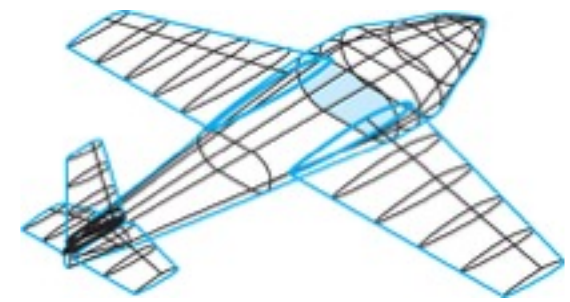
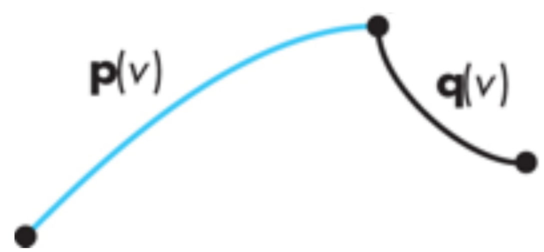
CS130 : Computer Graphics
Curves

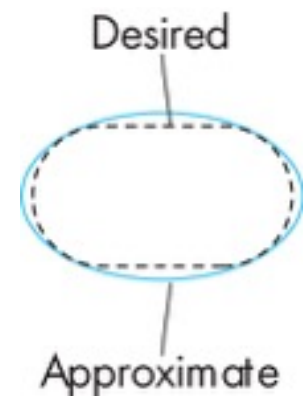
Tamar Shinar
Computer Science & Engineering
UC Riverside

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



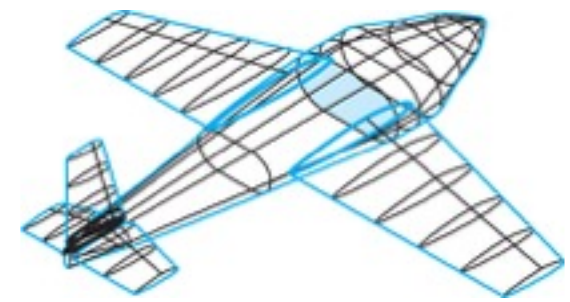


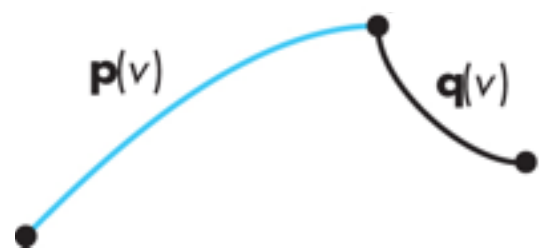


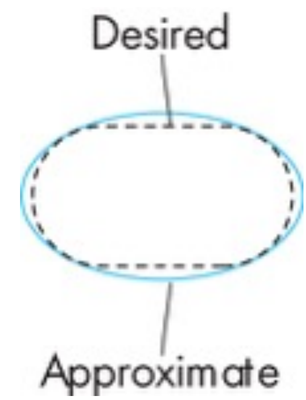
- local control - design each segment independently
- stability - small change in input values leads to small change in output

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





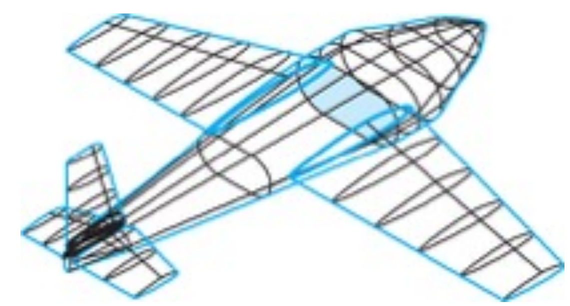


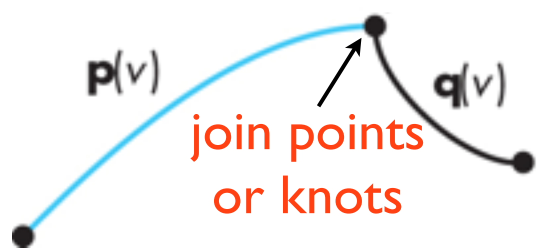
approximate out of a number of wood strips

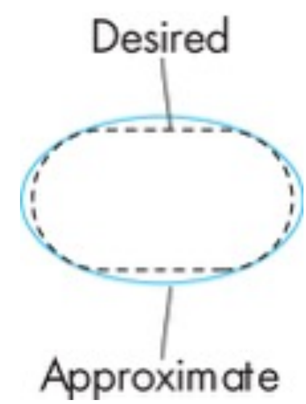
- local control - design each segment independently
- stability - small change in input values leads to small change in output

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





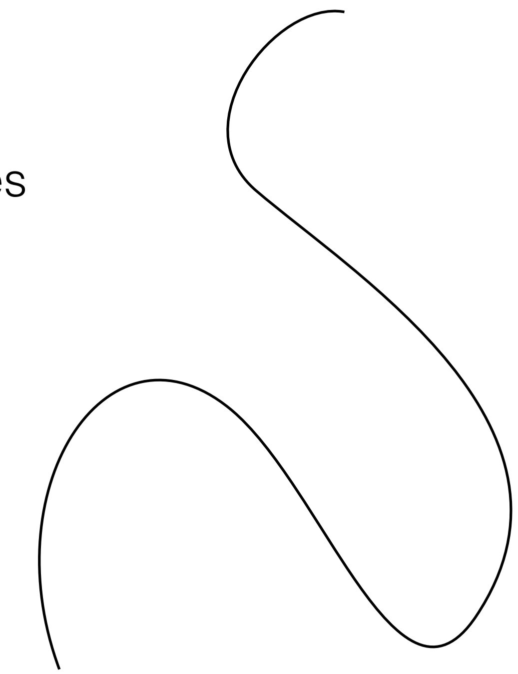


approximate out of a number of wood strips

- local control - design each segment independently
- stability - small change in input values leads to small change in output

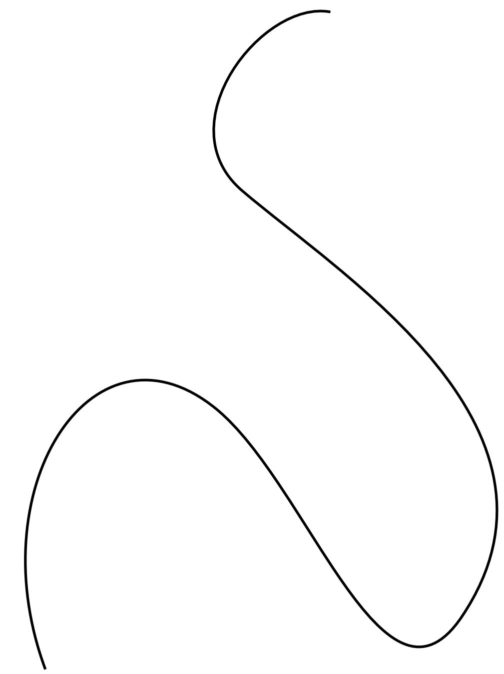
What is a curve?

