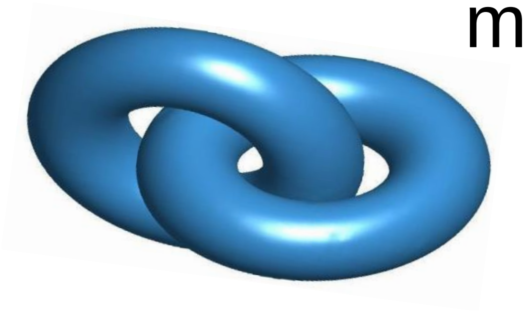
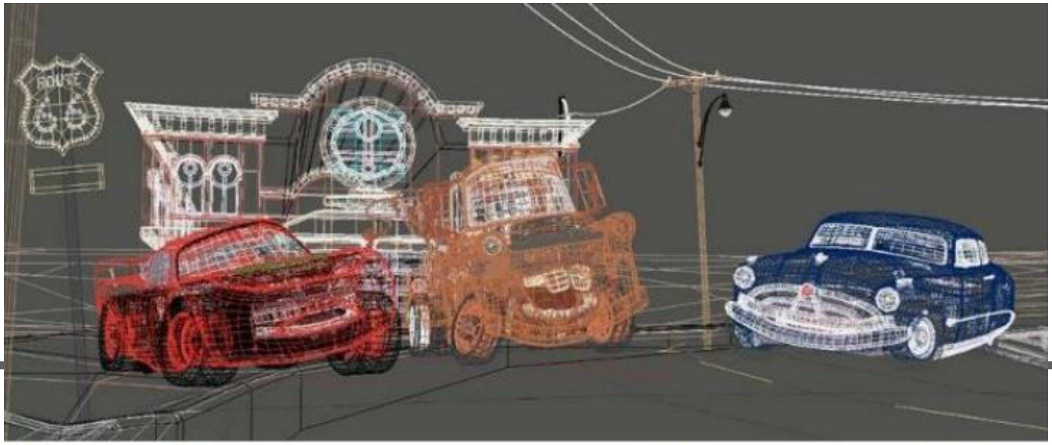
Introduction to Geometric Modeling

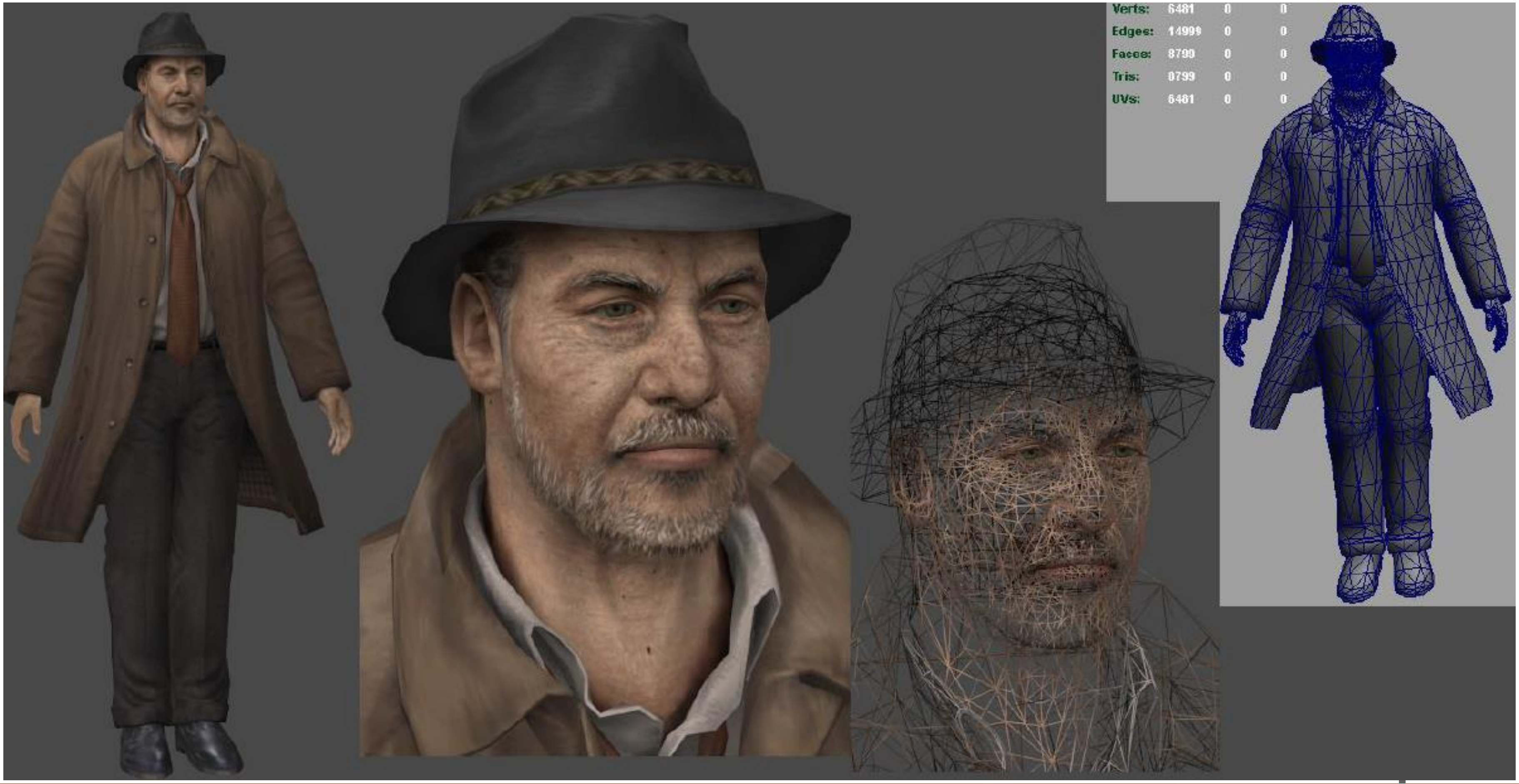
Computer graphics is rooted in the ability to mathematically describe reality.



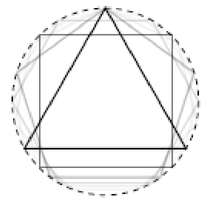
A complex object is represented by simple primitives, like points and lines..



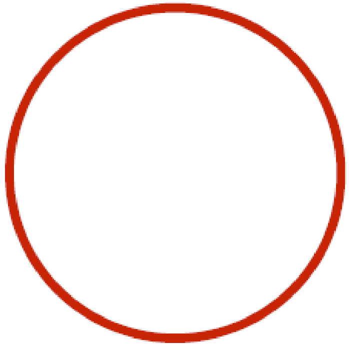

ROUTE
66


Verts: 6481 0 0
Edges: 14999 0 0
Faces: 8799 0 0
Tris: 8799 0 0
UVs: 6481 0 0

CG & Modeling



DISCRETE



IMPLICIT

$$x^2 + y^2 = 0$$

EXPLICIT

$$(\underbrace{\cos\theta}_{x}, \underbrace{\sin\theta}_{y})$$

- Representing shape
- Editing a shape/scene
  - Easy/Intuitive
  - fast
- Visualize a shape/scene
  - Fast

Geometric Primitives

- 0 dimension: points
- 1 dimension: lines/curves
- 2 dimension: mesh/surfaces
- 3 dimension: volumes

Representation:

There are many ways to describe geometry:

- Implicit
- Explicit, Parametric

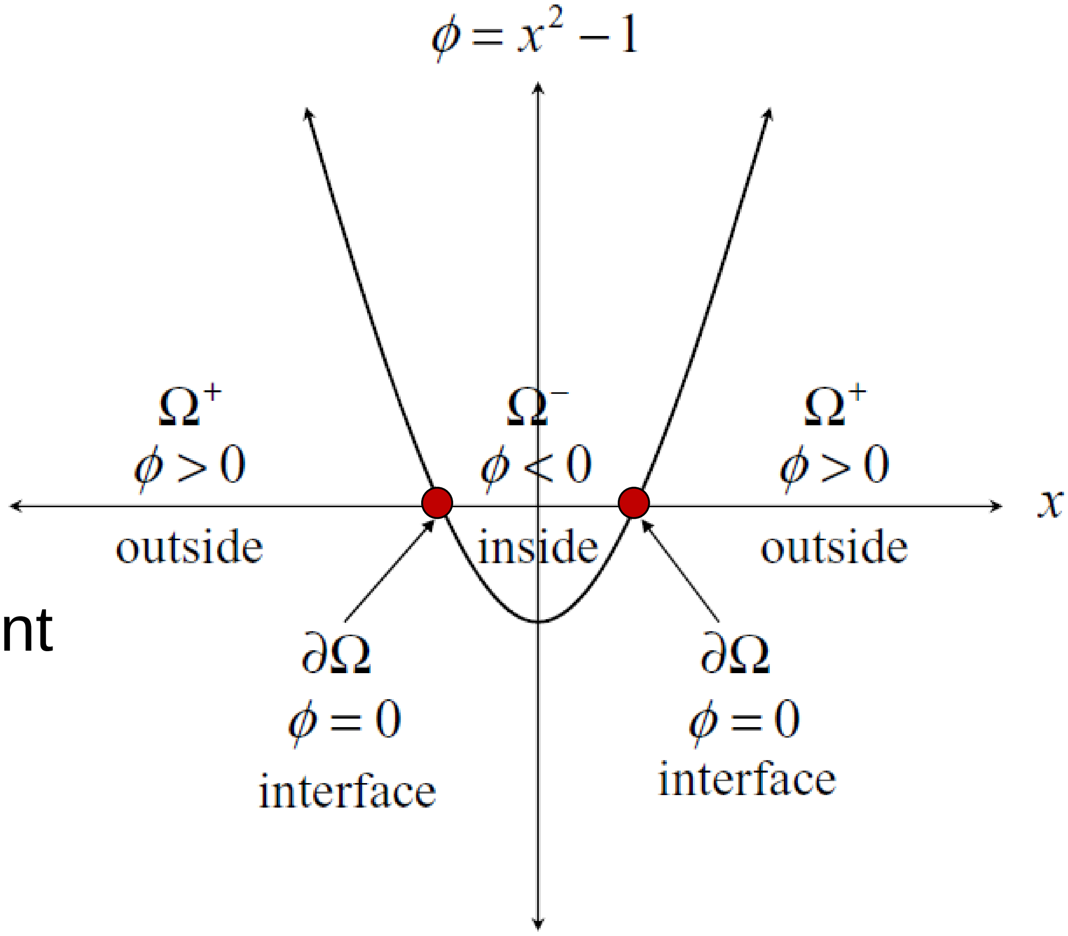
Points: implicit representation

Points x are defined by an implicit function **ø(x) such that**

ø(x) = 0 (isocontour)

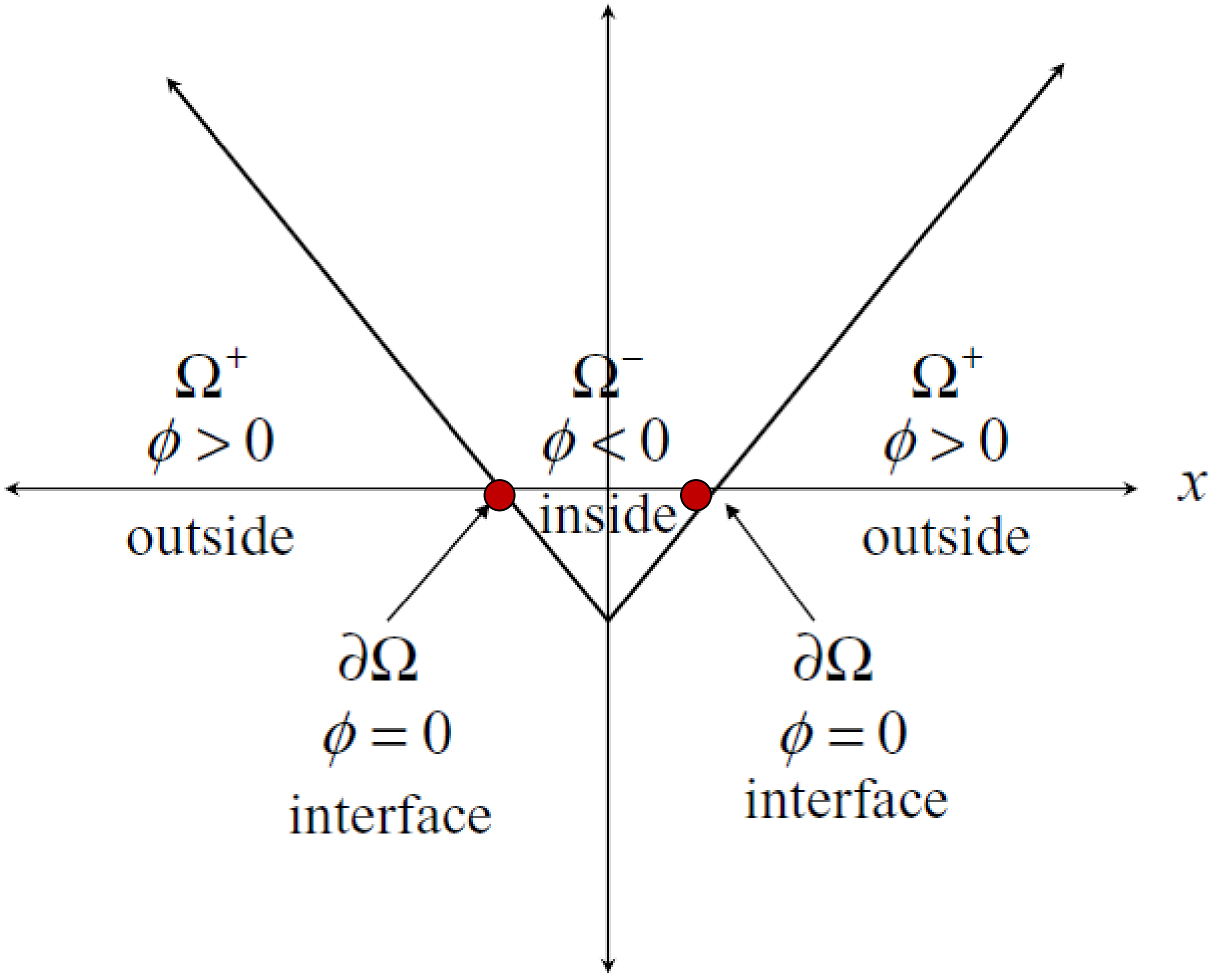
Interface (points)

$\partial\,\Omega$ ={-1,1}



In 1D, the interface is a point which split R into two subdomains (pos. e neg.)

$\phi = |x| - 1$

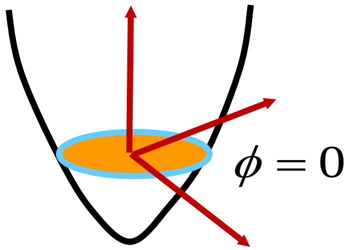


Ω^+
$\phi > 0$
outside
Ω^-
$\phi < 0$
inside
Ω^+
$\phi > 0$
outside
x
$\partial\Omega$
$\phi = 0$
interface
$\partial\Omega$
$\phi = 0$
interface

Curve: implicit representation

- The curve is described by the set of points (x,y) which satisfy ø(x,y) = 0 (isocontour)
- Example:

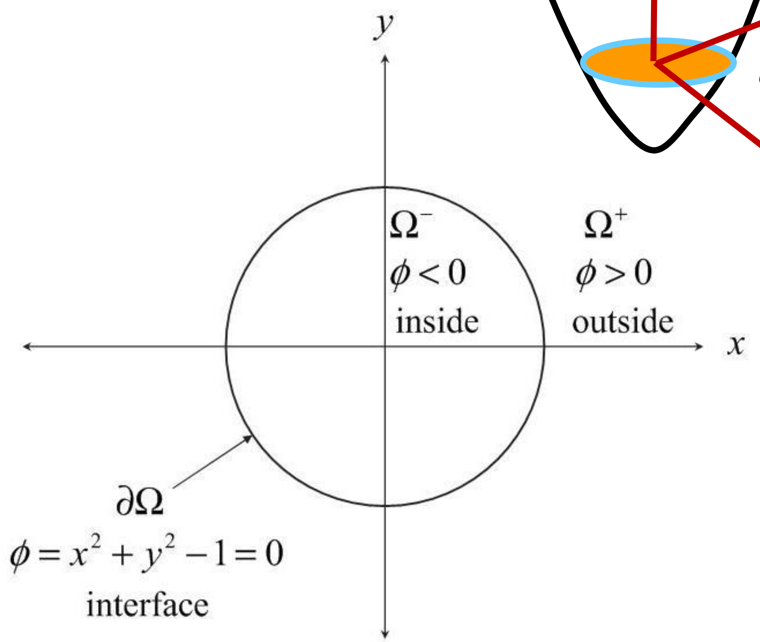
ø(x,y)= $x^2 + y^2 - 1$.

