)]:

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x,$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y,$$
$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z,$$
$$0 \le t \le 1$$

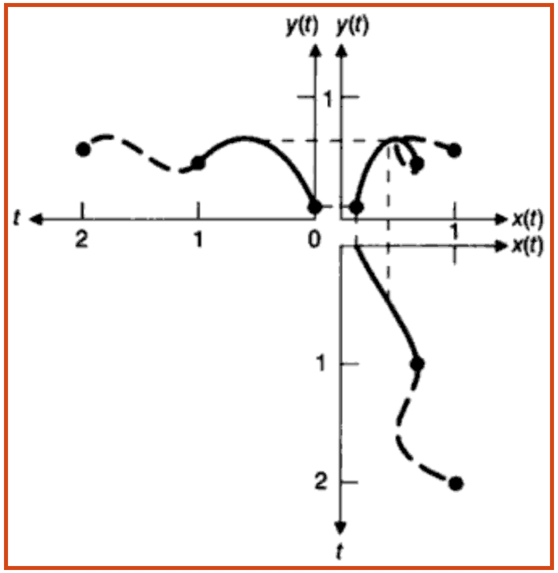
A more compact writing:

$$T = [t^3 \quad t^2 \quad t^1 \quad 1];$$

$$C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix};$$

$$Q(t) = [x(t) \quad y(t) \quad z(t)] = T \cdot C$$

An example of two joined parametric cubic curve segments and their polynomials



Continuity

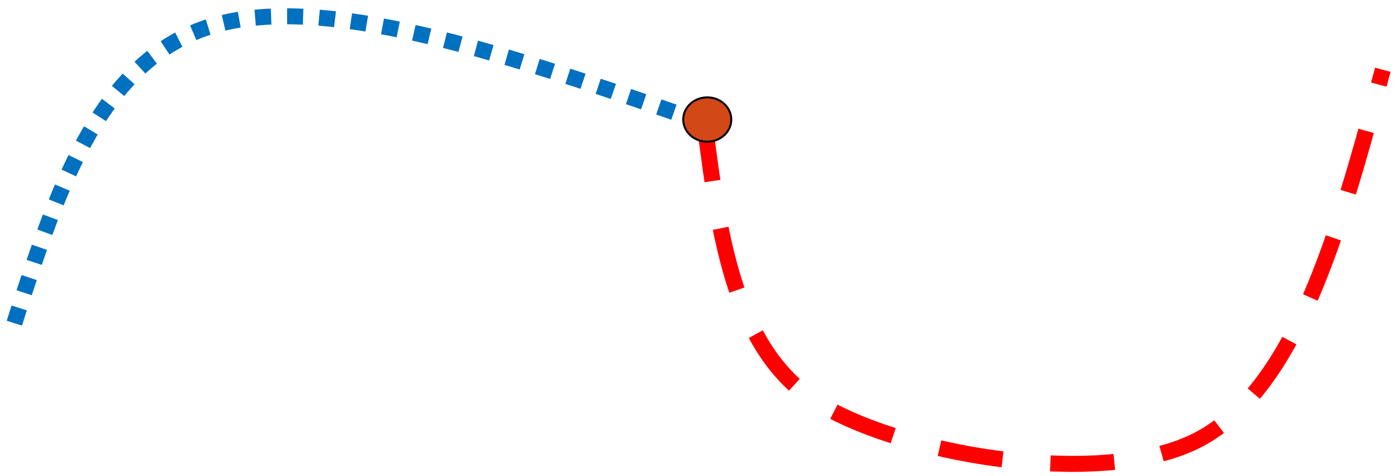
- One of the fundamental concepts
- Commonly used cases:

$$C^0, C^1, C^2$$

- Consider two curves: a(u) and b(u) (u is in [0,1])

Positional Continuity

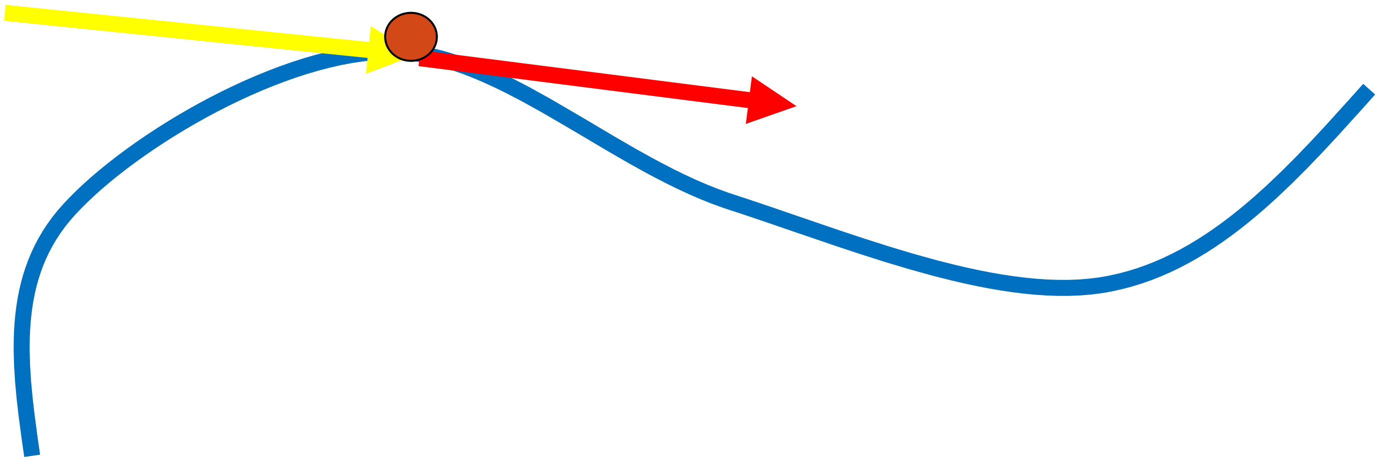
$$\mathbf{a}(1) = \mathbf{b}(0)$$



Derivative Continuity

$$\mathbf{a}(1) = \mathbf{b}(0)$$

$$\mathbf{a}'(1) = \mathbf{b}'(0)$$

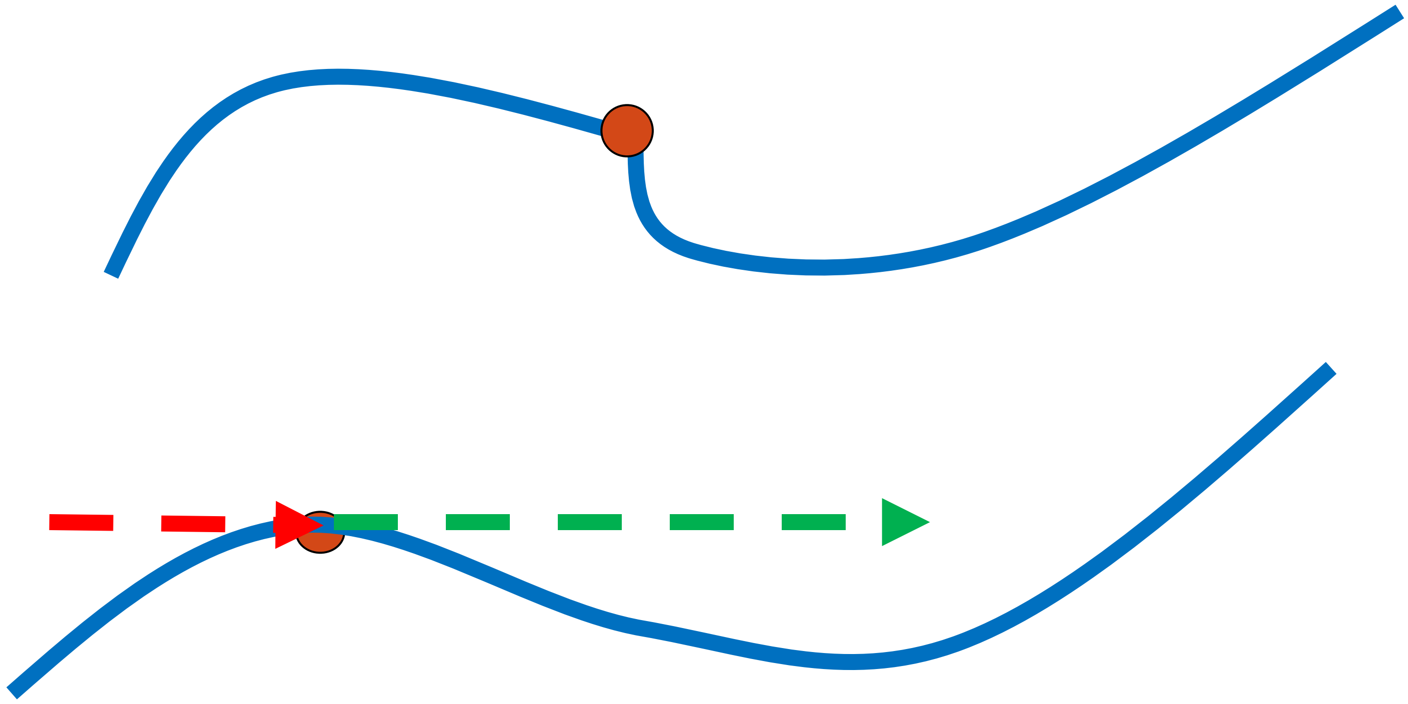


Parametric and General Continuity

- C^n continuity: derivatives (up to n-th) are the same at the joining point
$$\mathbf{a}^{(i)}(1) = \mathbf{b}^{(i)}(0)$$
$$i = 0,1,2,\ldots,n$$
- C^n → parametric continuity
  - Depending on parameterization, not just the geometry
  - Same geometry may have different parametric representations (re-parameterization)
- Another type of continuity: geometric continuity, denoted as G^n

Geometric Continuity

- G^0 and G^1



Geometric Continuity

- Only depend on the geometry, not the parameterization
- G^0 : the same joint
- G^1: two curve tangents at the joint align, but may (or may not) have the same magnitude
- G^n : → C^n after the reparameterization
- Which condition is stronger?

  - geometric continuity is a relaxed form of parametric continuity

Defining and Merging Curve Segments

- A curve segment is defined by constraints on endpoints, and tangent vectors (or higher degree derivatives)
- e.g. : on each dimension, a cubic polynomial curve has four coefficients ← four constraints will be needed to solve for the unknowns

→ Most commonly used in computer graphics
  - → Lower-degree polynomials give too little flexibility in controlling the shape of the curve (on position + tangent interpolation)
  - → Higher-degree polynomials can introduce unwanted wiggles and also require more computation

- Three common types of curve segments:
  - **Hermite** : defined by 2 endpoints + 2 endpoint tangent vectors
  - **Bezier** : defined by 2 endpoints and 2 other points (that control the endpoint tangent vectors)
  - **Several kinds of splines**: defined by 4 control points

How coefficients depend on constraints

- Given a cubic curve segment, only 12 coefficients to determine:
- On x(t) , only 4 , uniquely determined by 4 constraints
- Suppose we want to put constraints on positional and normal values x(0), x(1), x'(0), and x'(1)
- We can rewrite the representation

$$x(t) = [t^3 \quad t^2 \quad t \quad 1]\begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} = [t^3 \quad t^2 \quad t \quad 1]\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}\begin{bmatrix} x(0) \\ x(1) \\ x'(0) \\ x'(1) \end{bmatrix} = T \cdot M \cdot G_x$$

- It becomes a weighted sum of constraints
- A generalization of straight-line approximation

How coefficients depend on constraints

$$x(t) = [t^3 \quad t^2 \quad t \quad 1]\begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} = [t^3 \quad t^2 \quad t \quad 1]\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}\begin{bmatrix} x(0) \\ x(1) \\ x'(0) \\ x'(1) \end{bmatrix} = T \cdot M \cdot G_x$$

- If we know the matrix M, then given a set of new constraints, we know the curve immediately

$$x(t) = T \cdot \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x(0) \\ x(1) \\ x'(0) \\ x'(1) \end{bmatrix}$$

$$y(t) = T \cdot M^H \cdot \begin{bmatrix} y(0) & y(1) & y'(0) & y'(1) \end{bmatrix}^T$$

$$z(t) = T \cdot M^H \cdot \begin{bmatrix} z(0) & z(1) & z'(0) & z'(1) \end{bmatrix}^T$$

How coefficients depend on constraints

- Rewrite:

$$T = [t^3 \quad t^2 \quad t^1 \quad 1]; C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix};$$

$$Q(t) = [x(t) \quad y(t) \quad z(t)] = T \cdot C$$

$$= T \cdot M \cdot G = [t^3 \quad t^2 \quad t^1 \quad 1] \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}$$

Basis matrix

Geometric vectors (constraints, e.g. end points, tangent)

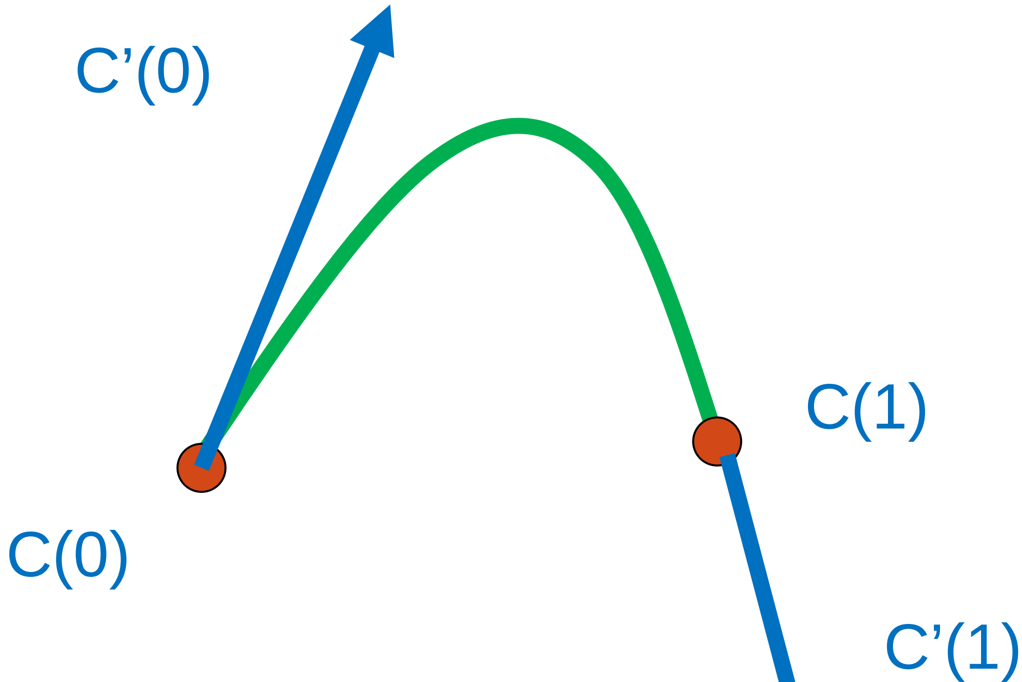
- On x(t):

$$x(t) = T \cdot M \cdot G_x = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} g_{x1} \\ g_{x2} \\ g_{x3} \\ g_{x4} \end{bmatrix}$$

→a curve is a weighted sum of a column (x, or y, or z) of elements of the geometry matrix

- A generalization of straight-line approximation

Cubic Hermite Curve



Cubic Hermite Curve

- Hermite curve

$$\mathbf{c}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}$$

- On each axis direction
  - 4 constraints = 2 end-points + 2 tangents at end-points

- Therefore:

$$\begin{bmatrix} x(0) \\ x(1) \\ x'(0) \\ x'(1) \end{bmatrix} = G_x^H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \cdot M^H \cdot G_x^H$$