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

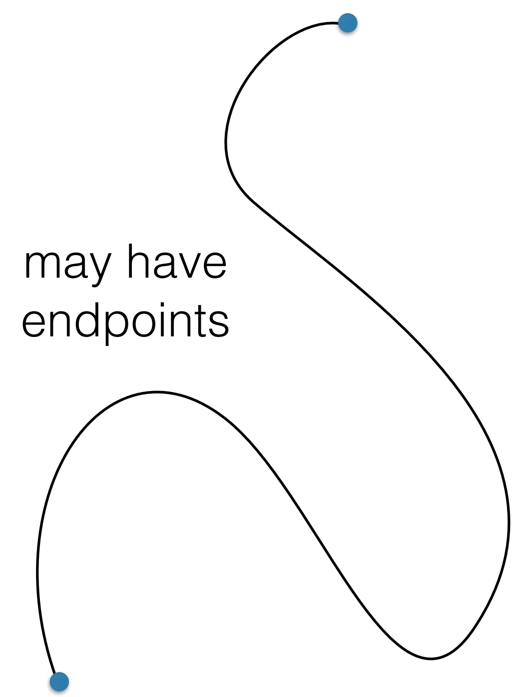


may be 2D, like on paper
or 3D, *space curve*

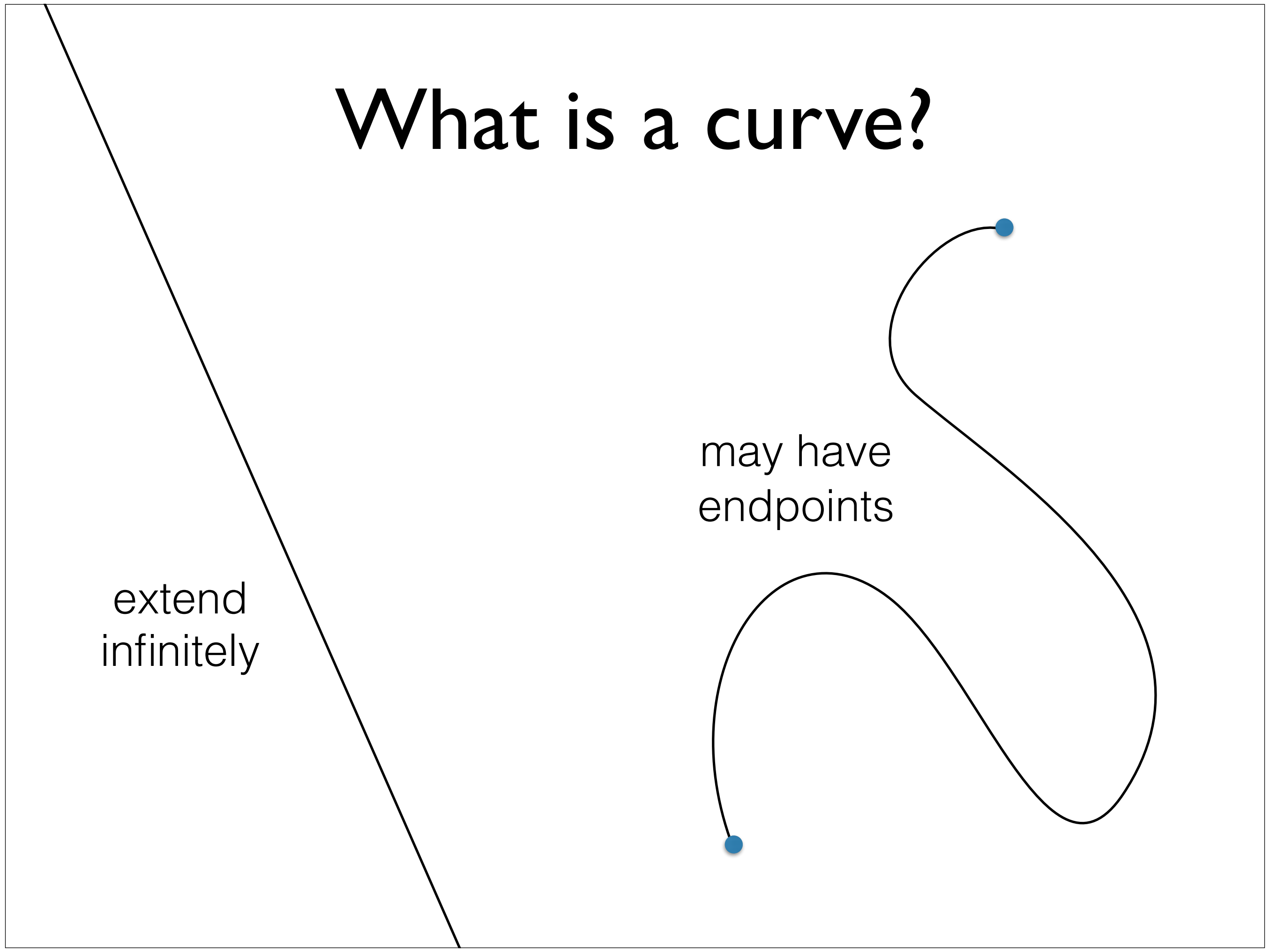
What is a curve?



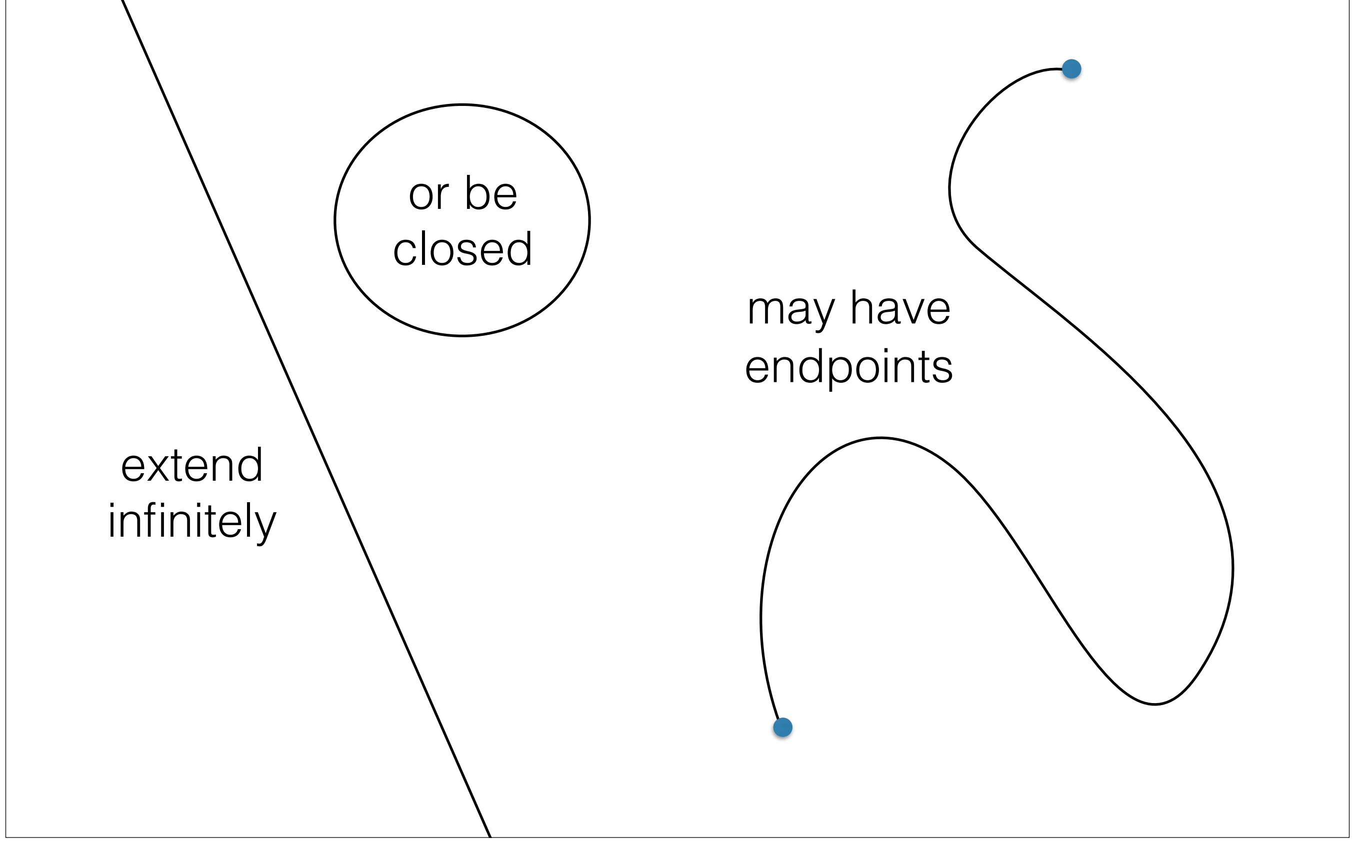
What is a curve?



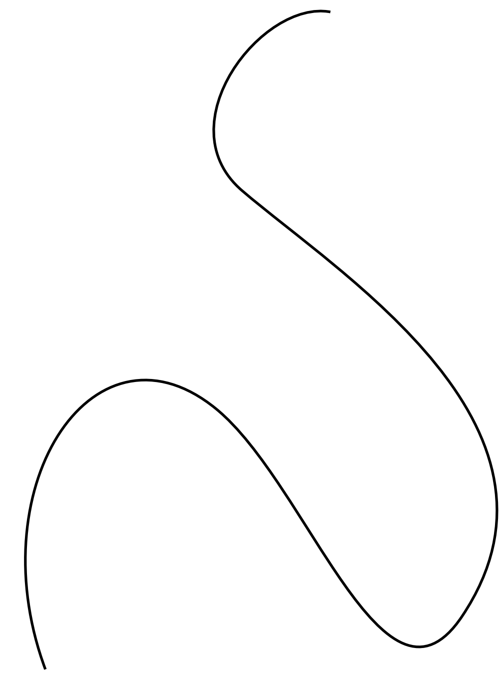
What is a curve?



What is a curve?



How do we specify a curve?

