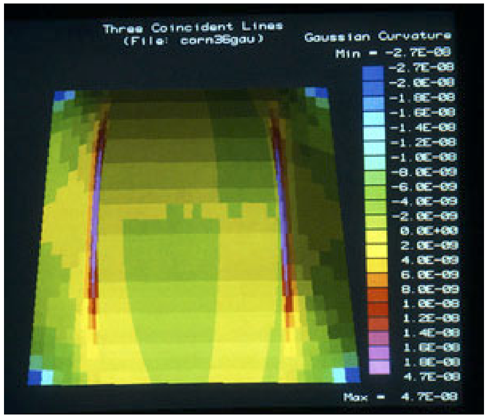
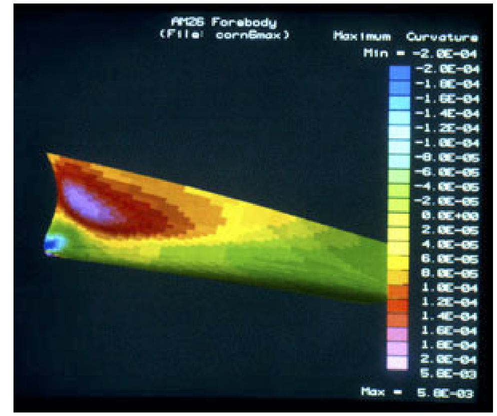



# Course 31
# NURBS
# (NonUniform Rational B-splines)
# a Primer

## Course Notes for SIGGRAPH 2003
## San Diego, California
## Tuesday, 29 July 2003

**David F. Rogers**
**Professor of Aerospace Engineering**
**United States Naval Academy**
**Annapolis, Maryland**

## Course Summary

A working knowledge of the underlying mathematics of NURBS is provided. An engineering approach is taken, i.e., just enough mathematics is presented to get the job done. We will concentrate on the fundamentals with the final goal the development of a fast rational B-spline (NURBS) surface algorithm

## Topics

Bézier Curves
Nonrational B-spline Curves
Rational B-spline (NURBS) Curves
Bézier Surfaces
Nonrational B-spline Surfaces
Rational B-spline (NURBS) Surfaces
A Fast NURBS Algorithm

# Speaker Biography

Dave Rogers is the author of the computer graphics classics, *Mathematical Elements for Computer Graphics* and *Procedural Elements for Computer Graphics* as well as *An Introduction to NURBS, With Historical Perspective*. He is also the coeditor of four books from the state-of-the-art series on computer graphics and has published two fluid dynamics texts – *Laminar Flow Analysis* and *Computer Aided Heat Transfer Analysis*. His books have been translated into six foreign languages.

Dr. Rogers was founder and former director of the Computer Aided Design/Interactive Graphics Group at the United States Naval Academy. His early classic work in the use of B-splines and NURBS for dynamic real-time manipulation of ship hull surfaces spawned both commercial and research programs. His early work was featured in the SIGGRAPH film *The Story of Computer Graphics*.

He was series editor for the Springer-Verlag series *Monographs in Visualization* and a founding editor of the journal *Computers & Education*. He is also a member of the editorial boards of *The Visual Computer* and *Computer Aided Design*.

He was the Fujitsu Scholar at the Royal Melbourne Institute of Technology in Australia, where he helped design and establish the computer graphics laboratory. He was also a Visiting Professor at the University of New South Wales in Australia, where he lectured on computer graphics. He studied naval architecture with the Royal Corps of Naval Constructors while an Honorary Research Fellow at University College London in England.

Professor Rogers was one of the original faculty who established the Aerospace Engineering Department at the United States Naval Academy in 1964. He is currently Professor of Aerospace Engineering and Director of Aeronautics at the Academy.

Kevin Sharer, CEO of Amgen and former student of Professor Rogers, recently endowed the David F. Rogers Chair of Aerospace Engineering at the United States Naval Academy in his honor.

Dave Rogers has both an experimental and a theoretical research background. His research interests include highly interactive graphics, computer aided design and manufacturing, numerical control, computer aided education, hypersonic viscous flow, boundary layer theory, computational fluid mechanics and flight dynamics.

He is an active pilot and holds an ATP (Air Transport Pilot) rating. He is chief pilot for the flight test course at the Academy. He has flown extensively throughout the Canadian High Arctic, including to Alert at 82 degrees 30 minutes north; across the North Atlantic to Iceland, Norway, Scotland and Ireland; to Alaska; and throughout the Bahamas and the Caribbean. His photographs of the Canadian High Arctic have been featured in a photography art show. Dave frequently flies his Bonanza to SIGGRAPH. He holds a Ph.D., as well as Bachelor's and Master's degrees, in aeronautical engineering from Rensselaer Polytechnic Institute.

## Schedule and Contents

**8:30 Bèzier and B-Spline Curves**

The Genesis of NURBS (10 mins)
- A Bit of History

Bèzier Curves (15 mins)
- Why study Bèzier curves?
- Definition
- Bernstein Basis Functions
- Examples
- Continuity

B-spline curves (30 mins)
- A Bit of History
- Definition and Characteristics
- Convex Hulls
- Knot Vectors
- Basis Functions
- Controls
- Examples

General Questions (5 mins)

**9:30 Advanced B-Spline Curves**

Rational B-spline Curves (NURBS) (35 mins)
- A Bit of History
- Definition and Characteristics
- Rational B-Spline Basis Functions
- Examples
- Conic Sections
- Circles
- Examples

General Questions (5 mins)

**10:15 Break** (15 mins)

**10:30 Bèzier and B-Spline Surfaces**

Bèzier Surfaces (30 mins)
- A Bit of History
- Definition and Characteristics
- Controls
- Examples

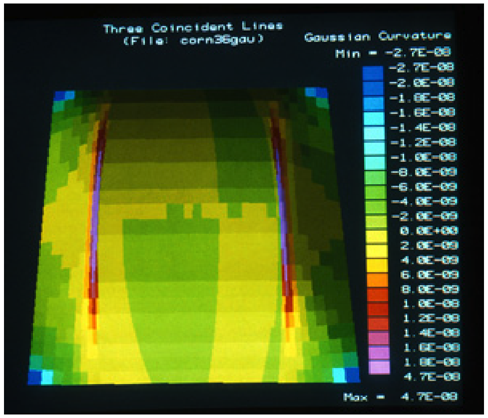
B-Spline Surfaces (30 mins)
- Definition and Characteristics
- Convex Hulls
- Local Control
- Examples

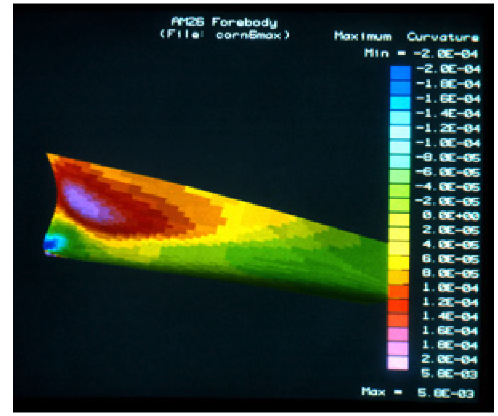
General Questions (5 mins)

**11:35 Rational B-Spline Surfaces (NURBS)**

- Rational B-Spline Surfaces (NURBS) (30 mins)
  - A Bit of History
  - Definition and Characteristics
  - Homogeneous Weight Effects
  - A Simple Algorithm
  - A Fast Rational B-Spline Algorithm
    - A Naive Algorithm
    - A More Efficient Algorithm
    - The Incremental Algorithm
- General Questions (10 mins)

**12:15 Course Ends**




# NURBS

# (NonUniform Rational B-splines)

# a Primer

David F. Rogers

Professor of Aerospace Engineering

United States Naval Academy

# Bézier's Contribution

# How it was

# Cubic Splines

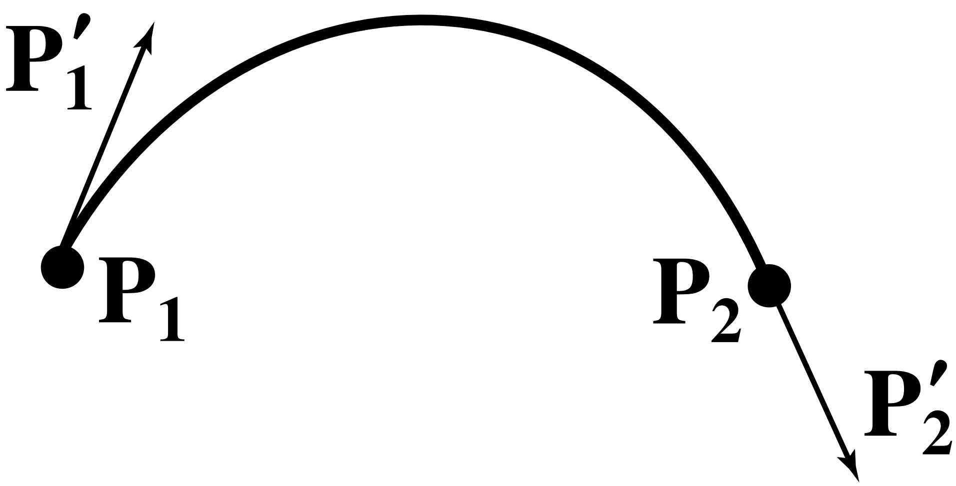
**Late 1960s early 1970s**

**Standard for fairing**

- **Ship hulls**
- **Automobile bodies**
- **etc.**

# Cubic Splines

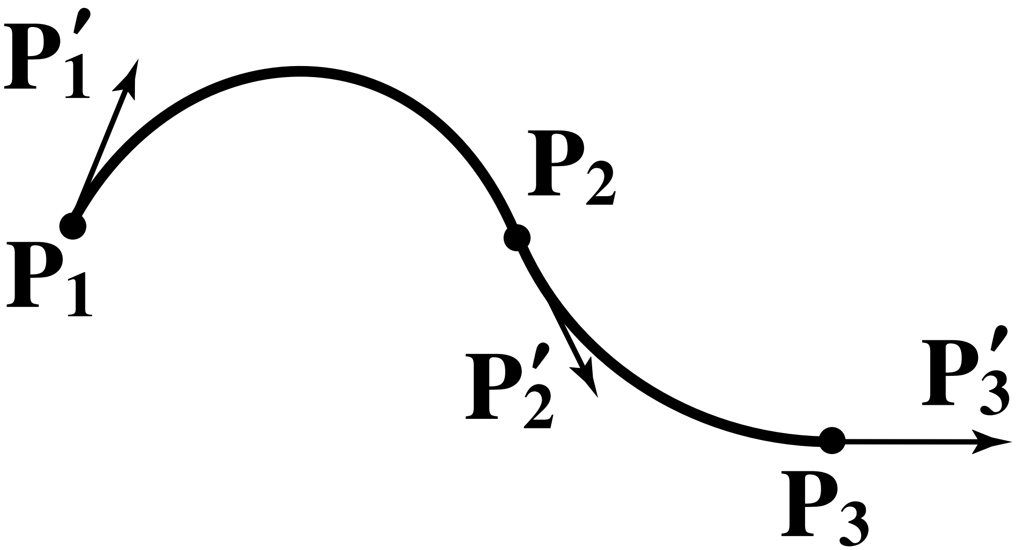
Data needed




2 End points

2 Slopes (tangent vectors)

# Cubic Splines



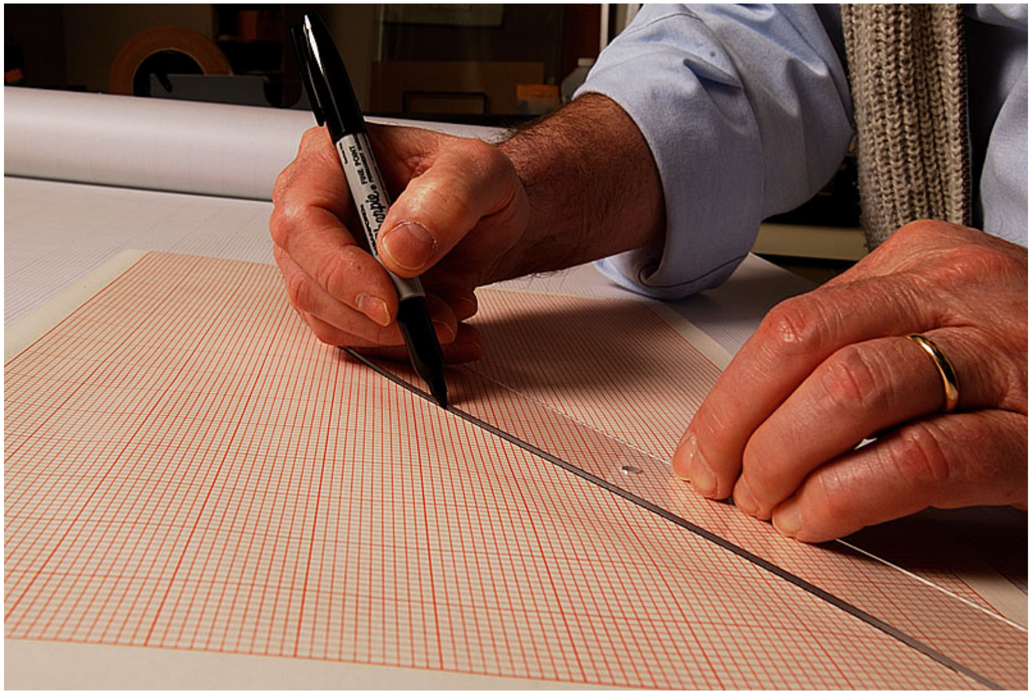

**Can calculate $P_2'$**

**Curvature continuity at $P_2$**

# Cubic Splines

## Problem – wiggles

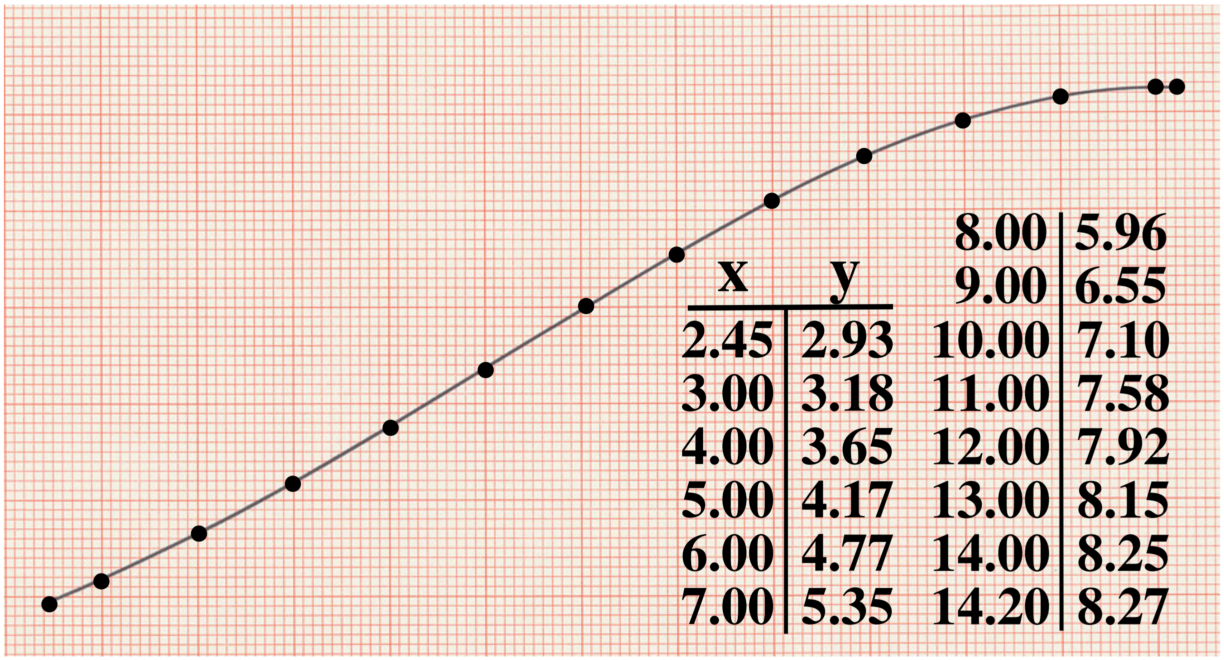
# Cubic Splines



## Creating smooth curve

# Cubic Splines

## Digitize points along the curve

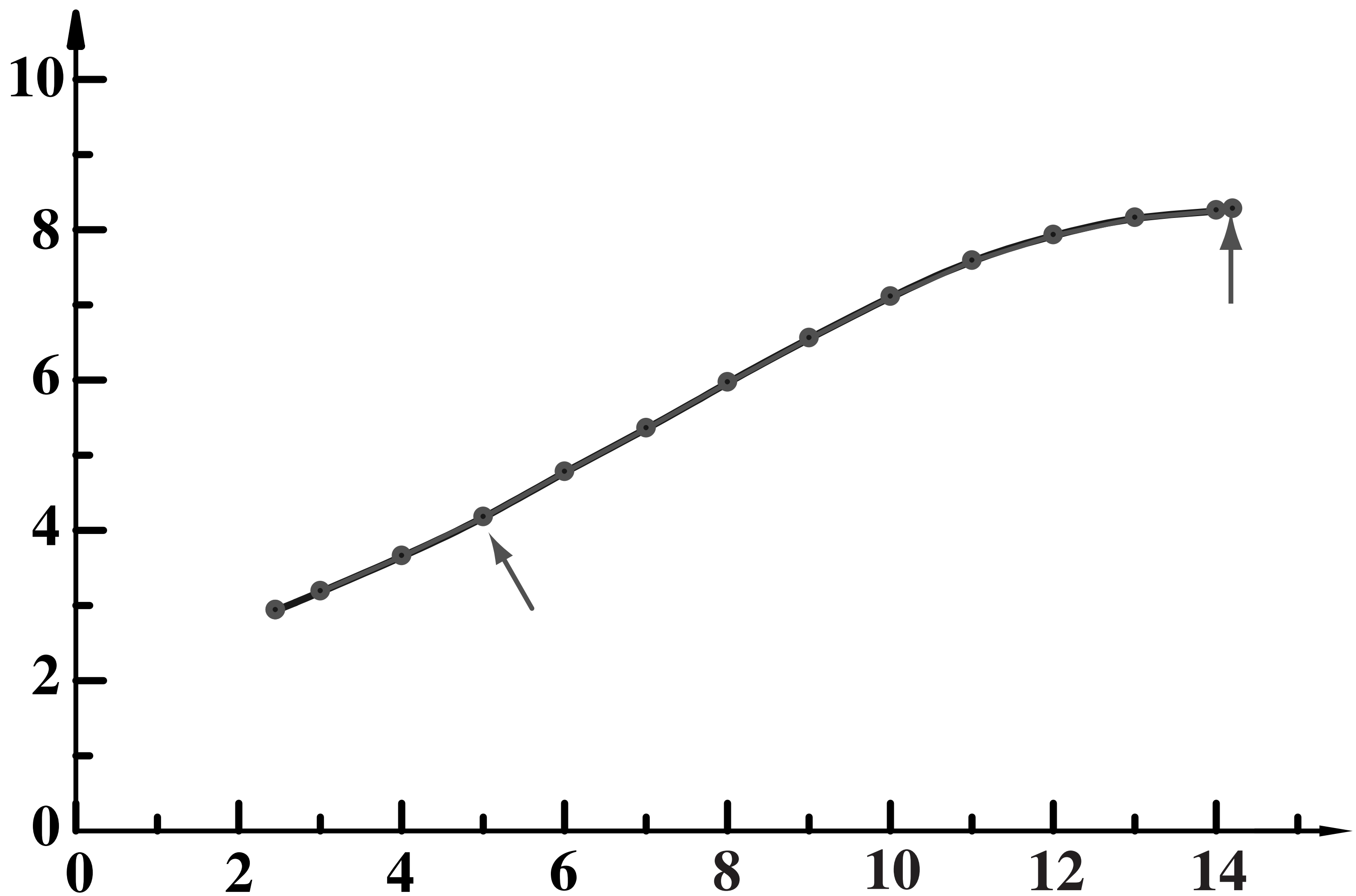


| x | y |
|---|---|
| 2.45 | 2.93 |
| 3.00 | 3.18 |
| 4.00 | 3.65 |
| 5.00 | 4.17 |
| 6.00 | 4.77 |
| 7.00 | 5.35 |
| 8.00 | 5.96 |
| 9.00 | 6.55 |
| 10.00 | 7.10 |
| 11.00 | 7.58 |
| 12.00 | 7.92 |
| 13.00 | 8.15 |
| 14.00 | 8.25 |
| 14.20 | 8.27 |

## List of digitize points

# Cubic Splines

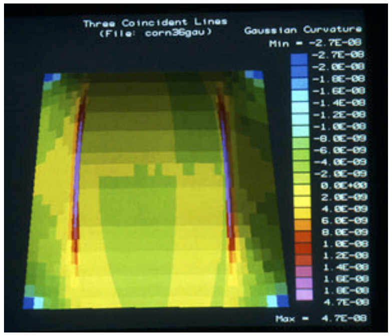
## A cubic spline curve generated from the digitized points



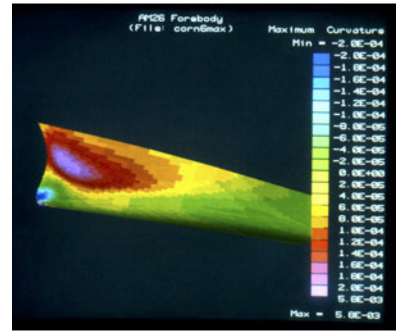

## Notice the wiggles

## Bézier's contribution

**Divorced the manipulation from the mathematics**



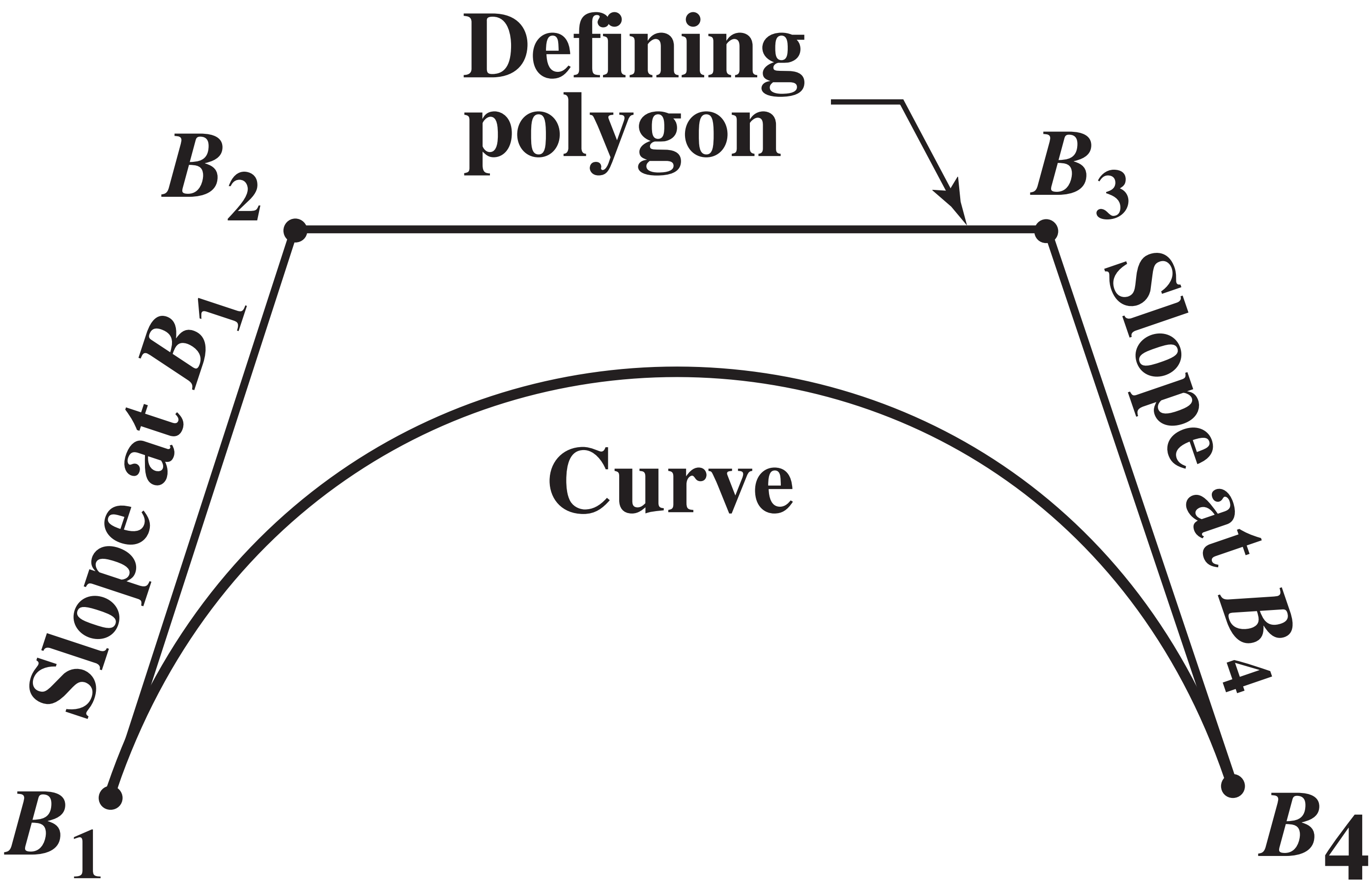

# Course 31
# NURBS
# (NonUniform Rational B-splines)




## Part 1 (8:40)
## Bézier and B-spline Curves

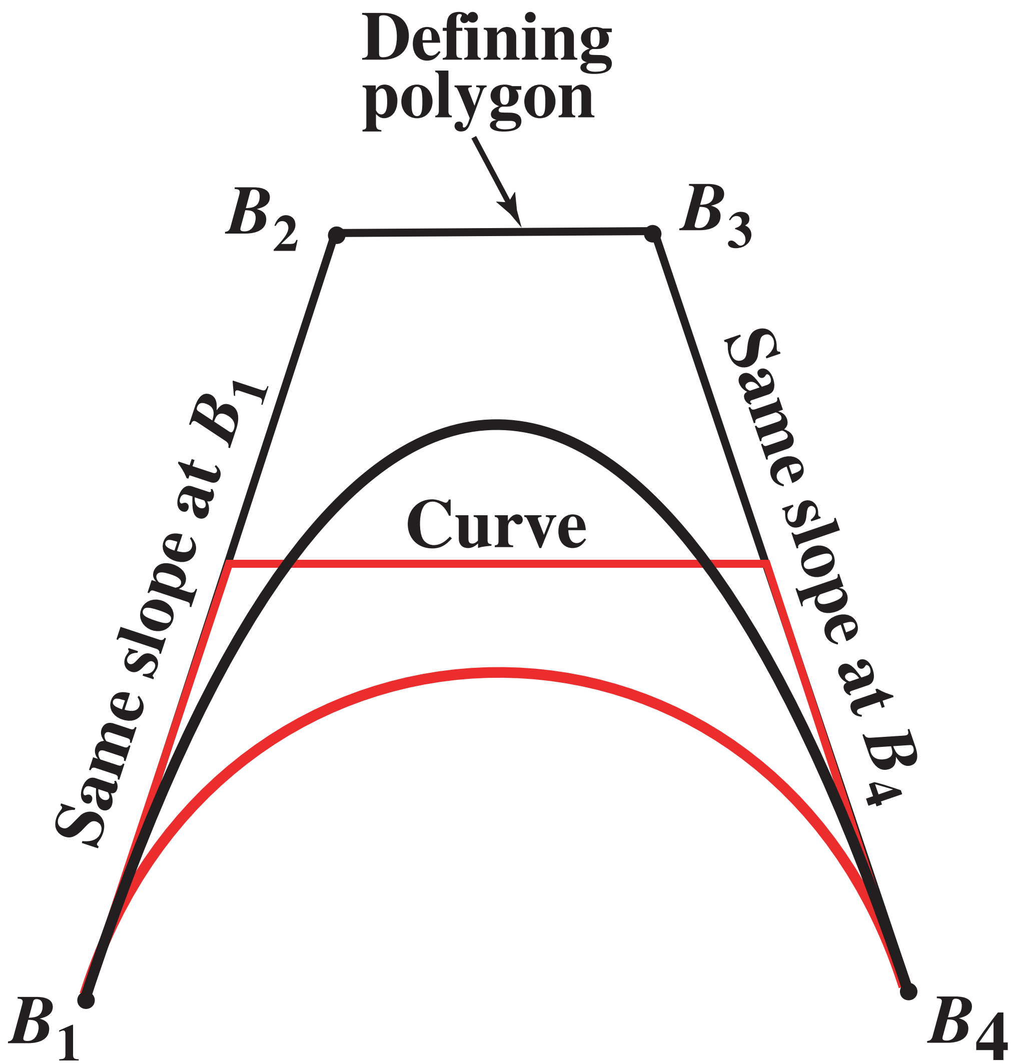
**David F. Rogers**
**Professor of Aerospace Engineering**
**United States Naval Academy**
**Annapolis, MD 21402**

