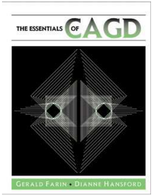
# The Essentials of CAGD

## Chapter 13: NURBS

Gerald Farin & Dianne Hansford

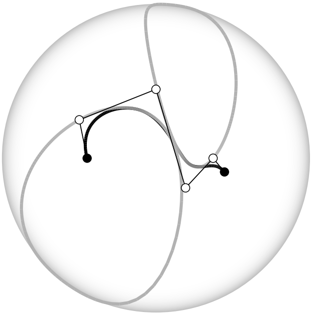
CRC Press, Taylor & Francis Group, An A K Peters Book
www.farinhansford.com/books/essentials-cagd





# Outline

# Introduction to NURBS



NURBS

Non-uniform Rational B-splines

Much of the previous discussion of B-spline curves and B-spline surfaces applies to NURBS

Here: focus on special features of NURBS

Most of these features are already exhibited by conics

# Conics

Conic sections: the oldest known curve form

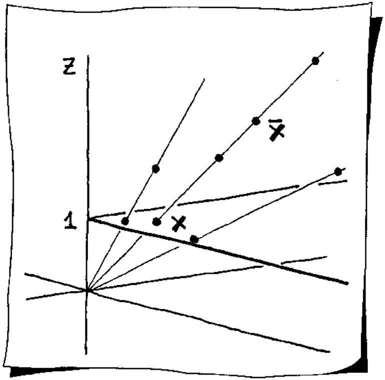
Essential to many CAD systems

Conics were the basis for the first “CAD” system

R. Liming in 1944
– Based the design of airplane fuselages
– *calculating* with conics as opposed to traditional *drafting* with conics

# Conics

*A conic section in $\mathbb{E}^2$ is the perspective projection of a parabola in $\mathbb{E}^3$*




Formulation as rational curves:

– Center of the projection: origin $\mathbf{0}$ (3D coordinate system)

– Projection plane: $z = 1$ (copy of $\mathbb{E}^2$)

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \longrightarrow \begin{bmatrix} x/z \\ y/z \end{bmatrix} = \mathbf{x}$$

Family of points $f\underline{\mathbf{x}}$ project onto $\mathbf{x}$

3D point $\underline{\mathbf{x}}$ called homogeneous form or homogeneous coordinates of $\mathbf{x}$

# Conics

Conic as a parametric rational quadratic curve

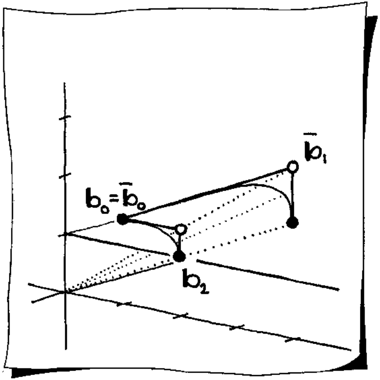
$$\mathbf{c}(t) = \frac{z_0\mathbf{b}_0 B_0^2(t) + z_1\mathbf{b}_1 B_1^2(t) + z_2\mathbf{b}_2 B_2^2(t)}{z_0 B_0^2(t) + z_1 B_1^2(t) + z_2 B_2^2(t)}$$

*weights* $z_0, z_1, z_2 \in \mathbb{R}$ *control points* $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{E}^2$

3D parabola projected onto the conic $\mathbf{c}$
has homogenous control points

$$z_0 \begin{bmatrix} \mathbf{b}_0 \\ 1 \end{bmatrix} \qquad z_1 \begin{bmatrix} \mathbf{b}_1 \\ 1 \end{bmatrix} \qquad z_2 \begin{bmatrix} \mathbf{b}_2 \\ 1 \end{bmatrix}$$

# Conics




Example:

Homogeneous control points

$$\underline{\mathbf{b}}_0 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad \underline{\mathbf{b}}_1 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \quad \underline{\mathbf{b}}_2 = \begin{bmatrix} 4 \\ 0 \\ 2 \end{bmatrix}$$