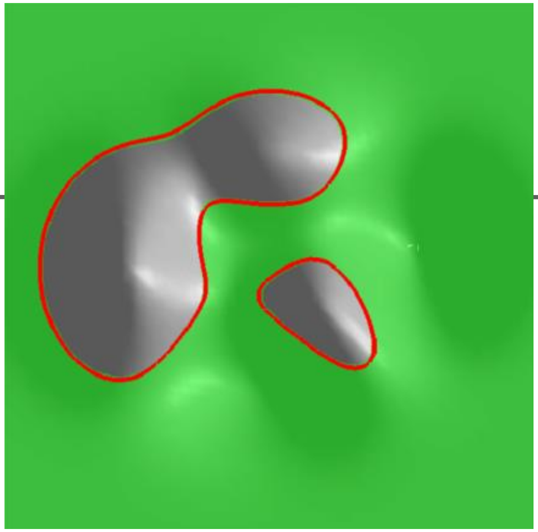


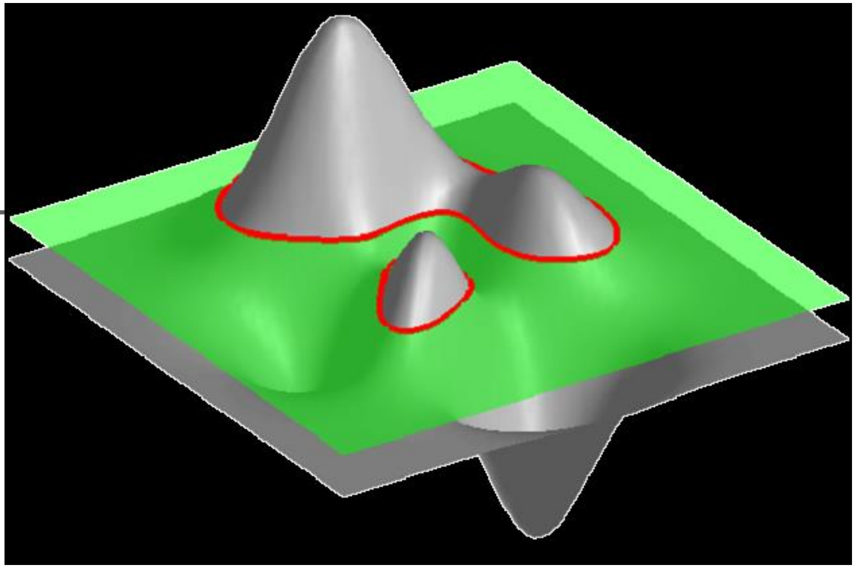
Interface (curve)

∂ Ω ={x/ ||x|| =1}

In 2D, the interface is a curve which separates R^2 into two subdomains.







- Gradient of an implicit function

$$\nabla \phi(x, y) = \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y} \right)$$

- The gradient is perpendicular to the isolevels

- The gradient has the direction of the outward normal N of the interface

$$\vec{N} = \frac{\nabla \phi}{|\nabla \phi|}$$

Explicit Representation

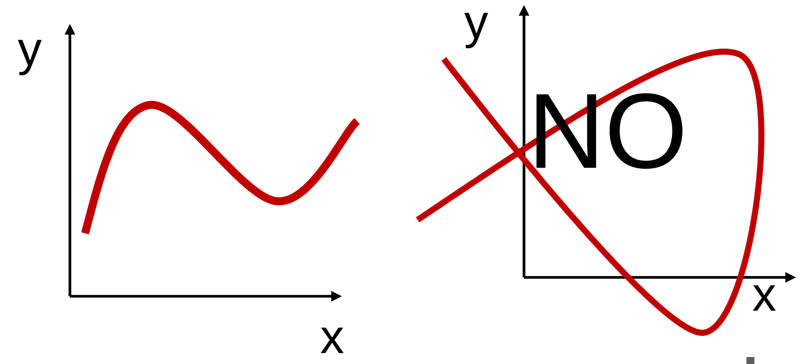
- Point: represented by its 2D/3D coords
- Curve: explicitly define points on it