# A Bézier curve



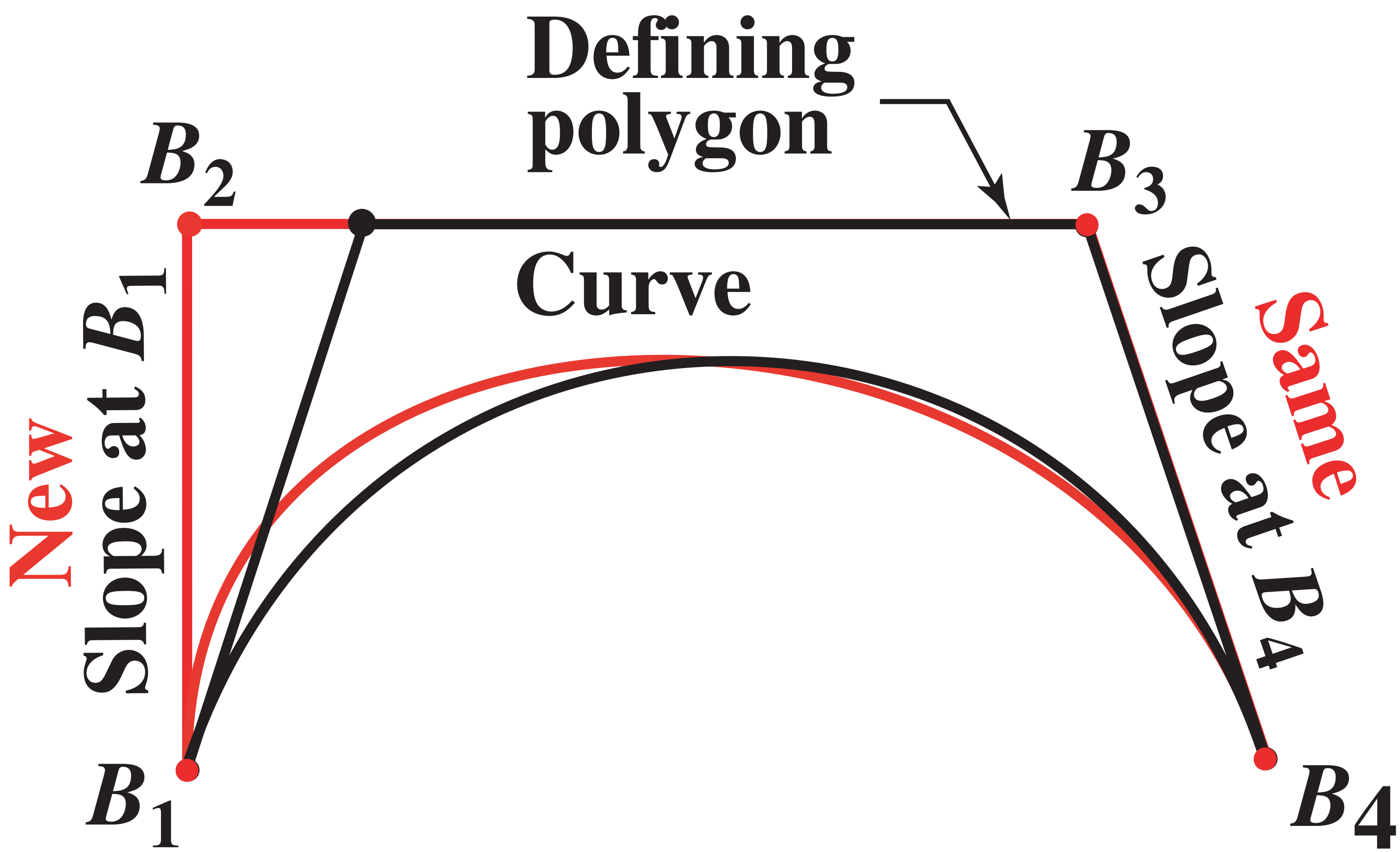
# Bézier curve – Controls

## Same slope – changed height

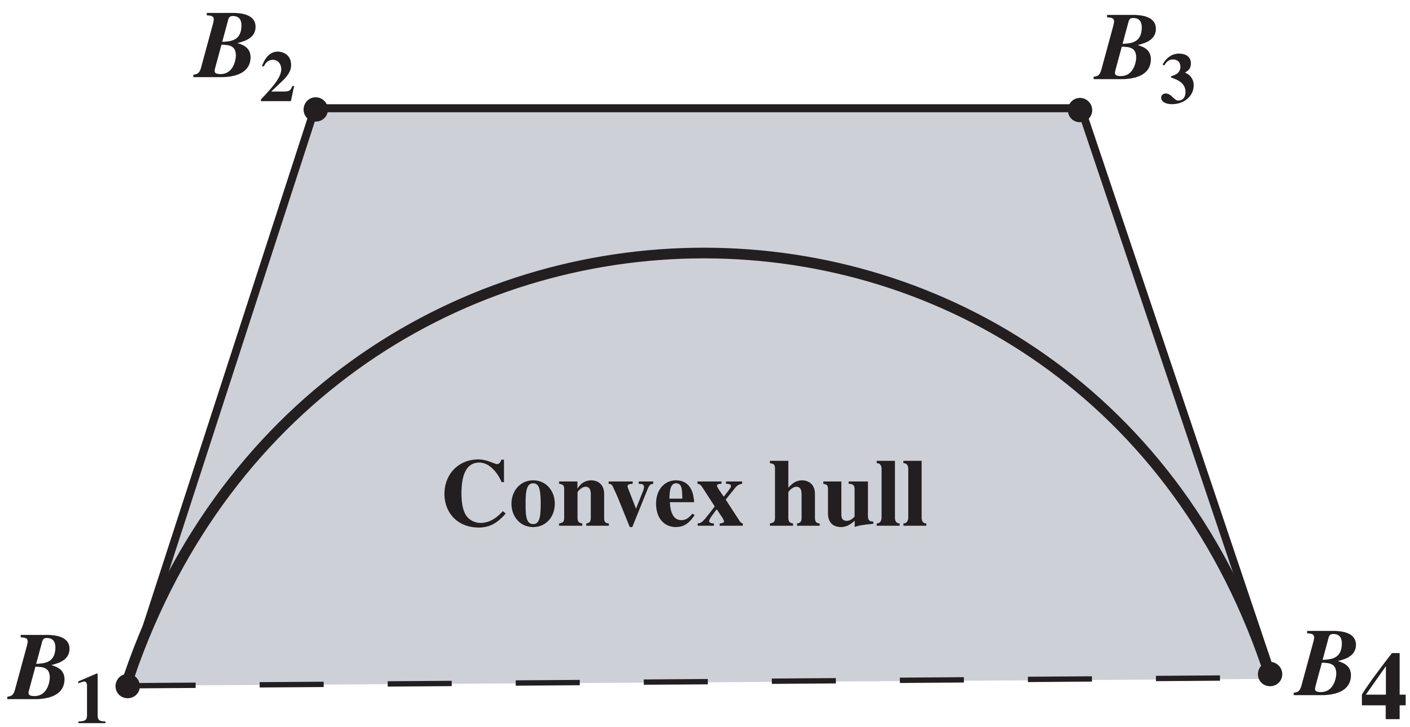


# Bézier curve – Controls

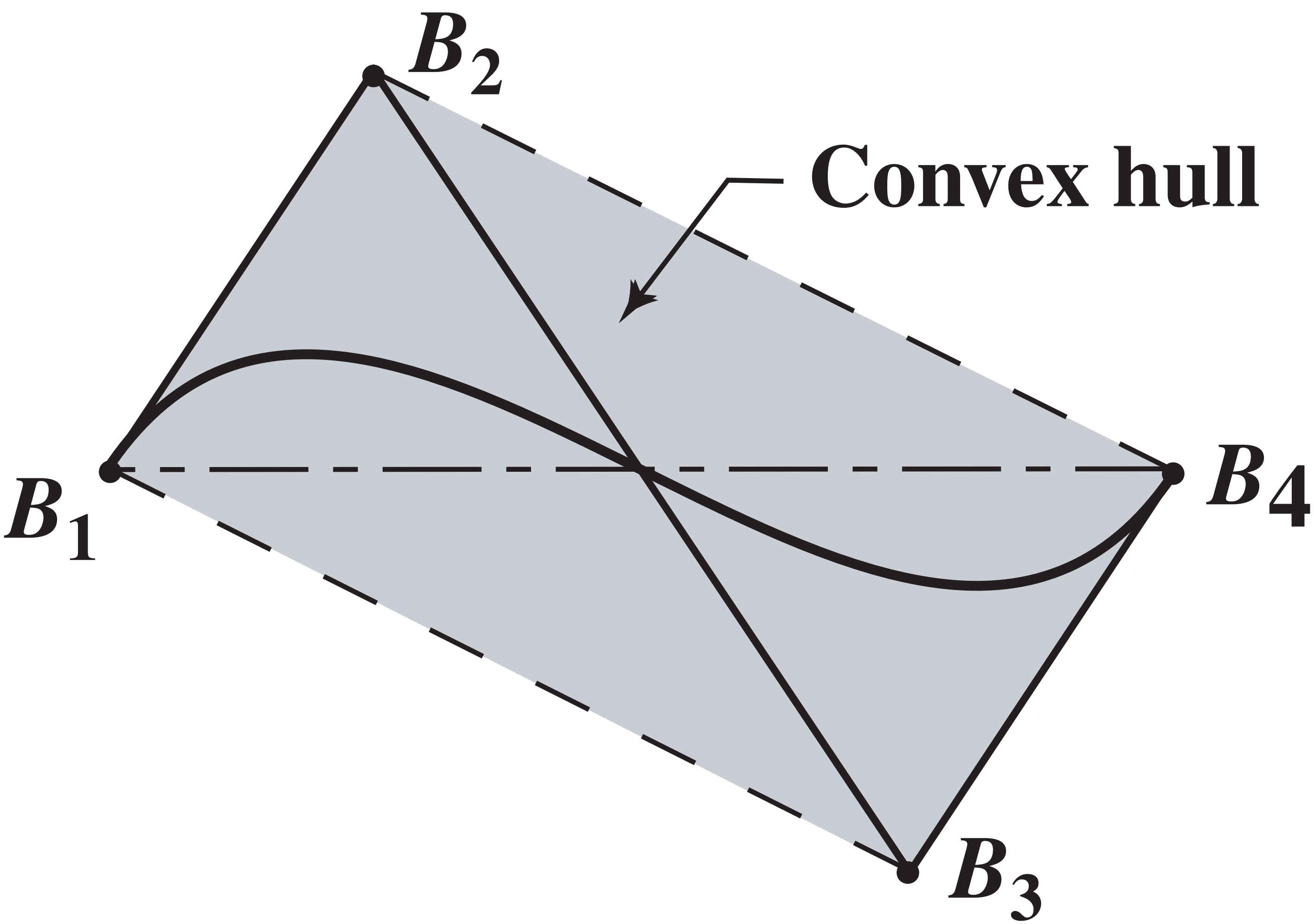
## Changed slope – original height



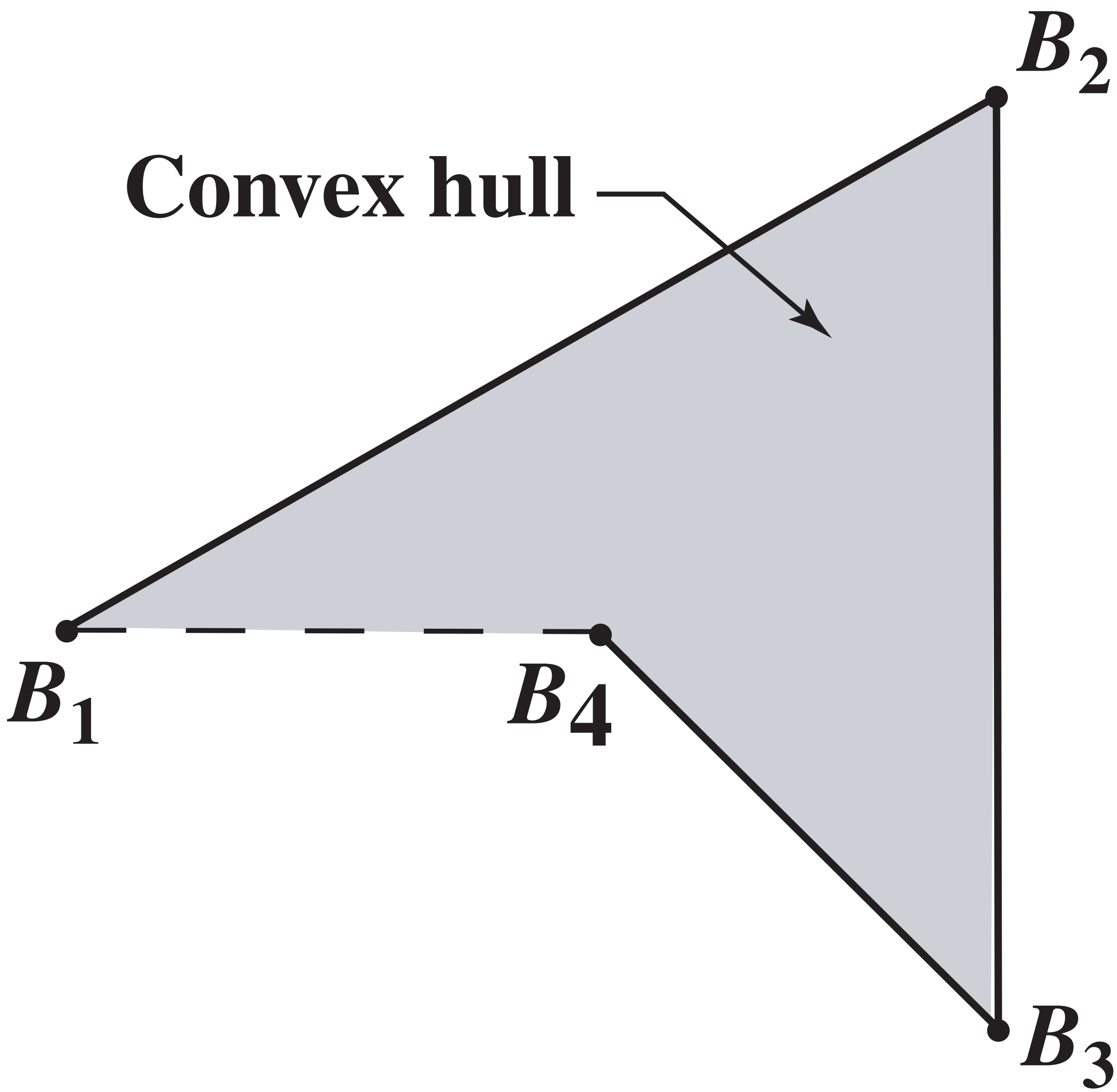
# Bézier curve – Convex hull



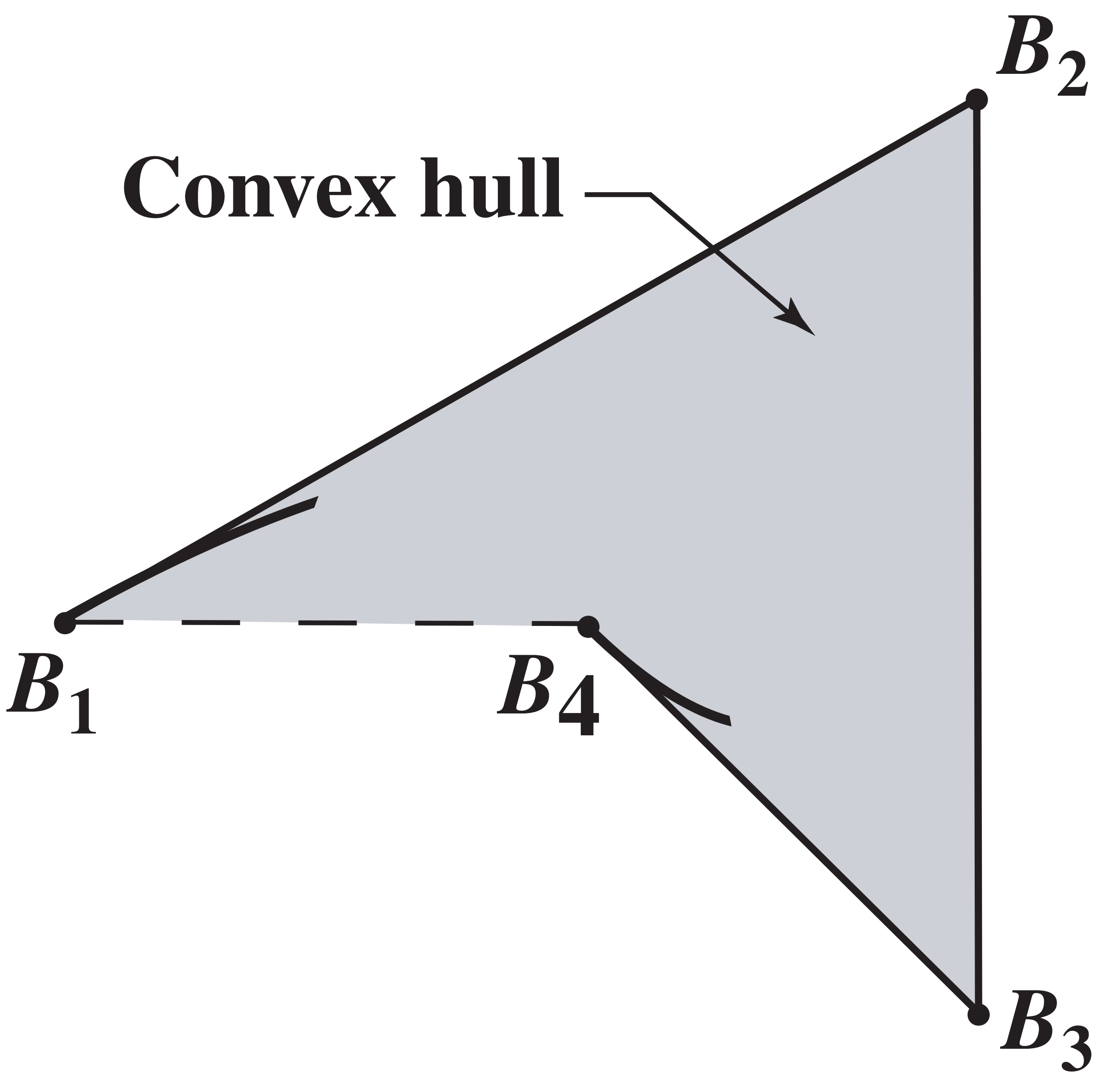
# Bézier curve – Convex hull



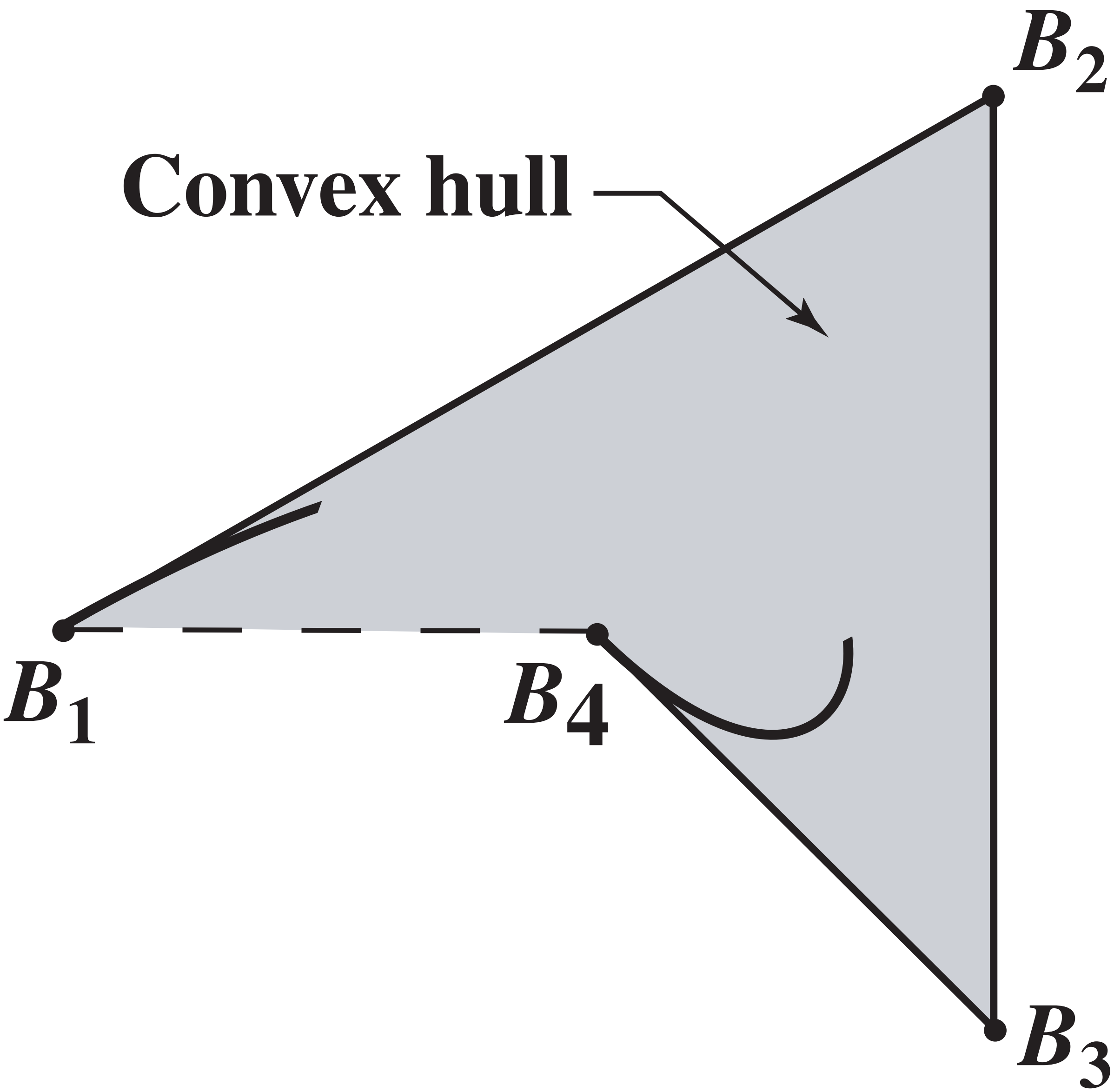
# Bézier curve – Drawing slopes



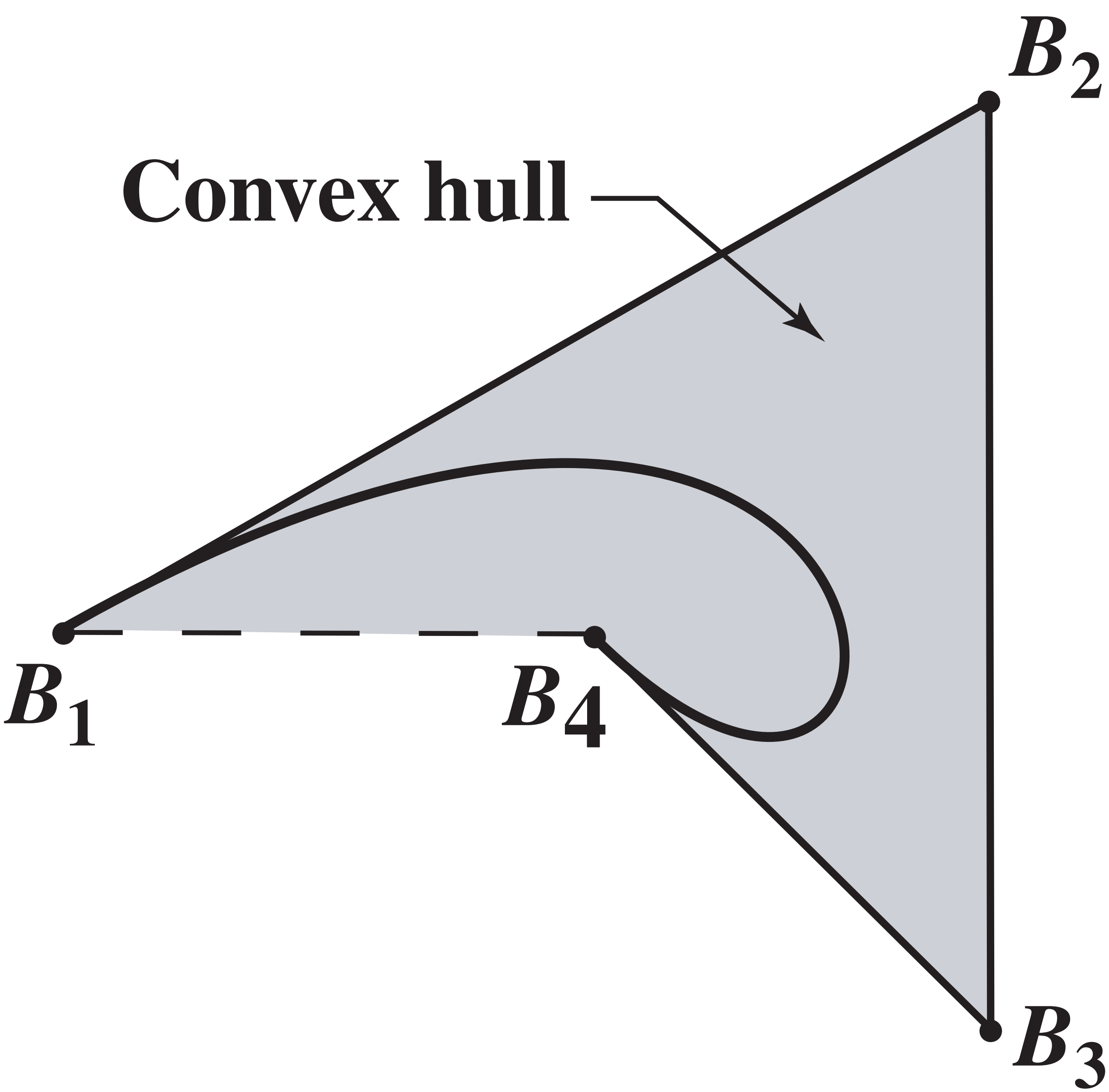
# Bézier curve — Drawing slopes



# Bézier curve — Drawing curves



# Bézier curve – Drawing curve



# Bézier curve – Mathematics

$$P(t) = \sum_{i=1}^{n} B_i J_{n,i}(t)$$

**where**

$$J_{n,i} = \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i}$$

**and**

$$\binom{n-1}{i-1} = \frac{(n-1)!}{(i-1)!\,(n-i)!}$$

# Bézier curve – Mathematics

$$P(t) = \sum_{i=1}^{n} B_i J_{n,i}(t)$$

**What does this mean?**

# Bézier curves – Mathematics

**Assume** $n = 4$

$$\begin{aligned} P(t) &= \sum_{i=1}^{n} B_i J_{n,i}(t) = \sum_{i=1}^{4} B_i J_{4,i}(t) \\ &= B_1 J_{4,1}(t) + B_2 J_{4,2}(t) + B_3 J_{4,3}(t) + B_4 J_{4,4}(t) \end{aligned}$$

**But what is this thing** $t$**?**

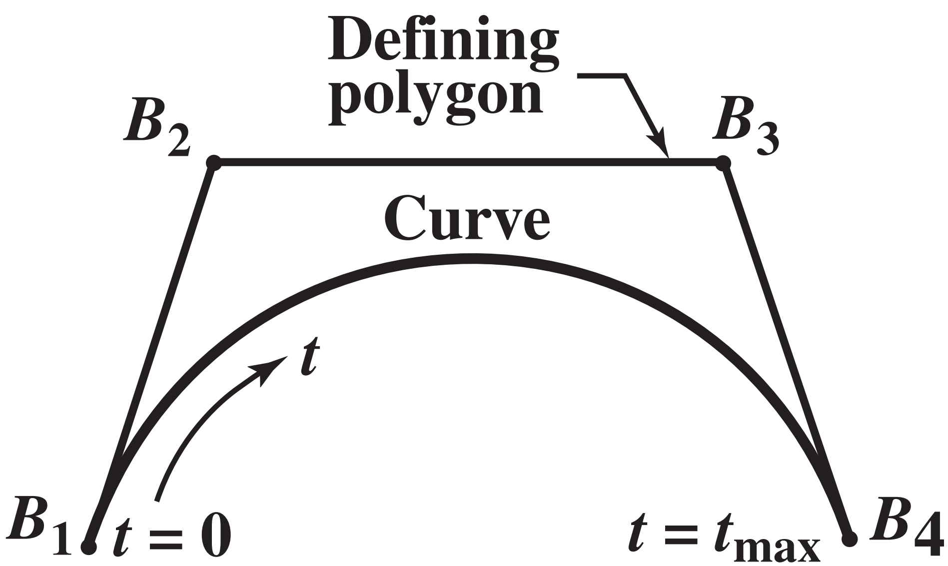
# Bézier curve – Mathematics

**But what is this thing $t$?**

**$t$ is a parameter**

**Bézier curves are parametric**

**$t$ runs along the curve $0 \leq t \leq t_{max}$**



# Bézier curves – Mathematics

## Be careful, there are three (3) components

**Assume** $n = 4$

$$\mathrm{X(t)} = \sum_{i=1}^{n} x_i J_{n,i}(t) = \sum_{i=1}^{4} x_i J_{4,i}(t)$$

$$= x_1 J_{4,1}(t) + x_2 J_{4,2}(t) + x_3 J_{4,3}(t) + x_4 J_{4,4}(t)$$

$$Y(t) = \sum_{i=1}^{n} y_i J_{n,i}(t) = \sum_{i=1}^{4} y_i J_{4,i}(t)$$

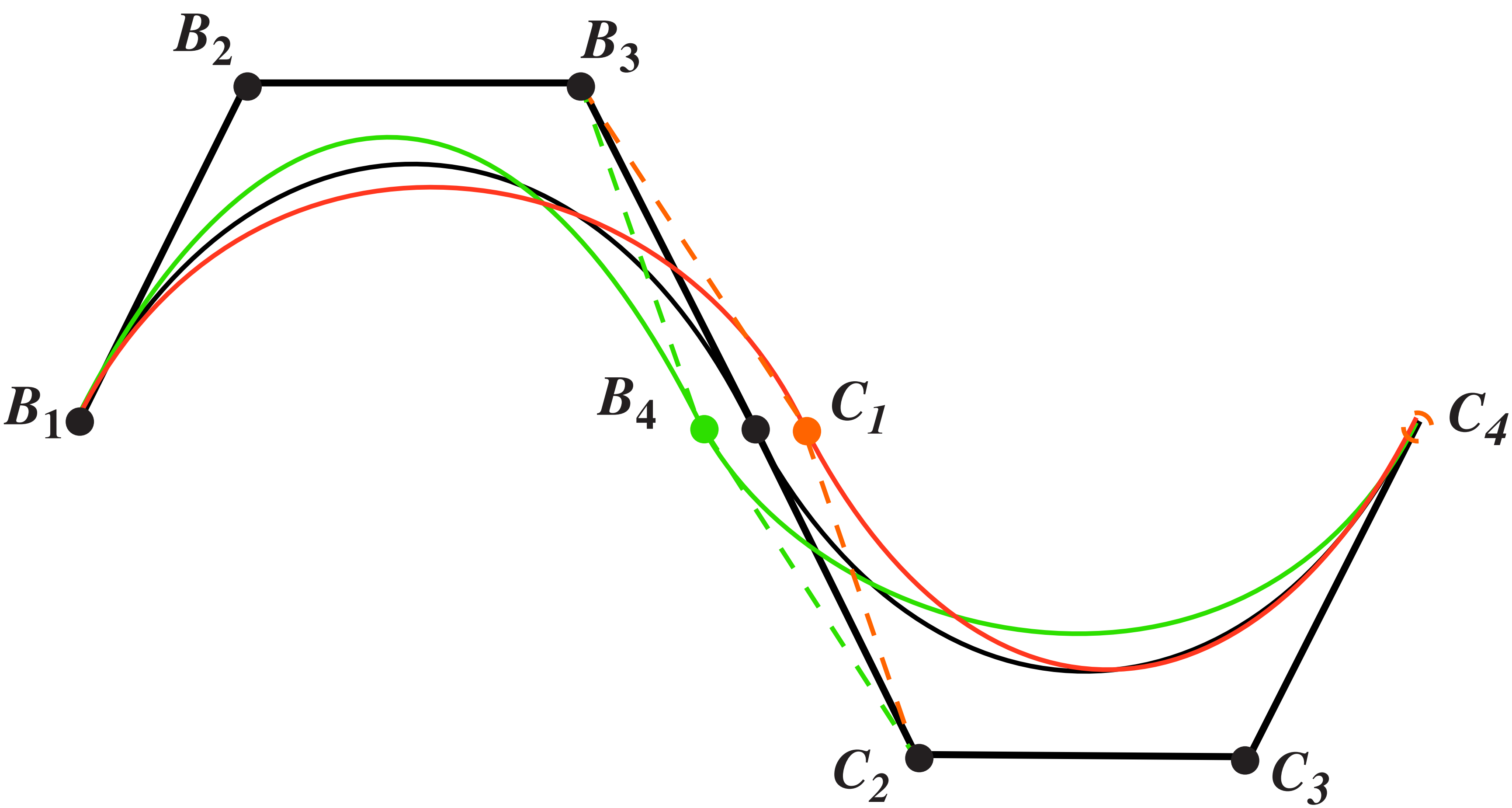
$$= y_1 J_{4,1}(t) + y_2 J_{4,2}(t) + y_3 J_{4,3}(t) + y_4 J_{4,4}(t)$$

$$Z(t) = \sum_{i=1}^{n} z_i J_{n,i}(t) = \sum_{i=1}^{4} z_i J_{4,i}(t)$$

$$= z_1 J_{4,1}(t) + z_2 J_{4,2}(t) + z_3 J_{4,3}(t) + z_4 J_{4,4}(t)$$

# Bézier curves – Continuity

## $B_3$, $B_4/C_1$, $C_2$ must lie in straight line



# Bézier curves – Geometric continuity

**Two curve segments joined at ends –**
**$G^0$ continuity**

**Two curve segments joined at ends –**
*and* **tangent vectors point in same direction**
**$G^1$ continuity**

**The tangent vector magnitudes**
*do not* **have to be the same**

**Less restrictive than parametric continuity**

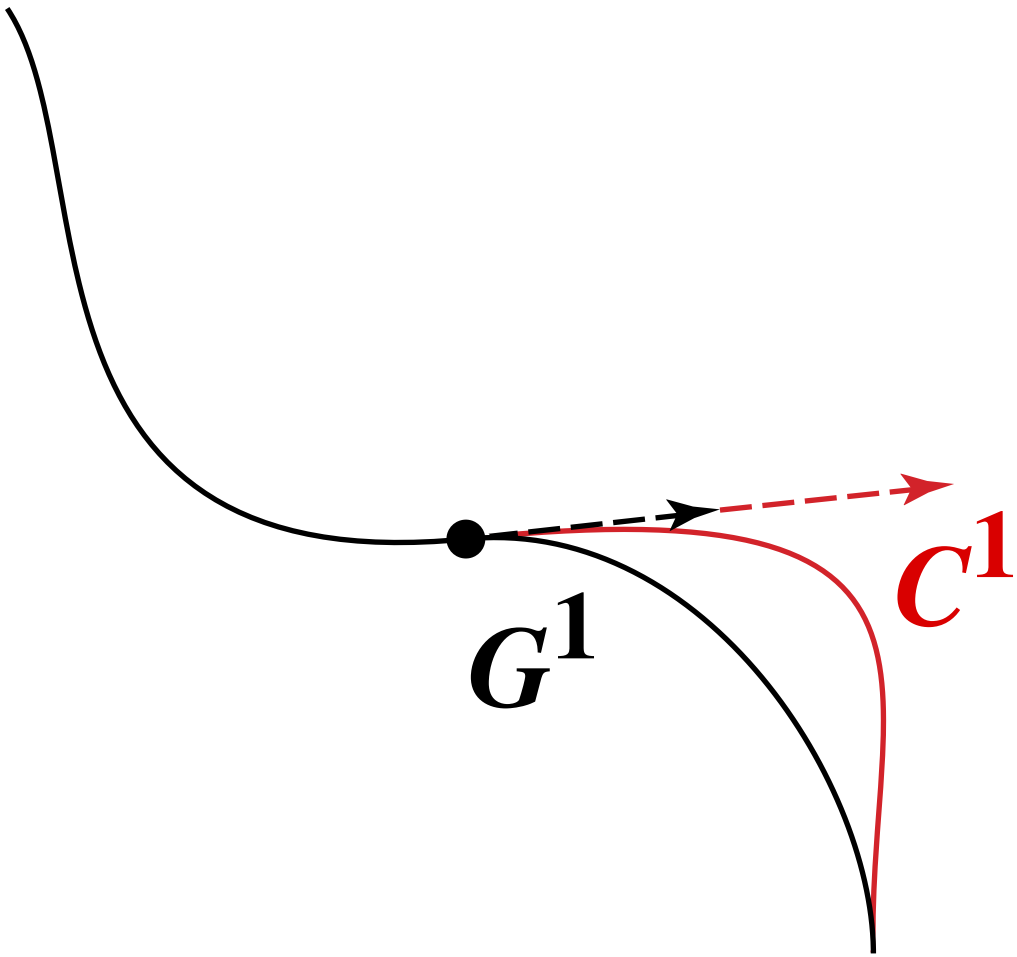
**Bézier curves – Parametric continuity**

**Two curve segments joined at ends –**
**$C^0$ continuity**

**Two curve segments joined at ends –**
*and* **tangent vectors in same direction –**
*and* **tangent vectors magnitudes the same**
**$C^1$ continuity**

**More restrictive than geometric continuity**

# Bézier curves – Continuity comparison



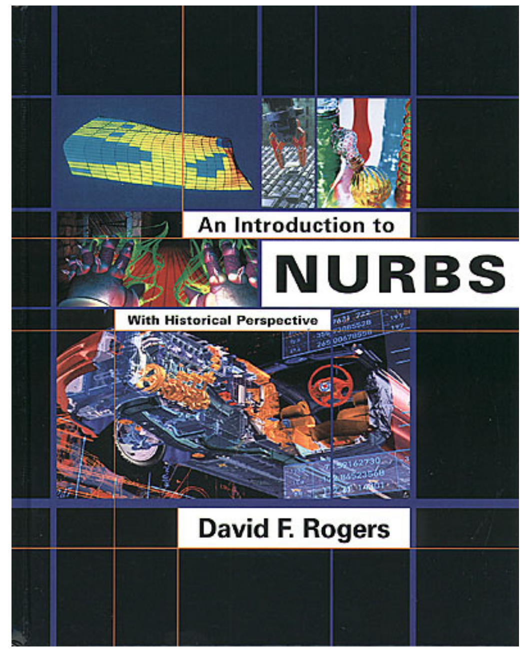
# Bézier Curves - Additional Topics

Degree elevation

Degree reduction

Subdivision

Reparameterization




Additional reading:

Rogers, D. F.
An Introduction to NURBS,
With Historical Perspective
Morgan Kaufmann Publishers, 2001

Piegl, L. & Tiller, W.
The NURBS Book, Springer-Verlag 1995

# B-spline curves

## A bit of history

Robin Forrest

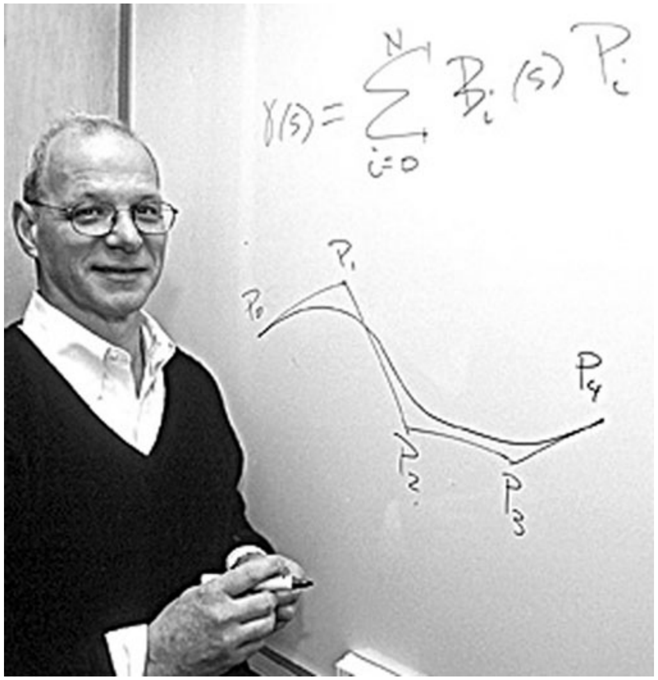
Robin, at Cambridge, knew French

Robin also knew about Bézier's work

Robin knew Bill Gordon at Syracuse University

# B-spline curves

## A bit of history




Rich Riesenfeld

Rich Riesenfeld was Gordon's PhD student

Gordon sent Riesenfeld to England to learn about Bézier's work

# B-spline curves

## Bézier curve difficulties

Bernstein basis is global
- No local control

Order (degree) fixed
- Equal to number of control vertices

High order (degree) required for flexibility
- Wiggles

Difficult to maintain continuity

# B-spline curves – Definition

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \quad t_{\mathbf{min}} \leq t < t_{\mathbf{max}}, \quad 2 \leq k \leq n+1$$

| | |
|---|---|
| $B_i$ **s** | **are the polygon control vertices** |
| $N_{i,k}(t)$ | **are the normalized B-spline basis functions of order** $k$ |
| $n+1$ | **is the number of control vertices** |

# B-spline curves – Basis functions

$$N_{i,1}(t) = \begin{cases} 1 & \textbf{if } x_i \leq t < x_{i+1} \\ 0 & \textbf{otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}$$

$x_i$ **s** **are the elements of a knot vector**

**Note** $0/0 \equiv 0$

# B-spline curves – Properties

$N_{i,k}(t) \equiv 1$ **for all** $t$

$N_{i,k}(t) \geq 0$ **for all** $t$

**Maximum order** $k_{max} = n + 1$

**Maximum degree,** $n$**, is one less than the order**

**Exhibits the variation diminishing property**

**Follows shape of the control polygon**

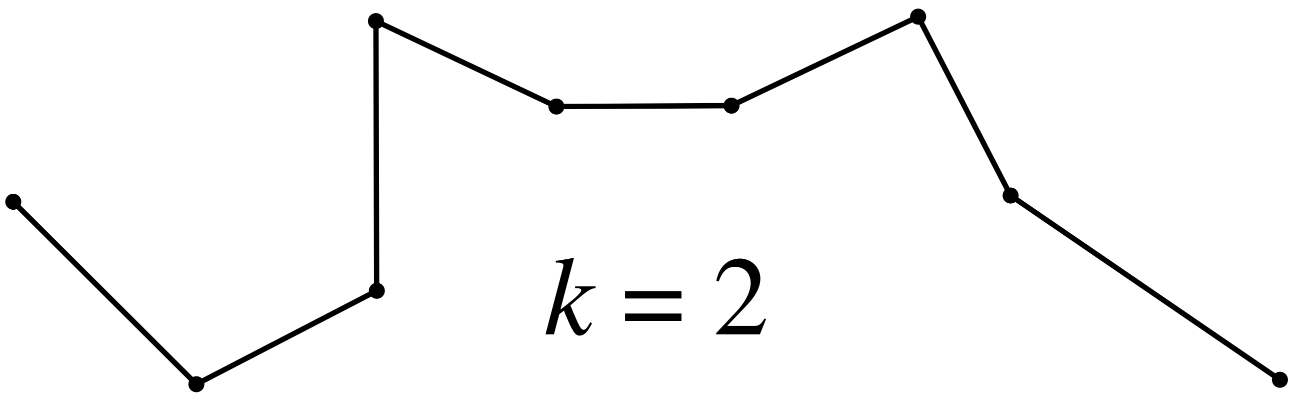
**Transform curve – transform control polygon**

**Everywhere** $C^{k-2}$ **continuous**

# B-spline curves – Convex hulls

**Stronger than for Bézier curves**

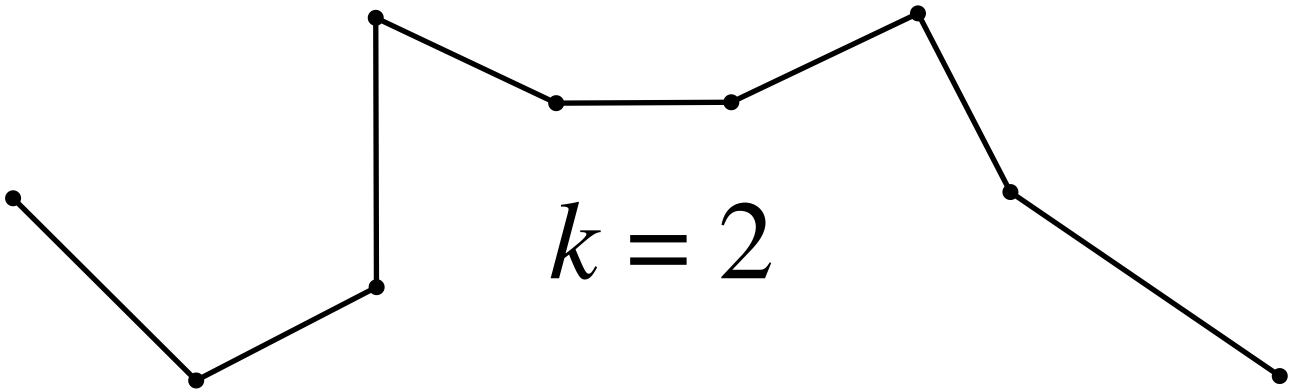
**A point on the curve $P(t)$ lies within the convex hull of $k$ neighboring control vertices**

