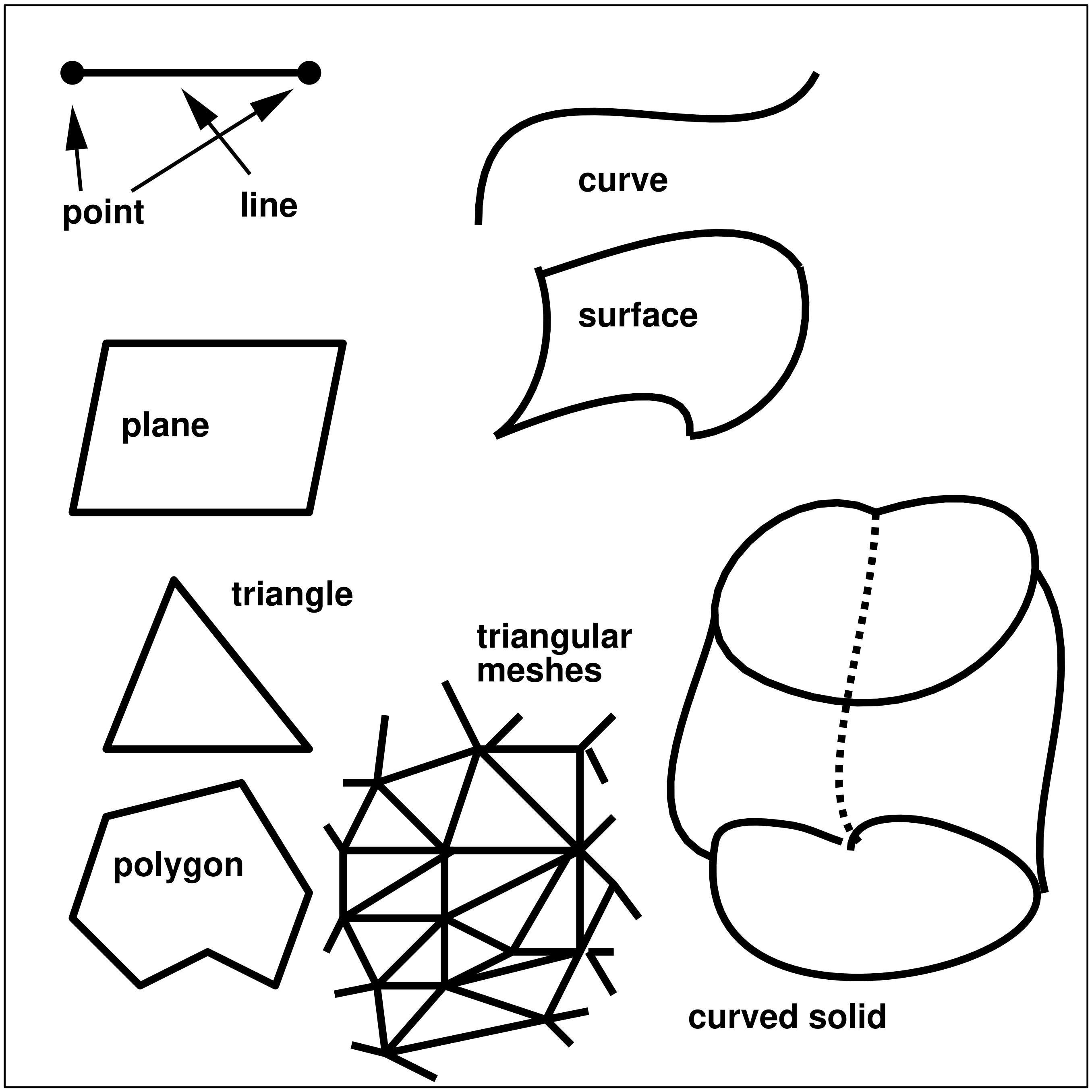
GEOMETRIC MODELING

- Why geometric modeling?
- Fundamental for visual computing
- Critical for virtual engineering
- Interaction
- Geometric information for decision making

Geometric Shapes



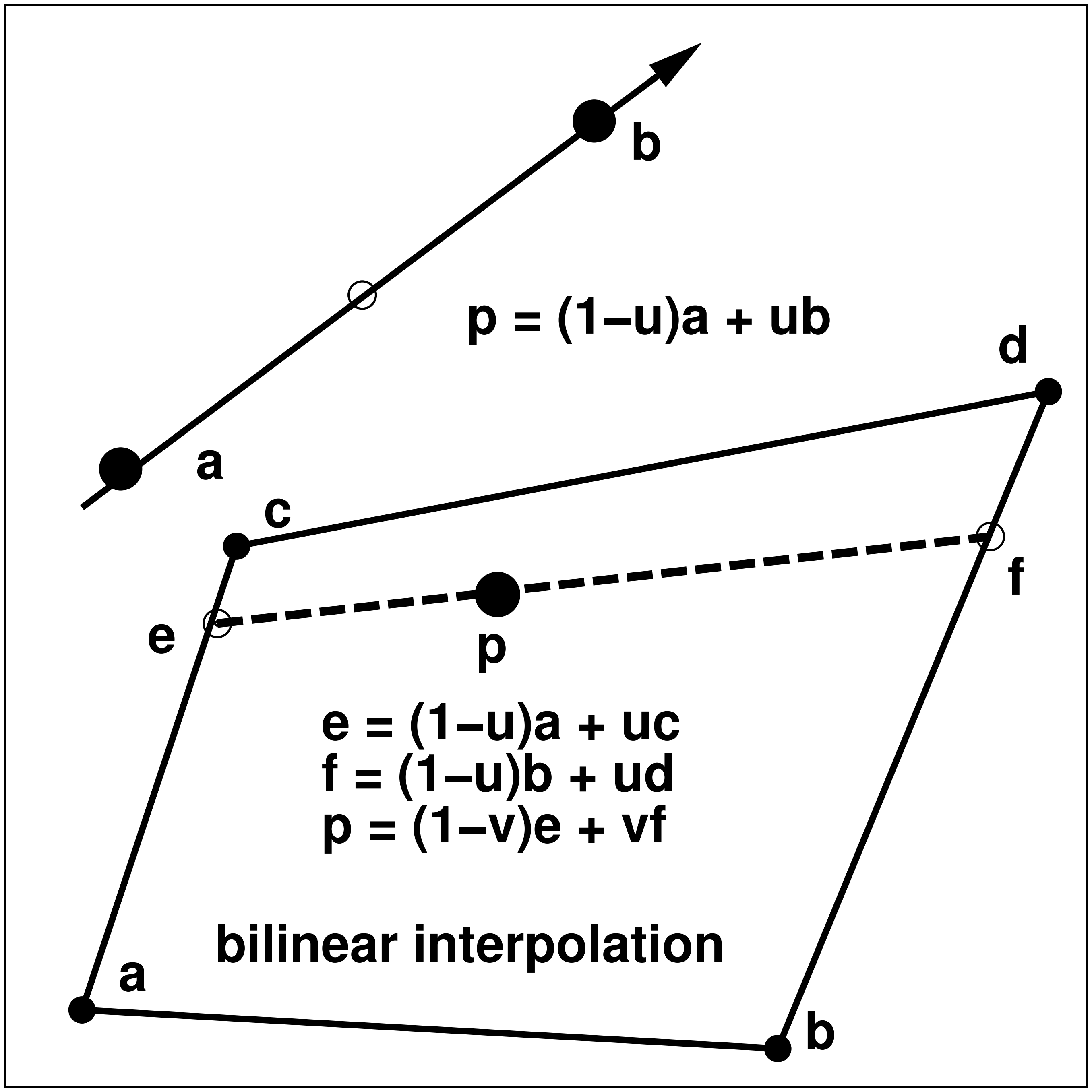
Basic Concepts

- **Linear interpolation:** $\mathbf{v} = \mathbf{v}_0(1-t) + \mathbf{v}_1(t)$
- **Local coordinates:** $\mathbf{v} \in [\mathbf{v}_0, \mathbf{v}_1]$**,** $t \in [0,1]$
- **Reparameterization:** $f(u)$**,** $u = g(v)$**,** $f(g(v)) = h(v)$
- **Affine transformation:**

$$f(ax+by) = af(x) + bf(y), a+b = 1$$

- **Polynomials**
- **Continuity**

Linear Interpolation



Fundamental Features

- Geometry
  - position, direction, length, area
  - normal, tangent, etc.
- Interaction
  - size, continuity
  - collision, intersection
- Topology
- Differential
  - curvature, arc-length
- Physical
- Computer representation & data structure
- Others!

Mathematical Formulations

- **Point:**

$$\mathbf{p} = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix}$$

- **Line:**

$$\mathbf{l}(u) = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix} u + \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix}$$

- **Quadratic curve:**

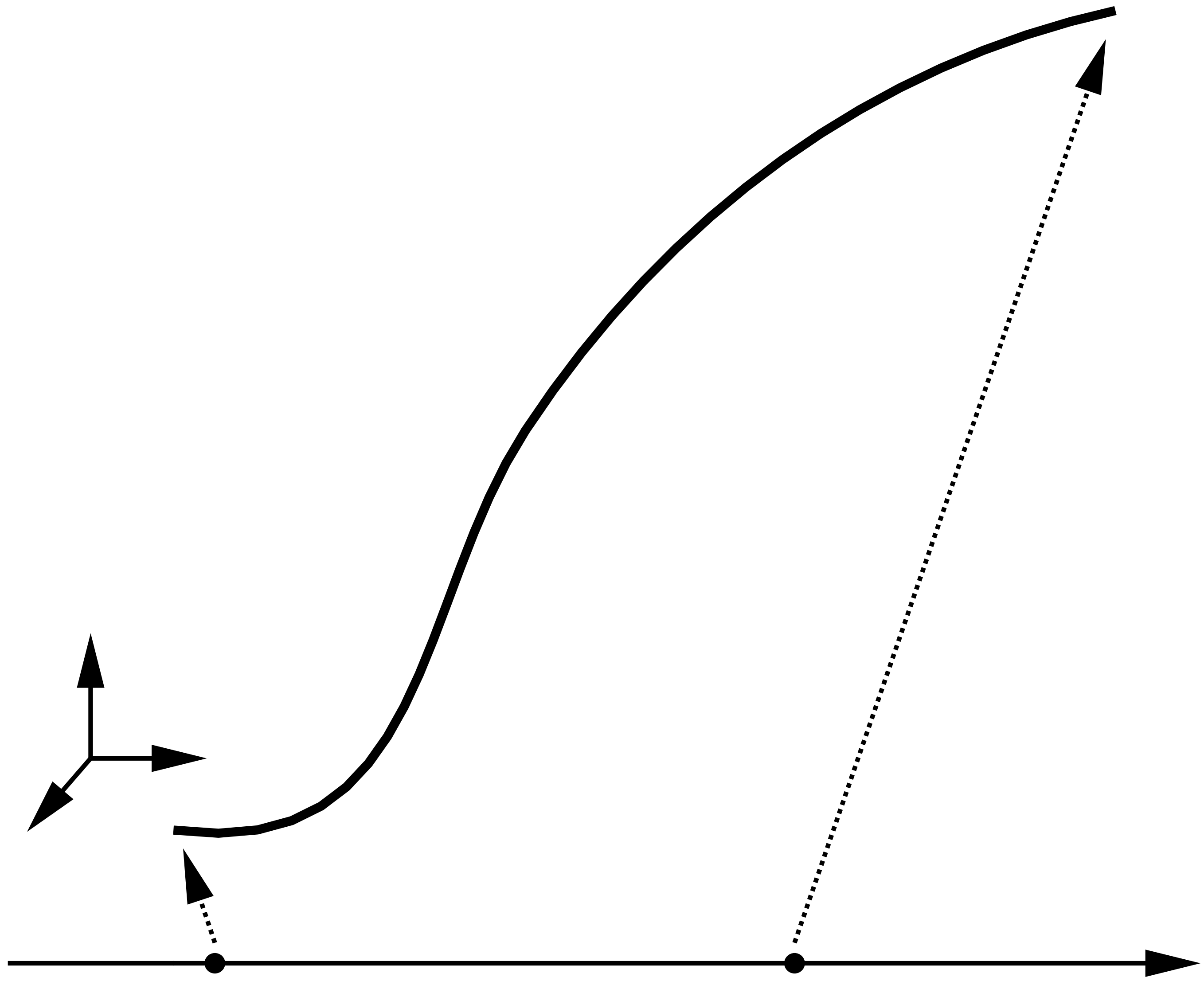
$$\mathbf{q}(u) = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix} u^2 + \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix} u + \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ \mathbf{c}_z \end{bmatrix}$$

- **parametric domain:** $u \in [u_s, u_e]$

- **Reparameterization:** $v = (u - u_s)/(u_e - u_s)$, $v \in [0, 1]$

- **High-order polynomials**

Parametric Polynomials



Parametric Polynomials

- **High-order polynomial**

$$\mathbf{c}(u) = \begin{bmatrix} \mathbf{a}_{0,x} \\ \mathbf{a}_{0,y} \\ \mathbf{a}_{0,z} \end{bmatrix} + \ldots + \begin{bmatrix} \mathbf{a}_{i,x} \\ \mathbf{a}_{i,y} \\ \mathbf{a}_{i,z} \end{bmatrix} u^i + \ldots + \begin{bmatrix} \mathbf{a}_{n,x} \\ \mathbf{a}_{n,y} \\ \mathbf{a}_{n,z} \end{bmatrix} u^n$$

- **No intuitive insight for the curve shape**

- **Difficult for piecewise smooth curves**

Lagrange Curves

- **Curve**

$$\mathbf{c}(u) = \begin{bmatrix} \mathbf{a}_{0,x} \\ \mathbf{a}_{0,y} \\ \mathbf{a}_{0,z} \end{bmatrix} L_0^n(u) + \ldots + \begin{bmatrix} \mathbf{a}_{n,x} \\ \mathbf{a}_{n,y} \\ \mathbf{a}_{n,z} \end{bmatrix} L_n^n(u)$$

- **Lagrange polynomials of degree n: $L_i^n(u)$**

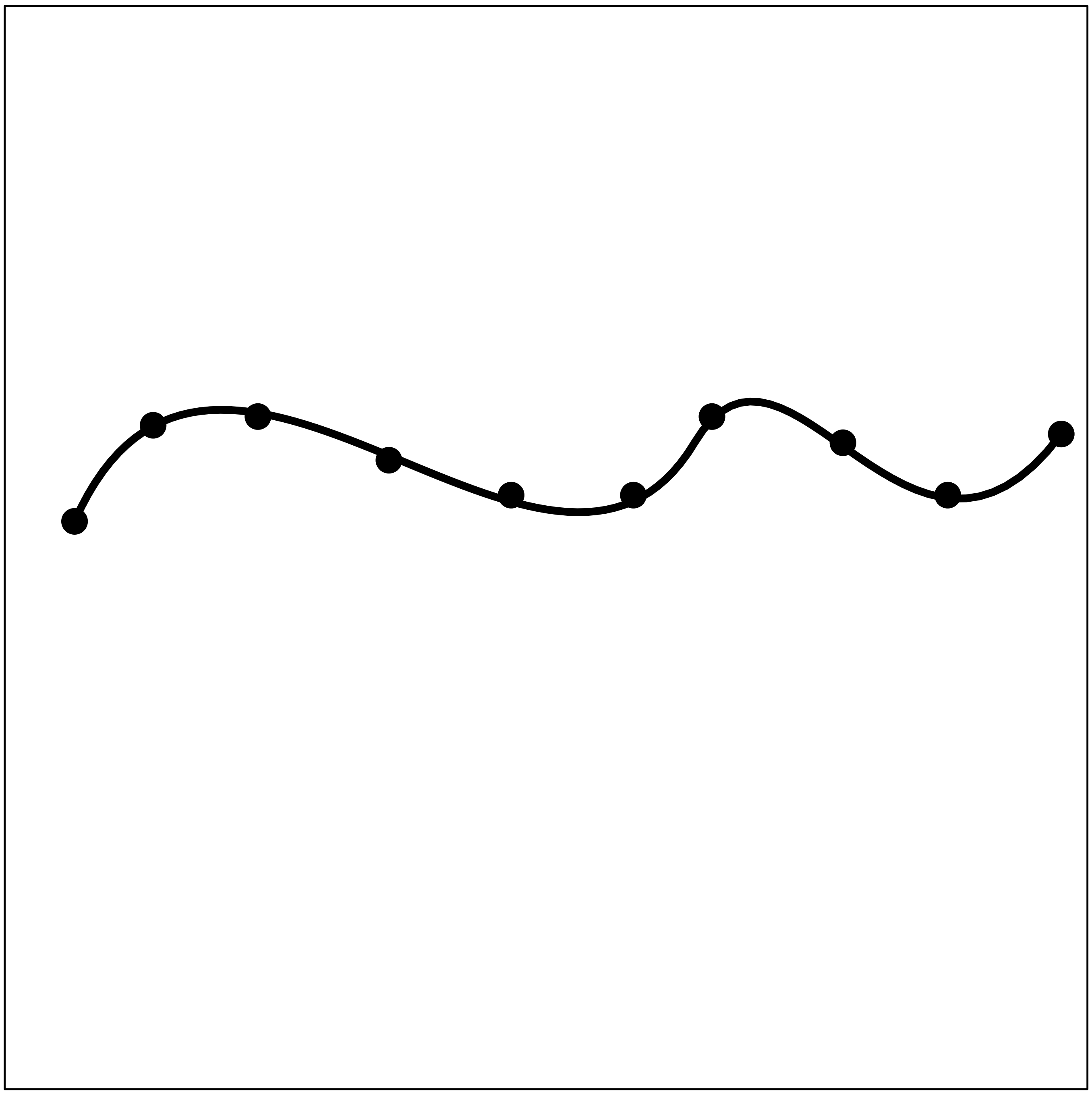
- **A knot sequence: $u_0, \ldots, u_n$**

