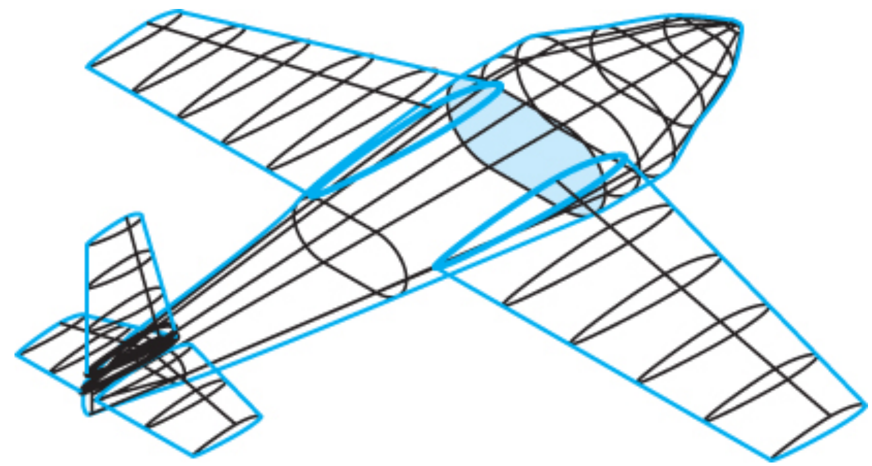
CS130 : Computer Graphics

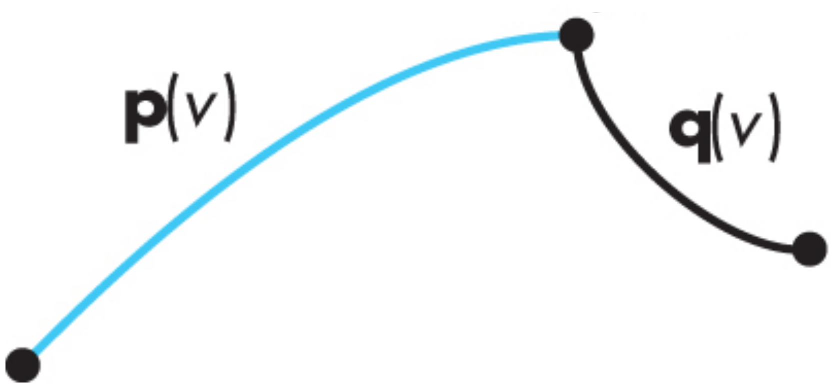
Curves

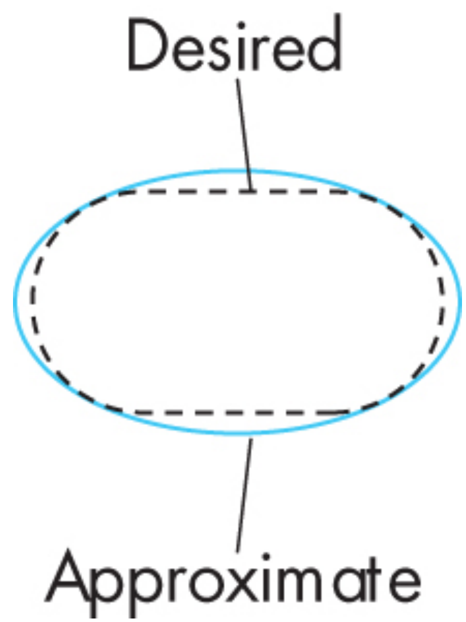
Tamar Shinar
Computer Science & Engineering
UC Riverside

Design considerations

- local control of shape
  - design each segment independently
- smoothness and continuity
- ability to evaluate derivatives
- stability
  - small change in input leads to small change in output
- ease of rendering

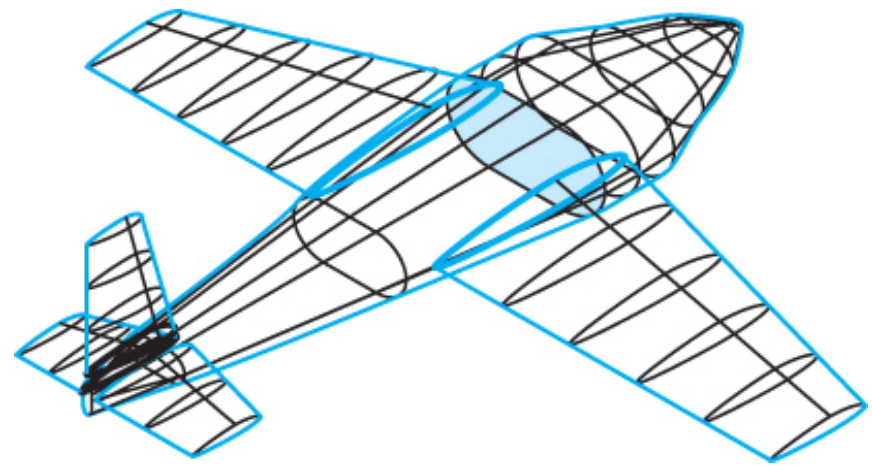


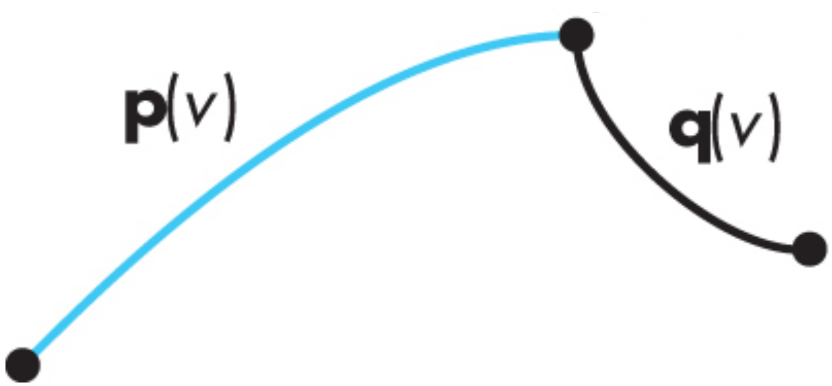


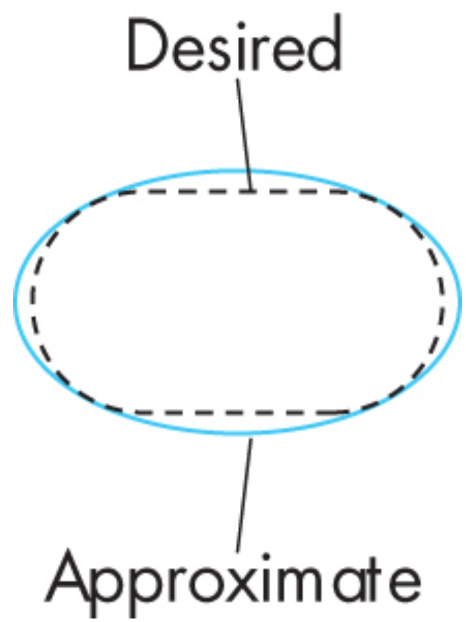


Design considerations

- local control of shape
  - design each segment independently
- smoothness and continuity
- ability to evaluate derivatives
- stability
  - small change in input leads to small change in output
- ease of rendering



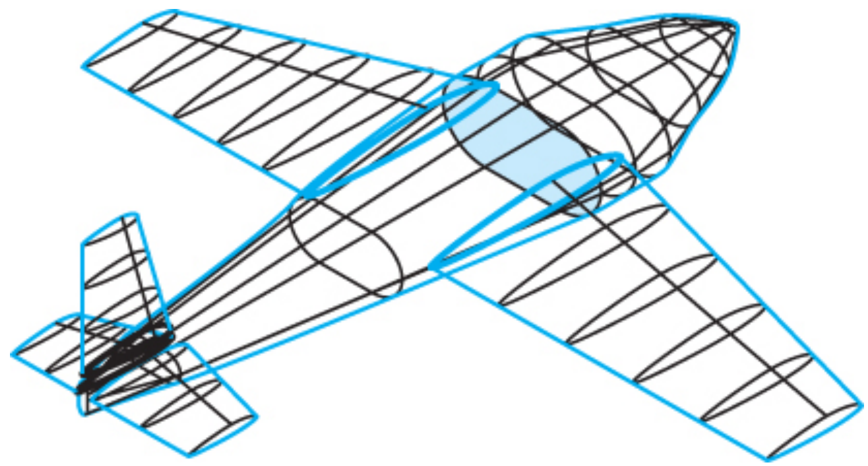


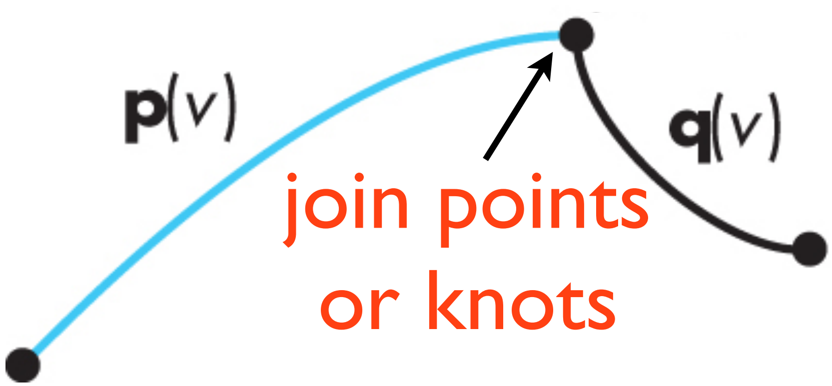


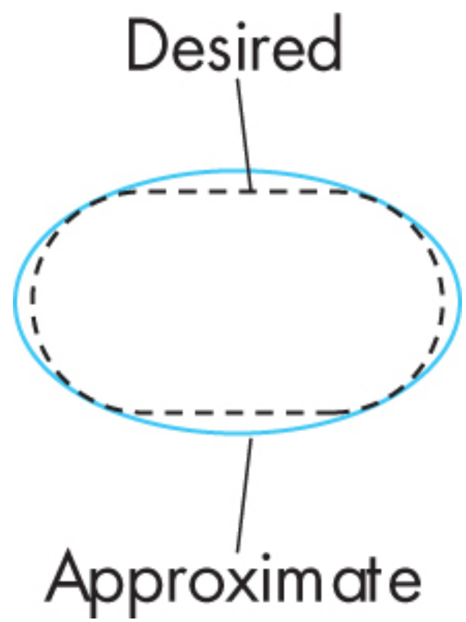
approximate out of a number of wood strips

Design considerations

- local control of shape
  - design each segment independently
- smoothness and continuity
- ability to evaluate derivatives
- stability
  - small change in input leads to small change in output
- ease of rendering





