MA 323 Geometric Modelling
Course Notes: Day 08
Cubic Curves and Cubic Hermite Interpolation

David L. Finn

December 10th, 2004

Yesterday, we considered parabolic curves and solved most of the motivating problems. Today, we want to extend these methods to create cubic curves. The only method we will not directly mention is de Casteljau's algorithm (we'll say more about that later), but there is an alternate though slightly different method called cubic Hermite interpolation.

8.1 Cubic Curves

We can generalize some of the constructions in the previous section to higher order curves. One problem with using parabolas as the basic curve element for modelling is that a parabola is a convex curve, convexity being defined as the curve lying on one one side of its tangent lines. This means that in order to model a curve that is not convex like S-curve one must use piecewise parabolic curves. By increasing the order of the polynomial curves that are used one can create a simple model of non-convex curve with one curve. The simplest polynomial curve that is not generically convex is a cubic curve. In fact, one of the reasons that cubic curves are heavily used in geometric modelling is because cubic curves are the lowest degree polynomial curves that are not generically convex.

Another advantage of cubic curves is that they can be joined in a C^2 manner meaning that the first and second derivatives at the point where they are joined agree, while quadratic polynomial curves (parabolic arcs) can not. This extra smoothness factor allows one to create a vast variety of smooth curves by joining simple polynomial curves. This smoothness allows one to model physical phenomenon very easily with cubic curves. The position, velocity and acceleration at one point completely determines a parabolic trajectory. This is a direct result of motion under constant acceleration and Taylor polynomials. This means that for motion modelling one can not model arbitrary changes in acceleration with parabolic arcs. Therefore, quadratic curves are not suitable for modelling motion based on the laws of physics, where motion is controlled through changing the acceleration of the particle. However, one can use cubic arcs. Given a position, velocity and acceleration at time $t = 0$ then a new acceleration at $t = 1$, one can construct a cubic curve that satisfies these properties (see the subsection on Piecewise Cubic Curves).

8.2 Interpolating Cubic Curves

A cubic curve $c(t) = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3$ is naturally defined by four points. We first consider the interpolation problem. Given four data points and four parameter values, we proceed exactly as we did for parabolic arcs. In this section, we will only use the matrix formulation in this section. We defer the basis function formulation until we cover Lagrange interpolation for the general case of higher order polynomials later in this chapter, though we write the answer explicitly. It is worth noting that as for the data fitting problem the parameter values for the interpolation problem may be chosen using equally spaced values, chord length values, or centripetal spaced values.

The interpolation problem for cubic curves is to find a curve $c(t) = a_0 + a_1\,t + a_2\,t^2 + a_3\,t^3$ that satisfies $c(t_i) = p_i$ where $i = 0, 1, 2, 3$ for given four points p_i and parameter values t_i. In the matrix formulation, we need to solve the system of equations

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}.$$

This is a matrix equation $P = T\,A$ where P is the matrix obtained by using the data points p_i as row vectors, A is the matrix of coefficient vectors, and T is the matrix of powers of the parameter values. The solution written out in long form is then obtained as

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \end{bmatrix}^{-1} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}.$$

or in short hand $A = T^{-1}\,P$. Provided that all the parameter values are different, the matrix T is invertible. The cubic curve can thus be represented as

$$\begin{aligned} c(t) &= \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ t_0 & t_1 & t_2 & t_3 \\ t_0^2 & t_1^2 & t_2^2 & t_3^2 \\ t_0^3 & t_1^3 & t_2^3 & t_3^3 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix} \\ &= \frac{(t-t_1)(t-t_2)(t-t_3)}{(t_0-t_1)(t_0-t_2)(t_0-t_3)}\,p_0 + \frac{(t-t_0)(t-t_2)(t-t_3)}{(t_1-t_0)(t_1-t_2)(t_1-t_3)}\,p_1 + \\ &\quad \frac{(t-t_0)(t-t_1)(t-t_3)}{(t_2-t_0)(t_2-t_1)(t_2-t_3)}\,p_2 + \frac{(t-t_0)(t-t_1)(t-t_2)}{(t_3-t_0)(t_3-t_1)(t_3-t_1)}\,p_3. \end{aligned}$$

Notice that the only changes from the quadratic case is the addition of an extra column to the P matrix and the A matrix, and filling out the T matrix. The answer as expressed is the basis function formulation.

