Chapter 12

Interval Bézier Curves

In recent years there has been considerable interest in approximating the curves and surfaces that arise in computer–aided design applications by other curves and surfaces that are of lower degree, of simpler functional form, or require less data for their specification (see, for example, [Hos87], [Hos88], [Lac88], [LM87], [Pat89], [SK91], [SWZ89], [WW88]). The motivation for this activity arises from the practical need to communicate product data between diverse CAD/CAM systems that impose fundamentally incompatible constraints on their canonical representation schemes, e.g., restricting themselves to polynomial (rather than rational) forms, or limiting the polynomial degrees that they accommodate.

While most of the approximation schemes in the references cited above attempt at minimum to guarantee that an approximation will satisfy a prescribed tolerance, once this has been achieved none of them proposes to carry detailed information on the approximation error forward to subsequent applications in other systems. Such information can be of crucial importance, for example, in tolerance analyses of the components of a mechanism. This chapter describes a form — the interval Bézier curve — that is capable of carrying such information, and we show how a complete description of approximation errors may be readily embodied in such forms in a straightforward and natural manner.

12.1 Interval arithmetic and interval polynomials

By a scalar interval $[\,a, b\,]$ we mean a closed set of real values of the form

$$[\,a, b\,] = \{\, t \mid a \le t \le b \,\}\,. \tag{12.1}$$

Given two such intervals $[\,a, b\,]$ and $[\,c, d\,]$, the result of a binary arithmetic operation $\star \in \{\,+\,,\,-\,,\,\cdot\,,\,/\,\}$ on them is defined to be the set of all values obtained by applying $\star$ to operands drawn from each interval:

$$[\,a, b\,] \star [\,c, d\,] = \{\, x \star y \mid x \in [\,a, b\,] \text{ and } y \in [\,c, d\,] \,\}\,. \tag{12.2}$$

Specifically, it is not difficult to verify that

$$\begin{aligned}
[\,a, b\,] + [\,c, d\,] &= [\,a + c, b + d\,]\,, \\
[\,a, b\,] - [\,c, d\,] &= [\,a - d, b - c\,]\,, \\
[\,a, b\,] \cdot [\,c, d\,] &= [\,\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)\,]\,, \\
[\,a, b\,] \,/\, [\,c, d\,] &= [\,a, b\,] \cdot [\,1/d, 1/c\,]\,,
\end{aligned} \tag{12.3}$$

where division is usually defined only for denominator intervals that do not contain 0. Occasionally, we shall wish to treat a single real value as a *degenerate* interval,

$$a = [\, a, a \,] \tag{12.4}$$

so that we can apply the rules (12.3) to mixed operands (e.g., $a + [\, b, c \,] = [\, a + b, a + c \,]$). We shall also make use of a convenient short–hand notation for intervals, denoting them by a single symbol enclosed in square parentheses, e.g., $[\, u \,] = [\, a, b \,]$ and $[\, v \,] = [\, c, d \,]$. (However, the reader is warned against allowing this notation to impart an undue sense of familiarity — note, for example, that $[\, u \,] - [\, u \,] \neq 0$ in general!)

It can be verified from (12.3) that interval addition and multiplication are commutative and associative, but that multiplication does not, in general, distribute over addition. A notable exception is the multiplication of a sum of intervals by a scalar value, for which

$$\alpha\,([\, u \,] + [\, v \,]) = \alpha [\, u \,] + \alpha [\, v \,] \tag{12.5}$$

holds for arbitrary real values α and intervals $[\, u \,]$, $[\, v \,]$. An example of the converse case — the sum of scalar multiples of a given interval — is given in equation (12.24) below. For a more rigorous and detailed discussion of these matters, see [Moo66], [Moo79].

Interval arithmetic offers an essentially infallible (although often pessimistic) means of monitoring error propagation in numerical algorithms that employ floating point arithmetic. Many familiar algorithms can be re–formulated to accept interval operands (e.g., [Han78b], [Han78a]). The use of interval techniques in the context of geometric modeling calculations has been discussed in [MK84].

An *interval polynomial* is a polynomial whose coefficients are intervals. We shall denote such polynomials in the form $[\, p \,](t)$ to distinguish them from ordinary (single–valued) polynomials. In general we express an interval polynomial of degree n in the form

$$[\, p \,](t) = \sum_{k=0}^{n} [\, a_k, b_k \,]\, B_k^n(t)\,, \tag{12.6}$$

in terms of the Bernstein polynomial basis

$$B_k^n(t) = \binom{n}{k} (1-t)^{n-k} t^k \quad \text{for } k = 0, \ldots, n \tag{12.7}$$

on $[\, 0, 1 \,]$. Using (12.3), the usual rules of polynomial arithmetic can be carried over with minor modifications to the case of interval polynomials (see [Rok75]; polynomial arithmetic in the Bernstein basis is described in [FR88]). Since the basis functions (12.7) are evidently all non–negative for $t \in [\, 0, 1 \,]$, we can also express (12.6) in the form

$$[\, p \,](t) = [\, p_{\min}(t), p_{\max}(t) \,] \quad \text{for all } t \in [\, 0, 1 \,]\,, \tag{12.8}$$

where

$$p_{\min}(t) = \sum_{k=0}^{n} a_k\, B_k^n(t) \quad \text{and} \quad p_{\max}(t) = \sum_{k=0}^{n} b_k\, B_k^n(t)\,. \tag{12.9}$$

A useful concept in dealing with interval polynomials defined over a finite domain as approxima-

tion tools is the *width* of such a polynomial,

$$\begin{aligned} \text{width}(\,[\,p\,](t)\,) &= \int_0^1 p_{\max}(t) - p_{\min}(t)\,dt \\ &= \sum_{k=0}^{n} (b_k - a_k) \int_0^1 B_k^n(t)\,dt \\ &= \frac{1}{n+1} \sum_{k=0}^{n} b_k - a_k\,. \end{aligned} \tag{12.10}$$

The last step follows from the fact that the definite integral of each of the basis functions (12.7) over $[\,0,1\,]$ is just $1/(n+1)$ [FR88].

The width (12.10) measures the total area enclosed between the two polynomials (12.9) that define the upper and lower bounds on $[\,p\,](t)$ over $t \in [\,0,1\,]$. The Bernstein–Bézier formulation (12.6) has the convenient feature that the width of an interval polynomial $[\,p\,](t)$ is simply the *average* of the widths $b_k - a_k$ of its coefficients.

A desirable characteristic of any scheme that calls for approximation by interval polynomials is to provide a means whereby successive approximations of uniformly decreasing width are generated. The approximated function can thus be confined within as small an area as desired.