Project onto the 2D points

$$\mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{b}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Weights: $z_i = 1,\ 2,\ 2$

$$\underline{\mathbf{x}}(0.5) = \begin{bmatrix} 2 \\ 5/4 \\ 7/4 \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} 8/7 \\ 5/7 \end{bmatrix} = \mathbf{x}(0.5)$$

# Reparametrization and Classification

It is possible to change the weights of a conic *without* changing its shape

Initial weights: $z_0, \quad z_1, \quad z_2$

Conic with weights $z_0, \quad cz_1, \quad c^2 z_2 \qquad c \neq 0$ has the same shape

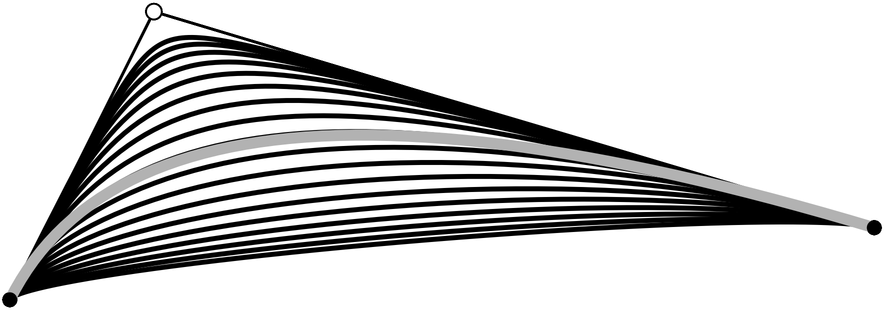
Conic in standard form: characterized by weights $1, \quad cz_1, \quad 1$
Steps:

1. Scale all weights so that $z_0 = 1 \quad \Rightarrow \quad 1, \ \hat{z}_1, \ \hat{z}_2$
2. Set $c = 1/\sqrt{\hat{z}_2} \quad \Rightarrow \quad 1, \ \tilde{z}_1, \ 1$

This change in weights *does* change how it is traversed
$\Rightarrow$ reparametrization

Example:
Initial weights $1, \quad 2, \quad 2 \qquad$ Let $c = 1/\sqrt{2}$
$\Rightarrow$ new weights in standard form: $1, \quad 2/\sqrt{2}, \quad 1$

# Reparametrization and Classification

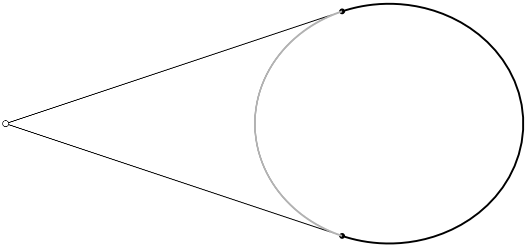


Conic is in standard form ⇒ easy to determine type:

- a hyperbola if $z_1 > 1$
- a parabola if $z_1 = 1$
- an ellipse if $z_1 < 1$

Identify these in the figure

# Reparametrization and Classification



Weights $z_0, \quad z_1, \quad z_2$ all $z_i > 0 \Rightarrow$ curve inside control polygon

Special reparametrization:
Setting $c = -1$ generates weights $z_0, \quad -z_1, \quad z_2$
$\Rightarrow$ evaluation for $t \in [0, 1]$ traces points in the complementary segment

# Derivatives

Conic section written as a rational function
Straightforward approach: derivatives need the quotient rule

Instead:
Conic $\mathbf{c}(t)$ is of the form $\mathbf{c}(t) = \mathbf{p}(t)/z(t)$ (polynomial numerator)

$$\mathbf{p}(t) = z(t)\mathbf{c}(t)$$

Polynomial curve differentiated using the product rule:

$$\dot{\mathbf{p}}(t) = \dot{z}(t)\mathbf{c}(t) + z(t)\dot{\mathbf{c}}(t)$$

Expression $\dot{\mathbf{c}}(t)$ is desired conic derivative

$$\dot{\mathbf{c}}(t) = \frac{1}{z(t)}[\dot{\mathbf{p}}(t) - \dot{z}(t)\mathbf{c}(t)]$$