- **$L_i^n(u_j) = \delta_{ij}$, δ is the Kronecker delta**

- **The curve interpolates all the a_i's**

- **Unwanted oscillation**

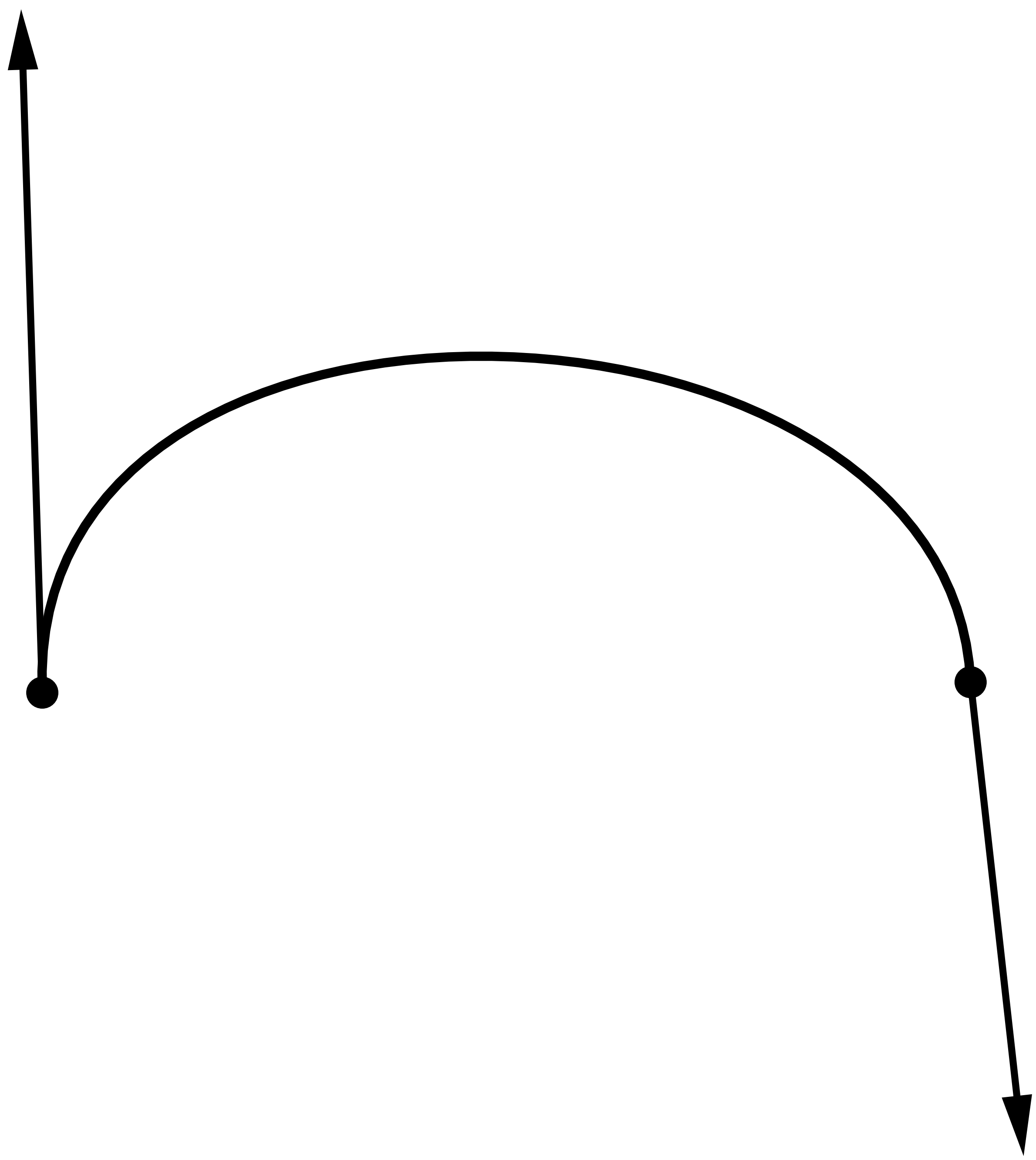
Lagrange Curve



Cubic Hermite Splines

- **Two vertices and two tangent vectors:** $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{d}_0$, **and** $\mathbf{d}_1$,
- $\mathbf{c}(0) = \mathbf{v}_0$, $\mathbf{c}(1) = \mathbf{v}_1$,
- $\mathbf{c}^{(1)}(0) = \mathbf{d}_0$, $\mathbf{c}^{(1)}(1) = \mathbf{d}_1$,
- $\mathbf{c}(u) = \mathbf{v}_0 H_0^3(u) + \mathbf{v}_1 H_1^3(u) + \mathbf{d}_0 H_2^2(u) + \mathbf{d}_1 H_3^3(u)$,
- $H_0^3(u) = 2u^3 - 3u^2 + 1$, $H_1^3(u) = -2u^3 + 3u^2$,
- $H_2^3(u) = u^3 - 2u^2 + u$, $H_3^3(u) = u^3 - u^2$
- **What do these basis functions look like?**

Cubic Hermite Splines



Hermite Curves

- **Curve**

$$\mathbf{c}(u) = \mathbf{v}_0^0 H_0^n(u) + \mathbf{v}_0^1 H_1^n(u) + \ldots + \mathbf{v}_0^{(n-1)/2} H_{(n-1)/2}^n(u)$$

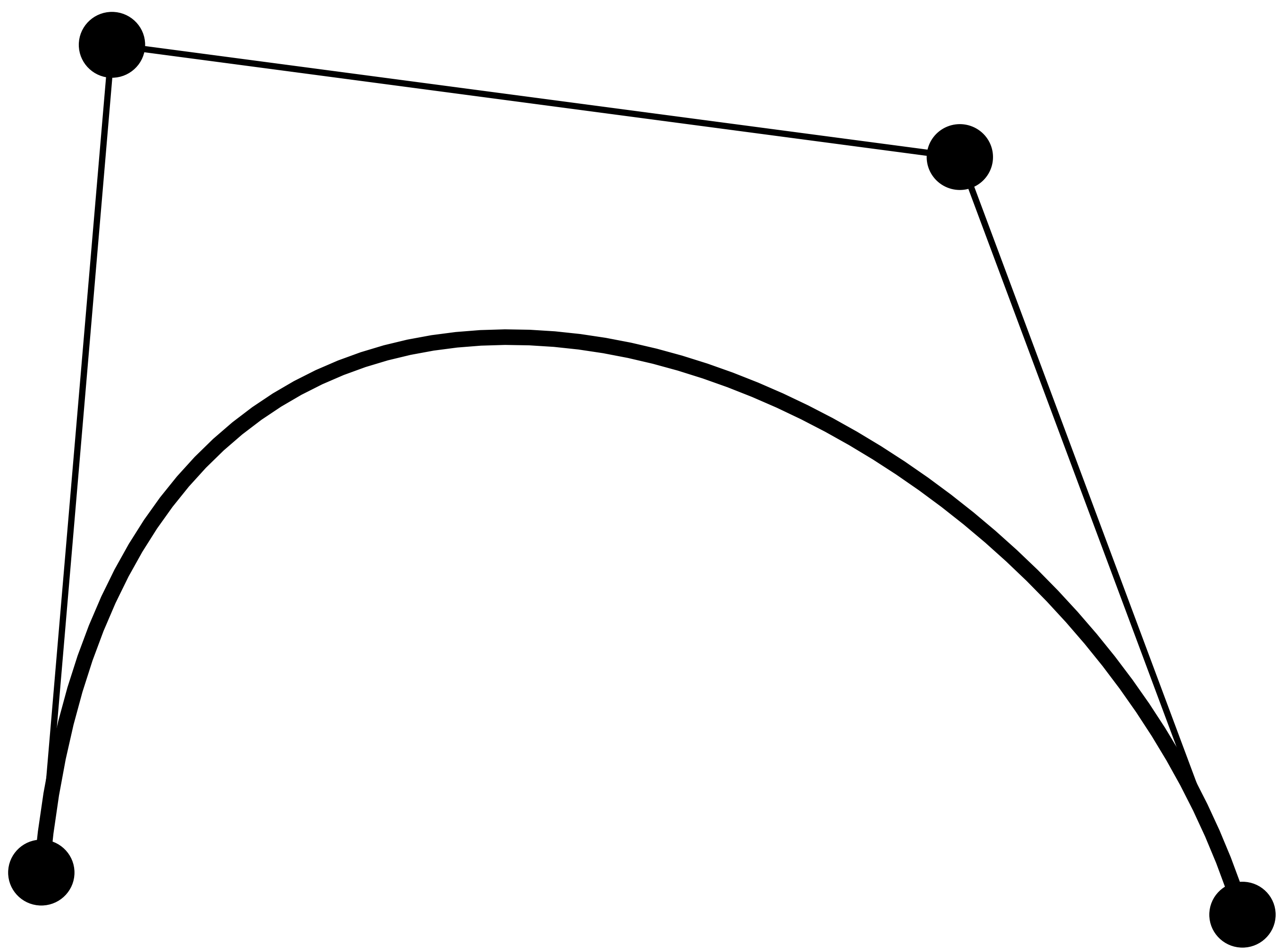
$$+\mathbf{v}_1^{(n-1)/2} H_{(n+1)/2}^n(u) + \ldots + \mathbf{v}_1^1 H_{(n-1)}^n(u) + \mathbf{v}_1^0 H_n^n(u)$$

- $\mathbf{v}_0^i = \mathbf{c}^{(i)}(0)$**,** $\mathbf{v}_1^i = \mathbf{c}^{(i)}(1)$**,** $i = 0, \ldots, (n-1)/2$**,** **and** n **is odd**

- **Geometric intuition**

- **High-order derivatives are required**

Cubic Bezier Curves



Bezier Splines

- **Curve**

$$\mathbf{c}(u) = \sum_{i=0}^{n} \mathbf{p}_i B_i^n(u)$$

- **Control points:** $\mathbf{p}_i$

- **Basis functions:**

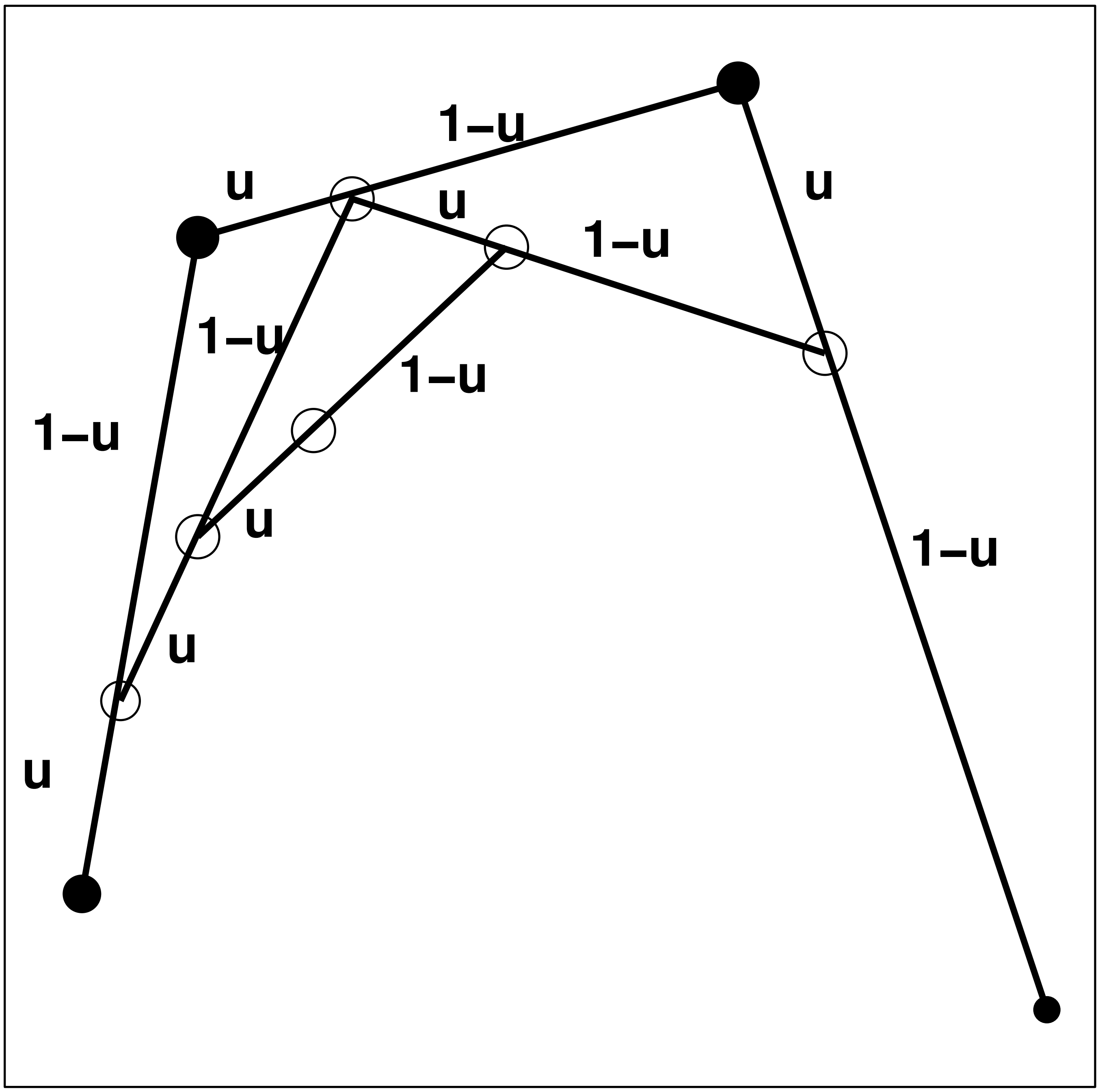
$$B_i^n(u) = \begin{pmatrix} n \\ i \end{pmatrix} (1-u)^{n-i} u^i$$

- **What do these basis functions look like?**

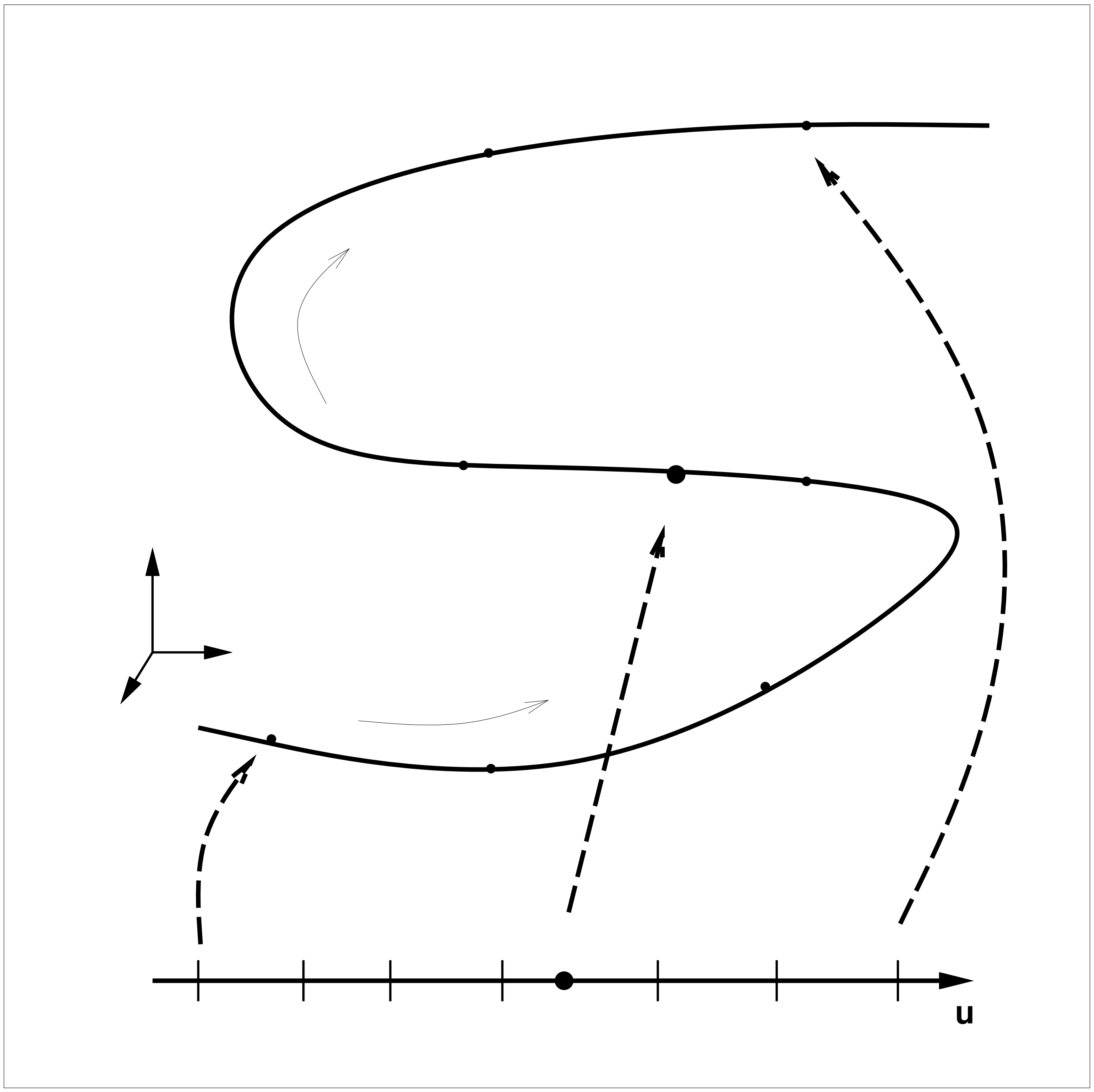
Bezier Splines Properties

- Basis functions are positive
- The sum of basis functions is unity
- Boundary interpolation
- Convex hull
- Recursive subdivision and evaluation
- Differentiation & integration
- Degree elevation
  - use polynomial of degree $n+1$ to express that of degree n
- Composite curves
- Geometric continuity
- Display of curve