12.2 Interval Bézier curves

We will define a vector–valued interval $[\,\mathbf{p}\,]$ in the most general terms as any compact set of points (x, y) in two dimensions. The addition of such sets is given by the *Minkowski sum*,

$$[\,\mathbf{p}_1\,] + [\,\mathbf{p}_2\,] = \{\,(x_1 + x_2, y_1 + y_2) \mid (x_1, y_1) \in [\,\mathbf{p}_1\,] \text{ and } (x_2, y_2) \in [\,\mathbf{p}_2\,]\,\}\,. \tag{12.11}$$

Such point–set operations arise, for example, in image analysis [Ser82]. However, since they are in general quite difficult to perform algorithmically (even when $[\,\mathbf{p}_1\,]$ and $[\,\mathbf{p}_2\,]$ are restricted to polygonal boundaries), it will be prudent for the present to restrict our attention to vector–valued intervals that are just the *tensor products* of scalar intervals,

$$[\,\mathbf{p}\,] = [\,a,b\,] \times [\,c,d\,] = \{\,(x,y) \mid x \in [\,a,b\,] \text{ and } y \in [\,c,d\,]\,\}\,. \tag{12.12}$$

We occasionally use the notation $([\,a,b\,],[\,c,d\,])$ instead of $[\,a,b\,] \times [\,c,d\,]$ for $[\,\mathbf{p}\,]$. Such vector–valued intervals are clearly just rectangular regions in the plane, and their addition is a trivial matter:

$$[\,\mathbf{p}_1\,] + [\,\mathbf{p}_2\,] = [\,a_1 + a_2, b_1 + b_2\,] \times [\,c_1 + c_2, d_1 + d_2\,]\,, \tag{12.13}$$

where $[\,\mathbf{p}_1\,] = [\,a_1, b_1\,] \times [\,c_1, d_1\,]$ and $[\,\mathbf{p}_2\,] = [\,a_2, b_2\,] \times [\,c_2, d_2\,]$. The extension of these ideas to vector–valued intervals in spaces of higher dimension is straightforward.

We note that there are other potentially useful shapes for interval control points (e.g., circular disks); the discussion that follows adapts readily to interval control points of more general shape, although specific algorithms may be more difficult to implement than in the tensor–product case.

Let us recall now that a Bézier curve on the parameter interval $[\,0,1\,]$ is defined by

$$\mathbf{P}(t) = \sum_{k=0}^{n} \mathbf{P}_k\, B_k^n(t)\,, \tag{12.14}$$

where the Bernstein basis function $B_k^n(t)$ are as defined above in (12.7). The vector coefficients $\mathbf{P}_k = (x_k, y_k)$ in (12.14) are called the *control points* of the curve.

An interval Bézier curve (i.e., a Bézier curve with *vector–interval control points*) is written in the form

$$[\,\mathbf{P}\,](t) = \sum_{k=0}^{n} [\,\mathbf{P}_k\,]\, B_k^n(t)\,, \tag{12.15}$$

where we assume that the $[\,\mathbf{P}_k\,]$ are rectangular (possibly degenerate) intervals of the form (12.12). Figure 12.1 illustrates a sample cubic interval Bézier curve.

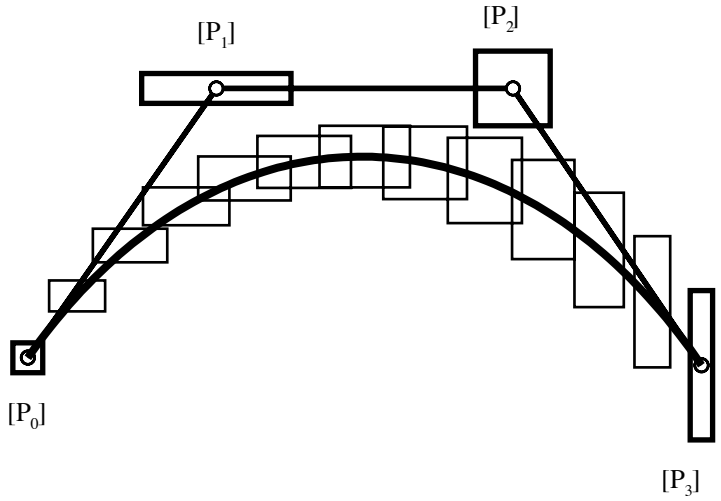

Figure 12.1: A cubic interval Bézier curve.

For each $t \in [\,0,1\,]$, the value $[\,\mathbf{P}\,](t)$ of the interval Bézier curve (12.15) is a vector interval that has the following significance: For any Bézier curve $\mathbf{P}(t)$ whose control points satisfy $\mathbf{P}_k \in [\,\mathbf{P}_k\,]$ for $k = 0, \ldots, n$, we have $\mathbf{P}(t) \in [\,\mathbf{P}\,](t)$. Likewise, the entire interval Bézier curve (12.15) defines a region in the plane that contains all Bézier curves whose control points satisfy $\mathbf{P}_k \in [\,\mathbf{P}_k\,]$ for $k = 0, \ldots, n$.

We shall not concern ourselves here with defining curves of the form (12.15) *ab initio*, but rather with the use of that form in approximating more complicated curves and surfaces by polynomial interpolation in such a manner that *the interval control points* $[\,\mathbf{P}_k\,]$ *reflect the approximation error*. Thus, the plane region defined by an interval approximant $[\,\mathbf{P}\,](t)$ to some other curve $\mathbf{r}(t)$ will be guaranteed to contain the latter.

While an initial approximant of this form may be unacceptably crude — defining an area that is too large for practical purposes — a process of refinement by subdivision will usually remedy this problem. A precise representation of the error incurred by approximations can be crucial in applications such as tolerance analysis, where a nominal approximant is of little value without information on its deviation from the exact form. The Bernstein–Bézier form offers an intuitive framework for expressing approximants, along with their error terms, as interval–valued polynomials.

12.2.1 Affine maps

A key operation in the de Casteljau subdivision and degree elevation algorithms for Bézier curves (see [Far90]) is computing the affine map

$$\mathbf{M}(p_0, p_1, t) = (1-t)\, p_0 + t\, p_1 \tag{12.16}$$

of two points p_0 and p_1 (we consider for now the case where p_0 and p_1 are simply scalar values). This map can be visualized as shown in Figure 12.2, where the values of p_0 and p_1 are taken as the vertical coordinates, and the values of t as horizontal coordinates. At $t = 0$, $y = p_0$ while at $t = 1$, $y = p_1$. The affine map is then represented graphically by drawing a straight line through p_0 and p_1. This line can be regarded as the affine map function for the two points: at any value of t, the affine map is simply the vertical coordinate of the line.

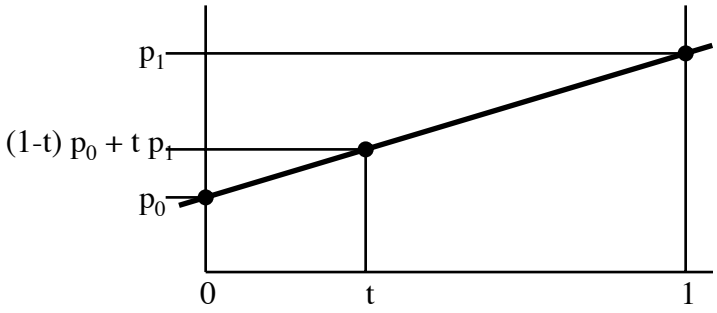

Figure 12.2: The affine map of two scalar points.

Consider next the affine map of two scalar intervals:

$$\begin{aligned} [\,\mathbf{M}\,]([\,a,b\,],[\,c,d\,],t\,) &= (1-t)\,[\,a,b\,] + t\,[\,c,d\,] \\ &= \{\,(1-t)\,u + t\,v \mid u \in [\,a,b\,] \text{ and } v \in [\,c,d\,]\,\}\,. \end{aligned} \tag{12.17}$$

The affine map (12.17) can be visualized as shown in Figure 12.3 — for a given value of t,

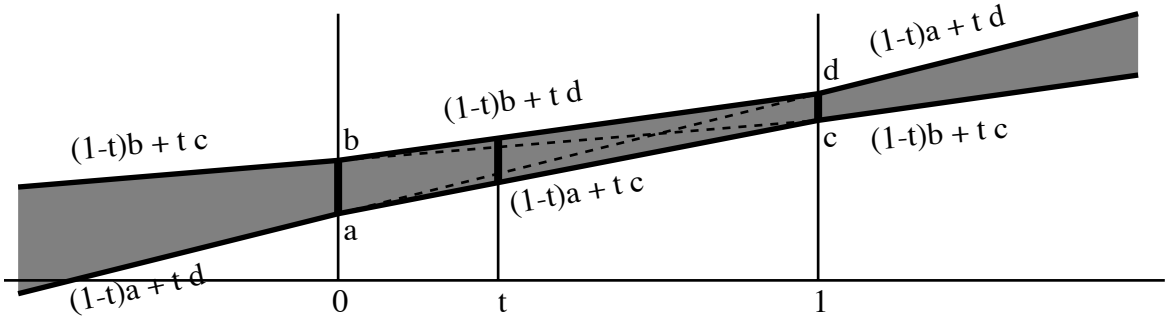

Figure 12.3: The affine map of two scalar intervals.

