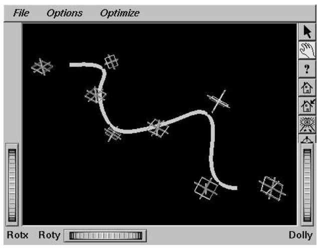
Lecture 17: Thursday, 4 November 1999

Administrative:

Per-team space arranged at `/mit/6.837/F99`
Each team should now have a shepherd TA
Written feedback in progress (from me, TAs)

Today:



(Warman Group)

Parametric Curves & Surfaces
- Hermite curves
- Bezier splines
- B-splines
- Spline bases and change of basis

Curved Primitives (FvDFHP §9.2)

Motivation:
- Necessary to model organic/machined surfaces
- Needed to "blend" differently-oriented primitives
- Designers tire of linear (polyhedral) primitives

Model plane and space curves first, then surfaces
- Applications beyond shape modeling:
  - Smooth variation of position, pose, etc.

Plane and space curves

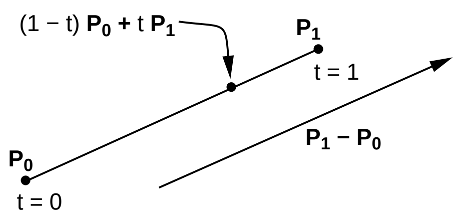
Want a curve primitive that has:
- Explicit representation (for rendering)
- Controllable start/end points
- Controllable start/end derivatives

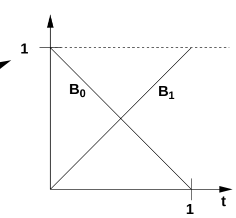
Try: implicit curves $Ax^2 + By^2 + ...F = 0$
- How to model "less" than the whole curve?
- How to fit to given position, derivative ?

Try: functions $y = f(x)$
- How to model double-valued or vertical curves?
- How to rotate the curve?

Parametric Curves

Express (simple) lerp in (complicated) new way





$$Q(t) = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix} = \begin{pmatrix} (P_0) & (P_1) \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}$$

What is the derivative $Q'(t)$ w.r.t. t?

$$Q'(t) = \frac{d}{dt}Q(t) = \begin{pmatrix} x'(t) \\ y'(t) \\ z'(t) \end{pmatrix} = \begin{pmatrix} (P_0) & (P_1) \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$= P_1 - P_0$$

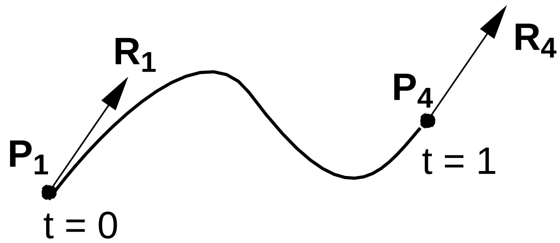
Note that $Q(t)$, $Q'(t)$ can be written as:

$Q(t) =$ Geometry $\mathbf{G}$ · Spline Basis $\mathbf{B}$ · Power Basis $\mathbf{T(t)}$

$$Q(t) = \mathbf{GBT(t)} \qquad Q'(t) = \mathbf{GBT'(t)}$$

Hermite Curves

Want to specify position, derivative at start, endpoint

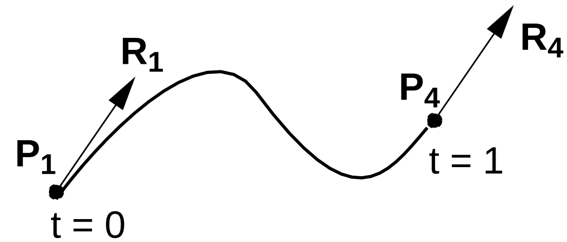


What do we know?

Linear, quadratic aren't sufficient. Why?

