NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Module 4: B-Splines

Learning Objectives

- Understand the definition of B-splines
- Understand the properties of B-splines
- Learn how to perform computations with B-splines

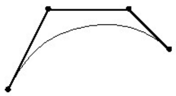
Sources

- Textbook (Chapter 3.3: Cubic splines)
- Joy's On-Line Geometric Modeling Notes (B-spline curves and patches)
  http://graphics.idav.ucdavis.edu/education/CAGDNotes/homepage.html
- Shene's Computing with Geometry Notes (Unit 5 and Unit 6)
  http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/notes.html

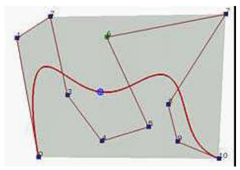
Outline

Introduction

- **Problem:** How can we efficiently and effectively design, represent and manipulate curves that can be used to interpolate or fit a (*large*) set of data points?
- **Background**
  - Bezier curves

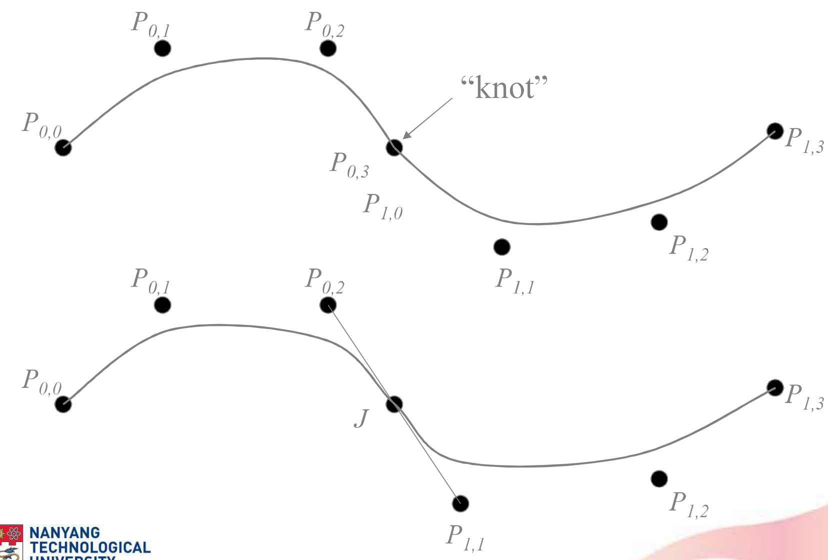


  - Increase the degrees of freedom
    - Use a higher degree Bezier curve (but with global control)
    - Use piecewise Bezier curves (but it is difficult to maintain continuity)



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

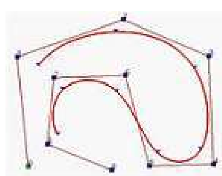
Piecewise Bezier curves



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

B-splines

- B-splines help to overcome these problems (local support, continuity control).
- Example: cubic B-spline curves

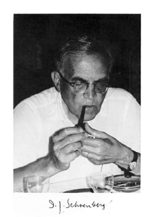


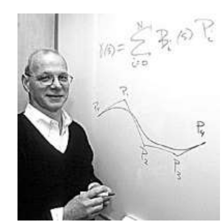
- A B-spline curve is defined in a similar fashion as a Bezier curve. That is, the curve is defined by the control polygon. However, the curve does not, in general, interpolate the control points.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

History

- Schoenberg: spline (1946)
- de Boor: recursive algorithm of B-spline (1966)
- Riesenfeld: B-spline for geometric design (1970s)

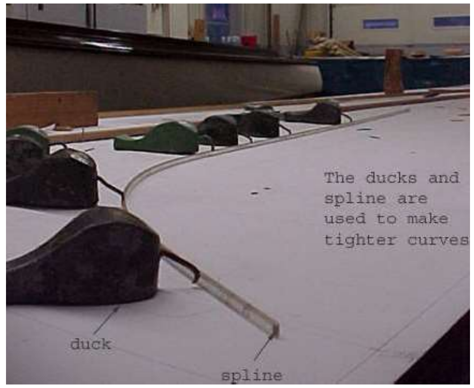




NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

What's a spline?

- Real world spline: a wooden beam which is used to draw smooth curves.

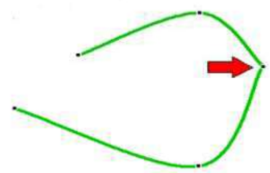


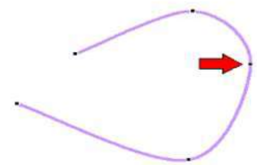
- Spline in mathematics: any composite curve formed with piecewise parametric polynomials subject to certain continuity conditions at the joints of the pieces.

