Bézier Splines

Sven

January 9, 2023

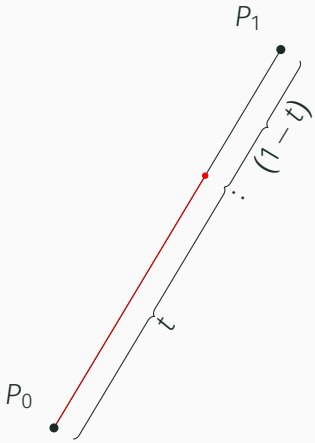
Chair i1, Theoretical Computer Science

Motivation

- Smooth and continuous curves that can be scaled indefinitely
- Relatively easy to calculate
- Essential for computer-aided design (CAD)
- Creation of illustrations (e.g. SVG)
- Used in game development (e.g. camera paths)
- Design of fonts

Bézier Curves

Linear Bézier Curves i



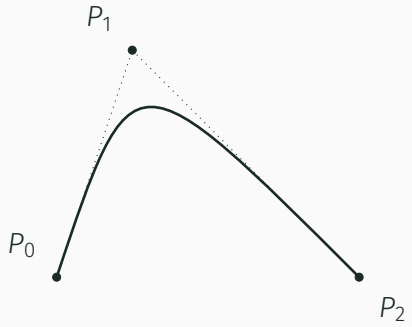
$$\boxed{t = \frac{2}{3}}$$

Linear Bézier Curves ii

$$B(t) = P_0 + t(P_1 - P_0) = (1 - t)P_0 + tP_1,\ 0 \leq t \leq 1$$

- equivalent to linear interpolation of points P_0 and P_1

Quadratic Bézier Curves i



$$P_0 = (0, 0)$$
$$P_1 = (1, 3)$$
$$P_2 = (4, 0)$$

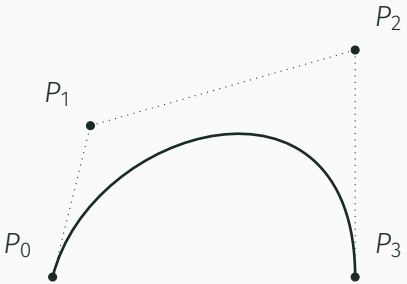
- P_1 is a control point

Quadratic Bézier Curves ii

$$B(t) = (1-t) \cdot \underbrace{\big((1-t)P_0 + tP_1\big)}_{\text{Linear Bézier Curve}} + t \cdot \underbrace{\big((1-t)P_1 + tP_2\big)}_{\text{Linear Bézier Curve}}, \ 0 \leq t \leq 1$$

- linear interpolant of corresponding points on linear Bézier curves from P_0 to P_1 and from P_1 to P_2

Cubic Bézier Curves i

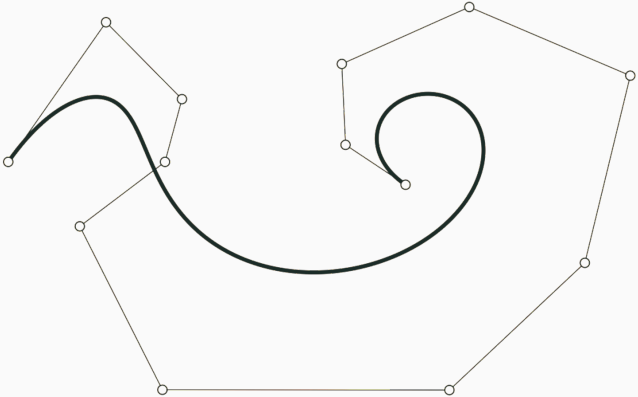


$$
\begin{aligned}
P_0 &= (0, 0) \\
P_1 &= (\tfrac{1}{2}, 2) \\
P_2 &= (4, 3) \\
P_3 &= (4, 0)
\end{aligned}
$$

Cubic Bézier Curves ii

$$B(t) = (1-t) \cdot \underbrace{B_{P_0,P_1,P_2}(t)}_{\substack{\text{Quadratic} \\ \text{Bézier Curve}}} + t \cdot \underbrace{B_{P_1,P_2,P_3}(t)}_{\substack{\text{Quadratic} \\ \text{Bézier Curve}}}, \ 0 \leq t \leq 1$$

Higher Order Bézier Curves



Bézier Curve of degree 12 (dodecic)[5]

Definitions for Bézier Curves

1. Recursive Definition (De Casteljau)

$$B_{P_0}(t) = P_0$$
$$B(t) = B_{P_0 P_1 \ldots P_n}(t) = (1-t)B_{P_0 P_1 \ldots P_{n-1}}(t) + t \cdot B_{P_1 \ldots P_n}(t)$$

De Casteljau's Algorithm - Evaluate

Idea

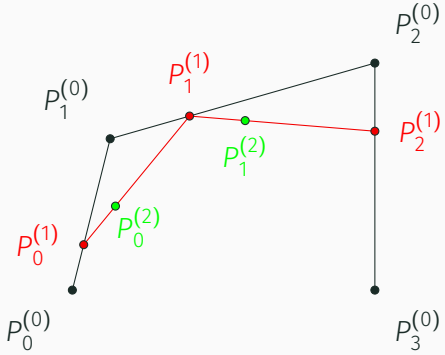
Split a Bézier Curve B of degree n into two Bézier Curves of degree $n - 1$ and interpolate. Repeat until there is nothing left to interpolate.

