BIT Numerical Mathematics
?, Vol. ?, No. ?, pp. ?–?

0006-3835/03/4301-0001 $16.00

ARC LENGTH ESTIMATION AND THE CONVERGENCE OF POLYNOMIAL CURVE INTERPOLATION *

M. S. FLOATER[1] †

[1] *Centre of Mathematics for Applications, Institute for Informatics, University of Oslo, P.B. 1053 Blindern, 0316 Oslo, Norway. email: michaelf@ifi.uio.no*

Abstract.

When fitting parametric polynomial curves to sequences of points or derivatives we have to choose suitable parameter values at the interpolation points. This paper investigates the effect of the parameterization on the approximation order of the interpolation. We show that chord length parameter values yield full approximation order when the polynomial degree is at most three. We obtain full approximation order for arbitrary degree by developing an algorithm which generates more and more accurate approximations to arc length: the lengths of the segments of an interpolant of one degree provide parameter intervals for interpolants of degree two higher. The algorithm can also be used to estimate the length of a curve and its arc-length derivatives.

AMS subject classification (2000): 65D05, 65D10.

Key words: Curve parameterization, curve length, arc length, Lagrange interpolation, Hermite interpolation, approximation order.

1 Introduction

In this paper we address the problem of interpolating a curve $C \subset \mathbb{R}^d$, $d \geq 2$, with a parametric polynomial curve, given only a sample of points $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n \in C$, in sequence; see Figure 1.1. This situation arises frequently in geometric modelling; the curve C might be an algebraic curve, a height contour, or perhaps a physical curve, and no parametric representation of C is available.

The standard approach to this problem is a two-step procedure. Firstly we choose a *parameterization*, i.e., an increasing sequence of parameter values

$$t_0 < t_1 < \cdots < t_n, \tag{1.1}$$

and secondly we find the unique polynomial $\mathbf{p}_n : [t_0, t_n] \to \mathbb{R}^d$ of degree $\leq n$ satisfying

$$\mathbf{p}_n(t_i) = \mathbf{x}_i, \qquad i = 0, 1, \ldots, n. \tag{1.2}$$

We have in mind here that n is relatively small. For large n it is more usual to replace $\mathbf{p}_n$ by an interpolating spline curve.

*Received June 2004. Revised February 2005. Communicated by Lothar Reichel.
†This work was supported by the Bemata program of the Norwegian Research Council.

A simple choice of parameter values is the uniform ones, $t_i = i$. However, it is known from long experience that the *chord length parameterization*

$$t_{i+1} - t_i = |\mathbf{x}_{i+1} - \mathbf{x}_i|, \tag{1.3}$$

where $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^d$, often leads to better behaved interpolants. This chordal parameterization was proposed as early as 1967 by Ahlberg, Nilson, and Walsh ([1], Sec. 2.6) for cubic spline interpolation, the motivation being that the chordal length $|\mathbf{x}_{i+1} - \mathbf{x}_i|$ approximates the arc length of the piece of C between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. Later Epstein [8] showed that chordal parameter intervals have the advantage that they can guarantee that an interpolating cubic spline curve will be regular (its first derivative never vanishes). Other choices of parameterization have been proposed both for polynomial and spline interpolation, such as the 'centripetal' one; see Lee [14].

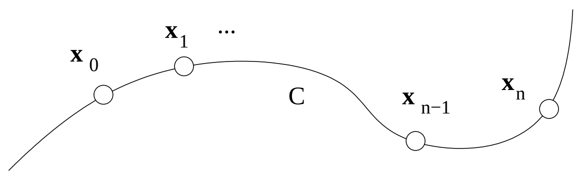

Figure 1.1: Sample of points from a curve.

Currently, however, little seems to be known about the effect of such parameterizations on the *approximation error*, viewed as the distance between $\mathbf{p}_n$ and C. The goal of this paper is to begin to address this important issue. Though we have in mind both polynomial and spline interpolation, we will focus here purely on the polynomial case, deferring a treatment of the spline case to a later paper. Ideally, we would like to determine parameter values t_i, viewed as functions $t_i(\mathbf{x}_0, \ldots, \mathbf{x}_n)$, which give 'full' approximation order when C is smooth enough, i.e., we would like the error between C and $\mathbf{p}_n$ to be of the order h^{n+1} as the points $\mathbf{x}_0, \ldots, \mathbf{x}_n$ get closer, where h is the distance along C from $\mathbf{x}_0$ to $\mathbf{x}_n$, analogous to the polynomial interpolation of functions.

In contrast, several papers [2, 5, 10, 13, 16, 17] deal with the approximation order of so-called geometric interpolants. These schemes, mainly Hermite, aim not only to retain full approximation order, but also to keep the degree of the interpolant low. The potential advantage of these schemes is that both the interpolant and parameterization are the simultaneous solutions to a set of equations. The disadvantage is that these equations are non-linear and only admit a solution under certain restrictions on the data points, and each scheme is dependent on the dimension d. Thus while it might be worth applying some of these non-linear schemes to specialized data sets such as points lying on a convex curve, or to special problems such as the conversion of circles and conic sections to polynomials [6, 9], it seems better in general to apply the two-step approach (1.1–1.2).

In this paper we show that chord length parameter values already guarantee full approximation order for $n \leq 3$. Thus quadratic and cubic interpolants have third and fourth order accuracy respectively. This helps to explain what has often been observed empirically; that chord lengths are 'good enough' for cubic interpolation. For $n \geq 4$, we develop an algorithm which recursively improves on chord lengths to generate parameter values which again guarantee full approximation order. Later we study curve length estimation, numerical differentiation, the extension to Hermite interpolation, and give some numerical examples.

2 A condition for full approximation order

We begin by deriving a natural and sufficient condition on the parameter values which guarantees full approximation order. In order to analyze the approximation error, let $\mathbf{f} : [a, b] \to \mathbb{R}^d$ be an arc-length parametric representation of C (or the piece of C in question). Thus $C = \{\mathbf{f}(s) : a \leq s \leq b\}$ and $\mathbf{f}$ is differentiable with $|\mathbf{f}'(s)| = 1$, for $a \leq s \leq b$. The representation $\mathbf{f}$ is unique up to orientation and a translation in s. We will further assume throughout that $\mathbf{f}$ has smoothness C^{n+1}.