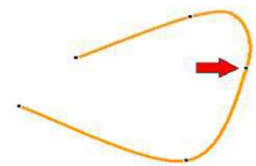
Measurement of continuity

Two curves: $\mathbf{r}_1(t), t \in [a,b]$ and $\mathbf{r}_2(t), t \in [b,c]$

- C^0 continuity: $\mathbf{r}_1(b) = \mathbf{r}_2(b)$
  - curve has no breaks (segments share the same points where join)
- C^1 continuity: $\mathbf{r}_1(b) = \mathbf{r}_2(b), \quad \mathbf{r}'_1(b) = \mathbf{r}'_2(b)$
  - 1st derivative is continuous
- C^2 continuity: $\mathbf{r}_1(b) = \mathbf{r}_2(b), \quad \mathbf{r}'_1(b) = \mathbf{r}'_2(b), \quad \mathbf{r}''_1(b) = \mathbf{r}''_2(b)$
  - 2nd derivative is continuous


C^0 continuity


C^1 continuity


C^2 continuity

Why does the continuity matter?

- Example 1 (modeling)

C^0 continuity

- Example 2 (animation)

Outline

§1. Introduction
§2. Formulation of B-splines
§3. Polar form / blossoming
§4. Applications
§5. Homework
§6. Summary

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

B-spline formulation

- **Definition of B-spline curves**
- **B-spline basis functions**
- **de Boor algorithm**
- **Properties of B-splines**

2.1 B-splines definition

Given

- control points P_i (i=0,…,n) called **de Boor points,** forming a control polygon;
- degree k;
- knot vector (or sequence) T = $\{u_0,\ldots,u_{n+k+1}\}$ where $u_0 \le \ldots \le u_{n+k+1}$ are the knots;

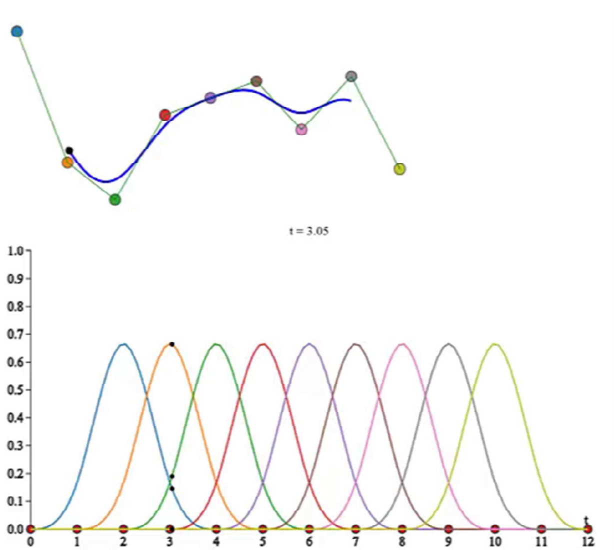
the B-spline curve of order (k+1) is defined by

$$r(u) = \sum_{i=0}^{n} P_i N_i^k(u), \qquad u \in [u_k, u_{n+1}]$$

where $N_i^k(u)$ are the **B-spline basis functions** defined over the knot vector T. The basis functions are ***piecewise*** degree k polynomials.

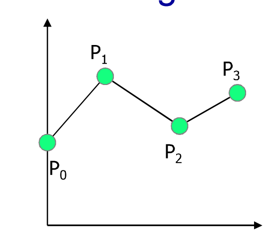
If all $u_{i+1}-u_i$ are the same, the curve is called the **uniform** B-spline curve; otherwise, it is a **non-uniform** B-spline curve.

Animation of a cubic B-spline curve



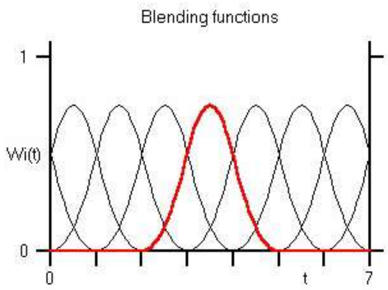
Example: degree 1 B-spline curve

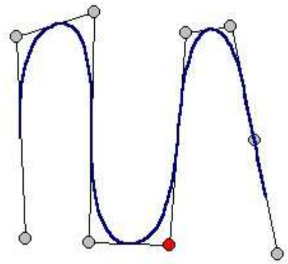
- 4 de Boor points (n=3), degree 1 (k=1), knot vector T = {0,1,2,3,4,5}
- Parameter domain of the curve is [1,4]. Or, this curve consists of 3 segments whose parameter domains are [1,2], [2,3], and [3,4], respectively.
- It is local since each de Boor point changes only 2 segments
- It is only C^0-continuous.



