GEOMETRICAL DIFFERENTIATION AND HIGH–ACCURACY CURVE INTERPOLATION

ROBERT SCHABACK
Institut für Numerische und Angewandte Mathematik
Universität Göttingen
Lotzestraße 16-18
W-3400 Göttingen
Germany

ABSTRACT: Let f be a smooth curve in $I\!R^d$, parametrized by arclength. If a large sample of data points $p_i = f(t_i)$ at unknown parameter values $t_i < t_{i+1}$ is given, one can use local n–th degree polynomial interpolation at parameters $s_i = \|P_i - P_\ell\| \text{sgn}(i - \ell)$ of data points P_i around a fixed point P_ℓ to calculate approximations to the derivatives $f^{(j)}(t_\ell)$ with accuracy $\mathcal{O}(h^{n+1-j})$, where $h := \max(t_i - t_{i-1})$ and $0 \le j \le k-1 \le n$. Using these as data for properly parametrized Hermite interpolation problems for polynomials of degree $\le 2k-1 \le n$ between successive data points, one can construct GC^{k-1} interpolants of f with accuracy $\mathcal{O}(h^{2k})$.

1. Introduction

The classical problem of numerical differentiation consists in finding an approximation of the j–th derivative $f^{(j)}(t^*)$ of some smooth real–valued function f on $[a, b] \subset I\!R$ in a given point $t^* \in [a, b]$, if $n + 1$ nodes

$$a \le t_0 < t_1 < \ldots < t_n \le b$$

and $n + 1$ real function values

$$f(t_0), f(t_1), \ldots, f(t_n)$$

are given. The standard approach simply takes the j–th derivative of the n–th degree polynomial p interpolating these data, and the error is easily evaluated from the representation

$$f(t) - p(t) = \left(\prod_{i=0}^{n}(t - t_i)\right) \Delta^{n+1}(t_0, t_1, \ldots, t_n, t) f, \tag{1..0.1}$$

where $\Delta^i(t_0, \ldots, t_i)f$ is the i–th divided difference of f with respect to the nodes $t_0, t_1, \ldots, t_i$. The j–th derivative of (1.1) at $t^* \in [t_0, t_n]$ can then be bounded by

$$|f^{(j)}(t^*) - p^{(j)}(t^*)| \le c \cdot h_t^{n+1-j} \cdot \max_{0 \le i \le j} \|f^{(n+1+i)}\|_{\infty,[a,b]} \tag{1..0.2}$$

with

$$h_t := \max_{1 \leq i \leq n} (t_i - t_{i-1}) \tag{1..0.3}$$

and a constant c which does not depend on f and the node distribution. Other approaches, like Sard's optimal approximation of linear functionals [4][5], and Micchelli's optimal recovery schemes [3], try to find a formula of a certain type, e.g.:

$$f'(t^*) \approx \sum_{i=0}^{n} \alpha_i f(t_i)$$

where the weights α_i are chosen to minimize the error in some well–defined sense.

In Computer–Aided–Design applications the situation is different. The given data only consist of an ordered set of points $P_0, P_1, \ldots, P_n$ in $I\!R^d$, which can be considered as a sample from the range $R := f([a,b])$ of a smooth and regular curve $f : [a,b] \to I\!R^d$. In particular, the points P_i may be written as $P_i = f(s_i)$ for some parameter values s_i which are not available and depend on the parametrization of f. Of course, the s_i might be chosen arbitrarily, but this will introduce some additional and hypothetical information. By **geometrical differentiation** we denote methods that construct data like tangent directions, curvature or torsion values at the P_i by exclusive use of the point sequence $P_0, \ldots, P_n$ and the geometry of the range R of the curve.

Given the range R of f, a canonical parametrization of f by arclength t can theoretically be constructed, and this parametrization depends only on R. Thus $P_i = f(t_i)$ can be assumed for the unknown arclength parametrization in order to derive error estimates.

The "**mesh width**" of the sample can be described by either

$$h_s := \max_{1 \leq i \leq n} \|P_i - P_{i-1}\|_2 = \max_{1 \leq i \leq n} \|f(t_i) - f(t_{i-1})\|_2 \tag{1..0.4}$$

or (1.3) as the maximum of chordlengths or arclengths between successive points.

Clearly, chordlength is numerically accessible while arclength is not. However, once arclength is small enough, the two are equivalent in the sense used for the notion of equivalence of norms:

Lemma 1.5 *If $f : [0,L] \to I\!R^d$ is a C^1 curve, parametrized by arclength, then there is a constant $h_0(f) \in (0,L)$ such that for any two arguments t and $t+h$ with*

$$0 \leq t < t+h \leq L, \;\; 0 < h \leq h_0(f)$$

the inequalities

$$\frac{1}{\sqrt{d+1}}\, h \leq \|f(t+h) - f(t)\|_2 \leq \sqrt{d} h \tag{1..0.6}$$

hold.

Proof. If we consider just one coordinate x of f, we have

$$|x(t+h) - x(t)| = |x'(\tau)|h, \quad t < \tau < t+h, \tag{1..0.7}$$

and since arclength parametrization implies $|x'(t)|^2 \leq \|f'\|_2^2 = 1$, we get the right–hand side of (1.6) by summing squares of components. Since $\|f'\|_2^2 = 1$ holds everywhere, there is a constant $h_0(f)$ such that on every subinterval I of $[0, L]$ of length $h \leq h_0(f)$ there is some component of f' whose absolute value is at least $(d+1)^{-1/2}$ on I. If x is such a component for given points t and $t+h$ with $h \leq h_0(t)$, then (1.7) implies

$$\|f(t+h) - f(t)\|^2 \geq |x(t+h) - x(t)|^2 \geq \frac{1}{d+1}\, h^2.$$

□

The main consequence of Lemma 1.1 is the equivalence of $\mathcal{O}(h_s^k)$ and $\mathcal{O}(h_t^k)$ error estimates for $h_s \to 0$ or $h_t \to 0$: both arclength and chordlength can be used to handle the asymptotics of error bounds. Thus we will generally use h as a symbol to mean h_t or h_s, if a fixed multiplicative constant does not matter.

