Parametric Surfaces

Michael Kazhdan

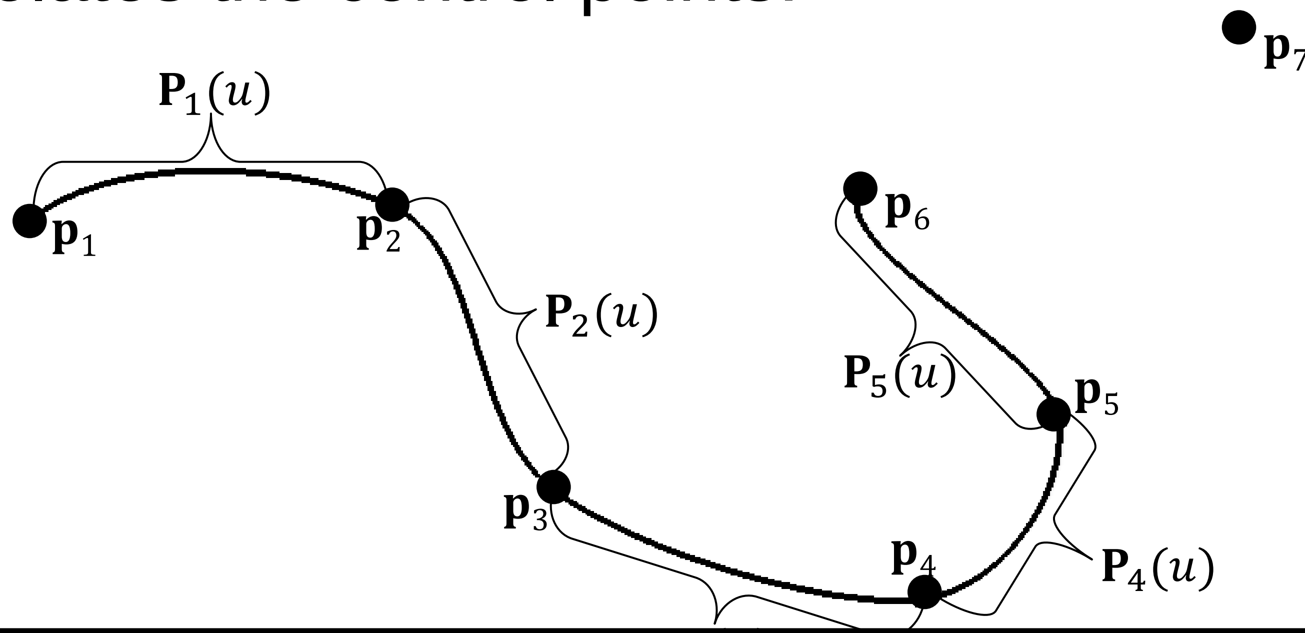
(601.457/657)

Outline

- Spline Surfaces
- Sweep Surfaces

Cubic Splines

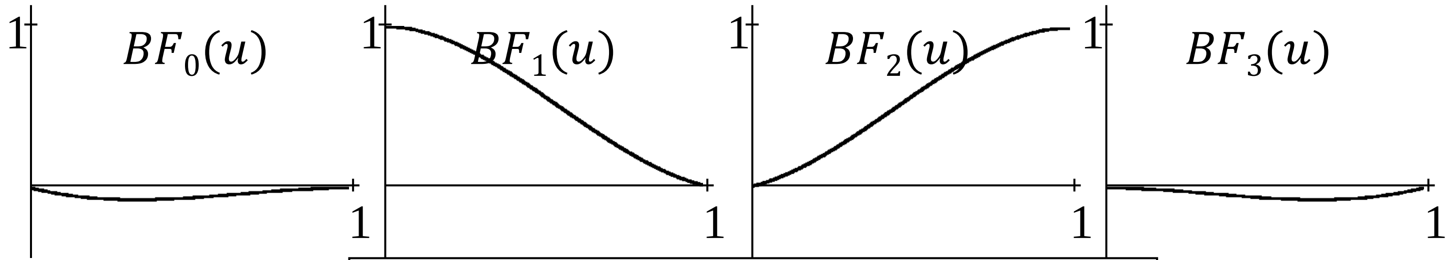
Given $n+1$ control points, $\{\mathbf{p}_0, \ldots, \mathbf{p}_n\}$, we define $n-2$ cubic polynomial functions $\{\mathbf{P}_1(u), \ldots, \mathbf{P}_{n-2}(u)\}$ that jointly describe a curve that approximates / interpolates the control points.



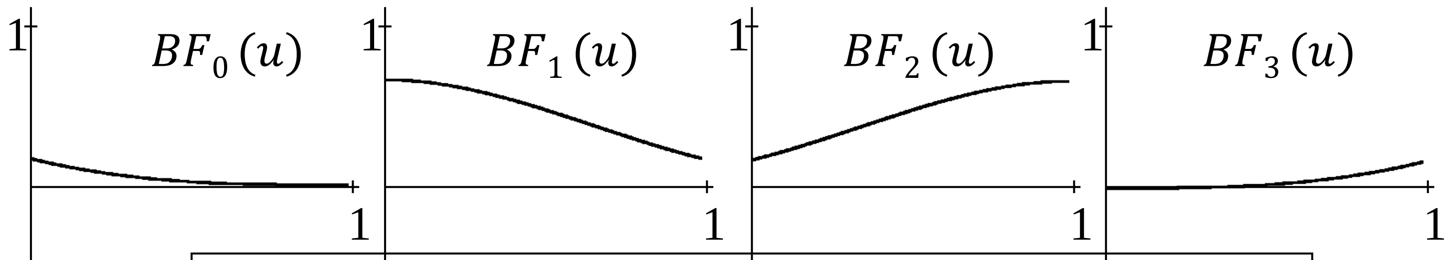
Each cubic function $\mathbf{P}_k(u)$ is defined on the interval $0 \le u \le 1$ and is determined by the points $\mathbf{p}_{k-1}$, $\mathbf{p}_k$, $\mathbf{p}_{k+1}$, and $\mathbf{p}_{k+2}$.

Cubic Blending Functions

Blending functions provide a way for expressing the functions $\mathbf{P}_k(u)$ as a weighted sum of the four control points $\mathbf{p}_{k-1}$, $\mathbf{p}_k$, $\mathbf{p}_{k+1}$, and $\mathbf{p}_{k+2}$:



Cardinal Blending Functions ($s = 1/2$)



Uniform Cubic B-Spline Blending Functions ($s = 1/2$)

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{k-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Blending Functions

For spline curves, we need/want:

- Translation Equivariance:
  $BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$ for all $0 \leq u \leq 1$.
- n-th Order Continuity:
  $$\begin{aligned} 0 &= BF_0^n(1) \\ BF_0^n(0) &= BF_1^n(1) \\ BF_1^n(0) &= BF_2^n(1) \\ BF_2^n(0) &= BF_3^n(1) \\ BF_3^n(0) &= 0 \end{aligned}$$
- Convex Hull Containment:
  $BF_0(u), BF_1(u), BF_2(u), BF_3(u) \geq 0$, for all $0 \leq u \leq 1$.
- Interpolation:
  $$\begin{pmatrix} BF_0(0) \\ BF_1(0) \\ BF_2(0) \\ BF_3(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} BF_0(1) \\ BF_1(1) \\ BF_2(1) \\ BF_3(1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{k-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Overview

From Curves to surfaces

- Spline Curves and Blending Functions
- Weighted Averaging
- Spline Surfaces
- Spline Surface Properties

Weighted Averaging

Suppose we have an array of values:

- $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, and $\mathbf{v}_4$,

and we have weights:

- α_1, α_2, α_3, and α_4, with $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$,
- β_1, β_2, β_3, and β_4, with $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$.

We can express the weighted average of the $\mathbf{v}_i$ in matrix form:

$$\sum_{i=1}^{4} \alpha_i \mathbf{v}_i = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \end{pmatrix} \qquad \sum_{i=1}^{4} \beta_i \mathbf{v}_i = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}$$

Weighted Averaging

If we have a matrix of values:

$$\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}$$

multiplying on the left by $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4)$ gives:

$$(\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4)\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}$$

Weighted Averaging

If we have a matrix of values:

$$\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}$$

multiplying on the left by $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4)$ gives:

$$(\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4)\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix} = \begin{pmatrix} \sum \alpha_i \mathbf{v}_{1i} \\ \sum \alpha_i \mathbf{v}_{2i} \\ \sum \alpha_i \mathbf{v}_{3i} \\ \sum \alpha_i \mathbf{v}_{4i} \end{pmatrix}^{\mathrm{T}}$$

… A row vector whose entries are the weighted average of the matrix's columns.

Weighted Averaging

Similarly, if we have a matrix of values:

$$\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}$$

multiplying on the right by $(\beta_1\ \beta_2\ \beta_3\ \beta_4)^{\mathsf{T}}$ gives:

$$\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} \sum \beta_j \mathbf{v}_{j1} \\ \sum \beta_j \mathbf{v}_{j2} \\ \sum \beta_j \mathbf{v}_{j3} \\ \sum \beta_j \mathbf{v}_{j4} \end{pmatrix}$$

… A column vector with entries that are the weighted average of the matrix's rows.

Weighted Averaging

Simultaneously multiplying on the left by $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4)$ and on the right by $(\beta_1\ \beta_2\ \ \beta_3\ \beta_4)^\top$ gives:

$$(\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4)\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}$$

Weighted Averaging

Simultaneously multiplying on the left by $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4)$ and on the right by $(\beta_1\ \beta_2\ \ \beta_3\ \beta_4)^\top$ gives:

$$(\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4)\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \sum_{i,j=1}^{4} \alpha_j \beta_i \mathbf{v}_{ij}$$

$\Rightarrow$ The weighted sum of the $\mathbf{v}_{ij}$, weighted by $\alpha_j\beta_i$.

<u>Claim</u>: This is a weighted <u>average</u> of the $\mathbf{v}_{ij}$:

To show this, we have to show that the total sum of the weights $\alpha_i\beta_j$ is equal to 1.