Any ordered sample of points $\mathbf{x}_0, \ldots, \mathbf{x}_n \in C$ can now be expressed as $\mathbf{x}_i = \mathbf{f}(s_i)$ for some increasing parameter values

$$a < s_0 < s_1 < \cdots < s_n < b. \tag{2.1}$$

The piece of C between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ is now $\mathbf{f}|_{[s_i, s_{i+1}]}$ and its length is

$$L(\mathbf{f}|_{[s_i,s_{i+1}]}) = \int_{s_i}^{s_{i+1}} |\mathbf{f}'(s)|\, ds = s_{i+1} - s_i.$$

Also h is now the length of $\mathbf{f}|_{[s_0,s_n]}$, which is simply $s_n - s_0$.

We will search for parameterizations $t_0 < \cdots < t_n$, such that $t_{i+1} - t_i \approx s_{i+1} - s_i$. We will in fact show that if this approximation is good enough then the polynomial interpolation (1.2) has full approximation order. To this end, it will help to make the following definition. We will say that a parameterization $t_0 < \cdots < t_n$, where each t_i is viewed as a function $t_i(\mathbf{x}_0, \ldots, \mathbf{x}_n)$, is *supportive* if

$$|(s_{i+1} - s_i) - (t_{i+1} - t_i)| = O((s_{i+1} - s_i)\rho_{i,n}(\mathbf{s})) \qquad \text{as } h \to 0, \tag{2.2}$$

for all $i = 0, 1, \ldots, n-1$, where

$$\rho_{i,n}(\mathbf{s}) = \prod_{j=0}^{i-1}(s_{i+1} - s_j) \prod_{k=i+2}^{n} (s_k - s_i),$$

and $\mathbf{s} = (s_0, s_1, \ldots, s_n)$. Notice that

$$(s_{i+1} - s_i)^{n-1} \leq \rho_{i,n}(\mathbf{s}) \leq h^{n-1}, \tag{2.3}$$

and so the intervals $t_{i+1} - t_i$ of a supportive parameterization approximate the arc lengths $s_{i+1} - s_i$ to order n. Note further that by dividing (2.2) by $s_{i+1} - s_i$, we see that

$$\frac{1}{2} \leq \frac{t_{i+1} - t_i}{s_{i+1} - s_i} \leq 2,$$

for small enough h. In fact, since $s_k - s_j = \sum_{i=j}^{k-1}(s_{i+1} - s_i)$ for any $0 \le j < k \le n$, it follows that

$$\frac{1}{2} \le \frac{t_k - t_j}{s_k - s_j} \le 2, \tag{2.4}$$

for small enough h.

To measure the distance between $\mathbf{f}$ and $\mathbf{p}_n$ we use reparameterization and define

$$d_P(\mathbf{f}|_{[s_0,s_n]}, \mathbf{p}_n) = \inf_{\phi} \|\mathbf{f} \circ \phi - \mathbf{p}_n\|,$$

where $\|\cdot\| = \max_{t_0 \le t \le t_n} |\cdot(t)|$, and the infimum is taken over strictly increasing, C^1 functions $\phi : [t_0, t_n] \to [s_0, s_n]$ with $\phi(t_0) = s_0$ and $\phi(t_n) = s_n$. It can be shown that $d_P(\mathbf{f}_1, \mathbf{f}_2)$ is a metric on pairs of regular curves $\mathbf{f}_1$ and $\mathbf{f}_2$ and that $d_P(\mathbf{f}_1, \mathbf{f}_2) \ge d_H(\mathbf{f}_1, \mathbf{f}_2)$ where d_H is the Hausdorff metric [15].

Theorem 2.1. *If the parameterization (1.1) is supportive then*

$$d_P(\mathbf{f}|_{[s_0,s_n]}, \mathbf{p}_n) = O(h^{n+1}) \quad \text{as } h \to 0. \tag{2.5}$$

Proof. Let $\phi : [t_0, t_n] \to \mathbb{R}$ be the Lagrange polynomial of degree $\le n$ satisfying $\phi(t_i) = s_i$ for $0 \le i \le n$. We will show that when h is small enough, ϕ is monotonically increasing and therefore a function $[t_0, t_n] \to [s_0, s_n]$. In this case the reparameterization

$$\mathbf{g}(t) = \mathbf{f}(\phi(t)), \qquad t \in [t_0, t_n], \tag{2.6}$$

is well-defined and satisfies $\mathbf{p}_n(t_i) = \mathbf{g}(t_i)$, $i = 0, 1, \ldots, n$. We can then use the usual Newton error formula

$$\mathbf{g}(t) - \mathbf{p}_n(t) = \psi_n(t)[t_0, \ldots, t_n, t]\mathbf{g}, \tag{2.7}$$

where $\psi_n(t) = \prod_{i=0}^{n}(t - t_i)$, giving, by the Genocchi-Hermite formula ([11], Sec. 6.1),

$$\|\mathbf{g} - \mathbf{p}_n\| \le (t_n - t_0)^{n+1} \|\mathbf{g}^{(n+1)}\|/(n+1)!.$$

Since $t_n - t_0 = O(h)$ by (2.4), this clearly establishes (2.5) as long as $\mathbf{g}^{(n+1)}$ is bounded as $h \to 0$. To show this consider the set partition version of Faa di Bruno's formula [12]

$$\mathbf{g}^{(k)}(t) = \sum_{\pi \in \Pi_k} \mathbf{f}^{(|\pi|)}(\phi(t)) \prod_{B \in \pi} \phi^{(|B|)}(t), \tag{2.8}$$

where Π_k is the set of partitions of the set $\{1, 2, \ldots, k\}$, B is a block in such a partition, and $|\cdot|$ denotes set cardinality. Since $\mathbf{f}', \mathbf{f}'', \ldots, \mathbf{f}^{(n+1)}$ are bounded by assumption, the formula with $k = n + 1$ shows that it suffices that the derivatives $\phi', \phi'', \ldots, \phi^{(n+1)}$ are bounded as $h \to 0$. The highest derivative, $\phi^{(n+1)}$, is trivially bounded since it is zero. To deal with $\phi', \ldots, \phi^{(n)}$, consider condition (2.2). Dividing it by $t_{i+1} - t_i$ and using the estimate (2.4) implies that for h small enough,

$$|[t_i, t_{i+1}]\phi - 1| \le K\rho_{i,n}(\mathbf{t}). \tag{2.9}$$

We claim that for all $k = 2, 3, \ldots, n$, and all $i = 0, 1, \ldots, n-k$,

$$|[t_i, \ldots, t_{i+k}]\phi| \leq 2^{k-1} K \prod_{j=0}^{i-1} (t_{i+k} - t_j) \prod_{\ell=i+k+1}^{n} (t_\ell - t_i). \tag{2.10}$$