Hermite Curves

Position and derivative at start, endpoint



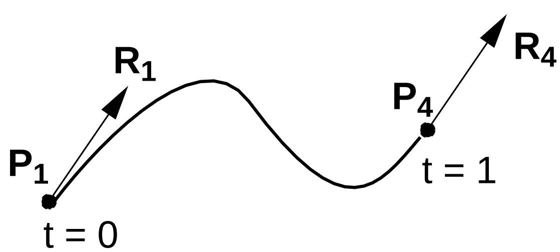
Knowns:

$$\begin{aligned}
\mathbf{G} &= \begin{pmatrix} P_1 & P_4 & R_1 & R_4 \end{pmatrix} \\
\mathbf{T(t)} &= \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}; \quad \mathbf{T'(t)} = \begin{pmatrix} 3t^2 \\ 2t \\ 1 \\ 0 \end{pmatrix} \\
Q(0) &= P_1 \\
Q(1) &= P_4 \\
Q'(0) &= R_1 \\
Q'(1) &= R_4
\end{aligned}$$

Hermite Curves

Given:

$Q(t) = GBT(t)$, $Q'(t) = GBT'(t)$;



Solve for B:

$$\begin{aligned}
GBT(0) &= P_1 \\
GBT(1) &= P_4 \\
GBT'(0) &= R_1 \\
GBT'(1) &= R_4
\end{aligned}$$

Hermite Matrix Formulation

$$\begin{pmatrix} (P_1) & (P_4) & (R_1) & (R_4) \end{pmatrix} B \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = P_1$$

$$\begin{pmatrix} (P_1) & (P_4) & (R_1) & (R_4) \end{pmatrix} B \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = P_4$$

$$\begin{pmatrix} (P_1) & (P_4) & (R_1) & (R_4) \end{pmatrix} B \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = R_1$$

$$\begin{pmatrix} (P_1) & (P_4) & (R_1) & (R_4) \end{pmatrix} B \begin{pmatrix} 3 \\ 2 \\ 1 \\ 0 \end{pmatrix} = R_4$$

Or, in matrix form as:

$$G = GB \begin{pmatrix} (T(0)) \\ (T(1)) \\ (T'(0)) \\ (T'(1)) \end{pmatrix} \quad \text{so} \quad B = \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}$$

$$Q(t) = GBT(t) = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix}$$

Hermite Curves

$$Q(t) = GBT = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix}$$

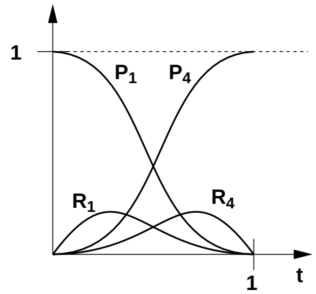
Write out $Q(t), Q'(t)$:

$$\begin{aligned} Q(t) &= (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4 \\ Q'(t) &= (6t^2 - 6t + 0)P_1 + (-6t^2 + 6t)P_4 + (3t^2 - 4t + 1)R_1 + (3t^2 - 2t)R_4 \end{aligned}$$

Check $Q(0), Q(1), Q'(0), Q'(1)$

Call polynomials in t the **Hermite basis**

Graph them with respect to t:

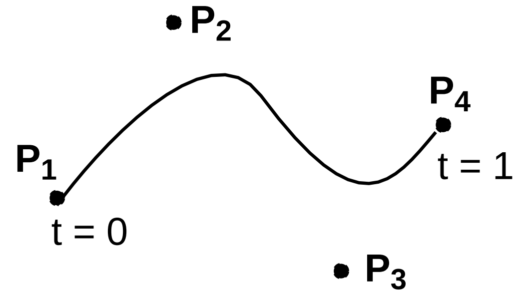


Note: ______________ !