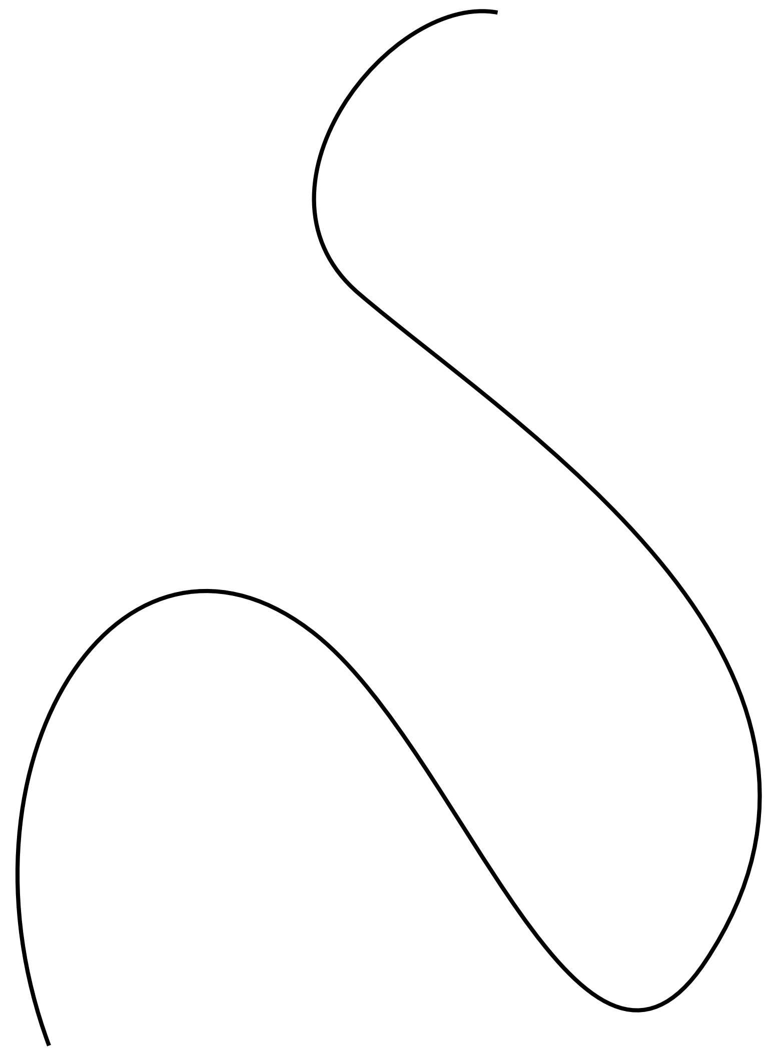


approximate out of a number of wood strips

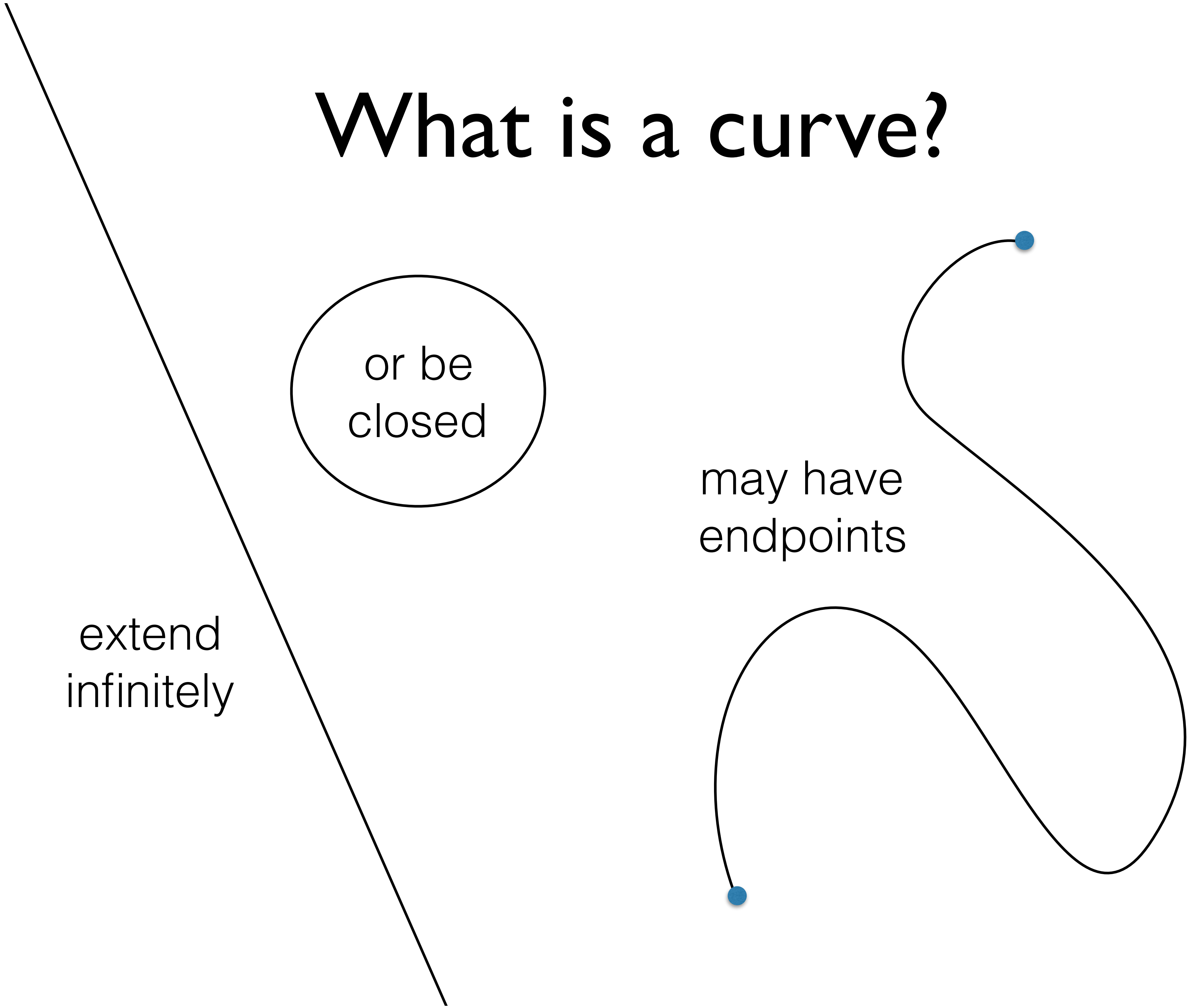
What is a curve?

intuitive idea:
  draw with a pen
  set of points the pen traces

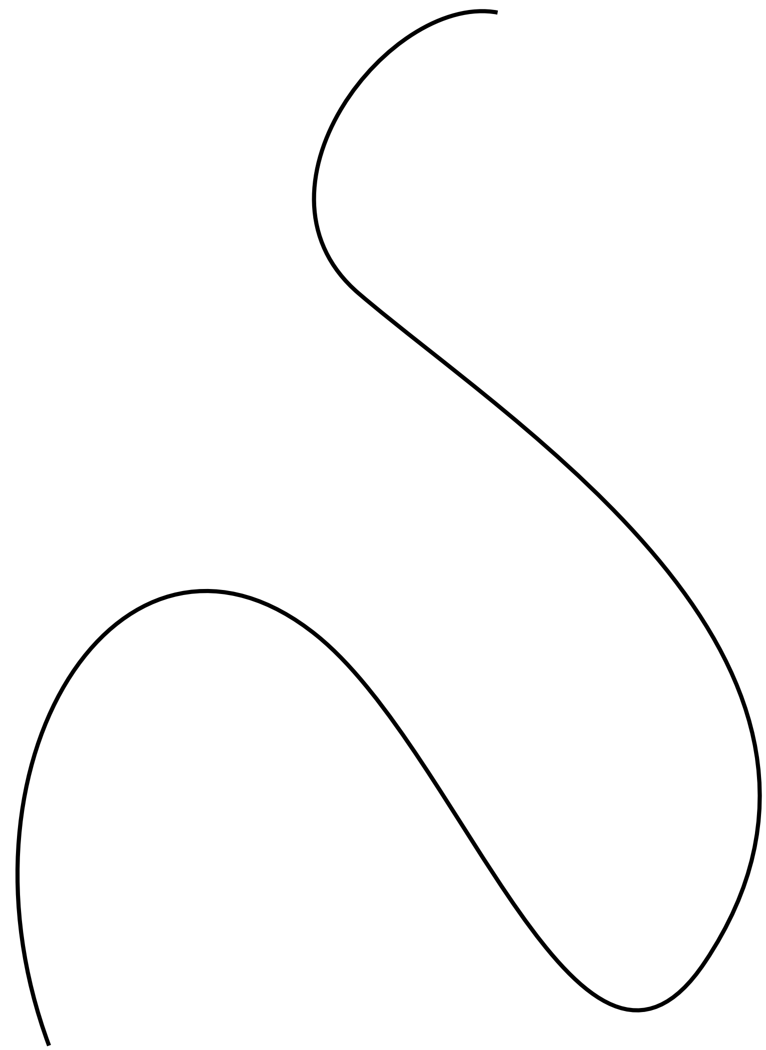
may be 2D, like on paper
or 3D, *space curve*



What is a curve?



How do we specify a curve?

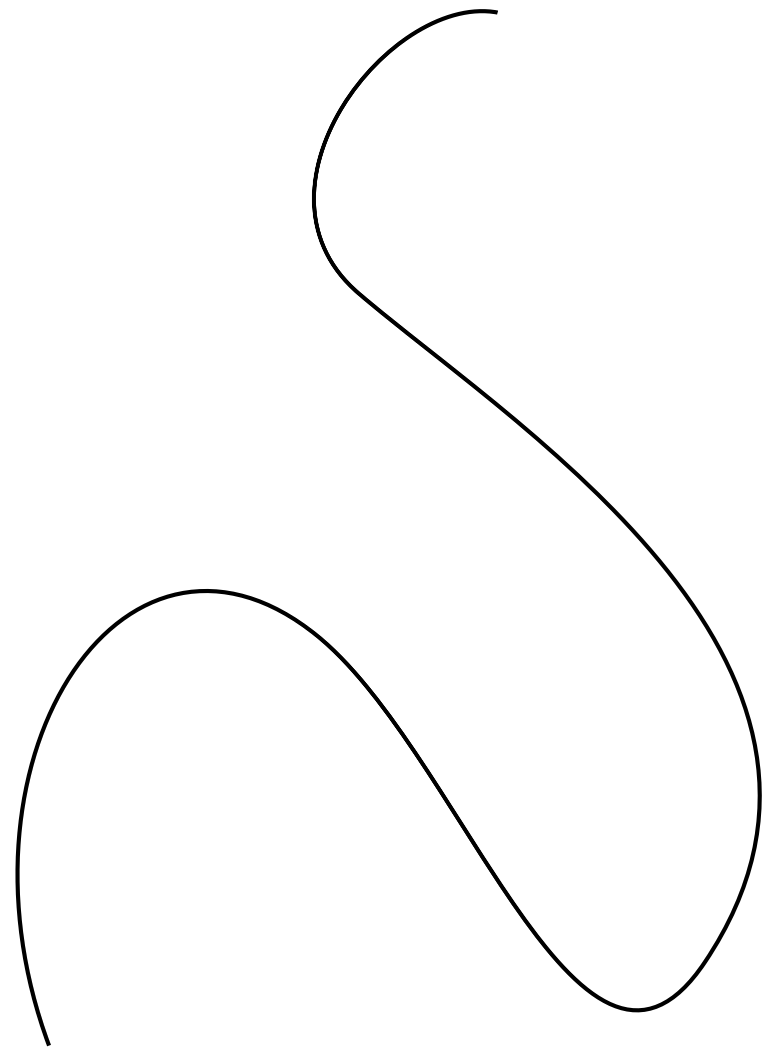


How do we specify a curve?

Implicit

(2D) f(x,y) = 0

test if (x,y) is on the curve

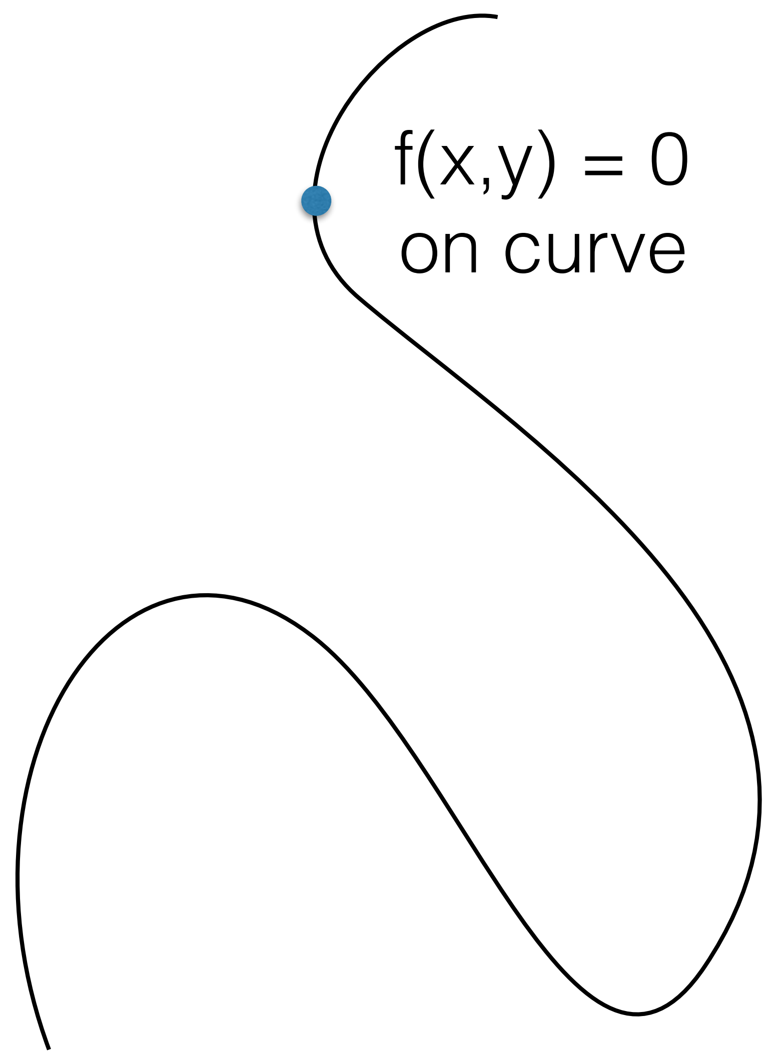


How do we specify a curve?

Implicit

(2D) f(x,y) = 0

test if (x,y) is on the curve

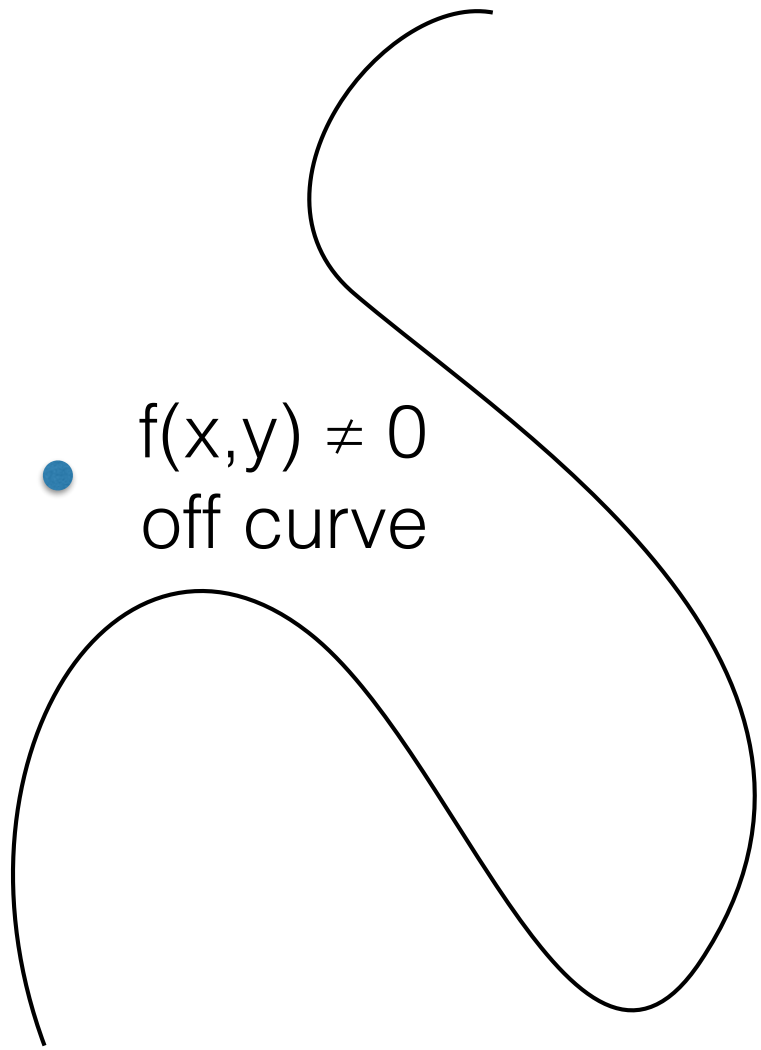


How do we specify a curve?

Implicit

(2D) $f(x,y) = 0$

test if (x,y) is on the curve



How do we specify a curve?

