Computer Animation and Games I

CM50244

Shape Representations

Animation Production

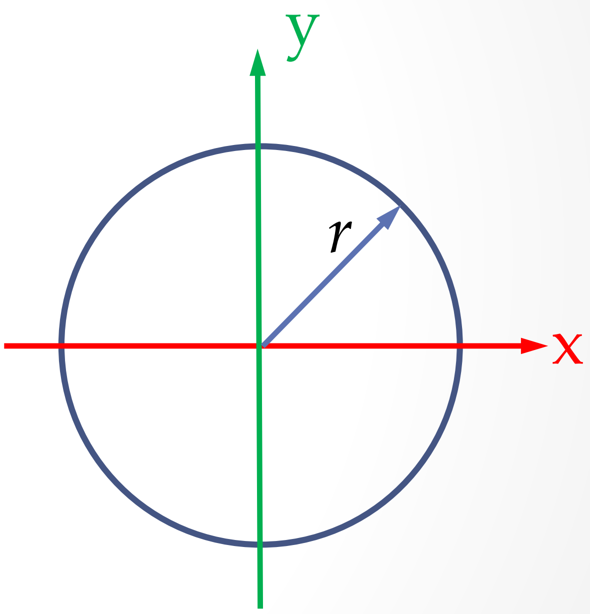
1. Story Board
2. Conceptual Art
3. Recording
4. Modeling
5. Rigging
6. Layout
7. Animation
8. Special Effects
9. Shading
10. Lighting
11. Rendering

Overview

- **2D Curve Representations**
- Bezier Curves
- 3D Surface Representations

2D Curve Representation

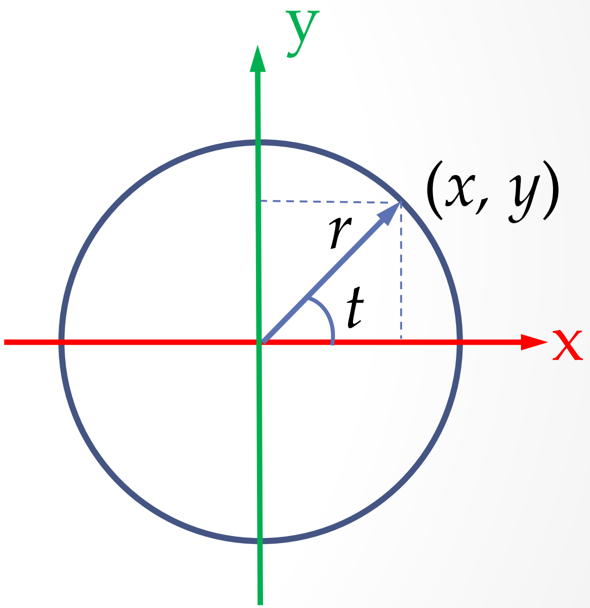
- **Parametric** representation
- **Implicit** representation
- **Explicit** representation



2D Curve Representation

- **Parametric** representation $x = x(t), \;\; y = y(t)$
  - the x and y coordinates of a curve point are functions of a single parameter *t*

$$x = r\cos t$$
$$y = r\sin t$$

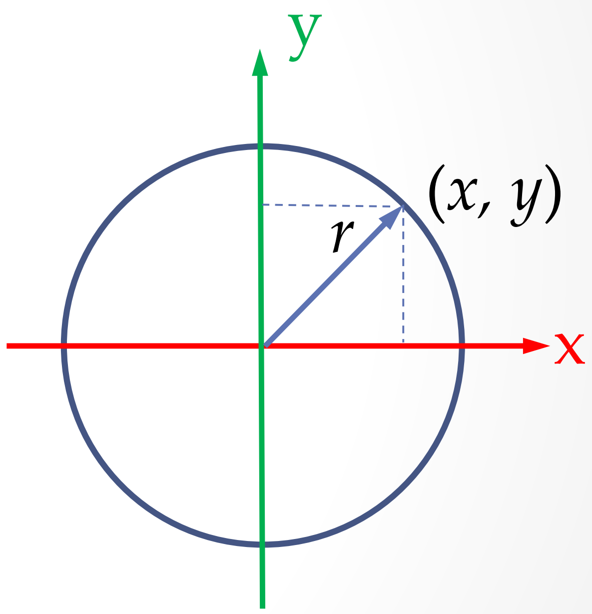


the parameter *t* provides an easy way to iterate all the points on the curve

2D Curve Representation

- **Implicit** representation $f(x, y) = 0$
  - for any curve point, x and y coordinates should satisfy a single equation

