# Designing Interpolating Curves

https://github.com/danielepanozzo/cg

Acknowledgement: Olga Sorkine-Hornung, Alexander Sorkine-Hornung, Ilya Baran

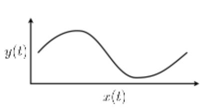
1

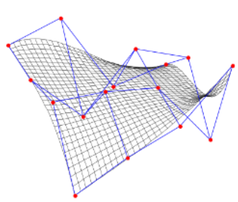
## How do we model shapes?



2

## Building blocks: curves and surfaces





3

## Modeling curves

- We need **mathematical concepts** to characterize the desired curve properties
- Notions from **curve geometry** help with designing user interfaces for curve creation and editing

4

## Line representation (review)

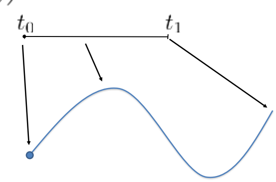
- Parametric equation: Consider the line through the point $P_0 = (x_0, y_0, z_0)$ and parallel to the non-zero vector $v = (a, b, c)$
- A point $P = (x, y, z)$ is on the line if and only if
  - $(x - x_0, y - y_0, z - z_0) = t * v$
    - $x = x_0 + t * a$
    - $y = y_0 + t * b$
    - $z = z_0 + t * c$

5

## 2D parametric curve

$$\mathbf{p}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \ t \in [t_0, t_1]$$

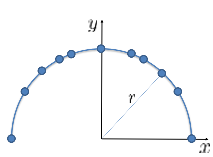
- $\mathbf{p}(t)$ must be continuous




6

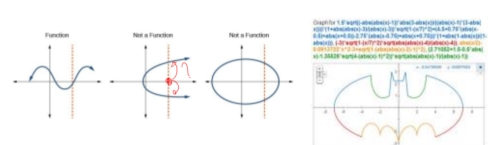
## A curve can be parameterized in many different ways

$$\begin{pmatrix} r\cos t \\ r\sin t \end{pmatrix}, t \in [0, \pi] \qquad \begin{pmatrix} -rt \\ r\sqrt{1-t^2} \end{pmatrix}, t \in [-1, 1]$$




7

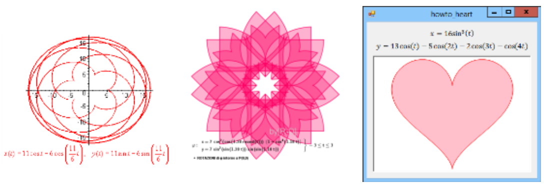
## Parameterized curves




8

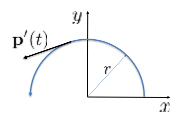
## Parameterized curves

- A parameterized curve can have any shape



9

## Tangent vector




$$\mathbf{p}(t) = \begin{pmatrix} r\cos t \\ r\sin t \end{pmatrix}$$

$$\mathbf{p}'(t) = \begin{pmatrix} -r\sin t \\ r\cos t \end{pmatrix}$$

$$\|\mathbf{p}'(t)\| = \text{speed}$$

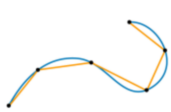
$$\frac{\mathbf{p}'(t)}{\|\mathbf{p}'(t)\|} = \mathbf{T}(t) = \text{unit tangent}$$

10

## Arc length

- How long is the curve between $t_0$ and $t$? How far does the particle travel?
- Speed is $\|\mathbf{p}'(t)\|$, so:

$$s(t) = \int_{t_0}^{t} \|\mathbf{p}'(t)\| dt$$



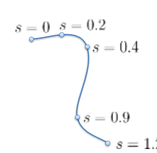
- Speed is nonnegative, so $s(t)$ is non-decreasing