We begin with the case $k = 2$ and observe that

$$|[t_i, t_{i+1}, t_{i+2}]\phi| \leq \frac{|[t_{i+1}, t_{i+2}]\phi - 1|}{t_{i+2} - t_i} + \frac{|[t_i, t_{i+1}]\phi - 1|}{t_{i+2} - t_i}. \tag{2.11}$$

Since

$$\prod_{\ell=i+3}^{n} (t_\ell - t_{i+1}) \leq \prod_{\ell=i+3}^{n} (t_\ell - t_i), \quad \text{and} \quad \prod_{j=0}^{i-1} (t_{i+1} - t_j) \leq \prod_{j=0}^{i-1} (t_{i+2} - t_j),$$

both terms on the right hand side of (2.11) are bounded above by

$$K \prod_{j=0}^{i-1} (t_{i+2} - t_j) \prod_{\ell=i+3}^{n} (t_\ell - t_i),$$

from which case $k = 2$ of (2.10) follows. The remaining cases ($k = 3, \ldots, n$) follow inductively in a similar fashion, using the inequality

$$|[t_i, \ldots, t_{i+k}]\phi| \leq \frac{|[t_{i+1}, \ldots, t_{i+k}]\phi|}{t_{i+k} - t_i} + \frac{|[t_i, \ldots, t_{i+k-1}]\phi|}{t_{i+k} - t_i}.$$

Inequalities (2.9–2.10) imply that $|[t_0, t_1]\phi - 1| = O(h^{n-1})$ and $|[t_0, \ldots, t_k]\phi| = O(h^{n-k})$ for $2 \leq k \leq n$. Now we turn to the derivatives of ϕ. From the Newton form of ϕ,

$$\phi'(t) - 1 = [t_0, t_1]\phi - 1 + \sum_{i=2}^{n} \psi'_{i-1}(t)[t_0, \ldots, t_i]\phi,$$

$$\phi^{(k)}(t) = \sum_{i=k}^{n} \psi_{i-1}^{(k)}(t)[t_0, \ldots, t_i]\phi, \quad k = 2, \ldots, n,$$

and due to the product rule, we have $\|\psi_{i-1}^{(k)}\| = O(h^{i-k})$ for $0 \leq k \leq i$. It follows that

$$\|\phi' - 1\| = O(h^{n-1}), \quad \text{and} \quad \|\phi^{(k)}\| = O(h^{n-k}) \quad k = 2, 3, \ldots, n. \tag{2.12}$$

Thus all derivatives of ϕ are bounded as $h \to 0$, and for small enough h, $\phi'(t) > 0$, in which case ϕ is monotonically increasing. □

We complete this section with a technical lemma which will be useful later.

LEMMA 2.2. *Under the assumptions of Theorem 2.1, all derivatives* $\mathbf{g}^{(k)}$ *and* $\mathbf{p}_n^{(k)}$, $k = 1, \ldots, n$, *are bounded as* $h \to 0$ *and both* $|\mathbf{g}'|$ *and* $|\mathbf{p}_n'|$ *are bounded away from zero for small enough* h.

PROOF. That $\|\mathbf{g}^{(k)}\|$ is bounded for $k = 1, \ldots, n$ follows from (2.8) and (2.12), and $|\mathbf{g}'|$ is bounded away from zero for small enough h because $|\mathbf{g}'(t)| = \phi'(t)$. The same properties hold for $\mathbf{p}_n$, because by the formula of [7, 18] and the Genocchi-Hermite formula,

$$\|\mathbf{g}^{(k)} - \mathbf{p}_n^{(k)}\| \le C_n(t_n - t_0)^{n+1-k}\|\mathbf{g}^{(n+1)}\| = O(h^{n+1-k}). \tag{2.13}$$

□

3 Chord length parameterization

With Theorem 2.1 in mind, consider next what is surely the simplest estimate for the length of a curve between two points, namely the length of the 'chord' (the line segment) between them. Suppose that $n = 1$, so that $a < s_0 < s_1 < b$ and consider the error between the length $L = |\mathbf{f}(s_1) - \mathbf{f}(s_0)|$ of the 'chord' and the length $h = s_1 - s_0$ of the curve piece $\mathbf{f}|_{[s_0,s_1]}$. The following derivation gives an explicit bound, of order $O(h^3)$, on the error between L and h requiring only the C^2 smoothness of $\mathbf{f}$. Note that since $\mathbf{f}$ is parameterized with respect to arc length, we have

$$\mathbf{f}'(s) \cdot \mathbf{f}'(s) = 1, \qquad \mathbf{f}'(s) \cdot \mathbf{f}''(s) = 0.$$

PROPOSITION 3.1.

$$0 \le h - L \le \frac{h^3}{12}\|\mathbf{f}''\|^2.$$

Note that $\|\mathbf{f}''\|$ is simply the maximum curvature of $\mathbf{f}$.

PROOF. Since L is the length of the shortest path between $\mathbf{f}(s_0)$ and $\mathbf{f}(s_1)$, we trivially have the lower bound $h - L \ge 0$. Regarding the upper bound, let $s = (s_0 + s_1)/2$ and consider the two Taylor series

$$\mathbf{f}(s_1) = \mathbf{f}(s) + \frac{h}{2}\mathbf{f}'(s) + \int_s^{s_1} (s_1 - t)\mathbf{f}''(t)\,dt,$$
$$\mathbf{f}(s_0) = \mathbf{f}(s) - \frac{h}{2}\mathbf{f}'(s) + \int_{s_0}^{s} (t - s_0)\mathbf{f}''(t)\,dt.$$

Taking the scalar product of the difference $\mathbf{f}(s_1) - \mathbf{f}(s_0)$ with itself yields

$$|\mathbf{f}(s_1) - \mathbf{f}(s_0)|^2 \ge h^2 + 2h\left(\int_s^{s_1} (s_1 - t)\mathbf{f}'(s)\cdot\mathbf{f}''(t)\,dt - \int_{s_0}^{s} (t - s_0)\mathbf{f}'(s)\cdot\mathbf{f}''(t)\,dt\right).$$

