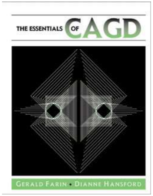
The Essentials of CAGD

Chapter 11: Working with B-spline Curves

Gerald Farin & Dianne Hansford

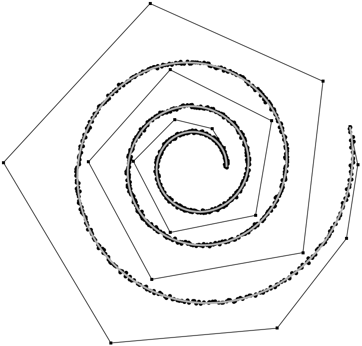
CRC Press, Taylor & Francis Group, An A K Peters Book
www.farinhansford.com/books/essentials-cagd



Outline

1. Introduction to Working with B-spline Curves
2. Designing with B-spline curves
3. Least Squares Approximation
4. Shape Equations
5. Cubic Spline Interpolation
6. Cubic Spline Interpolation in a Nutshell

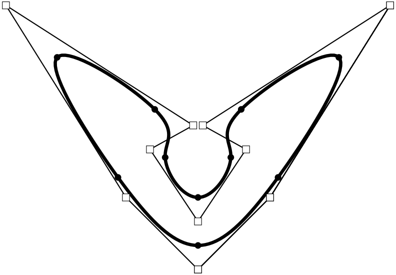
Introduction to Working with B-spline Curves



How to use B-spline curves?

B-spline curves popularity due to the many possible ways in which they can be “put to work”

Designing with B-spline curves



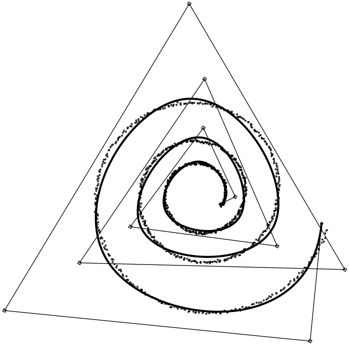
Find: a B-spline curve for the character “v” in some fancy font

Most basic design process:
Move individual control points until desired shape achieved

Manual/interactive method ok for final fine tuning of shape

Initial “guess” can be created faster with methods in this chapter

Introduction to Working with B-spline Curves



Many applications supply a large number of data points
$\Rightarrow$ from scanning devices

Find a cubic B-spline curve approximating their shape

Most popular method:
least squares approximation

Least Squares Approximation

Cubic B-spline curve defined by
– L polynomial segments
– Assume simple domain knots $\Rightarrow$ number of knots $K = L + 5$
– Knot sequence $u_0, \ldots, u_{K-1}$

Given P data points
– $\mathbf{p}_0, \ldots, \mathbf{p}_{P-1}$
– Each $\mathbf{p}_i$ associated with parameter value v_i

Find a cubic B-spline curve $\mathbf{x}(u)$
such that the distances $\|\mathbf{p}_i - \mathbf{x}(v_i)\|$ are small

Least Squares Approximation

B-spline curve

$$\mathbf{x}(u) = \mathbf{d}_0 N_0^3(u) + \ldots + \mathbf{d}_{D-1} N_{D-1}^3(u)$$

Given points $\mathbf{p}_i = \mathbf{x}(v_i)$ $i = 0, \ldots, P-1$ leading to

$$\begin{aligned} \mathbf{d}_0 N_0^3(v_0) + \ldots + \mathbf{d}_{D-1} N_{D-1}^3(v_0) &= \mathbf{p}_0 \\ &\vdots \\ \mathbf{d}_0 N_0^3(v_{P-1}) + \ldots + \mathbf{d}_{D-1} N_{D-1}^3(v_{P-1}) &= \mathbf{p}_{P-1} \end{aligned}$$