M^H= its inverse:

$$x(t) = T \cdot \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x'(0) \\ x'(1) \end{bmatrix}$$

$$y(t) = T \cdot M^H \cdot \begin{bmatrix} y(0) & y(1) & y'(0) & y'(1) \end{bmatrix}^T$$

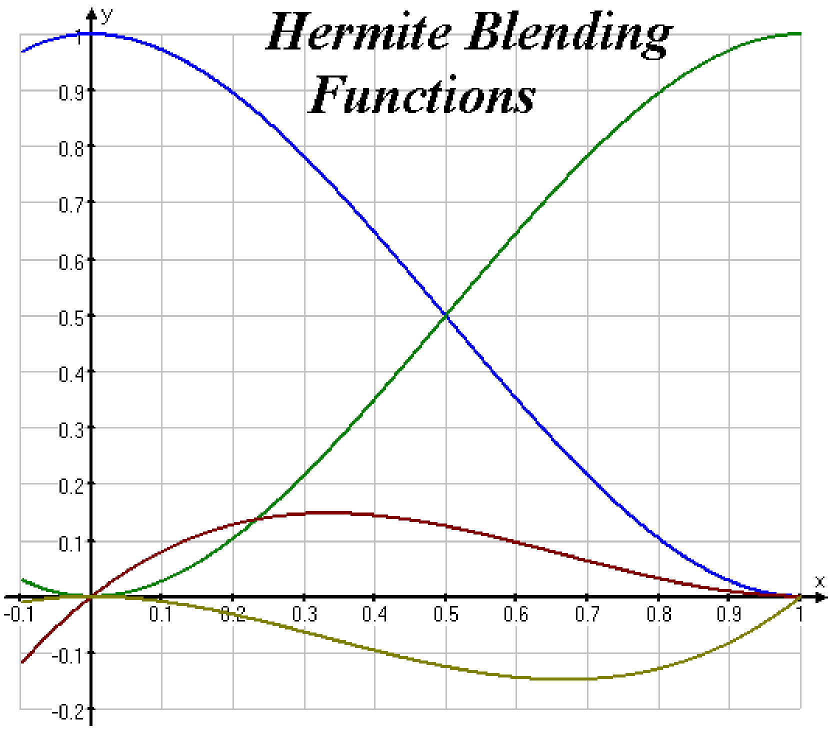
$$z(t) = T \cdot M^H \cdot \begin{bmatrix} z(0) & z(1) & z'(0) & z'(1) \end{bmatrix}^T$$

Hermite Curve

$$Q(t) = T \cdot M^H \cdot G^H = B^H \cdot G^H$$

- Basis functions

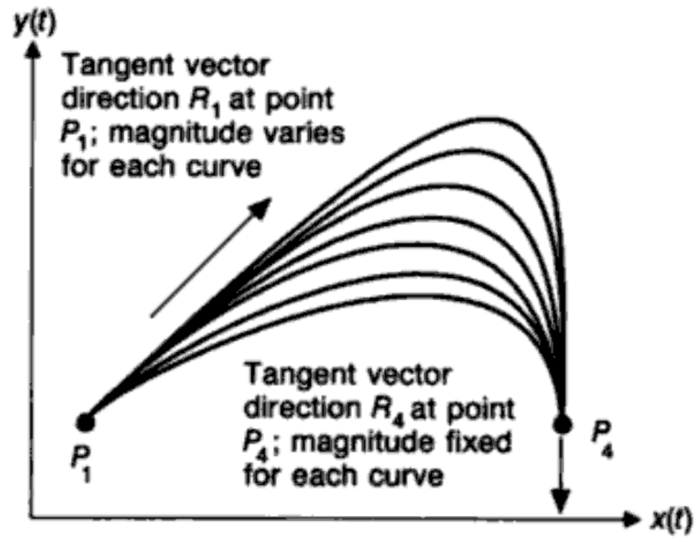
$$f_1(t) = 2t^3 - 3t^2 + 1$$
$$f_2(t) = -2t^3 + 3t^2$$
$$f_3(t) = t^3 - 2t^2 + t$$
$$f_4(t) = t^3 - t^2$$



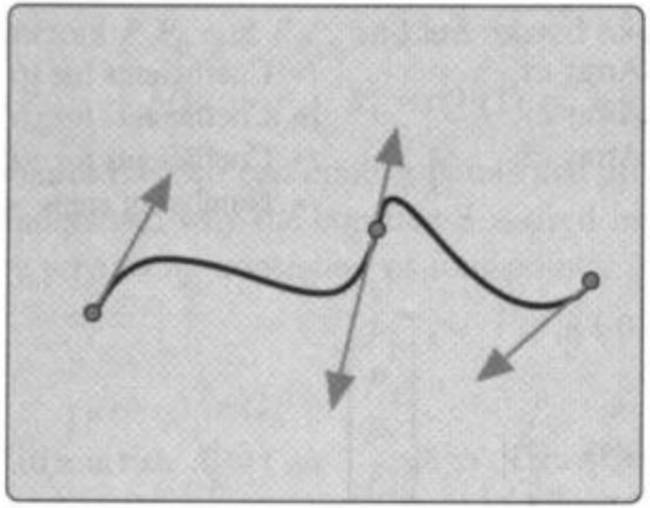
$$\mathbf{c}(t) = \mathbf{c}(0) f_1(t) + \mathbf{c}(1) f_2(t)$$
$$+ \mathbf{c}'(0) f_3(t) + \mathbf{c}'(1) f_4(t)$$

Series of Hermite Curves

- Tangent vector direction and the curve shape
  - increasing magnitude of R_1 → higher curves (right fig.)



- Continuity between two connecting Hermite cubic curves:
  - Same end-points
  - Same tangent vectors



High-Degree polynomials

- More degrees of freedom
- Easy to formulate
- Infinitely differentiable
- Drawbacks:
  - High-order
  - Global control
  - Expensive to compute, complex
  - Undulation

Piecewise Polynomials

- Piecewise --- different polynomials for different parts of the curve
- Advantages --- flexible, low-degree
- Disadvantages --- how to ensure smoothness at the joints (continuity)

Piecewise Hermite Curves

- How to build an interactive system to satisfy various constraints
- C0 continuity

$$\mathbf{a}(1) = \mathbf{b}(0)$$

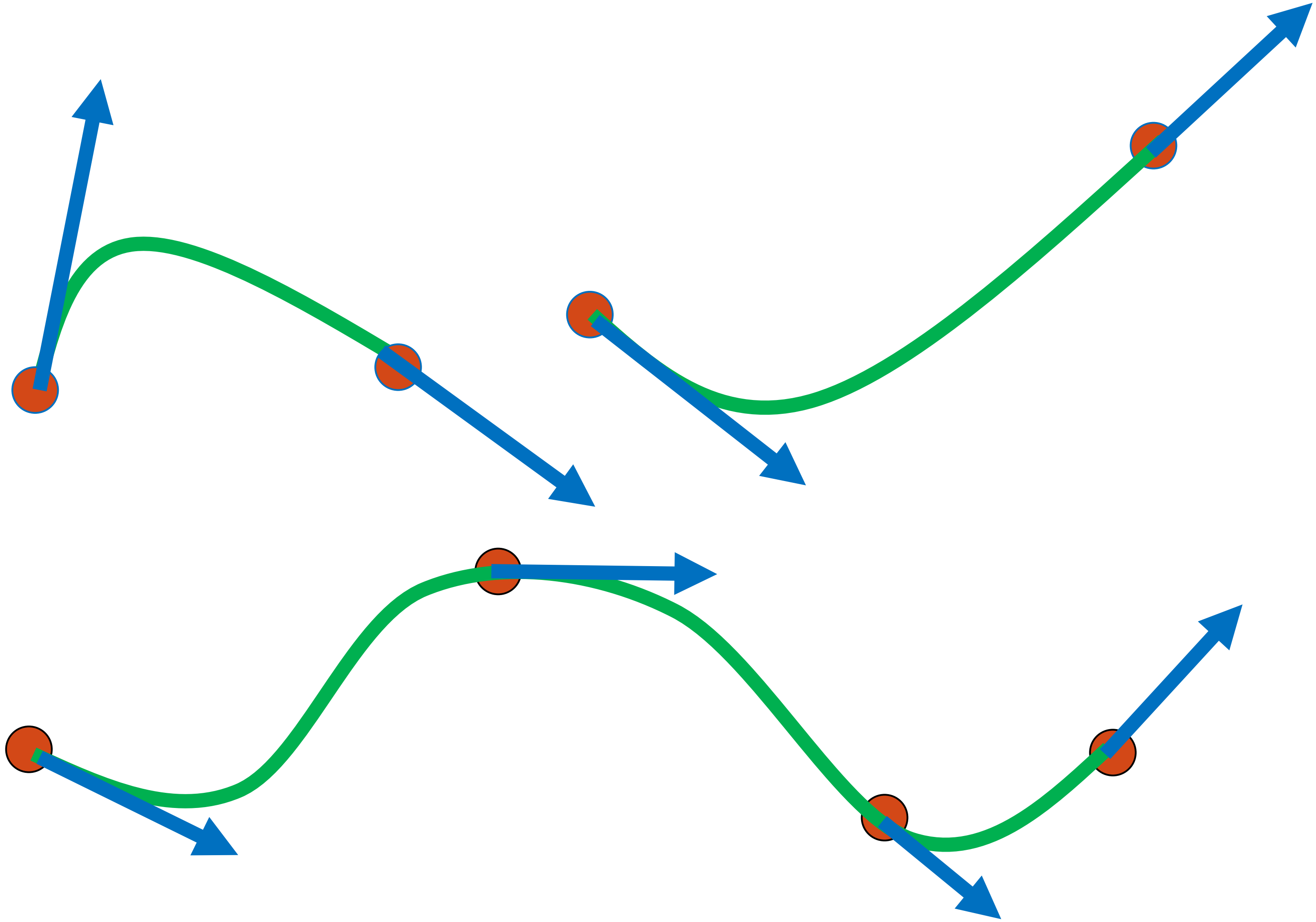
- C1 continuity

$$\mathbf{a}(1) = \mathbf{b}(0)$$
$$\mathbf{a}'(1) = \mathbf{b}'(0)$$

- G1 continuity

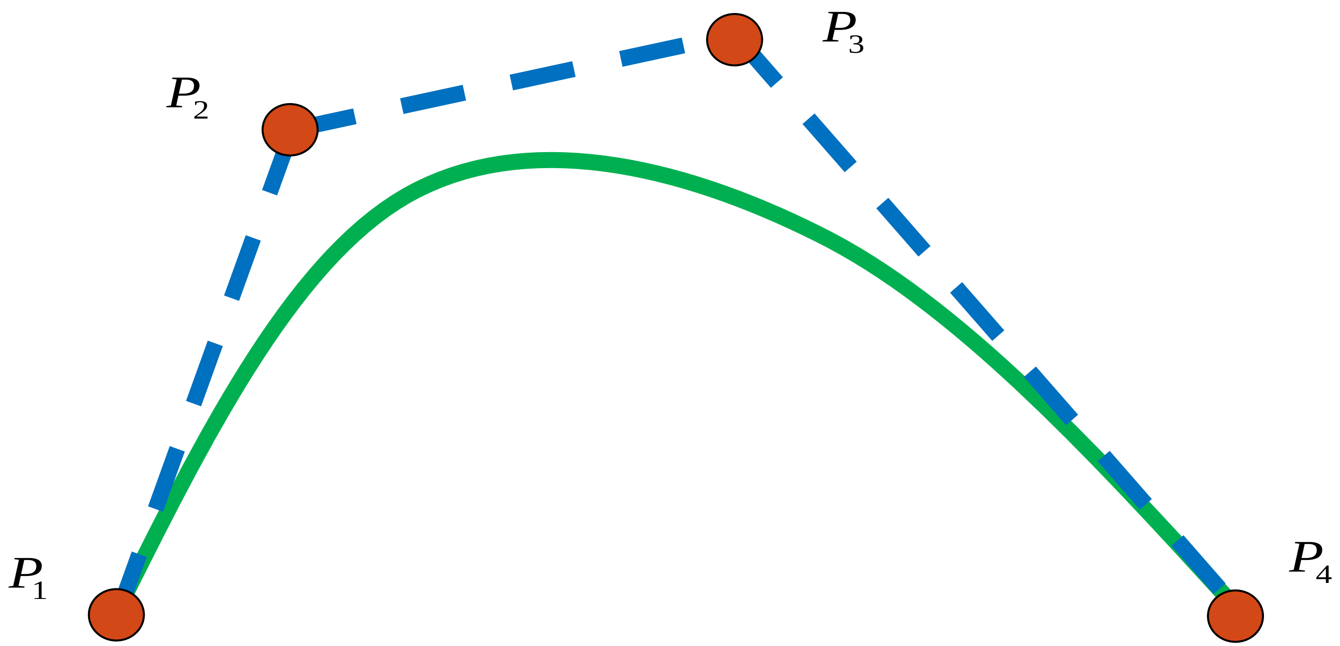
$$\mathbf{a}(1) = \mathbf{b}(0)$$
$$\mathbf{a}'(1) = \alpha\mathbf{b}'(0)$$

Piecewise Hermite Curves



Bezier Curve

Interpolate the two end control points, and approximates the other two points:



$$Q'(0) = 3(P_2 - P_1); Q'(1) = 3(P_4 - P_3)$$

Basis Matrix for Bezier Curve

- Following the last equation:

$$\begin{bmatrix} Q(0) \\ Q(1) \\ Q'(0) \\ Q'(1) \end{bmatrix} = G_x^H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = M^{HB} \cdot G^B$$

Bezier geometry vector

- Therefore, we derive the Bezier basis matrix from the Hermit form:

$$G^H = M^{HB} \cdot G^B; M^B = M^H \cdot M^{HB};$$

$$Q(t) = T \cdot M^H \cdot G^H = T \cdot M^H (M^{HB} \cdot G^B) = T \cdot M^B \cdot G^B;$$

$$M^B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \Rightarrow T \cdot M^B = \begin{bmatrix} B_0^3(t) = (1-t)^3 \\ B_1^3(t) = 3t(1-t)^2 \\ B_2^3(t) = 3t^2(1-t) \\ B_3^3(t) = t^3 \end{bmatrix}$$

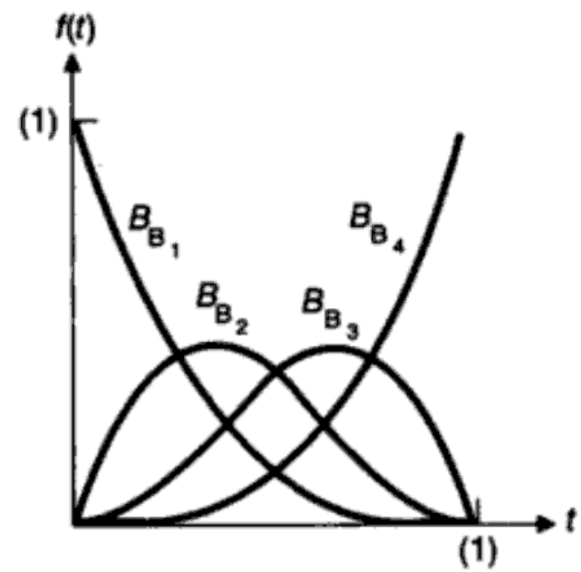
Bernstein Polynomials

- Bezier curve

$$\mathbf{c}(t) = \sum_{i=0}^{3} \mathbf{p}_i B_i^3(t)$$

- Control points and basis functions

$$B_0^3(t) = (1-t)^3$$
$$B_1^3(t) = 3t(1-t)^2$$
$$B_2^3(t) = 3t^2(1-t)$$
$$B_3^3(t) = t^3$$