Curve: defined by moving a point in space with 1 degree of freedom

1) Function Representation

Most familiar form of curve in 2D

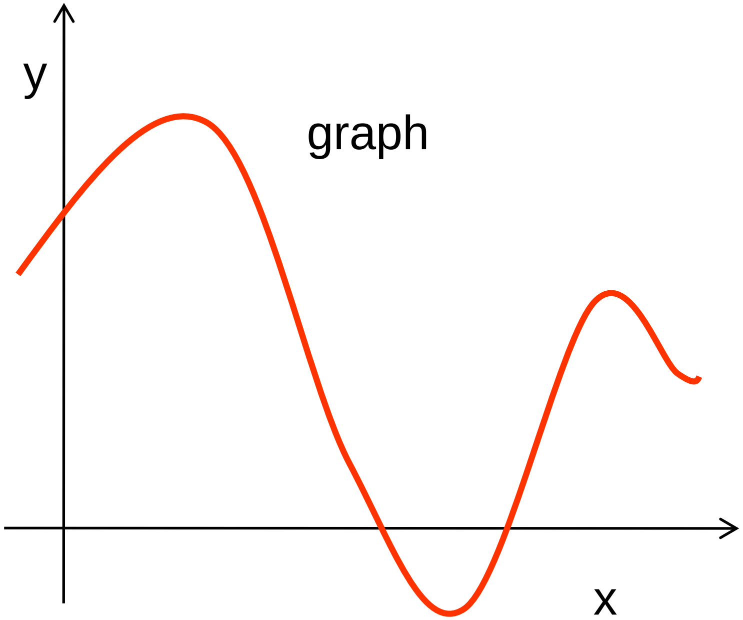
$$y=f(x)$$



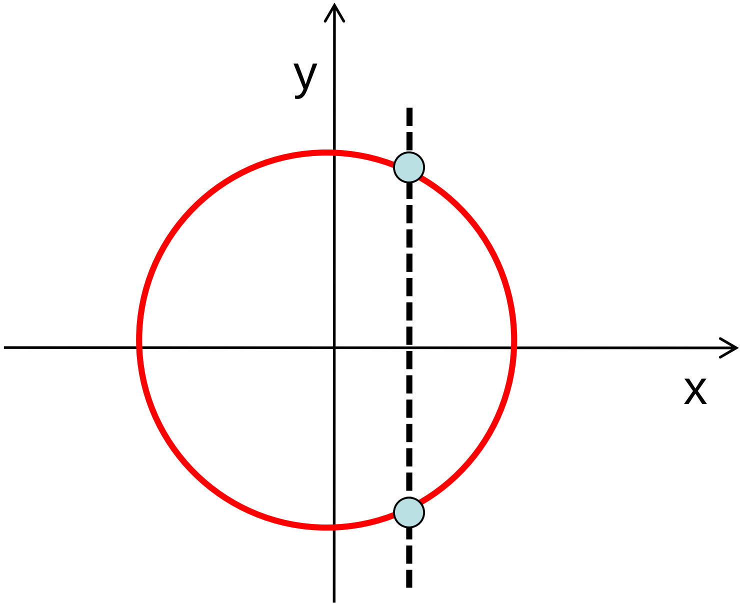
2) Parametric Equation

Curves: function representation

- Cannot represent all curves
- Vertical lines
- Circles

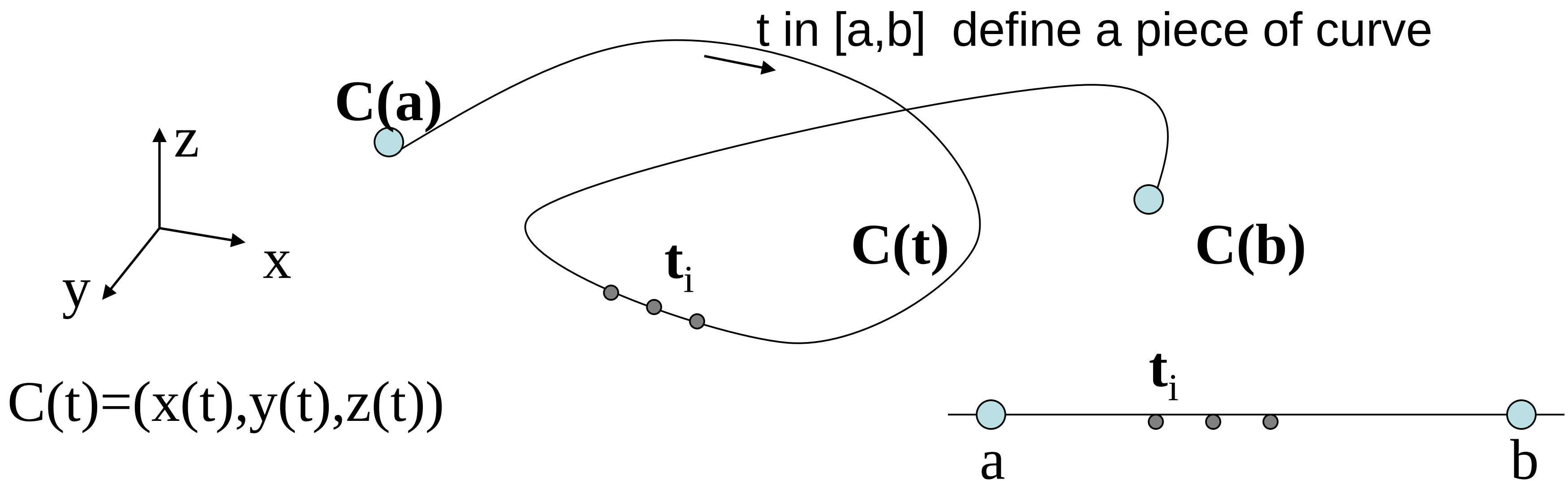


-Single-valued,
-no transformations



Curve: parametric form

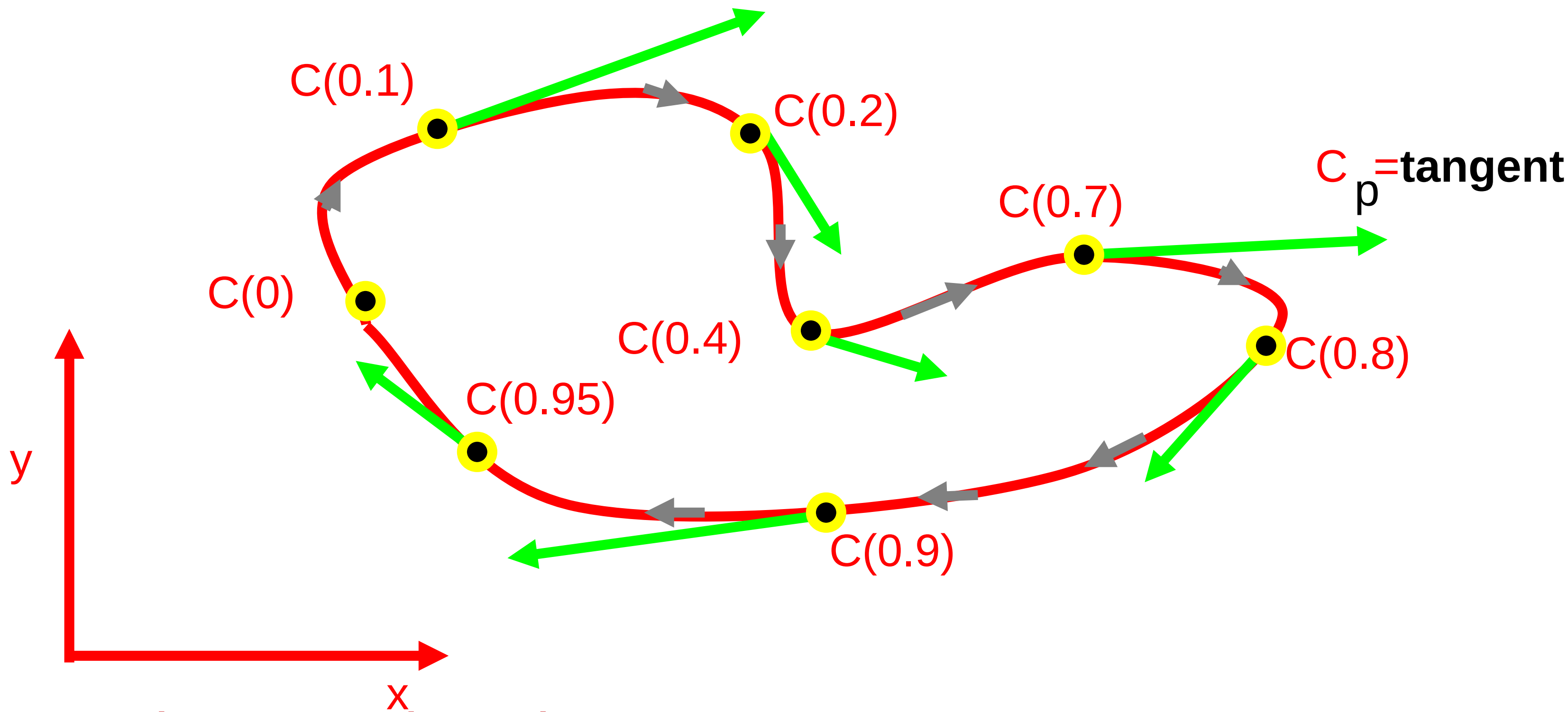
A parametric curve in space is a vector function
C(t)=(x(t),y(t),z(t)) of the parameter $t \in [a,b]$
Upon variation of t, the coordinates (x(t), y(t), z(t)) represent a point that moves on the curve.



In geometric modeling curves and surfaces are described in parametric form

Tangent vector

$C(t)=(x(t),y(t))$, $t \in [0,1]$ parametric domain



point motion speed on the curve :

derivatives at t **$C'(t)=(x'(t),y'(t))$**

Example

- Parametric Curve of degree n (n=3)

$$C(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} t + \begin{pmatrix} -9 \\ -3 \end{pmatrix} t^2 + \begin{pmatrix} 4 \\ 4 \end{pmatrix} t^3$$

- Derivative of a parametric curve (hodograph)

$$C'(t) = \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} + \begin{pmatrix} -18 \\ -6 \end{pmatrix} t + \begin{pmatrix} 12 \\ 12 \end{pmatrix} t^2$$

It ‘s a parametric curve of degree n-1

- Function curves are special cases of parametric curves

$$y = f(x) \quad \Leftrightarrow \quad C(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} t \\ f(t) \end{pmatrix}$$

- From parametric to implicit curves f(x,y)=0

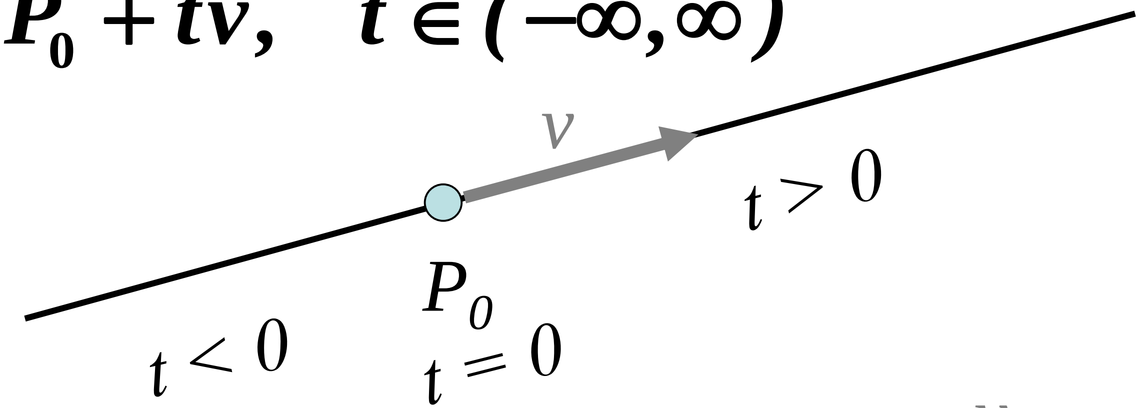
$$C(t) = \begin{pmatrix} x(t) = x0 + t(x1 - x0) \\ y(t) = y0 + t(y1 - y0) \end{pmatrix} = \begin{pmatrix} t = \dfrac{x - x0}{x1 - x0} \\ t = \dfrac{y - y0}{y1 - y0} \end{pmatrix}$$