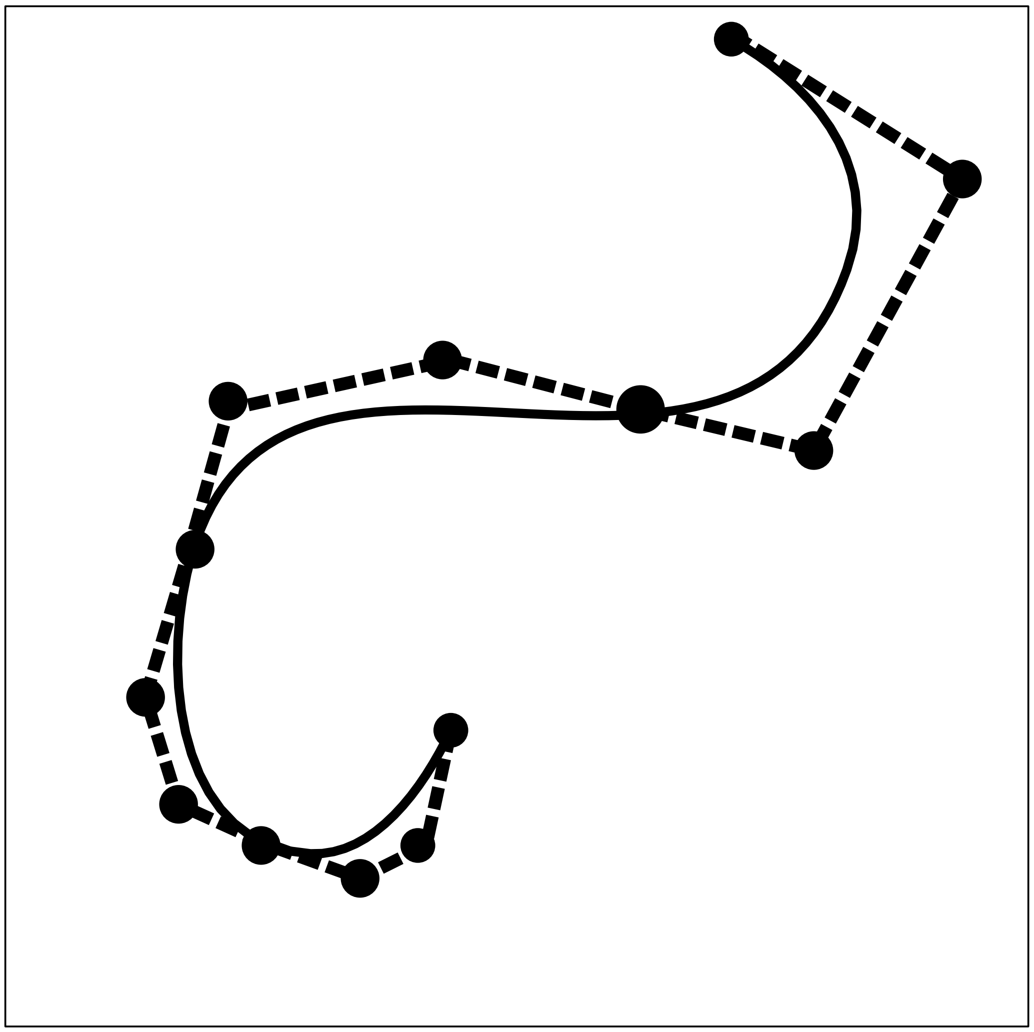
Cubic Bezier Algorithm



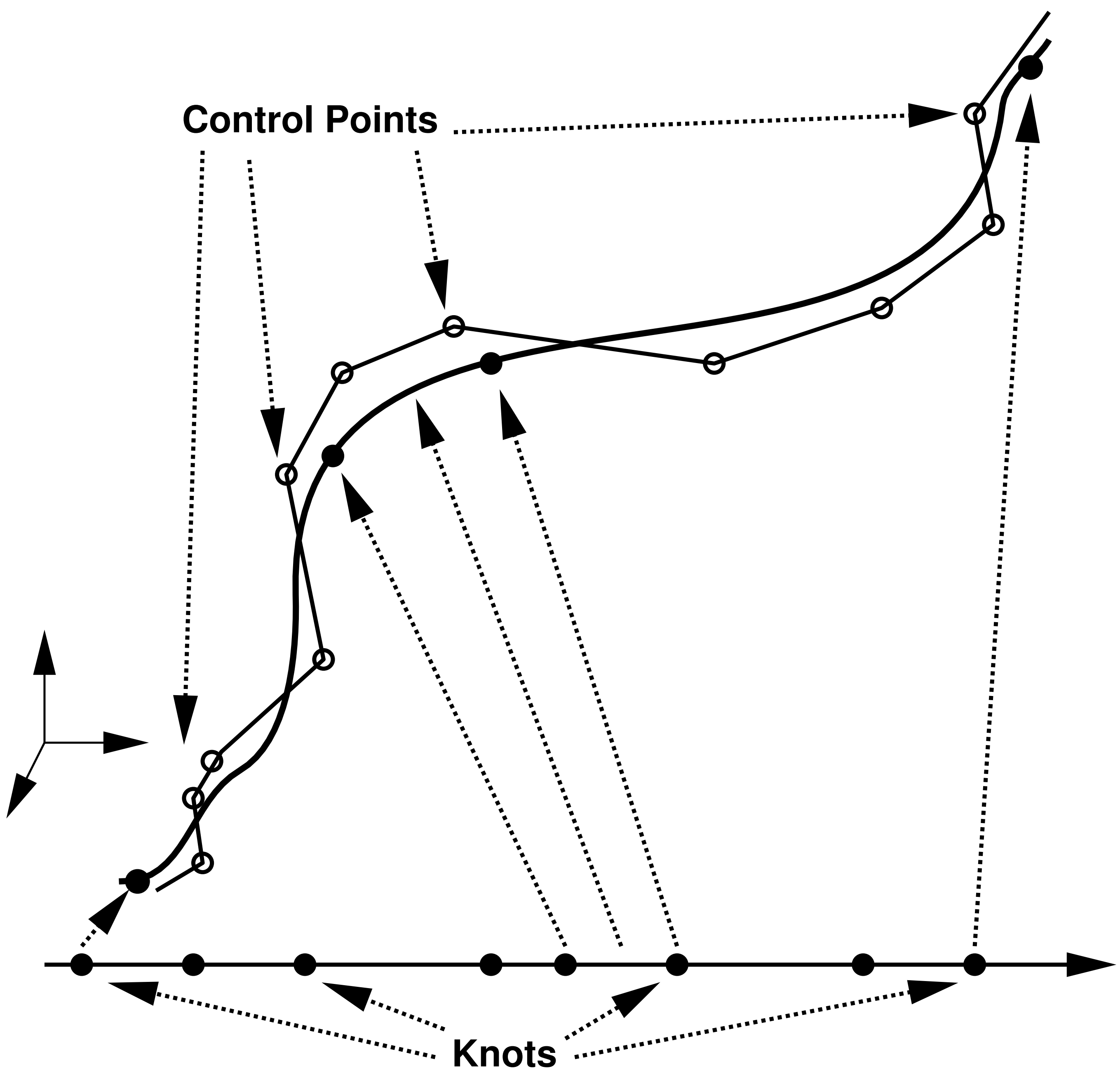
Piecewise Curves



Piecewise Bezier Curve



Piecewise B-splines



B-spline Curves

- **Curve:** $\mathbf{c}(u) = \sum_{i=0}^{n} \mathbf{p}_i B_{i,k}(u)$
- **Parametric variable:** u
- $n+1$ **control points:** $\mathbf{p}_i$
- $n+k+1$ **non-decreasing knots:** $u_0 \leq u_1 \leq \ldots \leq u_{n+k}$
- **Basis functions of degree** $k-1$**:** $B_{i,k}(u)$
- **Parametric domain:** $u_{k-1} \leq u \leq u_{n+1}$

B-Spline Basis Functions

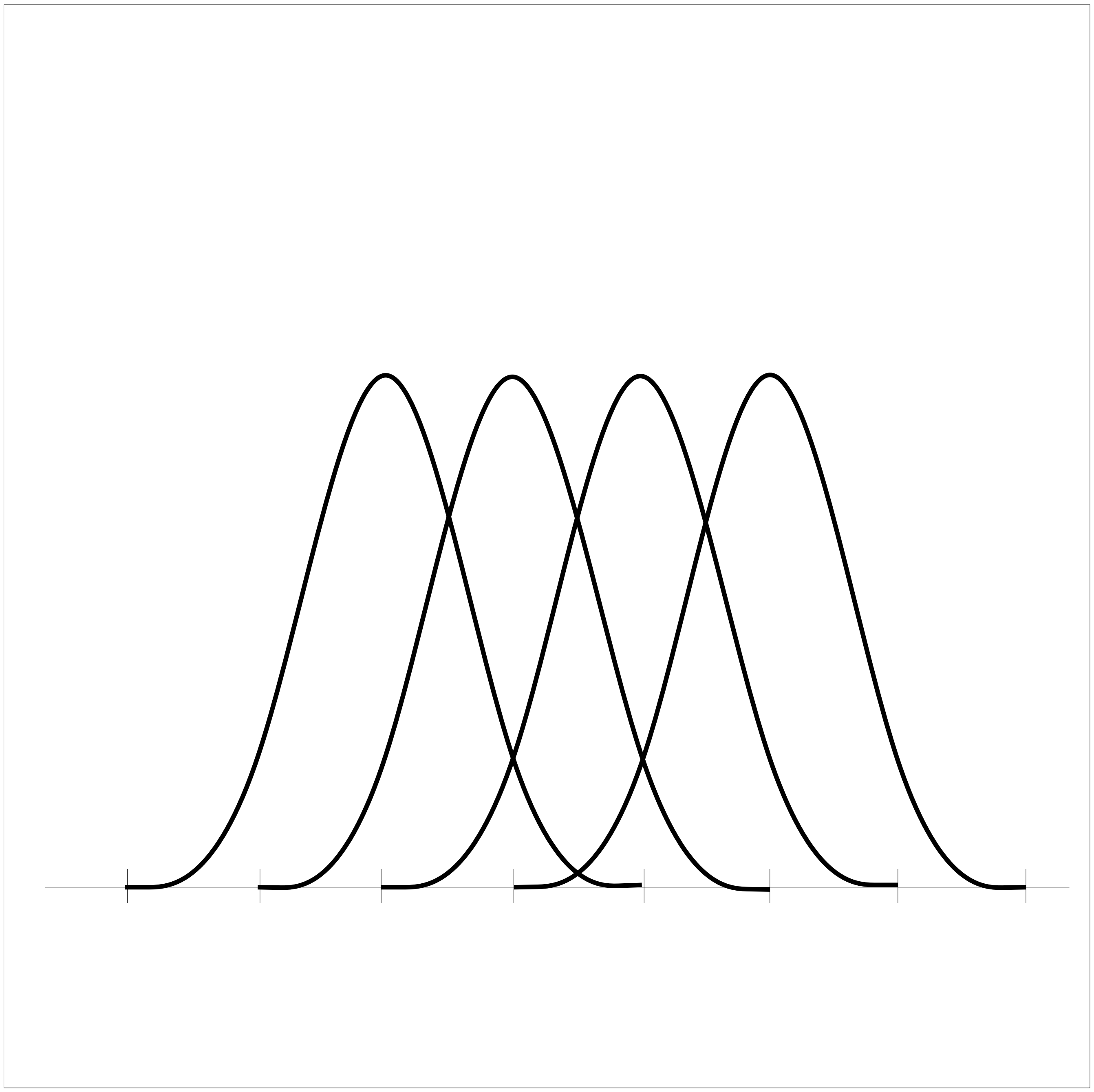
- $B_{i,1}(u) = \begin{cases} 1 & \textbf{for } u_i \leq u < u_{i+1} \\ 0 & \textbf{otherwise} \end{cases}$,

- $B_{i,k}(u) = \frac{u - u_i}{u_{i+k-1} - u_i} B_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} B_{i+1,k-1}(u).$

B-spline Basis Functions

- **One example: quadratic**
- **Knot vector:** $[0, 1, 2, 3, 4, 5, 6]$
- $B_{0,3}(u) = \begin{cases} \frac{1}{2}u^2, & \textbf{for } 0 \leq u < 1 \\ \frac{1}{2}u(2-u) + \frac{1}{2}(u-1)(3-u), & \textbf{for } 1 \leq u < 2 \\ \frac{1}{2}(3-u)^2 & \textbf{for } 2 \leq u < 3 \end{cases}$
- $B_{1,3}(u) = \begin{cases} \frac{1}{2}(u-1)^2, & \textbf{for } 1 \leq u < 2 \\ \frac{1}{2}(u-1)(3-u) + \frac{1}{2}(u-2)(4-u), & \textbf{for } 2 \leq u < 3 \\ \frac{1}{2}(4-u)^2, & \textbf{for } 3 \leq u < 4 \end{cases}$
- $B_{2,3}(u) = \ldots$
- $B_{3,3}(u) = \ldots$

B-Spline Basis Functions



B-spline Properties

- Control points and basis functions: $n + 1$
- Degree of basis functions: $k - 1$
- Partition of unity, positivity, and recursive evaluation for basis functions
- The degree is independent of the number of control points
- The curve is $C^{(k-2)}$ continuous in general
- Local control
- Special case: Bezier splines
- Efficient algorithms & tools
  - evaluation
  - knot insertion
  - degree elevation

  - derivative, integration
  - continuity

- Composite Bezier curves for B-splines

B-Spline Properties

- Basis functions
  - partition of unity
  - positivity
  - recursive evaluation
- Degree is independent of the number of control points
- Local control
- The curve is $C^{(k-2)}$ continuous in general
- Efficient algorithms & tools
- Special case: Bezier splines

From B-splines to NURBS

- What are NURBS???
- Non Uniform Rational B-Splines (NURBS)
- Rational curve motivation
  - polynomial, conic section, etc
  - industry standard
- B-splines

$$\mathbf{c}(u) = \sum_{i=0}^{n} \begin{bmatrix} \mathbf{p}_{i,x} w_i \\ \mathbf{p}_{i,y} w_i \\ \mathbf{p}_{i,z} w_i \\ w_i \end{bmatrix} B_{i,k}(u)$$

- NURBS

$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)}$$

From B-splines to NURBS

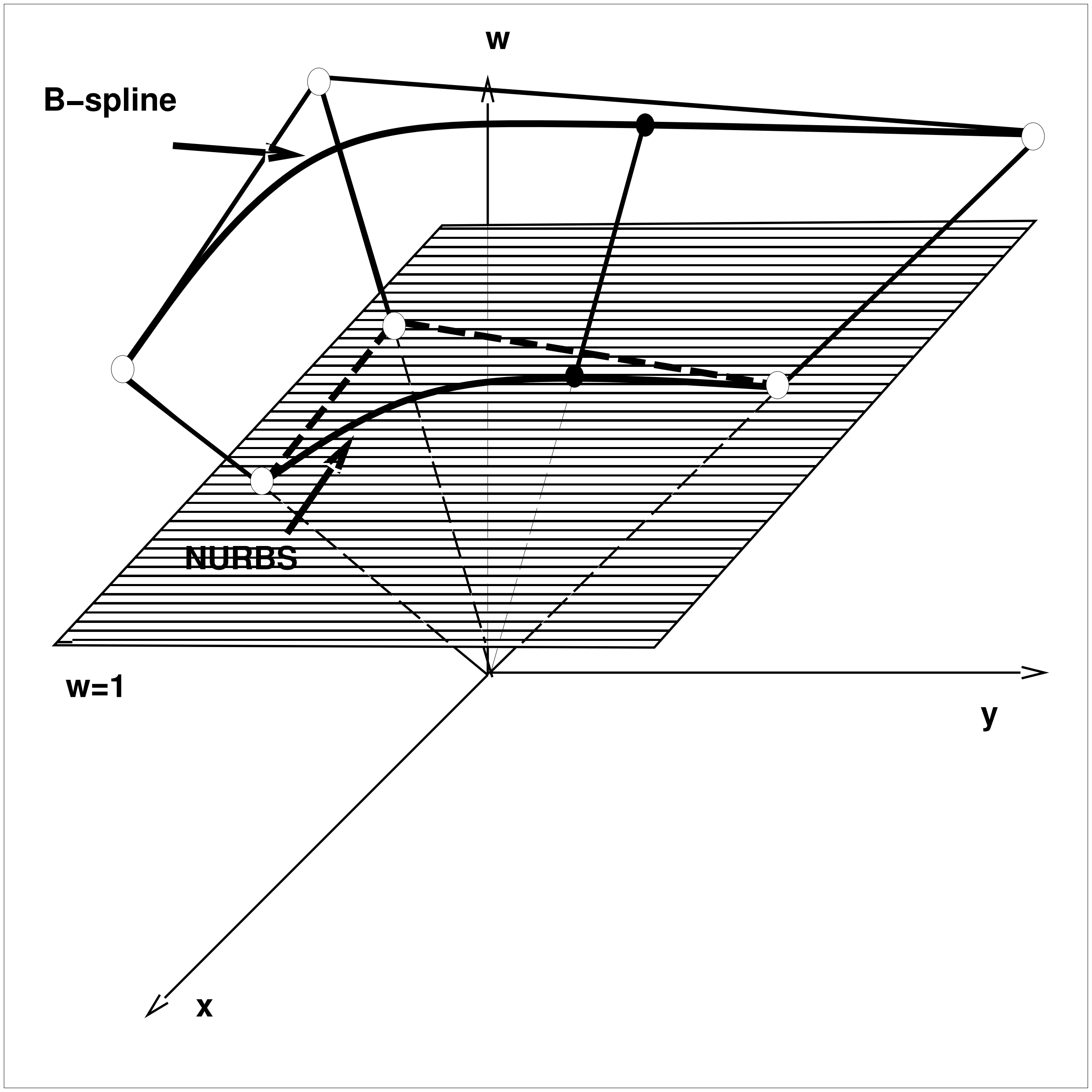
- **B-splines**

$$\mathbf{c}(u) = \sum_{i=0}^{n} \begin{bmatrix} \mathbf{p}_{i,x} w_i \\ \mathbf{p}_{i,y} w_i \\ \mathbf{p}_{i,z} w_i \\ w_i \end{bmatrix} B_{i,k}(u)$$