Now let F be a real–valued functional, not necessarily linear, on a set S of smooth regular curves, parametrized on some interval $[0, L]$, e.g.: curvature

$$F(f) = \kappa_f(\tau) = \frac{\|f'(\tau) \times f''(\tau)\|}{\|f'(\tau)\|_2^3}, \tag{1..0.8}$$

and let G_h be an approximation of F, based on data with density h. The quality of G_h can be measured by comparison to F on smooth regular curves f, parametrized by arclength, in the sense of

Definition 1.9 *A functional G_h is an m–th order approximation of F with respect to S and $h \to 0$, if there is a constant c such that for all $f \in S$ there are positive constants $h_0(f)$ and $K(f)$ with*

$$|F(f) - G_h(f)| \leq c \cdot h^m \cdot K(f)$$

for all $h \in (0, h_0(f)]$. □

The goal of this paper is to develop a general method for constructing high–oder approximations for geometric data of smooth and regular curves, e.g.: tangent directions, curvature or torsion values, or derivatives thereof with respect to arclength. Interpolation of curves will be a major application, because there are good high–order methods [1] requiring tangent or curvature data which must be constructed from positional data, if they are not available from other sources.

2. Local Polynomial Interpolants

If arclength values t_i of the data $P_i = f(t_i)$ were known and if the functional F had the form $F(f) = f^{(j)}(t^*)$, then vector–valued polynomial interpolation would be a very convenient tool to construct approximations of F. As a variation of this idea, one can consider polynomial interpolation at approximations s_i of the actual arclengths t_i. This requires a straightforward variation of the standard error estimate (1.2) for polynomial interpolation:

Lemma 2.10 *Let f be a C^{n+k} curve on $[a,b] \subset I\!R$, $1 \le k \le n$, parametrized by arclength. Consider n–th degree polynomial interpolation to given data $P_i = f(t_i)$, $0 \le i \le n$, at perturbed parameter values s_i, $0 \le i \le n$, satisfying*

$$\varphi(s_i) = t_i, \qquad 0 \le i \le n, \qquad c \le s_0 < s_1 < \ldots < s_n \le d \tag{2..0.11}$$

for a strictly monotonic reparametrization function

$$\varphi : [c,d] \to [a,b], \quad \varphi \in C^{n+k}[c,d].$$

Then the interpolant p to $P_i = f(t_i)$ at s_i has the classical error representation (1.1) *in the form*

$$(f \circ \varphi)(s) - p(s) = \left(\prod_{i=0}^{n} (s - s_i) \right) \Delta^{n+1}(s_0, \ldots, s_n, s)(f \circ \varphi) \tag{2..0.12}$$

and the derivatives satisfy

$$\|(f \circ \varphi)^{(j)}(s) - p^{(j)}(s)\| \le h_s^{n+1-j} \cdot K(n,k,f \circ \varphi)$$

for all $s \in [s_0, s_n]$, all $j \in \{0, \ldots, k-1\}$, where

$$h_s := \max_{1 \le i \le n} (s_i - s_{i-1}).$$

□

Here, K is dependent on the data distribution, because it contains derivatives of $f \circ \varphi$ up to order $n+k$. Furthermore, (2.12) is still dependent on the reparametrization function φ, and the next two sections will address this drawback.

3. Smoothly refinable parametrizations

We now consider strategies for determining "good" parameter values s_i. If convergence orders of Lemma 2.1 are to be kept as large as possible, the reparametrization functions φ of (2.11) should have derivatives of order up to $n+k$, which can be bounded independently of the density or position of the s_i and t_i, if the mesh width in the sense of (1.4) and (1.3) tends to zero. We then call such a strategy **smoothly refinable** of order $n+k$.

Example 3.13 The most obvious parametrization strategy uses successive chordlengths

$$s_i - s_{i-1} = \|P_i - P_{i-1}\|_2 = \|f(t_i) - f(t_{i-1})\|,$$

and sets $s_0 = t_0 = 0$ without loss of generality. For uniformly distributed data on a circular arc one can easily show that parametrization by successive chordlengths is smoothly refinable of arbitrary order. In general, successive chordlengths can be smoothly refinable only up to order three, as may be shown by taking divided differences of non–uniform samples of circular data. □

This eliminates successive chordlength parametrization as a tool for higher order geometric differentiation.

To overcome the difficulties with successive chordlength parametrization, one can take chordlengths with respect to a **fixed** point P_ℓ, $0 \leq l \leq n$, and define

$$s_i = s_\ell + \|P_i - P_\ell\| \cdot \mathrm{sgn}(i - \ell), \qquad 0 \leq i \leq n.$$

We call this a **locally centered chordlength parametrization** and get

Lemma 3.14 *If $f \in C^m[a,b]$, $m > 2$, is parametrized by arclength, then any locally centered chordlength parametrization is smoothly refinable of order m.*

Proof: Let $P_\ell = f(0)$ be the center of a given chordlength parametrization, and define the real–valued function

$$s(t) := \|f(t) - f(0)\|_2 \mathrm{sgn}(t), \qquad t \in [a,b] \ni 0.$$

Then $s(t_i) = s_i$ holds for $0 \leq i \leq n$, and $s^{-1} = \varphi$ is our candidate for a reparametrization function. Clearly, a simple Taylor expansion implies that

$$s^2(t) = \|f(t) - f(0)\|_2^2 = t^2 + t^4 \cdot q(t)$$

is a smooth function with $q \in C^{m-2}[a,b]$, and $s(t)$ has the form

$$s(t) = t\sqrt{1 + t^2 q(t)},$$

the square root taken to be positive. Around $t = 0$ the function $s(t)$ is in C^m and strictly monotonic. Thus $\varphi = s^{-1}$ exists and shares these properties. □

Applications of locally centered chordlength parametrization should make sure that the mesh width h_s of data P_i is small enough to make the s_i monotonic with respect to i.

Combining Lemmas 2.1 and 3.1 we get approximations of order h^{n+1-j} to j–th derivatives of $f \circ \varphi$ at the data points, whenever $0 \leq j \leq k-1$ and f is a C^{n+k} function, interpolated by n–th degree–polynomials at centered chordlength parameters. The details of the numerical realization are summarized in steps A1–A3 of the algorithm in section 7.

