Spline Notes

Marc Olano
University of Maryland, Baltimore County

October 23, 2007

1 Introduction

I. Modeled after drafting tool

A. Thin strip of wood or metal

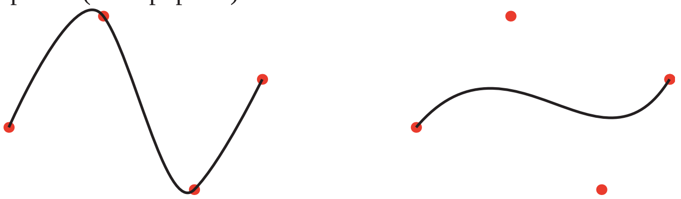
B. Control smooth curved path by running between small weights or *ducks*

Image from www.frets.com

II. Graphics version = parametric curve: $\vec{p}(s)$

A. Control Points

1. Points in space that control spline's shape. Like ducks in drafting splines
2. Interpolating = spline passes through control points
3. Approximating = spline passes near but not necessarily through control points (most popular)



B. Non-rational/ordinary: $\vec{P}(s)$ = vector of desired dimension

$$1D : \vec{P}(s) = (X(s)); \quad 2D : \vec{P}(s) = \begin{pmatrix} X(s) \\ Y(s) \end{pmatrix}; \quad 3D : \vec{P}(s) = \begin{pmatrix} X(s) \\ Y(s) \\ Z(s) \end{pmatrix}$$

C. Rational: $\vec{p}(s)$ = homogeneous vector

$$\vec{p}(s)=\begin{pmatrix} x(s)\\ y(s)\\ z(s)\\ z(s)\end{pmatrix} \quad \text{where} \quad \begin{pmatrix} X\\ Y\\ Z\end{pmatrix}=\begin{pmatrix} x/w\\ y/w\\ z/w\end{pmatrix}$$

D. Tangent, Normal, Binormal

A. Tangent, $\hat{T}$ = vector in direction of curve

B. Normal, $\hat{N}$ = vector normal to curve, in direction of greatest curvature.

C. Binormal, $\hat{B}$ = vector normal to curve, perpendicular to tangent and normal

D. Define in terms of 1st and 2nd derivatives of curve:

$$\begin{aligned}
\vec{P}'(s) &= \begin{pmatrix} X'(s)\\ Y'(s)\\ Z'(s)\end{pmatrix} = velocity(s)*\hat{T}(s)\\
\vec{P}''(s) &= \begin{pmatrix} X''(s)\\ Y''(s)\\ Z''(s)\end{pmatrix} = linearAccel(s)*\hat{T}(s)+centripidalAccel*\hat{N}(s)
\end{aligned}$$

E. Together, these make the *Frenet Frame* $\hat{T}$,$\hat{N}$,$\hat{B}$

$$\begin{aligned}
\hat{T}(s) &= normalize\left(\vec{P}'(s)\right)\\
\hat{B}(s) &= normalize\left(\vec{P}'(s)\times\vec{P}''(s)\right)\\
\hat{N}(s) &= \hat{B}(s)\times\hat{T}(s)
\end{aligned}$$

2 Polynomial splines

I. About the only splines you will ever see.

II. Most common in graphics: cubic (s^3) for general rendering, quintic (s^5) for CAD

$$\begin{aligned}
X(s) &= as^3+bs^2+cs+d &&= \begin{pmatrix} a & b & c & d\end{pmatrix}\begin{pmatrix} s^3\\ s^2\\ s\\ 1\end{pmatrix} &&= \begin{pmatrix} a & b & c & d\end{pmatrix}\vec{s}\\
X'(s) &= 3as^2+2bs+c &&= \begin{pmatrix} a & b & c & d\end{pmatrix}\begin{pmatrix} 3s^2\\ 2s\\ 1\\ 0\end{pmatrix} &&= \begin{pmatrix} a & b & c & d\end{pmatrix}\vec{s}'\\
X''(s) &= 6as+2b &&= \begin{pmatrix} a & b & c & d\end{pmatrix}\begin{pmatrix} 6s\\ 2\\ 0\\ 0\end{pmatrix} &&= \begin{pmatrix} a & b & c & d\end{pmatrix}\vec{s}''
\end{aligned}$$