$$\Rightarrow \frac{x - x0}{x1 - x0} = \frac{y - y0}{y1 - y0}$$

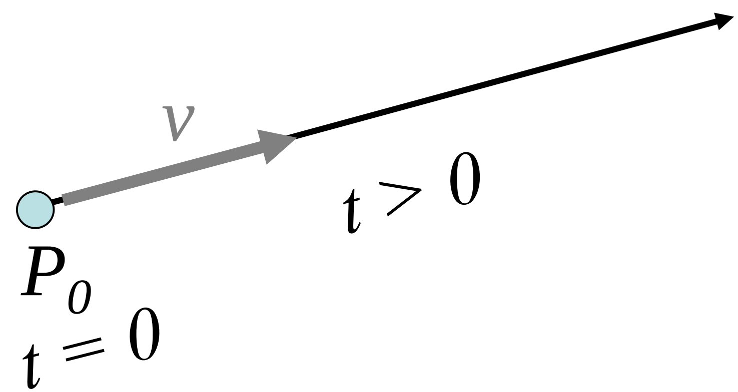
Parametric form

- **Line**

$$C(t) = P_0 + tv, \quad t \in (-\infty, \infty)$$



- **Ray**

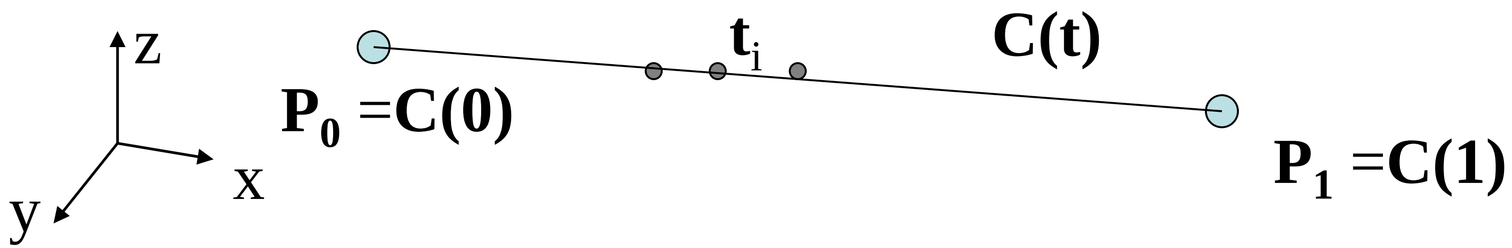


$$C(t) = P_0 + tv, \quad t \in (0, +\infty)$$

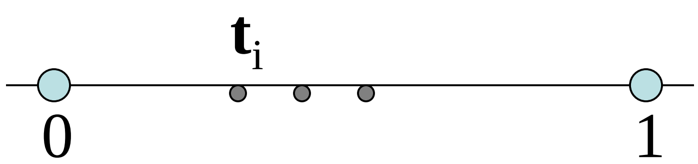
Segment: parametric form

$$C(t)=P_0+t(P_1-P_0),\quad t \text{ in } [0,1]$$

$$x(t)=x_0+t*(x_1-x_0)$$
$$y(t)=y_0+t*(y_1-y_0)$$
$$z(t)=z_0+t*(z_1-z_0)$$



$$x(t)=(1-t)x_0+t*x_1$$
$$y(t)=(1-t)y_0+t*y_1$$
$$z(t)=(1-t)z_0+t*z_1$$

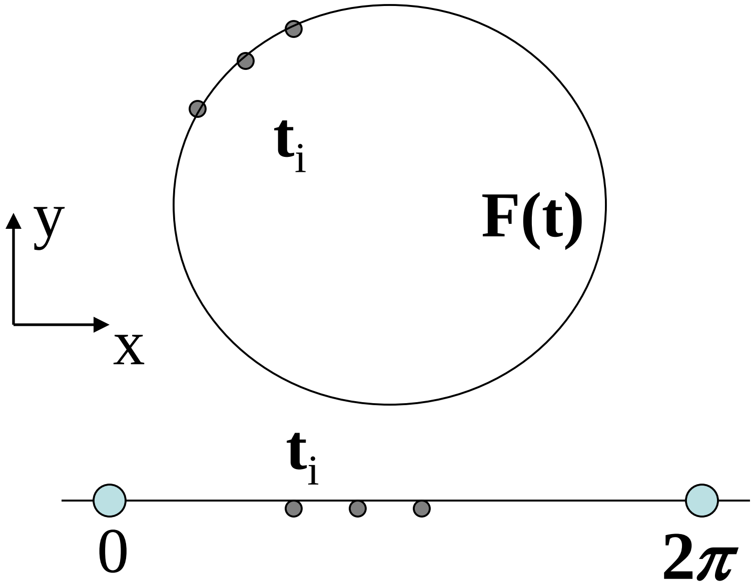


Circle: parametric form

Upon variation of t, the coordinates (x(t),y(t)) represent a point that moves on the circle.

Both parametric forms represent the unit circle:

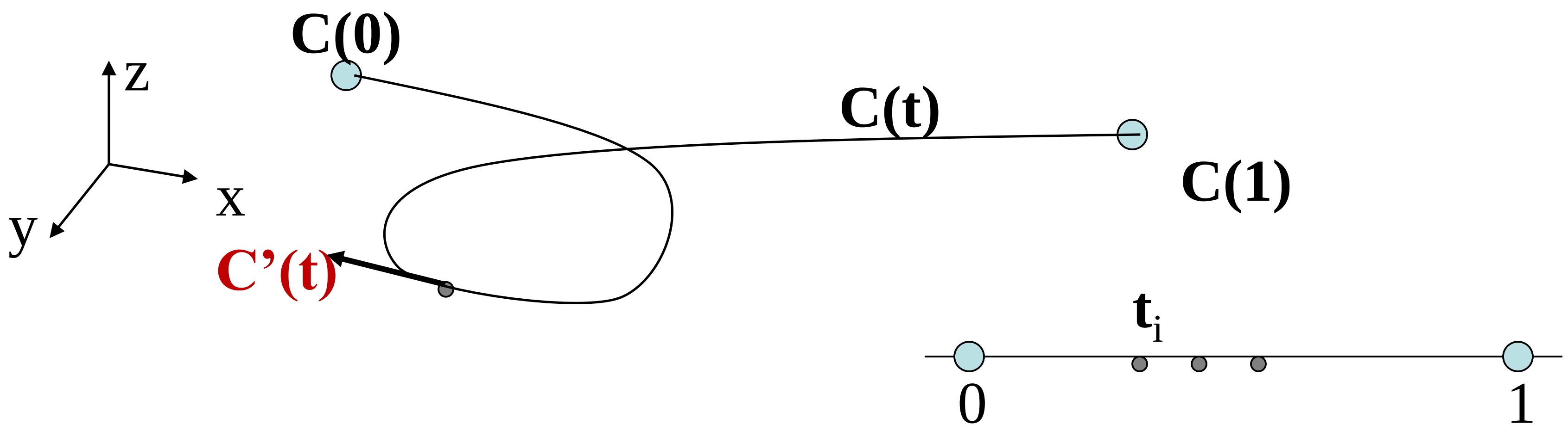
$x(t)=\cos(t)$ **$y(t)=\sin(t)$** **t in $[0,2\pi]$**	**$x(t)= 2t/(1+t^2)$** **$y(t)=(1-t^2)/(1+t^2)$** **$t=[0,1]$**



What, then, do they differ?

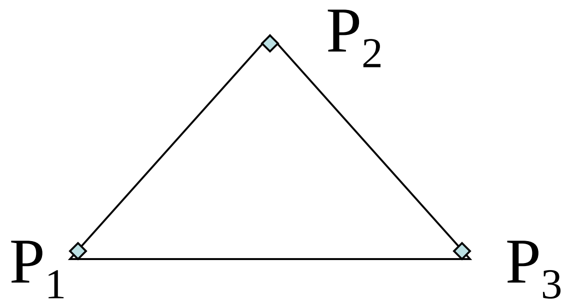
The parameterization, the motion of the point is different even if the path (the curve) is the same.

Physical model: a particle moving in time.
At each instant t, the position of the particle is (x(t), y(t), z(t)); two paths (curves) may be identical even if the velocity (parameterization) is different.