4. Elimination of parametrization

The results of the previous section yield high–order approximations of $(f \circ \varphi)^{(j)}$, but not of f itself. To eliminate the (unknown) reparametrization function φ, we define

$$g(s) = (f \circ \varphi)(s)$$

and use $g'(s) = (f' \circ \varphi)(s) \cdot \varphi'(s)$ to get derivatives

$$\begin{aligned} \varphi'(s) &= \|g'(s)\|_2 \\ \varphi^{(j)}(s) &= \frac{d^{j-1}}{ds^{j-1}} \|g'(s)\|_2, \quad j = 2, 3, \ldots \end{aligned} \tag{4..0.15}$$

of φ from the derivatives of g. Clearly,

$$(f' \circ \varphi)(s) = g'(s)/\|g'(s)\|$$

directly yields f' at $t = \varphi(s)$, and higher derivatives require the solution of equations

$$\begin{aligned} g''(s) &= (f' \circ \varphi)(s)(\varphi'(s))^2 + (f' \circ \varphi)(s)\varphi''(s) \\ g^{(j)}(s) &= (f^{(j)} \circ \varphi)(s)(\varphi'(s))^j + (f' \circ \varphi)(s)\varphi^{(j)}(s) + \text{ lower derivatives} \end{aligned} \tag{4..0.16}$$

for $f^{(j)}$ at $t = \varphi(s)$. If the data are dense enough, and if locally centered chordlength parametrization is used, there will be no problems with (4.15) because $\varphi' \approx 1$ for small h. The sample of points P_i can be rejected as being not dense enough, if the numerical test $\alpha^{-1} \geq \|g'\| = \|\varphi'\| \geq \alpha > 0$ for some $\alpha \in (0, 1)$ is not satisfied. Details are given in steps A4 and A5 of the algorithm in section 7.

If j–th derivatives of g contain an error of order h^{n+1-j}, some elementary calculations prove that j–th derivatives of φ and f, the latter taken at unknown arclength values $t = \varphi(s)$, also have errors of order $\mathcal{O}(h^{n+1-j})$, provided that the data are dense enough. The resulting derivatives of f at arclength parameters are now (asymptotically) independent of the parametrization chosen to supply the intermediate derivative of $f \circ \varphi$. Of course, this strategy for this elimination of parametrization effects will work in general, not just for the centered chordlength parametrization of the previous section.

Another approach to eliminate parametrization effects is to calculate curvature, torsion (or derivatives thereof with respect to arclength) directly from the derivatives of the curve $g = f \circ \varphi$. Since these results do not depend on parametrization, the contribution of φ is asymptotically eliminated.

In both cases there may be numerical problems due to cancellation effects and roundoff, if h is still large and high-order derivatives are calculated.

5. Application to curve interpolation

The previous section provided $\mathcal{O}(h^{n+1-j})$ approximations to j–th derivatives of curves $f \in C^{n+k}$, $1 \le j \le k-1$, parametrized by arclength. These can be put into existing Hermite interpolation schemes to generate piecewise interpolating curves. This is put on a rigorous basis by

Theorem 5.17 *Let $f \in C^{n+k}[0,h_0]$ with $h_0 > 0$ and $1 \le k \le n$ be given, and let p_h be the two–point Hermite interpolation polynomial of degree $\le 2k-1$ to data*

$$\begin{aligned} y_{0,j} &= f^{(j)}(0) + \eta_{0,j}, \quad |\eta_{0,j}| \le ch^{n+1-j},\ 0 \le j \le k-1 \\ y_{h,j} &= f^{(j)}(h) + \eta_{h,j}, \quad |\eta_{n,j}| \le ch^{n+1-j},\ 0 \le j \le k-1 \end{aligned} \tag{5..0.18}$$

on $[0,h]$ for $h \in (0,h_0]$. Then there exists a constant C, independent of h, such that the error bound

$$|f(t) - p_h(t)| \le C \cdot h^{\min(2k,n+1)}$$

holds for all $t \in [0,h]$.

Proof: Let q_h be the Hermite interpolation polynomial to exact data of f. Then the error has the classical representation

$$f(t) - q_h(t) = t^k (h-t)^k \Delta^{2k}(k\#0, k\#h, t) f$$

involving the $(2k)$–th generalized divided difference of f with repeated arguments, i.e.: the notation $k\#x$ means k repetitions of x as an argument. Since f is in C^{2k}, we get

$$|f(t) - q_h(t)| \le C_1 h^{2k}, \quad 0 \le t \le h,$$

where $C_1 = \dfrac{1}{(2k)!} \|f^{(2k)}\|_{\infty,[0,h_0]}$.

Both p_h and q_h can be represented via divided differences, and thus

$$p_h(t) - q_h(t) =$$

$$\sum_{j=0}^{2k-1} t^{m(j)} (h-t)^{j-m(j)} \Delta^j (m(j+1)\#0,\ (j+1-m(j+1))\#h)(p_h - q_h)$$

where $m(j) = \min(j,k)$. Therefore it suffices to prove

$$|\Delta^j (m(j+1)\#0, (j+1-m(j+1))\#h)(p_h - q_h)| \le C_2 h^{n+1-j} \tag{5..0.19}$$

inductively for $j = 0, 1, \ldots, 2k-1$. But these divided differences can be evaluated on the η values of (5.18), and all j–th divided differences with fully coalescing arguments are of order $\mathcal{O}(h^{n+1-j})$ by definition, where $0 \le j \le k-1$. If divided differences are formed via the usual recursive relation, the order drops by one. This proves (5.19). □

Theorem 5.17 has quite a number of applications. First consider the de–Boor–Höllig–Sabin method [1] for piecewise GC^2 interpolation of planar data by cubics. According to the first part of this paper, local interpolation by quintic polynomials at centered chordlengths will suffice to generate $\mathcal{O}(h^{6-j})$ estimates of j–th derivatives for $j = 1, 2$ to produce full order $\mathcal{O}(h^6)$ of the interpolation process. This follows from the proof technique in [1] and from Theorem 5.17 for $k = 3$, $n = 5$. Note that we require $f \in C^8$ for this method.

Piecewise GC^1 Hermite interpolation of planar data by quadratics was studied in [8], together with a direct method of determining tangent directions with accuracy $\mathcal{O}(h^3)$. The approach of the previous sections can also be used to supply such derivative estimates, using local cubic interpolants on centered chordlength parameters (case $k = 2$, $n = 3$ of Theorem 5.17).

For data in $\mathbb{R}^d$ with arbitrary d one can use the "$h/3$–rule" to determine a GC^1 piecewise cubic interpolant of accuracy $\mathcal{O}(h^4)$ from two positions P_0, P_1 and normalized tangent directions r_0, r_1 in P_0, P_1 by constructing Bernstein–Bézier control points

$$P_0, P_0 + \frac{h}{3}\, r_0, P_1 - \frac{h}{3}\, r_1, P_1$$

where $h = \|P_0 - P_1\|$ is the local chordlength. This method can be applied to purely positional data without loss of accuracy, if approximations of first derivatives of order $\mathcal{O}(h^3)$ are provided via first derivatives of local third–degree interpolants at centered chordlength parameters. This application will also be covered by Theorem 5.17 for $k = 2$, $n = 3$ together with the basic proof technique of [1]. Note that all of the simpler strategies for tangent estimation will not produce full fourth–order accuracy. For instance, the method of McConalogue [2] uses three–point estimates of tangents via quadratic interpolation, giving $\mathcal{O}(h^2)$ accuracy of tangent directions and an overall $\mathcal{O}(h^3)$ error of interpolation, as Theorem 5.17 shows for $k = n = 2$.

