Introduction

In vector graphics, curves are used to model smooth curves that can be scaled indefinitely. "Paths," as they are commonly referred to in image manipulation programs.

Parametric Curves

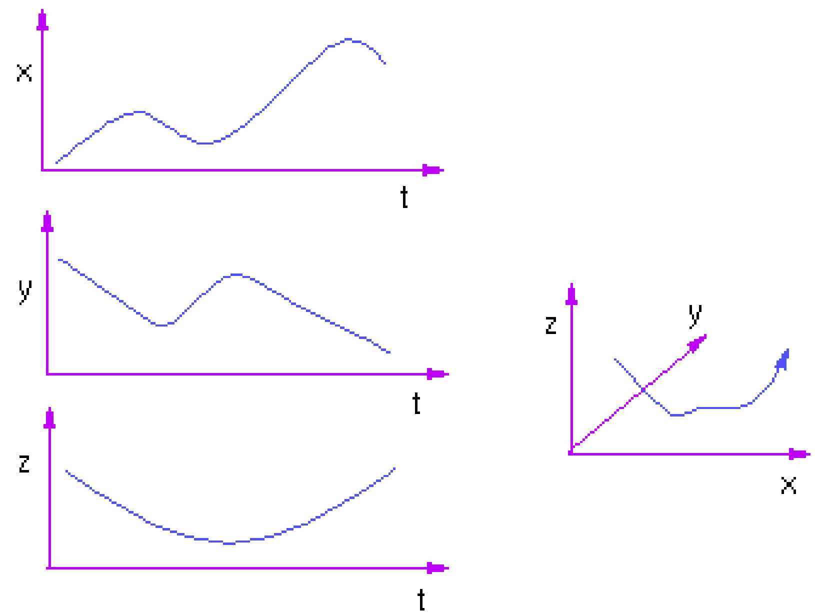
Parametric Cubic Curves

Cubic curves are commonly used in graphics because curves of lower order commonly have too little flexibility, while curves of higher order are usually considered unnecessarily complex and make it easy to introduce undesired wiggles.

A parametric cubic curve in 3D is defined by:

$$x(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$
$$y(t) = b_3t^3 + b_2t^2 + b_1t + b_0$$
$$z(t) = c_3t^3 + c_2t^2 + c_1t + c_0$$

Usually, we consider t = [0...1].



A compact version of the parametric equations can be written as follows:

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$$x(t) = T \cdot A$$

Similarly, we can write

y(t) = T B

z(t) = T C

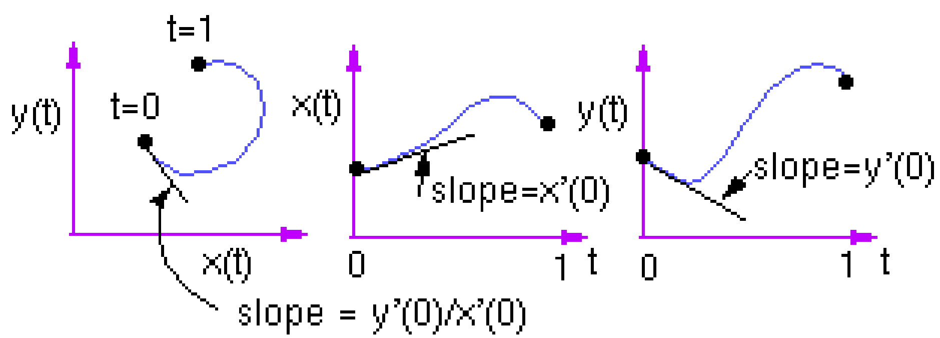
Each dimension is treated independently, so we can deal with curves in any number of dimensions.

The derivatives of the

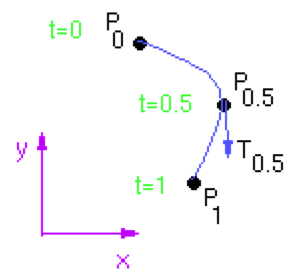
curve with respect to t can be expressed as follows:

$x'(t) = [3t^2 \;\; 2t \;\; 1 \;\; 0]\, A$

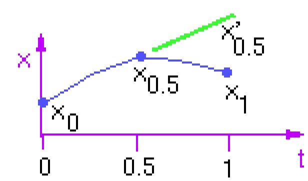
It is often convenient to think of the parameter t as being time in order to visualize some of the properties of the curve. The derivatives also have an intuitive interpretation in the cartesian space of the curve:



Suppose we wish to define a cubic curve such that the user specifies the position of two endpoints and a midpoint, as well as the tangent at the midpoint. The following figure illustrates some sample curves.



We'll first construct the cubic curve for x(t). There are four constraints that are to be met and four unknowns:



$$x(0) = [\,0\ 0\ 0\ 1\,]\ A$$

$$x(0.5) = [\,0.5^3\ \ 0.5^2\ \ 0.5^1\ \ 0.5^0\,]\ A$$

$$x'(0.5) = [\,3(0.5^2)\ \ 2(0.5)\ \ 1\ \ 0\,]\ A$$

$$x(1) = [\,1\ 1\ 1\ 1\,]\ A$$

or $G_X = B\,A$ where

$$G_X = \begin{bmatrix} x_0 \\ x_{0.5} \\ x'_{0.5} \\ x_1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.125 & 0.25 & 0.5 & 1 \\ 0.75 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We can solve for A using $\mathbf{A} = \mathbf{B}^{-1}\mathbf{x}$. The final equation for x is thus:

$$x(t) = T\,B^{-1}\,G_x$$

The matrix B^{-1} is often called the *basis matrix*, which we shall denote by *M*. We can thus write

$$x(t) = T\,M\,G_x$$

In this case, we have

$$M = \begin{bmatrix} -4 & 0 & -4 & 4 \\ 8 & -4 & 6 & -4 \\ -5 & 4 & -2 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Lastly, we can also write