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



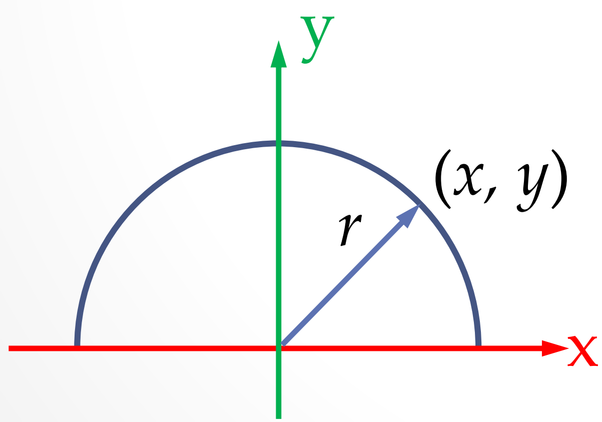
easily check if a point is on/inside/outside curve
by evaluating the function value

2D Curve Representation

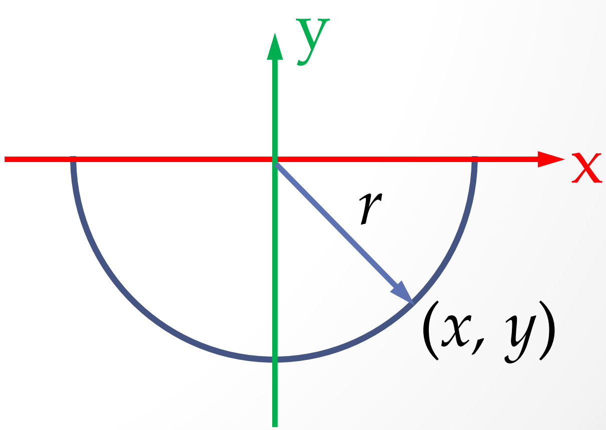
- **Explicit** representation $y = f(x)$
  - y coordinate is explicitly represented as a function of x coordinate

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

$$y = \sqrt{r^2 - x^2}$$



$$y = -\sqrt{r^2 - x^2}$$



simple since y coord. is explicitly represented by x coord.
but not convenient for multi-valued curve function

Which Rep. to Use?

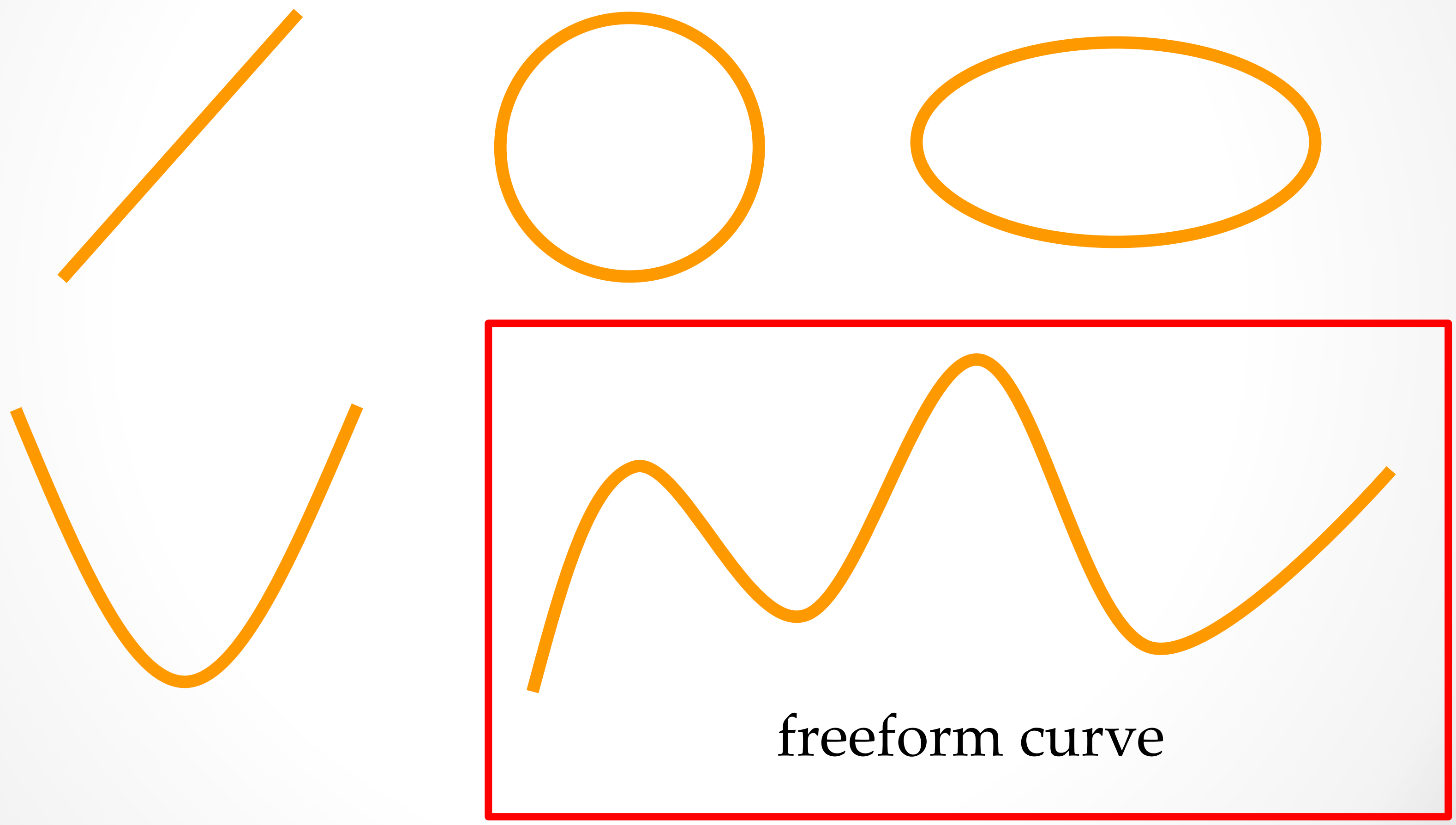
Rep.	Parametric	Implicit	Explicit
Math. Form	$x = x(t), \quad y = y(t)$	$f(x, y) = 0$	$y = f(x)$
Circle	$x = r\cos t$ $y = r\sin t$	$x^2 + y^2 - r^2 = 0$	$y = \sqrt{r^2 - x^2}$ $y = -\sqrt{r^2 - x^2}$

