Computational Graphics: Lecture 24

Alberto Paoluzzi

Tue, May 6, 2014

Outline: B-splines

Bezier curves in LAR

```
from splines import *
controlpoints = [[1,1],[2,3],[4,3],[3,1]]
dom = larDomain([32])
mapping = larBezierCurve(controlpoints)
obj = larMap(mapping)(dom)
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controlpoints)] ))
```

Figure : The 2D Bézier curve and the control polygon

Bezier curves in LAR

```
print dom,
>>> [[[0.0], [0.03125], [0.0625], ..., [0.90625], [0.9375], [0.96875], [1.0]],
    [[0, 1], [1, 2], [2, 3], ..., [29, 30], [30, 31], [31, 32]]]

print larBezierCurve(controlpoints),
>>> <function map_fn at 0x1174f0758>

print obj,
>>> [[[1.0, 1.0], [1.0965576171875, 1.181640625], [1.1982421875, 1.3515625], ...,
    [3.1533203125, 1.3515625], [3.0850830078125, 1.181640625], [3.0, 1.0]],
    [[0, 1], [1, 2], [2, 3], ..., [29, 30], [30, 31], [31, 32]]]
```

Uniform and Non-uniform B-splines

B-splines: Definition

Source from: **D.F. Rogers, An Introduction to NURBS**, highly recommended

Letting $P(t)$ be the position vector along the curve as a function of the parameter t, a B-spline curve is given by

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \qquad t_{\min} \le t < t_{\max}, \quad 2 \le k \le n+1 \qquad (3.1)$$

where the B_i are the position vectors of the $n+1$ control polygon vertices, and the $N_{i,k}$ are the normalized B-spline basis functions.

$N_{i,k}$: B-spline basis function with index i and order k

Normalized B-splines basis functions

For the ith normalized B-spline basis function of order k (degree $k-1$), the basis functions $N_{i,k}(t)$ are defined by the Cox–de Boor recursion formulas. Specifically

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } x_i \le t < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.2a}$$

and

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} \tag{3.2b}$$

order 1: step function

order k: convex combination of two functions of order $k-1$

knot values: non-decreasing sequence

The values of x_i are elements of a knot vector satisfying the relation $x_i \leq x_{i+1}$ (see Sec. 3.3). The parameter t varies from t_{min} to t_{max} along the curve $P(t)$.† The convention $\%_0 = 0$ is adopted.

B-splines: conditions satisfied

Formally, a B-spline curve is defined as a polynomial spline function of order k (degree $k-1$), because it satisfies the following two conditions:

$P(t)$ is a polynomial of degree $k-1$ on each interval $x_i \leq t < x_{i+1}$.

$P(t)$ and its derivatives of order $1, 2, \cdots, k-2$ are all continuous over the entire curve.

Thus, for example, a fourth-order B-spline curve is a piecewise cubic curve.

B-splines: properties

The sum of the B-spline basis functions for any parameter value t is (see [deBo72] and [Gord74])

$$\sum_{i=1}^{n+1} N_{i,k}(t) \equiv 1 \tag{3.3}$$

Each basis function is positive or zero for all parameter values, that is, $N_{i,k} \geq 0$.

Except for first-order basis functions, $k = 1$, each basis function has precisely one maximum value.

The maximum order of the curve equals the number of control polygon vertices. The maximum degree is one less.

The curve exhibits the variation-diminishing property. Thus, the curve does not oscillate about any straight line more often than its control polygon oscillates about the line.

The curve generally follows the shape of the control polygon.

Any affine transformation is applied to the curve by applying it to the control polygon vertices; i.e., the curve is transformed by transforming the control polygon vertices.

The curve lies within the convex hull of its control polygon.

B-splines: knot vector

Two types of knot vector are used:

- periodic
- open

B-splines: knot vector

Two types of knot vector are used:

- periodic
- open

in two subtypes:

- uniform
- nonuniform

B-splines: knot vector

Two types of knot vector are used:

- periodic
- open

in two subtypes:

- uniform
- nonuniform

Remark

open uniform knot vector has multiplicity of the knot values at the ends equal to the order k of the B-spline basis function

Uniform B-splines:

Examples from pyplasm or plasm.js

```
controls = [[0,0],[-1,2],[1,4],[2,3],[1,1],[1,2],[2.5,1],[2.5,3],[4,4],[5,0]];
knots = [0,0,0,0,1,2,3,4,5,6,7,7,7,7]
nubspline = NUBSPLINE(3)(knots)(controls)
VIEW(nubspline)    # DRAW(nubspline)  in plasm.js

controls = [[0,1],[1,1],[2,0],[3,0],[4,0],[5,-1],[6,-1]]
knots = [0,0,0,1,2,3,4,5,5,5]
nubspline = NUBSPLINE(2)(knots)(controls)
VIEW(STRUCT([nubspline,POLYLINE(controls)]))    # DRAW(nubspline)  in plasm.js

controls = [[0,1],[0,0],[1,0],[1,1],[0,1]]
knots = [0,0,0,1,2,3,3,3]
nubspline = NUBSPLINE(2)(knots)(controls)
VIEW(nubspline)    # DRAW(nubspline)  in plasm.js
```

Uniform B-splines:

Examples from `lar-cc`

```
controls = [[0,0],[-1,2],[1,4],[2,3],[1,1],[1,2],[2.5,1],[2.5,3],[4,4],[5,0]];
knots = [0,0,0,0,1,2,3,4,5,6,7,7,7,7]
bspline = BSPLINE(3)(knots)(controls)
dom = larIntervals([100])([knots[-1]-knots[0]])
obj = larMap(bspline)(dom)
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controls)] ))

controls = [[0,1],[1,1],[2,0],[3,0],[4,0],[5,-1],[6,-1]]
knots = [0,0,0,1,2,3,4,5,5,5]
bspline = BSPLINE(2)(knots)(controls)
dom = larIntervals([100])([knots[-1]-knots[0]])
obj = larMap(bspline)(dom)
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controls)] ))
```

Bezier curve as a B-spline curve

Examples from pyplasm or plasm.js

```
controls = [[0,1],[0,0],[1,1],[1,0]]
bezier = MAP( BEZIER(S1)(controls) )(INTERVALS(1)(32))
VIEW(bezier)
knots = [0,0,0,0,1,1,1,1]
nubspline = NUBSPLINE(3)(knots)(controls)
VIEW(nubspline)
VIEW(STRUCT([nubspline,bezier,POLYLINE(controls)]))
```

Bezier curve as a B-spline curve

Examples from `lar-cc`

```
controls = [[0,1],[0,0],[1,1],[1,0]]
bezier = larBezierCurve(controls)
dom = larIntervals([32])([1])
obj = larMap(bezier)(dom)
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controls)] ))

knots = [0,0,0,0,1,1,1,1]
bspline = BSPLINE(3)(knots)(controls)
dom = larIntervals([100])([knots[-1]-knots[0]])
obj = larMap(bspline)(dom)
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controls)] ))
```

