Cubic spline curves

1. Overview

Polynomial parametric curves of high degree have a disadvantage: Requirements placed on one stretch of such a curve can have a very strong effect some distance away. In Figure 1, the jump in the height of the data points near the middle has a strong effect on the interpolating polynomial curve near the ends.

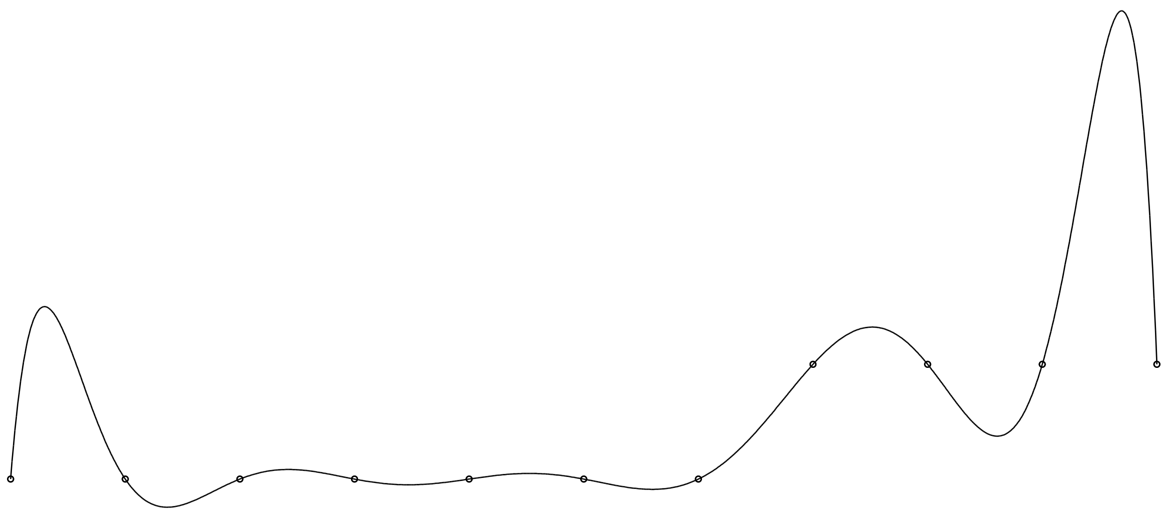

Figure 1: Lagrange interpolation of data points

In contrast, Figure 2 shows an example of a "cubic spline" curve through the same data points. Notice how it follows them much more closely.

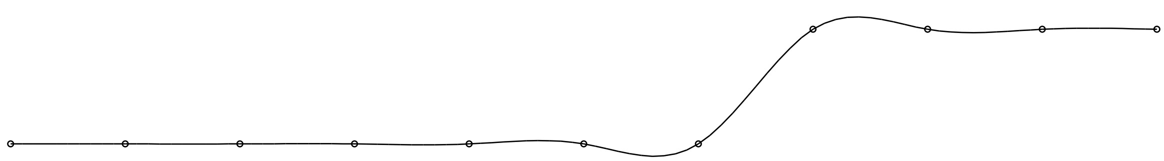

Figure 2: Spline interpolation of the same data points

The spline curve was constructed by using a different cubic polynomial curve between each two data points. In other words, it is a *piecewise cubic* curve, made of pieces of different cubic curves glued together. The pieces are so well matched where they are glued that the gluing is not obvious.

In fact, if the whole curve shown is described with a single function $P(t)$, then $P(t)$ is so smooth that it has a second derivative everywhere and this derivative is continuous.

Definition. A *cubic spline* curve is a piecewise cubic curve with continuous second derivative.

The word "spline" actually refers to a thin strip of wood or metal. At one time curves were designed for ships and planes by mounting actual strips of wood or metal so that they went through the desired data points but were otherwise free to move. For reasons of physics, such curves are approximately piecewise cubic with continuous second derivative, if they are suitably parameterized.

You may recall from calculus that the curvature of a curve at each point depends on the second derivative there. At the end points, an actual wood or metal strip has no reason to bend, and the second derivative of its curve is zero.

Definition. A cubic spline curve is *relaxed* if its second derivative is zero at each endpoint.

We shall concentrate on relaxed cubic spline curves. As you will see, they can be used either for controlled design (B-splines) or for interpolation. To describe the cubic pieces simply and conveniently, we shall use cubic Bézier curves.

2. Bézier curves with zero second derivative at one end

In order to handle the relaxed end conditions, we shall need to be able to tell when a Bézier curve has zero second derivative at one end. Recall that for a cubic Bézier curve $P(t)$ with control points P_0, P_1, P_2, P_3,

$$P''(0) = 6(P_0 - 2P_1 + P_2).$$

This quantity is zero when $2P_1 = P_0 + P_2$, or equivalently, when

$$P_1 = \tfrac{1}{2}P_0 + \tfrac{1}{2}P_2.$$

A similar relation holds in case $P''(1) = 0$. Even more simply:

Observation. $P''(0) = 0$ if and only if P_1 is the midpoint of the segment $\overline{P_0P_2}$; $P''(1) = 0$ if and only if P_2 is the midpoint of the segment $\overline{P_1P_3}$. Some examples are shown in Figure 3.

3. Gluing two Bézier curves

First attempt: Matching endpoints

Let us start with two Bézier curves that can be glued together but otherwise are not well matched. Let the first curve have control points P_0, P_1, P_2, P_3 and let the second curve have control points Q_0, Q_1, Q_2, Q_3, as shown