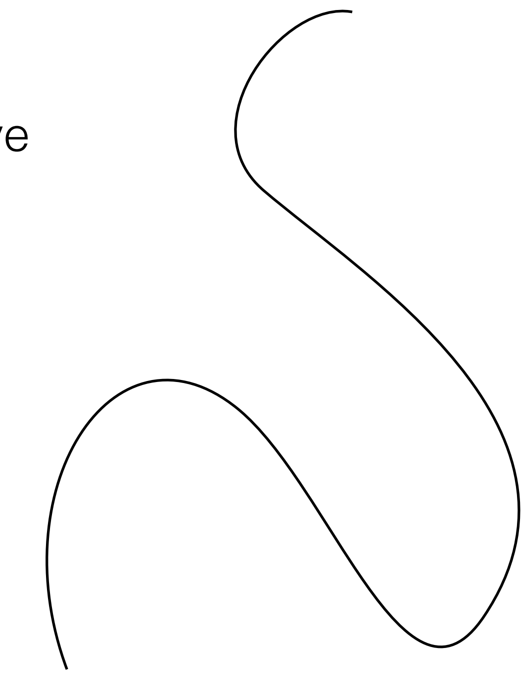


How do we specify a curve?

Implicit

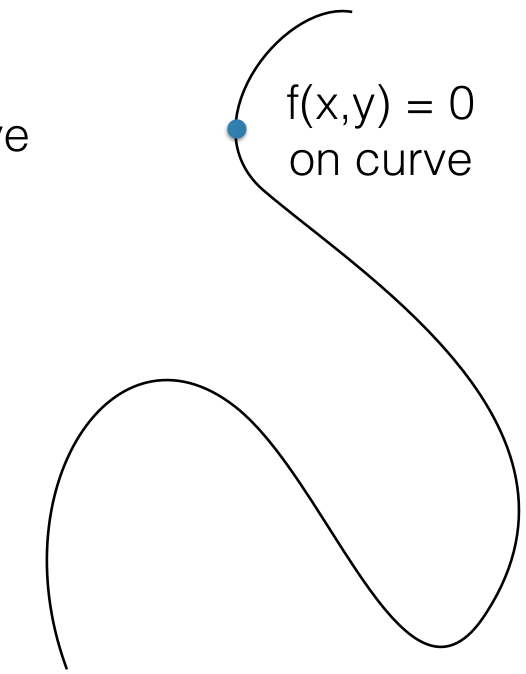
(2D) $f(x,y) = 0$

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

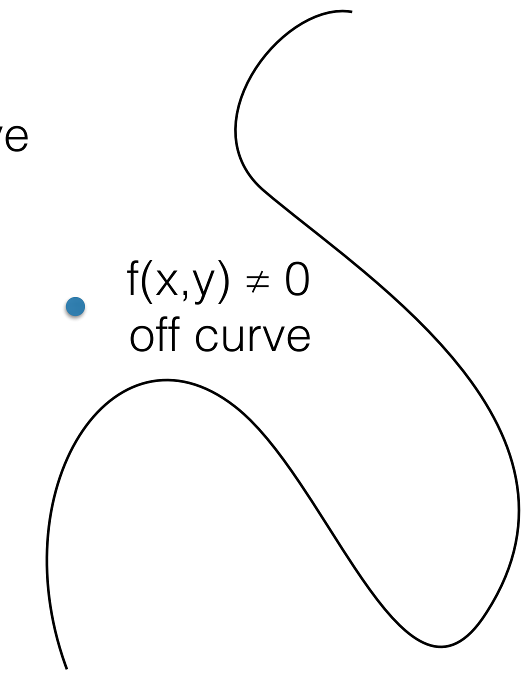


How do we specify a curve?

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

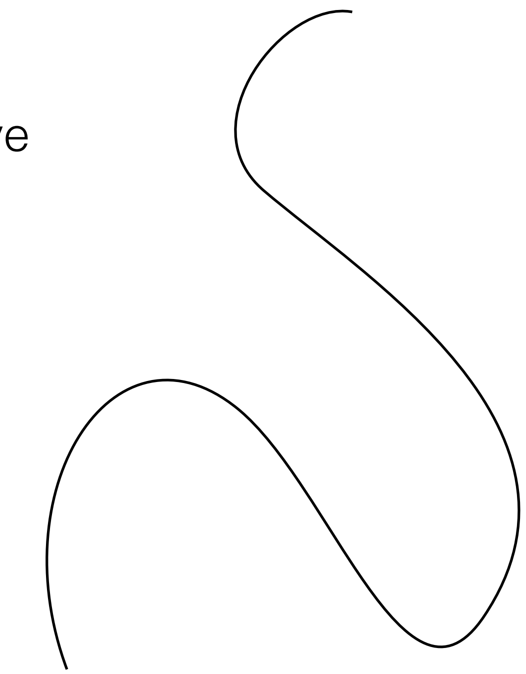
Parametric

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

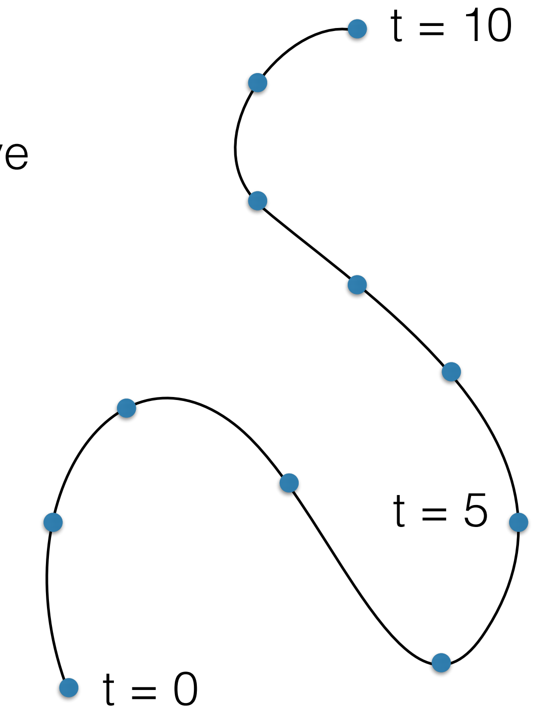
Parametric

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

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

map free *parameter* t

to points on the curve

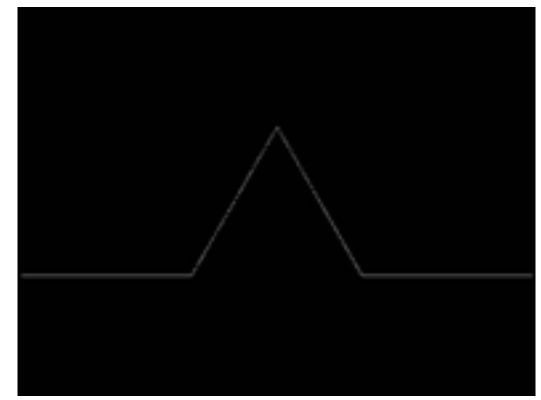


How do we specify a curve?

Implicit
(2D) f(x,y) = 0
test if (x,y) is on the curve

Parametric
(2D) (x,y) = **f**(t)
(3D) (x,y,z) = **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) = **f**(t)

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

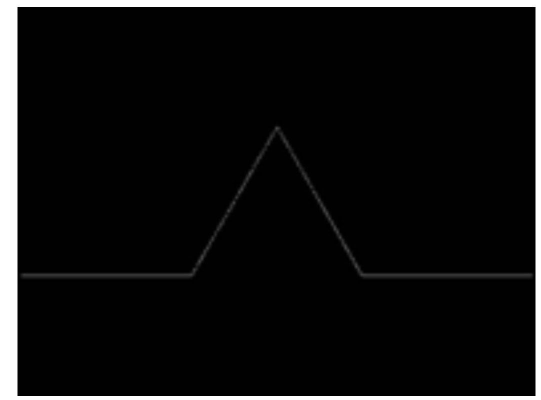
map free *parameter* t

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) = **f**(t)

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

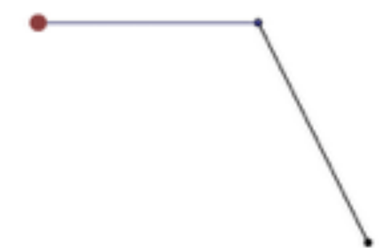
map free *parameter* t

to points on the curve

Procedural

e.g., fractals,

subdivision schemes



Bezier Curve

http://codegolf.stackexchange.com/questions/21178/animated-drawing-of-a-b%C3%A9zier-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)