11

## Arc length parameterization

- Every curve has a natural parameterization:

$$\mathbf{p}(s), \text{ such that } \|\mathbf{p}'(s)\| = 1$$

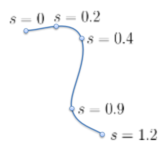



12

# Arc length parameterization

- Every curve has a natural parameterization:

$$\mathbf{p}(s), \text{ such that } \|\mathbf{p}'(s)\| = 1$$

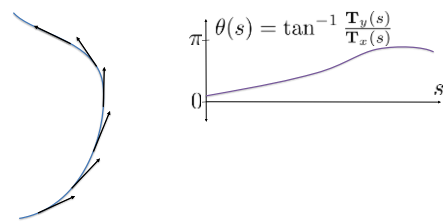



- Isometry between parameter domain and curve
- Tangent vector is unit-length: $\mathbf{p}'(s) = \mathbf{T}(s)$

13

# Curvature

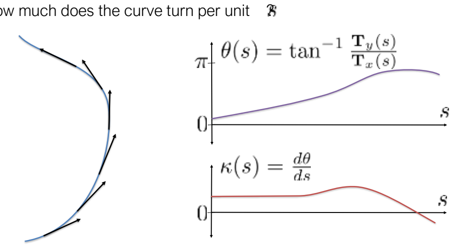
- How much does the curve turn per unit $s$




14

# Curvature

- How much does the curve turn per unit $s$




15

# Curvature of a circle

$\kappa(s) = \frac{d\theta}{ds} = \frac{1}{r}$

- Curvature of a circle:

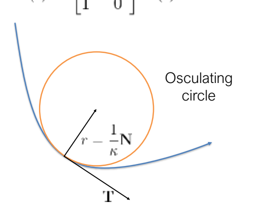
$$\mathbf{p}(s) = \begin{pmatrix} r\cos(s/r) \\ r\sin(s/r) \end{pmatrix}$$

$$\mathbf{p}'(s) = \begin{pmatrix} -\sin(s/r) \\ \cos(s/r) \end{pmatrix}$$

$$\theta(s) = \tan^{-1} \frac{\cos(s/r)}{-\sin(s/r)} = s/r - \pi/2$$

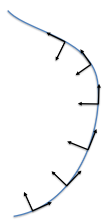
$$\kappa(s) = 1/r$$

$\mathbf{N}(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{T}(t)$ = Unit Normal




16

## Frenet Frame



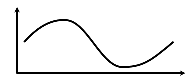
$$\mathbf{T}'(s) = \kappa(s)\mathbf{N}(s)$$
$$\mathbf{N}'(s) = -\kappa(s)\mathbf{T}(s)$$

$$\begin{pmatrix}\mathbf{T}'\\ \mathbf{N}'\end{pmatrix} = \begin{pmatrix}0 & \kappa\\ -\kappa & 0\end{pmatrix}\begin{pmatrix}\mathbf{T}\\ \mathbf{N}\end{pmatrix}$$

17

## Recap on parametric curves

$$\mathbf{p}(t) = \begin{pmatrix}x(t)\\ y(t)\end{pmatrix},\ t \in [t_0, t_1]$$



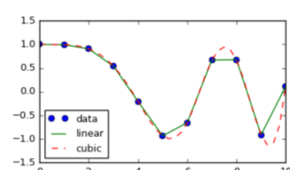
$$\|\mathbf{p}'(t)\| = \text{speed}$$
$$s(t) = \int_{t_0}^{t} \|\mathbf{p}'(t)\| dt$$
$$\kappa(s) = \frac{d\theta}{ds} = \mathbf{T}'(s) \cdot \mathbf{N}(s) \qquad \frac{\mathbf{p}'(t)}{\|\mathbf{p}'(t)\|} = \mathbf{T}(t)$$

18

Designging Curves:
Polynomials and Interpolation

19

## Interpolation

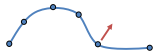



20

## Basic idea for curve design

- User gives us points. We need to connect the dots in a smooth way.

- The dots stay as “handles” on the curve. User can move the dots and the curve follows.

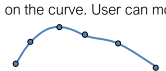


21

## Basic idea for curve design

- User gives us points. We need to connect the dots in a smooth way.

- The dots stay as “handles” on the curve. User can move the dots and the curve follows.