Convex combination

Cubic Bézier Curve

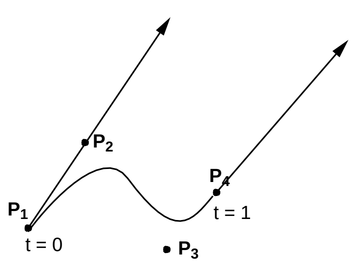
Given four control points P_1 .. P_4



$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4$$

Bézier Curve

$$\begin{aligned} Q(t) &= (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4 \\ Q'(t) &= -3(1-t)^2 P_1 + (9t^2 - 12t + 3)P_2 + (-9t^2 + 6t)P_3 + 3t^2 P_4 \end{aligned}$$

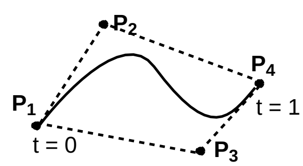


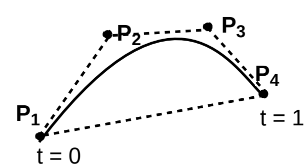
$$\begin{aligned} Q(0) &= P_1 \\ Q(1) &= P_4 \\ Q'(0) &= 3(P_2 - P_1) \\ Q'(1) &= 3(P_4 - P_3) \end{aligned}$$

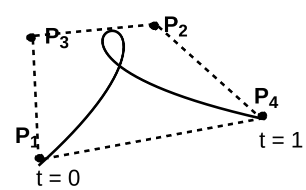
Property: Endpoint interpolation

Bézier Curve Properties

Convex hull property:

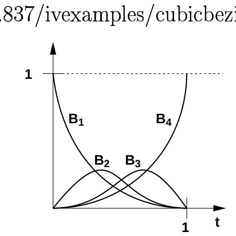






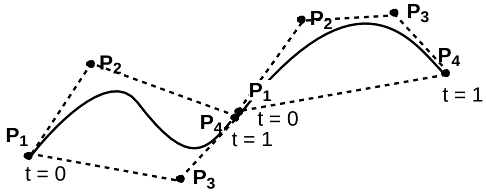
Is the curve planar ? Why or why not ?

[Demo: /mit/imagery4/6.837/ivexamples/cubicbezier.iv]



Tangent Matching

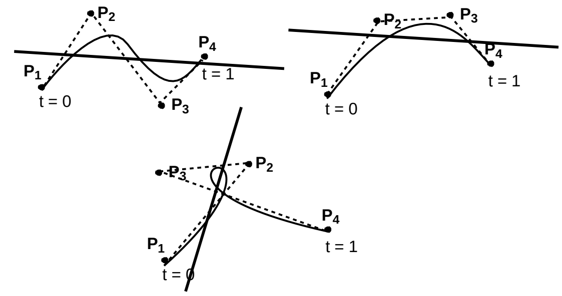
Can use control points to match tangents:



What are the design degrees of freedom?

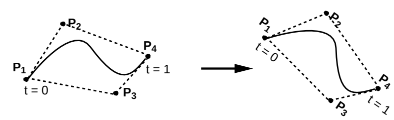
Variation Diminishing

Curve crosses line (plane) at most the number of times control polygon crosses line (plane)



Affine Invariance

Affine transformation, Bézier generation commute!



Note: does not commute with perspective! For that, we must use *rational* curves

Defined for any degree

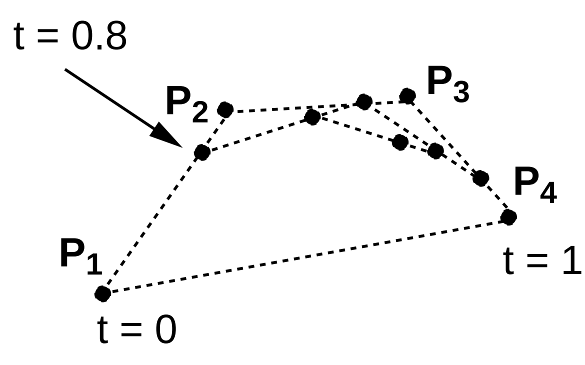
Use "Bernstein polynomials," one per control point

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \qquad 0 \le i \le n$$

Linear, Cubic, we've seen
What about quadratic?