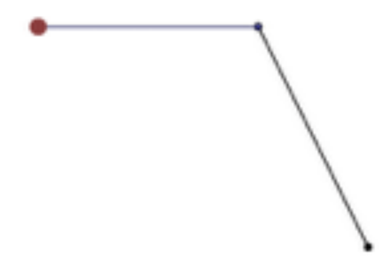
map free *parameter* t

to points on the curve

Procedural

e.g., fractals,

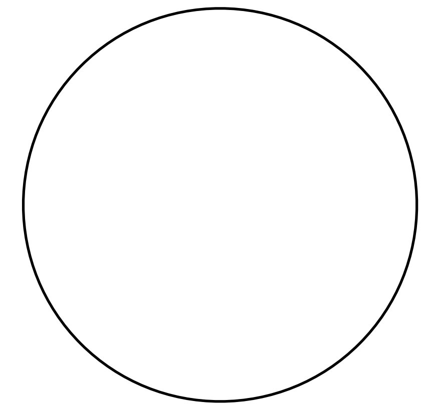
subdivision schemes



Bezier Curve

http://codegolf.stackexchange.com/questions/21178/animated-drawing-of-a-b%C3%A9zier-curve

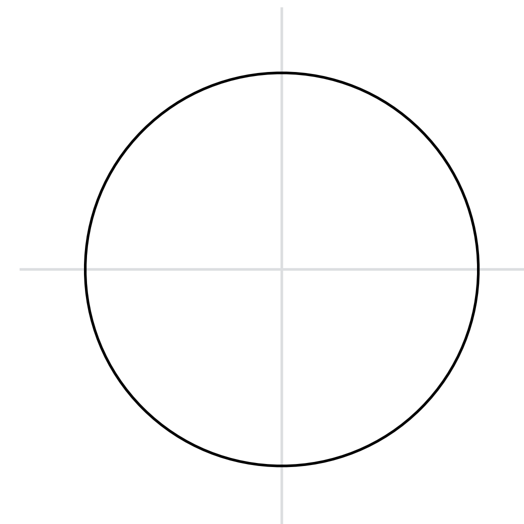
A curve may have multiple representations



A curve may have multiple representations

Implicit

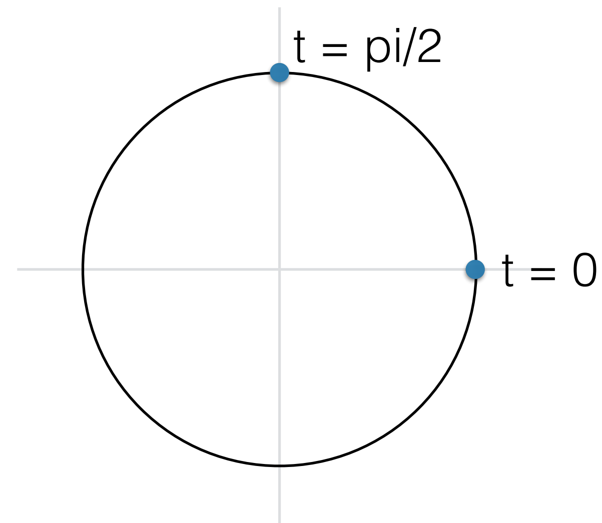
$f(x,y) = x^2 + y^2 - 1 = 0$



A curve may have multiple representations

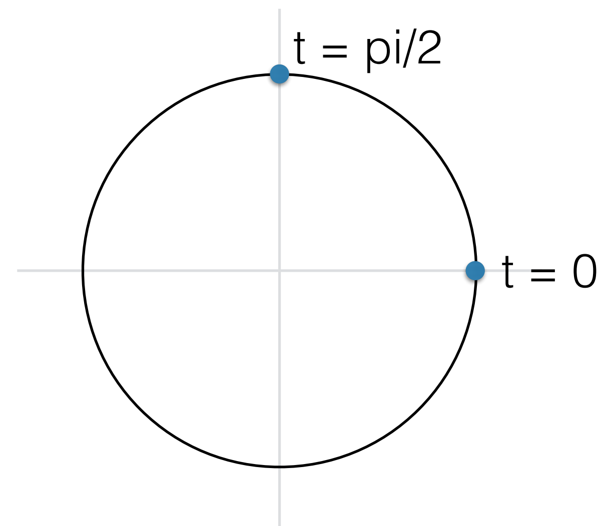
Parametric

(x,y) = **f**(t) = (cos t, sin t)



A curve may have multiple representations

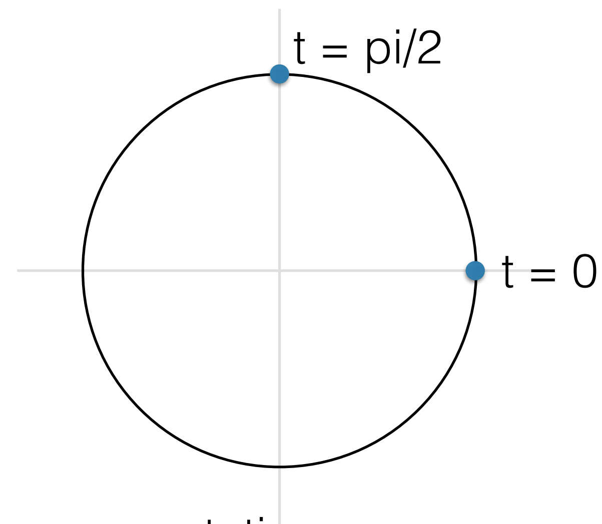
Parametric
(x,y) = **f**(t) = (cos t, sin t),
t in [0,2pi)



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

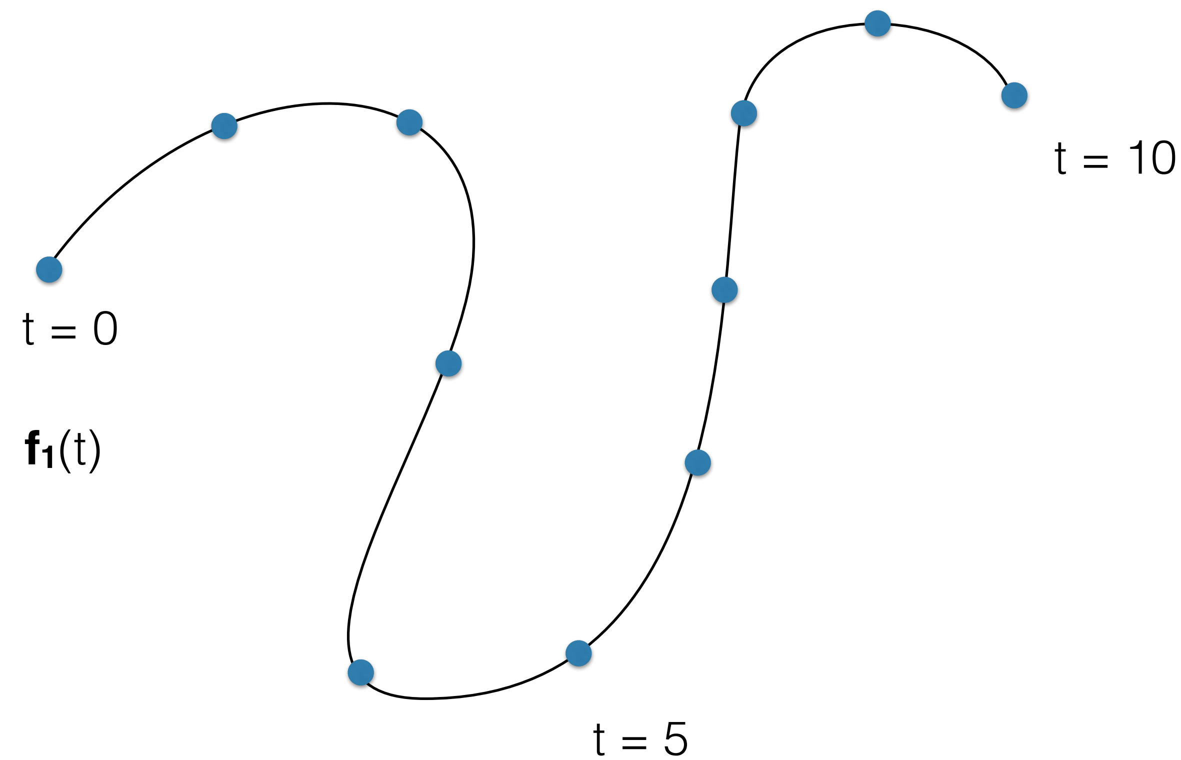
A curve may have multiple representations

Parametric
(x,y) = **f**(t) = (cos t, sin t),
t in [0,2pi)

