CSE-170 Computer Graphics

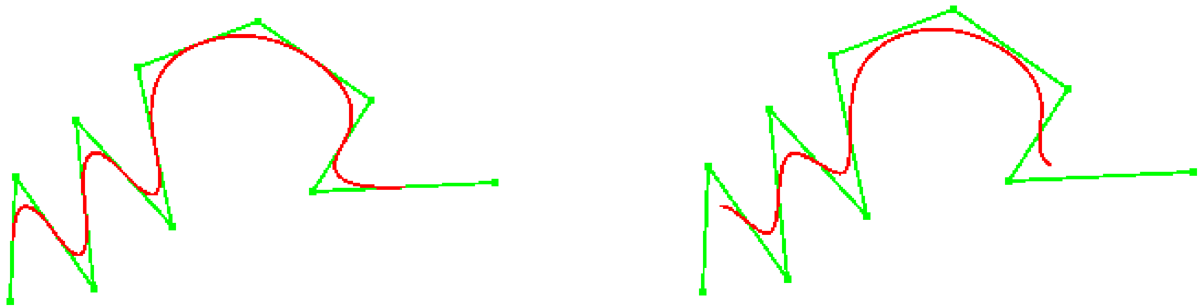
Lecture 21

B-Splines

Dr. Renato Farias
rfarias2@ucmerced.edu

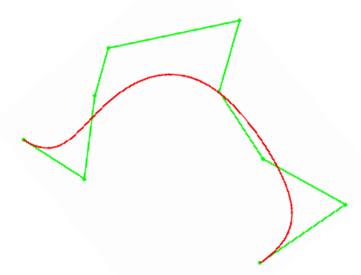
B-Splines

- "Basis Splines" can be defined for any number of control points
- Made of piecewise polynomials of chosen degree
- Main properties:
  - The degree does not depend on the number of control points (unlike Béziers)
  - Control polygon offers local curve modification/control
  - Continuity control w/ knot values: curve can be smooth or have sharp corners
- Degree 2 and 3 examples:



B-Splines vs. Béziers:

	Béziers	B-Splines
• Degree depends on number of control points:	True	False
• Local control:	no	yes
• Knot values:	no	yes
• End-point behavior:	reaches endpoints	varies