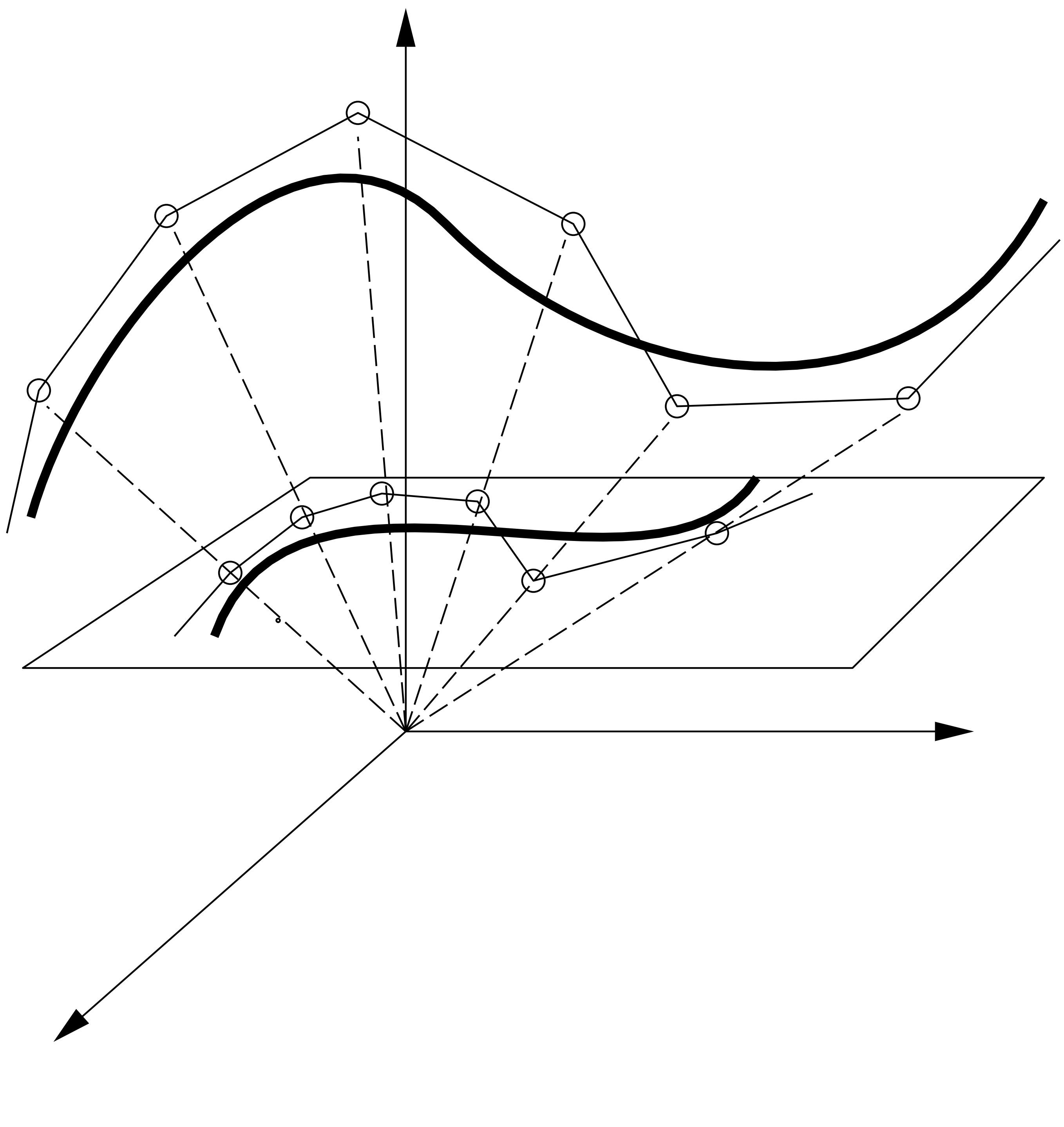
- **NURBS**

$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)}$$

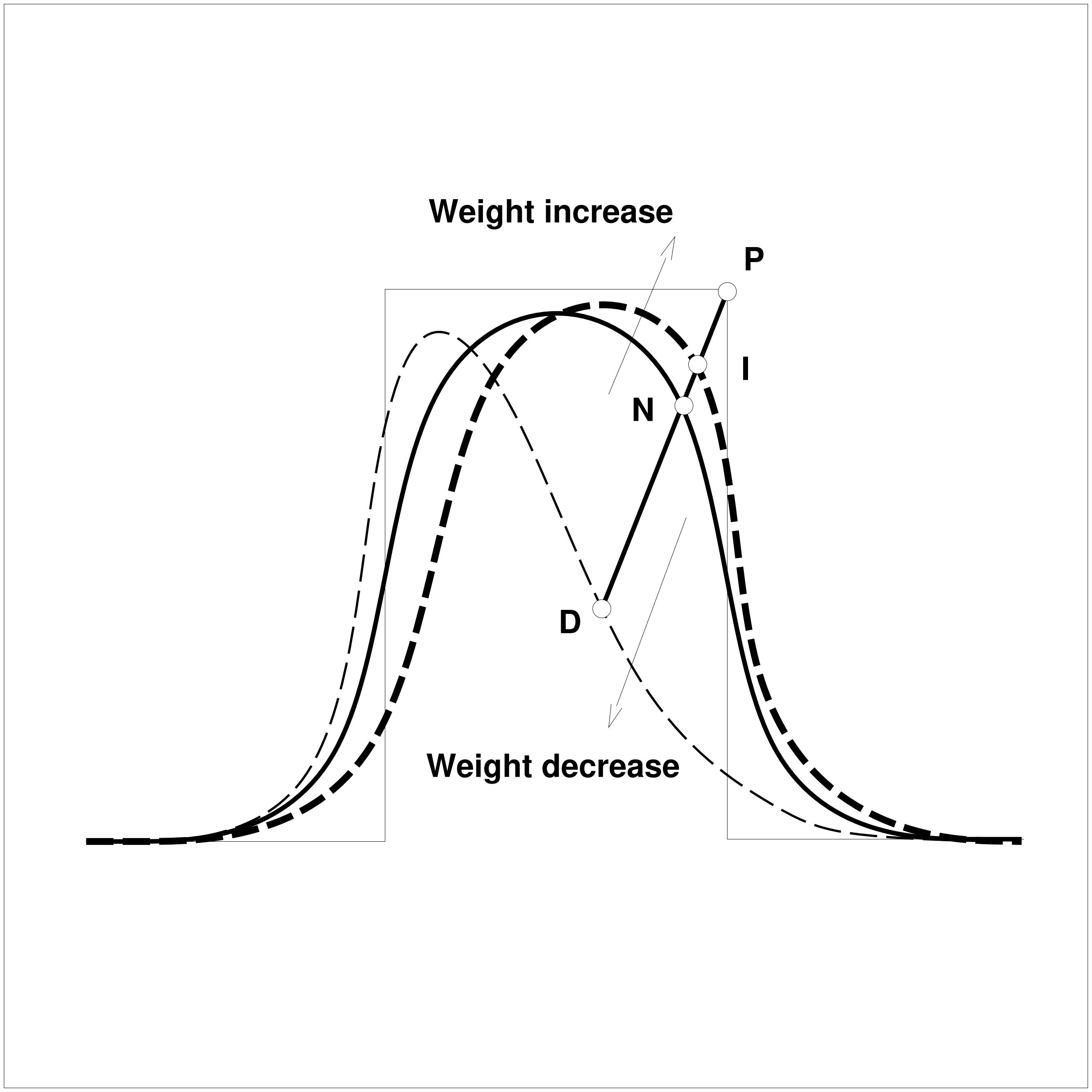
Rational Bezier Curve



From B-splines to NURBS



NURBS Weights



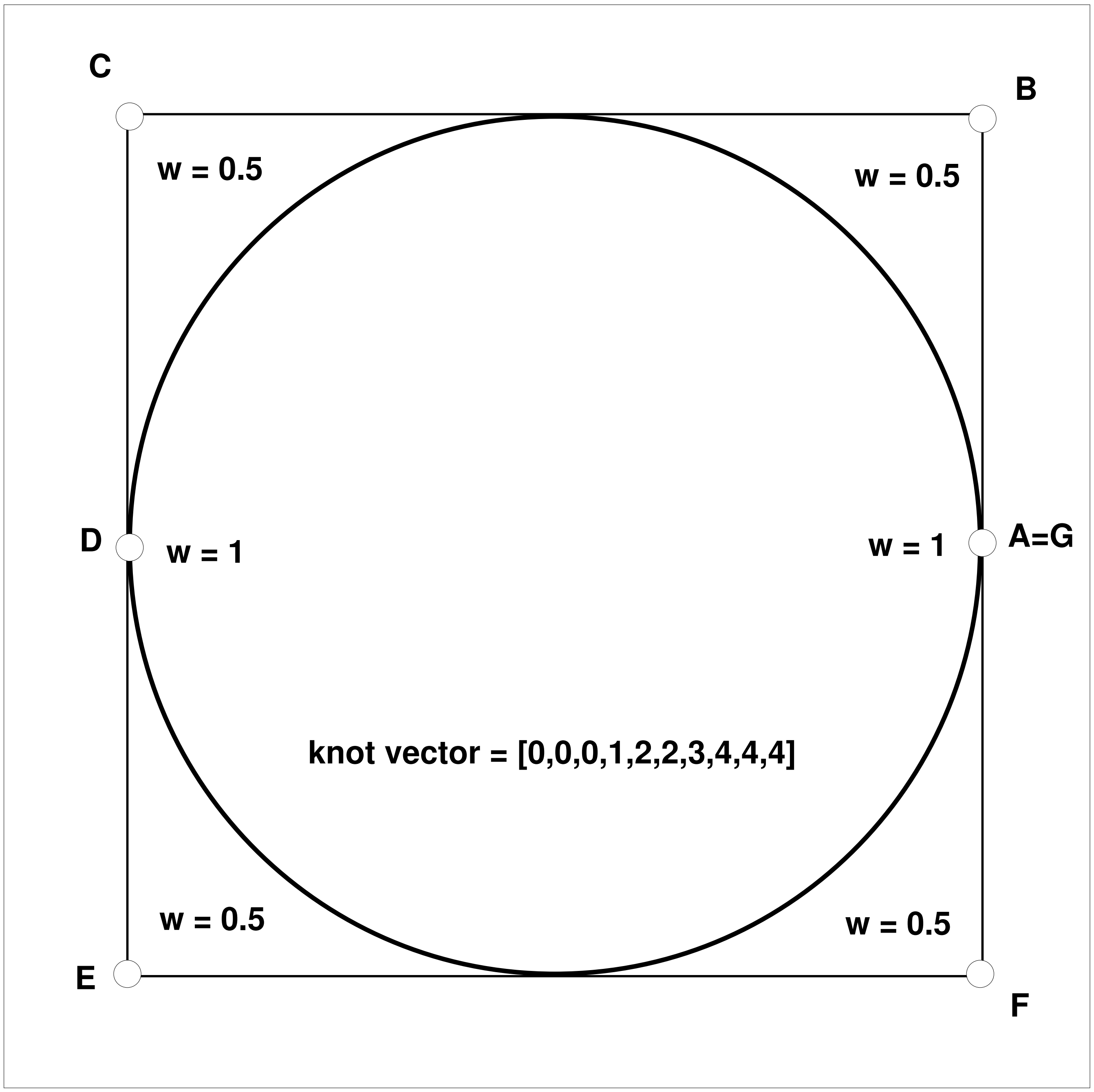
NURBS Properties

- Generalization of B-splines and Bezier splines
- Unified formulation for free-form and analytic shapes
- Weights as extra DOFs
- Various smoothness requirements
- Powerful geometric toolkits
- Efficient and fast evaluation algorithm
- Invariance under standard transformations
- Composite curves
- Continuity conditions

NURBS Properties

- **Generalization of B-splines**
- **Unified formulation for free-form and analytic shapes**
- **Weights as extra DOFs**
- **Various smoothness requirements**
- **Powerful geometric toolkits**
- **Efficient and fast evaluation algorithm**
- **Invariance under standard transformations**

NURBS Circle



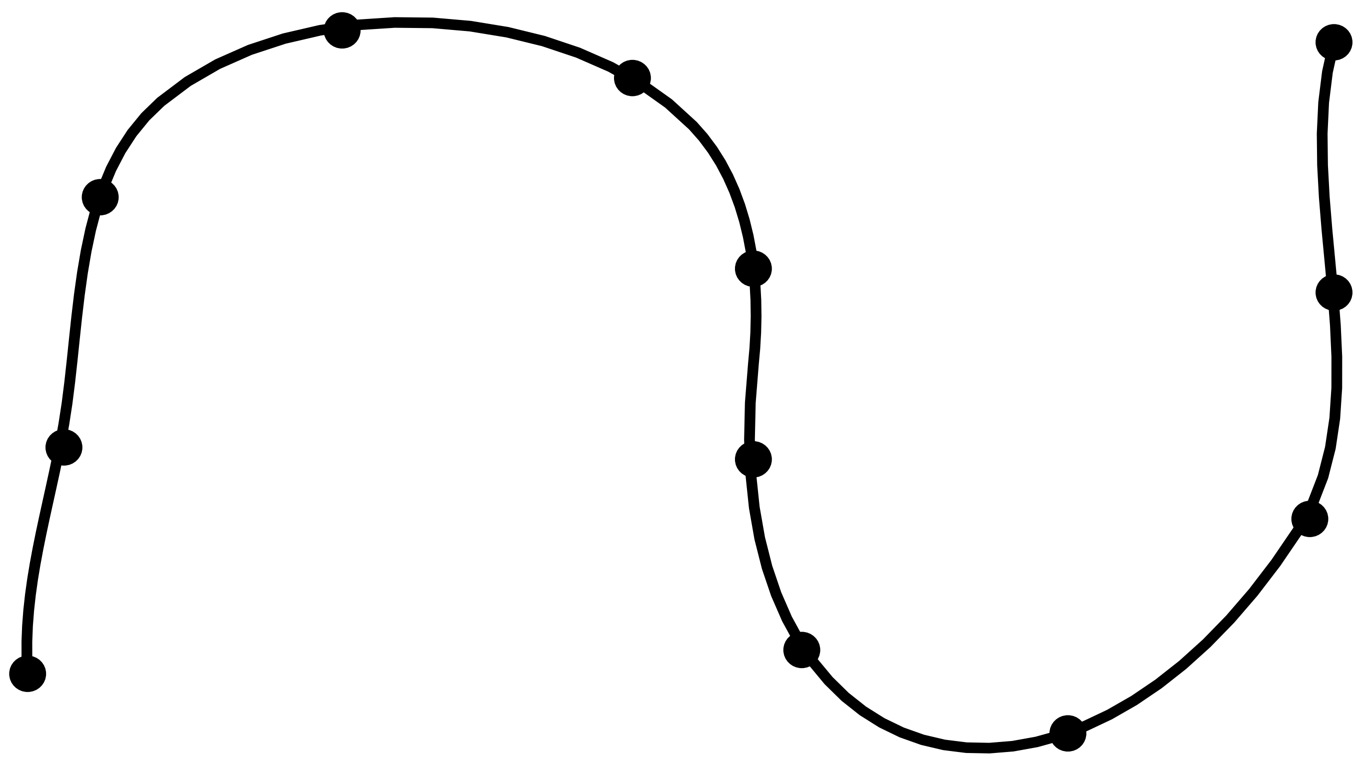
NURBS for Analytic Shapes

- Conic Sections
- Natural quadrics
- Extruded surfaces
- Ruled surfaces
- Surfaces of revolution

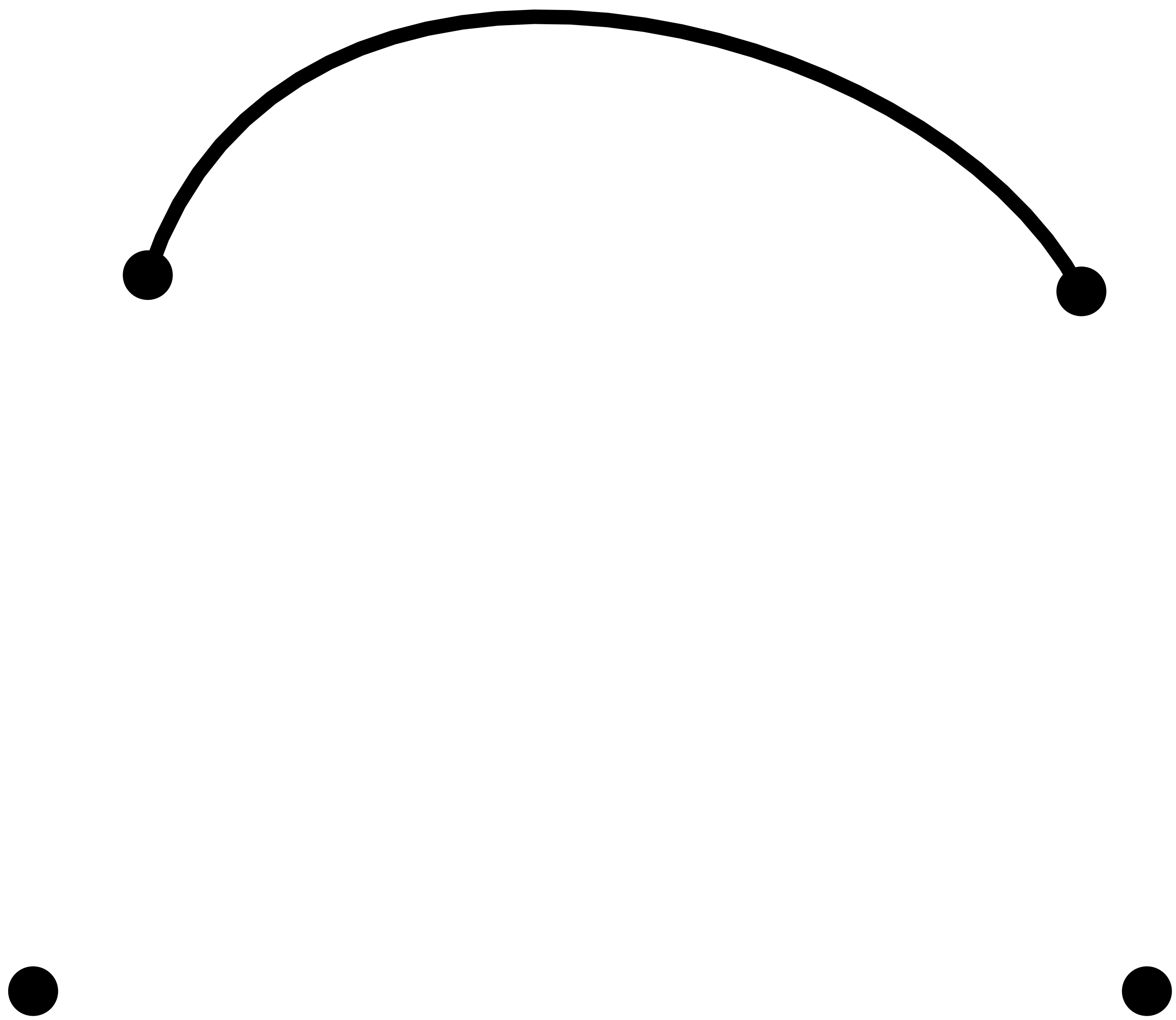
Natural C^2 Cubic Splines

- **A set of piecewise cubic polynomials**
- C^2 **continuity at each vertex**
- $\mathbf{c}_i(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}$