- **Parametric** rep. allows easy iteration along the path of a curve
- **Implicit** rep. allows quick test if a point is on/inside/outside a curve
- **Explicit** rep. is simple, but not convenient for multi-valued curve function

Overview

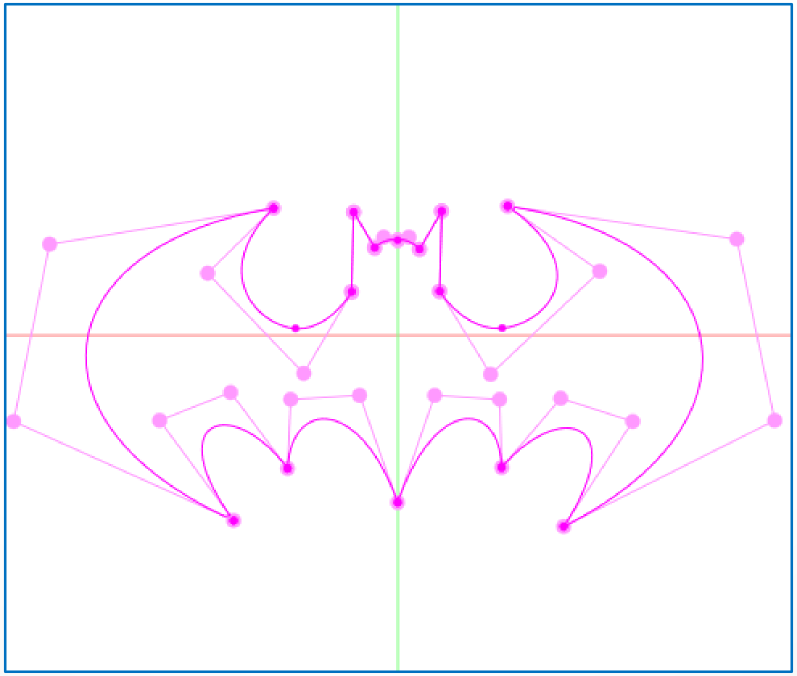
- 2D Curve Representations
- **Bezier Curves**
- 3D Surface Representations