22

## Polynomial curves

- Polynomials

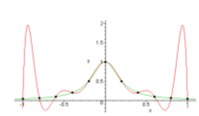
$$\mathbf{p}(t) = \begin{pmatrix} x_0 + x_1 t + x_2 t^2 + \cdots \\ y_0 + y_1 t + y_2 t^2 + \cdots \end{pmatrix} = \\ = \mathbf{p}_0 + \mathbf{p}_1 t + \mathbf{p}_2 t^2 + \cdots$$

- For degree $d$ we need $d+1$ points (“coefficients”)

23

## Polynomial curves

- For n data points, an approx. n-order polynomial will go through all the data points smoothly
- Often though, the interpolation behavior near the edges is problematic

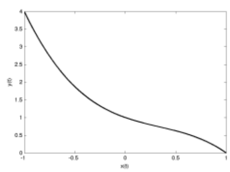


24

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 - t + t^2 - t^3$$



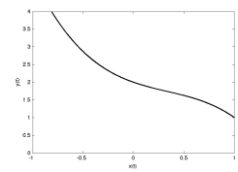
- Control shape?

25

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 2 - t + t^2 - t^3$$



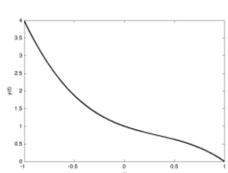
- Control shape?

26

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 - t + t^2 - t^3$$



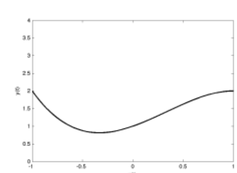
- Control shape?

27

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 + t + t^2 - t^3$$



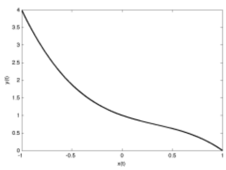
- Control shape?

28

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 - t + t^2 - t^3$$



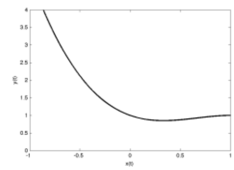
- Control shape?

29

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 - t + 2t^2 - t$$



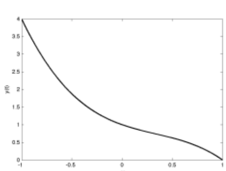
- Control shape?

30

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 - t + t^2 - t^3$$



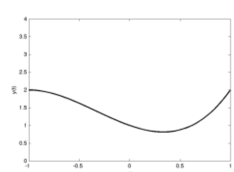
- Control shape?

31

## Polynomials

- Parametric form with polynomials

$$x(t) = t$$
$$y(t) = 1 - t + t^2 + t^3$$



- Control shape?
- Monomial coefficients are not intuitive

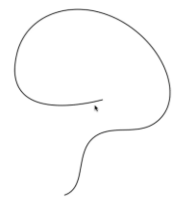
32

## Polynomial curves

- Idea: Specify a sequence of controls points, then the shave will be according to the control points

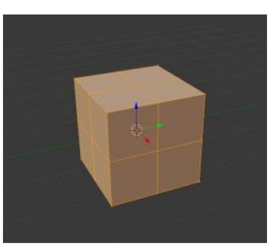
33

## Keynote Demo



34

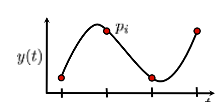
## Blender Demo



35

## Interpolation (1D)

- Interpolate control points
- Find polynomial with $y(t_i) = p_i$



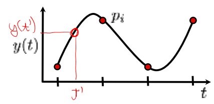

36

# Interpolation (1D)

- Interpolate control points
- Find polynomial with $y(t_i) = p_i$

$$y(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$$

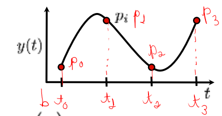
$$\begin{pmatrix} 1 & t & t^2 & t^3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = y(t)$$




37

# Interpolation (1D)

- Interpolate control points
- Find polynomial with $y(t_i) = p_i$
- Solve




$$\overset{A}{\begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \end{pmatrix}} \overset{x}{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}} = \overset{b}{\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}}$$