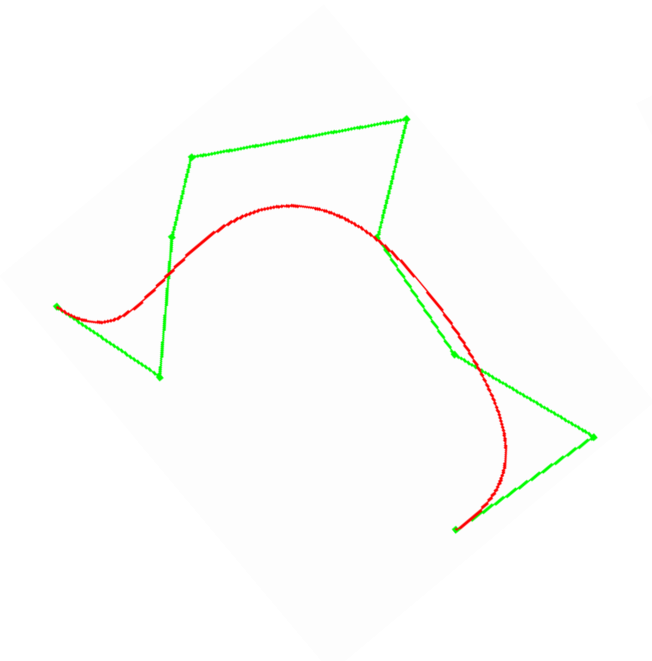


Examples: degree n quadratic cubic

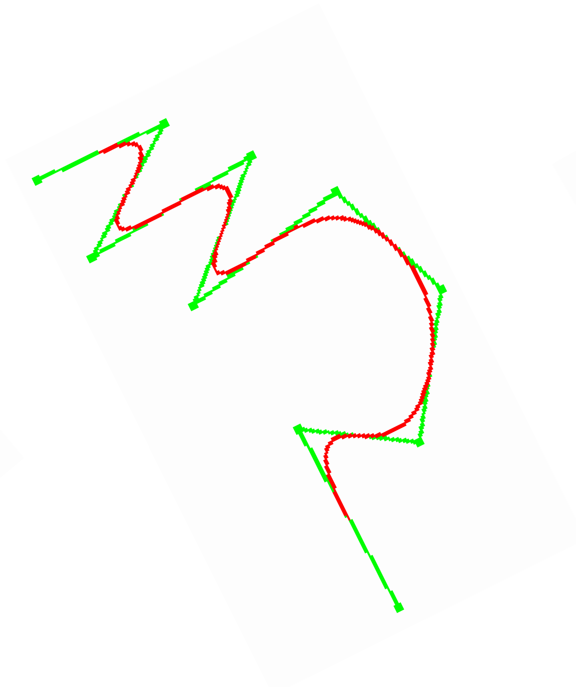
B-Splines vs. Béziers:

Make sure you can visually identify the main differences between these 3 cases:

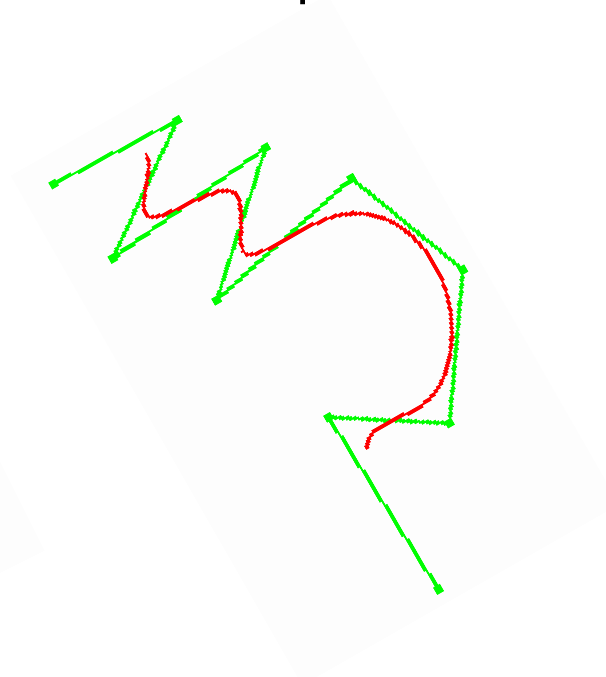
N-degree Bézier:



Quadratic B-Spline:



Cubic B-Spline:



Uniform B-Splines

Uniform B-Splines of degree 3

- We will focus our examples on B-Splines of degree 3

- Curve definition (for degree 3):

$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_i(u), u \in [3, n+1]$$

Uniform B-Splines of degree 3

- Properties of blending functions N (of degree 3):

$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_i(u), u \in [3, n+1]$$

a) 3rd degree polynomials, breaks at integer *u* values

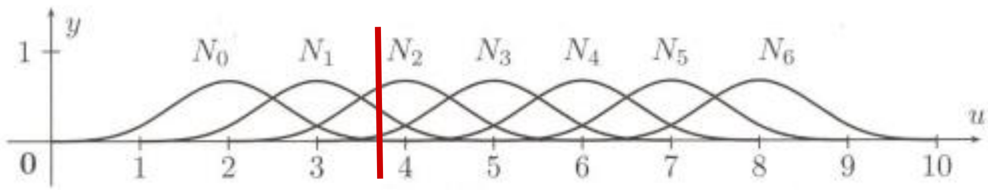
b) C^2-continous

c) They are translations of each other $\longrightarrow$ $N_i(u) = N_0(u-i)$

d) They form a partition of unity $\longrightarrow$ $\sum_i N_i(u) = 1, u \in [3, n+1]$

e) They are positive functions $\rightarrow$ $\forall u, N_i(u) \geq 0$

f) They have local support $\longrightarrow$ $\forall u : u \in [i+4, i], N_i(u) = 0$