2D Curve Examples



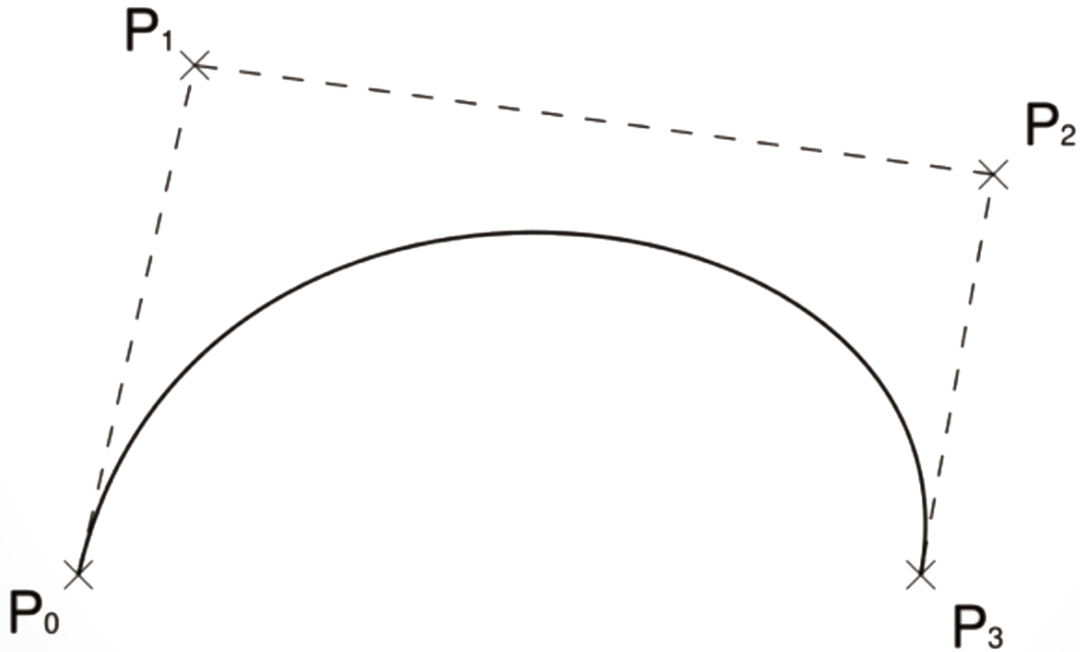
Freeform Curve Modeling

- Parametric curves are usually used
- Several standard parametric curves, e.g. Bézier
- Basic idea: using a few **control points** to Construct freeform curve from scratch
- Easy to deform



Bézier Curves

- A Bézier curve is specified using a set of control points



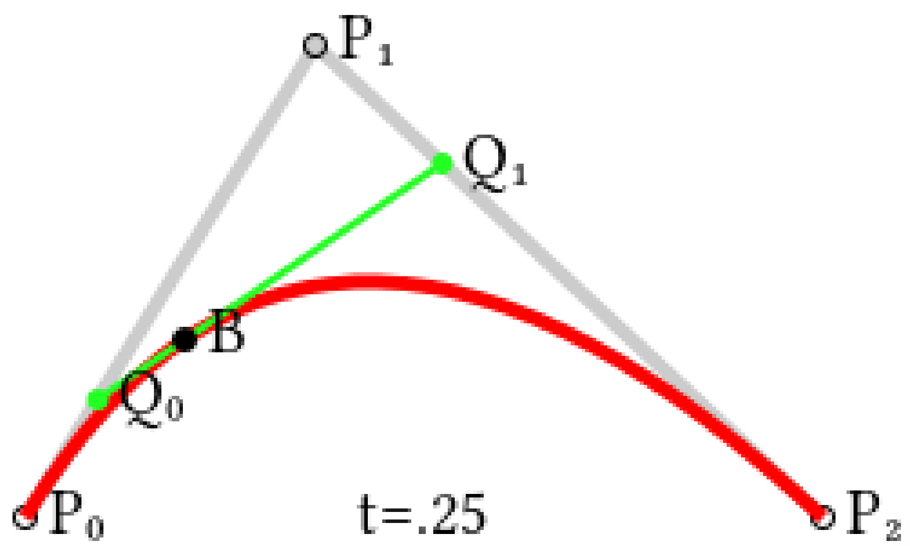
History of Bézier Curves

- The mathematical form has been known since 1912
- In 1962, a French engineer Pierre Bézier used them to design automobile bodies at Renault

Bézier Curve Demo

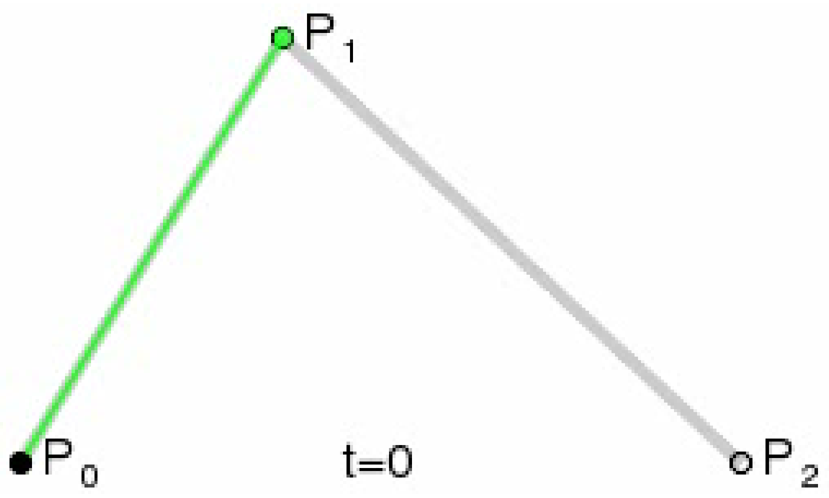
Bézier Curves

- **Recursive linear interpolation** (multi-linear construction) of line segments based on the control polygon