III. Constraints

A. Degree n polynomial has $n+1$ coefficients:
$as+b$; as^2+bs+c; as^3+bs^2+cs+d; ...

B. Solving for $n+1$ unknown coefficients requires $n+1$ *constraints* (so for cubic, you need 4)

Typical types of constraints:

1. Pass through X_i at s_i

$$\begin{aligned} X_i &= \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} s_i^3 \\ s_i^2 \\ s_i \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} a & b & c & d \end{pmatrix} \vec{s}_i \end{aligned}$$

2. Have derivative X_i' at s_i

$$\begin{aligned} X_i' &= \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 3s_i^2 \\ 2s_i \\ 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} a & b & c & d \end{pmatrix} \vec{s}_i\,' \end{aligned}$$

3. Have 2nd derivative X_i'' at s_i

$$\begin{aligned} X_i'' &= \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 6s_i \\ 2 \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} a & b & c & d \end{pmatrix} \vec{s}_i\,'' \end{aligned}$$

IV. Fitting many control points

A. Global: fit with large system / high-degree polynomial

1. + Single curve
2. – Every control point has some affect on the entire curve

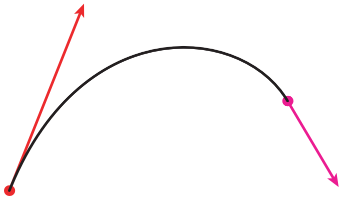
B. Piecewise: fit with many short curves, joined end-to-end

1. + Control points only affect local curve
2. – Must use some constraints to keep joins smooth

3 Spline zoo

I. Hermite

A. Give position and derivatives at each endpoint



X_0, X_0' at $s = s_0 = 0$; X_1, X_1' at $s = s_1 = 1$

Use conditions of the form

$$\begin{aligned}
X(0) &= X_0 &&= as_0^3 + by_0^2 + cs_0 + d &&= d \\
X(1) &= X_1 &&= as_1^3 + bs_1^2 + cs_1 + d &&= a + b + c + d \\
X'(0) &= X_0' &&= 3as_0^2 + 2bs_0 + c &&= c \\
X'(1) &= X_1' &&= 3as_1^2 + 2bs_1 + c &&= 3a + 2b + c
\end{aligned}$$

Or in matrix form:

$$\begin{aligned}
\begin{pmatrix} X_0 & X_1 & X_0' & X_1' \end{pmatrix} &= \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} \vec{s}_0 & \vec{s}_1 & \vec{s}_0{}' & \vec{s}_1{}' \end{pmatrix} \\
&= \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} a & b & c & d \end{pmatrix} \mathbf{H}
\end{aligned}$$

Even though $X(s)$ is a polynomial in s, it is **linear** in a, b, c and d. Solve as linear equation!

$$\begin{aligned}
\begin{pmatrix} a & b & c & d \end{pmatrix} &= \begin{pmatrix} X_0 & X_1 & X_0' & X_1' \end{pmatrix} \mathbf{H}^{-1} \\
&= \begin{pmatrix} X_0 & X_1 & X_0' & X_1' \end{pmatrix} \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}
\end{aligned}$$

