8. Introductory Material
- 8.1 Affine Maps
- 8.2 Curve representation -- Implicit form and Parametric form
- 8.3 Curve drawing

9. Interpolation and Piecewise Interpolation

10. Cubic Spline

11. Conics

12. Bezier Curves

13. B-spline Curves

14. Rational Curves (P75 - P80)

15. Back to Conics (P81 - P84)

16. Curve Manipulation (P85 - P86)

8.2 Curve Representation---Implicit form and Parametric form

Curves can be represented in two forms:

(1) Parametric representation

$x = x(t), \quad y = y(t)$

(2) Implicit representation

$f(x,y) = 0$

How a circle center at the origin with radius r can be represent in each form?

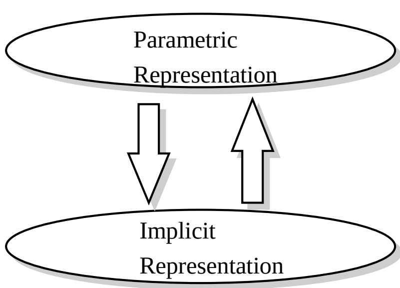
Parametric representation:

Implicit representation:

How a space curve (3D curve) can be represented in these two forms?

Parametric representation:

Implicit representation:



If we compare the parametric representation for curve C(t)=[X(t),Y(t)], to the implicit representation C(x,y)=0, we see advantages and disadvantages for each:

	Parametric	Implicit
Draw		
Transformations		
Is (x,y) on curve		

Standard Parametrization of Conics:

	Implicit	Parametric
Circle		
Ellipse		
Hyperbola		
Parabola		

Finding the parameter value corresponding to a given (x,y,z) coordinate on a parametric curve is termed <u>inversion</u>.

(a) Find the parameter value for the circle

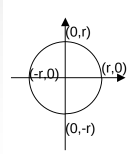
$X(t) = r\,(1-t^2)/(1+t^2),\ Y(t) = 2t/(1+t^2)$

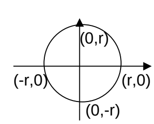
at the point (r,0),(0,r),(-r,0) and (0,-r)?

(b) What the curve will be draw by the following program segment ?

```
for t = -100 , 100   step 1
    draw(X(t),Y(t))
```

(c) Find the parameter value for the circle $X(t) = r\cos(t)$, $Y(t) = r\sin(t)$ at the point (r,0),(0,r),(-r,0) and (0,-r)?



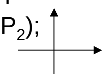


Curve Representation

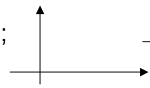
How to represents 2D curves in computer?

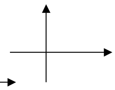
(1) Via the properties of the curve.

E.g. Line(P_1,P_2); Circle(P,r);



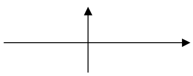
Ellipse(p,a,b,α);





(2) Via a set of data point.

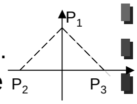
Distantage: waste memory, hard to know the properties of the curve.



(3) Via a set of coefficient value for polynomial function.

(4) Via a set of control points.

E.g. P_0,P_1,P_2 represent the curve



$\sum_{i=0}^{2} C_i^2 t^i(1-t)^{2-i}$

8.3 Curve Drawing

What we should know before drawing a circle (or arc) on screen?

How to draw the circle (or arc) after knowing the information (Assume the circle centered at the origin)?

(1) the parametric equation:

(2) The increase measurement of the angle:

(3) The starting point (x_0,y_0):

(4) The point (x_{i+1},y_{i+1}) after knowing (x_i,y_i) (Assume the circle centered at the origin):

How about the situation that the circle centered at (C_x,C_y)?

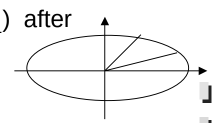
What infomation we should know before drawing ellipse on screen?

How to draw the picture (Assume the axes of ellipse are X-axis and Y-axis)?

(1) The parametric equations:

(2) It is trivial to get the increase measurement of the angle and the starting point.

(3) The point (x_{i+1},y_{i+1}) after knowing (x_i,y_i):



Homework:

How about the situation that the center of the ellipse is at (C_x , C_y) and the major axis of the ellipse makes an angle α with the horizontal ?

What infomation we should know before drawing parabola on screen?

How to draw the picture (Assume the axis of symmetry is parallel to X-axis and the vertex is at the origin.Furthermore,the parabola is open to the right)?

(1) The parametric equations:

(2) The point (x_{i+1},y_{i+1}) after knowing (x_i,y_i):

Homework:How about the apes that the parabola above is at (C_x,C_y) and the axis of symmetry makes an angle α with the horizontal?

What infomation we should know before drawing hyperbola on screen?

How to draw the picture (Assume the trsansverse axis is X-axis and the conjugate axis is Y-axis)?

(1) The parametric equations:

(2) The point (x_{i+1},y_{i+1}) after knowing (x_i,y_i):

$$X_{i+1}=a\sec(\theta+\Delta\theta)=a\cdot 1/\cos(\theta+\Delta\theta)$$

$$=a\frac{1}{\cos\theta\cos(\Delta\theta)-\sin\theta\sin(\Delta\theta)}=\frac{ab/\cos\theta}{b\cos(\Delta\theta)-\tan\theta\sin(\Delta\theta)}$$

$$=\frac{bX_i}{b\cos(\Delta\theta)-Y_i\sin(\Delta\theta)}$$

$$Y_{i+1}=b\tan(\theta+\Delta\theta)=\frac{b(\tan\theta+b\tan\Delta\theta)}{1-\tan\theta\tan\Delta\theta}=\frac{b(Y_i+b\tan(\Delta\theta))}{b-Y_i\tan(\Delta\theta)}$$

No matrix form for this parametric equation.

$$X_{i+1}=a\cosh(\theta+\Delta\theta)=a(\cosh\theta\cosh\Delta\theta+\sinh\theta\sinh\Delta\theta)$$
$$=X_i\cosh\Delta\theta+a/b\, Y_i\sinh\Delta\theta$$
$$Y_{i+1}=b\sinh(\theta+\Delta\theta)=b(\sinh\theta\cosh\Delta\theta+\cosh\theta\sinh\Delta\theta)$$
$$=b/a\, X_i\sinh\Delta\theta+Y_i\cosh\Delta\theta$$

$$[X_{i+1}\ Y_{i+1}\ 1]=[X_i\ Y_i\ 1]\begin{bmatrix}\cosh\Delta\theta & b/a\sinh\Delta\theta & 0\\ a/b\sinh\Delta\theta & \cosh\Delta\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

Curve Tracing:

How to sketch a Curve represented by a polynomial function?

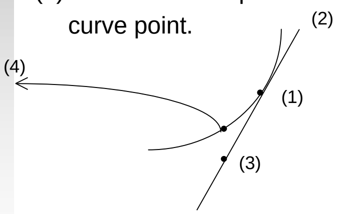
$f(x,y)=\sum_{i,j=0}^{i+j<=n} a_{ij}x^i y^j=0$

(1) Find a initial point p on the curve.

(2) Determine a local approximate at p.

(3) Select a suitable step size and step along the approximant.

(4) Refine the new point estimate to a curve point.



This process is not work for singular point. Consider the graph for $f(x,y)=x^3-y^2=0$ near the origin point. It is hard to find an local approximate at the origin because it is not differentiable at this point. We have to do the curve desingularization.

$f(x,y) = x^3 - y^2 = 0$ $f'(x',y')=0$

After the transformation $x'=x$ and $y'=y/x$, we find a curve f' which is differentiable at the origin. Then, we can trace f' as described before and find new point (x',y'). After that, the new point on f can be calculated by $x=x'$ and $y=x'y'$.