# Derivatives

Consider two conics

| | | | |
|---|---|---|---|
| #1 | $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ | $w_0, w_1, w_2$ | defined over $[u_0, u_1]$ |
| #2 | $\mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$ | $w_2, w_3, w_4$ | defined over $[u_1, u_2]$ |

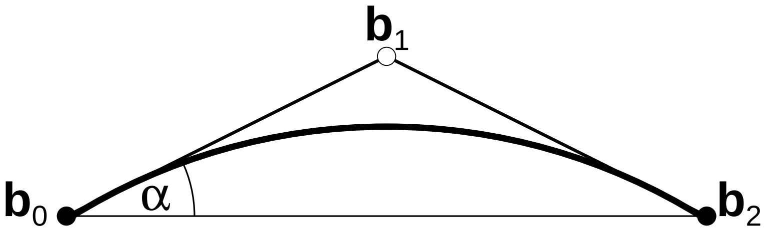
Both segments form a $C^1$ curve if

$$\frac{w_1}{u_1 - u_0}\Delta\mathbf{b}_1 = \frac{w_3}{u_2 - u_1}\Delta\mathbf{b}_2$$

Interval lengths appear due to application of the chain rule
– Composite curve defined with respect to global parameter $u$

Notice absence of the weight $w_2$ in the $C^1$ equation

# The Circle



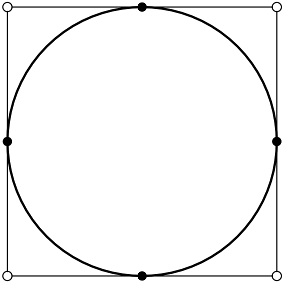

Circular arc: most widely used conic

Represent it as a rational quadratic Bézier curve:
– Control polygon must form an isosceles triangle (symmetry!)
– Weights 1, $z_1$, 1

$$z_1 = \cos\alpha$$

$$\alpha = \angle(\mathbf{b}_2, \mathbf{b}_0, \mathbf{b}_1)$$

# The Circle



Whole circle represented by piecewise rational quadratics:

**Method 1**:

– Represent one quarter with the control polygon

– Represent remaining part with the complementary segment

**Method 2**:

– Use four control polygons $\Rightarrow$ Convex hull property

# The Circle

Arc of a circle in sin/cos parametrization
⇒ Nice property: traverses circle with *unit speed*

Arc of a circle in rational quadratic form
⇒ Parameter $t$ does *not* traverse the circle with *unit speed*

Need numerical techniques to split arcs into equiangular segments

# Rational Bézier Curves

4D points and their 3D projections:

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \longrightarrow \begin{bmatrix} x/w \\ y/w \\ z/w \end{bmatrix} = \mathbf{x}$$

Degree $n$ Bézier curve in $\mathbb{E}^4$ projected into $w = 1$ hyperplane
$\Rightarrow$ Rational Bézier curve of degree $n$ in $\mathbb{E}^3$

$$\mathbf{x}(t) = \frac{w_0 \mathbf{b}_0 B_0^n(t) + \cdots + w_n \mathbf{b}_n B_n^n(t)}{w_0 B_0^n(t) + \cdots + w_n B_n^n(t)} \qquad \mathbf{x}(t), \quad \mathbf{b}_i \in \mathbb{E}^3$$

Homogeneous form of the curve:

$$\underline{\mathbf{x}}(t) = \underline{\mathbf{b}}_0 B_0^n(t) + \cdots + \underline{\mathbf{b}}_n B_n^n(t)$$

**Evaluation**:
de Casteljau algorithm to homogeneous form and project result into 3D

# Rational Bézier Curves

Example:

Control points:

$$\begin{bmatrix}-1\\0\end{bmatrix},\quad \begin{bmatrix}0\\1\end{bmatrix},\quad \begin{bmatrix}0\\-1\end{bmatrix},\quad \begin{bmatrix}1\\0\end{bmatrix}$$