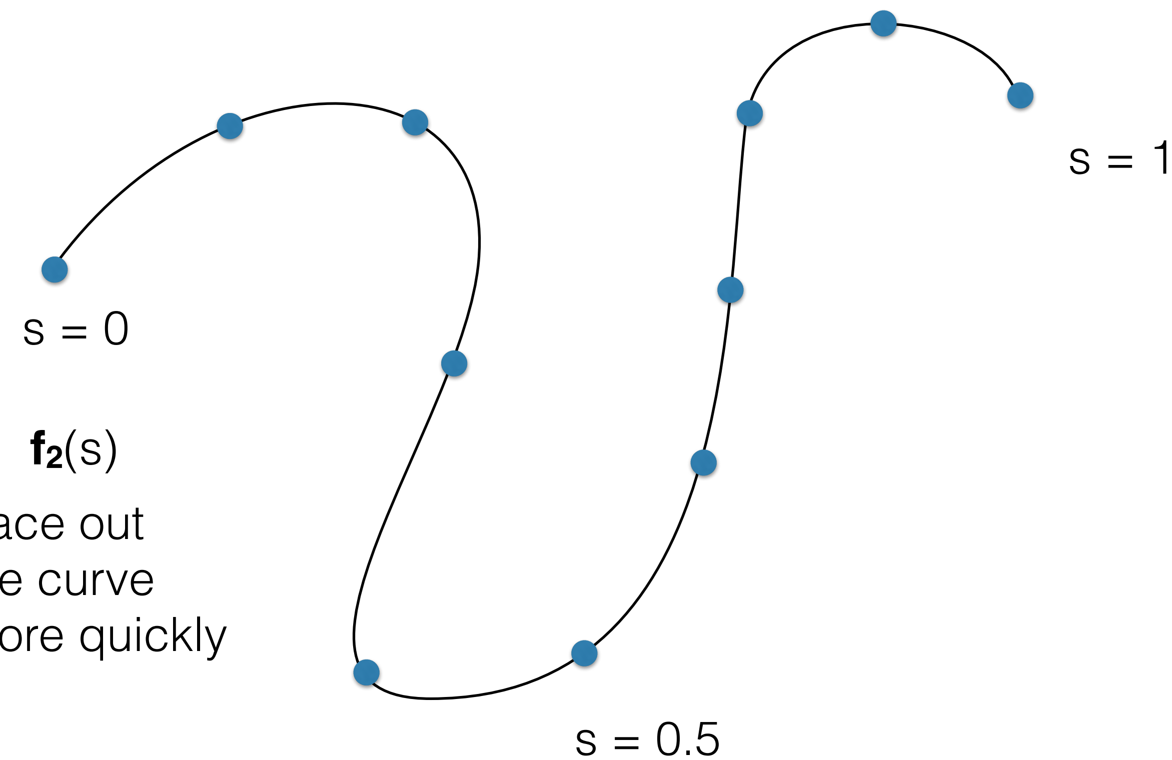


We will focus on parametric representations

Parameterization, re-parameterization



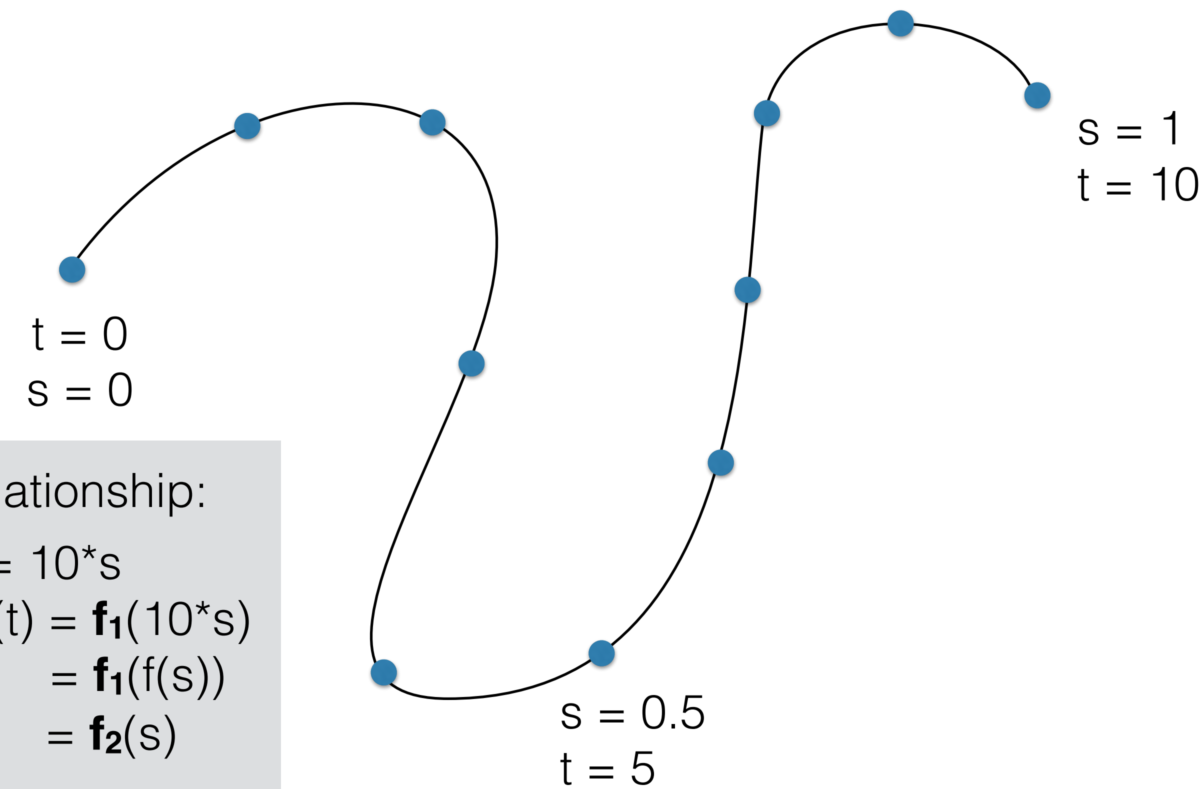
Parameterization, re-parameterization



f_2(s)

trace out
the curve
more quickly

Parameterization, re-parameterization



relationship:

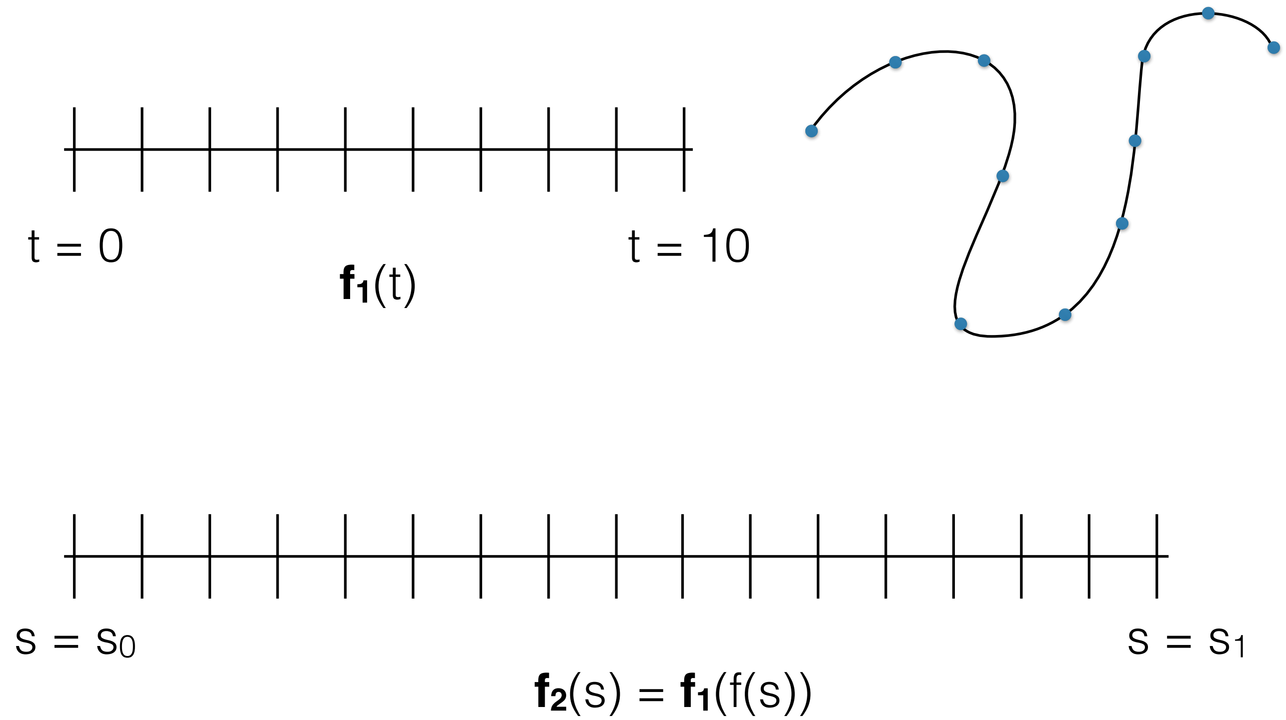
$t = 10*s$

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

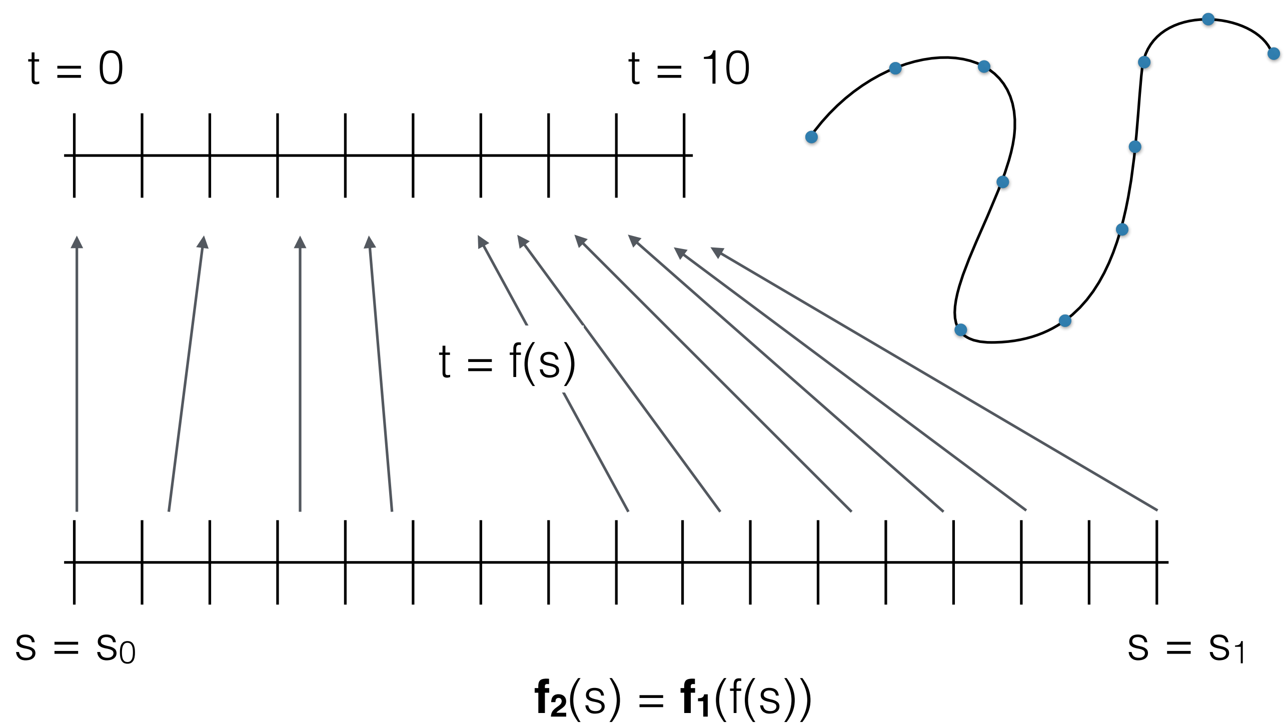
$= \mathbf{f_1}(f(s))$

$= \mathbf{f_2}(s)$

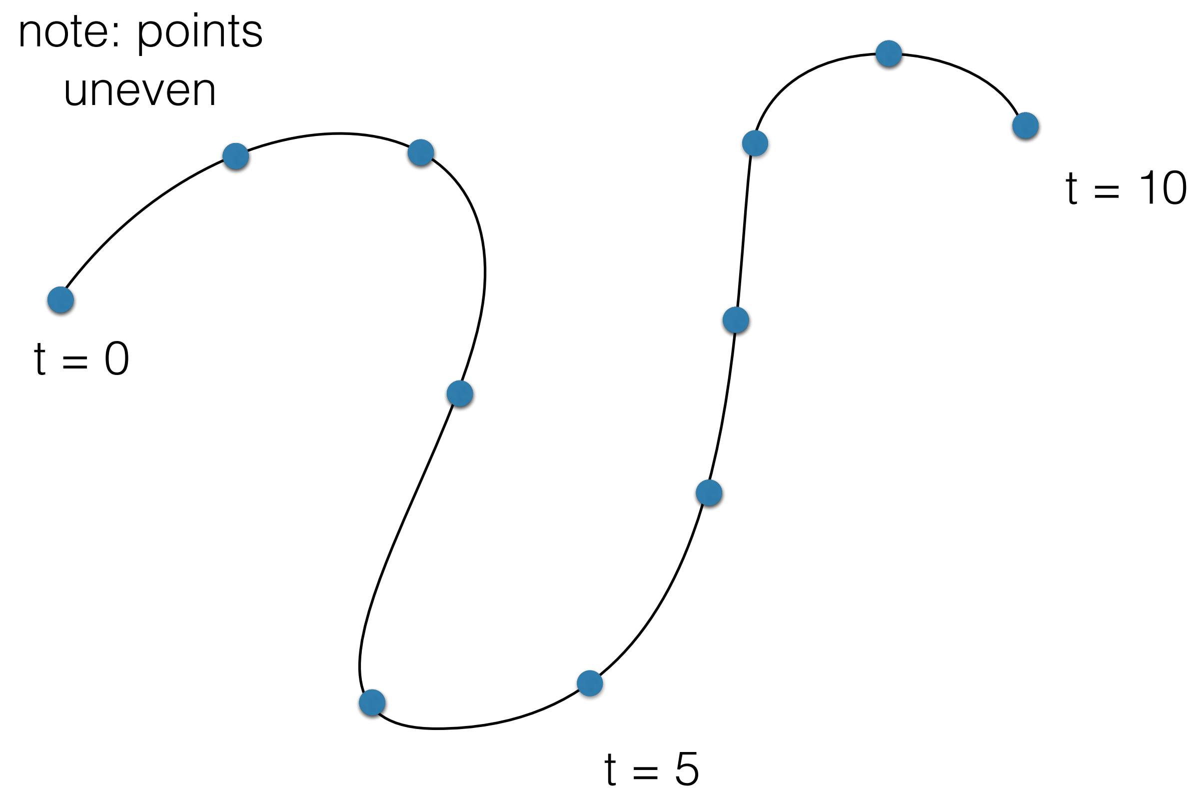
Parameterization, re-parameterization



Parameterization, re-parameterization

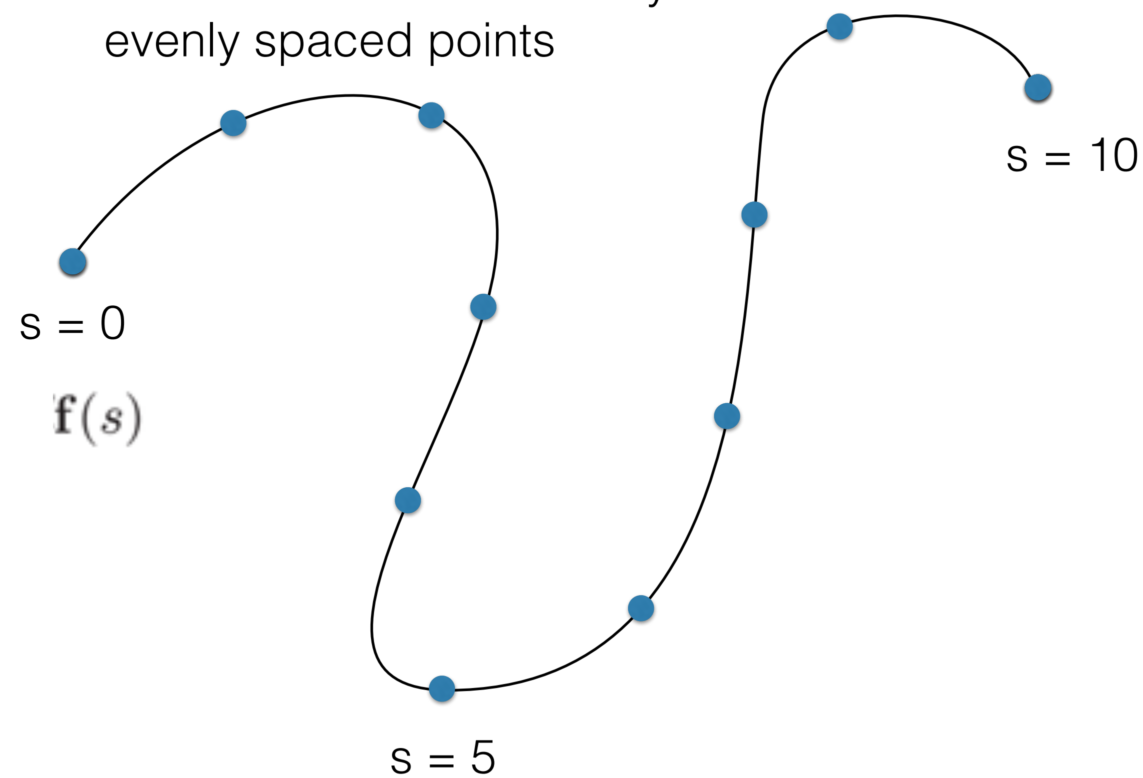


Natural parameterization

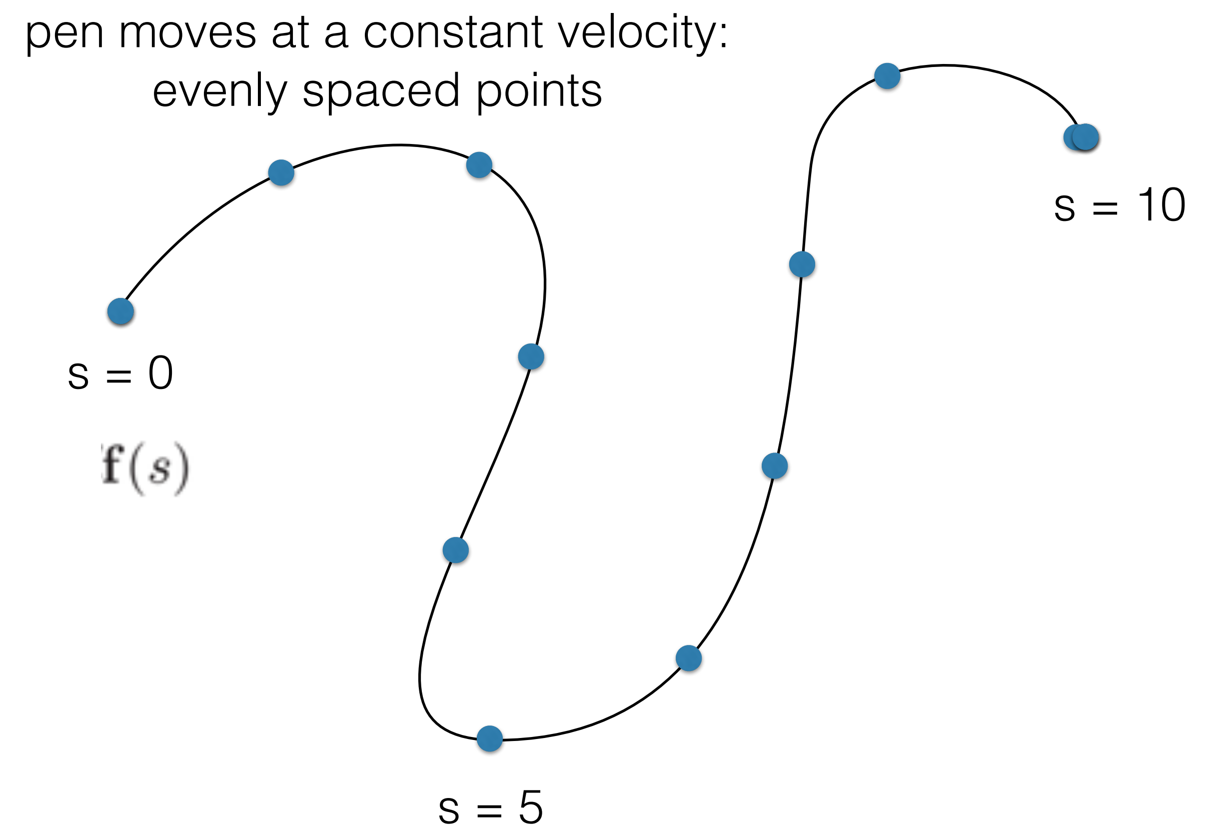


Natural parameterization

pen moves at a constant velocity:
evenly spaced points

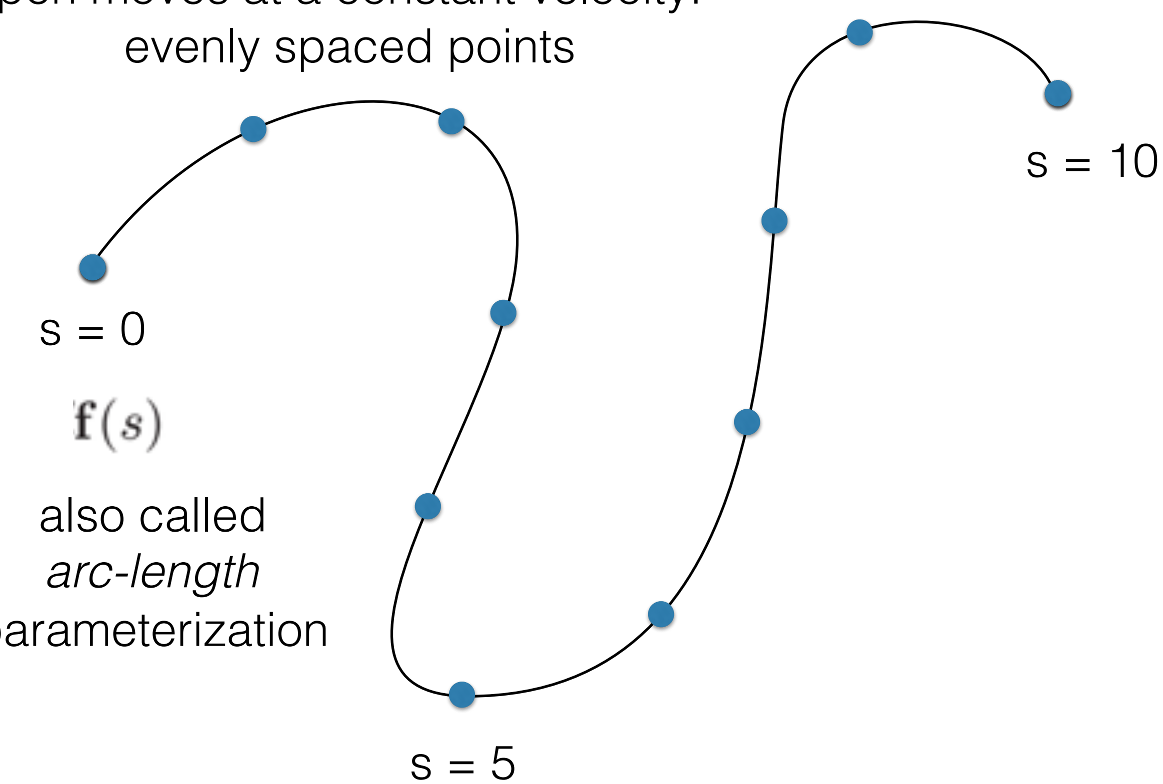


Natural parameterization



Natural parameterization

pen moves at a constant velocity:
evenly spaced points

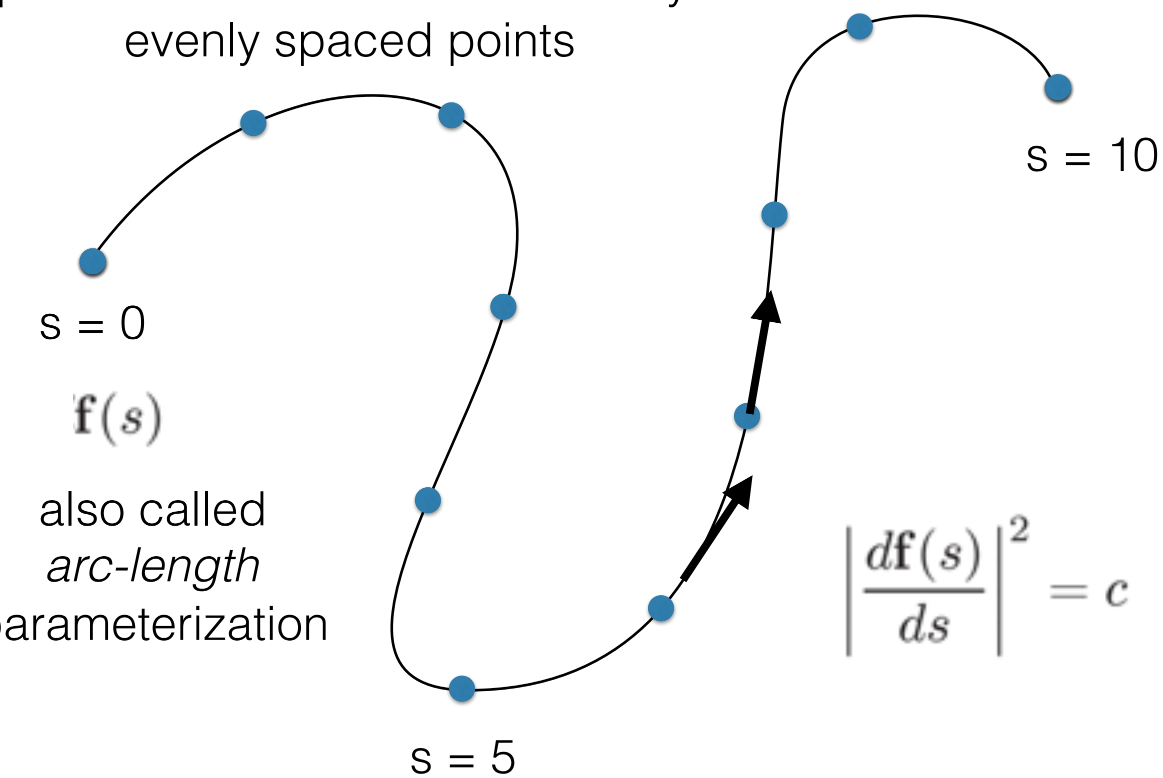


$\mathbf{f}(s)$

also called
arc-length