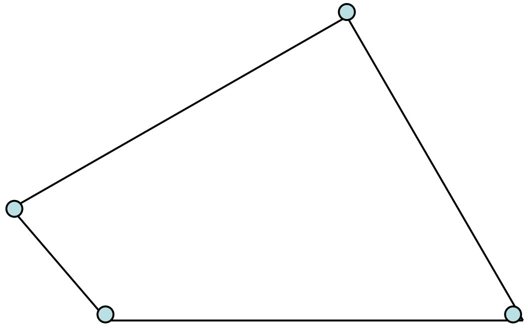


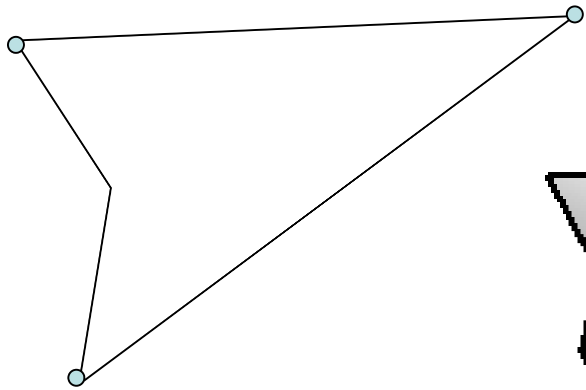
3D Geometric Primitives

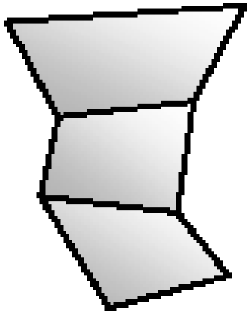
- 2D/3D polygons

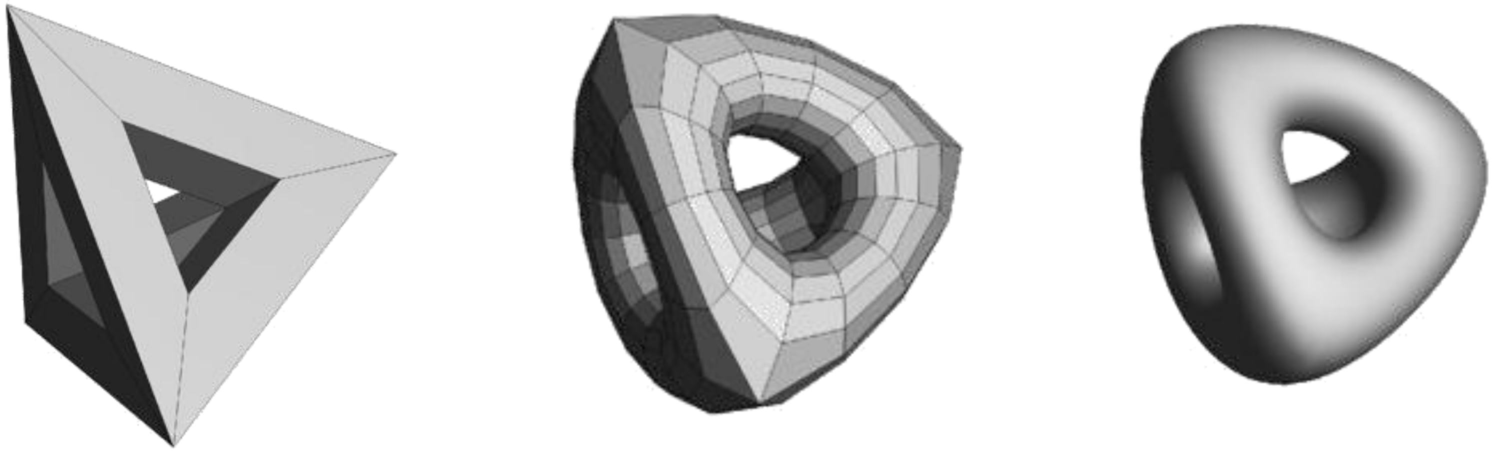


In 3D: poly must be planar!









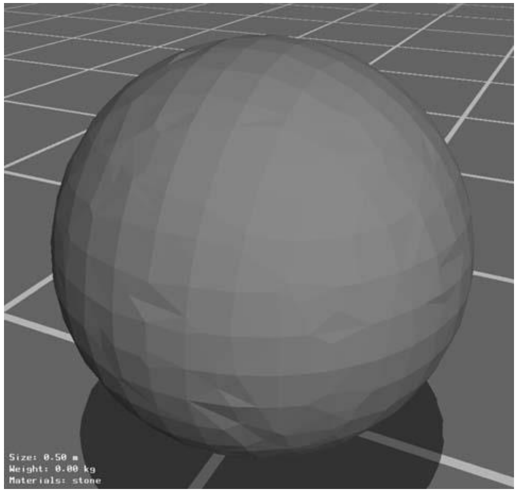
Polygonal mesh

Set of **edges**, **vertices** and **faces** connected in such a way that:

each edge is shared by at most two adjacent faces,
one side connects two vertices,
the faces are sequences of closed sides,
a vertex is shared by at least two sides.

Limits of Polygonal Meshes

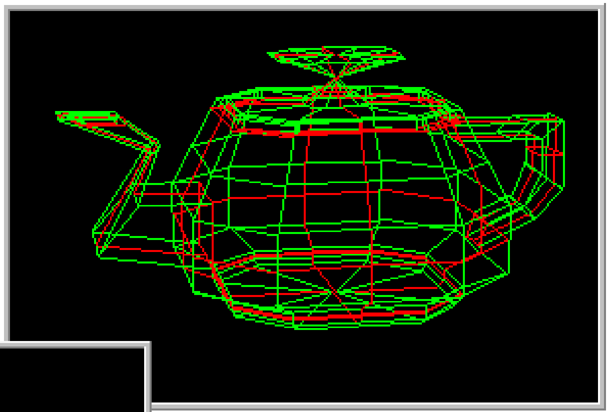
- flat facets
- fixed resolution
- difficult editing
- no natural parameterization



Smooth Surfaces

- Explicit / Implicit
- Naturally curved
- Closed / open
- Defined by control points /curves / interpolation / approximation
- planes

- Bézier Patch
- Spline Surfaces/NURBS



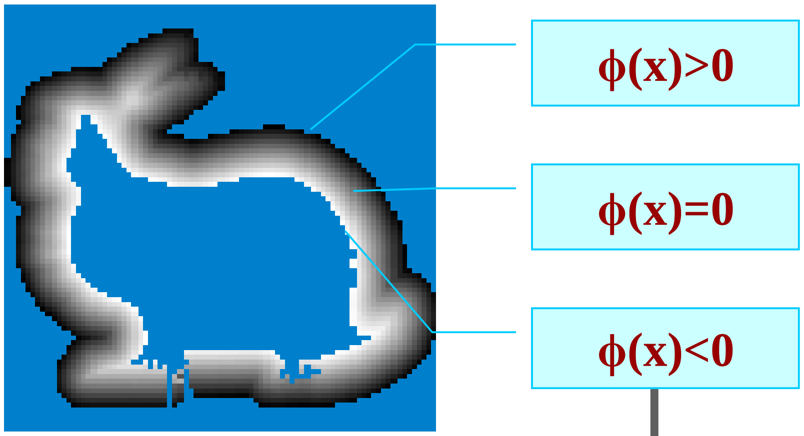
Surface: implicit representation

- Implicit surfaces represent a surface as a particular isocontour of a higher dimensional embedding function on $\Re^3$ $\phi(x,y,z)=0$

  E.g., unit sphere is all points x such that $x^2+y^2+z^2=1$

- The inside region Ω−,the outside region is Ω+, and the surface ∂Ω are all defined by the function:

$\phi(x,y,z) < 0$	inside
$\phi(x,y,z) > 0$	outside
$\phi(x,y,z) = 0$	on



Types of Implicit Surfaces

– Polynomial or *Algebraic*

– Non polynomial or *Transcendental*

- Exponential, trigonometric, etc.

– Computational

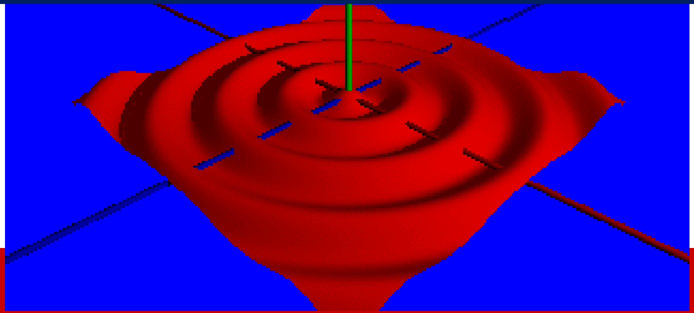
- Interpolation:

Generate surfaces that interpolate boundary points

- PDEs

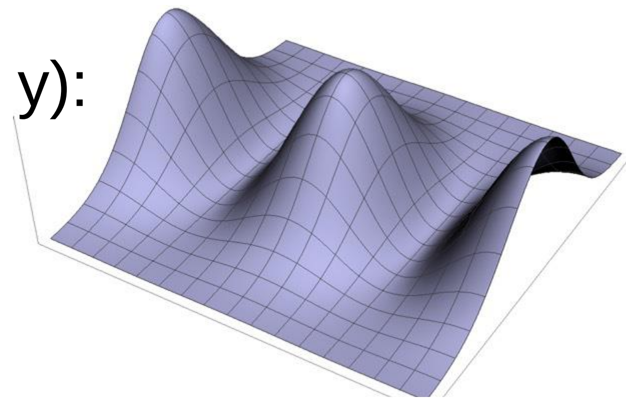
$$(2x^2 + y^2 + z^2 - 1)^3 - (0.1x^2 + y^2)z^3 = 0$$

$$|\nabla d(x)| = 1, \quad d(x) = 0, \quad x \in S$$



Surfaces: Explicit representation

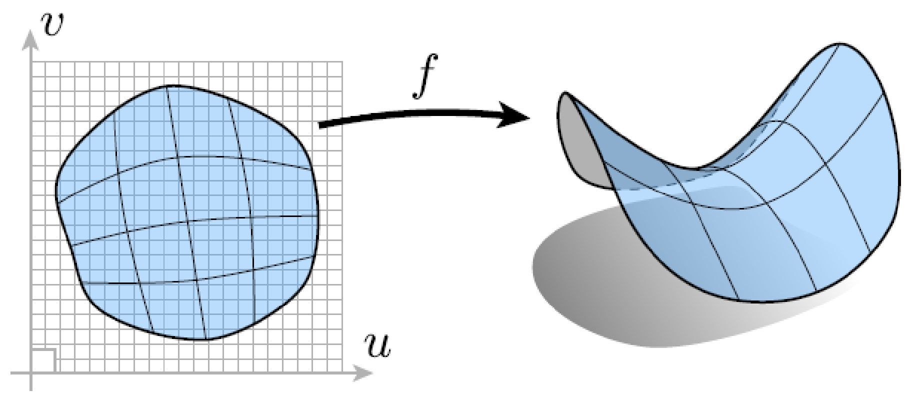
a) function form $z = f(x, y)$



z is the high of the point on the plane (x, y):