Weights: $1, 2, 1, 1$

Homogeneous control points

$$\begin{bmatrix}-1\\0\\1\end{bmatrix},\quad \begin{bmatrix}0\\2\\2\end{bmatrix},\quad \begin{bmatrix}0\\-1\\1\end{bmatrix},\quad \begin{bmatrix}1\\0\\1\end{bmatrix}$$

Applying the de Casteljau algorithm

$$\underline{\mathbf{x}}(0.5) = \begin{bmatrix}0.0\\0.375\\1.375\end{bmatrix} \qquad \text{then} \qquad \mathbf{x}(0.5) = \begin{bmatrix}0.0\\0.2727\end{bmatrix}$$

# Rational Bézier Curves

If all weights are one $\Rightarrow$ standard nonrational Bézier curve
– Denominator is identically equal to one

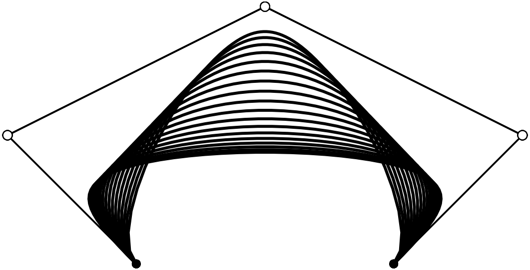
If some $w_i$ are negative: singularities may occur
$\Rightarrow$ Only deal with nonnegative $w_i$

If all $w_i$ are nonnegative, we have the convex hull property

Rational Bézier curves enjoy all the properties that their nonrational counterparts possess
– Example: affine invariance

# Rational Bézier Curves

Influence of the weights



Top curve corresponds to $w_2 = 10$
Bottom curve corresponds to $w_2 = 0.1$

# Rational Bézier Curves

Rational Bézier curves are projectively invariant

Projective map: $4 \times 4$ matrix $A$

$$\underline{\bar{\mathbf{x}}} = A\underline{\mathbf{x}}$$

Map will change the weights of a curve

– Example: Projective map of rational quadratic conics can map an ellipse to a hyperbola

# Rational Bézier Curves

Curvature at $t = 0$:

$$\kappa(0) = 2\frac{n-1}{n}\frac{w_0 w_2}{w_1}\frac{\mathrm{area}[\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2]}{\|\mathbf{b}_1 - \mathbf{b}_0\|^3}$$

Torsion at $t = 0$:

$$\tau(0) = \frac{3}{2}\frac{n-2}{n}\frac{w_0 w_3}{w_1 w_2}\frac{\mathrm{volume}[\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]}{\mathrm{area}[\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2]^2}$$

# Rational B-spline Curves

NonUniform Rational B-spline curveS NURBS

– CAD/CAM industry standard

$$\mathbf{x}(u) = \frac{w_0\mathbf{d}_0 N_0^n(u) + \ldots + w_{D-1}\mathbf{d}_{D-1} N_{D-1}^n(u)}{w_0 N_0^n(u) + \ldots + w_{D-1} N_{D-1}^n(u)}$$

All properties from the rational Bézier form carry over

– Example: convex hull property (for nonnegative weights)

– Example: affine and projective invariance

Designing with NURBS curves:

– Added freedom of changing weights

– Change of only one weight affects curve only locally

# Rational Bézier and B-spline Surfaces

Generalize Bézier and B-spline surfaces to rational
– Similar to curve case

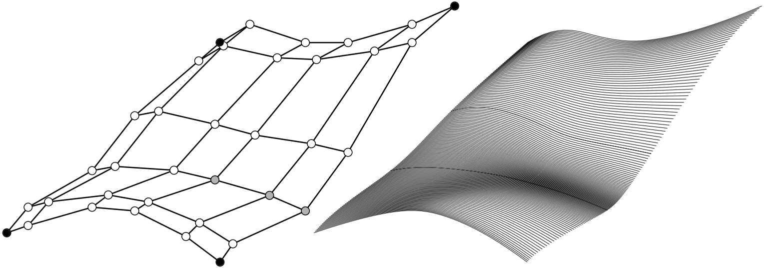
Rational Bézier or B-spline surface is projection of a 4D tensor product Bézier or B-spline surface