parameterization

Natural parameterization

pen moves at a constant velocity:
evenly spaced points



$\mathbf{f}(s)$

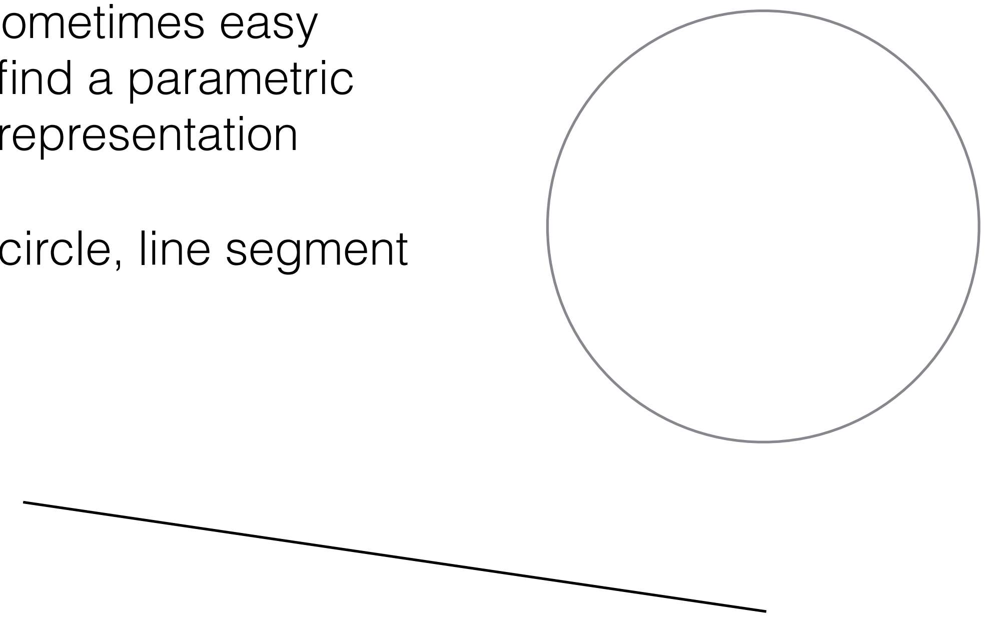
also called
arc-length
parameterization

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

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:

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

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
$\mathbf{f}_1$ and $\mathbf{f}_2$
to be from 0 to 1

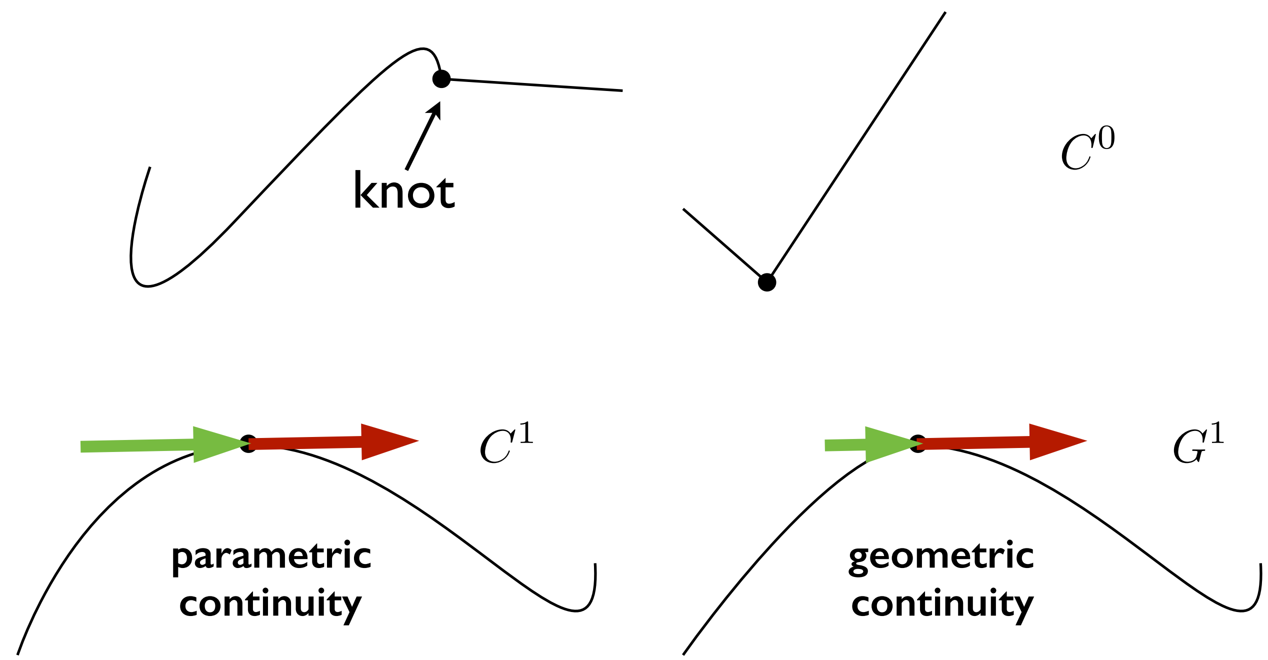
Curve Properties

Local properties:
- continuity
- position
- direction
- curvature

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

Interpolating vs. non-interpolating

Continuity: stitching curve segments together