B-spline curve: effect of double or triple control points

Examples from pyplasm or plasm.js

```
controls1 = [[0,0],[2.5,5],[6,2],[9,3]]
controls2 = [[0,0],[2.5,5],[2.5,5],[6,2],[9,3]]
controls3 = [[0,0],[2.5,5],[2.5,5],[2.5,5],[6,2],[9,3]]
knots = [0,0,0,0,1,1,1,1]
nubspline1 = NUBSPLINE(3)(knots)(controls1)
knots = [0,0,0,0,1,2,2,2,2]
nubspline2 = NUBSPLINE(3)(knots)(controls2)
knots = [0,0,0,0,1,2,3,3,3,3]
nubspline3 = NUBSPLINE(3)(knots)(controls3)
VIEW(STRUCT([nubspline1,nubspline2,nubspline3,POLYLINE(controls1)]))
```

B-spline curve: effect of double or triple control points

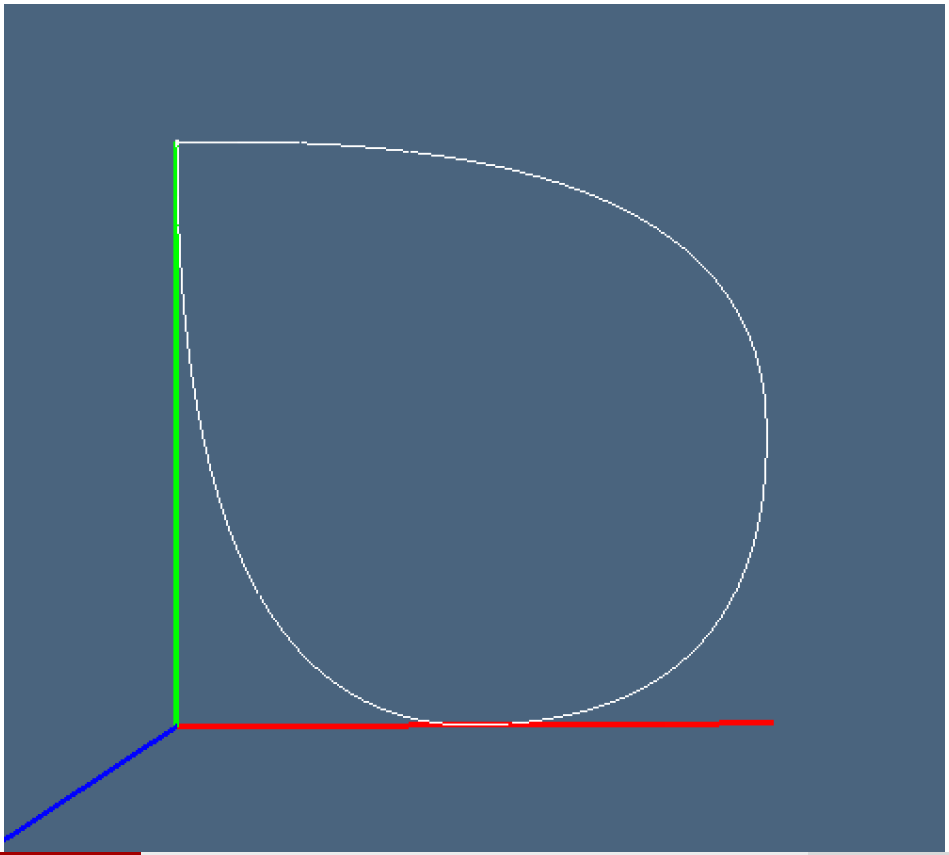
Examples from lar-cc

```
def larDom(knots,tics=32):
    domain = knots[-1]-knots[0]
    return larIntervals([tics*domain])([domain])

controls1 = [[0,0],[2.5,5],[6,1],[9,3]]
controls2 = [[0,0],[2.5,5],[2.5,5],[6,1],[9,3]]
controls3 = [[0,0],[2.5,5],[2.5,5],[2.5,5],[6,1],[9,3]]
knots = [0,0,0,0,1,1,1,1]
bspline1 = larMap( BSPLINE(3)(knots)(controls1) )(larDom(knots))
knots = [0,0,0,0,1,2,2,2,2]
bspline2 = larMap( BSPLINE(3)(knots)(controls2) )(larDom(knots))
knots = [0,0,0,0,1,2,3,3,3,3]
bspline3 = larMap( BSPLINE(3)(knots)(controls3) )(larDom(knots))

VIEW(STRUCT( CAT(AA(MKPOLS)([bspline1,bspline2,bspline3])) +
    [POLYLINE(controls1)]) )
```

Non periodic B-spline curve

Examples from pyplasm or plasm.js

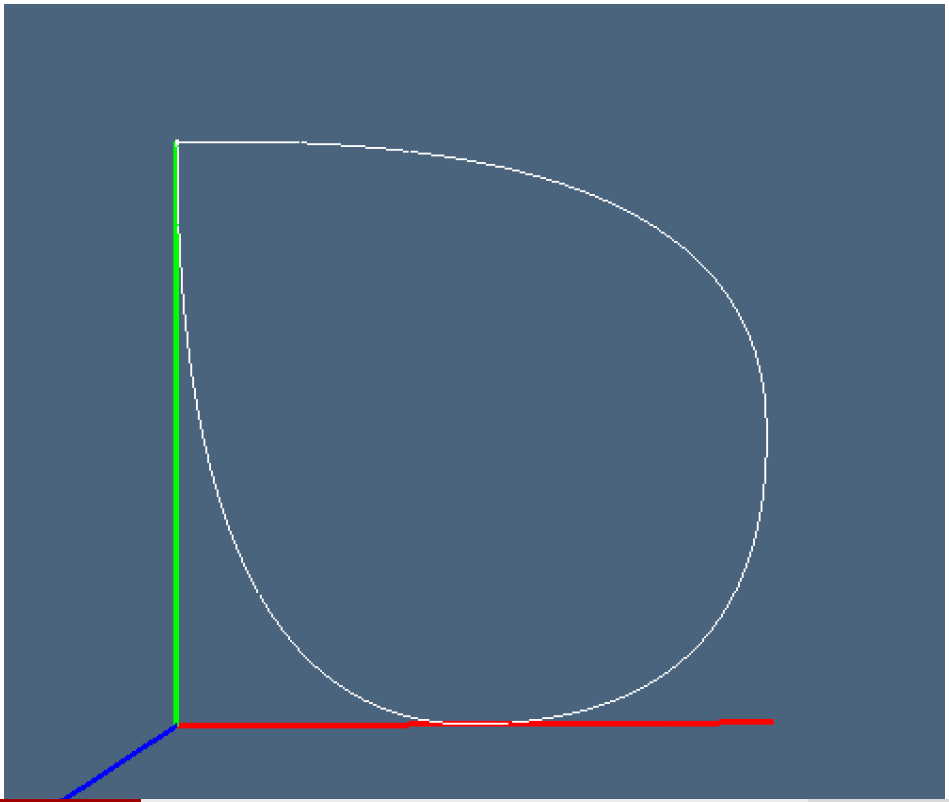
```
controls = [[0,1],[0,0],[1,0],[1,1],[0,1]]
knots = [0,0,0,1,2,3,3,3]
nubspline = NUBSPLINE(2)(knots)(controls)
VIEW(nubspline)    # DRAW(nubspline)  in plasm.js
```