De Casteljau's Algorithm - Evaluate

Idea

Split a Bézier Curve B of degree n into two Bézier Curves of degree $n-1$ and interpolate. Repeat until there is nothing left to interpolate.

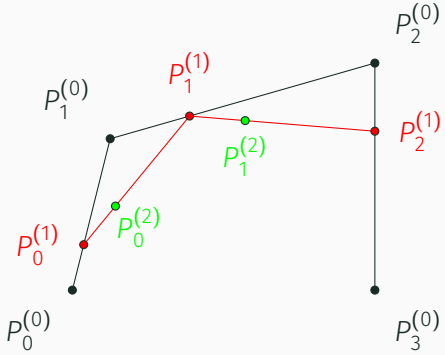
Let $t_0 = \frac{1}{3}$.

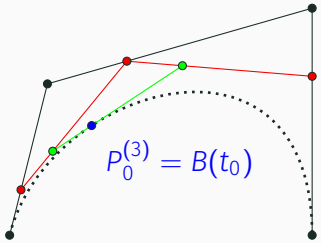


Idea

Split a Bézier Curve B of degree n into two Bézier Curves of degree $n-1$ and interpolate. Repeat until there is nothing left to interpolate.

Let $t_0 = \frac{1}{3}$.





De Casteljau's Algorithm - Split

Idea

Split a Bézier Curve B with control points $P_0, P_1, \ldots, P_n$ into two Bézier Curves with control points

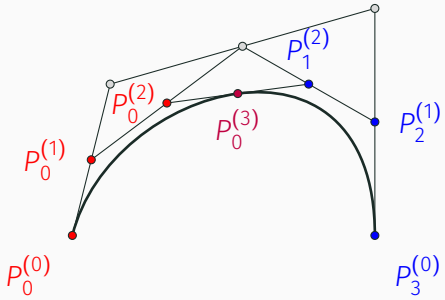
1. $P_0^{(0)}, P_0^{(1)}, \ldots, P_0^{(n)}$
2. $P_0^{(n)}, P_1^{(n-1)}, \ldots, P_n^{(0)}$

De Casteljau's Algorithm - Split

Idea

Split a Bézier Curve B with control points $P_0, P_1, \ldots, P_n$ into two Bézier Curves with control points

1. $P_0^{(0)}, P_0^{(1)}, \ldots, P_0^{(n)}$
2. $P_0^{(n)}, P_1^{(n-1)}, \ldots, P_n^{(0)}$

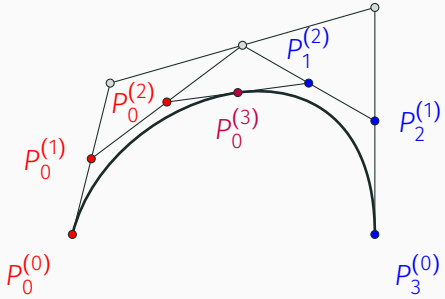


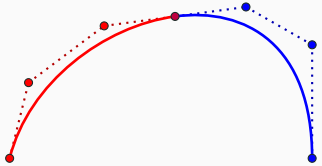
De Casteljau's Algorithm - Split

Idea

Split a Bézier Curve B with control points $P_0, P_1, \ldots, P_n$ into two Bézier Curves with control points

1. $P_0^{(0)}, P_0^{(1)}, \ldots, P_0^{(n)}$
2. $P_0^{(n)}, P_1^{(n-1)}, \ldots, P_n^{(0)}$





Transformation of Recursive Definition

Bézier Curve of degree 2 (quadratic)

$$(1-t)\cdot\big((1-t)P_0+tP_1\big)+t\big((1-t)P_1+tP_2\big)$$

Transformation of Recursive Definition

Bézier Curve of degree 2 (quadratic)

$$
\begin{aligned}
&(1-t)\cdot\big((1-t)P_0+tP_1\big)+t\big((1-t)P_1+tP_2\big)\\
&=(1-t)^2P_0+(t-t^2)P_1+(t-t^2)P_1+t^2P_2
\end{aligned}
$$

Transformation of Recursive Definition

Bézier Curve of degree 2 (quadratic)

$$
\begin{aligned}
& (1-t)\cdot\big((1-t)P_0 + tP_1\big) + t\big((1-t)P_1 + tP_2\big) \\
= {} & (1-t)^2P_0 + (t-t^2)P_1 + (t-t^2)P_1 + t^2P_2 \\
= {} & \quad P_0\cdot(1-t)^2 \\
& + \quad P_1\cdot 2(t-t^2) \\
& + \quad P_2\cdot t^2
\end{aligned}
$$