Weighted Averaging

Simultaneously multiplying on the left by $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4)$ and on the right by $(\beta_1\ \beta_2\ \ \beta_3\ \beta_4)^\top$ gives:

$$(\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4)\begin{pmatrix} \mathbf{v}_{11} & \mathbf{v}_{21} & \mathbf{v}_{31} & \mathbf{v}_{41} \\ \mathbf{v}_{12} & \mathbf{v}_{22} & \mathbf{v}_{32} & \mathbf{v}_{42} \\ \mathbf{v}_{13} & \mathbf{v}_{23} & \mathbf{v}_{33} & \mathbf{v}_{43} \\ \mathbf{v}_{14} & \mathbf{v}_{24} & \mathbf{v}_{34} & \mathbf{v}_{44} \end{pmatrix}\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \sum_{i,j=1}^{4} \alpha_j \beta_i \mathbf{v}_{ij}$$

$\Rightarrow$ The weighted sum of the $\mathbf{v}_{ij}$, weighted by $\alpha_j\beta_i$.

<u>Claim</u>: This is a weighted <u>average</u> of the $\mathbf{v}_{ij}$:

$$\sum_{i,j=1}^{4} \alpha_i \beta_j = \sum_{i=1}^{4} \alpha_i \left(\sum_{j=1}^{4} \beta_j \right)$$

$$= \sum_{i=1}^{4} \alpha_i = 1$$

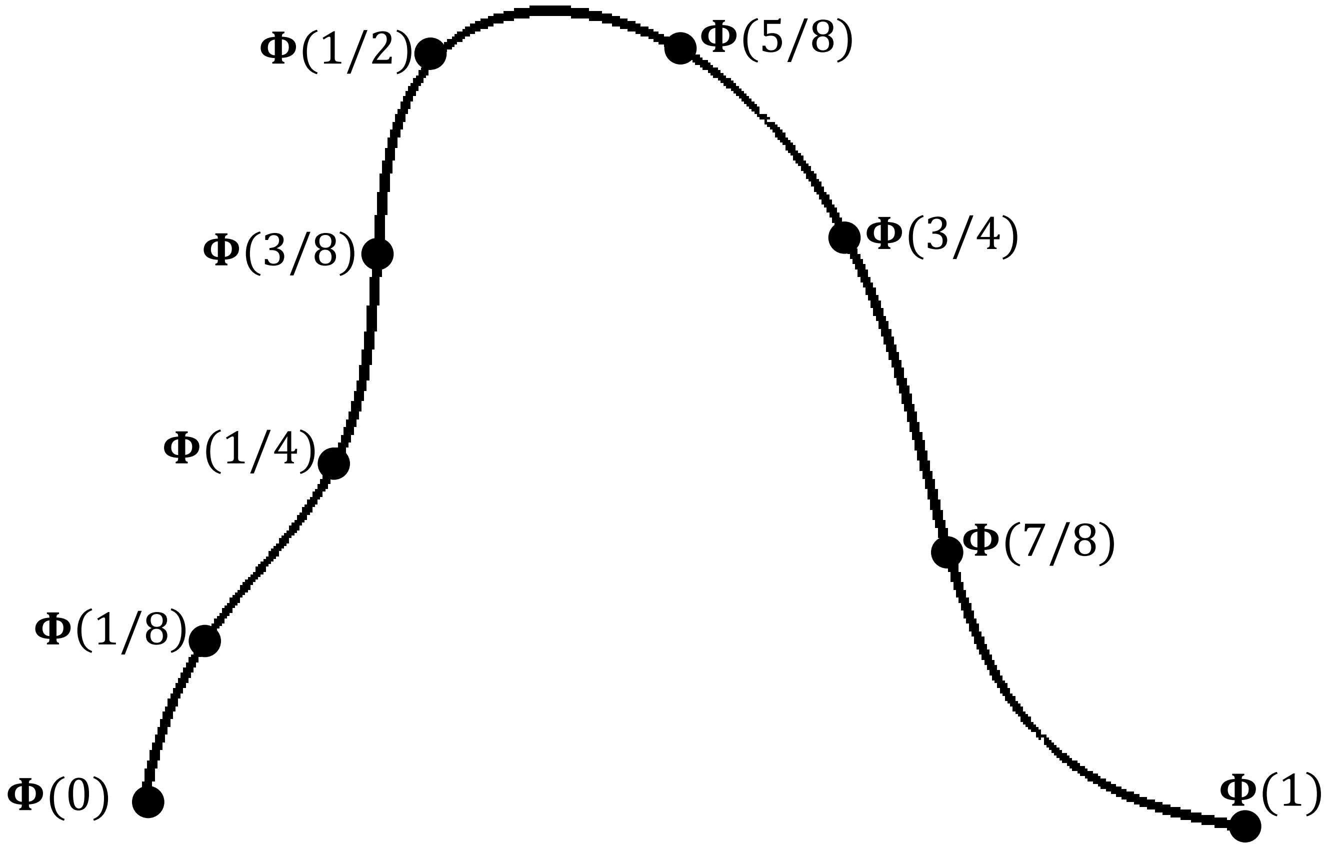
Overview

From Curves to surfaces

- Spline Curves and Blending Functions
- Weighted Averaging
- Spline Surfaces
- Spline Surface Properties

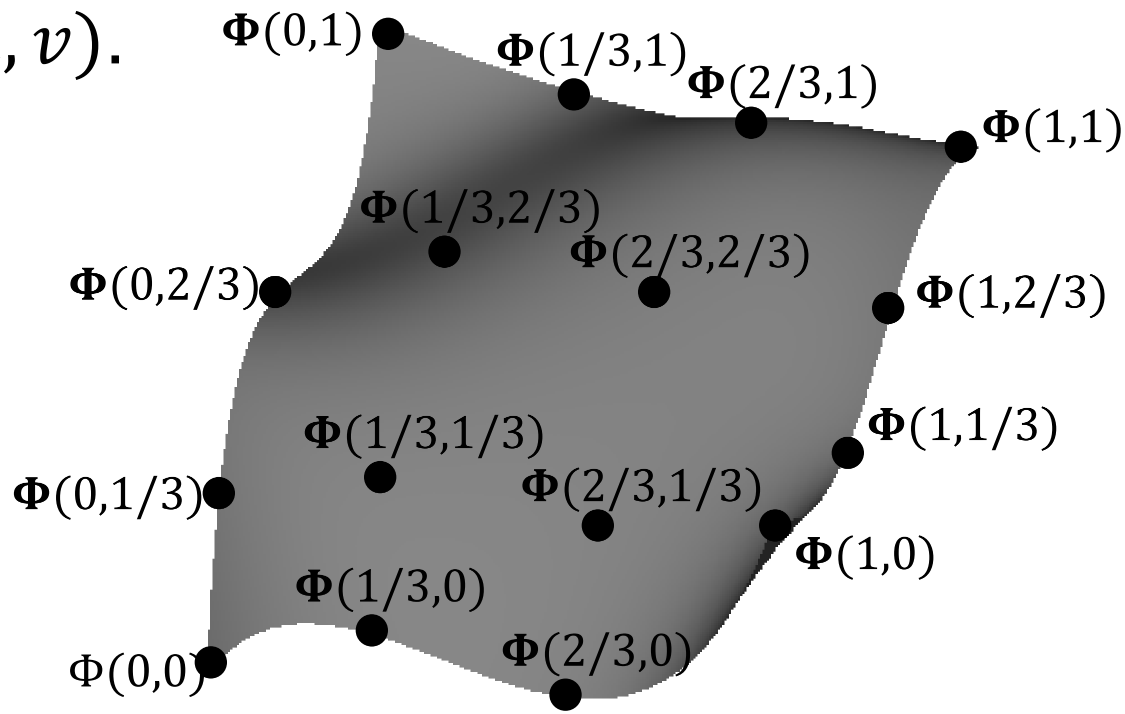
Spline Surfaces

- A parametric curve is a function in one variable $\mathbf{\Phi}(u)$ associating a position to every value of u.



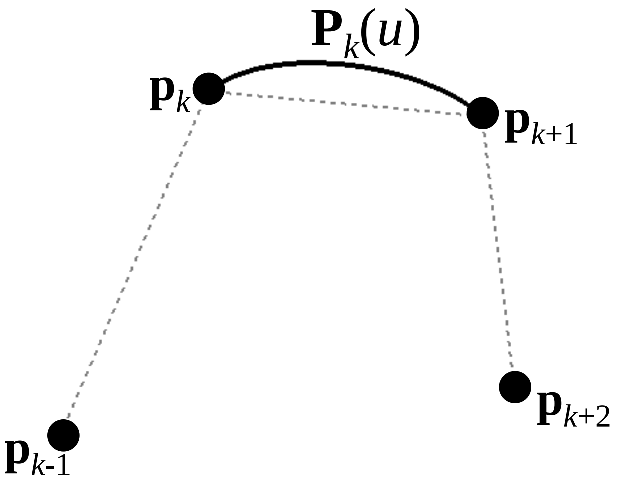
Spline Surfaces

- A parametric curve is a function in one variable $\mathbf{\Phi}(u)$ associating a position to every value of u.
- A parametric patch is a function in two variables $\mathbf{\Phi}(u,v)$ that associates a position to every pair of values of (u,v).



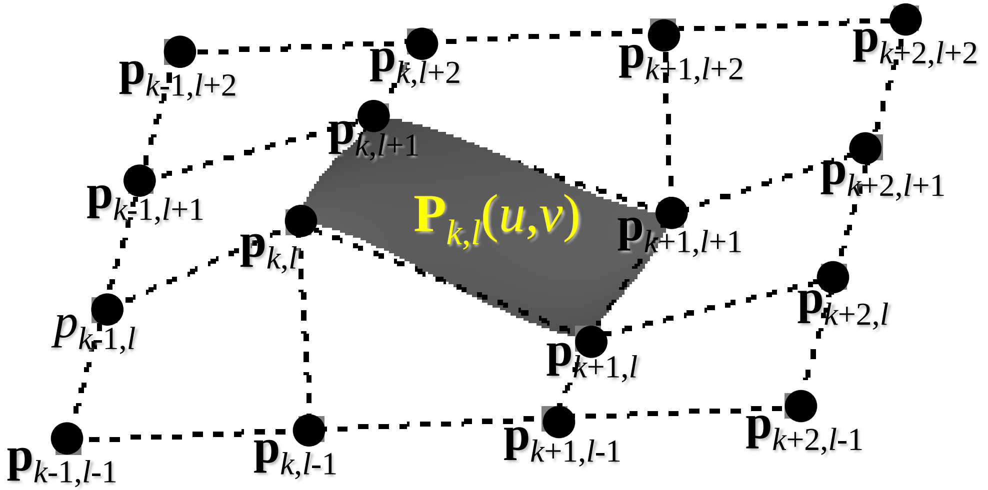
Spline Surfaces

- When considering spline curves, we use four control points to define a cubic polynomial $\mathbf{P}_k(u)$ in one variable ($0 \le u \le 1$).