Review:
An n-degree parametric curve

$$T = [t^n \quad \dots \quad t^1 \quad t^0];$$

$$C = \begin{bmatrix} c_n^x & c_n^y & c_n^z \\ c_{n-1}^x & c_{n-1}^y & c_{n-1}^z \\ \dots & \dots & \dots \\ c_0^x & c_0^y & c_0^z \end{bmatrix};$$

$$Q(t) = [x(t) \quad y(t) \quad z(t)] = T \cdot C = \sum_{i=0}^{n} \vec{c}_i t^i$$

a Degree-3 example:
A Cubic curve segment:

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x,$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y,$$
$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z,$$
$$0 \le t \le 1$$

Given 4 geometric constraint vectors: we can solve all unknown coefficients

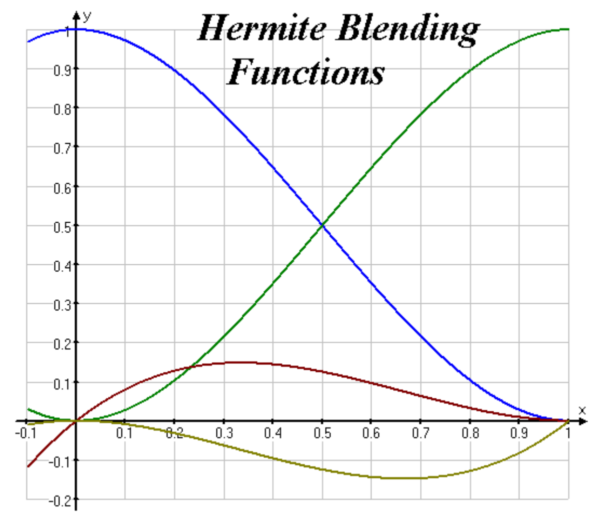
Different schemes: Hermite, Bezier…

- Mathmatically equivalent, one can convert to another
- Allow different constraint vectors

Cubic Hermite & Bezier Curves

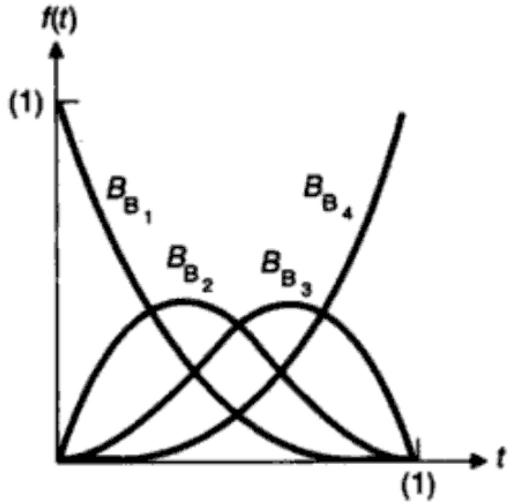
- Hermit Curves:

$$f_1(t) = 2t^3 - 3t^2 + 1$$
$$f_2(t) = -2t^3 + 3t^2$$
$$f_3(t) = t^3 - 2t^2 + t$$
$$f_4(t) = t^3 - t^2$$



- Bezier Curves:

$$B_0^3(t) = (1-t)^3$$
$$B_1^3(t) = 3t(1-t)^2$$
$$B_2^3(t) = 3t^2(1-t)$$
$$B_3^3(t) = t^3$$



Basic Properties of Bezier Cubic Curves

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}; C(t) = \sum_{i=0}^{i=m} B_i^n(t) P_i$$

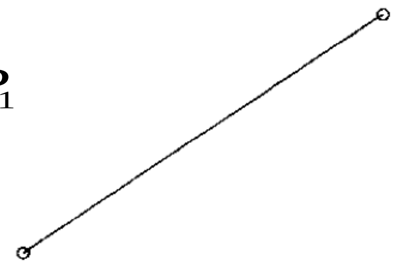
- End-point interpolation: curve passes the first and the last points
- The curve is a linear combination of control points and basis functions
- Basis functions
  - Are Polynomials
  - Partition of unity: Basis functions sum to one
  - Non-negative
- Convex hull (both necessary and sufficient)
- Predictability

Some Bezier curve examples

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}; C(t) = \sum_{i=0}^{i=m} B_i^n(t) P_i$$

- n=1 : linear interpolation

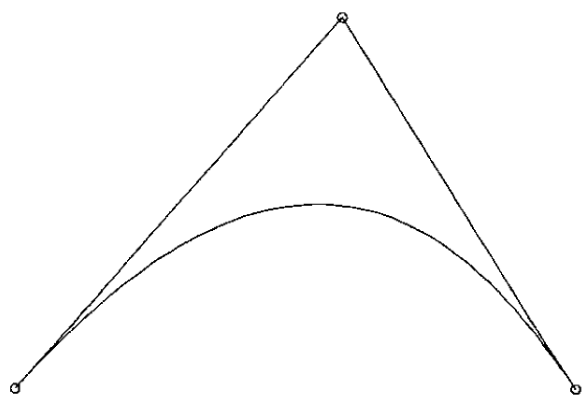
$$B_0^1(t) = 1-t; B_1^1(t) = t : C(t) = (1-t)P_0 + tP_1$$



- n=2 : linear interpolation

$$C(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

  - $\{P_0, P_1, P_2\}$ → control polygon
  - $P_0=C(0)$ and $P_2=C(1)$
  - Tangent directions at endpoints are parallel to P_1-P_0 and P_2-P_1
  - Curve contained in triangle $P_0P_1P_2$



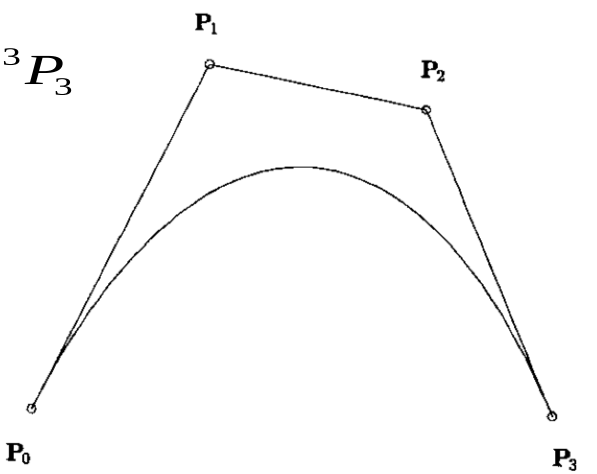
Some Bezier curve examples

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}; C(t) = \sum_{i=0}^{i=m} B_i^n(t) P_i$$

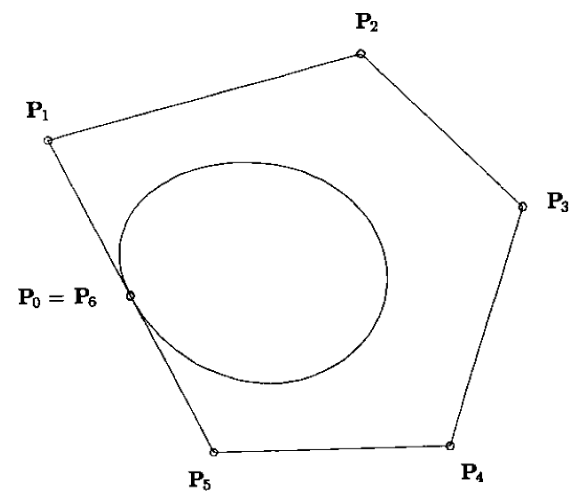
- n=3 : cubic Bezier curve

$$C(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

  - Control polygon (CP) approximates the curve shape, curve contained in this convex hull
  - Interpolate endpoints
  - Tangent at endpoints are parallel to P_1-P_0 and P_2-P_1
  - Variation diminishing property: no straight plane intersects the curve more times than it intersects the CP (curve doesn't wiggle more than CP)



- n=6 : a degree-6 closed Bezier curve
  - G1 continuous at C(0)=C(1)



Derivatives

- Tangent vectors are evaluated at the end-points

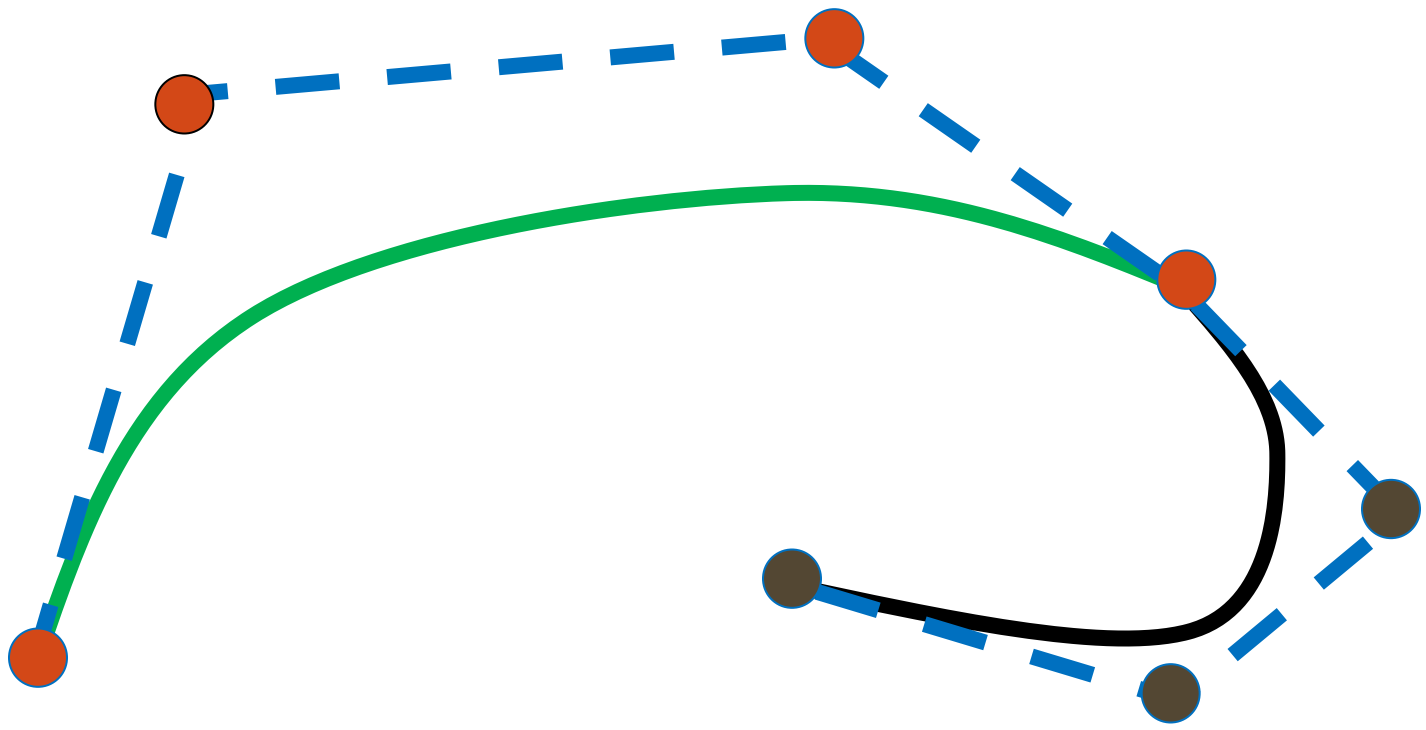
$$\mathbf{c}'(0) = 3(\mathbf{p}_1 - \mathbf{p}_0); \mathbf{c}'(1) = (\mathbf{p}_3 - \mathbf{p}_2)$$

- Second derivatives at end-points can also be easily computed:

$$\mathbf{c}^{(2)}(0) = 2 \times 3((\mathbf{p}_2 - \mathbf{p}_1) - (\mathbf{p}_1 - \mathbf{p}_0)) = 6(\mathbf{p}_2 - 2\mathbf{p}_1 + \mathbf{p}_0)$$

$$\mathbf{c}^{(2)}(1) = 2 \times 3((\mathbf{p}_3 - \mathbf{p}_2) - (\mathbf{p}_2 - \mathbf{p}_1)) = 6(\mathbf{p}_3 - 2\mathbf{p}_2 + \mathbf{p}_1)$$

Piecewise Bezier Curves



Piecewise Bezier Curves

- C0 continuity

$$\mathbf{p}_3 = \mathbf{q}_0$$

- C1 continuity

$$\begin{cases} \mathbf{p}_3 = \mathbf{q}_0 \\ (\mathbf{p}_3 - \mathbf{p}_2) = (\mathbf{q}_1 - \mathbf{q}_0) \end{cases}$$

- G1 continuity

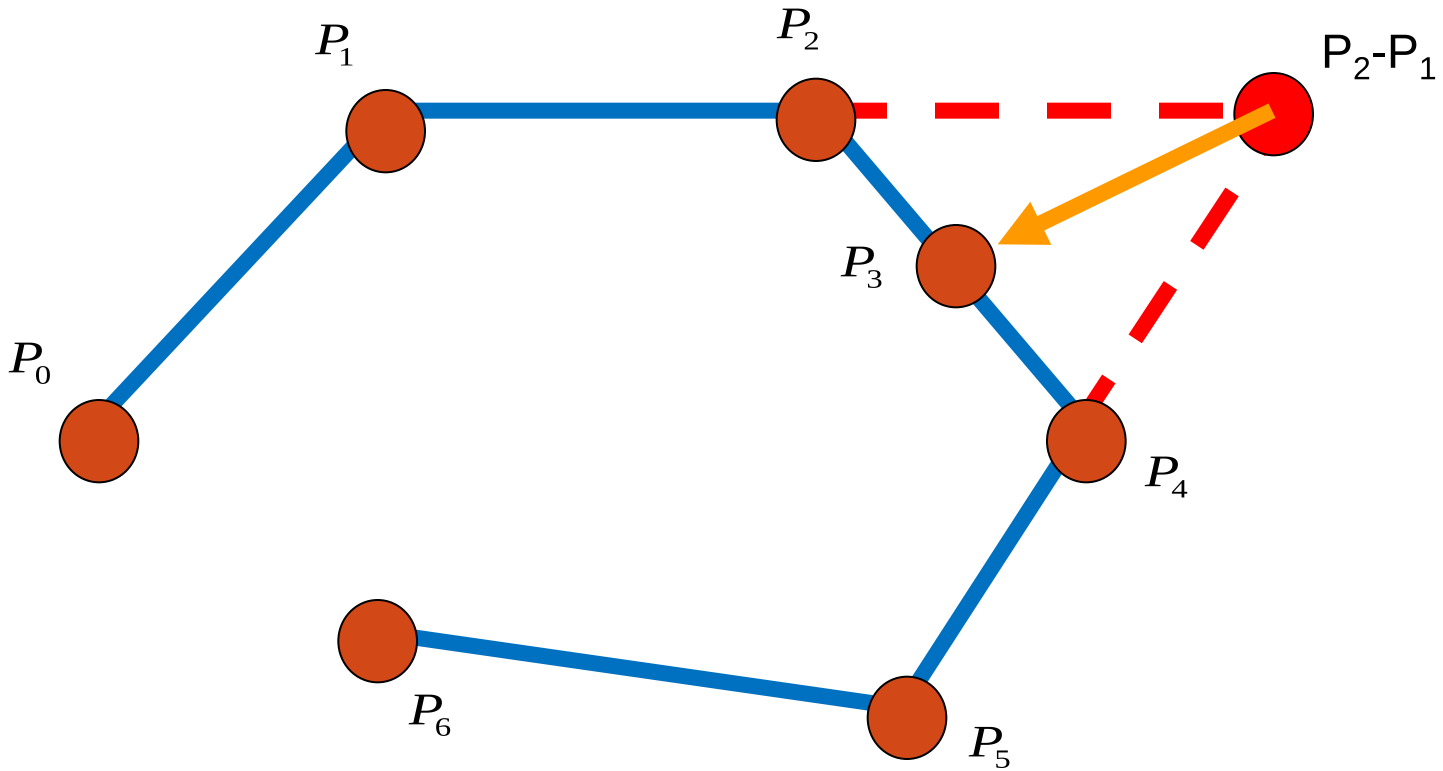
$$\begin{cases} \mathbf{p}_3 = \mathbf{q}_0 \\ (\mathbf{p}_3 - \mathbf{p}_2) = \alpha(\mathbf{q}_1 - \mathbf{q}_0) \end{cases}$$