Uniform B-Splines of degree 3

- Blending functions
  - There is only one set of blending functions that satisfy the desired properties:

$$R_0(u) = \frac{1}{6}u^3$$

$$R_1(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1)$$

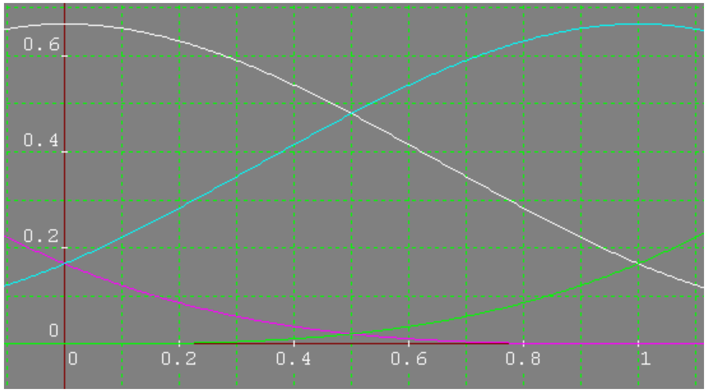
$$R_2(u) = \frac{1}{6}(3u^3 - 6u^2 + 4)$$

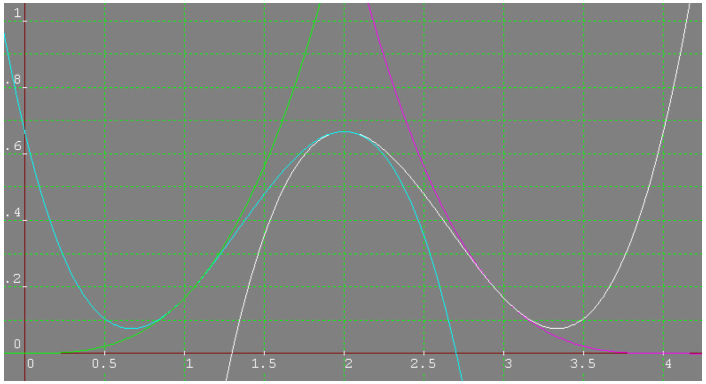
$$R_3(u) = \frac{1}{6}(1-u)^3$$

$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_i(u), u \in [3, n+1]$$

Each N function is composed of 4 R polynomials; for ex, N_0:

$$N_0(u) = \begin{cases} R_0(u), u \in [0,1] \\ R_1(u-1), u \in [1,2] \\ R_2(u-2), u \in [2,3] \\ R_3(u-3), u \in [3,4] \\ 0 \;\; otherwise \end{cases}$$





Uniform B-Splines of degree 3

- Blending functions
  - There is only one set of blending functions that satisfy the desired properties:

$$R_0(u) = \frac{1}{6}u^3$$

$$R_1(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1)$$

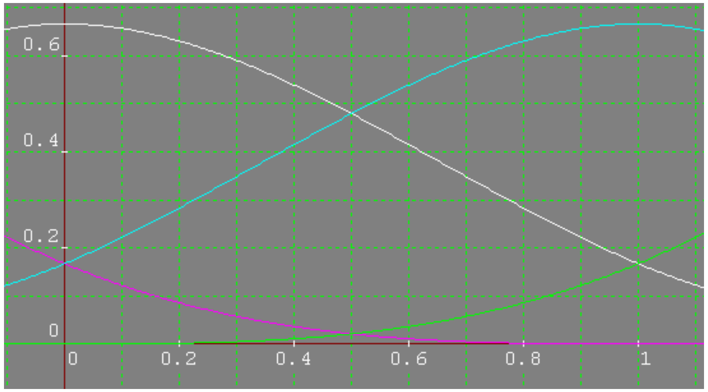
$$R_2(u) = \frac{1}{6}(3u^3 - 6u^2 + 4)$$

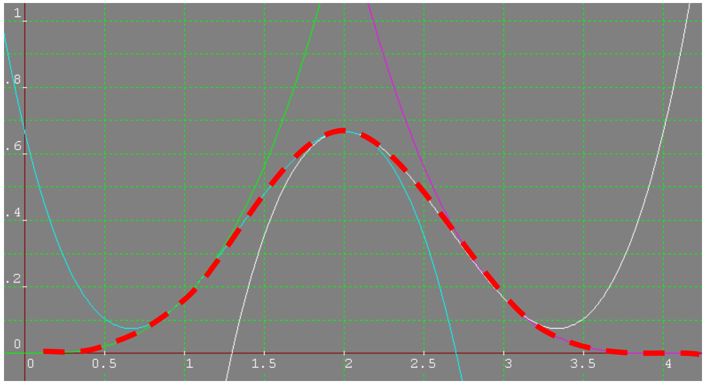
$$R_3(u) = \frac{1}{6}(1-u)^3$$

$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_i(u), u \in [3, n+1]$$

Each N function is composed of 4 R polynomials; for ex, N_0:

$$N_0(u) = \begin{cases} R_0(u), u \in [0,1] \\ R_1(u-1), u \in [1,2] \\ R_2(u-2), u \in [2,3] \\ R_3(u-3), u \in [3,4] \\ 0 \ \ otherwise \end{cases}$$