$x(t) = [\ f1(t)\ f2(t)\ f3(t)\ f4(t)\ ]\ G_x$

where f1(t) ... f4(t) are the functions obtained from the product T M. These are functions are called the *blending* or *basis functions*, because they serve to give a weighting to the various components of the geometry vector, G. In this case, these are

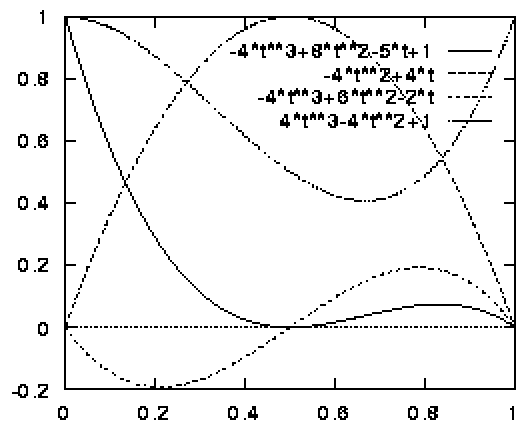
f1(t)= -4t^3 + 8t^2 - 5t + 1

f2(t)= -4t^2 + 4t

f3(t)= -4t^3 + 6t^2 - 2t

f4(t)= 4t^3 - 4t^2 + 1,

where f1(t) is the weighting function for P0, f2(t) is the weighting function for P0.5, f3(t) is the weighting function for T0.5, and f4(t) is the weighting function for P1. These basis functions look as follows:



The curves for y(t) and z(t) are contructed in an analogous fashion to that for x(t). The basis matrix and basis functions thus remain the same. Only the geometry vector changes. The geometry vector for y(t) gives the y components of P0, P0.5, T0.5, and P1. In general, we can write the curve as a single vector equation

P(t) = T M G

which encompasses the following three equations:

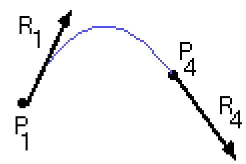
$x(t) = T\ M\ G_x$

$y(t) = T\ M\ G_y$

$z(t) = T\ M\ G_z$

Hermite Curves

As a second example, let's look at Hermite curves. Hermite curves are defined by two points and two tangent vectors.



Let's derive the equation for Hermite curves using the following geometry vector:

For Hermite curves, we impose the following constraints through the geometry matrix:

$$G_H = \begin{bmatrix} P_1 \\ P_2 \\ R_1 \\ R_4 \end{bmatrix}$$

where G_H is Hermite Geometry vector.

which indicates that the four columns (constraints) will be specified by two endpoints (P_1 and P_4) and two tangent vectors (R_1 and R_4). The x components of these constraints, can be expressed as the row vector:

$$G_{Hx} = \begin{bmatrix} P_1 \\ P_2 \\ R_1 \\ R_4 \end{bmatrix}_x$$

As before, we'll express the curve as:

$x(t) = T\ A_h$

$= T\ M_h\ G_h$

The constraints we'll use to define the curve are:

$x(0) = P1 = [\ 0\ 0\ 0\ 1\ ]\ A_h$

$x(1) = P4 = [\ 1\ 1\ 1\ 1\ ]\ A_h$

$x'(0) = R1 = [\ 0\ 0\ 1\ 0\ ]\ A_h$

$x'(1) = R4 = [\ 3\ 2\ 1\ 0\ ]\ A_h$

Writing these constraints in matrix form gives:

$G_h = B_h * A_h$

$A_h = (B_h)^{-1} * G_h$

$x(t) = T\ A_h$

$= T\ (B_h)^{-1} * G_h$

$= T\ M_h * G_h$

$x(t)=T.M_h.G_{Hx}$
so, we can also write
$y(t)=T.M_h.G_{Hy}$
$z(t)=T.M_h.G_{Hz}$

The inverse of $\mathbf{B_h}$ is thus defined as the basis matrix for the hermite curve.

$$M_h = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

solving these equation

$Q(t)=[\,x(t) \quad y(t) \quad z(t)] = T.M_H.G_h$

$$Q(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} * \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} P_1 \\ P_2 \\ R_1 \\ R_4 \end{bmatrix}$$

$$Q(t)=(2t^3-3t^2+1)P_1+(-2t^3+3t^2)P_4+(t^3-2t^2+t)R_1+(t^3-t^2)R_4$$

As before, the basis functions are the weighting factors for the terms in the geometry vector, and are given by the product T M_h. Thus, for basis functions for Hermite curves are

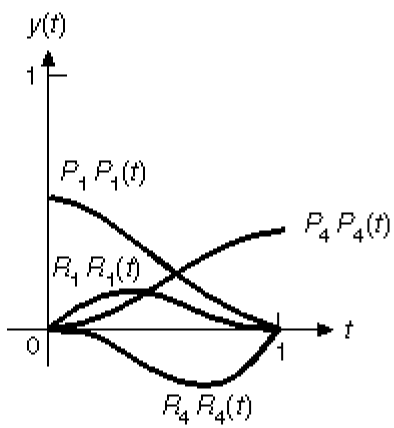
f1(t) = 2t^3 - 3t^2 + 1

f2(t) = -2t^3 + 3t^2

f3(t) = t^3 - 2t^2 + t

f4(t) = t^3 - t^2

These basis functions look as follows:



Application

Various curve functions are useful in object modeling, animation path specifications, data and function graphing, and other graphics applications. Commonly encountered curves include conics, trigonometric and exponential functions,probability distributions, general polynomials, and spline functions.

Bézier curves are also used in animation as a tool to control motion.

Bézier curves are also used in the time domain, particularly in animation and interface design,