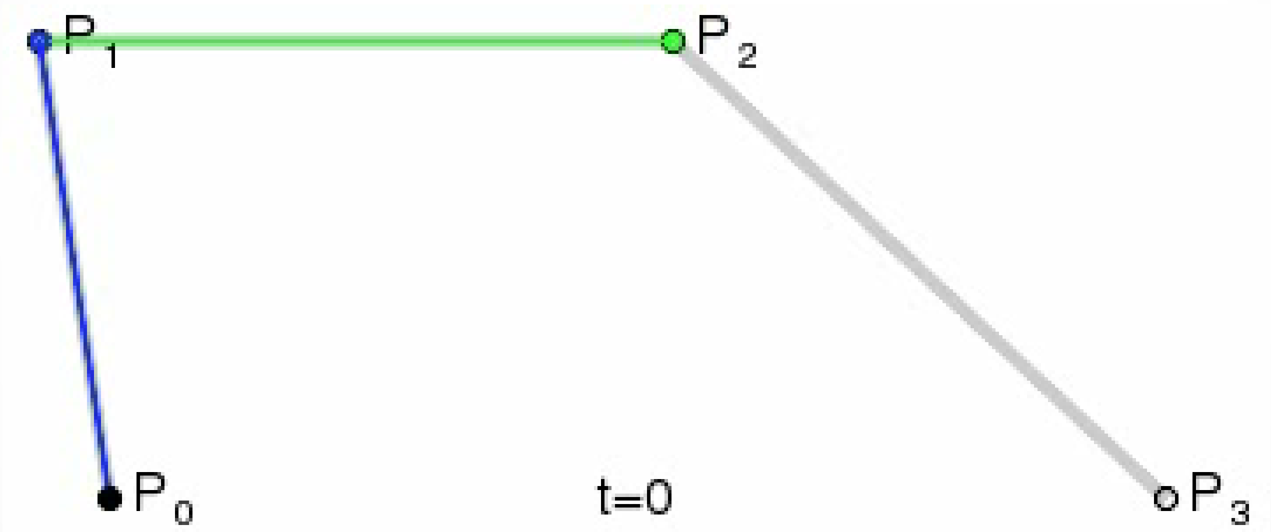
Quadratic Bézier Curves

- Based on the multi-linear construction, if t goes from 0 to 1, we can draw the whole Bézier curve.



Cubic Bézier Curves

- This recursive process can be continue to generate higher order curves, e.g., 3rd order



Algebraic Form of Bézier Curves

- Given n+1 control points $\mathbf{P}_i,\ (i=0,\ldots,n)$, Bézier curve can be defined as:

$$\mathbf{C}(t)=\sum_{i=0}^{n}\mathbf{P}_i B_{i,n}(t),\quad t\in[0,1]$$

- $B_{i,n}(t)$: *n*-th order Bernstein basis function

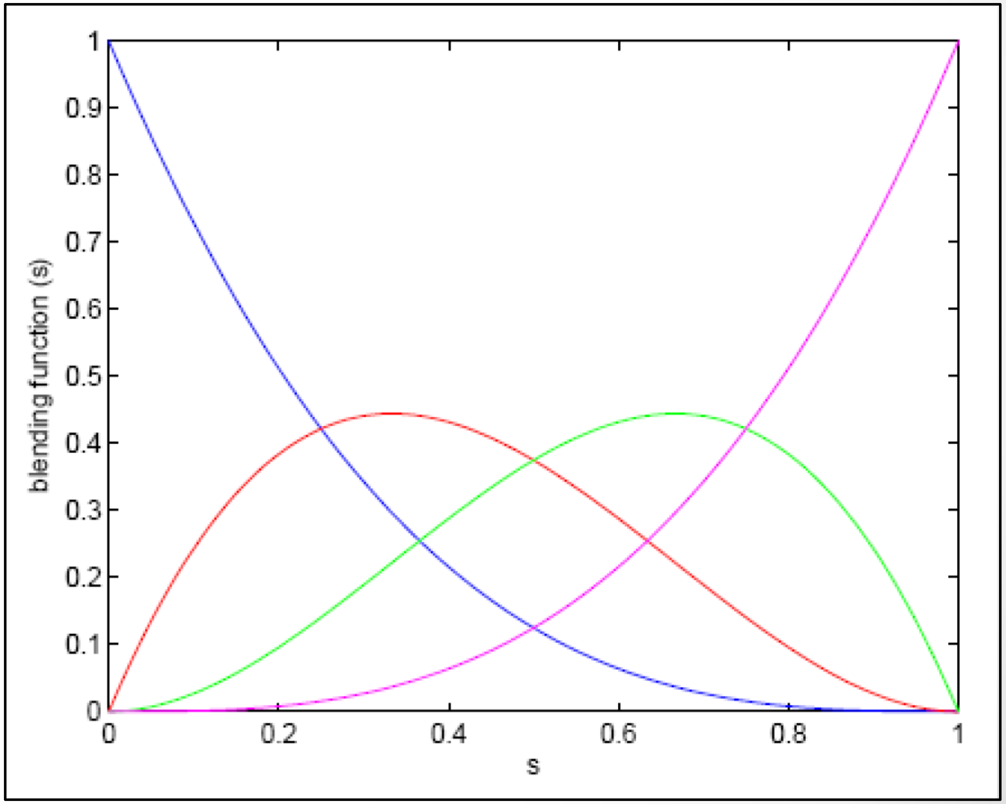
$$B_{i,n}(t)=C_n^i t^i(1-t)^{n-i}=\frac{n!}{i!(n-i)!}t^i\cdot(1-t)^{n-i}\quad (i=0,1,\cdots,n)$$

This form can be derived from multi-linear construction

Cubic Bézier Curves

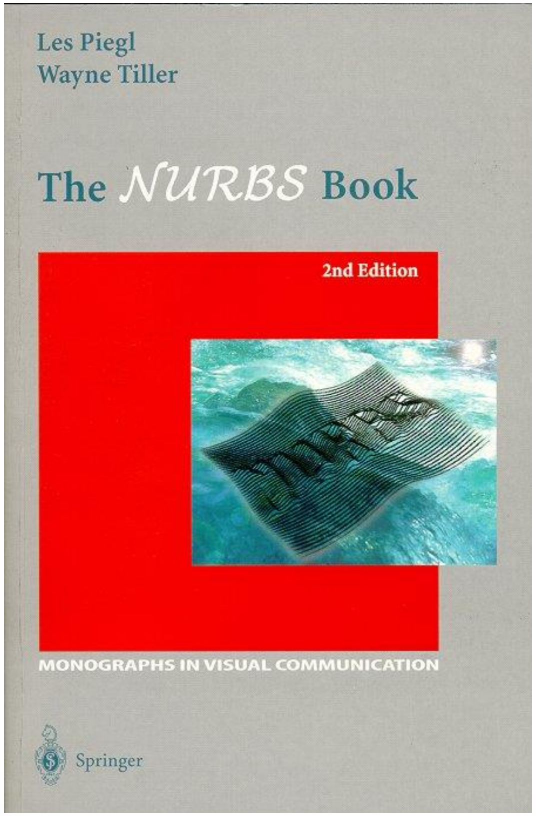
- When n=3, it is a weighted sum of **4 basis functions** (weights on the 4 control points)

$$\mathbf{p}(t) = (1\text{-}t)^3\,\mathbf{p}_0 + 3t(1\text{-}t)^2\,\mathbf{p}_1 + 3t^2(1\text{-}t)\,\mathbf{p}_2 + t^3\,\mathbf{p}_3$$



Other Parametric Curves

- B-Spline (**B**asis spline) Curves
- NURBS(**N**on-**U**niform **R**ational **B**-Spline) Curves

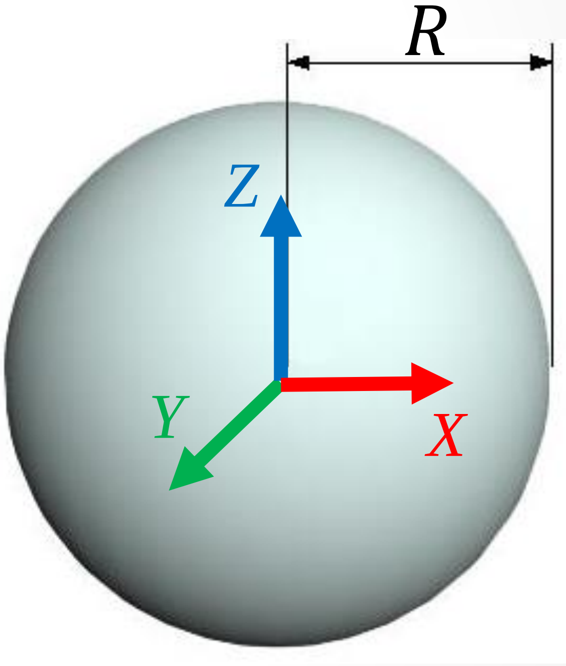


Overview

- 2D Curve Representations
- Bezier Curves
- **3D Surface Representations**

3D Surface Representation

- **Parametric** representation
- **Implicit** representation
- **Explicit** representation