Uniform B-Splines of degree 3

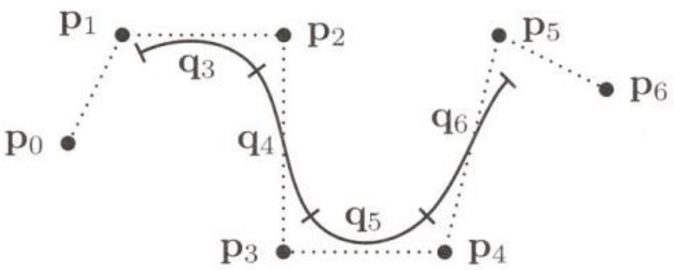
- Degree 3 case:

  <u>n+1 control points</u>

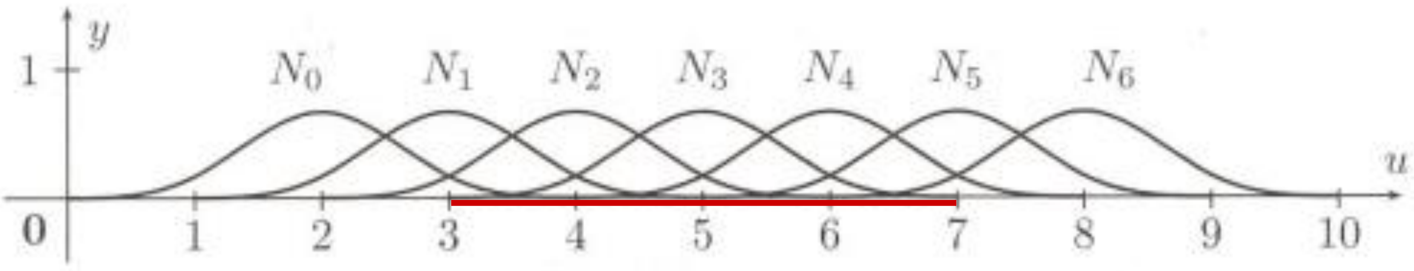
$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_i(u)$$

$u \in [3, n+1]$ ⟵ important

- Example: degree 3 curve with 7 control points, u in[3,7]:



  – basis functions, each w/ support [i,i+4]:

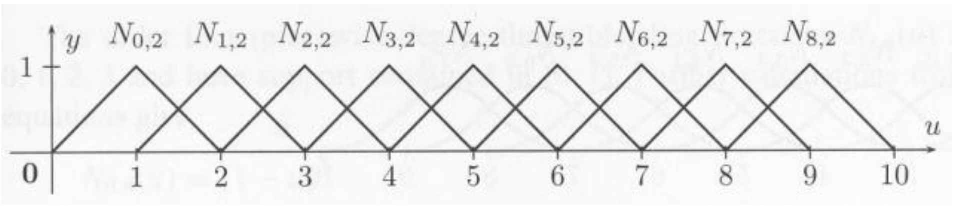


B-Splines of arbitrary degree

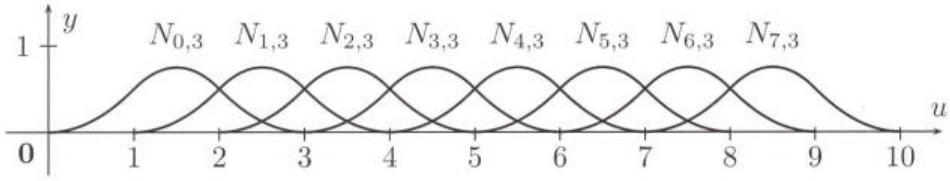
- Similar formulation can be defined for any order/degree (k=order, degree=k-1)