Transformation of Recursive Definition

Bézier Curve of degree 2 (quadratic)

$$
\begin{aligned}
& (1-t)\cdot\big((1-t)P_0 + tP_1\big) + t\big((1-t)P_1 + tP_2\big) \\
= {} & (1-t)^2P_0 + (t-t^2)P_1 + (t-t^2)P_1 + t^2P_2 \\
= {} & \quad P_0\cdot(1-t)^2 \\
& + \quad P_1\cdot 2(t-t^2) \\
& + \quad P_2\cdot t^2 \\
= {} & \quad P_0\cdot\tbinom{2}{0}\cdot t^0\cdot(1-t)^2 \\
& + \quad P_1\cdot\tbinom{2}{1}\cdot t^1\cdot(1-t)^1 \\
& + \quad P_2\cdot\tbinom{2}{2}\cdot t^2\cdot(1-t)^0
\end{aligned}
$$

Transformation of Recursive Definition

Bézier Curve of degree 2 (quadratic)

$$
\begin{aligned}
& (1-t)\cdot\big((1-t)P_0 + tP_1\big) + t\big((1-t)P_1 + tP_2\big) \\
&= (1-t)^2P_0 + (t-t^2)P_1 + (t-t^2)P_1 + t^2P_2 \\
&= \quad P_0\cdot(1-t)^2 \\
&\quad + \quad P_1\cdot 2(t-t^2) \\
&\quad + \quad P_2\cdot t^2 \\
&= \quad P_0\cdot\tbinom{2}{0}\cdot t^0\cdot(1-t)^2 \\
&\quad + \quad P_1\cdot\tbinom{2}{1}\cdot t^1\cdot(1-t)^1 \\
&\quad + \quad P_2\cdot\tbinom{2}{2}\cdot t^2\cdot(1-t)^0 \\
&= \sum_{i=0}^{2} P_i\cdot\binom{2}{i}\cdot t^i\cdot(1-t)^{2-i}
\end{aligned}
$$

Definitions for Bézier Curves

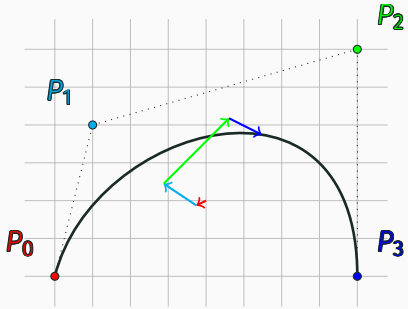
1. Recursive Definition (De Casteljau)

$$B_{P_0}(t) = P_0$$
$$B(t) = B_{P_0P_1\ldots P_n}(t) = (1-t)B_{P_0P_1\ldots P_{n-1}}(t) + t \cdot B_{P_1\ldots P_n}(t)$$

2. Bernstein Polynomial

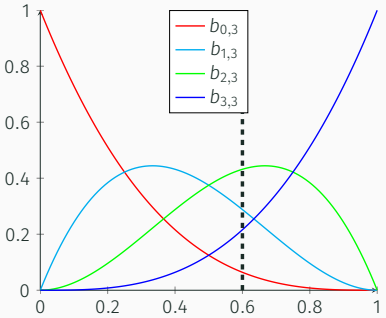
$$B(t) = \sum_{i=0}^{n} \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} \cdot P_i = \sum_{i=0}^{n} \underbrace{b_{i,n}(t)}_{\substack{\text{Bernstein} \\ \text{Basis} \\ \text{Polynomial}}} \cdot P_i$$

Interpretation of Bernstein Basis Polynomials



$t = 0.6$

$\sum_{i=0}^{n} b_{i,n}(t) \cdot \mathrm{P_i}$



Bernstein basis polynomials for 3rd degree curve

Polynomial Coefficients

$$
\begin{aligned}
B(t) = t^3(&- P_0 + 3P_1 - 3P_2 + P_3)\\
&+ t^2(\ \ 3P_0 - 6P_1 + 3P_2\ \ \ \ \ \ )\\
&+ t\ \ (-3P_0 + 3P_1\ \ \ \ \ \ \ \ \ \ )\\
&+ 1\ (\ \ \ P_0\ \ \ \ \ \ \ \ \ \ \ \ \ )\\
&\text{for } 0 \leq t \leq 1
\end{aligned}
$$

Polynomial Coefficients

$$
\begin{aligned}
B_x(t) = t^3(&-P_{0x} + 3P_{1x} - 3P_{2x} + P_{3x}) \\
&+ t^2(\ 3P_{0x} - 6P_{1x} + 3P_{2x}\ ) \\
&+ t\ (-3P_{0x} + 3P_{1x}\ ) \\
&+ 1\ (\ P_{0x}\ ) \\
\text{for } 0 \leq t &\leq 1
\end{aligned}
$$