B. Can reuse this matrix for **any** curve specified by value and derivatives at the endpoints

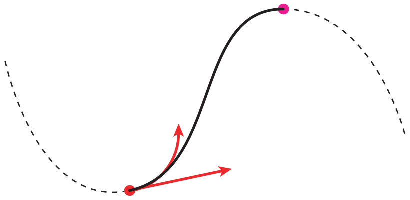
C. Can also reuse the same matrix for each component of the curve:

$$\begin{aligned}
\vec{P}(s) = \begin{pmatrix} X(s) \\ Y(s) \\ Z(s) \end{pmatrix} &= \begin{pmatrix} a_X & b_X & c_X & d_X \\ a_Y & b_Y & c_Y & d_Y \\ a_Z & b_Z & c_Z & d_Z \end{pmatrix} \begin{pmatrix} s^3 \\ s^2 \\ s \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \vec{A} & \vec{B} & \vec{C} & \vec{D} \end{pmatrix} \vec{s}
\end{aligned}$$

$$\begin{aligned}
\begin{pmatrix} a_X & b_X & c_X & d_X \\ a_Y & b_Y & c_Y & d_Y \\ a_Z & b_Z & c_Z & d_Z \end{pmatrix} &= \begin{pmatrix} X_0 & X_1 & X_0' & X_1' \\ Y_0 & Y_1 & Y_0' & Y_1' \\ Z_0 & Z_1 & Z_0' & Z_1' \end{pmatrix} \mathbf{H}^{-1} \\
\begin{pmatrix} \vec{A} & \vec{B} & \vec{C} & \vec{D} \end{pmatrix} &= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_0{}' & \vec{P}_0{}' \end{pmatrix} \mathbf{H}^{-1}
\end{aligned}$$

II. Natural

A. value, 1st and 2nd derivatives at $s = 0$ match previous



B. Even though segmented, each control point has **global** effect on all segments!

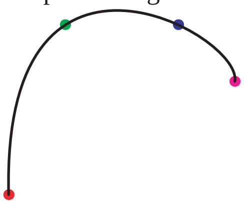
C. Create single linear system to solve for coefficients of all curve segments.

D. *Natural* because behavior is similar to drafting spline

III. Interpolating

A. To fit m points, need polynomial of degree $n = m - 1$

B. Single curve, each control point has global influence



C. Example: fit four points with a cubic
$\vec{P}_0$ at $s = s_0 = 0$, $\vec{P}_1$ at $s = s_1 = \frac{1}{3}$, $\vec{P}_2$ at $s = s_2 = \frac{2}{3}$, and $\vec{P}_3$ at $s = s_3 = 1$

$$\begin{aligned}
(\vec{P}_0 \quad \vec{P}_1 \quad \vec{P}_2 \quad \vec{P}_3) &= (\vec{A} \quad \vec{B} \quad \vec{C} \quad \vec{D})(\vec{s}_0 \quad \vec{s}_1 \quad \vec{s}_2 \quad \vec{s}_3) \\
&= (\vec{A} \quad \vec{B} \quad \vec{C} \quad \vec{D})\begin{pmatrix} 0 & \frac{1}{27} & \frac{8}{27} & 1 \\ 0 & \frac{1}{9} & \frac{4}{9} & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \\
&= (\vec{A} \quad \vec{B} \quad \vec{C} \quad \vec{D})\,\mathbf{M} \\
(\vec{A} \quad \vec{B} \quad \vec{C} \quad \vec{D}) &= (\vec{P}_0 \quad \vec{P}_1 \quad \vec{P}_2 \quad \vec{P}_3)\,\mathbf{M}^{-1} \\
&= (\vec{P}_0 \quad \vec{P}_1 \quad \vec{P}_2 \quad \vec{P}_3)\begin{pmatrix} -\frac{9}{2} & 9 & -\frac{11}{2} & 1 \\ \frac{27}{2} & -\frac{45}{2} & 9 & 0 \\ -\frac{27}{2} & 18 & -\frac{9}{2} & 0 \\ \frac{9}{2} & -\frac{9}{2} & 1 & 0 \end{pmatrix}
\end{aligned}$$

D. Basis functions
Full spline equation is

$$\vec{P}(s) = (\vec{P}_0 \quad \vec{P}_1 \quad \vec{P}_2 \quad \vec{P}_3)\,\mathbf{M}^{-1}\,\vec{s}$$