$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_{i,k}(u)$$

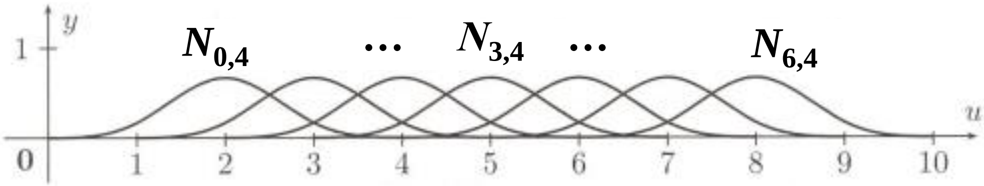
order 2
degree 1



order 3
degree 2



order 4
degree 3



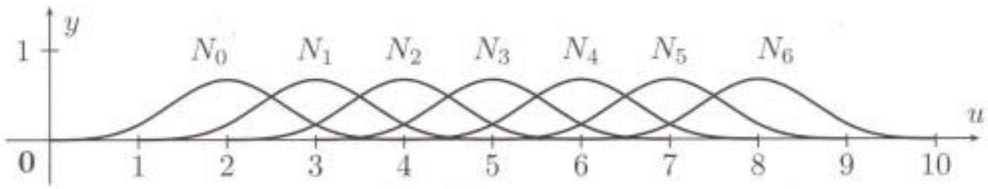
Non-Uniform B-Splines

Non-Uniform B-Splines

- Non-uniform B-Splines allow different behaviors to occur at each knot point

- Specified with "knot values" or a "knot vector"

- A "knot value" is a specific value that is given to modify the intervals along the *u* axis of the basis functions. The uniform version considers regular unit intervals for *u*:



Non-Uniform B-Splines

- For:
  - n+1 control points
  - order k, or degree m=k-1
- Then:
  - *l+1* knot values are given, *l*=*n*+*m*+1

knot values:

$$[u_0, u_1, \ldots, u_{l-1}, u_l]$$

$$u_0 \le u_1 \le \ldots \le u_{l-1} \le u_l$$

$$l = n + m + 1$$

$$\mathbf{q}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_{i,k}(u)$$

$$u \in [u_m, u_{n+1}]$$

Cox-de Boor Formula:

$$N_{i,1}(u) = \begin{cases} 1, u_i \le u \le u_{i+1} \\ 0, otherwise \end{cases}$$

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

B-Splines

- Example implementation of the N basis
  - Non-uniform: uses generic knot vector array
  - Uniform: knot vector will be [0,1,2,…] and thus can be derived from index i (just replace below u[i] by i)