Natural C^2 Cubic Splines



Cardinal Splines



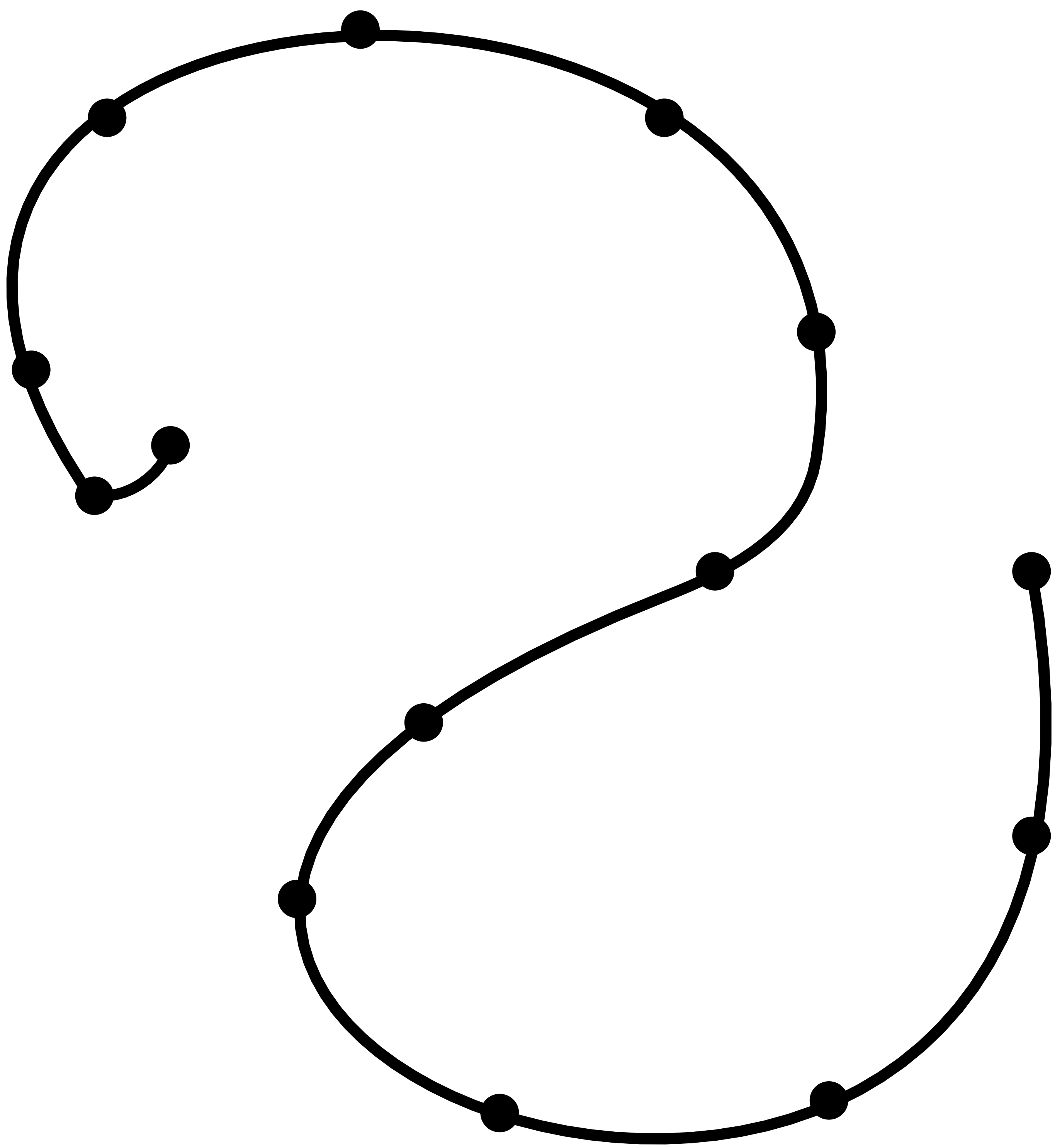
Cardinal Splines

- Four vertices: $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$,
- $\mathbf{c}(0) = \mathbf{v}_1$, $\mathbf{c}(1) = \mathbf{v}_2$,
- $\mathbf{c}^{(1)}(0) = \frac{1}{2}(1-\alpha)(\mathbf{v}_2 - \mathbf{v}_0)$,
- $\mathbf{c}^{(1)}(1) = \frac{1}{2}(1-\alpha)(\mathbf{v}_3 - \mathbf{v}_1)$
- Special case: Catmull-Rom splines when $\alpha = 0$
- More general case: Kochanek-Bartels splines
  - tension, bias, continuity parameters

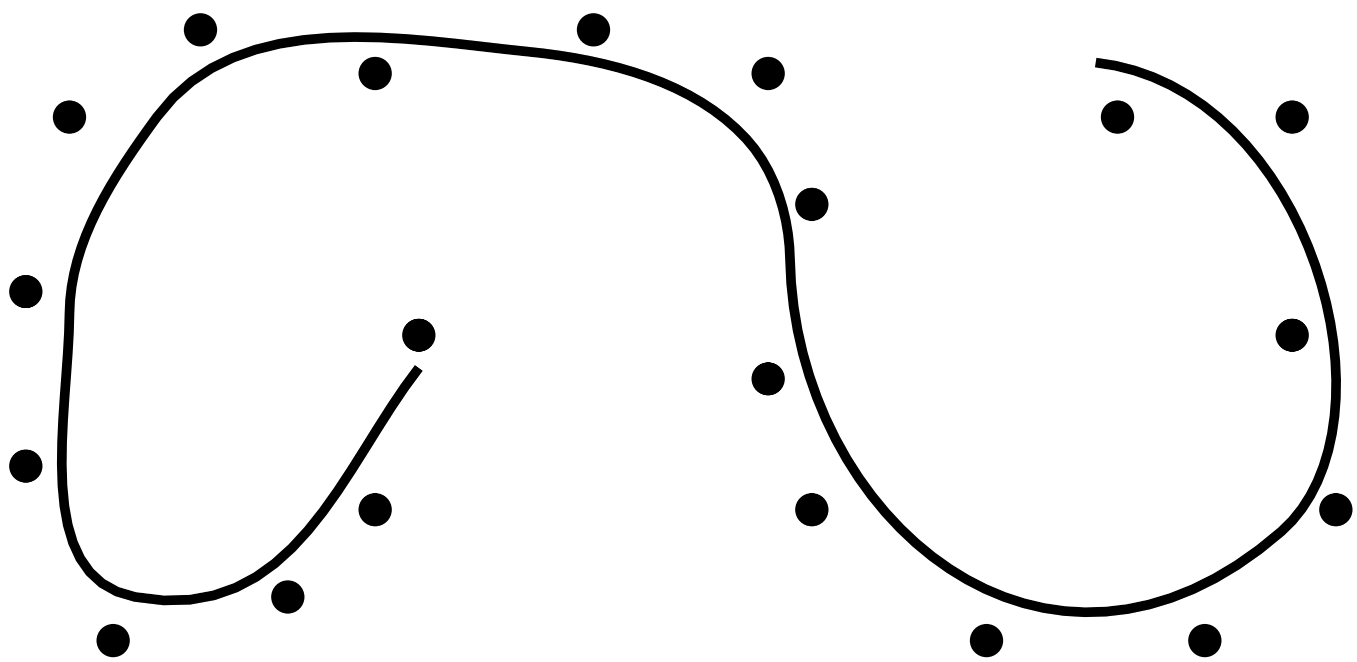
Kochanek-Bartels Splines

- **Four vertices:** $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, **and** $\mathbf{v}_3$
- $\mathbf{c}(0) = \mathbf{v}_1$, $\mathbf{c}(1) = \mathbf{v}_2$,
- $\mathbf{c}^{(1)}(0) =$
  $\frac{1}{2}(1-\alpha)[(1+\beta)(1-\gamma)(\mathbf{v}_1 - \mathbf{v}_0) + (1-\beta)(1+\gamma)(\mathbf{v}_2 - \mathbf{v}_1)]$
- $\mathbf{c}^{(1)}(1) =$
  $\frac{1}{2}(1-\alpha)[(1+\beta)(1+\gamma)(\mathbf{v}_2 - \mathbf{v}_1) + (1-\beta)(1-\gamma)(\mathbf{v}_3 - \mathbf{v}_2)]$
- **Tension parameter:** α
- **Bias parameter:** β
- **Continuity parameter:** γ