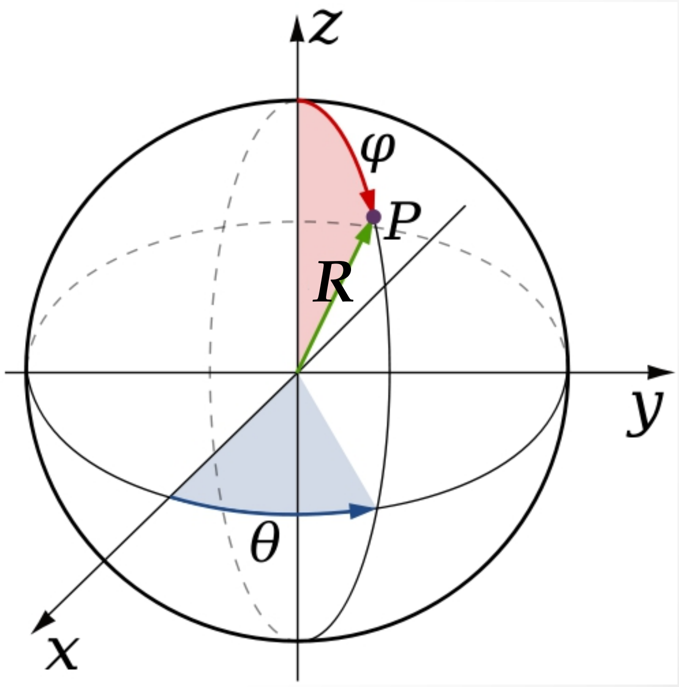


Surface Representation

- **Parametric** representation $x = x(u, v), y = y(u, v), z = z(u, v)$
  - the x, y, z coordinates of a surface point are functions of two parameters *u and v*

$$\mathbf{f} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad \mathcal{S}_\Omega = \mathbf{f}(\Omega)$$

$$x(\theta, \varphi) = R sin\varphi cos\theta$$
$$y(\theta, \varphi) = R sin\varphi sin\theta$$
$$z(\theta, \varphi) = R cos\varphi$$

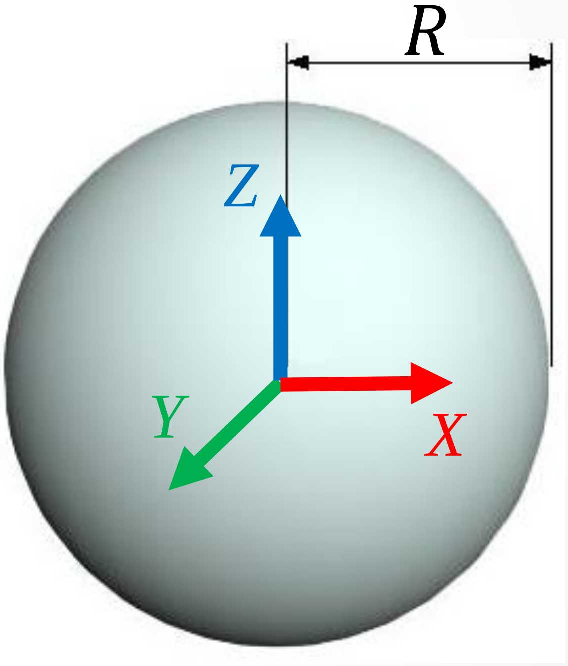


parameter (u, v) provide an easy way to iterate points on the surface

Surface Representation

- **Implicit** representation $f(x, y, z) = 0$
  - For any surface point, x, y, z coordinates should satisfy a single equation

$$x^2 + y^2 + z^2 - R^2 = 0$$



easily check if a point is on/inside/outside the surface by evaluating the function value

