

3 Hermite and Cubic Spline Interpolation

Suppose that we have $m + 1$ data points P_0, \dots, P_m through which we wish to draw a curve such as that shown in Figure 3.1 (in which $m = 6$).

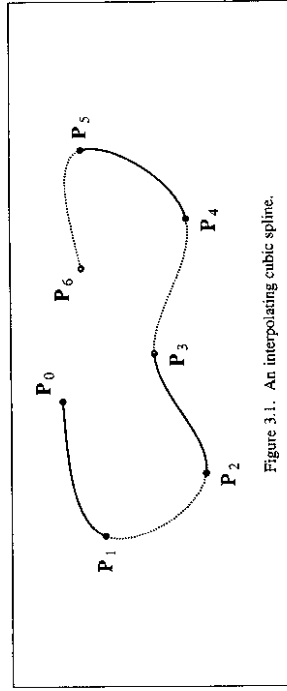


Figure 3.1. An interpolating cubic spline.

Each successive pair of data points is connected by a distinct curve segment. The i^{th} segment runs from P_i to P_{i+1} , and we will assume that the parameter \bar{u} runs correspondingly from the knot \bar{u}_i to the knot \bar{u}_{i+1} to generate this segment. This corresponds to the knot sequence and parameter range outlined in Chapter 2 with the special choices $\bar{u}_0 = \bar{u}_j = \bar{u}_{\min}$ and $\bar{u}_{\max} = \bar{u}_i = \bar{u}_m = \bar{u}_{\text{last}}$. Since each

such segment $Q_i(\bar{u})$ is represented parametrically as $(X_i(\bar{u}), Y_i(\bar{u}))$, we are really concerned with how the $X_i(\bar{u})$ and $Y_i(\bar{u})$ are determined by the points

$$P_i = (x_i, y_i).$$

In general, the x -coordinates $X(\bar{u})$ of points on a curve are determined solely by the x -coordinates x_0, \dots, x_m of the data points, and similarly $Y(\bar{u})$ is determined solely by the y -coordinates of the data points. Since both $X(\bar{u})$ and $Y(\bar{u})$ are treated in the same way we will discuss only $Y(\bar{u})$; indeed, to obtain curves in three dimensions we simply define a $Z(\bar{u})$ as well and let $Q_i(u)$ be given by $(X_i(u), Y_i(u), Z_i(u))$.

For ease of computation we will limit ourselves to the use of polynomials in defining $X_i(u)$, $Y_i(u)$ and $Z_i(u)$. Indeed cubic polynomials usually provide sufficient flexibility for many applications at reasonable cost. For the curve in Figure 3.1, then, $Y(\bar{u})$ is shown in Figure 3.2.

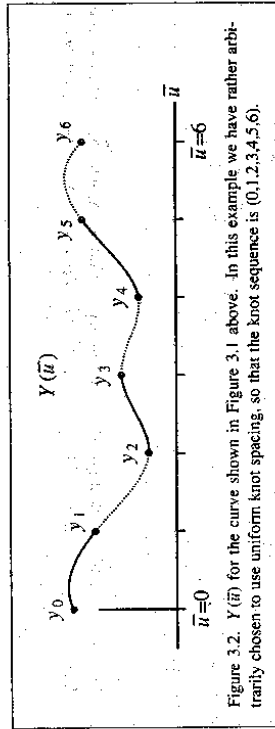


Figure 3.2. $Y(\bar{u})$ for the curve shown in Figure 3.1 above. In this example we have rather arbitrarily chosen to use uniform knot spacing, so that the knot sequence is $(0, 1, 2, 3, 4, 5, 6)$.