$[\,\mathbf{M}\,]([\,a,b\,],[\,c,d\,],t\,)$ is generated by drawing a vertical line at that t value and identifying all points on that line which are collinear with any point in $[\,a,b\,]$ at $t=0$ and any point in $[\,c,d\,]$ at $t=1$. Qualitatively, we see that the width of the affine map (12.17) lies between the widths of $[\,a,b\,]$ and $[\,c,d\,]$ when $t\in[\,0,1\,]$, whereas the width of the affine map grows linearly — without bound — for t outside the unit interval.

The affine map of two vector-valued intervals $[\mathbf{P}_0]$ and $[\mathbf{P}_1]$ as defined in equation 12.12 is simply

$$[\,\mathbf{M}\,]([\,\mathbf{P}_0],[\,\mathbf{P}_1],t\,)=\{\,(1-t)\,\mathbf{u}+t\,\mathbf{v}\mid\mathbf{u}\in[\,\mathbf{P}_0\,]\text{ and }\mathbf{v}\in[\,\mathbf{P}_1\,]\,\}\tag{12.18}$$

as shown in figure 12.4. Observe that the midpoint, width, and height of the affine map rectangle

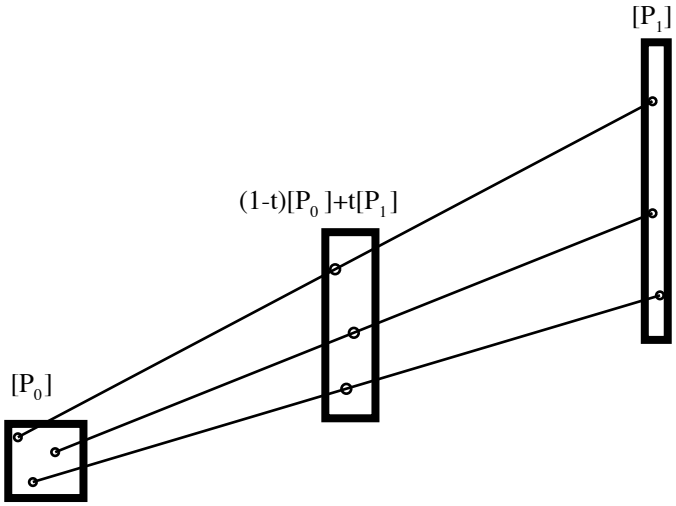

Figure 12.4: The affine map of two vector intervals.

are simply affine maps of the midpoints, widths, and heights of $[\mathbf{P}_0]$ and $[\mathbf{P}_1]$.

For a Bézier curve of the form (12.14), we may regard the de Casteljau algorithm as a repeated application of (12.16) to the control points $\{\mathbf{P}_k\}$. With $\mathbf{P}_k^{(0)}=\mathbf{P}_k$ for $k=0,\ldots,n$, we set

$$\mathbf{P}_k^{(r+1)}\;=\;\mathbf{M}\,(\,\mathbf{P}_{k-1}^{(r)}\,,\,\mathbf{P}_k^{(r)}\,,\,t\,)\quad\text{for }k=r+1,\ldots,n\tag{12.19}$$

and $r=0,\ldots,n-1$. This generates a triangular array of the quantities $\{\mathbf{P}_k^{(r)}\}$ — the final entry $\mathbf{P}_n^{(n)}$ in this array corresponds to the point $\mathbf{P}(t)$ on the curve (12.14), while the entries

$$\mathbf{P}_0^{(0)},\mathbf{P}_1^{(1)},\ldots,\mathbf{P}_n^{(n)}\quad\text{and}\quad\mathbf{P}_n^{(n)},\mathbf{P}_n^{(n-1)},\ldots,\mathbf{P}_n^{(0)}\tag{12.20}$$

on the left– and right–hand sides of the array are the control points for the subsegments of $\mathbf{P}(t)$ over the parameter intervals $[\,0,t\,]$ and $[\,t,1\,]$, respectively. To apply (12.19) to *interval* Bézier curves, we simply replace the quantities $\mathbf{P}_k^{(r)}$ by their interval counterparts, and invoke the appropriate rules of interval arithmetic, as illustrated in Figure 12.5. We discuss the de Casteljau algorithm further in the next section.

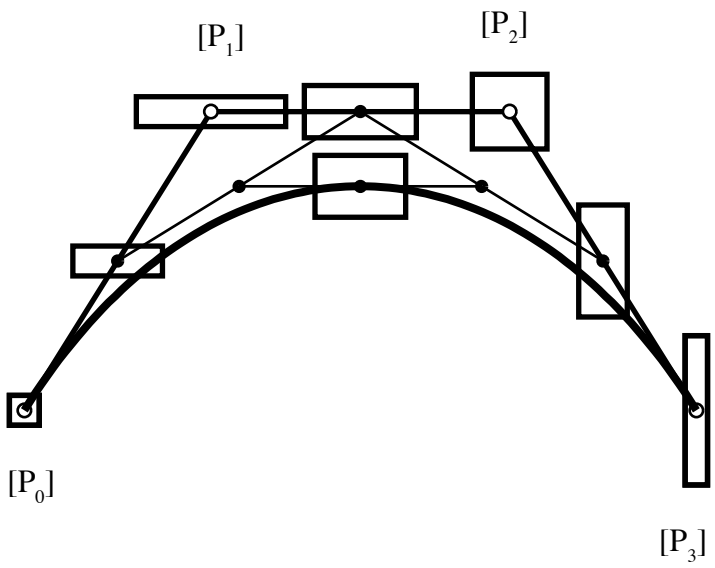

Figure 12.5: Interval de Casteljau algorithm.

12.2.2 Centered form

All the familiar algorithms and characteristics that we associate with Bézier curves — the subdivision and degree elevation algorithms, the variation–diminishing property and convex-hull confinement (see [Far90] for a review) — carry over to Bézier curves with interval coefficients under relatively minor modifications or re–interpretations.

For the case of tensor–product interval control points of the form (12.12), these operations can be more conveniently formulated if the control points are expressed in *centered form*. We re–write each control point $[\,\mathbf{P}_k\,]$ in the following manner:

$$\begin{aligned}
[\,\mathbf{P}_k\,] &= ([\,a_k, b_k\,], [\,c_k, d_k\,]) \\
&= \frac{1}{2}(a_k + b_k, c_k + d_k) \;+\; \frac{1}{2}(b_k - a_k, d_k - c_k)\,[\,-1, +1\,] \\
&= \overline{[\,\mathbf{P}_k\,]} \;+\; \mathbf{e}_k[\,i\,] \qquad (12.21)
\end{aligned}$$

where $[\,i\,]$ denotes the interval $[\,-1, +1\,]$. In (12.21), the point $\overline{[\,\mathbf{P}_k\,]}$ is the *center* of the vector interval $[\,\mathbf{P}_k\,]$, while the vector $\mathbf{e}_k$ is the *error* of $[\,\mathbf{P}_k\,]$. Note that the x and y components of $\mathbf{e}_k$ are necessarily non–negative.

Using centered form, the affine map of two interval points may be written as

$$\begin{aligned}
(1-t)\,[\,\mathbf{P}_0\,] + t\,[\,\mathbf{P}_1\,] &= (1-t)\,\{\overline{[\,\mathbf{P}_0\,]} + \mathbf{e}_0[\,i\,]\} \;+\; t\,\{\overline{[\,\mathbf{P}_1\,]} + \mathbf{e}_1[\,i\,]\} \\
&= \{(1-t)\,\overline{[\,\mathbf{P}_0\,]} + t\,\overline{[\,\mathbf{P}_1\,]}\} + \{(1-t)\,\mathbf{e}_0 + t\,\mathbf{e}_1\}[\,i\,]\,, \qquad (12.22)
\end{aligned}$$

where we assume that $0 \le t \le 1$. Thus, the affine map of two interval points can be computed by independently taking the affine maps of their centers and their errors.

For $0 \le t \le 1$, an interval Bézier curve may be expressed in centered form as follows:

$$\begin{aligned}
[\,\mathbf{P}\,](t) &= \sum_{k=0}^{n} [\,\mathbf{P}_k\,]\, B_k^n(t) \\
&= \sum_{k=0}^{n} \{\overline{[\,\mathbf{P}_k\,]} + \mathbf{e}_k[\,i\,]\}\, B_k^n(t) \\
&= \sum_{k=0}^{n} \overline{[\,\mathbf{P}_k\,]}\, B_k^n(t) \;+\; [\,i\,] \sum_{k=0}^{n} \mathbf{e}_k\, B_k^n(t) \\
&= \overline{[\,\mathbf{P}\,]}(t) + [\,i\,]\,\mathbf{e}(t) \qquad (12.23)
\end{aligned}$$

In bringing the interval $[\,i\,]$ outside the summation sign above we rely on the fact that, for each $k = 0, \ldots, n$, the x and y components of $\mathbf{e}_k$ are non–negative and $B_k^n(t) \ge 0$ for $t \in [\,0, 1\,]$.

Thus, the interval Bézier curve $[\,\mathbf{P}\,](t)$ over $0 \le t \le 1$ can be split into two independent "components" — a *center curve* $\overline{[\,\mathbf{P}\,]}(t)$, and an *error curve* $\mathbf{e}(t)$. Note that the control points $\mathbf{e}_k$ of $\mathbf{e}(t)$ all lie within the first quadrant, and consequently the error curve is itself contained within that quadrant for $t \in [\,0, 1\,]$.

Equation (12.22) suggests that for t outside the unit interval, the error curve should grow monotonically. Writing equation (12.17) in centered form with $[\,a, b\,] = p_0 + e_0[\,i\,]$ and $[\,c, d\,] = p_1 + e_1[\,i\,]$, and noting that

$$\alpha[\,i\,] + \beta[\,i\,] = (|\alpha| + |\beta|)\,[\,i\,] \qquad (12.24)$$

for arbitrary real numbers α and β, we see that the expressions

$$[\,\mathbf{M}\,](p_0 + e_0[\,i\,], p_1 + e_1[\,i\,], t)$$

$$\begin{aligned}
&= \begin{cases} p_0(1-t) + p_1 t \;+\; \{e_0(1-t) - e_1 t\}\,[\,i\,] & \text{for } t < 0\,, \\ p_0(1-t) + p_1 t \;+\; \{e_0(1-t) + e_1 t\}\,[\,i\,] & \text{for } 0 \le t \le 1\,, \\ p_0(1-t) + p_1 t \;+\; \{e_0(t-1) + e_1 t\}\,[\,i\,] & \text{for } t > 1\,, \end{cases} \qquad (12.25) \\
&= p_0(1-t) + p_1 t + \{e_0|1-t| + e_1|t|\}[i] \qquad (12.26)
\end{aligned}$$

describe the behavior of the affine map of two scalar intervals for all real t (see Figure 12.4).

The de Casteljau algorithm (12.19) is essentially a repeated application of the affine map (12.25). Figure 12.5 illustrates for the case $t = \frac{1}{2}$. By studying the behavior of the de Casteljau algorithm applied to an interval Bézier curves outside the unit interval, equation (12.25) leads to the following expressions:

$$[\,\mathbf{P}\,](t) = \begin{cases} \displaystyle\sum_{k=0}^{n} \overline{[\,\mathbf{P}_k\,]}\, B_k^n(t) + \mathbf{e}(t)[\,i\,] & \text{for } 0 \le t \le 1\,, \\ \displaystyle\sum_{k=0}^{n} \overline{[\,\mathbf{P}_k\,]}\, B_k^n(t) + \tilde{\mathbf{e}}(t)[\,i\,] & \text{for } t < 0 \text{ or } t > 1\,, \end{cases} \qquad (12.27)$$

where $\mathbf{e}(t)$ is the error curve for $t \in [\,0, 1\,]$ defined in (12.23), and

$$\tilde{\mathbf{e}}(t) = \sum_{k=0}^{n} (-1)^k \mathbf{e}_k\, B_k^n(t)\,. \qquad (12.28)$$

defines an error curve that is appropriate to the domains $t < 0$ and $t > 1$. (Alternately, we may substitute $|B_k^n(t)| = (-1)^k B_k^n(t)$ when $t < 0$, and $|B_k^n(t)| = (-1)^{n+k} B_k^n(t)$ when $t > 1$, into

$$[\,\mathbf{P}\,](t) \;=\; \sum_{k=0}^{n} \{\overline{[\,\mathbf{P}_k\,]} + \mathbf{e}_k[\,i\,]\}\, B_k^n(t) \tag{12.29}$$

$$= \sum_{k=0}^{n} \overline{[\,\mathbf{P}_k\,]}\, B_k^n(t) \;+\; [\,i\,] \sum_{k=0}^{n} \mathbf{e}_k\, |B_k^n(t)|\,,$$

and set $(-1)^n[\,i\,] = [\,i\,]$ in the case $t > 1$, in order to obtain (12.27)).

12.2.3 Error monotonicity

Both the x and y components of the error curve $\tilde{\mathbf{e}}(t)$ grow monotonically as t departs from the unit interval $[\,0, 1\,]$. To appreciate this, we note that the r–th derivative of the Bézier curve (12.14) can be written as a Bézier curve of degree $n - r$,

$$\mathbf{P}^{(r)}(t) = \frac{n!}{(n-r)!} \sum_{k=0}^{n-r} \Delta^r \mathbf{P}_k\, B_k^{n-r}(t)\,, \tag{12.30}$$

where the quantities $\{\Delta^r \mathbf{P}_k\}$ are the r–th *forward differences* of the control points, given by

$$\Delta^r \mathbf{P}_k = \sum_{j=0}^{r} (-1)^j \binom{r}{j} \mathbf{P}_{k+r-j} \quad \text{for } k = 0, \ldots, n-r\,. \tag{12.31}$$

The forward differences can be generated by the recursion

$$\Delta^{r+1} \mathbf{P}_k = \Delta^r \mathbf{P}_{k+1} - \Delta^r \mathbf{P}_k \tag{12.32}$$

which commences by setting $\Delta \mathbf{P}_k \,(= \Delta^1 \mathbf{P}_k) = \mathbf{P}_{k+1} - \mathbf{P}_k$.