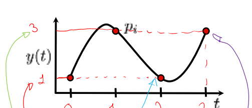
Vandermonde matrix

$\rightarrow Ax = b$

38

# Interpolation (1D)

- Interpolate control points
- Find polynomial with $y(t_i) = p_i$
- Solve

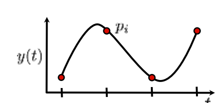



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 1 \\ 3 \end{pmatrix}$$

39

# Interpolation (1D)

- Interpolate control points
- Find polynomial with $y(t_i) = p_i$
- Solve




$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \end{pmatrix}^{-1} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

40

## Interpolation (1D)

- Polynomial fitting can be done explicitly:

$$y(t) = \sum_i p_i \prod_{j \neq i} \frac{t - t_j}{t_i - t_j} = \sum_i p_i L_i(t)$$

- Products $L_i(t)$ are called Lagrange polynomials

41

## Example of 1D Interpolation

42

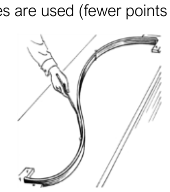
## Recap

- Monomial coefficients unintuitive
- Lagrange interpolation is better, but there is a global effect
- The possible curves are the same: just cubics

43

## Splines

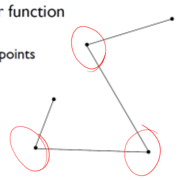
- Spline interpolate data
- Lower degree curves are used (fewer points can be used)



44

## Spline segments

- Piecewise linear
  - Each interval will be a linear function
    - $x(t) = at + b$
    - constraints are values at endpoints
    - $b = x_0$ ; $a = x_1 - x_0$
    - this is linear interpolation



45

## Spline segments

$$\begin{bmatrix} t & 1 \end{bmatrix} \begin{bmatrix} -p_0 + p_1 \\ p_0 \end{bmatrix} = t(p_1 - p_0) + p_0$$

- Vector formulation

$$x(t) = (x_1 - x_0)t + x_0$$
$$y(t) = (y_1 - y_0)t + y_0$$
$$\mathbf{f}(t) = (\mathbf{p}_1 - \mathbf{p}_0)t + \mathbf{p}_0$$

- Matrix formulation

$$\begin{bmatrix} -p_0 + p_1 \\ p_0 + 0p_1 \end{bmatrix}$$

$$\mathbf{f}(t) = \begin{bmatrix} t & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \end{bmatrix}$$

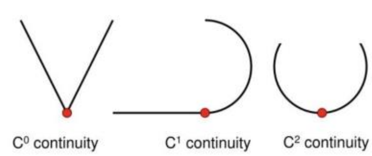
46

## Smoothness

- $C^0$-continuity: the segments meet at the nodes
- $C^1$-continuity: curve must be continuously derivable after the parameter one time.
  - The speed must be the same in the joint of both segments (the slopes at the joint must the equal)
- $C^2$-continuity: curve must be continuously derivable after the parameter twice.
  - The acceleration must be the same in the joint of both segments

47

## Smoothness




48

## Polynomials form a vector space

- You can add and subtract them
- You can multiply them by real numbers
- These operations follow the usual rules
  - Associativity, commutativity, inverses for addition
  - Distributivity for scalar multiplication
  - Etc.
- Anything you can do with vectors in general, you can do with polynomials.

49

## We work with vector spaces using bases

- Basis – list of vectors $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d)$
  - Linearly independent:

$$\sum_i c_i \mathbf{b}_i = 0 \quad \Longrightarrow \quad c_i = 0 \text{ for all } i$$

  - Span the space: for all vectors $\mathbf{v}$ there exist $c_i$ such that

$$\sum_i c_i \mathbf{b}_i = \mathbf{v}$$

- A basis lets us identify a vector by the unique linear combination

$$(c_0, c_1, \ldots, c_d)$$

50

## Review: Vector Space

- A set $V$ of vectors is called a vector space if:
  - ❑ $V$ contains the zero vector
  - ❑ For every vector $v$, if $V$ contains $v$ the it contains $\alpha v$ for every scalar $\alpha$
  - ❑ For every pair $u$ and $v$ of vectors, if $V$ contains $u$ and $v$ then it contains $u + v$