- C2 continuity

$$\begin{cases} \mathbf{p}_3 = \mathbf{q}_0 \\ (\mathbf{p}_3 - \mathbf{p}_2) = (\mathbf{q}_1 - \mathbf{q}_0) \\ \mathbf{p}_3 - 2\mathbf{p}_2 + \mathbf{p}_1 = \mathbf{q}_2 - 2\mathbf{q}_1 + \mathbf{q}_0 \end{cases}$$

- G2 continuity

Piecewise C2 Bezier Curves

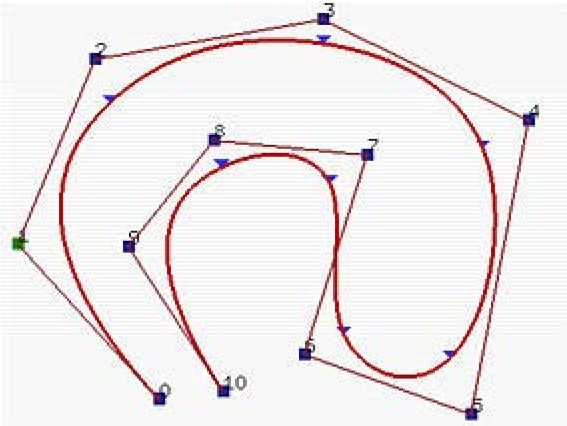


Spline

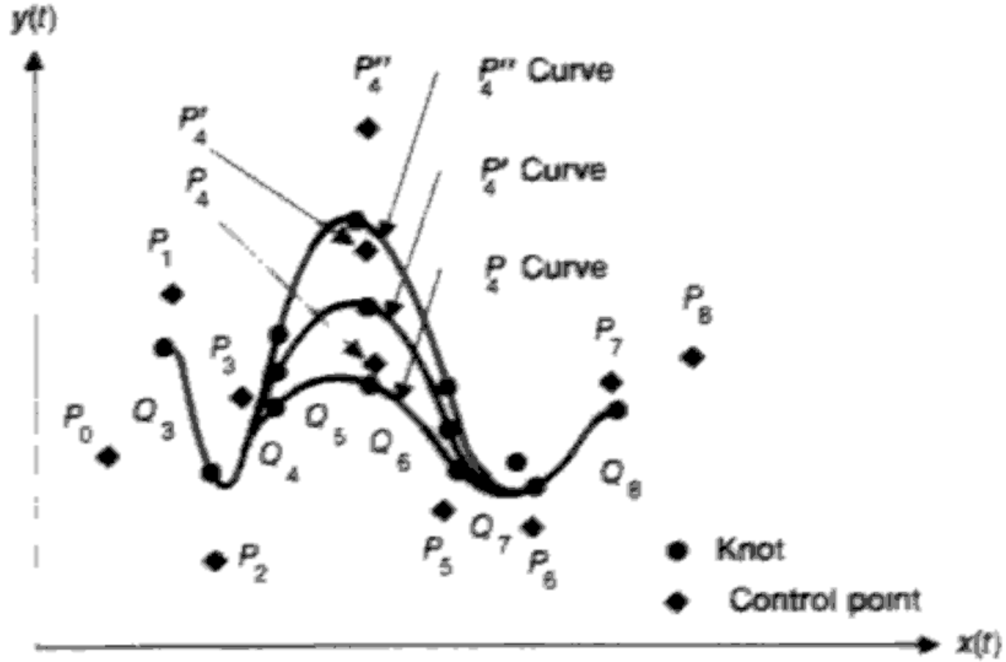
- Spline = long flexible strips of metal used by draftspersons to lay out the surfaces of airplanes, cars, and ships
- The metal splines, unless severely stressed, had second-order continuity
- B-Spline
  - Same basis function of Bezier Curves
  - Defined on a sequential parametric segments

$$\mathbf{S}(t) = \sum_{i=0}^{m-n} \mathbf{P}_i b_{i,n}(t) \ , \ t \in [t_{n-1}, t_{m-n}]$$

R. Bartels, J. Beatty, and B. Barsky, “An Introduction to Splines for Use in Computer Graphics and Geometric Modeling”, Morgan Kaufmann, 1987



B-Spline Definition



- Definitions:
  - Uniform B-spline: knots are spaced at equal intervals of the parameter t (e.g. $t_3=0$, $t_{i+1}-t_i=1$)
  - Nonuniform B-spline: …
  - Nonrational vs rational (later)
  - B-spline: →basis : splines are weighted sums of polynomial basis functions

B-Spline Basis Functions

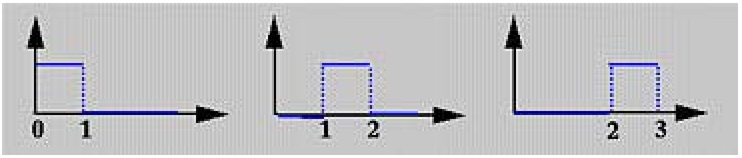
$$B_{i,1}(t) = \begin{cases} 1 & t_i <= t < t_{i+1} \\ 0 & otherwise \end{cases}$$

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

Bezier curve is a special case of it

B-Spline Basis Functions

- Degree-1:



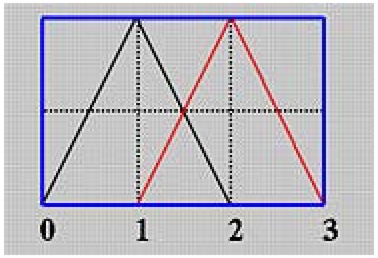
$$B_{0,1}(t) = \begin{cases} 1 & t \in [0,1] \\ 0 & o/w \end{cases}$$

- Degree-2:

$$B_{0,2}(t) = \begin{cases} t & t \in [0,1] \\ 2-t & t \in [1,2] \end{cases}$$

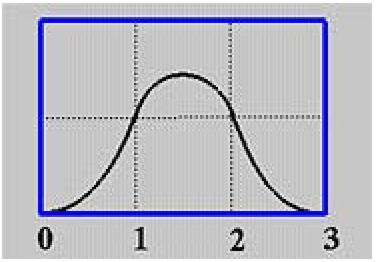
$$B_{1,2}(t) = \begin{cases} t-1 & t \in [1,2] \\ 3-t & t \in [2,3] \end{cases}$$

$$B_{2,2}(t) = \begin{cases} t-2 & t \in [2,3] \\ 4-t & t \in [3,4] \end{cases}$$



B-Spline Basis Functions

- Degree-3: Quadratic example (knot vector is [0,1,2,3,4,5,6])



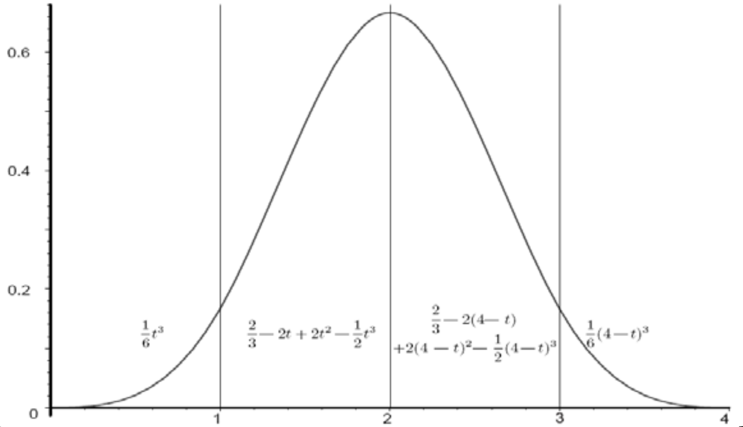
$$B_{0,3}(t) = \begin{cases} \frac{1}{2}t^2, & 0 <= t < 1 \\ \frac{1}{2}t(2-t) + \frac{1}{2}(t-1)(3-t), & 1 <= t < 2 \\ \frac{1}{2}(3-t)^2, & 2 <= t < 3 \end{cases}$$

$$B_{1,3}(t) = \begin{cases} \frac{1}{2}(t-1)^2, & 1 <= t < 2 \\ \frac{1}{2}(t-1)(3-t) + \frac{1}{2}(t-2)(4-t), & 2 <= t < 3 \\ \frac{1}{2}(4-t)^2, & 3 <= t < 4 \end{cases}$$

$$B_{2,3}(t) = \ldots\ldots$$

$$B_{3,3}(t) = \ldots\ldots$$

- Degree-4:



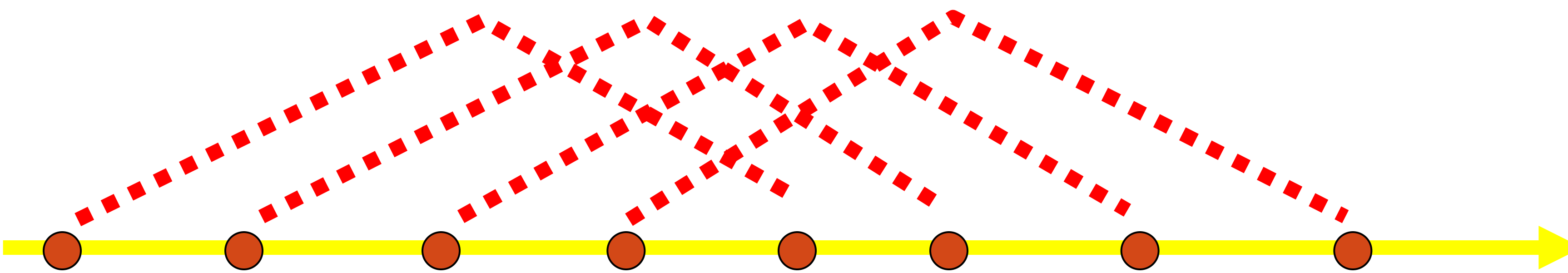
B-Spline Basis Functions

$$B_{0,1} \quad B_{1,1} \quad B_{2,1} \quad B_{3,1} \quad B_{4,1} \quad B_{5,1} \quad B_{6,1}$$

$$B_{0,2} \quad B_{1,2} \quad B_{2,2} \quad B_{3,2} \quad B_{4,2} \quad B_{5,2}$$

$$B_{0,3} \quad B_{1,3} \quad B_{2,3} \quad B_{3,3} \quad B_{4,3}$$

$$B_{0,4} \quad B_{1,4} \quad B_{2,4} \quad B_{3,4}$$

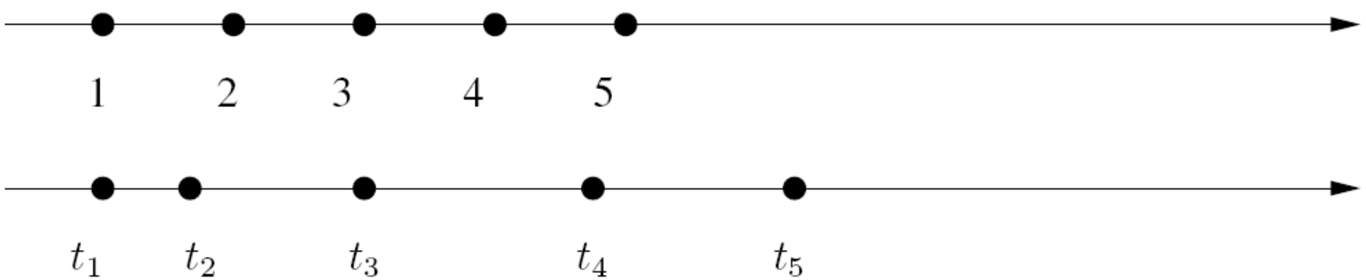


B-Spline Properties

- $B_{i,n}(t)$ is a piecewise polynomial of degree n, and with C^{n-1} continuity
- $B_{i,n}(t)$ has a support of length n+1
- Each curve segment is defined by n+1 control points, and each control point affects at most n+1 curve segments
- The degree of basis functions is independent of the number of control points
- Convex hull, local control
- Positivity, partition of unity, recursive evaluation

Uniform B-Spline

- Uniform vs Nonuniform:

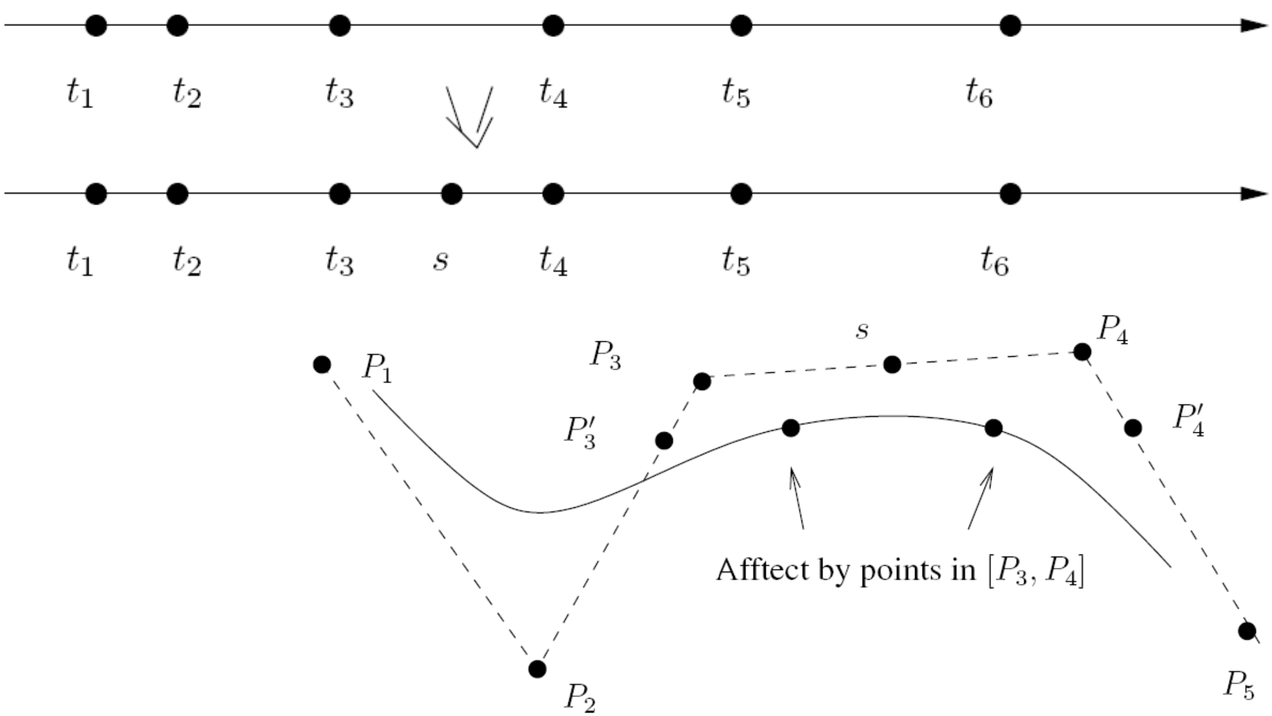


- Uniform Cubic B-spline (represented as Bezier control points)

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \\ \mathbf{p}_{i+3} \end{bmatrix}$$

Non-uniform B-Spline

- One of the most important advantage:
  - Knot insertion (locally adding a control point without changing the curve, for feature adjustment later)
    - Insert a new knot
    - Add a new control point, and update two control points



NURBS

- NURBS = Non Uniform Rational B-Splines
- Rational Functions → ratios of two polynomials
- Why: the need to represent some analytic shapes, for example, conic sections (e.g., circles, ellipses, parabolas)

→A non-uniform and rational extension of B-splines,
→A unified representation for polynomials, conic sections, etc.
→The industry standard representation

- Intuitively, rational representation adds weights to the control points, so that some control points are more important.