Non periodic B-spline curve

Examples from lar-cc

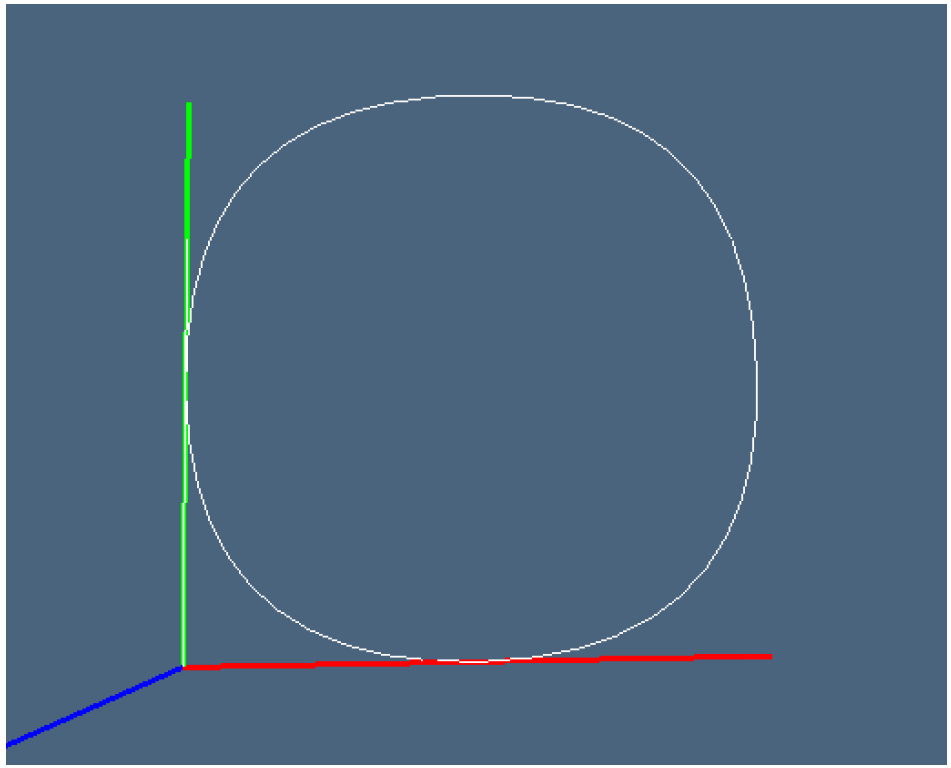
```
controls = [[0,1],[0,0],[1,0],[1,1],[0,1]]
knots = [0,0,0,1,2,3,3,3]
bspline = BSPLINE(2)(knots)(controls)
obj = larMap(bspline)(larDom(knots))
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controls)] ))
```



Periodic B-spline curve

Examples from pyplasm or plasm.js

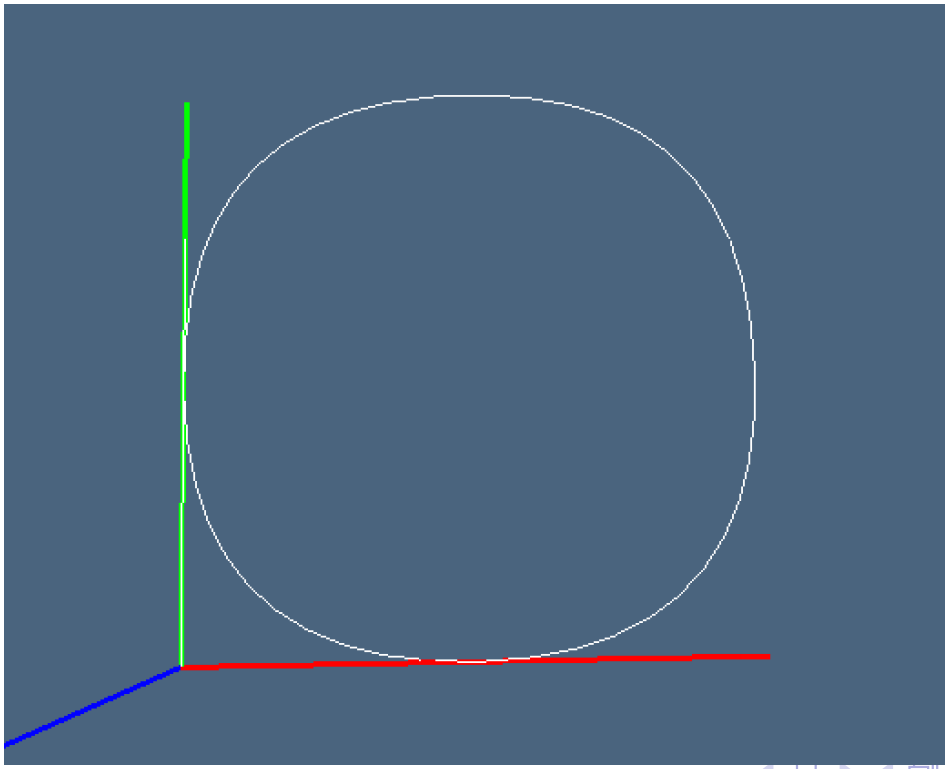
```
controls = [[0,1],[0,0],[1,0],[1,1],[0,1]]
knots = [0,1,2,3,4,5,6,7]
nubspline = NUBSPLINE(2)(knots)(controls)
VIEW(STRUCT([nubspline,POLYLINE(controls)]))    # DRAW(nubspline)  in plasm.js
```