6. Two–point Hermite interpolation

In case of prescribed parameter values t_i one can always generate a piecewise polynomial C^{k-1} interpolant from Lagrange data by the following purely local algorithm:

1. performing local interpolation on sets of $2k$ consecutive data points by polynomials of degree at most $2k - 1$;
2. taking derivatives of order $j = 0, 1, \ldots, k - 1$ of these local interpolants,
3. solving a symmetric two–point Hermite interpolation problem for polynomials of degree $\leq 2k - 1$ on each pair of consecutive data points, using the derivatives of the previous step.

If the data $P_i = f(t_i)$ are sampled from a C^{2k} function f, this process will have accuracy $\mathcal{O}(h^{2k})$, as follows from Lemma 2.1 and Theorem 5.17.

We now proceed to generalize this method to the parametric case. The data now consist of a large sample of points $P_i = f(t_i)$, $0 \leq i \leq N$ of a smooth and regular curve f with values in $\mathbb{R}^d$, parametrized by arclength, but we do not know the arclength values t_i. We can assume that the methods of the previous sections have been applied to yield approximative derivatives of f with respect to arclength up to order $k-1$ at the P_i. For this we make the implicit assumption that N is large enough with respect to k.

We now pick a pair $P_\ell, P_{\ell+1}$ of consecutive data points and want to apply polynomial Hermite interpolation of degree $\le 2k-1$ between $P_\ell = f(t_\ell)$ and $P_{\ell+1} = f(t_{\ell+1})$. This makes it necessary to introduce some parametrization again, because the exact arclength $\tau_\ell = t_{\ell+1} - t_\ell$ is not known. Furthermore, the interpolation should produce an overall GC^{k-1} curve when several patches are joined together. This will require C^{k-1} continuity after a suitable reparametrization.

We simply use chordlength $\sigma_\ell := \|P_{\ell+1} - P_\ell\|$ to parametrize the interpolant locally over $[0, \sigma_\ell]$, using the reparametrization function φ_ℓ of locally centered chordlength at P_ℓ, i.e.

$$s_i = s_\ell + \|P_i - P_\ell\| \cdot \operatorname{sgn}(i-\ell),\ t_i = \varphi_\ell(s_i).$$

This will satisfy

$$\varphi_\ell(0) = t_\ell, \qquad \varphi_\ell(\sigma_\ell) = t_{\ell+1}$$

if we set $s_\ell = 0$ without loss of generality. Thus, we apply nonparametric two–point Hermite interpolation to $f \circ \varphi_\ell$.

Data at the left endpoint $s_\ell = 0$ should then be

$$\begin{aligned} f_\ell^{(0)} &= P_\ell \\ (f \circ \varphi_\ell)^{(j)}(s_\ell),&\ \ 1 \le j \le k-1, \end{aligned} \tag{6..0.20}$$

while at the right endpoint $s_{\ell+1} = \sigma_\ell$ we should interpolate the values

$$\begin{aligned} f_{\ell+1}^{(0)} &= P_{\ell+1},\ \ 0 \le j \le k-1 \\ (f \circ \varphi_\ell)^{(j)}&(s_{\ell+1})_0 \end{aligned} \tag{6..0.21}$$

If f and φ_ℓ were known, including all derivatives of order $\le k-1$, this approach would work easily. In fact, each interpolant would be a piecewise Hermite interpolant to k derivatives of f with respect to arclength after elimination of the local parametrizations, which are uniformly bounded together with their derivatives. The overall accuracy would still be $\mathcal{O}(h^{2k})$, as follows from the proof technique of de Bo0r, Höllig, and Sabin [1].

To make the method feasible in practice, we have to take a closer look at the numerical process which replaces $(f \circ \varphi_\ell)^{(j)}$ by accessible values. Let p_ℓ be the polynomial used for local Lagrange interpolation of degree $\le 2k-1$ with centered chordlengths around P_ℓ, including $P_{\ell+1}$. This will interpolate $f \circ \varphi_\ell$ at centered chordlength abscissae such that

$$p_\ell^{(j)}(s) = (f \circ \varphi_\ell)^{(j)}(s)$$

holds exactly for $j = 0$ and certain chordlength values s_i including s_ℓ and $s_{\ell+1}$, while an error of order $2k - j$ occurs for $j > 0$ or arbitrary arguments near s_ℓ. The method of section 4 is then applied to get approximate values

$$\begin{array}{rcl} f_\ell^{(j)} & := & f^{(j)}(t_\ell) \quad + \quad \mathcal{O}(h^{2k-j}), \quad 0 \le j \le k-1 \\ \varphi_{\ell,i}^{(j)} & := & \varphi_\ell^{(j)}(s_i) \quad + \quad \mathcal{O}(h^{2k-j}) \quad 1 \le j \le k-1 \end{array} \tag{6..0.22}$$

for $i = \ell, \ell+1$. In (6.20) we can use the data $p_\ell^{(j)}(s_\ell)$ directly, but then in (6.21) we run into a problem caused by

$$(f \circ \varphi_\ell)^{(j)}(s_{\ell+1}) \neq p_{\ell+1}^{(j)}(s_{\ell+1}),$$

as used in the next segment. Note that the discrepancy in the above formula is only of order $\mathcal{O}(h^{2k-j})$, but we cannot ignore it without losing GC^{k-1} continuity.

We avoid this difficulty by modifying the actual interpolation data at $s_{\ell+1}$ in a suitable way: if we eliminate φ_ℓ from the data at $s_{\ell+1}$, we should arrive at $f_{\ell+1}^{(j)}$, the approximate chordlength derivative of f at $s_{\ell+1}$ in the next segment. This is accomplished by using (4.16) backwards, starting from the derivatives

$$\begin{array}{rcl} f_{\ell+1}^{(j)} & = & f^{(j)}(t_{\ell+1}) \quad + \quad \mathcal{O}(h^{2k-j}), \quad 0 \le j \le k-1 \\ \varphi_{\ell,\ell+1}^{(j)} & = & \varphi_\ell^{(j)}(s_{\ell+1}) \quad + \quad \mathcal{O}(h^{2k-j}), \quad 1 \le j \le k-1 \end{array}$$

and synthesizing an $\mathcal{O}(h^{2k-j})$ approximation $\hat{f}_{\ell+1}^{(j)}$ of $(f \circ \varphi_\ell)^{(j)}(s_{\ell+1})$ by evaluating the chain rule for (4.16).