Spline Surfaces

- When considering spline curves, we use four control points to define a cubic polynomial $\mathbf{P}_k(u)$ in one variable $(0 \leq u \leq 1)$.
- When considering spline surfaces, we use 4×4 control points to define a bi-cubic polynomial $\mathbf{P}_{k,l}(u,v)$ in two variables $(0 \leq u, v \leq 1)$.



Spline Surfaces

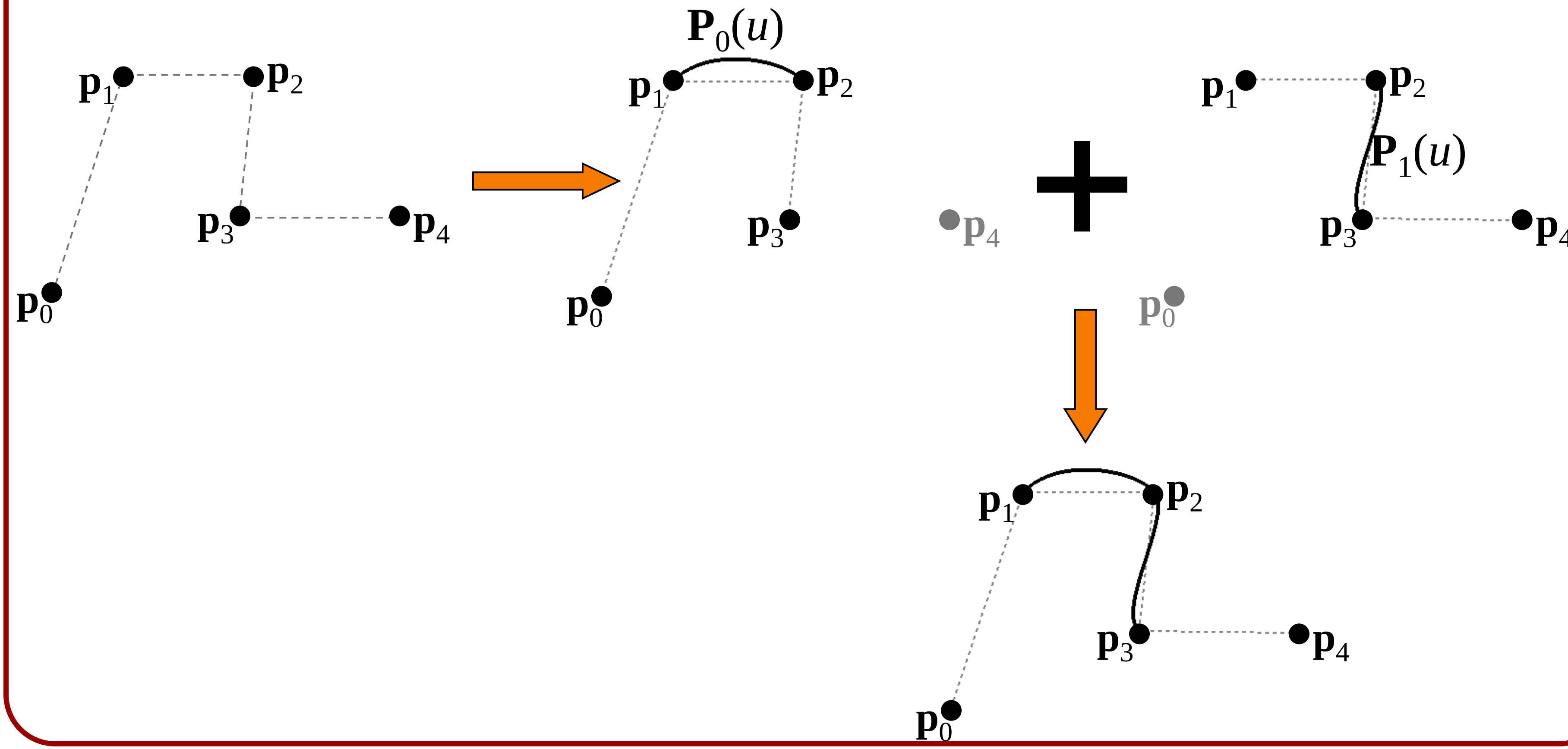
- When considering spline curves, we use four control points to define a cubic polynomial $\mathbf{P}_k(u)$ in one variable $(0 \le u \le 1)$.
- When considering spline surfaces, we use 4×4 control points to define a bi-cubic polynomial $\mathbf{P}_{k,l}(u, v)$ in two variables $(0 \le u, v \le 1)$.

A bi-cubic polynomial is a polynomial which is cubic in each variable:

$$
\begin{aligned}
\mathbf{P}(u,v) = \mathbf{a}u^3v^3 + \\
+\mathbf{b}u^3v^2 + \mathbf{c}u^2v^3 + \\
+\mathbf{d}u^2v^2 + \mathbf{e}u^1v^3 + \mathbf{f}u^3v^1 + \\
+\cdots
\end{aligned}
$$

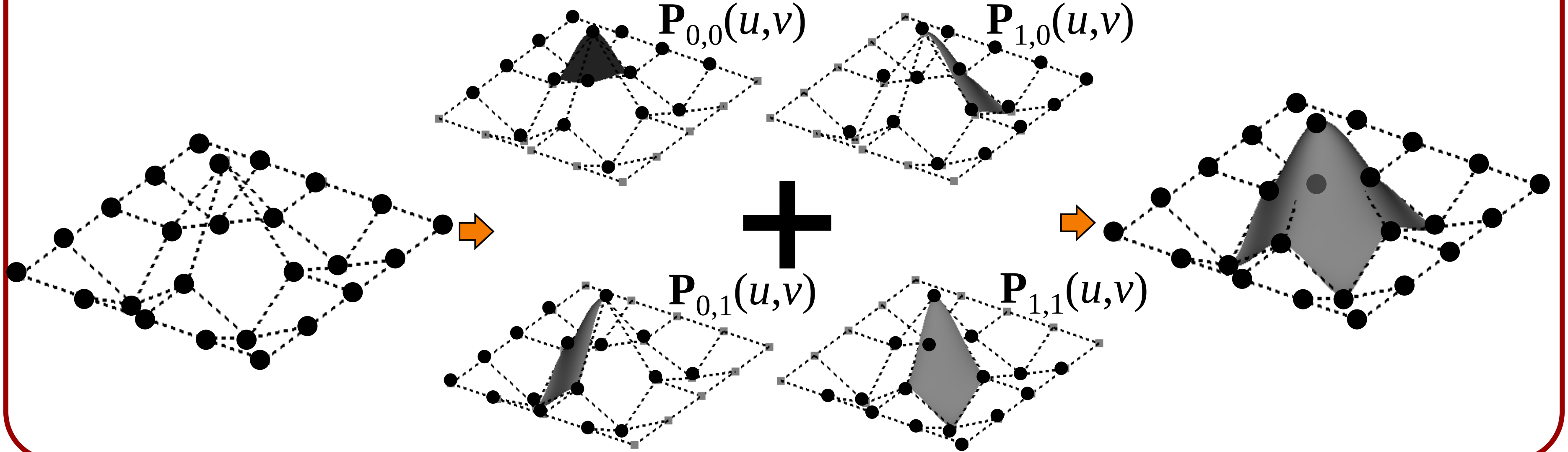
Spline Surfaces

- Given n points, we generate a piecewise cubic curve consisting of $n-3$ segments that approximate/interpolate the points.



Spline Surfaces

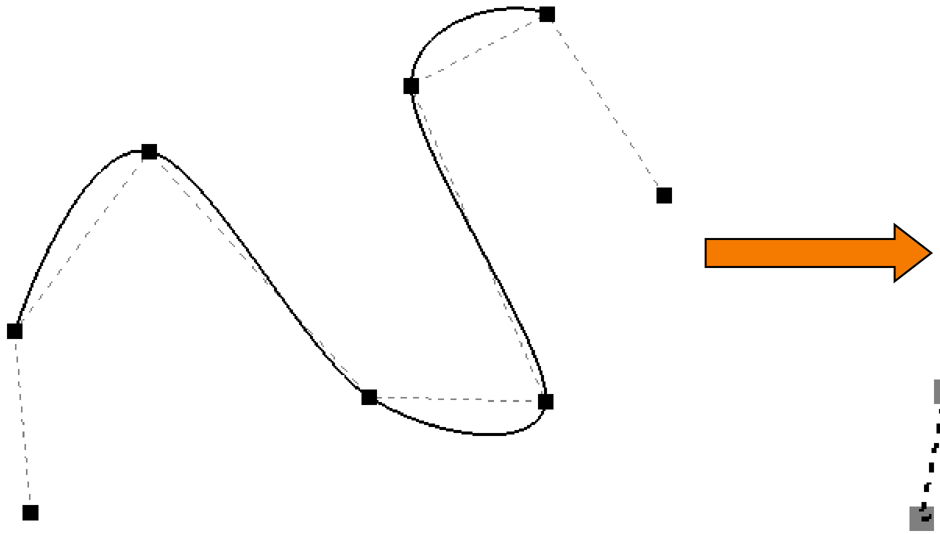
- Given n points, we generate a piecewise cubic curve consisting of $n - 3$ segments that approximate/interpolate the points.
- Given $n \times m$ points, we generate a piecewise bi-cubic surface, consisting of $(n - 3) \times (m - 3)$ patches that approximate/interpolate the points.