Example: a quadratic B-spline curve

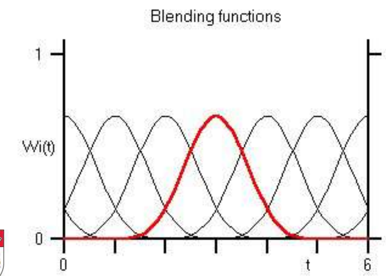
- 9 de Boor points (n=8), degree 2 (k=2), knot vector T = {-2,-1,0,1,2,3,4,5,6,7,8,9}
- Parameter domain of the curve is [0,7].
  The curve consists of 7 segments whose parameter domains are [0,1],[1,2], [2,3], [3,4],[4,5],[5,6], and [6,7] respectively. They are C^1–continuous.

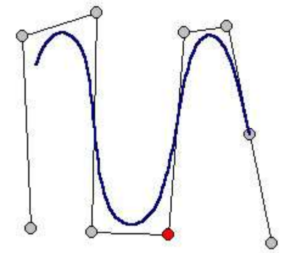




Example: a cubic B-spline curve

- 9 de Boor points (n=8), degree 3 (k=3), knot vector T = {-3,-2,-1,0,1,2,3,4,5,6,7,8,9}
- Parameter domain of the curve is [0,6].
  The curve consists of 6 segments whose parameter domains are [0,1],[1,2], [2,3], [3,4],[4,5],and [5,6] respectively. They are C^2–continuous.



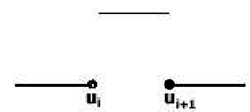


Demo

2.2 B-spline basis functions

- The basis functions are defined *recursively*:

$$N_i^0(u) = \begin{cases} 1, & u \in [u_i, u_{i+1}) \\ 0, & otherwise \end{cases}$$



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

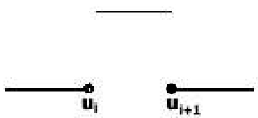
Note-- undetermined case: 0/0

- Question: verify that B-spline bases of degree *n* are non-zero only over *n*+1 intervals of the knot vector.

Degree 0 and 1 B-spline basis functions

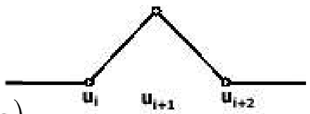
- Degree 0 B-spline basis

$$N_i^0(u) = \begin{cases} 1, & u \in [u_i, u_{i+1}) \\ 0, & otherwise \end{cases}$$



- Linear B-spline basis

$$N_i^1(u) = \begin{cases} \dfrac{u - u_i}{u_{i+1} - u_i}, & u \in [u_i, u_{i+1}) \\ \dfrac{u_{i+2} - u}{u_{i+2} - u_{i+1}}, & u \in [u_{i+1}, u_{i+2}) \end{cases}$$

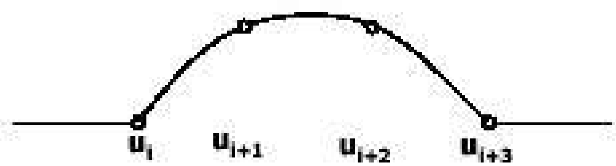


NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Degree 2 B-spline basis functions

- Quadratic B-spline basis

$$N_i^2(u) = \begin{cases} \dfrac{u - u_i}{u_{i+2} - u_i} \cdot \dfrac{u - u_i}{u_{i+1} - u_i}, & u \in [u_i, u_{i+1}) \\ \dfrac{u_{i+2} - u}{u_{i+2} - u_{i+1}} \cdot \dfrac{u - u_i}{u_{i+2} - u_i} + \dfrac{u_{i+3} - u}{u_{i+3} - u_{i+1}} \cdot \dfrac{u - u_{i+1}}{u_{i+2} - u_{i+1}}, & u \in [u_{i+1}, u_{i+2}) \\ \dfrac{u_{i+3} - u}{u_{i+3} - u_{i+1}} \cdot \dfrac{u_{i+3} - u}{u_{i+3} - u_{i+2}}, & u \in [u_{i+2}, u_{i+3}) \end{cases}$$



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Degree 3 B-spline basis functions

- Cubic B-spline basis

$$N_i^3(u) = \begin{cases} \dfrac{(u-u_i)^3}{(u_{i+1}-u_i)(u_{i+2}-u_i)(u_{i+3}-u_i)}, & u \in [u_i, u_{i+1}) \\[2ex] \dfrac{(u-u_i)^2(u_{i+2}-u)}{(u_{i+2}-u_{i+1})(u_{i+3}-u_i)(u_{i+2}-u_i)} + \dfrac{(u_{i+3}-u)(u-u_i)(u-u_{i+1})}{(u_{i+2}-u_{i+1})(u_{i+3}-u_{i+1})(u_{i+3}-u_i)} + \dfrac{(u_{i+4}-u)(u-u_{i+1})^2}{(u_{i+2}-u_{i+1})(u_{i+4}-u_{i+1})(u_{i+3}-u_{i+1})}, & u \in [u_{i+1}, u_{i+2}) \\[2ex] \dfrac{(u-u_i)(u_{i+3}-u)^2}{(u_{i+3}-u_{i+2})(u_{i+3}-u_{i+1})(u_{i+3}-u_i)} + \dfrac{(u_{i+4}-u)(u_{i+3}-u)(u-u_{i+1})}{(u_{i+3}-u_{i+2})(u_{i+4}-u_{i+1})(u_{i+3}-u_{i+1})} + \dfrac{(u_{i+4}-u)^2(u-u_{i+2})}{(u_{i+3}-u_{i+2})(u_{i+4}-u_{i+2})(u_{i+4}-u_{i+1})}, & u \in [u_{i+2}, u_{i+3}) \\[2ex] \dfrac{(u_{i+4}-u)^3}{(u_{i+4}-u_{i+3})(u_{i+4}-u_{i+2})(u_{i+4}-u_{i+1})}, & u \in [u_{i+3}, u_{i+4}) \end{cases}$$