B-Spline →

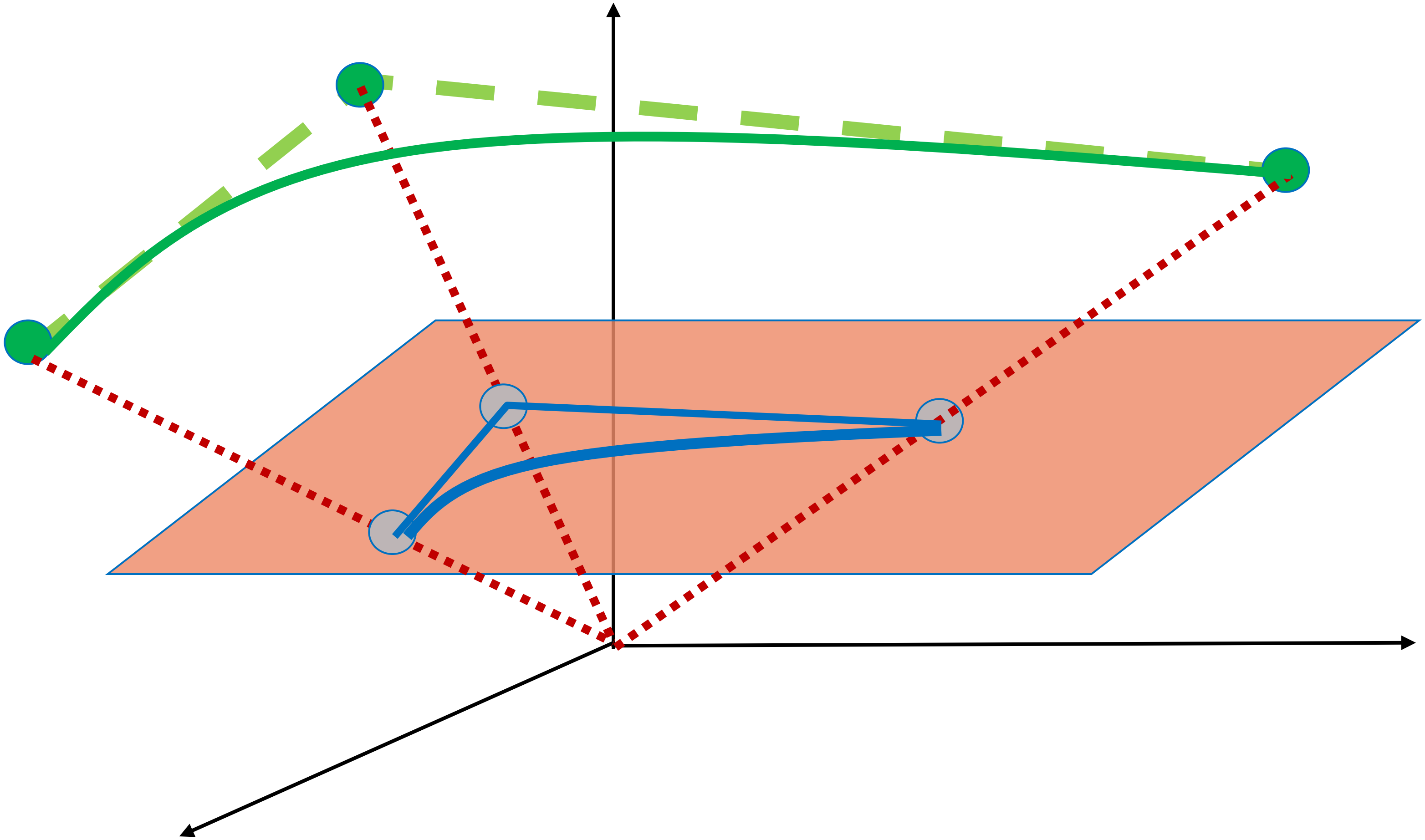
$$\mathbf{c}(u) = \sum_{i=0}^{n} \begin{bmatrix} \mathbf{p}_{i,x} w_i \\ \mathbf{p}_{i,y} w_i \\ \mathbf{p}_{i,z} w_i \\ w_i \end{bmatrix} B_{i,k}(u)$$

NURBS →

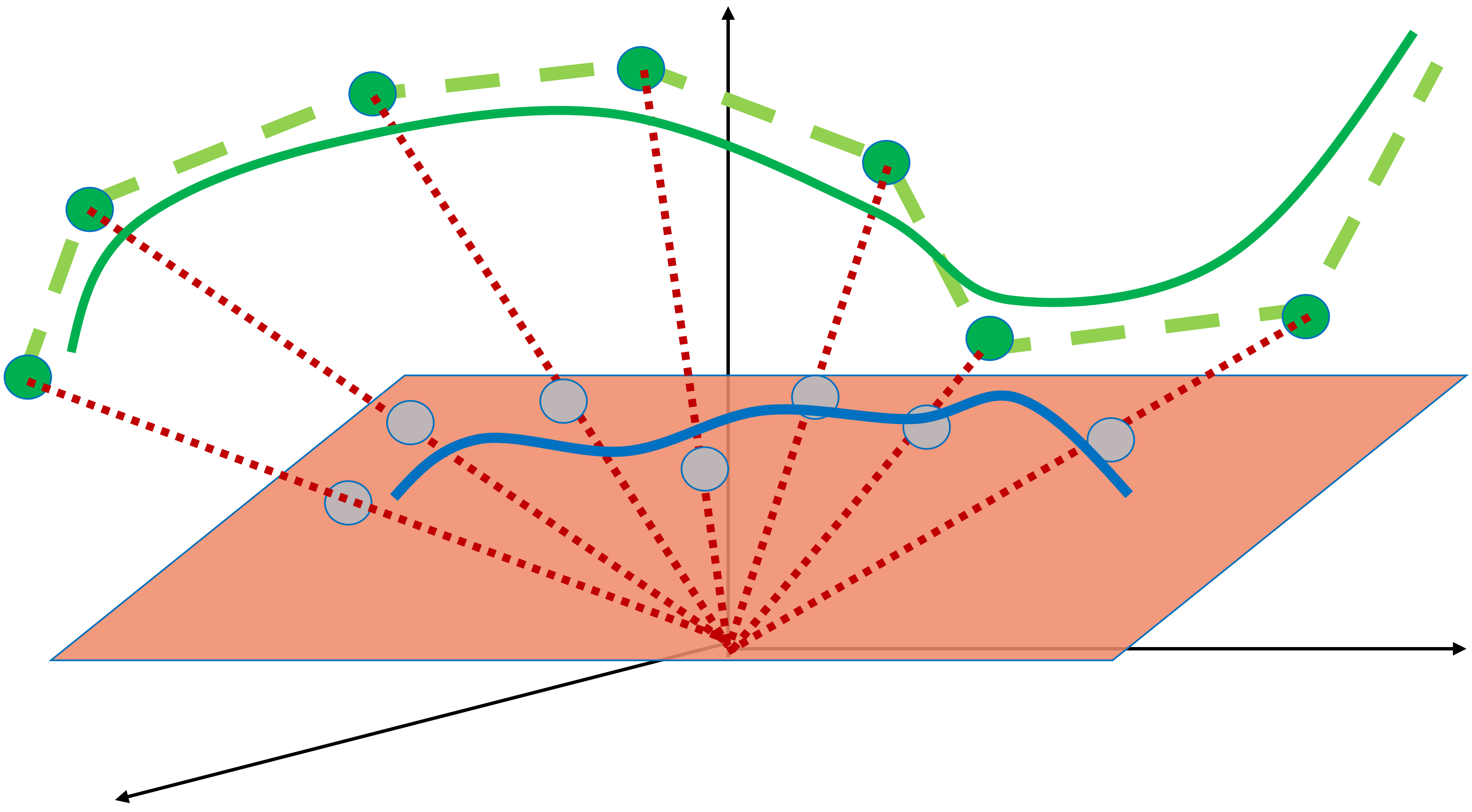
$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)}$$

Rational Bezier Curve

- Projecting a Bezier curve onto w=1 plane



From B-Splines to NURBS



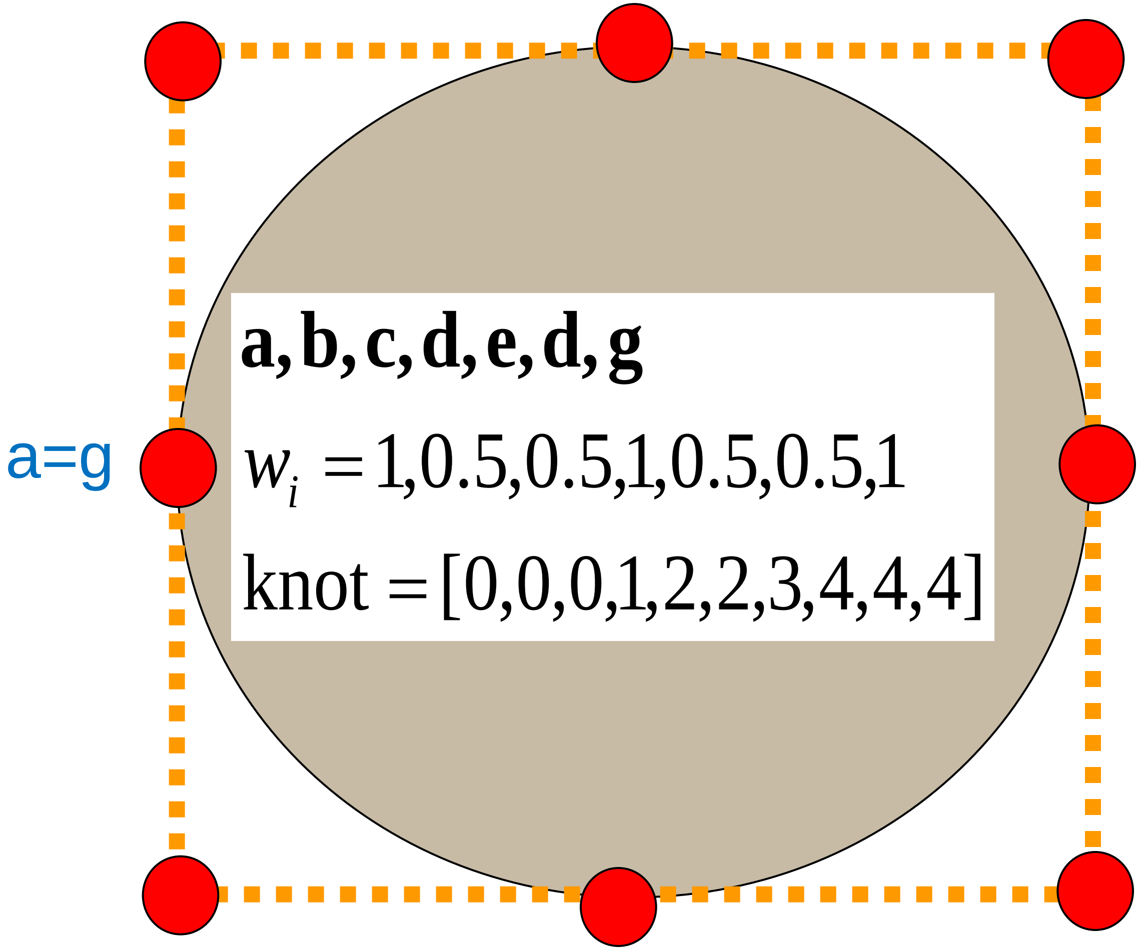
NURBS Weights

- Weight increase “attracts” the curve towards the associated control point
- Weight decrease “pushes away” the curve from the associated control point

NURBS for Analytic Shapes

- Conic sections
- Natural quadrics
- Extruded surfaces
- Ruled surfaces
- Surfaces of revolution

NURBS Circle



$$\mathbf{a, b, c, d, e, d, g}$$

$$w_i = 1, 0.5, 0.5, 1, 0.5, 0.5, 1$$

$$\text{knot} = [0,0,0,1,2,2,3,4,4,4]$$

… Will be explained in the next class…

NURBS Curve

- Geometric components
  - Control points, parametric domain, weights, knots
- Homogeneous representation of B-splines
- Geometric meaning --- obtained from projection
- Properties of NURBS
  - Represent standard shapes, invariant under perspective projection, B-spline is a special case, weights as extra degrees of freedom, common analytic shapes such as circles, clear geometric meaning of weights

Review

- Polynomial representation:
  - Curve as

$$x(t) = a_0 + a_1 t + \dots + a_n t^n$$
$$y(t) = b_0 + b_1 t + \dots + b_n t^n$$
$$z(t) = c_0 + c_1 t + \dots + c_n t^n$$

$\Rightarrow$

$$T = [t^n \quad \dots \quad t^1 \quad 1];$$

$$C = \begin{bmatrix} c_n^x & c_n^y & c_n^z \\ c_{n-1}^x & c_{n-1}^y & c_{n-1}^z \\ \dots & \dots & \dots \\ c_0^x & c_0^y & c_0^z \end{bmatrix};$$

$$Q(t) = [x(t) \quad y(t) \quad z(t)] = T \cdot C = \sum_{i=0}^{n} \vec{c}_i t^i$$

Or the weighted sum format with different types of basis functions:

For example, Bezier curve:

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

$$C(t) = \sum_{i=0}^{i=n} B_i^n(t) P_i$$

Rational Bezier Curves

- Although polynomials offer many advantages, there exist a number of important curve and surface types which cannot be represented precisely using polynomials (e.g. circles, ellipses, hyperbolas, cylinders, cones, spheres, etc.)
- Example: unit circle in the xy plane can't be represented using polynomial functions
  - If it has a parametric representation : $x(t) = a_0 + a_1 t + \ldots + a_n t^n$, $y(t) = b_0 + b_1 t + \ldots + b_n t^n$
  - Then $x^2+y^2-1=0$ implies:

$$\begin{aligned} 0 &= (a_0 + a_1 t + \ldots + a_n t^n)^2 + (b_0 + b_1 t + \ldots + b_n t^n)^2 - 1 \\ &= (a_0{}^2 + b_0{}^2 - 1) + 2(a_0 a_1 + b_0 b_1)t + (a_1{}^2 + 2a_0 a_2 + b_1{}^2 + 2b_0 b_2)t^2 \qquad (1) \\ &+ \ldots + 2(a_n a_{n-1} + b_n b_{n-1})t^{2n} - 1 + (a_n{}^2 + b_n{}^2)t^{2n} \end{aligned}$$

  - Equation (1) should hold for all t, which implies that all coefficients are zero

Rational Bezier Curves (cont.)