It will be easiest to continue the discussion by reparametrizing each segment Y_i separately by substituting u for \bar{u} as was described earlier. This means that $u = \bar{u} - i$ for the knot sequence given in Figure 3.2. Each $Y_i(u)$ is a cubic polynomial in the parameter u . We know two things in particular about

$$Y_i(u) = a_i + b_i u + c_i u^2 + d_i u^3,$$

namely that

$$Y_i(0) = y_i = a_i$$

$$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i.$$

Because we have four coefficients to determine, we need two other constraints to completely determine a particular $Y_i(u)$. One easy way to do this is to simply pick, arbitrarily, first derivatives D_i of $Y(u)$ at each knot \bar{u}_i , so that

$$Y_i^{(1)}(0) = D_i = b_i$$

$$Y_i^{(1)}(1) = D_{i+1} = b_i + 2c_i + 3d_i.$$

These four equations can be solved symbolically, once and for all, to yield

$$a_i = y_i$$

$$b_i = D_i$$

$$c_i = 3(y_{i+1} - y_i) - 2D_i - D_{i+1} \quad (3.1)$$

$$d_i = 2(y_i - y_{i+1}) + D_i + D_{i+1}.$$

Since we use D_i as the derivative at the left end of the i^{th} segment (i.e., as $Y_i^{(1)}(0)$) and at the right end of the $(i-1)^{\text{th}}$ segment (as $Y_{i-1}^{(1)}(1)$), $Y(u)$ has a continuous first derivative.

This technique is called *Hermite interpolation*. It can be generalized to higher-order polynomials.

How are the D_i specified? One possibility is to compute them automatically, perhaps by fitting a parabola through y_{i-1} , y_i , and y_{i+1} , and using its derivative at y_i as D_i ; arbitrary values (such as 0) can be used at the end points [Kochanek et al.82]. Or one can use for D_i the y component of a weighted average of the vector from P_{i-1} to P_i and the vector from P_{i+1} to P_i [Kochanek/Bartels84]. Or the user may specify derivative vectors directly. Some of these possibilities are discussed later in Chapter 21.

It is possible to arrange that successive segments match second as well as first derivatives at joints, using only cubic polynomials. Suppose, as above, that we want to interpolate the $(m+1)$ points P_0, \dots, P_m by such a curve. Each of the m segments $Y_0(u), \dots, Y_{m-1}(u)$ is a cubic polynomial determined by four coefficients. Hence we have $4m$ unknown values to determine. At each of the $(m-1)$ interior knots $\bar{u}_1, \dots, \bar{u}_{m-1}$ (where two segments meet) we have four con-

$$Y_{i-1}(1) = y_i, \quad Y_{i-1}^{(1)}(1) = Y_i^{(1)}(0)$$

$$Y_i(0) = y_i, \quad Y_i^{(2)}(1) = Y_i^{(2)}(0).$$

The result of this forward elimination process will be

$$\begin{bmatrix} 1 & \gamma_0 & & & & & & \\ & 1 & \gamma_1 & & & & & \\ & & 1 & \gamma_2 & & & & \\ & & & & \gamma_{m-2} & & & \\ & & & & 1 & \gamma_{m-1} & & \\ & & & & & 1 & & \\ & & & & & & & \delta_m \end{bmatrix} = \begin{bmatrix} D_0 & & & & & & & \\ & D_1 & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & D_m \end{bmatrix}$$

This directly yields the value of D_m , and it is then a simple matter to solve successively for D_{m-1}, \dots, D_0 in a process of backward substitution:

```

D_m ← δ_m
for i ← m-1 step -1 until 0 do
    D_i ← δ_i - γ_i D_{i+1}
endfor.

```

The multiplicative factors γ_i that accomplish the forward substitution need only be computed once. The δ_j 's must be computed and the backward substitution performed separately for each coordinate. When a data point is moved, the values $\delta_j, \dots, \delta_m$ must be recomputed and the entire backward substitution again performed.

3.2 Other End Conditions For Cubic Interpolating Splines

There are many other ways in which to determine the additional two constraints needed to define a C^2 continuous interpolating cubic spline fully. These conditions are most commonly applied to the ends of a curve; hence the name *end conditions*; the natural cubic splines offer an example of this. However, all that is really necessary is to provide the missing two conditions. Any two linear equations that are independent of those provided by the interpolation conditions could be used. They could involve data points or derivatives interior to the curve as well as at the ends. Whatever conditions are used, they will have some influence over the shape of the entire curve. For example, instead of fixing the second derivatives at the first and last knot to zero, we may fix the first derivatives there to be zero.