- single-valued
- No vertical tangent planes
- No changes

b) parametric form



not have these problems,

- Extend parametric curves
- Parametric variables *u* and v
- Isoparametric curves   f(u,v) and f(u,v),   for u,v fixed

$$f : \mathbb{R}^2 \to \mathbb{R}^3; \quad (u,v) \to \big(x(u,v), \quad y(u,v), \quad z(u,v)\big)$$

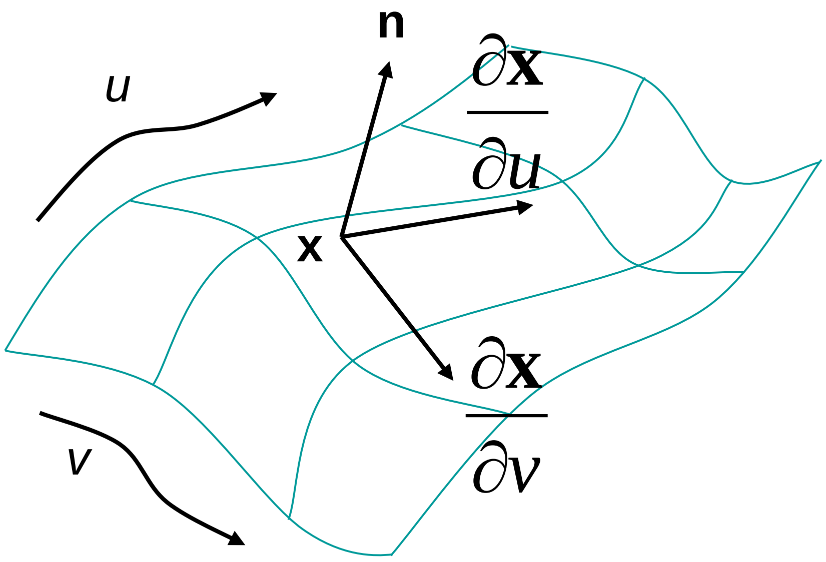
Parametric Surface

- **Derivatives**

  The two tangent vectors define a tangent plane at (u,v)
  To calculate the normal to the surface at a point (u,v), we compute the two tangents in this point and evaluate their cross product (normalized)

$$\mathbf{n}^* = \frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v}$$

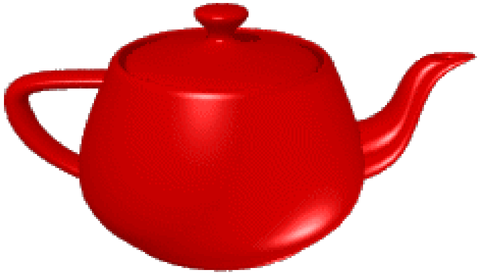
$$\mathbf{n} = \frac{\mathbf{n}^*}{\left|\mathbf{n}^*\right|}$$



Useful to understand what is the outer side of a face

Continuity

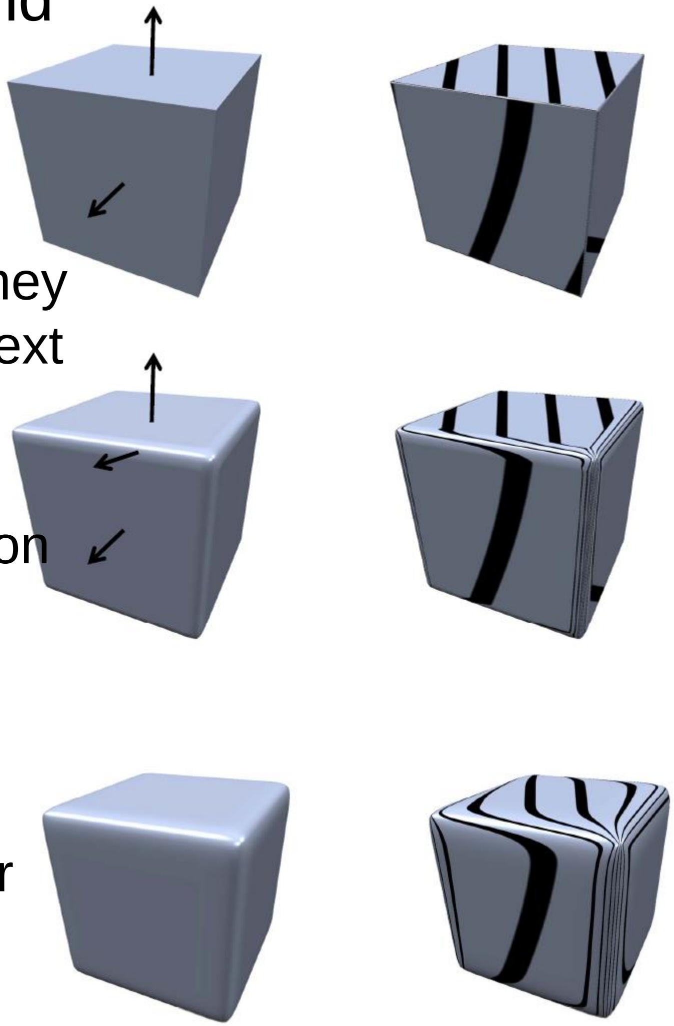
- C^0 curves/surfaces
  – without holes
  – "watertight"
- C^1 curves/surfaces
  – with continuous derivatives
  – “smooth, no faced"
- C^2 curves/surfaces
  – with continuous second derivatives
  – Important for shading and CAD/CAM



Visual surface analysis

Visually evaluates surface smoothness and continuity using a stripe map

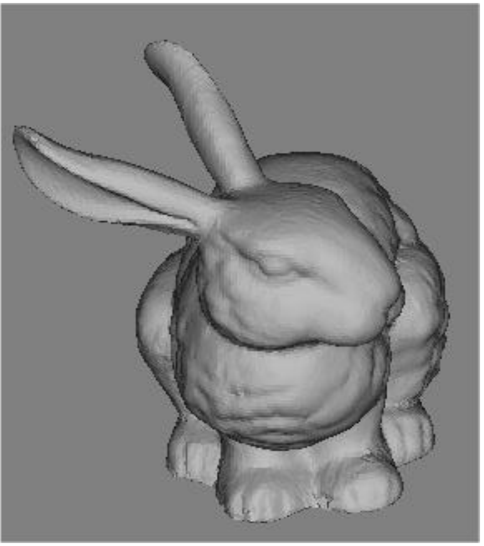
- Position continuity $G^0 = C^0$
  - If the stripes have kinks or jump sideways as they cross the connection from one surface to the next
- Tangent continuity G^1
  - If the stripes line up as they cross the connection but turn sharply at the connection, the position and tangency between the surfaces match
- Curvature continuity G^2
  - If the stripes match and continue smoothly over the connection, this means that the position, tangency, and curvature between the surfaces match.

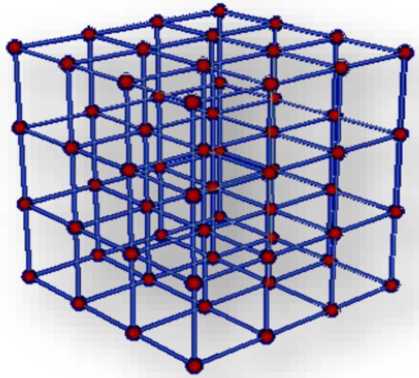


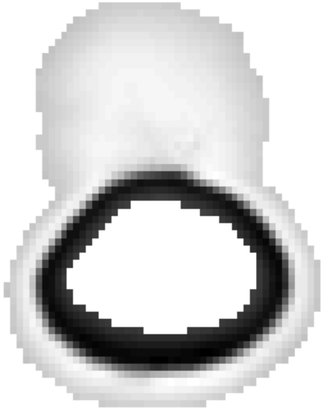
Surfaces: Explicit vs. Implicit form

- **Evaluation**:
  - (I) Grid vs. (E) explicit evaluation
- **Classification** of points as inside/outside to a given interface:
  - (I) check the sign of $\phi(x)$
  - (E) no easy for explicit form
- **Boolean Operation**
  - (I) easy
  - (E) no easy for explicit form
- **Editing**