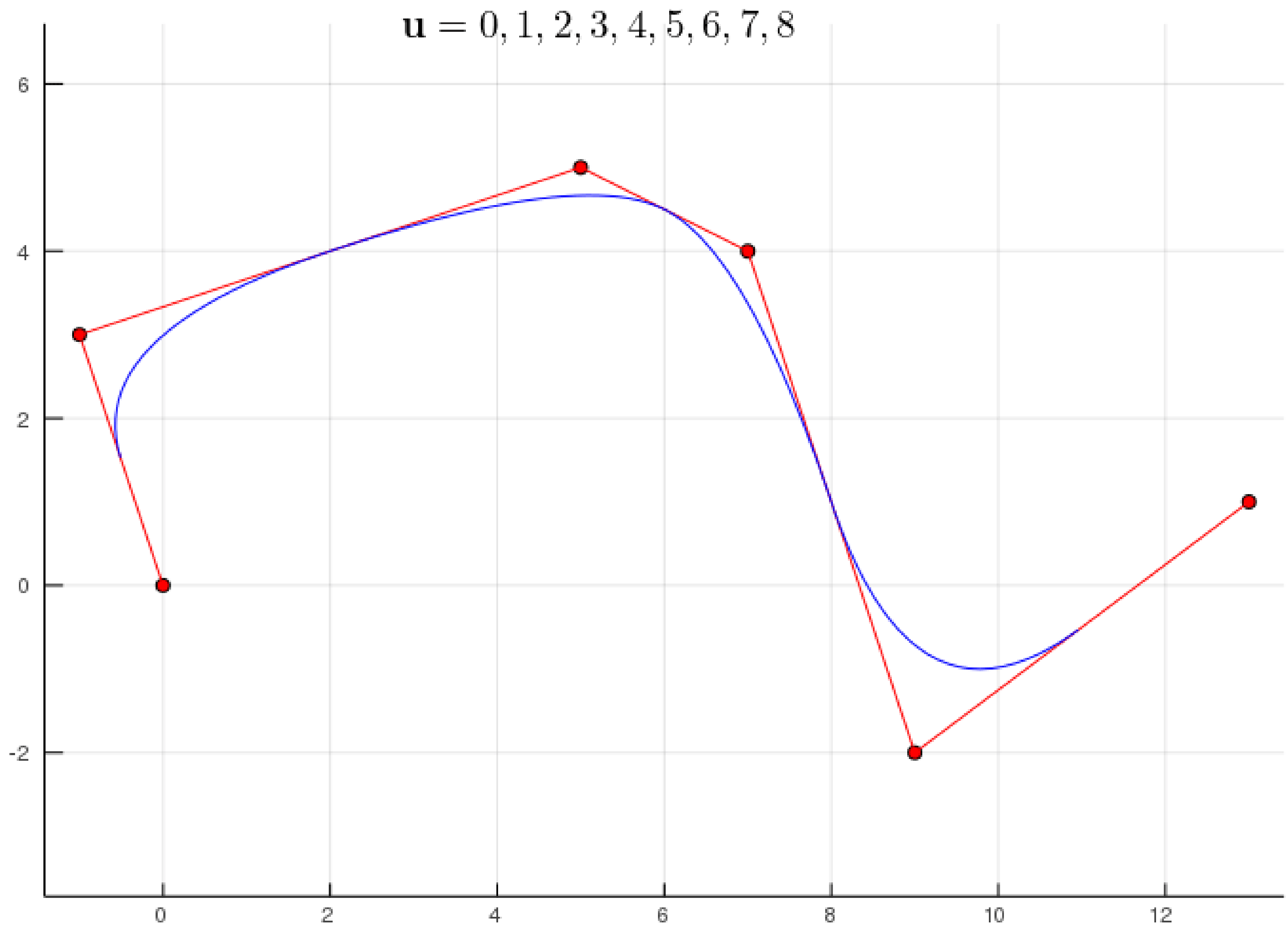
```
// n=pnts.size()-1=l-k-1, l=u.size()-1, order k,
static float N ( int i, int k, float t, const GsArray<float>& u )
 {
   if ( k==1 )
     return u[i]<=t && t<=u[i+1]? 1.0f:0;
   else
    return   ((t-u[i])/(u[i+k-1]-u[i])) * N(i,k-1,t,u) +
           ((u[i+k]-t)/(u[i+k]-u[i+1])) * N(i+1,k-1,t,u);
 }

// n=pnts.size()-1, i in [0,n], u in [3,n+1]
static float N ( int i, int k, float u )
 {
   float ui=float(i);
   if ( k==1 )
     return ui<=u && u<ui+1? 1.0f:0;
   else
    return   ((u-ui)/(k-1)) * N(i,k-1,u) +
           ((ui+k-u)/(k-1)) * N(i+1,k-1,u);
 }
```

Non-uniform knot vectors

- Can simulate Béziers, ex:
  - B-Spline order 4 is a cubic Bézier
  - Four control points, n=3
  - Knot vector: [0,0,0,0,1,1,1,1]