9. Curve Interpolation

Find an arbitrary curve which fits a set of data values is the problem of curve interpolation.

Lagrange Polynomial:

Let (x_i,y_i), where i=0,...,n, are 2D points. We would like to find a curve passing through these points. Let this curve be:

$f_n(x) = y_0L_{0,n}(x) + ... + y_nL_{n,n}(x)$

We hope that $f_n(x_i) = y_i$, that is,

$L_{i,n}(x_i)=1$ and $L_{i,n}(x_j) = 0$ for j <> i

We can define $L_{i,n}(x)$ as:

$L_{i,n}(x) =$ =

Now, $f_n(x)$ can be written as:

$f_n(x) =$

The degree of the polynomial is tied to the number of points used. If the number of the points increase, a higher degree curve will be found. The higher degree curve is not only excessive oscillation, but also numerical sensitivity.

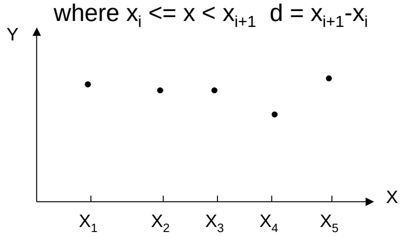
Example: Use a Lagrangian polynomial to interpolate the curve passing through (0,0), (1,2), (2,3),(3,5),(4,1)

Piecewise Linear:

If accuracy is not a major concern, piecewise linear interpolation may be an appropriate solution.

$f(x) = f(x_i)+[f(x_{i+1})-f(x_i)][(x-x_i)/d]$

where $x_i <= x < x_{i+1}$ $d = x_{i+1}-x_i$



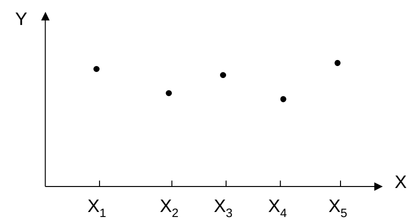
Example: Find the piecewise linear interpolation for the curve passing through the point (-1,1),(0,0) and (1,1).

Piecewise Quadric:

Let (x_i,y_i), where i = 0, ... , 2n and $x_i<x_j$ for i<j, are 2D points. We group three points to do the interpolation, so that n quadric curve will be produced. Consider the curve passing through three points (x_0,y_0), (x_1,y_1), (x_2,y_2).

Let $f(x) = a_2x^2+a_1x+a_0$,

from $y_i=f(x_i)$, i = 0,1,2, we find three equations with three variables (a_2,a_1 and a_0, unique solution in general case), so we find the curve.



Example: Find the curve passing through the points (-1,1),(0,0) and (1,1). Find a circle passing through these three points.

(1) $f(x)=a_2x^2+a_1x+a_0$

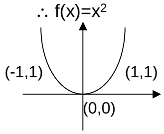
$1=f(-1)=a_2-a_1+a_0$

$0=f(0)=a_0$

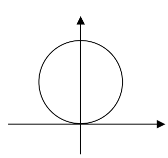
$1=f(1)= a_2+a_1+a_0$

$\Rightarrow \begin{cases} a_2-a_1=1 \\ a_2+a_1=1 \end{cases} \Rightarrow a_0=0,\ a_1=0,\ a_2=1$

$\therefore f(x)=x^2$



(2)The point (0,1) has equal to these three points. So, the circle $x^2+(y-1)^2=1$ passing through(-1,1) (0,0) and (1,1).



Notice that the lagrangian polynomial,piecewise linear and piecewise quadric are the form of $y=f(x)$, such as (1), and not the form of $f(x,y)=0$, such as (2)

Hermit Interpolation

The procedure for defining a cubic curve, using endpoints coordinates and tangent vectors at the endpoints, is one form of Hermit interpolation.

Let this cubic curve be:

$C(t) = a_3t^3+a_2t^2+a_1t+a_0$

We have $C'(t) = 3a_3t^2+2a_2t+a_1$

So $C(0) = a_0$

$C(1) = a_3+ a_2+ a_1+ a_0$

$C'(0) = a_1$

$C'(1) = 3a_3+2a_2+ a_1$

We have 4 equations with 4 variables, and a unique solution can be found in general case.

$a_0= C(0)$

$a_1= C'(0)$

$a_2= -3C(0)+3C(1)-2C'(0)- C'(1)$

$a_3= 2C(0) -2C(1)+ C'(0)+C'(1)$

$C(t) =$

Let $[T] = [t^3, t^2, t, 1]$, algebraic coefficients matrix $[A] = [a_3, a_2, a_1, a_0]^T$, geometric coefficients matrix $[G] = [C(0), C(1), C'(0), C'(1)]^T$ and Hermit matrix

$[M]=$

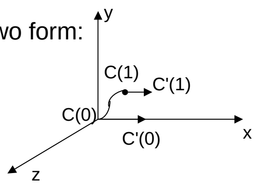
C(t) can be written in two form:

(1) Algebraic form:

$C(t) = [T][A]$

(2) Geometric form:

$C(t) = [T][M][G]$



Example: Find the parametric cubic curve C(t), knowing that:

$C(0)=(0,0,0)$; $C(1)=(2,2,2)$;

$C'(0)=[1,0,0]$; $C'(1)=[1,0,0]$;

Find the matrix of geometric coefficients for a parametric cubic curve, knowing that:

- for t = 0, (2,20,2) is a point on the curve and $C'(0) = (x_1, 0, 4x_1)$.
- for t = 1, (10,20,2) is a point on the curve and $C'(1) = (x_2, 0, -2x_2)$.
- for t = 0.5, (6,20,6) is a point on the curve.

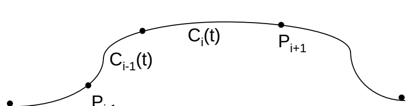
Homework: (Anand P277, Ex4) A parametric cubic curve passes through the points (0,0),(2,4),(4,3),(5,-2), which are parametrized at t = 0,1/4,3/4, and 1, respectively. Determine the geometric coefficient matrix and the slope of the curve when t=0.5.

10. Cubic Spline

From: Computer Graphics and Geometric Modeling for Engineers --- Anand

The term "*spline*" in computer graphics and geometric modeling refers to the general piecewise parametric representation of geometry with a specified level of parametric continuity. The *cubic spline* is represented by a picecewise cubic polynomial with second order derivative continuity at the common joints between segments.

Suppose there are m points $P_0, P_1, \ldots, P_{m-1}$, We want to find the cubic spline curve passing through these points. Let the curve $C_i(t)$ be the curve with P_i, P_{i+1} as endpoints, as the picture show below:



From the property of cubic spline, we have:

$C''_{i-1}(1) = C''_i(0)$

For the cubic polynomial expressed as:

$C_i(t)=a_{3,i}t^3+a_{2,i}t^2+a_{1,i}t+a_{0,i}$

the second derivative is:

So, $C''_{i-1}(1) = C''_i(0)$ implies

Substituting a_3, a_2 values, we have:

Let P_i and P'_i represent $C_i(0), C_i'(0)$ respectively, from the position continuity $P_i(1)=P_{i+1}(0)$, and the first derivate coincides $P'_i(1)=P'_{i+1}(0)$, the equation can be simplify as:

Now, we have m-2 equations (1 equation for each internal joints) with m variables ($P'_0, P'_1, \ldots, P'_{m-1}$), two more constraints must be imposed on the cubic spline. Two commonly used constraints are:

- Known end tangent vectors, P'_0 and P'_{m-1}.
- Second derivatives at the endpoints P_0 and P_{m-1} both made equal to zero; this is referred to as a natrual cubic spline.