Rational Bézier patch:

$$\mathbf{x}(u,v) = \frac{M^{\mathrm{T}}\mathbf{B}_w N}{M^{\mathrm{T}} W N}$$

– Matrix $\mathbf{B}_w$ has elements $w_{i,j}\mathbf{b}_{i,j}$
– Matrix $W$ has elements $w_{i,j}$ (weights)
Influence the shape of the surface

# Rational Bézier and B-spline Surfaces



Rational B-spline surface:

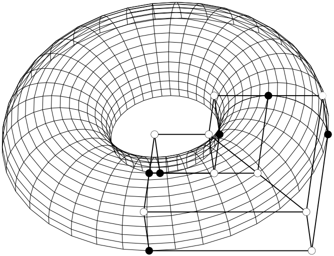
$$\mathbf{s}(u,v) = \frac{M^{\mathrm{T}} \mathbf{D}_w N}{M^{\mathrm{T}} W N}$$

Matrices $M$ and $N$ contain the B-spline basis functions in $u$ and $v$

Figure: weights of gray control points set to 3

# Surfaces of Revolution

Rational B-spline surfaces
allow exact representation of surfaces of revolution



Surface of revolution:
rotate a curve (generatrix)
around an axis

Generatrix:

$$\mathbf{g}(v) = \begin{bmatrix} r(v) \\ 0 \\ z(v) \end{bmatrix}$$

Planar curve in $(x, z)$-plane

Axis of revolution here: $z-$axis
(comes out of the center of
half-torus)

# Surfaces of Revolution

## Surface of revolution

$$\mathbf{x}(u,v) = \begin{bmatrix} r(v)\cos u \\ r(v)\sin u \\ z(v) \end{bmatrix}$$

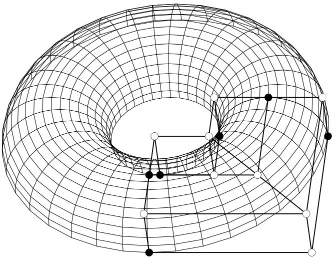
For fixed $v$:
isoparametric line $v = \mathit{const}$ traces out a circle of radius $r(v)$
– called a *meridian*

Control points of the generatrix

$$\mathbf{c}_i = \begin{bmatrix} x_i \\ 0 \\ z_i \end{bmatrix} \quad \text{and weights } w_i$$

# Surfaces of Revolution

Surface of revolution broken down into four symmetric pieces



– Rational quadratic in the parameter $u$

– Each piece one quadrant of $(x, y)-$plane

Over the first quadrant:
surface with three columns of control points and associated weights

$$\begin{bmatrix} x_i \\ 0 \\ z_i \end{bmatrix}, \quad \begin{bmatrix} x_i \\ x_i \\ z_i \end{bmatrix}, \quad \begin{bmatrix} 0 \\ x_i \\ z_i \end{bmatrix}$$

Weights $w_i, \frac{\sqrt{2}}{2} w_i, w_i$

Remaining three surface segments obtained by reflecting this one

# Surfaces of Revolution

Example: one-sixteenth of a *torus*

– created by revolving a quarter circle around the $z-$axis

– quarter circle defined in the $(x, z)$-plane and centered at $[2 \;\; 0 \;\; 0]^{\mathrm{T}}$]

Bézier points defining generatrix

$$\begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} \qquad \text{weights} \;\; 1, \quad \sqrt{2}/2, \quad 1$$

Control points for a rational biquadratic patch

$$\begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix} \qquad \text{with weights} \quad \begin{bmatrix} 1 & \frac{\sqrt{2}}{2} & 1 \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{\sqrt{2}}{2} \\ 1 & \frac{\sqrt{2}}{2} & 1 \end{bmatrix}$$
$$\begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$