Stability of Evaluation

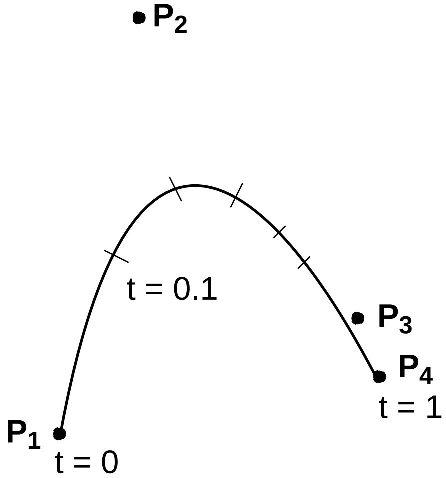
High degree power basis is numerically *unstable*
But... beautiful property of Bézier curves:
Evaluation via nested interpolation!



Works for any degree! (de Casteljau evaluation)
How long does it take?

Quality of Evaluation

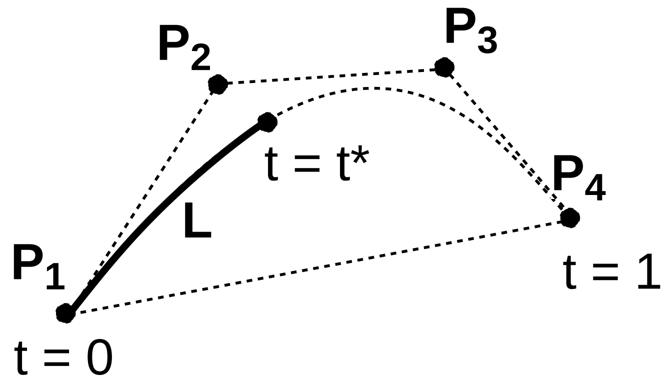
Still, *discretization artifacts* can arise



How can this problem be avoided?

Subdivision:

Split curve segment Q into two segments L, R



Why? How?

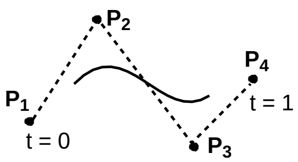
B-Splines (FvDFHP §9.2.4)

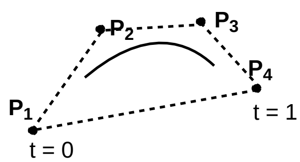
(Sometimes) Undesirable properties of Hermite, Bézier

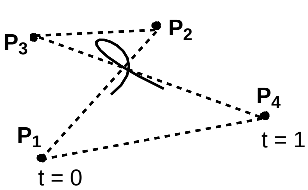
Asymmetry; non-local effect of point manipulation

Using B-splines addresses both issues

Given four control points P_1 .. P_4





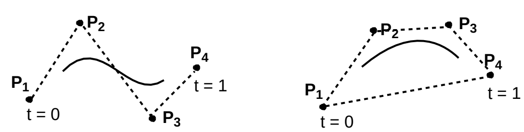


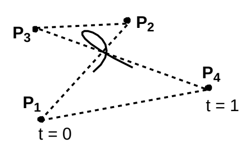
Define $Q(t)$ with different basis functions:

$$Q(t) = \frac{(1-t)^3}{6}P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6}P_{i-2} + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_{i-1} + \frac{t^3}{6}P_i$$

B-Splines

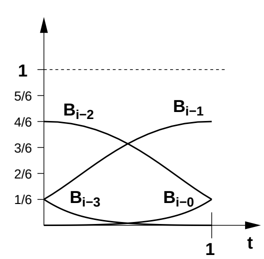
What are $Q(0)$, $Q(1)$? Why would you want to do this?





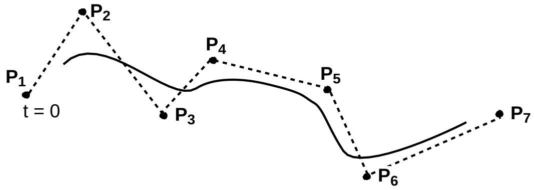
$$Q(t) = \frac{(1-t)^3}{6}P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6}P_{i-2} + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_{i-1} + \frac{t^3}{6}P_{i-0}$$

(in figure, $i = 4$)