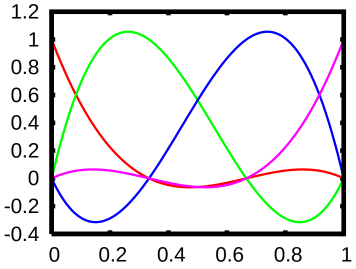
Grouping control points and matrix gives coefficients.

$$\vec{P}(s) = \left[\begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \mathbf{M}^{-1} \right] \vec{s}$$

Grouping matrix and polynomial terms gives a polynomial *basis functions* for each control point. Total curve is a weighted average of the control points, with the weights given by the basis functions:

$$\begin{aligned} \vec{P}(s) &= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \left[\mathbf{M}^{-1} \vec{s} \right] \\ &= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \begin{pmatrix} f_0(s) \\ f_1(s) \\ f_2(s) \\ f_3(s) \end{pmatrix} \\ &= \vec{P}_0(-\tfrac{9}{2}s^3 + 9s^2 - \tfrac{11}{2}s + 1) + \\ &\quad \vec{P}_1(\tfrac{27}{2}s^3 - \tfrac{45}{2}s^2 + 9s) + \\ &\quad \vec{P}_2(-\tfrac{27}{2}s^3 + 18s^2 - \tfrac{9}{2}s) + \\ &\quad \vec{P}_3(\tfrac{9}{2}s^3 - \tfrac{9}{2}s^2 + s) \end{aligned}$$

E. Here is a graph of these four basis functions



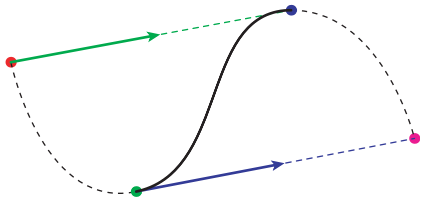
IV. Catmull-Rom

A. Too hard to give derivatives, so derive them from control points.

B. Endpoints $\vec{P}_1$, $\vec{P}_2$

C. Derivatives $\frac{1}{2}(\vec{P}_2 - \vec{P}_0)$, $\frac{1}{2}(\vec{P}_3 - \vec{P}_1)$

D. Chosen to pass through all points, match value and 1st derivative at 1st endpoint, and preserve local control

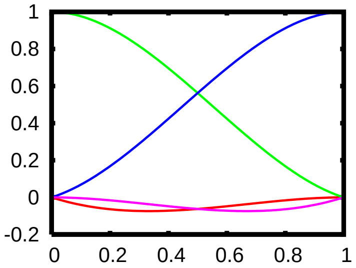


E. Matrix

$$\begin{pmatrix} a & b & c & d \end{pmatrix} = \begin{pmatrix} \vec{P}_1 & \vec{P}_2 & \frac{1}{2}(\vec{P}_2 - \vec{P}_0) & \frac{1}{2}(\vec{P}_3 - \vec{P}_1) \end{pmatrix} \mathbf{H}^{-1}$$

Can use that equation as-is, or rearrange to find basis functions

$$
\begin{aligned}
\vec{P}(s) &= \begin{pmatrix} \vec{P}_1 & \vec{P}_2 & \frac{1}{2}(\vec{P}_2 - \vec{P}_0) & \frac{1}{2}(\vec{P}_3 - \vec{P}_1) \end{pmatrix} \mathbf{H}^{-1}\,\vec{s} \\
&= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & -\frac{1}{2} & 0 \\ 1 & 0 & 0 & -\frac{1}{2} \\ 0 & 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \mathbf{H}^{-1}\,\vec{s} \\
&= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \begin{pmatrix} -\frac{1}{2}s^3 + s^2 - \frac{1}{2}s \\ \frac{3}{2}s^3 - \frac{5}{2}s^2 + 1 \\ -\frac{3}{2}s^3 + 2s^2 + \frac{1}{2}s \\ \frac{1}{2}s^3 - \frac{1}{2}s^2 \end{pmatrix}
\end{aligned}
$$