Top
C0: the curves are continuous, but have discontinuous first derivatives
Bottom
Left: At the knot, the curve has C1 continuity: the curve segments have common point and first derivative
Right: At the knot, the curve has G1 continuity: the curve segments have a common point, and parallel first derivatives of different magnitude

Finding a Parametric Representation

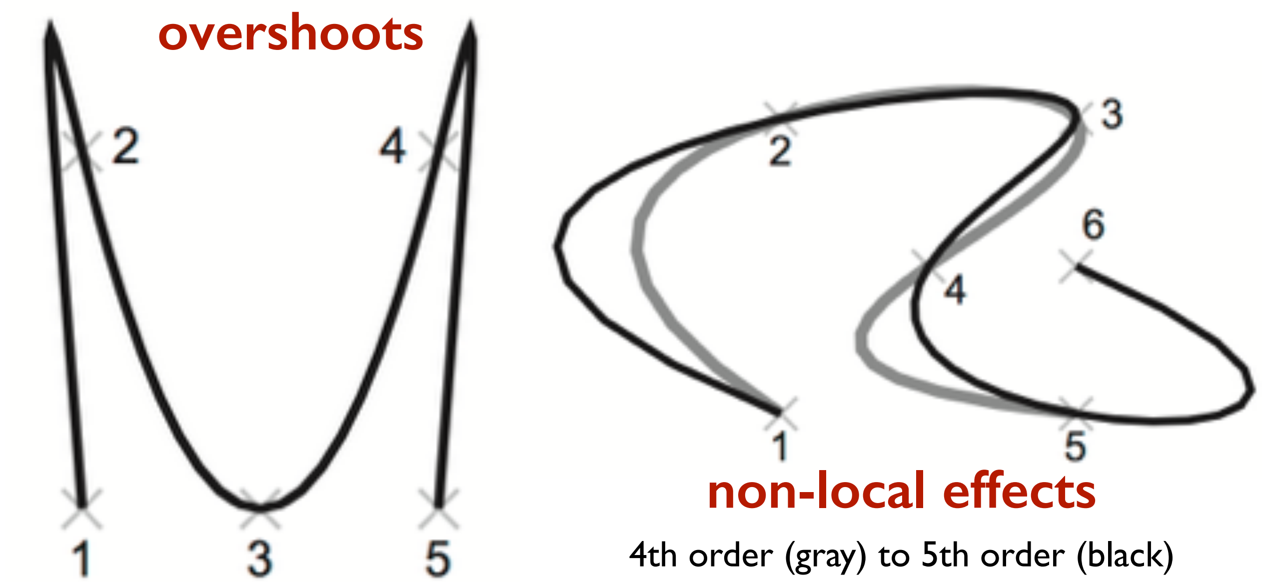
Polynomial Pieces

<whiteboard>

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



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

These images demonstrate problems with using higher order polynomials:
- overshoots
- non-local effects (in going from the 4th order polynomial in grey to the 5th order polynomial in black)