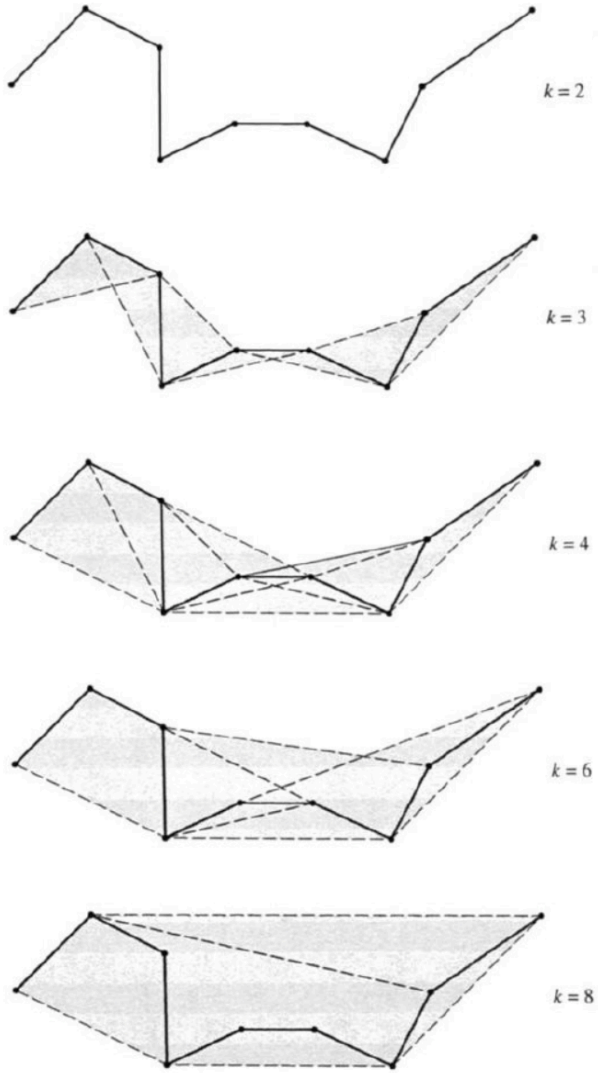
Periodic B-spline curve

Examples from lar-cc

```
controls = [[0,1],[0,0],[1,0],[1,1],[0,1]]
knots = [0,1,2,3,4,5,6,7]
bspline = BSPLINE(2)(knots)(controls)
obj = larMap(bspline)(larDom(knots))
VIEW(STRUCT( MKPOLS(obj) + [POLYLINE(controls)] ))
```



Convex hull properties of B-spline curves



Colinear curve segments

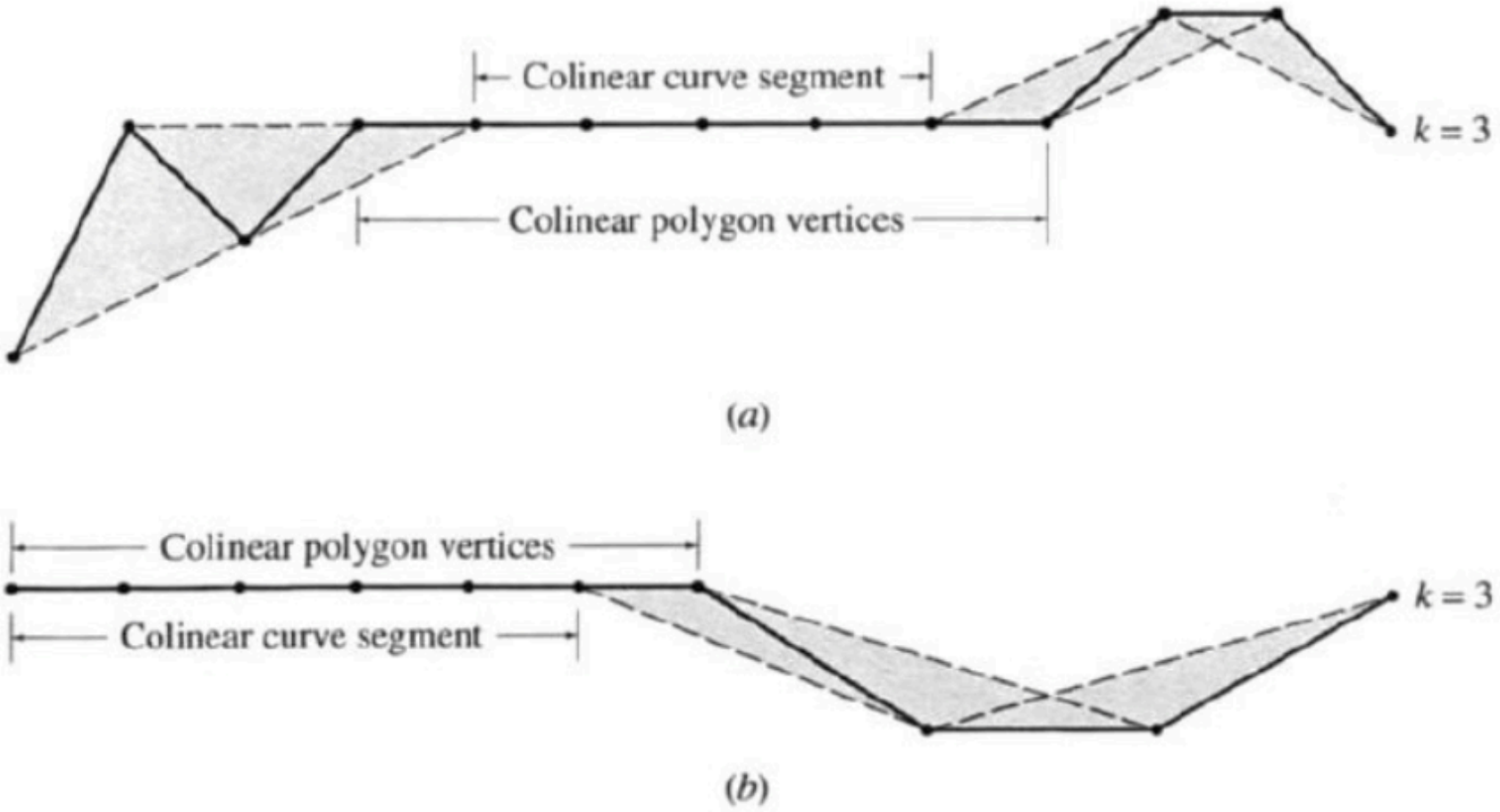


Figure 3.2 B-spline convex hull properties for colinear curve segments. (a) Within the control polygon vertices; (b) at the end of the control polygon vertices.