Basis function dependencies

- Form triangular pattern

$$\begin{array}{ccccc} N_i^k & & & & \\ N_i^{k-1} & N_{i+1}^{k-1} & & & \\ N_i^{k-2} & N_{i+1}^{k-2} & N_{i+2}^{k-2} & & \\ \vdots & \vdots & \vdots & \ddots & \\ N_i^0 & N_{i+1}^0 & N_{i+2}^0 & \cdots & N_{i+k}^0 \end{array}$$

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

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Basis function inverse dependencies

- Form triangular pattern

$$
\begin{array}{ccccc}
N_{i-k}^{k} & \cdots & N_{i-2}^{k} & N_{i-1}^{k} & N_{i}^{k} \\
 & \ddots & \vdots & \vdots & \vdots \\
 & & N_{i-2}^{2} & N_{i-1}^{2} & N_{i}^{2} \\
 & & & N_{i-1}^{1} & N_{i}^{1} \\
 & & & & N_{i}^{0}
\end{array}
$$

- Influence of a single first-order basis function N_i^0 on higher-order basis functions.

Properties of B-spline basis functions

- Partition of unity: $\sum_{i=0}^{n} N_i^k(u) \equiv 1$
- Positivity: $N_i^k(u) \geq 0$
- Compact support: $N_i^k(u) = 0, \quad for\ u \notin [u_i, u_{i+k+1}]$
- Continuity: $N_i^k(u)$ is C^{k-1} continuous.

2.3 de Boor algorithm

- Generalization of de Casteljau algorithm
- Evaluation of a point on the curve at *u*=*t* by successive linear interpolation: for a given $t \in [u_j, u_{j+1}]$, consider those points $P_{j-k}, \ldots, P_j$,

$$P_i^0 = P_i, \quad i = j-k, \ldots, j$$

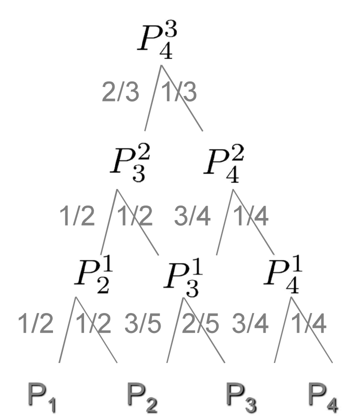
$$P_i^h = \left(1 - \frac{t-u_i}{u_{i+k+1-h} - u_i}\right) P_{i-1}^{h-1} + \frac{t-u_i}{u_{i+k+1-h} - u_i} P_i^{h-1}, \quad h > 0$$

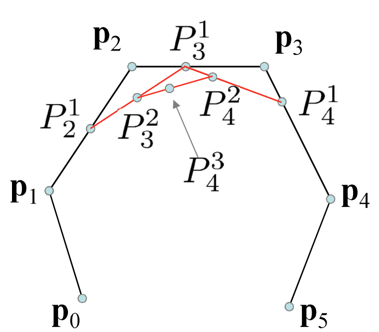
$$r(t) = P_j^k$$

Example: de Boor algorithm

Cubic, knot vector = [0 0 0 0 1 4 5 5 5 5]

=[$u_0,u_1,u_2,u_3,u_4,u_5,u_6,u_7,u_8,u_9$], evaluate at t = 2





$$\mathbf{p}_i^l = \frac{u_{4+i-l} - t}{u_{4+i-l} - u_i}\mathbf{p}_{i-1}^{l-1} + \frac{t-u_i}{u_{4+i-l} - u_i}\mathbf{p}_i^{l-1}$$

2.4 Properties

Affine invariance

– You can scale, rotate and translate the curve by scaling, rotating or translating the control points.

Excellent locality

– Change of one control point affects at most k+1 segments where k is the degree.