Since

$$\mathbf{f}'(s) = \mathbf{f}'(t) - \int_s^t \mathbf{f}''(u)\,du = \mathbf{f}'(t) + \int_t^s \mathbf{f}''(u)\,du,$$

it follows that

$$\begin{aligned}
&h^2 - L^2 \\
&\le 2h\left(\int_s^{s_1}(s_1-t)\int_s^t \mathbf{f}''(u)\cdot\mathbf{f}''(t)\,du\,dt + \int_{s_0}^{s}(t-s_0)\int_t^s \mathbf{f}''(u)\cdot\mathbf{f}''(t)\,du\,dt\right) \\
&\le 2h\left(\max_{s_0\le v\le s_1}|\mathbf{f}''(v)|\right)^2\left(\int_s^{s_1}(s_1-t)(t-s)\,dt + \int_{s_0}^{s}(t-s_0)(s-t)\,dt\right) \\
&= 4h\|\mathbf{f}''\|^2\int_0^{h/2} t(h/2-t)\,dt = \frac{h^4}{12}\|\mathbf{f}''\|^2,
\end{aligned}$$

and therefore

$$h - L = \frac{h^2-L^2}{h+L} \le \frac{h^4}{12(h+L)}\|\mathbf{f}''\|^2.$$

□

Returning to the problem of finding a supportive parameterization, with the chord length parameterization (1.3), Proposition 3.1 and (2.3) imply that

$$|(s_{i+1}-s_i)-(t_{i+1}-t_i)| = O\big((s_{i+1}-s_i)^n\big) = O((s_{i+1}-s_i)\rho_{i,n}(\mathbf{s})),$$

for both $n=2$ and $n=3$. This parameterization is therefore supportive for both $n=2$ and $n=3$, and provides full approximation order for quadratic and cubic interpolation.

4 Improved parameterization

For degrees $n \ge 4$, chord lengths can no longer be guaranteed to give full approximation order, as is clear from the numerical examples in Section 9. To overcome this, we will use the intuitive idea of improving the chord length estimate of arc length between two points by fitting a curve and using the length of the curve segment between the two points. Like the use of chord lengths, this kind of iterative improvement was suggested as a possibility for spline interpolation by Ahlberg, Nilson, and Walsh [1] but no analysis was given. Using Theorem 2.1 and the following estimate we can now turn this intuitive idea into an algorithm for generating a supportive parameterization for any n.

THEOREM 4.1. *If $t_0 < \cdots < t_n$ is a supportive parameterization and $\mathbf{p}_n$ is the Lagrange interpolant (1.2) then for $i = 0,1,\ldots,n-1$,*

$$|(s_{i+1}-s_i) - L(\mathbf{p}_n|_{[t_i,t_{i+1}]})| = O\big((s_{i+1}-s_i)^3\rho_{i,n}(\mathbf{s})\big) \quad \textit{as } h\to 0. \tag{4.1}$$

PROOF. Clearly, $s_{i+1}-s_i = L(\mathbf{g}|_{[t_i,t_{i+1}]})$ where $\mathbf{g}$ is the reparameterization of $\mathbf{f}$ in (2.6). Thus (4.1) compares the length of a piece of $\mathbf{p}_n$ with the corresponding piece of $\mathbf{g}$. This leads us to study the error $\mathbf{e}(t) = \mathbf{g}(t) - \mathbf{p}_n(t)$. We will show that for $t \in [t_i, t_{i+1}]$,

$$|\mathbf{e}(t)| \le (t_{i+1}-t_i)^2\rho_{i,n}(\mathbf{t})\|\mathbf{g}^{(n+1)}\|/(n+1)!, \tag{4.2}$$

$$|\mathbf{e}'(t)| \le (t_{i+1}-t_i)\rho_{i,n}(\mathbf{t})\|\mathbf{g}^{(n+1)}\|/n!. \tag{4.3}$$

Using the error formula (2.7), the bound (4.2) follows from the obvious estimate

$$|\psi_n(t)| \le (t_{i+1}-t_i)^2 \rho_{i,n}(\mathbf{t}), \qquad t_i \le t \le t_{i+1}.$$

To prove (4.3), we express $\mathbf{e}'$ in the form [7, 18]

$$\mathbf{e}'(t) = \sum_{j=0}^{n-1} \psi_n^j(t)[t_0,\dots,t_n,t]\mathbf{g} + \psi_n^n(t)[t_0,\dots,t_{n-1},t,t]\mathbf{g},$$

where $\psi_n^j(t) = \prod_{k=0,k\ne j}^n (t-t_k)$, from which it follows that

$$|\mathbf{e}'(t)| \le \max_{0\le j\le n} |\psi_n^j(t)|\,\|\mathbf{g}^{(n+1)}\|/n!,$$

and so equality (4.3) will be established if we can show that

$$|\psi_n^j(t)| \le (t_{i+1}-t_i)\rho_{i,n}(\mathbf{t}), \qquad t_i \le t \le t_{i+1}, \tag{4.4}$$

for any $j = 0,1,\dots,n$. This obviously holds for $j=i$ and $j=i+1$. For $j<i$,

$$|\psi_n^j(t)| \le (t_{i+1}-t_i)^2 \prod_{k=0,k\ne j}^{i-1} (t_{i+1}-t_k) \prod_{\ell=i+2}^{n} (t_\ell - t_i),$$

and (4.4) follows because $(t_{i+1}-t_i) \le (t_{i+1}-t_j)$. A similar treatment deals with the remaining case $j > i+1$.

Having now established the bounds (4.2–4.3) on $\mathbf{e}$ and $\mathbf{e}'$, observe that since

$$|\mathbf{g}'(t)|^2 - |\mathbf{p}_n'(t)|^2 = 2\mathbf{e}'(t)\cdot\mathbf{g}'(t) - \mathbf{e}'(t)\cdot\mathbf{e}'(t),$$

$$|L(\mathbf{g}|_{[t_i,t_{i+1}]}) - L(\mathbf{p}_n|_{[t_i,t_{i+1}]})| \le 2\left|\int_{t_i}^{t_{i+1}} \frac{\mathbf{e}'(t)\cdot\mathbf{g}'(t)}{|\mathbf{g}'(t)|+|\mathbf{p}_n'(t)|}\,dt\right| + \int_{t_i}^{t_{i+1}} \frac{|\mathbf{e}'(t)|^2}{|\mathbf{g}'(t)|+|\mathbf{p}_n'(t)|}\,dt.$$