In matrix form:

$$\begin{bmatrix} N_0^3(v_0) & \ldots & N_{D-1}^3(v_0) \\ & \vdots & \\ & \vdots & \\ N_0^3(v_{P-1}) & \ldots & N_{D-1}^3(v_{P-1}) \end{bmatrix} \begin{bmatrix} \mathbf{d}_0 \\ \vdots \\ \mathbf{d}_{D-1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 \\ \vdots \\ \vdots \\ \mathbf{p}_{P-1} \end{bmatrix}$$

$$M\mathbf{D} = \mathbf{P}$$

Least Squares Approximation

Linear system $M\mathbf{D} = \mathbf{P}$ is overdetermined
– Number P of data points $>$ number D of curve control points

Solution: multiply both sides by M^{T}:

$$M^{\mathrm{T}} M\mathbf{D} = M^{\mathrm{T}}\mathbf{P}$$

Linear system with D equations in D unknowns
– Square and symmetric coefficient matrix $M^{\mathrm{T}} M$
– Solution straightforward since $M^{\mathrm{T}} M$ invertible
as long as parameter values v_j must be "evenly" distributed in domain knots

Least Squares Approximation

Parameters to define: – How many segments L should the curve have?
– How to choose the knots u_j
– How to choose the parameter values v_i?

No universal answers – suggestions:

- Choose the parameters v_i according to the chord length
- Select $L \approx P/10$
- Choose u_i such that approximately ten v_j fall in each interval domain knot interval $[u_i, u_{i+1}]$

Shape Equations

Possible data point defects:

– noisy – unevenly distributed

$\Rightarrow$ Least squares approximation might fail to produce nice results

Solution: modify method with shape information

– Accept deviation from data for a better-shaped curve
– Formulate conditions for the control polygon's shape
– Assumption: a polygon is nice if it does not wiggle much

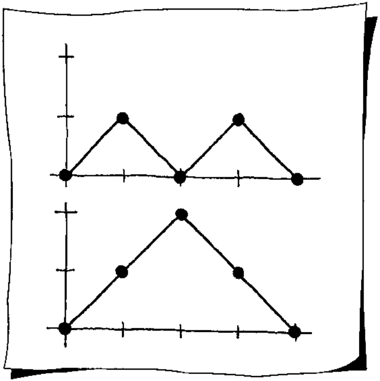
Expressed by computing *second differences* of control points

$$\Delta^2 \mathbf{d}_i = \mathbf{d}_i - 2\mathbf{d}_{i+1} + \mathbf{d}_{i+2}$$

Less wiggle $\Rightarrow$ smaller sum:

$$S = \|\Delta^2 \mathbf{d}_0\| + \ldots + \|\Delta^2 \mathbf{d}_{D-3}\|$$

Shape Equations



Example

Top polygon: $S = 6$

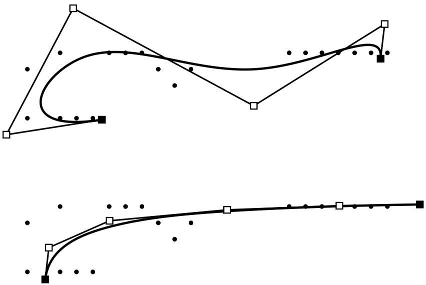
$$\Delta^2\mathbf{d}_0 = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \quad \Delta^2\mathbf{d}_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad \Delta^2\mathbf{d}_2 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

Bottom polygon: $S = 2$

$$\Delta^2\mathbf{d}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Delta^2\mathbf{d}_1 = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \quad \Delta^2\mathbf{d}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$\Rightarrow$ Bottom polygon is smoother

Shape Equations



Least squares approximation:
Add shape equations to the overdetermined system

$$\mathbf{d}_0 - 2\mathbf{d}_1 + \mathbf{d}_2 = \mathbf{0}$$

$$\vdots$$

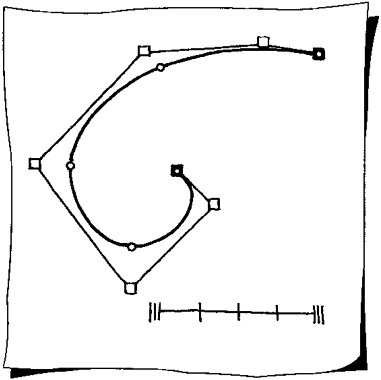
$$\mathbf{d}_{D-3} - 2\mathbf{d}_{D-2} + \mathbf{d}_{D-1} = \mathbf{0}$$