In the first case, the system of equations can be represented in matrix form as:

Solution of this matrix equation yields the values of all tangent vectors:

$$\begin{bmatrix} P'_0 \\ P'_1 \\ : \\ : \\ P'_{m-1} \end{bmatrix} =$$

In the second case, consider the second derivative of the curve $C_0(t)$:

$$C''_0(t)=6a_{3,0}t+2a_{2,0}$$

At the point P_0 (t=0), we have:

$$C''_0(0)=2a_{2,0}=0$$

From the value of $a_{2,0}$, we derive the equation:

With the similar process, consider the second derivative of the curve $C_{m-2}(t)$ at the point P_{m-1} (t=1), we derive the equation:

The m equations in m unknowns can be represented in matrix form as:

Example: Consider four 2D points

$P_0=(0,0)$, $P_1=(2,1)$, $P_2=(4,4)$, $P_3=(6,0)$

with given tangent vectors

$P'_0=[1\ 1]$ and $P'_3=[-1\ 1]$

Determine the values of the tangent vectors at P_1 and P_2 needed for a cubic spline interpolation.

Example:Solve the problem in the previous example, using a natural cubic spline. Calculate cubic spline values at t=1/3 and t=2/3 for each spline segment.

Conics

Conics are commonly described in

(1) implicit form:

$a_{20}x^2+a_{02}y^2+a_{11}xy+a_{10}x+a_{01}y+a_{00} = 0$

(2) matrix form:

Conic curves or sections are either central or noncentral. The central forms are those with a center, specifically the ellipse and the hyperbola. (A circle is a special case of an ellipse.) The parabola is the only noncentral conics.

Given specific values a_{ij}, how can we know whether it represents an ellipse, a hyperbola or a parabola (or degenerate forms)?

Consider the matrix form as:

$(1/2)[X][A][X]^T = 0$

Let's consider translation and rotation on this conic.

Translation of the conics :

The matrix form for the conic after the translation is:

$(1/2)[X][Tr][A][Tr]^T[X]^T = (1/2)[X][A'][X]^T = 0$

where the translation matrix

[Tr] =

From $[A']=[Tr][A][Tr]^T$, the elements in [A'] are:*

$a_{20}'=$

$a_{02}'=$

$a_{11}'=$

$a_{10}'=$

$a_{01}'=$

$a_{00}'=$

If the conic is central, the linear terms are eliminated or the center of the conic is translated to the origin.

That is:

$a_{10}' = a_{01}' = 0$

*Let R_i means the i-th row for [Tr], then $[A']=[Tr][A][Tr]^T =[R_i\ A\ R_j^T]$

By solving $a_{10}' = a_{01}' = 0$, we have:

[m n] [ ] =[ ] or [ ] [ ] = [ ]

which may be written as [L] [M] = [Q]

If [L] is singular, the solution for [M] does not exist and the conic is noncentral, i.e. a parabola. Otherwise, a solution for [M] exists and the conic is central.

If det[L] < 0, then the conic is a hyperbola.

If det[L] = 0, then the conic is a parabola.

If det[L] > 0, then the conic is an ellipse.

Example: Determine the type of conic described by:

$2x^2 - 72xy +23y^2 + 140x - 20y + 50 = 0$

Rotation of the conics :

The matrix form for the conic after the rotation is:

$[X][R][A][R]^T[X]^T = [X][A''][X]^T = 0$

where the rotation matrix

[R] = [ ] A = [ ]

From $[A'']=[R][A][R]^T$, the elements in [A''] are:

$a_{20}''=$

$a_{02}''=$

$a_{11}''=$

$a_{10}''=$

$a_{01}''=$

$a_{00}''=$

If the axes of the conic are parallel to the coordinate axes, then the cross-product term $a_{11}xy$ is not present.

What degree of the angle we should rotate so that $a_{11}'' = 0$?

Conics in standard form:

For a central conic the standard form places the center of the conic at the origin with its axes aligned with the coordinate axes.

For a noncentral conic, the standard form is with the axis of symmetry of the parabola coincident with the positive x-axis, with the vertex at the origin and the parabola opening to the right.

If the conic is central, then it is placed in the standard form by a combination of translation and rotation. Translation followed by rotation yields:

$$[X][Tr][R][A][R]^T[Tr]^T[X]^T = [X][A^{3'}][X]^T = 0$$

From $[A^{3'}]=[Tr][R][A][R]^T[Tr]^T$, the elements in [A''] are:

$a_{20}^{3'} =$

$a_{02}^{3'}=$

$a_{11}^{3'} =$

$a_{10}^{3'} =$

$a_{01}^{3'} =$

$a_{00}^{3'} =$

Now, the standard form of the conic can be written as:

$$[X][A^{3'}][X]^T = [x\ y\ 1]\begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & -\kappa \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

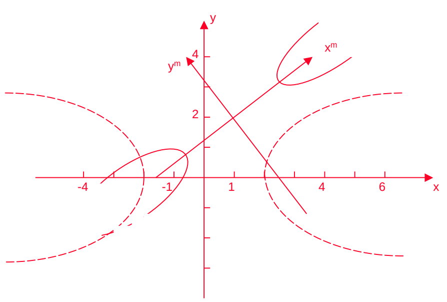
Write out the matrix into implicit form, we have:

Let's systematically investiage the results for various values of α, β, and κ.

κ	α	β	Conics
=	αβ>0		
=	αβ<0		
=	=	=	
=	one of α,β = 0		
>	>	>	
>	<	<	
>	αβ<0		
>	αβ=0		

Example: Transform the following conic into standard form:

$$2x^2 - 72xy + 23y^2 + 140x - 20v + 50 = 0$$



If the conic is noncentral (a parabola), the linear terms cannot both be eliminated. However, one of the linearly dependent terms along with one of the quadratic terms either in x or y can be eliminated.

Let consider the rotation (elimination a_{11}) followed by a translation. That is:

$$[X][R][Tr][A][Tr]^T[R]^T[X]^T = [X][A^{4'}][X]^T = 0$$

From $[A^{4'}]=[R][Tr][A][Tr]^T[R]^T$, the elements in $[A^{4'}]$ are:

$a_{20}^{4'} =$

$a_{02}^{4'}=$

$a_{11}^{4'} = 0$

$a_{10}^{4'} =$

$a_{01}^{4'} =$

$a_{00}^{4'} =$

Here, either $a_{20}^{4'}$ or $a_{02}^{4'}$ will be zero. We try to eliminate one of $a_{10}^{4'}$ and $a_{01}^{4'}$.

$a_{10}^{4'} = 0$ yield m =

$a_{01}^{4'} = 0$ yield n =

What is the relation among $a_{20}^{4'}, a_{02}^{4'}, a_{10}^{4'}$, and $a_{01}^{4'}$?

Assuming that the linear terms in y and the quadratic terms in x are eliminated ($a_{20}^{4'} = a_{01}^{4'} = 0$), the standard form of the parabola can be written as:

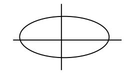
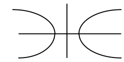
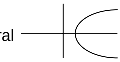
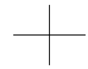
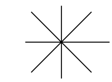
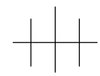
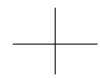
$$[A^{4'}] = \begin{pmatrix} 0 & 0 & \gamma \\ 0 & \beta & 0 \\ \gamma & 0 & -\kappa \end{pmatrix}$$

Write out the matrix into implicit form, we have:

The final step to transform the parabola into standard form is:

Exercise: Given [A], draw a flow chart which identify the types of the conic represented by matrix [A].