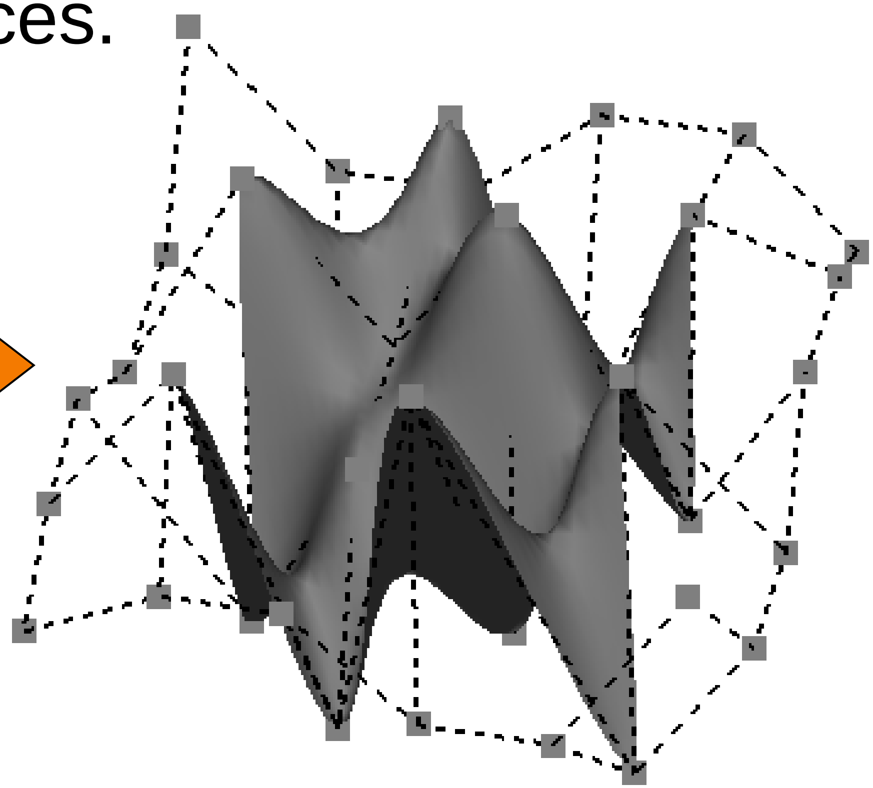


Spline Surfaces

We generate spline curves by using the blending function to compute the weighted average of the control points.

We do the same for surfaces.





Cubic Blending Functions

Recall

For a cubic segment of a spline curve, we can express the spline curve in matrix form as:

$$\mathbf{P}_k(u) = \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

Since the sum of the $BF_i(u)$ equals 1, this is a weighted average of the control points.

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{k-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Cubic Blending Functions

If we are given a 4×4 array of control points, we can define a bi-cubic spline patch similarly:

$$\mathbf{P}_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}$$

Since, the sum of the $BF_i(u)$ equals 1, $\mathbf{P}_{k,l}(u,v)$ is a weighted average of the control points.

Cubic Spline Patches

For example, computing the value of the patch at a point (u_0, v_0) amounts to:

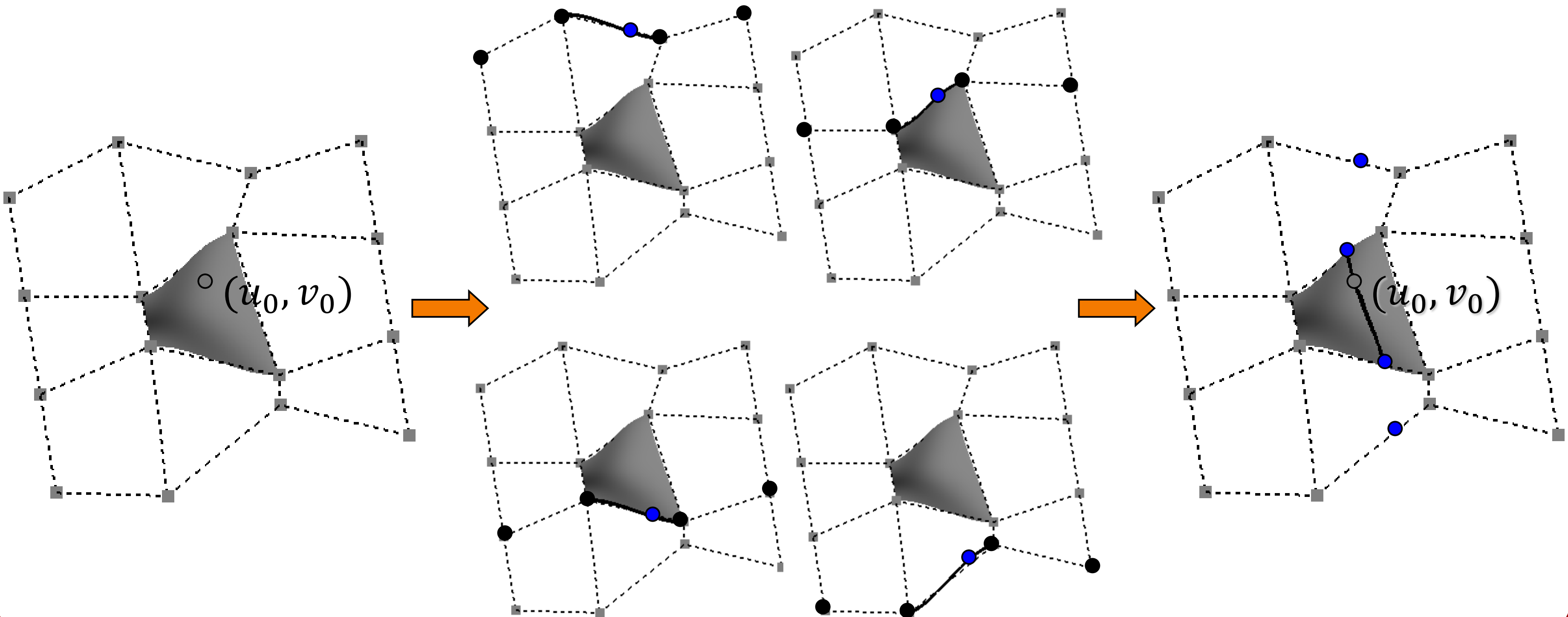
1. Averaging the rows using the weights $BF_i(u_0)$
2. Averaging the result using the weights $BF_i(v_0)$.

$$\mathbf{P}_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\top} \left(\begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix} \right)$$

Cubic Spline Patches

For example, computing the value of the patch at a point (u_0, v_0) amounts to:

1. Averaging the rows using the weights $BF_i(u_0)$
2. Averaging the result using the weights $BF_i(v_0)$.



Cubic Spline Patches

$$\mathbf{P}_{k,l}(u,v)=\begin{pmatrix}BF_0(v)\\BF_1(v)\\BF_2(v)\\BF_3(v)\end{pmatrix}^{\top}\begin{pmatrix}\mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1}\\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l}\\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1}\\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2}\end{pmatrix}\begin{pmatrix}BF_0(u)\\BF_1(u)\\BF_2(u)\\BF_3(u)\end{pmatrix}$$

Multiplying out the matrices we get:

$$\begin{aligned}\mathbf{P}_{k,l}(u,v) &= BF_0(u)\cdot BF_0(v)\cdot \mathbf{p}_{k-1,l-1} + BF_1(u)\cdot BF_0(v)\cdot \mathbf{p}_{k,l-1} + \cdots\\ &\quad + BF_0(u)\cdot BF_1(v)\cdot \mathbf{p}_{k-1,l} + BF_1(u)\cdot BF_1(v)\cdot \mathbf{p}_{k,l} + \cdots\\ &\quad + \cdots\end{aligned}$$

Or, if we set $BF_{i,j}(u,v)=BF_i(u)\cdot BF_j(v)$ we get:

$$\begin{aligned}\mathbf{P}_{k,l}(u,v) &= BF_{0,0}(u,v)\cdot \mathbf{p}_{k-1,l-1} + BF_{1,0}(u,v)\cdot \mathbf{p}_{k,l-1} + \cdots\\ &\quad + BF_{0,1}(u,v)\cdot \mathbf{p}_{k-1,l} + BF_{1,1}(u,v)\cdot \mathbf{p}_{k,l} + \cdots\\ &\quad + \cdots\end{aligned}$$

Cubic Spline Patches

$$\mathbf{P}_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}$$

Multiplying out the matrices we get:

$$\begin{aligned} \mathbf{P}_{k,l}(u,v) = & \; BF_0(u) \cdot BF_0(v) \cdot \mathbf{p}_{k-1,l-1} + BF_1(u) \cdot BF_0(v) \cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_0(u) \cdot BF_1(v) \cdot \mathbf{p}_{k-1,l} + BF_1(u) \cdot BF_1(v) \cdot \mathbf{p}_{k,l} + \cdots \\ & + \cdots \end{aligned}$$

Or, if we set $BF_{i,j}(u,v) = BF_i(u) \cdot BF_j(v)$ we get:

$$\begin{aligned} \mathbf{P}_{k,l}(u,v) = & \; BF_{0,0}(u,v) \cdot \mathbf{p}_{k-1,l-1} + BF_{1,0}(u,v) \cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_{0,1}(u,v) \cdot \mathbf{p}_{k-1,l} + BF_{1,1}(u,v) \cdot \mathbf{p}_{k,l} + \cdots \\ & + \cdots \end{aligned}$$

Cubic Spline Patches

$$\mathbf{P}_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}$$

Multiplying out the matrices we get:

$$\begin{aligned}\mathbf{P}_{k,l}(u,v) = {} & BF_0(u)\cdot BF_0(v)\cdot \mathbf{p}_{k-1,l-1} + BF_1(u)\cdot BF_0(v)\cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_0(u)\cdot BF_1(v)\cdot \mathbf{p}_{k-1,l} + BF_1(u)\cdot BF_1(v)\cdot \mathbf{p}_{k,l} + \cdots \\ & + \cdots\end{aligned}$$

Or, if we set $BF_{i,j}(u,v) = BF_i(u)\cdot BF_j(v)$ we get:

$$\begin{aligned}\mathbf{P}_{k,l}(u,v) = {} & BF_{0,0}(u,v)\cdot \mathbf{p}_{k-1,l-1} + BF_{1,0}(u,v)\cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_{0,1}(u,v)\cdot \mathbf{p}_{k-1,l} + BF_{1,1}(u,v)\cdot \mathbf{p}_{k,l} + \cdots \\ & + \cdots\end{aligned}$$

Cubic Spline Patches

$$\mathbf{P}_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}$$

Multiplying out the matrices we get:

$$\begin{aligned} \mathbf{P}_{k,l}(u,v) = {} & BF_0(u)\cdot BF_0(v)\cdot \mathbf{p}_{k-1,l-1} + BF_1(u)\cdot BF_0(v)\cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_0(u)\cdot BF_1(v)\cdot \mathbf{p}_{k-1,l} + BF_1(u)\cdot BF_1(v)\cdot \mathbf{p}_{k,l} + \cdots \\ & + \cdots \end{aligned}$$

Or, if we set $BF_{i,j}(u,v) = BF_i(u)\cdot BF_j(v)$ we get:

$$\begin{aligned} \mathbf{P}_{k,l}(u,v) = {} & BF_{0,0}(u,v)\cdot \mathbf{p}_{k-1,l-1} + BF_{1,0}(u,v)\cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_{0,1}(u,v)\cdot \mathbf{p}_{k-1,l} + BF_{1,1}(u,v)\cdot \mathbf{p}_{k,l} + \cdots \\ & + \cdots \end{aligned}$$