Since the x and y components of the control points of $\tilde{\mathbf{e}}(t)$ are of alternating sign, it is clear from (12.31) that the x and y components of the differences $\Delta^r \tilde{\mathbf{e}}_k$ are all of the *same* sign for $r = 1, \ldots, n$. Thus, the x and y components of all derivatives of $\tilde{\mathbf{e}}(t)$ have the same sign when t lies outside the unit interval, and the magnitude of $\tilde{\mathbf{e}}(t)$ grows monotonically for $t < 0$ or $t > 1$.

Interval arithmetic has a bad reputation for interval width inflation. That is, it often occurs when dealing with interval arithmetic that the widths of the intervals "balloon" so rapidly that the practical value of the interval technique is seriously impaired. For example, interval operations are generally not reversible:

$$([1, 2] + [3, 4]) - [3, 4] = [0, 3]. \tag{12.33}$$

Thus, it is noteworthy that ballooning does not occur with the de Casteljau algorithm when applied within the unit parameter interval. Even after repeated applications of the de Casteljau algorithm to an interval Bézier curve, a point $[\,\mathbf{P}\,](t)$ evaluated on a subdivided region of the curve has the same size as the given point evaluated on the original curve, as long as the subdivision always occurs within the $[\,0, 1\,]$ parameter domain. However, equation (12.27) shows that serious interval inflation will occur when subdividing outside the unit interval.

12.2.4 Envelopes of interval Bézier curves

At first sight, it may seem that making the transition from *interval polynomials* of the form (12.6) to *interval curves* of the form (12.15) is a straightforward matter. However, a number of subtle difficulties arise that deserve attention. It should be noted, for example, that while the bounds on the scalar–valued interval polynomial (12.6) are readily expressible as the two polynomials (12.9), the boundary of the area defined by an interval Bézier curve is *not* (in general) describable by just two polynomial Bézier curves. That boundary is actually the *envelope* of the continuum of rectangles

$$[\,x\,](t)\times[\,y\,](t) \quad \text{for } t \in [\,0,1\,]\,, \tag{12.34}$$

where $[\,x\,](t)$ and $[\,y\,](t)$ are the interval–polynomial components of (12.15).

The region of the plane covered by an interval Bézier curve is actually bounded by Bézier curves and by straight line segments. Consider, for example, the interval Bézier curve in Figure 12.6. There

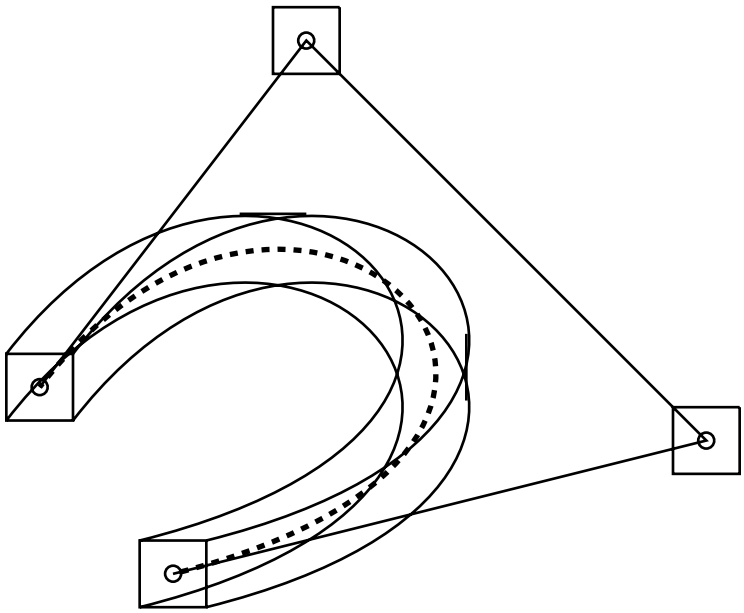

Figure 12.6: The envelope of an interval Bézier curve.

are four possible Bézier curves involved in the envelope for a curve over the unit parameter interval. Those four curves are defined by the four respective rectangle corners of the interval control points.

If the center curve $[\overline{\mathbf{P}}\,](t)$ is progressing NE or SW, the upper envelope bound is defined by control points lying in the top left corner of their respective intervals, and the lower envelope bound is defined by control points lying in the bottom right corner of their respective intervals. Likewise, a curve progressing NW or SE is bounded from above by a curve whose control points lie in top right corners, and from beneath by a curve whose control points lie in bottom left corners. In the case of a horizontal or vertical tangency, straight line segments are introduced into the envelope, equal in width to twice the interval error at the point of tangency.

Thus, we can represent the boundary of the region covered by an interval Bézier curve as a piecewise–Bézier curve of the same degree. Any linear segments that arise in this boundary due to points of horizontal or vertical tangency can be degree–elevated if desired, making the representation of the envelope compatible with standard spline formats.

12.2.5 Interval hodographs

The hodograph of a Bézier curve is simply the first derivative of that curve. The hodograph is itself expressible as a Bézier curve of degree one less than the given curve. If $\{\mathbf{P}_k\}$ are the control points of a Bézier curve of degree n, the control points of the hodograph are $n\Delta\mathbf{P}_k = n(\mathbf{P}_{k+1} - \mathbf{P}_k)$ for $k = 0, \ldots, n-1$ [B86].

We can readily extend this notion to interval Bézier curves, although some caution must be exercised in how the hodograph of an interval Bézier curve is to be interpreted. If the given interval Bézier curve (with control points $[\,\mathbf{P}_k\,]$) is taken to define the *set of all Bézier curves with fixed control points satisfying* $\mathbf{P}_k \in [\,\mathbf{P}_k\,]$, then the hodograph control points can correctly be computed by simply differencing the control points of the original interval Bézier curve. Thus, a point $[\,\mathbf{P}'\,](t)$ on the hodograph defines bounds on the tangent vector for any curve for which $\mathbf{P}_k \in [\,\mathbf{P}_k\,]$.

However, in some applications a given curve is represented by allowing control points to move about as a function of t, within the interval. For example, in [SK91] it was shown that a rational curve can be represented as a polynomial curve for which one of the control points is a rational curve. That "moving control point" can be replaced by an interval which bounds its movement, but the afore–mentioned hodograph does not correctly bound the derivative of the rational curve. The reason is that the moving control point itself moves as a function of t, and therefore the product rule must be applied to correctly represent the derivative of the curve.

12.3 Approximation by interval polynomials

Before proceeding to vector (curve) approximations, it will be convenient to discuss the approximation of scalar functions by interval polynomials expressed in Bernstein form.

Let the function $f(t)$ be differentiable $n+1$ times on the interval $[\,a, b\,]$, and let

$$a \le t_0 < t_1 < \cdots < t_n \le b \tag{12.35}$$

be an ordered sequence of $n+1$ distinct points on that interval. The *Lagrange interpolant* to the sampled values $f_k = f(t_k)$, $k = 0, 1, \ldots, n$ of $f(t)$ at these points is the unique polynomial of degree n given by

$$F_n(t) = \sum_{k=0}^{n} f_k\, L_k(t)\,, \tag{12.36}$$

where the $n+1$ linearly independent polynomials

$$L_k(t) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{t - t_j}{t_k - t_j} \quad \text{for } k = 0, \ldots, n \tag{12.37}$$

constitute the *Lagrange basis* for the sequence of nodes $t_0, t_1, \ldots, t_n$. Since the basis polynomials (12.37) satisfy the conditions

$$L_k(t_j) = \delta_{jk} = \begin{cases} 1 & if \text{j=k}, \\ 0 & otherwise, \end{cases} \tag{12.38}$$

for each $j = 0, \ldots, n$ and $k = 0, \ldots, n$ it is clear that the polynomial $F_n(t)$ reproduces the values of the function $f(t)$ at each of the nodes: $F_n(t_k) = f_k$ for $k = 0, \ldots, n$.