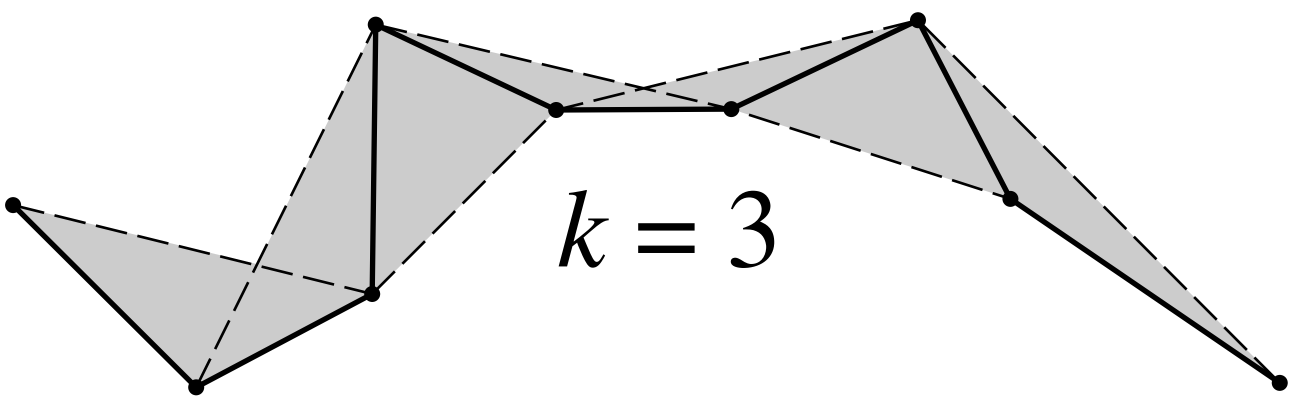



**Notice for order, $k = 2$ the degree is one – a straight line**

**The B-spline curve is the control polygon**

# B-spline curves – Convex hulls

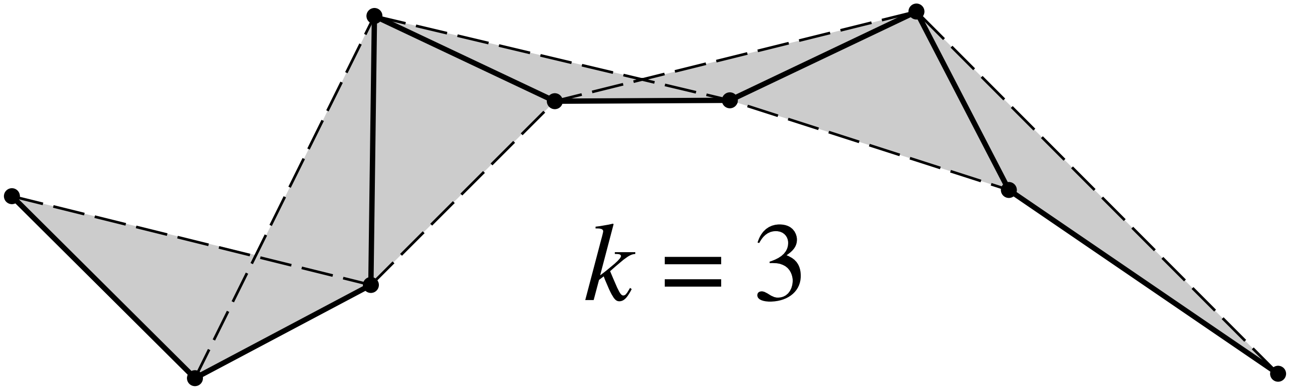


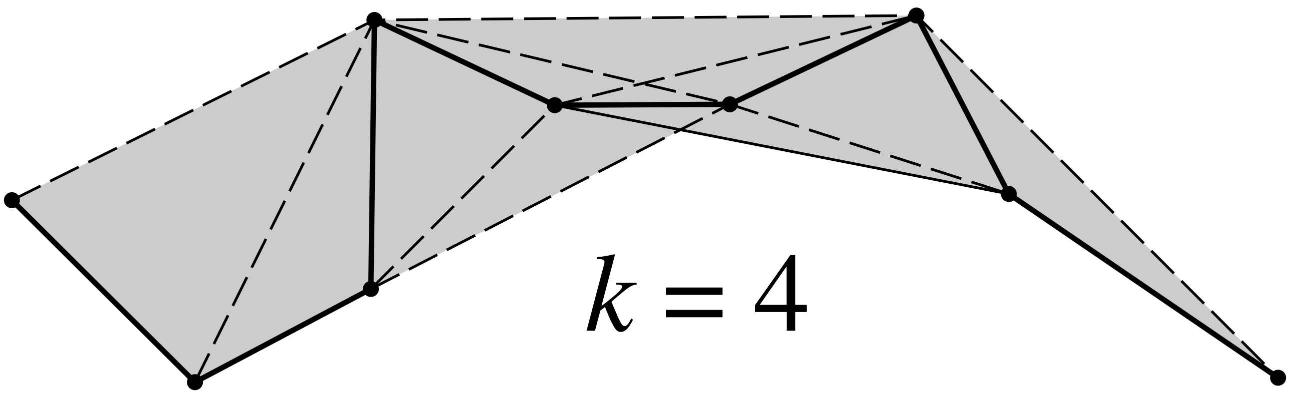

**For $k = 3$ a larger region may contain the curve**

**The B-spline curve will not exactly follow polygon**

# B-spline curves – Convex hulls



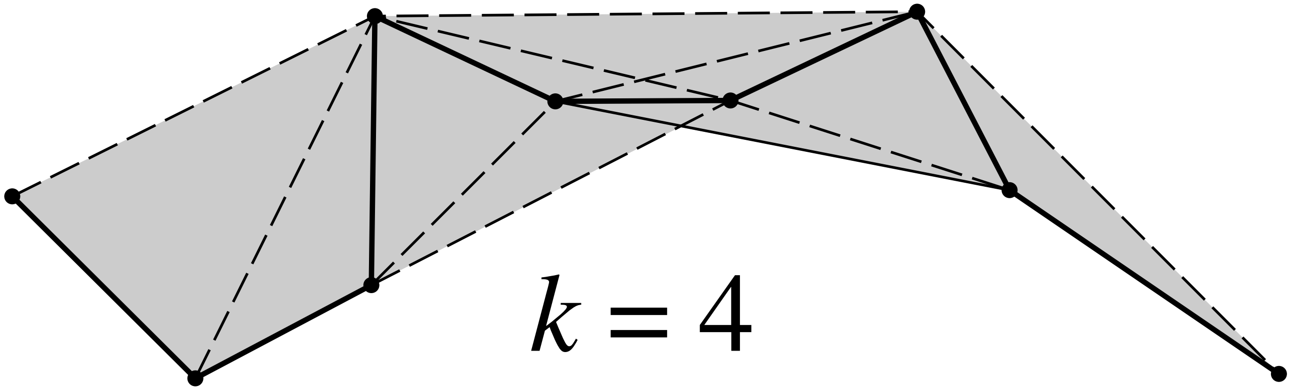
$k = 3$

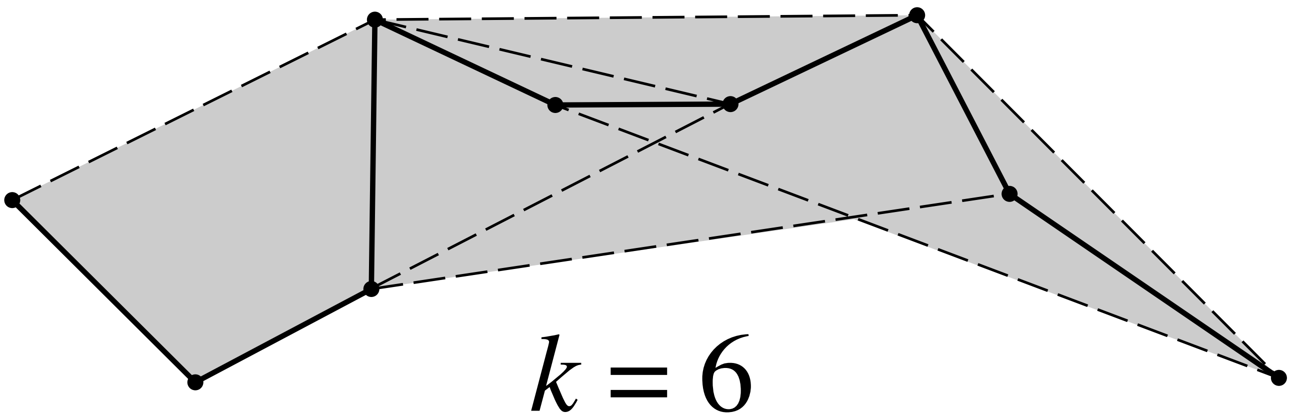


$k = 4$

**The higher the order the less closely the B-spline curve follows the control polygon**

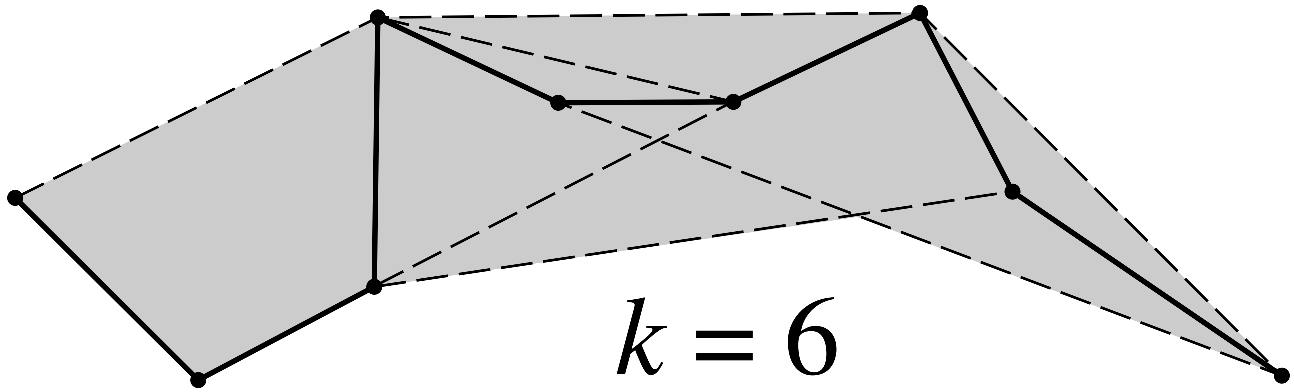
# B-spline curves – Convex hulls



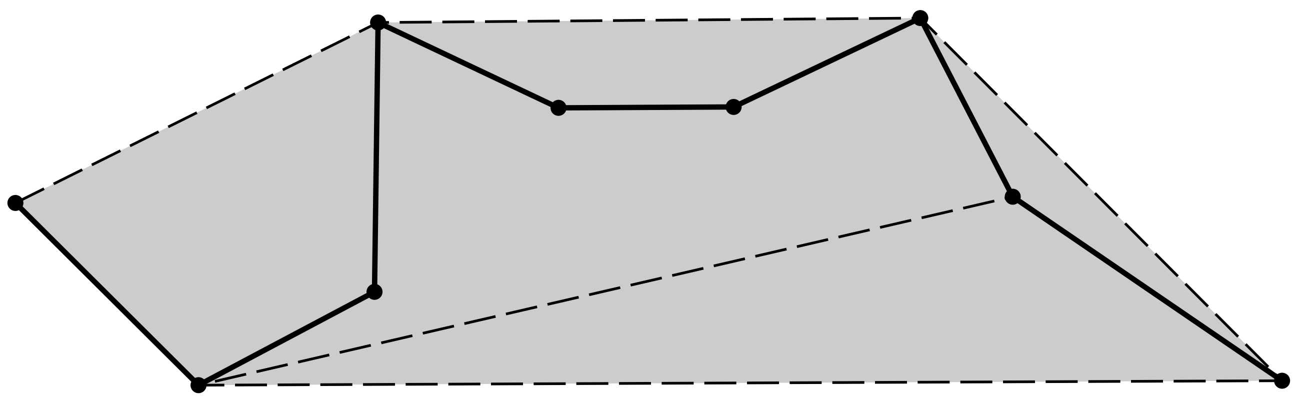




**The higher the order the less closely the B-spline curve follows the control polygon**

# B-spline curves – Convex hulls



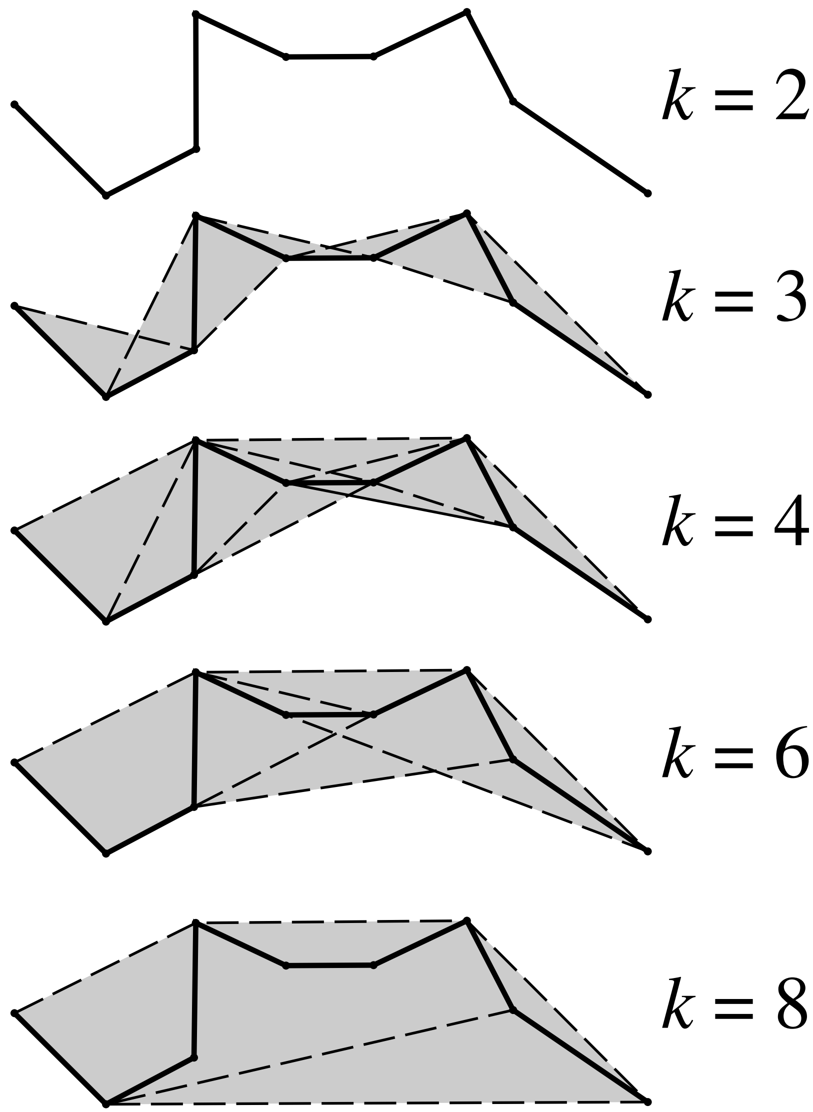
$k = 6$



$k = 8$

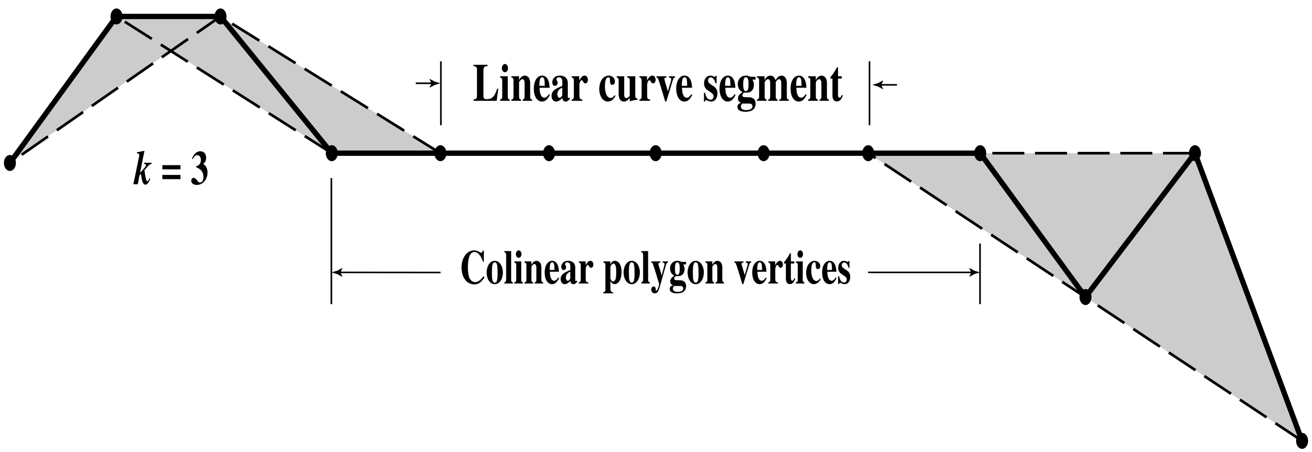
**The higher the order the less closely the B-spline curve follows the control polygon**

# B-spline curves – Convex hulls



# B-spline curves – Convex hulls

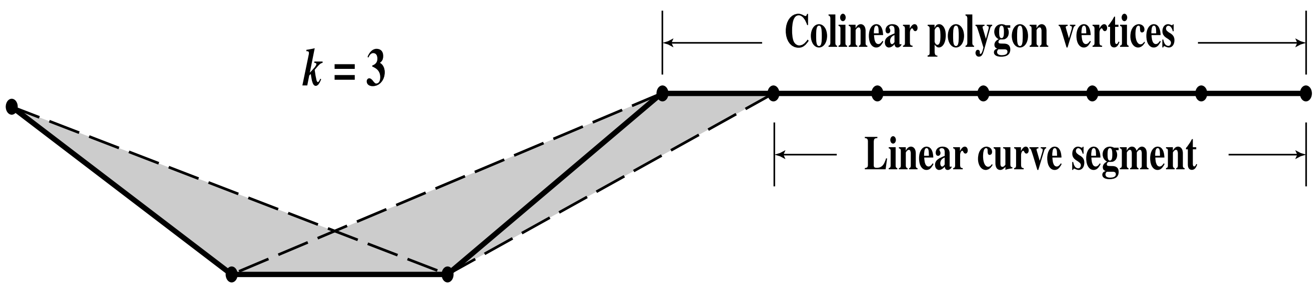
## Straight segments




Straight line results start and stop $k - 2$ spans from the ends of the colinear segments

# B-spline curves – Convex hulls

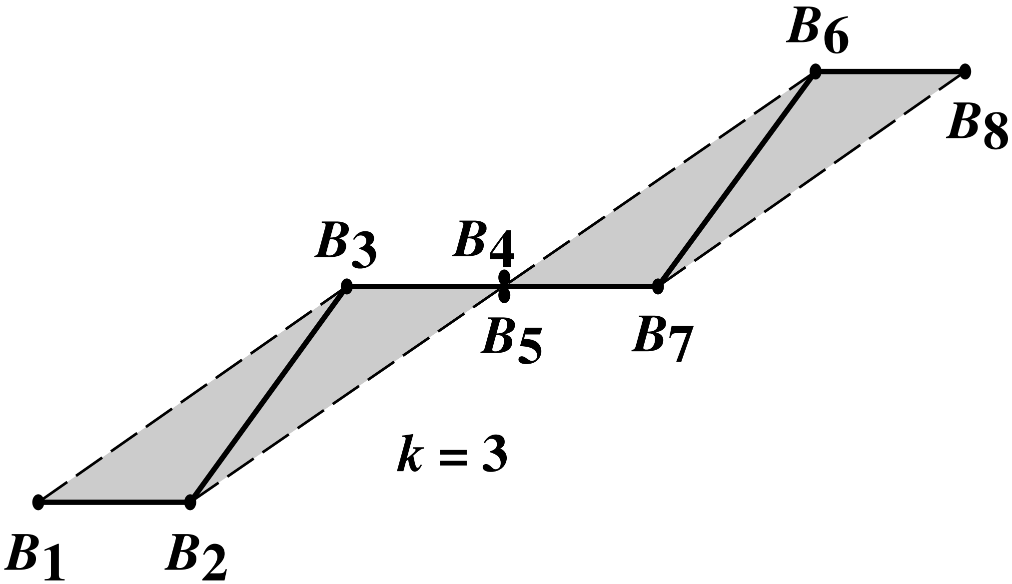
## Straight segments at ends




For $\ell$ colinear vertices then the number of linear segments at the end is at least $\ell - k + 1$

# B-spline curves – Convex hulls

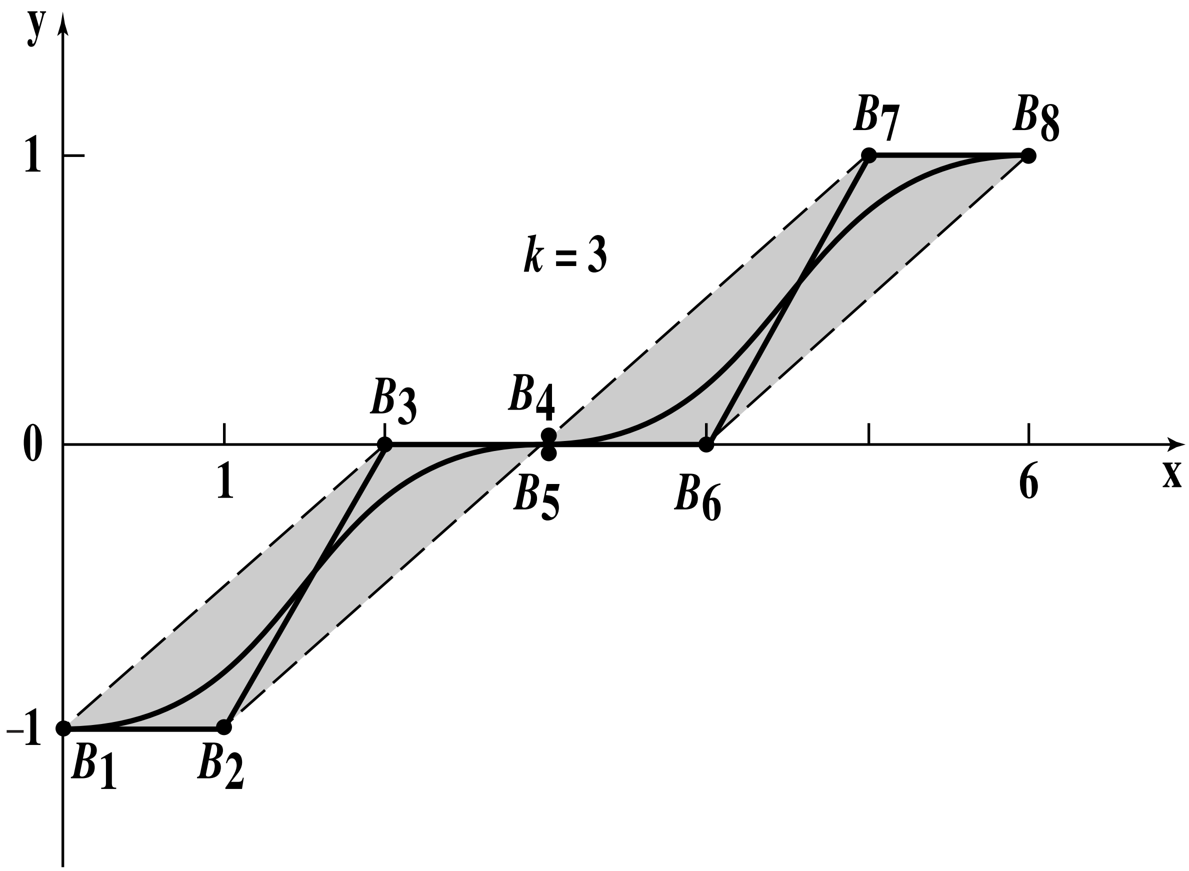
## Coincident vertices




**$k-1$ coincident vertices are required for the curve to pass through the vertices**

# B-spline curves – Convex hulls

## Coincident vertices




**The curve smoothly transitions through the coincident vertices with $C^{k-2}$ continuity**

# B-spline curves – Knot vectors

## Only requirement

$x_i \leq x_{i+1}$

## Uniform – evenly spaced

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -0.2 & -0.1 & 0 & 0.1 & 0.2 & 0.3 \end{bmatrix}$$

## Typically begin at zero

## May normalize to $0 \leq x_i \leq 1.0$

$$\begin{bmatrix} 0 & 0.2 & 0.4 & 0.6 & 0.8 & 1.0 \end{bmatrix}$$

# Knot vectors – Open uniform

## Multiplicity equal to $k$ at the ends

$$k = 2 \quad [0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3]$$
$$k = 3 \quad [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 3]$$
$$k = 4 \quad [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 3 \quad 3]$$

## or normalized

$$k = 4 \quad [0 \quad 0 \quad 0 \quad 0 \quad 1/3 \quad 2/3 \quad 1 \quad 1 \quad 1 \quad 1]$$

# Knot vectors – Open uniform

## Formal definition

$$x_i = 0 \qquad 1 \leq i \leq k$$

$$x_i = i - k \qquad k + 1 \leq i \leq n + 1$$

$$x_i = \underbrace{n - k + 2}_{\textbf{max knot value}} \qquad n + 2 \leq i \leq \underbrace{n + k + 1}_{\textbf{max no. of knots}}$$

## Curves behave most nearly like Bézier curves

# Knot vectors – Open nonuniform

$$[0 \quad 0 \quad 0 \quad 1 \quad 3/2 \quad 2 \quad 2 \quad 2]$$

$$[0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2]$$

## Repeating knot value

# Basis functions

$$N_{i,1}(t) = \begin{cases} 1 & \textbf{if } x_i \leq t < x_{i+1} \\ 0 & \textbf{otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}$$

**$x_i$ s are the elements of a knot vector**

**Note:** $0/0 \equiv 0$

**Recursion relation**

**Dependent on knot vector**

# Basis functions – Dependencies

## Form triangular pattern

$$
\begin{array}{llllll}
N_{i,k} & & & & & \\
N_{i,k-1} & N_{i+1,k-1} & & & & \\
N_{i,k-2} & N_{i+1,k-2} & N_{i+2,k-2} & & & \\
\vdots & & & \vdots & \ddots & \\
N_{i,1} & N_{i+1,1} & N_{i+2,1} & N_{i+3,1} & \cdot & N_{i+k-1,1}
\end{array}
$$

**The single basis function in the first row depends on all those in the last row**

# Basis functions – Inverse Dependencies

## Form triangular pattern

$$
\begin{array}{ccccccc}
N_{i-k+1,k} & \cdot & N_{i+k-1,k} & N_{i,k} & N_{i+1,k} & \cdot & N_{i+k-1,k} \\
\cdot & \cdot & & & & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & \\
 & & N_{i-1,2} & N_{i,2} & N_{i+1,2} & & \\
 & & & N_{i,1} & & &
\end{array}
$$

## Influence of a single first-order basis function $N_{i,1}$ on higher-order basis functions

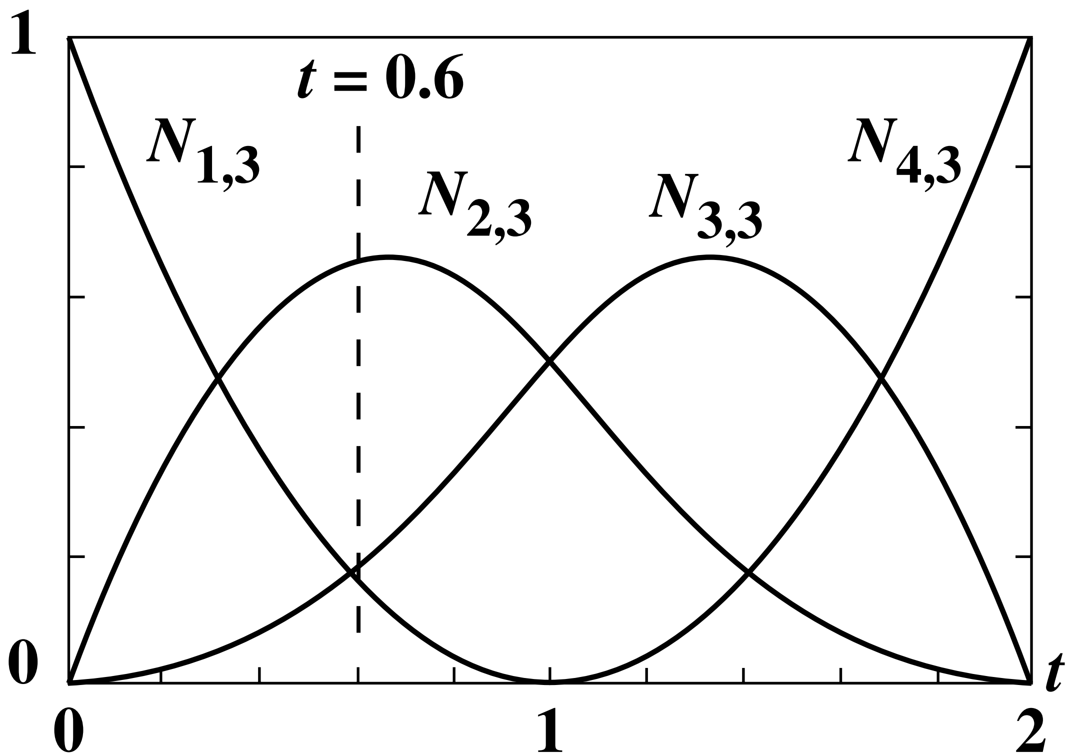
# Basis functions – Sum equals one at any t

**Example:** $n+1=4, \quad k=3$

$$[\,X\,] = [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2]$$

$\boldsymbol{t = 0.6}$

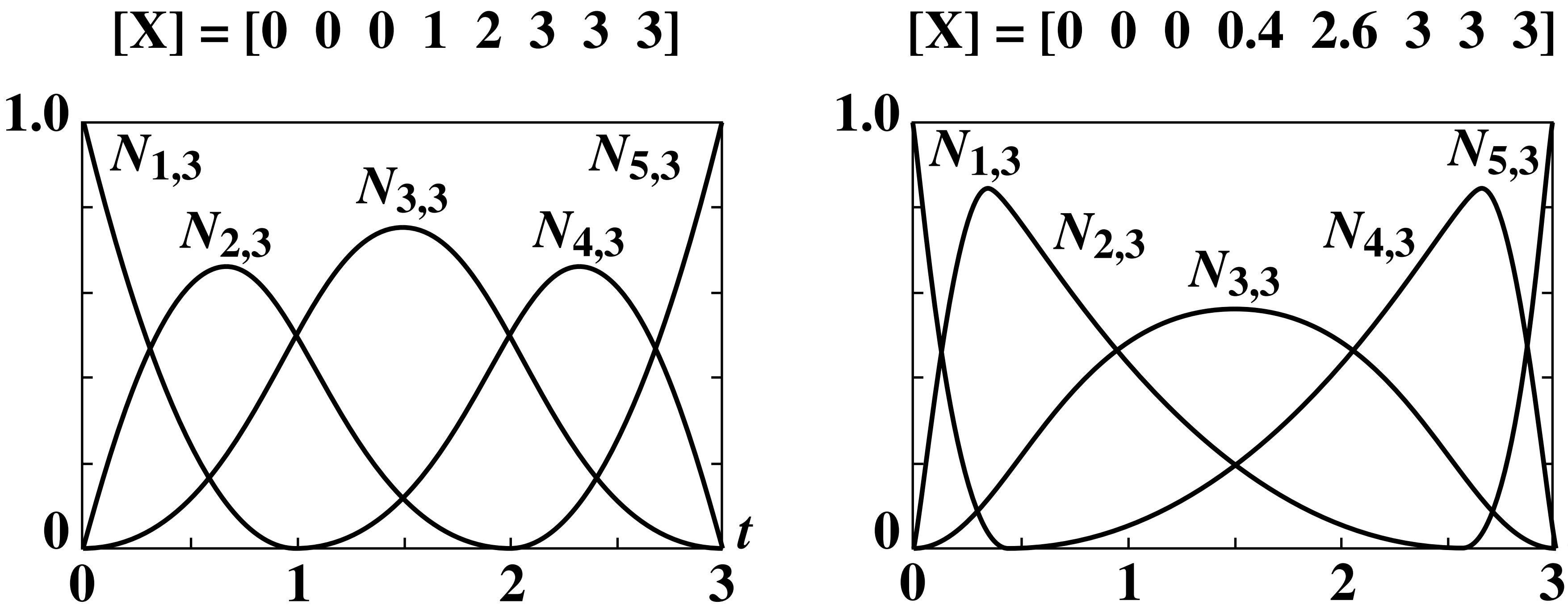
$N_{1,3} + N_{2,3} + N_{3,3} + N_{4,3} = 0.16 + 0.66 + 0.18 + 0.0 = 1.0$



# Basis functions – Comparisons

## Uniform and nonuniform knot vectors

$k = 3, \quad n + 1 = 5$



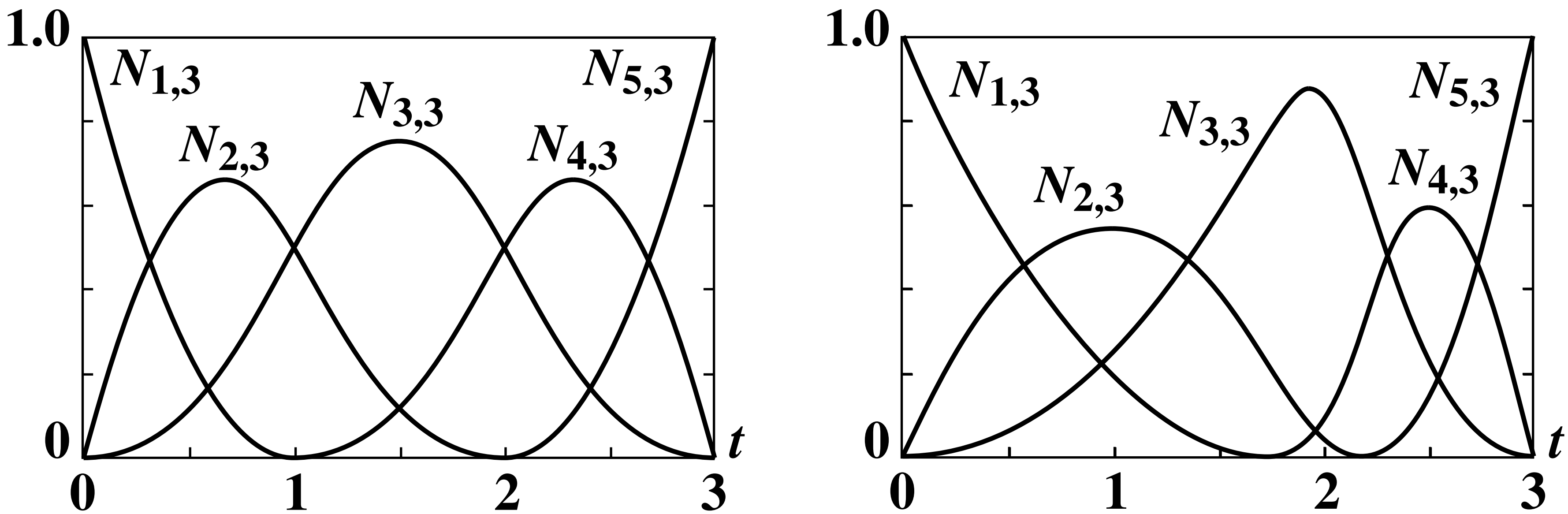

**Notice:** $N_{2,3}$ **and** $N_{4,3}$ **pulled left and right**
**More influence for** $B_{2,3}$ **and** $B_{4,3}$ **control vertices**
**Less for others**

# Basis functions – Comparisons

## Uniform and nonuniform knot vectors

$k = 3, \quad n + 1 = 5$

$[X] = [0\ 0\ 0\ 1\ 2\ 3\ 3\ 3]$ $\qquad$ $[X] = [0\ 0\ 0\ 1.8\ 2.2\ 3\ 3\ 3]$



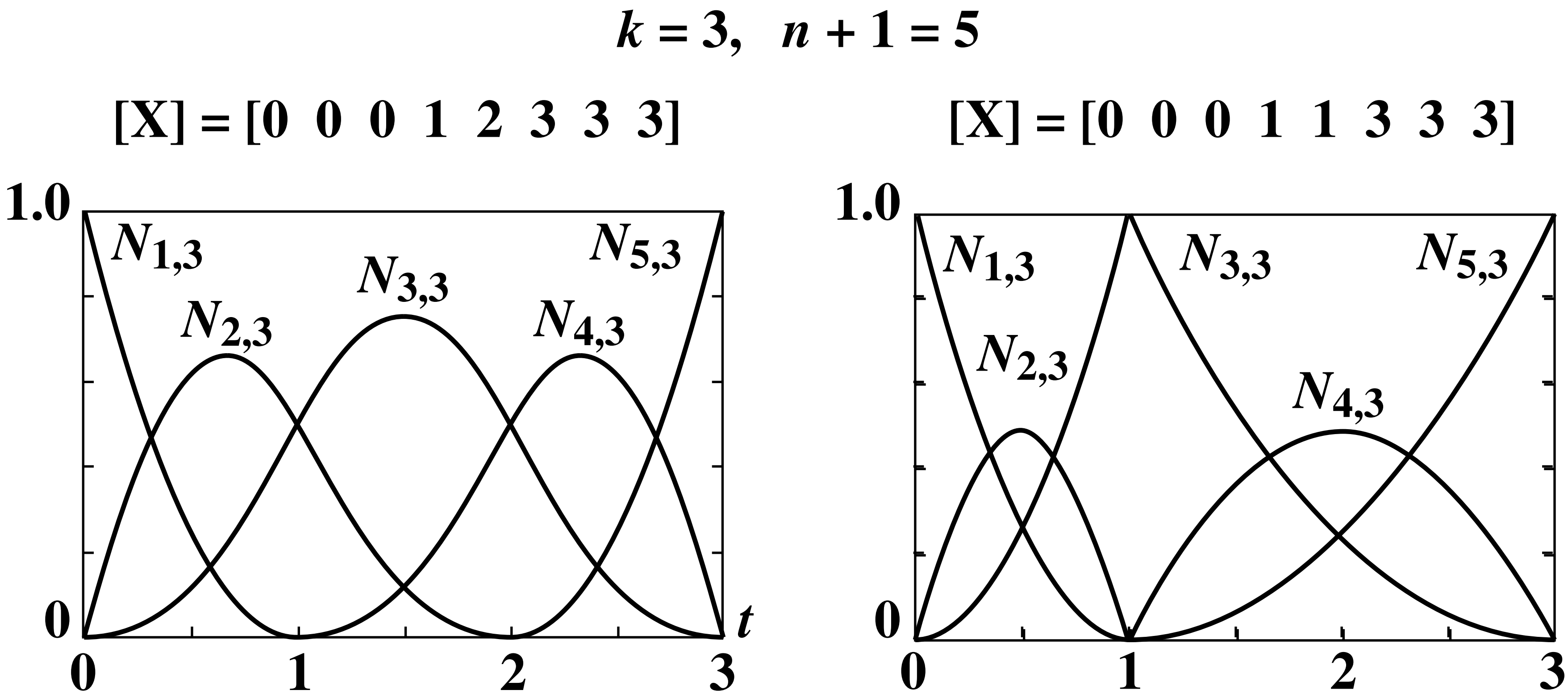

**Notice:** $N_{3,3}$ **pulled right and magnitude increased**

**More influence for** $B_{3,3}$ **control vertex**

**Less for others**

# Basis functions – Comparisons

## Multiple duplicate knot values



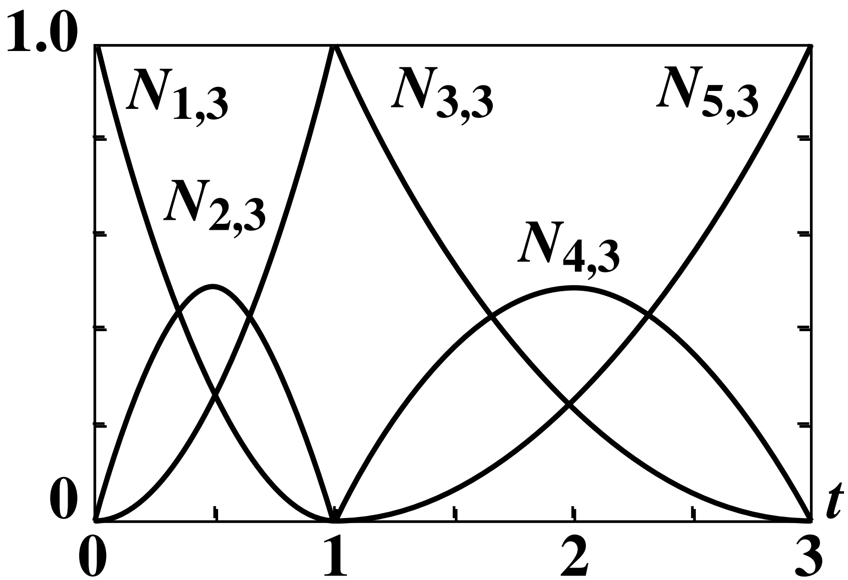

Notice: $N_{3,3} = 1$ at $t = 1$ while all others zero
Curve passes through $B_{3,3}$
Continuity reduced