Now by Lemma 2.2, $|\mathbf{g}'| + |\mathbf{p}_n'|$ is bounded away from zero for small enough h. Thus the bound (4.3) on $\mathbf{e}'$ implies that the second term on the right hand side is of order $O((t_{i+1}-t_i)^3\rho_{i,n}^2(\mathbf{t}))$, and therefore also of order $O((t_{i+1}-t_i)^3\rho_{i,n}(\mathbf{t}))$. Considering the first term on the right hand side, note that $\mathbf{e}(t_i) = \mathbf{e}(t_{i+1}) = \mathbf{0}$. So by integration by parts,

$$\int_{t_i}^{t_{i+1}} \frac{\mathbf{e}'(t)\cdot\mathbf{g}(t)}{|\mathbf{g}'(t)|+|\mathbf{p}_n'(t)|}\,dt = -\int_{t_i}^{t_{i+1}} \mathbf{e}(t)\cdot\frac{d}{dt}\Big(\frac{\mathbf{g}'(t)}{|\mathbf{g}'(t)|+|\mathbf{p}_n'(t)|}\Big)\,dt,$$

whose absolute value is, by virtue of (4.2), also of order $O((t_{i+1}-t_i)^3\rho_{i,n}(\mathbf{t}))$, because

$$\frac{d}{dt}\Big(\frac{\mathbf{g}'(t)}{|\mathbf{g}'(t)|+|\mathbf{p}_n'(t)|}\Big)$$

is bounded as $h\to 0$ by Lemma 2.2. Thus, by (2.4), we have shown that $|L(\mathbf{g}|_{[t_i,t_{i+1}]}) - L(\mathbf{p}_n|_{[t_i,t_{i+1}]})|$ is of order $O((s_{i+1}-s_i)^3\rho_{i,n}(\mathbf{s}))$. □

The similarity between the condition (2.2) of Theorem 2.1 and the estimate in Theorem 4.1 is striking. The only difference is that the latter estimate has two extra powers of $(s_{i+1} - s_i)$ in front, and this is what allows us to iteratively improve the accuracy of a parameterization, jumping two orders of approximation at each step.

We cannot in general find the length $L(\mathbf{p}_n|_{[t_i,t_{i+1}]})$ exactly. In practice we must use some approximation to it, $M(\mathbf{p}_n|_{[t_i,t_{i+1}]})$. As long as

$$|L(\mathbf{p}_n|_{[t_i,t_{i+1}]}) - M(\mathbf{p}_n|_{[t_i,t_{i+1}]})| = O\big((t_{i+1} - t_i)^{n+2}\big), \tag{4.5}$$

the approximation $M(\mathbf{p}_n|_{[t_i,t_{i+1}]})$ will also satisfy (4.1) by the triangle inequality. We can achieve the approximation order (4.5) by approximating the integral $\int_{t_i}^{t_{i+1}} q(t)\,dt$, where $q(t) = |\mathbf{p}_n'(t)|$, using a standard quadrature rule such as a Newton-Cotes or Gauss rule, with degree of precision n or higher, of the form

$$M_n(\mathbf{p}_n|_{[t_i,t_{i+1}]}) = \sum_{j=0}^{m} w_j q(x_j),$$

for appropriate points $x_0, \dots, x_m$ in the interval $[t_i, t_{i+1}]$. The error (4.5) then follows from the fact that the $(n+1)$-st derivative of q is bounded, due to the properties of $\mathbf{p}_n$ of Lemma 2.2.

5 Parameterization algorithm

Based on the theoretical results, we propose a general algorithm which computes a supportive parameterization for arbitrary degree n. For $n \le 3$, we use chord lengths, while for $n \ge 4$, we use recursion as follows. First set $t_0 = 0$ and then, for each $i = 0, \dots, n-1$, choose any $j \in \{0, 1, 2\}$ such that $j \le i < i + 1 \le j + n - 2$. We then apply the algorithm with n replaced by $n - 2$ to compute supportive parameter values $u_j < \cdots < u_{n+2-j}$ for the points $\mathbf{x}_j, \dots, \mathbf{x}_{n-2+j}$ and find the interpolant $\mathbf{p}_{n-2}$ such that $\mathbf{p}_{n-2}(u_k) = \mathbf{x}_k$, for $k = j, \dots, n-2+j$, and set $t_{i+1} = t_i + M_{n-2}(\mathbf{p}_{n-2}|_{[u_i,u_{i+1}]})$. It follows from Theorem 4.1 that the resulting parameterization is supportive.

For some i there is a choice of (at most three) polynomials to use to calculate $t_{i+1} - t_i$. We can make a computational saving by taking $j = 0$ for $i < n/2$ and $j = 2$ for $i \ge n/2$ for this requires finding only two polynomials of degree $n - 2$, as in the algorithm Param1 below.

6 Estimating curve length

If we merely want to estimate the length of the curve piece $\mathbf{f}|_{[s_0,s_n]}$ from the points $\mathbf{x}_0, \dots, \mathbf{x}_n$, we can use Param1 giving the algorithm Length below.
From Theorem 4.1, this estimate for the length h of $\mathbf{f}|_{[s_0,s_n]}$ has the approximation order

$$|h - \mathrm{Length}(\mathbf{x}_0, \dots, \mathbf{x}_n)| = O\big(h^{n+2}\big).$$

This length algorithm could be used piecewise to form a composite algorithm for approximating the length of a curve from a large sample of points, analogous to composite rules for numerical quadrature.

Algorithm: Param1

Param1($\mathbf{x}_0, \ldots, \mathbf{x}_n$) {
    $t_0 = 0$;
    if($n \le 3$) for($i = 0$; $i < n$; i++) $t_{i+1} = t_i + \|\mathbf{x}_{i+1} - \mathbf{x}_i\|$;
    else {
        $(u_0, \ldots, u_{n-2}) = \text{Param1}(\mathbf{x}_0, \ldots, \mathbf{x}_{n-2})$;
        Find polynomial $\mathbf{p}_{n-2}$ s.t. $\mathbf{p}_{n-2}(u_k) = \mathbf{x}_k$, $k = 0, \ldots, n-2$;
        for($i = 0$; $i < n/2$; i++) $t_{i+1} = t_i + M_{n-2}(\mathbf{p}_{n-2}|_{[u_i,u_{i+1}]})$;
        $(v_2, \ldots, v_n) = \text{Param1}(\mathbf{x}_2, \ldots, \mathbf{x}_n)$;
        Find polynomial $\mathbf{q}_{n-2}$ s.t. $\mathbf{q}_{n-2}(v_k) = \mathbf{x}_k$, $k = 2, \ldots, n$;
        for($i = n/2$; $i < n$; i++) $t_{i+1} = t_i + M_{n-2}(\mathbf{q}_{n-2}|_{[v_i,v_{i+1}]})$; }
    return $(t_0, \ldots, t_n)$; }