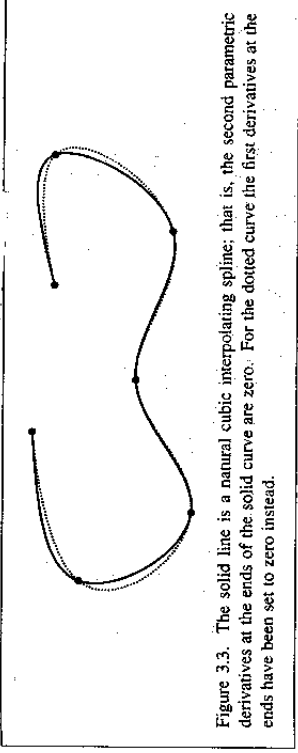


Figure 3.3. The solid line is a natural cubic interpolating spline; that is, the second parametric derivatives at the ends of the solid curve are zero. For the dotted curve the first derivatives at the ends have been set to zero instead.

Another possibility, which de Boor calls the *not-a-knot* condition [de Boor78], is to require C^3 continuity at the second and next-to-last knots u_1 and u_{m-1} . In effect the first two segments are a single polynomial, as are the last two.

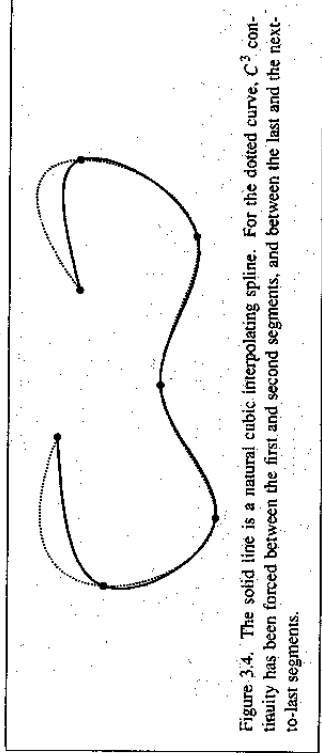


Figure 3.4. The solid line is a natural cubic interpolating spline. For the dotted curve, C^3 continuity has been forced between the first and second segments, and between the last and the next-to-last segments.

Yet another alternative, suggested by Forsythe, Malcolm and Moler [Forsythe et al.77], is to use the third derivatives of the cubic polynomials that interpolate the first and last four points as the third derivatives of the first and last segments. One might allow the user to explicitly supply any two of the first, second, or third derivative vectors at the ends. In any case, we can construct and solve a set of equations very much as we did for the natural cubic splines. Additional discussion of how this can be done, and algorithms, are given in Chapter 4 of [Forsythe et al.77] and in Chapter 4 of [de Boor78]. For a uniform knot vector, and indeed for any reasonable strictly increasing sequence of knots, these equations are well conditioned and can be solved easily and accurately.

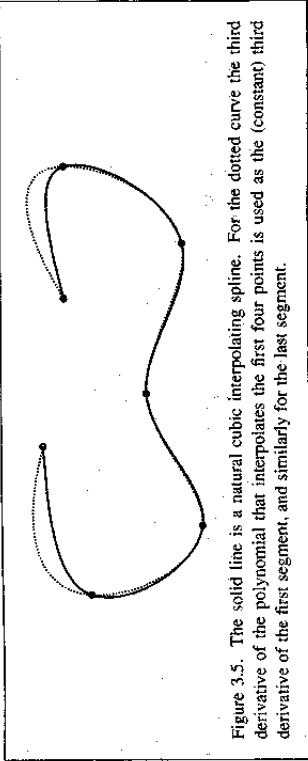


Figure 3.5. The solid line is a natural cubic interpolating spline. For the dotted curve the third derivative of the polynomial that interpolates the first four points is used as the (constant) third derivative of the first segment, and similarly for the last segment.