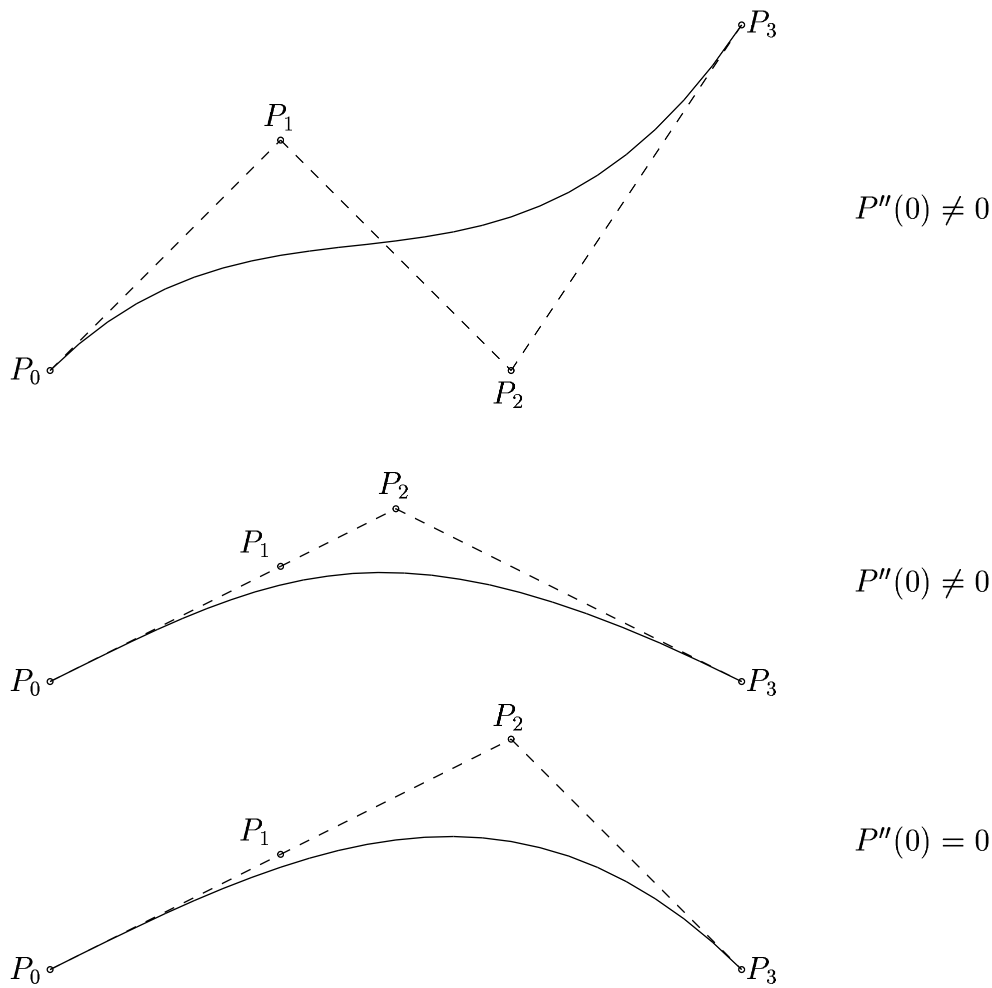

Figure 3: Examples of second derivatives

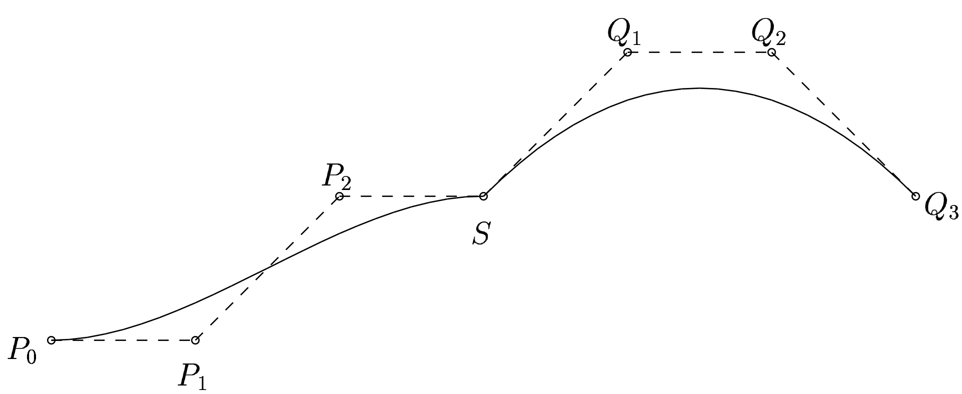

Figure 4: A coarse gluing

in Figure 4. Suppose that $P_3 = Q_0$. For convenience let this point of joining also be called S; i.e., $S = P_3 = Q_0$. The result is shown in Figure 4.

The curve has a corner, because at S the first Bézier curve has first derivative $3(S - P_2)$ and the other has first derivative $3(Q_1 - S)$, but the vectors $S - P_2$ and $Q_1 - S$ do not even have the same direction. (Recall that subtracting one point from another, say $A - B$, gives the vector from B to A.)

Second attempt: Matching values and first derivatives

A much better join is obtained if we require that $S - P_2 = Q_1 - S$, or equivalently, that S be the midpoint of the line segment $\overline{P_2Q_1}$, so that the first derivatives match at the point of gluing. Figure 5 shows an example where this condition is met.

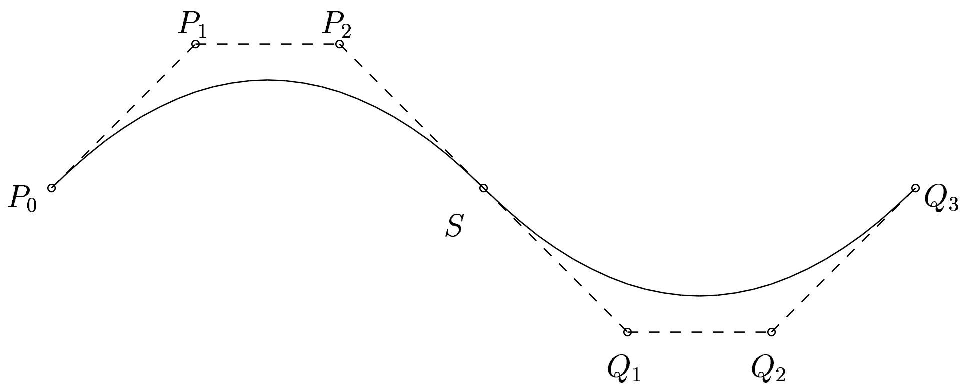

Figure 5: A better gluing, matching first derivatives

This example certainly looks smoother. However, it is still not ideal. Imagine taking a fast train ride along a track of this shape. On the first Bézier segment you are curving so that you are pushed against the left wall of the train; on the other you are curving the other way and are pushed against the right wall. At the point of joining you are jerked from one side of the train to the other. For an even smoother join, then, the curvature should be continuous. Because the curvature can be expressed in terms of the first and second derivatives, continuity of curvature can be achieved by matching second derivatives, as well as first derivatives, at the point of gluing.

Third attempt: Matching values, first derivatives, and second derivatives

Recall that at S, where $t = 1$ for the first Bézier curve and $t = 0$ for the other, the second derivatives of the Bézier curves are respectively $6(P_1 - 2P_2 + S)$ and $6(S - 2Q_1 + Q_2)$. Thus we want

$6(P_1 - 2P_2 + S) = 6(S - 2Q_1 + Q_2)$, or equivalently, $P_1 - 2P_2 = Q_2 - 2Q_1$.

Here is an interesting way of interpreting this equation. Negate both sides to get $2P_2 - P_1 = 2Q_1 - Q_2$. The motivation for doing this is that both sides now have coefficients summing to 1 and so should represent points, independently of coordinatization.

The left-hand side corresponds to a particular point A_+ on the line through P_1 and P_2. In fact, $A_+ = 2P_2 - P_1 = P_2 + 1 \cdot (P_2 - P_1)$, as shown in Figure 7. Let us call A_+ the *right apex* of the first control polygon. Similarly, the right-hand side of the equation is the *left apex* $A_- = 2Q_1 - Q_2$ of the second Bézier curve, as shown.

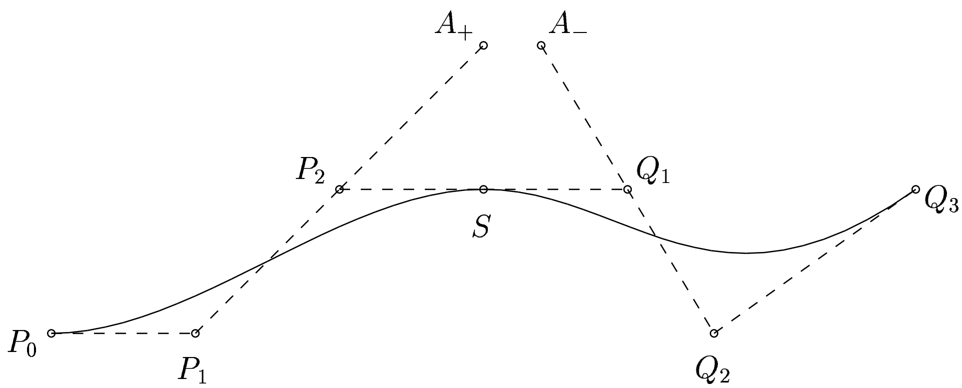

Figure 6: A gluing almost matching second derivatives

As you see, in this example the two apexes are not equal, so the equation is not satisfied and the second derivatives at S of the two Bézier curves still do not match. Figure 7 shows an example where they do match, with both apexes being at a common point A:

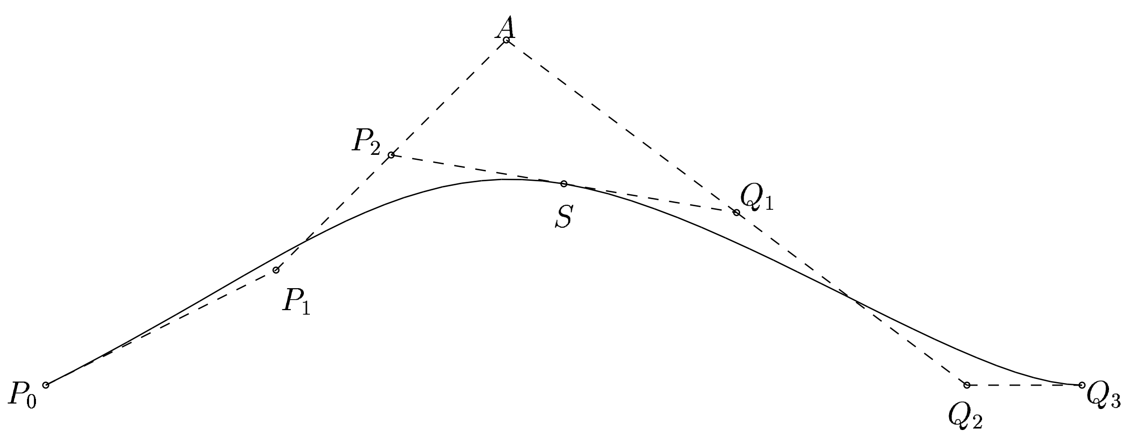

Figure 7: A gluing matching second derivatives

The relevant part of Figure 7 looks like the letter A or like an A-frame cabin.

Definition. An $A - frame$ is a figure with points as indicated, in which S is the midpoint of $\overline{P_2Q_1}$, P_2 is the midpoint of $\overline{P_1A}$, and Q_1 is the midpoint of $\overline{Q_2A}$. An example is shown in Figure 8.

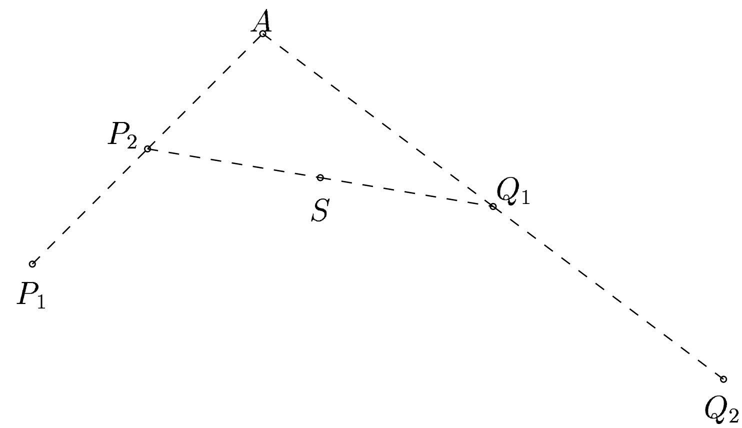

Figure 8: An A-frame

Thus we see:

Observation. If two Bézier curves are joined at a point S, both their first and second derivatives match at S if and only if their control polygons fit an A-frame.

Note: Matching third derivatives as well sounds promising, but is not helpful, as it forces both curves to be parts of just one third-degree curve. Thus the flexibility obtained by gluing curves is lost. See the Exercises.

4. B-spline curves

An easy way of making a controlled-design curve with many control points is to use B-spline curves. The ones we shall discuss are called *relaxed uniform cubic B-spline* curves. You start by specifying a control polygon of points $B_0, B_1, \ldots B_n$, and you end by getting a curve like the one in Figure 9.

Here is the method, if done by hand: Divide each leg of the control polygon in thirds by marking two "division" points. At each B_i except the first and last, draw the line segment between the two nearest "division" points, and call the midpoint S_i. Then you have made an A-frame with B_i at the apex, as shown. For completeness, let $S_0 = B_0$ and $S_n = B_n$. See if you can locate the four A-frames in Figure 10.

Finally, sketch a cubic Bézier curve from each point S_i to the next, using as Bézier control points the four points S_i, two "division" points, and S_{i+1}, as in Figure 11.

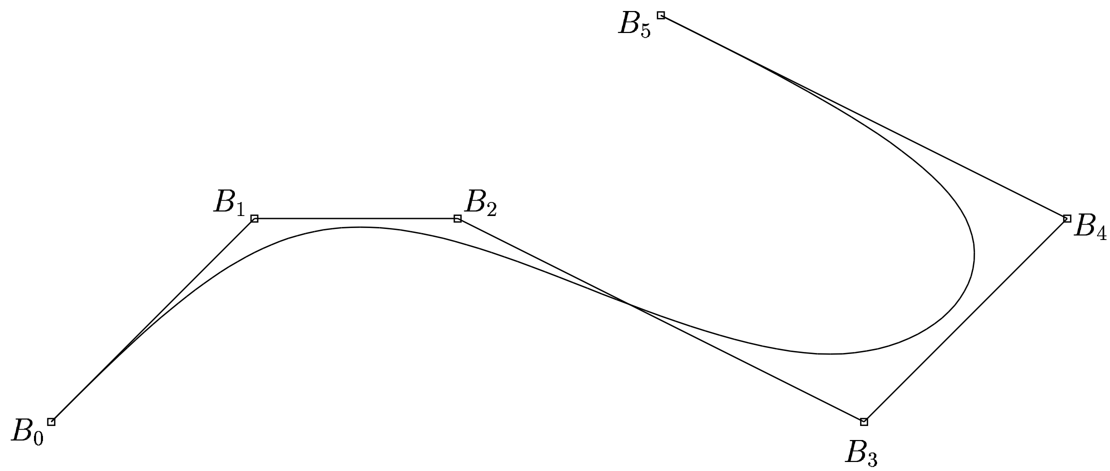

Figure 9: A relaxed uniform cubic B-spline curve

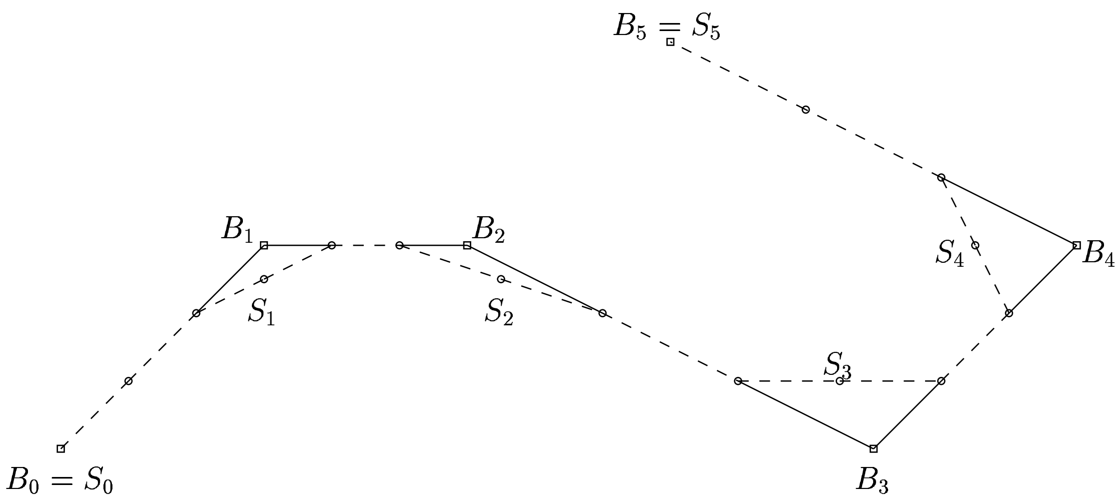

Figure 10: A-frames for a relaxed cubic B-spline

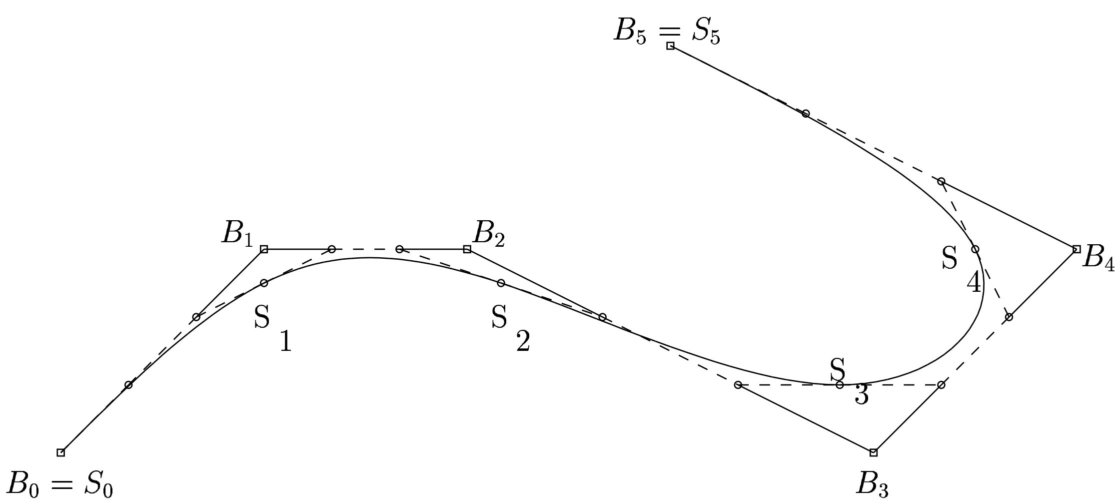

Figure 11: Construction of a relaxed cubic B-spline

As you see, the points of gluing meet the A-frame condition automatically and at the ends the second derivative is zero. Therefore you obtain a relaxed cubic spline curve.

The method as performed on a computer is the same; we merely need to find the Bézier control points in terms of the original B-spline control points:

The "division" points on the line segment from B_{i-1} to B_i are $\frac{2}{3}B_{i-1}+\frac{1}{3}B_i$ and $\frac{1}{3}B_{i-1}+\frac{2}{3}B_i$. Also, S_i is the average of the ends of its "cross-segment", so that

$S_i = \frac{1}{2}(\frac{1}{3}B_{i-1}+\frac{2}{3}B_i)+\frac{1}{2}(\frac{2}{3}B_i+\frac{1}{3}B_{i+1}) = \frac{1}{6}B_{i-1}+\frac{2}{3}B_i+\frac{1}{6}B_{i+1}$, for $i = 1,\ldots,n-1$.

To summarize the computer method:

Given B-spline control points $B_0,\ldots,B_n$, calculate $S_i = \frac{1}{6}B_{i-1}+\frac{2}{3}B_i+\frac{1}{6}B_{i+1}$, for $i=1,\ldots,n-1$, and let $S_0 = B_0$, $S_n = B_n$. There are n Bézier curves to plot; curve $\#i$ has control points S_{i-1}, $\frac{2}{3}B_{i-1}+\frac{1}{3}B_i$, $\frac{1}{3}B_{i-1}+\frac{2}{3}B_i$, and S_i. On curve $\#i$, you can plot points on the curve for, say, $t = 0, .05, .10, \ldots, .95, 1$.

Finally, let's consider the situation mathematically. Let $p_i(t)$ be the ith Bézier curve $(0 \le t \le 1)$. These n curves can be combined into a single curve $P(t)$ for $0 \le t \le n$ by letting

$P(t) = p_1(t)$ for $0 \le t \le 1$,

$P(t) = p_2(t-1)$ for $1 \le t \le 2$, etc. In general,

$P(t) = p_i(t-(i-1))$ for $i-1 \le t \le i$, where $i = 1,\ldots,n$.

Then $P(t)$ is a relaxed cubic spline curve. $P(t)$ is called a *uniform* spline curve because its domain $0 \le t \le n$ was made from intervals all of length 1. Non-uniform curves will be considered in Section 10.

An important virtue of B-spline curves is that the influence of individual control points is *local*. In fact, any one point on the curve is influenced by at most four of the B-spline control points. The reason is that for Bézier curve $\#i$, all four control points can be computed from a knowledge of B_{i-2}, B_{i-1}, B_i, and B_{i+1}. Similarly, control point B_i influences only four Bézier-curve segments: the two that join at S_i and the two additional ones joined to those. The local effect can be illustrated by changing a single control point. In Figure 12, two choices of the middle control point are indicated, along with the corresponding B-spline curves. Dots on the curves indicate some gluing points S_i.

What if you don't want relaxed end conditions? In that case, you can just use less of the curve, say the part from S_1 to S_{n-1}, i.e., $1 \le t \le n$. B_0 and B_n can still be used as control points to affect the shape of the part of the curve that you are using.

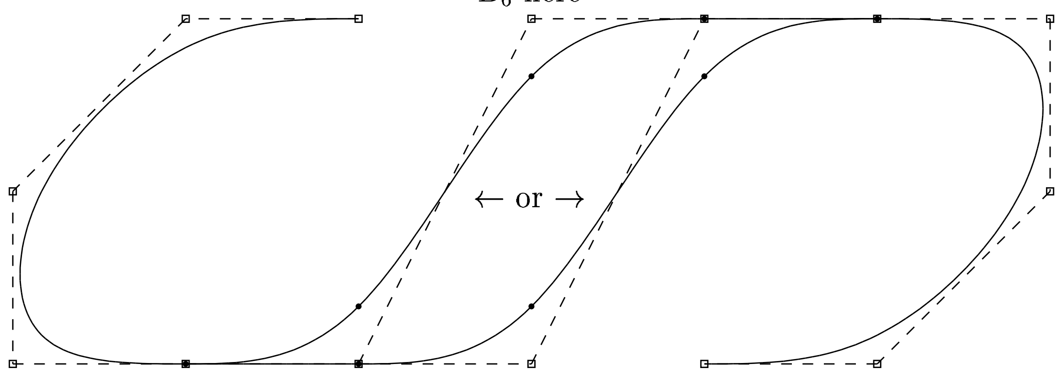

Figure 12: The effect of varying one control point

5. Interpolation by relaxed cubic splines

Suppose that you are interested not in controlled design but in interpolation. In other words, data points $S_0, \ldots, S_n$ are given and you want a relaxed cubic spline curve $P(t)$ for $0 \le t \le n$ such that $P(i) = S_i$; i.e., the curve goes *through* the data points.

One easy approach is to use B-splines as an intermediate step. This time, though, you know the points $S_0, \ldots, S_n$ and you must compute the appropriate control points $B_0, \ldots, B_n$ before you will be able to compute the Bézier control points for the individual pieces.

Of course, $B_0 = S_0$ and $B_n = S_n$. To find $B_1, \ldots, B_{n-1}$, we can use the linear equations already found above for the S_i in terms of the B_i and solve them treating the B_i as unknowns and the S_i as constants. When the linear equations are written in matrix form, they look like this (for $n = 5$):

$$\begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} (6S_1 - S_0) \\ 6S_2 \\ 6S_3 \\ (6S_4 - S_5) \end{bmatrix}$$

Here the B_i and S_i are *points*, so that in $\mathbf{R}^2$ they are *pairs* of numbers. These equations are equivalent to two sets of equations with the same coefficient matrix, one set for x coordinates and one for y coordinates. To solve them, though, it is easiest to use one of two more direct methods. Let M be the matrix of coefficients (which we can call the "1 4 1 matrix"), let B_* be the matrix whose rows are $B_1, \ldots B_{n-1}$, and let C be the matrix of constants on the right.

Method 1 (on a computer): Make the augmented matrix $[M|C]$ and row-reduce it completely. The answer will be $[I|B_*]$.

Method 2 (on homework problems or tests): You will be given M^{-1}; the solution is then $B_* = M^{-1}C$.

Note: The $(n-1) \times (n-1)$ matrix M is nonsingular for any n. In general, a matrix is guaranteed to be well behaved if its diagonal entries are all large in absolute value compared to the sums of the absolute values of the off-diagonal entries in each row.

Problem. Find Bézier control points for the four segments of the relaxed cubic spline curve through the data points $S_0 = (1, -1)$, $S_1 = (-1, 2)$, $S_2 = (1, 4)$, $S_3 = (4, 3)$, $S_4 = (7, 5)$, as shown in Figure 13. Useful information:

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}^{-1} = \frac{1}{56}\begin{bmatrix} 15 & -4 & 1 \\ -4 & 16 & -4 \\ 1 & -4 & 15 \end{bmatrix}.$$

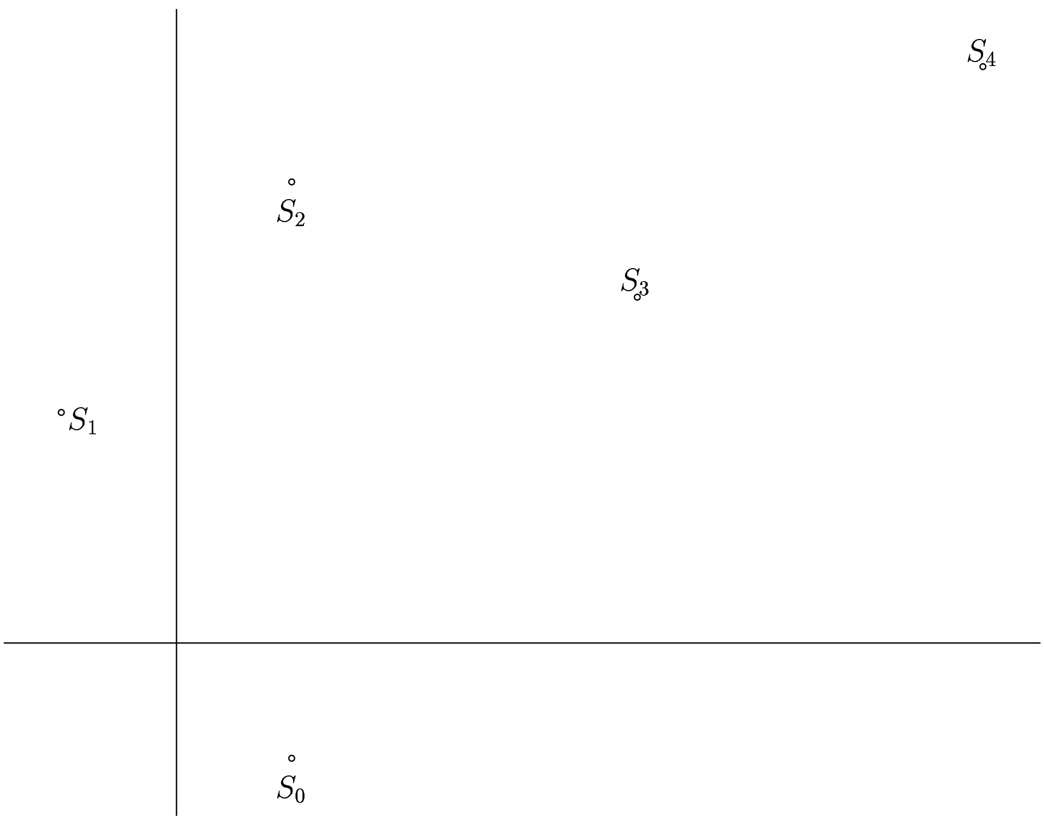

Figure 13: Data points for interpolation

Solution. $n = 4$, so the "1 4 1" equations in matrix form are 3×3:
$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} (6S_1 - S_0) \\ 6S_2 \\ (6S_3 - S_4) \end{bmatrix} = \begin{bmatrix} -7 & 13 \\ 6 & 24 \\ 17 & 13 \end{bmatrix}$. Solving these equations by using the matrix inverse, we get

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \frac{1}{56} \begin{bmatrix} 15 & -4 & 1 \\ -4 & 16 & -4 \\ 1 & -4 & 15 \end{bmatrix} \begin{bmatrix} -7 & 13 \\ 6 & 24 \\ 17 & 13 \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ 1 & 5 \\ 4 & 2 \end{bmatrix}$$

Therefore the B-spline control points are $B_0 = S_0 = (1, -1)$, $B_1 = (-2, 2)$, $B_2 = (1, 5)$, $B_3 = (4, 2)$, $B_4 = S_4 = (7, 5)$. The four sets of cubic Bézier control points are as follows, and together they give Bézier curves that go together to make the curve shown in Figure 14.

Bézier #1:	$(1, -1)$	$(0, 0)$	$(-1, 1)$	$(-1, 2)$
Bézier #2:	$(-1, 2)$	$(-1, 3)$	$(0, 4)$	$(1, 4)$
Bézier #3:	$(1, 4)$	$(2, 4)$	$(3, 3)$	$(4, 3)$
Bézier #4:	$(4, 3)$	$(5, 3)$	$(6, 4)$	$(7, 5)$

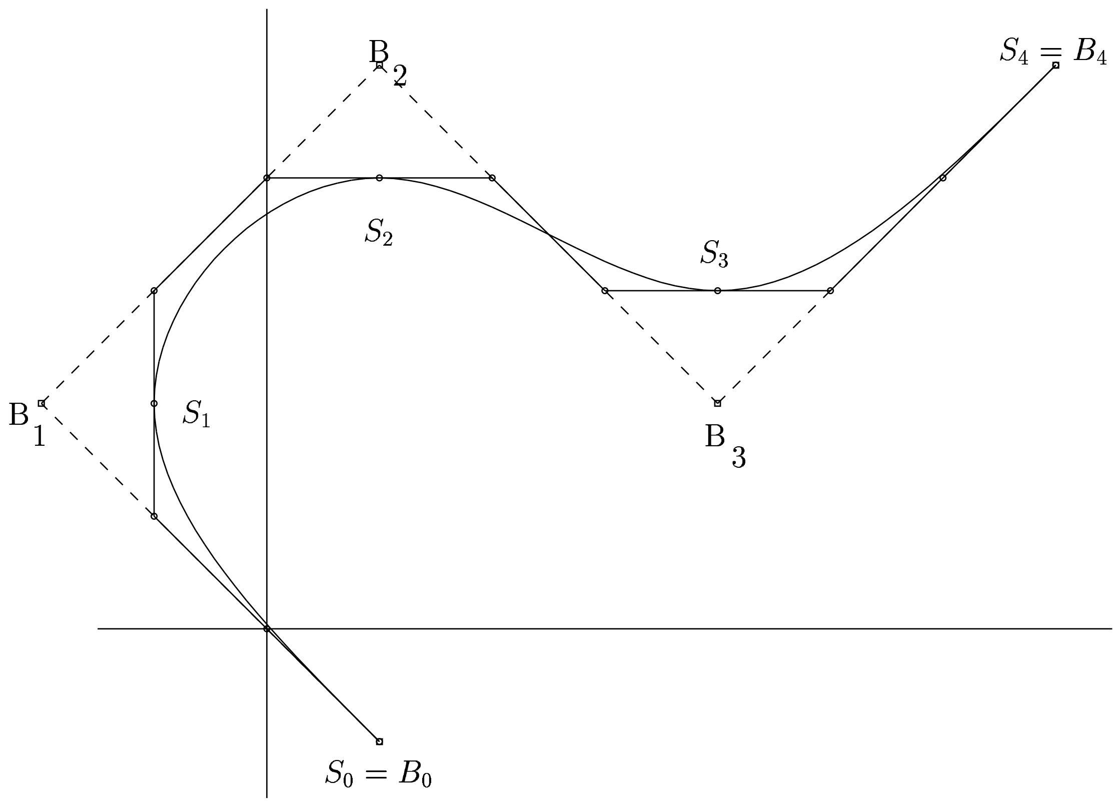

Figure 14: The interpolating relaxed B-spline curve

Derivation of the "1 4 1" *equations.* The first and last equations are different from the middle ones. For all equations, recall that $\frac{1}{6}B_{i-1} + \frac{2}{3}B_i + \frac{1}{6}B_{i+1} = S_i$. For the middle equations, just multiply this relation by 6 to get

$1 \cdot B_{i-1} + 4 \cdot B_i + 1 \cdot B_{i+1} = 6 \cdot S_i.$

For the first equation, $B_0 + 4B_1 + B_2 = 6S_1$; recall that $B_0 = S_0$ and subtract S_0 from both sides. The last equation is handled similarly, by using the fact that $B_n = S_n$.

Remark. As you see, we used B-splines to do interpolation. Usually when you hear someone talk about a "B-spline" problem, though, the control points B_i will be given; when you hear someone talk about a "spline interpolation" problem, the S_i will be given and the person may or may not be thinking of using B-splines to get the answer.

6. Higher dimensions and applications to animation.

Although we have been working in $\mathbf{R}^2$, everything that has been said so far applies in any number of dimensions. In m dimensions, the control points B_i or the data points S_i are in $\mathbf{R}^m$, and the values of $P(t)$ are also in $\mathbf{R}^m$.

The case $m = 3$ is just what you would think: The control points or data points give a curve in space.

Higher cases are useful in animation, as follows. Suppose that you have a series of cartoon frames representing the position of some character at times $t = 0, 1, \ldots, n$, and from them you would like to compute more frames in between to make a smooth-looking movie. In other words, you have some *key frames* and you want to interpolate more frames.

The first step is to represent all frames in numerical form, by choosing some uniform way of giving a list of numbers determining the position of the character. For example, suppose that the character is entirely made of straight lines between various vertices, and there are fifteen such vertices. Then each vertex can be described by two numbers, and the whole frame can be described by a list of thirty numbers.

The next step is conceptual—simply think of a frame as being *one* point in $\mathbf{R}^{30}$. Then your key frames are data points in $\mathbf{R}^{30}$, and the in-between frames will be on a curve in $\mathbf{R}^{30}$ that goes through the data points. To make such a curve, just use an interpolating relaxed cubic spline $P(t)$, following the method of §5.

The final step is to find the in-between frames. Their lists of numbers are found just by evaluating $P(t)$ for the desired values of t, perhaps every $\frac{1}{20}$-th of a time unit.

An example is shown in Figure 15, which was made by a former student in this course. The rows of frames should be regarded as being in one long sequence of frames. The key frames are indicated by an asterisk (*), and each time unit has been divided into six subintervals. A final key frame was used but is not shown.

Notes.

- It might be tempting to dispense with splines and just interpolate linearly between successive key frames. For example, at time $t = \frac{1}{3}$, just

Figure 15: An example of animation, by Sean Meyn

take $\frac{2}{3}$ of each number from the key frame at $t = 0$ plus $\frac{1}{3}$ of the corresponding number from the key frame at $t = 1$. However, this method is the same as connecting data points in $\mathbf{R}^{30}$ with straight lines, and the motion it produces will be very jerky.

- In animation, you must watch for cases where smooth second derivatives are not applicable. For example, when a character steps on the floor, there is a sudden stop of downward motion. Applying spline interpolation will produce a picture in which the foot continues below the floor!

7. A nonparametric version

The term "nonparametric" refers to the familiar situation $y = f(x)$ or $y = f(t)$, as opposed to the parametric situation $P(t) = (f_1(t), f_2(t), \ldots, f_m(t))$. A method for the non-parametric version gives a parametric method, and vice-versa, as follows:

If you have a nonparametric method, just apply it separately in each coordinate to obtain a parametric version. For example, if $n = 2$ and data points $(5, 2), (3, 7), (9, 1)$ are given, find $f_1(x)$ such that $f_1(0) = 5$, $f_1(1) = 3$, and $f_1(2) = 9$, and also $f_2(x)$ with values $2, 7, 1$ at those same x-values. Then let $P(t) = f_1(t), f_2(t)$. (Notice that t is now used in place of x.)

If you have a parametric method giving a curve $P(t)$ in $\mathbf{R}^m$, to get a nonparametric method just concentrate on the case $m = 1$. This would give you a point moving in one dimension, i.e., a moving number $x(t)$ in $\mathbf{R}$. However, you can think of the graph of x against t to get a better picture of the function. If you wish you can then put y for x and x for t to get a function $y = f(x)$.

The discussion so far applies to any kind of parametric versus nonparametric method, but it works in particular for uniform cubic B-splines and interpolating splines. In these cases, all functions involved are piecewise cubic. The method of Section 5 applies for interpolation, with plain numbers wherever points were mentioned. Instead of "Bézier curves" for the segments, we have simply Bézier functions. Instead of data points, we have simply data numbers.

8. Basis functions

Just to be specific, let's consider splines with $n = 4$. Make a nonparametric interpolating spline function $f_0(t)$ with data values $1, 0, 0, 0, 0$, using the method of Section 5. Then make a similar function $f_1(t)$ with data values $0, 1, 0, 0, 0$. Continuing this way you get functions $f_0(t), \ldots, f_4(t)$, as shown

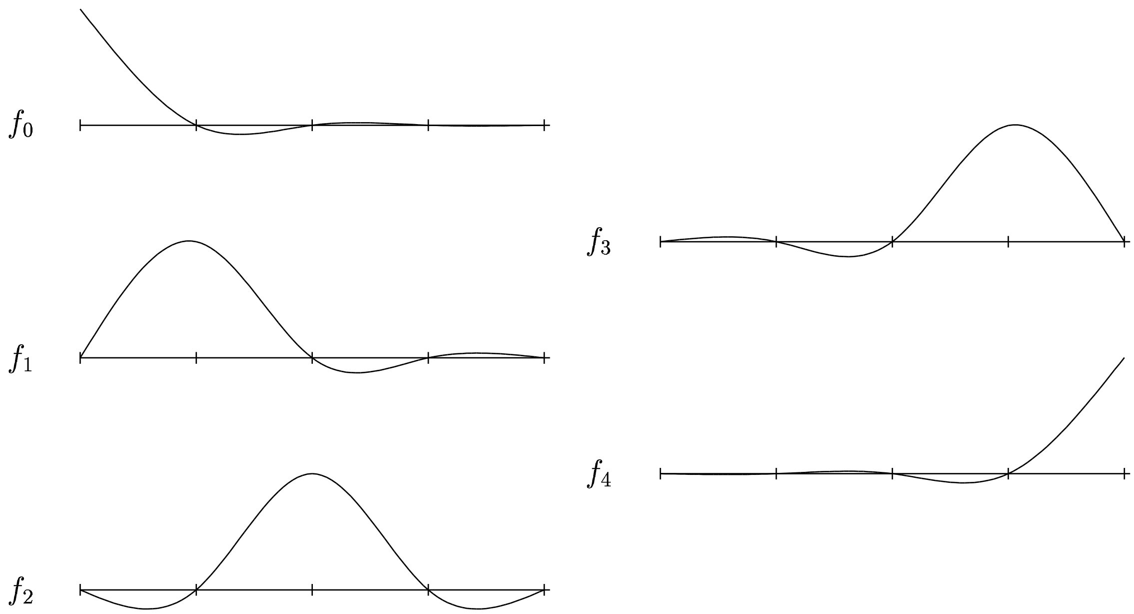

Figure 16: Basis functions for relaxed cubic splines, n=4

in Figure 16. In terms of linear algebra, these functions are a basis for the vector space of all uniform relaxed cubic spline functions with $n = 4$.

These basis functions have a practical use even for spline curves in higher dimensions with $n = 4$: Instead of computing interpolating splines directly, as in Section 5, just compute these basis functions. Then, given data points $S_0, \ldots, S_4$, just let $P(t) = f_0(t)S_0 + \ldots f_4(t)S_4$. This should be reminiscent of the construction of Lagrange polynomials, and the reasoning to show that they do interpolate the data points is the same as in the case of Lagrange.

This method is valuable when there is a reason to do as much pre-computing as possible, so that there is nothing left to do with each new set of data points except to take a linear combination at each time. Two occasions when there is such a reason are these:

- You have many different curves to do with different data points but always for $n = 4$.
- You need to interpolate in a space of high dimension, say for a complicated cartoon. This method avoids carrying the data through all the steps of row-reduction of the "1 4 1" matrix.

9. Other possible end conditions

Consider cubic spline curves that interpolate given data points. As we shall discuss below, you can specify two additional vector conditions. Before, these conditions were that the second derivative be zero at each end. Instead, we could make other requirements. One possibility would be to specify the *velocity* vector at each end, i.e., to specify that $P'(0) = \mathbf{v}$ and to specify that $P'(n) = \mathbf{w}$ for specific constant vectors $\mathbf{v}, \mathbf{w}$, instead of requiring that the second derivative be zero at each end. These are called "clamped" end conditions, because they force the curve to have a certain tangent direction at each end, as if they were held with a clamp. Since the ends are twisted by being clamped, we can no longer assume that the ends are relaxed.

Analysis of the clamped case.

We can no longer take $B_0 = S_0$ and $B_n = S_n$, since these choices are what result in having the second derivatives be $\mathbf{0}$ at each end. Therefore there are $n+1$ points B_i for which to solve. We need $n+1$ equations involving points. For $n-1$ of them, we can just use

$B_{i-1} + 4B_i + B_{i+1} = 6S_i$, as before (but now for $i = 1$ and $i = n-1$ as well). For the first end condition, use the first derivative property of Bézier curves: $P'(0) = 3(P_1 - P_0)$, which here is $3([\frac{2}{3}B_0 + \frac{1}{3}B_1] - S_0) = 2B_0 + B_1 - 3S_0$. This quantity is to equal $\mathbf{v}$. Therefore an equation is $2B_0 + B_1 = 3S_0 + \mathbf{v}$. Similarly, at the other end an equation is $B_{n-1} + 2B_n = 3S_n - \mathbf{w}$. In matrix form, the equations look like this (illustrated with $n = 4$):

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} (3S_0 + \mathbf{v}) \\ 6S_1 \\ 6S_2 \\ 6S_3 \\ (3S_4 - \mathbf{w}) \end{bmatrix}$$

The two extra conditions can be used other ways. If the first and last data points are the same, the two conditions can be used to equate the first derivatives of the two ends and also the second derivatives, so that a "periodic spline curve" is obtained. Another possibility is to require the third derivative to be continuous at S_1 and S_{n-1}.

10. Non-uniform spline curves

The spline curves we previously studied were of the form $P(t)$ for $0 \leq t \leq n$, where the points of gluing were $t = 1, 2, \ldots, n-1$. We say that the *knot points* are $0, 1, \ldots, n$. It is possible, more generally, to have knot points $t_0, \ldots, t_n$. As before, one can start with B-spline curves and then use them for spline interpolation. In this case, the simple graphical construction with A-frames is not possible, but there is a recursive construction that is not difficult to implement on the computer.

11. Problems

Problem DD-1. Sketch the relaxed cubic B-spline curve with control points $(0,0)$, $(1,0)$, $(1,1)$, $(0,1)$, $(0,0)$. (Calculate the Bézier control points, then sketch the Bézier curves freehand. Use a large enough scale that your sketch is meaningful.)

Problem DD-2. (a) In constructing a cubic B-spline curve from given control points $B_0, \ldots, B_n$, all the Bézier control points you generate are in the convex hull of the set of B_i. Why?

(b) Explain how it follows from (a) that the whole curve is in the convex hull of the set of B_i.

(c) If all the B_i are on your rectangular screen, is the whole cubic B-spline curve necessarily on the screen? (Why?)

Problem DD-3. (a) If you use B-spline control points $B_0, \ldots, B_{20}$ to get a B-spline curve $P(t)$ with $0 \le t \le n$, which B_i affect $P(5.3)$? (b) Which affect $P(6.0)$?

Problem DD-4. (a) For the relaxed cubic spline through data points S_0, S_1, S_2, find formulas for the B-spline and Bézier control points involved. (This is the case $n = 2$, so you won't need a matrix to solve equations. Your answers will be linear combinations of the data points.)

(b) By Lagrange, there is actually a single quadratic polynomial through the data points in (a). In general, will the two Bézier curves in (a) actually both match this one quadratic curve? (Say why or why not.)

(c) Sketch the relaxed cubic spline curve through data points $(1,0)$, $(1,1)$, $(0,1)$.

(d) Show that $P(\frac{1}{2}) = \frac{13}{32}S_0 + \frac{22}{32}S_1 - \frac{3}{32}S_2$ for the relaxed cubic spline curve $P(t)$ interpolating S_0, S_1, S_2, where $0 \le t \le 2$.

Problem DD-5. For the basis functions for relaxed cubic splines in the case $n = 2$, find their values at $t = \frac{1}{2}$. (You may use the results of Problem DD-4.)

Problem DD-6. Sketch the relaxed cubic spline through data points $(0,0)$, $(1,1)$, $(2,0)$, $(3,0)$. (Calculate the B-spline and Bézier control points, then sketch the Bézier curves freehand. Use a large enough scale that your sketch is meaningful. Useful information: $\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}^{-1} = \frac{1}{15}\begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix}$.)

Problem DD-7. Sketch the relaxed cubic spline through data points $(0,0)$, $(1,0)$, $(2,1)$, $(3,0)$, $(4,0)$. (Calculate the B-spline and Bézier control points, then sketch the Bézier curves freehand. Use a large enough scale that your sketch is meaningful.) Useful information: $\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}^{-1} = \frac{1}{56}\begin{bmatrix} 15 & -4 & 1 \\ -4 & 16 & -4 \\ 1 & -4 & 15 \end{bmatrix}$

Problem DD-8. Suppose two people are given control points $B_0, \dots, B_n$ and an affine transformation T (the same B_i and T for each person). The first person finds the B-spline curve $P(t)$ with control points B_i ($0 \le t \le n$) and then draws the curve $T(P(t))$. The second person finds the points $T(B_i)$ and uses them as control points to make a B-spline curve (also for $0 \le t \le n$). Do the two people get the same curve? (Explain.)