Implicit

(2D) $f(x,y) = 0$

test if (x,y) is on the curve

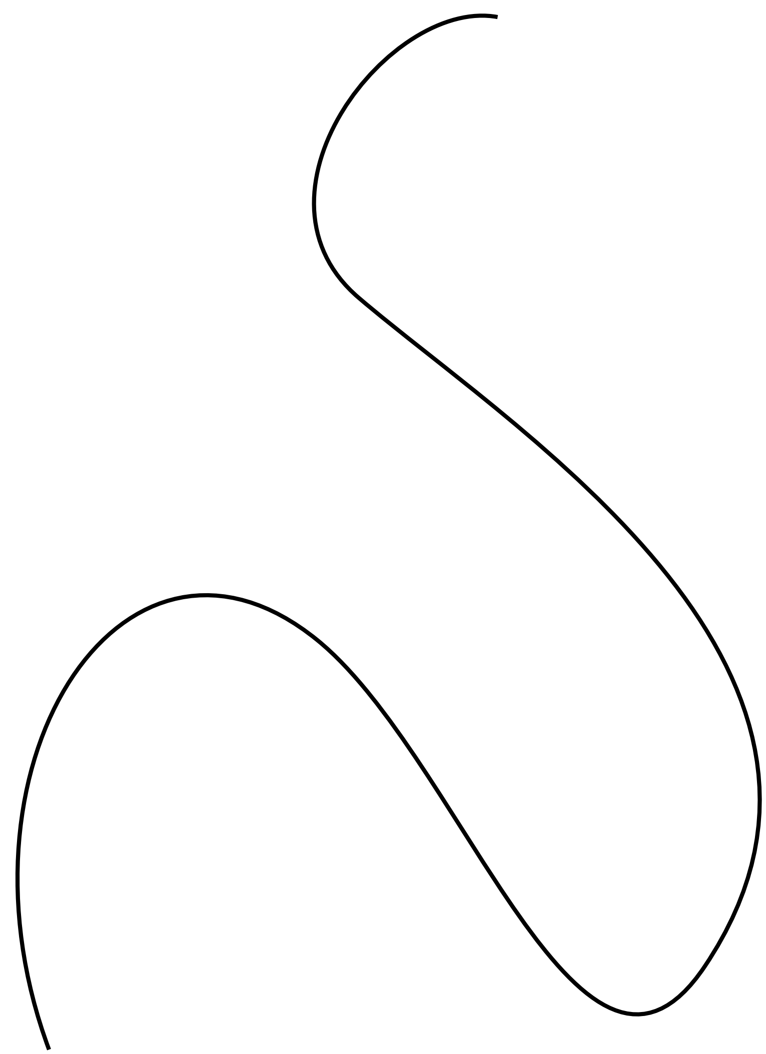
Parametric

(2D) $(x,y) = \mathbf{f}(t)$

(3D) $(x,y,z) = \mathbf{f}(t)$

map free *parameter* t

to points on the curve



How do we specify a curve?

Implicit

(2D) f(x,y) = 0

test if (x,y) is on the curve

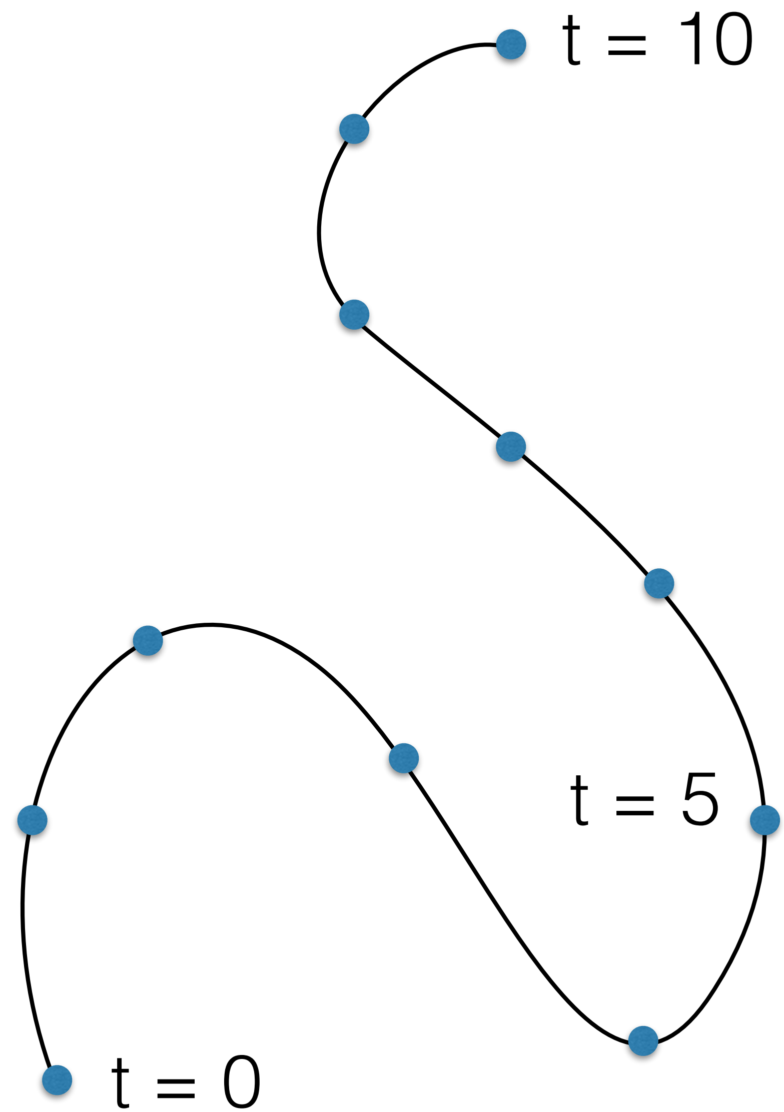
Parametric

(2D) (x,y) = **f**(t)

(3D) (x,y,z) = **f**(t)

map free *parameter* t

to points on the curve



How do we specify a curve?

Implicit

(2D) $f(x,y) = 0$

test if (x,y) is on the curve

Parametric

(2D) $(x,y) = \mathbf{f}(t)$

(3D) $(x,y,z) = \mathbf{f}(t)$

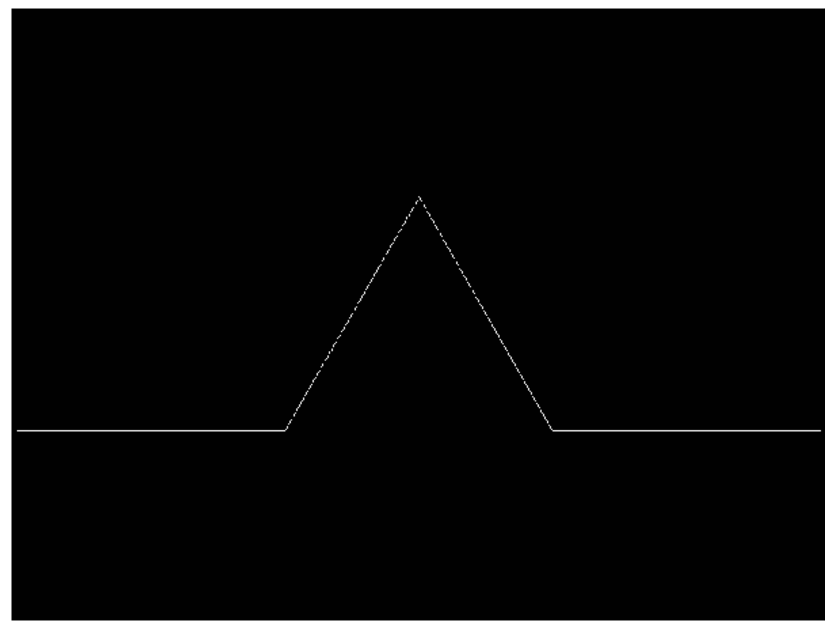
map free *parameter* t

to points on the curve

Procedural

e.g., fractals,

subdivision schemes



[George Reese]

Fractal: Koch Curve

How do we specify a curve?

Implicit

(2D) $f(x,y) = 0$

test if (x,y) is on the curve

Parametric

(2D) $(x,y) = \mathbf{f}(t)$

(3D) $(x,y,z) = \mathbf{f}(t)$

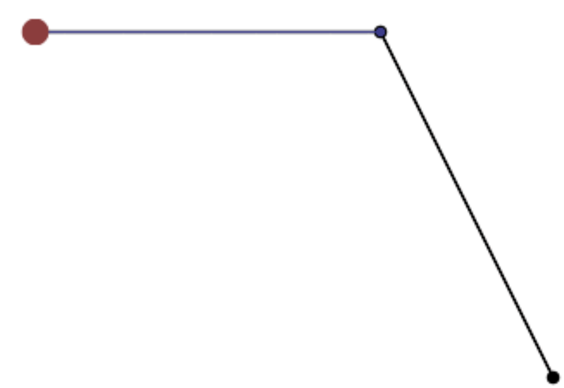
map free *parameter* t

to points on the curve

Procedural

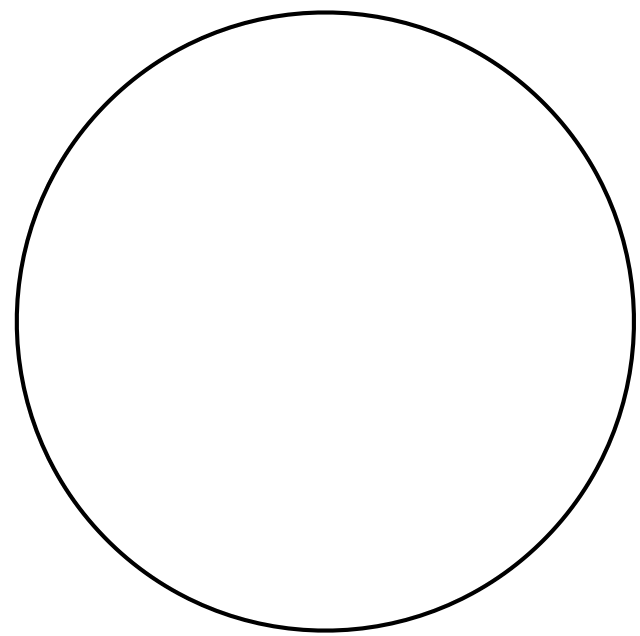
e.g., fractals,

subdivision schemes



Bezier Curve

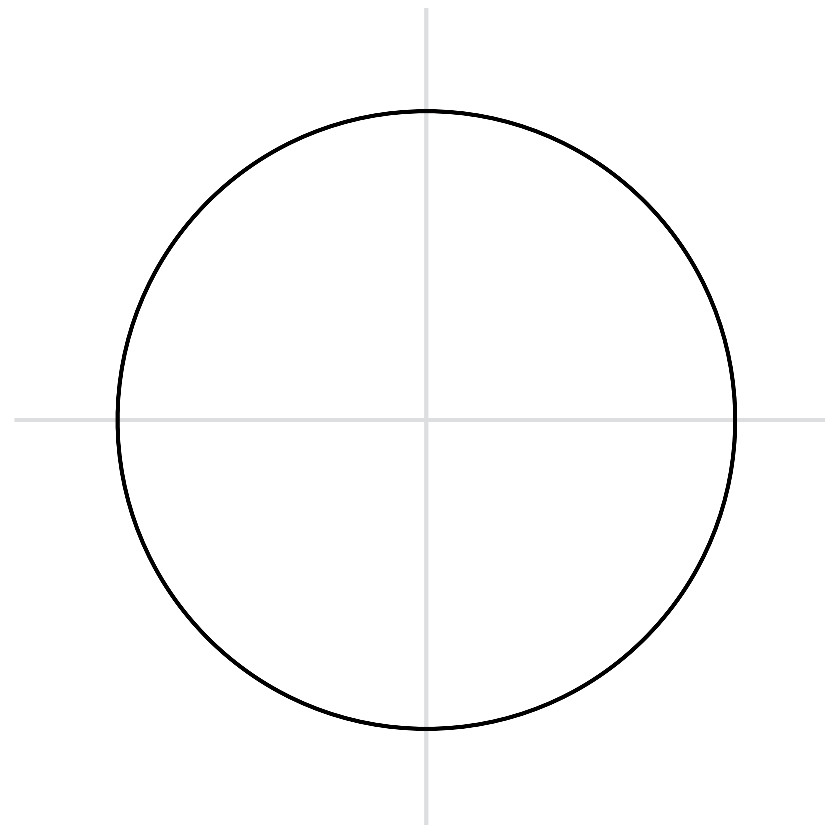
A curve may have multiple representations



