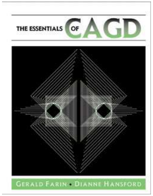
The Essentials of CAGD

Chapter 5: Putting Curves to Work

Gerald Farin & Dianne Hansford

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

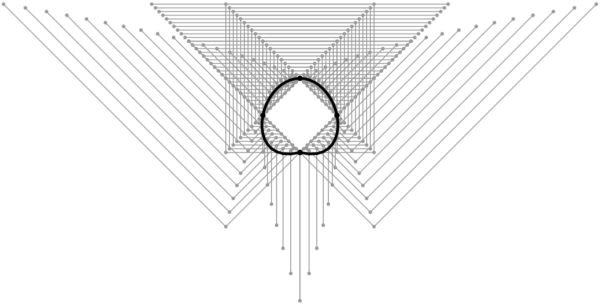


Outline

Introduction to Putting Curves to Work

Parametric curves describe geometric shapes

Design methods: *interpolation* and *approximation*

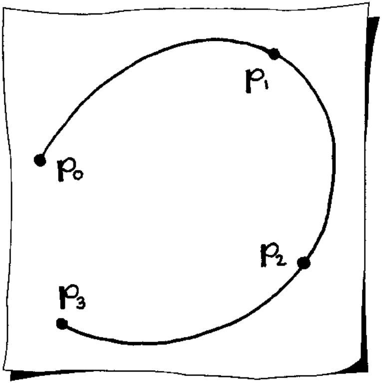


An interpolating polynomial curve

Evaluated at forty points

Intermediate steps in the computations shown (Aitken's algorithm)

Cubic Interpolation



Given: 2D or 3D points

$$\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$$

Find: curve passing through them
Called interpolation

Fit with a cubic Bézier curve

Assign parameter values

$$\mathbf{x}(t_i) = \mathbf{p}_i$$

Requirement: $t_i \leq t_{i+1}$

Example: $t_i = i/3$

Cubic Interpolation

Given: four point and parameter pairs $\mathbf{p}_i, t_i$

Find: a cubic Bézier curve $\mathbf{x}(t)$ such that

$$\mathbf{x}(t_i) = \mathbf{p}_i \qquad i = 0, 1, 2, 3$$

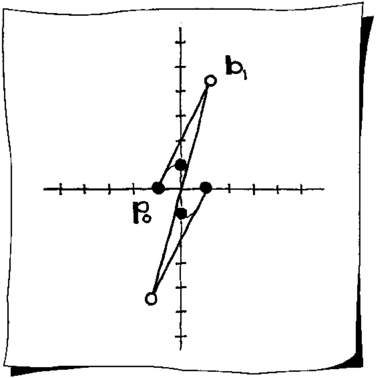
$$\mathbf{x}(t) = B_0^3(t)\mathbf{b}_0 + B_1^3(t)\mathbf{b}_1 + B_2^3(t)\mathbf{b}_2 + B_3^3(t)\mathbf{b}_3$$

$$\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} B_0^3(t_0) & B_1^3(t_0) & B_2^3(t_0) & B_3^3(t_0) \\ B_0^3(t_1) & B_1^3(t_1) & B_2^3(t_1) & B_3^3(t_1) \\ B_0^3(t_2) & B_1^3(t_2) & B_2^3(t_2) & B_3^3(t_2) \\ B_0^3(t_3) & B_1^3(t_3) & B_2^3(t_3) & B_3^3(t_3) \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

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

Solution: $\mathbf{B} = M^{-1}\mathbf{P}$

Cubic Interpolation



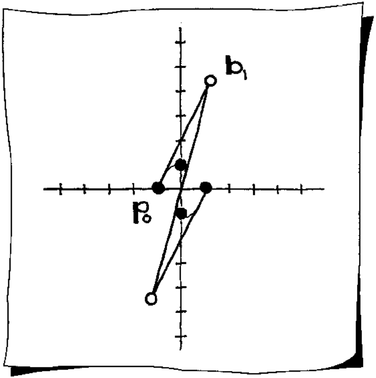
Example: Given $\mathbf{p}_i$ and $t_i = i/3$

$$\begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$M = \frac{1}{27}\begin{bmatrix} 27 & 0 & 0 & 0 \\ 8 & 12 & 6 & 1 \\ 1 & 6 & 12 & 8 \\ 0 & 0 & 0 & 27 \end{bmatrix}$$

$$\mathbf{B}: \quad \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$$

Cubic Interpolation



Example con't: Given $\mathbf{p}_i$ and $t_i = i/3$

$$\begin{bmatrix}-1\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\end{bmatrix} \quad \begin{bmatrix}0\\-1\end{bmatrix} \quad \begin{bmatrix}1\\0\end{bmatrix}$$