Algorithm: Length

Length($\mathbf{x}_0, \ldots, \mathbf{x}_n$) {
    $(u_0, \ldots, u_n) = \text{Param1}(\mathbf{x}_0, \ldots, \mathbf{x}_n)$;
    Find polynomial $\mathbf{p}_n$ s.t. $\mathbf{p}_n(u_k) = \mathbf{x}_k$, $k = 0, \ldots, n$;
    return $M_n(\mathbf{p}_n|_{[u_0,u_n]})$; }

7 Numerical differentiation

We saw earlier, in (2.13), that if $t_0 < \cdots < t_n$ is a supportive parameterization, then the derivatives of $\mathbf{p}_n$ approximate the derivatives of $\mathbf{g}$ in the same way as for functions, the order of accuracy falling by one as the derivatives increase by one. This gives a way of estimating the derivatives of $\mathbf{g}$ given only the sample of points $\mathbf{x}_0, \ldots, \mathbf{x}_n$. However, $\mathbf{g}$ is a reparameterization of the fixed, arc-length parameterized curve $\mathbf{f}$, and depends on the sample points $\mathbf{x}_i$, so the derivatives of $\mathbf{g}$ are of little interest. What we would prefer to do is to use $\mathbf{p}_n$ to approximate the derivatives of $\mathbf{f}$, since these are fixed, and moreover (since they are arc-length derivatives) they provide intrinsic properties of the curve, such as tangent directions and curvature. One way of doing this is to compute the *arc-length* derivatives of $\mathbf{p}_n$. Denoting differentiation with respect to arc length by D_s, we find

$$D_s\mathbf{p}_n = \frac{\mathbf{p}_n'}{|\mathbf{p}_n'|}, \qquad D_s^2\mathbf{p}_n = \frac{1}{|\mathbf{p}_n'|^2}\mathbf{p}_n'' - \frac{\mathbf{p}_n' \cdot \mathbf{p}_n''}{|\mathbf{p}_n'|^4}\mathbf{p}_n',$$

and so on ([3], Chap. 1). Applying the same formulas to $\mathbf{g}$, and using Lemma 2.2, repeated use of (2.13) implies

$$\|\mathbf{f}^{(k)} \circ \phi - D_s^k\mathbf{p}_n\| = \|D_s^k\mathbf{g} - D_s^k\mathbf{p}_n\| = O(h^{n+1-k}). \tag{7.1}$$

Interestingly though, due to the fact that a supportive parameterization approximates arc length, it turns out that the derivatives of $\mathbf{p}_n$ approximate the derivatives of $\mathbf{f}$ *directly*, with the loss of only one order of accuracy compared with (7.1).

THEOREM 7.1. *If $t_0 < \cdots < t_n$ is a supportive parameterization then for $1 \le k \le n$,*

$$\|\mathbf{f}^{(k)} \circ \phi - \mathbf{p}_n^{(k)}\| = O\big(h^{n-k}\big).$$

PROOF. By the triangle inequality,

$$\|\mathbf{f}^{(k)} \circ \phi - \mathbf{p}_n^{(k)}\| \le \|\mathbf{f}^{(k)} \circ \phi - \mathbf{g}^{(k)}\| + \|\mathbf{g}^{(k)} - \mathbf{p}_n^{(k)}\|,$$

and so from (2.13) it is enough to show that $\|\mathbf{f}^{(k)} \circ \phi - \mathbf{g}^{(k)}\| = O\big(h^{n-k}\big)$. By Faa di Bruno's formula (2.8), we find

$$\mathbf{g}^{(k)}(t) - \mathbf{f}^{(k)}(\phi(t)) = \sum_{j=1}^{k} c_j(t)\mathbf{f}^{(j)}(\phi(t)),$$

where $c_k = (\phi')^k - 1$ and for $1 \le j < k$, c_j is a linear combination of terms of the form

$$\phi^{(i_1)}\phi^{(i_2)}\cdots\phi^{(i_j)}, \tag{7.2}$$

where $i_1+i_2+\cdots+i_j = k$, $i_1, \ldots, i_j \ge 1$, and $i_r \ge 2$ for at least one r in $\{1, \ldots, j\}$. Then (2.12) shows both that $\|c_k\|$ is of order $O(h^{n-1}) = O(h^{n-k})$, and that the product (7.2) is of order $O(h^{n-i_r}) = O(h^{n-k})$, implying that $\|c_j\| = O(h^{n-k})$ for $1 \le j < k$. □

Notice that the case $k = 1$ implies

$$\||\mathbf{p}_n'| - 1\| = \||\mathbf{p}_n'| - |\mathbf{f}' \circ \phi|\| = O\big(h^{n-1}\big), \tag{7.3}$$

and so $\mathbf{p}_n$ is 'asymptotically' arc-length. Note also that Theorem 7.1 implies that

$$|\mathbf{f}^{(k)}(s_i) - \mathbf{p}_n^{(k)}(t_i)| = O(h^{n-k}), \tag{7.4}$$

which shows, for example, that $|\mathbf{p}_n''(t_i)|$ approximates the curvature of $\mathbf{f}$ at s_i.

Not only this, we get the *same* accuracy as in (7.1) if we apply one more step of the iterative parameterization algorithm, giving the algorithm Param2 below. Due to Theorem 4.1 the parameterization generated by Param2 has the property that

$$|(s_{i+1} - s_i) - (t_{i+1} - t_i)| = O((s_{i+1} - s_i)^3 \rho_{i,n}(\mathbf{s})) \quad \text{as } h \to 0. \tag{7.5}$$

The derivatives of ϕ then satisfy the bounds $\|\phi' - 1\| = O(h^{n+1})$ and $\|\phi^{(k)}\| = O(h^{n-k+2})$ for $k = 2, 3, \ldots, n$, which are two orders of h higher than in (2.12). By (2.8) this now implies

$$\|\mathbf{f}^{(k)} \circ \phi - \mathbf{g}^{(k)}\| = O\big(h^{n+2-k}\big), \tag{7.6}$$

Algorithm: Param2

Param2($\mathbf{x}_0, \ldots, \mathbf{x}_n$) {
    $(u_0, \ldots, u_n) = \mathrm{Param1}(\mathbf{x}_0, \ldots, \mathbf{x}_n)$;
    Find polynomial $\mathbf{p}_n$ s.t. $\mathbf{p}_n(u_k) = \mathbf{x}_k$, $k = 0, \ldots, n$;
    $t_0 = 0$;
    for($i = 0$; $i < n$; i++) $t_{i+1} = t_i + M_n(\mathbf{p}_n|_{[u_i, u_{i+1}]})$;
    return $(t_0, \ldots, t_n)$; }

and therefore

$$\|\mathbf{f}^{(k)} \circ \phi - \mathbf{p}_n^{(k)}\| = O\big(h^{n+1-k}\big),$$

which agrees with the accuracy of (7.1). Similarly the orders of convergence in (7.3–7.4) are also increased by one, namely to $O(h^n)$ and $O(h^{n-k+1})$.