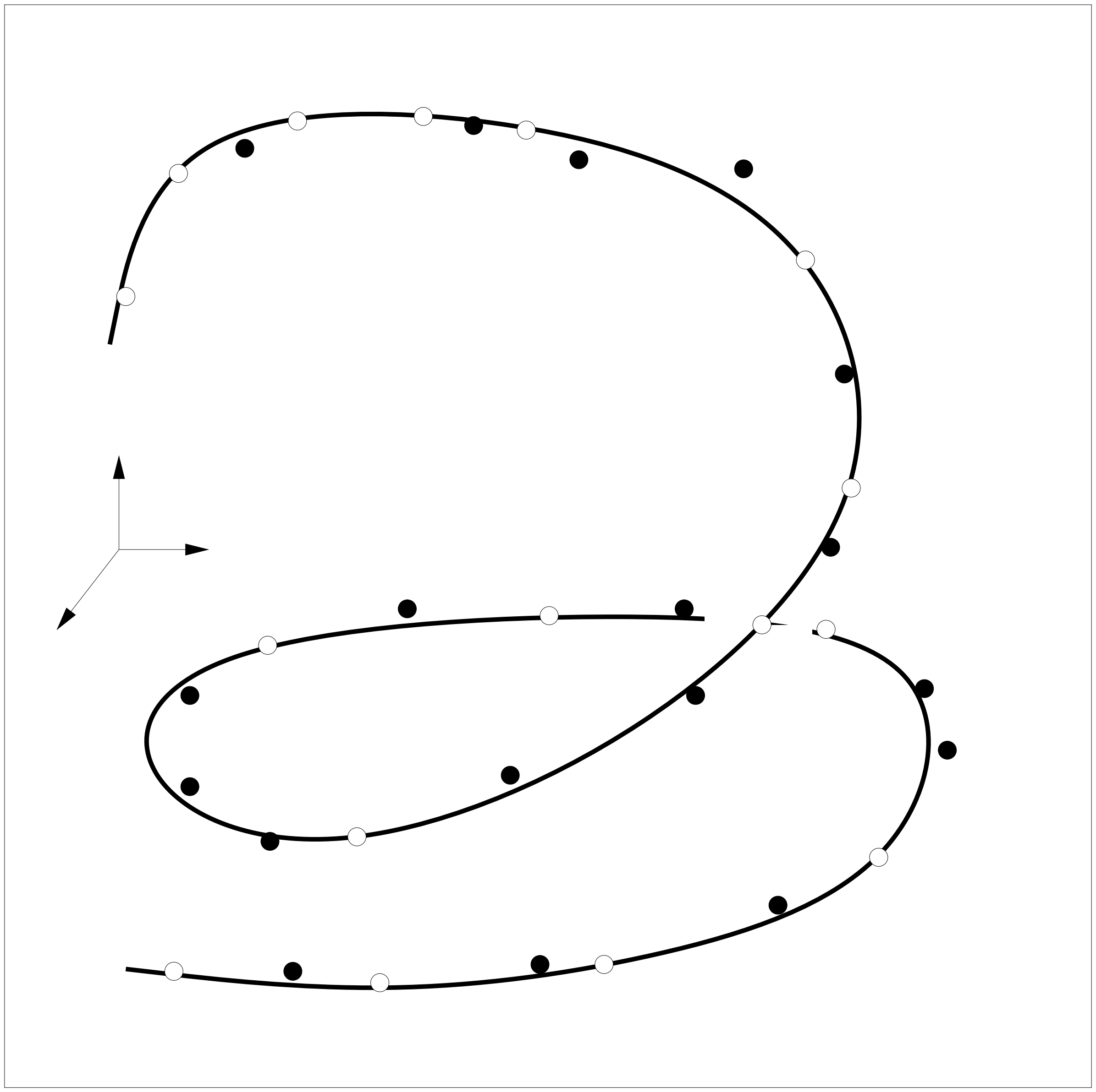
Interpolation Splines



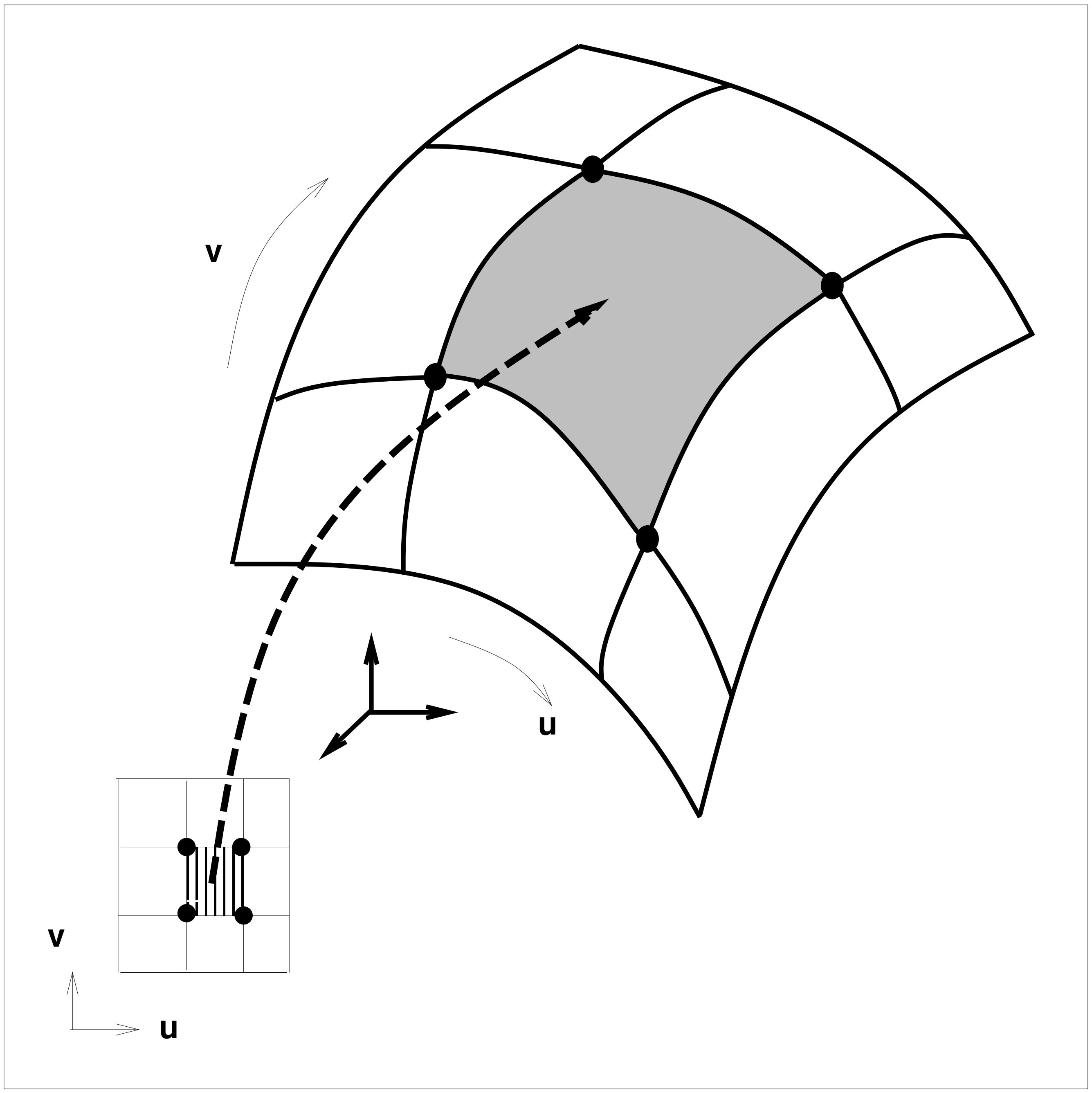
Approximation Splines



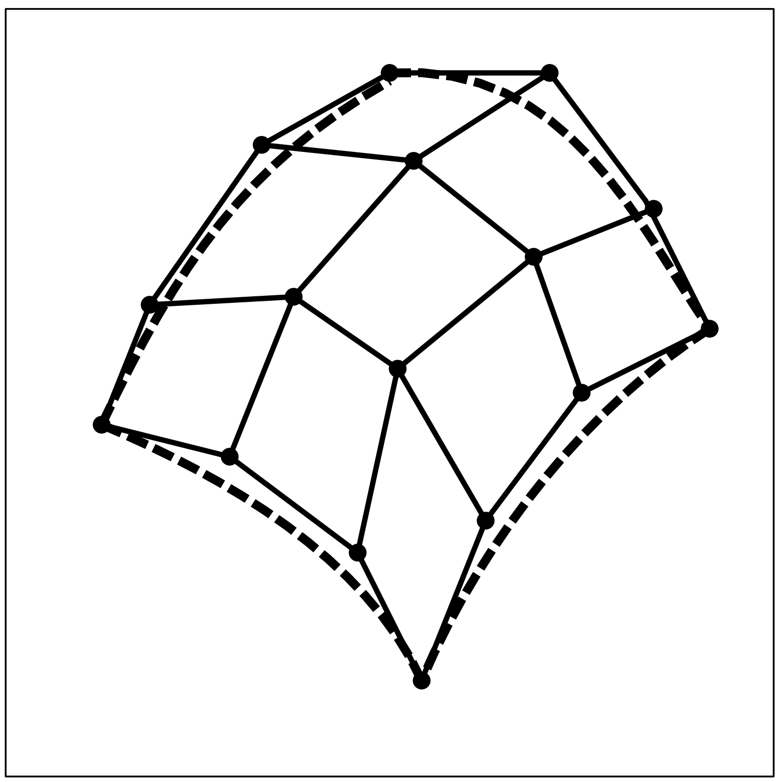
Interpolation/Approximation

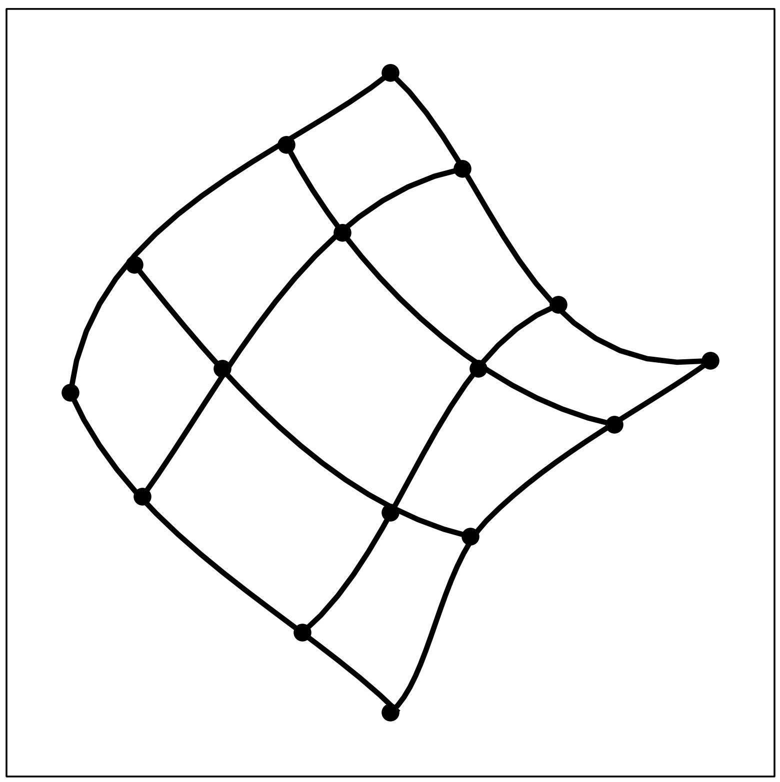


Rectangular Surface



Regular Surfaces





Tensor Product Surfaces

- **Where are they from?**
- **Monomial form**

$$\mathrm{s}(u,v) = \sum_i \sum_j \mathbf{a}_{i,j} u^i v^j$$

- **General case**

$$\mathrm{s}(u,v) = \sum_i \sum_j \mathbf{v}_{i,j} F_i(u) G_j(v)$$

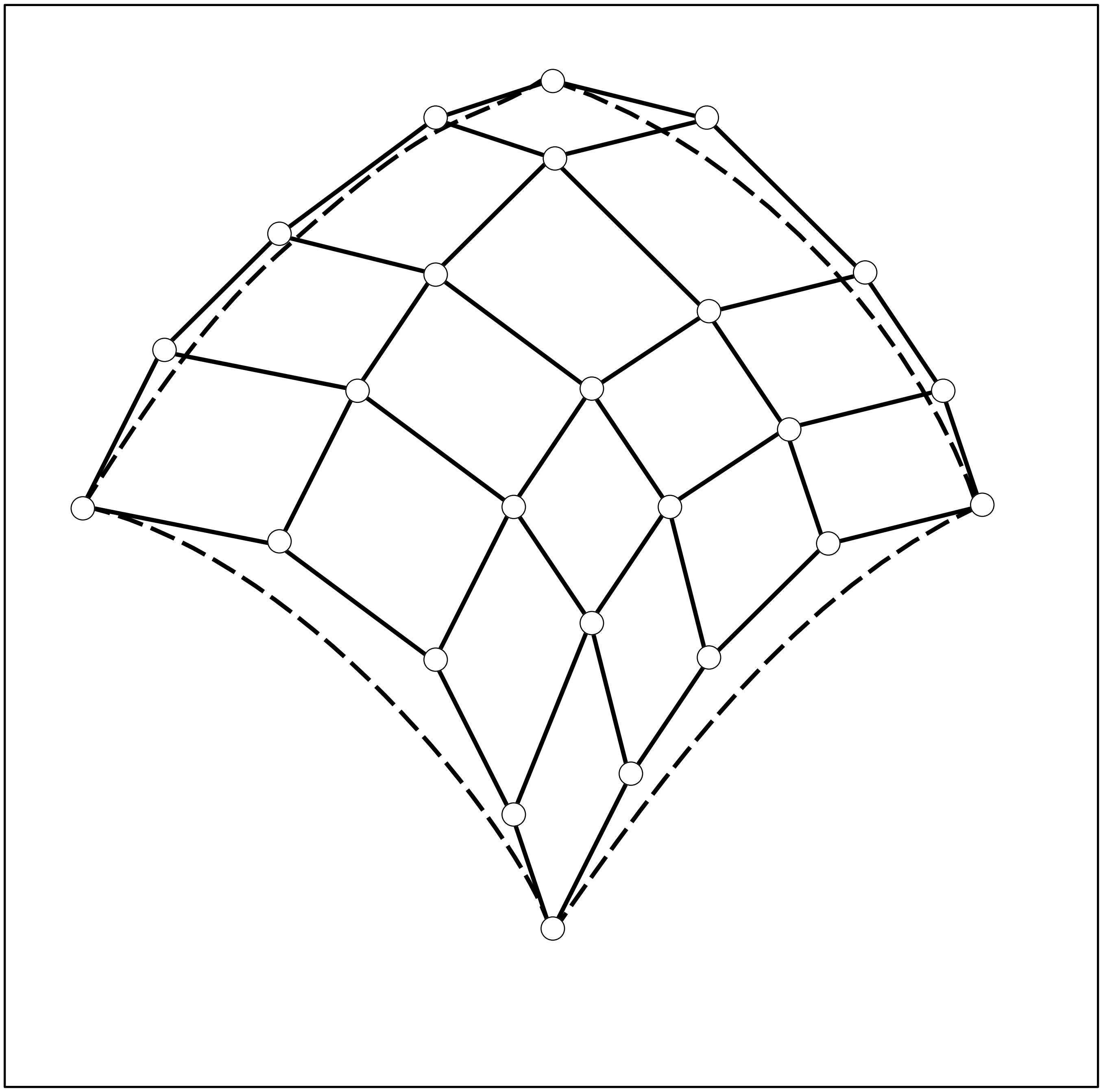
- **Bezier surface**

$$\mathrm{s}(u,v) = \sum_i \sum_j \mathbf{p}_{i,j} B_i^m(u) B_j^n(v)$$

- **B-spline surface**

$$\mathrm{s}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{i,j} B_{i,k}(u) B_{j,l}(v)$$

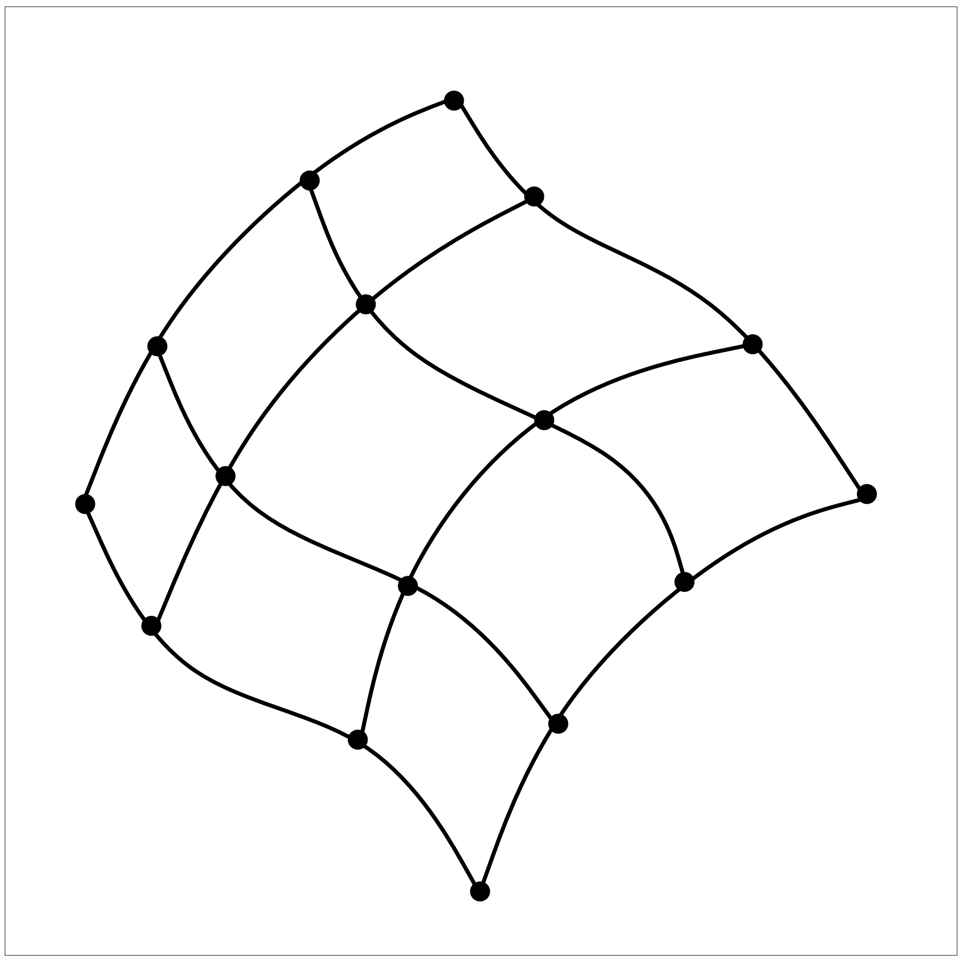
Bezier Surface



Surface Properties

- Inherit from their curve generators
- More!
- Efficient algorithms
- Continuity across boundaries
- Interpolation & approximation tools

Curve Network



Transfinite Methods

- **Bilinearly blended Coons patch**
  - **interpolate four boundary curves**
- **Bicubically blended Coons patch**
- **Gordon surfaces**
  - **interpolate a curve network**
- **Triangular extension**
  - **interpolate over triangles**

Coons Patch

- **Bilinearly blended Coons patch**

$$(P)\mathbf{f} = (P_1 \oplus P_2)\mathbf{f} = (P_1 + P_2 - P_1P_2)\mathbf{f}$$

$$(P_1)\mathbf{f} = \mathbf{f}(0,v)L_0^1(u) + \mathbf{f}(1,v)L_1^1(u)$$

$$(P_2)\mathbf{f} = \mathbf{f}(u,0)L_0^1(v) + \mathbf{f}(u,1)L_1^1(v)$$

- **Bicubically blended Coons patch**

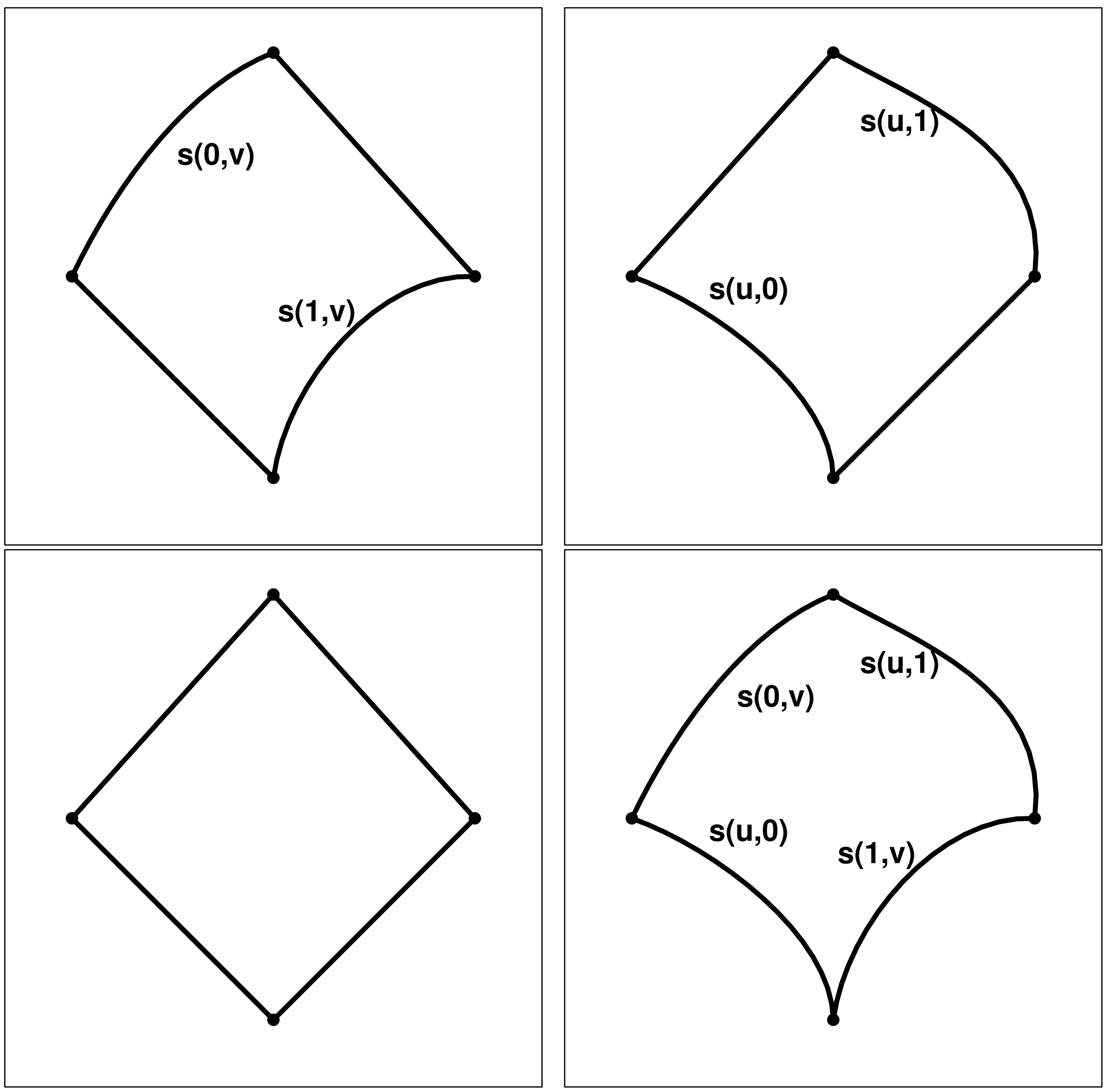
$$(P_1)\mathbf{f} = \mathbf{f}(0,v)H_0^3(u) + \mathbf{f}_u(0,v)H_1^3(u)$$

$$+\mathbf{f}_u(1,v)H_2^3(u) + \mathbf{f}(1,v)H_3^3(u)$$

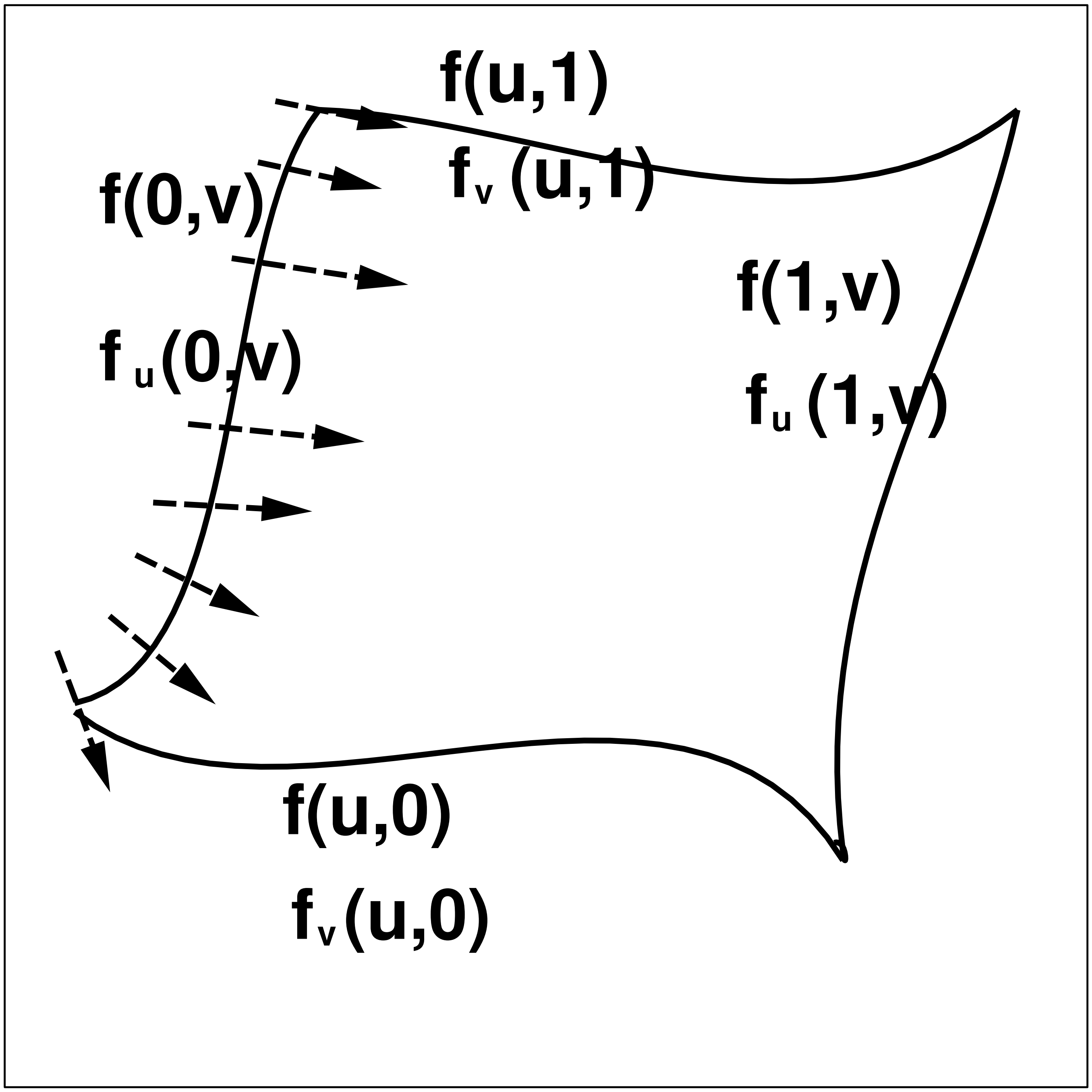
$$(P_2)\mathbf{f} = \mathbf{f}(u,0)H_0^3(v) + \mathbf{f}_v(u,1)H_1^3(v)$$

$$+\mathbf{f}_v(u,1)H_2^3(v) + \mathbf{f}(u,1)H_3^3(v)$$

Coons Patch



Coons Patch



Gordon Surfaces

- **Generalization of Coons techniques**
- **A set of curves**

$$\mathbf{f}(u_i, v), i = 0, \ldots, n$$

$$\mathbf{f}(u, v_j), j = 0, \ldots, m$$

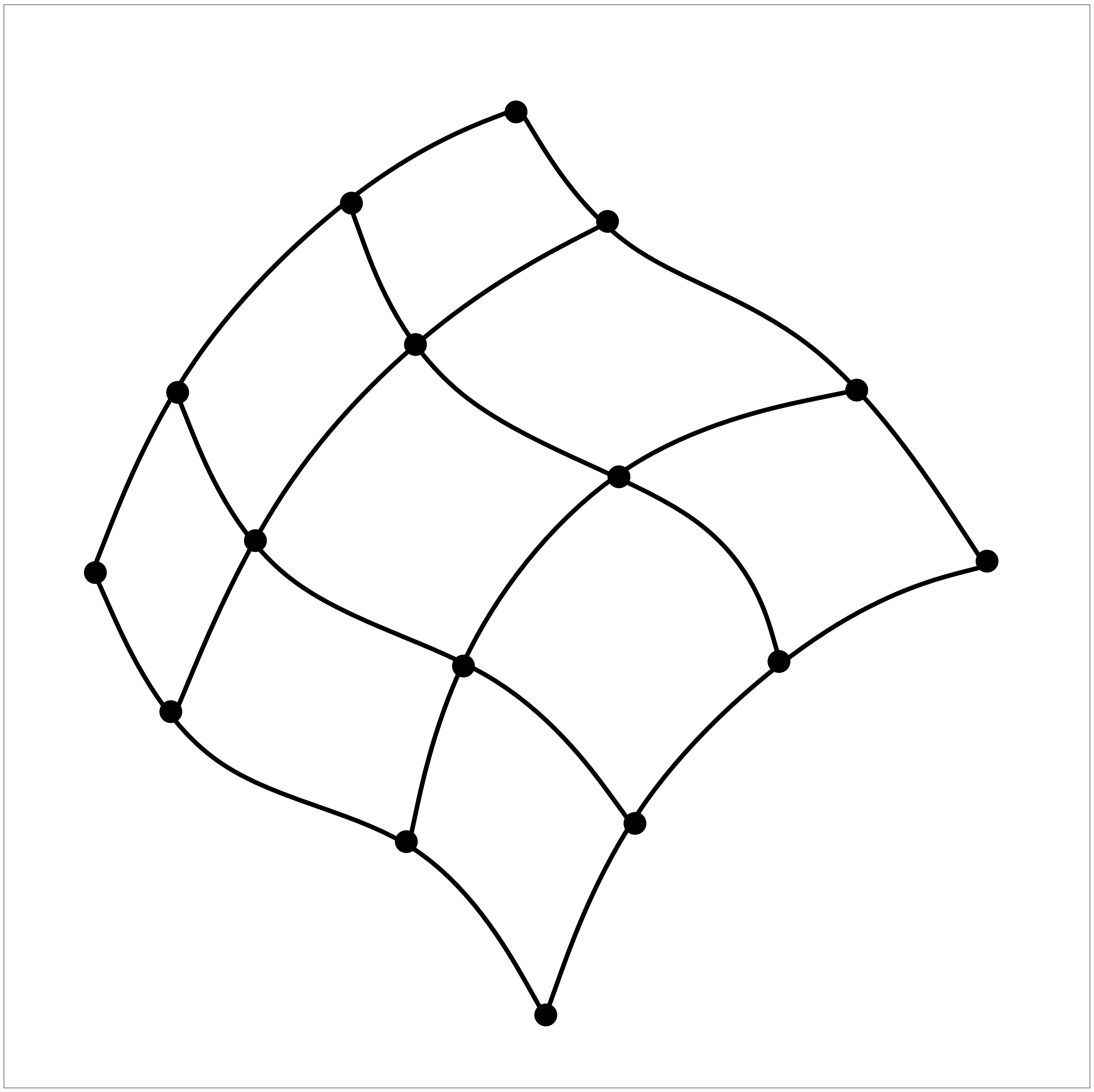
- **Boolean sum using Lagrange polynomials**