The degree of the global curve doesn't depend on the number of points

– Efficient for modelling curves with many points

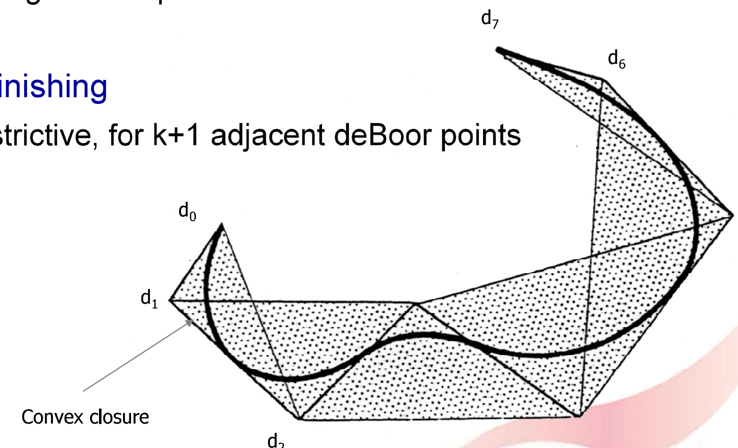
Properties

Strong convex hull

– A point on the curve lies within the convex hull of k+1 neighboring deBoor points

Variation diminishing

– More restrictive, for k+1 adjacent deBoor points



A C^1-continuous quadratic B-Spline

2.5 Recap of B-spline curves

For a B-spline curve $r(u)=\sum_{i=0}^{n} P_i N_i^k(u), \qquad u\in[u_k,u_{n+1}]$

- order = k+1
- degree = k
- number of de Boor points + order = number of knots
- The control points are P_i (i=0,…,n).
- The knots are $\{u_0,\dots,u_{n+k+1}\}$. The first and the last knots have no actual effect on the curve.
- The curve consists of (n+1-k) segments, which correspond to the knot spans $[u_k,u_{k+1}]$, $[u_{k+1},u_{k+2}]$, …, $[u_n,u_{n+1}]$.
- To compute a point on the curve for $t\in[u_j,u_{j+1}]$, only points $P_{j-k},\dots,P_j$ are involved.

Recap of B-spline curves

- The curve segment defined over knot span $[u_j,u_{j+1}]$ is contained in the convex hull of points $P_{j-k},\dots,P_j$.
- Moving a de Boor point P_j will affect the curve segment(s) defined over the knot span $[u_j,u_{j+k+1}]$.
- If all $u_{i+1}-u_i$ are the same, the curve is a uniform B-spline curve; otherwise, the curve is a non-uniform curve. Non-uniform includes
  - *different lengths of knot spans*
  - *multiple knots*
- At a simple knot u_i, the B-spline curve is C^{k-1} continuous.
- At a multiple knot u_i with multiplicity h, the B-spline curve is C^{k-h} continuous.

Recap of B-spline curves

- In case a multiple knot u_i has multiplicity *k* (assume $u_i = \ldots = u_{i+k-1}$), then the B-spline curve interpolates de Boor point P_{i-1}.
  - *If $u_1 = \ldots = u_k$, then the B-spline curve interpolates the first de Boor point.*
  - *If $u_{n+1} = \ldots = u_{n+k}$, then the B-spline curve interpolates the last de Boor point.*
  - *In particular, if the knot vector is $\{u_0, \ldots, u_0, u_{n+1}, \ldots, u_{n+1}\}$, the B-spline curve becomes a Bezier curve defined over $[u_0, u_{n+1}]$.*

Example

A degree 4 B-spline curve is defined by 8 control points P_0 to P_7 and knot vector {0,0,0,0,0,1,2,3,4,4,4,4,4}.

- order = 5
- 5 + 8 = 13 (number of knots)
- u0 = u1 = u2 = u3 = u4 = 0, u5 = 1, u6 = 2, u7 = 3, u8=…=u12 = 4
- u1=u2=u3=u4 ➔ the curve interpolates P_0 (i.e., r(0) = P_0).
- u8=u9=u10=u11 ➔ the curve interpolates P_7 (i.e., r(4) = P_7).
- The curve has 4 segments: [0,1],[1,2],[2,3],[3,4].
- Moving point P_5 will affect curve segments over [1,4].
- The segment with knot span [1,2] lies within the convex hull of points P_1 to P_5

More examples

- A degree 3 Bezier curve is a B-spline curve with knot vector {0,0,0,0,1,1,1,1}.

  8 = (3+1)+4

- A degree 2 Bezier curve is a B-spline curve with knot vector {0,0,0,1,1,1}

  6 = (2+1)+3

- A quadratic Bezier spline consisting of two quadratic Bezier curves with control points P_0,P_1,P_2 and P_2,P_3,P_4 can be viewed as a quadratic B-spline curve with control points P_0, P_1, P_2, P_3, P_4 and knot vector {0,0,0,1,1,2,2,2}.