51

51

## Review: Vector Space

- Examples:
  - ❑ A line passing through the origin
  - ❑ A plane that contains the origin
  - ❑ The solution of a homogeneous liner system!
  - ❑ $\mathbb{R}^2$ and $\mathbb{R}^3$

52

52

## Review: Subspace

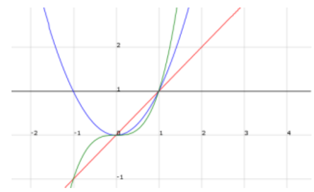
- If $V$ and $W$ are are vector spaces and $V$ is a subset of $W$, we say $V$ is a subspace of $W$
- Examples:
  - ❑ That line passing through the origin is a subspace of that plane containing the origin.
  - ❑ $V = \{a_0 + a_1 x + \ldots + a_n x^n \mid a_i \in \mathbb{R}\}$, i.e., polynomials of degree $\leq n$

53

53

## Monomial basis for cubics:

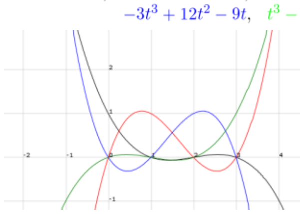
$$(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4) = (1, t, t^2, t^3)$$



54

## Interpolation basis for $t = (0, 1, 2, 3)$

$$\tfrac{1}{6}(-t^3 + 6t^2 - 11t + 6, \quad 3t^3 - 15t^3 + 18t, \quad -3t^3 + 12t^2 - 9t, \quad t^3 - 3t^2 + 2t)$$



55

## Change of basis

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}$$

56

# Change of basis

- Monomial to interpolation for $t = (0, 1, 2, 3)$

$$\begin{pmatrix} 1 & t & t^2 & t^3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} L_1 & L_2 & L_3 & L_4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

- $\begin{pmatrix} L_1 & L_2 & L_3 & L_4 \end{pmatrix}$ is the interpolation basis
- Columns are monomial basis elements in terms of interpolation basis elements
- Rows are interpolation coefficients in terms of monomial coefficients

57

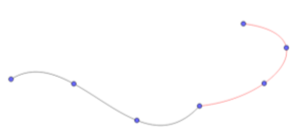
# Remark

- Basis change is just a matrix multiplication
- We need to find a good basis for intuitively designing curves

58

# How do we get smoothness?

- With Lagrange polynomials, it's hard to get tangents to match up

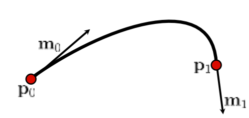


59

# Hermite Basis

- Instead of four points, specify two points and two derivatives:

$$\mathbf{p}(t_0) = \mathbf{p}_0$$
$$\mathbf{p}(t_1) = \mathbf{p}_1$$
$$\mathbf{p}'(t_0) = \mathbf{m}_0$$
$$\mathbf{p}'(t_1) = \mathbf{m}_1$$

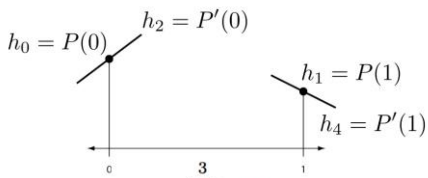



$$\mathbf{p}(t) = \mathbf{p}_0 \underline{h_{00}(t)} + \mathbf{p}_1 \underline{h_{10}(t)} + \mathbf{m}_0 \underline{h_{01}(t)} + \mathbf{m}_1 \underline{h_{11}(t)}$$

Cubic polynomials of the form $H(t) = h_0 + h_1 t + h_2 t^2 + h_3 t^3$

a b c d

60




$$P(t) = \sum_{i=0}^{3} h_i \, H_i(t)$$

**Given: values and derivatives at 2 points**

61

## Hermite Basis

**Assume cubic polynomial**

$$P(t) = a\,t^3 + b\,t^2 + c\,t + d$$
$$P'(t) = 3a\,t^2 + 2b\,t + c$$

**Solve for coefficients:**

$$P(0) = h_0 = d$$
$$P(1) = h_1 = a + b + c + d$$
$$P'(0) = h_2 = c$$
$$P'(1) = h_3 = 3a + 2b + c$$