We still have to prove that this process maintains overall GC^{k-1} continuity and $\mathcal{O}(h^{2k})$ accuracy. The latter fact is clear because our modifications do not spoil the accuracy, being of order $\mathcal{O}(h^{2k-j})$ when referring to j–th derivatives. To prove CC^{k-1} continuity we introduce the reparametrization function ψ_ℓ which is a Hermite interpolant of the numerically obtained derivative values $\varphi_{\ell,i}^{(j)}$ of (6.22) together with $\varphi_{\ell,i}^{(0)} := \varphi_\ell(s_i)$ for $i = \ell, \ell+1$. Then $\varphi_\ell - \psi_\ell = \mathcal{O}(h^{2k})$ holds between s_ℓ and $s_{\ell+1}$. because the derivatives of φ_ℓ are uniformly bounded. If h is small enough, ψ_ℓ we be strictly monotonic. If q_ℓ is the Hermite interpolant to the data $p_\ell^{(j)}(s_\ell)$ and $\hat{f}_{\ell+1}^{(j)}$, $0 \le j \le k-1$ at $s = s_\ell$ and $s = s_{\ell+1}$, respectively, our construction guarantees that $q_\ell \circ \psi_\ell^{-1}$ interpolates the data $f_\ell^{(j)}$ and $f_{\ell+1}^{(j)}$, $0 \le j \le k-1$ at $t = t_\ell$ and $t = t_{\ell+1}$ respectively, because we used the exact chain rule for derivative values of q_ℓ and ψ_ℓ. But this means that the functions q_ℓ form a GC^{k-1} curve, since after reparametrization they coincide of order $k-1$ at the breakpoints. We summarize:

Theorem 6.23 *Let f be a regular C^{2k} curve, parametrized by arclength. The the above process, given in algorithmic form in the next section, provides a piecewise polynomial GC^{k-1} interpolant of accuracy $\mathcal{O}(h^{2k})$.* □

Of course, the above approach is biased towards the left endpoint P_ℓ, because we used the local chordlength parametrization φ_ℓ centered at P_ℓ. A similar interpolation can be done on

$[-\sigma_\ell, 0]$, using local chordlengths centered at $P_{\ell+1}$, and we found it practically useful to take means of the two solutions in order to maintain symmetry and to avoid instabilities due to the calculations based on (4.15) and (4.16). Furthermore, the local estimation of derivatives via interpolation at centered chordlengths should be of order $\leq 2k$ instead of $2k-1$ to get symmetry-preserving formulae based on an odd number of points. Both modifications do not affect our theoretical results, and they are incorporated into the algorithmic formulation of the method in the next section.

7. Algorithm

For quick reference and easier programming, we summarize our method in algorithmic form.

Data: $N \in I\!N,\ d \in I\!N,\ P_0, \ldots, P_N \in I\!R^d$ with $P_i \neq P_{i-1}$ for $1 \leq i \leq N$. It is implicitly assumed that the data are a large sample of points $P_i = f(t_i)$ from the range of a smooth and regular curve with values in $I\!R^d$, parametrized by arclength, for arclength values satisfying $t_{i-1} < t_i,\ 1 \leq i \leq N$.

Parameters: Choose numbers $k \in I\!N_{\geq 0}$ with $2k \leq N$ and $\alpha \in (0,1)$. The final order of accuracy is $\mathcal{O}(h^{2k})$ if $f \in C^{2k}$, and overall GC^{k-1} continuity will be achieved. Large values of α increase the safety of the method, but will at the same time restrict the range of admissible applications.

Step A: Local Interpolation at centered chordlengths.

For all ℓ with $0 \leq \ell \leq N$ do :

A1: Calculate centered chordlength values around P_ℓ:

$$\begin{aligned} \ell^* &:= \max(k, \min(N-k, \ell)) = \left\{ \begin{array}{cc} k & 0 \leq \ell < k \\ \ell & k \leq \ell \leq N-k \\ N-k & N-k < \ell \leq N \end{array} \right\} \\ s_i &:= \|P_i - P_\ell\| \operatorname{sgn}(i-\ell),\ \ \ell^* - k \leq i \leq \ell^* + k. \end{aligned}$$

A2: If the s_i are not strictly monotonic, give up. The data do not form a sufficiently dense sample from a smooth and regular curve. Use a more robust, but less accurate method.

A3: Calculate a local interpolant p_ℓ around P_ℓ:

$$\begin{aligned} p_\ell &:= \text{polynomial interpolant of data } (s_i, P_i), \\ &\quad\ \ \ell^* - k \leq i \leq \ell^* + k, \text{ of degree } \leq 2k \\ N_\ell &:= \left\{ \begin{array}{cc} \{\ell, \ell+1\} & \ell = 0 \\ \{\ell-1, \ell, \ell+1\} & 1 \leq \ell < N \\ \{\ell-1, \ell\} & \ell = N \end{array} \right\} \end{aligned}$$

Evaluate $p_\ell^{(j)}(s_i)$ for all $0 \leq j < k$, all neighbors s_i for $i \in N_\ell$, and store these values.

A4: Use (4.15) to get approximate derivatives of the reparametrization function φ_ℓ via

$$\begin{aligned}
\varphi_{\ell,i}^{(1)} &:= \|p_\ell'(s_i)\|_2 && \text{for } k \geq 2\\
\varphi_{\ell,i}^{(2)} &:= p_\ell'(s_i)^T p_\ell''(s_i)/\varphi_{\ell,i}^{(1)} && \text{for } k \geq 3\\
\varphi_{\ell,i}^{(3)} &:= (p_\ell''(s_i)^T p_\ell''(s_i) + p_\ell'(s_i)^T p_\ell'''(s_i) - (\varphi_{\ell,i}^{(2)})^2)/\varphi_{\ell,i}^{(1)} && \text{for } k \geq 4\\
\varphi_{\ell,i}^{(4)} &:= (3p_\ell''(s_i)^T p_\ell'''(s_i) + p_\ell'(s_i)^T p_\ell^{(4)}(s_i) - 3\varphi_{\ell,i}^{(2)}\varphi_{\ell,i}^{(3)})/\varphi_{\ell,i}^{(1)} && \text{for } k \geq 5\\
&\text{etc.}
\end{aligned}$$

for $i \in N_\ell$ and store these values. After evaluating $\varphi_{\ell,i}^{(1)}$, test for

$$0 < \alpha \leq \varphi_{\ell,i}^{(1)} \leq 1/\alpha$$

and give up, if the test fails. In this case the data are no sufficiently dense sample from a smooth and regular curve. Use a more robust, but less accurate method.