For these reasons, we might call a parameterization satisfying (7.5) *super-supportive*. A further aspect of a super-supportive parameterization is that due to (7.6), we have $\|\mathbf{f}^{(n+1)} \circ \phi - \mathbf{g}^{(n+1)}\| = O(h)$, and so the interpolation error is the same asymptotically as for the arc length parameterization $t_{i+1} - t_i = s_{i+1} - s_i$.

In conclusion, Param2 is superior to Param1 from several points of view, although it requires more computation. One can thus argue that chord length parameterization is not quite 'good enough' for cubic interpolation after all. If Param2 is applied to the cubic case, the convergence order of the derivatives increases by one, and only then is the error asymptotically the same as for arc length.

8 Hermite interpolation

The theory generalizes to Hermite interpolation. Hermite data can be specified by allowing some of the s_i in (2.1) to coalesce. For example, the chordal parameterization becomes

$$t_{i+1} - t_i = \begin{cases} |\mathbf{f}(s_{i+1}) - \mathbf{f}(s_i)| & \text{if } s_{i+1} > s_i; \\ 0 & \text{if } s_{i+1} = s_i. \end{cases}$$

More generally, if, for example,

$$s_{i-1} < s_i = \cdots = s_{i+k} < s_{i+k+1}, \tag{8.1}$$

then in any parameterization we will set $t_i = \cdots = t_{i+k}$, and $\mathbf{p}_n$ in (1.2) will be understood to be a Hermite interpolant satisfying the conditions

$$\mathbf{p}_n^{(j)}(t_i) = \mathbf{f}^{(j)}(s_i), \qquad j = 0, 1, \ldots, k, \tag{8.2}$$

(among others). We continue to assume that $\mathbf{f}$ is parameterized with respect to arc length, so the derivatives of $\mathbf{p}_n$ specified in (8.2) are *arc length derivatives*.

To extend Theorem 2.1 to the Hermite case we replace the Lagrange interpolant ϕ by a Hermite one. In the example of (8.1), we set $\phi(t_i) = s_i$, $\phi'(t_i) = 1$, and $\phi''(t_i) = \cdots = \phi^{(k)}(t_i) = 0$. Using (2.8), this ensures that

$$\mathbf{p}_n^{(j)}(t_i) = \mathbf{g}^{(j)}(t_i), \qquad j = 0, 1, \ldots, k.$$

The rest of the proof then extends in a straightforward way. For example, equation (2.9) trivially holds if $t_i = t_{i+1}$ since in that case $\phi'(t_i) = 1$. The bound on $[t_i, \ldots, t_{i+k}]\phi$ in (2.10) trivially holds in the case that $t_i = \cdots = t_{i+k}$ because then it equals $\phi^{(k)}(t_i)/k!$ which is zero by definition. Theorem 4.1 obviously extends to the Hermite case as well.

If the Newton form of interpolation is used in Param1 and Param2, it is easy to generalize them to the Hermite case. The divided differences can be computed in the standard way; see e.g. [4], p. 68.

As an example, consider the important two-point cubic Hermite interpolation, which in simplified notation can be specified as

$$\mathbf{p}_3^{(k)}(t_i) = \mathbf{f}^{(k)}(s_i), \qquad i = 0, 1, \qquad k = 0, 1. \tag{8.3}$$

Our analysis shows that if $\mathbf{f} \in C^4[a, b]$, the chordal values

$$t_1 - t_0 = |\mathbf{f}(s_1) - \mathbf{f}(s_0)|, \tag{8.4}$$

yield fourth-order accuracy, i.e., $d_P(\mathbf{f}|_{[s_0,s_1]}, \mathbf{p}_3) = O\big((s_1 - s_0)^4\big)$ as $s_1 - s_0 \to 0$. In contrast, if we use the chordal values (8.4) for the quintic two-point Hermite interpolation

$$\mathbf{p}_5^{(k)}(t_i) = \mathbf{f}^{(k)}(s_i), \qquad i = 0, 1, \qquad k = 0, 1, 2,$$

the accuracy does not increase to sixth order, as the numerical examples in Section 9 clearly show. This problem can be fixed by Param1: we first let $\mathbf{p}_3$ be some chordal cubic interpolant to four Hermite data which include $\mathbf{f}(s_0)$ and $\mathbf{f}(s_1)$, for example $\mathbf{p}_3$ in (8.3–8.4) (Param1 as written would use the less symmetric conditions $\mathbf{p}_3(t_0) = \mathbf{f}(s_0)$ and $\mathbf{p}_3^{(k)}(t_1) = \mathbf{f}^{(k)}(s_1)$, $k = 0, 1, 2$, but this would do equally well). We then set

$$u_1 - u_0 = M_3(\mathbf{p}_3|_{[t_0,t_1]}), \tag{8.5}$$

and then if

$$\mathbf{p}_5^{(k)}(u_i) = \mathbf{f}^{(k)}(s_i), \qquad i = 0, 1, \qquad k = 0, 1, 2, \tag{8.6}$$

and $\mathbf{f} \in C^6[a, b]$, we have $d_P(\mathbf{f}|_{[s_0,s_1]}, \mathbf{p}_5) = O\big((s_1 - s_0)^6\big)$ as $s_1 - s_0 \to 0$.

Hermite interpolation is of special importance because it can be used to build spline interpolants given a sequence of points and derivatives. Piecing together cubic Hermite interpolants of the form (8.3–8.4) will give a C^1 cubic spline curve with fourth order accuracy. Piecing together quintic Hermite interpolants of the form (8.5–8.6) will give a C^2 quintic spline curve with sixth order accuracy.

9 Numerical examples

Param1 and Param2 were tested on four examples where points and derivatives were sampled from the unit circle $\mathbf{f}(s) = (\cos s, \sin s)$, at the values (angles) s_i^k, $i = 0, 1, \ldots, n$, for each $k = 0, 1, \ldots, 5$, where $s_i^k = s_i/2^k$. For each k, the polynomial curve $\mathbf{p}_n^k$ was computed based on the parameterizations: arc length (for comparison), chord length, Param1 and Param2. The distance between $\mathbf{p}_n^k$ and $\mathbf{f}$ was computed numerically by taking the maximum distance from $\mathbf{p}_n^k$ to the circle over 300 uniformly sampled points.