Cubic Spline Patches

$$\mathbf{P}_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}$$

Multiplying out the matrices we get:

$$\begin{aligned} \mathbf{P}_{k,l}(u,v) = & BF_0(u)\cdot BF_0(v)\cdot \mathbf{p}_{k-1,l-1} + BF_1(u)\cdot BF_0(v)\cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_0(u)\cdot BF_1(v)\cdot \mathbf{p}_{k-1,l} + \boxed{BF_1(u)\cdot BF_1(v)\cdot \mathbf{p}_{k,l}} + \cdots \\ & + \cdots \end{aligned}$$

Or, if we set $BF_{i,j}(u,v) = BF_i(u)\cdot BF_j(v)$ we get:

$$\begin{aligned} \mathbf{P}_{k,l}(u,v) = & BF_{0,0}(u,v)\cdot \mathbf{p}_{k-1,l-1} + BF_{1,0}(u,v)\cdot \mathbf{p}_{k,l-1} + \cdots \\ & + BF_{0,1}(u,v)\cdot \mathbf{p}_{k-1,l} + \boxed{BF_{1,1}(u,v)\cdot \mathbf{p}_{k,l}} + \cdots \\ & + \cdots \end{aligned}$$

Cubic Spline Patches

$$P_{k,l}(u,v) = \begin{pmatrix} BF_0(v) \\ BF_1(v) \\ BF_2(v) \\ BF_3(v) \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \begin{pmatrix} BF_0(u) \\ BF_1(u) \\ BF_2(u) \\ BF_3(u) \end{pmatrix}$$

Recall that we can write out blending functions as:

$$(BF_0(u) \quad BF_1(u) \quad BF_2(u) \quad BF_3(u)) = \mathbf{M}_{\text{Spline}} U$$

with $U^{\top} = (u^3 \; u^2 \; u \;\; 1)$ and $\mathbf{M}_{\text{Spline}}$ the spline matrix.

This gives:

$$P_{k,l}(u,v) = V^{\top} \mathbf{M}_{\text{Spline}}^{\top} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \mathbf{M}_{\text{Spline}} U$$

with $V^{\top} = (v^3 \; v^2 \; v \; 1)$.

Cubic Spline Patches

$$\mathbf{P}_{k,l}(u,v)=\begin{pmatrix}BF_0(v)\\BF_1(v)\\BF_2(v)\\BF_3(v)\end{pmatrix}^\top\begin{pmatrix}\mathbf{p}_{k-1,l-1}&\mathbf{p}_{k,l-1}&\mathbf{p}_{k+1,l-1}&\mathbf{p}_{k+2,l-1}\\\mathbf{p}_{k-1,l}&\mathbf{p}_{k,l}&\mathbf{p}_{k+1,l}&\mathbf{p}_{k+2,l}\\\mathbf{p}_{k-1,l+1}&\mathbf{p}_{k,l+1}&\mathbf{p}_{k+1,l+1}&\mathbf{p}_{k+2,l+1}\\\mathbf{p}_{k-1,l+2}&\mathbf{p}_{k,l+2}&\mathbf{p}_{k+1,l+2}&\mathbf{p}_{k+2,l+2}\end{pmatrix}\begin{pmatrix}BF_0(u)\\BF_1(u)\\BF_2(u)\\BF_3(u)\end{pmatrix}$$

Recall that we can write out blending functions as:

$$(BF_0(u)\quad BF_1(u)\quad BF_2(u)\quad BF_3(u))=\mathbf{M}_{\text{Spline}}U$$

Surface splines that are obtained from curve splines in this way are referred to as tensor product splines.

$$\mathbf{P}_{k,l}(u,v)=V^\top\mathbf{M}_{\text{Spline}}^\top\begin{pmatrix}\mathbf{p}_{k-1,l-1}&\mathbf{p}_{k,l-1}&\mathbf{p}_{k+1,l-1}&\mathbf{p}_{k+2,l-1}\\\mathbf{p}_{k-1,l}&\mathbf{p}_{k,l}&\mathbf{p}_{k+1,l}&\mathbf{p}_{k+2,l}\\\mathbf{p}_{k-1,l+1}&\mathbf{p}_{k,l+1}&\mathbf{p}_{k+1,l+1}&\mathbf{p}_{k+2,l+1}\\\mathbf{p}_{k-1,l+2}&\mathbf{p}_{k,l+2}&\mathbf{p}_{k+1,l+2}&\mathbf{p}_{k+2,l+2}\end{pmatrix}\mathbf{M}_{\text{Spline}}U$$

with $V^\top=(v^3\ v^2\ v\ 1)$.

Cubic Spline Patches

We can choose our favorite spline curve (Cardinal, uniform cubic-B, etc.) and use its blending functions to define a spline patch:

$$\mathbf{P}_{k,l}(u,v) = V^\top \boxed{\mathbf{M}_{\text{Spline}}^\top} \begin{pmatrix} \mathbf{p}_{k-1,l-1} & \mathbf{p}_{k,l-1} & \mathbf{p}_{k+1,l-1} & \mathbf{p}_{k+2,l-1} \\ \mathbf{p}_{k-1,l} & \mathbf{p}_{k,l} & \mathbf{p}_{k+1,l} & \mathbf{p}_{k+2,l} \\ \mathbf{p}_{k-1,l+1} & \mathbf{p}_{k,l+1} & \mathbf{p}_{k+1,l+1} & \mathbf{p}_{k+2,l+1} \\ \mathbf{p}_{k-1,l+2} & \mathbf{p}_{k,l+2} & \mathbf{p}_{k+1,l+2} & \mathbf{p}_{k+2,l+2} \end{pmatrix} \boxed{\mathbf{M}_{\text{Spline}}} U$$

Overview

From Curves to surfaces

- Spline Curves and Blending Functions
- Weighted Averaging
- Spline Surfaces
- Spline Surface Properties

Blending Functions

For spline curves, we want:

- Translation Equivariance:

  $BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$ for all $0 \leq u \leq 1$.

- n-th Order Continuity:

$$
\begin{aligned}
0 &= BF_0^n(1) \\
BF_0^n(0) &= BF_1^n(1) \\
BF_1^n(0) &= BF_2^n(1) \\
BF_2^n(0) &= BF_3^n(1) \\
BF_3^n(0) &= 0
\end{aligned}
$$

- Convex Hull Containment:

  $BF_0(u), BF_1(u), BF_2(u), BF_3(u) \geq 0$, for all $0 \leq u \leq 1$.

- Interpolation:

$$
\begin{matrix} BF_0(0) \\ BF_1(0) \\ BF_2(0) \\ BF_3(0) \end{matrix} = \begin{matrix} 0 \\ 1 \\ 0 \\ 0 \end{matrix} \quad \text{and} \quad \begin{matrix} BF_0(1) \\ BF_1(1) \\ BF_2(1) \\ BF_3(1) \end{matrix} = \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix}
$$

Do tensor product splines satisfy these conditions?

Surface Spline Properties

- Translation equivariance:
  - As in the curve case, we need the sum of the blending functions $BF_{i,j}(u,v)$ to be equal to one.
  - But since
$$BF_{i,j}(u,v) = BF_i(u) \cdot BF_j(v)$$
if the $BF_i(u)$ are weighting functions that sum to 1, then the tensor product functions $BF_{i,j}(u,v)$ also sum to 1.

Surface Spline Properties

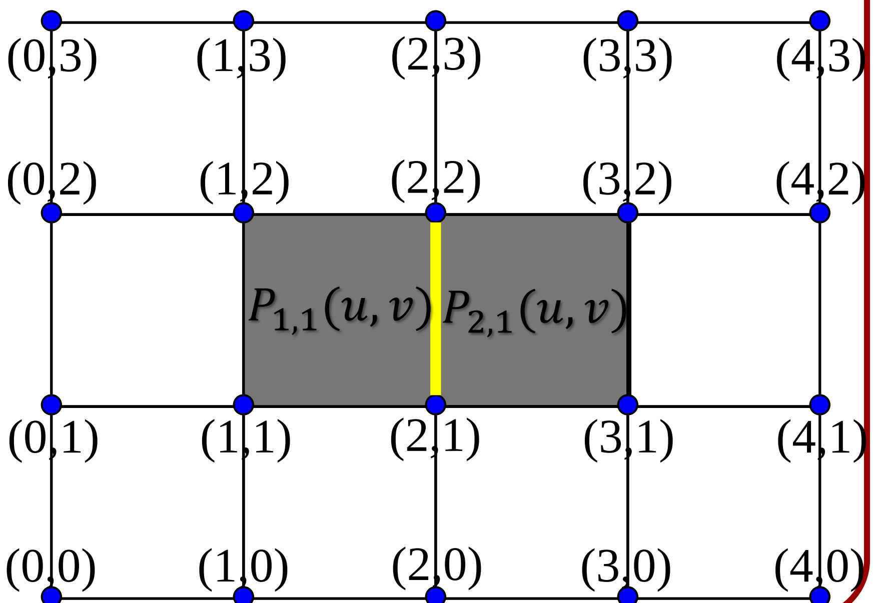
- Continuity:
  - W.L.O.G. consider continuity along the yellow edge:

$$0 = \mathbf{P}_{1,1}(1,v) - \mathbf{P}_{2,1}(0,v) \qquad \forall 0 \le v \le 1$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot \mathbf{p}_{i,j} - \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(0,v)\cdot \mathbf{p}_{i+1,j}$$

Re-index the second term so that the control point indices match.