CS148 Lecture 8 Pat Hanrahan, Fall 2009

62

## Hermite Basis

$x = A^{-1}b$

$$h_0 = d$$
$$h_1 = a + b + c + d$$
$$h_2 = c$$
$$h_3 = 3a + 2b + c$$

$$\underbrace{\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_{x}$$

CS148 Lecture 8 Pat Hanrahan, Fall 2009

63

## Hermite Basis

- $\mathbf{X(t) = t^T M_H q}$
  - $\mathbf{q}$ is the control vector

$$\mathbf{M_H} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

$A^{-1}$

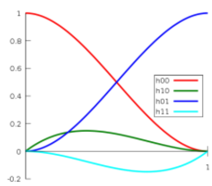
$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_0' \\ x_1 \\ x_1' \end{bmatrix}$$

64

## Hermite Basis

- Instead of four points, specify two points and two derivatives:

$$\begin{pmatrix} 1 - 3t^2 + 2t^3 \\ 3t^2 - 2t^3 \\ t - 2t^2 + t^3 \\ -t^2 + t^3 \end{pmatrix}^T$$




65

## Smooth Curves

- General parametric form
  - Weighted sum of coefficients and basis functions

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i F_i^n(t)$$

Coefficients $\mathbf{c}_i \in \mathbb{R}^k$

Basis functions $F_i^n(t) \in \Pi^n$

66

## What might be "good" basis functions?

- Intuitive editing
  - Control points are coefficients
  - Predictable behavior
  - No oscillation
  - Local control

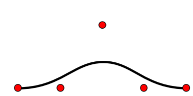
$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i F_i^n(t)$$

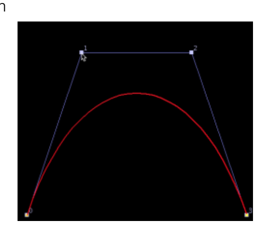
- Mathematical guarantees
  - Smoothness, affine invariance, linear precision, ...
- Efficient processing and rendering

67

## What might be "good" basis functions?

- Approximation instead of interpolation
- Bézier- and B-Spline curves





68

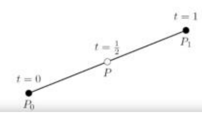
## Parametric equations - review

- A parameter, $t$, is used to determine the value of the variables, e.g.,

$$x(t) = (1-t)x_0 + tx_1,$$
$$y(t) = (1-t)y_0 + ty_1,$$

where $0 \le t \le 1$. Let $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$ and $P = (x, y)$ then

$$P(t) = (1-t)P_0 + tP_1.$$

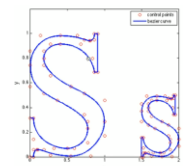



Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

69

## Bézier curves

- Bézier curve is a parametric curve used to draw smooth lines
- Pierre Bézier used them for designing cars at Renault
- Common applications: CAD software, 3D modelling and typefaces

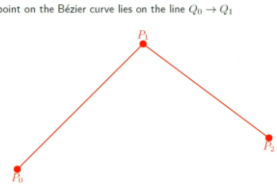


Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

70

## Bézier curves

- $Q_0$ and $Q_1$ lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
- The point on the Bézier curve lies on the line $Q_0 \to Q_1$

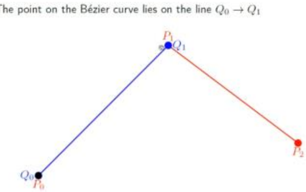



Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

71

## Bézier curves

- $Q_0$ and $Q_1$ lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
- The point on the Bézier curve lies on the line $Q_0 \to Q_1$

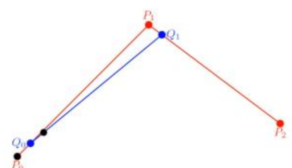



Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

72

## Bézier curves

- $Q_0$ and $Q_1$ lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
- The point on the Bézier curve lies on the line $Q_0 \to Q_1$