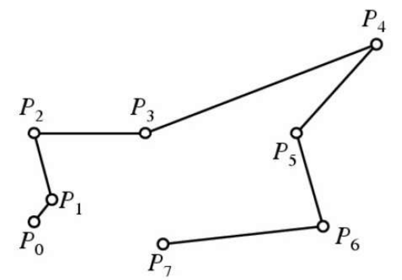
  8 = (2+1)+5

Question for you

A B-spline curve of degree four, $P(t)$, is defined by the control points P_0, P_1, …, P_7 that are shown in Figure Q4(b) and the knot vector [0,0,0,0,0,1,2,4,5,5,5,5,5].

(i) Sketch the convex hull for the curve segment defined on knot span (2,4) according to the strong convex hull property.

(ii) Suggest how to modify the control points to make the curve segment on knot span (2,4) become a straight line segment.



Outline

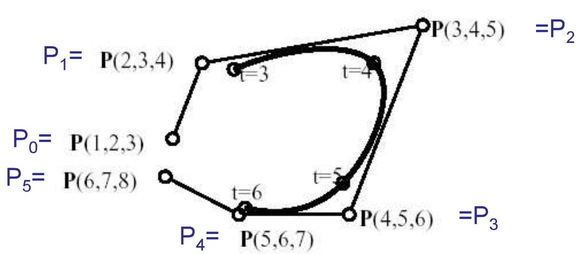
Polar form

- ***Polar form*** is a *labeling* scheme for control points of B-splines, developed by **Dr. L Ramshaw**. Its underlying theory is based on symmetric polynomials and a technique called ***blossoming***.

- In polar form, control points are referred to as *polar values*. Most important algorithms for Bezier and B-spline curves can be derived from the following rules for polar values:

Rule 1

- For a degree k B-spline curve with a knot vector of $\{u_0, u_1, u_2, u_3, \ldots\}$, the arguments of the polar values consist of group of k adjacent knots from the knot vector, with the i^{th} polar value being $P(u_{i+1}, u_{i+2}, \ldots, u_{i+k})$.



knot vector = {0,1,2,3,4,5,6,7,8,9}

Rule 2 & Rule 3

- A polar value is symmetric in its arguments. This means that the order of the arguments can be changed without changing the polar value. For example, P(1,0,0,2) = P(0,1,0,2) = P(2,1,0,0).
- Given $P(u_1, u_2, \ldots, u_{k-1}, a)$ and $P(u_1, u_2, \ldots, u_{k-1}, b)$, we can compute $P(u_1, u_2, \ldots, u_{k-1}, c)$ by linear interpolation, where c is any value:

$$P(u_1,u_2,\ldots,u_{k-1},c)=\frac{b-c}{b-a}P(u_1,u_2,\ldots,u_{k-1},a)+\frac{c-a}{b-a}P(u_1,u_2,\ldots,u_{k-1},b)$$

$P(u_1, u_2, \ldots, u_{k-1}, c)$ is said to be an **affine combination** of $P(u_1, u_2, \ldots, u_{k-1}, a)$ and $P(u_1, u_2, \ldots, u_{k-1}, b)$.

Question for you

Q: Polar values $P(0,1,2)$, $P(1,4,2)$, and $P(2,4,4)$ have coordinates (2,2), (6,6), and (6,0), respectively. Compute the coordinates of polar value $P(2,2,2)$.

Outline

Applications

- How to insert a knot
- How to compute a point on a B-spline curve
- How to extract Bezier curves from B-splines

Common strategies

1) Find the correspondence between the given control points and the polar values based on the initial knot vector
2) Find the new knot vector
3) List the new polar values based on the new knot vector
4) Compute the geometry of the new polar values from the known polar values.

4.1 Knot insertion

Problem: Given a cubic B-spline with control points P0, P1, P2, P3, P4, P5, and knot vector {0,0,0,0,1,3,4,4,4,5}, find the new control point after inserting a new knot of 2.

Solution:

– The initial knot vector is {0,0,0,0,1,3,4,4,4,5}. Thus

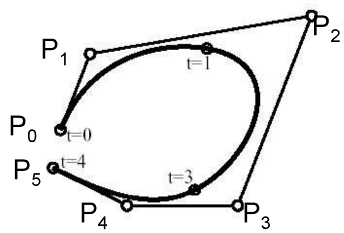
P(0,0,0) = P0, P(0,0,1) = P1,

P(0,1,3) = P2, P(1,3,4) = P3,

P(3,4,4) = P4, P(4,4,4) = P5

– The new knot vector is

{0,0,0,0,1,2,3,4,4,4,5}.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Knot insertion

– The polar values based on the new knot vector are P(0,0,0), P(0,0,1), P(0,1,2), P(1,2,3), P(2,3,4), P(3,4,4), P(4,4,4).