Surface Spline Properties

- <u>Continuity</u>:
  - W.L.O.G. consider continuity along the yellow edge:

$$0 = \mathbf{P}_{1,1}(1,v) - \mathbf{P}_{2,1}(0,v) \qquad \forall 0 \le v \le 1$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot\mathbf{p}_{i,j} - \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(0,v)\cdot\mathbf{p}_{i+1,j}$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot\mathbf{p}_{i,j} - \sum_{i=1}^{4}\sum_{j=0}^{3} B_{i-1,j}(0,v)\cdot\mathbf{p}_{i,j}$$

Decompose the equation in terms of the control points shared by both patches.
$\mathbf{p}_{i,j}$ w/ $1 \le i \le 3$ and $0 \le j \le 3$

Surface Spline Properties

- Continuity:
  - W.L.O.G. consider continuity along the yellow edge:

$$0 = \mathbf{P}_{1,1}(1,v) - \mathbf{P}_{2,1}(0,v) \qquad \forall 0 \le v \le 1$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot \mathbf{p}_{i,j} - \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(0,v)\cdot \mathbf{p}_{i+1,j}$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot \mathbf{p}_{i,j} - \sum_{i=1}^{4}\sum_{j=0}^{3} B_{i-1,j}(0,v)\cdot \mathbf{p}_{i,j}$$

$$\Downarrow$$

$$0 = \sum_{j=0}^{3} B_{0,j}(1,v)\cdot \mathbf{p}_{0,j} + \sum_{i=1}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot \mathbf{p}_{i,j} - \sum_{i=1}^{3}\sum_{j=0}^{3} B_{i-1,j}(0,v)\cdot \mathbf{p}_{i,j} - \sum_{j=0}^{3} B_{3,j}(0,v)\cdot \mathbf{p}_{4,j}$$

Combine terms using the same control points.

Surface Spline Properties

- Continuity:
  - W.L.O.G. consider continuity along the yellow edge:

$$0 = \mathbf{P}_{1,1}(1,v) - \mathbf{P}_{2,1}(0,v) \qquad \forall 0 \le v \le 1$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot\mathbf{p}_{i,j} - \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(0,v)\cdot\mathbf{p}_{i+1,j}$$

$$\Downarrow$$

$$0 = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot\mathbf{p}_{i,j} - \sum_{i=1}^{4}\sum_{j=0}^{3} B_{i-1,j}(0,v)\cdot\mathbf{p}_{i,j}$$

$$\Downarrow$$

$$0 = \sum_{j=0}^{3} B_{0,j}(1,v)\cdot\mathbf{p}_{0,j} + \sum_{i=1}^{3}\sum_{j=0}^{3} B_{i,j}(1,v)\cdot\mathbf{p}_{i,j} - \sum_{i=1}^{3}\sum_{j=0}^{3} B_{i-1,j}(0,v)\cdot\mathbf{p}_{i,j} - \sum_{j=0}^{3} B_{3,j}(0,v)\cdot\mathbf{p}_{4,j}$$

$$\Downarrow$$

$$0 = \sum_{j=0}^{3} B_{0,j}(1,v)\cdot\mathbf{p}_{0,j} + \sum_{i=1}^{3}\sum_{j=0}^{3} \left(B_{i,j}(1,v) - B_{i-1,j}(0,v)\right)\cdot\mathbf{p}_{i,j} - \sum_{j=0}^{3} B_{3,j}(0,v)\cdot\mathbf{p}_{4,j}$$

Surface Spline Properties

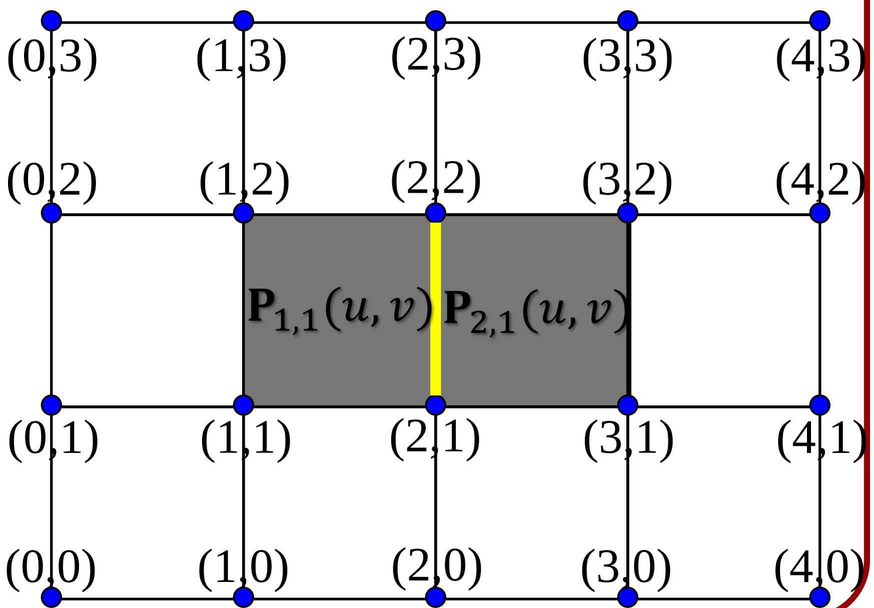
- <u>Continuity</u>:
  - W.L.O.G. consider continuity along the yellow edge:

$$0 = \sum_{j=0}^{3} B_{0,j}(1,v)\cdot \mathbf{p}_{0,j} + \sum_{i=1}^{3}\sum_{j=0}^{3}\left(B_{i,j}(1,v) - B_{i-1,j}(0,v)\right)\cdot \mathbf{p}_{i,j} - \sum_{j=0}^{3} B_{3,j}(0,v)\cdot \mathbf{p}_{4,j}$$

For this to be true for all control points $\mathbf{p}_{ij}$, we need:

- $B_{0,j}(1,v) = B_{3,j}(0,v) = 0$
- $B_{1,j}(1,v) = B_{0,j}(0,v)$
- $B_{2,j}(1,v) = B_{1,j}(0,v)$
- $B_{3,j}(1,v) = B_{2,j}(0,v)$

for all $v \in [0,1]$



Surface Spline Properties

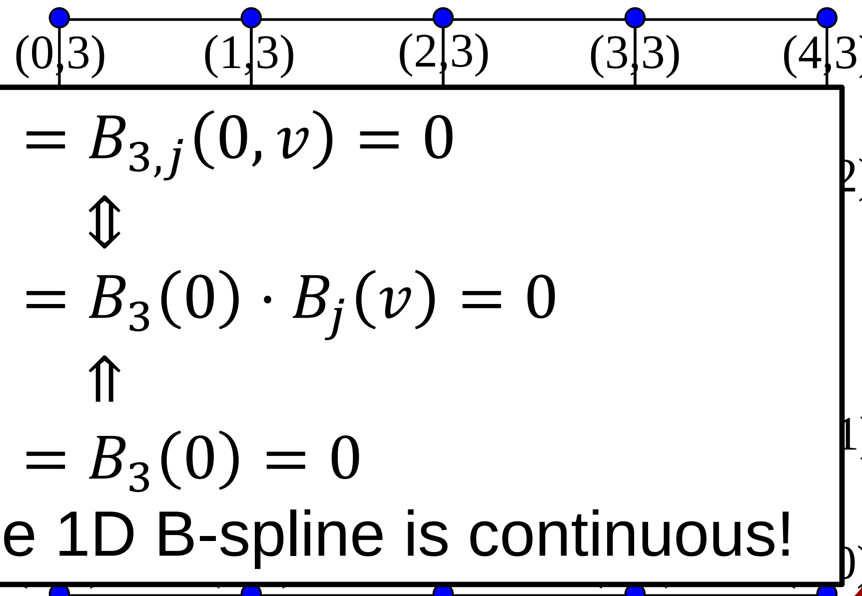
- Continuity:
  - W.L.O.G. consider continuity along the yellow edge:

$$0=\sum_{j=0}^{3} B_{0,j}(1,v)\cdot\mathbf{p}_{0,j}+\sum_{i=1}^{3}\sum_{j=0}^{3}\left(B_{i,j}(1,v)-B_{i-1,j}(0,v)\right)\cdot\mathbf{p}_{i,j}-\sum_{j=0}^{3} B_{3,j}(0,v)\cdot\mathbf{p}_{4,j}$$

For this to be true for all control points $\mathbf{p}_{ij}$, we need:

  - $B_{0,j}(1,v)=B_{3,j}(0,v)=0$
  - $B_{1,j}(1,v)=B_{0,j}(0,v)$
  - $B_{2,j}$
  - $B_{3,j}$

for all



$$B_{0,j}(1,v)=B_{3,j}(0,v)=0$$
$$\Updownarrow$$
$$B_0(1)\cdot B_j(v)=B_3(0)\cdot B_j(v)=0$$
$$\Uparrow$$
$$B_0(1)=B_3(0)=0$$

Which is satisfied if the 1D B-spline is continuous!

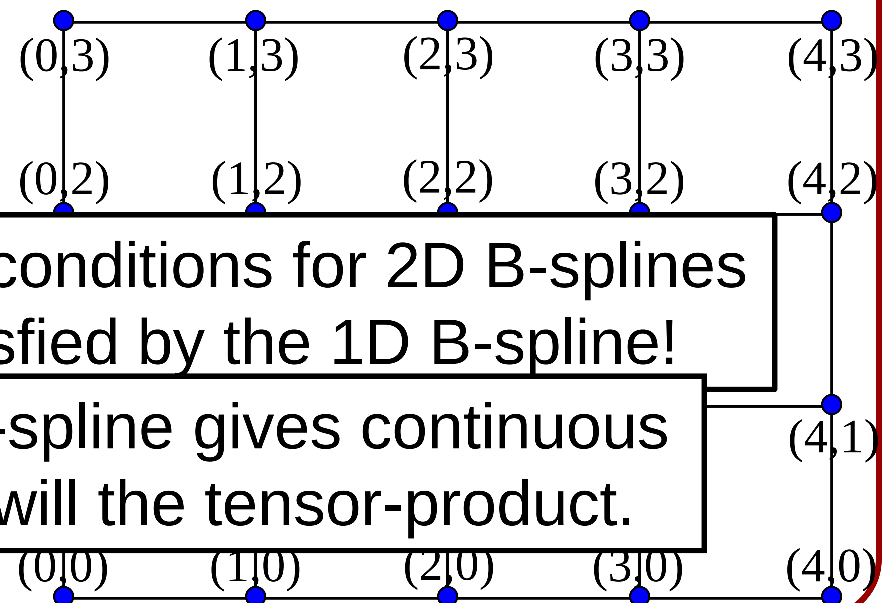
Surface Spline Properties

- <u>Continuity</u>:
  - W.L.O.G. consider continuity along the yellow edge:

$$0 = \sum_{j=0}^{3} B_{0,j}(1,v) \cdot \mathbf{p}_{0,j} + \sum_{i=1}^{3}\sum_{j=0}^{3}\left(B_{i,j}(1,v) - B_{i-1,j}(0,v)\right) \cdot \mathbf{p}_{i,j} - \sum_{j=0}^{3} B_{3,j}(0,v) \cdot \mathbf{p}_{4,j}$$

For this to be true for all control points $\mathbf{p}_{ij}$, we need:

- $B_{0,j}(1,v) = B_{3,j}(0,v) = 0$
- $B_{1,j}(1,v) = B_{0,j}(0,v)$
- $B_{2,j}(1,v) = B_{1,j}(0,v)$
- $B_{3,j}(1,v) = B_{2,j}(0,v)$



Similarly, the other continuity conditions for 2D B-splines are satisfied if they are satisfied by the 1D B-spline!