12.3.1 Remainder formulae and interval approximants

At any other point on $[\,a, b\,]$, however, the values of $F_n(t)$ and $f(t)$ disagree in general. Thus, we define the error $E_n(t)$ of the Lagrange interpolant by

$$E_n(t) = f(t) - F_n(t)\,, \tag{12.39}$$

and if we have information on the behavior of the derivative $f^{(n+1)}(t)$ over the interval $[\,a, b\,]$, we may express (12.39) by the *Cauchy remainder* formula [Dav63]:

$$E_n(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (t - t_k) \quad \text{for some } \xi \in (a, b)\,. \tag{12.40}$$

Although, for each t, the value ξ at which the right hand side of (12.40) gives the error $E_n(t)$ of the Lagrange interpolant is not easily determined, if we know *upper and lower bounds* on $f^{(n+1)}(t)$ over $[\,a, b\,]$,

$$f_{\min}^{(n+1)} \le f^{(n+1)}(t) \le f_{\max}^{(n+1)} \quad \text{for all } t \in (a, b) \tag{12.41}$$

then we may write

$$f(t) \in F_n(t) \,+\, \frac{[\,f_{\min}^{(n+1)}, f_{\max}^{(n+1)}\,]}{(n+1)!} \prod_{k=0}^{n} (t - t_k)\,, \tag{12.42}$$

where the right hand side is regarded as *an interval polynomial* $[\,F_{n+1}\,](t)$ *of degree* $n + 1$.

In the particular form (12.42), only the remainder term has a non–degenerate interval coefficient, but if we choose to represent this interval polynomial in another basis — for example, as

$$[\,F_{n+1}\,](t) = \sum_{k=0}^{n+1} [\,a_k, b_k\,] \binom{n+1}{k} \frac{(b-t)^{n+1-k}(t-a)^k}{(b-a)^{n+1}} \tag{12.43}$$

in the Bernstein basis of degree $n + 1$ on $[\,a, b\,]$ — we would find that in general *each* coefficient $[\,a_k, b_k\,]$ is a proper interval of finite width. The formulae giving the interval coefficients of (12.43) in terms of the nodes (12.35), the function values $f_0, f_1, \ldots, f_n$ at those nodes, and the derivative bounds (12.41) for $f(t)$ on $[\,a, b\,]$ are cumbersome to quote in full generality; we shall give explicit formulae below only for the simpler cases of practical interest.

12.3.2 Hermite interpolation

Hermite interpolation is another useful means of polynomial approximation. In general this involves the interpolation of the values and derivatives of $f(t)$ at specified points. Informally, we may regard Hermite interpolation as a limiting form of Lagrange interpolation in which a sequence $m + 1$ consecutive nodes

$$t_k, t_{k+1}, \ldots, t_{k+m+1} \tag{12.44}$$

are allowed to merge; then $F_n(t)$ would be required to match the value $f(t_k)$ and first m derivatives

$$f'(t_k), f''(t_k), \ldots, f^{(m)}(t_k) \tag{12.45}$$

of $f(t)$ at t_k (i.e., $F_n(t)$ has an m–fold osculation to $f(t)$ at t_k).

In particular, suppose that $t_0, t_1, \ldots, t_r$ is a monotone sequence of $r+1$ distinct nodes, and we associate non–negative integers $m_0, m_1, \ldots, m_r$ with these nodes, such that $m_0+m_1+\cdots+m_r+r = n$. Then the unique polynomial $F_n(t)$ that satisfies the interpolation problem

$$F_n^{(i)}(t_k) = f^{(i)}(t_k) \quad \text{for} \;\; i = 0, \ldots, m_r \;\; \text{and} \;\; k = 0, \ldots, r\,, \tag{12.46}$$

where $f^{(0)}(t) \equiv f(t)$ and $F_n^{(0)}(t) \equiv F_n(t)$, has the remainder term

$$E_n(t) = f(t) - F_n(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{r} (t-t_k)^{m_k+1}\,. \tag{12.47}$$

From the above, we can write down an expression analogous to (12.42), namely

$$f(t) \in F_n(t) \;+\; \frac{[\, f_{\min}^{(n+1)}, f_{\max}^{(n+1)}\,]}{(n+1)!} \prod_{k=0}^{r} (t-t_k)^{m_k+1}\,. \tag{12.48}$$

The interval–valued error term on the right has the same coefficient as in (12.43), but its dependence on t is different.

An important instance of the general Hermite problem (12.46) is the symmetric interpolation of values and derivatives to order $(n-1)/2$ of a function $f(t)$ at the end–points of the unit interval $[\,0,1\,]$ by a polynomial $F_n(t)$ of odd degree n,

$$F_n^{(i)}(0) = f^{(i)}(0) \quad \text{and} \quad F_n^{(i)}(1) = f^{(i)}(1) \qquad \text{for} \;\; i = 0, \ldots, (n-1)/2\,. \tag{12.49}$$

If we define the Hermite basis $\{H_k^n(t)\}$ of odd degree n on $[\,0,1\,]$ by demanding that the boundary conditions

$$\left.\frac{d^{\,j} H_k^n}{dt^{\,j}}\right|_{t=0} = \left.\frac{d^{\,j} H_{n-k}^n}{dt^{\,j}}\right|_{t=1} = \delta_{jk} \tag{12.50}$$

be satisfied for $j,k = 0, \ldots, (n-1)/2$ (where δ_{jk} is the Kronecker symbol given by (12.38) above), we can write down the solution to (12.49) in the form

$$F_n(t) \;=\; \sum_{k=0}^{(n-1)/2} f^{(k)}(0) H_k^n(t) + f^{(k)}(1) H_{n-k}^n(t)\,. \tag{12.51}$$

The remainder term (12.40) in this case becomes

$$E_n(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\, t^{(n+1)/2} (t-1)^{(n+1)/2}\,. \tag{12.52}$$

For example, the cubic Hermite basis on $[\,0,1\,]$ is given by

$$\begin{aligned} H_0^3(t) = 2t^3 - 3t^2 + 1\,,\;\; H_1^3(t) = t^3 - 2t^2 + t\,, \\ H_2^3(t) = t^3 - t^2\,,\;\; H_3^3(t) = -2t^3 + 3t^2\,, \end{aligned} \tag{12.53}$$

and plays an important rôle in the construction of cubic splines [dB78].