Periodic uniform B-spine basis functions

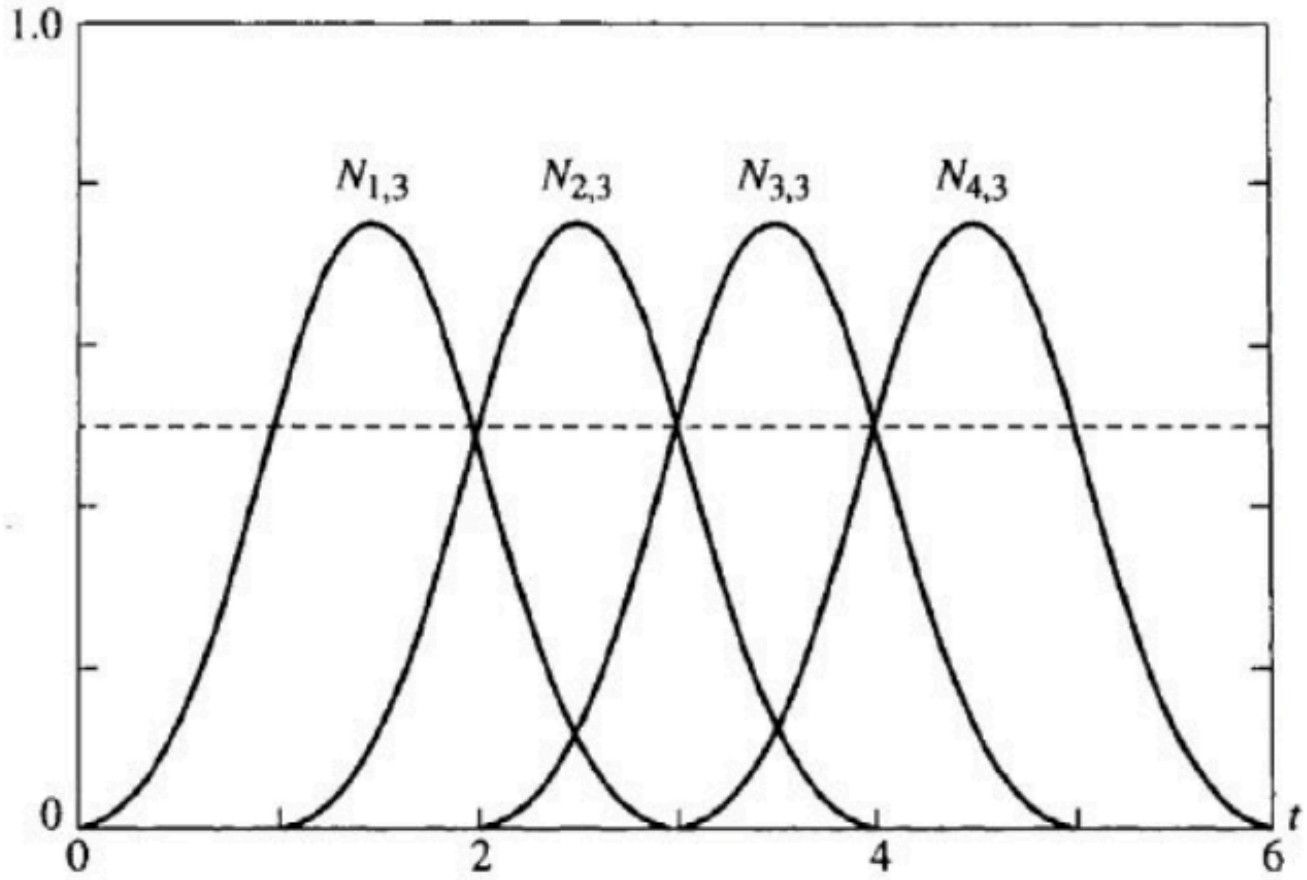


Figure 3.5 Periodic uniform B-spline basis functions, with $[X] = [0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]$, $n + 1 = 4$, $k = 3$.

Open uniform B-spine basis functions

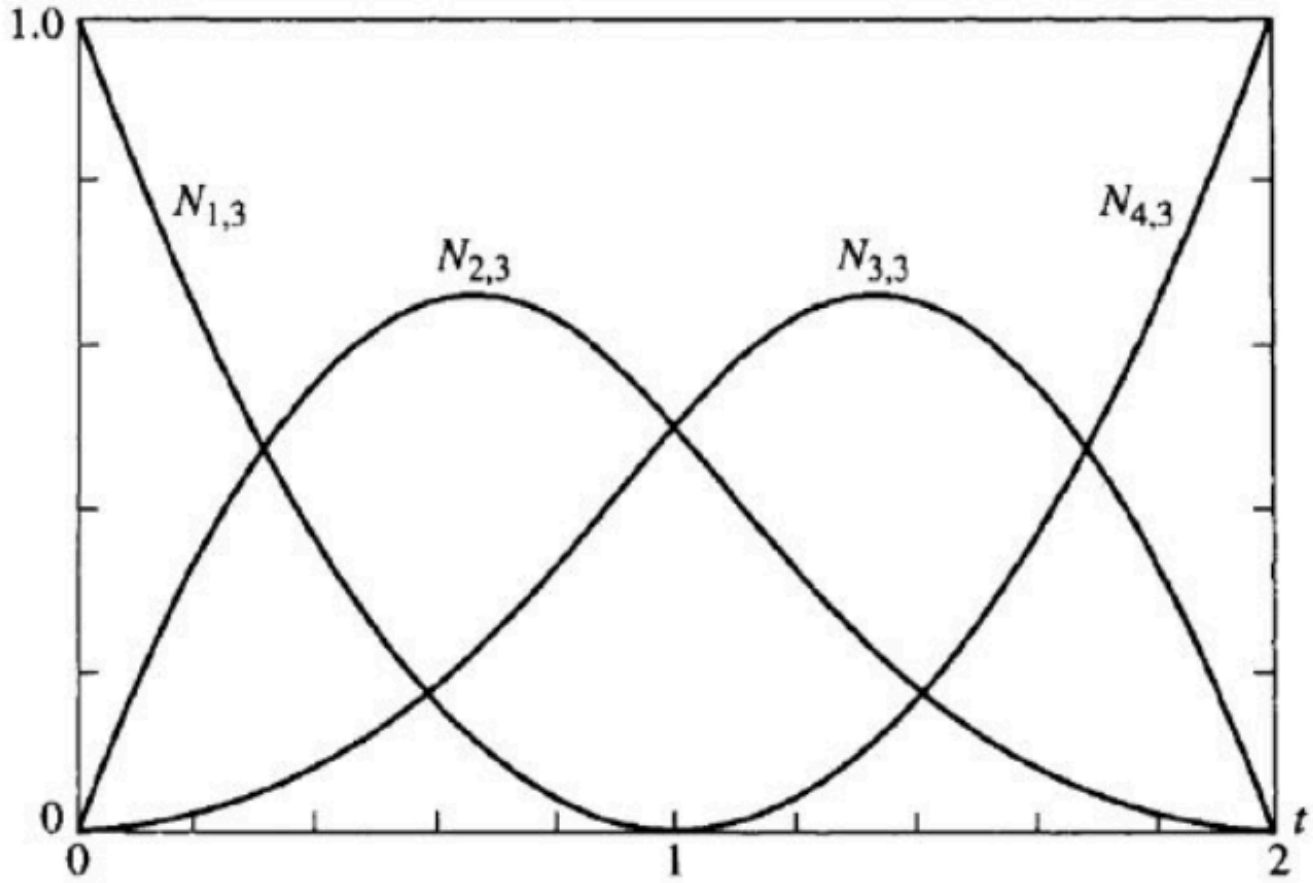


Figure 3.6 Open uniform B-spline basis functions, with $[X] = [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 2]$, $k = 3$, $n + 1 = 4$.