# Basis functions – Comparisons

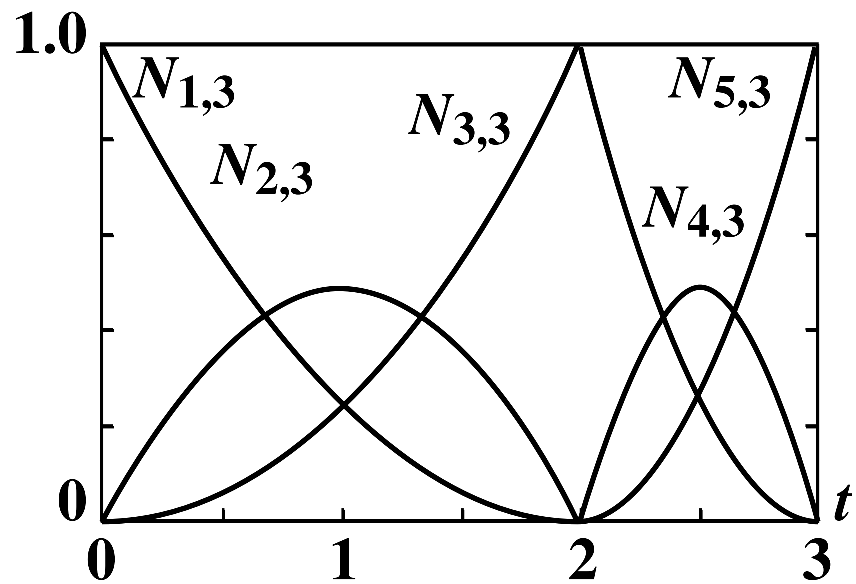
## Multiple duplicate knot values

$k = 3, \quad n + 1 = 5$

$[X] = [0\ 0\ 0\ 1\ 1\ 3\ 3\ 3]$




$[X] = [0\ 0\ 0\ 2\ 2\ 3\ 3\ 3]$




**Notice: Now $N_{3,3} = 1$ at $t = 2$ while all others zero**

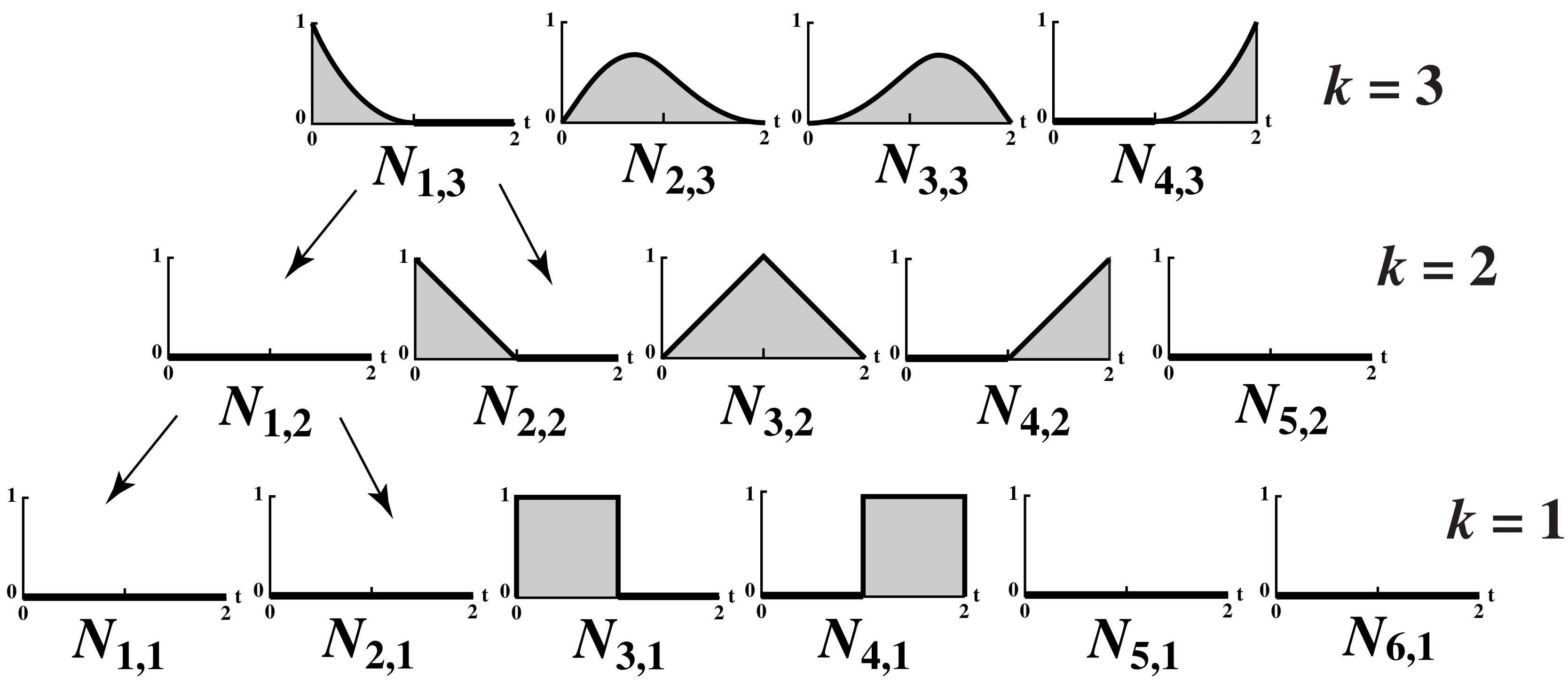
**Curve passes through $B_{3,3}$**

**Continuity reduced**

# Basis functions – Buildup

**Example:** $n+1=4,\quad k=3$

$$[\,X\,]=[0\quad 0\quad \underbrace{0\quad 1\quad 2}_{\text{two spans}}\quad 2\quad 2]$$



# Basis functions — Calculation

**Example:** $n+1=4$, $\quad k=3$

$$[\,X\,] = [0 \quad 0 \quad \underbrace{0 \quad 1 \quad 2}_{\text{two spans}} \quad 2 \quad 2]$$

$0 \le t < 1$

$$N_{3,1}(t)=1;\ N_{i,1}(t)=0,\ i \neq 3$$

$$N_{2,2}(t)=1-t;\ N_{3,2}(t)=t;\ N_{i,2}(t)=0,\ i \neq 2,3$$

$$N_{1,3}(t)=(1-t)^2;\ N_{2,3}(t)=t(1-t)+\frac{(2-t)}{2}t$$

$$N_{3,3}(t)=\frac{t^2}{2};\ N_{i,3}(t)=0,\ i \neq 1,2,3$$

# Basis functions — Calculation

**Example:** $n+1=4$**,** $\quad k=3$

$$[\,X\,]=[0 \quad 0 \quad \underbrace{0 \quad 1 \quad 2}_{\text{two spans}} \quad 2 \quad 2]$$

$1 \leq t < 2$

$N_{4,1}(t)=1;\ N_{i,1}(t)=0,\ i\neq 4$

$N_{3,2}(t)=(2-t);\ N_{4,2}(t)=(t-1);\ N_{i,2}(t)=0,$
$i\neq 3,4$

$$N_{2,3}(t)=\frac{(2-t)^2}{2};$$

$$N_{3,3}(t)=\frac{t(2-t)}{2}+(2-t)(t-1);$$

$N_{4,3}(t)=(t-1)^2;\ N_{i,3}(t)=0,\ i\neq 2,3,4$

# B-spline & Bézier Curves

**If** $k = n + 1$

*and*

**an open knot vector is used**

*then*

**the B-spline curve is a Bézier curve**

**The knot vector is:**

$k$ **zeros followed by** $k$ **ones**

$k = 4, \; [X] = [\,0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1\,]$

# B-spline Curves – Controls

**Change type of knot vector**

**open uniform**

**open nonuniform**

**Change order $k$**

**Change number/position of control vertices**

**Use multiple coincident control vertices**

**Use multiple equal internal knot values**

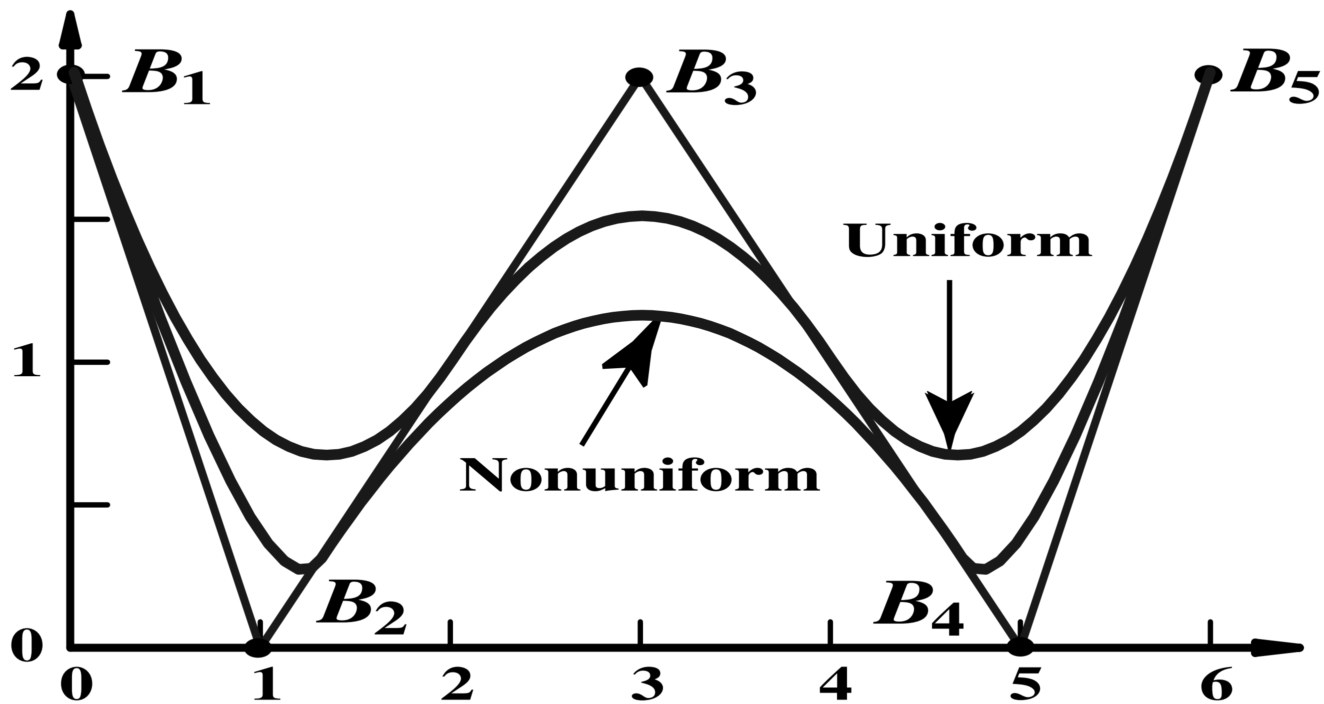
# B-spline Curves – Controls

## Type of knot vector

$n + 1 = 5, \quad k = 3$

$[\,X\,] = [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 3]$ **Open uniform**

$[\,X\,] = [0 \quad 0 \quad 0 \quad 0.4 \quad 2.6 \quad 3 \quad 3 \quad 3]$ **Open nonuniform**



# B-spline Curves – Control

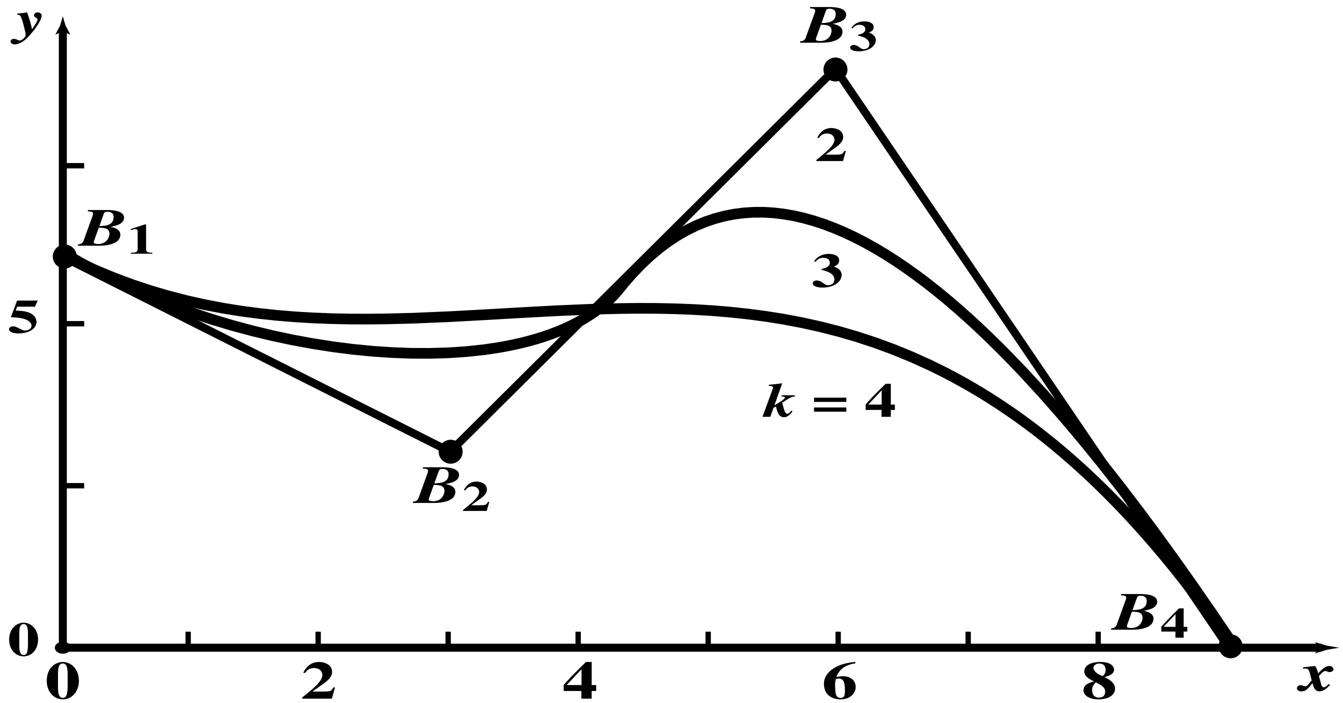
## Change order

$n+1=4$, $\quad k=2,3,4$

$k=2 \qquad [X]=[0 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 3]$

$k=3 \qquad [X]=[0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 2 \;\; 2]$

$k=4 \qquad [X]=[0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1]$



# B-spline Curves – Control

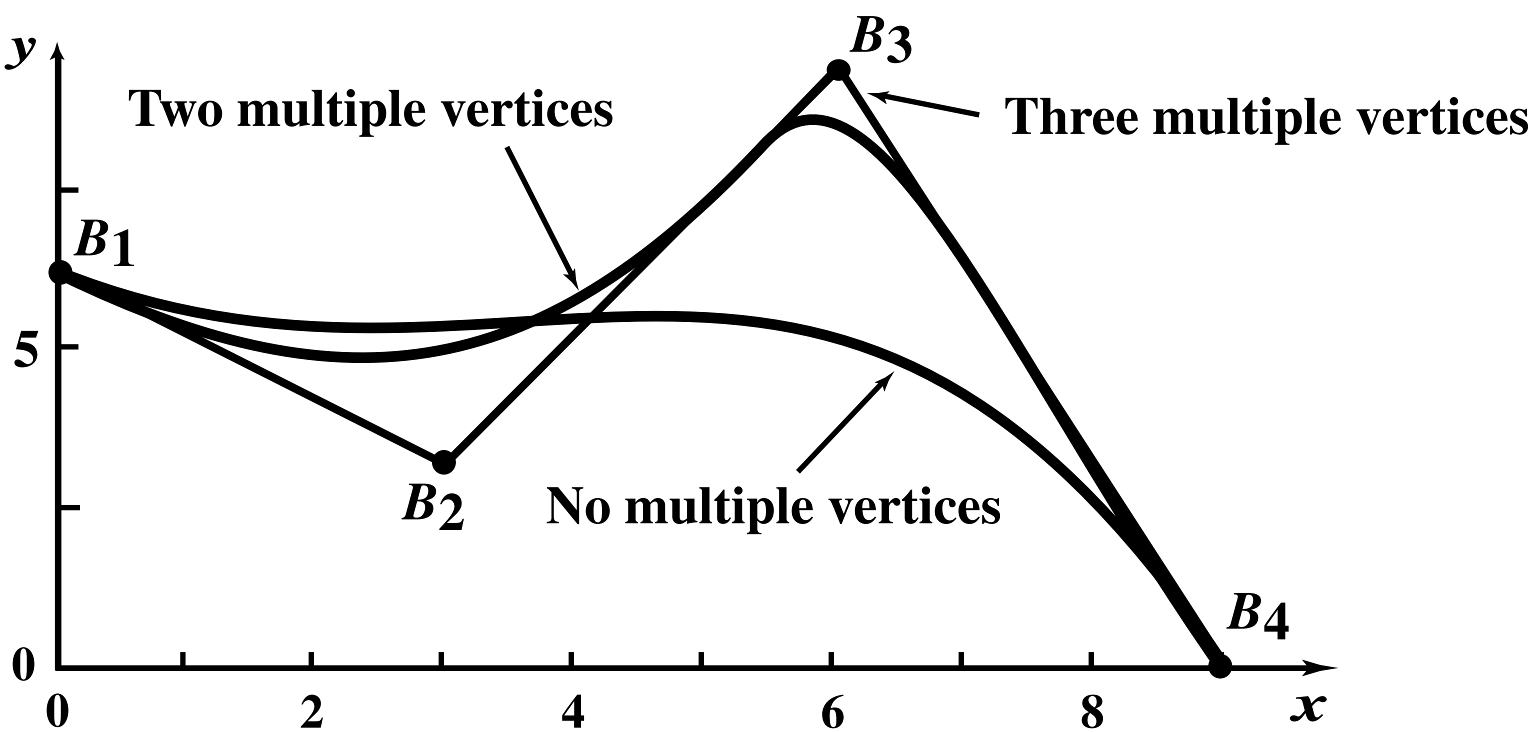
## Multiple Coincident Vertices

$k = 4, \quad n+1 = 4,\ 5,\ 6$

$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1]$ **Single vertex**

$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 2 \;\; 2 \;\; 2]$ **Double vertex**

$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 3 \;\; 3 \;\; 3]$ **Triple vertex**



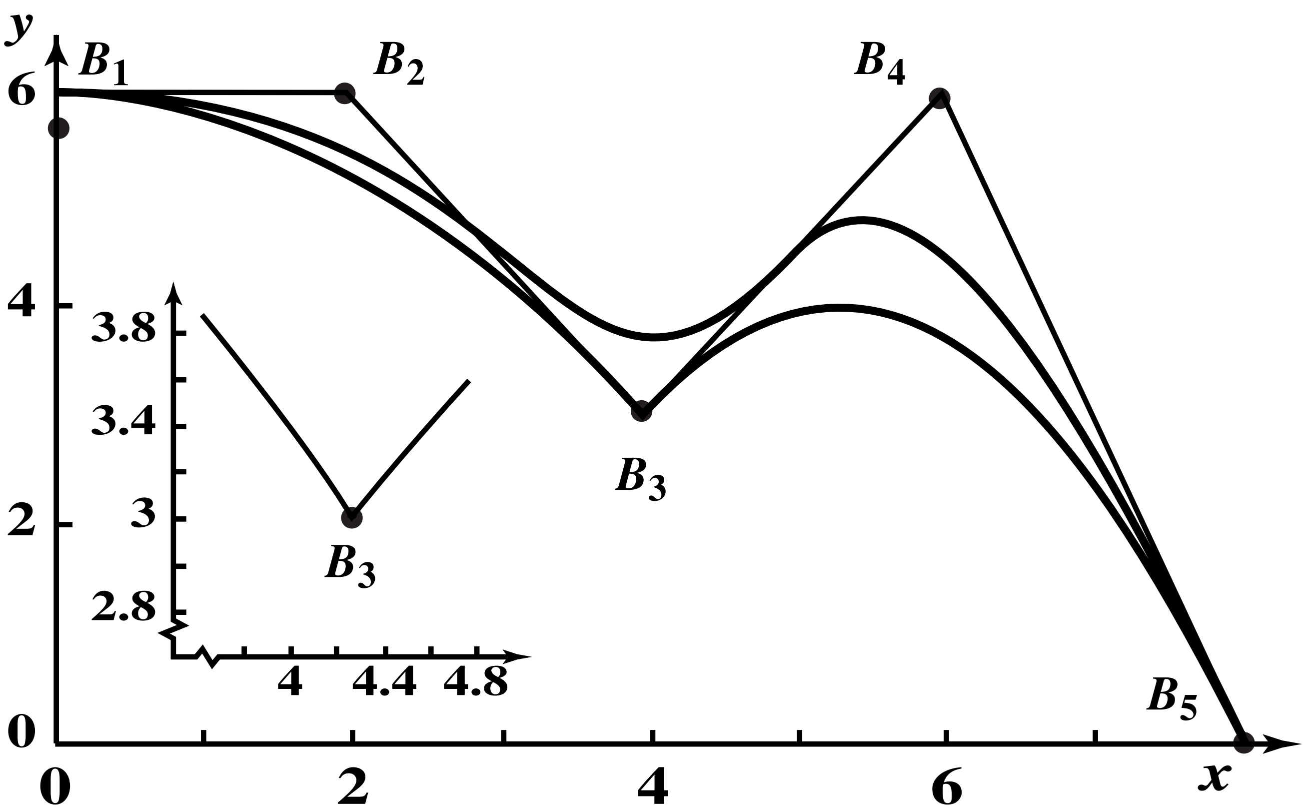
# B-spline Curves – Control

## Multiple Internal Knot Values

$n + 1 = 5, \quad k = 3$

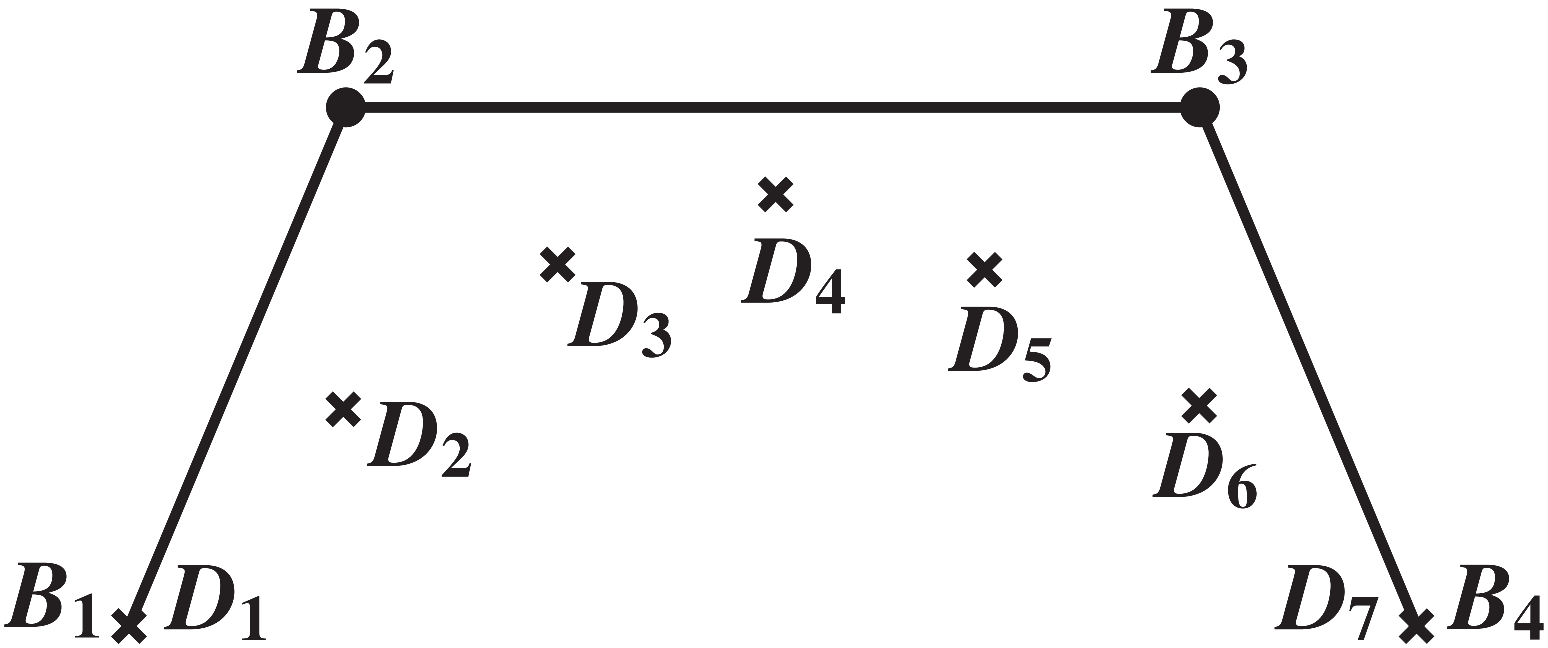
$[\,X\,] = [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 3]$

$[\,X\,] = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 3 \quad 3 \quad 3]$ **Duplicate knot value**



# B-spline Curves – Fitting

**Given a data set find a B-spline curve that "fairs" the data**




**For each data point**

$$D_j(t_j) = N_{1,k}(t_j)B_1 + N_{2,k}(t_j)B_2 + \cdots + N_{n+1,k}(t_j)B_{n+1}$$

**where** $2 \le k \le n+1 \le j$

# B-spline Curves – Fitting

**Writing the equation for each of $j$ data points yields**

$$D_1(t_1) = N_{1,k}(t_1)B_1 + N_{2,k}(t_1)B_2 + \cdots + N_{n+1,k}(t_1)B_{n+1}$$

$$D_2(t_2) = N_{1,k}(t_2)B_1 + N_{2,k}(t_2)B_2 + \cdots + N_{n+1,k}(t_2)B_{n+1}$$

$$\vdots$$

$$D_j(t_j) = N_{1,k}(t_j)B_1 + N_{2,k}(t_j)B_2 + \cdots + N_{n+1,k}(t_j)B_{n+1}$$

**where** $2 \leq k \leq n+1 \leq j$

**This system of equations is more compactly written in matrix form**

# B-spline Curves – Fitting

## In matrix form

$$[\,D\,] = [\,N\,][\,B\,]$$

**where**

$$[\,D\,]^T = [\,D_1(t_1) \quad D_2(t_2) \quad \cdots \quad D_j(t_j)\,]$$

$$[\,B\,]^T = [\,B_1 \quad B_2 \quad \cdots \quad B_{n+1}\,]$$

$$[\,N\,] = \begin{bmatrix} N_{1,k}(t_1) & \cdots & \cdots & N_{n+1,k}(t_1) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ N_{1,k}(t_j) & \cdots & \cdots & N_{n+1,k}(t_j) \end{bmatrix}$$

## Three problems

# B-spline Curves – Fitting

## Parameter value for each $D_j(t_j)$

## A useful approximation is the chord distance

$$t_1 = 0$$

$$\frac{t_\ell}{t_{\max}} = \frac{\sum_{\mathrm{s}=2}^{\ell} |D_{\mathrm{s}} - D_{\mathrm{s}-1}|}{\sum_{\mathrm{s}=2}^{j} |D_{\mathrm{s}} - D_{\mathrm{s}-1}|} \qquad \ell \geq 2$$

**The maximum parameter value, $t_{\max}$, is usually the maximum value of the knot vector**

# B-spline Curves – Fitting

**Number control vertices equals number of data points,** $2 \leq k \leq n+1 = j$

$$[\,D\,] = [\,N\,]\,[\,B\,]$$

$[N]$ **is square**

**Control polygon is obtained directly by matrix inversion.**

$$[\,B\,] = [\,N\,]^{-1}\,[\,D\,] \qquad 2 \leq k \leq n+1 = j$$

**The resulting B-spline curve passes through each data point.**

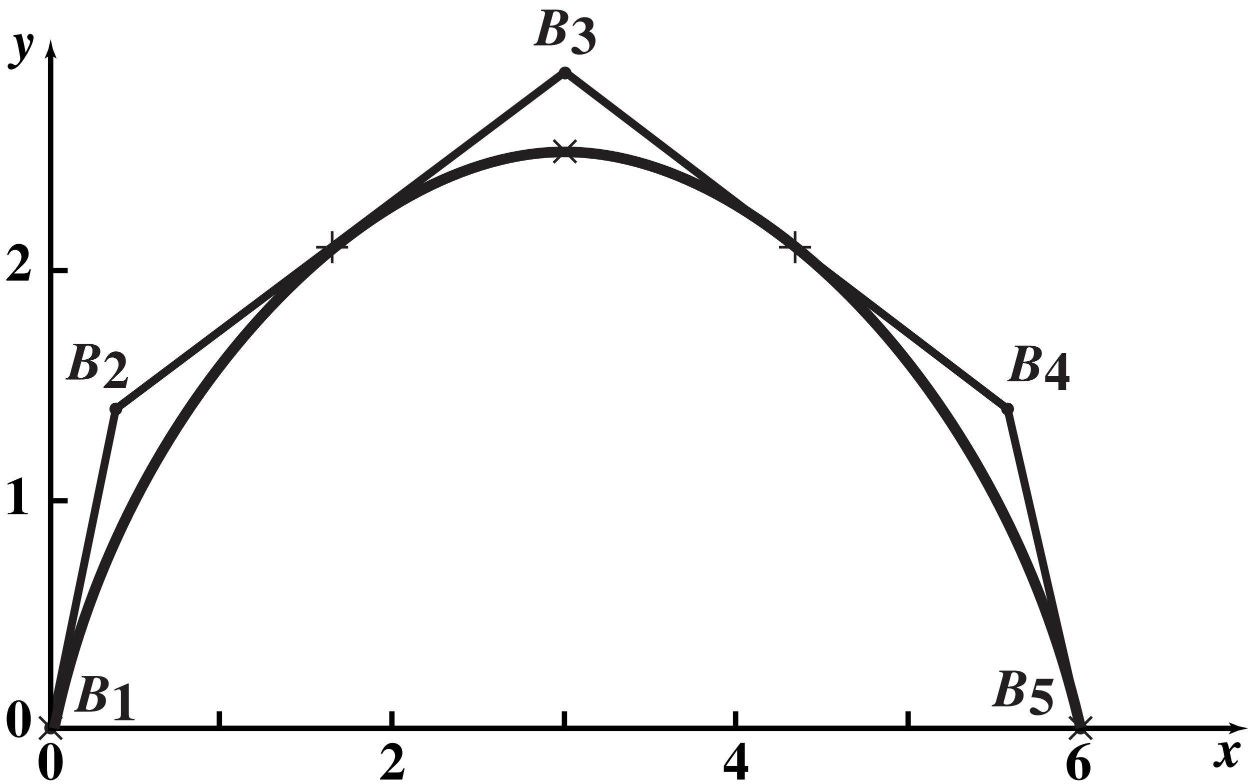
**But, it may wiggle.**

# B-spline Curves – Fitting

## Example

**Number control vertices is equal to the number of data points,** $2 \le k \le n+1 = j$

$k = 3$**,** $n+1 = 5$**,** $j = 5$ $[N]$ **is square**




**Curve passes through each data point**

# B-spline Curves - Fitting

**Number control vertices is less than the number of data points,** $2 \leq k \leq n+1 < j$

$$[D] = [N][B]$$

$[N]$ **is NOT square**

**Control polygon is obtained in a mean sense**

$$[N]^T [D] = [N]^T [N][B]$$

$$[B] = [[N]^T [N]]^{-1} [N]^T [D]$$

**The resulting B-spline curve does NOT pass through each data point**

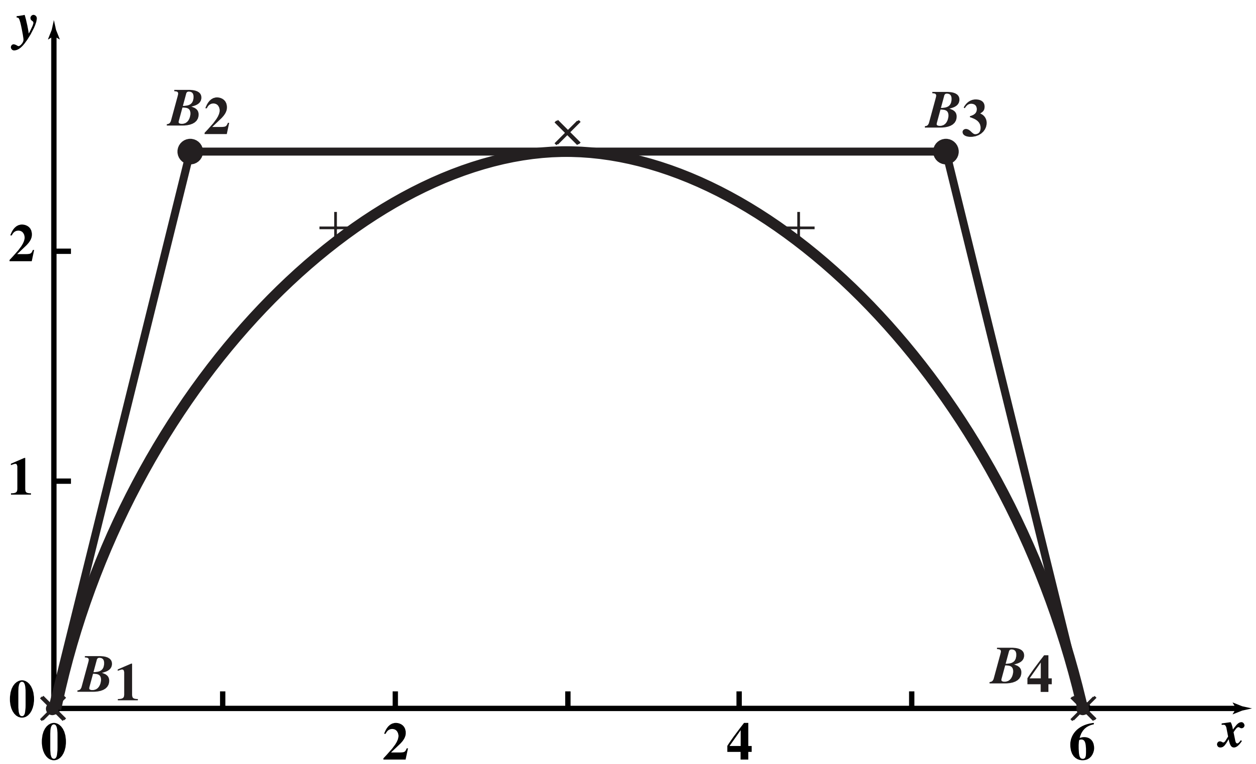
**The curve is "faired" through the data points**

# B-spline Curves – Fitting

## Example

**Number control vertices is not equal to the number of data points,** $2 \leq k \leq n+1 < j$

$k = 3$, $n+1 = 4$, $j = 5$ $[N]$ **is not square**




**Curve does not pass through each data point**

# B-spline Curves - Conic sections

**Nonrational B-spline curves cannot precisely represent the conic sections**

**Circles**

**Ellipses**

**Parabolas**

**Hyperbolas**

# B-spline Curves - Additional Topics

Degree elevation

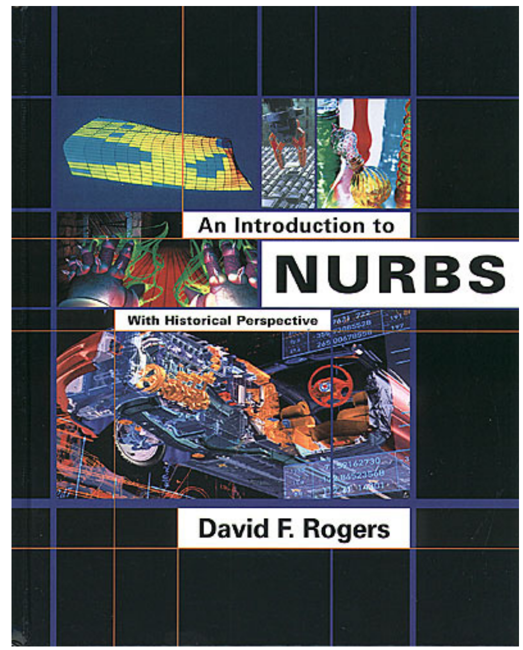
Degree reduction

Knot insertion

Subdivision

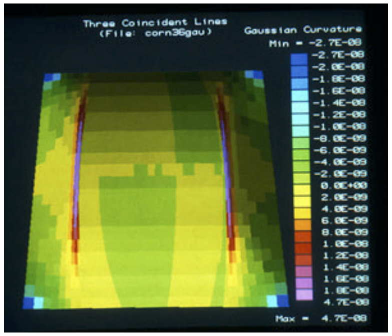
Reparameterization

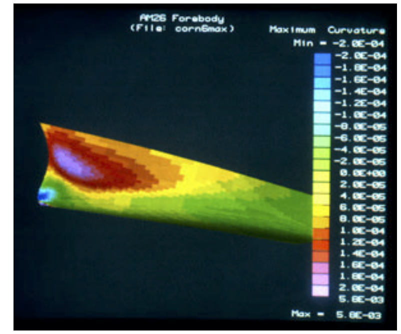
Additional reading:




Rogers, D. F.
An Introduction to NURBS,
With Historical Perspective
Morgan Kaufmann Publishers, 2001