Linear system solver returns

$$\begin{bmatrix}-1\\7/6\\-7/6\\1\end{bmatrix} \quad \begin{bmatrix}0\\9/2\\-9/2\\0\end{bmatrix}$$

$\mathbf{b}_i$ for interpolating cubic:

$$\begin{bmatrix}-1\\0\end{bmatrix} \quad \begin{bmatrix}7/6\\9/2\end{bmatrix} \quad \begin{bmatrix}-7/6\\-9/2\end{bmatrix} \quad \begin{bmatrix}1\\0\end{bmatrix}$$

Interpolation Beyond Cubics

Polynomial interpolation for given data points

$$\mathbf{p}_0, \ldots, \mathbf{p}_n$$

Also given: corresponding parameter values $t_0, \ldots, t_n$

Interpolation problem leads to the linear system

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

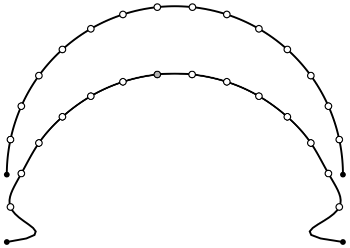
M is an $(n+1) \times (n+1)$ matrix with elements

$$m_{i,j} = B_j^n(t_i)$$

Solve using any linear system solver

Interpolation Beyond Cubics

Polynomial interpolation is guaranteed to work
Does not always produce satisfying results for higher degrees



Top: 16 points on a semicircle

Bottom: one data point changed
x-coordinate of gray data point modified by 0.002

A small change in data can lead to large changes in the interpolating curve
$\Rightarrow$ ill-conditioned process

Interpolation Beyond Cubics

Interpolating curve can be in form other than Bézier
– Different polynomial forms will give the identical result

Example: monomial form

$$\mathbf{x}(t) = \mathbf{a}_0 + \mathbf{a}_1 t + \ldots + \mathbf{a}_n t^n$$

Unknowns are the coefficients $\mathbf{a}_i$

$$\text{Linear system:} \quad \mathbf{P} = M\mathbf{A}$$

M is an $(n+1) \times (n+1)$ matrix with elements

$$m_{i,j} = t_i^j$$

Interpolation Beyond Cubics

Example: Lagrange polynomials

$$L_i^n(t) = \frac{(t - t_0) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_n)}{(t_i - t_0) \dots (t_i - t_{i-1})(t_i - t_{i+1}) \dots (t_i - t_n)}$$

$(* - t_i)$ term missing in numerator and denominator of i^{th} polynomial

Allow a very direct form for the interpolant:

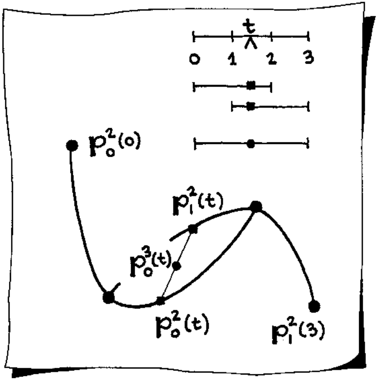
$$\mathbf{x}(t) = L_0^n(t)\mathbf{p}_0 + \dots + L_n^n(t)\mathbf{p}_n$$

$\Rightarrow$ Data points appear explicitly

– Called the cardinal form of the interpolant

Aitken's Algorithm

Recursive algorithm to compute points on interpolating polynomial curve
– Some of the characteristics of the de Casteljau algorithm



Derive via cubic case

Start with 2 quadratic curves
$\mathbf{p}_0^2(t)$ through $\mathbf{p}_0, \mathbf{p}_1$, $\mathbf{p}_2$
$\mathbf{p}_1^2(t)$ through $\mathbf{p}_1, \mathbf{p}_2$, $\mathbf{p}_3$