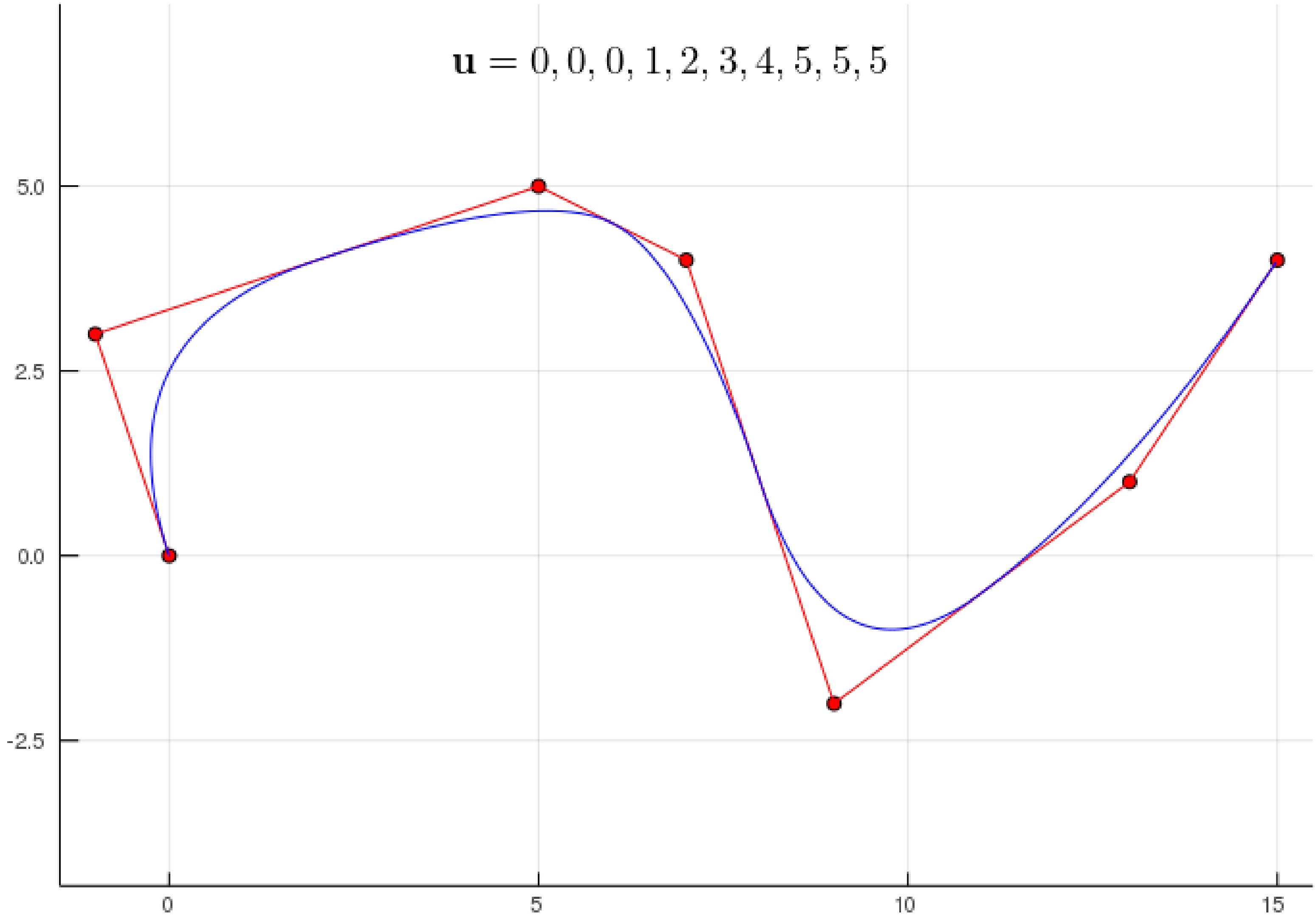
- Knot u_i with multiplicity w
  - When same knot value u_i appears w times
  - Curve will have only (order-w-1)th derivative continuity at u_i
  - Allows to define "sharp corners"
  - Ex: order:4 degree: 3
    multiplicity:3 C^0 continuity