V. Cardinal

A. Catmull-Rom is good, but not enough control

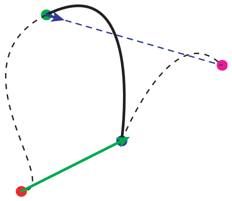
B. Add tension parameter, τ to derivative terms

C. Endpoints $\vec{P}_1$, $\vec{P}_2$

D. Derivatives $\frac{(1-\tau_1)}{2}(\vec{P}_2 - \vec{P}_0)$, $\frac{(1-\tau_2)}{2}(\vec{P}_3 - \vec{P}_1)$

E. $\tau = 1 \Rightarrow$ cusp
$0 < \tau < 1 \Rightarrow$ sharper than Catmull-Rom
$\tau = 0 \Rightarrow$ Catmull-Rom
$\tau < 0 \Rightarrow$ flatter than Catmull-Rom

F. Example with $\tau_1 = -1$ and $\tau_2 \approx 1$



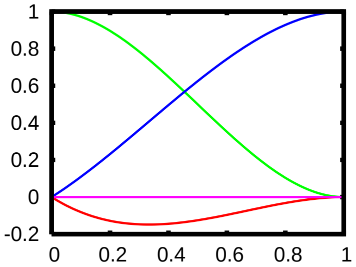
G. Matrix

$$
\begin{pmatrix} a & b & c & d \end{pmatrix} = \begin{pmatrix} \vec{P}_1 & \vec{P}_2 & \frac{1-\tau_1}{2}(\vec{P}_2 - \vec{P}_0) & \frac{1-\tau_2}{2}(\vec{P}_3 - \vec{P}_1) \end{pmatrix} \mathbf{H}^{-1}
$$

H. Basis functions:

$$\begin{aligned}
\vec{P}(s) &= \begin{pmatrix} \vec{P}_1 & \vec{P}_2 & \frac{1-\tau_1}{2}(\vec{P}_2-\vec{P}_0) & \frac{1-\tau_2}{2}(\vec{P}_3-\vec{P}_1) \end{pmatrix} \mathbf{H}^{-1}\,\vec{s} \\
&= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & -\frac{1-\tau_1}{2} & 0 \\ 1 & 0 & 0 & -\frac{1-\tau_2}{2} \\ 0 & 1 & \frac{1-\tau_1}{2} & 0 \\ 0 & 0 & 0 & \frac{1-\tau_2}{2} \end{pmatrix} \mathbf{H}^{-1}\,\vec{s} \\
&= \begin{pmatrix} \vec{P}_0 & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 \end{pmatrix} \begin{pmatrix} -\frac{1-\tau_1}{2}s^3 + (1-\tau_1)s^2 - \frac{1-\tau_1}{2}s \\ \left(2-\frac{1-\tau_2}{2}\right)s^3 + \left(\frac{1-\tau_2}{2}-3\right)s^2 + 1 \\ \left(\frac{1-\tau_1}{2}-2\right)s^3 + (3-(1-\tau_1))s^2 + \frac{1-\tau_1}{2}s \\ \frac{1-\tau_2}{2}s^3 - \frac{1-\tau_2}{2}s^2 \end{pmatrix}
\end{aligned}$$

I. Graph of basis functions for $\tau_1 = -1$, $\tau_2 = 1$



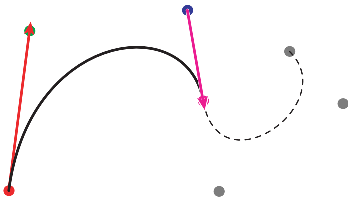
VI. Beziér

A. Most common

B. Endpoints $\vec{P}_0, \vec{P}_3$

C. Derivatives $\vec{P}_1 - \vec{P}_0, \vec{P}_3 - \vec{P}_2$

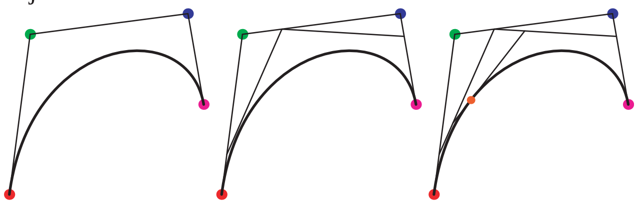
D. Match derivatives by choice of interior control points locations on successive segments