Construct interpolating cubic:

$$\mathbf{p}_0^3(t) = \frac{t_3 - t}{t_3 - t_0}\mathbf{p}_0^2(t) + \frac{t - t_0}{t_3 - t_0}\mathbf{p}_1^2(t)$$

Aitken's Algorithm

$$\mathbf{p}_0^3(t) = \frac{t_3 - t}{t_3 - t_0}\mathbf{p}_0^2(t) + \frac{t - t_0}{t_3 - t_0}\mathbf{p}_1^2(t)$$

Verify interpolation to all four data points

Check $\mathbf{p}_0$:

$$\mathbf{p}_0^3(t_0) = \frac{t_3 - t_0}{t_3 - t_0}\mathbf{p}_0^2(t_0) + \frac{t_0 - t_0}{t_3 - t_0}\mathbf{p}_1^2(t_0) = \mathbf{p}_0$$

Check $\mathbf{p}_1$:

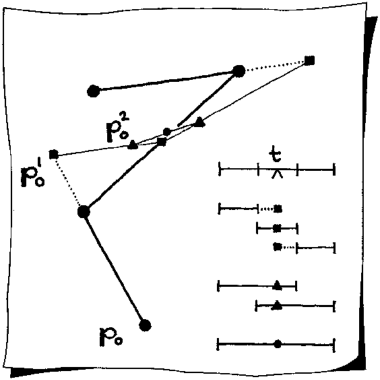
– Observe that the factors sum to one

– Both $\mathbf{p}_0^2(t_1) = \mathbf{p}_1$ and $\mathbf{p}_1^2(t_1) = \mathbf{p}_1$

$\Rightarrow \mathbf{p}_0^3(t_1) = \mathbf{p}_1$

Same idea for $\mathbf{p}_2$ and $\mathbf{p}_3$

Aitken's Algorithm



Finding the quadratic interpolants
– Same process works again:

$$\mathbf{p}_0^2(t) = \frac{t_2 - t}{t_2 - t_0}\mathbf{p}_0^1(t) + \frac{t - t_0}{t_2 - t_0}\mathbf{p}_1^1(t)$$

$$\mathbf{p}_1^2(t) = \frac{t_3 - t}{t_3 - t_1}\mathbf{p}_1^1(t) + \frac{t - t_1}{t_3 - t_1}\mathbf{p}_2^1(t)$$

New terms $\mathbf{p}_i^1$ are simply linear interpolants of the data

$$\mathbf{p}_1^1(t) = \frac{t_2 - t}{t_2 - t_1}\mathbf{p}_1 + \frac{t - t_1}{t_2 - t_1}\mathbf{p}_2$$

Aitken's Algorithm

Just as in the de Casteljau algorithm
Convenient to arrange the intermediate points in a triangular array:

$$\begin{array}{llll}
\mathbf{p}_0 & & & \\
\mathbf{p}_1 & \mathbf{p}_0^1 & & \\
\mathbf{p}_2 & \mathbf{p}_1^1 & \mathbf{p}_0^2 & \\
\mathbf{p}_3 & \mathbf{p}_2^1 & \mathbf{p}_1^2 & \mathbf{p}_0^3
\end{array}$$

Left-most column: given points (and parameter values)

Aitken's algorithm computes the points in each successive column

Point on the curve is $\mathbf{p}_0^3$

Aitken's Algorithm

Example: Evaluate at $t = 1.5$

Interpolating cubic through

$$\mathbf{p}_0 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \qquad \mathbf{p}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{p}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \qquad \mathbf{p}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$(t_0, t_1, t_2, t_3) = (0, 1, 2, 3)$$

$$\begin{matrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} & & & \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix} & & \\ \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0.125 \\ 0.375 \end{bmatrix} & \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} -0.5 \\ -1.5 \end{bmatrix} & \begin{bmatrix} -0.125 \\ -0.375 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{p}_0^3(1.5) \end{matrix}$$

A sampling of the computation of the intermediate points:

$$\mathbf{p}_0^1 = -0.5\mathbf{p}_0 + 1.5\mathbf{p}_1 \qquad \mathbf{p}_0^2 = 0.25\mathbf{p}_0^1 + 0.75\mathbf{p}_1^1 \qquad \mathbf{p}_0^3 = 0.5\mathbf{p}_0^2 + 0.5\mathbf{p}_1^2$$