The first test (Table 9.1) was on Lagrange cubics, with angles $(s_0, s_1, s_2, s_3) = (0.0, 0.2, 1.0, 1.5)$ chosen deliberately to be non-uniform. The second test (Table 9.2) was on Lagrange quintics, with $(s_0, \ldots, s_5) = (0.0, 0.1, 1.0, 1.7, 2.0, 2.2)$. The third and fourth tests (Tables 9.3 and 9.4) were on Hermite cubics and quintics, with angles $(s_0, s_1, s_2, s_3) = (0, 0, 2, 2)$ and $(s_0, \ldots, s_5) = (0, 0, 0, 2, 2, 2)$.

In all the examples, the numerically approximated approximation orders all agree with the theoretical ones. The orders of the error using Param1 are the same as those for arc length, and the error using Param2 is asymptotically the same as for arc length. In these particular examples Param2 gives a somewhat smaller error than Param1. Uniform and centripetal parameterizations were also tested in the four examples but they gave second-order accuracy at best.

Table 9.1: Errors and orders of Lagrange cubics.

k	Arc	order	Chord (Param1)	order	Param2
0	3.54e-03		5.59e-03		4.10e-03
1	2.26e-04	3.97	3.60e-04	3.96	2.35e-04
2	1.42e-05	3.99	2.27e-05	3.99	1.43e-05
3	8.87e-07	4.00	1.42e-06	4.00	8.89e-07
4	5.54e-08	4.00	8.87e-08	4.00	5.55e-08
5	3.47e-09	4.00	5.54e-09	4.00	3.47e-09

Table 9.2: Errors and orders of Lagrange quintics.

k	Arc	order	Chord	order	Param1	order	Param2
0	3.52e-04		4.27e-04		5.08e-04		2.95e-04
1	6.36e-06	5.79	2.41e-05	4.15	8.33e-06	5.93	6.10e-06
2	1.03e-07	5.95	1.60e-06	3.91	1.32e-07	5.98	1.02e-07
3	1.63e-09	5.99	1.02e-07	3.97	2.07e-09	5.99	1.62e-09
4	2.54e-11	6.00	6.42e-09	3.99	3.24e-11	6.00	2.54e-11
5	3.98e-13	6.00	4.02e-10	4.00	5.06e-13	6.00	3.98e-13

Table 9.3: Errors and orders of Hermite cubics.

k	Arc	order	Chord (Param1)	order	Param2
0	3.90e-02		1.06e-01		6.61e-02
1	2.56e-03	3.93	7.49e-03	3.82	3.17e-03
2	1.62e-04	3.98	4.83e-04	3.95	1.72e-04
3	1.02e-05	4.00	3.04e-05	3.99	1.03e-05
4	6.36e-07	4.00	1.91e-06	4.00	6.38e-07
5	3.97e-08	4.00	1.19e-07	4.00	3.98e-08

Table 9.4: Errors and orders of Hermite quintics.

k	Arc	order	Chord	order	Param1	order	Param2
0	1.32e-03		6.50e-02		7.64e-03		2.45e-03
1	2.14e-05	5.94	3.98e-03	4.03	7.66e-05	6.64	2.42e-05
2	3.38e-07	5.99	2.45e-04	4.02	9.42e-07	6.35	3.46e-07
3	5.29e-09	6.00	1.53e-05	4.01	1.36e-08	6.11	5.32e-09
4	8.28e-11	6.00	9.54e-07	4.00	2.08e-10	6.03	8.29e-11
5	1.29e-12	6.00	5.96e-08	4.00	3.24e-12	6.01	1.29e-12

Acknowledgment.

Many thanks to Carl de Boor and the referee for patiently reading the original manuscript. Their comments helped to improve it.

REFERENCES

1. J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The theory of splines and their applications*, Academic Press, New York, 1967.
2. C. de Boor, K. Höllig K., and M. Sabin, High accuracy geometric Hermite interpolation, Comp. Aided Geom. Design **4** (1987), 269–278.
3. M. P. do Carmo, *Differential geometry of curves and surfaces*, Prentice-Hall, New Jersey, 1976.
4. S. D. Conte and C. de Boor, *Elementary numerical analysis*, McGraw-Hill, 1980.
5. W. Degen, High accurate rational approximation of parametric curves, Comp. Aided Geom. Design **10** (1993), 293–313.
6. T. Dokken, M. Dæhlen, T. Lyche, and K. Mørken, Good approximation of circles by curvature-continuous Bézier curves, Comp. Aided Geom. Design **7** (1990), 33–41.
7. T. Dokken and T. Lyche, A divided difference formula for the error in Hermite interpolation, BIT **19** (1979), 539–540.
8. M. P. Epstein, On the influence of parametrization in parametric interpolation, SIAM J. Numer. Anal. **13** (1976), 261–268.
9. M. S. Floater, An $O(h^{2n})$ Hermite approximation for conic sections, Comp. Aided Geom. Design **14** (1997), 135–151.

10. T. A. Grandine and T. A. Hogan, A parametric quartic spline interpolant to position, tangent and curvature, Computing **72** (2004), 65–78.
11. E. Isaacson and H. B. Keller, *Analysis of numerical methods*, Wiley, 1966.
12. W. P. Johnson, The curious history of Faa di Bruno's formula, Amer. Math. Monthly **109** (2002), 217–234.
13. K. Höllig and J. Koch, Geometric Hermite interpolation, CAGD **12** (1995), 567–580.
14. E. T. Y. Lee, Choosing nodes in parametric curve interpolation, Computer-Aided Design **21** (1989), 363–370.
15. T. Lyche and K. Mørken, A metric for parametric approximation, in *Curves and Surfaces*, P. J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters, Wellesley, (1994), 311-318.
16. K. Mørken and K. Scherer, A general framework for high-accuracy parametric interpolation, Math. Comp. **66** (1997), 237–260.
17. R. Schaback, Optimal geometric Hermite interpolation of curves, in *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Lyche & L. L. Schumaker (eds.), Vanderbilt University Press, Nashville, (1998), 417–428.
18. X. Wang, On remainders of numerical differentiation formulas, Ke Xue Tong Bao (in Chinese) **24(19)** (1979), 869–872.