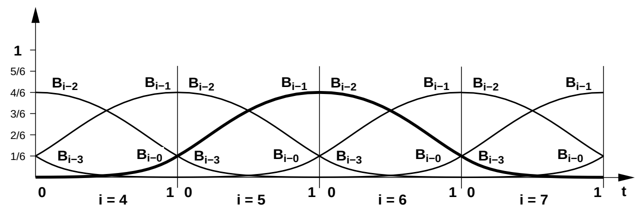
B-Splines

Single segment unremarkable. But *join* segments:



Now notation makes sense:

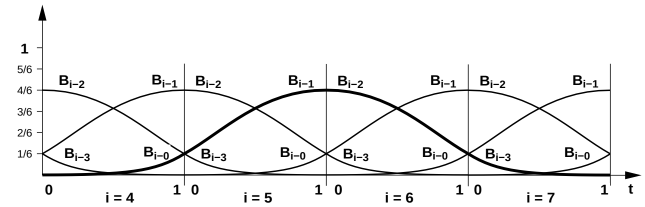
Each interval $0 \leq t \leq 1$ *indexed* by i



[Demo: /mit/imagery4/6.837/ivexamples/cubicbspline.iv]

B-Splines

What do we gain? (Contrast to Bézier)



Symmetry: every point plays the same role!

Smoothness: curve is C^2 everywhere

Local control:

Every point has limited effect on curve

What do we lose?

Spline Bases

Recall compact spline representation

$Q(t) =$ Geometry $\mathbf{G}$ $\cdot$ Spline Basis $\mathbf{B}$ $\cdot$ Power Basis $\mathbf{T(t)}$

But $Q(t)$ expressible as Hermite, Bézier , ...

$Q(t) = \mathbf{G_H} \cdot \mathbf{B_H} \cdot \mathbf{T(t)} = \mathbf{G_B} \cdot \mathbf{B_B} \cdot \mathbf{T(t)}$

What is the relationship between $\mathbf{G_H}$ and $\mathbf{G_B}$?

We can convert among several representations !

Basis Unification

Spline bases:

$$B_{Hermite} = \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}$$

$$B_{Bezier} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

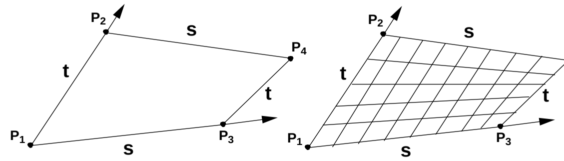
$$B_{B-Spline} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

$$Q(t) = GBT = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix}$$

Parametric Surfaces

What do we "want" from a surface primitive?
- Interpolation of corner points
- Tangent control
- Local control

First attempt: **Bilinear interpolation**
- (analogous to linear interpolation)
- Bi-lerp a (typically non-planar) quadrilateral

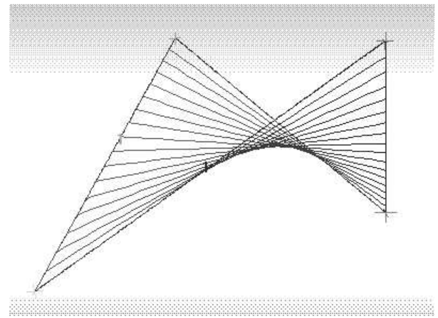


Notation: $\mathbf{L}(P_1, P_2, \alpha) \equiv (1-\alpha)P_1 + \alpha P_2$

$$Q(s,t) = \mathbf{L}(\mathbf{L}(P_1, P_2, t), L(P_3, P_4, t), s)$$

Bilinear Interpolation

How does this surface primitive work?
- Interpolates endpoints
- Tangent control?



What is *implicit degree* of bilinear patch
- (i.e., degree of equivalent polynomial in x, y, z)?