Piegl, L. & Tiller, W.
The NURBS Book, Springer-Verlag 1995




# Course 31
# NURBS
# (NonUniform Rational B-splines)

## Part 2 (9:30)
## Rational B-spline Curves (NURBS)

**David F. Rogers**
**Professor of Aerospace Engineering**
**United States Naval Academy**
**Annapolis, MD 21402**

# Rational B-spline Curves (NURBS)

**Bézier and nonrational B-splines are a subset (special case) of rational B-splines (NURBS)**

**Rational B-splines provide a single precise mathematical form for:**

- **lines**
- **planes**
- **conic sections (circles, ellipses . . .)**
- **free form curves**
- **quadric surfaces**
- **sculptured surfaces**

# Rational B-splines

Ken Versprille

**First to discuss rational B-splines**
**PhD dissertation at Syracuse University**

# Rational B-spline curves – Definition

**Defined in 4-D homogeneous coordinate space**
**Projected back into 3-D physical space**

**In 4-D**

$$P(t) = \sum_{i=1}^{n+1} B_i^h N_{i,k}(t)$$

**where**

| | |
|---|---|
| $B_i^h$**s** | **are the 4-D homogeneous control vertices** |
| $N_{i,k}(t)$**s** | **are the nonrational B-spline basis functions** |
| $k$ | **is the order of the basis functions** |

# Rational B-spline curves – Definition

**Projected back into 3-D physical space**

**Divide through by homogeneous coordinate**

$$P(t) = \frac{\sum_{i=1}^{n+1} B_i h_i N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}(t)} = \sum_{i=1}^{n+1} B_i R_{i,k}(t)$$

$B_i$**s are the 3-D control vertices**

$$R_{i,k}(t) = \frac{h_i N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}(t)} \qquad h_i \geq 0$$

$R_{i,k}(t)$**s are the rational B-spline basis functions**

# Rational B-spline curves – Properties

$\sum_{i=1}^{n+1} R_{i,k}(t) \equiv 1$ **for all** $t$

$R_{i,k}(t) \geq 0$ **for all** $t$

$R_{i,k}(t), \quad k > 1$ **has precisely one maximum**

$k_{max} = n + 1$

**Maximum degree** $= n$

**Exhibits variation diminishing property**

**Follows shape of the control polygon**

**Transform curve – transform control polygon**

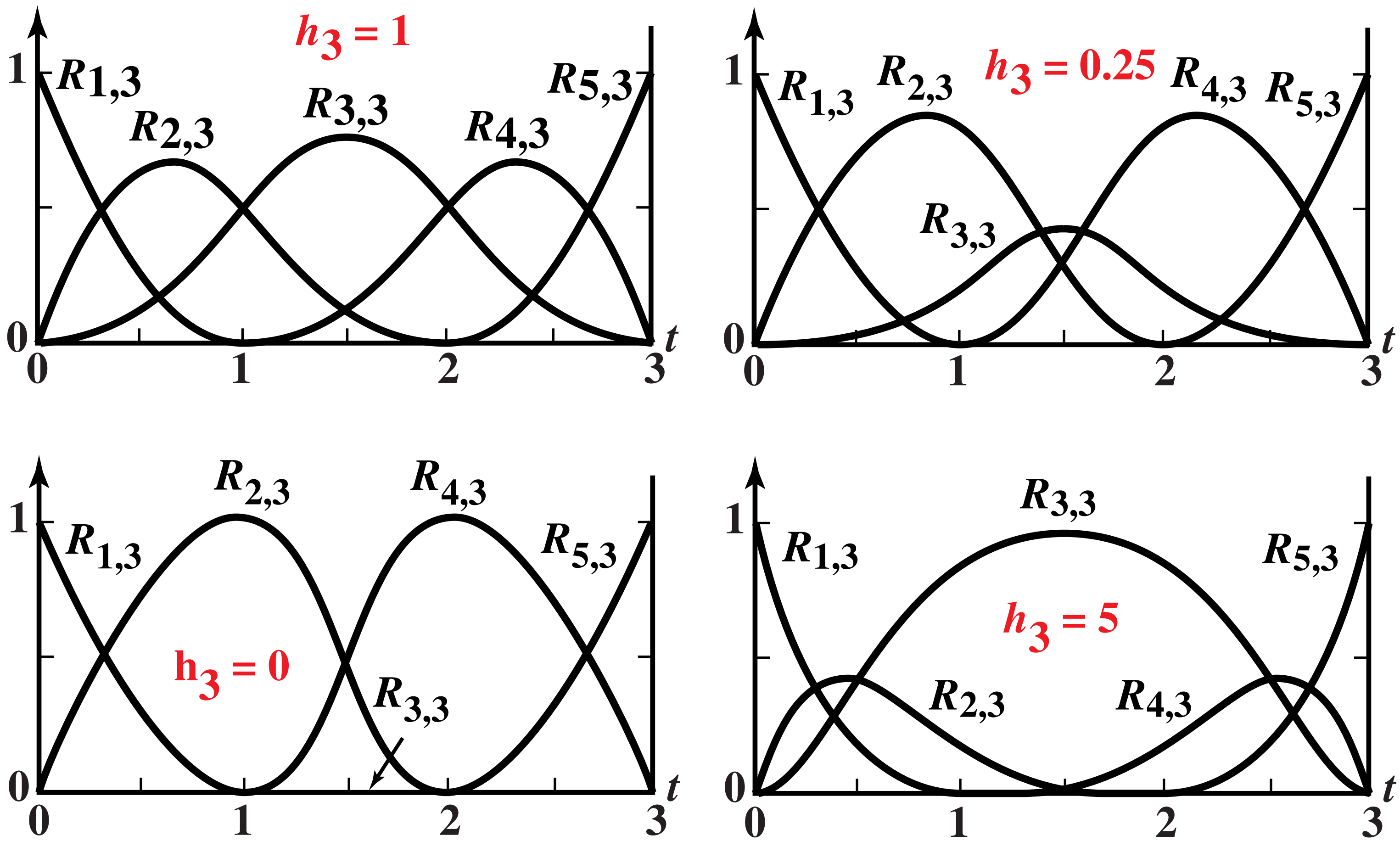
**Lies within union of convex hulls of** $k$
**successive control vertices if** $h_i > 0$

**Everywhere** $C^{k-2}$ **continuous**

# Rational B-spline basis functions

## Comparisons: $n+1=5, \quad k=3$

$$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 3 \;\; 3], \qquad [\,H\,] = [1 \;\; 1 \;\; h_3 \;\; 1 \;\; 1]$$



# Rational B-spline curves – Control

Same as nonrational B-splines

plus

Manipulation of the homogeneous weighting factor

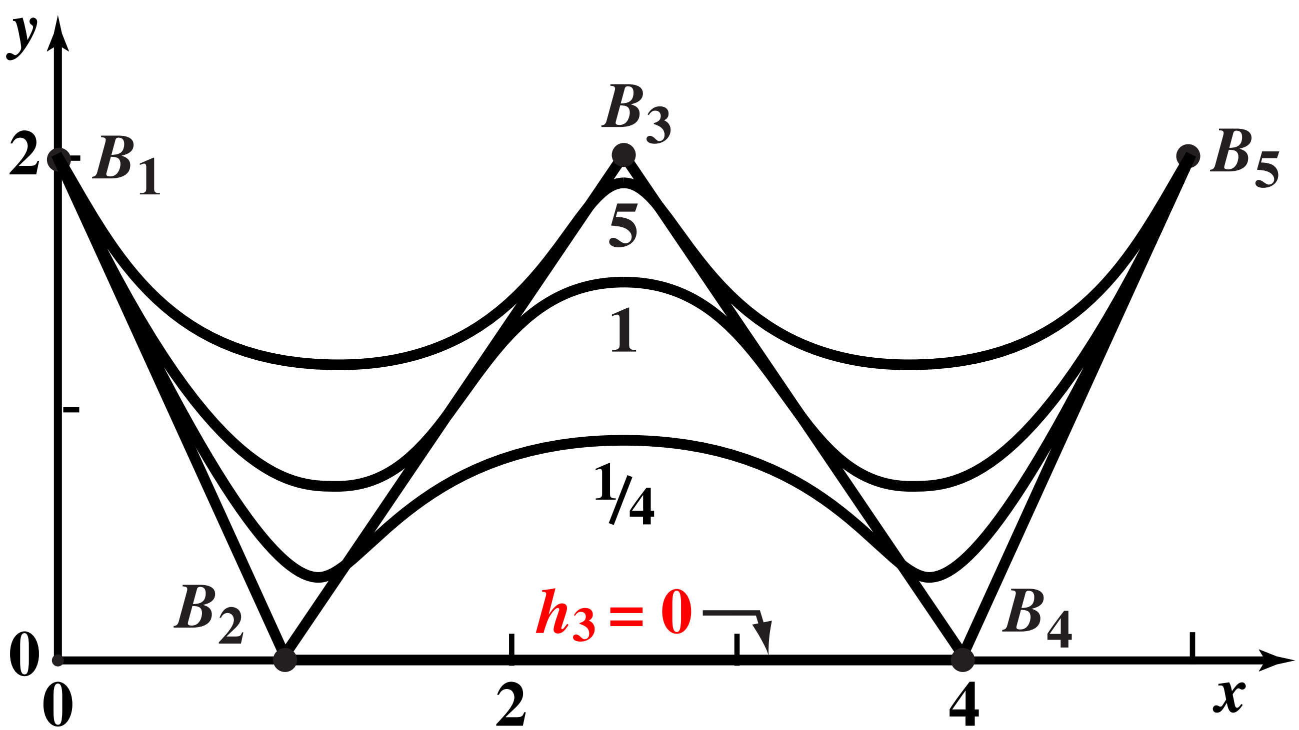
# Rational B-spline Curves – Control

## Homogeneous weighting factor

$n + 1 = 5, \quad k = 3$

$[\,X\,] = [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 3]$

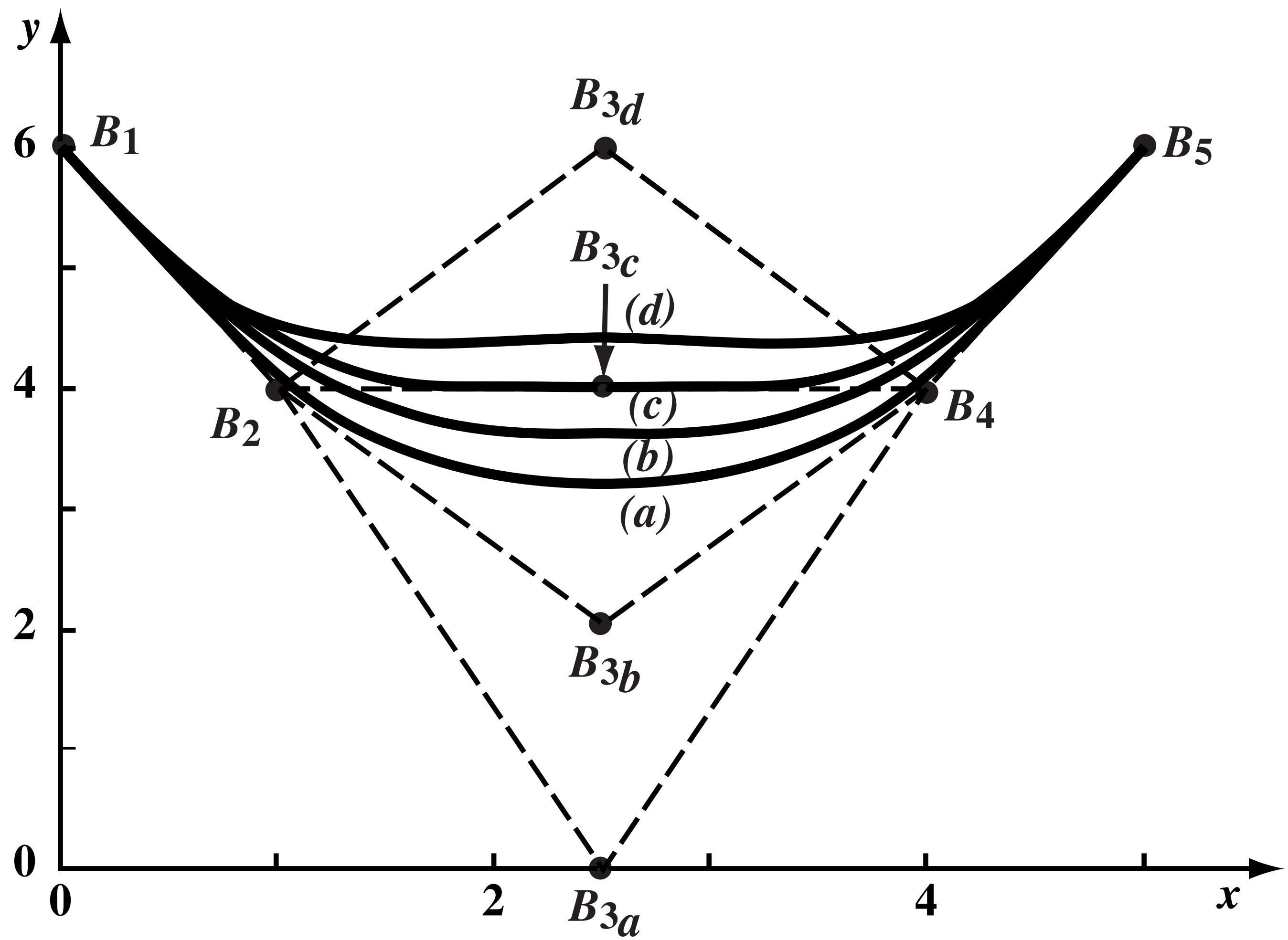
$[\,H\,] = [1 \quad 1 \quad h_3 \quad 1 \quad 1\,]$



# Rational B-spline Curves – Control

## Move single vertex, $n + 1 = 5$, $k = 4$

$$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 2 \;\; 2 \;\; 2], \qquad [\,H\,] = [1 \;\; 1 \;\; 1/4 \;\; 1 \;\; 1]$$



# Rational B-spline Curves – Control

## Multiple vertices

$[\,X\,] = [0\;\;0\;\;0\;\;0\;\;1\;\;2\;\;2\;\;2\;\;2]$ $n+1 = 5$, $k = 4$

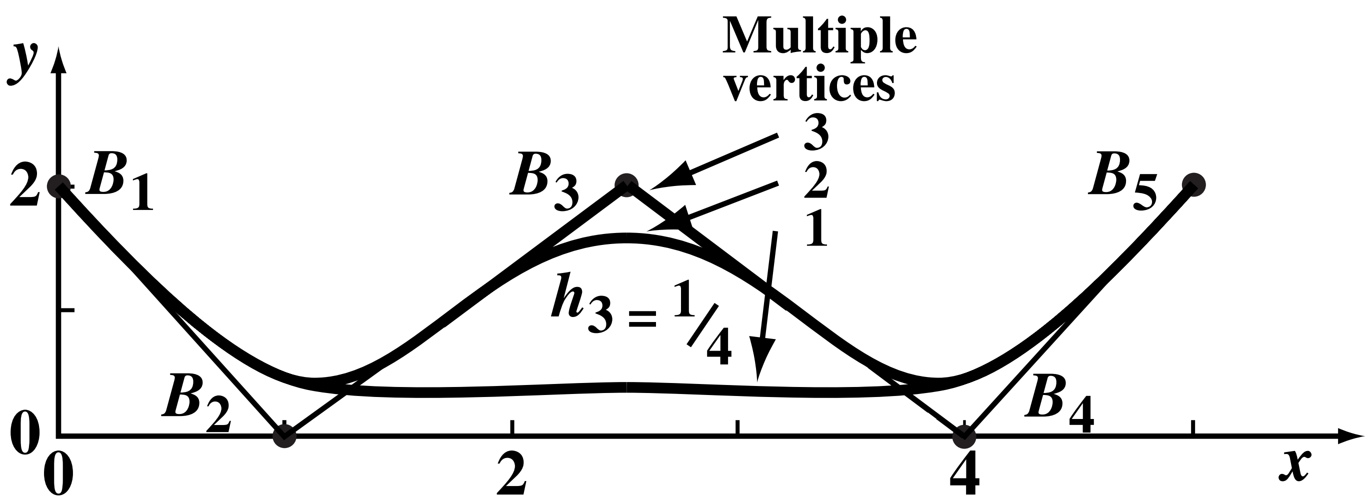
$[\,H\,] = [1\;\;1\;\;{}^1\!/_4\;\;1\;\;1]$ **single vertex**

$[\,X\,] = [0\;\;0\;\;0\;\;0\;\;1\;\;2\;\;3\;\;3\;\;3\;\;3]$ $n+1 = 6$, $k = 4$

$[\,H\,] = [1\;\;1\;\;{}^1\!/_4\;\;{}^1\!/_4\;\;1\;\;1]$ **double vertex**

$[\,X\,] = [0\;\;0\;\;0\;\;0\;\;1\;\;2\;\;3\;\;4\;\;4\;\;4\;\;4]$ $n+1 = 7$, $k = 4$

$[\,H\,] = [1\;\;1\;\;{}^1\!/_4\;\;{}^1\!/_4\;\;{}^1\!/_4\;\;1\;\;1]$ **triple vertex**



# Rational B-spline Curves – Conic Sections

**Conic sections described by quadratic curves**

**Consider quadratic rational B-spline**

$[\,X\,] = [0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1]$; $n+1=3$, $k=3$

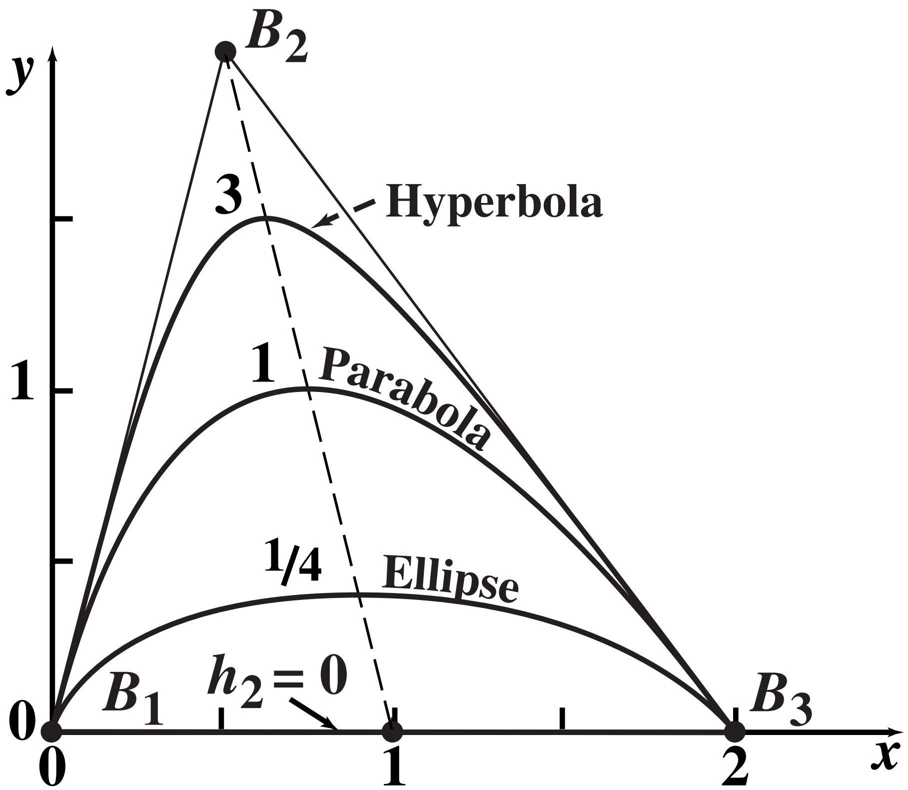
$$P(t) = \frac{h_1 N_{1,3}(t) B_1 + h_2 N_{2,3}(t) B_2 + h_3 N_{3,3}(t) B_3}{h_1 N_{1,3}(t) + h_2 N_{2,3}(t) + h_3 N_{3,3}(t)}$$

**A third-order rational Bézier curve**

**Convenient to assume** $h_1 = h_3 = 1$

$$P(t) = \frac{N_{1,3}(t) B_1 + h_2 N_{2,3}(t) B_2 + N_{3,3}(t) B_3}{N_{1,3}(t) + h_2 N_{2,3}(t) + N_{3,3}(t)}$$

# Rational B-spline Curves – Conic Sections




| | |
|---|---|
| $h_2 = 0$ | **a straight line** |
| $0 < h_2 < 1$ | **an elliptic curve segment** |
| $0 < h_2 = 1$ | **a parabolic curve segment** |
| $h_2 > 1$ | **a hyperbolic curve segment** |

# Rational B-spline Curves

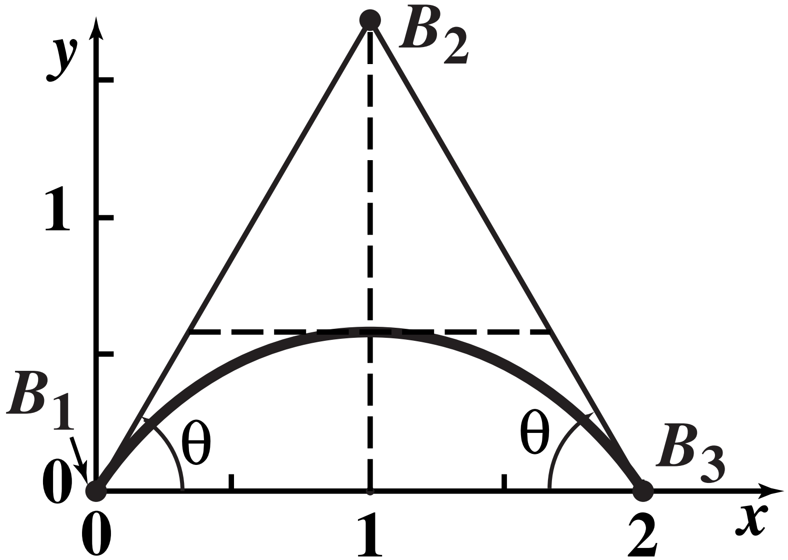
**Conic Sections – Circles**

**Control vertices form isosceles triangle**

**Multiple internal knot values**

**Specific value of the homogeneous weight,** $h_2 = 1/2$

$n+1 = 3,\ k = 3, \quad [\,X\,] = [\,0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\,], \quad [\,H\,] = [\,1\ \ {}^1\!/\!_2\ \ 1\,]$




**Full circle uses three 120° arcs**

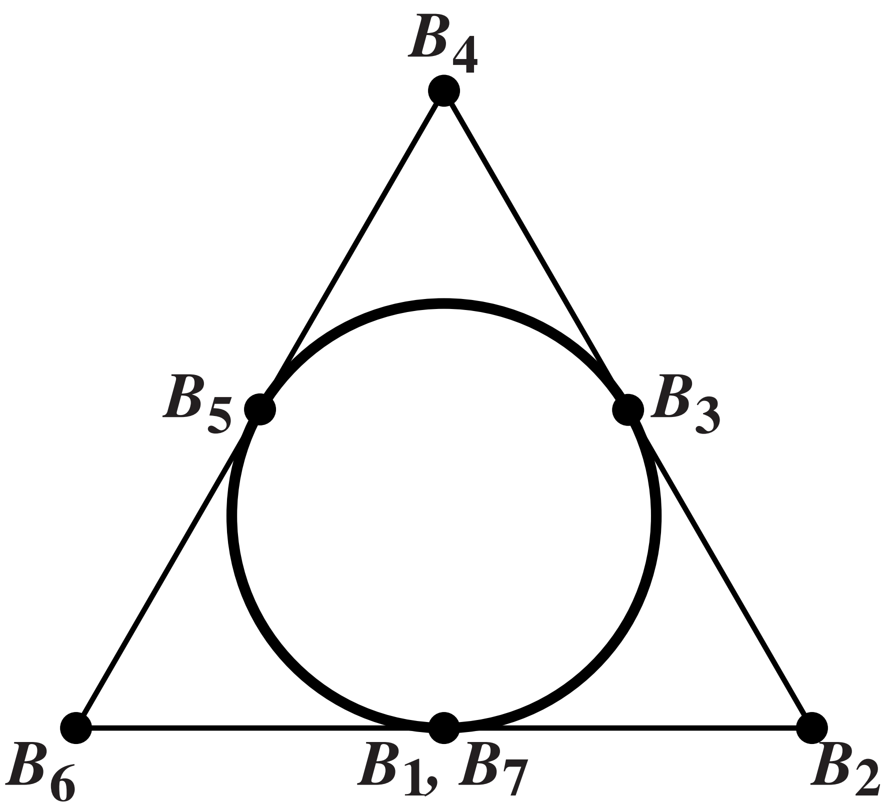
# Rational B-spline Curves

## Conic Sections – Circles

## Three 120° arcs

$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 2 \;\; 2 \;\; 3 \;\; 3 \;\; 3]$; $k = 3$

$[\,H\,] = [1 \;\; 1/2 \;\; 1 \;\; 1/2 \;\; 1 \;\; 1/2 \;\; 1\,]$



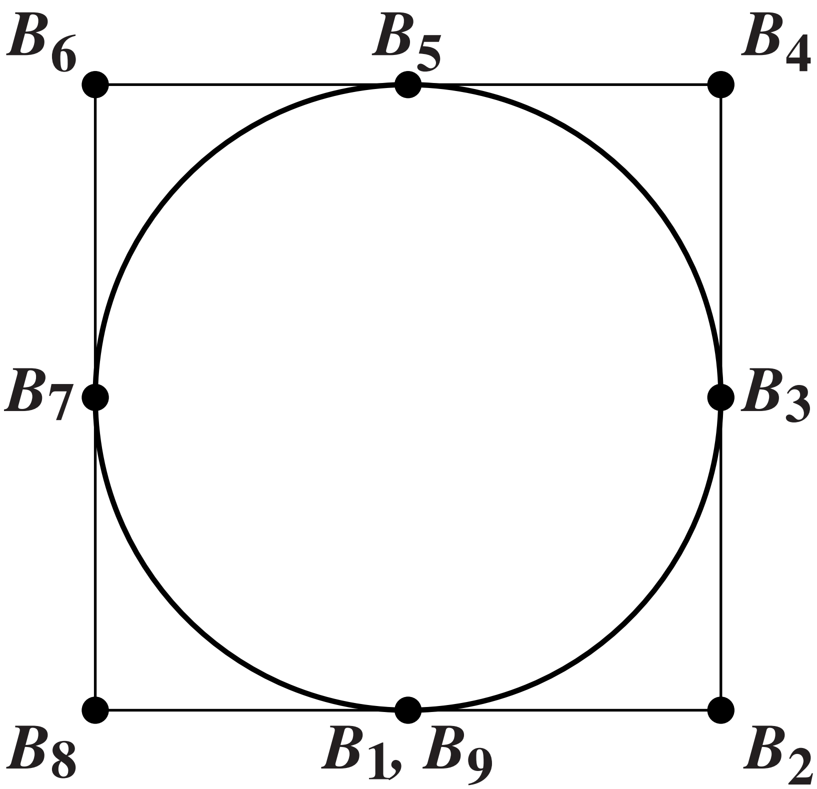
# Rational B-spline Curves

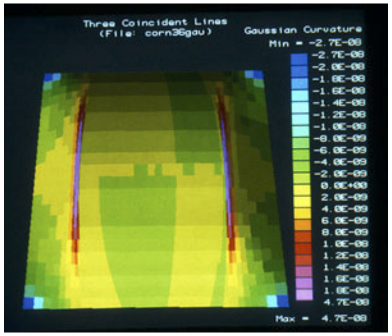
## Conic Sections – Circles

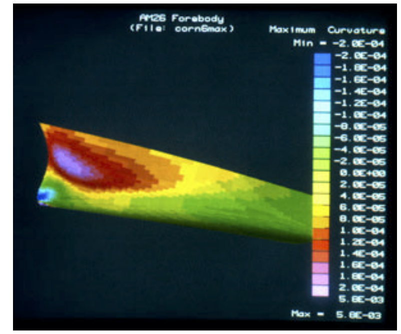
### Four 90° arcs

$[\,X\,] = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 2 \;\; 2 \;\; 3 \;\; 3 \;\; 4 \;\; 4 \;\; 4]$; $k = 3$

$[\,H\,] = [1 \;\; \sqrt{2}/2 \;\; 1 \;\; \sqrt{2}/2 \;\; 1 \;\; \sqrt{2}/2 \;\; 1 \;\; \sqrt{2}/2 \;\; 1\,]$





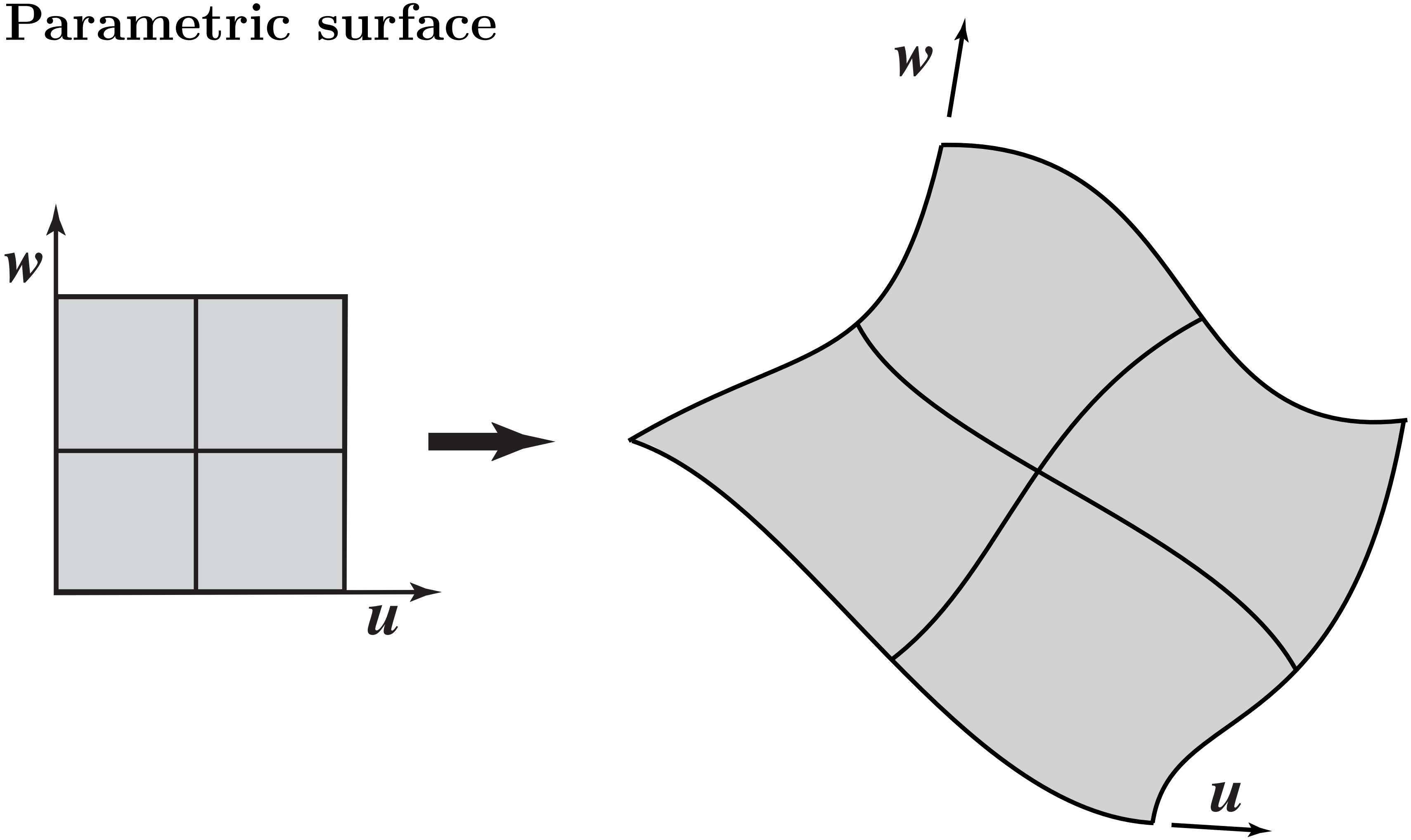

# Course 31
# NURBS
# (NonUniform Rational B-splines)

## Part 3 (10:30)
## Bézier and B-spline Surfaces

David F. Rogers
Professor of Aerospace Engineering
United States Naval Academy
Annapolis, MD 21402

# Bézier Surfaces

## Parametric surface




**Two degrees of freedom:** $u, w$

# Bézier Surfaces – Definition

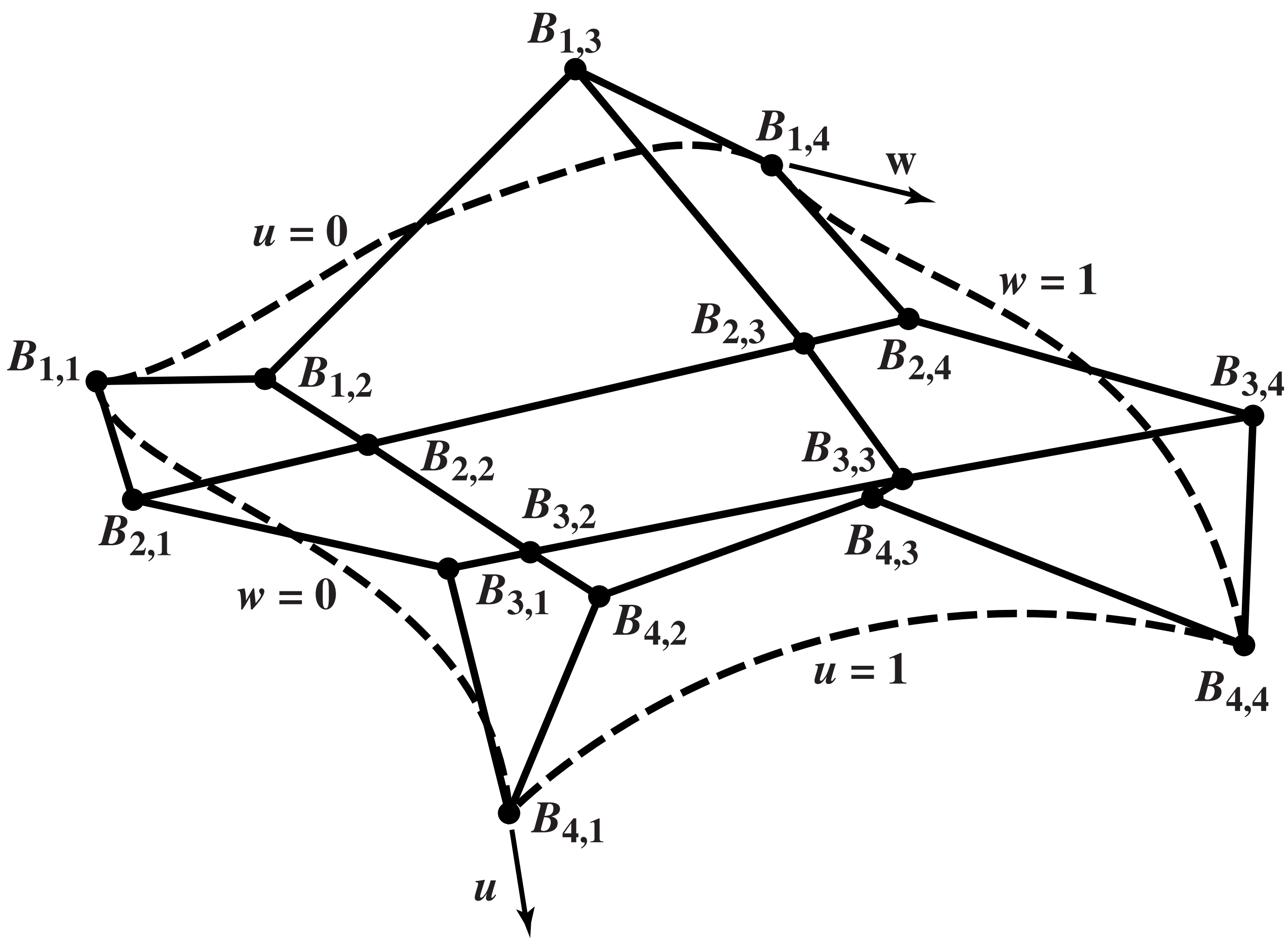
$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} J_{n,i}(u) K_{m,j}(w); \quad 0! \equiv 1; \quad 0^0 \equiv 1$$

**where**

$$J_{n,i}(u) = \binom{n-1}{i-1} u^{i-1}(1-u)^{n-i}; \quad K_{m,j}(w) = \binom{m-1}{j-1} w^{j-1}(1-w)^{m-j}$$