Likewise one may construct the basis function representation of $c(t)$ by determining the cubic functions $L_i^3(t)$ such that

$$L_i^3(t_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

to write the cubic function as

$$c(t) = L_0^3(t)\,p_0 + L_1^3(t)\,p_1 + L_2^3(t)\,p_2 + L_3^3\,p_3.$$

The advantage of the basis function approach is that it avoids solving the system directly.

8.3 Hermite Interpolation

We now consider a different type of interpolation problem. In this problem, one prescribes for data, a sequence of points and tangent vectors at these points. This can also be considered a method for modelling motion, as the tangent vectors can be viewed as controls as prescribing the velocity of the particle at the given points

This form of interpolation is called Hermite interpolation, after the French mathematician Charles Hermite. The cubic Hermite interpolation problem is to find a cubic curve that passes through two points with given tangent vectors at these points. Geometrically, one may view the input as four points, the initial and terminal points of each tangent vector, as the initial point of the tangent vector is the point at which the vector is tangent. Viewing the input of a Hermite interpolation in this form is an important observation for a designer. In fact, Adobe Illustrator uses this type of viewpoint to draw curves with its Pen Tool. You place the first point, then drag to the next point (this forms a tangent vector) at the first point. The second point and tangent vector are placed in the same manner. The curve is actually created interactively as the user places the second point and drags to create the tangent vector. For the interaction, the program starts with a tangent vector equal to zero. Thus creating the first curve without waiting for the user to drag to create a nonzero tangent vector.

Cubic Hermite interpolation is therefore a more natural design tool than cubic interpolation where the designer chooses four point that curve must pass through. In Hermite interpolation, the designer chooses only the endpoints of a cubic segment and then interactively shapes the curve as desired. This allows very easy interaction, and is the reason for the use in Adobe Illustrator. The only problem is that it may take some time getting accustomed to the interaction, and becoming proficient in control the curve through specifying the tangent vectors at the end points.

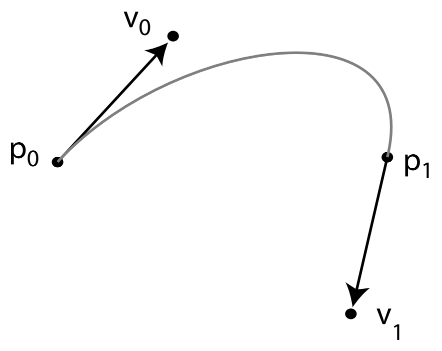

Figure 1: Construction of Curve by Hermite Interpolation

To determine the cubic curve, defined by $c(0) = p_0$, $c'(0) = v_0$, $c(1) = p_1$, $c'(1) = v_1$, we set up the necessary equations and solve. The equations are

$$\begin{aligned} p_0 &= a_0 \\ v_0 &= a_1 \\ p_1 &= a_0 + a_1 + a_2 + a_3 \\ v_1 &= a_1 + 2a_2 + 3a_3 \end{aligned}$$

Solving these equations, we find

$$\begin{aligned}
a_0 &= p_0 \\
a_1 &= v_0 \\
a_2 &= v_1 + 3(p_1 - p_0) - 2(v_0 + v_1) \\
a_3 &= v_0 + v_1 - 2(p_1 - p_0)
\end{aligned}$$

This yields with some rearranging the polynomial

$$c(t) = (1 - 3t^2 + 2t^3)\, p_0 + (t - 2t^2 + t^3)\, v_0 + (-t^2 + t^3)\, v_1 + (3t^2 - 2t^3)\, p_1$$

The functions $H_0^3(t) = (1 - 3t^2 + 2t^3)$, $H_1^3(t) = (t - 2t^2 + t^3)$, $H_2^3(t) = (-t^2 + t^3)$ and $H_3^3(t) = (3t^2 - 2t^3)$ are graphed below.