Desire interpolation of control points
- (at least at corners)

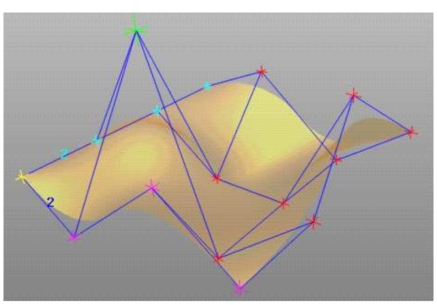
And **independent** control of tangents
- (again, at least at corners)

By inspection, linear, bilinear don't work
- Why don't quadratic or biquadratic patches work?

Bicubic Bézier Patch

Define "Tensor-product" Bézier surface

Notation: $\mathbf{CB}(P_1, P_2, P_3, P_4, \alpha)$ is Bézier curve with control points P_i evaluated at α



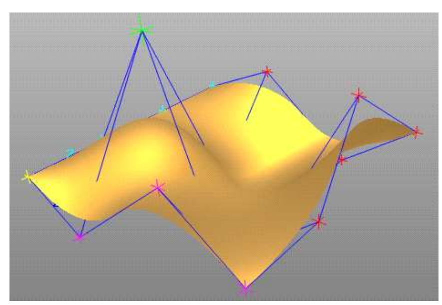
Then define Bézier patch $Q(s,t)$ as

$$Q(s,t) = \mathbf{CB}(\;\begin{array}{l} \mathbf{CB}(P_{00}, P_{01}, P_{02}, P_{03}, t), \\ \mathbf{CB}(P_{10}, P_{11}, P_{12}, P_{13}, t), \\ \mathbf{CB}(P_{20}, P_{21}, P_{22}, P_{23}, t), \\ \mathbf{CB}(P_{30}, P_{31}, P_{32}, P_{33}, t), \\ s) \end{array}$$

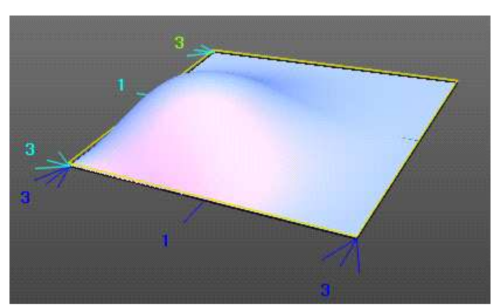
Bicubic Bézier Patch

Can rewrite Bézier patch equation as:

$$\mathbf{Q}(s,t) = \sum B_{ij}(s,t)\mathbf{P}_{ij}$$

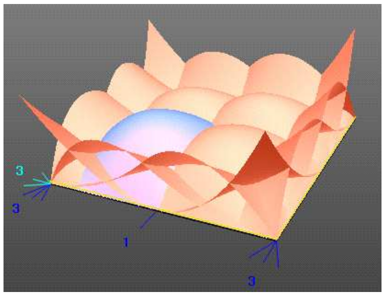


Look at function multiplying **one** control point:

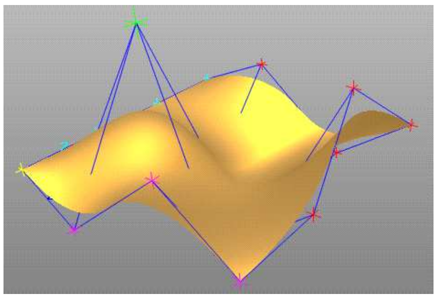


Bézier Basis Functions

What do we know about the basis functions B_{ij}?

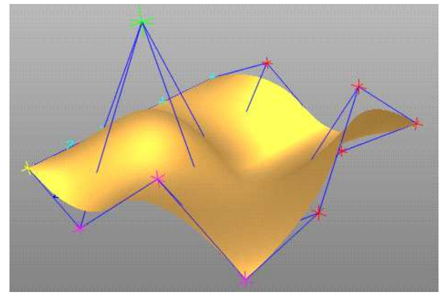


What properties does this imply? (Locality?)



Bézier Patch Properties

Corner interpolation



What is the *normal* to the patch at (s, t)?
Hint: consider curves of constant s or t

What is the *implicit degree* of the patch?
(i.e., degree of equivalent polynomial in x, y, z)?
[Demo: `/mit/6.837/trimnurbs/trimnurbs`]