Non-uniform B-spine basis functions

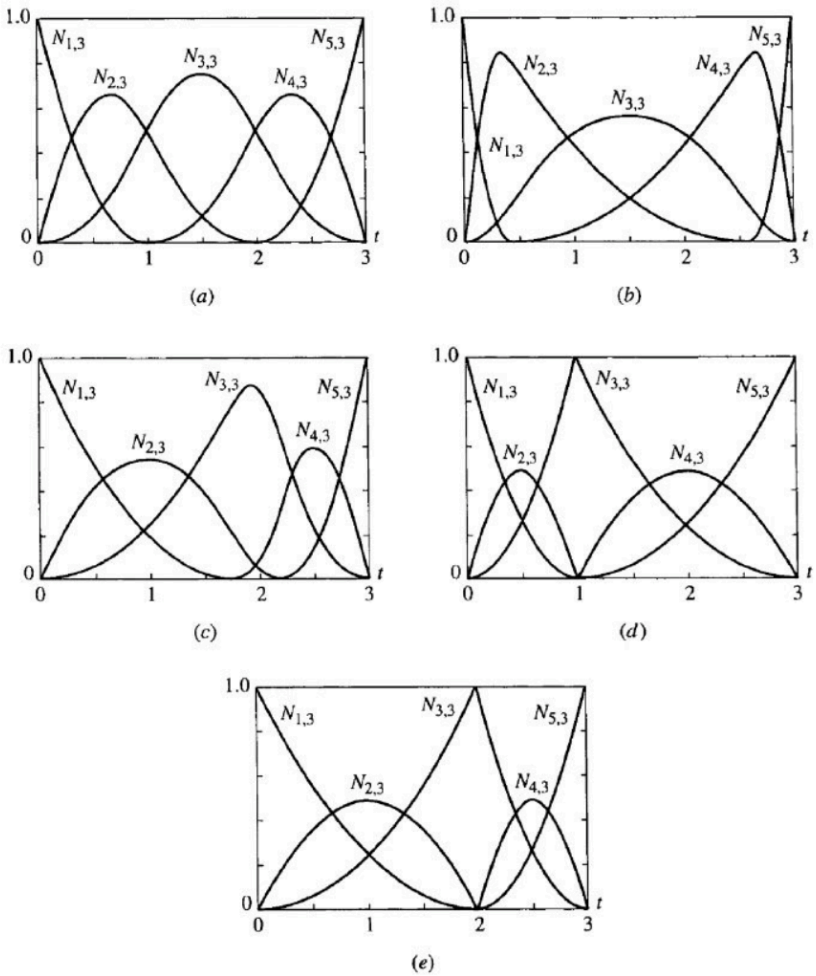


Figure 3.7 Nonuniform basis functions for $n + 1 = 5$, $k = 3$ compared to the open uniform basis function shown in (a). (a) $[X] = [0\ \ 0\ \ 0\ \ 1\ \ 2\ \ 3\ \ 3\ \ 3]$; (b) $[X] = [0\ \ 0\ \ 0\ \ 0.4\ \ 2.6\ \ 3\ \ 3\ \ 3]$; (c) $[X] = [0\ \ 0\ \ 0\ \ 1.8\ \ 2.2\ \ 3\ \ 3\ \ 3]$; (d) $[X] = [0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 3\ \ 3\ \ 3]$; (e) $[X] = [0\ \ 0\ \ 0\ \ 2\ \ 2\ \ 3\ \ 3\ \ 3]$.

Periodic basis functions build-up

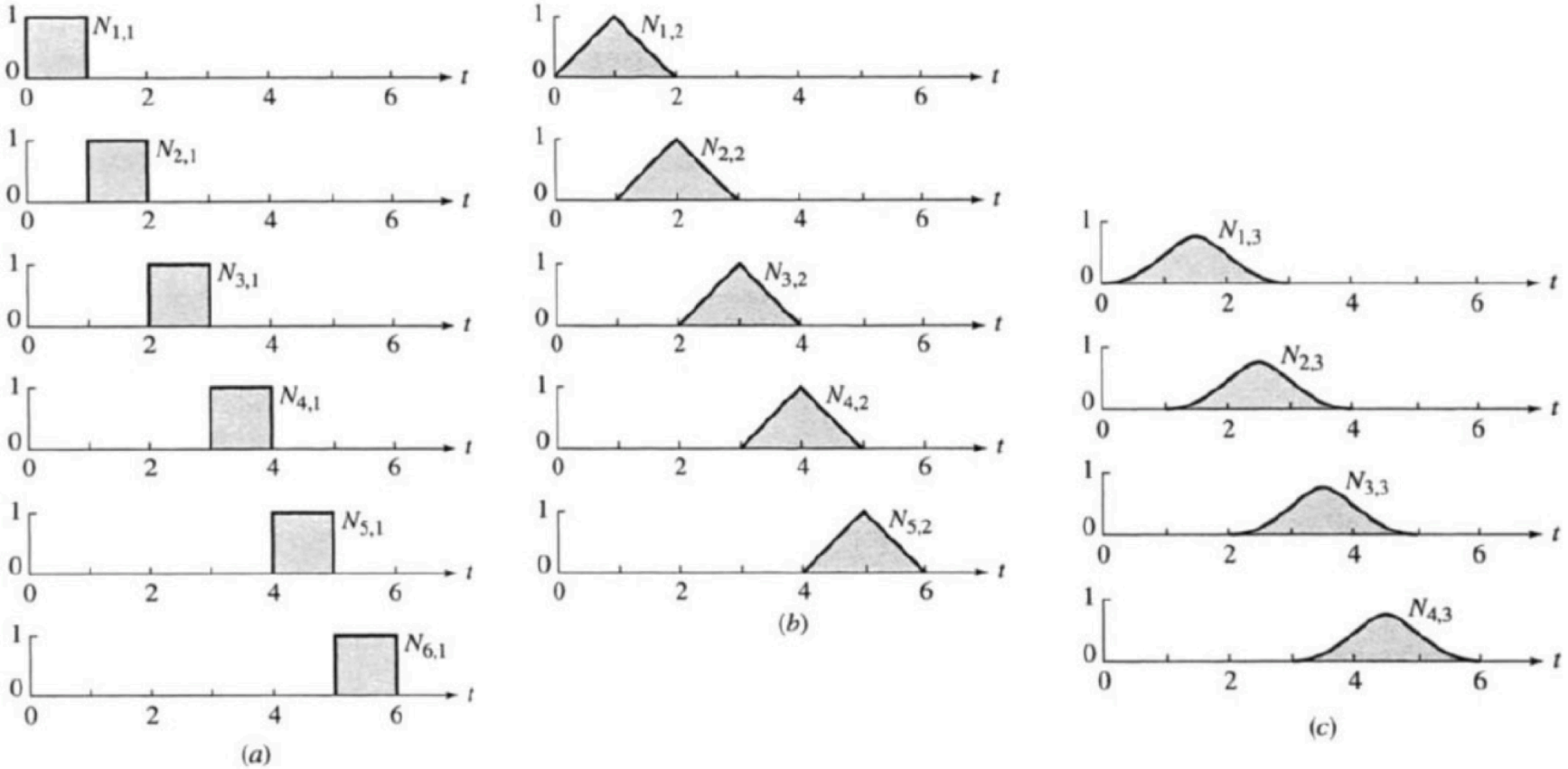


Figure 3.8 Periodic basis function buildup, $n + 1 = 4$. (a) $k = 1$; (b) $k = 2$; (c) $k = 3$.