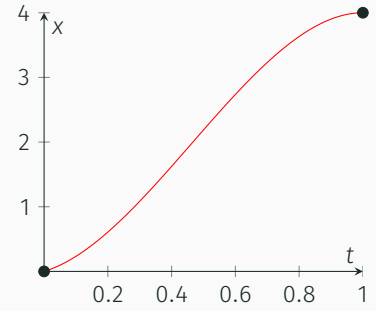
Polynomial Coefficients

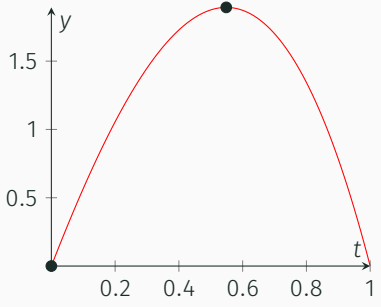
$$
\begin{aligned}
B_x(t) = t^3(&- P_{0x} + 3P_{1x} - 3P_{2x} + P_{3x}) \\
&+ t^2(\ 3P_{0x} - 6P_{1x} + 3P_{2x} \ ) \\
&+ t \ (-3P_{0x} + 3P_{1x} \ ) \\
&+ 1 \ (\ P_{0x} \ )
\end{aligned}
$$

for $0 \leq t \leq 1$

$$
\begin{aligned}
B_y(t) = t^3(&- P_{0y} + 3P_{1y} - 3P_{2y} + P_{3y}) \\
&+ t^2(\ 3P_{0y} - 6P_{1y} + 3P_{2y} \ ) \\
&+ t \ (-3P_{0y} + 3P_{1y} \ ) \\
&+ 1 \ (\ P_{0y} \ )
\end{aligned}
$$

for $0 \leq t \leq 1$





Definitions for Bézier Curves

1. Recursive Definition (De Casteljau)

$$B_{P_0}(t) = P_0$$
$$B(t) = B_{P_0P_1\ldots P_n}(t) = (1-t)B_{P_0P_1\ldots P_{n-1}}(t) + t \cdot B_{P_1\ldots P_n}(t)$$

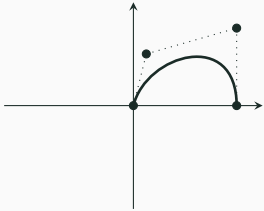
2. Bernstein Polynomial

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} \cdot P_i = \sum_{i=0}^{n} b_{i,n}(t) \cdot P_i$$

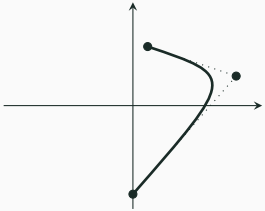
3. Polynomial Coefficients

$$x(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n,$$
$$y(t) = b_0 + b_1 t + b_2 t^2 + \cdots + b_n t^n$$

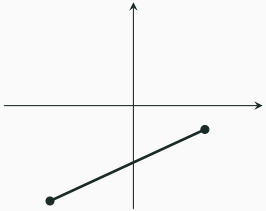
Derivatives



Bezier Curve



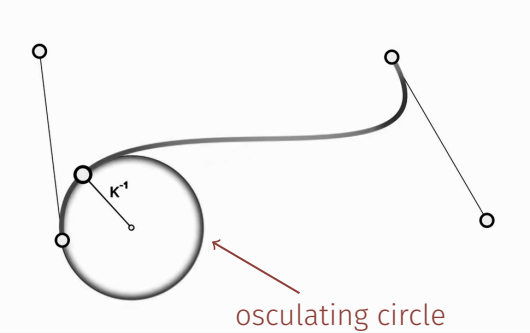
First derivative
(velocity)



Second derivative
(acceleration)

Curvature

$$\kappa = \frac{\det(B', B'')}{\|B'\|^3}, \qquad r = \kappa^{-1}$$

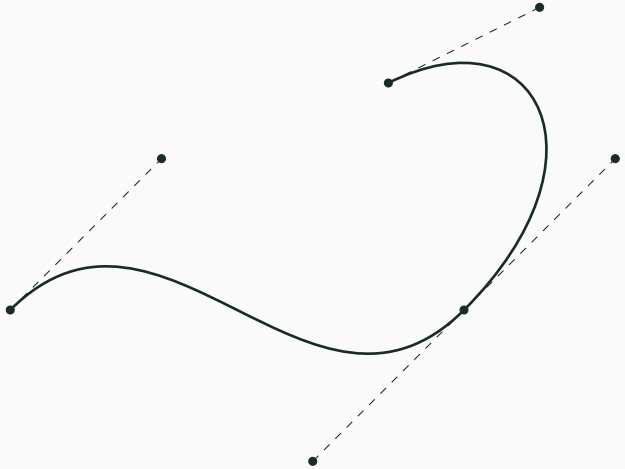


Source: [5]