Surface Representation

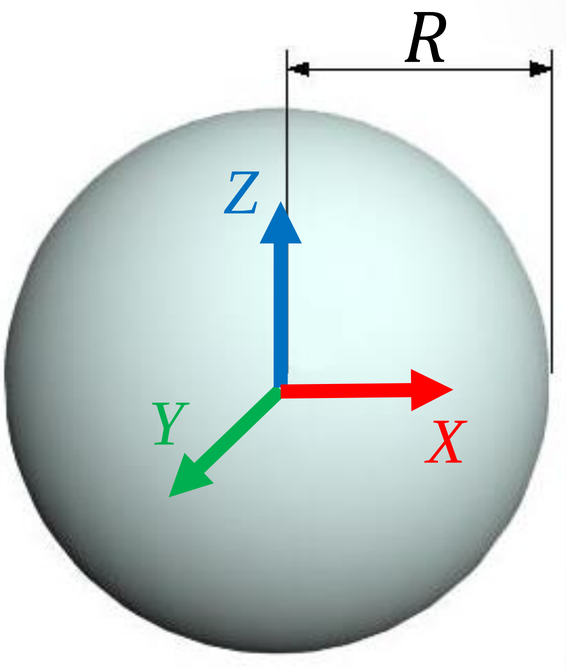
- **Explicit** representation $z = f(x, y)$
  - z coordinate is explicitly represented as a function of x and y coordinates

$$\boxed{x^2 + y^2 + z^2 - R^2 = 0}$$

$$\downarrow$$

$$z = \sqrt{R^2 - x^2 - y^2}$$

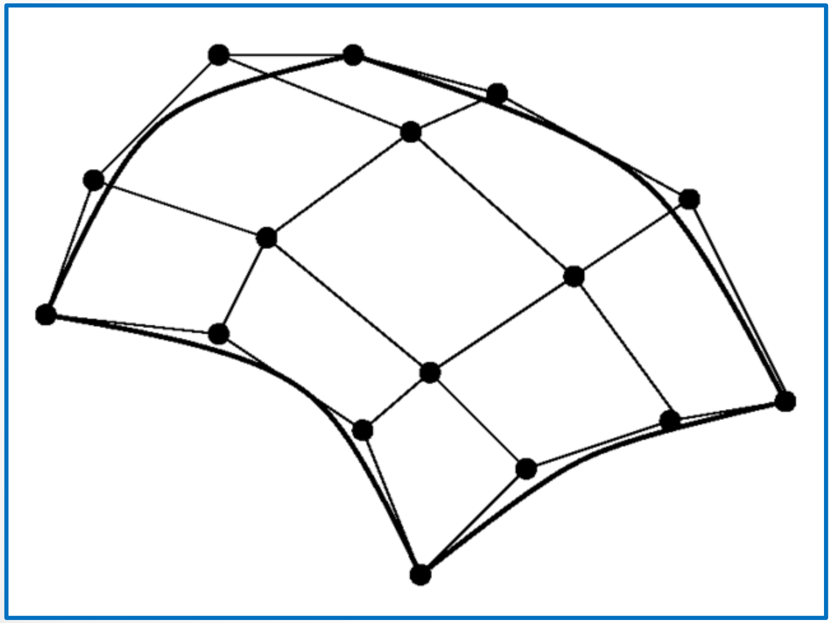
$$z = -\sqrt{R^2 - x^2 - y^2}$$

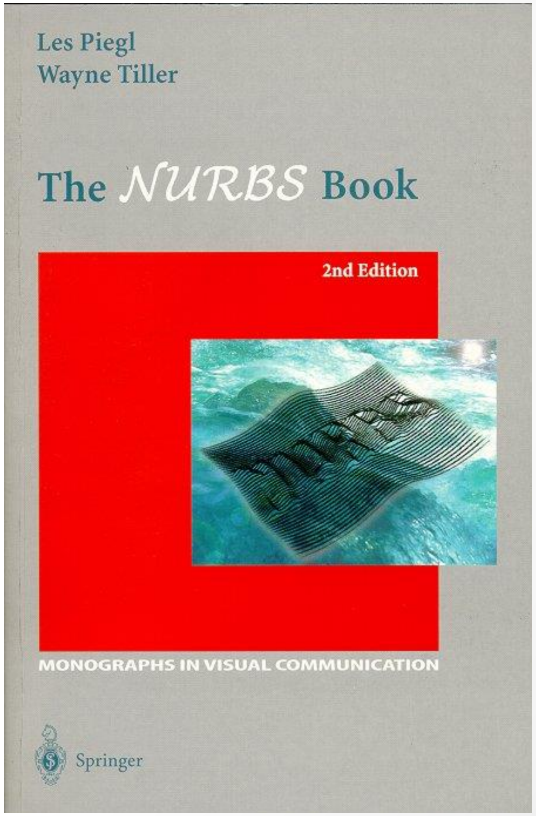


simple since z coord. is explicitly represented by x and y coord.
but not convenient for multi-valued surface function

Freeform Surface Modeling

- Extend 2D curve theory to 3D surface
  - Bézier surface
  - B-Spline surface
  - NURBS surface





Mesh Representation

- A **discrete** 3D surface representation

$$M = (V, E, F)$$

$V: mesh\ vertex\ set$

$E: mesh\ edge\ set$

$F: mesh\ face\ set$