Open basis functions build-up

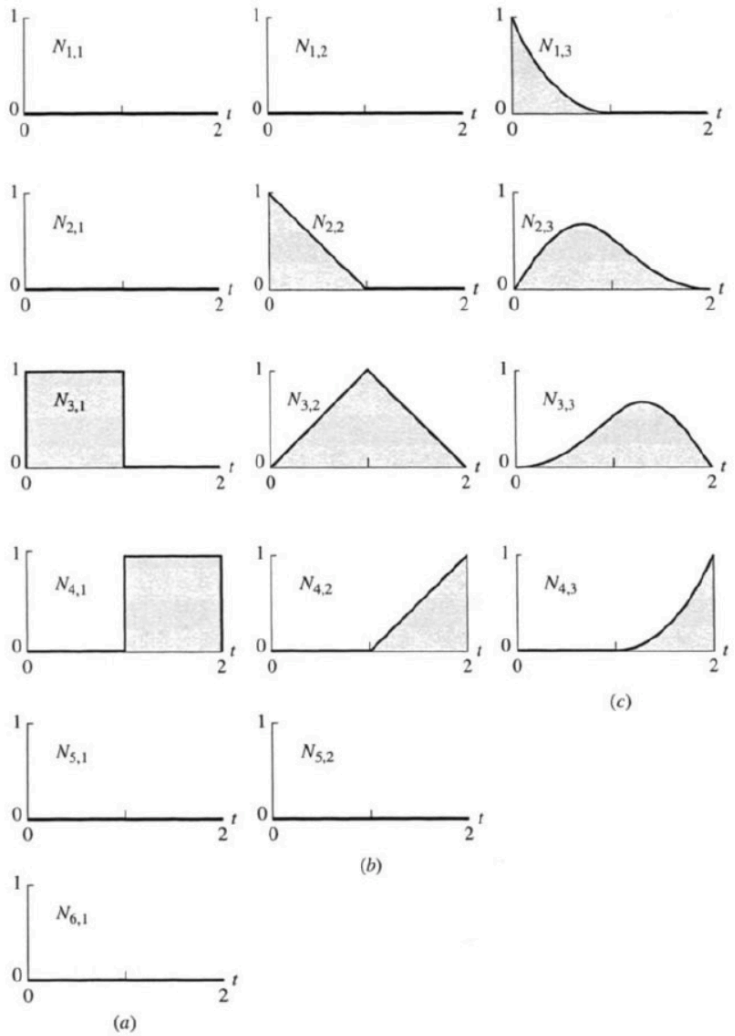


Figure 3.9 Open basis function buildup, $n + 1 = 4$. (a) $k = 1$; (b) $k = 2$; (c) $k = 3$.

Global control: effect of varying order

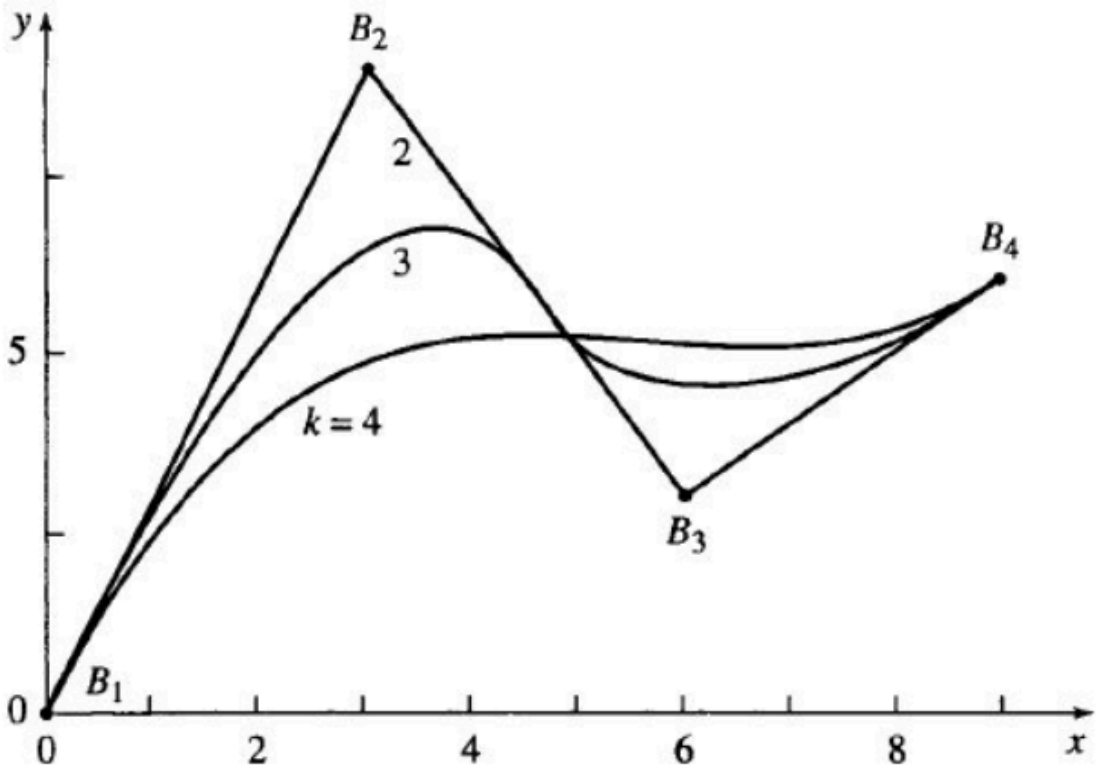


Figure 3.10 Effect of varying order on B-spline curves.

Local control: effect of multiple vertices

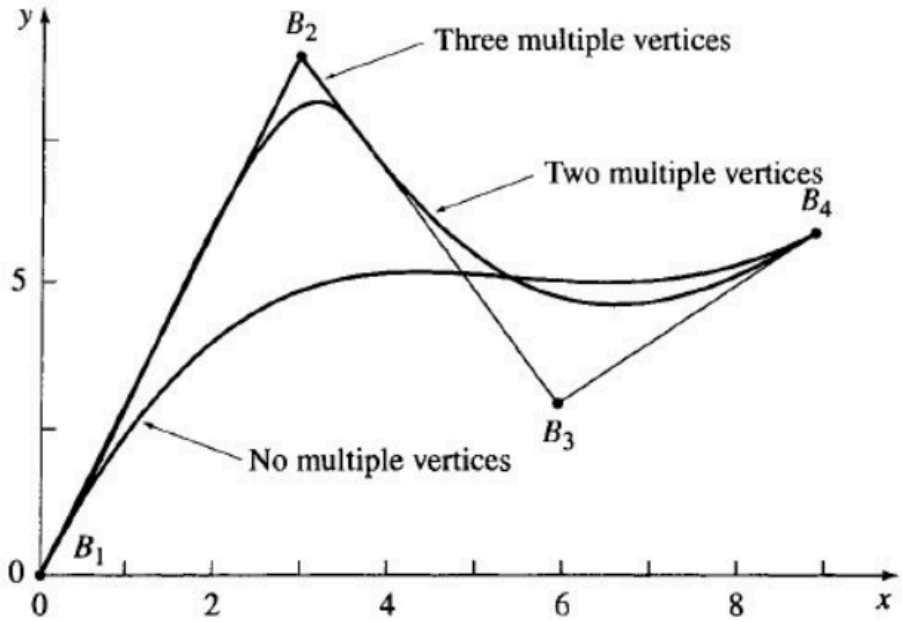


Figure 3.11 Effect of multiple vertices at B_2 on a B-spline curve, $k = 4$.