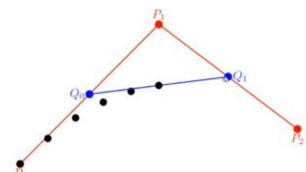



Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

73

## Bézier curves

- $Q_0$ and $Q_1$ lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
- The point on the Bézier curve lies on the line $Q_0 \to Q_1$

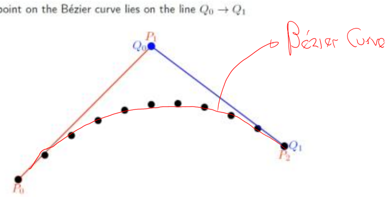



Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

74

## Bézier curves

- $Q_0$ and $Q_1$ lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
- The point on the Bézier curve lies on the line $Q_0 \to Q_1$




Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

75

## Bézier curves

| $t$ | $Q_0$ | $Q_1$ | $C(t)$ |
|---|---|---|---|
| 0 | $P_0$ | $P_1$ | $Q_0$ |
| 1 | $P_1$ | $P_2$ | $Q_1$ |

- $Q_0$ and $Q_1$ are points on the lines $P_0 \to P_1$ and $P_1 \to P_2$

$$Q_0 = (1-t)P_0 + tP_1,$$
$$Q_1 = (1-t)P_1 + tP_2.$$

- $C(t)$ is a point on the Bézier curve on the line $Q_0 \to Q_1$

$$C(t) = (1-t)Q_0 + tQ_1.$$

$$C(t) = (1-t)\left[(1-t)P_0 + tP_1\right] + t\left[(1-t)P_1 + tP_2\right]$$

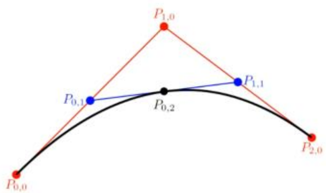
- Combining gives

$$\boxed{C(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2.}$$

Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

76

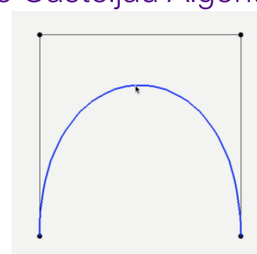
## De Casteljau's algotrithm




Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

77

## De Casteljau Algorithm



i33www.ira.uka.de/applets/mocca/html/noplugin/BezierCurve/AppDeCasteljau/index.html

78

## De Casteljau's algotrithm

$P_{0,2} = (1-t)P_{0,1} + tP_{1,1}$ $\quad$ $P_{0,1} = (1-t)P_{0,0} + tP_{1,0}$

$P_{1,1} = (1-t)P_{1,0} + tP_{2,0}$

- The derivation process use here is known as **de Casteljau's algorithm**.

$P_{0,2} = (1-t^2)P_{0,0} + t(1-t)P_{1,0} + t(1-t)P_{1,0} + t^2 P_{2,0}$

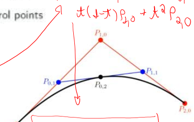
- Let $P_{i,j}$ denote the control points where $P_{i,0}$ are the original control points $P_0$ to $P_2$, $P_{i,1}$ are the points $Q_0$ to $Q_1$ and $P_{0,2}$ is $C(t)$ then

$$P_{i,j} = (1-t)P_{i,j-1} + tP_{i+1,j-1}.$$

- $P_{i,j}$ depends on the points $P_{i,j-1}$ and $P_{i+1,j-1}$, i.e.,

$$\begin{array}{ccc} P_{i,j-1} & \rightarrow & P_{i,j} \\ & \nearrow & \\ P_{i+1,j-1} & & \end{array}$$

where the horizontal arrow denotes the $(1-t)$ coefficient and the diagonal arrow denotes the $t$ coefficient




$$\boxed{\begin{aligned} P_{0,2} = {} & (1-t^2)P_{0,0} + \\ & 2t(1-t)P_{1,0} + \\ & t^2 P_{2,0} \end{aligned}}$$

$P_{0,2} = (1-t)[(1-t)P_{0,0} + tP_{1,0}] + t[(1-t)P_{1,0} + tP_{2,0}]$

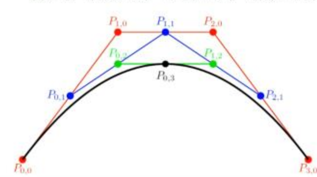
Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