E. Matrix

$$\begin{pmatrix} a & b & c & d \end{pmatrix} = \begin{pmatrix} \vec{P}_0 & \vec{P}_3 & \vec{P}_1 - \vec{P}_0 & \vec{P}_3 - \vec{P}_2 \end{pmatrix} \mathbf{H}^{-1}$$

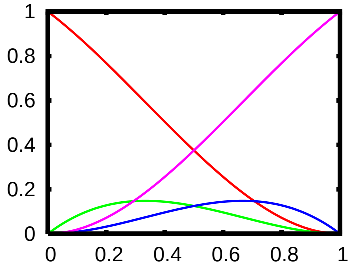
F. Can also find points on Beziér curves using *deCastlejau's algorithm*. To find the point 1/n of the way along the curve, find the points 1/n of the way between each pair of control points. Then find the point 1/n of the way between each pair of points just found. Then find the point 1/n of the way between the two points just found.



deCastlejau's algorithm generalizes easily to Beziér curves of higher (or lower) degree than just cubic. Just iterate the successive subdivisions as many times as necessary until you get to a single point.

G. Basis functions

$$\begin{aligned}\vec{P}(s) &= (\vec{P}_0 \quad \vec{P}_3 \quad \vec{P}_1-\vec{P}_0 \quad \vec{P}_3-\vec{P}_2)\ \mathbf{H}^{-1}\ \vec{s} \\ &= (\vec{P}_0 \quad \vec{P}_1 \quad \vec{P}_2 \quad \vec{P}_3)\begin{pmatrix}1 & 0 & -1 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\\ 0 & 1 & 0 & 1\end{pmatrix}\mathbf{H}^{-1}\ \vec{s} \\ &= (\vec{P}_0 \quad \vec{P}_1 \quad \vec{P}_2 \quad \vec{P}_3)\begin{pmatrix}s^3-s^2-s+1\\ s^3-2s^2+s\\ -s^3+s^2\\ -s^3+2s^2\end{pmatrix}\end{aligned}$$



These are known as the *Bernstein basis functions*. Beziér curves can be extended to any degree r polynomial using the n basis functions

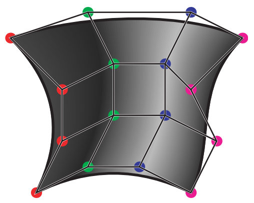
$$B_{n,r}(s) = \binom{n}{r} t^r (1-t)^{n-r}$$

In terms of the Bernstein basis functions, the curve equation is

$$\vec{P}(s) = \sum_{i=0}^{n} \vec{P}_i B_{n,i}(s)$$

4 Patches

I. Create a rectangular patch by smoothly changing each control points of one spline curve along paths given by crossing set of spline curves.



II. The patch is symmetric in which direction you consider to be the control splines and which direction is the splines that sweep out the surface.

$$\begin{aligned}
\vec{P}(s,t) &= \textstyle\sum_{i=0}^{n} \left(\sum_{j=0}^{m} \vec{P}_{i,j} B_{n,i}(s) \right) B_{n,j}(t) \\
&= \textstyle\sum_{i=0}^{n} \sum_{j=0}^{m} \left(\vec{P}_{i,j} B_{n,i}(s) B_{m,j}(t) \right) \\
&= \textstyle\sum_{j=0}^{m} \sum_{i=0}^{n} \left(\vec{P}_{i,j} B_{n,i}(s) B_{m,j}(t) \right) \\
&= \textstyle\sum_{j=0}^{m} \left(\sum_{i=0}^{n} \vec{P}_{i,j} B_{m,j}(t) \right) B_{n,i}(s)
\end{aligned}$$

III. $\vec{P}_{i,j}$ is known as the control mesh

IV. Render by subdividing (not covered here) or tessellating into polygons.