Summary of Conic Sections

Name	Equation	Conditions	Type	Sketch
Ellipse	$\alpha x^2+\beta y^2=\kappa$	$\kappa, \alpha, \beta > 0$	Central	
Hyperbola	$\alpha x^2+\beta y^2=\kappa$	$\beta < 0 < \kappa, \alpha$	Central	
Parabola	$\alpha x^2+\beta y^2=0$ $\beta x^2+\alpha y^2=0$		Noncentral	
Empty set	$\alpha x^2+\beta y^2=\kappa$	$\alpha,\beta < 0 < \kappa$	(Central)	(No sketch)
Point	$\alpha x^2+\beta y^2=0$	$\alpha, \beta > 0$	Central	
Pair of lines	$\alpha x^2+\beta y^2=0$	$\beta < 0 < \alpha$	Central	
Parallel lines	$\alpha x^2 = \kappa$	$\alpha , \kappa > 0$	Central	
Empty set	$\alpha x^2 = \kappa$	$\alpha < 0 < \kappa$	(Central)	No sketch
'Repeated' line	$\alpha x^2 =0$		Central	

Translation of the conics (m,n) units

12. Bezier Curve

<u>Bernstein Polynomialss</u>

A Bernstein polynomial is defined by:

$$B_{i,n}(t) = C^n_i t^i (1-t)^{n-i} \quad 0 <= i <= n$$

where n is the degree of the polynomial and $C^n_i = n! / i!(n-i)!$

Honer's method for Bernstein polynimials can be:

$B_{i,n}(t) = (1-t)^n C^n_i u^i \quad u = t/(1-t); \ 0<=t<=1/2$

$B_{i,n}(t) = B_{i,n}(T)$ where $T=1-t; \quad 1/2<=t<=1$

The properties for the Bernstein polynomial are:

(1) Partition of unity: $B_{0,n}(t)+...+B_{n,n}(t)=1$

(2) Recursion: $B_{i,n}(t) = (1-t)B_{i,n-1}(t)+tB_{i-1,n-1}(t)$

(3) Derivative:$(d/dt)B_{i,n}(t)=n[(B_{i-1,n-1}(t)-B_{i,n-1}(t))]$

(4) Linear precision:

(5) Symmetric with respect to t and 1-t:

$B_{i,n}(t) = B_{n-1,n}(1-t)$

(6) Degree elevation formula:

$B_{i,n}(t)$ =

(7) Subdivision:$B^n_i(ct) = \Sigma B^j_n(c)\Sigma^n_j(t)$

(8) Product:

$B_{i,m}(t)B_{j,n}(t)$ =

(9) Integral:

Bezier Curve

Bezier curves employ control points, that is, an ordered set of points $(P_0,P_1,\ldots,P_n)$ that approximate the curve. A Bezier curve of degree n, specified by n+1 control points, is a parametric function of the following form:

$C^n(t)$ =

where the vectors P_i represent the n+1 control points. $B_{i,n}(t)$ is the blending function for the Bezier representation.

The polygon G form by $P_0,P_1,\ldots,P_n$ is called the Bezier polygon or control polygon of the curve C(t). Sometimes we also write $C^n(t) = B[P_0,P_1,\ldots,P_n;t]=B[G;t]$ or, shorter, $C^n=B[P_0,P_1,\ldots,P_n]=BG$.

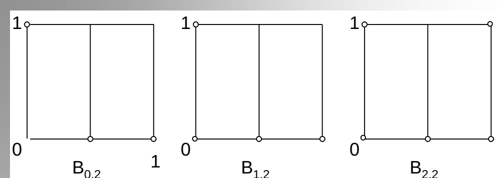
Consider the Bezier curve with control points: $P_i = (i/n, 1)$ for one i;

$P_j = (j/n, 0)$ for all j ¹ i;

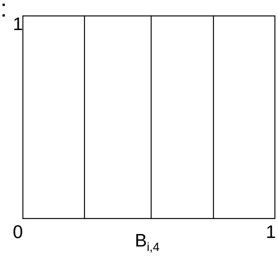
Then

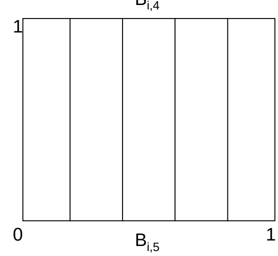
$C^n(t) = \sum_{j=0}^{n}$

So, we can easily find the control points for the curve of Bernstein polynomial $B_{i,n}(t)$.



With the same idea, $B_{i,n}(t)$ where n=4,5 are:





The de Casteljau Algorithm

We give a simple construction for the generation of a parabola; the straightforward generalization will then lead to Bezier curves. Let P_0, P_1, P_2 be any three points in R^2 (or R^3), and let t be a real number. Construct

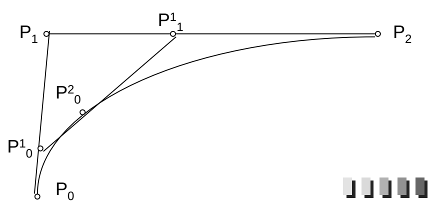
(1) $P^1_0(t) = (1-t)P_0 + tP_1$

(2) $P^1_1(t) = (1-t)P_1 + tP_2$

(3) $P^2_0(t) = (1-t)P^1_0(t) + tP^1_1(t)$

Inserting the first two equations into the third one, we obtain a quadratic expression in t and so $P^2_0(t)$ traces out a parabola as t varies from - to .

The above construction consists of repeated linear interpolation.



This algorithm can be generalized to generate a polynomial curve of arbitrary degree n:

de Casteljau algorithm:

Given: $P_0, P_1, \ldots, P_n$ are points in R^3 and t is a real number.

Set $P^r_i(t) = (1-t)P^{r-1}_i(t) + tP^{r-1}_{i+1}(t)$ where

$r = 1,\ldots,n$ and $i = 0,\ldots,n-r$ and $P^0_i(t) = P_i$.

Then $P^n_0(t)$ is the point with parameter value t on the Bezier curve generate by the control points $P_0, P_1, \ldots, P_n$.

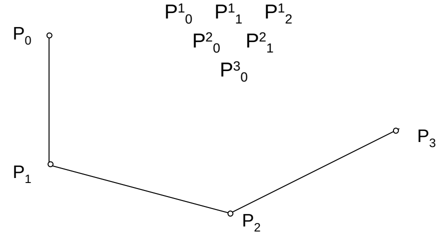
The intermediate coefficients $P^r_i(t)$ are conveniently written into a triangular array of points, the de Casteljau scheme:

$P_0 \quad P_1 \quad P_2 \quad P_3$

$P^1_0 \quad P^1_1 \quad P^1_2$

$P^2_0 \quad P^2_1$

P^3_0



Some properties of Bezier curves

1 Convex hull property: (the curve is inside the control polygon). This follows, since for 0<= t<=1, the Bernstein polynomials are nonnegative, and their sum is equal to one.

Let C and C' are two Bezier curves and G and G' are their control polygons. Do these two curves intersect if G and G' are not intersect? How about G and G' do intersect?

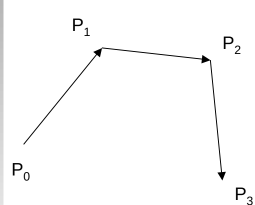
2 Endpoint interpolation: (the curve passes the endpoints of the control points).

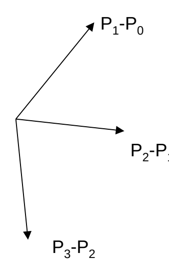
3 Symmetry:($B[P_0, P_1, \ldots, P_n ; t] = B[P_n, \ldots, P_1, P_0 ; 1-t]$)

4 Pseudo-local control: If we move only one of the control polygon vertices, say p_i, then the curve is mostly affected by this change in the region of the curve around the parameter value i/n. This makes the effect of the change reasonably predictable, although the change does affect the whole curve.

5 The derivative of a Bezier curve:

$(d/dt)C^n(t) =$





6. Degree Elevattion

What we should notice if we want construct a piecewise Bezier curve which has C^1 continuity?

What we should do if we want construct a closed curve?