Overdetermined linear system becomes even more overdetermined

Top: without shape equations
Bottom: with shape equations

Cubic Spline Interpolation

Interpolation: # *given* data equals # *unknown* control points



Given P data points

$\mathbf{p}_0, \ldots, \mathbf{p}_{P-1}$

Interpolate with a cubic B-spline curve $\mathbf{x}(u)$

End knots of multiplicity three:

$u_0 = u_1 = u_2$

$u_3, \ldots, u_{K-4}$

$u_{K-3} = u_{K-2} = u_{K-1}$

Junction points paired with $\mathbf{p}_i$

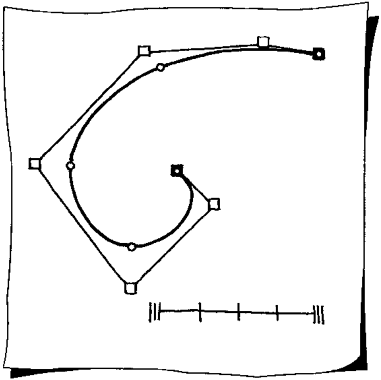
$\mathbf{x}(u_2) = \mathbf{p}_0, \ldots$

$\Rightarrow P - 1$ curve segments

$\Rightarrow K = P + 4$ knots

$\Rightarrow D = P + 2$ control points

Cubic Spline Interpolation



Example

Given $P = 5$ data points

Need $K = 5 + 4 = 9$ knots

$$0,\ 0,\ 0,\ 1,\ 2,\ 3,\ 4,\ 4,\ 4$$

$\Rightarrow D = 7$ control points

$$\mathbf{d}_0, \ldots, \mathbf{d}_6$$

Cubic Spline Interpolation

Two more data items are needed than the curve has junction points

Solution: add two more data items at the ends of the curve

$$\mathbf{t}_s = \dot{\mathbf{x}}(u_2) \qquad \text{start tangent}$$
$$\mathbf{t}_e = \dot{\mathbf{x}}(u_{K-3}) \quad \text{end tangent}$$

These are called end conditions

Knots u_2 and u_{K-3} are the first and last domain knots

Bessel tangents method: extract tangents from interpolating parabola through first and last three data points
– See The Essentials of CAGD for equation details

Cubic Spline Interpolation

Interpolation conditions:

$$\begin{aligned}
\mathbf{p}_0 &= \mathbf{x}(u_2)\\
\mathbf{t}_s &= \dot{\mathbf{x}}(u_2)\\
\mathbf{p}_1 &= \mathbf{x}(u_3)\\
&\vdots\\
\mathbf{t}_e &= \dot{\mathbf{x}}(u_{K-3})\\
\mathbf{p}_{P-1} &= \mathbf{x}(u_{K-3})
\end{aligned}$$

Triple end knots result in

$$\mathbf{d}_0 = \mathbf{p}_0 \quad \text{and} \quad \mathbf{d}_{D-1} = \mathbf{p}_{P-1}$$

$\Rightarrow$ eliminates two unknowns
$\Rightarrow$ eliminates two equations

Simplified interpolation conditions:

$$\begin{aligned}
\mathbf{t}_s &= \dot{\mathbf{x}}(u_2)\\
\mathbf{p}_1 &= \mathbf{x}(u_3)\\
&\vdots\\
\mathbf{p}_{P-2} &= \mathbf{x}(u_{K-4})\\
\mathbf{t}_e &= \dot{\mathbf{x}}(u_{K-3})
\end{aligned}$$

For unknowns $\mathbf{d}_1, \ldots, \mathbf{d}_{D-2}$

Cubic Spline Interpolation

Example

Revisit previous example

$$
\begin{aligned}
\mathbf{p}_0 &= \mathbf{x}(0)\\
\mathbf{t}_s &= \dot{\mathbf{x}}(0)\\
\mathbf{p}_1 &= \mathbf{x}(1)\\
\mathbf{p}_2 &= \mathbf{x}(2)\\
\mathbf{p}_3 &= \mathbf{x}(3)\\
\mathbf{t}_e &= \dot{\mathbf{x}}(4)\\
\mathbf{p}_4 &= \mathbf{x}(4)
\end{aligned}
$$