A curve may have multiple representations

Implicit

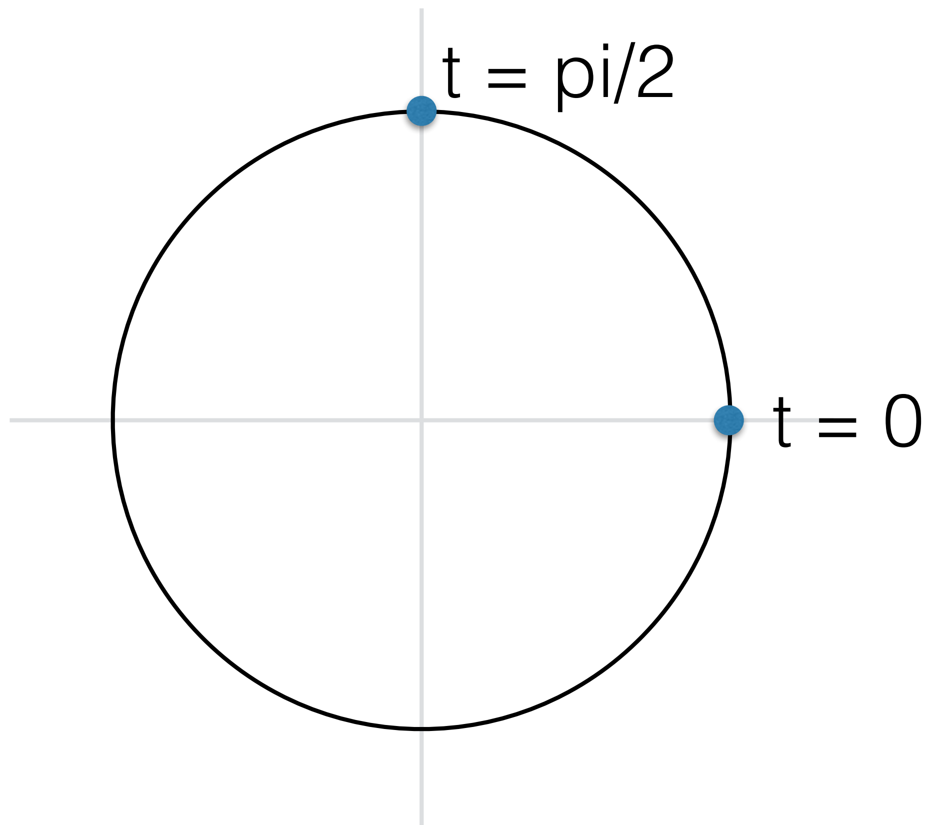
$f(x,y) = x^2 + y^2 - 1 = 0$



A curve may have multiple representations

Parametric

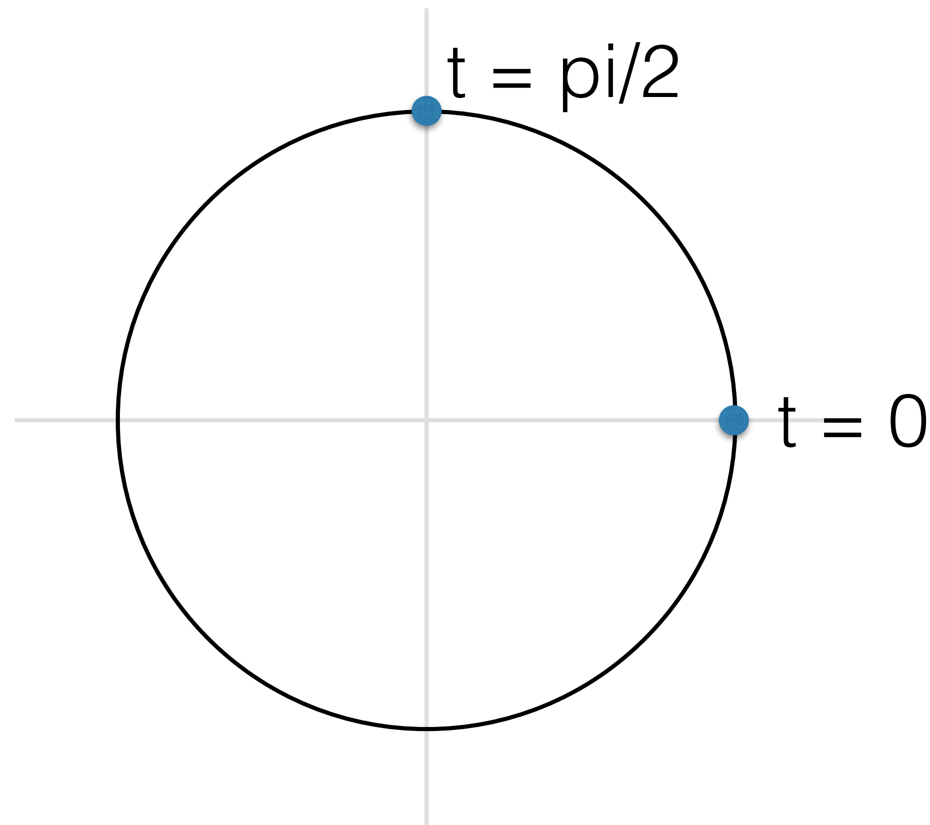
$(x,y) = \mathbf{f}(t) = (\cos t, \sin t)$



A curve may have multiple representations

Parametric

$(x,y) = \mathbf{f}(t) = (\cos t, \sin t),$
$t \text{ in } [0,2\pi)$



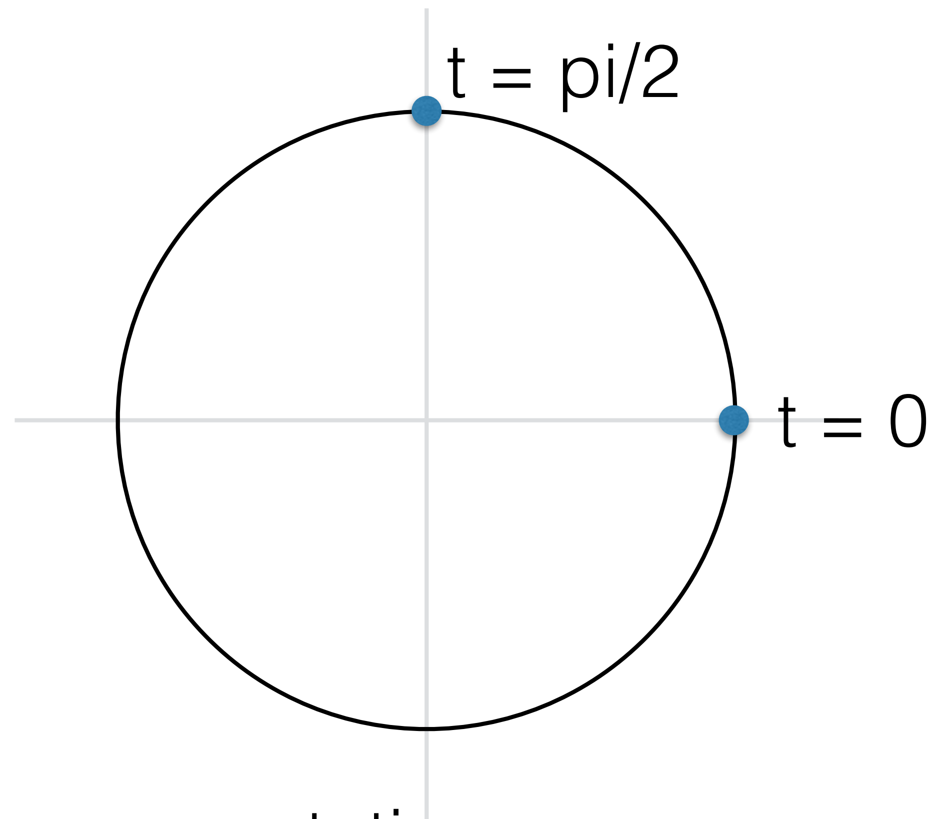
Same curve (set of points),
but different mathematical representation!

A curve may have multiple representations

Parametric

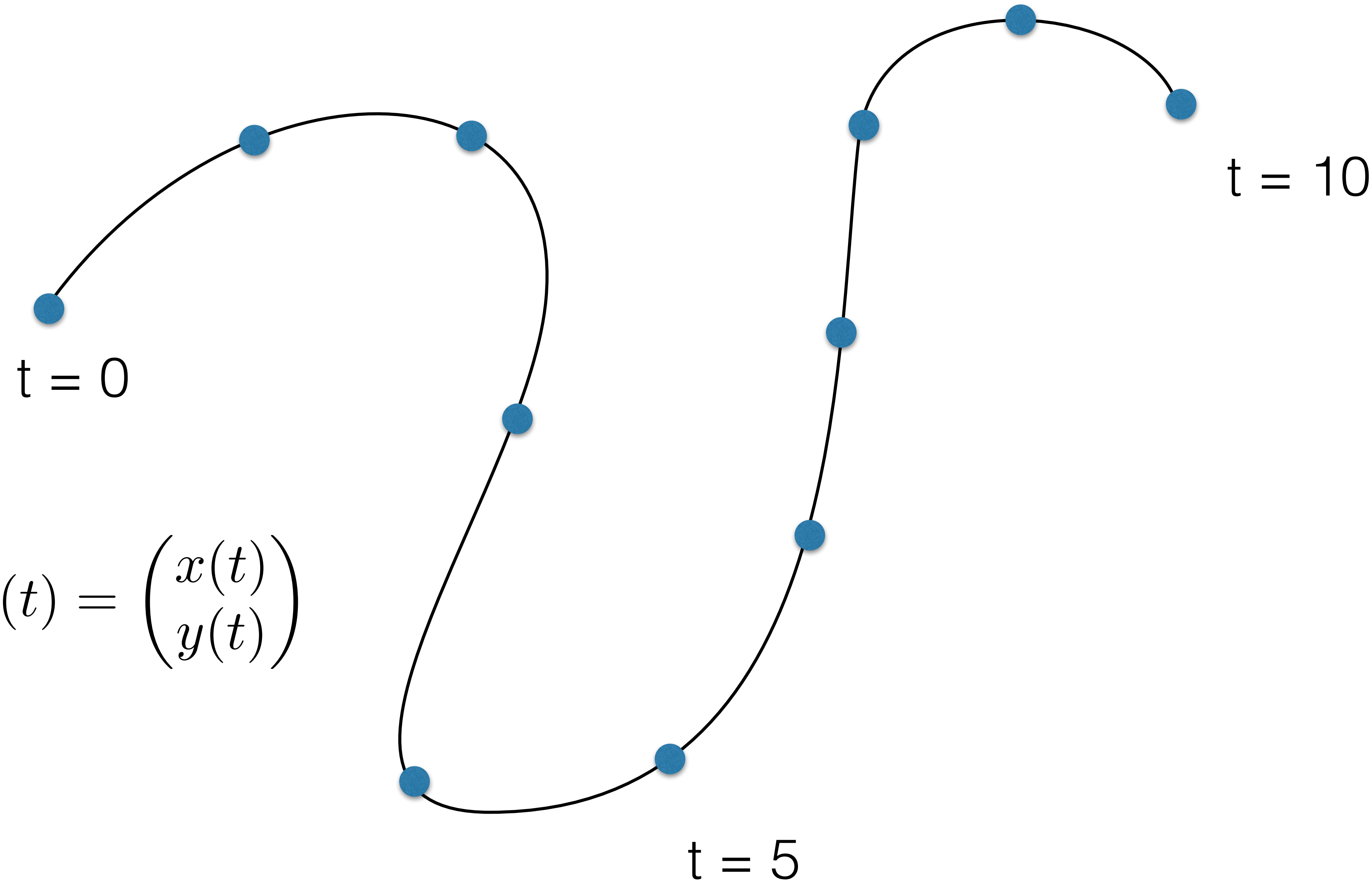
$(x,y) = \mathbf{f}(t) = (\cos t, \sin t),$

$t \in [0, 2\pi)$



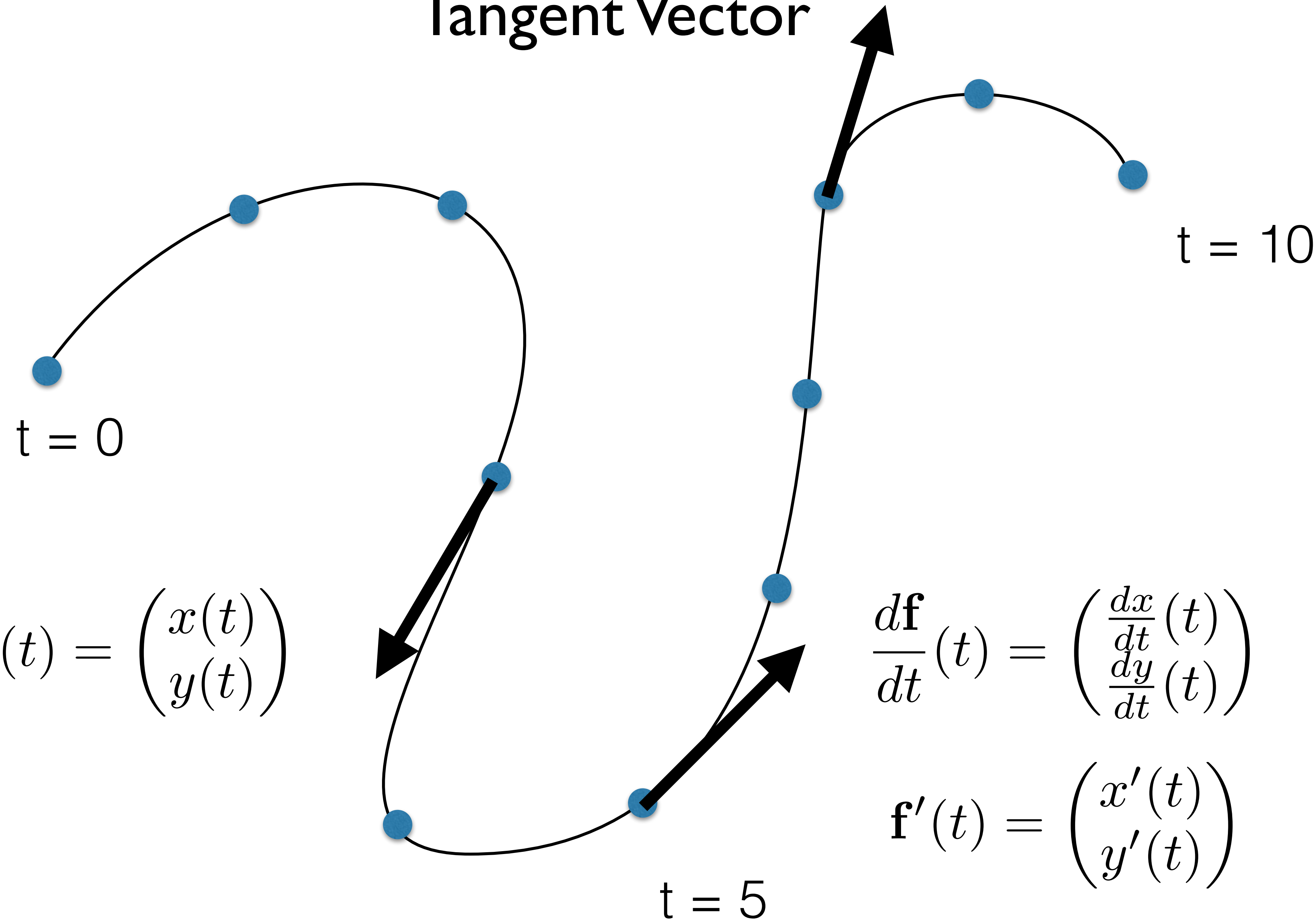
We will focus on parametric representations