We now consider some simple special cases. Let the function $f(t)$ be twice differentiable on the interval $[\,0,1\,]$, and let $f_0 = f(0)$ and $f_1 = f(1)$ be its end–point values on that interval. The linear interpolant to these values is just

$$F_1(t) = f_0(1-t) + f_1 t\,, \tag{12.54}$$

and we may write

$$f(t) = F_1(t) \,+\, \frac{f''(\xi)}{2}\, t(t-1) \quad \text{for some } \xi \in (0,1)\,. \tag{12.55}$$

Thus, if the second derivative $f''(t)$ is confined to the interval

$$[\, f'' \,] = [\, f''_{\min}, f''_{\max} \,]\,, \tag{12.56}$$

for all $t \in [\,0,1\,]$, we have

$$f(t) \in F_1(t) \,-\, \frac{[\, f'' \,]}{2}\, t(1-t) \quad \text{for all } t \in [\,0,1\,]\,. \tag{12.57}$$

In order to formulate the above as a quadratic interval polynomial $[\, F_2(t) \,]$, we degree–elevate (12.54) by multiplying the right hand side by $1 = (1-t) + t$ to obtain

$$f(t) \in [\, F_2 \,](t) = \sum_{k=0}^{2} [\, F_{2,k} \,]\, B_k^2(t) \tag{12.58}$$

where

$$[\, F_{2,0} \,] = f_0\,,\ [\, F_{2,1} \,] = \frac{2f_0 + 2f_1 - [\, f'' \,]}{4}\,,\ [\, F_{2,2} \,] = f_1\,. \tag{12.59}$$

Thus, in the case of a linear end–point interpolant, we see that the resulting quadratic interval polynomial bounds $f(t)$ on $[\,0,1\,]$ from below and above by the parabolas

$$\begin{aligned} F_{2,\min}(t) &= f_0 B_0^2(t) \,+\, \frac{2f_0 + 2f_1 - f''_{\max}}{4} B_1^2(t) \,+\, f_1 B_2^2(t)\,, \\ F_{2,\max}(t) &= f_0 B_0^2(t) \,+\, \frac{2f_0 + 2f_1 - f''_{\min}}{4} B_1^2(t) \,+\, f_1 B_2^2(t)\,. \end{aligned} \tag{12.60}$$

For the case of cubic Hermite interpolation to the function values f_0, f_1 and derivatives f'_0, f'_1 at the end–points of $[\,0,1\,]$, the Bernstein–Bézier form of the interpolant is

$$F_3(t) = f_0 B_0^3(t) \,+\, \frac{3f_0 + f'_0}{3} B_1^3(t) \,+\, \frac{3f_1 - f'_1}{3} B_2^3(t) \,+\, f_1 B_3^3(t)\,. \tag{12.61}$$

Thus, if the fourth derivative lies within the interval $[\, f^{(4)} \,] = [\, f^{(4)}_{\min}, f^{(4)}_{\max} \,]$ for all $t \in [\,0,1\,]$, we may write

$$f(t) \in F_3(t) + \frac{[\, f^{(4)} \,]}{24}\, t^2(t-1)^2 = \sum_{k=0}^{4} [\, F_{4,k} \,]\, B_k^4(t) = [\, F_4 \,](t)\,, \tag{12.62}$$

where the coefficients of the quartic interval polynomial on the right are given by

$$\begin{aligned} [\, F_{4,0} \,] = f_0\,,\ [\, F_{4,1} \,] &= f_0 + \frac{1}{4} f'_0\,, \\ [\, F_{4,2} \,] &= \frac{1}{2}(f_0 + f_1) + \frac{1}{6}(f'_0 - f'_1) + \frac{1}{144}[\, f^{(4)} \,]\,, \\ [\, F_{4,3} \,] = f_1 - \frac{1}{4} f'_1\,,\ [\, F_{4,4} \,] &= f_1\,. \end{aligned} \tag{12.63}$$

It should be noted that in all cases of symmetric Hermite interpolation of function values and derivatives to order $(n-1)/2$ at the end points of an interval, only the middle coefficient

$[\,F_{n+1,(n+1)/2}\,]$ of the interpolating interval polynomial $[\,F_{n+1}\,](t)$ has non–zero width, since the residual (12.52) contributes only to the term involving the basis function $B^{n+1}_{(n+1)/2}(t)$. The width of this middle coefficient is just

$$\frac{f^{(n+1)}_{\max} - f^{(n+1)}_{\min}}{\big(\,(n+1)\,n\,\cdots\,\frac{1}{2}(n+3)\,\big)^2}\,. \tag{12.64}$$

Note also that for end–point Hermite interpolation, the width of the interval polynomial $[\,F_{n+1}\,](t)$ always degenerates to zero at $t = 0$ and $t = 1$.

Example 1. With $f(t) = e^t$ we have $f(0) = f'(0) = 1$, $f(1) = f'(1) = e$, and $f^{(4)}(t) \in [\,1, e\,]$ for $t \in [\,0, 1\,]$. The interval control points (12.63) of the Hermite interpolant reflecting these values are then:

$$[\,F_{4,0}\,] = 1\,,\ [\,F_{4,1}\,] = \frac{5}{4}\,,\ [\,F_{4,2}\,] = \frac{1}{144}[\,97 + 48e, 96 + 49e\,]\,, \tag{12.65}$$

$$[\,F_{4,3}\,] = \frac{3}{4}e\,,\ [\,F_{4,4}\,] = e\,.$$

We can gain a better idea of the width of the middle control point from the approximate form $[\,F_{4,2}\,] \approx [\,1.5797, 1.5916\,]$. From (12.10) we now see that the overall width of the approximant $[\,F_4\,](t)$ over $[\,0, 1\,]$ is $(e-1)/720 \approx 0.0024$. Thus, the upper and lower bounds on $f(t) = e^t$ deviate by no more than 1 per cent over the entire interval $[\,0, 1\,]$, and on average much less than this.

Example 2. With $f(t) = \sin(\pi t/2)$ we have $f(0) = 0$, $f'(0) = \pi/2$, $f(1) = 1$, $f'(1) = 0$, and $f^{(4)}(t) \in [\,0, \pi^4/16\,]$ for $t \in [\,0, 1\,]$. In this case, we have

$$[\,F_{4,0}\,] = 0\,,\ [\,F_{4,1}\,] = \frac{\pi}{8}\,,\ [\,F_{4,2}\,] = \frac{1}{2304}[\,192(6+\pi), 192(6+\pi) + \pi^4\,]\,, \tag{12.66}$$

$$[\,F_{4,3}\,] = 1\,,\ [\,F_{4,4}\,] = 1\,.$$

and the interval–valued coefficient is approximately $[\,F_{4,2}\,] \approx [\,0.7618, 0.8041\,]$. The overall width of $[\,F_4\,](t)$ over $[\,0, 1\,]$ is now $\pi^4/11520 \approx 0.0085$, somewhat larger than in the preceding example.

12.3.3 Estimating bounds on derivatives

If the function $f(t)$ to be approximated on $[\,0, 1\,]$ is actually a *polynomial*, we can appeal to its Bernstein–Bézier form,

$$f(t) = \sum_{k=0}^{n} p_k\, B^n_k(t)\,, \tag{12.67}$$

to obtain the derivative bounds required by the approximation procedure. We recall that the r–th derivative of $f(t)$ can be written as

$$f^{(r)}(t) = \frac{n!}{(n-r)!} \sum_{k=0}^{n-r} \Delta^r p_k\, B^{n-r}_k(t)\,, \tag{12.68}$$

and by the convex hull property we then have

$$\frac{n!}{(n-r)!} \min_k \Delta^r p_k \;\le\; f^{(r)}(t) \;\le\; \frac{n!}{(n-r)!} \max_k \Delta^r p_k \quad \text{for } t \in [\,0, 1\,]\,. \tag{12.69}$$

12.4 Approximation by interval Bézier curves

For brevity we restrict our discussion in this paper to the approximation of plane curves; the extension to three dimensions is straightforward. Much of the preceding discussion concerning the interpolation of values sampled from scalar functions applies also to the interpolation of vector–valued functions, i.e., parametric curves. There is, however, an important difference regarding the interpretation of the remainder term that deserves attention.