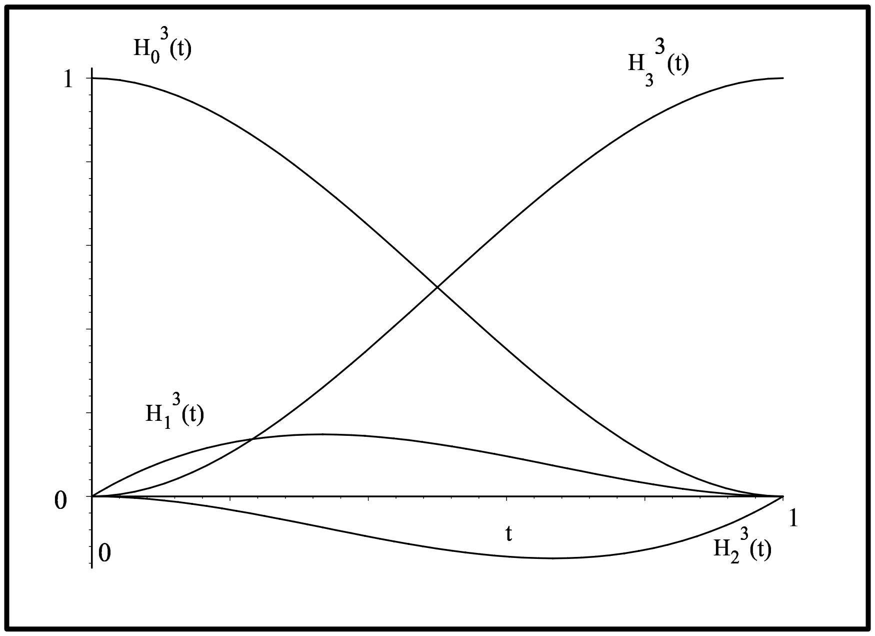

Figure 2: Cubic Hermite Basis Functions $H_i0^3(t)$

Notice the symmetry of the functions H_0^3 and H_3^3 about the line $t = \frac{1}{2}$, and the symmetry of the functions H_1^3 and $-H_2^3$ about the line $t = \frac{1}{2}$.

8.4 Piecewise Cubic Curves

The solution to Hermite interpolation allows one to chain together cubic curves to obtain piecewise cubic curves. One can also use the same method for constructing piecewise linear curves and the first method for piecewise circular curves. Specifically, construct a piecewise cubic curve from a sequence of $3n + 1$ points to be interpolated. Let p_0, p_1, p_2, p_3, p_4, $\cdots$, p_{3n} and construct a cubic curve by interpolation from the points p_{3i-3}, p_{3i-2}, p_{3i-1}, p_{3i} for $i = 1, 2, \cdots, n$.

An advantage of cubic Hermite interpolation is that it is easy to construct a smooth piecewise cubic curve. A second advantage of cubic Hermite curves that is important in geometric design is the symmetry in the data provided, at each point there is a tangent vector. This allows one to modify either the point or the tangent vector to affect a change in the curve. In the straight interpolation method, one can only modify points that the curves pass through. The effect of changing any one point can be hard to predict especially changing the joint points p_{3i}.

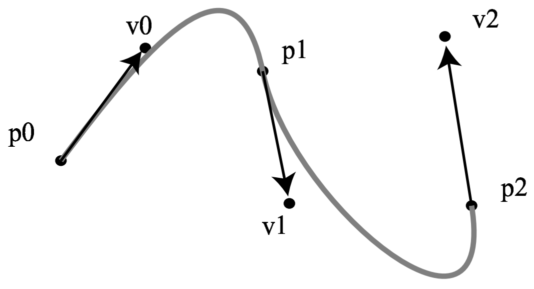

Figure 3: Hermite Spline Construction

The final advantage of the cubic Hermite curves is that the admit very simple piecewise descriptions. Let the given data be p_0, v_0, p_1, v_1, $\cdots$, p_n, v_n so that we have n cubic curves. Each individual cubic curve has the description

