

Written by Robert Dunlop
Microsoft DirectX MVP



[Demo w/ Source Code for this article](#)

Introduction

Splines are a mathematical means of representing a curve, by specifying a series of points at intervals along the curve and defining a function that allows additional points within an interval to be calculated. There are various functions available to approximate a curve, but in this article we will focus on a variety of spline known as the Catmull-Rom spline.

Catmull-Rom Splines

The points that define a spline are known as "Control Points". One of the features of the Catmull-Rom spline is that the specified curve will pass through all of the control points - this is not true of all types of splines.

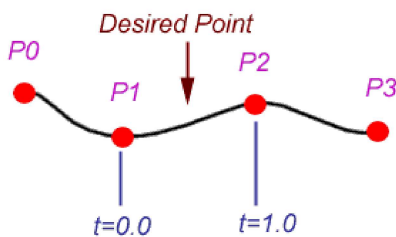


Figure 1

To calculate a point on the curve, two points on either side of the desired point are required, as shown on the left. The point is specified by a value t that signifies the portion of the distance between the two nearest control points.

Given the control points P_0 , P_1 , P_2 , and P_3 , and the value t , the location of the point can be calculated as (assuming uniform spacing of control points):

$$q(t) = 0.5 * (1.0f, t, t^2, t^3) * \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Equation 1

To put that another way:

$$q(t) = 0.5 * ((2 * P_1) + (-P_0 + P_2) * t + (2 * P_0 - 5 * P_1 + 4 * P_2 - P_3) * t^2 + (-P_0 + 3 * P_1 - 3 * P_2 + P_3) * t^3)$$

Equation 2

This formula gives Catmull-Rom spline the following characteristics:

- The spline passes through all of the control points. (see Fig. 1)
- The spline is C^1 continuous, meaning that there are no discontinuities in the tangent direction and magnitude. (See Fig. 2)
- The spline is not C^2 continuous. The second derivative is linearly interpolated within each segment, causing the curvature to vary linearly over the length of the segment.
- Points on a segment may lie outside of the domain of $P_1 \rightarrow P_2$. (See Fig. 2)

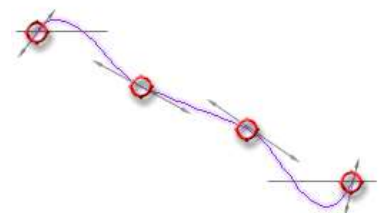


Figure 2

While a spline segment is defined using four control points, a spline may have any number of additional control points. This results in a continuous chain of segments, each defined by the two control points that form the endpoints of the segments, plus an additional control

point on either side of the endpoints. Thus for a given segment with endpoints P_n and P_{n+1} , the segment would be calculated using $[P_{n-1}, P_n, P_{n+1}, P_{n+2}]$.

Because a segment requires control points to the outside of the segment endpoints, the segments at the extreme ends of the spline cannot be calculated. Thus, for a spline with control points 1 through N, the minimum segment that can be formulated is $P_1 \leftrightarrow P_2$, and the maximum segment is $P_{N-3} \leftrightarrow P_{N-2}$. Thus, to define S segments, S+3 control points are required.

Support of Catmull-Rom Splines in DirectX

The D3DX math libraries provide functions for calculating a point on a Catmull-Rom segment, using 2D, 3D, or 4D vectors. These functions are:

```
D3DXVec2CatmullRom(D3DXVECTOR2* pOut,
    CONST D3DXVECTOR2* pV1,
    CONST D3DXVECTOR2* pV2,
    CONST D3DXVECTOR2* pV3,
    CONST D3DXVECTOR2* pV4,
    FLOAT s)
```

```
D3DXVec3CatmullRom(D3DXVECTOR3* pOut,
    CONST D3DXVECTOR3* pV1,
    CONST D3DXVECTOR3* pV2,
    CONST D3DXVECTOR3* pV3,
    CONST D3DXVECTOR3* pV4,
    FLOAT s)
```

```
D3DXVec4CatmullRom(D3DXVECTOR4* pOut,
    CONST D3DXVECTOR4* pV1,
    CONST D3DXVECTOR4* pV2,
    CONST D3DXVECTOR4* pV3,
    CONST D3DXVECTOR4* pV4,
    FLOAT s)
```

Each of these functions accepts four control points and a positional weight from 0.0 to 1.0, and returns an interpolated vector. The returned value will fall in the range of *pV2 (when s==0.0) to *pV3 (when s==1.0).

Sample: Path Interpolation using Catmull-Rom Splines

The downloadable demo available below demonstrates the use of a Catmull-Rom spline for path interpolation of a vehicle traveling down an infinite highway. Use the up/down arrow keys to control rate of travel. You can toggle between Catmull-Rom based interpolation and linear interpolation using the buttons on the right side, to see the difference that spline interpolation makes in providing a smooth animation path.

Source code, pre-compiled executable and media included.



[Download Demo](#)
[lightmap.zip](#)
[347KB](#)

Requires DirectX 8.0 Runtimes or Later

This site, created by DirectX MVP Robert Dunlop and aided by the work of other volunteers, provides a free on-line resource for DirectX programmers.

Special thanks to www.mvps.org, for providing a permanent home for this site.



Visitors Since 1/1/2000: Hit Counter
 Last updated: 07/26/05.