– Compute the polar values:

P(0,0,0) = P0,

P(0,0,1) = P1

P(0,1,2) = (1/3)*P(0,0,1) + (2/3)*P(0,1,3)

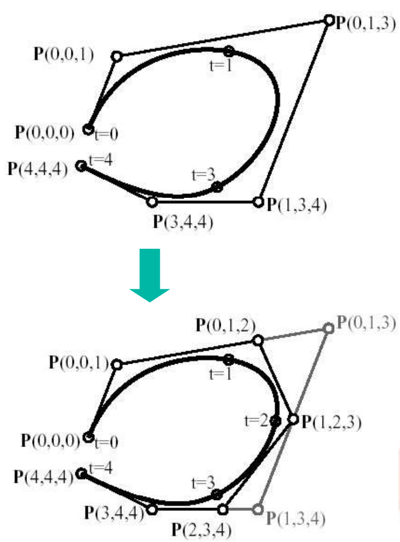
……

P(4,4,4) = P5.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Knot insertion

	Initial	After Knot Insertion
Knot Vector:	0,0,0,0,1,3,4,4,4,5	0,0,0,0,1,2,3,4,4,4,5
Control Points:	P(0,0,0)	P(0,0,0)
	P(0,0,1)	P(0,0,1)
		P(0,1,2)
	P(0,1,3)	
		P(1,2,3)
	P(1,3,4)	
		P(2,3,4)
	P(3,4,4)	P(3,4,4)
	P(4,4,4)	P(4,4,4)



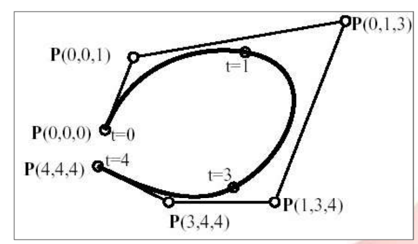
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

4.2 de Boor algorithm

Problem: Given a cubic B-spline with control points P0, P1, P2, P3, P4, P5, and knot vector {0,0,0,0,1,3,4,4,4,4}, find the point on the curve whose parameter value is 2.

Solution:

- The initial knot vector is {0,0,0,0,1,3,4,4,4,5}. Thus
  P(0,0,0) = P0, P(0,0,1) = P1,
  P(0,1,3) = P2, P(1,3,4) = P3,
  P(3,4,4) = P4, P(4,4,4) = P5

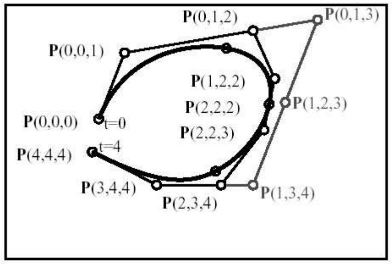


knot vector = {0,0,0,0,1,3,4,4,4,4}

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

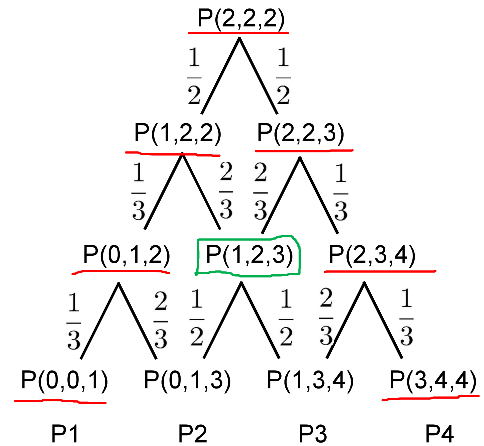
de Boor algorithm

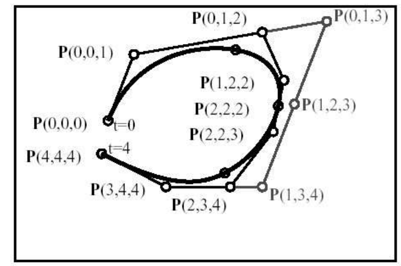
- The new knot vector is {0,0,0,0,1,2,2,2,3,4,4,4,5}.
- The polar values based on the new knot vector are P(0,0,0), P(0,0,1), P(0,1,2), P(1,2,2), P(2,2,2), P(2,2,3), …
- We want to compute P(2,2,2). ……



old knot vector = {0,0,0,0,1,3,4,4,4,4}
new knot vector = {0,0,0,0,1,2,2,2,3,4,4,4,4}

de Boor algorithm

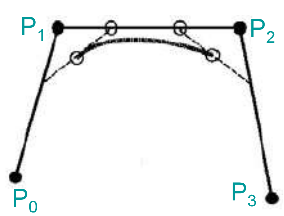




4.3 Extract Bezier from cubic B-splines

Problem: A cubic B-spline curve is defined by de Boor points P_0,P_1,P_2,P_3, and knot vector {-3,-2,-1,0,1,2,3,4}. Convert it into Bezier representation.