**with** $$\binom{n-1}{i-1} = \frac{(n-1)!}{(i-1)!(n-i)!}; \quad \binom{m-1}{j-1} = \frac{(m-1)!}{(j-1)!(m-j)!}$$

# Bézier Surfaces – Polygon Net




**Not necessarily square**

**Each boundary is a Bézier curve**

# Bézier surfaces – Characteristics

Degree one less than number of control vertices in each parametric direction

Continuity two less than number of control vertices in each parametric direction

Surface follows shape of control net

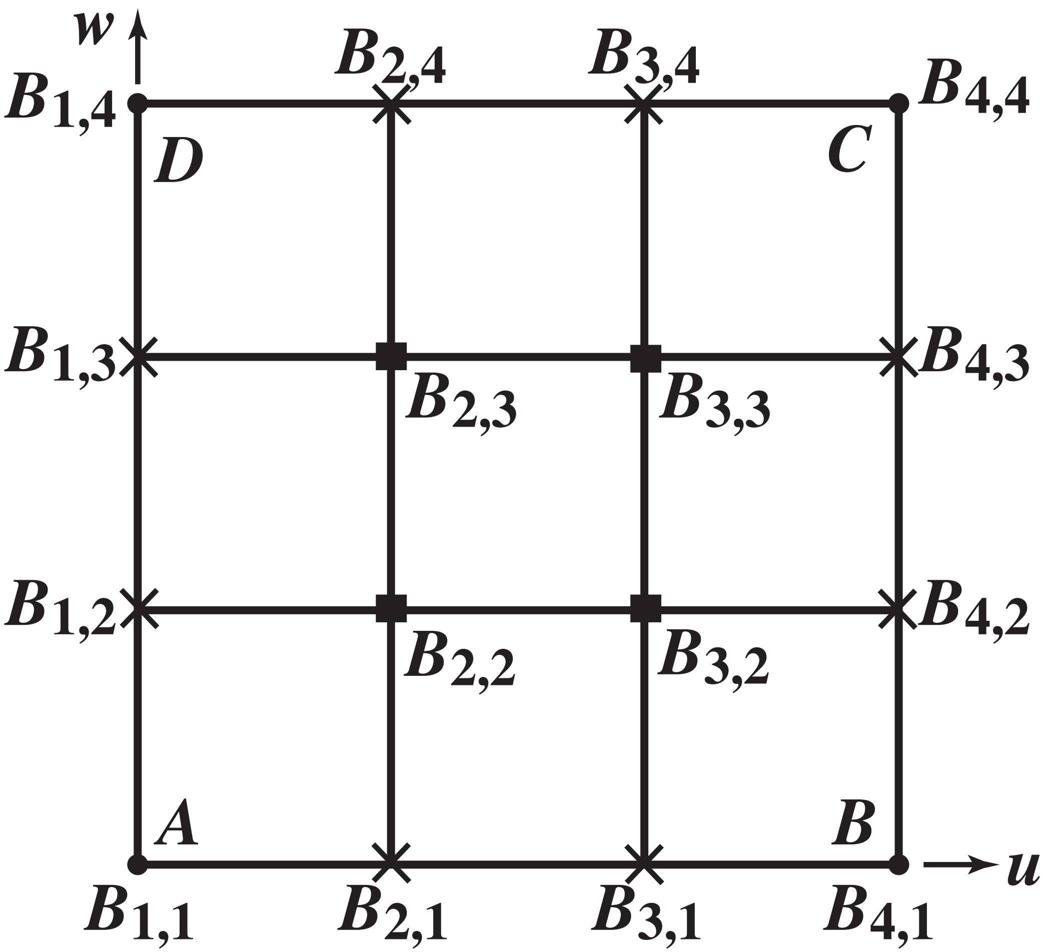
Surface and control net coincident only at corner points

Surface lies within convex hull of control net

Does not exhibit variation diminishing property

Transform surface – transform control net

# Bézier Surfaces – Controls

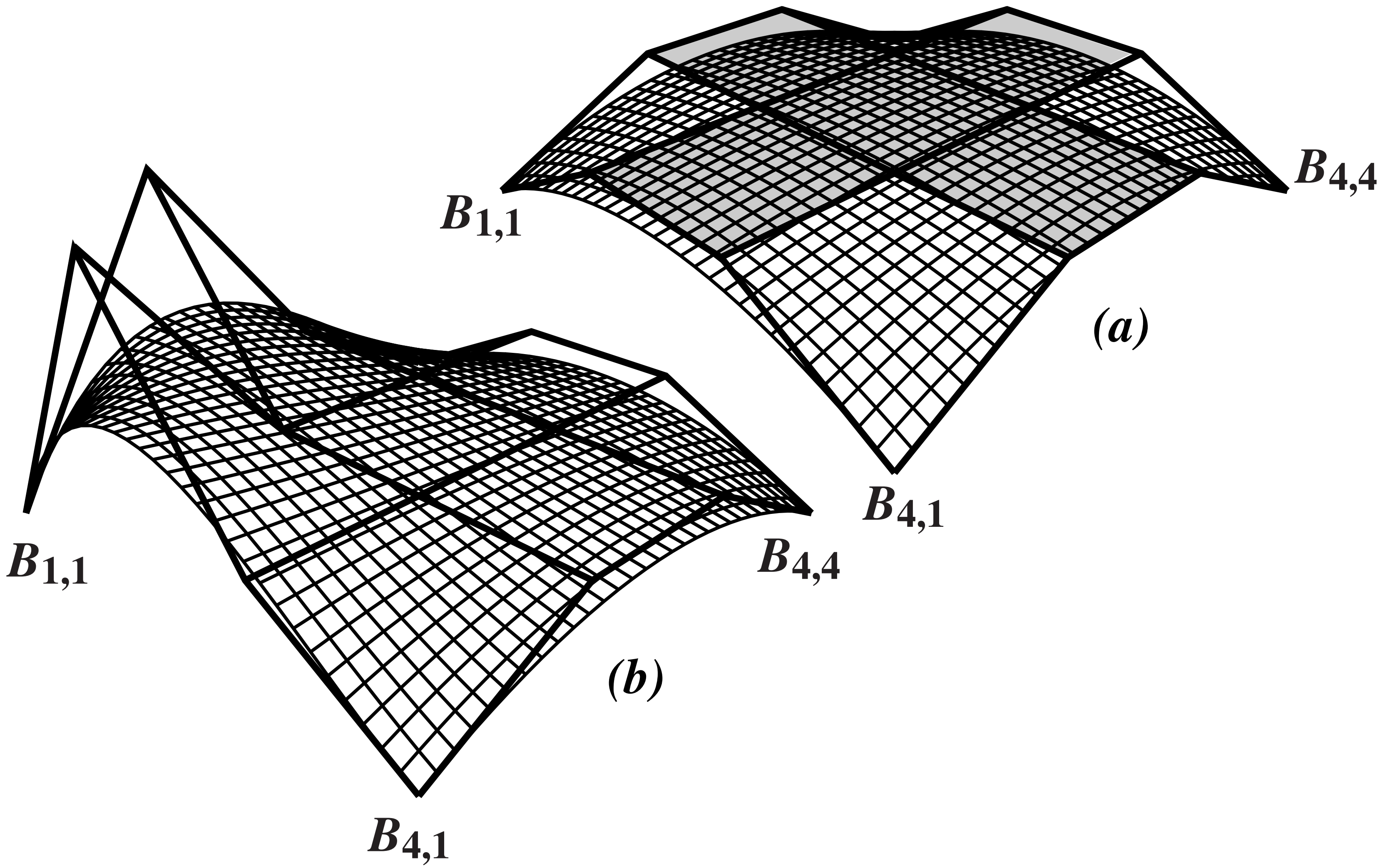



× control tangent vectors at corners

■ influence "internal curvature" at corners

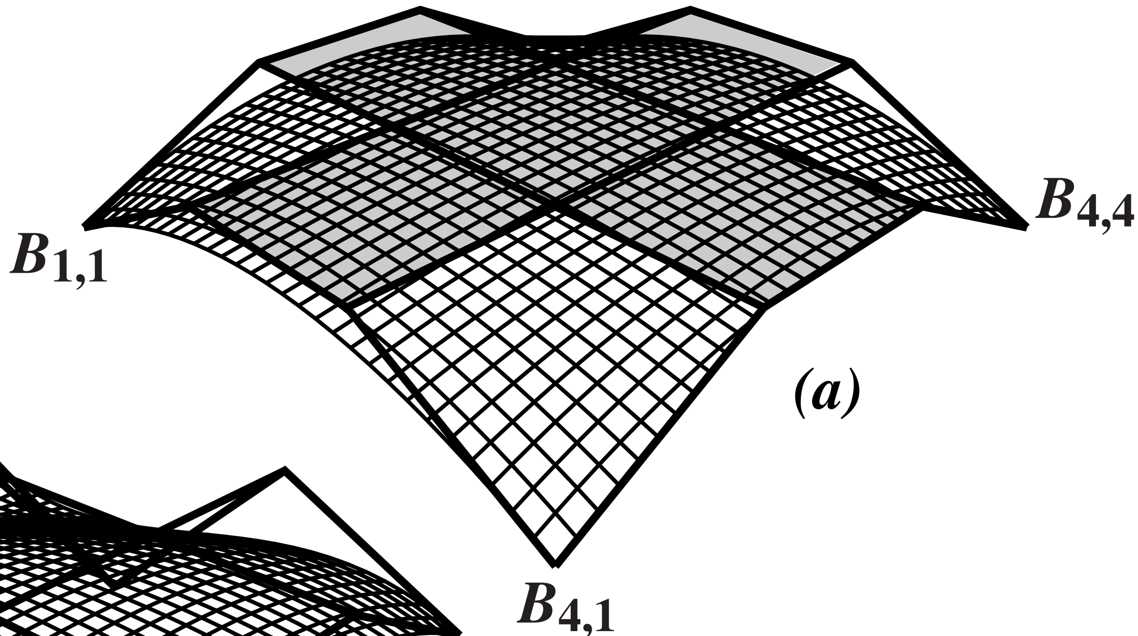
# Bézier Surfaces – Controls

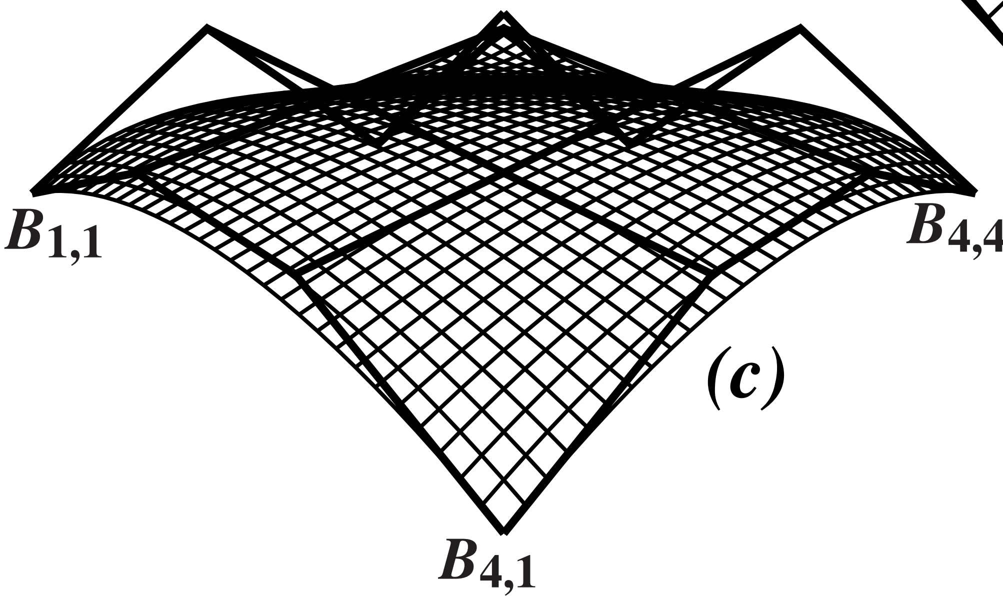
## Effect of tangent vector magnitude



# Bézier Surfaces – Controls

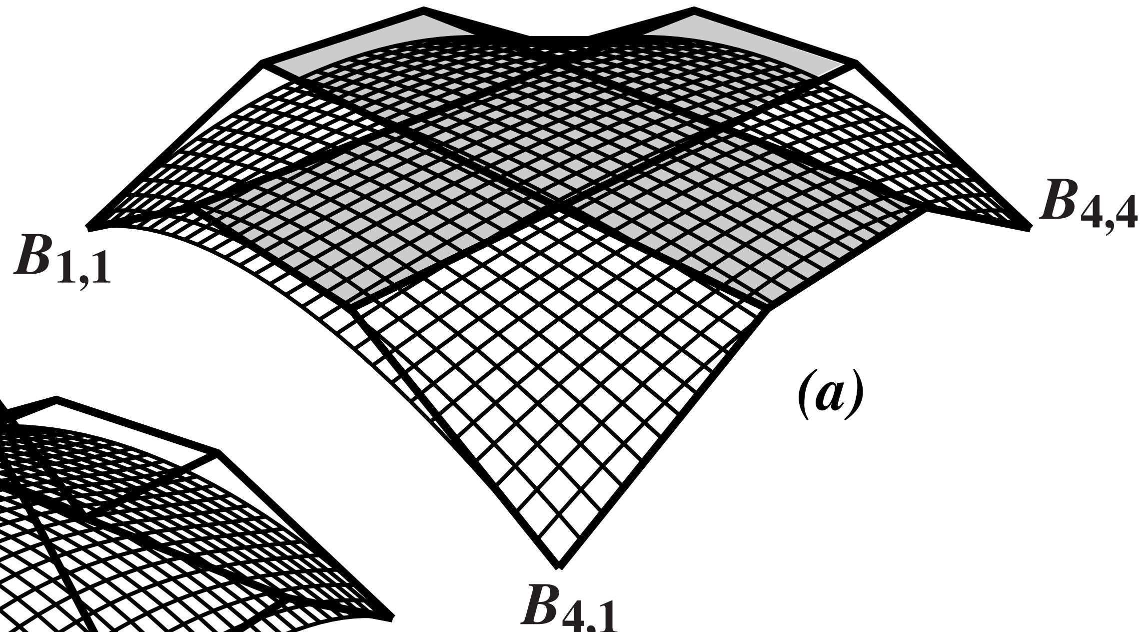
## Effect of tangent vector direction



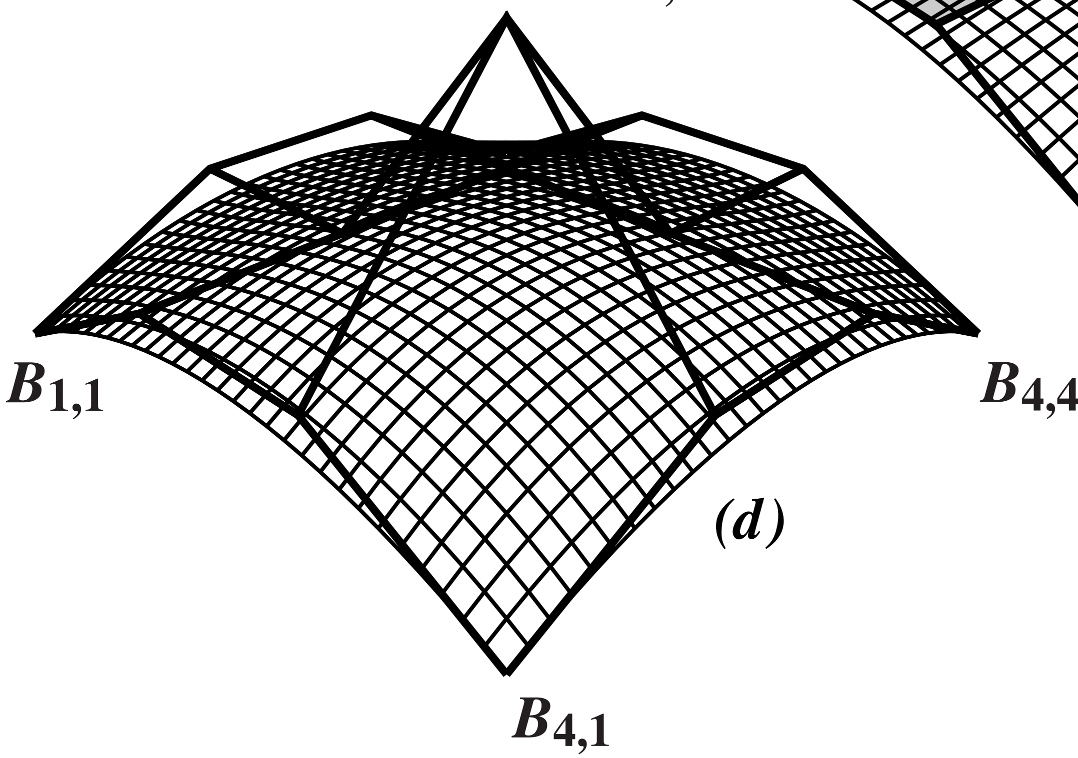



# Bézier Surfaces – Controls

## Effect of twist vector magnitude

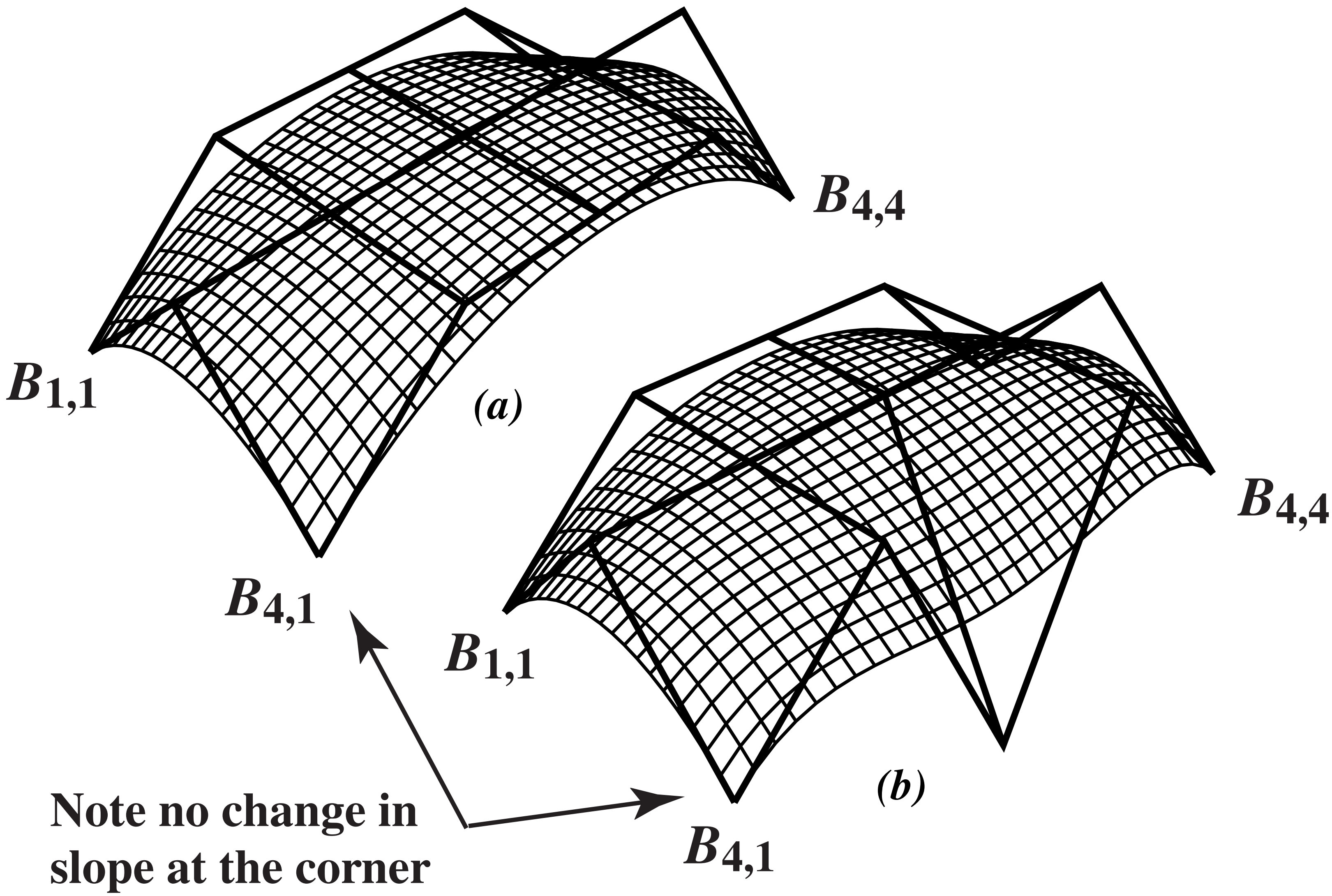



*(a)*




*(d)*

# Bézier Surfaces – Local Control



# Nonrational B-spline Surfaces – Definition

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,\ell}(w)$$

**where**

$$N_{i,1}(u) = \begin{cases} 1 & \textbf{if } x_i \le u < x_{i+1} \\ 0 & \textbf{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{(u - x_i) N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

**and**

$$M_{j,1}(w) = \begin{cases} 1 & \textbf{if } y_j \le w < y_{j+1} \\ 0 & \textbf{otherwise} \end{cases}$$

$$M_{j,\ell}(w) = \frac{(w - y_j) M_{j,\ell-1}(w)}{y_{j+l-1} - y_j} + \frac{(y_{j+\ell} - w) M_{j+1,\ell-1}(w)}{y_{j+\ell} - y_{j+1}}$$

# B-spline surfaces – Characteristics

**Maximum order, $k, \ell$ is the number of control vertices in each parametric direction**

**Continuity $C^{k-2}$, $C^{\ell-2}$ in each parametric direction**

**Variation diminishing property is not known**

**Transform surface – transform control net**

**Influence of single control vertex is $\pm k/2$, $\pm \ell/2$**

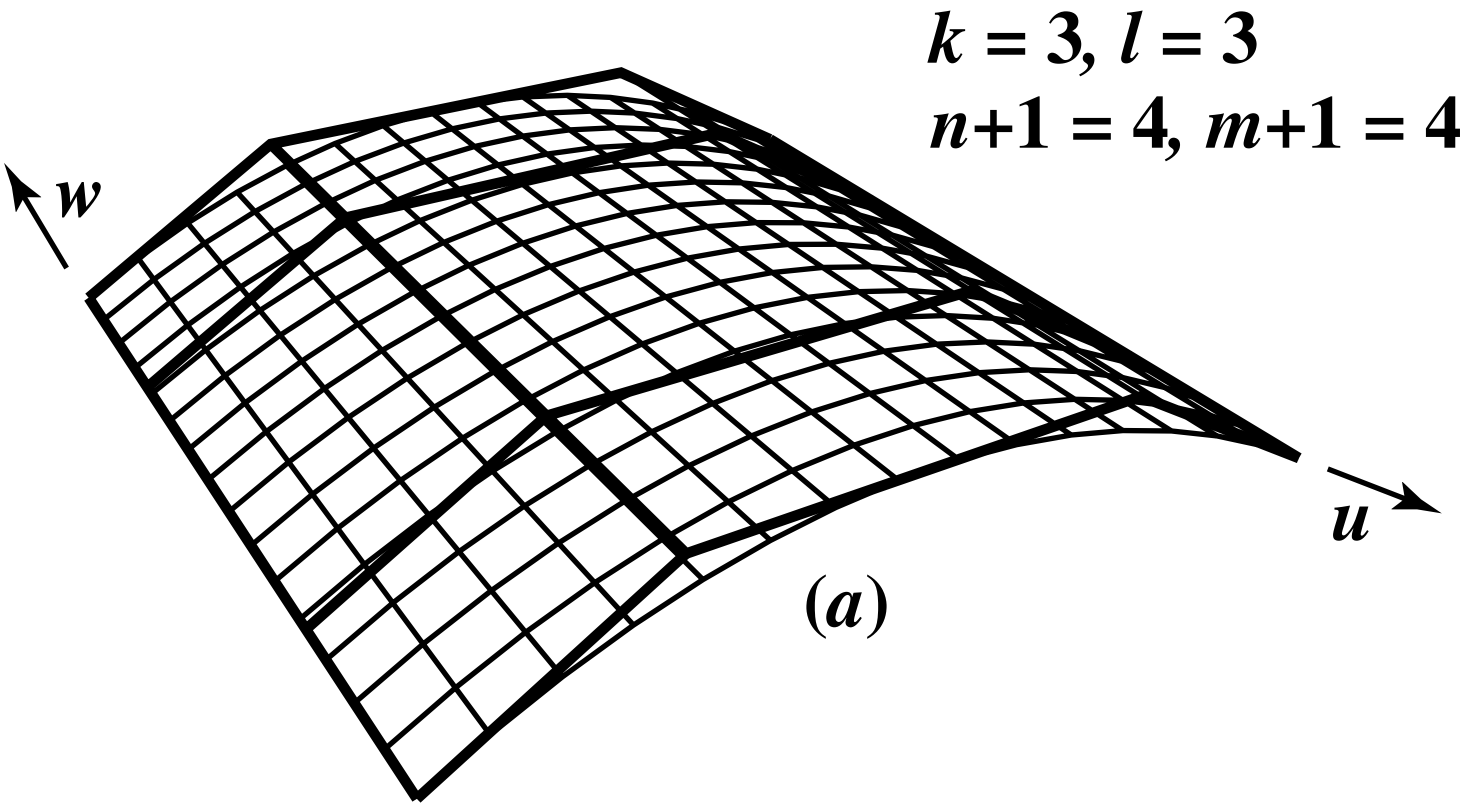
**If $n+1=k$, $m+1=\ell$ a Bézier surface results**

**Triangulated, the control net forms a planar approximation to the surface**

**Lies within the union of convex hulls of $k$, $\ell$ neighboring control vertices**

# Nonrational B-spline Surfaces

## Colinear net lines



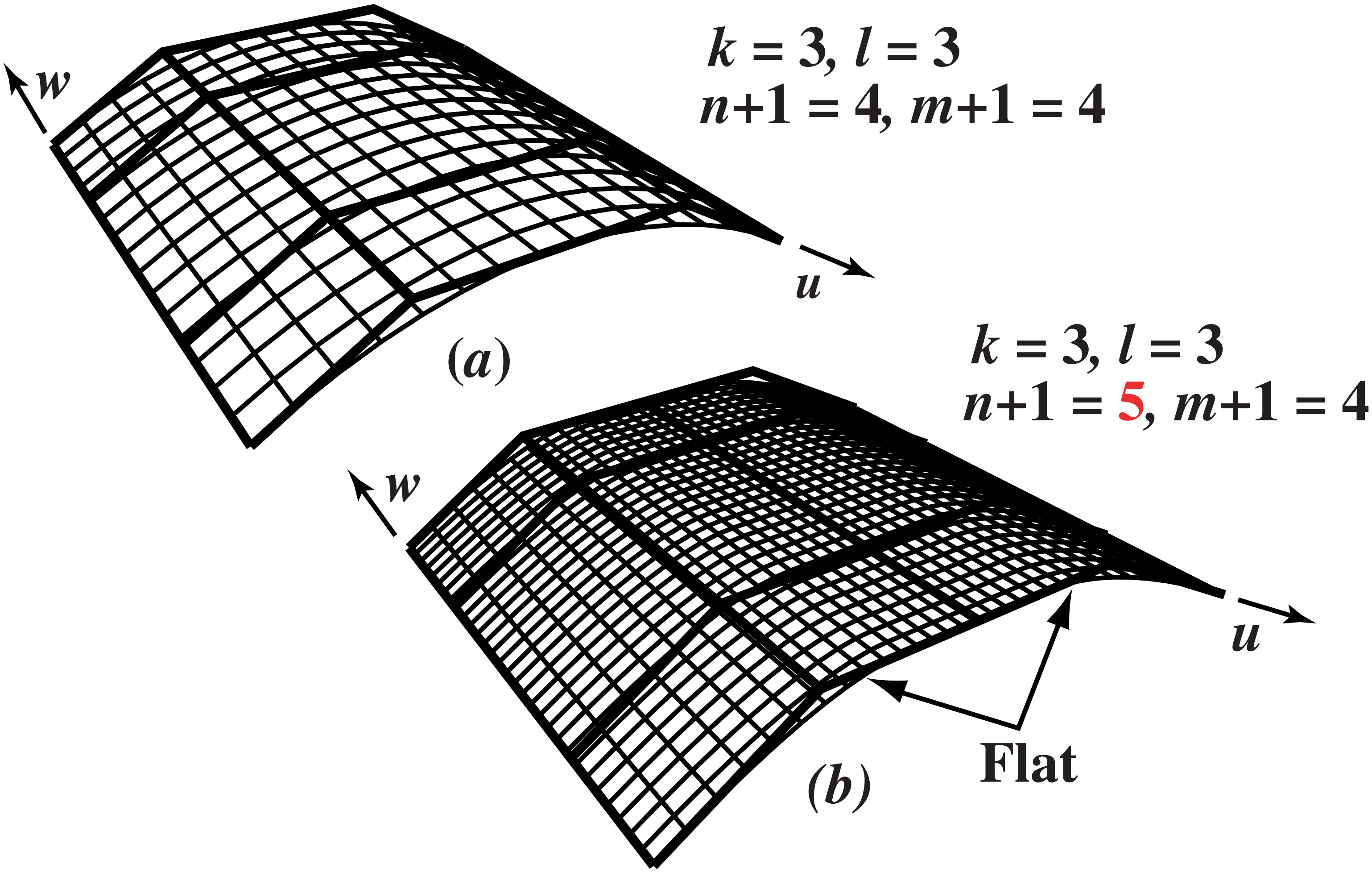

(a)