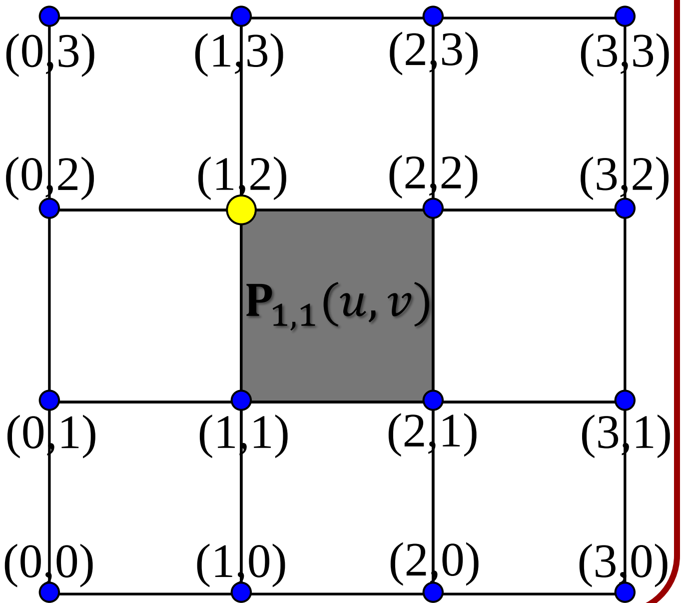
More generally, if the 1D B-spline gives continuous n-th order derivatives, so will the tensor-product.

Surface Spline Properties

- Convex hull containment:
  - For convex hull containment we need the weights of the blending function to be non-negative.
  - If the $BF_i(u)$ are non-negative, then since
    $$BF_{i,j}(u,v) = BF_i(u) \cdot BF_j(v)$$
    the $BF_{i,j}(u,v)$ will also be non-negative.

Surface Spline Properties

- <u>Interpolation</u>:
  - For the spline surface to interpolate, it must satisfy:
    - » $BF_{1,1}(0,0) = BF_{1,2}(0,1) = BF_{2,1}(1,0) = BF_{2,2}(1,1) = 1.$
    - » All the other blending functions evaluate to 0 at the end-points.
  - The spline curve is interpolating if:
    - » $BF_0(0) = BF_2(0) = BF_3(0) = 0$
    - » $BF_0(1) = BF_1(1) = BF_3(1) = 0$
    - » $BF_1(0) = 1$
    - » $BF_2(1) = 1$



$$BF_{1,2}(0,1) = BF_1(0) \cdot BF_2(1)$$

Surface Spline Properties

We began by describing some of the properties that we would like spline curves to satisfy:

- Translation equivariance
- Continuity
- Convex hull containment
- Interpolation

If the curve spline satisfies these properties, then so will the tensor product spline!

Surface Spline Properties

We began by describing some of the properties that we would like spline curves to satisfy:

- Translation equivariance
- Continuity
- Convex hull containment
- Interpolation

As with curves, we can handle boundaries by:

- Ignoring them
- Doubling up
- Introducing cylindrical/toroidal periodicity

If the … hen so will th

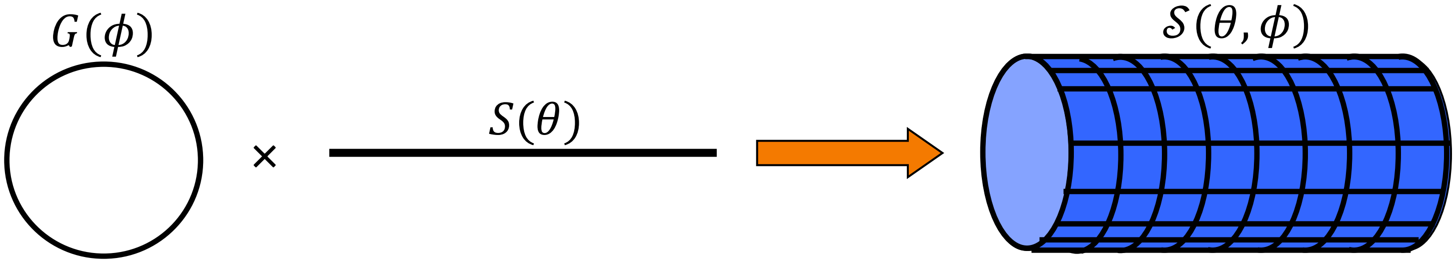
Surface Spline Demo

Outline

- Spline Surfaces
- Sweep Surfaces

Sweeps

Given a 3D *sweep curve* $S(\theta)$ and a 2D *generating curve* $G(\phi)$, define the *sweep surface* $\mathcal{S}(\theta,\phi)$ as the sweep of C along H:



In this example, the sweep curve is used to translate the generating curve:

$$\mathcal{S}(\theta,\phi) = S(\theta) + G(\phi)$$

We can define more complex sweep surfaces.

Example: Seashells

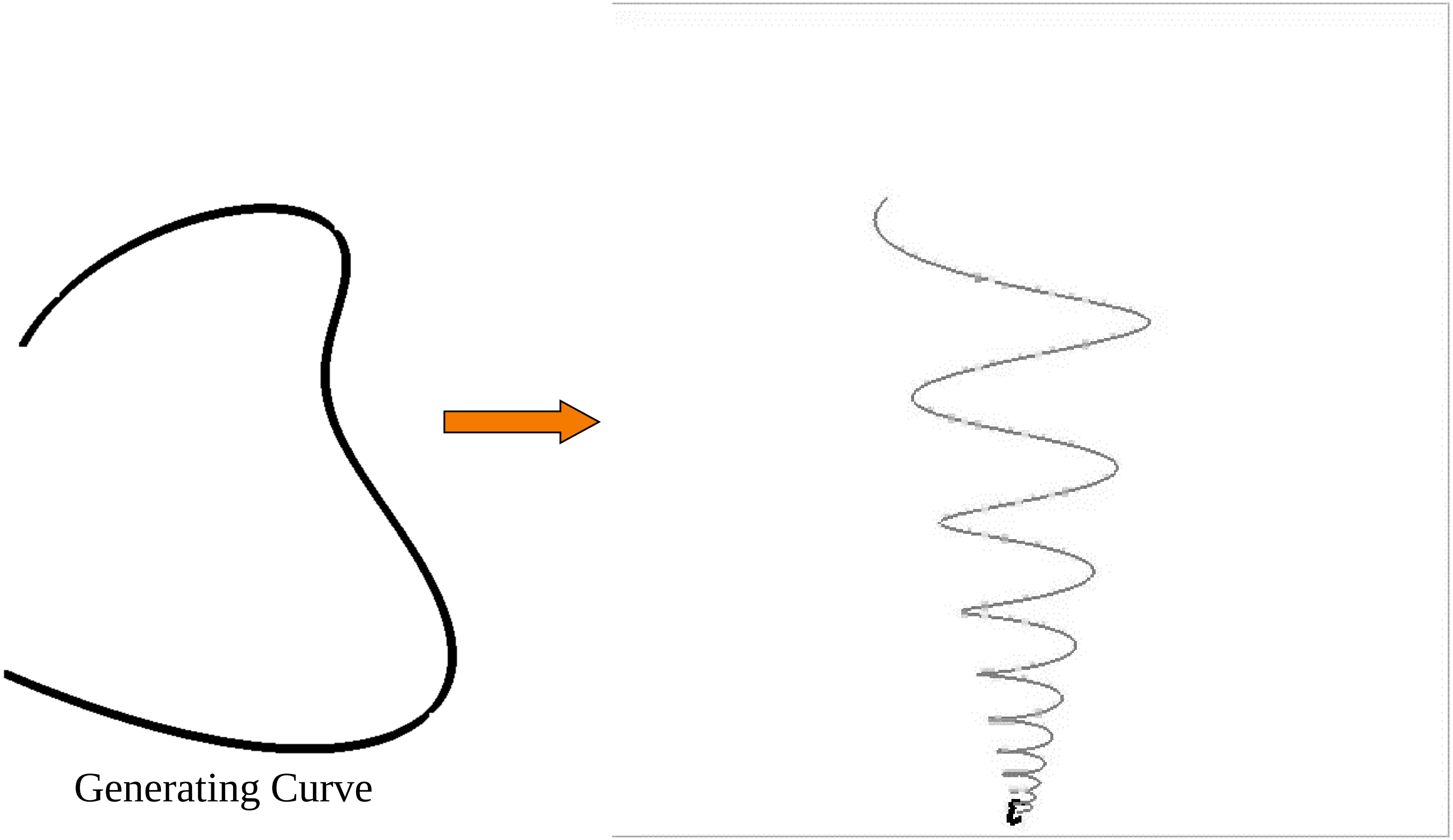
- Create 3D polygonal surface models of seashells

"Modeling Seashells,"
Deborah Fowler, Hans Meinhardt,
and Przemyslaw Prusinkiewicz,
Computer Graphics (SIGGRAPH 92),
Chicago, Illinois, July, 1992, p 379-387.