$$(P_1)\mathbf{f} = \sum_{i=0}^{n} \mathbf{f}(u_i, v) L_i^n(u)$$

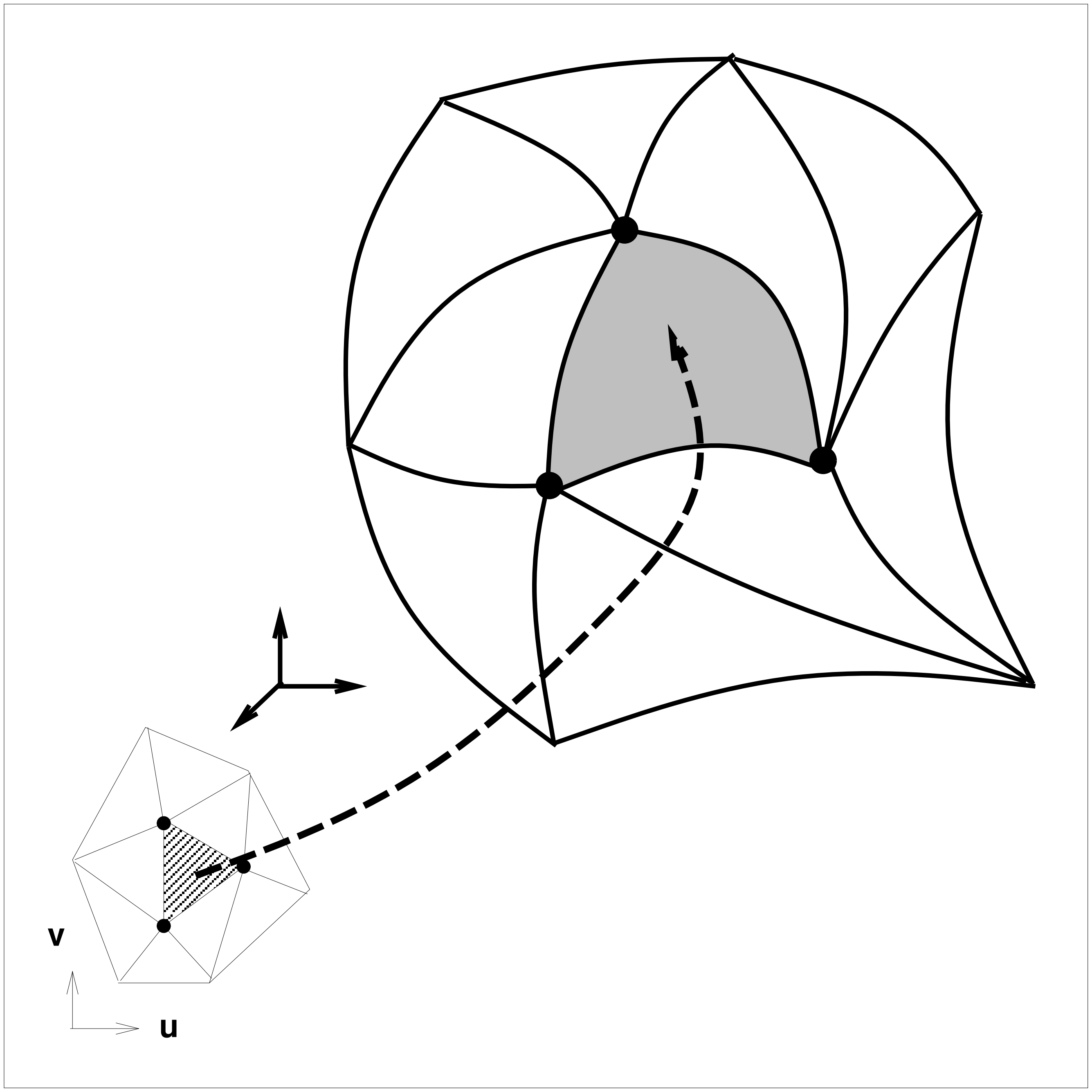
$$(P_2)\mathbf{f} = \sum_{j=0}^{m} \mathbf{f}(u, v_j) L_j^m(v)$$

$$(P)\mathbf{f} = (P_1 \oplus P_2)\mathbf{f} = (P_1 + P_2 - P_1 P_2)\mathbf{f}$$

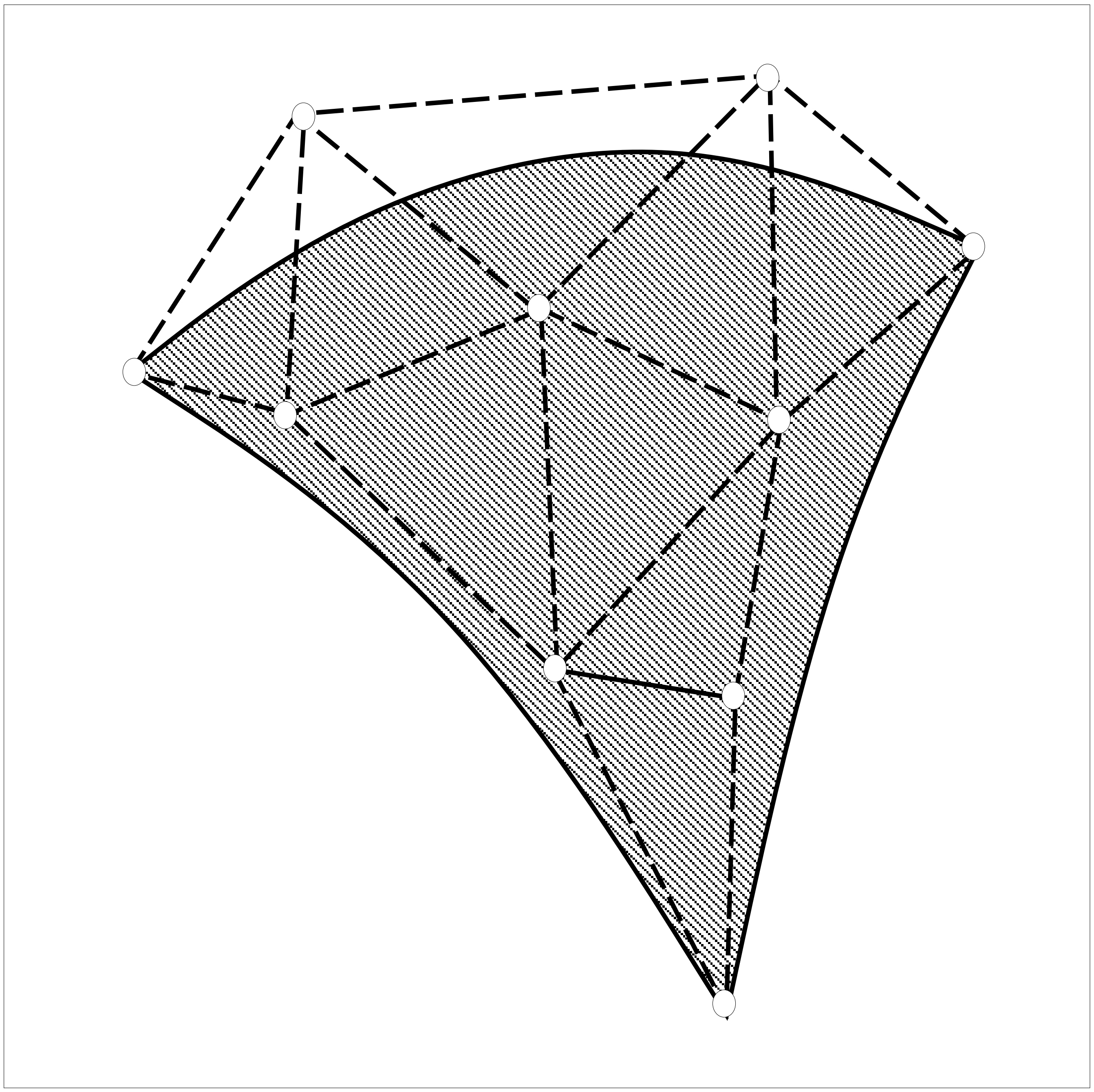
Curve Network



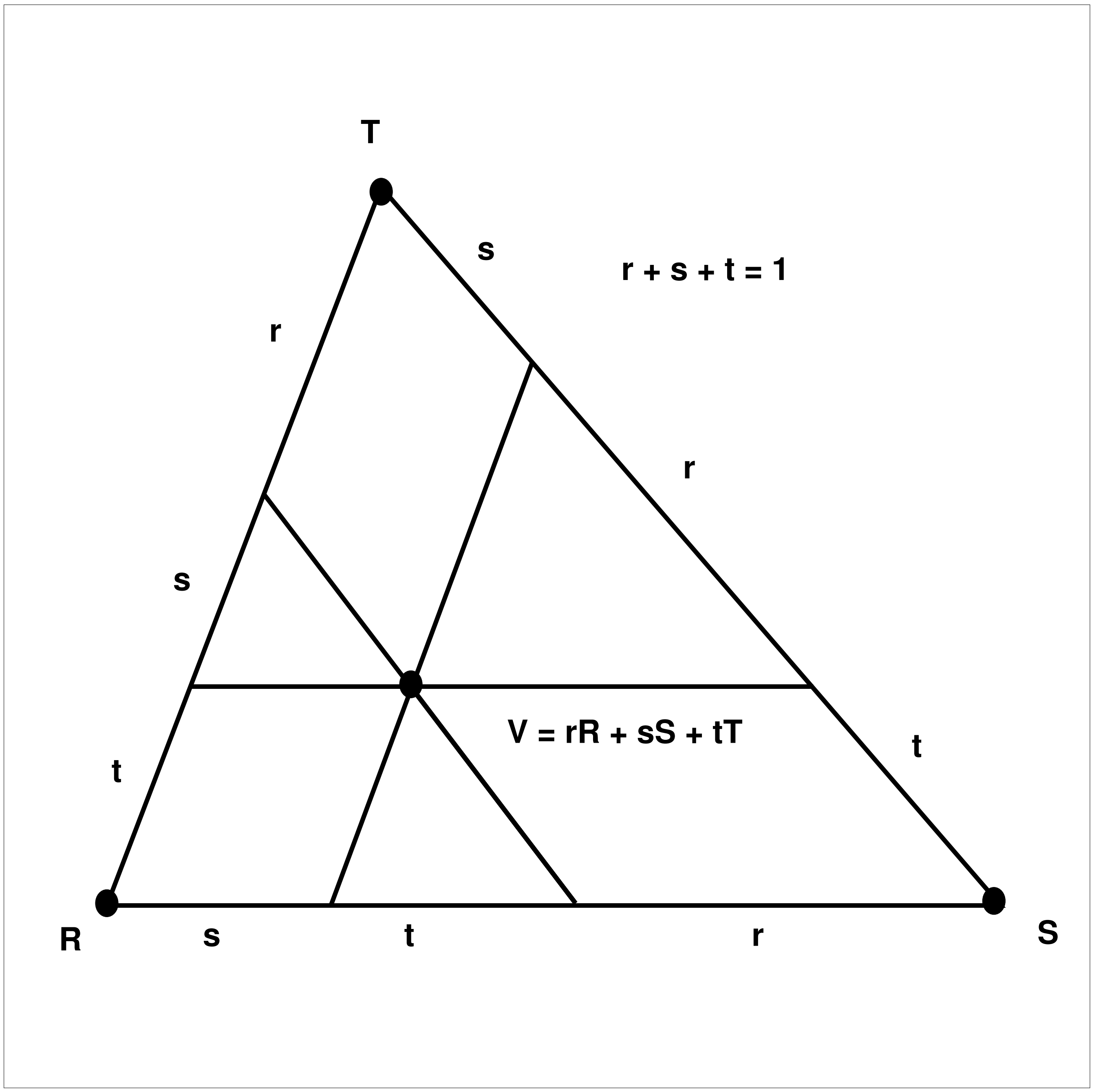
Triangular Surfaces



Triangular Bezier Surface



Barycentric Coordinates



Bezier Formulation

- Triangular Bezier Surface

$$\mathrm{s}(u,v) = \sum_{i,j,k>=0} \mathbf{p}_{i,j,k} B^n_{i,j,k}(r,s,t),$$

  where $i+j+k=n$, $r+s+t=1$, and they are local barycentric coordinates

- Basis functions

$$B^n_{i,j,k}(r,s,t) = \frac{n!}{i!j!k!} r^i s^j t^k$$

- Properties
  - recursive evaluation
  - directional derivatives
  - degree elevation
  - subdivision
  - composite surfaces

Triangular Interpolation

- $(P_1)\mathrm{f} = \mathrm{f}(r,0,t)L_0^1(\alpha) + \mathrm{f}(r,s,0)L_1^1(\alpha)$,
  where $\alpha = \frac{s}{s+t}$

- $(P_2)\mathrm{f} = \mathrm{f}(r,s,0)L_0^1(\beta) + \mathrm{f}(0,s,t)L_1^1(\beta)$,
  where $\beta = \frac{t}{r+t}$

- $(P_3)\mathrm{f} = \mathrm{f}(0,s,t)L_0^1(\gamma) + \mathrm{f}(r,0,t)L_1^1(\gamma)$,
  where $\gamma = \frac{r}{r+s}$

- **The Boolean sum of any two operators results the same!**

$$(P_{12})\mathrm{f} = (P_1 \oplus P_2)\mathrm{f}$$

$$(P_{13})\mathrm{f} = (P_1 \oplus P_3)\mathrm{f}$$

$$(P_{23})\mathrm{f} = (P_2 \oplus P_3)\mathrm{f}$$

- **Use cubic blending functions for C^1 interpolation!**

$$(Q_1)\mathrm{f} = \mathrm{f}(r,0,t)H_0^3(\alpha) + D_\alpha \mathrm{f}(r,0,t)H_1^3(\alpha) + D_\alpha \mathrm{f}(r,s,0)H_2^3(\alpha) + \mathrm{f}(r,s,0)H_3^3(\alpha)$$