**Ruled in the $w$ direction**

**Smoothly curved in the $u$ direction**

# Nonrational B-spline Surfaces

## Colinear net lines

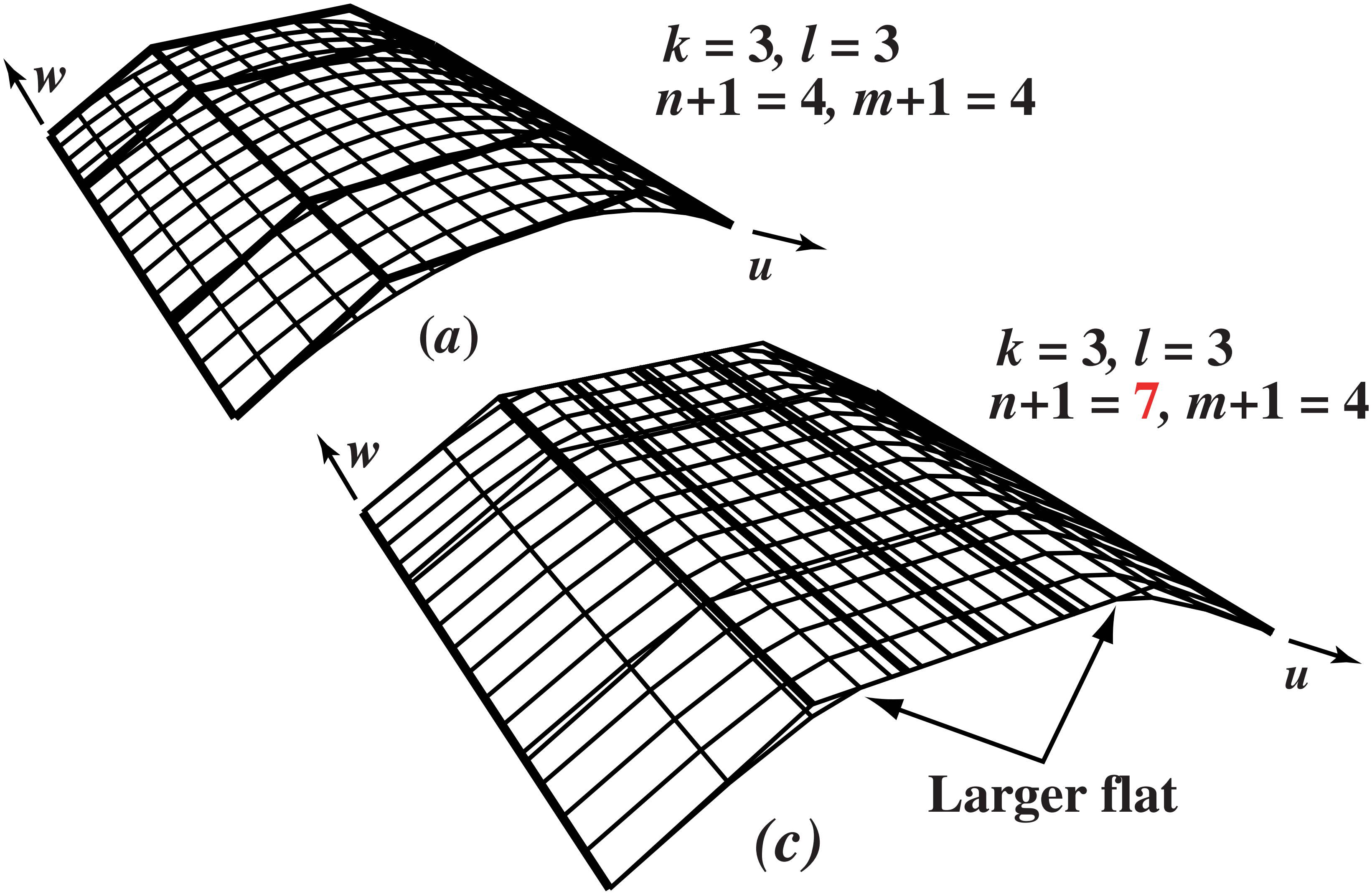



Ruled in the $w$ direction

Embedded flat area in the $u$ direction

# Nonrational B-spline Surfaces

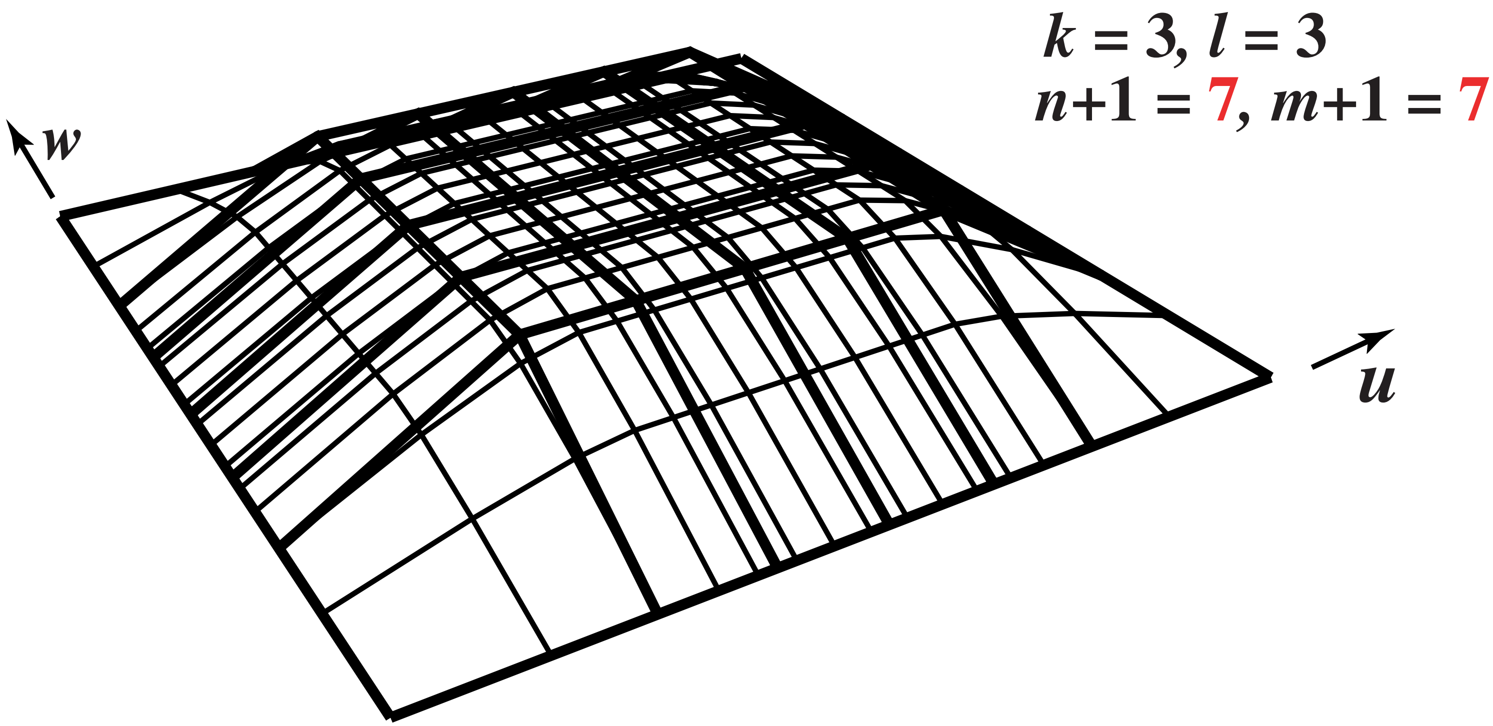
## Colinear net lines




## Larger embedded flat area in the $u$ direction

# Nonrational B-spline Surfaces

## Colinear net lines



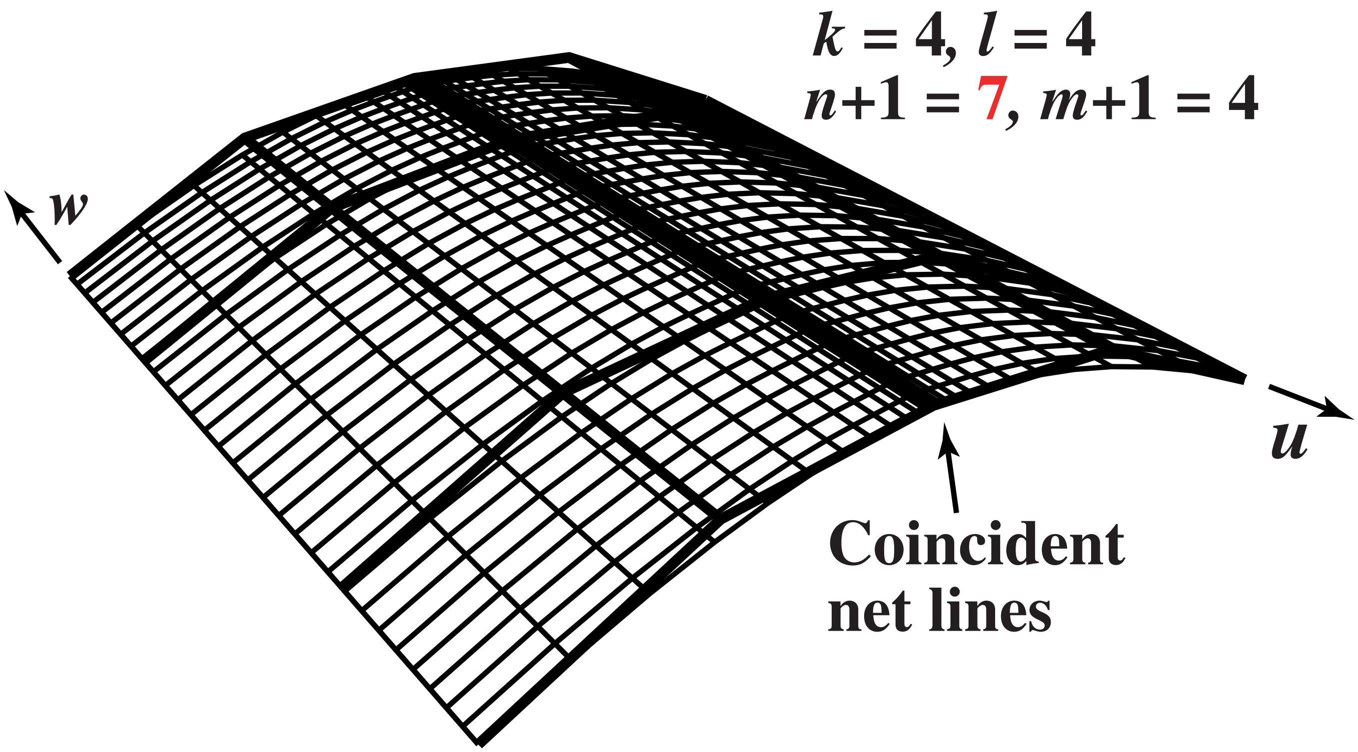

Embedded flat area in the center

Embedded flat area on each side

Curved corners

# Nonrational B-spline Surfaces

## Colinear net lines

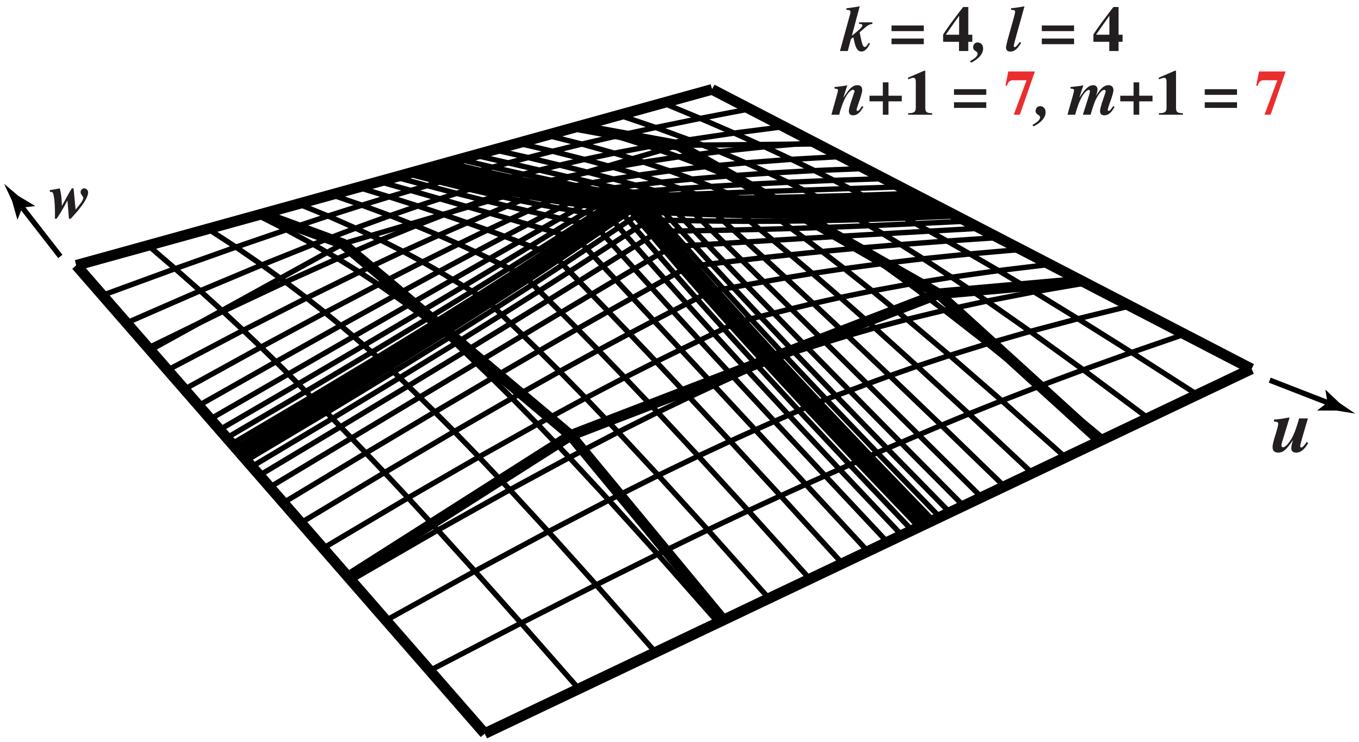



Three coincident net lines in the $w$ direction generate hard line in surface

Still $C^{k-2}$, $C^{\ell-2}$ continuous in both parametric directions

# Nonrational B-spline Surfaces

## Coincident net lines

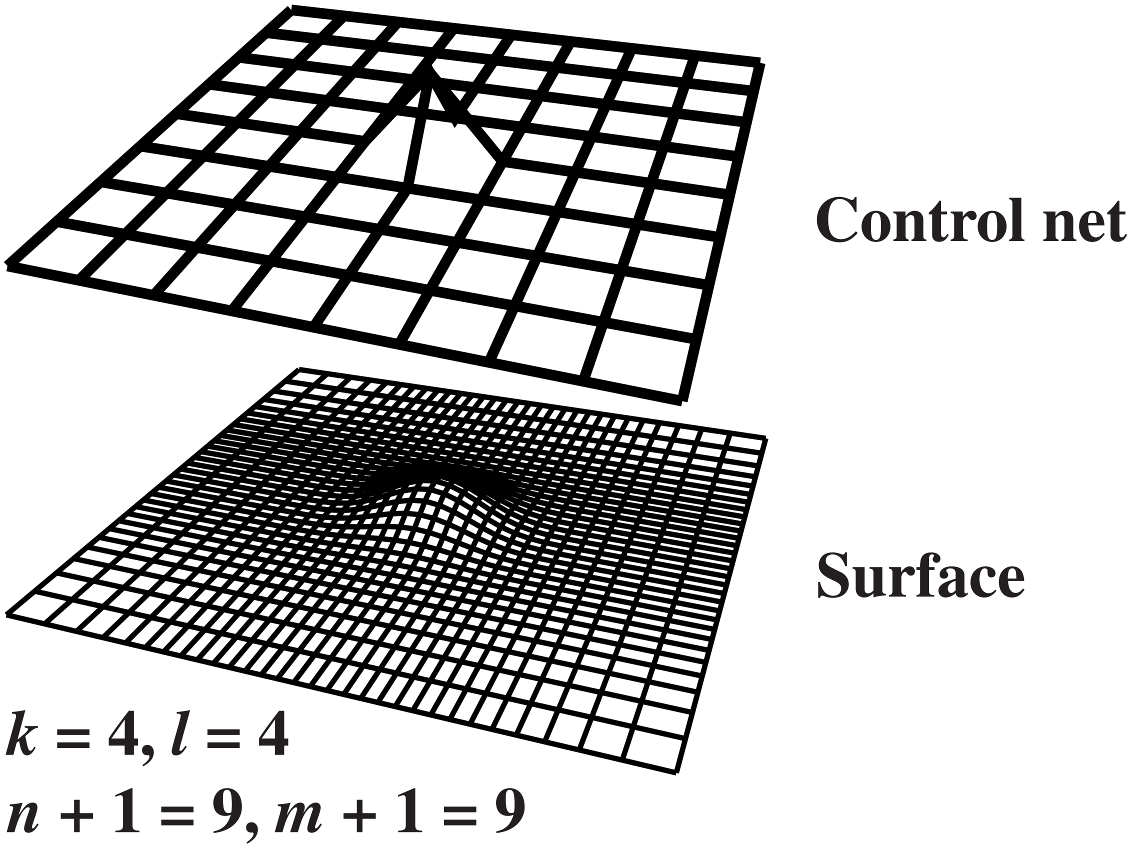



**Three coincident net lines in $u$ and $w$ directions generate hard two hard lines and a point in the surface**

**Still $C^{k-2}$, $C^{\ell-2}$ continuous in both parametric directions**

# Nonrational B-spline Surfaces

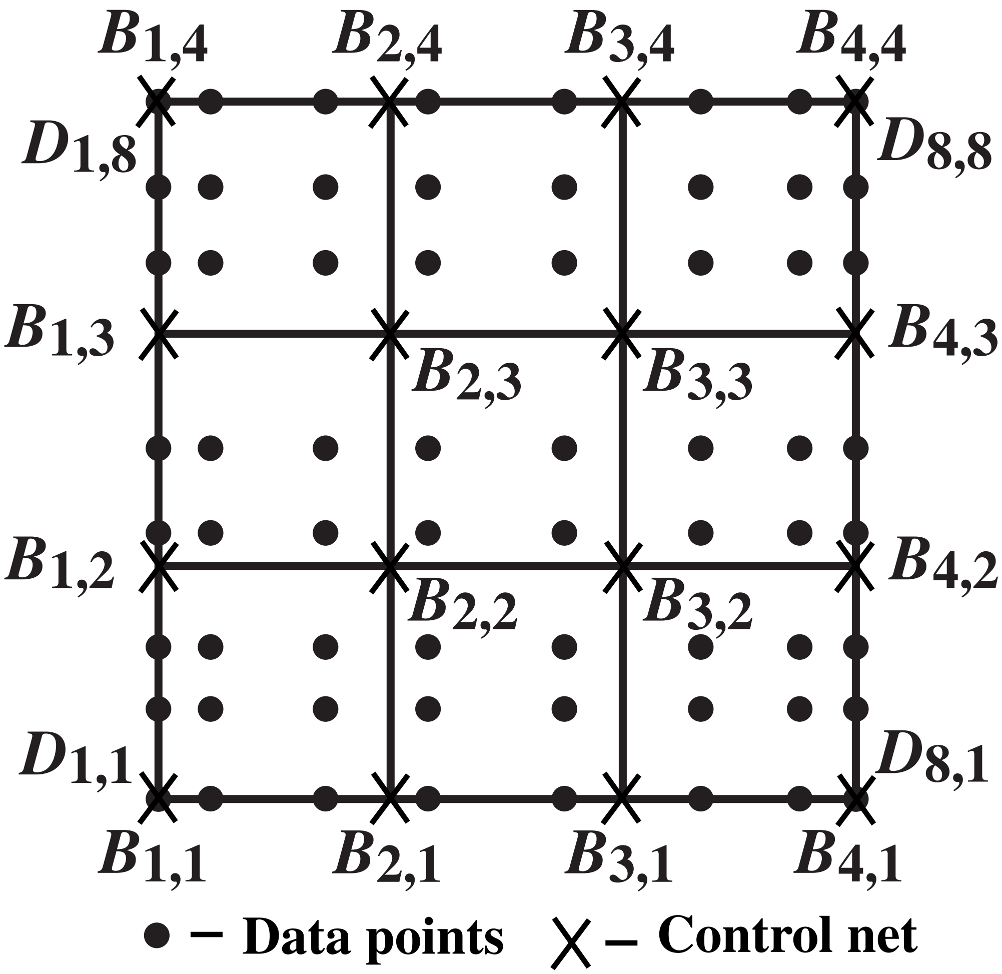
## Local control




**Local influence is** $\pm k/2$, $\pm \ell/2$

# Nonrational B-spline Surfaces

## Surface fitting

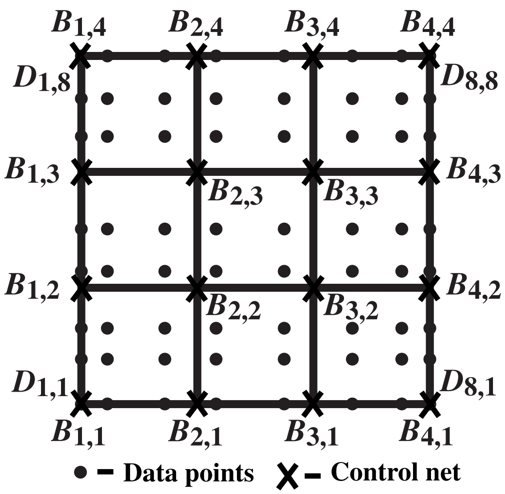



**Topologically rectangular $r \times s$ data set**

**Control net is $2 \leq k \leq n+1 \leq r$ and $2 \leq \ell \leq m+1 \leq s$**

# Nonrational B-spline Surfaces

## Surface fitting




$$D_{1,1}(u_1, w_1) =$$

$$N_{1,k}(u_1)\left[M_{1,\ell}(w_1)B_{1,1} + \cdots + M_{m+1,\ell}(w_1)B_{1,m+1}\right] +$$

$$\vdots$$

$$N_{n+1,k}(u_1)\left[M_{1,\ell}(w_1)B_{n+1,1} + \cdots + M_{m+1,\ell}(w_1)B_{n+1,m+1}\right]$$

$u_1, w_1$ **are the parameter values for each data point**

# Nonrational B-spline Surfaces
## Surface fitting

## Rewrite in matrix form

$$[\,D\,] = [\,C\,]\,[\,B\,]$$

**where**

$[\,D\,]$ **is an** $r * s \times 3$ **matrix of the data points**

$[\,C\,]$ **is an** $r * s \times n * m$ **matrix of the products of the B-spline basis functions, i.e.,** $C_{i,j} = N_{i,k} M_{j,\ell}$

$[\,B\,]$ **is an** $n * m \times 3$ **matrix of the three-dimensional control net vertices**

# B-spline surfaces – Fitting

**Parameter value for each $C_{ij}(u_i, w_j)$**

**A useful approximation is the chord distance**
**For $r$ data points in the $u$ parametric direction**

$$u_1 = 0 \qquad \frac{u_q}{u_{\max}} = \frac{\sum_{g=2}^{q} |D_{g,p} - D_{g-1,p}|}{\sum_{g=2}^{r} |D_{g,p} - D_{g-1,p}|} \qquad \begin{array}{l} 1 \le p \le s \\ 1 \le q \le r \end{array}$$

**Similarly for $s$ data points in the $w$ direction**

**The maximum parameter value is usually the maximum value of the knot vector in each direction**

# B-spline Surfaces – Fitting

**Number control vertices equals number of data points**

$$[\,D\,] = [\,C\,]\,[\,B\,]$$

$[C]$ **is square**

**Control net is obtained by matrix inversion**

$$[\,B\,] = [\,C\,]^{-1}\,[\,D\,]$$

**The resulting B-spline surface passes through each data point**

**But, it may wiggle.**

# B-spline Surfaces – Fitting

**Number control vertices does not equal the number of data points**

$$[\,D\,] = [\,C\,]\,[\,B\,]$$

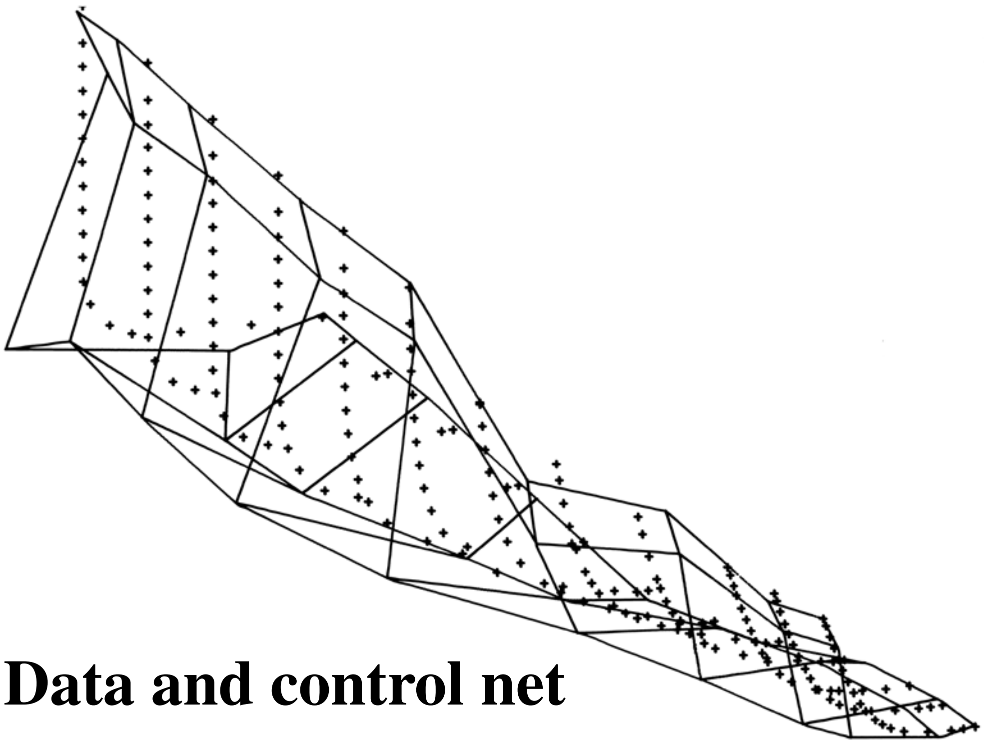
$[C]$ **is not square**

**Control net is obtained in a mean sense by**

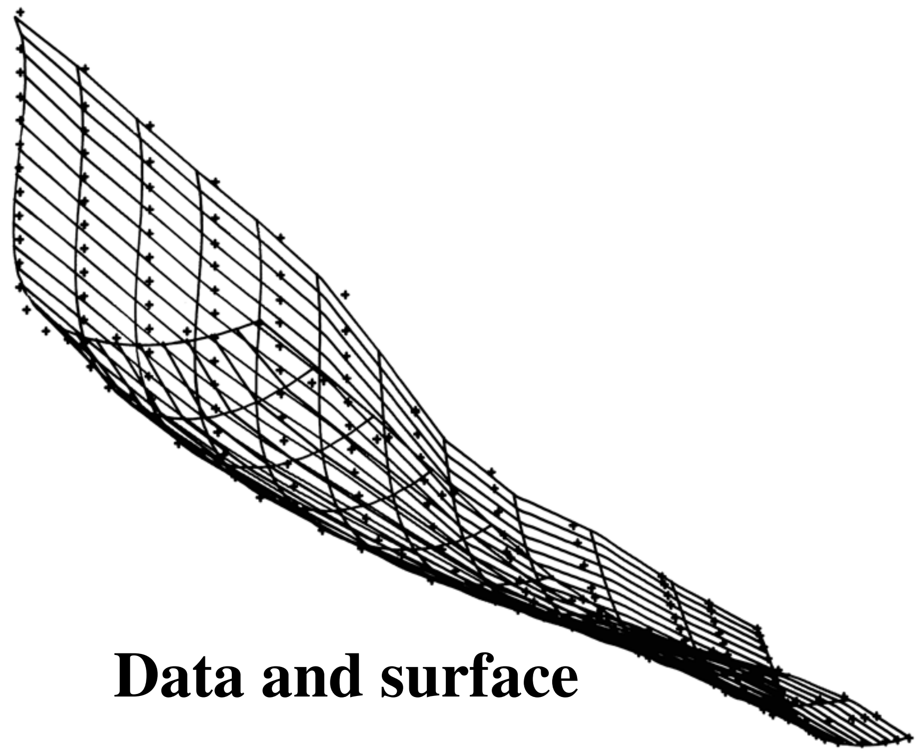
$$[\,B\,] = \left[\,[\,C\,]^T\,[\,C\,]\,\right]^{-1}[\,C\,]^T\,[\,D\,]$$

**The resulting B-spline surface does not pass through each data point**

# B-spline Surfaces
## Fitting – Example



**Data and control net**



**Data and surface**

# B-spline Surfaces - Additional Topics

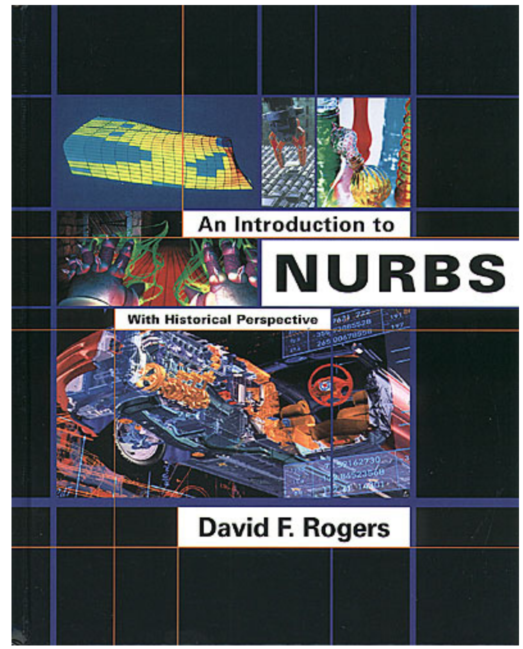
Degree elevation and reduction

Derivatives

Knot insertion

Subdivision

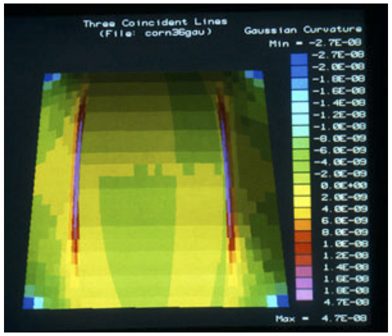
Reparameterization



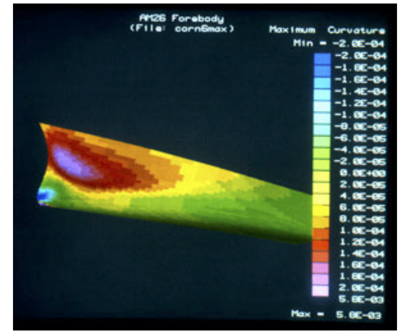

Additional reading:

Rogers, D. F.
An Introduction to NURBS,
With Historical Perspective
Morgan Kaufmann Publishers, 2001

Piegl, L. & Tiller, W.
The NURBS Book, Springer-Verlag, 1995




# Course 31
# NURBS
# (NonUniform Rational B-splines)




## Part 4 (11:35)
## Rational B-spline Surfaces

**David F. Rogers**
**Professor of Aerospace Engineering**
**United States Naval Academy**
**Annapolis, MD 21402**

# NURBS Surfaces – Definition

**In four-dimensional homongeneous coordinate space**

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B^h_{i,j} N_{i,k}(u) M_{j,\ell}(w)$$

**And projecting back into three space**

$$Q(u,w) = \frac{\displaystyle\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\displaystyle\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,\ell}(w)} = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} S_{i,j}(u,w)$$

**where**

$B_{i,j}$**s are the 3-D control net vertices**
$S_{i,j}$**s are the bivariate rational B-spline surface basis functions**

# NURBS Surfaces – Definition

**Basis functions**

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} S_{i,j}(u,w)$$

**where**

$$S_{i,j}(u,w) = \frac{h_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\displaystyle\sum_{i1=1}^{n+1} \sum_{j1=1}^{m+1} h_{i1,j1} N_{i1,k}(u) M_{j1,\ell}(w)} = \frac{h_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\mathbf{Sum}(u,w)}$$

**and**

$$\mathbf{Sum}(u,w) = \sum_{i1=1}^{n+1} \sum_{j1=1}^{m+1} h_{i1,j1} N_{i1,k}(u) M_{j1,\ell}(w)$$

**Convenient, but not necessary, to assume** $h_{i,j} \geq 0$ **for all** $i, j$

# NURBS Surfaces – Definition

## Basis functions

$$S_{i,j}(u,w) = \frac{h_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\mathbf{Sum}(u,w)}$$

$$\mathbf{Sum}(u,w) = \sum_{i1=1}^{n+1} \sum_{j1=1}^{m+1} h_{i1,j1} N_{i1,k}(u) M_{j1,\ell}(w)$$

**$S_{i,j}(u,w)$s are not the product of $R_{i,k}(u)$ and $R_{j,\ell}(w)$**

**Similar shapes and characteristics to $N_{i,k}(u)M_{j,\ell}(w)$**

# NURBS surfaces – Characteristics

$$\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} S_{i,j}(u, w) \equiv 1$$

$$S_{i,j}(u, w) \geq 0$$

**Maximum order is the number of control vertices in each parametric direction**

**Continuity $C^{k-2}$, $C^{\ell-2}$ in each parametric direction**

**Transform surface – transform control net**

**The variation-diminishing property not known**

# NURBS surfaces – Characteristics

**Influence of single control vertex is $\pm k/2$, $\pm \ell/2$**

**If $n+1 = k$, $m+1 = \ell$, a rational Bézier surface results**

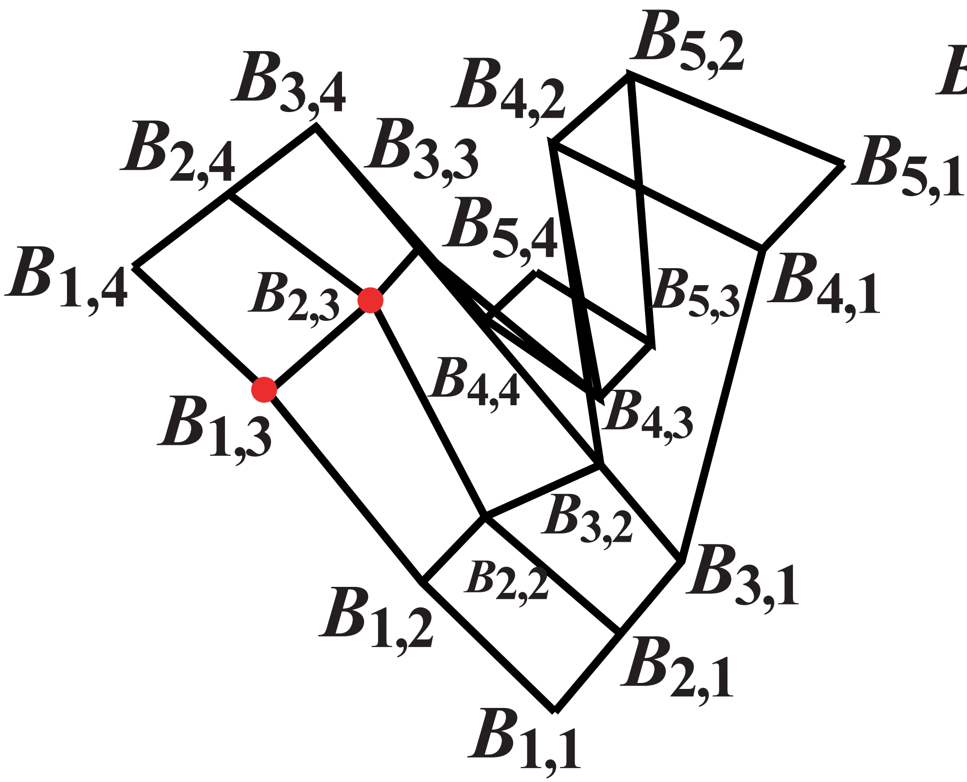
**If $n+1 = k$, $m+1 = \ell$ and $h_{ij} = 1$, a nonrational Bézier surface results**

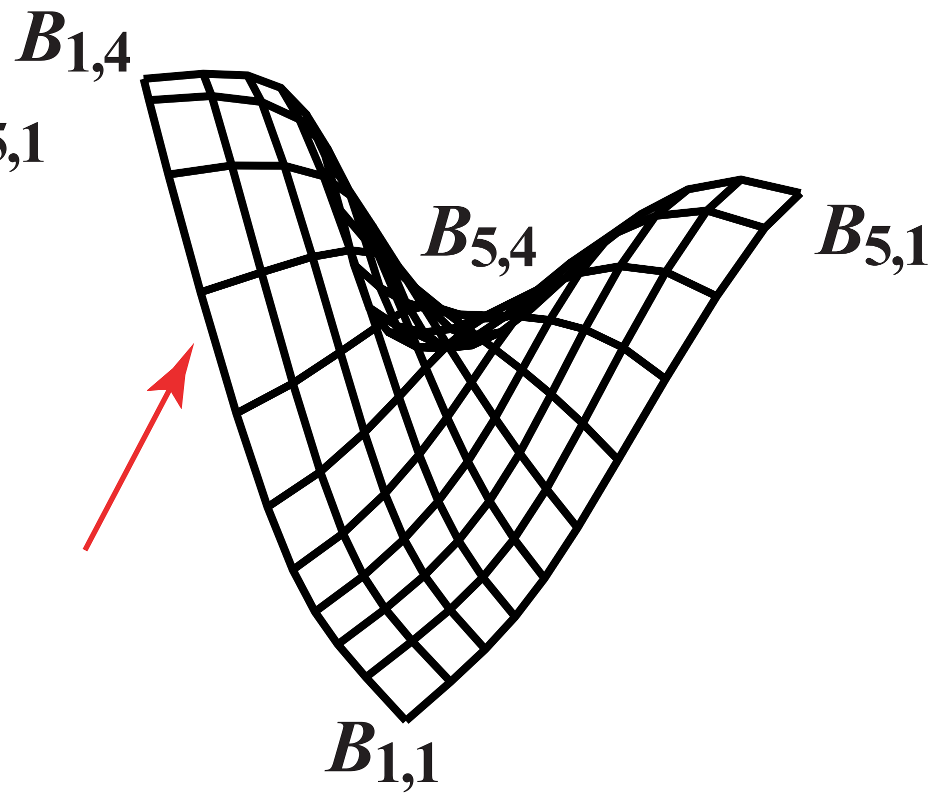
**Triangulated, the control net forms a planar approximation to the surface**