Solution:



- P_0 = P(-2,-1,0), P_1 = P(-1,0,1), P_2 = P(0,1,2), P_3 = P(1,2,3)
- Inserting knots of 0 and 1 twice gives the new knot vector {-3,-2,-1,0,0,0,1,1,1,2,3,4}.
- Then we have polar values P(-2,-1,0), P(- 1,0,0), P(0,0,0), P(0,0,1), P(0,1,1), P(1,1,1), P(1,1,2), P(1,2,3).
- Compute the Bezier control points which are just P(0,0,0), P(0,0,1), P(0,1,1) and P(1,1,1).

Extract Bezier from cubic B-splines

P_0 = P(-2,-1,0), P_1 = P(-1,0,1), P_2 = P(0,1,2), P_3 = P(1,2,3)

How to compute P(0,0,0), P(0,0,1), P(0,1,1) and P(1,1,1)?

$$P(0,0,1)=\frac{2-0}{2-(-1)}P(-1,0,1)+\frac{0-(-1)}{2-(-1)}P(0,1,2)=\frac{2}{3}P_1+\frac{1}{3}P_2$$

$$P(0,1,1)=\frac{2-1}{2-(-1)}P(-1,0,1)+\frac{1-(-1)}{2-(-1)}P(0,1,2)=\frac{1}{3}P_1+\frac{2}{3}P_2$$

$$P(-1,0,0)=\frac{1-0}{1-(-2)}P(-2,-1,0)+\frac{0-(-2)}{1-(-2)}P(-1,0,1)=\frac{1}{3}P_0+\frac{2}{3}P_1$$

$$P(1,1,2)=\frac{3-1}{3-0}P(0,1,2)+\frac{1-0}{3-0}P(1,2,3)=\frac{2}{3}P_2+\frac{1}{3}P_3$$

$$P(0,0,0)=\frac{1-0}{1-(-1)}P(-1,0,0)+\frac{0-(-1)}{1-(-1)}P(0,0,1)=\frac{P(-1,0,0)+P(0,0,1)}{2}=\frac{P_0+4P_1+P_2}{6}$$

$$P(1,1,1)=\frac{2-1}{2-0}P(0,1,1)+\frac{1-0}{2-0}P(1,1,2)=\frac{P(0,1,1)+P(1,1,2)}{2}=\frac{P_1+4P_2+P_3}{6}$$

Extract Bezier from degree 2 B-splines

Problem: A degree 2 B-spline curve is defined by de Boor points P_0, P_1, P_2, and knot vector {-2,-1,0,1,2,3}. Convert it into Bezier representation.

Solution:

- $P_0 = P(-1,0)$, $P_1 = P(0,1)$, $P_2 = P(1,2)$
- Inserting knots of 0 and 1 once gives the new knot vector {-2,-1,0,0,1,1,2,3}.
- Then we have polar values P(-1,0), P(0,0), P(0,1), P(1,1), P(1,2).
- Compute the Bezier control points which are just P(0,0), P(0,1), P(1,1).

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Extract Bezier from degree 2 B-splines

$$P(0,0) = \frac{1-0}{1-(-1)} P(-1,0) + \frac{0-(-1)}{1-(-1)} P(0,1) = \frac{1}{2} P_0 + \frac{1}{2} P_1$$

$$P(1,1) = \frac{2-1}{2-0} P(0,1) + \frac{1-0}{2-0} P(1,2) = \frac{1}{2} P_1 + \frac{1}{2} P_2$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Homework

Q1. A cubic B-spline curve $P(t)$ is defined by de Boor points $P_0, P_1, \dots, P_9$ and knot sequence [-1,1,2,4,5,5,8,10,11,12,13,14,16,17].

1) How many curve segments is this B-spline curve composed of?
2) What is the order of continuity of the curve at t=5?
3) Which control points affect $P(6)$?
4) Express $P(5)$ in terms of the de Boor points.
5) Suggest how to modify the knots such that the modified B-spline curve goes through P_3.

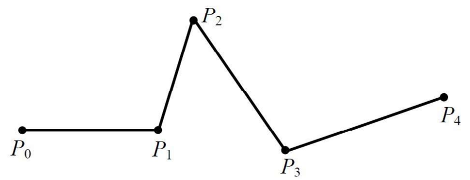
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Homework (cont)

Q2. Polyline $P_0P_1P_2P_3P_4$ shown in the figure serves as the control polygon for the following curves:

1) A Bezier curve;
2) A cubic B-spline curve with knots {0,1,2,3,4,5,6,7,8};
3) A cubic B-spline curve with knots {0,1,2,3,4,5,5,5,8};
4) A quadratic B-spline curve with knots {0,1,2,3,4,5,6,7};
5) A quadratic B-spline curve with knots {0,1,2,3,3,5,6,7}.

Draw these curves with their control polygons.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Summary

- B-spline formulation & basis functions
- B-spline properties
- Using polar form to perform computations on B-splines

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

End

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE