12
Spline Interpolation

Given a set of points, it is easy to compute a polynomial that passes through the points. The Lagrange polynomial (LP) of Section 10.2 is an example of such a polynomial. However, as the discussion in Section 8.8 (especially Exercise 8.16) illustrates, a curve based on a high-degree polynomial may wiggle wildly and its shape may be far from what the user has in mind. In practical work we are normally interested in a smooth, tight curve that proceeds from point to point such that each segment between two points is a smooth arc. The spline approach to curve design, discussed in this chapter, constructs such a curve from individual segments, each a simple curve, generally a parametric cubic (PC). This chapter illustrates spline interpolation with four examples, cubic splines (Section 12.1), the Akima spline (Section 12.2), cardinal splines (Section 12.5), and Kochanek–Bartels splines (Section 12.8). Another important type, the B-spline, is the topic of Chapter 14. Other types of splines are known and are discussed in the scientific literature. A short history of splines can be found in [Schumaker 81] and [Farin 04].

For those looking for other texts on splines, the bibliography lists several books by Gerald Farin, and I would also like to recommend [Späth 95a,b] and [Dierckx 95].

Definition: A spline is a set of polynomials of degree k that are smoothly connected at certain data points. At each data point, two polynomials connect, and their first derivatives (tangent vectors) have the same values. The definition also requires that all their derivatives up to the $(k-1)$st be the same at the point.

D. Salomon, *The Computer Graphics Manual*, Texts in Computer Science,
DOI 10.1007/978-0-85729-886-7_12, © Springer-Verlag London Limited 2011

12.1 The Cubic Spline Curve

The cubic spline was originally introduced by James Ferguson in [Ferguson 64]. Given n data points that are numbered $\mathbf{P}_1$ through $\mathbf{P}_n$, there are infinitely many curves that pass through all the points in order of their numbers (Figure 12.1a), but the eye often tends to trace *one* imaginary smooth curve through the points, especially if the points are arranged in a familiar pattern. It is therefore useful to have an algorithm that does the same. Since the computer does not recognize familiar patterns the way humans do, such a method should be interactive, thereby allowing the user to create the desired curve.

The cubic spline method is such an algorithm. Given n data points, it makes it possible to construct a smooth curve that passes through the points (see definition of data points in Section 8.6). The curve consists of $n-1$ individual Hermite segments that are smoothly connected at the $n-2$ interior points and that are easy to compute and display. For the segments to meet at the interior points, their tangent vectors (first derivatives) must be the same at each interior point. An added feature of cubic splines is that their second derivatives are also the same at the interior points. The cubic spline method is interactive. The user can control the shape of the curve by varying the two extreme tangent vectors at the beginning and the end of the curve.

Given the n data points $\mathbf{P}_1$, $\mathbf{P}_2$, through $\mathbf{P}_n$, we look for $n-1$ parametric cubics $\mathbf{P}_1(t)$, $\mathbf{P}_2(t), \ldots, \mathbf{P}_{n-1}(t)$ such that $\mathbf{P}_k(t)$ is the polynomial segment from point $\mathbf{P}_k$ to point $\mathbf{P}_{k+1}$ (Figure 12.1b). The PCs will have to be smoothly connected at the $n-2$ interior points $\mathbf{P}_2$, $\mathbf{P}_3, \ldots, \mathbf{P}_{n-1}$, which means that their first derivatives will have to match at every interior point. The definition of a spline requires that their second derivatives match too. This requirement (the boundary condition of the cubic spline) is important because it provides the necessary equations and also results in a tight curve in the sense that once the curve is drawn, the eye can no longer detect the positions of the original data points.

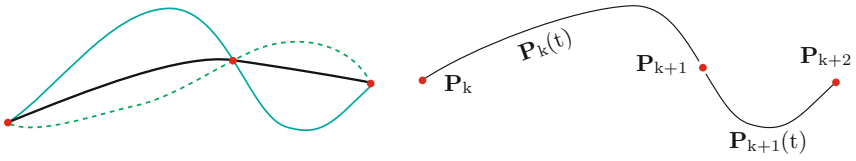


Figure 12.1: (a) Three Different Curves. (b) Two Segments.

The principle of cubic splines is to divide the set of n points into $n-1$ overlapping pairs of two points each and to fit a Hermite segment (Equations (11.4) and (11.5)) to each pair. The pairs are $(\mathbf{P}_1, \mathbf{P}_2)$, $(\mathbf{P}_2, \mathbf{P}_3)$, and so on, up to $(\mathbf{P}_{n-1}, \mathbf{P}_n)$. Recall that a Hermite curve segment is specified by two points and two tangents. In our case, all the points are given, so the only unknowns are the tangent vectors. In order for segments $\mathbf{P}_k(t)$ and $\mathbf{P}_{k+1}(t)$ to connect smoothly at point $\mathbf{P}_{k+1}$, the end tangent of $\mathbf{P}_k(t)$ has to equal the start tangent of $\mathbf{P}_{k+1}(t)$. Thus, there is only one tangent vector per point, for a total of n unknowns.

The unknown tangent vectors are computed as the solutions of a system of n equations. The equations are derived from the requirement that the second derivatives of the individual segments match at every interior point. However, there are only $n-2$ interior points, so we can only have $n-2$ equations, enough to solve for only $n-2$ unknowns.

The key to resolving this shortage of equations is to ask the user to provide the software with the values of two tangent vectors (normally the first and last ones). Once this is done, the equations can easily be solved, yielding the remaining $n-2$ tangents. This seems a strange way to solve equations, but it has the advantage of being interactive. If the resulting curve looks wrong, the user can repeat the calculation with two new tangent vectors. Before delving into the details, here is a summary of the steps involved.

1. The n data points are input into the program.
2. The user provides values (guesses or estimates) for two tangent vectors.
3. The program sets up $n-2$ equations, with the remaining $n-2$ tangent vectors as the unknowns, and solves them.
4. The program loops $n-1$ times. In each iteration, it selects two adjacent points and their tangent vectors to compute one Hermite segment.

We start with three adjacent points, $\mathbf{P}_k$, $\mathbf{P}_{k+1}$, and $\mathbf{P}_{k+2}$, of which $\mathbf{P}_{k+1}$ must be an interior point and the other two can be either interior or endpoints. Thus, k varies from 1 to $n-2$. The Hermite segment from $\mathbf{P}_k$ to $\mathbf{P}_{k+1}$ is denoted by $\mathbf{P}_k(t)$, which implies that $\mathbf{P}_k(0)=\mathbf{P}_k$ and $\mathbf{P}_k(1)=\mathbf{P}_{k+1}$. The tangent vectors of $\mathbf{P}_k(t)$ at the endpoints are still unknown and are denoted by $\mathbf{P}_k^t$ and $\mathbf{P}_{k+1}^t$. The first step is to express segment $\mathbf{P}_k(t)$ geometrically, in terms of the two endpoints and the two tangents. Applying Equation (11.4) to our segment results in

$$\begin{aligned}\mathbf{P}_k(t) = \mathbf{P}_k + \mathbf{P}_k^t t + \left[3(\mathbf{P}_{k+1}-\mathbf{P}_k) - 2\mathbf{P}_k^t - \mathbf{P}_{k+1}^t\right] t^2 \\ + \left[2(\mathbf{P}_k-\mathbf{P}_{k+1}) + \mathbf{P}_k^t + \mathbf{P}_{k+1}^t\right] t^3.\end{aligned} \tag{12.1}$$

When the same equation is applied to the next segment $\mathbf{P}_{k+1}(t)$ (from $\mathbf{P}_{k+1}$ to $\mathbf{P}_{k+2}$), it becomes

$$\begin{aligned}\mathbf{P}_{k+1}(t) = \mathbf{P}_{k+1} + \mathbf{P}_{k+1}^t t + \left[3(\mathbf{P}_{k+2}-\mathbf{P}_{k+1}) - 2\mathbf{P}_{k+1}^t - \mathbf{P}_{k+2}^t\right] t^2 \\ + \left[2(\mathbf{P}_{k+1}-\mathbf{P}_{k+2}) + \mathbf{P}_{k+1}^t + \mathbf{P}_{k+2}^t\right] t^3.\end{aligned} \tag{12.2}$$

◇ **Exercise 12.1:** Where do we use the assumption that the first derivatives of segments $\mathbf{P}_k(t)$ and $\mathbf{P}_{k+1}(t)$ are equal at the interior point $\mathbf{P}_{k+1}$?

Next, we use the requirement that the second derivatives of the two segments be equal at the interior points. The second derivative $\mathbf{P}^{tt}(t)$ of a Hermite segment $\mathbf{P}(t)$ is obtained by differentiating Equation (11.1)

$$\mathbf{P}^{tt}(t) = 6\mathbf{a}t + 2\mathbf{b}. \tag{12.3}$$

Equality of the second derivatives at the interior point $\mathbf{P}_{k+1}$ implies

$$\mathbf{P}_k^{tt}(1) = \mathbf{P}_{k+1}^{tt}(0) \qquad \text{or} \qquad 6\mathbf{a}_k\times 1 + 2\mathbf{b}_k = 6\mathbf{a}_{k+1}\times 0 + 2\mathbf{b}_{k+1}. \tag{12.4}$$

Using the values of **a** and **b** from Equations (12.1) and (12.2), we get

$$\begin{aligned} &6\left[2(\mathbf{P}_k-\mathbf{P}_{k+1})+\mathbf{P}_k^t+\mathbf{P}_{k+1}^t\right]+2\left[3(\mathbf{P}_{k+1}-\mathbf{P}_k)-2\mathbf{P}_k^t-\mathbf{P}_{k+1}^t\right] \\ &\quad=2\left[3(\mathbf{P}_{k+2}-\mathbf{P}_{k+1})-2\mathbf{P}_{k+1}^t-\mathbf{P}_{k+2}^t\right], \end{aligned} \tag{12.5}$$

which, after simple algebraic manipulations, becomes

$$\mathbf{P}_k^t+4\mathbf{P}_{k+1}^t+\mathbf{P}_{k+2}^t=3(\mathbf{P}_{k+2}-\mathbf{P}_k). \tag{12.6}$$

The three quantities on the left side of Equation (12.6) are unknown. The two quantities on the right side are known.

Equation (12.6) can be written $n-2$ times for all the interior points $\mathbf{P}_{k+1}=\mathbf{P}_2, \mathbf{P}_3, \ldots, \mathbf{P}_{n-1}$ to obtain a system of $n-2$ linear algebraic equations expressed in matrix form as

$$\underbrace{\begin{pmatrix} 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 \\ & & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 & 4 & 1 \end{pmatrix}}_{n} \begin{pmatrix} \mathbf{P}_1^t \\ \mathbf{P}_2^t \\ \vdots \\ \mathbf{P}_n^t \end{pmatrix} = \begin{pmatrix} 3(\mathbf{P}_3-\mathbf{P}_1) \\ 3(\mathbf{P}_4-\mathbf{P}_2) \\ \vdots \\ 3(\mathbf{P}_n-\mathbf{P}_{n-2}) \end{pmatrix}. \tag{12.7}$$

Equation (12.7) is a system of $n-2$ equations in the n unknowns $\mathbf{P}_1^t, \mathbf{P}_2^t, \ldots, \mathbf{P}_n^t$. A practical approach to the solution is to let the user specify the values of the two extreme tangents $\mathbf{P}_1^t$ and $\mathbf{P}_n^t$. Once these values have been substituted in Equation (12.7), it's easy to solve it and obtain values for the remaining $n-2$ tangents, $\mathbf{P}_2^t$ through $\mathbf{P}_{n-1}^t$. The n tangent vectors are now used to calculate the original coefficients **a**, **b**, **c**, and **d** of each segment by means of Equations (11.3), (11.4), or (11.7), which should be written and solved $n-1$ times, once for each segment of the spline.

The reader should notice that the matrix of coefficients of Equation (12.7) is tridiagonal and therefore diagonally dominant and thus nonsingular. This means that the system of equations can always be solved and that it has a unique solution. (Matrices and their properties are discussed in texts on linear algebra.)

This approach to solving Equation (12.7) is called the *clamped* end condition. Its advantage is that the user can vary the shape of the curve by entering new values for $\mathbf{P}_1^t$ and $\mathbf{P}_n^t$ and recalculating. This allows for interactive design, where each step brings the curve closer to the desired shape. Figure 12.1a is an example of three cubic splines that pass through the same points and differ only in $\mathbf{P}_1^t$ and $\mathbf{P}_n^t$. It illustrates how the shape of the entire curve can be radically changed by modifying the two extreme tangents.

It is possible to let the user specify any two tangent vectors, not just the two extreme ones. However, varying the two extreme tangents is a natural way to edit and reshape the curve in practical applications.

Tension control. Section 11.2.3 shows how to control the tension of a Hermite segment by varying the magnitudes of the tangent vectors. Since a cubic spline is based on Hermite segments, its tension can also be controlled in the same way. The user may input a tension parameter s and the software simply multiplies every tangent vector by

s. Small values of s correspond to high tension, so a user-friendly algorithm inputs a parameter T in the interval $[0,1]$ and multiplies each tangent vector by $s=\alpha(1-T)$ for some predetermined α. Large values of T (close to 1) correspond to small s and therefore to high tension, while small values of T correspond to s close to α. This makes T a natural tension parameter. Section 12.5 has the similar relation $T=1-2s$, which makes more sense for cardinal splines.

The downside of the cubic spline is the following:

1. There is no local control. Modifying the extreme tangent vectors changes Equation (12.7) and results in a different set of n tangent vectors. The entire curve is modified!

2. Equation (12.7) is a system of n equations that, for large values of n, may be too slow to solve.

Picnic Blues (anagram of Cubic Spline).

12.1.1 Example

Given the four points $\mathbf{P}_1=(0,0)$, $\mathbf{P}_2=(1,0)$, $\mathbf{P}_3=(1,1)$, and $\mathbf{P}_4=(0,1)$, we are looking for three Hermite segments $\mathbf{P}_1(t)$, $\mathbf{P}_2(t)$, and $\mathbf{P}_3(t)$ that will connect smoothly at the two interior points $\mathbf{P}_2$ and $\mathbf{P}_3$ and will constitute the spline. We further select an initial direction $\mathbf{P}_1^t=(1,-1)$ and a final direction $\mathbf{P}_4^t=(-1,-1)$. Figure 12.2 shows the points, the two extreme tangent vectors, and the resulting curve.

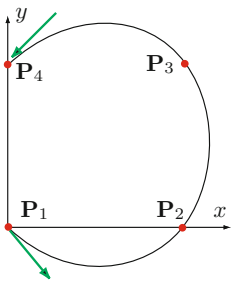


Figure 12.2: A Cubic Spline Example.

We first write Equation (12.7) for our special case ($n=4$)

$$\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} (1,-1) \\ \mathbf{P}_2^t \\ \mathbf{P}_3^t \\ (-1,-1) \end{pmatrix} = \begin{pmatrix} 3[(1,1)-(0,0)] \\ 3[(0,1)-(1,0)] \end{pmatrix} = \begin{pmatrix} (3,3) \\ (-3,3) \end{pmatrix},$$

$$\text{or} \qquad \begin{aligned} (1,-1)+4\mathbf{P}_2^t+\mathbf{P}_3^t &= (3,3), \\ \mathbf{P}_2^t+4\mathbf{P}_3^t+(-1,-1) &= (-3,3). \end{aligned}$$

This is a system of two equations in two unknowns. It is easy to solve and the solutions are $\mathbf{P}_2^t=(\frac{2}{3},\frac{4}{5})$ and $\mathbf{P}_3^t=(-\frac{2}{3},\frac{4}{5})$.

We now write Equation (11.7) three times, for the three spline segments. For the first segment, Equation (11.7) becomes

$$\begin{aligned}\mathbf{P}_1(t) &= (t^3, t^2, t, 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} (0,0) \\ (1,0) \\ (1,-1) \\ (\frac{2}{3}, \frac{4}{5}) \end{pmatrix} \\ &= (-\tfrac{1}{3}, -\tfrac{1}{5})t^3 + (\tfrac{1}{3}, \tfrac{6}{5})t^2 + (1,-1)t.\end{aligned}$$

The second segment is calculated in a similar way:

$$\begin{aligned}\mathbf{P}_2(t) &= (t^3, t^2, t, 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} (1,0) \\ (1,1) \\ (\frac{2}{3}, \frac{4}{5}) \\ (-\frac{2}{3}, \frac{4}{5}) \end{pmatrix} \\ &= (0, -\tfrac{2}{5})t^3 + (-\tfrac{2}{3}, \tfrac{3}{5})t^2 + (\tfrac{2}{3}, \tfrac{4}{5})t + (1,0).\end{aligned}$$

Finally, we write, for the third segment,

$$\begin{aligned}\mathbf{P}_3(t) &= (t^3, t^2, t, 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} (1,1) \\ (0,1) \\ (-\frac{2}{3}, \frac{4}{5}) \\ (-1,-1) \end{pmatrix} \\ &= (\tfrac{1}{3}, -\tfrac{1}{5})t^3 - (\tfrac{2}{3}, \tfrac{3}{5})t^2 + (-\tfrac{2}{3}, \tfrac{4}{5})t + (1,1),\end{aligned}$$

which completes the example.

◇ **Exercise 12.2:** Check to make sure that the three polynomial segments really connect at the two interior points. What are the tangent vectors at the points?

◇ **Exercise 12.3:** Redo the example of this section with an indefinite initial direction $\mathbf{P}_1^t = (0,0)$. What does it mean for a curve to start going in an indefinite direction?

12.1.2 Relaxed Cubic Splines

The original approach to the cubic spline curve is for the user to specify the two extreme tangent vectors. This approach is known as the *clamped* end condition. It is possible to have different end conditions, and the one described in this section is based on the simple idea of setting the two extreme second derivatives of the curve, $\mathbf{P}_1^{tt}(0)$ and $\mathbf{P}_{n-1}^{tt}(1)$, to zero. If we think of the second derivative as the acceleration of the curve (see the particle paradigm of Section 8.6), then this end condition implies constant speeds and therefore small curvatures at both ends of the curve. This is why this end condition is called *relaxed*.

It is easy to calculate the relaxed cubic spline. The second derivative of the parametric cubic $\mathbf{P}(t)$ is $\mathbf{P}^{tt}(t) = 6\mathbf{a}t + 2\mathbf{b}$ (Equation (12.3)). The end condition $\mathbf{P}_1^{tt}(0) = 0$ implies $2\mathbf{b}_1 = 0$ or, from Equation (11.3),

$$-3\mathbf{P}_1 + 3\mathbf{P}_2 - 2\mathbf{P}_1^t - \mathbf{P}_2^t = 0, \quad \text{which yields} \quad \mathbf{P}_1^t = \tfrac{3}{2}(\mathbf{P}_2 - \mathbf{P}_1) - \tfrac{1}{2}\mathbf{P}_2^t. \tag{12.8}$$

The other end condition, $\mathbf{P}_{n-1}^{tt}(1) = 0$, implies $6\mathbf{a}_{n-1} + 2\mathbf{b}_{n-1} = 0$ or, from Equation (11.3)

$$6\left(2\mathbf{P}_{n-1} - 2\mathbf{P}_n + \mathbf{P}_{n-1}^t + \mathbf{P}_n^t\right) + 2\left(-3\mathbf{P}_{n-1} + 3\mathbf{P}_n - 2\mathbf{P}_{n-1}^t - \mathbf{P}_n^t\right) = 0,$$

or

$$\mathbf{P}_n^t = \tfrac{3}{2}(\mathbf{P}_n - \mathbf{P}_{n-1}) - \tfrac{1}{2}\mathbf{P}_{n-1}^t. \tag{12.9}$$

Substituting Equations (12.8) and (12.9) in Equation (12.7) results in

$$\underbrace{\begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 \\ & & & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & \cdots & \cdots & 1 & 4 & 1 \end{bmatrix}}_{n} \begin{bmatrix} \frac{3}{2}(\mathbf{P}_2 - \mathbf{P}_1) - \frac{1}{2}\mathbf{P}_2^t \\ \mathbf{P}_2^t \\ \vdots \\ \mathbf{P}_{n-1}^t \\ \frac{3}{2}(\mathbf{P}_n - \mathbf{P}_{n-1}) - \frac{1}{2}\mathbf{P}_{n-1}^t \end{bmatrix} \quad (n-2) \tag{12.10}$$

$$= \begin{bmatrix} 3(\mathbf{P}_3 - \mathbf{P}_1) \\ 3(\mathbf{P}_4 - \mathbf{P}_2) \\ \vdots \\ 3(\mathbf{P}_{n-1} - \mathbf{P}_{n-3}) \\ 3(\mathbf{P}_n - \mathbf{P}_{n-2}) \end{bmatrix}.$$

This is a system of $n-2$ equations in the $n-2$ unknowns $\mathbf{P}_2^t, \mathbf{P}_3^t, \ldots, \mathbf{P}_{n-1}^t$. Calculating the relaxed cubic spline is done in the following steps:

1. Set up Equation (12.10) and solve it to obtain the $n-2$ interior tangent vectors.
2. Use $\mathbf{P}_2^t$ to calculate $\mathbf{P}_1^t$ from Equation (12.8). Similarly, use $\mathbf{P}_{n-1}^t$ to calculate $\mathbf{P}_n^t$ from Equation (12.9).
3. Now that the values of all n tangent vectors are known, write and solve Equation (11.4) or (11.7) $n-1$ times, each time calculating one spline segment.

The clamped cubic spline is interactive. The curve can be modified by varying the two extreme tangent vectors. The relaxed cubic spline, on the other hand, is not interactive. The only way to edit or modify it is to move the points or add points. The points, however, are data points that may be dictated by the problem on hand or that may be given by a user or a client, so it may not always be possible to move them.

Example: We use the same four points $\mathbf{P}_1 = (0,0)$, $\mathbf{P}_2 = (1,0)$, $\mathbf{P}_3 = (1,1)$, and $\mathbf{P}_4 = (0,1)$ of Section 12.1.1. The first step is to set up Equation (12.10) and solve it to obtain the two interior tangent vectors $\mathbf{P}_2^t$ and $\mathbf{P}_3^t$.

$$\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} (\frac{3}{2},0) - \frac{1}{2}\mathbf{P}_2^t \\ \mathbf{P}_2^t \\ \mathbf{P}_3^t \\ (-\frac{3}{2},0) - \frac{1}{2}\mathbf{P}_3^t \end{pmatrix} = \begin{pmatrix} (3,3) \\ (-3,3) \end{pmatrix}.$$

The solutions are

$$\mathbf{P}_2^t = \left(\frac{3}{5}, \frac{2}{3}\right), \quad \mathbf{P}_3^t = \left(-\frac{3}{5}, \frac{2}{3}\right).$$

The second step is to calculate $\mathbf{P}_1^t$ and $\mathbf{P}_4^t$

$$\mathbf{P}_1^t = \frac{3}{2}(\mathbf{P}_2 - \mathbf{P}_1) - \frac{1}{2}\mathbf{P}_2^t = \left(\frac{3}{2}, 0\right) - \frac{1}{2}\left(\frac{3}{5}, \frac{2}{3}\right) = \left(\frac{6}{5}, -\frac{1}{3}\right),$$
$$\mathbf{P}_4^t = \frac{3}{2}(\mathbf{P}_4 - \mathbf{P}_3) - \frac{1}{2}\mathbf{P}_3^t = \left(-\frac{3}{2}, 0\right) - \frac{1}{2}\left(-\frac{3}{5}, \frac{2}{3}\right) = \left(-\frac{6}{5}, -\frac{1}{3}\right).$$

Now that the values of all four tangent vectors are known, the last step is to write and solve Equation (11.4) or (11.7) three times to calculate each of the three segments of our example curve.

For the first segment, Equation (11.7) becomes

$$\begin{aligned}\mathbf{P}_1(t) &= (t^3, t^2, t, 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} (0,0) \\ (1,0) \\ (\frac{6}{5}, -\frac{1}{3}) \\ (\frac{3}{5}, \frac{2}{3}) \end{pmatrix} \\ &= (-\tfrac{1}{5}, \tfrac{1}{3})t^3 + (\tfrac{6}{5}, -\tfrac{1}{3})t.\end{aligned}$$

For the second segment, Equation (11.7) becomes

$$\begin{aligned}\mathbf{P}_2(t) &= (t^3, t^2, t, 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} (1,0) \\ (1,1) \\ (\frac{3}{5}, \frac{2}{3}) \\ (-\frac{3}{5}, \frac{2}{3}) \end{pmatrix} \\ &= (0, -\tfrac{2}{3})t^3 + (-\tfrac{3}{5}, 1)t^2 + (\tfrac{3}{5}, \tfrac{2}{3})t + (1, 0).\end{aligned}$$

◇ **Exercise 12.4:** Compute the third Hermite segment.

12.1.3 Cyclic Cubic Splines

The *cyclic* end condition is ideal for a closed cubic spline (Section 12.1.5) and also for a periodic cubic spline (Section 12.1.4). The condition is that the tangent vectors be equal at the two extremes of the curve (i.e., $\mathbf{P}_1^t = \mathbf{P}_n^t$) and the same for the second derivatives $\mathbf{P}_1^{tt} = \mathbf{P}_n^{tt}$. Notice that the curve doesn't have to be closed, i.e., a segment from $\mathbf{P}_n$ to $\mathbf{P}_1$ is not required.

Applying Equation (11.1) to the first condition yields

$$\mathbf{P}_1^t(0) = \mathbf{P}_{n-1}^t(1)$$

or

$$3\mathbf{a}_1 t^2 + 2\mathbf{b}_1 t + \mathbf{c}_1|_{t=0} = 3\mathbf{a}_{n-1}t^2 + 2\mathbf{b}_{n-1}t + \mathbf{c}_{n-1}|_{t=1}$$

or

$$\mathbf{c}_1 = 3\mathbf{a}_{n-1} + 2\mathbf{b}_{n-1} + \mathbf{c}_{n-1}. \tag{12.11}$$

Applying Equation (12.3) to the second condition yields

$$\mathbf{P}_1^{tt}(0) = \mathbf{P}_{n-1}^{tt}(1)$$

or

$$6\mathbf{a}_1 t + 2\mathbf{b}_1|_{t=0} = 6\mathbf{a}_{n-1}t + 2\mathbf{b}_{n-1}|_{t=1}$$

or

$$2\mathbf{b}_1 = 6\mathbf{a}_{n-1} + 2\mathbf{b}_{n-1}. \tag{12.12}$$

Subtracting Equations (12.11) and (12.12) yields $\mathbf{c}_1 - 2\mathbf{b}_1 = -3\mathbf{a}_{n-1} + \mathbf{c}_{n-1}$ or, from Equation (11.3),

$$\mathbf{P}_1^t - 2[-3\mathbf{P}_1 + 3\mathbf{P}_2 - 2\mathbf{P}_1^t - \mathbf{P}_2^t] = -3[2\mathbf{P}_{n-1} - 2\mathbf{P}_n + \mathbf{P}_{n-1}^t + \mathbf{P}_n^t] + \mathbf{P}_{n-1}^t.$$

This can be written

$$\mathbf{P}_1^t + 4\mathbf{P}_1^t + 3\mathbf{P}_n^t = 6(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_n - \mathbf{P}_{n-1}) - (\mathbf{P}_2^t + \mathbf{P}_{n-1}^t).$$

Using the end condition $\mathbf{P}_1^t = \mathbf{P}_n^t$, we get

$$\mathbf{P}_1^t = \mathbf{P}_n^t = \tfrac{3}{4}\left(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_n - \mathbf{P}_{n-1}\right) - \tfrac{1}{4}\left(\mathbf{P}_2^t + \mathbf{P}_{n-1}^t\right). \tag{12.13}$$

Substituting Equation (12.13) in Equation (12.7) results in

$$\underbrace{\begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 \\ & & & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & \cdots & \cdots & 1 & 4 & 1 \end{bmatrix}}_{n} \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} n-2 \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} \frac{3}{4}\left(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_n - \mathbf{P}_{n-1}\right) - \\ -\frac{1}{4}\left(\mathbf{P}_2^t + \mathbf{P}_{n-1}^t\right) \\ \mathbf{P}_2^t \\ \vdots \\ \mathbf{P}_{n-1}^t \\ \frac{3}{4}\left(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_n - \mathbf{P}_{n-1}\right) - \\ -\frac{1}{4}\left(\mathbf{P}_2^t + \mathbf{P}_{n-1}^t\right) \end{bmatrix} = \begin{bmatrix} 3(\mathbf{P}_3 - \mathbf{P}_1) \\ 3(\mathbf{P}_4 - \mathbf{P}_2) \\ \vdots \\ 3(\mathbf{P}_{n-1} - \mathbf{P}_{n-3}) \\ 3(\mathbf{P}_n - \mathbf{P}_{n-2}) \end{bmatrix}, \tag{12.14}$$

which is a system of $n-2$ equations in the $n-2$ unknowns $\mathbf{P}_2^t$, $\mathbf{P}_3^t, \ldots, \mathbf{P}_{n-1}^t$. Notice that in the case of a closed curve, these equations are somehow simplified because the two extreme points $\mathbf{P}_1$ and $\mathbf{P}_n$ are identical. Calculating the cyclic cubic spline is done in the following steps:

1. Set up Equation (12.14) and solve it to obtain the $n-2$ interior tangent vectors.
2. Use $\mathbf{P}_2^t$ and $\mathbf{P}_{n-1}^t$ to calculate $\mathbf{P}_1^t$ and $\mathbf{P}_n^t$ from Equation (12.13).
3. Now that the values of all n tangent vectors are known, write and solve Equation (11.4) or (11.7) $n-1$ times, each time calculating one spline segment.

Example: We select the five points $\mathbf{P}_1 = \mathbf{P}_5 = (0,-1)$, $\mathbf{P}_2 = (1,0)$, $\mathbf{P}_3 = (0,1)$, and $\mathbf{P}_4 = (-1,0)$ and calculate the cubic spline with the cyclic end condition for these points. Notice that the curve is closed since $\mathbf{P}_1 = \mathbf{P}_5$. Also, since the points are symmetric about the origin, we can expect the resulting four PC segments to be similar. We start with Equation (12.14)

$$\begin{bmatrix} 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \frac{3}{4}(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_5 - \mathbf{P}_4) - \frac{1}{4}(\mathbf{P}_2^t + \mathbf{P}_4^t) \\ \mathbf{P}_2^t \\ \mathbf{P}_3^t \\ \mathbf{P}_4^t \\ \frac{3}{4}(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_5 - \mathbf{P}_4) - \frac{1}{4}(\mathbf{P}_2^t + \mathbf{P}_4^t) \end{bmatrix} = \begin{bmatrix} 3(\mathbf{P}_3 - \mathbf{P}_1) \\ 3(\mathbf{P}_4 - \mathbf{P}_2) \\ 3(\mathbf{P}_5 - \mathbf{P}_3) \end{bmatrix},$$

which is solved to yield $\mathbf{P}_2^t = (0, 3/2)$, $\mathbf{P}_3^t = (-3/2, 0)$, and $\mathbf{P}_4^t = (0, -3/2)$. These values are used to solve Equation (12.13)

$$\mathbf{P}_1^t = \mathbf{P}_5^t = \tfrac{3}{4}\left(\mathbf{P}_2 - \mathbf{P}_1 + \mathbf{P}_5 - \mathbf{P}_4\right) - \tfrac{1}{4}\left(\mathbf{P}_2^t + \mathbf{P}_4^t\right),$$

which gives $\mathbf{P}_1^t = \mathbf{P}_5^t = (3/2, 0)$. The four segments can now be calculated in the usual way. For the first segment, Equation (11.7) becomes

$$\begin{aligned} \mathbf{P}_1(t) &= (t^3, t^2, t, 1) \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} (0,-1) \\ (1,0) \\ (\frac{3}{2},0) \\ (0,\frac{3}{2}) \end{pmatrix} \\ &= -(\tfrac{1}{2},\tfrac{1}{2})t^3 + (0,\tfrac{3}{2})t^2 + (\tfrac{3}{2},0)t + (0,-1). \end{aligned}$$

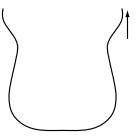
For the second segment, Equation (11.7) becomes

$$\begin{aligned} \mathbf{P}_2(t) &= (t^3, t^2, t, 1) \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} (1,0) \\ (0,1) \\ (0,\frac{3}{2}) \\ (-\frac{3}{2},0) \end{pmatrix} \\ &= (\tfrac{1}{2},-\tfrac{1}{2})t^3 + (-\tfrac{3}{2},0)t^2 + (0,\tfrac{3}{2})t + (1,0). \end{aligned}$$

◇ **Exercise 12.5:** Compute the third and fourth Hermite segments.

Notice how the symmetry of the problem causes the coefficients of $\mathbf{P}_1(t)$ and $\mathbf{P}_3(t)$ to have opposite signs, and the same for the coefficients of $\mathbf{P}_2(t)$ and $\mathbf{P}_4(t)$.

It is also possible to have an *anticyclic* end condition for the cubic spline. It requires that the two extreme tangent vectors have the same magnitudes but opposite directions $\mathbf{P}_1^t = -\mathbf{P}_n^t$ and the same condition for the second derivatives $\mathbf{P}_1^{tt} = -\mathbf{P}_n^{tt}$. Such an end condition makes sense for curves such as the cross section of a vase or any other surface of revolution.



Following steps similar to the ones for the cyclic case, we get for the anticyclic end condition

$$\mathbf{P}_1^t = -\mathbf{P}_n^t = \frac{3}{4}\left(\mathbf{P}_2 - \mathbf{P}_1 - \mathbf{P}_n + \mathbf{P}_{n-1}\right) - \frac{1}{4}\left(\mathbf{P}_2^t - \mathbf{P}_{n-1}^t\right). \tag{12.15}$$

◇ **Exercise 12.6:** Given the three points $\mathbf{P}_1 = (-1, 0)$, $\mathbf{P}_2 = (0, 1)$, and $\mathbf{P}_3 = (1, 0)$, calculate the anticyclic cubic spline for them and compare it to the clamped cubic spline for the same points.

12.1.4 Periodic Cubic Splines

A periodic function $f(x)$ is one that repeats itself. If p is the period of the function, then $f(x + p) = f(x)$ for any x. A two-dimensional cubic spline is periodic if it has the same extreme tangent vectors (i.e., if it starts and ends going in the same direction) and if its two extreme points $\mathbf{P}(0)$ and $\mathbf{P}(1)$ have the same y coordinate. If the curve satisfies these conditions, then we can place consecutive copies of it side by side and the result would look like a single periodic curve.

The case of a three-dimensional periodic cubic spline is less clear. It seems that the two extreme points can be any points (they don't have to have the same y or z coordinates or any other relationship), so the condition for periodicity is that the curve will have the same start and end tangents, i.e., it will be cyclic.

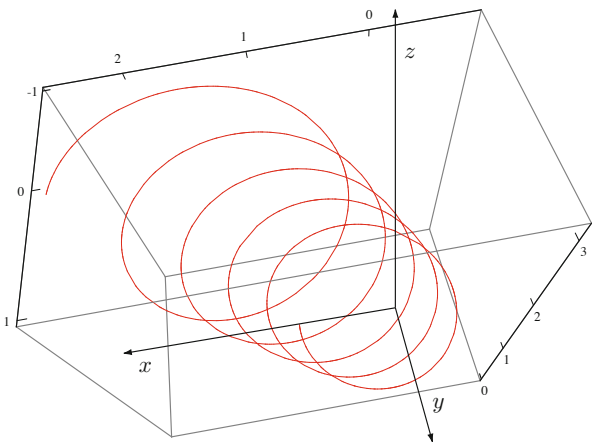
Example: Exercise 8.11 shows that the parametric expression $(\cos t, \sin t, t)$ describes a helix (see also Section 9.4.1 for a double helix). Modifying this expression to $\mathbf{P}(t) = (0.05t + \cos t, \sin t, .1t)$ creates a helix that moves in the x direction as it climbs up in the z direction. Figure 12.3 shows its behavior. This curve starts at $\mathbf{P}(0) = (1, 0, 0)$ and ends at $\mathbf{P}(10\pi) = (0.5\pi + 1, 0, \pi)$. There is no special relation between the start and end points, but the curve is periodic since both its start and end tangents equal $\mathbf{P}^t(0) = \mathbf{P}^t(10\pi) = (0.05, 1, 0.1)$. We can construct another period of this curve by copying it, moving the copy parallel to itself, and placing it such that the start point of the copy is at the end point of the original curve.

Notice that it is possible to make the start and end points even more unrelated by, for example, tilting the helix also in the y direction as it climbs up in the z direction. This kind of effect is achieved by an expression such as

$$\mathbf{P}(t) = (0.05t + \cos t, -0.05t^2 + \sin t, 0.1t).$$

12.1.5 Closed Cubic Splines

A closed cubic spline has an extra curve segment from $\mathbf{P}_n$ to $\mathbf{P}_1$ that closes the curve. In such a curve, every point is interior, so Equation (12.7) becomes a system of n equations in the same n unknowns. No user input is needed, which implies that the only way to control or modify such a curve is to move, add, or delete points. It is convenient to define the two additional points $\mathbf{P}_{n+1} \stackrel{\text{def}}{=} \mathbf{P}_1$ and $\mathbf{P}_{n+2} \stackrel{\text{def}}{=} \mathbf{P}_2$. Equation (12.7) then becomes




```
(* tilted helix as a periodic curve *)
ParametricPlot3D[{.05t+Cos[t],Sin[t],.1t},{t,0,10Pi},
 Ticks->{{-1,0,1,2},{-1,0,1},{0,1,2,3}},
 PlotPoints->100,PlotStyle->Red]
```

Figure 12.3: A Tilted Helix as a Periodic Curve.

$$n\left\{\underbrace{\begin{bmatrix} 1 & 4 & 1 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & \cdots & 0 \\ & & & \ddots & \ddots & & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 & 4 & 1 \\ 1 & \cdots & \cdots & \cdots & 0 & 1 & 4 \\ 4 & 1 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}}_{n}\right\} \begin{bmatrix} \mathbf{P}_1^t \\ \mathbf{P}_2^t \\ \vdots \\ \mathbf{P}_{n-1}^t \\ \mathbf{P}_n^t \end{bmatrix} = \begin{bmatrix} 3(\mathbf{P}_3 - \mathbf{P}_1) \\ 3(\mathbf{P}_4 - \mathbf{P}_2) \\ \vdots \\ 3(\mathbf{P}_{n+1} - \mathbf{P}_{n-1}) \\ 3(\mathbf{P}_{n+2} - \mathbf{P}_n) \end{bmatrix}. \tag{12.16}$$

Example: Given the four points of Section 12.1.1, $\mathbf{P}_1 = (0,0)$, $\mathbf{P}_2 = (1,0)$, $\mathbf{P}_3 = (1,1)$, and $\mathbf{P}_4 = (0,1)$, we are looking for four Hermite segments $\mathbf{P}_1(t)$, $\mathbf{P}_2(t)$, $\mathbf{P}_3(t)$, and $\mathbf{P}_4(t)$ that would connect smoothly at the four points. Equation (12.16) becomes

$$\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{P}_1^t \\ \mathbf{P}_2^t \\ \mathbf{P}_3^t \\ \mathbf{P}_4^t \end{pmatrix} = \begin{bmatrix} 3(\mathbf{P}_3 - \mathbf{P}_1) \\ 3(\mathbf{P}_4 - \mathbf{P}_2) \\ 3(\mathbf{P}_1 - \mathbf{P}_3) \\ 3(\mathbf{P}_2 - \mathbf{P}_4) \end{bmatrix}. \tag{12.17}$$

Its solutions are $\mathbf{P}_1^t = (3/4, -3/4)$, $\mathbf{P}_2^t = (3/4, 3/4)$, $\mathbf{P}_3^t = (-3/4, 3/4)$, and $\mathbf{P}_4^t = (-3/4, -3/4)$, and the four spline segments are

$$\mathbf{P}_1(t) = (-1/2, 0)t^3 + (3/4, 3/4)t^2 + (3/4, -3/4)t,$$

$$\mathbf{P}_2(t) = (0,-1/2)t^3 + (-3/4,3/4)t^2 + (3/4,3/4)t + (1,0),$$
$$\mathbf{P}_3(t) = (1/2,0)t^3 + (-3/4,-3/4)t^2 + (-3/4,3/4)t + (1,1),$$
$$\mathbf{P}_4(t) = (0,1/2)t^3 + (3/4,-3/4)t^2 + (-3/4,-3/4)t + (0,1).$$

12.1.6 Nonuniform Cubic Splines

All the different types of cubic splines discussed so far assume that the parameter t varies in the interval $[0,1]$ in each segment. These types of cubic spline are therefore *uniform* or *normalized*. The *nonuniform cubic spline* is obtained by adding another parameter t_k to each spline segment and letting t vary in the interval $[0,t_k]$. Since there are $n-1$ spline segments connecting the n data points, this adds $n-1$ parameters to the curve, which makes it easier to fine-tune the shape of the curve. The nonuniform cubic splines are especially useful in cases where the data points are nonuniformly spaced. In regions where the points are closely spaced, the normalized cubic spline tends to develop loops and overshoots. In regions where the points are widely spaced, it tends to "cut corners," i.e., to be too tight. Careful selection of the t_k parameters can overcome these tendencies.

The calculation of the nonuniform cubic spline is based on that of the uniform version. We simply rewrite some of the basic equations, substituting t_k for 1 as the final value of t. We start with Equation (11.2) that becomes, for the first spline segment,

$$\begin{aligned}
\mathbf{a}\cdot 0^3 + \mathbf{b}\cdot 0^2 + \mathbf{c}\cdot 0 + \mathbf{d} &= \mathbf{P}_1,\\
\mathbf{a}(t_1)^3 + \mathbf{b}(t_1)^2 + \mathbf{c}(t_1) + \mathbf{d} &= \mathbf{P}_2,\\
3\mathbf{a}\cdot 0^2 + 2\mathbf{b}\cdot 0 + \mathbf{c} &= \mathbf{P}_1^t,\\
3\mathbf{a}(t_1)^2 + 2\mathbf{b}(t_1) + \mathbf{c} &= \mathbf{P}_2^t,
\end{aligned}$$

with solutions

$$\begin{aligned}
\mathbf{a} &= \frac{2(\mathbf{P}_1-\mathbf{P}_2)}{t_1^3} + \frac{\mathbf{P}_1^t}{t_1^2} + \frac{\mathbf{P}_2^t}{t_1^2},\\
\mathbf{b} &= \frac{3(\mathbf{P}_2-\mathbf{P}_1)}{t_1^2} - \frac{2\mathbf{P}_1^t}{t_1} - \frac{\mathbf{P}_2^t}{t_1},\\
\mathbf{c} &= \mathbf{P}_1^t,\\
\mathbf{d} &= \mathbf{P}_1.
\end{aligned} \tag{12.18}$$

Equation (11.4) now becomes

$$\mathbf{P}(t) = \left[\frac{2(\mathbf{P}_1-\mathbf{P}_2)}{t_1^3} + \frac{\mathbf{P}_1^t}{t_1^2} + \frac{\mathbf{P}_2^t}{t_1^2}\right]t^3 + \left[\frac{3(\mathbf{P}_2-\mathbf{P}_1)}{t_1^2} - \frac{2\mathbf{P}_1^t}{t_1} - \frac{\mathbf{P}_2^t}{t_1}\right]t^2 + \mathbf{P}_1^t t + \mathbf{P}_1. \tag{12.19}$$

Equation (12.4) becomes

$$\mathbf{P}_k^{tt}(t_k) = \mathbf{P}_{k+1}^{tt}(0) \qquad \text{or} \qquad 6\mathbf{a}_k \times t_k + 2\mathbf{b}_k = 6\mathbf{a}_{k+1}\times 0 + 2\mathbf{b}_{k+1}, \tag{12.20}$$

and Equation (12.5) is now

$$\begin{aligned} &2\left[\frac{3(\mathbf{P}_{k+1}-\mathbf{P}_k)}{t_k^2}-\frac{2\mathbf{P}_k^t}{t_k}-\frac{\mathbf{P}_{k+1}^t}{t_k}\right]+6t_k\left[\frac{2(\mathbf{P}_k-\mathbf{P}_{k+1})}{t_k^3}+\frac{\mathbf{P}_k^t}{t_k^2}+\frac{\mathbf{P}_{k+1}^t}{t_k^2}\right] \\ &\quad=2\left[\frac{3(\mathbf{P}_{k+2}-\mathbf{P}_{k+1})}{t_{k+1}^2}-\frac{2\mathbf{P}_{k+1}^t}{t_{k+1}}-\frac{\mathbf{P}_{k+2}^t}{t_{k+1}}\right]. \end{aligned} \tag{12.21}$$

Equation (12.6) now becomes

$$\begin{aligned} &t_{k+1}\mathbf{P}_k^t+2(t_k+t_{k+1})\mathbf{P}_{k+1}^t+t_k\mathbf{P}_{k+2}^t \\ &\quad=\frac{3}{t_kt_{k+1}}\left[t_k^2(\mathbf{P}_{k+2}-\mathbf{P}_{k+1})+t_{k+1}^2(\mathbf{P}_{k+1}-\mathbf{P}_k)\right]. \end{aligned} \tag{12.22}$$

This produces the new version of Equation (12.7)

$$\begin{aligned} &\underbrace{\begin{bmatrix} t_2 & 2(t_1+t_2) & t_1 & 0 & 0 & \cdots & 0 \\ 0 & t_3 & 2(t_2+t_3) & t_2 & 0 & \cdots & 0 \\ & & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & t_{n-1} & 2(t_{n-1}+t_{n-2}) & t_{n-2} \end{bmatrix}}_{n} \begin{bmatrix} \mathbf{P}_1^t \\ \mathbf{P}_2^t \\ \vdots \\ \mathbf{P}_n^t \end{bmatrix} \\ &\quad=\begin{bmatrix} \frac{3}{t_1t_2}\left[t_1^2(\mathbf{P}_3-\mathbf{P}_2)+t_2^2(\mathbf{P}_2-\mathbf{P}_1)\right] \\ \frac{3}{t_2t_3}\left[t_2^2(\mathbf{P}_4-\mathbf{P}_3)+t_3^2(\mathbf{P}_3-\mathbf{P}_2)\right] \\ \vdots \\ \frac{3}{t_{n-2}t_{n-1}}\left[t_{n-2}^2(\mathbf{P}_n-\mathbf{P}_{n-1})+t_{n-1}^2(\mathbf{P}_{n-1}-\mathbf{P}_{n-2})\right] \end{bmatrix}. \end{aligned} \tag{12.23}$$

This is again a system of $n-2$ equations in the n unknowns $\mathbf{P}_1^t$, $\mathbf{P}_2^t$,..., $\mathbf{P}_n^t$. After the user inputs the guessed or estimated values for the two extreme tangent vectors $\mathbf{P}_1^t$ and $\mathbf{P}_n^t$, this system can be solved, yielding the values of the remaining $n-2$ tangent vectors. Each of the $n-1$ spline segments can now be calculated by means of Equation (12.18) that is written here for the first segment in compact form

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}=\begin{pmatrix} 2/t_1^3 & -2/t_1^3 & 1/t_1^2 & 1/t_1^2 \\ -3/t_1^2 & 3/t_1^2 & -2/t_1 & -1/t_1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_1^t \\ \mathbf{P}_2^t \end{pmatrix}. \tag{12.24}$$

Notice how each of Equations (12.18) through (12.24) reduces to the corresponding original equation when all the t_i are set to 1. The nonuniform cubic spline can now be calculated in the following steps:

1. The user inputs the values of the two extreme tangent vectors and the values of the $n-1$ parameters t_k. The software sets up and solves Equation (12.23) to calculate the remaining tangent vectors.

2. The software sets up and solves Equation (12.24) $n-1$ times, once for each of the spline segments.

3. Each segment $\mathbf{P}_k(t)$ is plotted by varying t from 0 to t_k.

Before looking at an example, it is useful to try to understand the advantage of having the extra parameters t_k. Equation (12.18) shows that a large value of t_k for spline segment $\mathbf{P}_k(t)$ means small $\mathbf{a}$ and $\mathbf{b}$ coefficients (since t_k appears in the denominators), and hence a small second derivative $\mathbf{P}_k^{tt}(t) = 6\mathbf{a}_k + 2\mathbf{b}_k$ for that segment. Since the second derivative can be interpreted as the acceleration of the curve, we can predict that a large t_k will result in small overall acceleration for segment k. Thus, most of the segment will be close to a straight line. This is also easy to see when we substitute small $\mathbf{a}$ and $\mathbf{b}$ in $\mathbf{P}_k(t) = \mathbf{a}t^3 + \mathbf{b}t^2 + \mathbf{c}t + \mathbf{d}$. The dominant part of the segment becomes $\mathbf{c}t + \mathbf{d}$, which brings it close to linear. If the start and end directions of the segment are very different, the entire segment cannot be a straight line, so, in order to minimize its overall second derivative, the segment will end up consisting of two or three parts, each close to a straight line, with short, highly-curved corners connecting them (Figure 12.4). Such a geometry has a small overall second derivative. This knowledge is useful when designing curves, which is why the nonuniform cubic spline should not be dismissed as impractical. It may be the best method for certain curves.

Figure 12.4: Curves with Small Overall Second Derivative.

Example: The four points of Section 12.1.1 are used in this example. They are $\mathbf{P}_1 = (0,0)$, $\mathbf{P}_2 = (1,0)$, $\mathbf{P}_3 = (1,1)$, and $\mathbf{P}_4 = (0,1)$. We also select the same initial and final directions $\mathbf{P}_1^t = (1,-1)$ and $\mathbf{P}_4^t = (-1,-1)$. We decide to use $t_k = 2$ for each of the three spline segments to illustrate how large t_k values create a curve very different from the one of Section 12.1.1. Equation (12.23) becomes

$$\begin{bmatrix} t_2 & 2(t_1+t_2) & t_1 & 0 \\ 0 & t_3 & 2(t_2+t_3) & t_2 \end{bmatrix} \begin{bmatrix} (1,-1) \\ \mathbf{P}_2^t \\ \mathbf{P}_3^t \\ (-1,-1) \end{bmatrix} = \begin{bmatrix} \frac{3}{t_1 t_2}[t_1^2(\mathbf{P}_3 - \mathbf{P}_2) + t_2^2(\mathbf{P}_2 - \mathbf{P}_1)] \\ \frac{3}{t_2 t_3}[t_2^2(\mathbf{P}_4 - \mathbf{P}_3) + t_3^2(\mathbf{P}_3 - \mathbf{P}_2)] \end{bmatrix}.$$

For $t_1 = t_2 = t_3 = 2$, this yields $\mathbf{P}_2^t = (1/6, 1/2)$ and $\mathbf{P}_3^t = (-1/6, 1/2)$. Equation (12.24) is now written and solved three times:

$$\text{Segment 1} \quad \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} 2/t_1^3 & -2/t_1^3 & 1/t_1^2 & 1/t_1^2 \\ -3/t_1^2 & 3/t_1^2 & -2/t_1 & -1/t_1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} (0,0) \\ (1,0) \\ (1,-1) \\ (1/6,1/2) \end{bmatrix}.$$

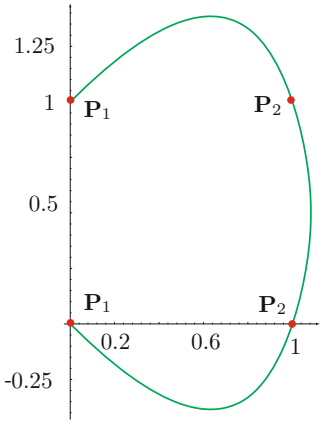
$$\text{Segment 2} \quad \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} 2/t_2^3 & -2/t_2^3 & 1/t_2^2 & 1/t_2^2 \\ -3/t_2^2 & 3/t_2^2 & -2/t_2 & -1/t_2 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} (1,0) \\ (1,1) \\ (1/6,1/2) \\ (-1/6,1/2) \end{bmatrix}.$$

$$\text{Segment 3} \quad \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} 2/t_3^3 & -2/t_3^3 & 1/t_3^2 & 1/t_3^2 \\ -3/t_3^2 & 3/t_3^2 & -2/t_3 & -1/t_3 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} (1,1) \\ (0,1) \\ (-1/6,1/2) \\ (-1,-1) \end{bmatrix}.$$

This yields the coefficients for the three spline segments:

$$\begin{aligned} \mathbf{P}_1(t) &= (1/24,-1/8)t^3 + (-1/3,3/4)t^2 + (1,-1)t, \\ \mathbf{P}_2(t) &= (0,0)t^3 + (-1/12,0)t^2 + (1/6,1/2)t + (1,0), \\ \mathbf{P}_3(t) &= -(1/24,1/8)t^3 + (-1/12,0)t^2 + (-1/6,1/2)t + (1,1). \end{aligned}$$

The result is shown in Figure 12.5. It should be compared with the uniform curve of Figure 12.2 that's based on the same four points. (Recall that t varies from 0 to 2 in each of the segments above.)




```
(* Nonuniform cubic spline example *)
C1:=ParametricPlot[{1/24,-1/8}t^3+{-1/3,3/4}t^2+{1,-1}t, {t,0,2}];
C2:=ParametricPlot[{-1/12,0}t^2+{1/6,1/2}t+{1,0}, {t,0,2}];
C3:=ParametricPlot[-{1/24,1/8}t^3+{-1/12,0}t^2+{-1/6,1/2}t+{1,1}, {t,0,2}];
Show[C1, C2, C3, PlotRange->All, AspectRatio->Automatic]
```

Figure 12.5: A Nonuniform Cubic Spline Example.

12.1.7 Fair Cubic Splines

The term "fair curve" refers to a spline curve where each segment is close to a circular arc. Such a curve does not have very flat or very curved segments (e.g., segments with loops) and is generally considered pleasing to the eye. It is useful in artistic applications and in font design, where the aim is to get beautiful curves rather than a precise fit to a given set of points. The approach presented here is based on [Manning 74] and is illustrated by Figure 12.6. In Figure 12.6a we see four Hermite segments connecting the same two endpoints, all with 45° tangent vectors. The difference between them is the magnitude of the vectors. If we denote the distance between the points l, then Figure 12.6a(i)–(iii) shows curves whose tangent vectors have magnitudes smaller than, equal to, and greater than l, respectively. In Figure 12.6a(iv), the left tangent has magnitude $> l$ and the right one $< l$, resulting in a curve that's "pulled" to the right. Of these four, curve (ii) can be considered "fair," since it is close to a special circular arc, namely the arc that passes through the endpoints of the curve *at the same angle* as the tangent vectors.

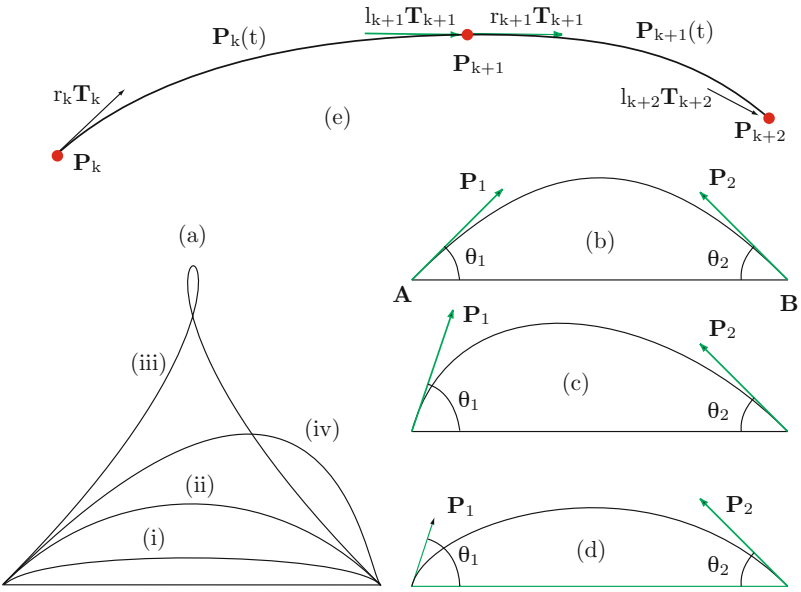


Figure 12.6: Varying Tangent Vector Magnitudes.

◇ **Exercise 12.7:** Two given points $\mathbf{P}_1$ and $\mathbf{P}_2$ define a straight segment $\mathbf{P}_1 \rightarrow \mathbf{P}_2$. Your task is to construct a "fair" Hermite segment with $\mathbf{P}_1$ and $\mathbf{P}_2$ as its endpoints. Here is what you need to show. Imagine the circular arc that passes through $\mathbf{P}_1$ and $\mathbf{P}_2$ and makes angles θ with the line $\mathbf{P}_1 \rightarrow \mathbf{P}_2$. Show that if the center point $\mathbf{P}(0.5)$ of

the Hermite segment is located on this circular arc, then the magnitudes of the tangent vectors satisfy

$$|\mathbf{P}_1^t| = |\mathbf{P}_2^t| = \frac{2|\mathbf{P}_2 - \mathbf{P}_1|}{1 + \cos\theta}. \tag{12.25}$$

Figure 12.6b shows a Hermite segment with tangent vectors whose magnitudes satisfy Equation (12.25). The curve is therefore close to the special circular arc mentioned earlier. In Figure 12.6c, the left tangent vector $\mathbf{P}_1$ has been rotated (but has the same magnitude), thereby increasing angle θ_1 and pulling the curve to the left. Figure 12.6d shows how the curve can (approximately) be returned to its former shape by shortening $\mathbf{P}_1$. This suggests a way to build a complete cubic spline where every segment is fair. At every internal point $\mathbf{P}_{k+1}$, the "incoming" and "outgoing" tangent vectors should go in the same direction but should have *different magnitudes*. Figure 12.6e shows the terminology used. The incoming tangent vector at interior point $\mathbf{P}_{k+1}$ is $\mathbf{P}_k^t(1) = l_{k+1}\mathbf{T}_{k+1}$, where all $\mathbf{T}_i$ are unit vectors and l_{k+1} is a scalar (the magnitude of the tangent). Similarly, the outgoing tangent vector at $\mathbf{P}_{k+1}$ is $\mathbf{P}_{k+1}^t(0) = r_{k+1}\mathbf{T}_{k+1}$, the same unit vector, but with a different magnitude r_{k+1}.

Equation (12.25) can be considered the definition of a fair Hermite segment in the special case $\theta_1 = \theta_2$. The previous paragraph suggests a way to extend it to cases where $\theta_1 \neq \theta_2$. We need to express the magnitudes of the tangents $\mathbf{P}_1^t$ and $\mathbf{P}_2^t$ in terms of the endpoints and the angles θ_1 and θ_2 such that $\theta_1 > \theta_2$ will result in $|\mathbf{P}_1^t| < |\mathbf{P}_2^t|$. One way to achieve this effect is to define

$$r_1 = |\mathbf{P}_1^t| = \frac{2|\mathbf{P}_2 - \mathbf{P}_1|}{1 + \alpha\cos\theta_2 + (1-\alpha)\cos\theta_1}, \tag{12.26}$$

$$l_2 = |\mathbf{P}_2^t| = \frac{2|\mathbf{P}_2 - \mathbf{P}_1|}{1 + \alpha\cos\theta_1 + (1-\alpha)\cos\theta_2}, \tag{12.27}$$

where α is a parameter in the range $[0, 1]$ to be determined by the user. This, of course, is not the only way to define a fair curve, but this definition has two useful properties:

1. For $\theta_1 = \theta_2$, Equations (12.26) and (12.27) reduce to Equation (12.25).
2. If $\theta_1 > \theta_2$, then $\cos\theta_1 < \cos\theta_2$ (for fair curves, we can assume angles between $0°$ and $90°$). In order to achieve $|\mathbf{P}_1^t| < |\mathbf{P}_2^t|$, we need a situation where $(1-\alpha)\cos\theta_1 < \alpha\cos\theta_2$. This is satisfied when $1 - \alpha \leq \alpha$, i.e., when $0.5 \leq \alpha \leq 1$. [Manning 74] suggests $\alpha = 2/3$, but it seems that α should be left as a user-defined parameter, especially for closed fair curves, which are discussed later, where there is no other parameter for the user to adjust.

The condition for slope continuity at the $n-2$ interior points is thus written $h_{k+1}\mathbf{P}_k^t(1) = \mathbf{P}_{k+1}^t(0)$, where h_{k+1} is the ratio of the tangent vectors' magnitudes. We denote by $\mathbf{T}_k$ a unit tangent vector, so we can write $h_{k+1}l_{k+1}\mathbf{T}_{k+1} = r_{k+1}\mathbf{T}_{k+1}$ or $h_{k+1} = r_{k+1}/l_{k+1}$ for $k = 1, 2, \ldots, n-2$. The two tangents $\mathbf{P}_k^t(1)$ and $\mathbf{P}_{k+1}^t(0)$ go in the same direction, so h_{k+1} must be positive.

The quantities $l_2, l_3, \ldots, l_n$ and $r_1, r_2, \ldots, r_{n-1}$ make a total of $2n-2$ unknowns that have to be calculated. The equations to calculate them are obtained by the requirement that the curvatures of the spline segments be equal at the $n-2$ interior points. When

Equation (8.19) is used for the curvature, this requirement becomes

$$\frac{d^2\mathbf{P}_k(1)}{ds^2} = \frac{d^2\mathbf{P}_{k+1}(0)}{ds^2}$$

or

$$\frac{\mathbf{P}_k^{tt}(1)}{\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1)} - \frac{\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^{tt}(1)}{\left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1)\right)^2}\mathbf{P}_k^t(1) = \frac{\mathbf{P}_{k+1}^{tt}(0)}{\mathbf{P}_{k+1}^t(0)\bullet\mathbf{P}_{k+1}^t(0)} - \frac{\mathbf{P}_{k+1}^t(0)\bullet\mathbf{P}_{k+1}^{tt}(0)}{\left(\mathbf{P}_{k+1}^t(0)\bullet\mathbf{P}_{k+1}^t(0)\right)^2}\mathbf{P}_{k+1}^t(0).$$

This is simplified by substituting $h_{k+1}\mathbf{P}_k^t(1) = \mathbf{P}_{k+1}^t(0)$ and multiplying by $(\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1))$, yielding

$$\mathbf{P}_k^{tt}(1) - \frac{\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^{tt}(1)}{\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1)}\mathbf{P}_k^t(1) = \frac{\mathbf{P}_{k+1}^{tt}(0)}{h_{k+1}^2} - \frac{h_{k+1}^2\left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_{k+1}^{tt}(0)\right)}{h_{k+1}^4\left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1)\right)}\mathbf{P}_k^t(1)$$

or

$$h_{k+1}^2\mathbf{P}_k^{tt}(1) - \mathbf{P}_{k+1}^{tt}(0) = \frac{h_{k+1}^2\left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^{tt}(1)\right) - \left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_{k+1}^{tt}(0)\right)}{\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1)}\mathbf{P}_k^t(1). \quad (12.28)$$

Equation (12.28) can be written

$$h_{k+1}^2\mathbf{P}_k^{tt}(1) - \mathbf{P}_{k+1}^{tt}(0) = M_{k+1}\mathbf{P}_k^t(1), \quad (12.29)$$

where the quantity M_{k+1} that is defined by

$$M_{k+1} \stackrel{\text{def}}{=} \frac{h_{k+1}^2\left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^{tt}(1)\right) - \left(\mathbf{P}_k^t(1)\bullet\mathbf{P}_{k+1}^{tt}(0)\right)}{\mathbf{P}_k^t(1)\bullet\mathbf{P}_k^t(1)},$$

is a scalar combining all the scalar quantities from the right-hand side of Equation (12.28).

> Fair: To draw and adjust (the lines of a ship's hull being designed) to produce regular surfaces of the correct form.
>
> — A dictionary definition.

The next step is to replace the two second derivatives on the left side of Equation (12.29) with expressions containing the unknowns l_k, r_k, and $\mathbf{T}_k$ (and, perhaps, some known quantities, such as the points $\mathbf{P}_k$). This will provide equations whose solutions will yield the tangent vectors $\mathbf{P}_k^t$ at all the points. We start with the second derivative of a PC $\mathbf{P}^{tt}(t) = 6\mathbf{a}t + 2\mathbf{b}$ (Equation (12.3)), where $\mathbf{a}$ and $\mathbf{b}$ are given by Equation (11.3). The two second derivatives used in Equation (12.29) are (see also Figure 12.6e) as follows:

1. From segment $\mathbf{P}_k(t)$,

$$\begin{aligned}\mathbf{P}_k^{tt}(1) &= 6\mathbf{a}_k \times 1 + 2\mathbf{b}_k \\ &= 6[2(\mathbf{P}_k - \mathbf{P}_{k+1}) + \mathbf{P}_k^t + \mathbf{P}_{k+1}^t] + 2[3(\mathbf{P}_{k+1} - \mathbf{P}_k) - 2\mathbf{P}_k^t - \mathbf{P}_{k+1}^t] \\ &= -6(\mathbf{P}_{k+1} - \mathbf{P}_k) + 2\mathbf{P}_k^t + 4\mathbf{P}_{k+1}^t \\ &= -6(\mathbf{P}_{k+1} - \mathbf{P}_k) + 2r_k\mathbf{T}_k + 4l_{k+1}\mathbf{T}_{k+1}. \end{aligned} \tag{12.30}$$

2. From segment $\mathbf{P}_{k+1}(t)$,

$$\begin{aligned}\mathbf{P}_{k+1}^{tt}(0) &= 6\mathbf{a}_{k+1} \times 0 + 2\mathbf{b}_{k+1} \\ &= 2[3(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}) - 2\mathbf{P}_{k+1}^t - \mathbf{P}_{k+2}^t] \\ &= 6(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}) - 4r_{k+1}\mathbf{T}_{k+1} - 2l_{k+2}\mathbf{T}_{k+2}. \end{aligned} \tag{12.31}$$

Substituting Equations (12.30) and (12.31) into Equation (12.29) and taking into account that $h_k = r_k/l_k$ for $k = 1, 2, \ldots, n-2$, we get

$$\begin{aligned}&\frac{r_{k+1}^2}{l_{k+1}^2}\left[-6(\mathbf{P}_{k+1} - \mathbf{P}_k) + 2r_k\mathbf{T}_k + 4l_{k+1}\mathbf{T}_{k+1}\right] \\ &\quad - \left[6(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}) - 4r_{k+1}\mathbf{T}_{k+1} - 2l_{k+2}\mathbf{T}_{k+2}\right] \\ &= M_{k+1}\mathbf{P}_k^t(1).\end{aligned}$$

Multiplying by l_{k+1}^2 and simplifying yields

$$\begin{aligned}&r_{k+1}^2[-6(\mathbf{P}_{k+1} - \mathbf{P}_k) + 2r_k\mathbf{T}_k + 4l_{k+1}\mathbf{T}_{k+1}] \\ &\quad - l_{k+1}^2[6(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}) - 4r_{k+1}\mathbf{T}_{k+1} - 2l_{k+2}\mathbf{T}_{k+2}] \\ &= l_{k+1}^2 M_{k+1}\mathbf{P}_k^t(1) = l_{k+1}^3 M_{k+1}\mathbf{T}_{k+1}\end{aligned}$$

or

$$\begin{aligned}&-6r_{k+1}^2(\mathbf{P}_{k+1} - \mathbf{P}_k) + 2r_{k+1}^2 r_k\mathbf{T}_k + 4r_{k+1}^2 l_{k+1}\mathbf{T}_{k+1} \\ &\quad - 6l_{k+1}^2(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}) + 4l_{k+1}^2 r_{k+1}\mathbf{T}_{k+1} + 2l_{k+1}^2 l_{k+2}\mathbf{T}_{k+2} \\ &= l_{k+1}^3 M_{k+1}\mathbf{T}_{k+1}.\end{aligned}$$

Dividing by 2 and changing signs results in

$$\begin{aligned}&3r_{k+1}^2(\mathbf{P}_{k+1} - \mathbf{P}_k) - 3l_{k+1}^2(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}) - r_{k+1}^2 r_k\mathbf{T}_k - l_{k+1}^2 l_{k+2}\mathbf{T}_{k+2} \\ &= -\frac{1}{2}l_{k+1}^3 M_{k+1}\mathbf{T}_{k+1} - 2r_{k+1}^2 l_{k+1}\mathbf{T}_{k+1} + 2l_{k+1}^2 r_{k+1}\mathbf{T}_{k+1} \stackrel{\text{def}}{=} L_{k+1}\mathbf{T}_{k+1},\end{aligned} \tag{12.32}$$

where L_{k+1} is defined as everything that multiplies $\mathbf{T}_{k+1}$ on the right-hand side of Equation (12.32).

Equation (12.32) is a vector equation that can be written $n-2$ times, for $k = 1, 2, \ldots, n-2$. It therefore yields $2(n-2)$ or $3(n-2)$ equations, depending on whether

the original data points $\mathbf{P}_k$ are two- or three-dimensional. However, more equations are needed. To derive them, we turn our attention to Figure 12.7, which shows the relation between a unit tangent vector $\mathbf{T}$ and the angle θ between it and the line connecting the two endpoints

$$\cos\theta_k = \mathbf{T}_k \bullet \frac{\mathbf{P}_{k+1} - \mathbf{P}_k}{|\mathbf{P}_{k+1} - \mathbf{P}_k|}.$$

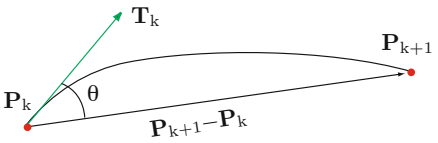


Figure 12.7: Relation Between θ and $\mathbf{T}$.

For segment $\mathbf{P}_k(t)$, we get from Equation (12.27)

$$\begin{aligned} l_{k+1} = |\mathbf{P}^t_{k+1}| &= \frac{2|\mathbf{P}_{k+1} - \mathbf{P}_k|}{1 + \alpha\mathbf{T}_k \bullet \frac{\mathbf{P}_{k+1}-\mathbf{P}_k}{|\mathbf{P}_{k+1}-\mathbf{P}_k|} + (1-\alpha)\mathbf{T}_{k+1} \bullet \frac{\mathbf{P}_{k+1}-\mathbf{P}_k}{|\mathbf{P}_{k+1}-\mathbf{P}_k|}} \\ &= \frac{2|\mathbf{P}_{k+1} - \mathbf{P}_k|}{1 + [\alpha\mathbf{T}_k + (1-\alpha)\mathbf{T}_{k+1}] \bullet \frac{\mathbf{P}_{k+1}-\mathbf{P}_k}{|\mathbf{P}_{k+1}-\mathbf{P}_k|}}. \end{aligned} \tag{12.33}$$

From Equation (12.26)

$$\begin{aligned} r_k = |\mathbf{P}^t_k| &= \frac{2|\mathbf{P}_{k+1} - \mathbf{P}_k|}{1 + \alpha\mathbf{T}_{k+1} \bullet \frac{\mathbf{P}_{k+1}-\mathbf{P}_k}{|\mathbf{P}_{k+1}-\mathbf{P}_k|} + (1-\alpha)\mathbf{T}_k \bullet \frac{\mathbf{P}_{k+1}-\mathbf{P}_k}{|\mathbf{P}_{k+1}-\mathbf{P}_k|}} \\ &= \frac{2|\mathbf{P}_{k+1} - \mathbf{P}_k|}{1 + [\alpha\mathbf{T}_{k+1} + (1-\alpha)\mathbf{T}_k] \bullet \frac{\mathbf{P}_{k+1}-\mathbf{P}_k}{|\mathbf{P}_{k+1}-\mathbf{P}_k|}}. \end{aligned} \tag{12.34}$$

Equations (12.33) and (12.34) are scalar. They can be written for each of the $n-1$ spline segments (i.e., for $k = 1, 2, \ldots, n-1$), providing $2n-2$ additional equations. The total number of equations is thus $(2n-4)+(2n-2) = 4n-6$ for the two-dimensional case and $(3n-6)+(2n-2) = 5n-8$ for the three-dimensional case. The unknowns are

$$l_2, l_3, \ldots, l_n, r_1, r_2, \ldots, r_{n-1}, L_2, L_3, \ldots, L_{n-1} \text{ and } \mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_n,$$

a total of $(n-1)+(n-1)+(n-2)+n = 4n-4$ unknowns. It is important to realize that in the two-dimensional case, each unit vector $\mathbf{T}_k$ is equivalent to only one unknown (since it can be written $(\cos\theta, \sin\theta)$ for some angle θ). In the three-dimensional case, each $\mathbf{T}_k$ is equivalent to two unknowns, so the number of unknowns in that case is $5n-4$. We thus end up with $4n-6$ equations and $4n-4$ unknowns (in the two-dimensional case) or $5n-8$ equations and $5n-4$ unknowns (in the three-dimensional case). We seem to be two or four equations short, but we already know from past experience with cubic

splines that this apparent problem can be turned to our advantage. The solution, of course, is to ask the user to supply the values of the two extreme unit tangents $\mathbf{T}_1$ and $\mathbf{T}_n$. This provides the two or four necessary values to bring the number of unknowns down to the number of equations. This, together with the free parameter α, turns fair cubic splines into an interactive method.

The discussion so far has assumed an open curve, but closed curves are also useful in practical problems. In fact, the original fair cubic splines were developed by [Manning 74] to help design insoles for shoes; a useful, practical example of a closed curve. A closed curve does not have endpoints; every point can be considered interior. Equation (12.32) can thus be written n times, providing $2n$ or $3n$ equations. A closed curve also requires n segments, instead of $n-1$. Each of Equations (12.33) and (12.34) can thus be written n times, once for each segment. The total number of equations is therefore $4n$ or $5n$. The unknowns are

$$l_1, l_2, \ldots, l_n, r_1, r_2, \ldots, r_n, L_1, L_2, \ldots, L_n \text{ and } \mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_n,$$

a total of $4n$ or $5n$ unknowns. A closed curve can thus be computed based on the n data points, without any additional user input. If it is unsatisfactory, it can be edited by moving the points or changing the value of the parameter α.

One drawback of this method is that the equations are not linear. This makes it complicated to solve them and it also means that there may be either no solutions or several different solutions. A simple, iterative algorithm for solving the equations is the following:

1. Guess reasonable initial values for the unit tangents $\mathbf{T}_k$. A good idea is to set $\mathbf{T}_k$ in the direction $\mathbf{P}_{k+1} - \mathbf{P}_k$.

2. Using these values, solve Equations (12.33) and (12.34) to calculate initial values for all the l_k and r_k unknowns.

3. Substitute the current values of $\mathbf{T}_k$, l_k, and r_k in the left-hand side of Equation (12.32) to obtain better values for the products $L_k\mathbf{T}_k$. Once such a product is known, $\mathbf{T}_k$ can be calculated since its magnitude is 1.

4. Repeat steps 2 and 3 until the process converges (i.e., until none of l_k, r_k, and $\mathbf{T}_k$ changes between consecutive iterations by more than a preset threshold).

As has been mentioned earlier, this process is not guaranteed to converge, or it may converge to one of several possible solutions. Another difficulty has to do with the constants L_k. We never need their magnitudes, but their signs are important, since they give the sense of direction at the interior points. One way to handle the L_k is to keep the angle between the two vectors $(\mathbf{P}_{k+2} - \mathbf{P}_{k+1})$ and $(\mathbf{P}_{k+1} - \mathbf{P}_k)$ small for every interior point $\mathbf{P}_{k+1}$ (see Figure 12.6e). This will produce individual spline segments, none of which turns too much, resulting in tangents $\mathbf{T}_k$, $\mathbf{T}_{k+1}$, and $\mathbf{T}_{k+2}$ that don't differ much in direction. Such a situation corresponds to $L_k > 0$ for every k. An alternative is to keep the sign of each of the new products $L_k\mathbf{T}_k$ obtained in step 3 identical to the sign of $\mathbf{T}_k$ used earlier in that step. This guarantees that none of the new $\mathbf{T}_k$ will change much in direction during an iteration.

12.2 The Akima Spline

In the late 1960s, Hiroshi Akima was looking for a cubic spline curve that would look natural and smooth, like a curve drawn intuitively by hand. The result turned out to be successful and the Akima spline has become the algorithm of choice for several illustration and drawing applications. The Akima algorithm has been published in [Akima 70], [Akima 78], and [Akima 91], with Matlab code available at [Akima-matlab 10].

The curves produced by the Akima spline are often very similar to those obtained by cubic splines, but are less prone to wiggling. This feature is especially noticeable when one data point is an outlier, as illustrated by Figure 12.8. The figure shows several data points and it is obvious that one of them lies at an unexpected location, far from the other points. A cubic spline has been computed for the points and it is clear that it wiggles between points. The Akima spline, on the other hand, is smooth overall and resembles a curve we expect a person to draw for the given points.

Definition of outlier
A value that "lies outside" (is much smaller or larger than) most of the other values in a set of data. A person who lives away from his place of work. An observation that lies outside the overall pattern of a distribution. The presence of an outlier normally indicates some sort of problem. A point which a data set is better off without (this is an embarrassing secret of the statistical trade).

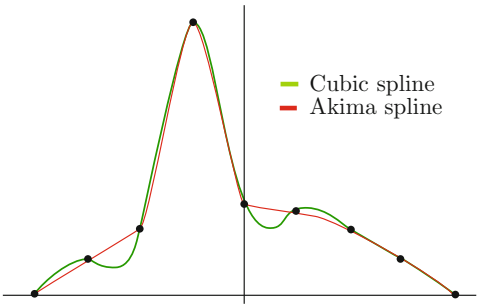


Figure 12.8: Two Splines at an Outlier.

The original Akima spline was developed for explicit curves (i.e., curves expressed as $y = f(x)$, also referred to as single-valued functions). Each spline segment from point x_i to point x_{i+1} is a cubic polynomial, and the main idea is to compute the slope $s_i \stackrel{\text{def}}{=} f'(x_i)$ of the segment at point x_i explicitly, by means of a simple expression that depends on the point, its two immediate predecessors and its two immediate successors. The method is therefore local, and moving a point affects at most five spline segments. (This explains its usefulness when some data points are outliers. It also implies that the

number of data points must be at least five.) Another notable feature is the absence of the large system of equations, which is the cornerstone of the traditional cubic splines. A possible shortcoming is the fact that the second derivatives of the spline segments are not continuous at the data points.

Figure 12.9 shows two examples of the construction that is used to compute the slope. Five data points, numbered 1 through 5, are shown. The points are connected with straight segments (secants) whose slopes are denoted by m_1 through m_4. Thus, for example, $m_1 = (y_2 - y_1)/(x_2 - x_1)$. The desired slope at point 3 is also shown, as a short straight segment.

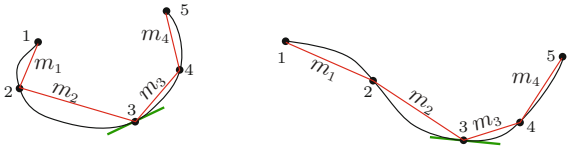


Figure 12.9: Construction for Computing the Slope.

The chief innovation of the Akima algorithm is the expression for computing the slope at point 3

$$s_3 = \frac{|m_4 - m_3|m_2 + |m_2 - m_1|m_3}{|m_4 - m_3| + |m_2 - m_1|}. \tag{12.35}$$

In reference [Akima 70], the developer of this method, Hiroshi Akima, shows why this expression makes sense. Figure 12.10a again shows five points. The straight segments $\overline{12}$, $\overline{23}$, $\overline{43}$, and $\overline{54}$ are extended to determine the locations of auxiliary points A and B. Segment $\overline{CD}$ is the desired slope at point 3, and it is determined by satisfying the relation

$$\left|\frac{\overline{2C}}{\overline{CA}}\right| = \left|\frac{\overline{4D}}{\overline{DB}}\right|.$$

About a dozen steps (not listed here) are needed to arrive from this relation to Equation (12.35).

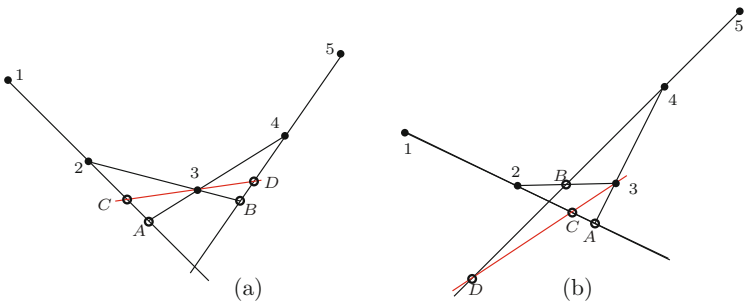


Figure 12.10: Construction for Determining the Slope.

Notice that in Figure 12.10b the two segments $\overline{4D}$ and $\overline{DB}$ seem to go in "opposite" directions, which is why the relation above uses absolute values.

Equation (12.35) implies that the slope at point 3 depends only on the slopes of the four secants. It is independent of the lengths of the secants and is invariant under a linear transformation of the coordinate system. It is also obvious that the equation makes sense in the following special cases: If $m_1 = m_2$ and $m_3 \neq m_4$, then $s = m_2$. If $m_3 = m_4$ and $m_1 \neq m_2$, then $s = m_3$. If $m_2 = m_3$, then $s = m_2 = m_3$. However, in the special case $m_1 = m_2 \neq m_3 = m_4$, Equation (12.35) is indeterminate and the slope is simply defined as the average $(m_2 + m_3)/2$.

Once we accept Equation (12.35), it is applied to points (x_i, y_i) and (x_{i+1}, y_{i+1}) to compute slopes s_i and s_{i+1}. These six numbers (four coordinates and two slopes) are then used to compute the cubic polynomial segment from x_i to x_{i+1}. The polynomial itself has the form

$$y_i = p_0 + p_1(x - x_i) + p_2(x - x_i)^2 + p_3(x - x_i)^3, \quad \text{where} \quad x_i \le x \le x_{i+1}, \tag{12.36}$$

and it is easy to show that the four parameters p_k are expressed in terms of the six numbers as

$$p_0 = y_i, \tag{12.37}$$
$$p_1 = s_i, \tag{12.38}$$
$$p_2 = \big[3(y_{i+1} - y_i)/(x_{i+1} - x_i) - 2s_i - s_{i+1}\big]/(x_{i+1} - x_i), \tag{12.39}$$
$$p_3 = \big[s_i + s_{i+1} - 2(y_{i+1} - y_i)/(x_{i+1} - x_i)\big]/(x_{i+1} - x_i)^2. \tag{12.40}$$

◇ **Exercise 12.8:** Only subscripts i and $i+1$ appear in Equations (12.37) through (12.40), but the slope at each data point i depends on the point and its four near neighbors. Where are the subscripts of the other neighbors?

Computing and drawing an Akima spline for a set of n data points is done in $n - 1$ iterations. For each pair of consecutive points, the four polynomial coefficients are computed and the curve segment, Equation (12.36), is drawn. While experimenting with the curve, it is a good idea to save the four polynomial coefficients of each segment. If the user decides to move a data point, the software has to recompute the coefficients of at most five polynomial segments.

The next point to consider is the computation of the first two and last two spline segments. The developer of this method, Hiroshi Akima, adopts the following idea. To compute the first segment, from point 1 to point 2, assume two imaginary points, 0 and -1, to the left of point 1, and compute slopes m_{-1} and m_0. To compute the last segment, assume two more points $n + 1$ and $n + 2$, and compute slopes m_n and m_{n+1}. The computations are done as follows. For a non-periodic curve, $m_{-1} = 3m_1 - 2m_2$, $m_0 = 2m_1 - m_2$, $m_n = 2m_{n-1} - m_{n-2}$, and $m_{n+1} = 3m_{n-1} - 2m_{n-2}$. For a periodic curve, $m_{-1} = m_{n-2}$, $m_0 = m_{n-1}$, $m_n = m_1$, and $m_{n+1} = m_2$.

12.3 The Quadratic Spline

The cubic spline curve is useful in certain practical applications, which raises the question of splines of different degrees based on the same concepts. It turns out that splines of degrees higher than 3 are useful only for special applications because they are more computationally intensive and tend to have many undesirable inflection points (i.e., they tend to wiggle excessively). Splines of degree 1 are, of course, just straight segments connected to form a polyline, but quadratic (degree-2) splines can be useful in certain applications. Such a spline is easy to derive and to compute. Each spline segment is a quadratic polynomial, i.e., a parabolic arc, so it results in fewer oscillations in the curve. On the other hand, quadratic spline segments connect with at most C^1 continuity because their second derivative is a constant. Thus, a quadratic spline curve may not be as tight as a cubic spline that passes through the same points.

The quadratic spline curve is derived in this section based on the variant Hermite segment of Section 11.7. Each segment $\mathbf{P}_i(t)$ is therefore a quadratic polynomial defined by its two endpoints $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ and by its start tangent vector $\mathbf{P}_i^t$. Equation (11.33) shows that the end tangent of such a segment is $\mathbf{P}_i^t(1) = 2(\mathbf{P}_{i+1} - \mathbf{P}_i) - \mathbf{P}_i^t$. The first two spline segments are

$$\begin{aligned}\mathbf{P}_1(t) &= (\mathbf{P}_2 - \mathbf{P}_1 - \mathbf{P}_1^t)t^2 + \mathbf{P}_1^t t + \mathbf{P}_1,\\ \mathbf{P}_2(t) &= (\mathbf{P}_3 - \mathbf{P}_2 - \mathbf{P}_2^t)t^2 + \mathbf{P}_2^t t + \mathbf{P}_2.\end{aligned}$$

At their joint point $\mathbf{P}_2$ they have the tangent vectors $\mathbf{P}_1^t(1) = 2(\mathbf{P}_2 - \mathbf{P}_1) - \mathbf{P}_1^t$ and $\mathbf{P}_2^t(0) = \mathbf{P}_2^t$. In order to achieve C^1 continuity we have to have the boundary condition $\mathbf{P}_1^t(1) = \mathbf{P}_2^t(0)$ or $2(\mathbf{P}_2 - \mathbf{P}_1) - \mathbf{P}_1^t = \mathbf{P}_2^t$. This equation can be written $\mathbf{P}_1^t + \mathbf{P}_2^t = 2(\mathbf{P}_2 - \mathbf{P}_1)$, and when duplicated $n-1$ times, for the points $\mathbf{P}_1$ through $\mathbf{P}_{n-1}$, the result is

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 1 & 1 & 0 & \cdots & 0 & 0\\ & & & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix}}_{n} \begin{bmatrix} \mathbf{P}_1^t\\ \mathbf{P}_2^t\\ \vdots\\ \mathbf{P}_n^t \end{bmatrix} = 2 \begin{bmatrix} \mathbf{P}_2 - \mathbf{P}_1\\ \mathbf{P}_3 - \mathbf{P}_2\\ \vdots\\ \mathbf{P}_n - \mathbf{P}_{n-1} \end{bmatrix}. \qquad (n-1) \tag{12.41}$$

As with the cubic spline, there are more unknowns than equations (n unknowns and $n-1$ equations), and the standard technique is to ask the user to provide a value for one of the unknown tangent vectors, normally $\mathbf{P}_1^t$.

Example: We select the four points of Section 12.1.1, namely $\mathbf{P}_1 = (0,0)$, $\mathbf{P}_2 = (1,0)$, $\mathbf{P}_3 = (1,1)$, and $\mathbf{P}_4 = (0,1)$. We also select the same start tangent $\mathbf{P}_1^t = (1,-1)$. Equation (12.41) becomes

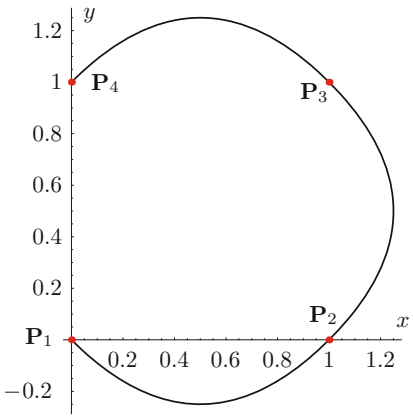
$$\begin{pmatrix} 1 & 1 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{P}_1^t\\ \mathbf{P}_2^t\\ \mathbf{P}_3^t\\ \mathbf{P}_4^t \end{pmatrix} = 2 \begin{pmatrix} \mathbf{P}_2 - \mathbf{P}_1\\ \mathbf{P}_3 - \mathbf{P}_2\\ \mathbf{P}_4 - \mathbf{P}_3 \end{pmatrix} = \begin{pmatrix} (2,0)\\ (0,2)\\ (-2,0) \end{pmatrix},$$

with solutions $\mathbf{P}_2^t = (1,1)$, $\mathbf{P}_3^t = (-1,1)$, and $\mathbf{P}_4^t = (-1,-1)$. The three spline segments

become

$$\begin{aligned}
\mathbf{P}_1(t) &= (\mathbf{P}_2 - \mathbf{P}_1 - \mathbf{P}_1^t)t^2 + \mathbf{P}_1^t t + \mathbf{P}_1 = (t, t^2 - t),\\
\mathbf{P}_2(t) &= (\mathbf{P}_3 - \mathbf{P}_2 - \mathbf{P}_2^t)t^2 + \mathbf{P}_2^t t + \mathbf{P}_2 = (-t^2 + t + 1, t),\\
\mathbf{P}_3(t) &= (\mathbf{P}_4 - \mathbf{P}_3 - \mathbf{P}_3^t)t^2 + \mathbf{P}_3^t t + \mathbf{P}_3 = (-t + 1, -t^2 + t + 1).
\end{aligned}$$

Their tangent vectors are $\mathbf{P}_1^t(t) = (1, 2t-1)$, $\mathbf{P}_2^t(t) = (-2t+1, 1)$, and $\mathbf{P}_3^t(t) = (-1, -2t+1)$. It is easy to see that $\mathbf{P}_1^t(1) = \mathbf{P}_2^t(0) = (1, 1)$ and $\mathbf{P}_2^t(1) = \mathbf{P}_3^t(0) = (-1, 1)$. Also, the end tangent of the entire curve is $\mathbf{P}_3^t(1) = (-1, -1)$, the same as for the cubic case. The complete spline curve is shown in Figure 12.11.




```
(*quadratic spline example*)
C1:=ParametricPlot[{t,t^2-t},{t,0,1}];
C2:=ParametricPlot[{-t^2+t+1,t},{t,0,1}];
C3:=ParametricPlot[{-t+1,-t^2+t+1},{t,0,1}];
C4=Graphics[{Red, AbsolutePointSize[6],Point[{0,0}],
 Point[{1,0}],Point[{1,1}],Point[{0,1}]}];
Show[C1,C2,C3,C4,PlotRange->All, AspectRatio->Automatic]
```

Figure 12.11: A Quadratic Spline Example.

12.4 The Quintic Spline

The derivation of the cubic spline is based on the requirement (boundary condition) that the second derivatives of the individual segments be equal at the interior points. This produces $n-2$ equations to compute the first derivatives, but makes it impossible to control the values of the second derivatives. In cases where the designer wants to specify the values of the second derivatives, higher-degree polynomials must be used. A degree-5 (quintic) polynomial is a natural choice. Section 11.3 discusses the similar case of the quintic Hermite segment.

The approach to the quintic spline is similar to that of the cubic spline. The spline is a set of $n-1$ segments, each a quintic polynomial, so we have to compute the coefficients of each segment from the boundary conditions. A general quintic spline segment from point $\mathbf{P}_k$ to point $\mathbf{P}_{k+1}$ is given by Equation (11.16), duplicated here

$$\mathbf{P}_k(t) = \mathbf{a}_k t^5 + \mathbf{b}_k t^4 + \mathbf{c}_k t^3 + \mathbf{d}_k t^2 + \mathbf{e}_k t + \mathbf{f}_k. \tag{11.16}$$

The six coefficients are computed from the following six boundary conditions

$$\begin{aligned} \mathbf{P}_k(0) = \mathbf{P}_k, \quad \mathbf{P}_k(1) = \mathbf{P}_{k+1}, \quad \mathbf{P}'_k(1) = \mathbf{P}'_{k+1}(0), \\ \mathbf{P}''_k(1) = \mathbf{P}''_{k+1}(0), \quad \mathbf{P}'''_k(1) = \mathbf{P}'''_{k+1}(0), \quad \mathbf{P}''''_k(1) = \mathbf{P}''''_{k+1}(0). \end{aligned}$$

(Notice that these conditions involve the first four derivatives. Experience indicates that better-looking splines are obtained when the boundary conditions are based on an even number of derivatives, which is why the quintic, and not the quartic, polynomial is a natural choice.)

The boundary conditions can be written explicitly as follows:

$$\begin{aligned} &\mathbf{f}_k = \mathbf{P}_k, && a \\ &\mathbf{a}_k + \mathbf{b}_k + \mathbf{c}_k + \mathbf{d}_k + \mathbf{e}_k + \mathbf{f}_k = \mathbf{f}_{k+1} = \mathbf{P}_{k+1}, && b \\ &5\mathbf{a}_k + 4\mathbf{b}_k + 3\mathbf{c}_k + 2\mathbf{d}_k + \mathbf{e}_k = \mathbf{e}_{k+1}, && c \\ &20\mathbf{a}_k + 12\mathbf{b}_k + 6\mathbf{c}_k + 2\mathbf{d}_k = 2\mathbf{d}_{k+1}, && (12.42)d \\ &60\mathbf{a}_k + 24\mathbf{b}_k + 6\mathbf{c}_k = 6\mathbf{c}_{k+1}, && e \\ &120\mathbf{a}_k + 24\mathbf{b}_k = 24\mathbf{b}_{k+1}. && f \end{aligned}$$

These equations are now used to express the six coefficients of each of the $n-1$ quintic polynomials in terms of the second and fourth derivatives.

Equation (12.42)f results in $24\mathbf{b}_{k+1} = \mathbf{P}''''_{k+1}(0)$ or $\mathbf{b}_k = \frac{1}{24}\mathbf{P}''''_k(0)$. This also implies $\mathbf{a}_k = \frac{1}{120}[\mathbf{P}''''_{k+1}(0) - \mathbf{P}''''_k(0)]$. Equation (12.42)$d$ implies $2\mathbf{d}_{k+1} = \mathbf{P}''_{k+1}(0)$ or $\mathbf{d}_k = \frac{1}{2}\mathbf{P}''_k(0)$. Now that we have expressed $\mathbf{a}_k$, $\mathbf{b}_k$, and $\mathbf{d}_k$ in terms of the second and fourth derivatives, we substitute them in Equation (12.42)d to get the following expression for $\mathbf{c}_k$

$$\mathbf{c}_k = \frac{1}{6}[\mathbf{P}''_{k+1}(0) - \mathbf{P}''_k(0)] - \frac{1}{36}[\mathbf{P}''''_{k+1}(0) + 2\mathbf{P}''''_k(0)].$$

The last coefficient to be expressed in terms of the (still unknown) second and fourth derivatives is $\mathbf{e}_k$. This is done from $\mathbf{P}_k(1) = \mathbf{P}_{k+1}$ and results in

$$\mathbf{e}_k = [\mathbf{P}_{k+1} - \mathbf{P}_k] - \frac{1}{6}[\mathbf{P}''_{k+1}(0) + 2\mathbf{P}''_k(0)] + \frac{1}{360}[7\mathbf{P}''''_{k+1}(0) + 8\mathbf{P}''''_k(0)].$$

When these expressions for the six coefficients are combined with $\mathbf{P}'_{k-1}(1) = \mathbf{P}'_k(0)$, all the terms with first and third derivatives are eliminated, and the result is a relation between the (unknown) second and fourth derivatives and the (known) data points

$$\begin{aligned}&[\mathbf{P}_{k-1} - \mathbf{P}_k] + \frac{1}{6}[\mathbf{P}''_{k-1}(0) + 2\mathbf{P}''_k(0)] - \frac{1}{360}[7\mathbf{P}''''_{k-1}(0) + 8\mathbf{P}''''_k(0)] \\ &= [\mathbf{P}_{k+1} - \mathbf{P}_k] - \frac{1}{6}[\mathbf{P}''_{k+1}(0) + 2\mathbf{P}''_k(0)] + \frac{1}{360}[7\mathbf{P}''''_{k+1}(0) + 8\mathbf{P}''''_k(0)].\end{aligned} \tag{12.43}$$

When these expressions for the six coefficients are similarly combined with $\mathbf{P}'''_{k-1}(1) = \mathbf{P}'''_k(0)$, the result is another relation between the second and fourth derivates

$$-\mathbf{P}''_{k-1}(0) + 2\mathbf{P}''_k(0) - \mathbf{P}''_{k+1}(0) + \frac{1}{6}\mathbf{P}''''_{k-1}(0) + \frac{2}{3}\mathbf{P}''''_k(0) + \frac{1}{6}\mathbf{P}''''_{k+1}(0) = 0. \tag{12.44}$$

Each of Equations (12.43) and (12.44) is $n-1$ equations for $k = 1, 2, \ldots, n-1$, so we end up with $2(n-1)$ equations with the $2n$ second and fourth unknown derivatives. As in the case of the cubic spline, we complete this system of equations by guessing values for some extreme derivatives. The simplest end condition is to require

$$\mathbf{P}''''_1(0) = \mathbf{P}''''_{n-1}(1) = \mathbf{P}''''_1(0) = \mathbf{P}''''_{n-1}(1) = 0,$$

which implies $\mathbf{P}''_1(0) = \mathbf{P}''_1(1) - \frac{1}{6}\mathbf{P}''''_1(1)$ and $\mathbf{P}''_n(0) = \mathbf{P}''_{n-1}(1) - \frac{1}{6}\mathbf{P}''''_{n-1}(1)$ and makes it possible to eliminate $\mathbf{P}''_1(0)$ and $\mathbf{P}''_n(0)$ from Equations (12.43) and (12.44). [Späth 83] shows that the end result is the system of equations

$$\begin{bmatrix}\mathbf{A} & -\mathbf{B} \\ \mathbf{C} & \mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{P}'' \\ \mathbf{P}''''\end{bmatrix} = \begin{bmatrix}\mathbf{D} \\ \mathbf{0}\end{bmatrix}, \tag{12.45}$$

where

$$\mathbf{P}'' = \left(\mathbf{P}''_1(0), \ldots, \mathbf{P}''_{n-1}(0)\right)^T, \quad \mathbf{P}'''' = \left(\mathbf{P}''''_1(0), \ldots, \mathbf{P}''''_{n-1}(0)\right)^T,$$

$$\mathbf{D} = \left[6[(\mathbf{P}_2 - \mathbf{P}_1) - (\mathbf{P}_1 - \mathbf{P}_0)], \ldots, 6[(\mathbf{P}_n - \mathbf{P}_{n-1}) - (\mathbf{P}_{n-1} - \mathbf{P}_{n-2})]\right]^T,$$

and

$$\mathbf{A} = \begin{bmatrix}5 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 5\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix}26 & 7 & & & & \\ 7 & 16 & 7 & & & \\ & 7 & 16 & 7 & & \\ & & \ddots & & & \\ & & & 7 & 16 & 7 \\ & & & & 7 & 26\end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix}6 & -6 & & & & \\ -6 & 12 & -6 & & & \\ & -6 & 12 & -6 & & \\ & & \ddots & & & \\ & & & -6 & 12 & -6 \\ & & & & 6 & 6\end{bmatrix}.$$

Notice that matrices **A**, **B**, and **C** are tridiagonal and symmetric. In addition, **A** and **B** are diagonally dominant, while **C** is nonnegative definite. This guarantees that the block matrix of Equation (12.45) will have an inverse, which implies that the system of equations has a unique solution.

Solving the system of Equations (12.45) means expressing the second and fourth derivatives of the spline segments in terms of the data points (the known quantities). Once this is done, the six coefficients of each of the $n-1$ spline segments can be expressed in terms of the data points, and the segments can be constructed.

12.5 Cardinal Splines

The cardinal spline is another example of how Hermite interpolation is applied to construct a spline curve. The cardinal spline overcomes the main disadvantages of cubic splines, namely the lack of local control and the need to solve a system of linear equations that may be large (its size depends on the number of data points). Cardinal splines also offer a natural way to control the tension of the curve by modifying the magnitudes of the tangent vectors (Section 11.2.3). The price for all this is the loss of second-order continuity. Strictly speaking, this loss means that the cardinal spline isn't really a spline (see the definition of splines on Page 577), but its form, its derivation, and its behavior are so similar to those of other splines that the name "cardinal spline" has stuck.

Figure 12.12a illustrates the principle of this method. The figure shows a curve that passes through seven points. The curve looks continuous but is constructed in segments, two of which are thicker than the others. The first thick segment, the one from $\mathbf{P}_2$ to $\mathbf{P}_3$, starts in the direction from $\mathbf{P}_1$ to $\mathbf{P}_3$ and ends going in the direction from $\mathbf{P}_2$ to $\mathbf{P}_4$. The second thick segment, from $\mathbf{P}_5$ to $\mathbf{P}_6$, features the same behavior. It starts in the direction from $\mathbf{P}_4$ to $\mathbf{P}_6$ and ends going in the direction from $\mathbf{P}_5$ to $\mathbf{P}_7$.

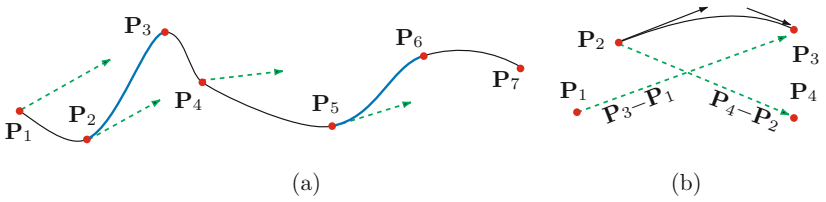


Figure 12.12: Tangent Vectors in a Cardinal Spline.

The cardinal spline for n given points is calculated and drawn in segments, each depending on four points only. Each point participates in at most four curve segments, so moving one point affects only those segments and not the entire curve. This is why the curve features *local control*. The individual segments connect smoothly; their first derivatives are equal at the connection points (the curve therefore has first-order continuity). However, the second derivatives of the segments are generally different at the connection points.

The first step in constructing the complete curve is to organize the points into $n-3$ highly-overlapping groups of four consecutive points each. The groups are

$$[\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4], \quad [\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5], \quad [\mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5, \mathbf{P}_6], \ldots, [\mathbf{P}_{n-3}, \mathbf{P}_{n-2}, \mathbf{P}_{n-1}, \mathbf{P}_n].$$

Hermite interpolation is then applied to construct a curve segment $\mathbf{P}(t)$ for each group. Denoting the four points of a group by $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$, the two interior points $\mathbf{P}_2$ and $\mathbf{P}_3$ become the start and end points of the segment and the two tangent vectors become $s(\mathbf{P}_3 - \mathbf{P}_1)$ and $s(\mathbf{P}_4 - \mathbf{P}_2)$, where s is a real number. Thus, segment $\mathbf{P}(t)$ goes from $\mathbf{P}_2$ to $\mathbf{P}_3$ and its two extreme tangent vectors are proportional to the vectors $\mathbf{P}_3 - \mathbf{P}_1$ and $\mathbf{P}_4 - \mathbf{P}_2$ (Figure 12.12b). The proportionality constant s is related to the tension parameter T. Note how there are no segments from $\mathbf{P}_1$ to $\mathbf{P}_2$ and from $\mathbf{P}_{n-1}$ to $\mathbf{P}_n$. These segments can be added to the curve by adding two new extreme points $\mathbf{P}_0$ and $\mathbf{P}_{n+1}$. These points can also be employed to edit the curve, because the first segment, from $\mathbf{P}_1$ to $\mathbf{P}_2$, starts going in the direction from $\mathbf{P}_0$ to $\mathbf{P}_2$, and similarly for the last segment.

The particular choice of the tangent vectors guarantees that the individual segments of the cardinal spline will connect smoothly. The end tangent $s(\mathbf{P}_4 - \mathbf{P}_2)$ of the segment for group $[\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4]$ is identical to the start tangent of the next group, $[\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5]$.

Segment $\mathbf{P}(t)$ is therefore defined by

$$\begin{aligned} \mathbf{P}(0) &= \mathbf{P}_2, \quad \mathbf{P}(1) = \mathbf{P}_3, \\ \mathbf{P}^t(0) &= s(\mathbf{P}_3 - \mathbf{P}_1), \quad \mathbf{P}^t(1) = s(\mathbf{P}_4 - \mathbf{P}_2) \end{aligned} \tag{12.46}$$

and is easily calculated by applying Hermite interpolation (Equation (11.7)) to the four quantities of Equation (12.46)

$$\begin{aligned} \mathbf{P}(t) &= (t^3, t^2, t, 1) \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_2 \\ \mathbf{P}_3 \\ s(\mathbf{P}_3 - \mathbf{P}_1) \\ s(\mathbf{P}_4 - \mathbf{P}_2) \end{pmatrix} \\ &= (t^3, t^2, t, 1) \begin{pmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{pmatrix}. \end{aligned} \tag{12.47}$$

Tension in the cardinal spline can now be controlled by changing the lengths of the tangent vectors by means of parameter s. A long tangent vector (obtained by a large s) causes the curve to continue longer in the direction of the tangent. A short tangent has the opposite effect; the curve moves a short distance in the direction of the tangent, then quickly changes direction and moves toward the end point. A zero-length tangent (corresponding to $s = 0$) produces a straight line between the endpoints (infinite tension). In principle, the parameter s can be varied from 0 to ∞. In practice, we use only values in the range $[0, 1]$. However, since $s = 0$ produces maximum tension, we cannot

intuitively think of s as the tension parameter and we need to define another parameter, T inversely related to s.

The tension parameter T is defined as $s = (1 - T)/2$, which implies $T = 1 - 2s$. The value $T = 0$ results in $s = 1/2$. The curve is defined as having tension zero in this case and is called the *Catmull–Rom spline* [Catmull and Rom 74]. Section 12.6 includes a detailed derivation of this type of spline as a blend of two parabolas. Increasing T from 0 to 1 decreases s from 1/2 to 0, thereby reducing the magnitude of the tangent vectors down to 0. This produces curves with more tension. Exercise 11.7 tells us that when the tangent vectors have magnitude zero, the Hermite curve segment is a straight line, so the entire cardinal spline curve becomes a set of straight segments, a polyline, the curve with maximum tension. Decreasing T from 0 to -1 increases s from 1/2 to 1. The result is a curve with more slack at the data points.

To illustrate this behavior mathematically, we rewrite Equation (12.47) explicitly to show its dependence on s:

$$\begin{aligned}\mathbf{P}(t) = {} & s(-t^3 + 2t^2 - t)\mathbf{P}_1 + s(-t^3 + t^2)\mathbf{P}_2 + (2t^3 - 3t^2 + 1)\mathbf{P}_2 \\ & + s(t^3 - 2t^2 + t)\mathbf{P}_3 + (-2t^3 + 3t^2)\mathbf{P}_3 + s(t^3 - t^2)\mathbf{P}_4.\end{aligned} \tag{12.48}$$

For $s = 0$, Equation (12.48) becomes $(2t^3 - 3t^2 + 1)\mathbf{P}_2 + (-2t^3 + 3t^2)\mathbf{P}_3$, which can be simplified to $(3t^2 - 2t^3)(\mathbf{P}_3 - \mathbf{P}_2) + \mathbf{P}_2$. Substituting $u = 3t^2 - 2t^3$ reduces this to $u(\mathbf{P}_3 - \mathbf{P}_2) + \mathbf{P}_2$, which is the straight line from $\mathbf{P}_2$ to $\mathbf{P}_3$.

For large s, we use Equation (12.48) to calculate the mid-curve value $\mathbf{P}(0.5)$:

$$\begin{aligned}\mathbf{P}(0.5) &= \frac{s}{8}\left[(\mathbf{P}_3 - \mathbf{P}_1) + (\mathbf{P}_2 - \mathbf{P}_4)\right] + 0.5(\mathbf{P}_2 + \mathbf{P}_3) \\ &= \frac{s}{8}\left[\mathbf{P}^t(0) - \mathbf{P}^t(1)\right] + 0.5(\mathbf{P}_2 + \mathbf{P}_3).\end{aligned}$$

This is an extension of Equation (Ans.13). The first term is the difference of the two tangent vectors, multiplied by $s/8$. As s grows, this term grows without limit. The second term is the midpoint of $\mathbf{P}_2$ and $\mathbf{P}_3$. Adding the two terms (a vector and a point) produces a point that may be located far away (for large s) from the midpoint, showing that the curve moves a long distance away from the start point $\mathbf{P}_2$ before changing direction and starting toward the end point $\mathbf{P}_3$. Large values of s therefore feature a loose curve (low tension).

Thus, the tension of the curve can be increased by setting s close to 0 (or, equivalently, setting T close to 1); it can be decreased by increasing s (or, equivalently, decreasing T toward 0).

◇ **Exercise 12.9:** What happens when $T > 1$?

Setting $T = 0$ results in $s = 0.5$. Equation (12.47) reduces in this case to

$$\mathbf{P}(t) = (t^3, t^2, t, 1)\begin{pmatrix} -0.5 & 1.5 & -1.5 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{pmatrix}, \tag{12.49}$$

a curve known as the Catmull–Rom spline. Its basis matrix is termed the parabolic blending matrix.

Example: Given the four points $(1,0)$, $(3,1)$, $(6,2)$, and $(2,3)$, we apply Equation (12.47) to calculate the cardinal spline segment from $(3,1)$ to $(6,2)$:

$$\begin{aligned}\mathbf{P}(t) &= (t^3, t^2, t, 1)\begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} (1,0) \\ (3,1) \\ (6,2) \\ (2,3) \end{bmatrix} \\ &= t^3(4s-6, 4s-2) + t^2(-9s+9, -6s+3) + t(5s, 2s) + (3,1).\end{aligned}$$

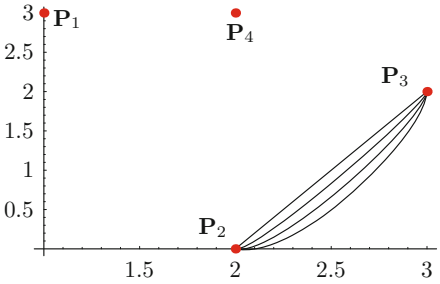
For high tension (i.e., $T = 1$ or $s = 0$), this reduces to the straight line

$$\mathbf{P}(t) = (-6,-2)t^3 + (9,3)t^2 + (3,1) = (3,1)(-2t^3 + 3t^2) + (3,1) = (3,1)u + (3,1).$$

For $T = 0$ (or $s = 1/2$), this cardinal spline reduces to the Catmull–Rom curve

$$\mathbf{P}(t) = (-4,0)t^3 + (4.5,0)t^2 + (2.5,1)t + (3,1). \tag{12.50}$$

Figure 12.13 shows an example of a similar cardinal spline (the points are different) with four values 0, 1/6, 2/6, and 3/6 of the tension parameter.




```
(* Cardinal spline example *)
T={t^3,t^2,t,1};
H[s_]:={{-s,2-s,s-2,s},{2s,s-3,3-2s,-s},{-s,0,s,0},{0,1,0,0}};
B={{1,3},{2,0},{3,2},{2,3}};
s=3/6; (* T=0 *)
g1=ParametricPlot[T.H[s].B,{t,0,1}];
s=2/6; (* T=1/3 *)
g2=ParametricPlot[T.H[s].B,{t,0,1}];
s=1/6; (* T=2/3 *)
g3=ParametricPlot[T.H[s].B,{t,0,1}];
s=0; (* T=1 *)
g4=ParametricPlot[T.H[s].B,{t,0,1}];
g5=Graphics[{AbsolutePointSize[4], Table[Point[B[[i]]],{i,1,4}] }];
Show[g1,g2,g3,g4,g5, PlotRange->All]
```

Figure 12.13: A Cardinal Spline Example.

12.6 Parabolic Blending: Catmull–Rom Curves

The Catmull–Rom curve (or the Catmull–Rom spline) is the special case of a cardinal spline with tension $T = 0$. In this section, we describe an approach to the Catmull–Rom spline where each spline segment is derived as the blend of two parabolas.

This approach to the Catmull–Rom curve proceeds in the following steps:

1. Organize the points in overlapping groups of three consecutive points each. The groups are

$$[\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3], \quad [\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4], \quad [\mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5], \quad \cdots \quad [\mathbf{P}_{n-2}, \mathbf{P}_{n-1}, \mathbf{P}_n].$$

2. Fit two parabolas, one through the first three points, $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$, and the other through the overlapping group, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$.

3. Calculate the first curve segment from $\mathbf{P}_2$ to $\mathbf{P}_3$ as a linear blend of the two parabolas, using the two barycentric weights $1 - t$ and t.

4. Fit a third parabola, through points $\mathbf{P}_3$, $\mathbf{P}_4$, and $\mathbf{P}_5$ and calculate the second curve segment, from $\mathbf{P}_3$ to $\mathbf{P}_4$, as a linear blend of the second and third parabolas.

5. Repeat until the last segment, from $\mathbf{P}_{n-2}$ to $\mathbf{P}_{n-1}$, is calculated as a linear blend of the $(n-3)$rd and the $(n-2)$nd parabolas.

Each parabola is defined by three points (which, of course, are on the same plane) and is therefore flat. However, the two parabolas that make up the segment are not generally on the same plane, so their blend is not necessarily flat and can twist in space.

The two original parabolas are denoted by $\mathbf{Q}(u) = (u^2, u, 1)\mathbf{H}_{123}$ and $\mathbf{R}(w) = (w^2, w, 1)\mathbf{H}_{234}$, where $\mathbf{H}_{123}$ and $\mathbf{H}_{234}$ are column vectors, each depending on the three points involved. They will have to be calculated. The expression for the blended segment is $\mathbf{P}(t) = (1-t)\mathbf{Q}(u)+t\mathbf{R}(w)$. Since this expression depends on t only, we have to express parameters u and w in terms of t. We try the linear expressions $u = at + b$, $w = ct + d$.

To calculate a, b, c, and d, we write the end conditions for the two parabolas and for the curve segment (Figure 12.14a):

$$\begin{aligned} &\mathbf{Q}(0) = \mathbf{P}_1, \quad \mathbf{Q}(0.5) = \mathbf{P}_2, \quad \mathbf{Q}(1) = \mathbf{P}_3, \\ &\mathbf{R}(0) = \mathbf{P}_2, \quad \mathbf{R}(0.5) = \mathbf{P}_3, \quad \mathbf{R}(1) = \mathbf{P}_4, \\ &\mathbf{P}(0) = \mathbf{P}_2, \qquad\qquad\quad \mathbf{P}(1) = \mathbf{P}_3. \end{aligned}$$

For point $\mathbf{P}_2$, we get (1) $u = 0.5$ and $t = 0$, implying $b = 0.5$, and (2) $w = 0$ and $t = 0$, implying $d = 0$. For point $\mathbf{P}_3$, we similarly get (1) $u = 1$ and $t = 1$, implying $a + b = 1 \Rightarrow a = 0.5$, and (2) $w = 0.5$ and $t = 1$, implying $c = 0.5$. This results in $u = (1 + t)/2$ and $w = t/2$.

Therefore, for the first parabola, we get

$$\left.\begin{aligned} \mathbf{Q}(0) &= \mathbf{P}_1 = (0, 0, 1)\mathbf{H}_{123}, \\ \mathbf{Q}(0.5) &= \mathbf{P}_2 = (1/4, 1/2, 1)\mathbf{H}_{123}, \\ \mathbf{Q}(1) &= \mathbf{P}_3 = (1, 1, 1)\mathbf{H}_{123}. \end{aligned}\right\} \Longrightarrow \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1/4 & 1/2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \mathbf{H}_{123} = \mathbf{M}\mathbf{H}_{123},$$

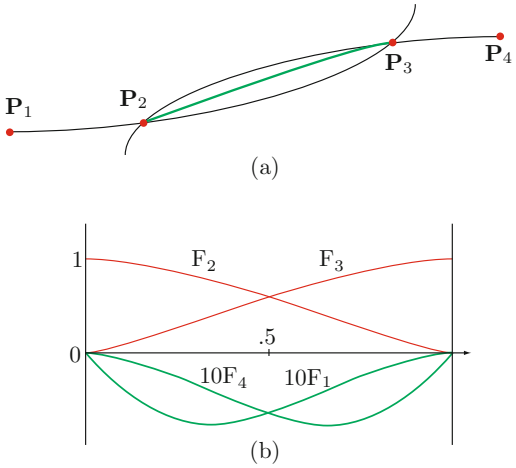


Figure 12.14: Parabolic Blending: (a) Two Parabolas. (b) The Blend Functions.

This can be solved for $\mathbf{H}_{123}$

$$\mathbf{H}_{123} = \mathbf{M}^{-1} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = \begin{pmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix}.$$

So the first parabola is

$$\mathbf{Q}(u) = (u^2, u, 1)\mathbf{M}^{-1} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix}.$$

The second parabola is obtained similarly:

$$\mathbf{R}(w) = (w^2, w, 1)\mathbf{M}^{-1} \begin{pmatrix} \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{pmatrix}.$$

The first curve segment is therefore

$$\begin{aligned} \mathbf{P}(t) &= (1-t)\mathbf{Q}(u) + t\mathbf{R}(w) \\ &= (1-t)(u^2, u, 1)\mathbf{M}^{-1} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} + t(w^2, w, 1)\mathbf{M}^{-1} \begin{pmatrix} \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{pmatrix} \\ &= (1-t)(2u^2 - 3u + 1, -4u^2 + 4u, 2u^2 - u) \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} \end{aligned}$$

$$
\begin{aligned}
&+ t(2w^2 - 3w + 1, -4w^2 + 4w, 2w^2 - w) \begin{pmatrix} \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{pmatrix} \\
&= (-0.5t^3 + t^2 - 0.5t)\mathbf{P}_1 + (1.5t^3 - 2.5t^2 + 1)\mathbf{P}_2 \\
&\quad + (-1.5t^3 + 2t^2 + 0.5t)\mathbf{P}_3 + (0.5t^3 - 0.5t^2)\mathbf{P}_4 \qquad (12.51) \\
&= (t^3, t^2, t, 1) \begin{pmatrix} -0.5 & 1.5 & -1.5 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{pmatrix} \\
&= (t^3, t^2, t, 1)\mathbf{BP}, \qquad (12.52)
\end{aligned}
$$

where $\mathbf{B}$ is called the parabolic blending matrix.

The other segments are calculated similarly. Note that, in practice, there is no need to calculate the parabolas. The program simply executes a loop where in each iteration, it uses Equation (12.51) with the next group of points to calculate the next segment.

The Catmull–Rom curve starts at point $\mathbf{P}_2$ and ends at $\mathbf{P}_{n-1}$. To make it pass through all n points $\mathbf{P}_1, \ldots, \mathbf{P}_n$, we add two more points $\mathbf{P}_0$ and $\mathbf{P}_{n+1}$. In practice, we normally select them as $\mathbf{P}_0 = \mathbf{P}_1$ and $\mathbf{P}_{n+1} = \mathbf{P}_n$. The first group of points is now $\mathbf{P}_0, \ldots, \mathbf{P}_3$, and the last one is $\mathbf{P}_{n-2}, \ldots, \mathbf{P}_{n+1}$. This also makes the method more interactive, since two more points can be repositioned to edit the shape of the curve. The curve can also be closed, if the first and last points are set to identical values.

Equation (12.51) gives the representation of the Catmull–Rom curve curves in terms of the four blending functions

$$F_1(t) = (-0.5t^3 + t^2 - 0.5t), \quad F_2(t) = (1.5t^3 - 2.5t^2 + 1),$$

$$F_3(t) = (-1.5t^3 + 2t^2 + 0.5t), \quad F_4(t) = (0.5t^3 - 0.5t^2).$$

Note how F_1 and F_4 are negative (Figure 12.14b), how F_2 and F_3 are symmetric, and how the four functions are barycentric.

◇ **Exercise 12.10:** Prove the first-order continuity of the parabolic curve.

Example: Given the five points $(1, 0)$, $(3, 1)$, $(6, 2)$, $(2, 3)$, and $(1, 4)$, we calculate the Catmull–Rom curve from $(1, 0)$ to $(1, 4)$. The first step is to add two more points, one on each end. We simply duplicate each of the two endpoints, ending up with seven points. The first segment is (from Equation (12.51))

$$
\begin{aligned}
\mathbf{P}_1(t) &= (-0.5t^3 + t^2 - 0.5t)(1, 0) + (1.5t^3 - 2.5t^2 + 1)(1, 0) \\
&\quad + (-1.5t^3 + 2t^2 + 0.5t)(3, 1) + (0.5t^3 - 0.5t^2)(6, 2) \\
&= (-0.5t^3 + 1.5t^2 + t + 1, -0.5t^3 + t^2 + 0.5t).
\end{aligned}
$$

This segment goes from point $(1, 0)$ (for $t = 0$) to point $(3, 1)$ (for $t = 1$). The next

segment, from $(3,1)$ to $(6,2)$, is similarly

$$\begin{aligned}\mathbf{P}_2(t) &= (-0.5t^3 + t^2 - 0.5t)(1,0) + (1.5t^3 - 2.5t^2 + 1)(3,1) \\ &\quad + (-1.5t^3 + 2t^2 + 0.5t)(6,2) + (0.5t^3 - 0.5t^2)(2,3) \\ &= (-4,0)t^3 + (4.5,0)t^2 + (2.5,1)t + (3,1).\end{aligned}$$

This is identical to Equation (12.50). Calculating the other two segments is left as an exercise.

12.6.1 Generalized Parabolic Blending

The previous discussion assumes that the n points are roughly equally spaced. This is why we could write $\mathbf{Q}(0.5) = \mathbf{P}_2$ and $\mathbf{R}(0.5) = \mathbf{P}_3$. This assumption is sometimes true in practical work. In cases where it isn't true, it is possible to write $\mathbf{Q}(\alpha) = \mathbf{P}_2$ and $\mathbf{R}(\beta) = \mathbf{P}_3$ (where $0 \le \alpha, \beta \le 1$ and their values depend on the placement of the points) and derive the expression for the curve from there.

Here is a summary of the results: The three parameters are now related by $u = (1-\alpha)t + \alpha$ and $w = \beta t$. The two parabolas are given by

$$\begin{pmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\end{pmatrix} = \begin{pmatrix}0 & 0 & 1\\ \alpha^2 & \alpha & 1\\ 1 & 1 & 1\end{pmatrix}\mathbf{H}_{123},$$

implying

$$\mathbf{H}_{123} = \mathbf{M}^{-1}\begin{pmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\end{pmatrix} = \begin{pmatrix}\frac{1}{\alpha} & \frac{-1}{\alpha(1-\alpha)} & \frac{1}{1-\alpha}\\ \frac{-(1+\alpha)}{\alpha} & \frac{1}{\alpha(1-\alpha)} & \frac{-\alpha}{1-\alpha}\\ 1 & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\end{pmatrix},$$

and

$$\begin{pmatrix}\mathbf{P}_2\\ \mathbf{P}_3\\ \mathbf{P}_4\end{pmatrix} = \begin{pmatrix}0 & 0 & 1\\ \beta^2 & \beta & 1\\ 1 & 1 & 1\end{pmatrix}\mathbf{H}_{234},$$

implying

$$\mathbf{H}_{234} = \mathbf{M}^{-1}\begin{pmatrix}\mathbf{P}_2\\ \mathbf{P}_3\\ \mathbf{P}_4\end{pmatrix} = \begin{pmatrix}\frac{1}{\beta} & \frac{-1}{\beta(1-\beta)} & \frac{1}{1-\beta}\\ \frac{-(1+\beta)}{\beta} & \frac{1}{\beta(1-\beta)} & \frac{-\beta}{1-\beta}\\ 1 & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{P}_2\\ \mathbf{P}_3\\ \mathbf{P}_4\end{pmatrix}.$$

The final expression of the curve is

$$\mathbf{P}(t) = (t^3, t^2, t, 1)\begin{pmatrix}\frac{-(1-\alpha)^2}{\alpha} & \frac{(1-\alpha)+\alpha\beta}{\alpha} & \frac{-(1-\alpha)-\alpha\beta}{1-\beta} & \frac{\beta^2}{1-\beta}\\ \frac{2(1-\alpha)^2}{\alpha} & \frac{-2(1-\alpha)-\alpha\beta}{\alpha} & \frac{2(1-\alpha)-\beta(1-2\alpha)}{1-\beta} & \frac{-\beta^2}{1-\beta}\\ \frac{-(1-\alpha)^2}{\alpha} & \frac{(1-2\alpha)}{\alpha} & \alpha & 0\\ 0 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\\ \mathbf{P}_4\end{pmatrix}.$$

12.6.2 Bessel's Algorithm

The cardinal spline and the Catmull–Rom curve are based on the particular way the two extreme tangent vectors of each four-point segment are defined. Equation (12.46) defines $\mathbf{P}^t(0) = s(\mathbf{P}_3 - \mathbf{P}_1)$ and $\mathbf{P}^t(1) = s(\mathbf{P}_4 - \mathbf{P}_2)$. So far, these definitions, which seem arbitrary, have been used because they make sense. They can, however, be explained (or justified) by a simple method called *Bessel's algorithm*. The idea is to calculate a quadratic interpolating polynomial $\mathbf{Q}_s(t)$ for the first three points $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$ and define $\mathbf{P}^t(0)$ as the tangent vector of $\mathbf{Q}_s(t)$ at point $\mathbf{P}_1$ (Figure 12.15). Similarly, another quadratic interpolating polynomial $\mathbf{Q}_e(t)$ is calculated for the last three points $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$, and $\mathbf{P}^t(1)$ is defined as the tangent vector of $\mathbf{Q}_e(t)$ at point $\mathbf{P}_2$.

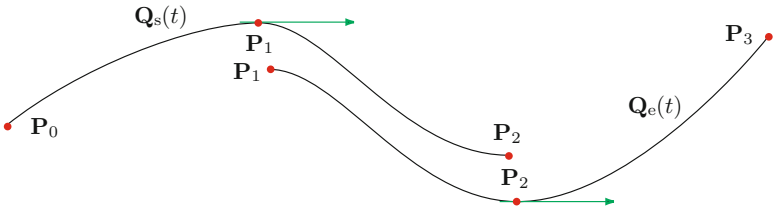


Figure 12.15: Bessel's Algorithm.

Friedrich Wilhelm Bessel (1784–1846): German astronomer and mathematician. Best known for making the first accurate measurement of the distance to a star.

The uniform quadratic Lagrange polynomial (Equation (10.11)) is used as our interpolating polynomial:

$$\begin{aligned}\mathbf{Q}_s(t) &= \frac{t^2-3t+2}{2}\mathbf{P}_0 - (t^2-2t)\mathbf{P}_1 + \frac{t^2-t}{2}\mathbf{P}_2 \\ &= (t^2,t,1)\begin{pmatrix} 1/2 & -1 & 1/2 \\ -3/2 & 2 & -1/2 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}.\end{aligned}$$

The parameter t varies in the range $[0,2]$, so $\mathbf{Q}_s(1)$ gives the middle point. The tangent vector of $\mathbf{Q}_s(t)$ is

$$\begin{aligned}\mathbf{Q}_s^t(t) &= \frac{2t-3}{2}\mathbf{P}_0 - (2t-2)\mathbf{P}_1 + \frac{2t-1}{2}\mathbf{P}_2 \\ &= (2t,1,0)\begin{pmatrix} 1/2 & -1 & 1/2 \\ -3/2 & 2 & -1/2 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}.\end{aligned}$$

Thus, $\mathbf{Q}_s^t(1) = (\mathbf{P}_2 - \mathbf{P}_0)/2$. Similarly,

$$\begin{aligned}\mathbf{Q}_e(t) &= \frac{t^2-3t+2}{2}\mathbf{P}_1 - (t^2-2t)\mathbf{P}_2 + \frac{t^2-t}{2}\mathbf{P}_3 \\ &= (t^2,t,1)\begin{pmatrix} 1/2 & -1 & 1/2 \\ -3/2 & 2 & -1/2 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix}\mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3\end{pmatrix},\end{aligned}$$

which yields $\mathbf{Q}_e^t(1) = (\mathbf{P}_3 - \mathbf{P}_1)/2$.

It is also possible to use the nonuniform quadratic Lagrange polynomial (Equation (10.12)). If we select

$$\mathbf{Q}_s(t) = (t^2,t,1)\begin{pmatrix} \dfrac{1}{\Delta_0(\Delta_0+\Delta_1)} & -\dfrac{1}{\Delta_0\Delta_1} & \dfrac{1}{(\Delta_0+\Delta_1)\Delta_1} \\ \dfrac{-1}{\Delta_0+\Delta_1} - \dfrac{1}{\Delta_0} & \dfrac{1}{\Delta_0} + \dfrac{1}{\Delta_1} & -\dfrac{1}{\Delta_1} + \dfrac{1}{\Delta_0+\Delta_1} \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix}\mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2\end{pmatrix}, \quad (12.53)$$

then the tangent vector at point $\mathbf{P}_1$ becomes

$$\begin{aligned}\mathbf{Q}_s^t(\Delta_0) &= -\frac{\Delta_1}{\Delta_0(\Delta_0+\Delta_1)}\mathbf{P}_0 + \frac{\Delta_1-\Delta_0}{\Delta_0\Delta_1}\mathbf{P}_1 + \frac{\Delta_0}{(\Delta_0+\Delta_1)\Delta_1}\mathbf{P}_2 \\ &= \left(\frac{\Delta_1}{\Delta_0+\Delta_1}\right)\left(\frac{\mathbf{P}_1-\mathbf{P}_0}{\Delta_0}\right) + \left(\frac{\Delta_0}{\Delta_0+\Delta_1}\right)\left(\frac{\mathbf{P}_2-\mathbf{P}_1}{\Delta_1}\right).\end{aligned} \quad (12.54)$$

It is easy to see that Equation (12.54) reduces to $(\mathbf{P}_2 - \mathbf{P}_0)/2$ when $\Delta_0 = \Delta_1 = 1$.

◇ **Exercise 12.11:** Use Equation (10.12) to represent $\mathbf{Q}_e(t)$ and calculate the tangent vector $\mathbf{Q}_e^t(\Delta_1)$.

12.7 Catmull–Rom Surfaces

The cardinal spline or the Catmull–Rom curve can easily be extended to a surface that's fully defined by a rectangular grid of data points. In analogy to the Catmull–Rom curve segment—which involves four points but only passes through the two interior points—a single Catmull–Rom surface patch is specified by 16 points, the patch is anchored at the four middle points and spans the area delimited by them.

We start with a group of $m \times n$ data points roughly arranged in a rectangle. We examine all the overlapping groups that consist of 4×4 adjacent points, and we calculate a surface patch for each group. Some of the groups are shown in Figure 12.16.

$$
\begin{array}{llll}
\mathbf{P}_{40}\mathbf{P}_{41}\mathbf{P}_{42}\mathbf{P}_{43} & \mathbf{P}_{41}\mathbf{P}_{42}\mathbf{P}_{43}\mathbf{P}_{44} & \mathbf{P}_{42}\mathbf{P}_{43}\mathbf{P}_{44}\mathbf{P}_{45} & \cdots\ \mathbf{P}_{4,n-3}\mathbf{P}_{4,n-2}\mathbf{P}_{4,n-1}\mathbf{P}_{4n} \\
\mathbf{P}_{30}\mathbf{P}_{31}\mathbf{P}_{32}\mathbf{P}_{33} & \mathbf{P}_{31}\mathbf{P}_{32}\mathbf{P}_{33}\mathbf{P}_{34} & \mathbf{P}_{32}\mathbf{P}_{33}\mathbf{P}_{34}\mathbf{P}_{35} & \cdots\ \mathbf{P}_{3,n-3}\mathbf{P}_{3,n-2}\mathbf{P}_{3,n-1}\mathbf{P}_{3n} \\
\mathbf{P}_{20}\mathbf{P}_{21}\mathbf{P}_{22}\mathbf{P}_{23} & \mathbf{P}_{21}\mathbf{P}_{22}\mathbf{P}_{23}\mathbf{P}_{24} & \mathbf{P}_{22}\mathbf{P}_{23}\mathbf{P}_{24}\mathbf{P}_{25} & \cdots\ \mathbf{P}_{2,n-3}\mathbf{P}_{2,n-2}\mathbf{P}_{2,n-1}\mathbf{P}_{2n} \\
\mathbf{P}_{10}\mathbf{P}_{11}\mathbf{P}_{12}\mathbf{P}_{13} & \mathbf{P}_{11}\mathbf{P}_{12}\mathbf{P}_{13}\mathbf{P}_{14} & \mathbf{P}_{12}\mathbf{P}_{13}\mathbf{P}_{14}\mathbf{P}_{15} & \cdots\ \mathbf{P}_{1,n-3}\mathbf{P}_{1,n-2}\mathbf{P}_{1,n-1}\mathbf{P}_{1n} \\
 & & & \\
\mathbf{P}_{30}\mathbf{P}_{31}\mathbf{P}_{32}\mathbf{P}_{33} & \mathbf{P}_{31}\mathbf{P}_{32}\mathbf{P}_{33}\mathbf{P}_{34} & \mathbf{P}_{32}\mathbf{P}_{33}\mathbf{P}_{34}\mathbf{P}_{35} & \cdots\ \mathbf{P}_{3,n-3}\mathbf{P}_{3,n-2}\mathbf{P}_{3,n-1}\mathbf{P}_{3n} \\
\mathbf{P}_{20}\mathbf{P}_{21}\mathbf{P}_{22}\mathbf{P}_{23} & \mathbf{P}_{21}\mathbf{P}_{22}\mathbf{P}_{23}\mathbf{P}_{24} & \mathbf{P}_{22}\mathbf{P}_{23}\mathbf{P}_{24}\mathbf{P}_{25} & \cdots\ \mathbf{P}_{2,n-3}\mathbf{P}_{2,n-2}\mathbf{P}_{2,n-1}\mathbf{P}_{2n} \\
\mathbf{P}_{10}\mathbf{P}_{11}\mathbf{P}_{12}\mathbf{P}_{13} & \mathbf{P}_{11}\mathbf{P}_{12}\mathbf{P}_{13}\mathbf{P}_{14} & \mathbf{P}_{12}\mathbf{P}_{13}\mathbf{P}_{14}\mathbf{P}_{15} & \cdots\ \mathbf{P}_{1,n-3}\mathbf{P}_{1,n-2}\mathbf{P}_{1,n-1}\mathbf{P}_{1n} \\
\mathbf{P}_{00}\mathbf{P}_{01}\mathbf{P}_{02}\mathbf{P}_{03} & \mathbf{P}_{01}\mathbf{P}_{02}\mathbf{P}_{03}\mathbf{P}_{04} & \mathbf{P}_{02}\mathbf{P}_{03}\mathbf{P}_{04}\mathbf{P}_{05} & \cdots\ \mathbf{P}_{0,n-3}\mathbf{P}_{0,n-2}\mathbf{P}_{0,n-1}\mathbf{P}_{0n}
\end{array}
$$

Figure 12.16: Points for a Catmull–Rom Surface Patch.

The expression of the surface is obtained by applying the technique of Cartesian product (Section 8.12) to the Catmull–Rom curve. Equation (8.25) produces

$$\mathbf{P}(u,w) = (u^3, u^2, u, 1)\mathbf{B}\mathbf{P}\mathbf{B}^T \begin{pmatrix} w^3 \\ w^2 \\ w \\ 1 \end{pmatrix}, \tag{12.55}$$

where $\mathbf{B}$ is the parabolic blending matrix of Equation (12.49)

$$\mathbf{B} = \begin{pmatrix} -0.5 & 1.5 & -1.5 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

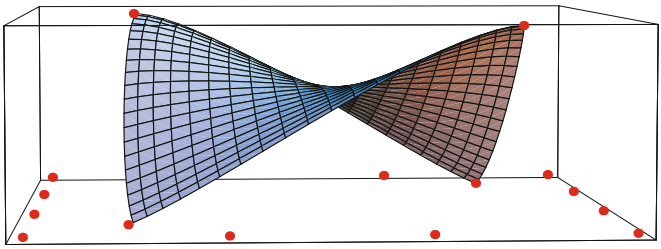
and $\mathbf{P}$ is a matrix consisting of the 4×4 points participating in the patch

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{i+3,j} & \mathbf{P}_{i+3,j+1} & \mathbf{P}_{i+3,j+2} & \mathbf{P}_{i+3,j+3} \\ \mathbf{P}_{i+2,j} & \mathbf{P}_{i+2,j+1} & \mathbf{P}_{i+2,j+2} & \mathbf{P}_{i+2,j+3} \\ \mathbf{P}_{i+1,j} & \mathbf{P}_{i+1,j+1} & \mathbf{P}_{i+1,j+2} & \mathbf{P}_{i+1,j+3} \\ \mathbf{P}_{i,j} & \mathbf{P}_{i,j+1} & \mathbf{P}_{i,j+2} & \mathbf{P}_{i,j+3} \end{pmatrix}.$$

Notice that the patch spans the area bounded by the four central points. In general, the entire surface spans the area bounded by the four points $\mathbf{P}_{11}$, $\mathbf{P}_{1,n-1}$, $\mathbf{P}_{m-1,1}$, and $\mathbf{P}_{m-1,n-1}$. If we want the surface to span the area bounded by the four corner points $\mathbf{P}_{00}$, $\mathbf{P}_{0n}$, $\mathbf{P}_{m0}$, and $\mathbf{P}_{mn}$, we have to create two new extreme rows and two new extreme columns of points, by analogy with the Catmull–Rom curve.

Example: Given the following coordinates for 16 points in file `CRpoints`

```
0 0 0   1   0 0   2     0 0   3 0 0
0 1 0  .5  .5 1   2.5  .5 0   3 1 0
0 2 0  .5 2.5 0   2.5 2.5 1   3 2 0
0 3 0   1   3 0   2     3 0   3 3 0
```



```
0 0 0  1  0 0 2   0 0 3 0 0
0 1 0  .5  .5 1 2.5  .5 0 3 1 0
0 2 0  .5 2.5 0 2.5 2.5 1 3 2 0
0 3 0  1 3  0 2  3  0 3 3 0
```

```
Clear[Pt,Bm,CRpatch,g1,g2];
Pt=ReadList["CRpoints",{Number,Number,Number},RecordLists->True];
Bm:={{-.5,1.5,-1.5,.5},{1,-2.5,2,-.5},{-.5,0,.5,0},{0,1,0,0}};
CRpatch[i_]:=(*1st patch,rows 1-4*)
 {u^3,u^2,u,1}.Bm.Pt[[{1,2,3,4},{1,2,3,4},i]].
 Transpose[Bm].{w^3,w^2,w,1};
g1=Graphics3D[{Red, AbsolutePointSize[6],
 Table[Point[Pt[[i,j]]],{i,1,4},{j,1,4}]}];
g2=ParametricPlot3D[{CRpatch[1],CRpatch[2],CRpatch[3]},
 {u,0,.98},{w,0,1}];
Show[g1,g2,ViewPoint->{-4.322,0.242,0.306},PlotRange->All]
```

Figure 12.17: A Catmull–Rom Surface Patch.

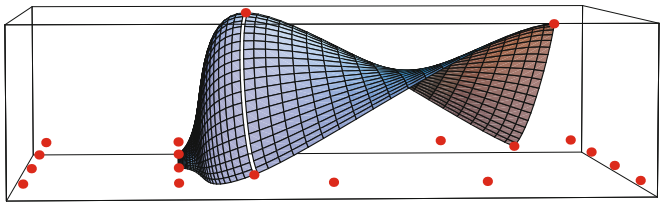
the *Mathematica* code of Figure 12.17 reads the file and generates the Catmull–Rom patch. Note how the patch spans only the four center points and how the z coordinates of 0 and 1 create the particular shape of the patch.

Example: (extended) We now add four more points to file `CRpoints`, and use rows 2–5 to calculate and display another patch. Notice the five values of y compared to the four values of x. The code of Figure 12.18 reads the extended file and generates and displays both patches. Each patch spans four points, but they share the two points $(0.5, 2.5, 0)$ and $(2.5, 2.5, 1)$. Note how they connect smoothly.

Tension can be added to a Catmull–Rom surface patch in the same way that it is added to a Catmull–Rom curve or to a cardinal spline. Figure 12.19 illustrates how smaller values of s create a surface closer to a flat plane.

12.8 Kochanek–Bartels Splines

The Kochanek–Bartels spline method [Kochanek and Bartels 84] is an extension of the cardinal spline. In addition to the tension parameter T, this method introduces two new parameters, c and b to control the *continuity* and *bias*, respectively, of individual curve segments. The curve is a spline computed from a set of n data points, and the three



```
0 0 0  1  0 0 2   0 0 3 0 0
0 1 0  .5  .5 1 2.5  .5 0 3 1 0
0 2 0  .5 2.5 0 2.5 2.5 1 3 2 0
0 3 0  1 3  0 2  3  0 3 3 0
0 4 0  1 4  0 2  4  0 3 4 0

Clear[Pt,Bm,CRpatch,CRpatchM,g1,g2,g3];
Pt=ReadList["CRpoints",{Number,Number,Number},RecordLists->True];
Bm:={{-.5,1.5,-1.5,.5},{1,-2.5,2,-.5},{-.5,0,.5,0},{0,1,0,0}};
CRpatch[i_]:=(*1st patch,rows 1-4*){u^3,u^2,u,1}.Bm.
 Pt[[{1,2,3,4},{1,2,3,4},i]].Transpose[Bm].{w^3,w^2,w,1};
CRpatchM[i_]:=(*2nd patch,rows 2-5*){u^3,u^2,u,1}.Bm.
 Pt[[{2,3,4,5},{1,2,3,4},i]].Transpose[Bm].{w^3,w^2,w,1};
g1=Graphics3D[{Red,AbsolutePointSize[6],
 Table[Point[Pt[[i,j]]],{i,1,5},{j,1,4}]}];
g2=ParametricPlot3D[{CRpatch[1],CRpatch[2],CRpatch[3]},
 {u,0,.98},{w,0,1}];
g3=ParametricPlot3D[{CRpatchM[1],CRpatchM[2],CRpatchM[3]},
 {u,0,1},{w,0,1}];
Show[g1,g2,g3,PlotRange->All]
```

Figure 12.18: Two Catmull–Rom Surface Patches.

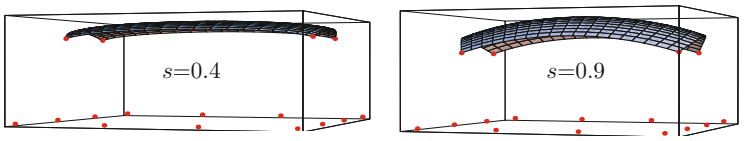
shape parameters can be specified separately for each point or can be global. Thus, the user/designer has to specify either 3 or $3n$ parameters.

Consider an interior point $\mathbf{P}_k$ where two spline segments meet. When the "arriving" segment arrives at the point it is moving in a certain direction that we call the arriving tangent vector. Similarly, the "departing" segment starts at the point while moving in a direction that we call the departing tangent vector. The three shape parameters control these two tangent vectors in various ways. The tension parameter varies the magnitudes of the arriving and departing vectors. The bias parameter rotates both tangents by the same amount from their "natural" direction, and the continuity parameter rotates each tangent separately, so they may no longer point in the same direction.

A complete Kochanek–Bartels spline passes through n given data points $\mathbf{P}_1$ through $\mathbf{P}_n$ and is computed and displayed in the following steps:

1. The designer (or user) adds two new points $\mathbf{P}_0$ and $\mathbf{P}_{n+1}$. Recall that each cardinal spline segment is determined by a group of four points but it goes from the second point to the third one. Adding point $\mathbf{P}_0$ makes it possible to have a segment from $\mathbf{P}_1$ to $\mathbf{P}_2$, and similarly for the new point $\mathbf{P}_{n+1}$. All the original n points are now interior.

2. Two tangent vectors, arriving and departing, are computed for each of the n interior points from Equations (12.56) and (12.57). The arriving tangent at $\mathbf{P}_1$ and the




```
(* A Catmull-Rom surface with tension *)
Clear[Pt,Bm,CRpatch,g1,g2,s];
Pt={{{0,3,0},{1,3,0},{2,3,0},{3,3,0}},
 {{0,2,0},{.1,2,.9},{2.9,2,.9},{3,2,0}},
 {{0,1,0},{.1,1,.9},{2.9,1,.9},{3,1,0}},
 {{0,0,0},{1,0,0},{2,0,0},{3,0,0}}};
Bm:={{-s,2-s,s-2,s},{2s,s-3,3-2s,-s},{-s,0,s,0},{0,1,0,0}};
CRpatch[i_]:=(*rows 1-4*){u^3,u^2,u,1}.Bm.
 Pt[[{1,2,3,4},{1,2,3,4},i]].Transpose[Bm].{w^3,w^2,w,1};
g1=Graphics3D[{Red,AbsolutePointSize[6],
 Table[Point[Pt[[i,j]]],{i,1,4},{j,1,4}]}];
s=.4;
g2=ParametricPlot3D[{CRpatch[1],CRpatch[2],CRpatch[3]},
 {u,0,1},{w,0,1}];
Show[g1,g2,ViewPoint->{1.431,-4.097,0.011},PlotRange->All]
```

Figure 12.19: A Catmull–Rom Surface Patch with Tension.

departing tangent at $\mathbf{P}_n$ are not used, so the total number of tangents to compute is $2n-2$.

3. The $n+2$ points are divided into $n-1$ overlapping groups of four points each, and a Hermite curve segment is computed and displayed for each group. The computations are similar to those for the cardinal spline, the only difference being that the tangent vectors are computed in a special way.

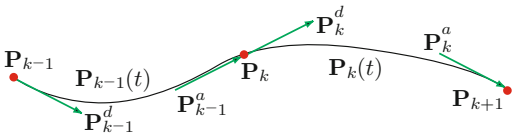


Figure 12.20: Two Kochanek–Bartels Spline Segments.

Figure 12.20 shows two spline segments $\mathbf{P}_{k-1}(t)$ and $\mathbf{P}_k(t)$ that meet at interior point $\mathbf{P}_k$. This point is the last endpoint of segment $\mathbf{P}_{k-1}(t)$ and the first endpoint of segment $\mathbf{P}_k(t)$. We denote the two tangent vectors at $\mathbf{P}_k$ by $\mathbf{P}^a_{k-1} \stackrel{\text{def}}{=} \mathbf{P}^t_{k-1}(1)$ and $\mathbf{P}^d_k \stackrel{\text{def}}{=} \mathbf{P}^t_k(0)$. In a cardinal spline the two tangents $\mathbf{P}^a_{k-1}$ and $\mathbf{P}^d_k$ are identical and are proportional to the vector $\mathbf{P}_{k+1}-\mathbf{P}_{k-1}$ (the chord surrounding $\mathbf{P}_k$). This guarantees a smooth connection of the two segments. In a Kochanek–Bartels spline, the two tangents are computed as shown here, they have the same magnitude, but may point in different directions. Notice that the two endpoints of segment $\mathbf{P}_k(t)$ are $\mathbf{P}_k$ and $\mathbf{P}_{k+1}$ and its two extreme tangent vectors are $\mathbf{P}^d_k$ and $\mathbf{P}^a_k$. Here is how the tangent vectors are computed.

Tension. In a cardinal spline, tension is controlled by multiplying the tangent vectors by a parameter s. Small values of s produce high tension, so the tension parameter T is defined by $s = (1 - T)/2$. Thus, we can express the tangents as

$$\frac{1-T}{2}(\mathbf{P}_{k+1} - \mathbf{P}_{k-1}) = (1-T)\frac{1}{2}\big((\mathbf{P}_{k+1} - \mathbf{P}_k) + (\mathbf{P}_k - \mathbf{P}_{k-1})\big).$$

This can be interpreted as $(1 - T)$ multiplied by the average of the "arriving" chord $(\mathbf{P}_k - \mathbf{P}_{k-1})$ and the "departing" chord $(\mathbf{P}_{k+1} - \mathbf{P}_k)$. In a Kochanek–Bartels spline, the tension parameter contributes the same quantity

$$(1 - T_k)\frac{1}{2}\big((\mathbf{P}_{k+1} - \mathbf{P}_k) + (\mathbf{P}_k - \mathbf{P}_{k-1})\big)$$

to the two tangents $\mathbf{P}^a_{k-1}$ and $\mathbf{P}^d_k$ at point $\mathbf{P}_k$. The value $T_k = 1$ results in tangent vectors of zero magnitude, which corresponds to maximum tension. The value $T_k = 0$ (zero tension) results in a contribution of $(\mathbf{P}_{k+1} - \mathbf{P}_{k-1})/2$ to both tangent vectors. The value $T_k = -1$ results in twice that contribution and therefore to long tangents and low tension.

Continuity. Curves are important in computer animation. An object being animated is often moved along a curve and the (virtual) camera may also move along a path. Sometimes, an animation path should not be completely smooth, but should feature jumps and jerks at certain points. This effect is achieved in a Kochanek–Bartels spline by separately rotating $\mathbf{P}^a_{k-1}$ and $\mathbf{P}^d_k$, so that they point in different directions. The contributions of the continuity parameter to these vectors are

$$\begin{aligned}
\text{contribution to}\quad \mathbf{P}^a_{k-1}\quad \text{is}\quad & \frac{1-c_k}{2}(\mathbf{P}_k - \mathbf{P}_{k-1}) + \frac{1+c_k}{2}(\mathbf{P}_{k+1} - \mathbf{P}_k),\\
\text{contribution to}\quad \mathbf{P}^d_k\quad \text{is}\quad & \frac{1+c_k}{2}(\mathbf{P}_k - \mathbf{P}_{k-1}) + \frac{1-c_k}{2}(\mathbf{P}_{k+1} - \mathbf{P}_k),
\end{aligned}$$

where c_k is the continuity parameter at point $\mathbf{P}_k$. The value $c_k = 0$ results in $\mathbf{P}^a_{k-1} = \mathbf{P}^d_k$ and therefore in a smooth curve at $\mathbf{P}_k$. For $c_k \neq 0$, the two tangents are different and the curve has a sharp corner (a kink or a cusp) at point $\mathbf{P}_k$, a corner that becomes more pronounced for large values of c_k. The case $c_k = -1$ implies $\mathbf{P}^a_{k-1} = \mathbf{P}_k - \mathbf{P}_{k-1}$ (the arriving chord) and $\mathbf{P}^d_k = \mathbf{P}_{k+1} - \mathbf{P}_k$ (the departing chord). The case $c_k = 1$ produces tangent vectors in the opposite directions: $\mathbf{P}^a_{k-1} = \mathbf{P}_{k+1} - \mathbf{P}_k$ and $\mathbf{P}^d_k = \mathbf{P}_k - \mathbf{P}_{k-1}$. These three extreme cases are illustrated in Figure 12.21.

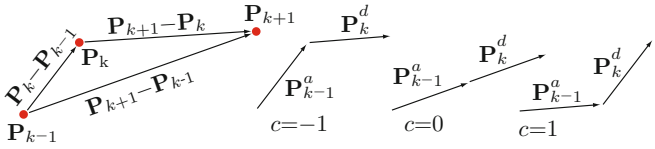


Figure 12.21: Effects of the Continuity Parameter.

Tension and continuity may have the same effect, yet they affect the dynamics of the curve in different ways as illustrated by Figure 12.22. Part (a) of the figure shows five points and a two-segment Kochanek–Bartels spline from $\mathbf{P}_1$ through $\mathbf{P}_2$ to $\mathbf{P}_3$. Both the tension and continuity parameters are set to zero at $\mathbf{P}_2$, so the direction of the curve at this point is the direction of the chord $\mathbf{P}_3 - \mathbf{P}_1$. Setting $T = 1$ at $\mathbf{P}_2$ increases the tension to maximum at that point, thereby changing the curve to two straight segments (part (b) of the figure). However, if we leave T at zero and set $c = -1$ at $\mathbf{P}_2$, the resulting curve will have the same shape (the direction of the arriving tangent $\mathbf{P}_1^a$ is from $\mathbf{P}_1$ to $\mathbf{P}_2$ while the direction of the departing tangent $\mathbf{P}_2^d$ is from $\mathbf{P}_2$ to $\mathbf{P}_3$).

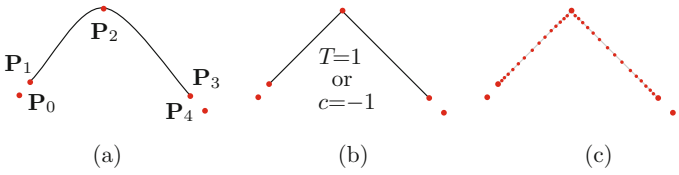


Figure 12.22: Different Dynamics of Tension and Continuity.

Thus, maximum tension and minimum continuity may result in identical geometries, but not in identical curves. These parameters have different effects on the *speed* of the curve as illustrated in Part (c) of the figure. Specifically, infinite tension results in nonuniform speed. If the first spline segment $\mathbf{P}_1(t)$ is plotted by incrementing t in equal steps, the resulting points are first bunched together, then feature larger gaps, and finally become dense again.

When the user specifies high (or maximum) tension at a point, the tangent vectors become short (or zero) at the point, but they get longer as the curve moves away from the point. The speed of the curve is determined by the size of its tangent vector, which is why high tension results in nonuniform speed. In contrast, low tension does not affect the magnitude of the tangent vectors, which is why it does not affect the speed. When low continuity results in a straight segment, the speed will be uniform. Curved segments, however, always have variable speed regardless of the continuity parameters at the endpoints of the segment.

◇ **Exercise 12.12:** Compute the tangent vector of the cardinal spline for $s = 0$ and show that its length is zero for $t = 0$ and $t = 1$, but is nonzero elsewhere.

Bias. In a cardinal spline with zero tension, both tangent vectors at point $\mathbf{P}_k$ have the value

$$\frac{1}{2}(\mathbf{P}_{k+1} - \mathbf{P}_{k-1}) = \frac{1}{2}\big((\mathbf{P}_k - \mathbf{P}_{k-1}) + (\mathbf{P}_{k+1} - \mathbf{P}_k)\big),$$

implying that the direction of the curve at point $\mathbf{P}_k$ is the average of the two chords connecting at $\mathbf{P}_k$.

The Kochanek–Bartels spline introduces an additional (sometimes misunderstood) parameter b_k to control the direction of the curve at $\mathbf{P}_k$ by rotating $\mathbf{P}_{k-1}^a$ and $\mathbf{P}_k^d$ by the same amount. The contribution of the bias parameter to the arriving and departing

tangents is set (somewhat arbitrarily) to

$$\frac{1+b_k}{2}\,(\mathbf{P}_k-\mathbf{P}_{k-1})+\frac{1-b_k}{2}\,(\mathbf{P}_{k+1}-\mathbf{P}_k)\,.$$

Setting $b_k=1$ changes both tangents to $\mathbf{P}_k-\mathbf{P}_{k-1}$, the chord on the left of $\mathbf{P}_k$. The other extreme value, $b_k=-1$, changes them to the chord on the right of $\mathbf{P}_k$. Figure 12.23 illustrates the effects of the three extreme values of b_k.

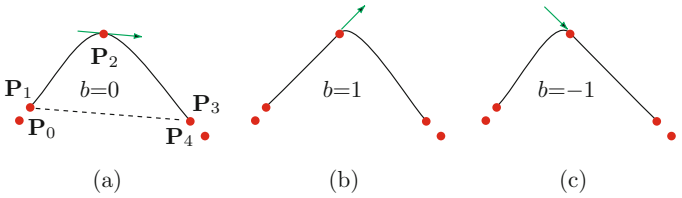


Figure 12.23: Effect of the Bias Parameter b.

Bias is used in computer animation to obtain the effect of overshooting a point ($b_k=1$) or undershooting it ($b_k=-1$).

The three shape parameters are incorporated in the tangent vectors as follows: the tangent vector that departs point $\mathbf{P}_k$ is defined by

$$\mathbf{P}_k^d=\mathbf{P}_k^t(0)=\frac{1}{2}(1-T_k)(1+b_k)(1-c_k)(\mathbf{P}_k-\mathbf{P}_{k-1})+\frac{1}{2}(1-T_k)(1-b_k)(1+c_k)(\mathbf{P}_{k+1}-\mathbf{P}_k). \tag{12.56}$$

Similarly, the tangent vector arriving at point $\mathbf{P}_{k+1}$ is defined by

$$\begin{aligned}\mathbf{P}_k^a=\mathbf{P}_k^t(1)&=\frac{1}{2}(1-T_{k+1})(1+b_{k+1})(1+c_{k+1})(\mathbf{P}_{k+1}-\mathbf{P}_k)\\&+\frac{1}{2}(1-T_{k+1})(1-b_{k+1})(1-c_{k+1})(\mathbf{P}_{k+2}-\mathbf{P}_{k+1}).\end{aligned} \tag{12.57}$$

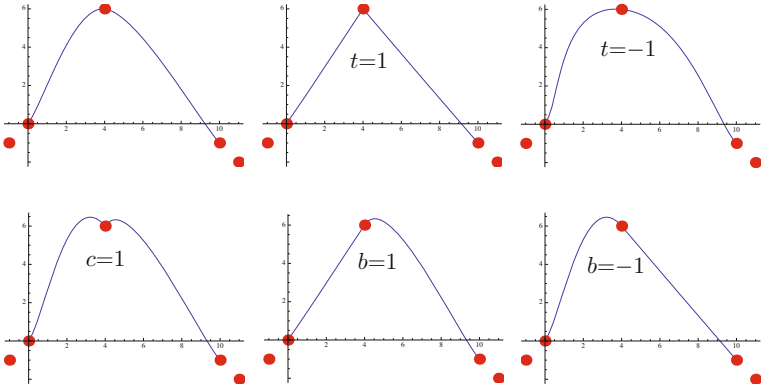
As a result, the Kochanek–Bartels curve segment $\mathbf{P}_k(t)$ from $\mathbf{P}_k$ to $\mathbf{P}_{k+1}$ is constructed by the familiar expression

$$\mathbf{P}_k(t)=(t^3,t^2,t,1)\mathbf{H}\begin{pmatrix}\mathbf{P}_k\\ \mathbf{P}_{k+1}\\ \mathbf{P}_k^d\\ \mathbf{P}_k^a\end{pmatrix},$$

where $\mathbf{H}$ is the Hermite matrix, Equation (11.7). Notice that the segment depends on six shape parameters, three at $\mathbf{P}_k$ and three at $\mathbf{P}_{k+1}$. The segment also depends on four points $\mathbf{P}_{k-1}$, $\mathbf{P}_k$, $\mathbf{P}_{k+1}$, and $\mathbf{P}_{k+2}$.

Note also that the second derivatives of this curve are generally not continuous at the data points.

Example: The three points $\mathbf{P}_1 = (0,0)$, $\mathbf{P}_2 = (4,6)$, and $\mathbf{P}_3 = (10,-1)$ are given, together with the extra points $\mathbf{P}_0 = (-1,-1)$ and $\mathbf{P}_4 = (11,-2)$. Up to nine shape parameters can be specified (three parameters for each of the three interior points). Figure 12.24 shows the curve with all shape parameters set to zero, and the effects of setting T to 1 (maximum tension) and to -1 (a loose curve), setting c to 1, and setting b to 1 (overshoot) and -1 (undershoot), all in $\mathbf{P}_2$. The *Mathematica* code that computed the curves is also included.




```
Clear[T, H, B, pts, Pa, Pd, te, bi, co];
(*Kochanek Bartels 3+2 points*)
T = {t^3, t^2, t, 1};
H = {{2, -2, 1, 1}, {-3, 3, -2, -1}, {0, 0, 1, 0}, {1, 0, 0, 0}};
Pd[k_] := (1 - te[[k + 1]]) (1 + bi[[k + 1]]) (1 +
    co[[k + 1]]) (pts[[k + 1]] - pts[[k]])/
   2 + (1 - te[[k + 1]]) (1 - bi[[k + 1]]) (1 -
    co[[k + 1]]) (pts[[k + 2]] - pts[[k + 1]])/2;
Pa[k_] := (1 - te[[k + 2]]) (1 + bi[[k + 2]]) (1 -
    co[[k + 2]]) (pts[[k + 2]] - pts[[k + 1]])/
   2 + (1 - te[[k + 2]]) (1 - bi[[k + 2]]) (1 +
    co[[k + 2]]) (pts[[k + 3]] - pts[[k + 2]])/2;
pts := {{-1, -1}, {0, 0}, {4, 6}, {10, -1}, {11, -2}};
te = {0, 0, 0, 0, 0}; bi = {0, 0, 0, 0, 0}; co = {0, 0, 0, 0, 0};
B = {pts[[2]], pts[[3]], Pd[1], Pa[1]};
Simplify[T.H.B];
Simplify[D[T.H.B, t]];
g1 = ParametricPlot[T.H.B, {t, 0, 1}, PlotRange -> All];

B = {pts[[3]], pts[[4]], Pd[2], Pa[2]};
Simplify[T.H.B];
Simplify[D[T.H.B, t]];
g2 = ParametricPlot[T.H.B, {t, 0, 1}, PlotRange -> All];
g3 = Graphics[{Red, AbsolutePointSize[6],
  Table[Point[pts[[i]]], {i, 1, 5}]}];
Show[g1, g2, g3, PlotRange -> All]
```

Figure 12.24: Effects of the Three Parameters in the Kochanek–Bartels Spline.

12.9 Fitting a PC to Experimental Points

The spline methods discussed so far use data points. The curve methods of Chapters 13 and 14 use control points. In the case of data points, the curve has to pass through all of them. Control points exert a pull on the curve, so each of them pulls the curve toward itself (Section 8.6). The method described here, due to [Plass and Stone 83], uses *experimental points* ("epoints" for short). Such points are typically obtained as a result of a science experiment, but can also be input by scanning an image. Given n epoints $\mathbf{P}_1$, $\mathbf{P}_2$,..., $\mathbf{P}_n$, the problem is calculating a PC curve that will pass close to all the points but will not necessarily pass *through* them. A user-controlled tolerance parameter controls the closeness of the fit.

Since a PC is fully defined by means of just four coefficients, it cannot have a very complex shape, so it may not be able to follow a set of epoints that meander all over the place. In such a case, the curve will have to be calculated as a spline where each segment is a PC and the segments fit together, either smoothly or with corner joints. In this section, we show how to calculate one such PC, so we assume that the n epoints do not describe a complex curve. To distinguish between simple and complex curves quantitatively, we connect the n epoints with $n-1$ straight segments, resulting in an open polygon. Experience shows that the set of points is simple and will allow a PC to follow it if (1) all the angles between consecutive segments are in the range $[135°, 180°]$, (2) the curve has at most one loop, and (3) if it does not have a loop, it can have at most two inflection points.

We denote our single PC segment by

$$\begin{aligned}\mathbf{P}(t) &= \big(u(t), w(t)\big)\\ &= \mathbf{a}_3t^3+\mathbf{a}_2t^2+\mathbf{a}_1t+\mathbf{a}_0\\ &= (a_{3x},a_{3y})t^3+(a_{2x},a_{2y})t^2+(a_{1x},a_{1y})t+(a_{0x},a_{0y}),\end{aligned} \tag{12.58}$$

where the four vector quantities **a**, **b**, **c**, and **d** have to be determined. Together, they constitute eight numbers, so we can say that a PC segment has eight degrees of freedom. To understand the method, let's imagine that we have somehow found a PC segment $\mathbf{P}(t)$ that passes close to all n epoints. We can use this PC to find the n values of the parameter t where the curve passes closest to each of the n epoints. Denoting these values by t_1, t_2,..., t_n, we use them to label the epoints $\mathbf{P}_{t_1}$, $\mathbf{P}_{t_2}$,..., $\mathbf{P}_{t_n}$. Now imagine the opposite situation where we still don't know the PC segment, but we already have the epoints somehow labeled correctly. Using the coordinates of the n epoints and the n values of t, we could, in such a case, calculate a curve using the least-squares fitting technique.

The idea is to start with an initial set of estimated t values, use least squares to calculate a PC segment from this set, use this PC to calculate a better set of t values, and repeat until the curve obtained is close enough to all the epoints. Convergence is not guaranteed, but experience shows that epoints that satisfy the three conditions stated earlier normally result in a reasonably shaped curve in just a few iterations.

The initial set of estimated t values is based on the lengths of the polygon's edges. Denoting the polygon vertices (i.e., the epoints) by $\mathbf{P}_i=(x_i,y_i)$, we define a quantity

s_k as the sum of the polygon's edges from $\mathbf{P}_1$ to $\mathbf{P}_k$:

$$s_1 = 0, \quad s_k = \sum_{i=1}^{k-1} |\mathbf{P}_{i+1} - \mathbf{P}_i| = \sum_{i=1}^{k-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad k = 2, 3, \ldots, n.$$

The initial value of t_k is now defined as the ratio s_k/s_n, resulting in $t_1 = 0$, $t_n = 1$, and, in general, $0 \le t_i \le 1$.

◇ **Exercise 12.13:** Given the eight epoints $\mathbf{P}_1 = (2, 5)$, $\mathbf{P}_2 = (2, 8)$, $\mathbf{P}_3 = (5, 11)$, $\mathbf{P}_4 = (8, 8)$, $\mathbf{P}_5 = (11, 4)$, $\mathbf{P}_6 = (14, 8)$, $\mathbf{P}_7 = (13, 8)$, and $\mathbf{P}_8 = (11, 10)$, draw them in the xy plane, draw the open polygon connecting them, indicate the "bad" polygon vertices, and calculate the quantities s_k and t_k.

Once a set of t_i values is available, a PC curve segment can be calculated by least squares. The principle is to compute values for the four coefficients $\mathbf{a}_i$ that will minimize the expression

$$S(\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) = \sum_{j=1}^{n} (\mathbf{P}(t_j) - \mathbf{P}_j)^2 = \sum_{j=1}^{n} \left(\sum_{i=0}^{3} \mathbf{a}_i t_j^i - \mathbf{P}_j \right)^2.$$

We consider this expression a function S of the four coefficients $\mathbf{a}_i$ and minimize it by (1) writing the four partial derivatives of S,

$$\frac{\partial S(\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)}{\partial \mathbf{a}_k} = \sum_{j=1}^{n} 2 \sum_{i=0}^{3} \left(\mathbf{a}_i t_j^i - \mathbf{P}_j \right) t_j^k, \quad 1 \le k \le 4,$$

(2) equating each to zero,

$$\sum_{i=0}^{3} \left(\sum_{j=1}^{n} t_j^i t_j^k \right) \mathbf{a}_i = \sum_{j=1}^{n} \mathbf{P}_j t_j^k, \quad 1 \le k \le 4,$$

which can also be written

$$\begin{aligned} &\mathbf{a}_0(t_1^0 t_1^k + t_2^0 t_2^k + \cdots + t_n^0 t_n^k) + \mathbf{a}_1(t_1^1 t_1^k + t_2^1 t_2^k + \cdots + t_n^1 t_n^k) \\ &\quad + \mathbf{a}_2(t_1^2 t_1^k + t_2^2 t_2^k + \cdots + t_n^2 t_n^k) + \mathbf{a}_3(t_1^3 t_1^k + t_2^3 t_2^k + \cdots + t_n^3 t_n^k) \\ &\quad = \mathbf{P}_1 t_1^k + \mathbf{P}_2 t_2^k + \cdots + \mathbf{P}_n t_n^k, \quad 1 \le k \le 4, \end{aligned} \tag{12.59}$$

and then (3) solving the resulting system of four linear equations in the four unknowns $\mathbf{a}_i$.

Having produced values for the four coefficients $\mathbf{a}_i$, we use the resulting PC to calculate a better set of t values. For each epoint, we find the value of t that produces the point nearest it on the PC and assign that t value to the epoint. Mathematically, this amounts to finding the minimum distance between an epoint $\mathbf{P}_j = (x_j, y_j)$ and the curve $\mathbf{P}(t) = (u(t), w(t))$. Since the distance involves a square root, we use the square

of the distance (a similar method is used in the Bresenham–Michener circle method, Section 3.8.3).

Our problem is, therefore, to minimize the function

$$D(t) = |\mathbf{P}(t) - \mathbf{P}_j| = \big(u(t) - x_j\big)^2 + \big(w(t) - y_j\big)^2,$$

and we do this by differentiating it with respect to t, equating the derivative to zero, and solving for t. Thus,

$$2(u(t) - x_j)u^t(t) + 2(w(t) - y_j)w^t(t) = 0. \tag{12.60}$$

Since $u(t)$ and $w(t)$ are cubic polynomials in t, their derivatives are quadratic polynomials. The left side of Equation (12.60) is thus a degree-5 polynomial in t, so a numerical solution is required. We use the Newton–Raphson method, a general, fast, iterative method for finding roots of functions. Given a function $f(t)$, the method requires an initial value of t (a guess or an estimate) and updates this value by the iteration

$$t \leftarrow t - \frac{f(t)}{f'(t)}.$$

If the initial value is close to a root, convergence is fast but is not guaranteed. In our case, function $f(t)$ is given by Equation (12.60), and we always have an estimate for t. Our Newton–Raphson iteration thus becomes

$$t \leftarrow t - \frac{2(u(t) - x_j)u^t(t) + 2(w(t) - y_j)w^t(t)}{u^t(t)^2 + w^t(t)^2 + (u(t) - x_j)u^{tt}(t) + (w(t) - y_j)w^{tt}(t)}.$$

Experience shows that one iteration is enough to produce a new t value much better than its predecessor.

The new t values may be located outside the interval $[0, 1]$, so one last step is needed, where all n new t values are linearly scaled to bring them back into the right interval, if necessary. Here is how it's done.

If the new t_1 is positive, it should not be scaled or changed in any way. This is the algorithm's way of telling us that a better fit would be achieved if the curve did not start at the first epoint. However, if t_1 becomes negative (e.g., if $t_1 = -\alpha$), it should be incremented by α to bring it back to zero, and all the other t_i's should be incremented by quantities that get smaller with i until they reach zero for $i = n$ (i.e., t_n should not be changed). Similarly, if the new t_n is less than 1, it should not be scaled, but if it exceeds 1 (by a quantity β), it should be decremented by β and all the other t_i's should be decremented by quantities that get smaller with i until they reach zero for $i = 1$.

Once this is grasped, it is easy to guess how a general value t_i should be scaled. It should be incremented by α multiplied by some weight and decremented by β multiplied by another weight, such that the weights add up to 1. If the new t_1 is positive, α should be set to 0. Similarly, if the new t_n is less than 1, β should be set to 0. The result is

$$t_i \leftarrow t_i + \alpha\frac{n-i}{n-1} - \beta\frac{i-1}{n-1}. \tag{12.61}$$

◇ **Exercise 12.14:** Given the eight new t values, -0.1, 0.1, 0.2, 0.3, 0.4, 0.6, 0.8, and 1.2, use Equation (12.61) to scale them.

These are the steps of the iteration. The loop continues until none of the t values changes significantly or, alternatively, until the maximum distance between an epoint and the curve falls below a preset threshold. If this does not happen after a certain, fixed number of iterations, the loop stops and displays an error message (curve does not converge to epoints). Figure 12.25 is an example of a spline fitting a set of 20 epoints. It is easy to see how the fit improves even after a small number of iterations.

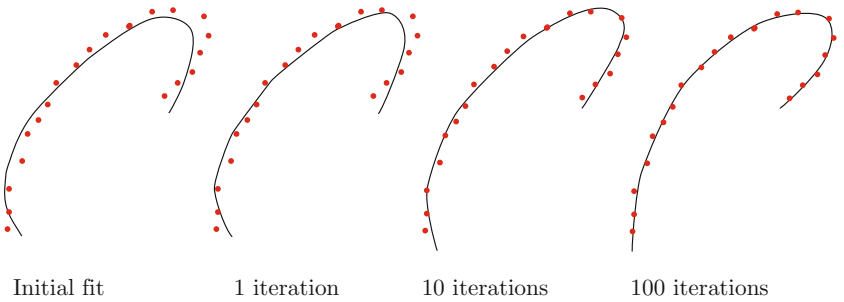


Figure 12.25: Spline Fit to Many Epoints.

We next discuss how to add constraints to the PC segment that's being calculated. When the initial t values are calculated by $t_k = s_k/s_n$, the first value, t_1, becomes zero, and the last value, t_n, is set to 1. The PC segment thus starts at the first epoint $\mathbf{P}_1$ and ends at $\mathbf{P}_n$. When the t values are updated in an iteration, both t_1 and t_n may get new values. If t_1 goes below zero, it is scaled back to zero. However, if it becomes positive (e.g., $t_1 = 0.05$), it is not changed. This means that point $\mathbf{P}(0.05)$ on the curve would be closest to $\mathbf{P}_1$. The start of the curve (point $\mathbf{P}(0)$) would, in this case, be located "before" $\mathbf{P}_1$. A similar situation may happen at the end of the curve, where $\mathbf{P}(1)$ may move "past" the last epoint $\mathbf{P}_n$.

Fitting a PC segment to a set of epoints in this way generally means that the curve may be "longer" than the set. Sometimes, we want the curve to start and end at the two extreme epoints, so we have to "constrain" it. Another aspect of constraining arises when we are given a complex set of epoints, where more than one PC segment is needed to fit all the points. In such a case, we have to consider the problem of joining individual segments. A segment may therefore have to be constrained by specifying its start and/or end tangent vectors.

The point to understand is that each added constraint reduces the quality of the fit. The reason is that a PC depends on four vector coefficients, which constitute eight scalar quantities (it has eight degrees of freedom). Adding a constraint means fixing one or more of those quantities, thereby reducing the number of degrees of freedom, and thus leading to a worse fit. The number of constraints should, therefore, be kept small (no more than one or two).

Adding constraints is done by generalizing the cubic polynomials $u(t)$ and $w(t)$. Instead of writing them in the form

$$\big(u(t), w(t)\big) = \mathbf{a}_3 t^3 + \mathbf{a}_2 t^2 + \mathbf{a}_1 t + \mathbf{a}_0,$$

we express them as

$$\big(u(t), w(t)\big) = \mathbf{a}_1 F_1(t) + \mathbf{a}_2 F_2(t) + \mathbf{a}_3 F_3(t) + \mathbf{a}_4 F_4(t),$$

where the $F_i(t)$ are any four linearly independent cubic polynomials. Both $u(t)$ and $w(t)$ remain cubic polynomials, but certain choices of the $F_i(t)$'s may make it easy to constrain the endpoints or the extreme tangents of the PC segment.

One such choice is the set of four *Hermite blending functions* of Equation (11.6), duplicated here:

$$\begin{aligned} F_1(t) &= 2t^3 - 3t^2 + 1, \quad F_2(t) = -2t^3 + 3t^2, \\ F_3(t) &= t^3 - 2t^2 + t, \quad F_4(t) = t^3 - t^2. \end{aligned} \tag{11.6}$$

We know from Equation (11.5) that if a PC segment $\mathbf{P}(t)$ is expressed as the weighted sum

$$\mathbf{P}(t) = \mathbf{a}_1 F_1(t) + \mathbf{a}_2 F_2(t) + \mathbf{a}_3 F_3(t) + \mathbf{a}_4 F_4(t), \tag{12.62}$$

then $\mathbf{a}_1$ and $\mathbf{a}_2$ are the endpoints of the segment, and $\mathbf{a}_3$ and $\mathbf{a}_4$ are its two extreme tangents. We can now add constraints by using Equation (12.62) instead of Equation (12.58) and preassigning values to some of the four $\mathbf{a}_i$ coefficients. For example, if we want the initial tangent vector to be in the "up" direction $(0, 1)$, we assign $\mathbf{a}_3$ the value $(0, 1)$ and end up with Equation (12.59) becoming a system of three equations in the three unknowns $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_4$. It is now obvious that the more constraints (i.e., the more coefficients are assigned values and eliminated from Equation (12.59)), the fewer are the possibilities for fitting the PC segment to the epoints. There is, therefore, a trade-off between a good fit and more constraints.

> ... and then in midair the elasticity makes the shape rebound, so what you have is not a circle but some linked spline curves, not exactly symmetrical, because the ball flattens on one side ...
>
> —John Updike, *Roger's version* (1996)