- Example: unit circle in the xy plane can't be represented using polynomial functions

$$0 \equiv (a_0 + a_1 t + \ldots + a_n t^n)^2 + (b_0 + b_1 t + \ldots + b_n t^n)^2 - 1$$
$$= (a_0{}^2 + b_0{}^2 - 1) + 2(a_0 a_1 + b_0 b_1)t + (a_1{}^2 + 2a_0 a_2 + b_1{}^2 + 2b_0 b_2)t^2$$
$$+ \ldots + 2(a_n a_{n-1} + b_n b_{n-1})t^{2n} - 1 + (a_n{}^2 + b_n{}^2)t^{2n}$$

  - (cont.)
  - (1) $a_n{}^2 + b_n{}^2 = 0 \Rightarrow a_n = b_n = 0$
  - (2) $a_{n-1}{}^2 + 2a_{n-2}a_n + b_{n-1}{}^2 + 2b_{n-2}b_n = 0 \Rightarrow a_{n-1} = b_{n-1} = 0$
  - (3) …
  - (n) $a_1{}^2 + 2a_0 a_2 + b_1{}^2 + 2b_0 b_2 = 0 \Rightarrow a_0 = b_0 = 0$

    But this implies 0=(0+0-1)=-1

    This proves a circle can't be represented by a polynomial form.

- Conic sections can be represented by rational functions:
  - Unit circle: $x(t) = \dfrac{1-t^2}{1+t^2};\ y(t) = \dfrac{2t}{1+t^2}$
  - Ellipse (major radius 2 on y-axis, and minor radius 1 on x-axis): $x(t) = \dfrac{1-t^2}{1+t^2};\ y(t) = \dfrac{4t}{1+t^2}$
  - Hyperbola, center at (0, 4/3), with y-axis the transverse axis:

$$x(t) = \frac{-1+2t}{1+2t-2t^2};\ y(t) = \frac{4t(1-t)}{1+2t-2t^2}$$

Rational Bezier curve

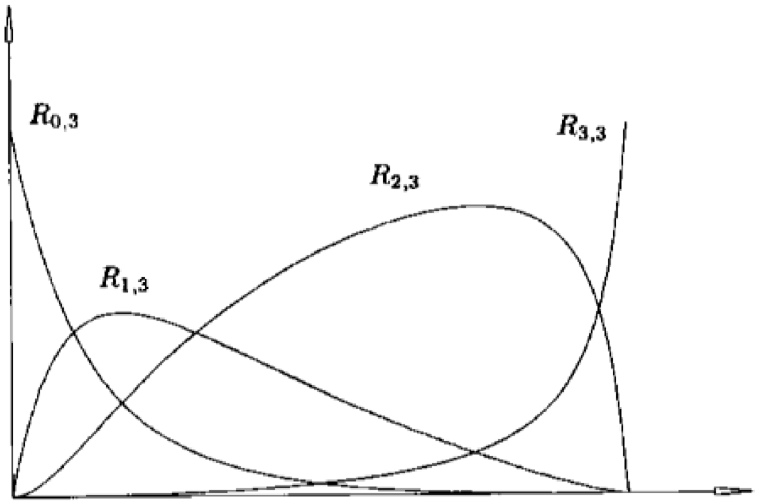
- nth-degree rational Bezier curve:

$$C(t)=\frac{\sum_{i=0}^{n}B_{i,n}(t)w_iP_i}{\sum_{i=0}^{n}B_{i,n}(t)w_i},0\le t\le 1$$

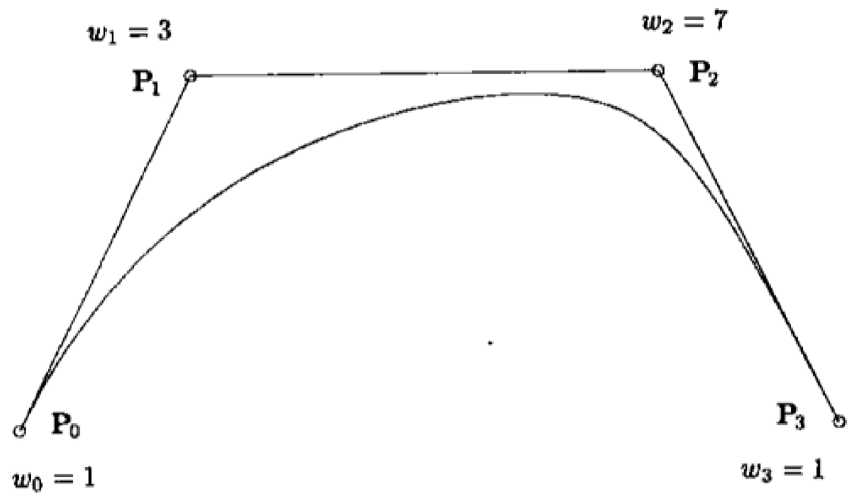
or $C(t)=\sum_{i=0}^{n}R_{i,n}(t)P_i,0\le t\le 1$ where $R_{i,n}(t)=\frac{B_{i,n}(t)w_i}{\sum_{i=0}^{n}B_{i,n}(t)w_i},0\le t\le 1$

- Properties:
  - Nonnegativity, partition of unity, endpoints interpolation
  - $B_{i,n}(t)$ are a special case of the $R_{i,n}(t)$
  - Convex hull property, affine transformation invariance, variation diminishing property
  - The k^{th} derivative at $t=0$ ($t=1$) depends on the first (last) $k+1$ control points and weights, in particular $C'(0)$ and $C'(1)$ are parallel to P_1-P_0 and P_n-P_{n-1} respectively.

Rational Bezier curve example



(a) Basis functions;



(b) Bézier curve.

Using Homogeneous Coordinates

A 2D curve example:

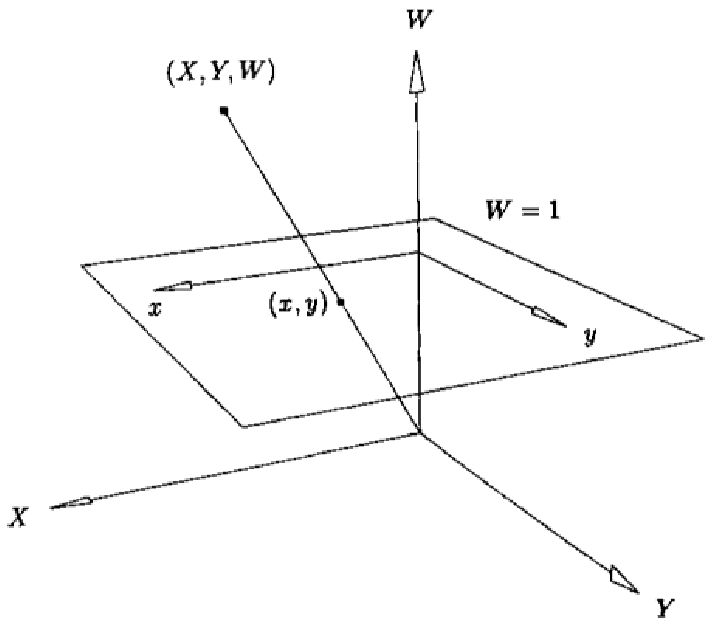
- Given a set of control points $\{P_i\}$, and weights $\{w_i\}$
- Construct the weighted control points Q_i $(w_i x_i, w_i y_i, w_i)$
- In 3D:

$$x^H(t) = \sum_{i=0}^{n} B_{i,n}(t) w_i x_i$$

$$y^H(t) = \sum_{i=0}^{n} B_{i,n}(t) w_i y_i$$

$$w(t) = \sum_{i=0}^{n} B_{i,n}(t) w_i$$

- Project it back onto the w=1 plane:



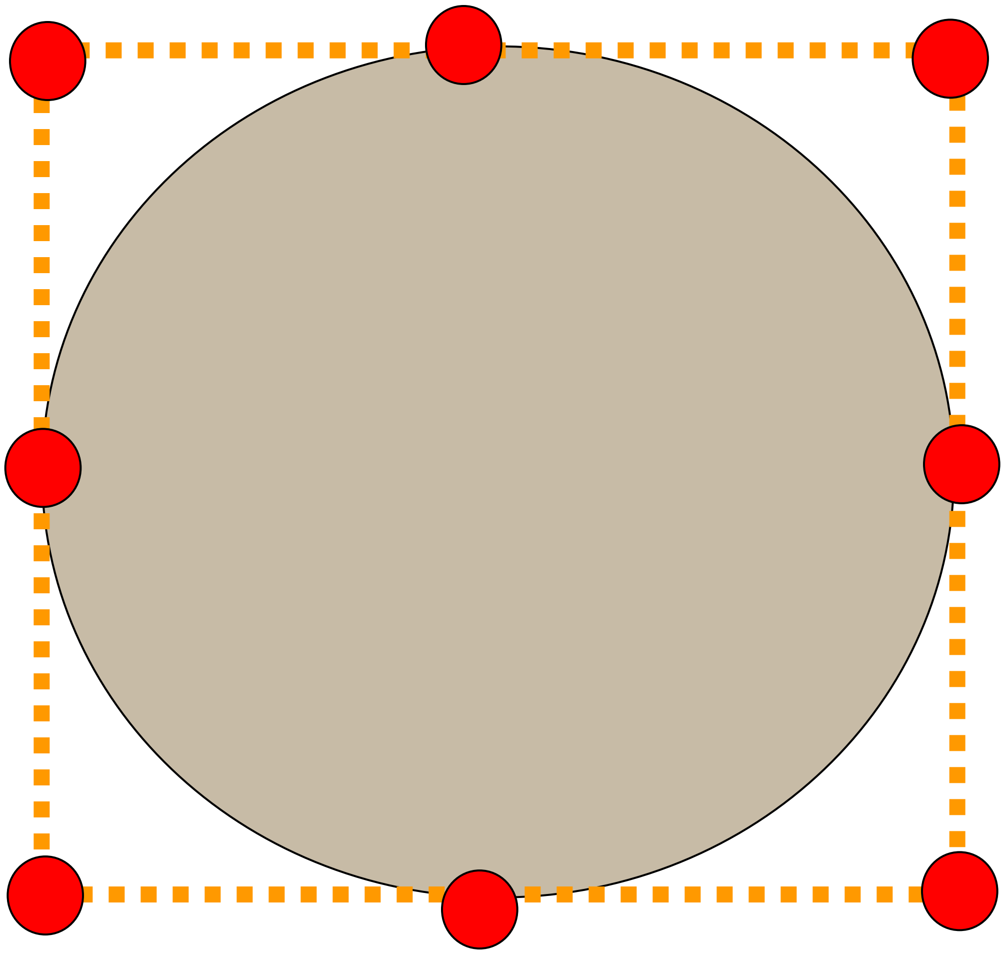
$$x(t) = \frac{x^H(t)}{w(t)} = \frac{\sum_{i=0}^{n} B_{i,n}(t) w_i x_i}{\sum_{i=0}^{n} B_{i,n}(t) w_i}, \quad y(t) = \frac{y^H(t)}{w(t)} = \frac{\sum_{i=0}^{n} B_{i,n}(t) w_i y_i}{\sum_{i=0}^{n} B_{i,n}(t) w_i}$$

$$C(t) = \frac{\sum_{i=0}^{n} B_{i,n}(t) w_i P_i}{\sum_{i=0}^{n} B_{i,n}(t) w_i}$$

Example:

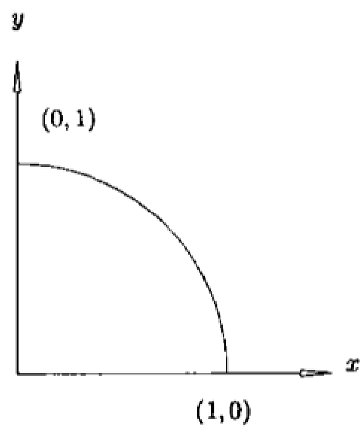
Rational functions for unit circle

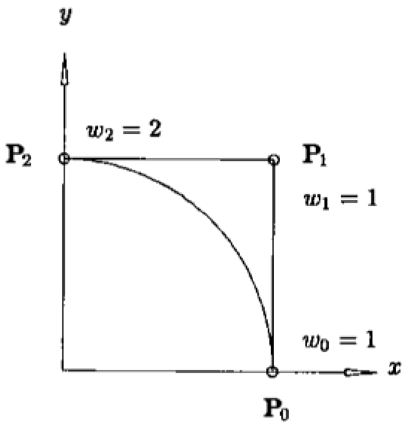
Where to put control points, and how to set their weights?



Suppose you know: a parametric representation of a circle is $x(t) = \frac{1-t^2}{1+t^2}; y(t) = \frac{2t}{1+t^2}$

The unit circle example





Look at one quadrant of the unit circle: $\left(x(t)=\dfrac{1-t^2}{1+t^2}, y(t)=\dfrac{2t}{1+t^2}\right), 0 \le t \le 1$

- The quadric curve should have P_0, P_1, and P_2 placed as shown in the right figure. (Why?)
- For the weights, we have: $w(t)=1+t^2=\sum_{i=0}^{n} B_{i,2}(t)w_i = (1-t)^2 w_0 + 2t(1-t)w_1 + t^2 w_2$

  with $t=0,0.5,1$ → $w_0, w_1, w_2 = 1,1,2$
- Get homogeneous coordinates

$$\begin{array}{lcl} P_0=(1,0) & & Q_0=(1,0,1) \\ P_1=(1,1) & \Rightarrow & Q_1=(1,1,1) \\ P_2=(0,1) & & Q_2=(0,2,1) \end{array}$$

The unit circle example

The 3D parametric curve is a parabolic arc:

$$C^H(t) = \sum_{i=0}^{n} B_{i,2}(t) Q_i$$

Which projects onto a circular arc on the W=1 plane

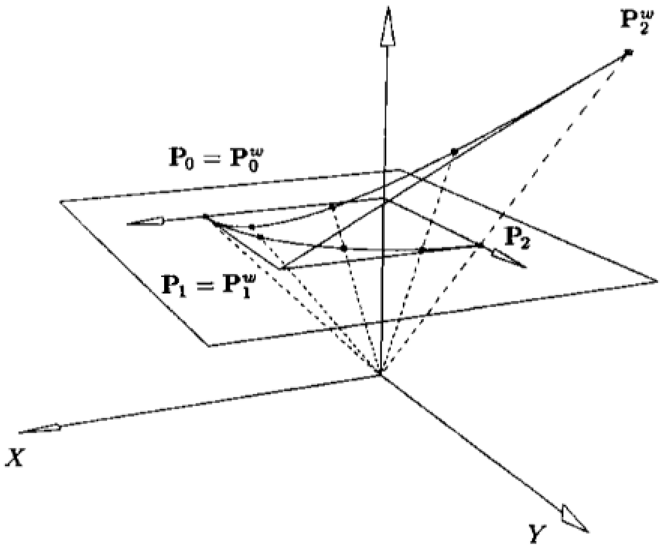
On any given t, for example, $t=1/2$

$$C^H(\frac{1}{2}) = \sum_{i=0}^{2} B_{i,2}(\frac{1}{2}) Q_i$$

$$= \left(1-\left(\frac{1}{2}\right)\right)^2 (1,0,1) + 2\left(1-\left(\frac{1}{2}\right)\right)\left(\frac{1}{2}\right)(1,1,1) + \left(\frac{1}{2}\right)^2 (0,2,2)$$

$$= (\frac{3}{4}, 1, \frac{5}{4})$$

$$\Rightarrow \quad C(\frac{1}{2}) = (3/5, 4/5)$$



NURBS Curves

An p-degree NURBS Curve:

$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)} \quad \text{where} \quad B_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & o/w \end{cases}$$

$$B_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} B_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} B_{i+1,k-1}(u)$$

Note:

- Computation of a set of basis functions requires specification of a knot vector U and the degree k
- It may yield the quotient $0/0$, we define it to be zero
- $B_{i,p}(u)$ defined on the entire real line, but only the $[u_0, u_m]$ is of interest.
- The interval $[u_i, u_{i+1})$ is called the ith knot span, and can have zero length
- The computation of pth-degree functions generates a trucated triangular table
- Exercise: Curve with $U = \{\underbrace{0, \ldots, 0}_{k+1}, \underbrace{1, \ldots, 1}_{k+1}\}$ is a generalized p-degree Bezier representation.