Aitken's Algorithm

n^{th} degree interpolating curve

for $r = 1, \ldots, n$
  for $i = 0, \ldots, n - r$

$$\mathbf{p}_i^r(t) = \frac{t_{i+r} - t}{t_{i+r} - t_i}\mathbf{p}_i^{r-1}(t) + \frac{t - t_i}{t_{i+r} - t_i}\mathbf{p}_{i+1}^{r-1}(t)$$

Linear interpolation between $\mathbf{p}_i^{r-1}$ and $\mathbf{p}_{i+1}^{r-1}$ over $[t_{i+r}, t_i]$

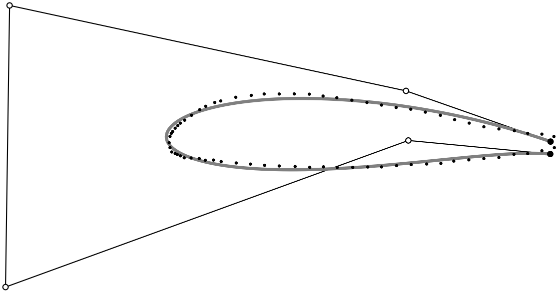
$\Rightarrow$ Affine map of interval onto line through $\mathbf{p}_i^{r-1}$ and $\mathbf{p}_{i+1}^{r-1}$

Example: See chapter introduction Figure

Polynomial interpolation is a *global* operation
– Every data point involved in calculation

Approximation

Some data not suited to interpolation
– Too many data points ⇒ Higher degree interpolation is ill-conditioned
– Data may be noisy



Approximation: curve passes “near” points
– Still captures shape suggested by given points

Approximation

Least squares approximation

Given: $l+1$ data points $\mathbf{p}_0, \ldots, \mathbf{p}_l$ and parameter values t_i

Find: polynomial curve $\mathbf{x}(t)$ of a given degree n
such that distances $\|\mathbf{p}_i - \mathbf{x}(t_i)\|$ are small

Ideal situation:

$\mathbf{p}_i = \mathbf{x}(t_i) \quad i = 0, \ldots, l \qquad \Rightarrow \qquad \mathbf{b}_0 B_0^n(t_i) + \ldots + \mathbf{b}_n B_n^n(t_i) = \mathbf{p}_i$

$$\begin{bmatrix} B_0^n(t_0) & \ldots & B_n^n(t_0) \\ & \vdots & \\ & \vdots & \\ B_0^n(t_l) & \ldots & B_n^n(t_l) \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 \\ \vdots \\ \vdots \\ \mathbf{p}_l \end{bmatrix}$$

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

Approximation

Least squares approximation continued

Assume number of data points $l >$ degree n of the curve
$\Rightarrow$ linear system is overdetermined

Multiply both sides by M^{T}:

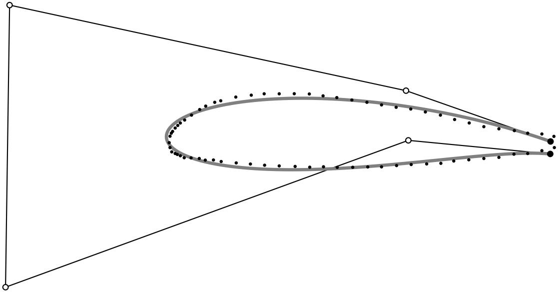
$$M^{\mathrm{T}} M\mathbf{B} = M^{\mathrm{T}}\mathbf{P}$$

Linear system with $n+1$ equations in $n+1$ unknowns
– Square and symmetric coefficient matrix $M^{\mathrm{T}}M$
– $M^{\mathrm{T}}M$ always invertible
– System of normal equations

Curve $\mathbf{B}$ is the *one* polynomial of degree n
which minimizes the sum of the $\|\mathbf{p}_i - \mathbf{x}(t_i)\|$

Approximation

Example: 79 data (noisy) points from a cross section of a wing
Parameter values selected to reflect the spacing of the data



Approximated by a least squares quintic

Choice of the "right" degree for this type of problem not easy
– Trial and error or application dependent

Finding the Right Parameters

Input to both curve interpolation and approximation:
1) data points $\mathbf{p}_i \quad i = 0, l$
2) associated parameter values t_i
In many applications parameter values must be chosen

Some choices:

Uniform set of parameters: $t_i = i/l$

Chord length parameters: parameters reflect the spacing of the data points

$$
\begin{aligned}
t_0 &= 0 \\
t_1 &= t_0 + \|\mathbf{p}_1 - \mathbf{p}_0\| \\
&\vdots \\
t_l &= t_{l-1} + \|\mathbf{p}_l - \mathbf{p}_{l-1}\|
\end{aligned}
$$

Finding the Right Parameters

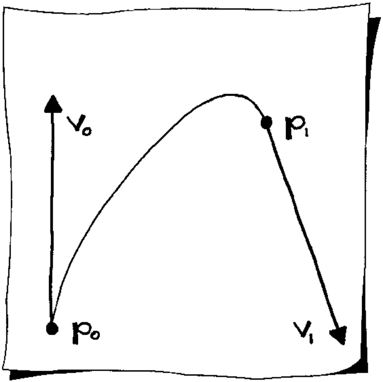
Normalize the parameters: scaling between zero and one

$$t_i = \frac{t_i - t_0}{t_l - t_0}$$

Chord length method superior to uniform method (mostly)
– Considers geometry of the data

Hermite Interpolation

Curve fitting to points and tangent vectors

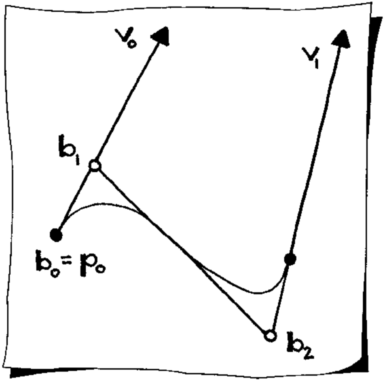


Given: two points $\mathbf{p}_0, \mathbf{p}_1$
and two tangent vectors $\mathbf{v}_0, \mathbf{v}_1$

Find: cubic polynomial interpolant $\mathbf{x}(t)$ such that

$$\mathbf{x}(0) = \mathbf{p}_0$$
$$\dot{\mathbf{x}}(0) = \mathbf{v}_0$$
$$\dot{\mathbf{x}}(1) = \mathbf{v}_1$$
$$\mathbf{x}(1) = \mathbf{p}_1$$

Hermite Interpolation



Write $\mathbf{x}(t)$ in cubic Bézier form

$$\mathbf{b}_0 = \mathbf{p}_0 \qquad \mathbf{b}_3 = \mathbf{p}_1$$

Recall endpoint derivative for Bézier curves:

$$\dot{\mathbf{x}}(0) = 3\Delta\mathbf{b}_0 \qquad \dot{\mathbf{x}}(1) = 3\Delta\mathbf{b}_2$$

$\Rightarrow$ Easily solve for $\mathbf{b}_1$ and $\mathbf{b}_2$:

$$\mathbf{b}_1 = \mathbf{p}_0 + \frac{1}{3}\mathbf{v}_0 \qquad \mathbf{b}_2 = \mathbf{p}_1 - \frac{1}{3}\mathbf{v}_1$$

Hermite Interpolation

Rewrite interpolant so given data appear *explicitly*

$$\mathbf{x}(t) = \mathbf{p}_0 B_0^3(t) + (\mathbf{p}_0 + \frac{1}{3}\mathbf{v}_0) B_1^3(t) + (\mathbf{p}_1 - \frac{1}{3}\mathbf{v}_1) B_2^3(t) + \mathbf{p}_1 B_3^3(t).$$

Rearrange and form cubic Hermite polynomials $H_i^3(t)$:

$$\mathbf{x}(t) = \mathbf{p}_0 H_0^3(t) + \mathbf{v}_0 H_1^3(t) + \mathbf{v}_1 H_2^3(t) + \mathbf{p}_1 H_3^3(t)$$

$$\begin{aligned} H_0^3(t) &= B_0^3(t) + B_1^3(t) \\ H_1^3(t) &= \frac{1}{3} B_1^3(t) \\ H_2^3(t) &= -\frac{1}{3} B_2^3(t) \\ H_3^3(t) &= B_2^3(t) + B_3^3(t) \end{aligned}$$

Cardinal form for the interpolant to point and tangent data

Length of $\mathbf{v}_0$ and $\mathbf{v}_1$ important factor for curve's shape