**If $h_{i,j} \geq 0$, surface lies within union of convex hulls of $k$, $\ell$ neighboring control vertices**

# NURBS surfaces – Characteristics

$h_{i,j} \geq 0$, effect of zero weights



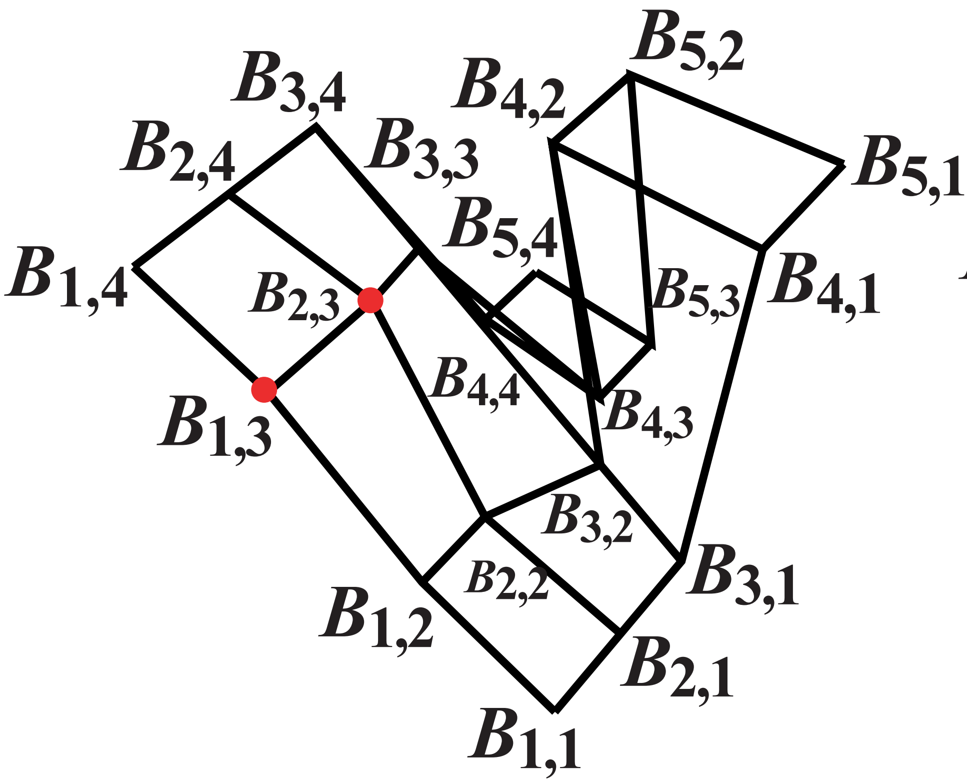



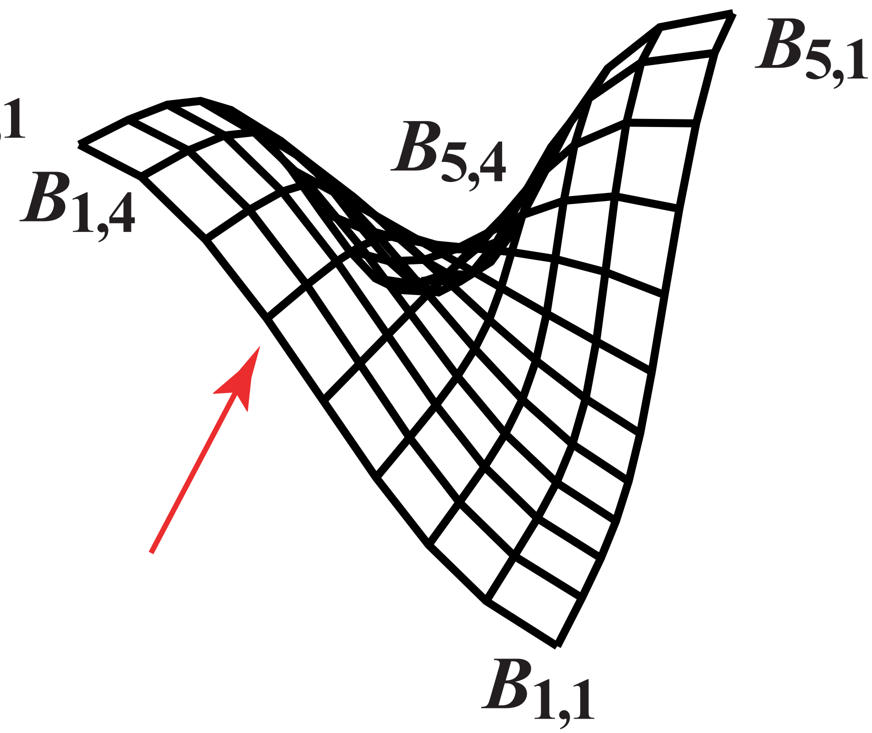

$n+1=5,\ m+1=4,\ k=l=4,\ h_{1,3}=h_{2,3}=0$

**Notice the straight edge and flat surface indicated by the red arrow**

# NURBS surfaces – Characteristics

## $h_{i,j} \geq 0$ effect of homogeneous weights



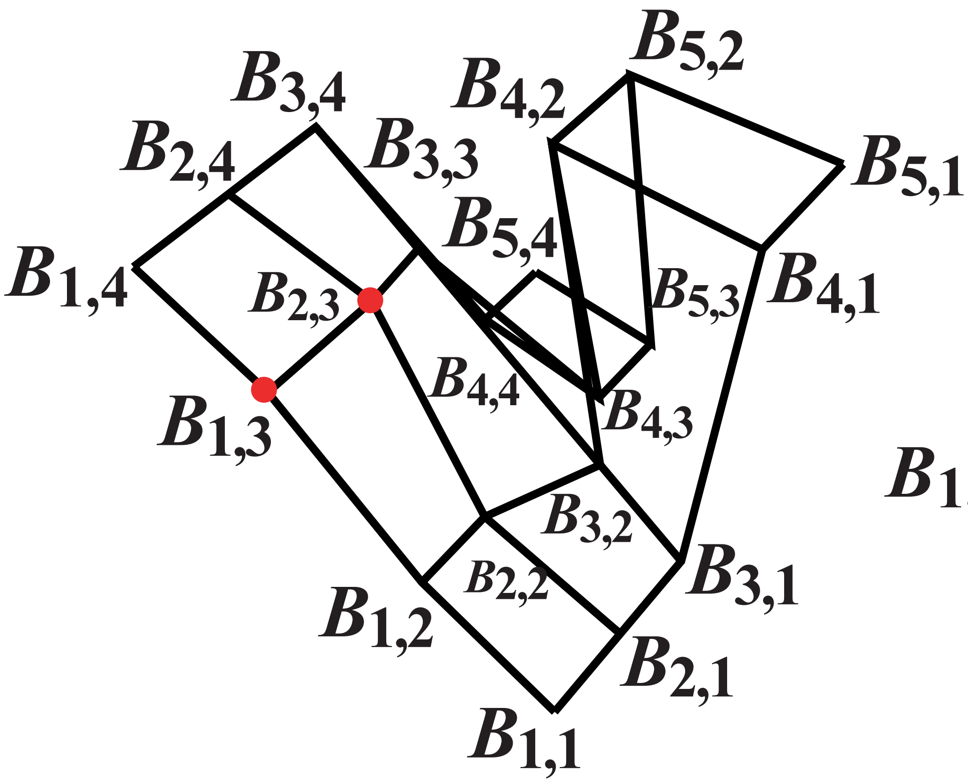



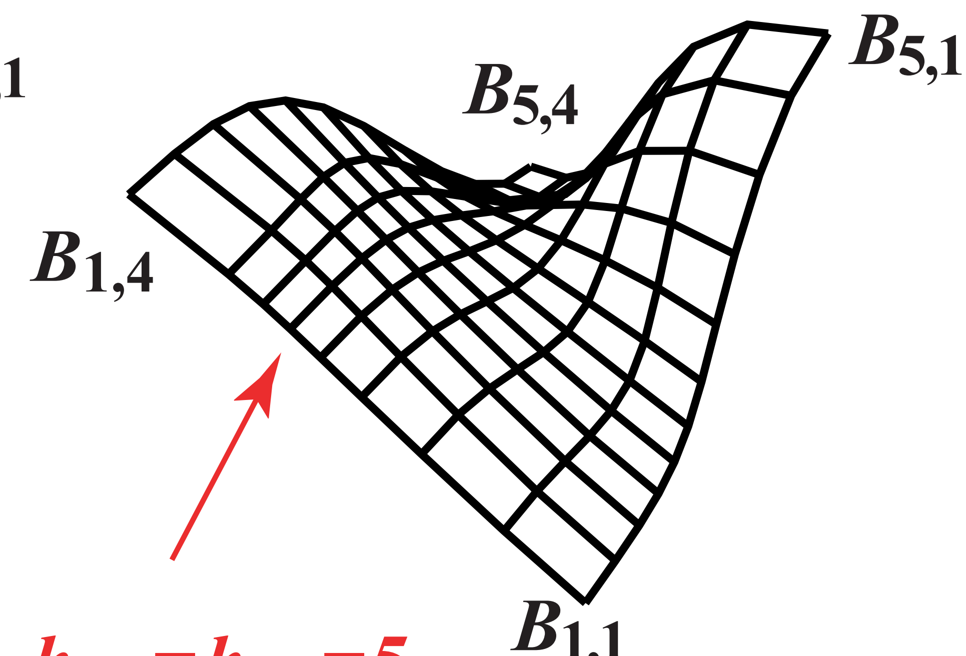

$n+1=5,\ m+1=4,\ k=l=4,\ h_{1,3}=h_{2,3}=1$

**Notice the curved edge and surface indicated by the red arrow**

# NURBS surfaces – Characteristics

$h_{i,j} \geq 0$ effect of homogeneous weights



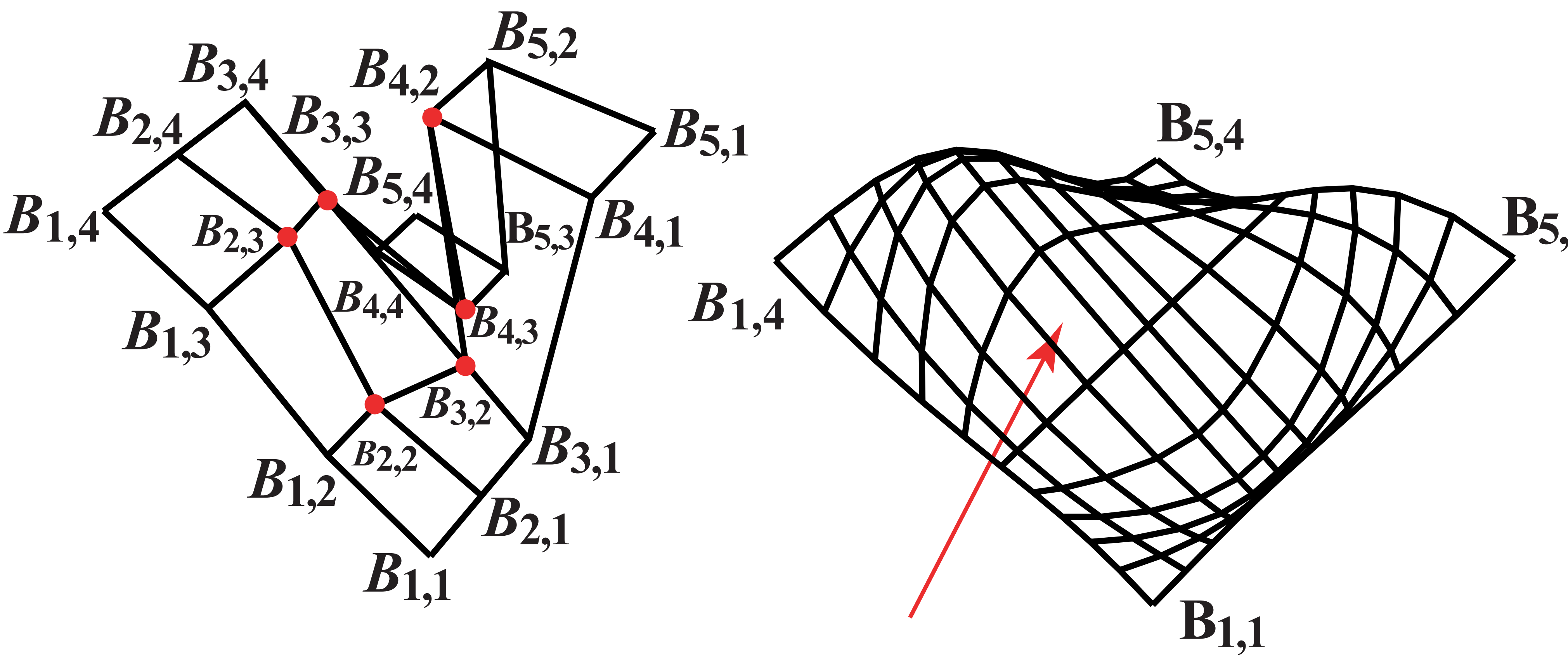




$n+1=5$, $m+1=4$, $k=l=4$, $h_{1,3}=h_{2,3}=5$

**Notice the flat edge and surface indicated by the red arrow**

# NURBS surfaces – Characteristics

## $h_{i,j} \geq 0$ effect of homogeneous weights

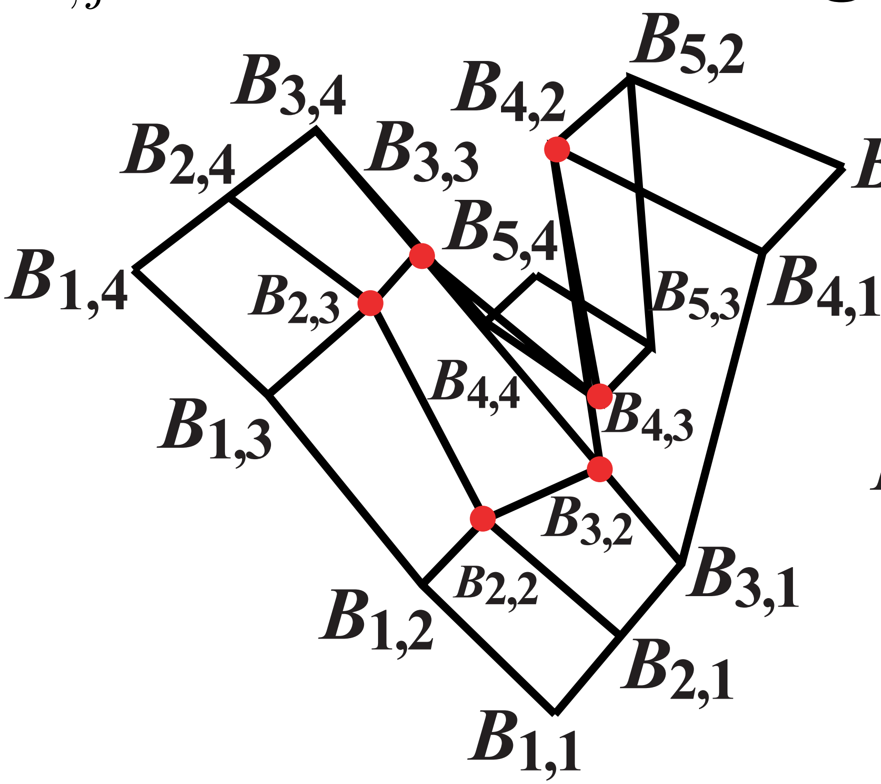


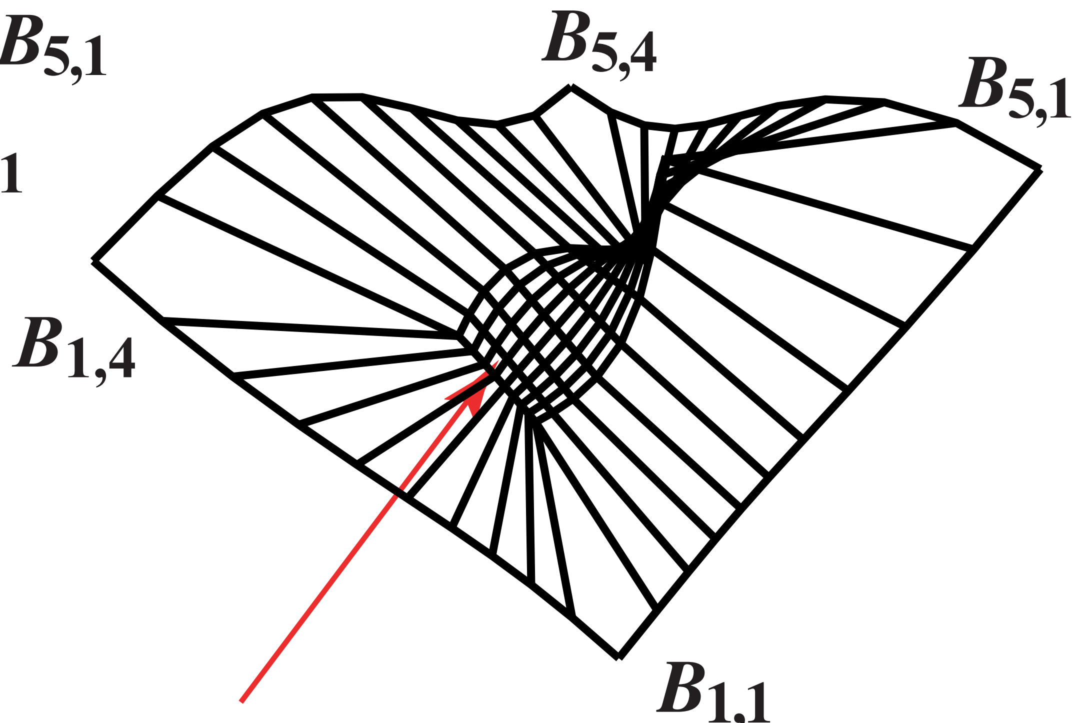

$n+1=5,\ m+1=4,\ k=l=4$ **All interior $h_{i,j} = 0$**

**Notice the edge and the surface interior indicated by the red arrow**

# NURBS surfaces – Characteristics

$h_{i,j} \geq 0$ effect of homogeneous weights



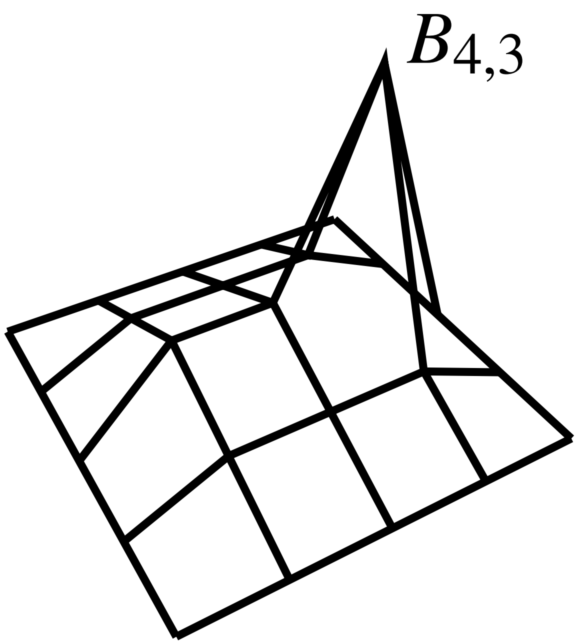

$n+1=5,\ m+1=4,\ k=l=4$ All interior $h_{i,j} = 500$

Notice the surface interior indicated by the red arrow

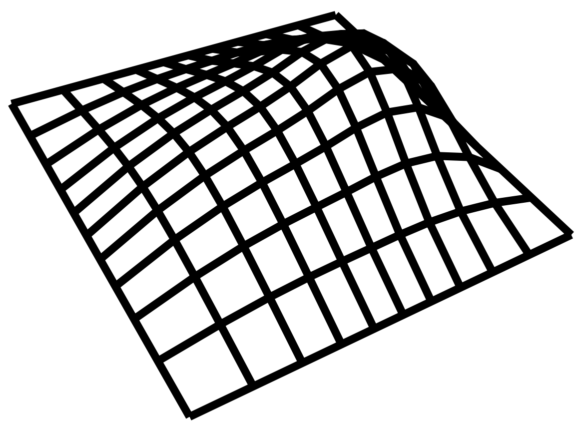
This is a terrible parameterization

# NURBS surfaces – Characteristics

## $h_{i,j} \geq 0$, effect of homogeneous weights



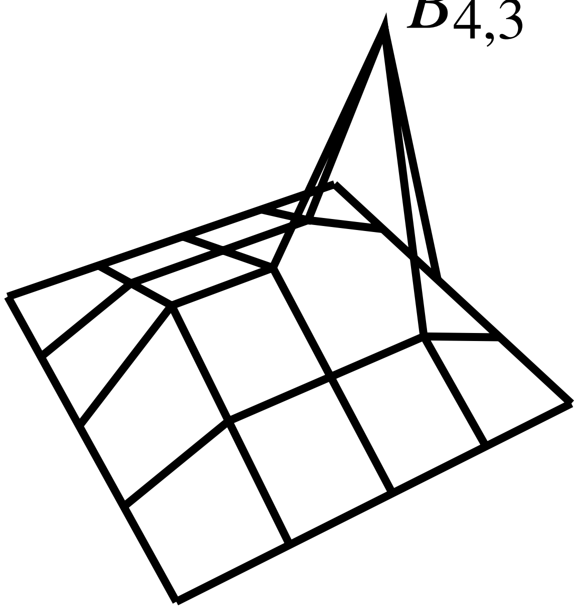

**Control net**



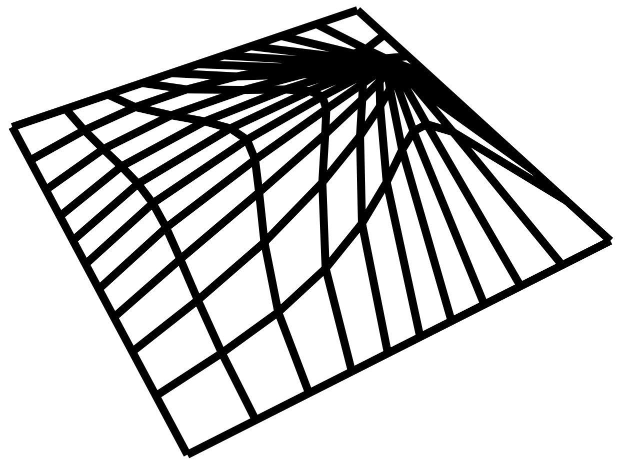
**Surface $h_{4,3} = 1$**

# NURBS surfaces – Characteristics

## $h_{i,j} \geq 0$, effect of homogeneous weights



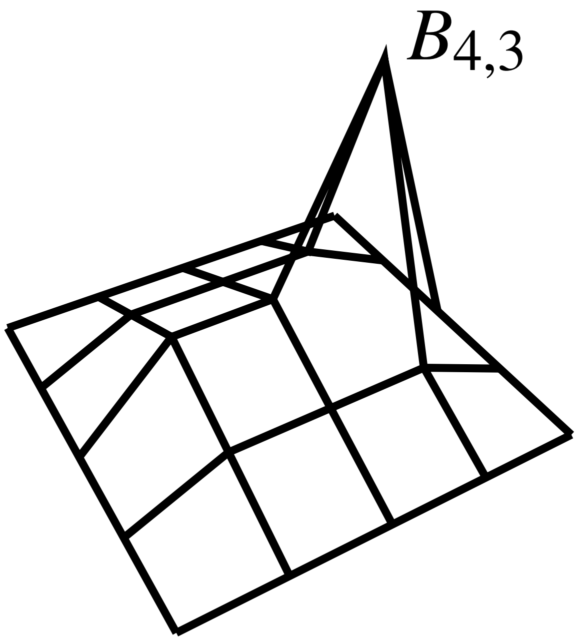

**Control net**



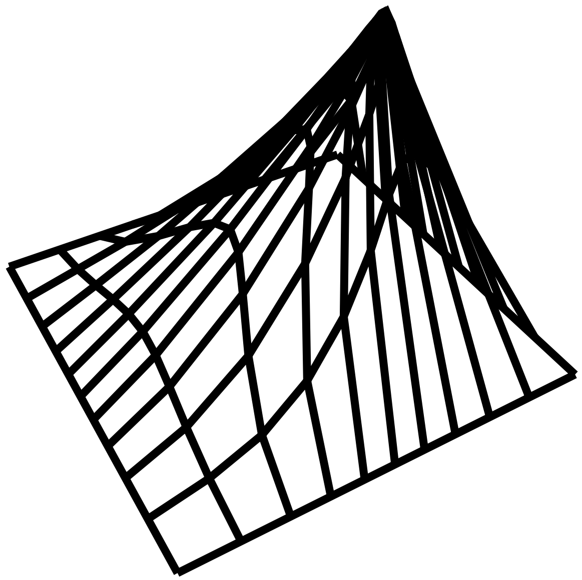
**Surface $h_{4,3} = 5$**

# NURBS surfaces – Characteristics

## $h_{i,j} \geq 0$, effect of homogeneous weights



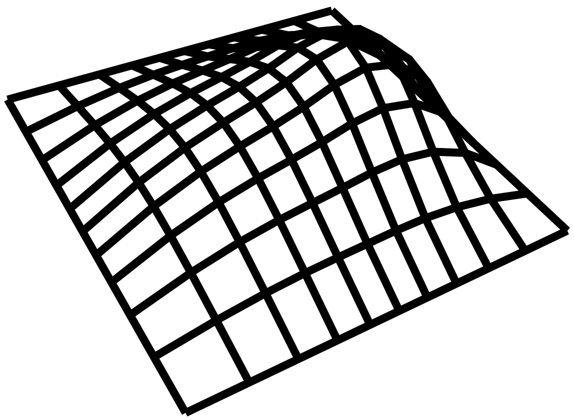

**Control net**



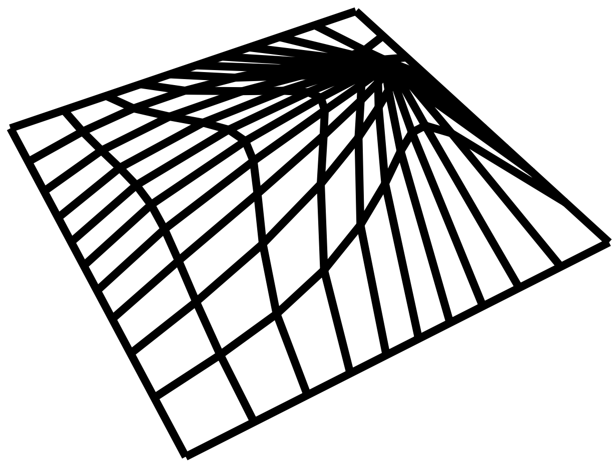
**Surface $h_{4,3} = 50$**

# NURBS surfaces – Characteristics

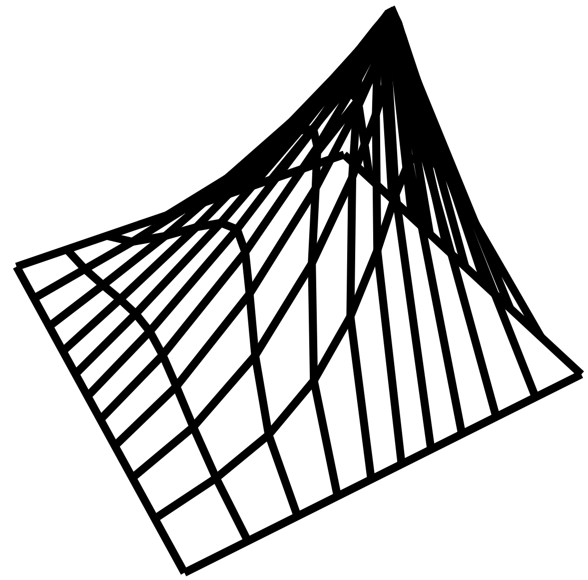
## $h_{i,j} \geq 0$, effect of homogeneous weights - comparison



$h_{4,3} = 1$



$h_{4,3} = 5$



$h_{4,3} = 50$

# NURBS Surfaces – Algorithm

**Nonrational B-spline surface –** $h_{i,j} = 1$ **for all** $i, j$

**Hence**

$$\mathbf{Sum}(u,w) = \sum_{i1=1}^{n+1} \sum_{j1=1}^{m+1} h_{i1,j1} N_{i1,k}(u) M_{j1,\ell}(w) = 1 \ \ \textbf{for all} \ \ u, w$$

**and** $S_{i,j}(u,w)$ **reduces to**

$$S_{i,j}(u,w) = \frac{h_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\mathbf{Sum}(u,w)} = N_{i,k}(u) M_{j,\ell}(w)$$

**which yields**

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} S_{i,j}(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,\ell}(w)$$

**which suggests that the core algorithm is two nested loops**

# NURBS Surfaces – Algorithm

## Nonrational B-spline surface – Example

**Writing out for $n+1=4$, $m+1=4$, $k=\ell=4$ yields**

$$Q(u,w) = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} B_{i,j}N_{i,k}(u)M_{j,\ell}(w) = \sum_{i=1}^{4}\sum_{j=1}^{4} B_{i,j}N_{i,4}(u)M_{j,4}(w)$$

**or**

$$\begin{aligned} Q(u,w) = {} & N_{1,4}(B_{1,1}M_{1,4} + B_{1,2}M_{2,4} + B_{1,3}M_{3,4} + B_{1,4}M_{4,4}) \\ & + N_{2,4}(B_{2,1}M_{1,4} + B_{2,2}M_{2,4} + B_{2,3}M_{3,4} + B_{2,4}M_{4,4}) \\ & + N_{3,4}(B_{3,1}M_{1,4} + B_{3,2}M_{2,4} + B_{3,3}M_{3,4} + B_{3,4}M_{4,4}) \\ & + N_{4,4}(B_{4,1}M_{1,4} + B_{4,2}M_{2,4} + B_{4,3}M_{3,4} + B_{4,4}M_{4,4}) \end{aligned}$$

**The inner loop is within the ( )**
**The outer loop is the multiplier $N_{i,j}(\ )$**
**The knot vectors and basis functions are also needed**

# NURBS Surfaces – Algorithm

## Naive nonrational B-spline surface algorithm

**Specify number of control vertices in the $u$, $w$ directions**
**Specify order in each of the $u$, $w$ directions**
**Specify number of isoparametric lines in each of the $u$, $w$ directio**
**Specify the control net, store in an array**
**Calculate the knot vector in the $u$ direction, store in an array**
**Calculate the knot vector in the $w$ direction, store in an array**
**For each parametric value, $u$**
  **Calculate the basis functions, $N_{i,k}(u)$, store in an array**
  **For each parametric value, $w$**
    **Calculate the basis functions, $M_{j,\ell}(w)$, store in an array**
        **For each control vertex in the $u$ direction**
          **For each control vertex in the $w$ direction**
            **Calculate the surface point, $Q(u, w)$, store in an array**
          **end loop**
        **end loop**
  **end loop**
**end loop**

# NURBS Surfaces – Algorithm

**Rational B-spline (NURBS) surface**

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} \frac{h_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\mathbf{Sum}(u,w)}$$

**and**

$$\mathbf{Sum}(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,\ell}(w)$$

**Two differences from the nonrational B-spline surface:**

- **Calculate and divide by the Sum$(u,w)$ function**
- **Multiply by** $h_{i\,j}$

**Let's look at calculating the Sum$(u,w)$ function**

# NURBS Surfaces – Algorithm

**Calculating the $\mathbf{Sum}(u, w)$ function**

$$\mathbf{Sum}(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,\ell}(w)$$

**Writing this out for $n + 1 = m + 1 = 4$, $k = \ell = 4$ yields**

$$\begin{aligned}
\mathbf{Sum}(u, w) &= \sum_{i=1}^{4} \sum_{j=1}^{4} h_{i,j} N_{i,4}(u) M_{j,4}(w) \\
&= N_{1,4}(h_{1,1}M_{1,4} + h_{1,2}M_{2,4} + h_{1,3}M_{3,4} + h_{1,4}M_{4,4}) \\
&\quad + N_{2,4}(h_{2,1}M_{1,4} + h_{2,2}M_{2,4} + h_{2,3}M_{3,4} + h_{2,4}M_{4,4}) \\
&\quad + N_{3,4}(h_{3,1}M_{1,4} + h_{3,2}M_{2,4} + h_{3,3}M_{3,4} + h_{3,4}M_{4,4}) \\
&\quad + N_{4,4}(h_{4,1}M_{1,4} + h_{4,2}M_{2,4} + h_{4,3}M_{3,4} + h_{4,4}M_{4,4})
\end{aligned}$$

**Same form as the nonrational B-spline surface except $h_{ij}$ instead of $B_{ij}$ – use the same algorithm**

# NURBS Surfaces – Algorithm

## Algorithm for the Sum$(u, w)$ function

**Assume the $N_{i,k}$ and $M_{j,\ell}$ basis functions are available**
**Assume the homogeneous weights, $h_{i,j}$, are available**

**For each control vertex in the $u$ direction**
    **For each control vertex in the $w$ direction**
        **Calculate and store the Sum$(u, w)$ function**
    **end loop**
**end loop**

# NURBS Surfaces – Algorithm

**Naive rational B-spline (NURBS) surface algorithm**

**The inner loop now becomes**

```
For each parametric value, u
  Calculate the basis functions, N_{i,k}(u), store in an array
  For each parametric value, w
    Calculate the basis functions, M_{j,ℓ}(w), store in an array
 ⇒  Calculate the Sum(u,w) function
        For each control vertex in the u direction
          For each control vertex in the w direction
            Calculate and store the surface point, Q(u,w)
          end loop
        end loop
  end loop
end loop
```

# NURBS Surfaces – Algorithm

**Nonrational B-spline surface**

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,\ell}(w)$$

**Rational B-spline (NURBS) surface**

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} \frac{B_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\mathbf{Sum}(u,w)}$$

**Comparing shows the rational (NURBS) algorithm requires**
- **an additional multiply**
- **a division**
- **calculation of the Sum$(u,w)$ function**

**Results in approximately 1/3 more computational effort**

# NURBS Surfaces – Algorithm

These naive algorithms are very memory efficient

However, they are computationally inefficient

Computational efficiency improved by
- avoiding the division by the Sum$(u, w)$ function
  - by converting it to a multiply using the reciprocal
- avoiding entire computations

# NURBS Surfaces – Algorithm

**More efficient NURBS algorithm**

**Recall for $n+1 = m+1 = 3$, $k = \ell = 3$ the NURBS surface is**

$$
\begin{aligned}
Q(u,w) = {} & \frac{N_{1,3}}{\text{Sum}}(h_{1,1}B_{1,1}M_{1,3} + h_{1,2}B_{1,2}M_{2,3} + h_{1,3}B_{1,3}M_{3,3}) \\
& + \frac{N_{2,3}}{\text{Sum}}(h_{2,1}B_{2,1}M_{1,3} + h_{2,2}B_{2,2}M_{2,3} + h_{2,3}B_{2,3}M_{3,3}) \\
& + \frac{N_{3,3}}{\text{Sum}}(h_{3,1}B_{3,1}M_{1,3} + h_{3,2}B_{3,2}M_{2,3} + h_{3,3}B_{3,3}M_{3,3})
\end{aligned}
$$

**Recall that in many cases the basis functions are zero**

**If $N_{i,j}(u,w) = 0$, then we can avoid the entire calculation in ( ) and the division (multiply) by Sum$(u,w)$ (the reciprocal)**

**If $M_{i,j}(u,w) = 0$, then we can avoid two multiplies in ( )**

**Storing the reciprocal of Sum$(u,w)$ saves a divide at the expense of a multiply**

# NURBS Surfaces – Algorithm

**More efficient rational B-spline (NURBS) surface algorithm**

**The inner loop now becomes**

**For each parametric value,** $u$
  **Calculate the basis functions,** $N_{i,k}(u)$**, store in an array**
  **For each parametric value,** $w$
    **Calculate the basis functions,** $M_{j,\ell}(w)$**, store in an array**
$\Rightarrow$ **Calculate and save the reciprocal of Sum(**$u, w$**)**
      **For** $i = 1$ **to** $n + 1$ *For each control vertex in the u direction*
  $\Rightarrow$ **If** $N_{i,k}(u) \neq 0$ **then**
        **For** $j = 1$ **to** $m + 1$ *For each control vertex in the w direction*
    $\Rightarrow$ **If** $M_{j,\ell}(w) \neq 0$ **then**
          **Calculate** $Q(u, w) = Q(u, w) + h_{i,j} N_{i,k}(u) M_{j,\ell}(w) \mathbf{Sum}(u, w)$
        **end if**
        **end loop**
      **end if**
      **end loop**
      **Store** $Q(u, w)$**; Reinitalize** $Q(u, w) = 0$
  **end loop**
**end loop**

# NURBS Surfaces – Algorithm

The improved naive algorithms are still very memory efficient

The simple changes, based on the underlying mathematics, increase the computational efficiency by 25% or more

In the late 1970s this algorithm provided the basis for a real time interactive nonrational B-spline surface design system based on directly manipulating the control net – SIGGRAPH '80 paper

The machine was a 16 bit minicomputer with 64 Kbytes of memory driving an Evans & Sutherland Picture System I

Can we do better – Yes!

# NURBS Surfaces – Algorithm

When modifying a B-spline surface, a designer typically works with a control net:

- of constant control net size, $n+1,\ m+1$, in each direction
- of constant order, $k,\ \ell$, in each parametric direction
- with a constant number, $p_1,\ p_2$, of isoparametric lines in each parametric direction

Hence, $n+1$, $m+1$, $k$, $\ell$, $p_1$ and $p_2$ do not change

If these values do not change, neither do the basis functions, $N_{i,k}(u)$ and $M_{j,\ell}(w)$, nor the Sum$(u,w)$ function

Thus, precalculating and storing the product $N_{i,k}(u)M_{j,\ell}(w)/\mathbf{Sum}(u,w)$ further increases the efficiency

However, we leave this specific efficiency increase as an exercise

# NURBS Surfaces – Algorithm

**When modifying a NURBS surface control net, a designer typically manipulates:**

**a single control net vertex,** $B_{ij}$

**or**

**the value of a single homongeneous weight,** $h_{ij}$

**Also, assume** $n+1$, $m+1$, $k$, $\ell$, $p_1$ **and** $p_2$ **do not change**

**Writing the NURBS surface equation for both the new and old surfaces and subtracting yields**

$$\mathbf{Sum}_{\text{new}}(u,w)Q_{\text{new}}(u,w) = \mathbf{Sum}_{\text{old}}(u,w)Q_{\text{old}}(u,w)$$
$$+ (h_{i,j_{\text{new}}}B_{i,j_{\text{new}}} - h_{i,j_{\text{old}}}B_{i,j_{\text{old}}})N_{i,k}(u)M_{j,\ell}(w)$$

**which represents an incremental calculation for the new surface**

# NURBS Surfaces – Algorithm

**Only a single control vertex changes**

**If $h_{i,j}$ does not change, then $\mathbf{Sum}(u,w)$ does not change and**

$$\mathbf{Sum}_{\text{new}}(u,w)Q_{\text{new}}(u,w) = \mathbf{Sum}_{\text{old}}(u,w)Q_{\text{old}}(u,w)$$

$$+ (h_{i,j_{\text{new}}}B_{i,j_{\text{new}}} - h_{i,j_{\text{old}}}B_{i,j_{\text{old}}})N_{i,k}(u)M_{j,\ell}(w)$$

**becomes**

$$Q_{\text{new}}(u,w) = Q_{\text{old}}(u,w) + (B_{i,j_{\text{new}}} - B_{i,j_{\text{old}}})\frac{h_{i,j}(u)N_{i,k}(u)M_{j,\ell}(w)}{\mathbf{Sum}(u,w)}$$

**Thus, incremental calculation of the new surface requires four multiplies, one subtract, one add**

**for each $u, w$**

# NURBS Surfaces – Algorithm

**Only a single homogeneous weight changes**

**If** $h_{i,j}$ **changes, then Sum**$(u, w)$ **also changes and**

$$\mathbf{Sum}_{\text{new}}(u,w)Q_{\text{new}}(u,w) = \mathbf{Sum}_{\text{old}}(u,w)Q_{\text{old}}(u,w)$$
$$+ (h_{i,j_{\text{new}}} B_{i,j_{\text{new}}} - h_{i,j_{\text{old}}} B_{i,j_{\text{old}}}) N_{i,k}(u) M_{j,\ell}(w)$$

**becomes**

$$Q_{\text{new}}(u,w) = \frac{\mathbf{Sum}_{\text{old}}(u,w)}{\mathbf{Sum}_{\text{new}}(u,w)}\, Q_{\text{old}}(u,w)$$
$$+ \left(h_{i,j_{\text{new}}} - h_{i,j_{\text{old}}}\right) \frac{B_{i,j} N_{i,k}(u) M_{j,\ell}(w)}{\mathbf{Sum}_{\text{new}}(u,w)}$$

**Thus, incremental calculation of the new surface requires**

**six multiplies, one subtract, one add**

**calculation of the new Sum**$(u, w)$ **function**

**for each** $u, w$

# NURBS Surfaces – Algorithm

**Incremental $\mathbf{Sum}(u, w)$ calculation**

**Writing the $\mathbf{Sum}(u, w)$ expression for both the new and old surfaces and subtracting yields**

$$\mathbf{Sum}_{\text{new}}(u, w) = \mathbf{Sum}_{\text{old}}(u, w) + (h_{i,j_{\text{new}}} - h_{i,j_{\text{old}}}) N_{i,k}(u) M_{j,\ell}(w)$$

**which represents an incremental calculation for the new $\mathbf{Sum}(u, w)$ function**

**Thus, calculating the new $\mathbf{Sum}(u, w)$ requires two multiplies, a subtract and an add**

**If either $N_{i\,k}(u)$ or $M_{j,\ell}(w)$ are zero, the $\mathbf{Sum}(u, w)$ function does not change**

# NURBS Surfaces – Algorithm

## Nonrational B-spline surface incremental calculation

**Recall**

$$\mathbf{Sum}_{\text{new}}(u,w)Q_{\text{new}}(u,w) = \mathbf{Sum}_{\text{old}}(u,w)Q_{\text{old}}(u,w)$$

$$+ (h_{i,j_{\text{new}}} B_{i,j_{\text{new}}} - h_{i,j_{\text{old}}} B_{i,j_{\text{old}}}) N_{i,k}(u) M_{j,\ell}(w)$$

**If $\mathbf{Sum}(u,w) = 1$ and all $h_{i,j} = 1$, a nonrational B-spline surface is generated. The result is**

$$Q_{\text{new}}(u,w) = Q_{\text{old}}(u,w) + (B_{i,j_{\text{new}}} - B_{i,j_{\text{old}}}) N_{i,k}(u) M_{j,\ell}(w)$$

**Thus, calculating the new surface requires two multiplies, a subtract and an add for each $u, w$**

**If either $N_{i\,k}(u)$ or $M_{j,\ell}(w)$ are zero, the surface point at $u, w$ does not change**

# NURBS Surfaces – Algorithm

Implemented in 1981 and published in 1982

The algorithms provide
- dynamic real time interactive manipulation of
  - spatial position control net vertex
  - homogeneous weight
- on modest computer systems

# Fast NURBS Surface Algorithm

**Use** $itest = (n+1) + (m+1)k + \ell + p_1 + p_2)$
**to determine if a complete new surface**
**is required**

**if** $(itest \neq (n+1) + (m+1)k + \ell + p_1 + p_2)$ **then**
  **calculate complete new surface (see previous)**
**else**
  **calculate incremental change to the surface**
**end if**

# Fast NURBS Surface Algorithm

**if** $(itest \neq (n+1) + (m+1)k + \ell + p_1 + p_2)$ **then**
 **calculate incremental change, if any,**
 **in the spatial coordinate or homogeneous**
 **weight of the vertex being manipulated**
  **if (***any coordinate or weight changed***) then**
   **if (***homogeneous weight changed***) then**
    **save the old Sum**$(u, w)$ **function**
    **calculate the new Sum**$(u, w)$ **function**
    **if (***no change in homogeneous weight***) then**
     **control net vertex changed**
     **calculate change in surface for each** $u, w$
    **else**
     **homogeneous weight changed**
     **calculate change in surface for each** $u, w$
    **end if**
   **end if**
   **save current vertex coordinates as old**
   **save current homogeneous weight as old**
  **end if**
**end if**

# Fast NURBS Surface Algorithm

**Efficiency improvement**

**only spatial coordinate changes – factor of 38**

**only homogeneous weight changes – factor of 15**

**over the naive algorithms**

# NURBS Surfaces

## Additional topics

Effect of multiple coincident knot values
Effect of internal nonuniform knot values
Effect of negative weights
Reparameterization
Derivatives – Curvature
Bilinear surfaces
Ruled/Developable surfaces
Sweep surfaces
Surfaces of revolution
Conic volumes
Subdivision
Trim surfaces
Surface fitting
Constrained surface fitting

# NURBS Surfaces

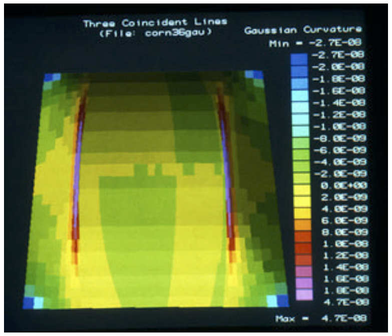
References


Rogers, D.F., B-spline curves and surfaces for ship hull design, Proc. of SNAME, SCAHD 77, First International Symposium on Computer Aided Hull Surface Definition, Annapolis, MD, 26–27 September 1977.

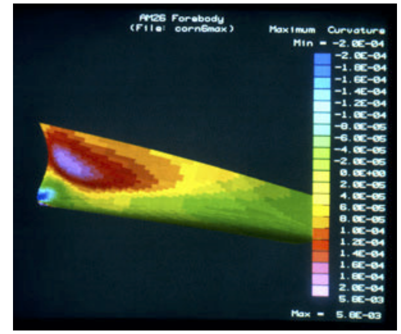
Rogers, D.F., and Satterfield, S.G., B-spline surfaces for ship hull design, Computer Graphics, Vol. 14, pp. 211–217, 1980 (SIGGRAPH 80).

Rogers, D.F., and Satterfield, S.G., Dynamic B-spline surfaces, Proc. of the Fourth International Conference on Computer Applications in the Automation of Shipyard Operation and Applications in the design (ICCAS 82), 7–10 June 1982, Annapolis, MD, pp. 189–196, North Holland, 1982.

Rogers, D. F. and Adlum, L., Dynamic Rational and Nonrational B-spline Surface for Display and Manipulation, invited paper in the commemorative issue honoring Pierre Bezier on his 80th birthday, Computer Aided Design Journal, Vol. 22, pp. 609-616, 1990.




# Course 31
# NURBS
# (NonUniform Rational B-splines)




## Errata at

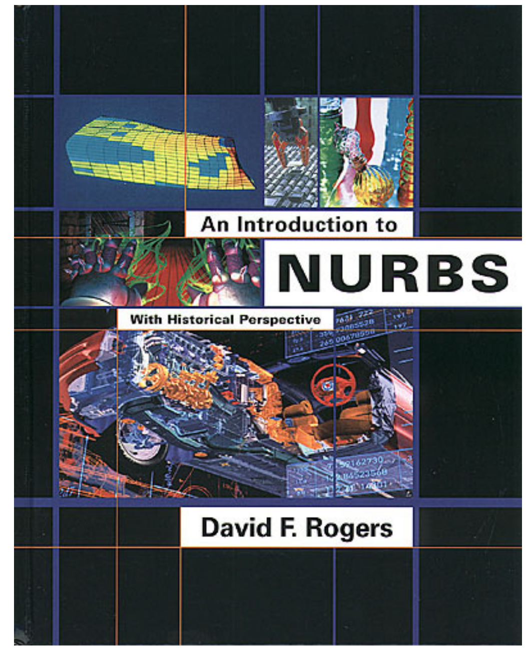
**www.nar-associates.com/nurbs/nurbs.html**

**David F. Rogers**
**Professor of Aerospace Engineering**
**United States Naval Academy**
**Annapolis, MD 21402**

# NURBS Surfaces

## Recommended Books

Rogers, D.F., An Introduction to NURBS
With Historical Perspective
Morgan Kaufmann Publishers, 2001




Piegl, L. and Tiller, W.,
The NURBS Book
Springer-Verlag, 1995

## More advanced

Cohen, E, Riesenfeld, R, and Elber, G.
Geometric Modeling with Splines
A.K. Peters, 2001