Implicit : Evaluation







Implicit : Classification

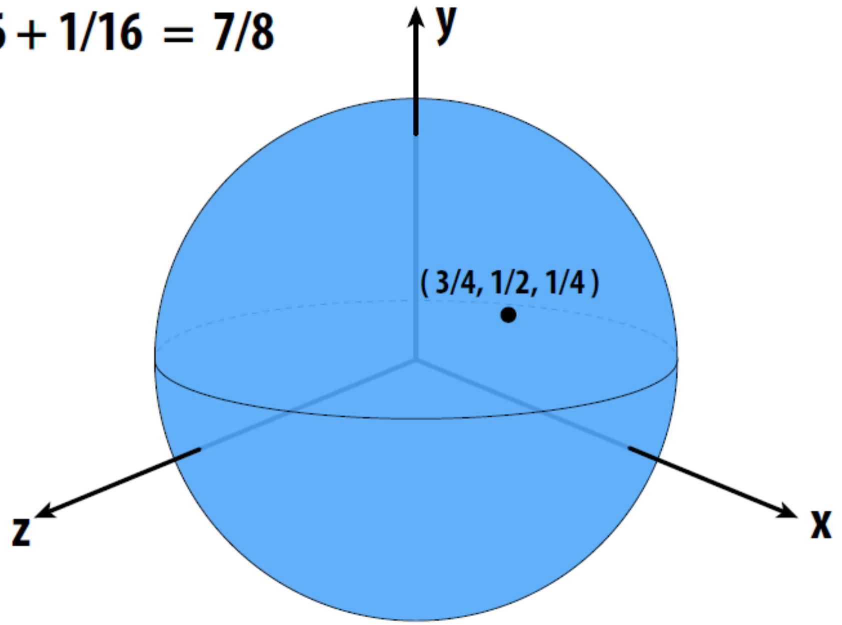
Check if this point is inside the unit sphere

How about the point (3/4, 1/2, 1/4)?

$9/16 + 4/16 + 1/16 = 7/8$

$7/8 < 1$

YES.

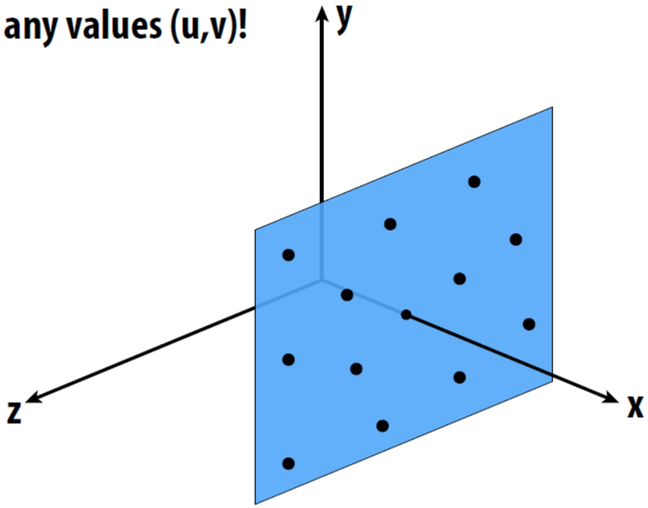


Implicit surfaces make other tasks easy (like inside/outside tests).

Sampling an explicit surface

My surface is f(u, v) = (1.23, u, v).

Just plug in any values (u,v)!



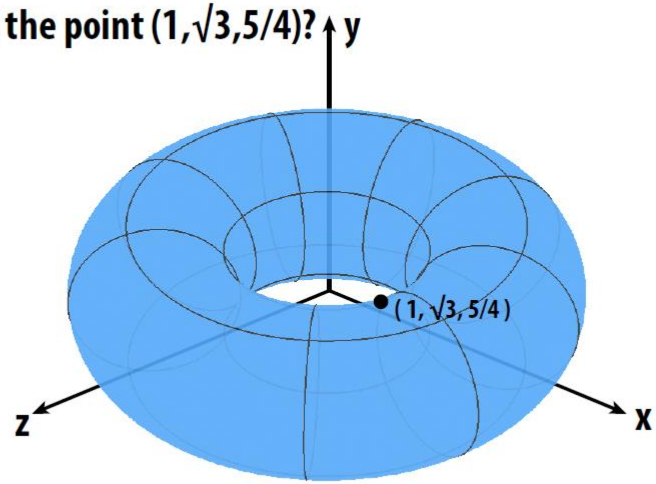
Explicit surfaces make some tasks easy (like sampling).

Explicit

Check if this point is inside the torus

My surface is f(u,v) = (2+cos(u))cos(v), 2+cos(u))sin(v), sin(u))

How about the point (1,√3,5/4)?



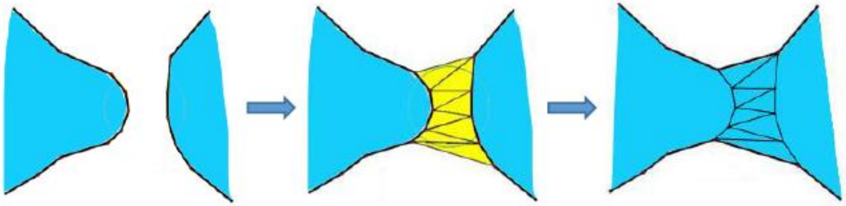
...NO!

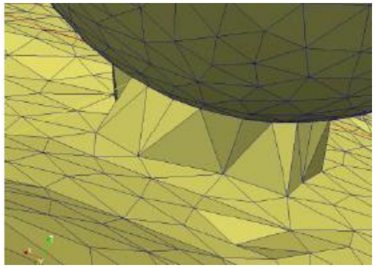
Explicit surfaces make other tasks hard (like inside/outside tests).

Detect **changes in topology**

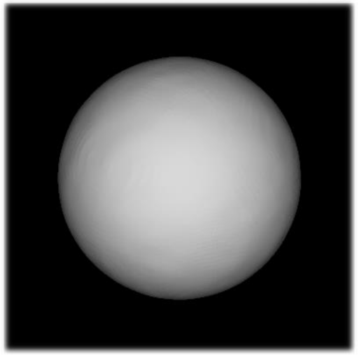
Implicit surfaces are good for handling complicated surfaces like water

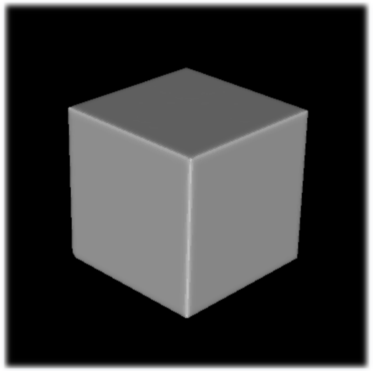
whereas a triangle representation has issues with editing, merging and pinching, overturning waves, etc.





Implicit : Boolean Operations



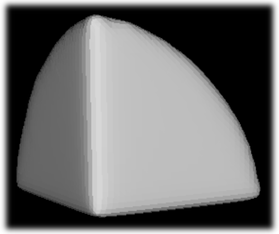


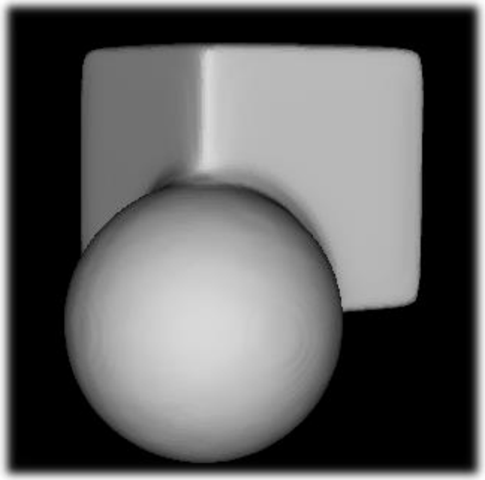
UNION $A \cup B$ $\min(\phi_A(x), \phi_B(x))$

INTERSECTION $A \cap B$ $\max(\phi_A(x), \phi_B(x))$

DIFFERENCE $A - B$ $\max(\phi_A(x), -\phi_B(x))$

Create complex objects using boolean operators on simple objects





Inter sec tion *Difference* $A - B$ *Union*

what’s the best way to encode geometry on a computer?

Some representations work better than others—depends on the task!

Useful Resources

There are many modeling programs that are designed to help you create and modify models

- Blender(free)
- SketchUp(free)
- Meshlab(free)

Other modeling programs include Autodesk 3DSMax, Maya, Rhinoceros, AutoCAD, Lightwave, Bryce, Hexagon, etc.

Model Database

- Aim@Shape
- Archive3D (everyday objects, e.g., desks, chairs, sofa, etc)
- GrabCAD (mechanical objects, e.g., robots, planes, cars, warships, etc)
- TurboSquid (largest model database in the world, but only part of them are free)
- 3DWarehouse (architecture, e.g. buildings, bridges, furniture, etc.)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Serena Morigi

Dipartimento di Matematica

serena.morigi@unibo.it

http://www.dm.unibo.it/~morigi