Example: Given two Bezier curves defined by: $B[(2,3,4),(3,1,5)(x,y,z),(3,4,3)]$

$B[(3,4,3),(2,6,0),(5,7,5)(5,2,3)]$

Establish the algebraic conditions that x, y, z must satisfy to ensure C^1 continuity.

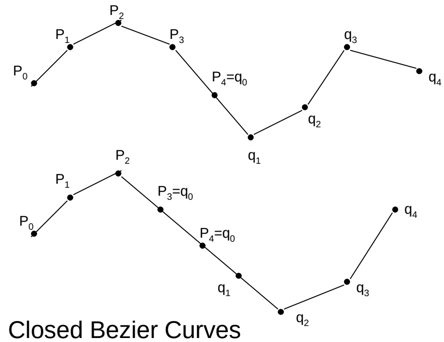
Piecewise Bezier Curve

C^0 continuity: $p_n=q_0$

C^1 continuity: $p_{n-1},p_n=q_0,q_1$ colinear.

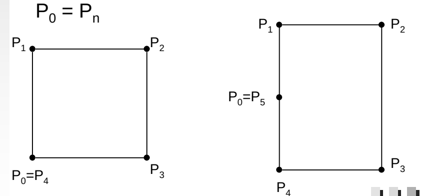
C^2 continuity: $p_{n-2},p_{n-1},p_n=q_0,q_1,q_2$ colinear.

Picture from: Mortenson's book.



Closed Bezier Curves

$P_0 = P_n$



Picture from: Anald's book.

Matrix form for Bezier curve

Bezier curve $B[P_0,P_1,...,P_n]$ can be conveniently expressed in matrix form.

$C^n(t) = B[P_0,P_1,...,P_n]$

$= [B_{0,n}\ B_{1,n} \ldots B_{n,n}]\ [P_0\ P_1 \ldots P_n]^T$

$= [t^n\ t^{n-1} \ldots t\ \ 1]\ [m_{ij}]\ [P_0\ P_1 \ldots P_n]^T$

where m_{ij} =

The cubic Bezier curve can be rewritten in matrix form as:

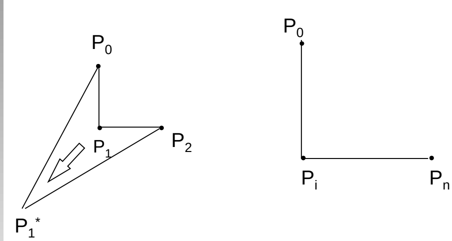
$C^3(t) = B[P_0,P_1,P_2,P_3]$

$= [B_{0,3}\ B_{1,3}\ B_{2,3}\ B_{3,3}]\ [P_0\ P_1\ P_2\ P_3]^T$

$= [(1-t)^3\ 3t(1-t)^2\ 3t^2(1-t)\ t^3][P_0\ P_1\ P_2\ P_3]^T$

$= [t^3\ t^2\ t\ 1] \left[\quad \right] [P_0\ P_1\ P_2\ P_3]^T$

Cubic Bezier Curves



Cubic Bezier curves and their modification:

(a) Moving point p_1 to p_1^* "pulls" the curve toward that vertex.

(b) By specifying multiple coincident points at a vertex, we pull the curve in closer and closer to that vertex.

From: Mortenson's book

13 B-Spline Curves

B-spline curves are similar to Bezier curves in that a set of blending functions combines the effects of n+1 control points P_i given by:

$$C(t) = \sum_{i=0}^{n} N_{i,k}(t)P_i$$

Compare with Bezier curves, the most important difference is the way the blending function $N_{i,k}(t)$ are formulated.

$$N_{i,1}(t) = 1 \quad \text{if } t_i <= t < t_{i+1}$$
$$= 0 \quad \text{otherwise}$$

and

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

where k controls the degree (k-1) of the resulting polynomial in t and thus also controls the continuity of the curve. (What number control the degree of the Bezier curve?) The t_i are called *knot values* and $[t_0\ t_1\ \dots\ t_{n+k}]$ are called *knot vector*. They relate the parametric value t to the P_i control points.

The knot vector $[t_0, t_1, \dots, t_{n+k}]$ can be classified as:

(1) Uniform/periodic

A uniform knot vector has equispaced t_i values, so that $t_i - t_{i-1} = a$ for all intervals, and a is a real number.

e.g.

(2) Nonperiodic

A nonperiodic or open knot vector has repeated knot values at the ends with multiplicity equal to the order of the function k and internal knots equally spaced.

e.g.

(3) Nonuniform

If the repeated knot values at the ends with multipility is not equal to the order of the function k,or the internal knots are not equally spaced, the knot vector is said to be nonuniform.

e.g.

Since the knot vectors influence the shape of the B-spline, it can be said, in general, that B-spline curves have this classification.

Nonperiedic B-spline Curve

For an open curve, the t_i are:(define 0/0=1)

$t_i = 0$ if i < k

$t_i = i-k+1$ if k <= i <= n

$t_i = n-k+2$ if i>n

with 0 <= i <= n+k

The range of the parametric variable t is

0 <= t <= n-k+2

Let's see how these equations compute the blending functions $N_{i,k}$ for k =1,2 and 3.

Given six control points (n=5) and k=1, we find that:

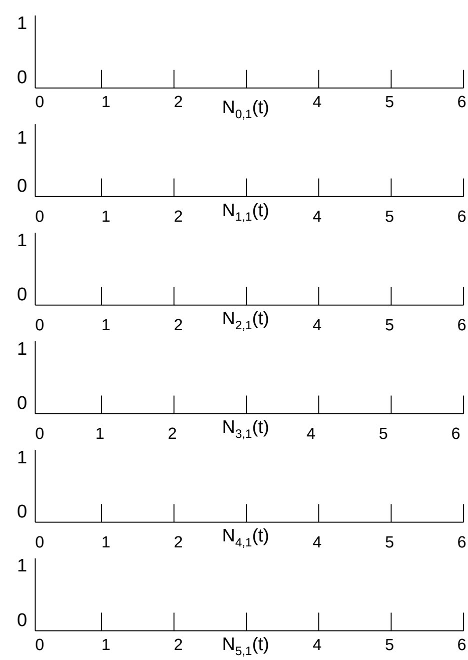
0 £ i £ 6 and 0 £ t £ 6

$[t_0\ t_1\ t_2\ t_3\ t_4\ t_5\ t_6] = [\,0\ 1\ 2\ 3\ 4\ 5\ 6\,]$

$N_{0,1}(t)=$ $N_{1,1}(t)=$

$N_{2,1}(t)=$ $N_{3,1}(t)=$

$N_{4,1}(t)=$ $N_{5,1}(t)=$



If we apply these blending functions to any set of six control points P_i, i=0,...,5, what kind of curve we find?

Next, for the $N_{i,2}(t)$ blending functions with n=5 and k=2, we find that

£ i £ and £ t £

$[t_0\ t_1\ t_2\ t_3\ t_4\ t_5\ t_6\ t_7] = [\qquad\qquad]$

$N_{0,1}(t) =$ $N_{1,1}(t) =$

$N_{2,1}(t) =$ $N_{3,1}(t) =$

$N_{4,1}(t) =$ $N_{5,1}(t) =$

$N_{0,2}(t) = (1-t)N_{1,1}(t)$

$N_{1,2}(t) = tN_{1,1}(t) + (2-t)N_{2,1}(t)$

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

$N_{3,2}(t) = (t-2)N_{3,1}(t) + (4-t)N_{4,1}(t)$

$N_{4,2}(t) = (t-3)N_{4,1}(t) + (5-t)N_{5,1}(t)$

$N_{5,2}(t) = (t-4)N_{5,1}(t)$

If we now apply these blending functions to any set of six control points P_i,i=0,....,5 what kind of curve we find? The curve is C^0, C^1 or C^2 curve?