A5: Use (4.16) to eliminate the parametrization effect from derivatives of p_ℓ

$$\begin{aligned}
f_\ell^{(0)} &:= P_\ell && \text{for } k \geq 1\\
f_\ell^{(1)} &:= p_\ell'(s_\ell)/\varphi_{\ell,\ell}^{(1)} && \text{for } k \geq 2\\
f_\ell^{(2)} &:= (p_\ell''(s_\ell) - \varphi_{\ell,\ell}^{(2)} f_\ell^{(1)})/(\varphi_{\ell,\ell}^{(1)})^2 && \text{for } k \geq 3\\
f_\ell^{(3)} &:= (p_\ell'''(s_\ell) - \varphi_{\ell,\ell}^{(3)} f_\ell^{(1)} - 3\varphi_{\ell,\ell}^{(1)}\varphi_{\ell,\ell}^{(2)} f_\ell^{(2)})/(\varphi_{\ell,\ell}^{(1)})^3 && \text{for } k \geq 4\\
f_\ell^{(4)} &:= (p_\ell^{(4)}(s_\ell) - \varphi_{\ell,\ell}^{(4)} f_\ell^{(1)} - 4\varphi_{\ell,\ell}^{(1)}\varphi_{\ell,\ell}^{(3)} f_\ell^{(2)}\\
&\quad -3(\varphi_{\ell,\ell}^{(2)})^2 f_\ell^{(2)} - 6(\varphi_{\ell,\ell}^{(1)})^2\varphi_{\ell,\ell}^{(2)} f_\ell^{(3)})/(\varphi_{\ell,\ell}^{(1)})^4 && \text{for } k \geq 5\\
&\text{etc.}
\end{aligned}$$

and store these values, which will be the "geometric" Hermite data for the next step.

Step B: Local two-point Hermite interpolation.

For all ℓ with $0 \leq \ell \leq N-1$ do :

B1: Prepare data for interpolation between P_ℓ and $P_{\ell+1}$, using (4.16) and the reparametrization function φ_ℓ:

$$\begin{aligned}
\sigma_\ell &:= \|P_\ell - P_{\ell+1}\|_2 \\
\widehat{f}^{(0)}_{\ell+1} &:= f^{(0)}_{\ell+1} \\
\widehat{f}^{(1)}_{\ell+1} &:= \varphi^{(1)}_{\ell,\ell+1} f^{(1)}_{\ell+1} \\
\widehat{f}^{(2)}_{\ell+1} &:= \varphi^{(2)}_{\ell,\ell+1} f^{(1)}_{\ell+1} + (\varphi^{(1)}_{\ell,\ell+1})^2 f^{(2)}_{\ell+1} \\
\widehat{f}^{(3)}_{\ell+1} &:= \varphi^{(3)}_{\ell,\ell+1} f^{(1)}_{\ell+1} + 3\varphi^{(1)}_{\ell,\ell+1}\varphi^{(2)}_{\ell,\ell+1}(\sigma_\ell) f^{(2)}_{\ell+1} + (\varphi^{(1)}_{\ell,\ell+1})^3 f^{(3)}_{\ell+1} \\
\widehat{f}^{(4)}_{\ell+1} &:= \varphi^{(4)}_{\ell,\ell+1} f^{(1)}_{\ell+1} + 3(\varphi^{(2)}_{\ell,\ell+1})^2(\sigma_\ell) f^{(2)}_{\ell+1} + 4\varphi^{(1)}_{\ell,\ell+1}\varphi^{(3)}_{\ell,\ell+1} f^{(2)}_{\ell+1} \\
&\quad +5(\varphi^{(1)}_{\ell,\ell+1})^2\varphi^{(2)}_{\ell,\ell+1} f^{(3)}_{\ell+1} + (\varphi^{(1)}_{\ell,\ell+1})^4 f^{(4)}_{\ell+1} \\
\text{etc.} &
\end{aligned}$$

for the appropriate values of k.

B2: Let $\widehat{q}_\ell$ be the nonparametric Hermite interpolant of degree $\leq 2k-1$ of data $p^{(j)}_\ell(s_\ell)$ at 0 and of $\widehat{f}^{(j)}_{\ell+1}$ at σ_ℓ, where $0 \leq j < k$. Store $\widehat{q}_\ell$ in some form or other.

B3: Prepare data for interpolation between P_ℓ and $P_{\ell+1}$, now using the reparametrization function $\varphi_{\ell+1}$:

$$\begin{aligned}
\widetilde{f}^{(0)}_{\ell} &:= f^{(0)}_{\ell} \\
\widetilde{f}^{(1)}_{\ell} &:= \varphi^{(1)}_{\ell+1,\ell} f^{(1)}_{\ell} \\
\widetilde{f}^{(2)}_{\ell} &:= \varphi^{(2)}_{\ell+1,\ell} f^{(1)}_{\ell} + (\varphi^{(1)}_{\ell+1,\ell})^2 f^{(2)}_{\ell} \\
\widetilde{f}^{(3)}_{\ell} &:= \varphi^{(3)}_{\ell+1,\ell} f^{(1)}_{\ell} + 3\varphi^{(1)}_{\ell+1,\ell}\varphi^{(2)}_{\ell+1,\ell} f^{(2)}_{\ell} + (\varphi^{(1)}_{\ell+1,\ell})^3 f^{(3)}_{\ell} \\
\widetilde{f}^{(4)}_{\ell} &:= \varphi^{(4)}_{\ell+1,\ell} f^{(1)}_{\ell} + 3(\varphi^{(2)}_{\ell+1,\ell})^2 f^{(2)}_{\ell} + 4\varphi^{(1)}_{\ell+1,\ell}\varphi^{(3)}_{\ell+1,\ell} f^{(2)}_{\ell} \\
&\quad +5(\varphi^{(1)}_{\ell+1,\ell})^2\varphi^{(2)}_{\ell+1,\ell} f^{(3)}_{\ell} + (\varphi^{(1)}_{\ell+1,\ell})^4 f^{(4)}_{\ell} \\
\text{etc.} &
\end{aligned}$$

for the appropriate values of k.

B4: Let $\widetilde{q}_\ell$ be the nonparametric Hermite interpolant of degree $\leq 2k-1$ of data $\widetilde{f}^{(j)}_\ell$ at $-\sigma_\ell$ and of $p^{(j)}_{\ell+1}(\sigma_{\ell+1})$ at 0, where $0 \leq j < k$. Store $\widetilde{q}_\ell$ in some form or other.

B5: For evaluation of the solution between P_ℓ and $P_{\ell+1}$, use the polynomial

$$q(s) := \frac{1}{2}\left(\widehat{q}_\ell(s) + \widetilde{q}_\ell(s-\sigma_\ell)\right),\ s \in [0, \sigma_\ell],$$

or a similar weighted mean between $\widehat{q}$ and $\widetilde{q}$.

Remarks

There can be m additional data points between the endpoints P_ℓ, $P_{\ell+1+m}$ of a local Hermite interpolation. The degree must then be $2k-1+m$ and the approximation order will be

$\mathcal{O}(h^{2k+m})$, provided that geometric differentiation of data from $f \in C^{2k+m}$ is done with local polynomials of degree at least $2k-1+m$.

At the end of the range of points, e.g. on $[P_0, P_1]$ or $[P_{N-1}, P_N]$, one does not need derivative values, because no GC^{k-1} continuity must be guaranteed. Then a non–symmetric Hermite interpolation problem with chordlength parameters locally centered at P_1 or P_{N-1} is sufficient, giving the same order of accuracy. This improvement was not incorporated into the examples of the last section.

An alternative approach would avoid the reparametrization of the $f_\ell^{(j)}$ values by direct interpolation on $[0, t^*]$ for a high–accuracy estimate t^* of the actual arclength $t_{\ell+1} - t_\ell$. However, it then seems to be difficult to reach $\mathcal{O}(h^{2k})$ accuracy at comparable costs.

We note that our method is not convexity preserving in general, and it may produce bad results for coarse or noisy data sets. It is designed for sufficiently dense samples of exact data from smooth curves, and its major feature is its convergence order, which may be arbitrarily high for sufficiently smooth curves. So far, the convexity preserving method of highest approximation order is the rational GC^2 interpolation of [6][7], being of order four.

8. Examples

We start with an equidistant sample of 17 points on a semi–circle. Figure 1 will show the $\mathcal{O}(h^4)$ solution with GC^1 continuity, obtained by our method for $k = 2$, together with a plot of curvature. The curvature discontinuities are not visible, but the actual curvature is not constant, as can be seen from the curvature derivative in Figure 2. The plots of interpolant and curvature look all the same for higher order solutions, and therefore we only show higher derivatives of curvature in Figures 3 and 4. The effect of the boundary is quite substantial, because there we have to evaluate the local high-order interpolant p_ℓ of steps A3-A5 near the endpoints of its range.

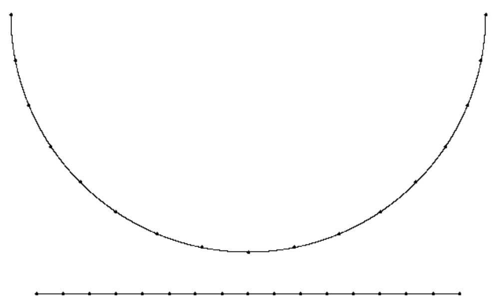

Figure 1: 17 points on a semi–circle, $k = 2$, with curvature plot

Figure 5 shows polygonal interpolation of 33 points sampled from part of a spiral in $\mathbb{R}^3$, seen from above and from the side. The spiral takes a 270 degree turn and has monotonic

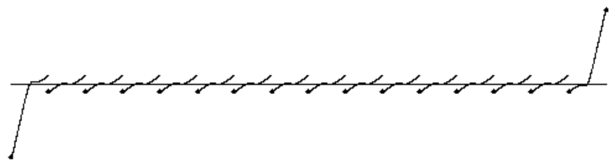

Figure 2: Same data, $k = 2$, first derivative of curvature

Figure 3: Same data, $k = 4$, first derivative of curvature

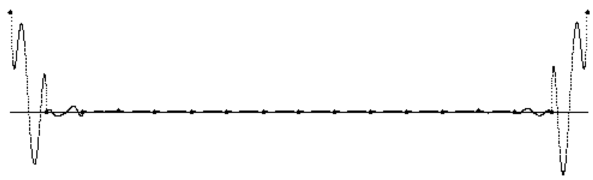

Figure 4: Same data, $k = 5$, second derivative of curvature

and nonzero curvature and torsion. Reproduction of the curve shape was perfect within plot precision in all cases. Thus we plot some of the instances where problems like discontinuities or peaks for higher derivatives of curvature and torsion occur.

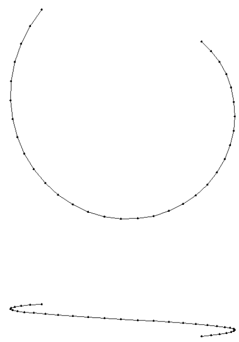

Figure 5: 33 points on spiral in $\mathbb{R}^3$, $k = 1$, polygonal interpolant

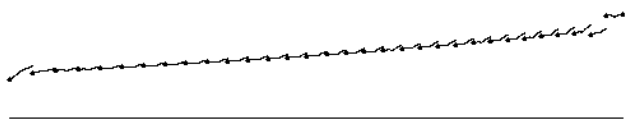

Figure 6: Same data, $k = 2$, first derivative of curvature

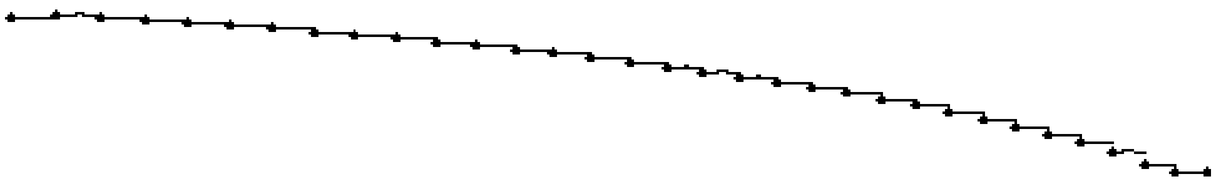

Figure 7: Same data, $k = 2$, torsion

One can clearly see that the degree of the local interpolants is too high; there are "unnecessary" wiggles in the curves of Figures 10 and 11, which still have to be continuous by construction.