Problem DD-9. Suppose three consecutive B-spline control points are evenly spaced on a straight line. (a) Describe the A-frame of the middle control point of the three. (b) What is the second derivative of the curve at the middle control point?

Problem DD-10. For the Bézier curves making up a B-spline curve, each Bézier curve depends on only several of the control points B_i. Therefore there is no harm in using an infinite list of control points. In fact, the list could be $\dots, B_{-1}, B_0, B_1, B_2, \dots$.

(a) Sketch the B-spline curve you get if the infinite list of B_i keeps going around the corners of a Box: $B_0 = (1,1)$, $B_1 = (-1,1)$, $B_2 = (-1,-1)$, $B_3 = (1,-1)$, and $B_4 = B_0$, $B_5 = B_1$, etc., and also $B_{-1} = B_3$, $B_{-2} = B_2$, etc. (Calculate the S_i and the Bézier control points precisely and sketch the Bézier curves freehand.)

(b) Is the curve you get an exact circle? How do you know?

Problem DD-11. (a) Suppose two cubic polynomials $p(t)$ and $q(t)$ have equal values at $t = t_0$, equal first derivatives, equal second derivatives, *and* equal third derivatives. Then they must be the same polynomial. Why? (Quote Taylor's Theorem.) (b) State a similar fact about cubic parametric curves, and say why it follows from (a). (c) In trying to glue cubic Bézier curves to make a single B-spline curve, we matched first and second derivatives, but not the third. Why not?

Problem DD-12. (a) Find a matrix M so that the Bézier curve with control points $P_0, \ldots, P_3$ can be written as $P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$.

(Method: Each column of M tells the expansion of a Bernstein polynomial in powers of t.)

(b) Suppose you have B-spline control points $B_0, \ldots, B_n$. Find a matrix H so that the Bézier control points of Bézier curve $\#i$ can be written as

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = H \begin{bmatrix} B_{i-2} \\ B_{i-1} \\ B_i \\ B_{i+1} \end{bmatrix}.$$

(See the explanation of the computer method in Section 4.)

(c) Explain how to use (a) and (b) to get a matrix expression for the point on Bézier curve $\#i$ corresponding to a given t (with $0 \le t \le 1$), in terms of the B_i.

Problem DD-13. Sketch the clamped cubic spline curve $P(t)$ through $(0,0)$, $(1,1)$, $(2,2)$ with $P'(0) = (1,0)$ and $P'(2) = (1,0)$.

(Method: Calculate B-spline and Bézier control points and then sketch the Bézier curves freehand.) Useful information:

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}^{-1} = \frac{1}{12} \begin{bmatrix} 7 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 7 \end{bmatrix}$$

Problem DD-14. Sketch one cartoon frame with a simple character and some coordinate axes. Give a list of numbers that describes the cartoon. Say what each number represents, for example, the x and y coordinates of a specific point. Give enough information that if the numbers were changed, someone else could draw the new position of the character.

Problem DD-15. The leg and foot of a cartoon character are shown in keyframes at times $t = 0, 1, 2$, in Figure 17. Sketch the interpolated frame for $t = .5$, using relaxed cubic spline interpolation. (Calculate the position of the heel precisely; for this you may use the result of Problem DD-4 for the case $n = 2$ at time $\frac{1}{2}$. You may use intuition for the other two points of the character.)

Problem DD-16. Suppose that you were to compute a large number of frames based on the keyframes in Figure 17, for $0 \le t \le 2$. In what way

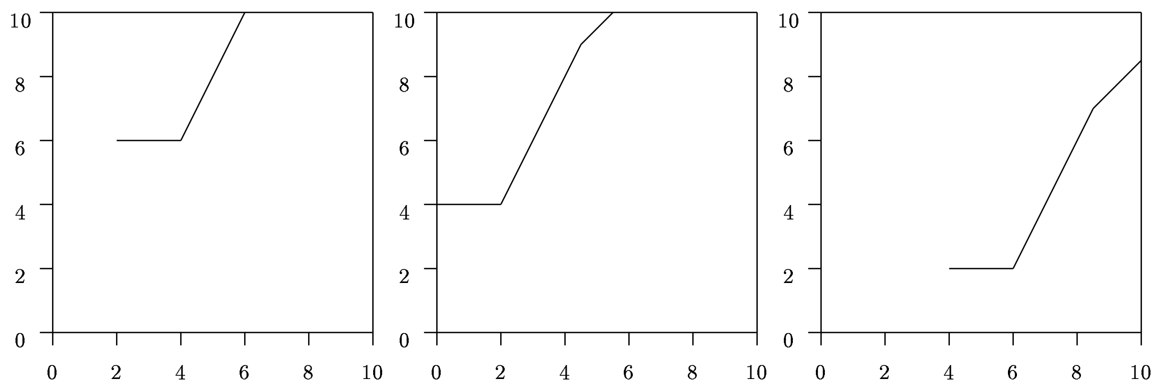

Figure 17: Cartoon frames at times 0, 1, 2

would the motion look better than if you had simply interpolated linearly between the first two key frames and then interpolated linearly between the second and third?

Problem DD-17. Sketch the nonparametric uniform relaxed cubic spline function with data values $0, 1, 0, -1, 0$. (Method: This is the case $n = 4$. Use the method of Section 8 and the basis functions from Figure 16.)

Problem DD-18. For two Bézier curves $P(t)$ and $Q(t)$, with respective lists of control points $P_0, \ldots, P_3$ and $Q_0, \ldots, Q_3$, linearly interpolating between the curves is the same as linearly interpolating between the control points. For example, if R_i is the point one-third of the way from P_i to Q_i for each i, then the Bézier curve $R(t)$ with control points $R_0, \ldots, R_3$ is one-third of the way from $P(t)$ to $Q(t)$, in the sense that for each t, the point $R(t)$ is one-third of the way from the point $P(t)$ to the point $Q(t)$.

(a) Prove the statement in this last sentence. (Let $R_0 = \frac{2}{3}P_0 + \frac{1}{3}Q_0$, etc.)

(b) Sketch the Bézier curve that is one-third of the way from the Bézier curve with control points $(0, 0)$, $(6, 0)$, $(6, 6)$, $(0, 6)$ to the Bézier curve with control points $(0, 0)$, $(6, 0)$, $(6, 6)$, $(12, 6)$.

Problem DD-19. It would be possible to use Lagrange interpolation for animation, but would the results be very good? To see what might happen, imagine a cartoon character walking along between times $t = 0$ and $t = 10$. She is walking at constant speed to the right, and is walking on one level until time $t = 7$, when she goes up one step and then walks at the new level for the rest of the time. What motion would Lagrange interpolation give? (Consult Figure 1.)

Problem DD-20. For the B-spline curve $B(t)$ with control points $B_0, \ldots, B_n$, show that for $i = 1, \ldots, n-1$,

(a) $B'(i) =$ half the vector from B_{i-1} to B_{i+1},

(b) $B''(i) =$ twice the vector from B_1 to the midpoint of the segment from B_{i-1} to B_{i+1}.

Remark. These facts make it easy to look at the control polygon for a B-spline and see where the first and second derivatives will be larger and where they will be smaller. Although these facts apply only at the points $B(i) = S_i$, in the case of the second derivative it is easy to tell what happens between: Since that $B(t)$ itself is piecewise cubic, $B'(t)$ is piecewise quadratic, and $B''(t)$ is piecewise linear, so the second derivative changes linearly from each point S_i to the next, the points where you do know the second derivative—and this includes the ends, since the second derivative there is $\mathbf{0}$.