3.3 Knot Spacing

Although the end conditions discussed above affect the entire curve, their principal influence is felt at the endpoints. Gross changes to a curve's shape can be made anywhere, without moving the interpolation points, by varying the knot spacing. (See Figure 3.6.)

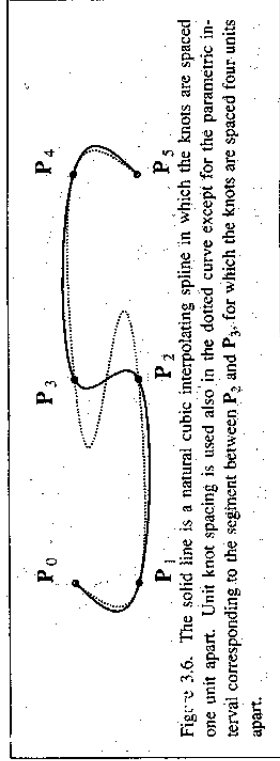


Figure 3.6. The solid line is a natural cubic interpolating spline in which the knots are spaced one unit apart. Unit knot spacing is used also in the dotted curve except for the parametric interval corresponding to the segment between P_2 and P_3 , for which the knots are spaced four units apart.

With the single exception of Figure 3.6, we have used a uniform knot sequence in defining the interpolating cubic spline curves discussed above. The knot vector for the solid curve in Figure 3.6 is

0, 1, 2, 3, 4, 5

while the dotted curve interpolates the same data points, but for the knot vector

0, 1, 2, 6, 7, 8.

Thus knot spacing can be used to influence shape; the more difficult question is how that influence can be controlled intuitively.

Uniform knot spacing is one obvious way to define a knot sequence. The Euclidean distance between data points is a second natural choice for the length of the parametric interval over which u varies in defining a segment.

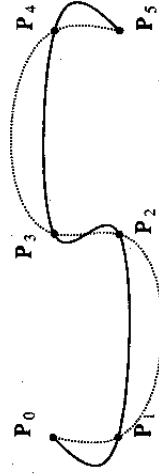


Figure 3.7. The solid line in the above figure is a natural cubic interpolating spline in which the knots are spaced a unit apart. In the case of the dotted curve, the knots corresponding to two successive data points differ in value by the Euclidean distance separating the two points.

3.4 Closed Curves

It is sometimes useful to generate closed curves such as in Figure 3.8. In this case, equation (3.2) applies at each of the m points, with the caveat that indices must be computed modulo $m+1$. The system of equations that results looks a little different:

$$\begin{bmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & 1 & 4 & 1 & \\ & & & \ddots & \ddots & \ddots \\ 1 & & & & & 1 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ \vdots \\ \vdots \\ D_m \end{bmatrix} = \begin{bmatrix} 3(y_1 - y_m) \\ 3(y_2 - y_0) \\ \vdots \\ \vdots \\ 3(y_m - y_{m-2}) \\ 3(y_0 - y_{m-1}) \end{bmatrix}$$

Basically one solves this system as one solved for the D_i for an open curve. Doing forward elimination, however, it is necessary to compute and save nonzero values for entries in the rightmost column and to successively cancel the leftmost nonzero value in the bottom row. The analogous change must be made to the back substitution process as well.

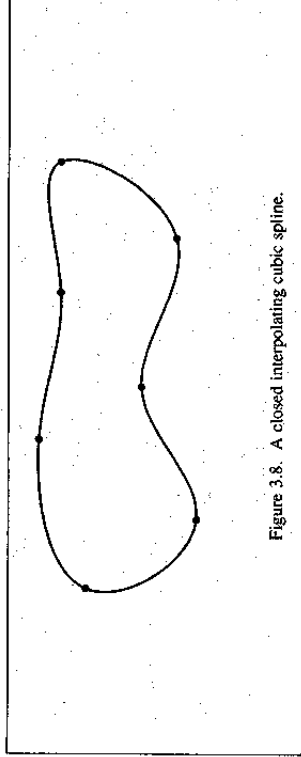


Figure 3.8. A closed interpolating cubic spline.