Fowler et al. Figure 7

Example: Seashells

- Sweep generating curve around helico-spiral axis



Generating Curve

Example: Seashells

- Sweep generating curve around helico-spiral axis

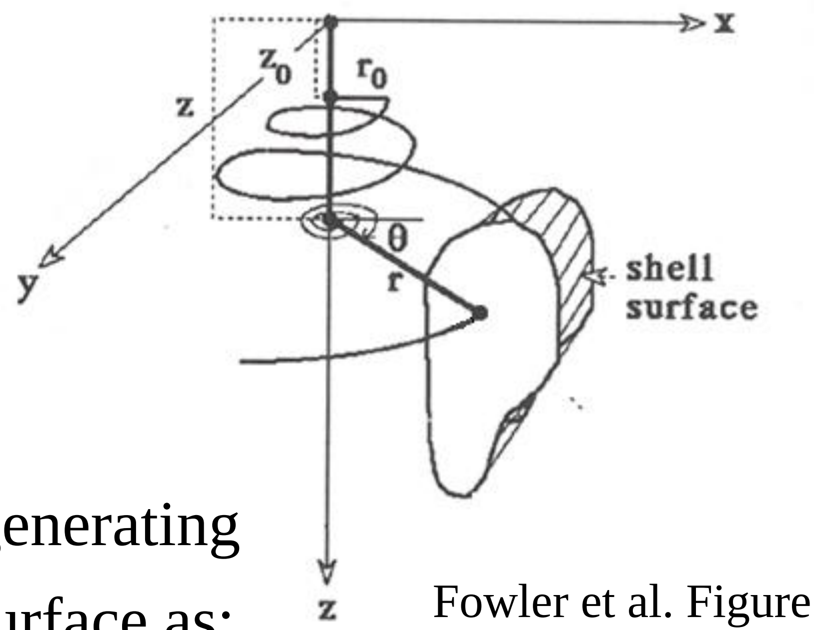
Helico-Spiral definition:

$$H(\theta) = \left(\cos\theta \cdot r(\theta), z(\theta), \sin\theta \cdot r(\theta)\right)$$

Angle: θ

Radius: $r(\theta) = e^{\lambda\theta}$

Height: $z(\theta) = e^{\mu\theta}$



Fowler et al. Figure 1

If $G(\phi) = \left(G_x(\phi), G_y(\phi)\right)$ is the generating curve, we can try to represent the surface as:

$$\mathcal{S}(\theta, \phi) = S(\theta) + \left(G_x(\phi), G_y(\phi), 0\right) \cdot r(\theta)$$

Example: Seashells

- Sweep generating curve around helico-spiral axis

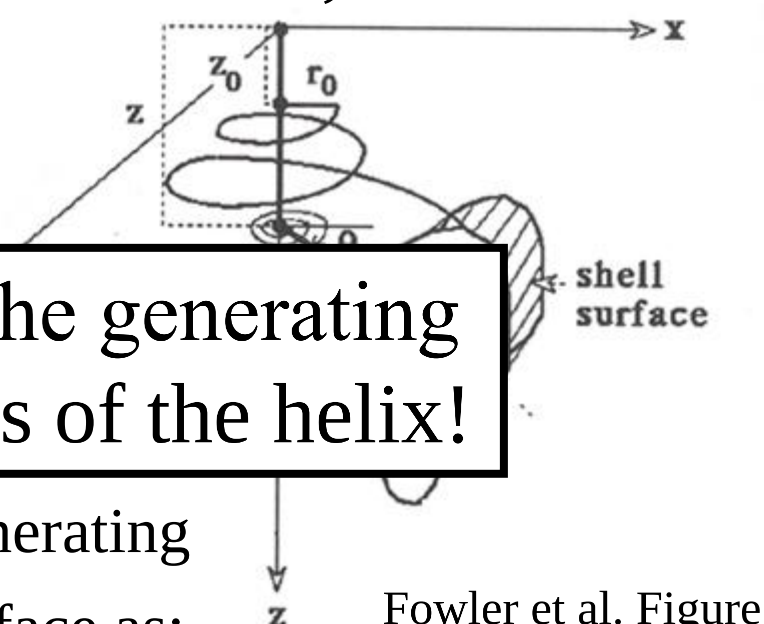
Helico-Spiral definition:

$$H(\theta) = \left(\cos\theta \cdot r(\theta), z(\theta), \sin\theta \cdot r(\theta)\right)$$

Angle: θ

Radius: $r(\theta) = e^{\lambda\theta}$

Height: $z(\theta) = e^{\mu\theta}$



Fowler et al. Figure 1

This doesn't rotate the generating curve around the axis of the helix!

If $G(\phi) = \left(G_x(\phi), G_y(\phi)\right)$ is the generating curve, we can try to represent the surface as:

$$\mathcal{S}(\theta, \phi) = S(\theta) + \left(G_x(\phi), G_y(\phi), 0\right) \cdot r(\theta)$$

Example: Seashells

- Sweep generating curve around helico-spiral axis

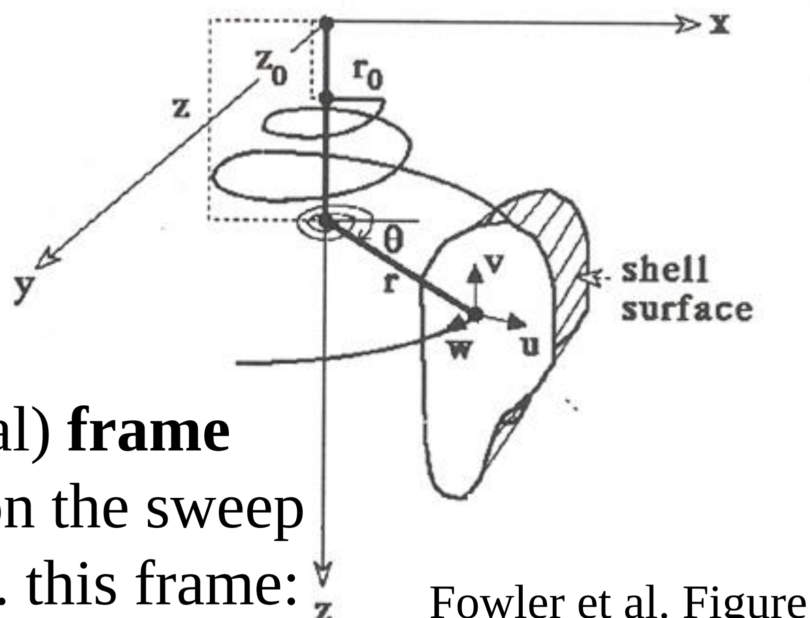
Helico-Spiral definition:

$$H(\theta) = \left(\cos\theta \cdot r(\theta), z(\theta), \sin\theta \cdot r(\theta)\right)$$

Angle: θ

Radius: $r(\theta) = e^{\lambda\theta}$

Height: $z(\theta) = e^{\mu\theta}$



Fowler et al. Figure 1

Instead, compute a local (orthogonal) **frame** $\{\vec{u}(\theta), \vec{v}(\theta), \vec{w}\{(\theta)\}$ at each point on the sweep curve and describe the surface w.r.t. this frame:

$$\mathcal{S}(\theta, \phi) = S(\theta) + \left(\vec{u}(\theta) \cdot G_x(\phi) + \vec{v}(\theta) \cdot G_y(\phi)\right) \cdot r(\theta)$$

Example: Seashells

- Swee … iral axis

$\vec{u}(\theta)$ and $\vec{v}(\theta)$ define the plane that is perpendicular to the curve H at θ:

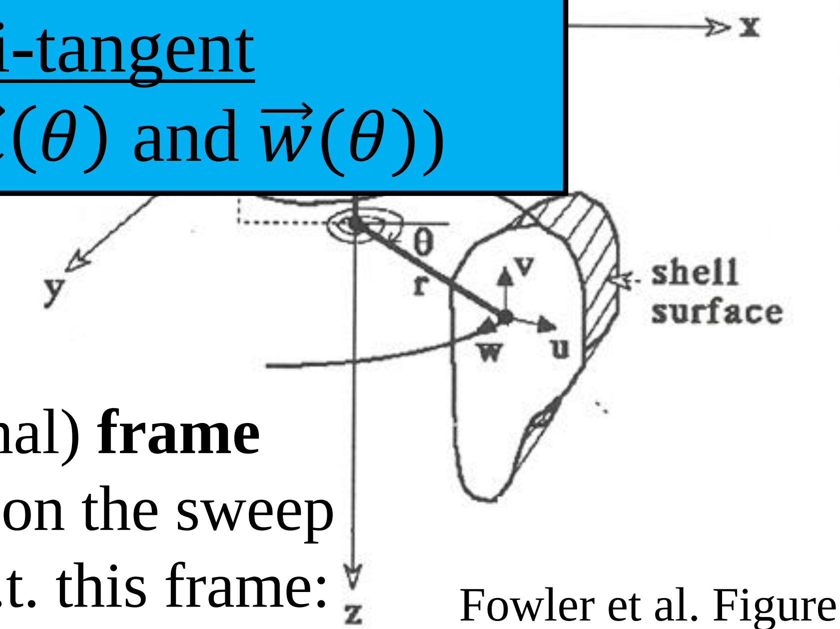
- $\vec{w}(\theta)$ is the curve <u>tangent</u>
- $\vec{u}(\theta)$ is the curve <u>normal</u>
- $\vec{v}(\theta)$ is the curve <u>bi-tangent</u> (perpendicular to $\vec{u}(\theta)$ and $\vec{w}(\theta)$)

Helico-S

Angle:

Radius:

Height: $z(\theta) = e^{\mu\theta}$



Fowler et al. Figure 1

Instead, compute a local (orthogonal) **frame** $\{\vec{u}(\theta), \vec{v}(\theta), \vec{w}\{(\theta)\}$ at each point on the sweep curve and describe the surface w.r.t. this frame:

$$\mathcal{S}(\theta,\phi) = S(\theta) + \left(\vec{u}(\theta)\cdot G_x(\phi) + \vec{v}(\theta)\cdot G_y(\phi)\right)\cdot r(\theta)$$

Example: Seashells

- Generate different shells by varying parameters

Different helico-spirals

Fowler et al. Figure 2

Example: Seashells

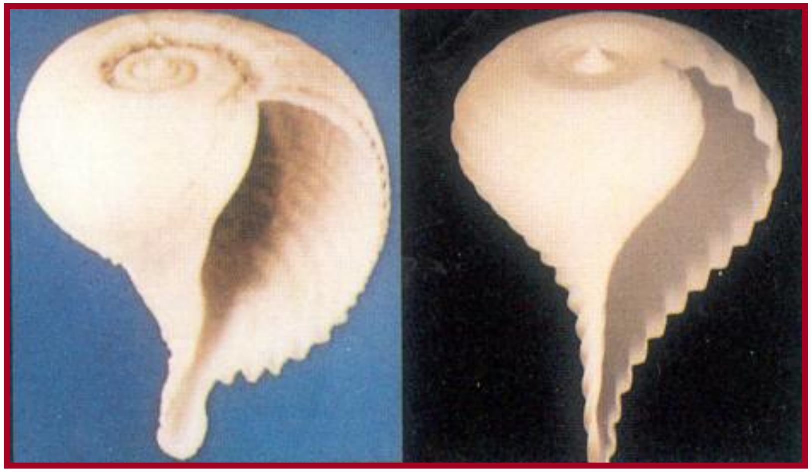
- Generate different shells by varying parameters

Different generating curves

Fowler et al. Figure 3

Example: Seashells

Generate many interesting shells with a simple procedural model!



Fowler et al. Figures 4,5,7