$C(t) = \Sigma N_{i,k}P_i = (1-t)P_0 + tP_1$ 0 £ t < 1

$(2-t)P_1 + (t-1)P_2$ 1 £ t < 2

$(3-t)P_2 + (t-2)P_3$ 2 £ t < 3

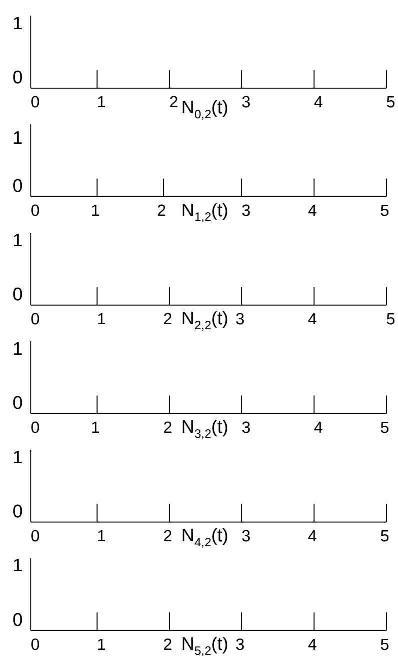
$(4-t)P_3 + (t-3)P_4$ 3 £ t < 4

$(5-t)P_4 + (t-4)P_5$ 4 £ t < 5

It contains line segments connecting

P_0,P_1,P_2,P_3,P_4 and P_5

So, it is C^0 curve.



Finally, for the $N_{i,3}(t)$ blending functions with n=5 and k=3, we find that:

£ i £ and £ t £

$[t_0\ t_1\ t_2\ t_3\ t_4\ t_5\ t_6\ t_7\ t_8] = [\qquad\qquad]$

$N_{0,1}(t) =$ $N_{1,1}(t) =$

$N_{2,1}(t) =$ $N_{3,1}(t) =$

$N_{4,1}(t) =$ $N_{5,1}(t) =$

$N_{0,2}(t) = 0$

$N_{1,2}(t) = (1-t)N_{2,1}(t)$

$N_{2,2}(t) = tN_{2,1}(t) + (2-t)N_{3,1}(t)$

$N_{3,2}(t) = (t-1)N_{3,1}(t) + (3-t)N_{4,1}(t)$

$N_{4,2}(t) = (t-2)N_{4,1}(t) + (4-t)N_{5,1}(t)$

$N_{5,2}(t) = (t-3)N_{5,1}(t)$

$N_{0,3}(t) = (1-t)^2 N_{2,1}(t)$

$N_{1,3}(t) = (1/2)t(4-3t)N_{2,1}(t) + (1/2)(2-t)^2 N_{3,1}(t)$

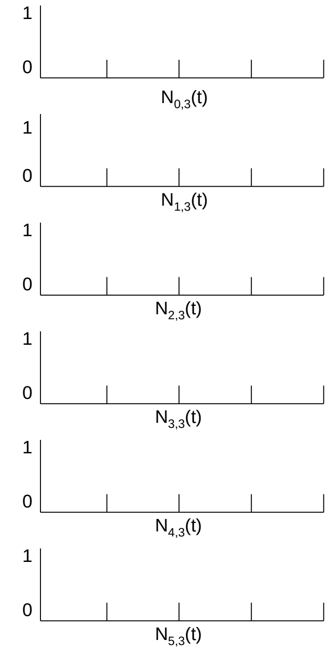
$N_{2,3}(t) = (1/2)t^2 N_{2,1}(t) + (1/2)(-2t^2+6t-3)N_{3,1}(t) + (1/2)(3-t)^2 N_{4,1}(t)$

$N_{3,3}(t) = (1/2)(t-1)^2 N_{3,1}(t) + (1/2)(-2t^2+10t-11) N_{4,1}(t) + (1/2)(4-t)^2 N_{5,1}(t)$

$N_{4,3}(t) = (1/2)(t-2)^2 N_{4,1}(t) + (1/2)(-3t^2+20t-32) N_{5,1}(t)$

$N_{5,3}(t) = (t-3)^2 N_{5,1}(t)$

If we now apply these blending functions to any set of six control points P_i, for i = 0,.....,5 the curve we find is:



A nonperiodic or open knot vector has repeated knot values at the ends with multiplicity equal to the order of the function k and internal knots equal spaced. For example, assuming a control polygon with four vertices:

Order (k)	No. of knots (m+1=n+k+1)	Nonperiodic Knot Vector
2	6	[0 0 1 2 3 3]
3	7	[0 0 0 1 2 2 2]
4	8	[0 0 0 0 1 1 1 1]

From: Anand's book.

Comparison between Bezier and nonperiodic B-spline Curve:(The Bezier representation is a special case of a nonperiodic B-spline, where the number of vertices used equal the order of the curve. The knot vector, in this case, becomes [0 ... 0 1 ... 1] with k 0's and 1's)

1. End point interpolation:

2. Local control of the curve:

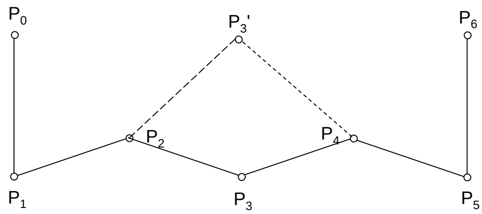
   Each segment of a B-spline curve is influenced by only k control points, and conversely each control point influences only k curve segments.

3. Convex hull property:

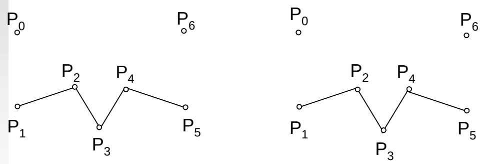
4. Degree of the curves are decided by:

5. Continuity:

Local control for a quadratic (k=3) nonperiodic B-spline curve.



Convex hull property of Bezier curve and strongly convex hull for (nonperiodic) B-spline curve.



Uniform B-spline Curve

For a uniform B-spline curve, $t_i = i$ with $0<= i<=n+k$. The range of the parametric variable t is $(k-1) <= t <= (n+1)$ where k is the order of the curve, n+1 is number of the control points. The number of knots can be calculate as n+k+1.

Notice that the range of the parametric variable t is $(k-1)<=t<=n+1$.

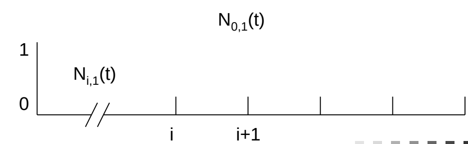
Let's see how these equations compute the blending function $N_{i,k}$ for k=1,2 and 3.

Given six control points (n=5) and k=1, we find that:

$0<=t<=6$ and $t_i=i$ for all i where $0<=i<=6$

$N_{i,1}(t) =$ 1 for $i<=t<i+1$

0 eleswhere



If we apply these blending functions to any set of six control points P_i, i=1,...,6, what kind of curve we find?