Parametric Form



$$\mathbf{f}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

Parametric Form
Tangent Vector

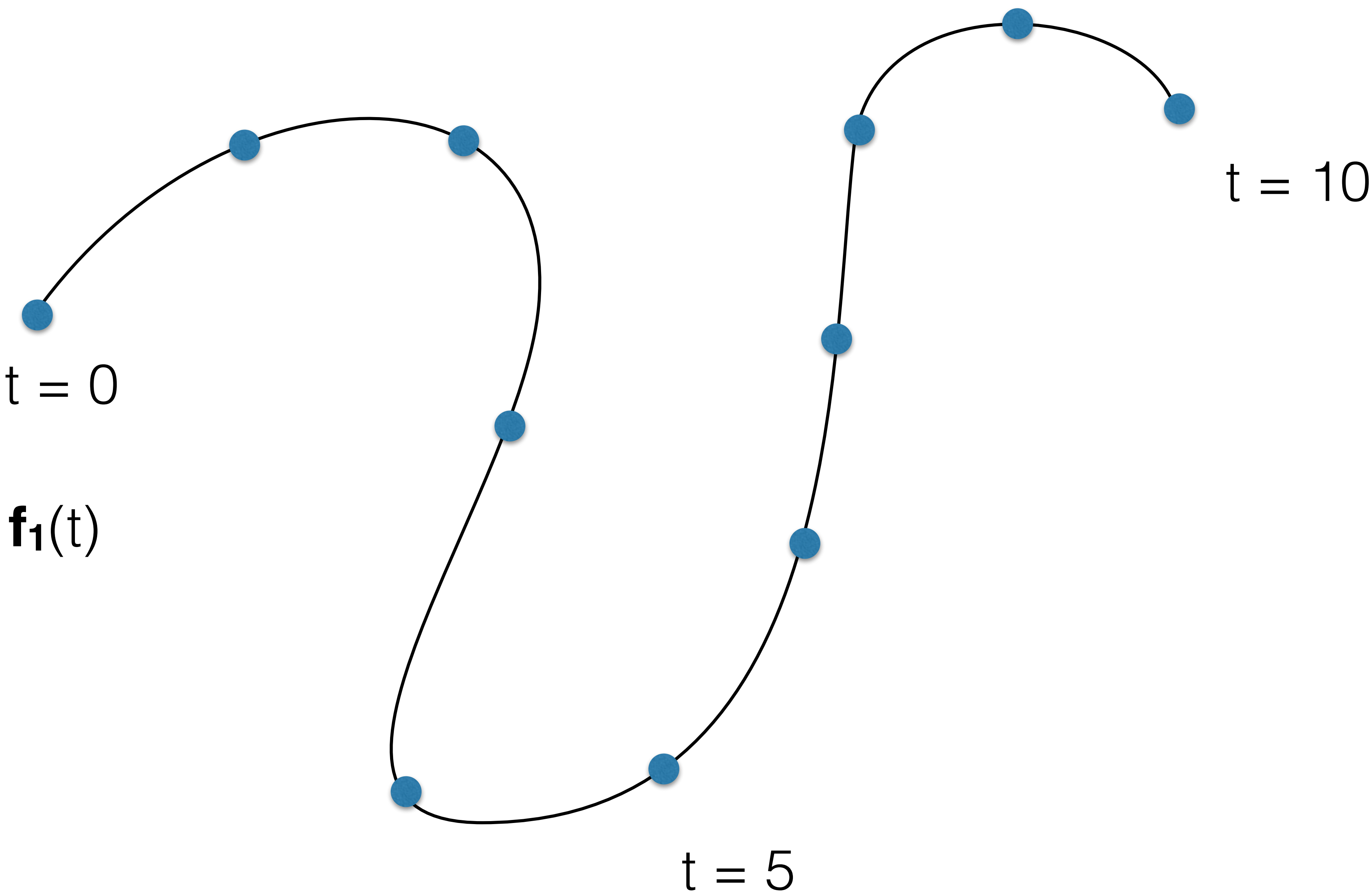


$$\mathbf{f}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

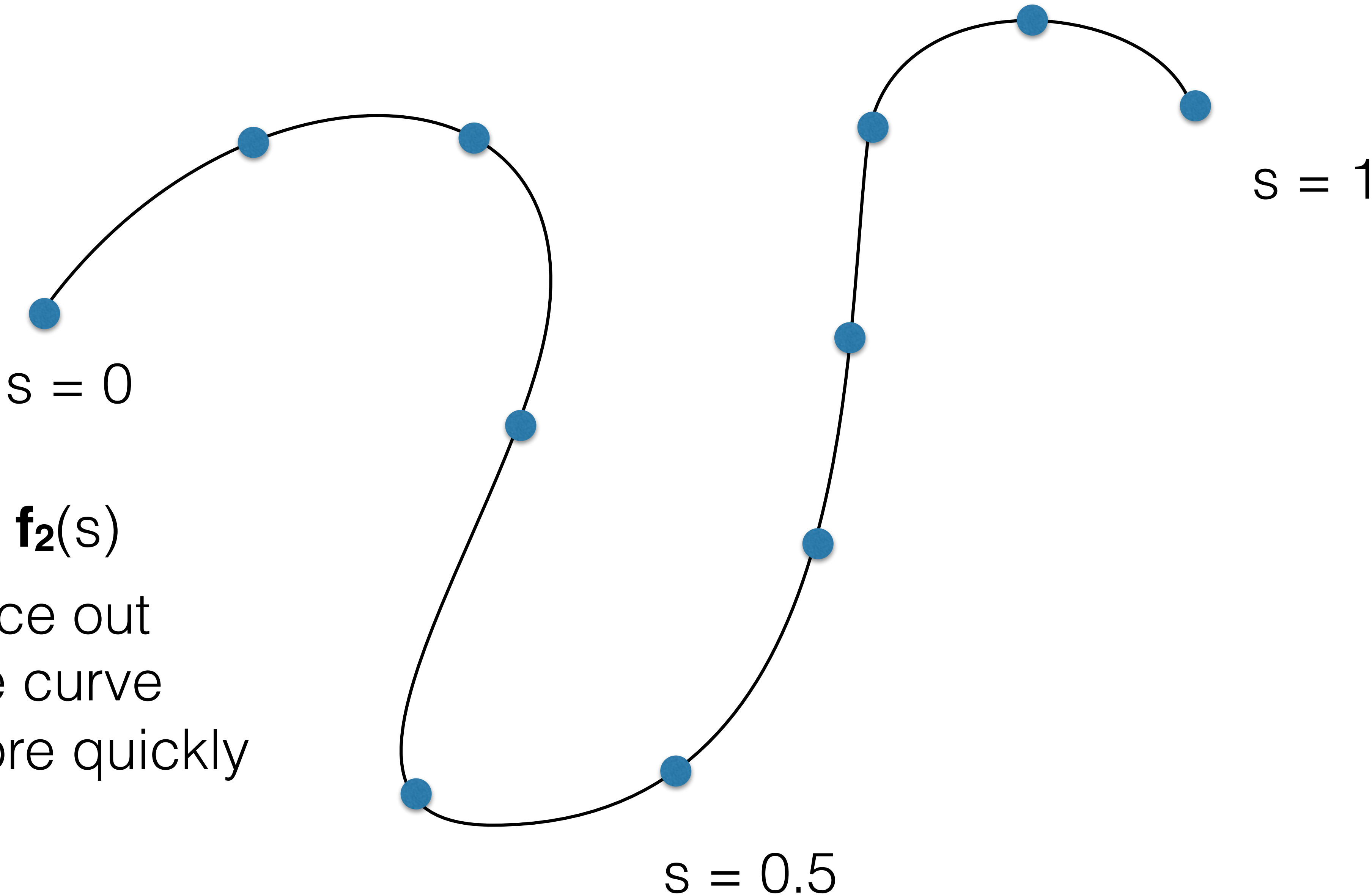
$$\frac{d\mathbf{f}}{dt}(t) = \begin{pmatrix} \frac{dx}{dt}(t) \\ \frac{dy}{dt}(t) \end{pmatrix}$$

$$\mathbf{f}'(t) = \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix}$$

Parameterization, re-parameterization



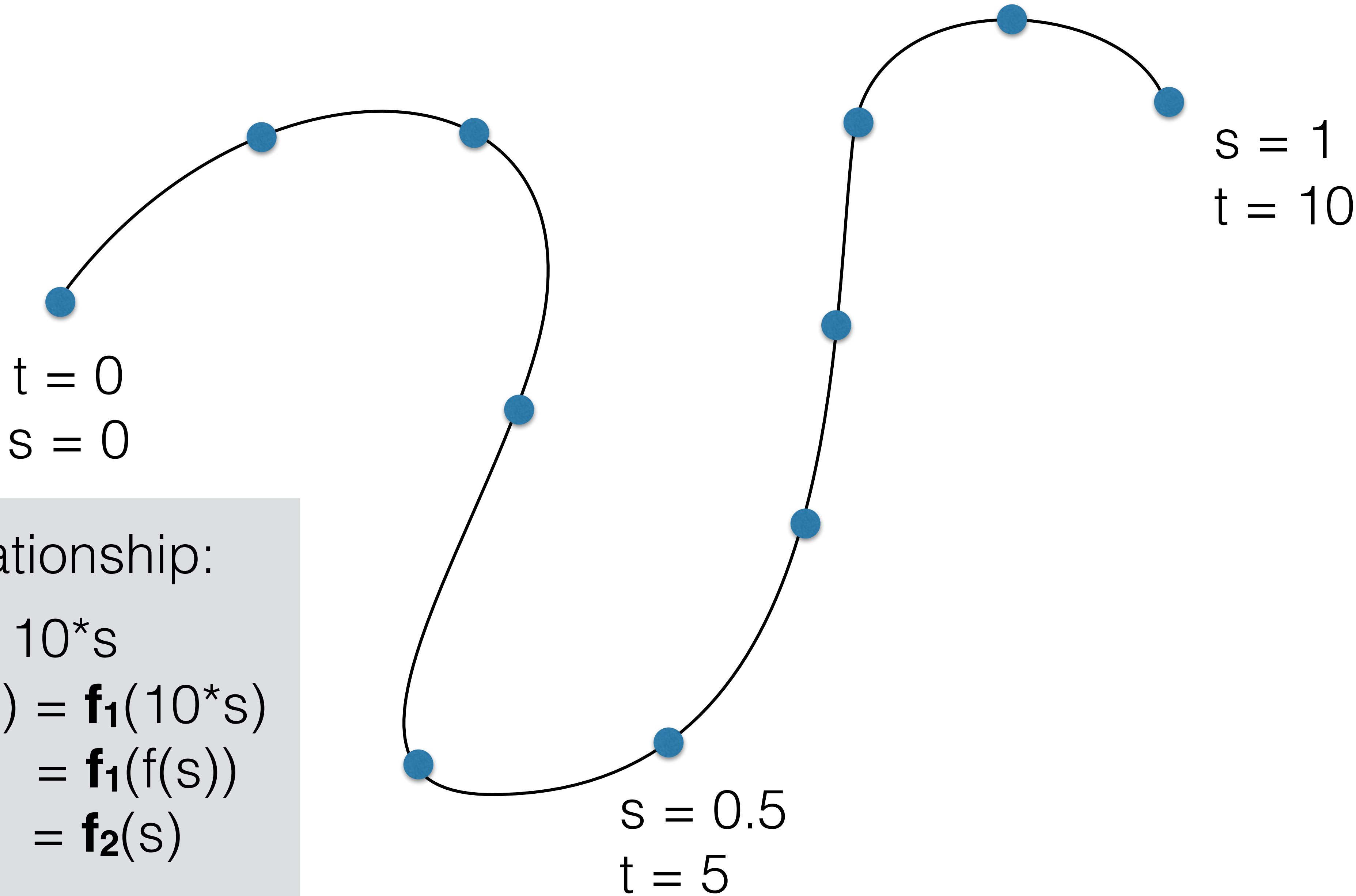
Parameterization, re-parameterization



$\mathbf{f_2}(s)$

trace out the curve more quickly

Parameterization, re-parameterization



relationship:

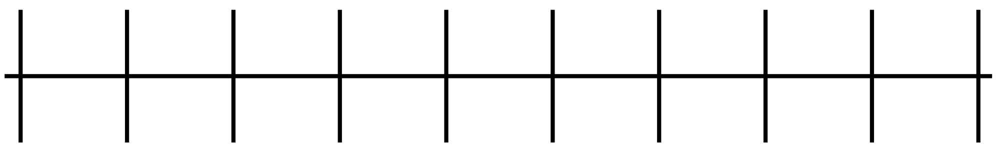
$$t = 10*s$$

$$\mathbf{f_1}(t) = \mathbf{f_1}(10*s)$$

$$= \mathbf{f_1}(f(s))$$

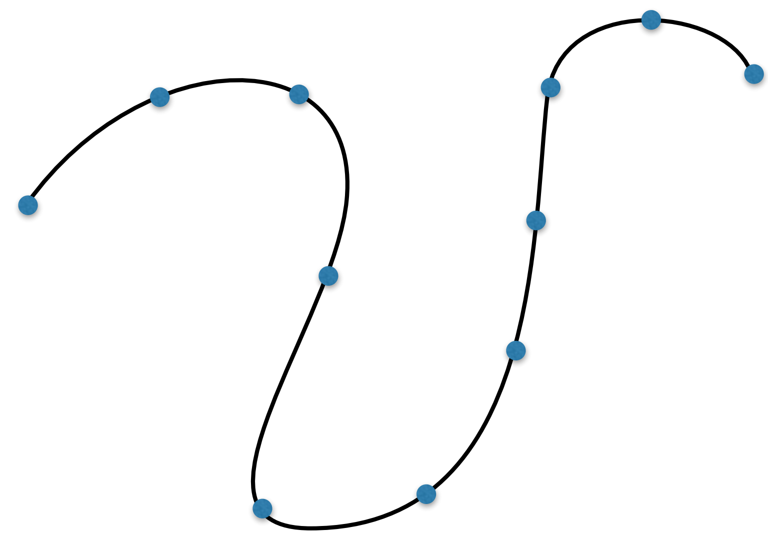
$$= \mathbf{f_2}(s)$$

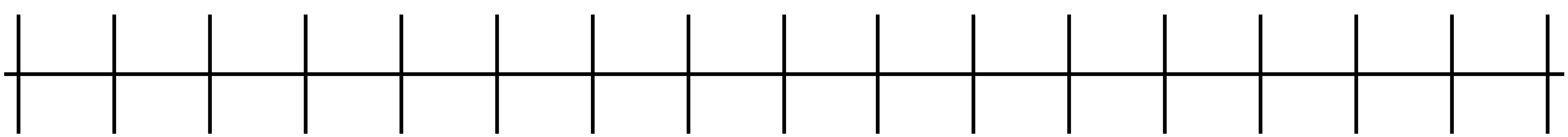
Parameterization, re-parameterization



$t = 0$ $t = 10$

$\mathbf{f_1}(t)$

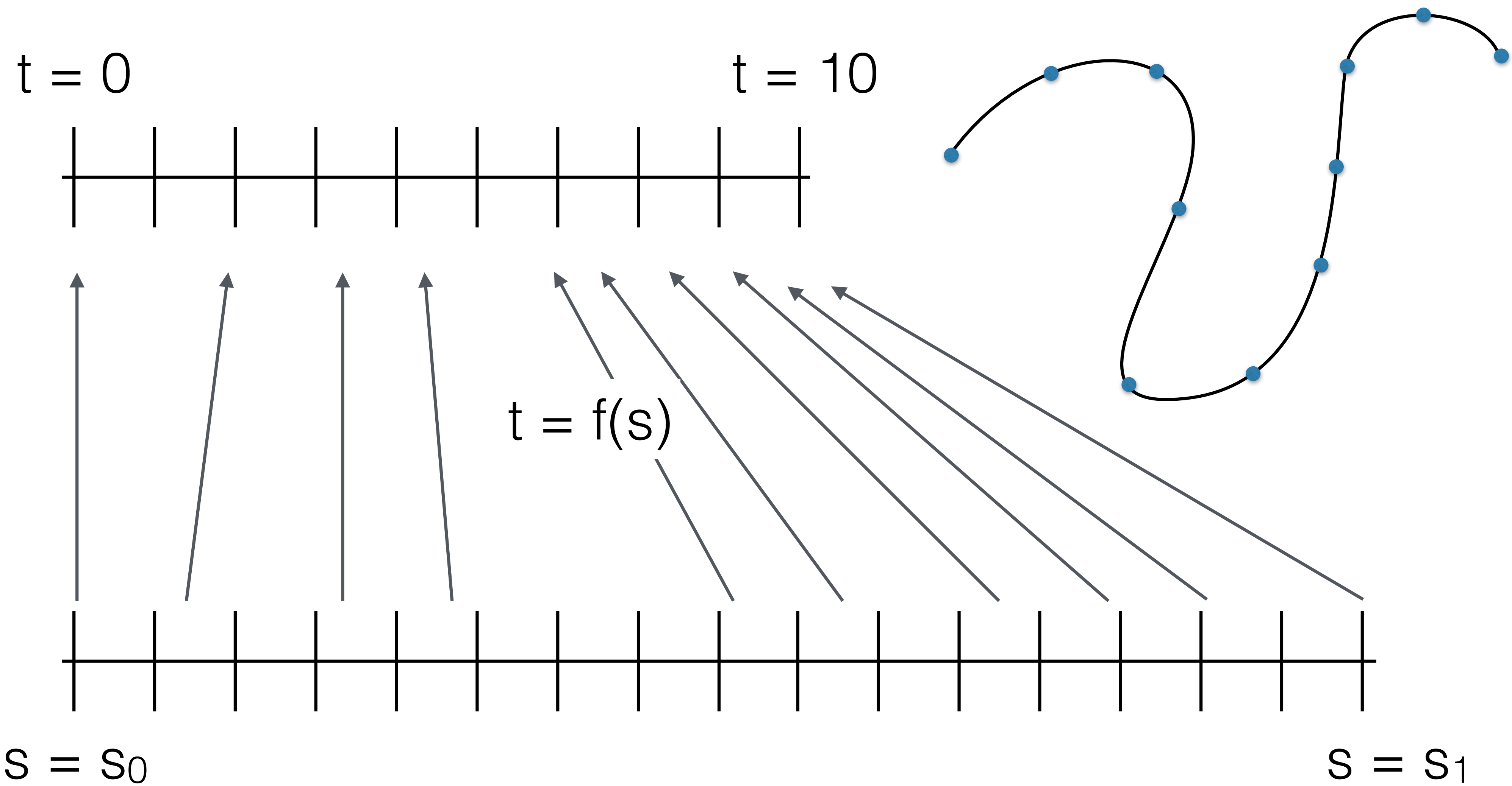




$s = s_0$ $s = s_1$

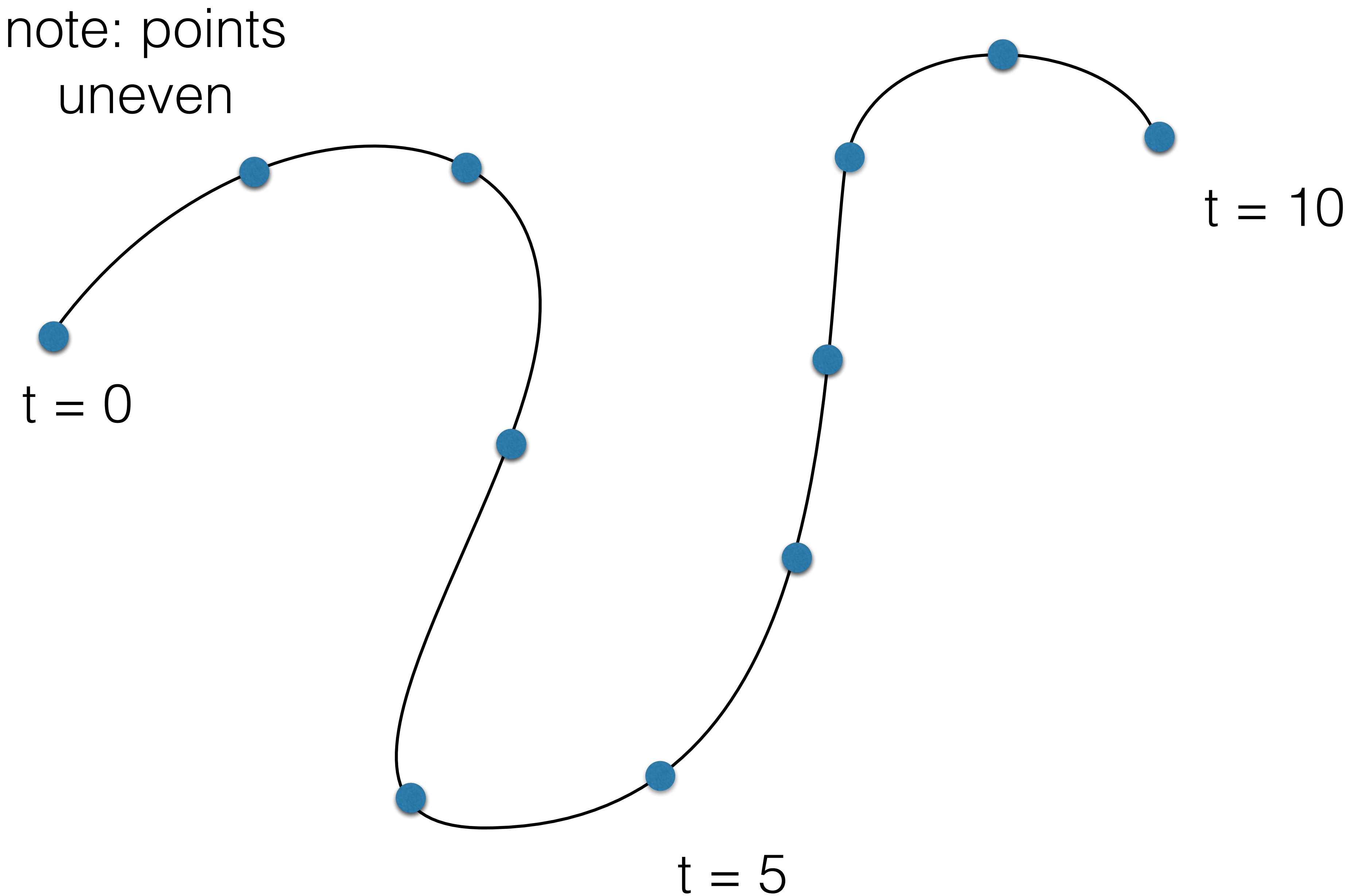
$\mathbf{f_2}(s) = \mathbf{f_1}(f(s))$

Parameterization, re-parameterization



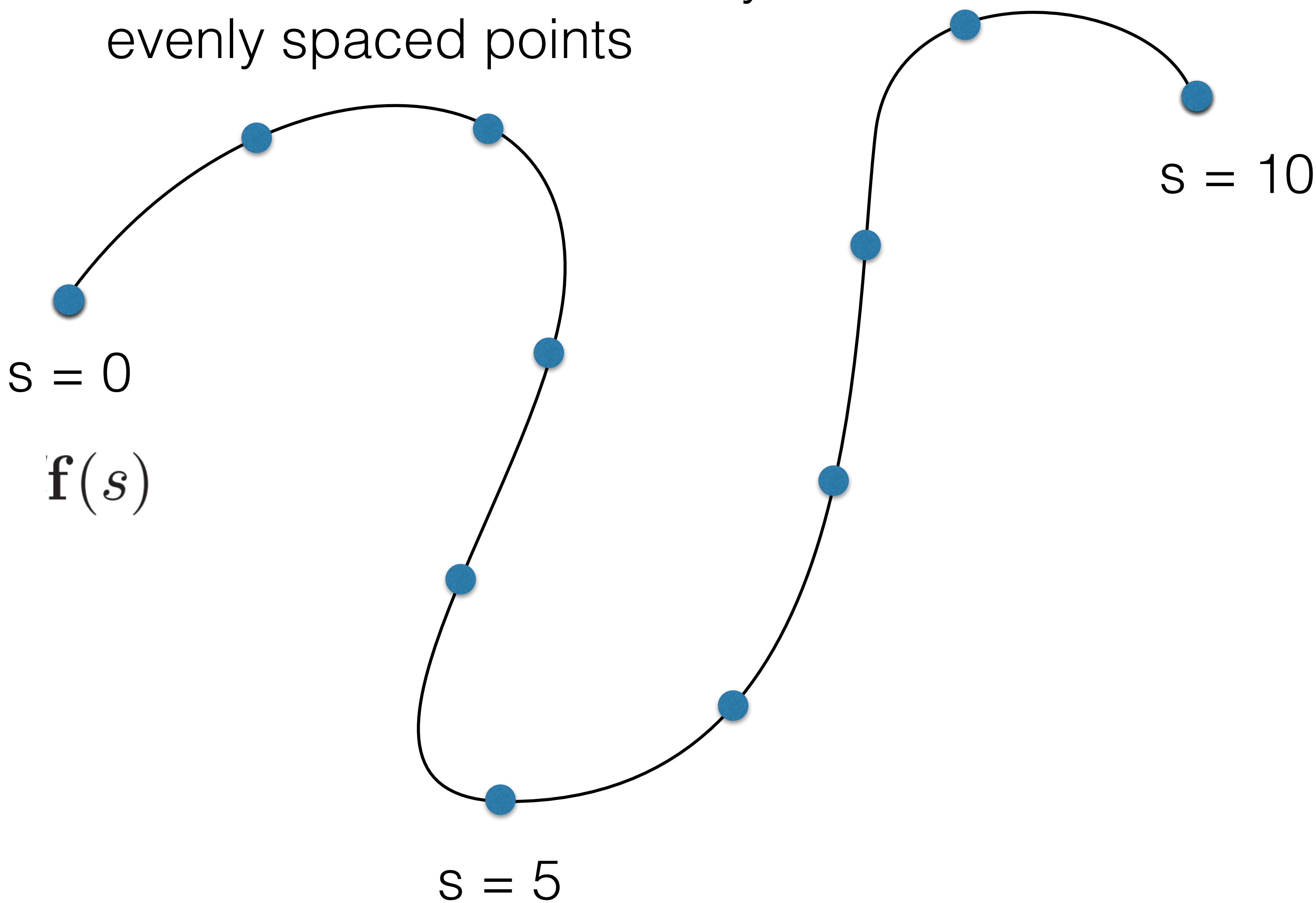
$\mathbf{f_2}(s) = \mathbf{f_1}(f(s))$

Natural parameterization



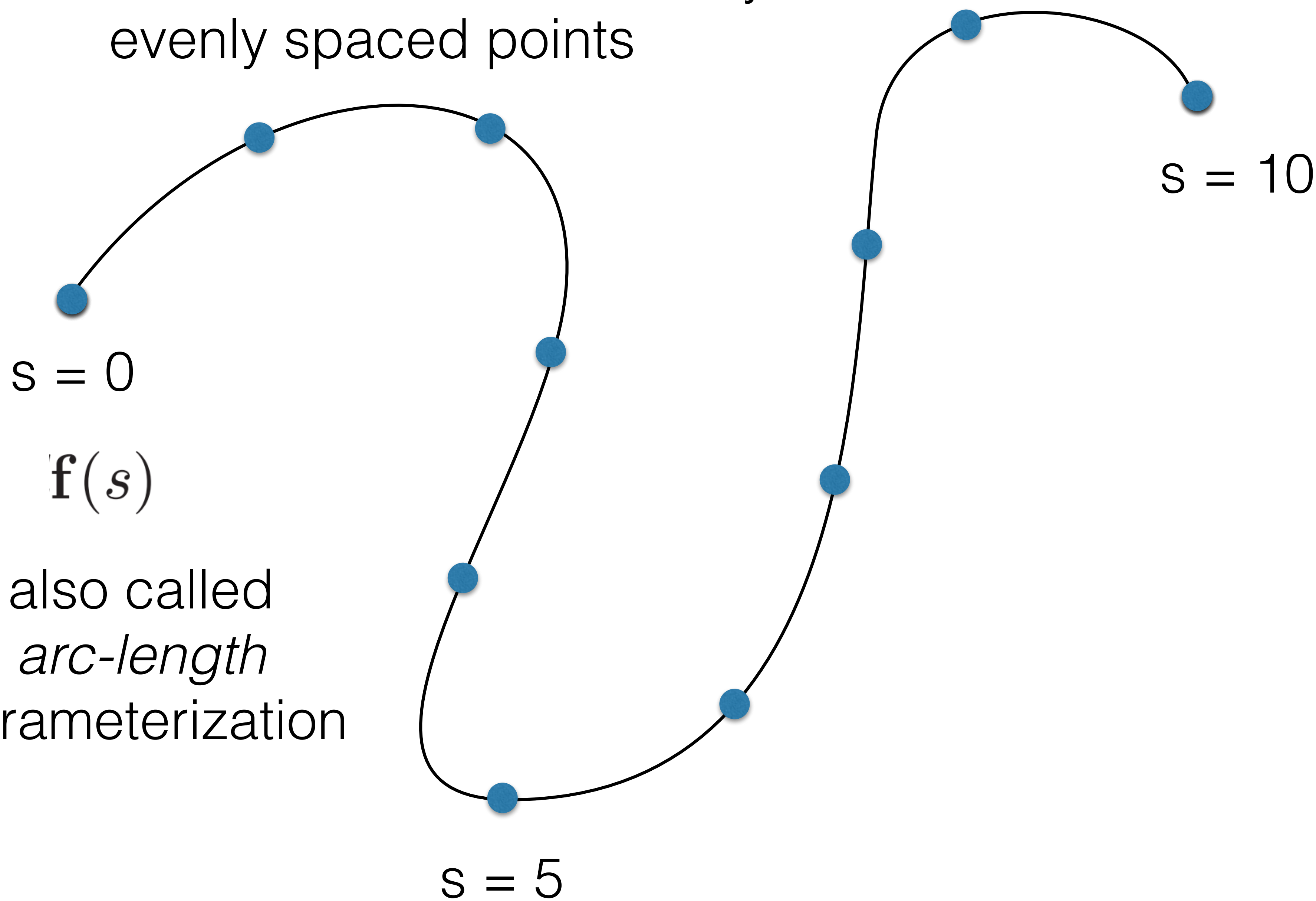
Natural parameterization

pen moves at a constant velocity:
evenly spaced points



Natural parameterization

pen moves at a constant velocity:
evenly spaced points

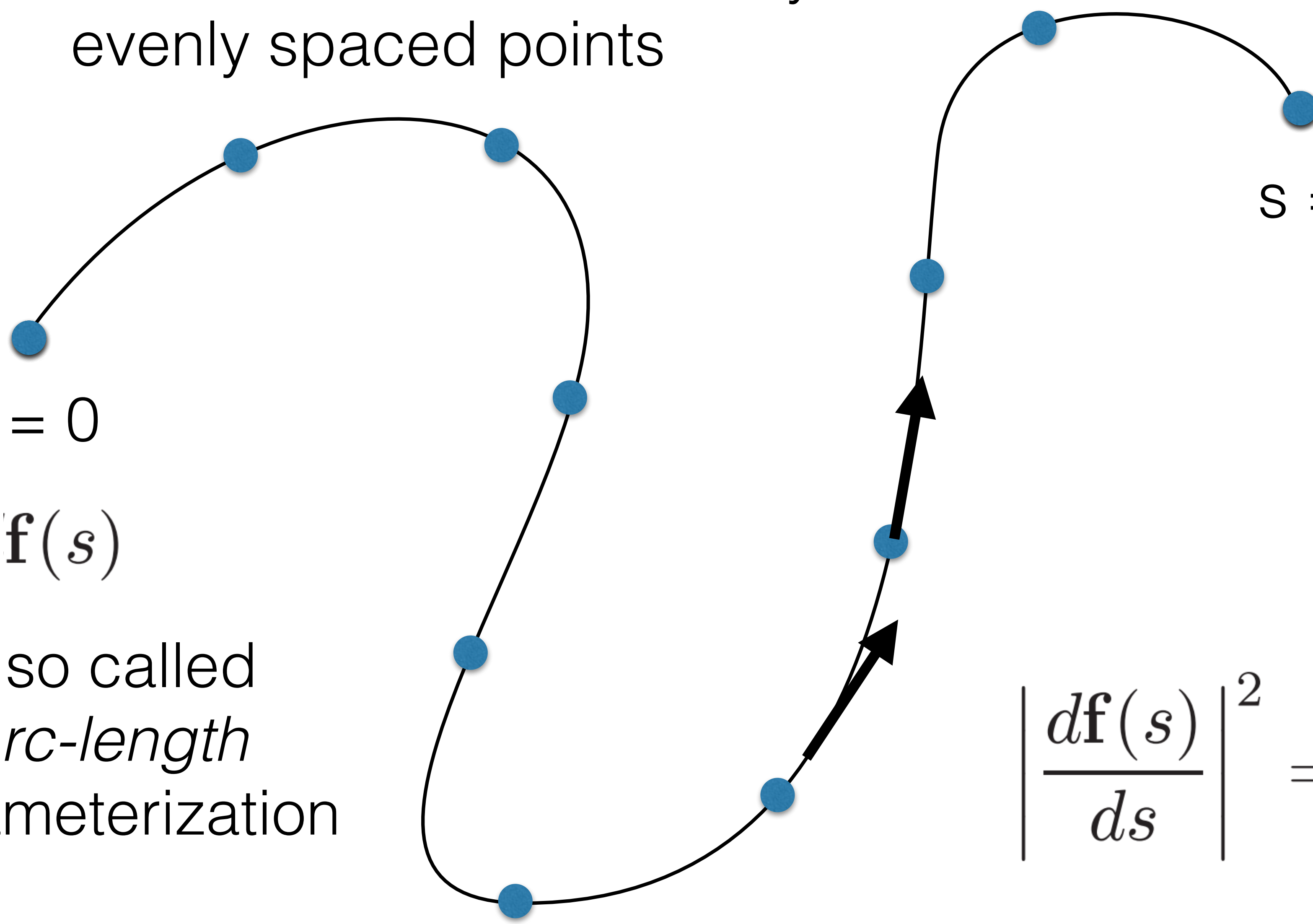


$\mathbf{f}(s)$