Local control: effect of moving vertices

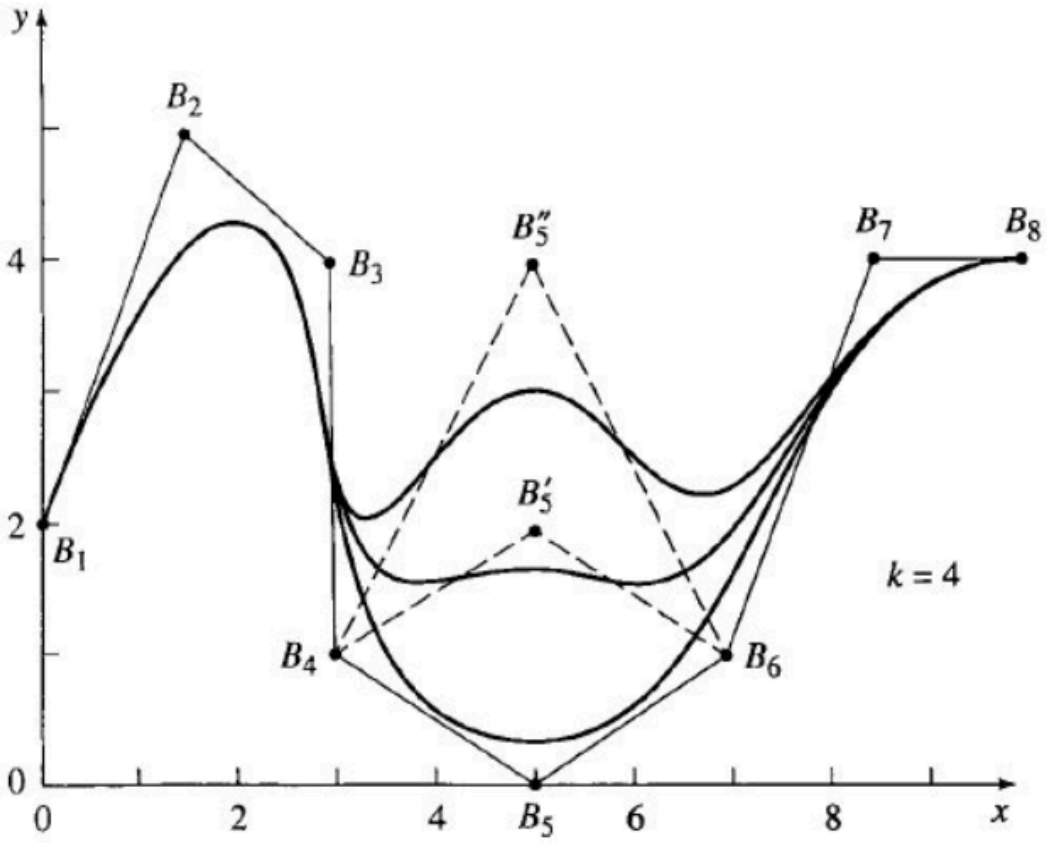


Figure 3.12 Local control of B-spline curves.

Example: closed periodic B-spline curve

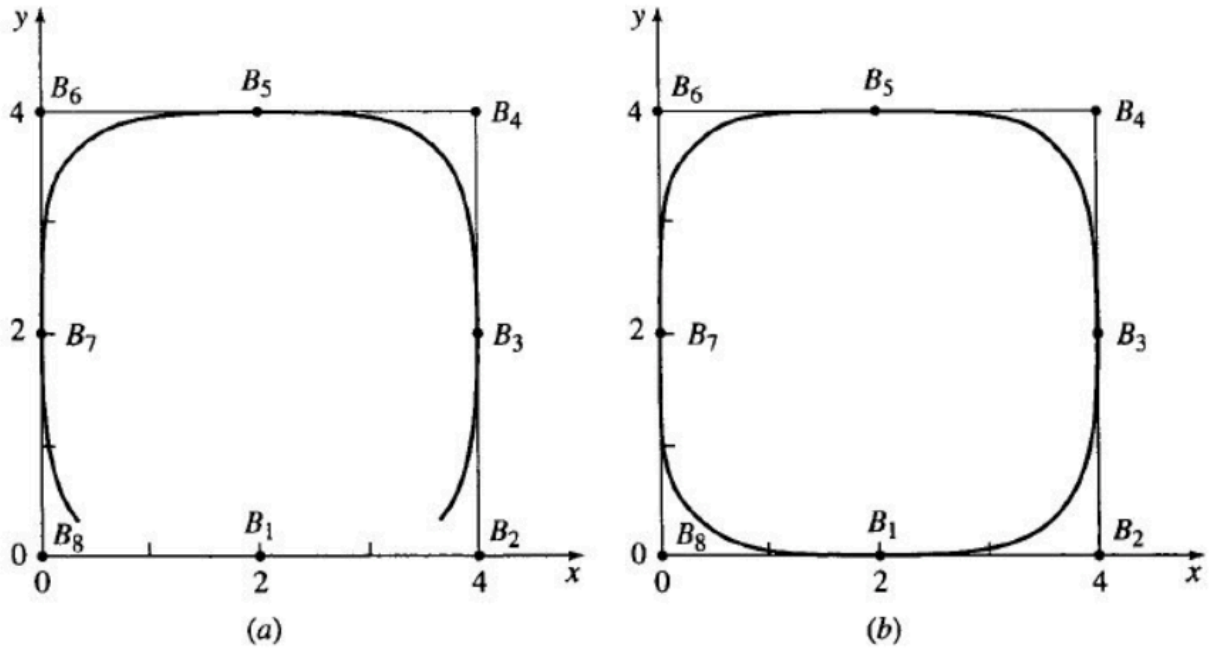


Figure 3.17 Closed periodic B-spline curve. (a) $B_1B_2B_3B_4B_5B_6B_7B_8B_1$ as the control polygon; (b) $B_8B_1B_2B_3B_4B_5B_6B_7B_8B_1B_2$ as the control polygon.

References

GP4CAD book

David F. Rogers, An introduction to NURBS, with historical perspective, Morgan Kaufmann, 2000