Assigning $\mathbf{d}_0 = \mathbf{p}_0$ and $\mathbf{d}_6 = \mathbf{p}_4$
System becomes

$$
\begin{aligned}
\mathbf{t}_s &= \dot{\mathbf{x}}(0)\\
\mathbf{p}_1 &= \mathbf{x}(1)\\
\mathbf{p}_2 &= \mathbf{x}(2)\\
\mathbf{p}_3 &= \mathbf{x}(3)\\
\mathbf{t}_e &= \dot{\mathbf{x}}(4)
\end{aligned}
$$

$\Rightarrow$ five equations for the unknowns $\mathbf{d}_1, \ldots, \mathbf{d}_5$

Cubic Spline Interpolation

Each data point yields an equation of the form

$$\mathbf{p}_i = \mathbf{d}_0 N_0^3(u_{2+i}) + \ldots + \mathbf{d}_{D-1} N_{D-1}^3(u_{2+i})$$

Due to the local support property of B-spline curves

$$\mathbf{p}_i = \mathbf{d}_i N_i^3(u_{2+i}) + \mathbf{d}_{i+1} N_{i+1}^3(u_{2+i}) + \mathbf{d}_{i+2} N_{i+2}^3(u_{2+i})$$

$\Rightarrow$ tridiagonal structure

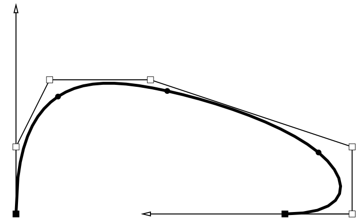
End conditions: first and last equation in the system
– For tridiagonal structure, must involve only the first and last unknowns

For the special case of equally spaced interior knots

$$6\mathbf{p}_i = \mathbf{d}_i + 4\mathbf{d}_{i+1} + \mathbf{d}_{i+2}$$

for each equation involving a data point

Cubic Spline Interpolation



Example

- Equally spaced knots
- End tangent equations: $\mathbf{t}_s = 3(\mathbf{d}_1 - \mathbf{d}_0)$ and $\mathbf{t}_e = 3(\mathbf{d}_6 - \mathbf{d}_5)$

$$\begin{bmatrix} 1 & & & & & \\ 3/2 & 7/2 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & 1 & 7/2 & 3/2 & \\ & & & & 1 & \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \\ \mathbf{d}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_0 + \frac{1}{3}\mathbf{t}_s \\ 6\mathbf{p}_1 \\ 6\mathbf{p}_2 \\ 6\mathbf{p}_3 \\ \mathbf{d}_6 - \frac{1}{3}\mathbf{t}_e \end{bmatrix}$$

Cubic Spline Interpolation in a Nutshell

Input:
– Data points $\mathbf{p}_0, \ldots, \mathbf{p}_{P-1}$
– A cubic B-spline knot sequence

$$u_0 = u_1 = u_2, \quad u_3, \ldots, u_{K-4}, \quad u_{K-3} = u_{K-2} = u_{K-1}$$

$K = P + 4 \quad \Rightarrow \quad P - 1$ curve segments

Find: cubic B-spline interpolant
– Control points $\mathbf{d}_0, \ldots, \mathbf{d}_{D-1}$ where $D = P + 2$
– Each data point $\mathbf{p}_i$ is associated with parameter u_{2+i}

Compute:

- Set $\mathbf{d}_0 = \mathbf{p}_0$ and $\mathbf{d}_{D-1} = \mathbf{p}_{P-1}$
- Create tangents $\mathbf{t}_s$ and $\mathbf{t}_e$ using Bessel tangent equations
- Set up the tridiagonal linear system of equations
- Solve the $(D-2) \times (D-2)$ linear system for $\mathbf{d}_1, \ldots, \mathbf{d}_{D-2}$