Figure 12 shows polygonal interpolation of 71 points sampled from part of a Lissajous type figure in $\mathbb{R}^3$, seen from above and from the side. There are two peaks of curvature and torsion which should be reproduced by the interpolant. Again, we found that reproduction of the curve shape was perfect within plot precision in all cases, and thus we plot only some

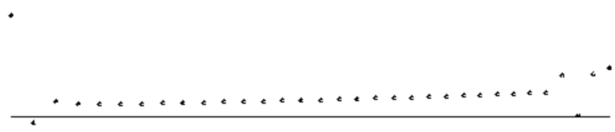

Figure 8: Same data, $k = 3$, second derivative of curvature

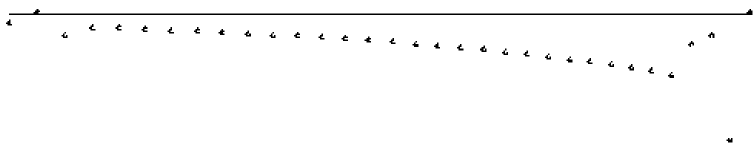

Figure 9: Same data, $k = 3$, first derivative of torsion

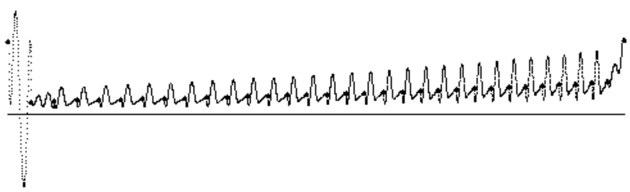

Figure 10: Same data, $k = 5$, second derivative of curvature

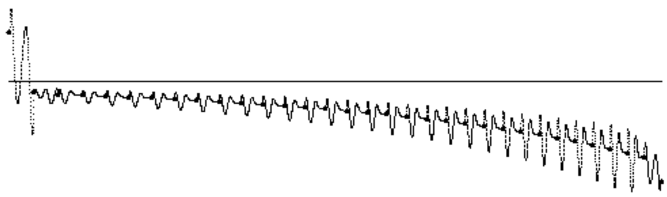

Figure 11: Same data, $k = 5$, first derivative of torsion

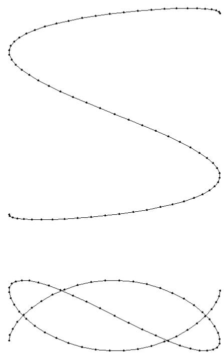

Figure 12: 71 points on space curve, polygonal interpolant

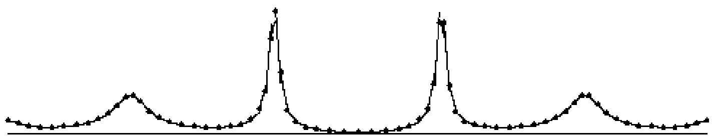

Figure 13: Same data, $k = 2$, GC^1, discontinuous curvature

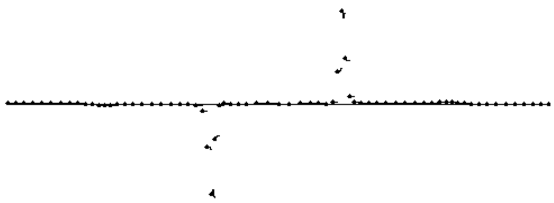

Figure 14: Same data, $k = 2$, GC^1, discontinuous torsion

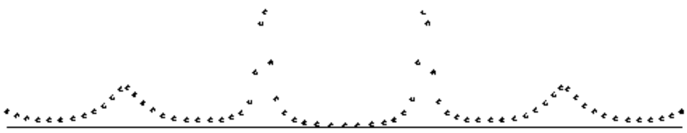

Figure 15: Same data, $k = 3$, GC^2, curvature

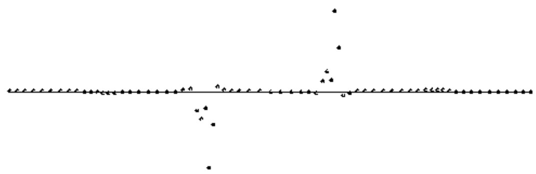

Figure 16: Same data, $k = 3$, GC^2, discontinuous torsion

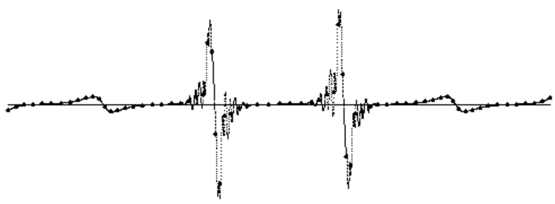

Figure 17: Same data, $k = 4$, GC^3, first derivative of curvature

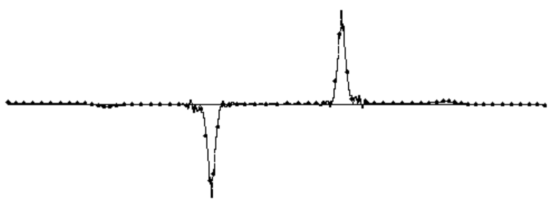

Figure 18: Same data, $k = 4$, GC^3, torsion

curvature and torsion data. Here, the relative heights of the peaks of curvature and torsion are about 9.5 and 30.0.

Further research should try to achieve high orders of accuracy with low polynomial degrees, e.g.: along the lines of the paper [1] by deBoor, Höllig, and Sabin, or by rational interpolation.

References

[1] deBoor, C., Höllig, K., and Sabin, M., High Accuracy Geometric Hermite Interpolation, Computer Aided Geometric Design 4, 269–278, 1987

[2] McConalogue, D.J., A quasi–intrinsic scheme for passing a smooth curve through a discrete set of points, The Computer Journal,13 (1970) 392–396

[3] Micchelli, C. A., Rivlin, Th. J., A Survey of Optimal Recovery, in C. A. Micchelli, Th. J. Rivlin (eds.): Optimal Estimation in Approximation Theory, Plenum Press, New York–London 1977

[4] Sard, A., Linear Approximation, Math. Surveys 9, Amer. Math. Soc., Providence 1963

[5] Sard, A., Optimal approximation, J. Funct. Anal. 1 (1967), 222-244

[6] Schaback, R., Interpolation in $\mathbb{R}^2$ by piecewise quadratic visually C^2 Bezier polynomials, Computer Aided Geometric Design 6 (1989) 219–233

[7] Schaback, R., On Global GC^2 Convexity Preserving Interpolation of Planar Curves by Piecewise Bezier Polynomials, in T. Lyche and L.L. Schumaker (eds.): "Mathematical Methods in Computer Aided Geometric Design", Academic Press 1989, 539–547

[8] Schaback, R., Convergence of Planar Curve Interpolation Schemes, in C.K. Chui, L.L. Schumaker and J.D. Ward (eds): "Approximation Theory VI", 1989, 581-584