$$c_k(s) = H^3_0(s)\, p_k + H^3_1(s)\, v_k + H^3_2(s)\, v_{k+1} + H^3_3(s)\, p_{k+1}$$

where $0 \le s \le 1$. The piecewise description can then be written by letting $s = t - k$, as

$$c(t) = \begin{cases} c_0(t-0) & \text{for } 0 \le t \le 1, \\ c_1(t-1) & \text{for } 1 \le t \le 2, \\ \vdots & \vdots, \\ c_k(t-k) & \text{for } k \le t \le k+1, \\ \vdots & \vdots, \\ c_{n-1}(t-(n-1)) & \text{for } n-1 \le t \le n \end{cases}$$

One can easily alter the above description allowing the individual points to have prescribed parameter values, by letting $s = (t - t_k)/(t_{k+1} - t_k)$ for the kth curve. We used equally spaced integer values for simplicity.

Piecewise cubic curves constructed through Hermite interpolation are guaranteed to be C^1 (parametrically smooth and therefore also G^1 geometrically smooth). One can also modify the construction of cubic Hermite curves to construct piecewise cubic curves that are geometrically smooth but are not parametrically smooth (see exercises). We note that it is also possible to easily construct piecewise cubic curves that are C^2, that is have two continuous derivatives. A simple method to construct a C^2 curve is to construct parametrically prescribing at $t = 0$ the point, the first derivative, and the second derivative then prescribing at subsequent t values the value of the second derivative. We thus have the interpolation problem find a cubic curve that satisfies $c(0) = p_0$, $c'(0) = v_0$, $c''(0) = a_0$ and $c''(1) = a_1$ which has solution

$$c(t) = p_0 + v_0\, t + \frac{1}{2}\, a_0\, t^2 + \frac{1}{6}\,(a_1 - a_0)\, t^3.$$

This can continued knowing $c(1) = p_1 = p_0 + v_0 + \frac{1}{3}\, a_0 + \frac{1}{6}\, a_1$, $c'(1) = v_1 = v_0 + \frac{1}{2}\, a_0 + \frac{1}{2}\, a_1$, $c''(1) = a_1$ and prescribing $c''(2) = a_2$.

There is a problem with this method of constructing C^2 piecewise cubic curves. The method is numerically unstable. Round-off errors in the computations will accumulate and greatly affect the shape of the curves. In particular, the points $p_n = c(n)$ for integers $n \geq 1$ where the second derivatives are prescribed will vary greatly with small changes in the data provided. This means that one is unable to predict the shape of the curve from the given information. This is similar to the fact that Taylor polynomials of a function only provide good approximations locally, at distances away from the point where the derivatives are known one can not accurately predict the value of the function.

8.5 Fitting Data to Cubics

Fitting data to a cubic is much the same as fitting data to a quadratic, especially using the matrix version of least squares. The set up is exactly the same the only difference is the number of variables in the problem. Consider the following example.

Find the cubic function that is best fits the data in the table

x	1.0	2.0	1.5	4.0	3.0	2.0
y	2.0	3.0	3.5	2.5	0.0	0.5

Since there are no given parameter (t) values, we will use equally spaced values with spacing $\Delta t = 0.5$ starting with $t_0 = 0$. This is an arbitrary choice, different choices will yield possibly different curves. Therefore, we need to solve the matrix system

$$\begin{bmatrix} 1.0 & 2.0 \\ 2.0 & 3.0 \\ 1.5 & 3.5 \\ 4.0 & 2.5 \\ 3.0 & 0.0 \\ 2.0 & 0.5 \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.50 & 0.25 & 0.125 \\ 1.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 1.50 & 2.25 & 3.375 \\ 1.0 & 2.0 & 4.0 & 8.0 \\ 1.0 & 2.50 & 6.25 & 15.625 \end{bmatrix} \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix}$$

Using least squares on the above system $P = T\,A$ we have the solution $A = (T^tT)^{-1}T^tP$ as

$\mathbf{a}_i$	x	y
$\mathbf{a}_0$	1.12	1.82
$\mathbf{a}_1$	-0.30	6.03
$\mathbf{a}_2$	2.48	-5.90
$\mathbf{a}_3$	-0.89	1.30

Different solutions arise when different parameter values are used. For instance, in the graph below, the dark curve is the solution above, the medium gray curve uses chord length parameters $t_0 = 0$ then $t_{i+1} = t_i + d_i$ where $d_i = \|p_{i+1} - p_i\|$, and the light gray curve uses centripetal parameters $t_0 = 0$ then $t_{i+1} = t_i + d_i^{1/2}$.