Non-uniform knot vectors



https://opensourc.es/blog/b-spline/

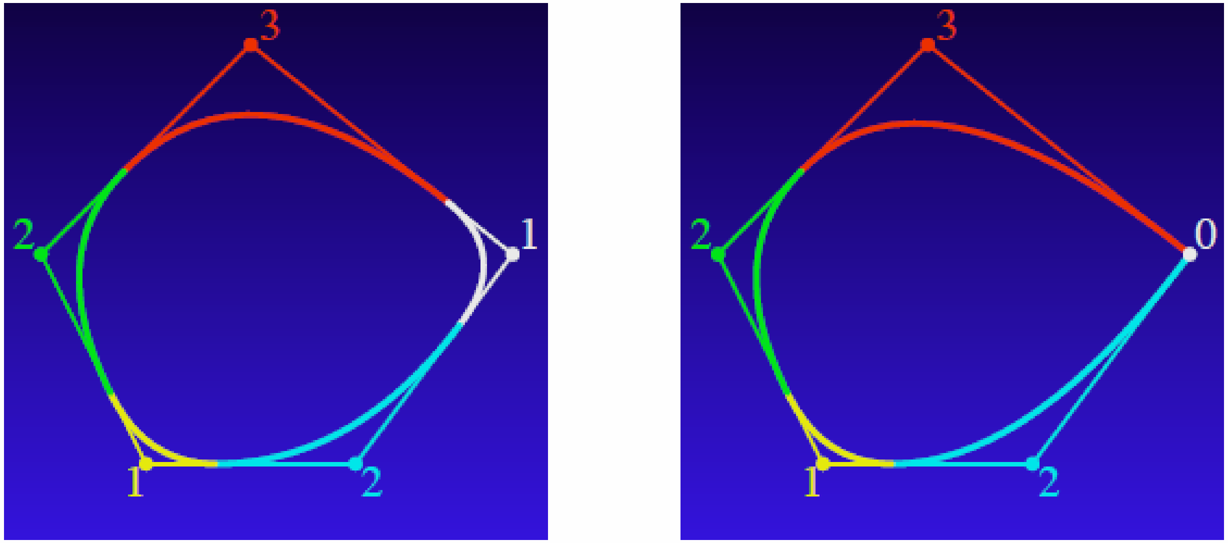
Non-uniform knot vectors



https://opensourc.es/blog/b-spline/

Non-uniform knot vectors

- Examples:



(a) Example 1. (b) Example with Zero Knot Interval.

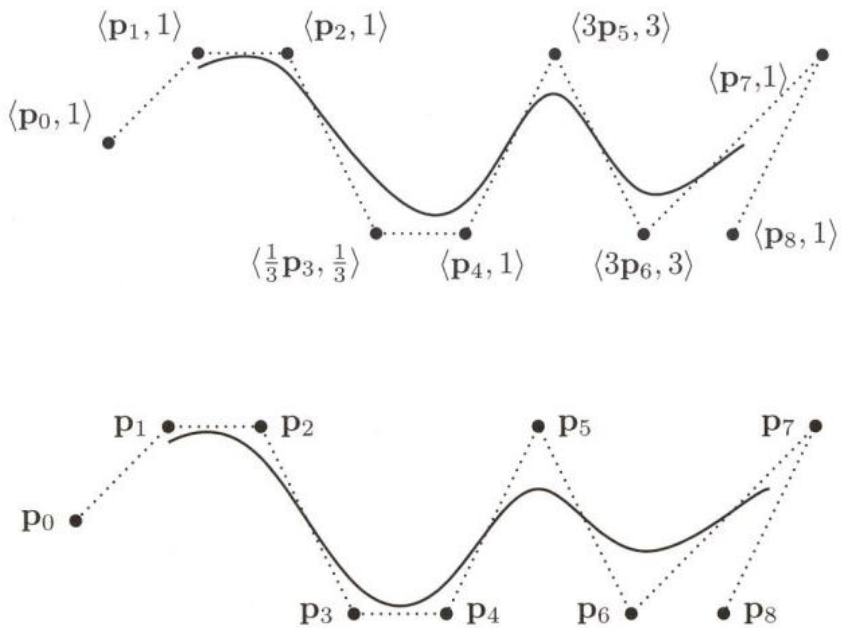
Figure 6.21: Quadratic B-Spline Curves.

  – Note: closed curves can be obtained by overlapping the final and first segment(s) of the control polygon (for the quadratic case)

Non-Uniform Rational B-Splines (NURBS)

NURBS

- NURBS: Non-Uniform Rational B-Splines
  - Control points are in homogeneous coordinates
  - Same as in the Rational Bézier case
  - The *w* component acts as a weight factor



NURBS

- Circles:
  - A non-rational B-spline curve cannot exactly represent a circle, but NURBS *can* be used to represent circles, and also all conics

- B-Spline Surfaces
  - Can be defined similarly to Bézier Patches

Extra

- For more information:
  - Available online: Computer Aided Geometric Design, by Thomas Sederberg
  - Curves and Surfaces for Computer Aided Geometric Design, by Gerald Farin

- Interesting Links
  - http://www.ibiblio.org/e-notes/Splines/Basis.htm
  - http://www.ibiblio.org/e-notes/Splines/Intro.htm