Bézier Splines

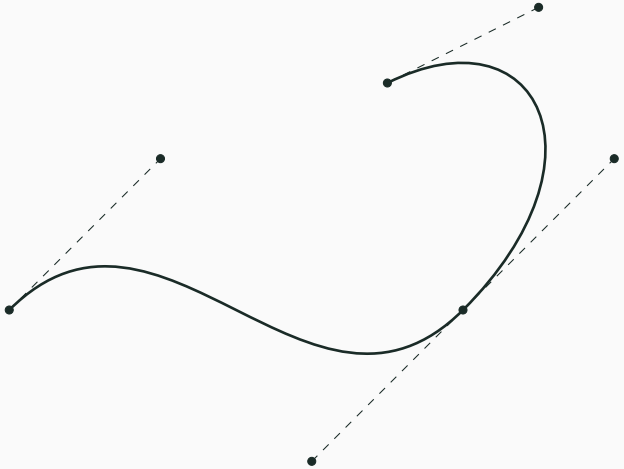
Why Splines?

- Higher order are curves expensive to calculate



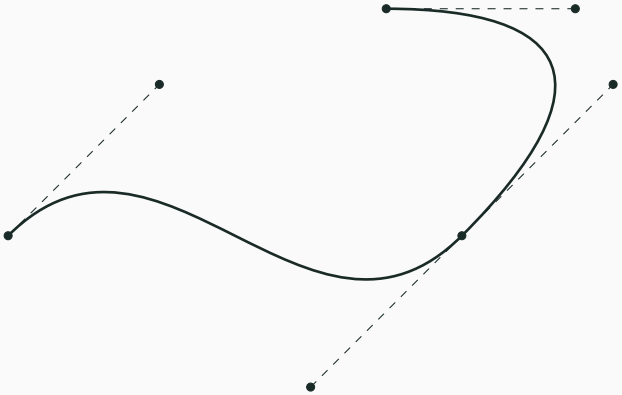
Why Splines?

- Higher order are curves expensive to calculate
- Local control



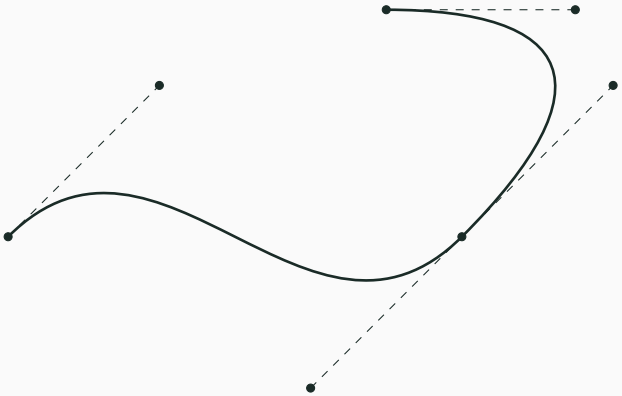
Why Splines?

- Higher order are curves expensive to calculate
- Local control



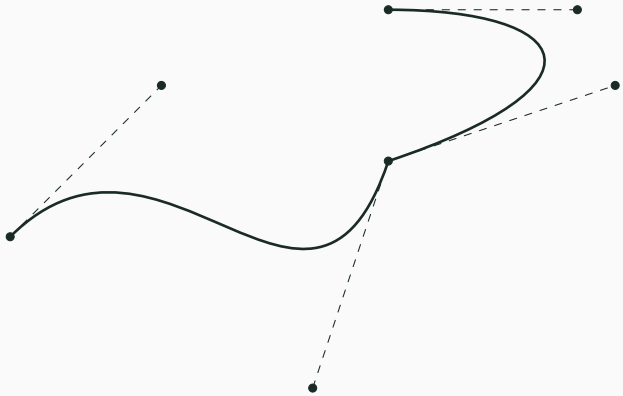
Why Splines?

- Higher order are curves expensive to calculate
- Local control
- Curve through points



Why Splines?

- Higher order are curves expensive to calculate
- Local control
- Curve through points

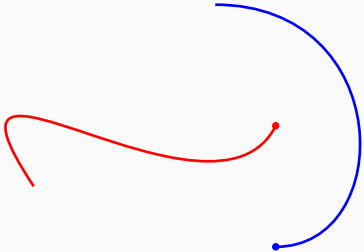


C^0 Continuity

Bézier Curves meet at the same point (positional continuity)

C^0 Continuity

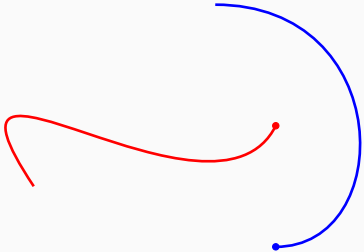
Bézier Curves meet at the same point (positional continuity)



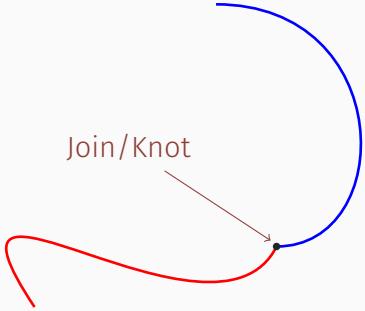
not C^0 continuous

C^0 Continuity

Bézier Curves meet at the same point (positional continuity)

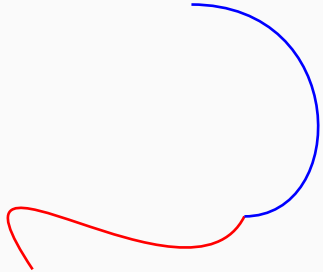


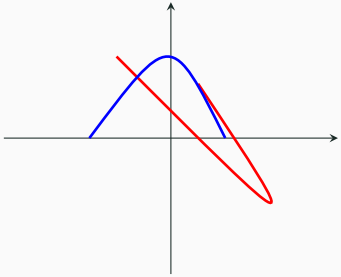
not C^0 continuous



C^0 continuous

Problem with C^0

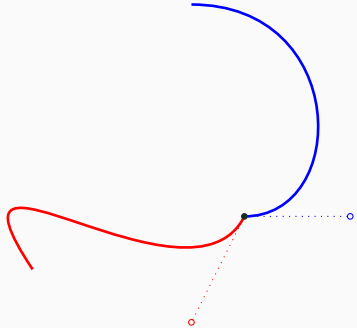




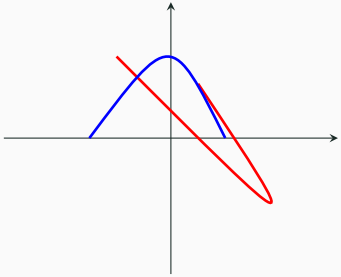
- First derivative (velocity) is *not* continuous

First derivative

Problem with C^0

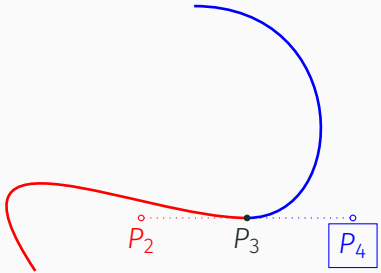


- First derivative (velocity) is *not* continuous
- Tangent Points are split/broken

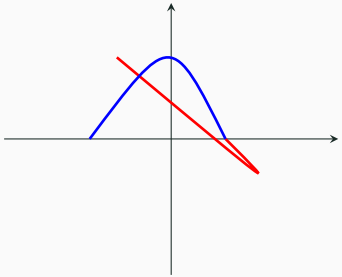


First derivative

C^1 Continuity

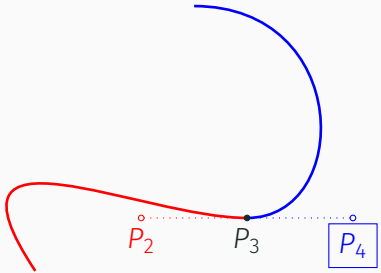


- First derivative (velocity) is continuous
- Tangent Points are mirrored
- $P_4 = 2P_3 - P_2$

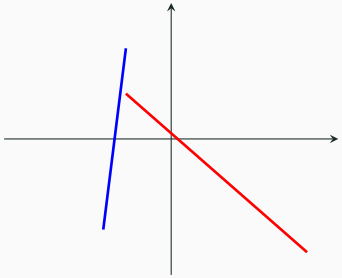


First derivative

C^1 Continuity

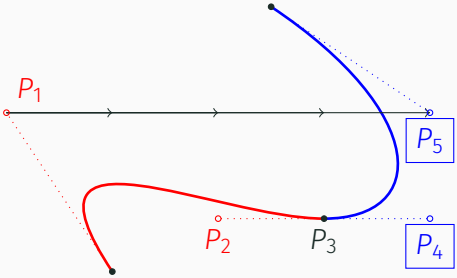


- First derivative (velocity) is continuous
- Tangent Points are mirrored
- $P_4 = 2P_3 - P_2$
- Second derivative (acceleration) is *not* continuous

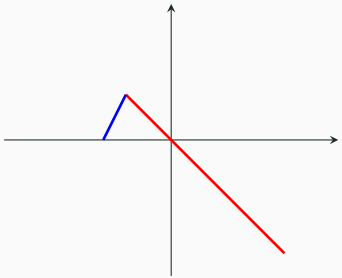


Second derivative

C^2 Continuity

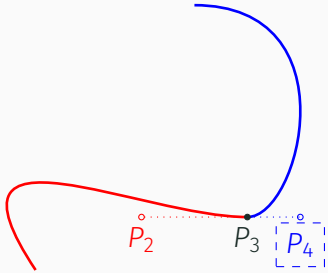


- $P_5 = P_1 + 4(P_3 - P_2)$



Second derivative

G^1 Continuity

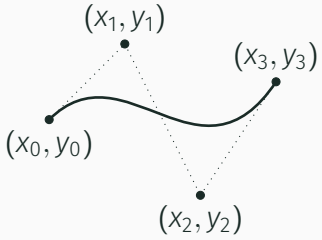


- Less strict than C^1
- Tangent points are collinear
- Is also called tangent continuity

Bézier Curves in PostScript

The `curveto` Operator

- Appends the path by a cubic Bézier Curve
- Syntax: $x_1\ y_1\ x_2\ y_2\ x_3\ y_3$ `curveto`
- Is used internally for all curves (e.g. `arc`)
- Relative version: `rcurveto`



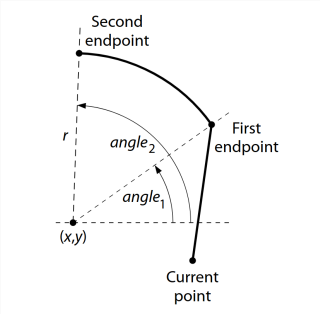
The `setflat` Operator

- Sets the flatness parameter in the graphics state
- Flatness is the error tolerance
- Syntax: *num* `setflat`
- Measured in output device pixels
- Range is 0.2 to 100.0
- Flattening is automatically done when path is used to control painting (e.g. `stroke`)
  - Explicit: `flattenpath`

Exaggerated example of a flattened curve

The `arc` Operator

- Appends the path by an arc of a circle
- Syntax: *x y angle*$_1$ *angle*$_2$ `arc`
- Possibly preceded by a straight line
- Only an approximation of an arc

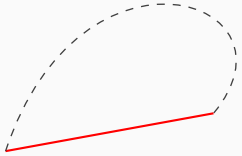


Source: [1]

Linear Approximation

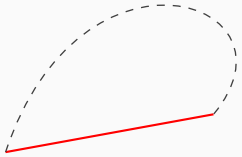
Recursive Subdivision

1. Start approximating curve with one line



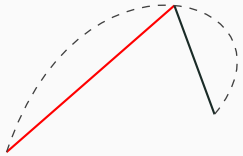
Recursive Subdivision

1. Start approximating curve with one line
2. Check whether curve is flat enough
   - 2.1 No, split curve into left and right with $t_0 = 0.5$ and check again
   - 2.2 Yes, done.



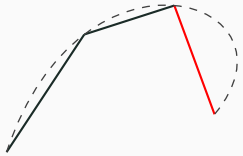
Recursive Subdivision

1. Start approximating curve with one line
2. Check whether curve is flat enough
   - 2.1 No, split curve into left and right with $t_0 = 0.5$ and check again
   - 2.2 Yes, done.



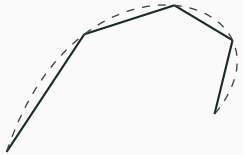
Recursive Subdivision

1. Start approximating curve with one line
2. Check whether curve is flat enough
   - 2.1 No, split curve into left and right with $t_0 = 0.5$ and check again
   - 2.2 Yes, done.



Recursive Subdivision

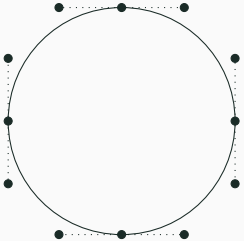
1. Start approximating curve with one line
2. Check whether curve is flat enough
   - 2.1 No, split curve into left and right with $t_0 = 0.5$ and check again
   - 2.2 Yes, done.



Approximation of Circular Arcs

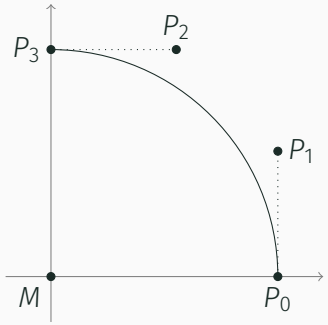
Approximation of a Circle

- Use inexpensive cubic Bézier Curves
- Not a perfect circle but visually not noticeable
- Commonly approximated with 4 cubic Bézier Curves



90° Arc

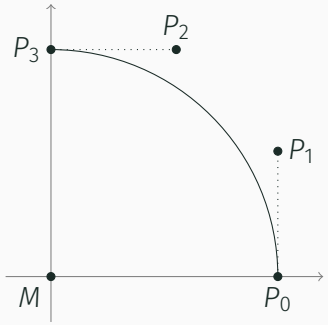
$$
\begin{aligned}
M &= (0 \mid 0) \\
P_0 &= (r \mid 0) \\
P_1 &= (r \mid r\kappa) \\
P_2 &= (r\kappa \mid r) \\
P_3 &= (0 \mid r)
\end{aligned}
$$



- Slopes of tangents at P_0 and P_3 coincide with the arc

90° Arc

$$
\begin{aligned}
M &= (0 \mid 0) \\
P_0 &= (r \mid 0) \\
P_1 &= (r \mid r\kappa) \\
P_2 &= (r\kappa \mid r) \\
P_3 &= (0 \mid r)
\end{aligned}
$$