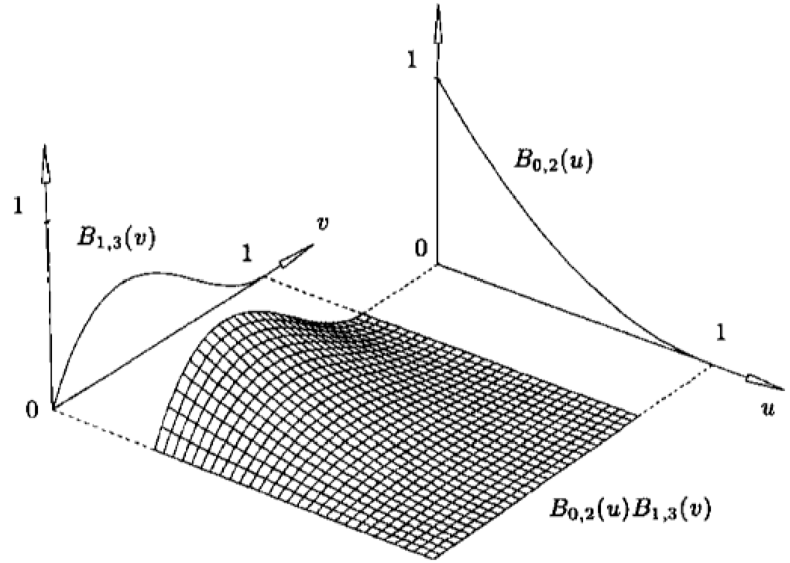
Spline Surface

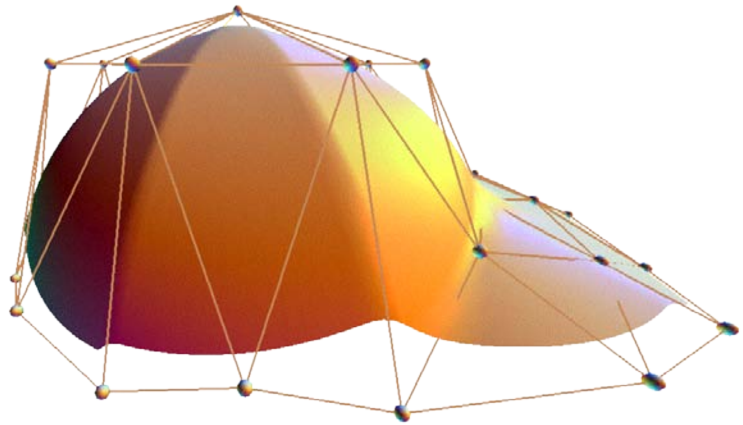
A curve → vector function of one parameter, mapping of a straight line segment into Euclidean 3D space

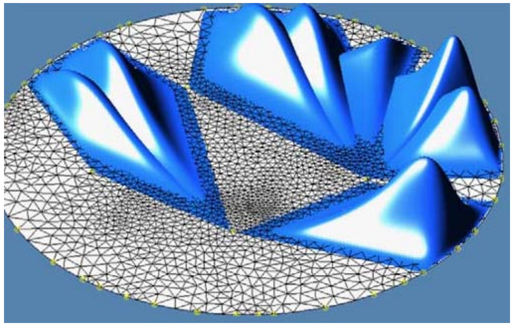
A surface → a vector-valued function of two parameters, mapping of a region into Euclidean 3D space

Spline Surface Categories (classified by domain schemes):

- ❑Tensor product patches
- ❑Triangular patches
- ❑…







Tensor Product Surfaces

Basis functions:
 bivariate functions of u and v
(constructed as products of univariate basis functions)

A tensor product surface $S^T(u,v) = (x(u,v), y(u,v), z(u,v)) = \sum_{i=0}^{n}\sum_{j=0}^{m} f_i(u) g_j(v) b_{i,j};$

$$where \begin{cases} b_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ 0 \le u, v \le 1 \end{cases}$$

The (u,v) domain of this mapping is a square (rectangle)

$$S^T(u,v) = [f_i(u)]^T [b_{i,j}][g_j(u)]$$

(n+1)*(m+1) matrix of 3D points

$$S^T(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} f_i(u) g_j(v) b_{i,j}$$

An example

A general parametric surface:

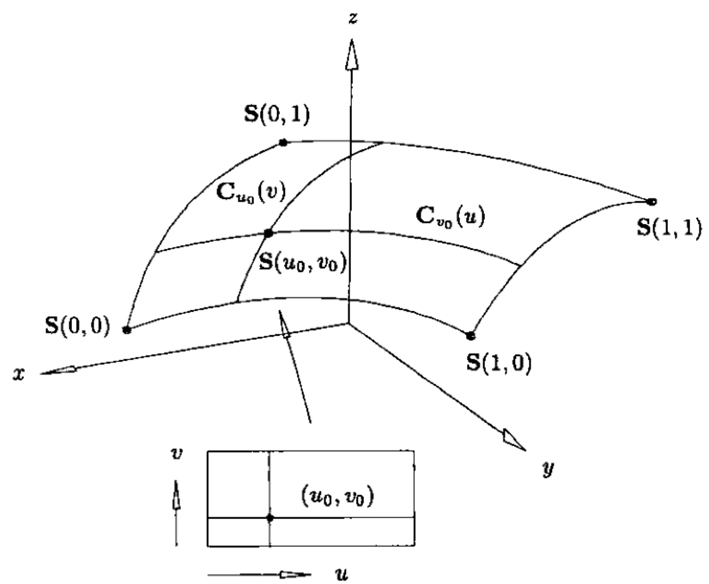
$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{a}_{i,j} u^i v^j = [u^i]^T [\mathbf{a}_{i,j}][v^j] \qquad \begin{cases} \mathbf{a}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ 0 \le u,\ v \le 1 \end{cases}$$

In tensor product representation: $f^i \rightarrow u_i$ and $g_j \rightarrow v^j$, basis functions make the products $\{u^i v^j\}$.

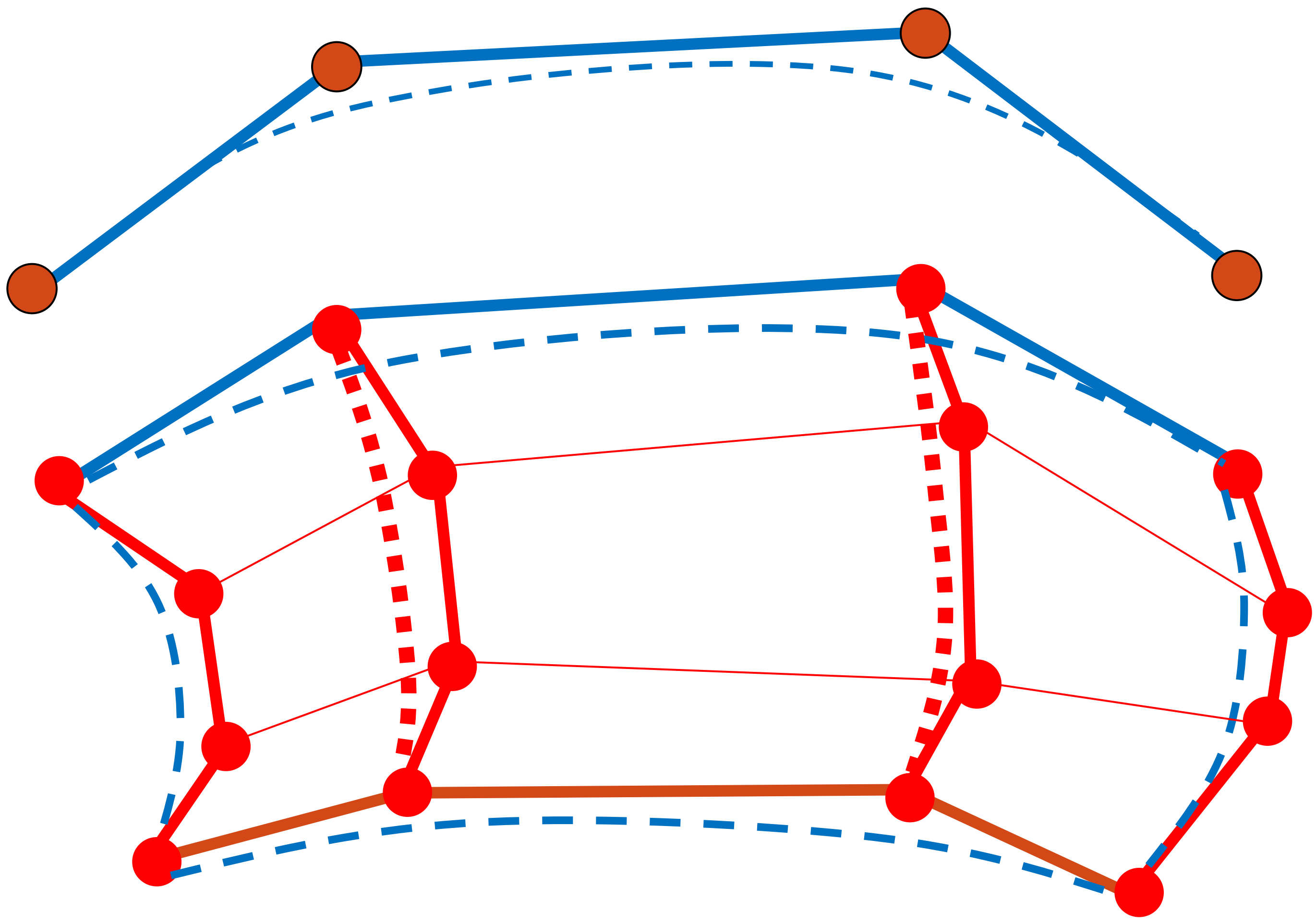
If we fix $u=u_0$ then we get an iso-curve:

$$\mathbf{C}_{u_0}(v) = \mathbf{S}(u_0, v) = \sum_{j=0}^{m}\left(\sum_{i=0}^{n} \mathbf{a}_{i,j}\, u_0^i\right) v^j = \sum_{j=0}^{m} \mathbf{b}_j(u_0)\, v^j$$

$$\mathbf{b}_j(u_0) = \sum_{i=0}^{n} \mathbf{a}_{i,j}\, u_0^i$$



Nonrational Bezier Surfaces



Nonrational Bezier Surfaces

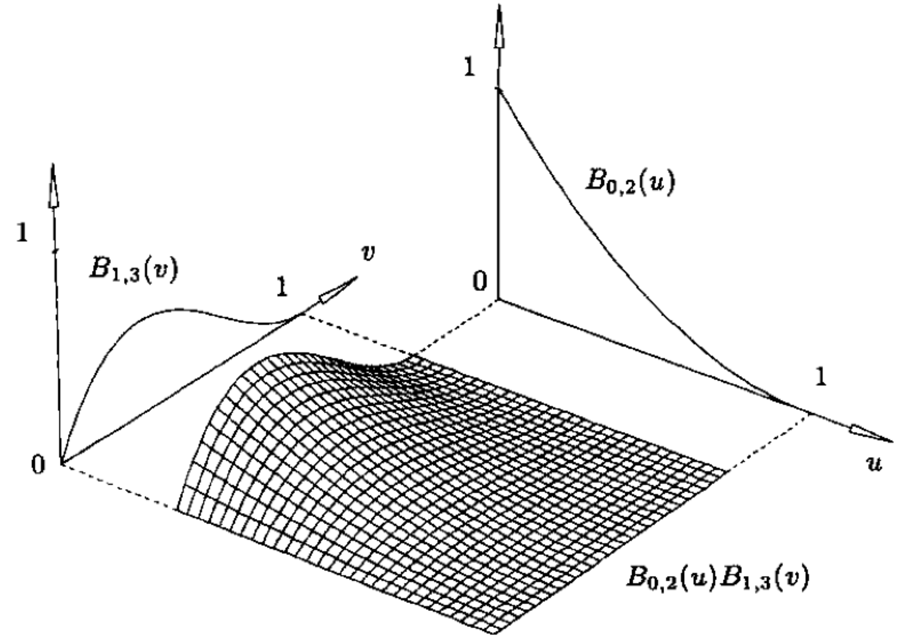
A bidirectional net of control points and products of the univariate Bernstein polynomials:

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v)\mathbf{P}_{i,j} \qquad 0 \le u,v \le 1$$

For fixed $u=u_0$: we get a Bezier curve

$$\begin{aligned}\mathbf{C}_{u_0}(v) = \mathbf{S}(u_0,v) &= \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u_0)B_{j,m}(v)\mathbf{P}_{i,j}\\ &= \sum_{j=0}^{m} B_{j,m}(v)\left(\sum_{i=0}^{n} B_{i,n}(u_0)\mathbf{P}_{i,j}\right)\\ &= \sum_{j=0}^{m} B_{j,m}(v)\mathbf{Q}_j(u_0)\end{aligned}$$

$$\mathbf{Q}_j(u_0) = \sum_{i=0}^{n} B_{i,n}(u_0)\mathbf{P}_{i,j} \qquad j = 0,\ldots,m$$



Properties of Nonrational Bezier Surfaces

- Non-negativity.

$$B_{i,n}(u)B_{j,m}(v) \geq 0 \text{ for all } i, j, u, v$$

- Partition of Unity.

$$\sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = 1 \text{ for all } u \text{ and } v$$

- S(u,v) is contained in the convex hull of its control points.

- Affine Transformation Invariance.

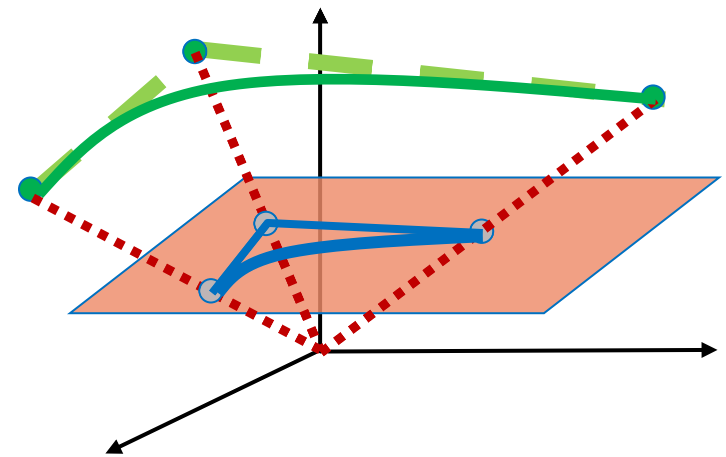
- The surface interpolates the four corner control points.

NURBS Curves → Surfaces

- NURBS curves.
- Tensor product?
- Question: can we get NURBS surface this way?
- Answer: NO!

  → NURBS are not tensor-product surfaces
- Can we have NURBS surface?
- YES.