Next, for the $N_{i,2}(t)$ blending functions with n=5 and k=2, we find that:

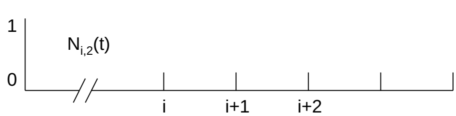
$1<=t<=6$ and $t_i=i$ for all i where $0<=i<=7$

$N_{i,1}(t) =$ 1 for $i<=t<i+1$ $(0<=i<=6)$

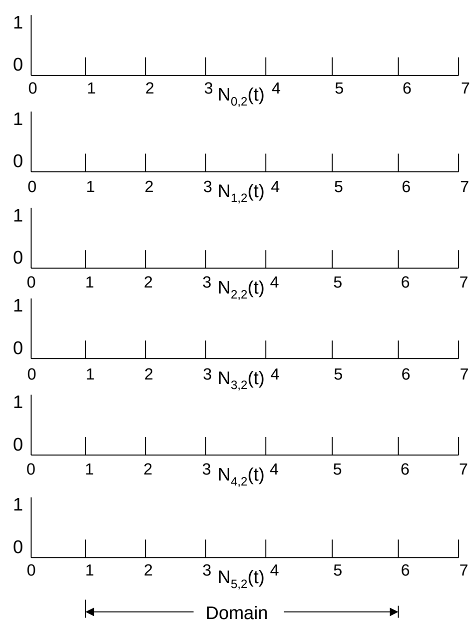
0 eleswhere

$N_{i,2}(t)=(t-t_i)N_{i,1}(t)+(t_{i+2}-t)N_{i+1,1}(t)$

$=(t-i)N_{i,1}(t)+(i+2-t)N_{i+1,1}(t)$ $(0<=i<=5)$



If we now apply these blending functions to any set of six control points P_i, what kind of curve we find? The curve is C^0, C^1 or C^2 curve?



Finally, for the $N_{i,3}(t)$ blending functions with n=5 and k=3, we find that:

$2<=t<=6$ and $t_i=i$ for all i where $0<=i<=8$

$N_{i,1}(t) = 1$ for $i<=t<i+1$ $(0<=i<=7)$

0 eleswhere

$N_{i,2}(t)=(t-t_i)N_{i,1}(t)+(t_{i+2}-t)N_{i+1,1}(t)$ $(0<=i<=6)$

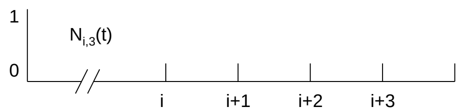
$=(t-i)N_{i,1}(t)+(i+2-t)N_{i+1,1}(t)$

$N_{i,3}(t)=(1/2)(t-t_i)N_{i,2}(t)+(1/2)(t_{i+3}-t)N_{i+1,2}(t)$ $(0<=i<=5)$

$=(1/2)(t-i)^2N_{i,1}(t)$

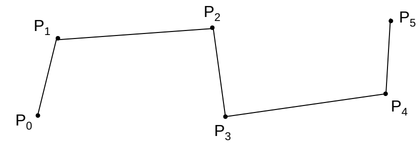
$+(1/2)[-2t^2+(4i+6)t-(2i^2+6i+3)]N_{i+1,1}(t)$

$+(1/2)(i+3-t)^2N_{i+2,1}(t)$

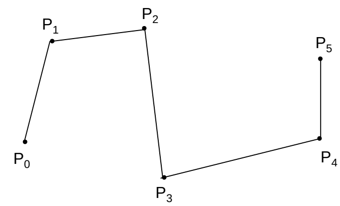


Notice the range of the domain t.

Periodic B-spline and nonperiodic B-spline



Nonperiodic B-spline curve (n=5, k=3)



Period B-spline curve (n=5, k=3; n=5,k=4)

Notice the difference between periodic B-spline and nonperiodic B-spline. Notice also that neither the k=3 curve nor the k=4 curve passes through any of the control points in periodic B-spline.

Picture From Moterson's book.

Uniform quadratic B-splines

Let k=3, we have

$N_{i,1}(t) = 1$ for $i<=t<=i+1$

0 elsewhere

$N_{i,2}(t) = (t-i)N_{i,1}(t)+(i+2-t)N_{i+1,1}(t)$

$N_{i,3}(t) = (1/2)(t-i)^2N_{i,1}(t)$

$+(1/2)[(t-i)(i+2-t)+(3+i-t)(t-i-1)]N_{i+1,1}(t)$

$+(1/2)(i+3-t)^2N_{i+2,1}(t)$

Let C(t) be the uniform quadratic B-spline curve with n+1 control points. That is,

$C(t) = N_{0,3}(t)P_0 + ... + N_{n,3}(t)P_n$

we want to find the expression C(t) for the interval $i+2<=t<i+3$, call it $C_i(t)$

$C_i(t)= (1/2)(i+3-t)^2P_i$

$+ (1/2)[(t-i-1)(i+3-t)+(4+i-t)(t-i-2)]P_{i+1}$

$+ (1/2)(t-i-2)^2P_{i+2}$

There are computational advantages to reparametrizing the interval so that $0<=t<1$ and then identifying the interval by subscripting C(t) as $C_i(t)$ for the *i*th interval.

To reparametrize the abouve equation, replace t by t+i+2, so that

$C_i(t) = (1/2)[(1-t)^2P_i+(-2t^2+2t+1)P_{i+1}+t^2P_{i+2}$

We can easily rewrite the equation into matrix notation:

$$C_i(t) = (1/2)[t^2\ t\ 1] \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} [P_i\ P_{i+1}\ P_{i+2}]^T$$

The analogous form for cubic B-splines (k=4) is:

$C_i(t) = (1/6)TMP$ where

$T = [t^3\ t^2\ t\ 1]$

$$M = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

$P = [P_i\ P_{i+1}\ P_{i+2}\ P_{i+3}]^T$

$0<=t<=1$ and $0<=i<=n-3$ for open curves

Closed Periodic B-spline Curves

Uniform B-splines are well suited to represent closed curves. All that is needed is a change in the number of segments used. We modify the previous equation as:

$C_i(t) = (1/6)TMP'$ where

$$P' = \begin{pmatrix} P_{i \bmod (n+1)} \\ P_{i+1 \bmod (n+1)} \\ P_{i+2 \bmod (n+1)} \\ P_{i+3 \bmod (n+1)} \end{pmatrix}$$

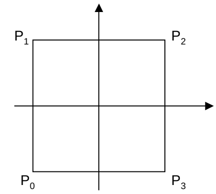
$0 \le t \le 1$ and $0 \le i \le n$ for closed curves

(Notice that the number of curves change from n-2 to n+1)

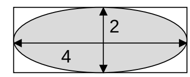
Example: Find the starting and ending locations for a uniform quadratic B-slpine segment.

Example: Use four control points to approxiate a circle by a closed, uniform, quadratic B-spline. Check the error incurred in the approximation at t=0.5 in the first segment.

P_0:(-r,-r) , P_1:(-r,r) , P_2:(r,r) , P_3:(r,-r)



Example: Use a uniform quadratic B-spline curve with four control points to describe an ellipse whose major axis has a length of four units and minor axis two units.



Conversion Between Representations

A freeform cubic curve is described by equation of the form:

x = TMP

where T = [t^3 t^2 t 1], P is the matrix of control points (or geometric coefficients) and M is the basis matrix. Corresponding values of y and z can be similarly found.

To change from one type of representation to another, the equation $x = TM_fP_f = TM_tP_t$ yields $P_t = M_t^{-1}M_fP_f$

Example: Given a cubic Bezier curve represented by the control point P_1(-6,0,0), p_2(-3,4,0),p_3(3,-4,0) and P_4(6,0,0), find:

(a) The control points that would reproduce this curve as a uniform cubic B-spline.

(b) The geometric coefficient matrix that would reproduce this curve as a Hermit.

Rational Curves

The term "rational" means these functions are obtained by the "ratio" of two polynomials. They are invariant under projective transformations. That is, the perspective projection of a rational curve is itself a rational curve, which is not true for the nonrational or integral curves. The rational polynomial functions represent the conics and freeform in one form.

Both Bezier and B-spline curves posses a rational form.

	Bezier	B-spline
Nonrational (Integral)	$Q(t) = \Sigma B_{i,n}(t)V_i$	$P(t) = \Sigma N_{i,n}(t)V_i$
Rational	$Q(t) = \dfrac{\Sigma B_{i,n}(t)w_iV_i}{\Sigma B_{i,n}(t)w_i}$	$P(t) = \dfrac{\Sigma N_{i,n}(t)w_iV_i}{\Sigma N_{i,n}(t)w_i}$

The perspective projection of a nonrational (or integral) curve is not a nonrational (or integral) curve. Why?