- Slopes of tangents at P_0 and P_3 coincide with the arc
- Find $\kappa \in [0, 1]$ with least approximation error

Cubic Bézier Curve

$$x_\kappa = (1-t)^3 r + 3t(1-t)^2 r + 3t^2(1-t)\kappa r,$$
$$y_\kappa = t^3 r + 3t^2(1-t) r + 3t(1-t)^2 \kappa r$$

Error

$$f_\kappa(t) = \frac{x_\kappa(t)^2 + y_\kappa(t)^2 - r^2}{r^2}$$

$f_\kappa(t) = 0$ iff $(x_\kappa \mid y_\kappa)$ is on the circle

Error Analysis ii

We want the curve to be on the circle at the bisector of the angle.

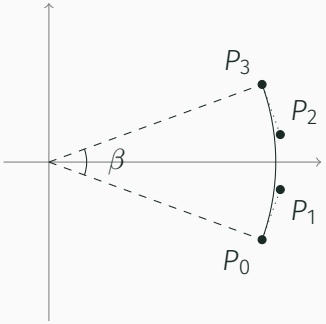
$$0 = f_\kappa(0.5) = \frac{9\kappa^2 + 24\kappa - 16}{32} \implies \kappa = \frac{4}{3}(\sqrt{2} - 1) \approx 0.55228$$

- $\kappa \approx 0.55228$ frequently referred to as magic number
- No error at $t \in \{0, 0.5, 1\}$, else positive
- Maximum error $f_\kappa(t) = 5.45 \cdot 10^{-4}$

$< 90°$ Arc

- Let A be an arc of sweep $\beta < 90°$ and radius $r = 1$
- Assume A is bisected by the x-axis
- Let $\phi = \frac{\beta}{2}$
- Let B be a Bézier Curve

$$B(\frac{1}{2}) = (1 \mid 0)$$
$$P_0 = (\cos(\phi) \mid -\sin(\phi))$$
$$P_3 = (\cos(\phi) \mid \sin(\phi))$$



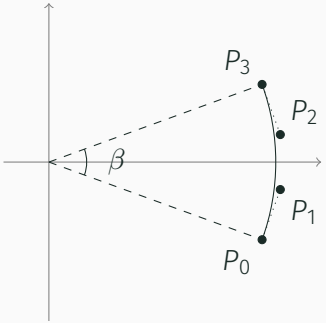
$< 90°$ Arc

Force slope of tangent of B at P_0 to coincide with those of A

$$m_0 = \frac{y_0}{x_0}, \quad m_0^t = \frac{-x_0}{y_0} = \frac{(y_0 - y_1)}{(x_0 - x_1)}$$

$$x_0 x_1 = 1 - y_0 y_1$$

Analog for P_3.

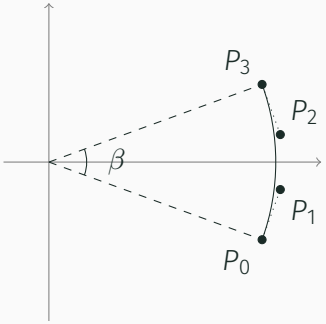


$< 90°$ Arc

After substituting x_3 with x_0, y_3 with $-y_0$ and y_2 with $-y_1$, one ends up with

$$x_0 = \cos(\frac{\beta}{2}) \qquad y_0 = \sin(\frac{\beta}{2})$$
$$x_3 = x_0 \qquad y_3 = -y_0$$
$$x_1 = \frac{4 - x_0}{3} \qquad y_1 = \frac{(1 - x_0)(3 - x_0)}{3y_0}$$
$$x_2 = x_1 \qquad y_2 = -y_1$$

Curve can be rotated, scaled and translated to approximate any arc.



Thanks for your attention! Any questions?

Sources i

CORPORATE Adobe Systems Inc.
PostScript language reference.
Addison-Wesley Longman Publishing Co., Inc., 1999.

ArtifexSoftware.
ghostpdl.
`https://github.com/ArtifexSoftware/ghostpdl`.
Accessed: 2022-12-14.

Sources ii

Richard A DeVeneza.
How to determine the control points of a bézier curve that approximates a small circular arc.
https://www.tinaja.com/glib/bezcirc2.pdf.
Accessed: 2022-12-27.

Kaspar Fischer.
Piecewise linear approximation of bezier curves.
In *HTTP://HCKLBRRFNN. WORDPRESS. COM/2012/08/20/PIECEWISE-LINEAR-APPROXIMATION-OF-BEZIER-CURVES/*.
Citeseer, 2000.

Freya Holmér.
The continuity of splines.
https://www.youtube.com/watch?v=jvPPXbo87ds.
Accessed: 2022-12-09.

Aleksas Riškus.
Approximation of a cubic bézier curve by circular arcs and vice versa.
Information Technology and control, 35(4), 2006.