NURBS Curves



$$c(u) = \frac{\sum_{i=1}^{n} p_i w_i B_{i,k}(u)}{\sum_{i=1}^{n} w_i B_{i,k}(u)}$$

$$\begin{bmatrix} c_x / c_w \\ c_y / c_w \\ c_z / c_w \end{bmatrix} \Leftarrow \begin{bmatrix} c_x(u) \\ c_y(u) \\ c_z(u) \\ c_w(u) \end{bmatrix} = \sum_{i=1}^{n} B_{i,k}(u) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix}$$

NURBS Surface

- NURBS surface definition:
  - A NURBS surface of degree k in u direction and degree l in the v direction is:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{p}_{i,j} w_{i,j} B_{i,k}(u) B_{j,l}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} B_{i,k}(u) B_{j,l}(v)}$$

- Geometric interpolation
- Not the tensor-product formulation. (Compare it with Bezier and B-spline construction)

NURBS Surface

$$s(u) = \frac{\sum_{i,j=1}^{n} p_{ij} w_{ij} B_{i,k}(u) B_{j,l}(v)}{\sum_{i,j=1}^{n} w_{ij} B_{i,k}(u) B_{j,l}(v)}$$

$$\begin{bmatrix} s_x / s_w \\ s_y / s_w \\ s_z / s_w \end{bmatrix} \Leftarrow \begin{bmatrix} s_x(u) \\ s_y(u) \\ s_z(u) \\ s_w(u) \end{bmatrix} = \sum_{i,j=1}^{n} B_{i,k}(u) B_{j,l}(v) \begin{bmatrix} w_{ij} x_{ij} \\ w_{ij} y_{ij} \\ w_{ij} z_{ij} \\ w_{ij} \end{bmatrix}$$

NURBS Surface

- Parametric variables: u and v
- Control points and their associated weights: (m+1)(n+1)
- Degrees of basis functions: (k-1) and (l-1)
- Knot sequence:

$$u_0 <= u_1 <= \ldots\ldots <= u_{m+k}$$

$$v_0 <= v_1 <= \ldots\ldots <= v_{n+l}$$

- Parametric domain:

$$u_{k-1} <= u <= u_{m+1}$$

$$v_{l-1} <= v <= v_{n+1}$$

NURBS Surface Property

Nonnegativity: $R_{i,j}(u,v) \geq 0$ for all i, j, u, and v;

Partition of unity: $\sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u,v) = 1$ for all $(u,v) \in [0,1] \times [0,1]$;

Local support: $R_{i,j}(u,v) = 0$ if (u,v) is outside the rectangle given by $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$;

In any given rectangle of the form $[u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, at most $(p+1)(q+1)$ basis functions are nonzero, in particular the $R_{i,j}(u,v)$ for $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$ are nonzero;

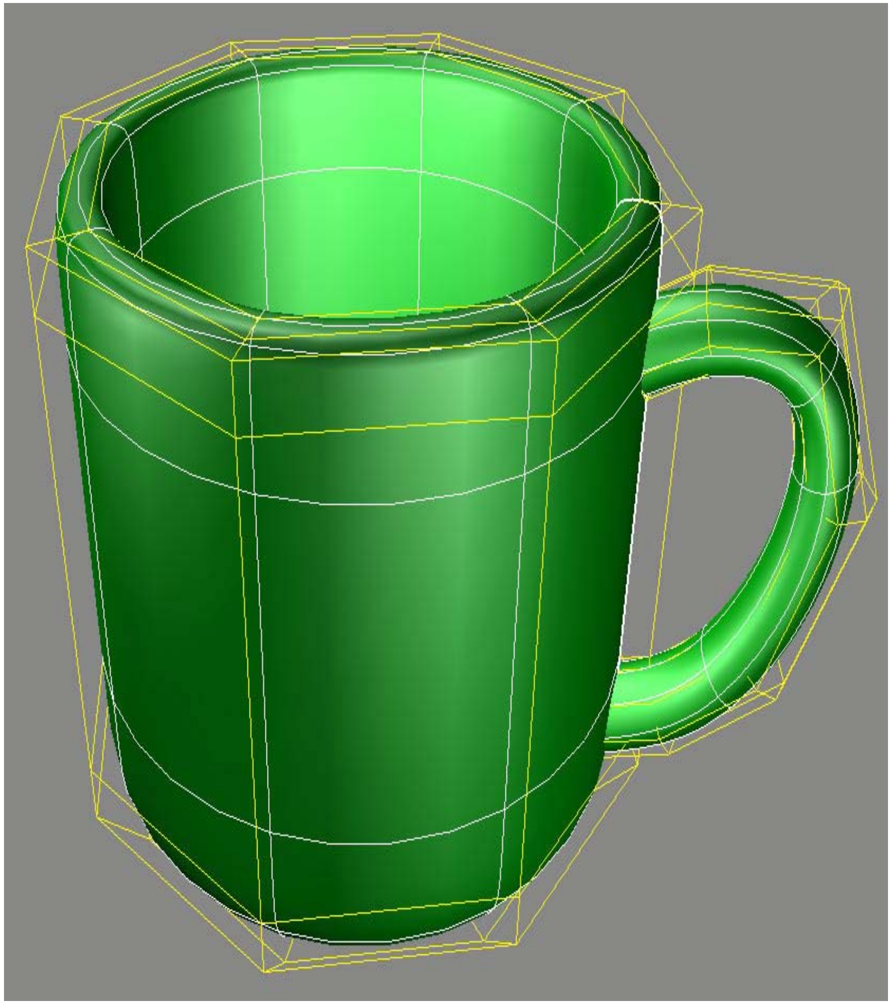
Corner point interpolation: $\mathbf{S}(0,0) = \mathbf{P}_{0,0}$, $\mathbf{S}(1,0) = \mathbf{P}_{n,0}$, $\mathbf{S}(0,1) = \mathbf{P}_{0,m}$, and $\mathbf{S}(1,1) = \mathbf{P}_{n,m}$;

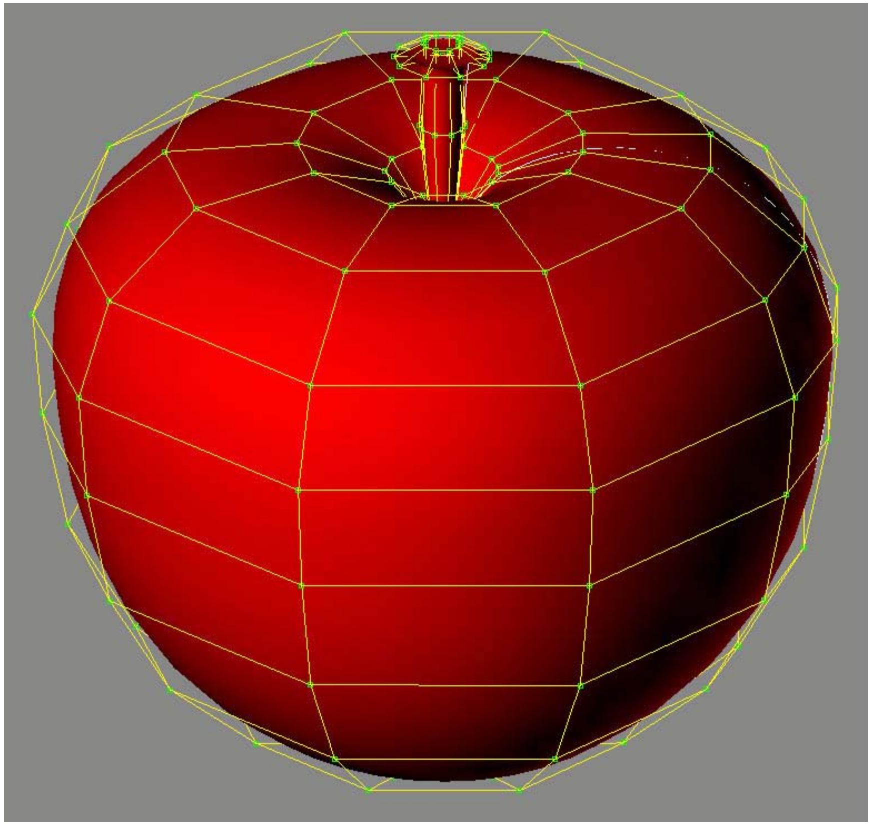
Affine invariance: an affine transformation is applied to the surface by applying it to the control points;

Strong convex hull property: assume $w_{i,j} \geq 0$ for all i, j. If $(u,v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, then $\mathbf{S}(u,v)$ is in the convex hull of the control points $\mathbf{P}_{i,j}$, $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$;

Local modification: if $\mathbf{P}_{i,j}$ is moved, or $w_{i,j}$ is changed, it affects the surface shape only in the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$;

NURBS Surface Examples





NURBS Surfaces

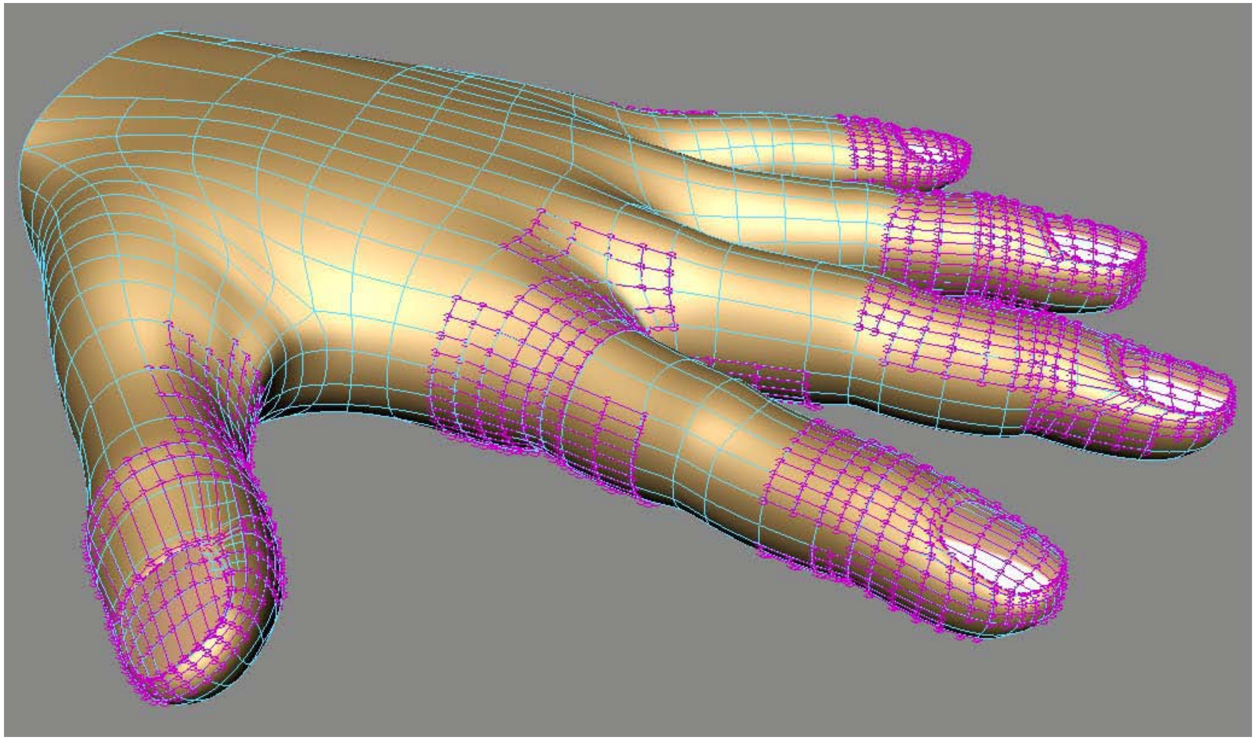
- Good for
  - Mechanical, manufactured parts
  - Smooth free-form surface representation
- Bad for
  - Non-genus-0 surfaces
  - Interactive design of free-form surfaces

Why NURBS

- Support free-form curves/surfaces modeling.
- Represent standard analytic shapes precisely.
- Local support.
- Convex hull.
- Affine transformation invariant.
- Strict analytic form for evaluation (important in CAD/CAM/CAE).

Why NOT NURBS

- Hard to model arbitrary topology.
- Regularity of tensor-product control polygon poses difficulty for level of detail.



Allow T-junctions
→ T-splines
(details in
EE7000 course)

Parametric Solids

- Tricubic solid

$$\mathbf{p}(u,v,w)=\sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3}\mathbf{a}_{ijk}u^i v^j w^k$$

$$u,v,w\in[0,1]$$

- Bezier solid

$$\mathbf{p}(u,v,w)=\sum_i\sum_j\sum_k \mathbf{p}_{ijk}B_i(u)B_j(v)B_k(w)$$

- B-spline solid

$$\mathbf{p}(u,v,w)=\sum_i\sum_j\sum_k \mathbf{p}_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)$$

- NURBS solid

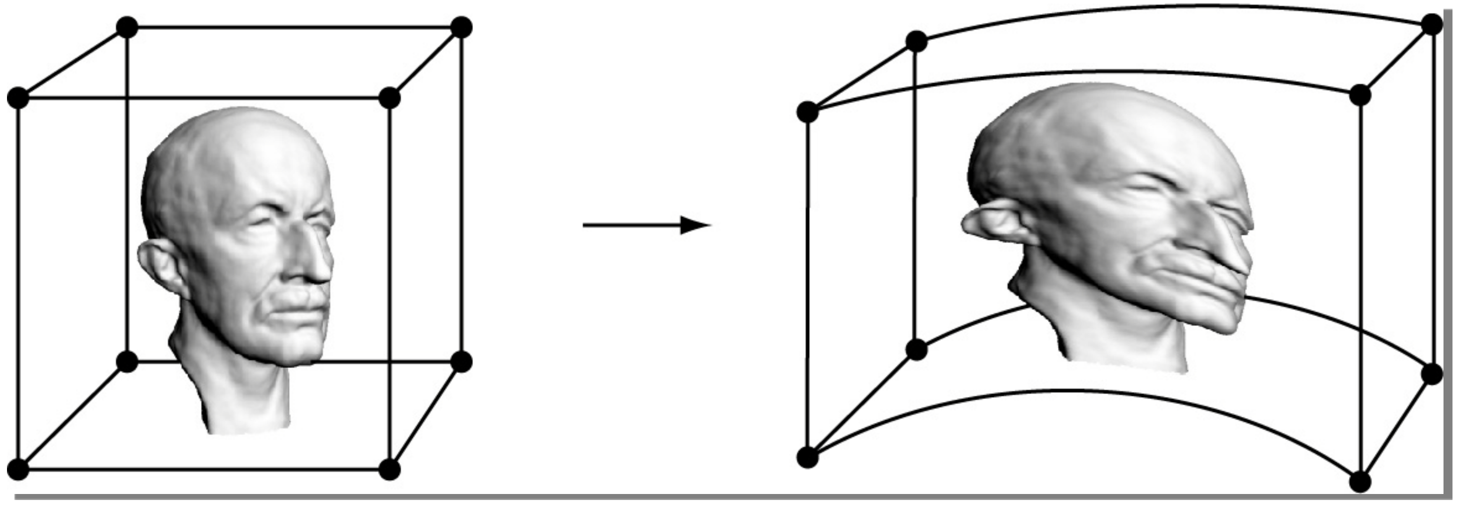
$$\mathbf{p}(u,v,w)=\frac{\sum_i\sum_j\sum_k \mathbf{p}_{ijk}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}{\sum_i\sum_j\sum_k q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}$$

Free-form Deformation

- Geometric objects are embedded into a space
- The surrounding space is represented by using commonly-used, popular splines
- Free-form deformation of the surrounding space
- All the embedded (geometric) objects are deformed accordingly, the quantitative measurement of deformation is obtained from the displacement vectors of the trivariate splines that define the surrounding space
- Essentially, the deformation is governed by the trivariate, volumetric splines
- Very popular in graphics and related fields

(Will be discussed in EE7000 course.)

Free-form Deformations



(courtesy of Pauly et al.)