$$(Q_2)\mathrm{f} = \ldots$$

$$(Q_3)\mathbf{f} = \dots$$

- **Convex combination (Gregory's method)**

$$(T_1)\mathbf{f} = \mathbf{f}(r, 0, t) + \alpha D_\alpha \mathbf{f}(r, 0, t)$$

$$(T_2)\mathbf{f} = \dots$$

$$(T_3)\mathbf{f} = \dots$$

$$(T_{12})\mathbf{f} = (T_1 \oplus T_2)\mathbf{f}$$

$$(T_{13})\mathbf{f} = \dots$$

$$(T_{23})\mathbf{f} = \dots$$

$$(T)\mathbf{f} = (a_1 T_{23} + a_2 T_{13} + a_3 T_{12})\mathbf{f}$$

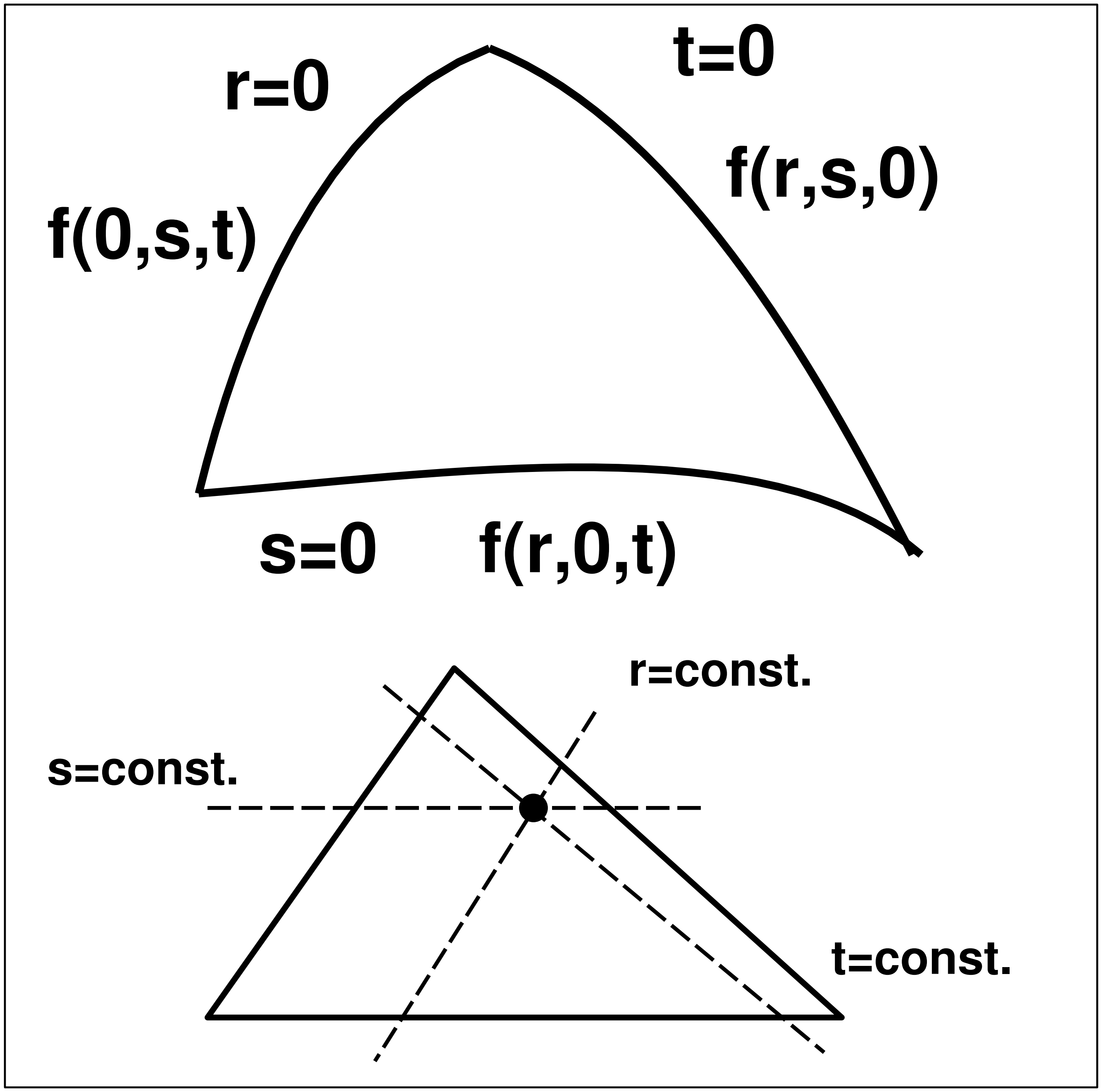
$$a_1 = \frac{s^2}{r^2 + s^2 + t^2}$$

$$a_2 = \dots$$

$$a_3 = \dots$$

- **Pentagonal patch!**

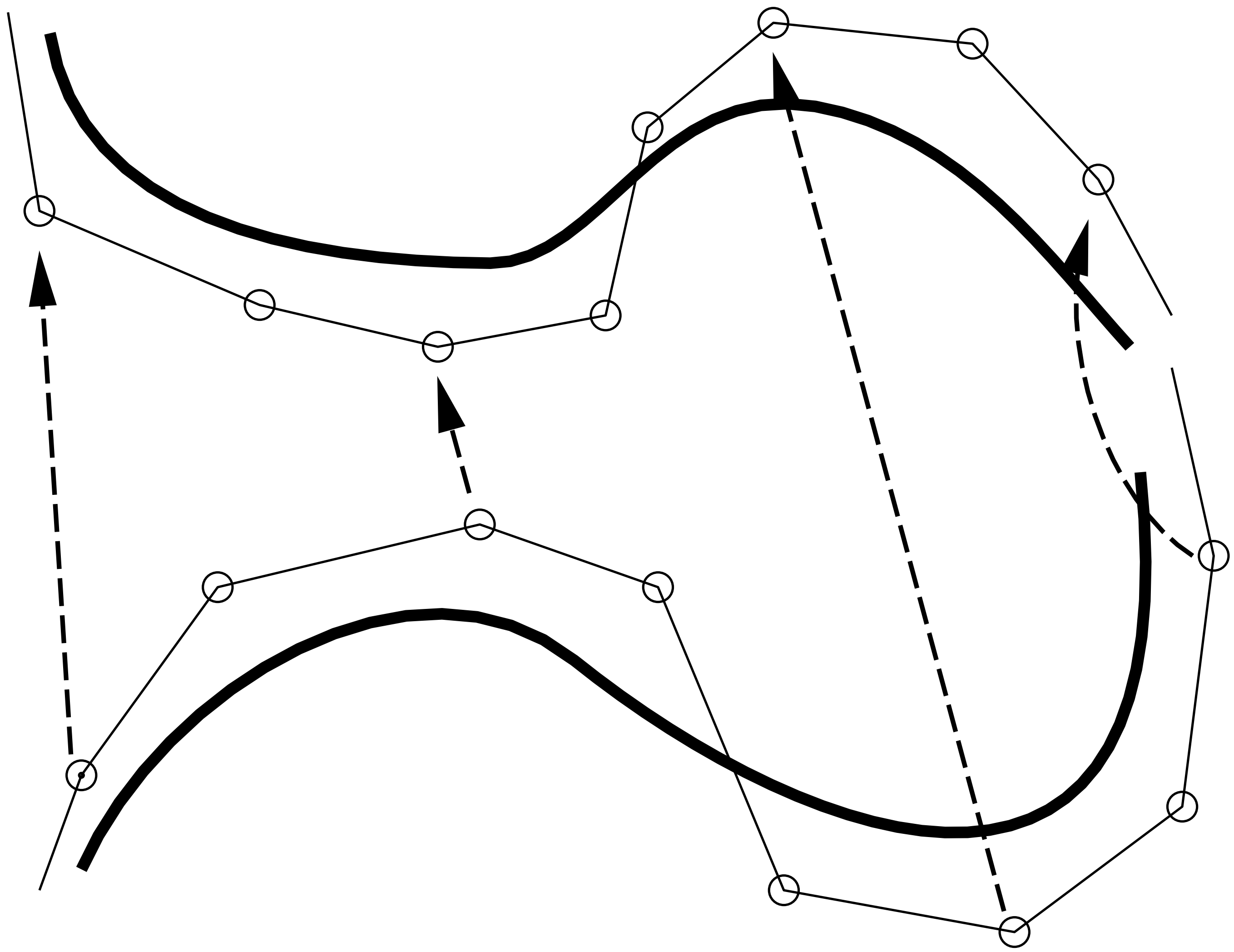
Triangular Coons-Gordon Surface



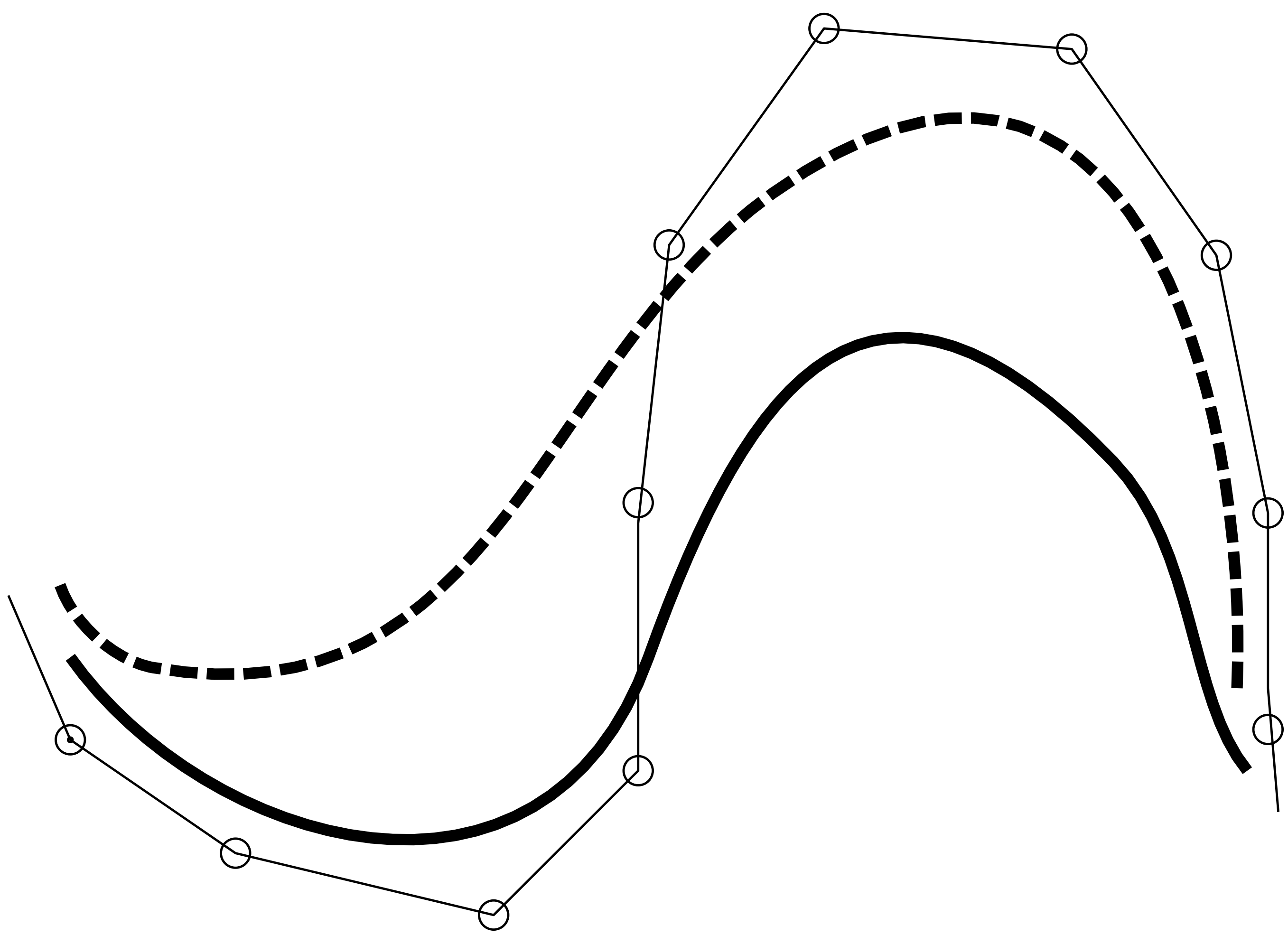
Surface Summary

- How many ways to construct surfaces???
- Multiple patches
- “Carpet method”
- A set of boundary curves
- Subdivision
- Others!

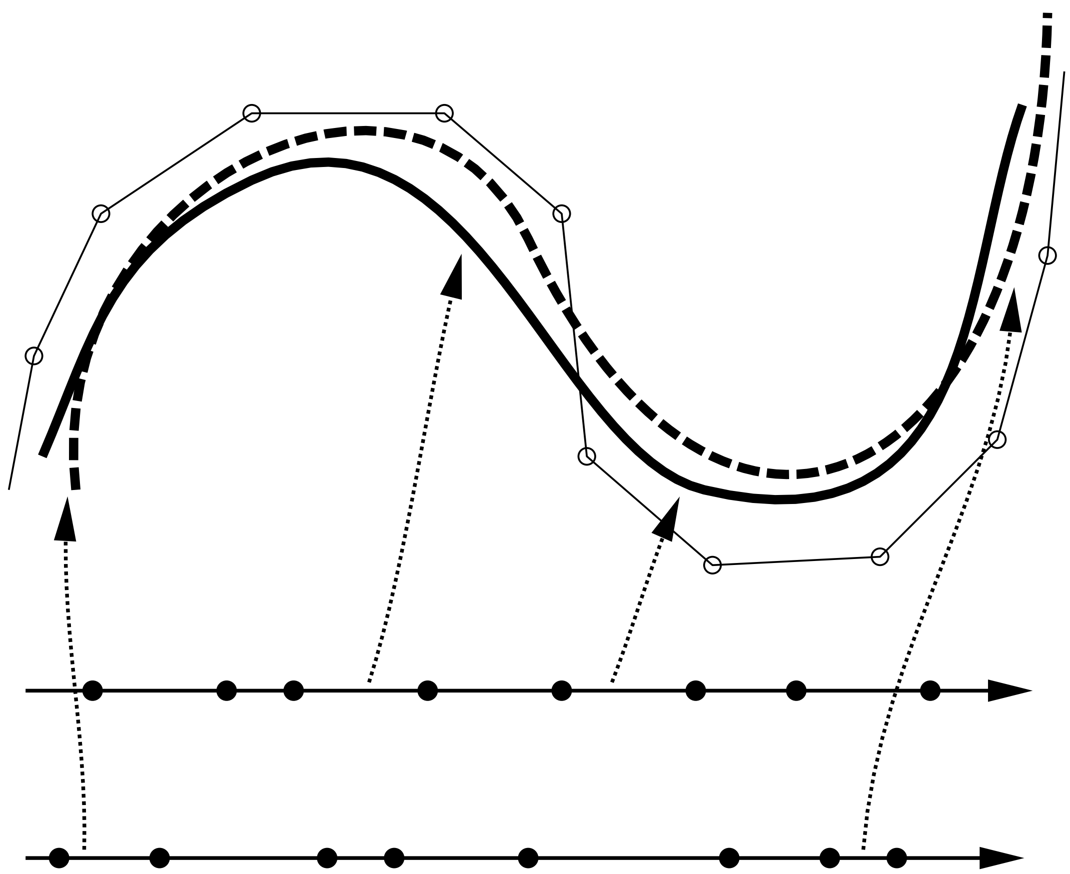
Control Point Manipulation



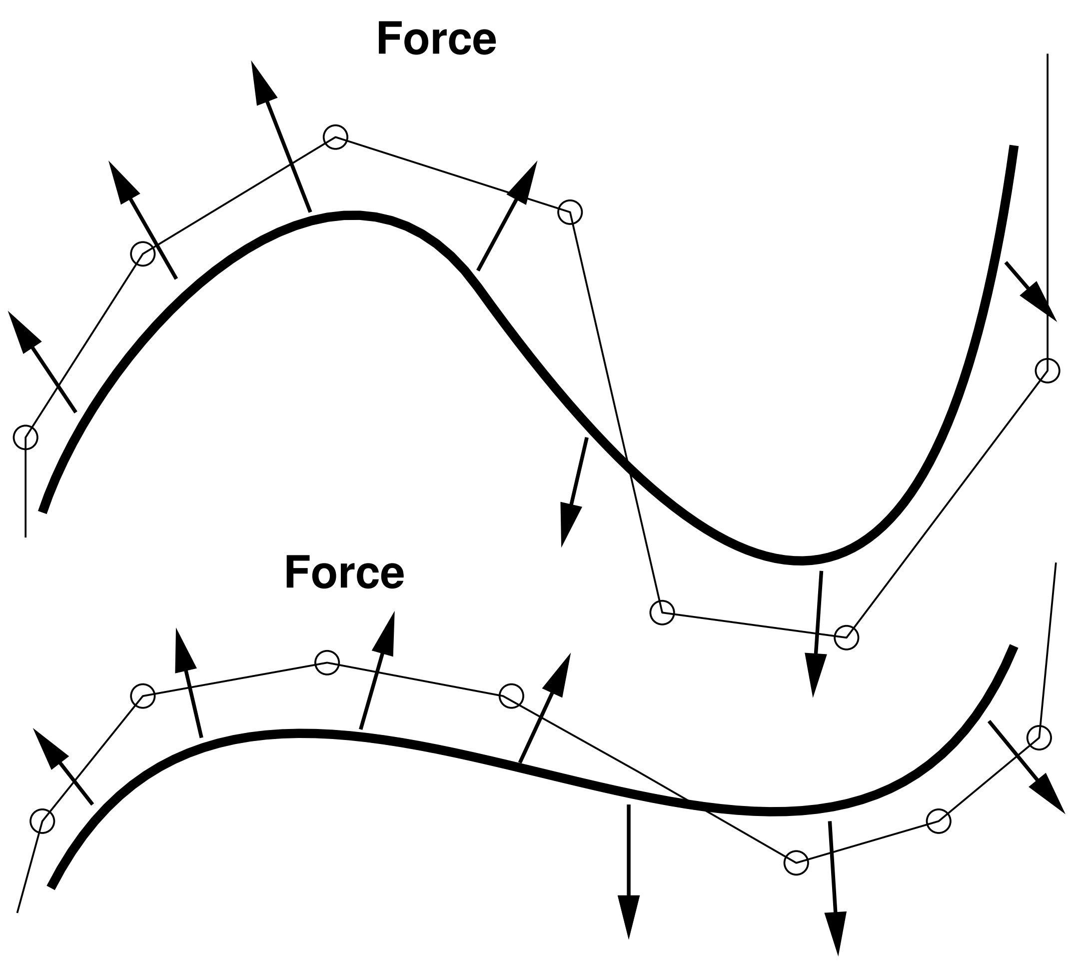
Weight Modification



Knot Vector Variation



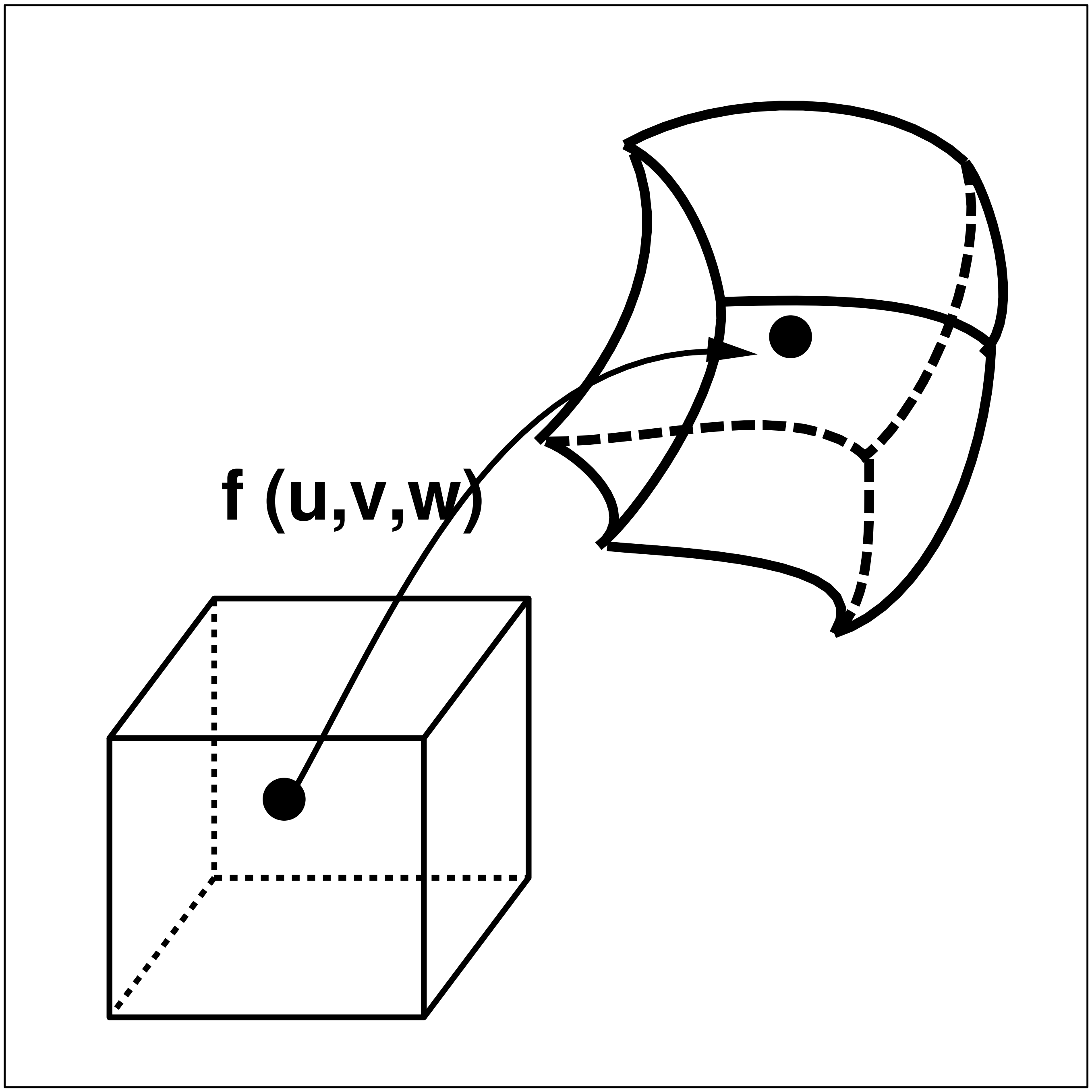
Physics-Based Modeling



Geometric Modeling Techniques

- Interpolation/Approximation
  - regular dataset
  - scattered dataset
  - curve network
- Optimization
- Interaction
- Constraints
- Continuity
- Differential geometry
- Other advanced operations
  - blending, rounding, etc

Solid



What Will Be Covered Next?

- **Deformable models**
- **"Deformable models",**
  D. Terzopoulos and K. Fleischer
  The Visual Computer, **4, 306-331, 1988**
- **Related publications: Terzopoulos's Siggraph'87 and Siggraph'88 papers!**