also called *arc-length* parameterization

Natural parameterization

pen moves at a constant velocity:
evenly spaced points



$\mathbf{f}(s)$

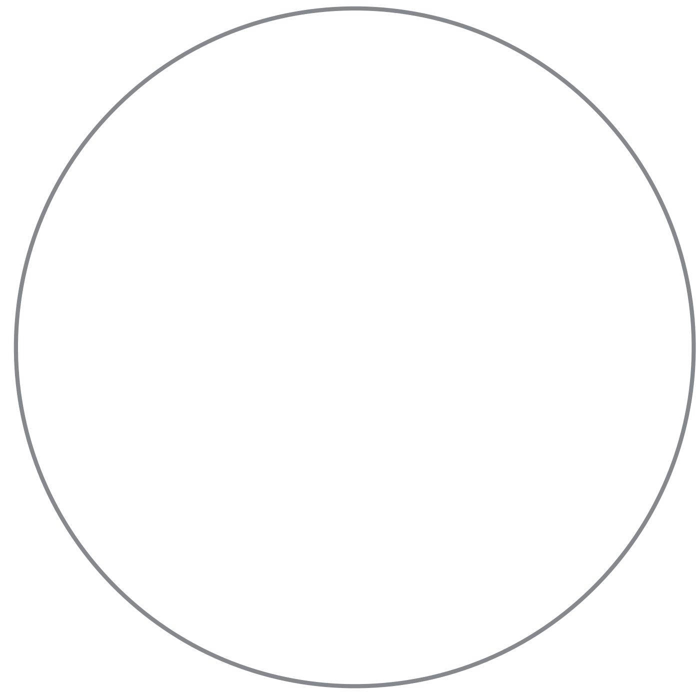
also called
arc-length
parameterization

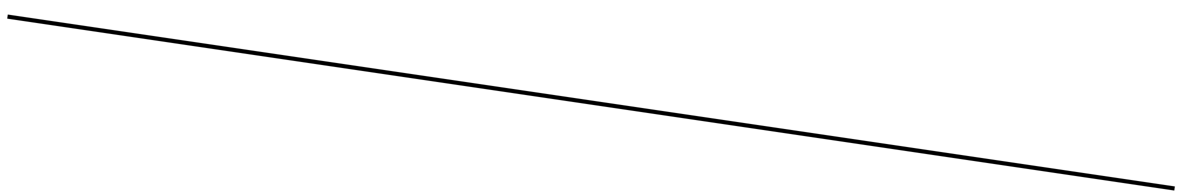
$$\left|\frac{d\mathbf{f}(s)}{ds}\right|^2 = c$$

piecewise parametric representation

sometimes easy
to find a parametric
representation

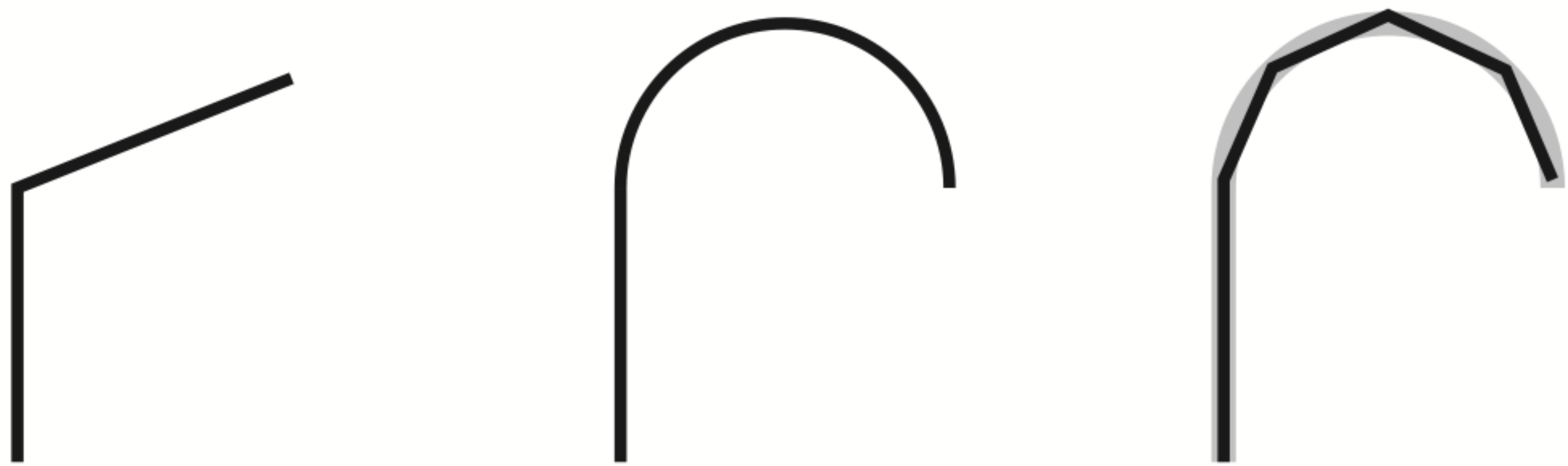
e.g., circle, line segment





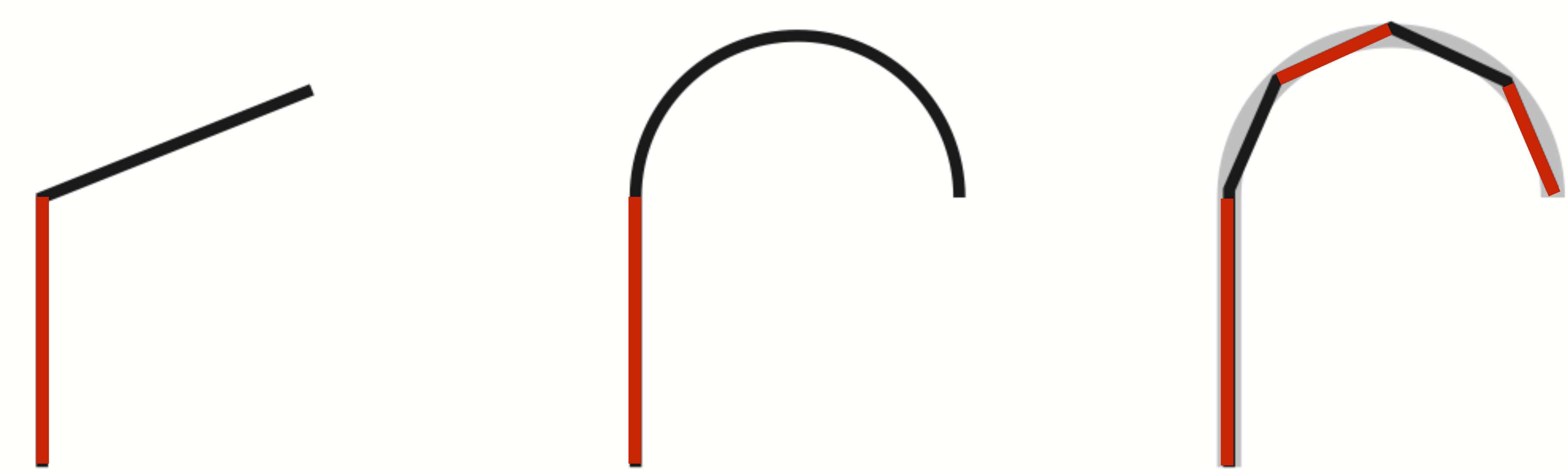
piecewise parametric representation

in other cases, not obvious



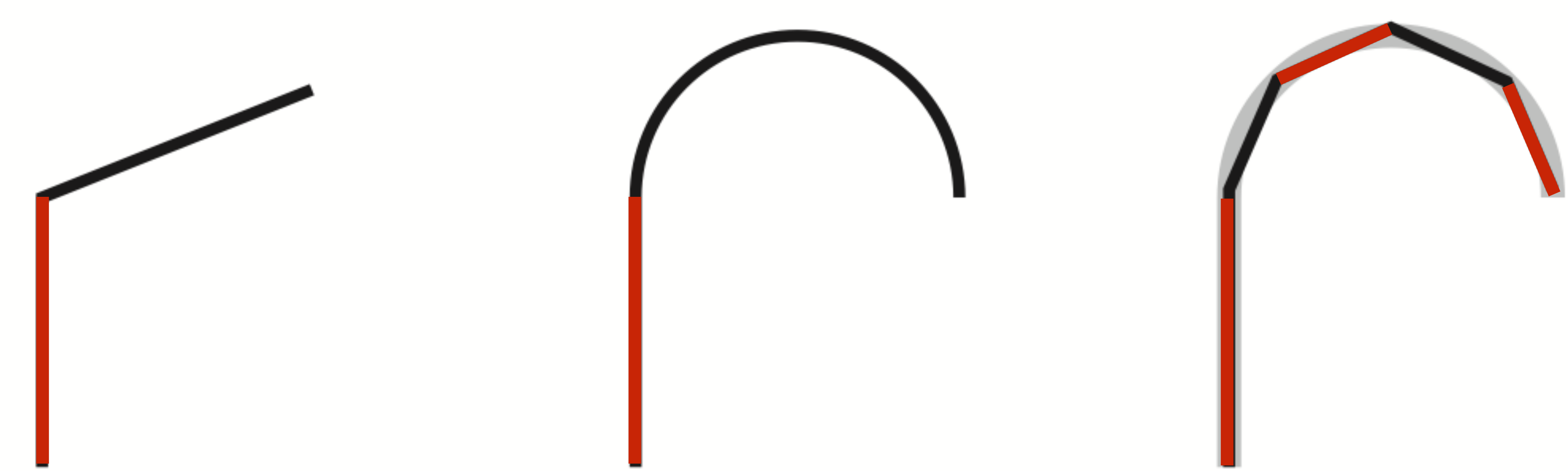
piecewise parametric representation

strategy: break into simpler pieces



piecewise parametric representation

strategy: break into simpler pieces

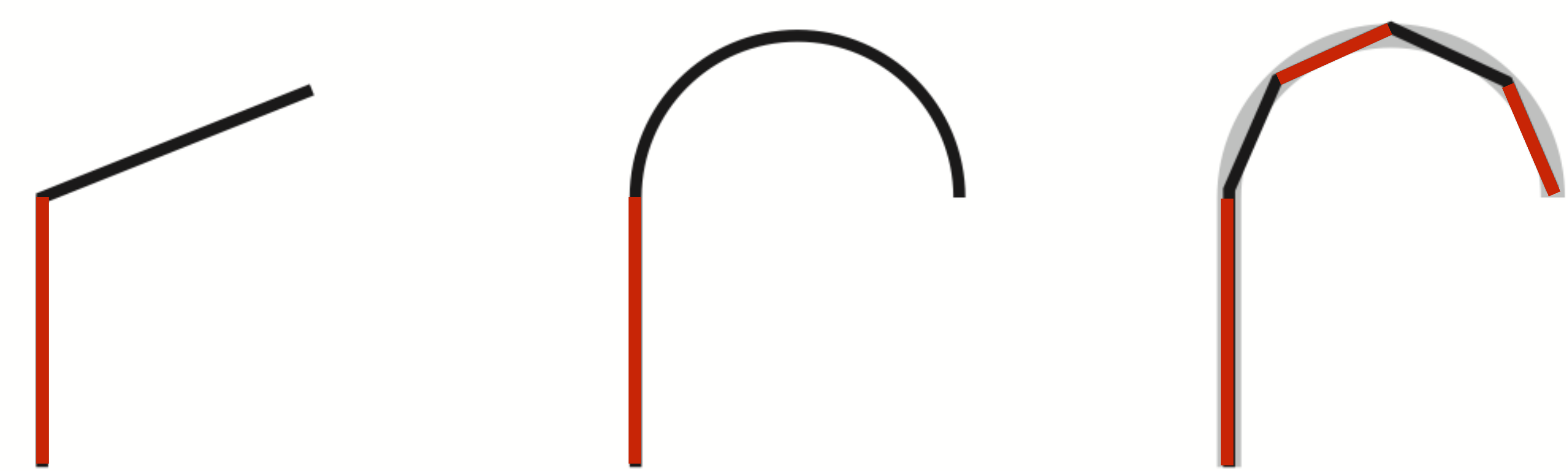


switch between functions that represent pieces:

$$\mathbf{f}(u) = \begin{cases} \mathbf{f}_1(2u) & u \le 0.5 \\ \mathbf{f}_2(2u-1) & u > 0.5 \end{cases}$$

piecewise parametric representation

strategy: break into simpler pieces



switch between functions that represent pieces:

$$\mathbf{f}(u) = \begin{cases} \mathbf{f}_1(2u) & u \leq 0.5 \\ \mathbf{f}_2(2u-1) & u > 0.5 \end{cases}$$

map the inputs to
f$_1$ and **f**$_2$
to be from 0 to 1

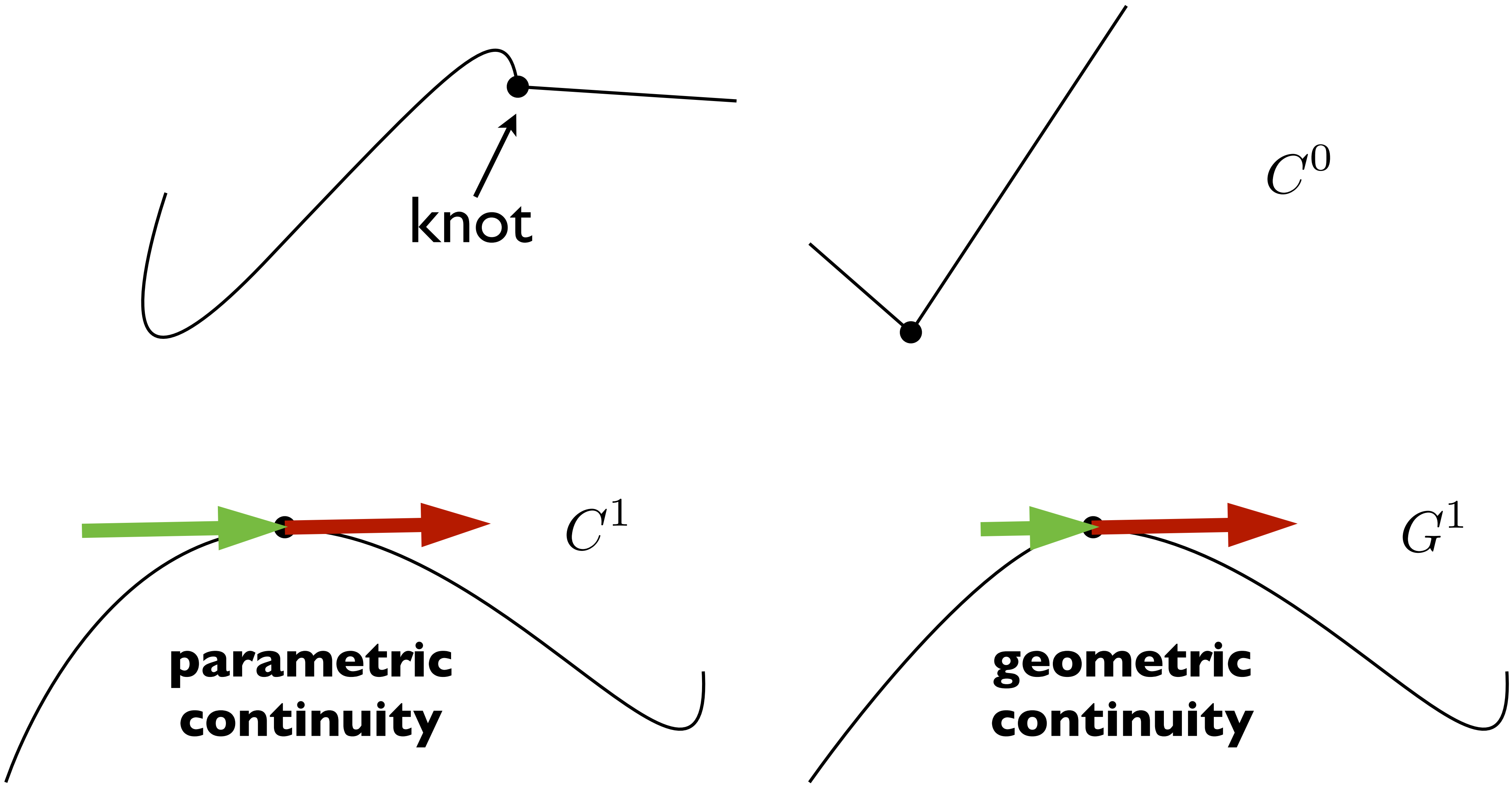
Curve Properties

Local properties:
- continuity
- position
- direction
- curvature

Global properties (examples):
- closed curve
- curve crosses itself

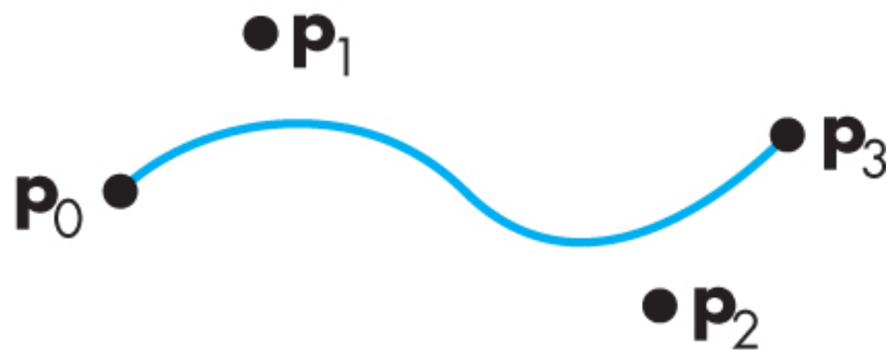
Interpolating vs. non-interpolating

Continuity: stitching curve segments together



Interpolating vs. Approximating Curves

Interpolating



Approximating
(non-interpolating)

Finding a Parametric Representation

Polynomial Pieces

$$f(u) = a_0 + a_1 u + a_2 u^2 + \cdots + a_n u^n$$

Polynomial Pieces

coefficients ... **n = degree**

$$f(u) = a_0 + a_1 u + a_2 u^2 + \cdots + a_n u^n$$

Polynomial Pieces

coefficients ... **n = degree**

$$f(u) = a_0 + a_1 u + a_2 u^2 + \cdots + a_n u^n$$

“canonical form” (monomial basis)

Blending functions are more convenient basis than monomial basis

- "canonical form" (monomial basis)

$$\mathbf{f}(u) = \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2 + \mathbf{a}_3 u^3$$

- "geometric form" (blending functions)

$$\mathbf{f}(u) = b_0(u)\mathbf{p}_0 + b_1(u)\mathbf{p}_1 + b_2(u)\mathbf{p}_2 + b_3(u)\mathbf{p}_3$$

Blending functions are more convenient basis than monomial basis

- "canonical form" (monomial basis)

$$\mathbf{f}(u) = \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2 + \mathbf{a}_3 u^3$$

- "geometric form" (blending functions)

$$\mathbf{f}(u) = b_0(u)\mathbf{p}_0 + b_1(u)\mathbf{p}_1 + b_2(u)\mathbf{p}_2 + b_3(u)\mathbf{p}_3$$

Blending functions are more convenient basis than monomial basis

$$f(u) = a_0 + a_1 u + a_2 u^2 + a_3 u^3$$

$$\mathbf{u} = \begin{pmatrix} 1 \\ u \\ u^2 \\ u^3 \end{pmatrix} \qquad \mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$f(u) = \mathbf{u} \cdot \mathbf{a} = \mathbf{u}^T \mathbf{a}$$

Blending functions are more convenient basis than monomial basis

$$C\mathbf{a} = \mathbf{p}$$

$$\mathbf{a} = C^{-1}\mathbf{p} = B\mathbf{p}$$

$$\mathbf{p} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

$$\begin{aligned} f(u) = \mathbf{u}^T\mathbf{a} &= \mathbf{u}^T(B\mathbf{p}) \\ &= (\mathbf{u}^T B)\mathbf{p} \\ &= \mathbf{b}(u)^T\mathbf{p} \end{aligned}$$

$$\mathbf{b}(u) = \begin{pmatrix} b_0(u) \\ b_1(u) \\ b_2(u) \\ b_3(u) \end{pmatrix}$$

Blending functions are more convenient basis than monomial basis

$$C\mathbf{a} = \mathbf{p}$$

$$\mathbf{a} = C^{-1}\mathbf{p} = B\mathbf{p}$$

$$\mathbf{p} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

$$f(u) = \mathbf{u}^T\mathbf{a} = \mathbf{u}^T(B\mathbf{p})$$

$$= (\mathbf{u}^T B)\mathbf{p}$$

$$= \mathbf{b}(u)^T\mathbf{p}$$

$$\mathbf{b}(u) =$$

Some examples <whiteboard>

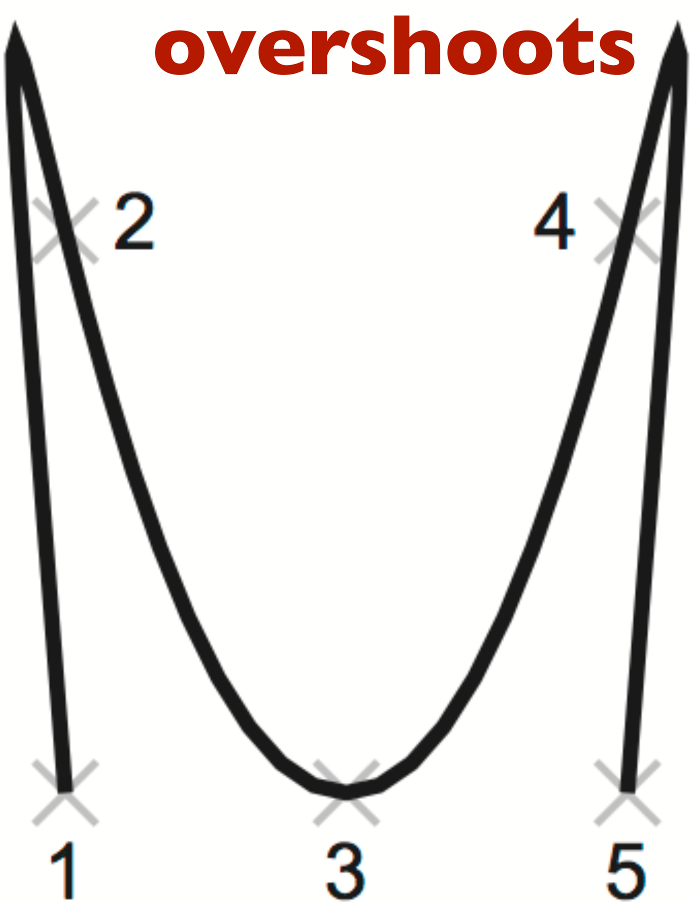
$$b_3(u)$$

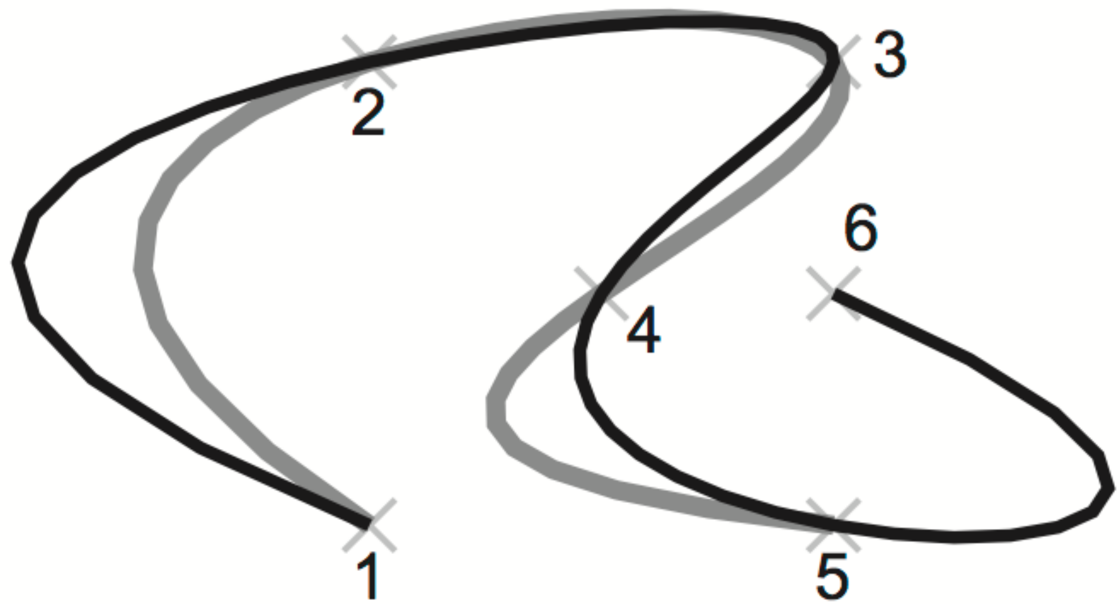
Interpolating Polynomials

Interpolating polynomials

- Given n+1 data points, can find a unique interpolating polynomial of degree n
- Different methods:
    - Vandermonde matrix
    - Lagrange interpolation
    - Newton interpolation

higher order interpolating polynomials are rarely used





non-local effects

4th order (gray) to 5th order (black)