If $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$ is a sequence of points corresponding to $n+1$ distinct parameter values $t_0, t_1, \ldots, t_n$ on a plane parametric curve $\mathbf{P}(t) = \{x(t), y(t)\}$, the Lagrange interpolant $\mathbf{P}_n(t)$ to these points is simply

$$\mathbf{P}_n(t) = \sum_{k=0}^{n} \mathbf{P}_k \, L_k(t) \,, \tag{12.70}$$

where the Lagrange basis is as defined in (12.37) above. But the remainder formula in the vector case is *not* obtained by merely substituting $\mathbf{P}^{(n+1)}(\xi)$ in place of $f^{(n+1)}(\xi)$ in equation (12.40). Rather, we must write the errors for the x and y components of $\mathbf{P}_n(t) = \{X_n(t), Y_n(t)\}$ separately

$$E_{n,x}(t) = x(t) - X_n(t) \quad \text{and} \quad E_{n,y}(t) = y(t) - Y_n(t) \,, \tag{12.71}$$

and we then have

$$x(t) = X_n(t) + \frac{x^{(n+1)}(\xi_1)}{(n+1)!} \prod_{k=0}^{n} (t - t_k) \quad \text{and} \quad y(t) = Y_n(t) + \frac{y^{(n+1)}(\xi_2)}{(n+1)!} \prod_{k=0}^{n} (t - t_k) \tag{12.72}$$

for some $\xi_1, \xi_2 \in (a, b)$, where $\xi_1 \neq \xi_2$ in general. However, defining a vector–valued interval for $\mathbf{P}^{(n+1)}(t)$ over $t \in [\, a, b \,]$ in the form

$$[\, \mathbf{P}^{(n+1)} \,] = [\, x_{\min}^{(n+1)}, x_{\max}^{(n+1)} \,] \times [\, y_{\min}^{(n+1)}, y_{\max}^{(n+1)} \,] \tag{12.73}$$

allows us to express the remainder term as

$$\mathbf{P}(t) \in \mathbf{P}_n(t) + \frac{[\, \mathbf{P}^{(n+1)} \,]}{(n+1)!} \prod_{k=0}^{n} (t - t_k) \,. \tag{12.74}$$

In formulating interval–polynomial approximants to parametric curves, it is natural to consider also *piecewise* interval polynomial forms, i.e., *interval splines*. This extension is not difficult, and we will give here only a brief sketch of some of the pertinent ideas.

It is well known that a curvature–continuous piecewise–cubic curve can be constructed so as to interpolate any ordered sequence of points $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$ in the plane (which may or may not be sampled from some other curve). Typically, constructing such a curve requires imposing appropriate "end conditions" and then solving a tridiagonal system of linear equations for parametric derivatives $\mathbf{P}'_0, \mathbf{P}'_1, \ldots, \mathbf{P}'_n$ to be assigned to the data points (see [dB78] or [Far90]). For each span k of the spline curve, we then construct the cubic Hermite interpolant $\mathbf{P}(t)$ on $t \in [\, 0, 1 \,]$ to the end–points $\mathbf{P}(0) = \mathbf{P}_{k-1}$, $\mathbf{P}(1) = \mathbf{P}_k$ and end–derivatives $\mathbf{P}'(0) = \mathbf{P}'_{k-1}$, $\mathbf{P}'(1) = \mathbf{P}'_k$,

$$\mathbf{P}(t) = \mathbf{P}(0)H_0^3(t) + \mathbf{P}'(0)H_1^3(t) + \mathbf{P}'(1)H_2^3(t) + \mathbf{P}(1)H_3^3(t) \,. \tag{12.75}$$

If the data points $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$ were actually sampled from another parametric curve $\mathbf{r}(t)$ and we knew vector–valued bounds on the fourth derivative $\mathbf{r}^{(4)}(t)$ between consecutive points, we could

replace the cubic Hermite arcs (12.75) by quartic interval Bézier arcs of the form

$$[\,\mathbf{P}\,](t) = \sum_{k=0}^{4} [\,\mathbf{P}_{4,k}\,]\, B_k^n(t) \tag{12.76}$$

with control points given by

$$\begin{aligned} [\,\mathbf{P}_{4,0}\,] = \mathbf{P}(0)\,,\ [\,\mathbf{P}_{4,1}\,] = \mathbf{P}(0) + \frac{1}{4}\mathbf{P}'(0)\,, \\ [\,\mathbf{P}_{4,2}\,] = \frac{1}{2}(\mathbf{P}(0) + \mathbf{P}(1)) + \frac{1}{6}(\mathbf{P}'(0) - \mathbf{P}'(1)) + \frac{1}{144}[\,\mathbf{P}^{(4)}\,]\,, \\ [\,\mathbf{P}_{4,3}\,] = \mathbf{P}(1) - \frac{1}{4}\mathbf{P}'(1)\,,\ [\,\mathbf{P}_{4,4}\,] = \mathbf{P}(1)\,. \end{aligned} \tag{12.77}$$

At each of the data points $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$ the width of the interval spline shrinks to zero.

Example 3. A quarter circle, $\mathbf{P}(t) = (\cos\frac{\pi t}{2}, \sin\frac{\pi t}{2})$, can be approximated as a quartic interval Bézier using the coefficients expressed in equation (12.67):

$$[\,\mathbf{P}_{4,0}\,] = (1,0);\ \ [\,\mathbf{P}_{4,1}\,] = (1, \frac{\pi}{8});\ \ [\,\mathbf{P}_{4,2}\,] = ([0.7618, 0.8041], [0.7618, 0.8041]); \tag{12.78}$$

$$[\,\mathbf{P}_{4,3}\,] = (\frac{\pi}{8}, 1);\ \ [\,\mathbf{P}_{4,4}\,] = (0,1)\,.$$

While a quarter circle can be expressed exactly as a rational curve, the current approximation

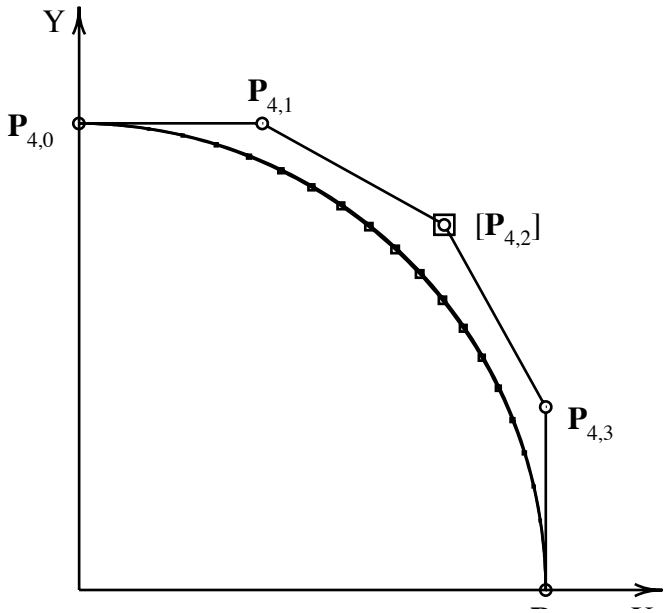

Figure 12.7: Approximate arc length parametrization of circle.

is useful because it approximates an *arc length* parametrization of the circle within two digits of accuracy.

In practice, one might simply choose the middle control point to be at the interval center $[\,\mathbf{P}_{4,2}\,] = (0.78295, 0.78295)$, and use the control point interval as an assurance that the largest possible error, which occurs at $t = 0.5$, is

$$\binom{4}{2}\left(\frac{0.8041 - 0.7618}{2}\right)\left(\frac{1}{2}\right)^4 = 0.0079. \tag{12.79}$$