8.6 Exercises

1. (Computational) Given the data below

t	x	y
0.0	1.0	2.0
0.3	2.0	3.0
0.6	4.0	4.0
1.0	5.0	2.0

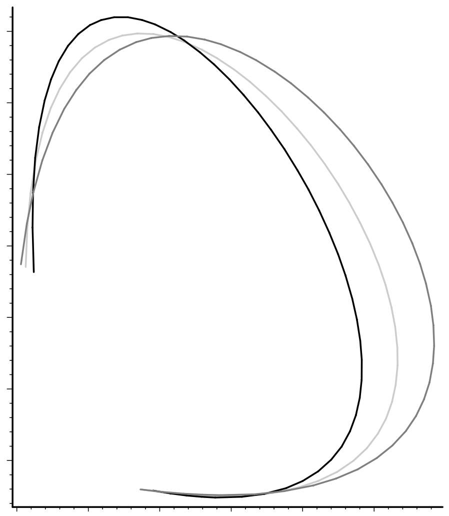
Figure 4: Data Fitted with Different Parameter Values

(a) Find the cubic curve that satisfies the above data.

(b) Find the Hermite data that creates the same curve.

2. (Interactive) Complete the interactive exercises associated with cubic Hermite curves.

3. (Computational/Thought) Sketch the piecewise cubic Hermite curve with the data in the figure below.

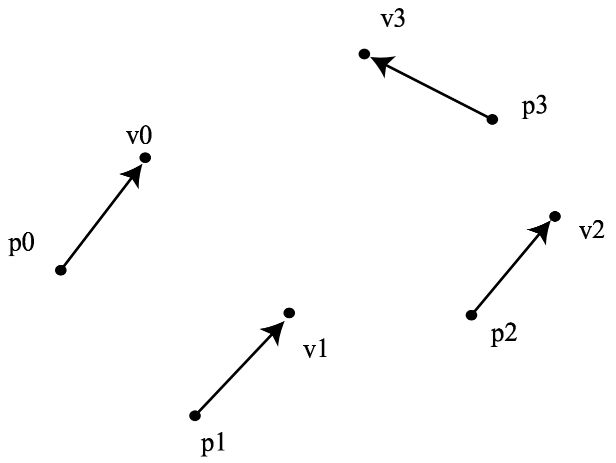

Figure 5: Sketch the piecewise cubic Hermite curve with this data

4. (Computational) Given the data

$$\boxed{\begin{array}{c} c(0) = [1,2] \\ c'(0) = [1,0] \\ c''(0) = [1,1] \\ c''(1) = [-1,1] \\ c''(2) = [1,2] \end{array}}$$

(a) Find a piecewise circular curve that satisfies this data.

(b) Repeat the above problem using $c''(0) = [1.01, 0.99]$, $c''(1) = [-1.01, 0.99]$, and $c''(2) = [0.98, 2.01]$.

(c) How does $c(1)$ and $c(2)$ differ from the curves in the previous two parts of this exercise.

5. (Computational) Given the data

x	1.2	0.8	1.5	1.8	3.0	3.0	2.5
y	2.2	2.5	3.2	3.0	4.0	2.0	1.8

(a) Find the best fit cubic to the data using equally spaced parameters with increment $\Delta t = 1.0$.

(b) Find the best fit cubic to the data using chord length parameters.

(c) Find the best fit cubic to the data using centripetal spaced parameters.

(d) Plot each of cubics generated above, which is provides the best fit to the data in your opinion. Why?