79

## Cubic Bézier curve

- A cubic Bézier curve is defined by 4 control points: $P_{0,0}$, $P_{1,0}$, $P_{2,0}$ and $P_{3,0}$

$$P_{0,3} = (1-t)^3 P_{0,0} + 3t(1-t)^2 P_{1,0} + 3t^2(1-t)P_{2,0} + t^3 P_{3,0}.$$




Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

80

## Bernstein Polynomials

- Bernstein polynomials $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$
- Binomial coefficients $\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{if } 0 \le i \le n \\ 0 & \text{otherwise} \end{cases}$

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i B_i^n(t)$$

- linear: $\mathbf{p}(t) = \mathbf{c}_0(1-t) + \mathbf{c}_1 t$
- quadratic: $\mathbf{p}(t) = \mathbf{c}_0(1-t)^2 + \mathbf{c}_1 2t(1-t) + \mathbf{c}_2 t^2$
- cubic: $\mathbf{p}(t) = \mathbf{c}_0(1-t)^3 + \mathbf{c}_1 3t(1-t)^2 + \mathbf{c}_2 3t^2(1-t) + \mathbf{c}_3 t^3$

81

## Cubic Bernstein Polynomials

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i},$$

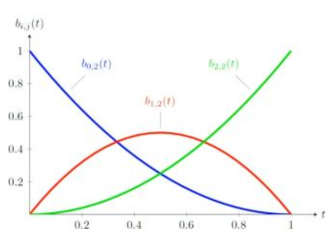
- For example, the Bernstein polynomials for a cubic Bézier curve are

$$b_{0,3}(t) = \binom{3}{0} t^0 (1-t)^{3-0} = (1-t)^3,$$
$$b_{1,3}(t) = \binom{3}{1} t^1 (1-t)^{3-1} = 3t(1-t)^2,$$
$$b_{2,3}(t) = \binom{3}{2} t^2 (1-t)^{3-2} = 3t^2(1-t),$$
$$b_{3,3}(t) = \binom{3}{3} t^3 (1-t)^{3-3} = t^3.$$

Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

82

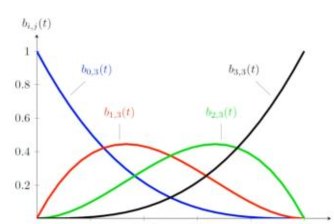
## Bernstein polynomials - Quadratic




Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

83

## Bernstein polynomials - Cubic




Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

84

## Matrix form of a Bézier curve

- In order to save computational effort, Bézier curves are precalculated and expressed in matrix form as follows:

$$C(t) = \begin{pmatrix} P_0 & P_1 & \cdots & P_{n-1} & P_n \end{pmatrix} M \begin{pmatrix} t^n \\ t^{n-1} \\ \vdots \\ t \\ 1 \end{pmatrix},$$

where $M$ is an $(n+1) \times (n+1)$ matrix.

Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

85

## Matrix form of a Bézier curve

$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}$ $\quad$ $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}$

- Consider the quadratic Bézier curve with the brackets expanded out

$$C(t) = (t^2 - 2t + 1)P_0 + (-2t^2 + 2t)P_1 + t^2 P_2,$$

$\begin{bmatrix} -2 & 2 & 0 \end{bmatrix} \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}$

this can be expressed in matrix form as

$$C(t) = \begin{pmatrix} P_0 & P_1 & P_2 \end{pmatrix} \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^2 \\ t \\ 1 \end{pmatrix}.$$

Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

86

## Matrix form of a Bézier curve

- Similarly a cubic Bézier curve can be expressed using

$$C(t) = \begin{pmatrix} P_0 & P_1 & P_2 & P_3 \end{pmatrix} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}.$$

Dr Jon Shiach - https://youtu.be/2HvH9cmHbG4

87

## Disadvantages

- Still global support of basis functions for each curve segment
- Insertion of new control points?
- Continuity conditions restrict control polygon

88