Consider the Bezier formulation in 4D homogeneous space, this would result in the expression:

$$Q^w(t) = \Sigma B_{i,n}(t)V^w_i$$

where

$Q^w(t)$: points on the curve in 4D homogeneous space -- coordinates $(w_x(t),w_y(t),w_z(t),w)$

$B_{i,n}(t)$: standard Bezier blending function

V^w_i : control points in 4D homogeneous space

The 3D projection of the 4D control points V^w_i is $V_i = V^w_i/w_i$, where w_i is the weight for the control point V_i. Analogously, the points Q(t) on the curve can be written in rational form as the projection from 4D to 3D space:

$$Q(t) = Q^w(t)/w(t) = (\Sigma B_{i,n}(t)w_iV_i)/(\Sigma B_{i,n}(t)w_i)$$

If $w_i = 1$ for all i, Q(t) is a nonrational curve. In some other case, Q(t) is a rational curve.

If $w_i >= 0$ for all i, the convex hull property for the curve Q(t) are still valid. Q(t) also has end point interpolation property.

If w_{i-1} and w_{i+1} are fixed, an increase in the value of W_i will pull the curve toward V_i.

Rational curves has been gaining popularity in CAD, and today many commerical systems use these representations which include Bezier and all forms of B-splines (uniform/periodic, nonperiodic and nonuniform). The most common scheme, however, appears to be the nonuniform rational B-spline, commonly referred to as NURB, popular because the NURB representation includes all B-splines and Bezier curves. It has the capability of representing a wide range of shapes, including conics, using one cannonical form. (From Anand, "Computer Graphics and Geometric Modeling for Engineerings", 1993)

Bezier curves are said to lack local control.

Nonperiodic B-spline exhibiting local control.

In rational curve, the increase in the value of w_i will pull the curve toward V_i.

Picture from Anand's book.

Nonperiodic cubic rational B-spline and nonuniform cubic rational B-spline.

Picture from Anand's book.

(a) (b)

(c)

(a) A nonrational curve with a change in one control point.

(b) A nonrational curve with a change in one weight.

(c) A rational B-spline curve whose weight of the indicated control point is changed. The curve is only affected locally.

Picture from Farin's book.

Conics as Rational Bezier

A conic section in R^2 is the projection of a parabola in R^3 into a plane.

Theorem: Let Q(t) in R^2 be a point on a conic. Then there exist numbers w_0, w_1, w_2 in R^2 and points b_0, b_1, b_2 in R^2 such that

$$Q(t) = \frac{w_0 b_0 B_{0,2}(t) + w_1 b_1 B_{1,2}(t) + w_2 b_2 B_{2,2}(t)}{w_0 B_{0,2}(t) + w_1 B_{1,2}(t) + w_2 B_{2,2}(t)}$$

Proof:

Gerald Farin, "Curves and Surface for Computer Aided Geometric Design", p179, Academic Press, 1988.

We call the points b_i the control polygon of the conic Q; the number w_i are called weights of the corresponding control polygon vertices. Thus the conic control polygon is the projection of the control polygon with vertices $[w_i b_i \; w_i]$, which is the control polygon of the 3D parabola that we projected onto C.

Conic sections: in the two shown examples, $w_0=w_1=1$. As w_1 becomes larger, the conic is "pulled" towards b_1.

Picture from Farin's book.

Conics as Rational B-spline

What value of k and n of rational B-spline are best suited to represent the conics?

Defining the quadratic rational B-spline by three control points, with 0 <= t <= 1 and a knot vector [t] = [0 0 0 1 1 1], yields:

$$P(t) = \frac{w_0 b_0 N_{0,3}(t) + w_1 b_1 N_{1,3}(t) + w_2 b_2 N_{2,3}(t)}{w_0 N_{0,3}(t) + w_1 N_{1,3}(t) + w_2 N_{2,3}(t)}$$

This equation defines a family of conics, with each conic passing through b_0 and b_2, and tangent to the line segment from b_0 to b_1 and from b_1 to b_2.

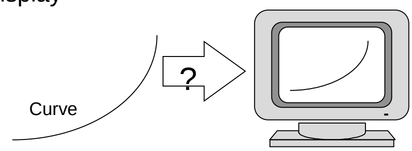
The value of w_1 determines what conic type will be obtained, such that:

$w_1 = 0$	line segment
$0 < w_1 < 1$	elliptic segment
$w_1 = 1$	parabolic segment
$w_1 > 1$	hyperbolic segment

Picture from Anand's book.

Curve Manipulations

1. Display



2. Transformation
3. Evaluating points on curves

4. Segmentation

Segmentation or curve splitting is defined as replacing one existing curve by one or more curve segments of the same curve type such that the shape of the composite curve is identical to that of the original curve.

5.57 5.58

Segment of a circle for modeling purposes. Reparametrization of a segmented curve.

5. Trimming

Trimming is mathematically identical to segmentation. The only difference between the two is that the result of trimming a curve is only one segment of the curve bounded by the trimming boundaries.

Trimming can truncate or extend a curve.

5-59

(a) Truncated curve (b) Extended curve

6. Blending

The blending problem can be stated as: Given two curve segments, find the conditions for the two segments to be continuous at the joint.

7. Offset curve (2D) and surface

Offset curve of a specified curve f is the curve which has equal distance to f.

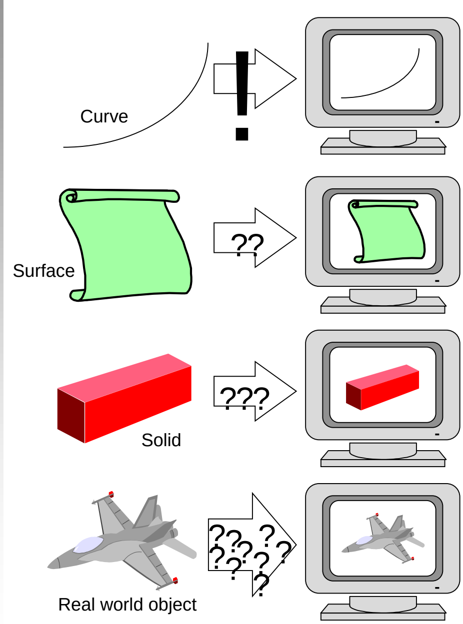
8. Voronoi curve (2D)

In 2D, Voronoi curve is the curve which has equal distance to two or more other curves.

9. Curve/Curve intersection

From Zeid's book.

Computer Aided Geometric Design



CAGD

From: A survey of curve and surface methods in CAGD
--- Bohm, Farin and Kahmann

CAGD - short for Computer-Aided Geometric Design - is concerned with the approximation and representation of curves and surfaces that arise when these objects have to processed by a computer.

Designing curves and surfaces plays an important role in the construction of quite different products such as car bodies, ship hulls, airplane fuselages and wings, propeller blades, shoe insoles, bottles, etc, etc, but also in the description of geological, physical and even medical phenomena.

Before the advent of computers, these design problems were dealt with by means descripitive geometry. A surface was defined by a set of curves, usually plane sections plus some characteristic feature lines. This information was sufficient to manufacture templates, and the templates were used to produce (wooden) master models. The stamps and dies were obtained from the master models by means of copymilling.

In the late fifties, it became possible to drive these milling machines by "numerical control". i.e. the machining instructions could be generated by a computer program. In order to fully exploit this capability, it was necessary to store the surface definition in a computer-compatible form. The problem thus arose how to translate existing surface definitions into a "computerized" format. i.e. how to design a "mathematic model".