Extrapolation Methods for Approximating Arc Length and Surface Area

Michael S. Floater[*], Atgeirr F. Rasmussen[†] and Ulrich Reif[‡]

November 15, 2006

Abstract

A well-known method of estimating the length of a parametric curve in $\mathbb{R}^d$ is to sample some points from it and compute the length of the polygon passing through them. In this paper we show that for uniform sampling of regular smooth curves Richardson extrapolation can be applied repeatedly giving a sequence of derivative-free length estimates of arbitrarily high orders of accuracy. Further, a similar result is derived for the approximation of the area of parametric surfaces.

Math Subject Classification: Primary: 65D30, 65B05, Secondary: 41A58, 65D17.
Keywords: Richardson extrapolation, arc length, surface area

1 Introduction

Computing the arc length of a parametric curve is a standard task in applied geometry, and many papers have been written about it or related issues, including [12], [17], [16], [11], [13], [19], [18], [2], [5], [3], [8], [9] and [10]. The purpose of this paper is to analyze the accumulated chord-length approximation to arc length, and show that it has an asymptotic h^2-expansion making it suitable for Richardson extrapolation.

[*]Centre of Mathematics for Applications, Department of Informatics, University of Oslo, PO Box 1053, Blindern, 0316 Oslo, Norway, *email: michaelf@ifi.uio.no*

[†]Centre of Mathematics for Applications, Department of Informatics, University of Oslo, PO Box 1053, Blindern, 0316 Oslo, Norway, *email: atgeirr@math.uio.no*

[‡]Department of Mathematics, Darmstadt University of Technology, Schlossgartenstraße 7, 64289 Darmstadt, Germany, *email: reif@mathematik.tu-darmstadt.de*

Let $\mathbf{f} : [0,1] \to \mathbb{R}^d$, $d \geq 2$, be a regular parametric curve, by which we mean that $\mathbf{f}$ is differentiable and $\mathbf{f}'(t) \neq 0$ for all $t \in [0,1]$. Its length is given by the integral formula

$$L(\mathbf{f}) := \int_0^1 |\mathbf{f}'(t)|\, dt,$$

where $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^d$. It is typically difficult or even impossible to find a close-form solution to this integral, and one has to resort to numerical approximation. For example, polynomial curves of degree ≥ 2 are never arc-length parametrized according to [7] and their speed functions are only easily integrated if they are PH-curves, see [6]. One way of approximating $L(\mathbf{f})$ is to apply one of the many known quadrature methods to the scalar function $q(t) := |\mathbf{f}'(t)|$. If $\mathbf{f}$ is smooth enough, one can obtain arbitrarily high rates of approximation this way. For an analysis of this, see [10].

However, approximating the integral of q requires the computation of (first) derivatives of $\mathbf{f}$ which can be awkward or time-consuming in practice. A popular alternative which avoids this problem is to use the 'chordal' method: we choose some uniform partition $0 = t_0 < t_1 < \cdots < t_n = 1$, where $t_i = ih$ and $h = 1/n$, and sum up the lengths of the chords (line segments) with end points $\mathbf{f}(t_i), \mathbf{f}(t_{i+1})$, giving the approximation

$$L_h(\mathbf{f}) := \sum_{i=0}^{n-1} |\mathbf{f}(t_{i+1}) - \mathbf{f}(t_i)| \approx L(\mathbf{f}), \tag{1}$$

which is simply the length of the polygon passing through the sample points $\mathbf{f}(t_i)$. It can be shown that

$$L(\mathbf{f}) - L_h(\mathbf{f}) = O(h^2)$$

as $h \to 0$, provided $\mathbf{f} \in C^2[0,1]$. Thus the method has second order accuracy. A proof of this and generalizations can be found in [10].

In this paper we show that we can apply a standard extrapolation technique, often known as Richardson extrapolation, to the chord length method on equally spaced points in order to raise the order of accuracy first to $O(h^4)$, then to $O(h^6)$ and so on, provided $\mathbf{f}$ has enough bounded derivatives. For example, consider the first extrapolation. For small h, we would expect the finer approximation $L_{h/2}(\mathbf{f})$ to be more accurate by a factor of ≈ 4, but in order to increase the *order* of approximation, we take a linear combination of it and the coarser approximation. It turns out that if we weight the two approximations by the coefficients $4/3$ and $-1/3$, giving

$$L(\mathbf{f}) \approx \frac{4}{3} L_{h/2}(\mathbf{f}) - \frac{1}{3} L_h(\mathbf{f}), \tag{2}$$

the order of approximation is raised to $O(h^4)$. In fact it was shown in [10] that the error of the method (2) is $O(h^4)$ provided $\mathbf{f} \in C^4[0,1]$. This was done by recognizing this method as the result of using quadratic polynomial interpolation and a certain open Newton-Cotes quadrature rule.

In the next section, we derive more information about the error in the fourth order rule (2) and show that for smooth $\mathbf{f}$ Richardson extrapolation can be applied indefinitely, giving methods with accuracy $O(h^6)$, $O(h^8)$ and so on. This method is an alternative to the general approach suggested in [10]. In Section 3, we formulate an algorithm and assess its performance. Then, in Section 4, we apply the idea of extrapolation to approximate surface area, where the basic approximation scheme uses cross products of diagonals of quadrilateral facets obtained by uniform sampling.

2 Approximation of Arc Length

Validating Richardson extrapolation according to (2) for the chord length rule (1) is simply a matter of showing that its error $L(\mathbf{f})-L_h(\mathbf{f})$ can be expanded in powers of h, where the coefficients are *independent* of h. We will show in fact that the expansion contains only even powers of h. Moreover the coefficient of the h^2 term can be expressed in terms of the curvature vector

$$\boldsymbol{\kappa} := \frac{1}{|\mathbf{f}'|}\left(\frac{\mathbf{f}'}{|\mathbf{f}'|}\right)', \tag{3}$$

which is the derivative of the normalized tangent vector with respect to arc length. A key part in proving this is the following expansion for the length of the vector

$$\mathbf{d}_h(t) := \mathbf{f}(t+h/2) - \mathbf{f}(t-h/2)$$

which, for given $h \in (0, 1/2)$, is regarded as a function of $t \in [h/2, 1-h/2]$.

Lemma 1 *Let $\mathbf{f} : [0,1] \to \mathbb{R}^d$, $d \geq 2$, be a regular parametric curve with $\mathbf{f} \in C^{2k+1}[0,1]$. Then there exist functions $c_r \in C^{2k-2r}[0,1]$ such that*

$$|\mathbf{d}_h(t)|/h = \sum_{r=0}^{k} c_r(t)h^{2r} + o(h^{2k}), \quad t \in [h/2, 1-h/2]. \tag{4}$$

The first two coefficients are

$$c_0 = |\mathbf{f}'|, \quad c_1 = \frac{\mathbf{f}' \cdot \mathbf{f}'''}{24\,|\mathbf{f}'|}. \tag{5}$$

Throughout, for an expression $g(t,h)$ depending on t and h, the Landau symbol $o(h^\ell)$ refers to *uniform convergence* with respect to t. That is, we write $g(t,h) = o(h^\ell)$ if for all $\varepsilon > 0$, there is some $h_0 > 0$ such that for all $h \in (0, h_0]$ and all $t \in [h/2, 1-h/2]$,

$$|g(t,h)| \leq \varepsilon h^\ell.$$

Proof. By Taylor's theorem,

$$\mathbf{f}(t \pm h/2) = \sum_{j=0}^{2k+1} \left(\frac{\pm h}{2}\right)^j \frac{\mathbf{f}^{(j)}(t)}{j!} + o(h^{2k+1}),$$

and so

$$\mathbf{d}_h/h = \sum_{j=0}^{k} \mathbf{a}_j h^{2j} + o(h^{2k}), \quad \mathbf{a}_j := \frac{\mathbf{f}^{(2j+1)}}{2^{2j}\,(2j+1)!}. \tag{6}$$

The first two $\mathbf{a}_j$ are

$$\mathbf{a}_0 = \mathbf{f}', \quad \mathbf{a}_1 = \frac{1}{24}\,\mathbf{f}'''.$$

Next, taking the inner product of $\mathbf{d}_h$ with itself gives

$$|\mathbf{d}_h|^2/h^2 = \sum_{j=0}^{k} b_j h^{2j} + o(h^{2k}), \quad b_j := \sum_{\ell=0}^{j} \mathbf{a}_\ell \cdot \mathbf{a}_{j-\ell}. \tag{7}$$

The first two b_j are

$$b_0 = \mathbf{a}_0 \cdot \mathbf{a}_0 = |\mathbf{f}'|^2, \quad b_1 = 2\,\mathbf{a}_0 \cdot \mathbf{a}_1 = \frac{1}{12}\,\mathbf{f}' \cdot \mathbf{f}'''.$$

Taking square roots, (7) gives

$$|\mathbf{d}_h|/h = (b_0 + \Delta)^{1/2}, \quad \Delta := \sum_{j=1}^{k} b_j h^{2j} + o(h^{2k})$$

where $\Delta \to 0$ as $h \to 0$. The Taylor expansion of the right hand side with respect to Δ reads

$$(b_0 + \Delta)^{1/2} = \sum_{\ell=0}^{k} \gamma_\ell\, b_0^{1/2-\ell} \Delta^\ell + o(\Delta^k), \quad \gamma_\ell := \frac{1}{\ell!} \prod_{p=0}^{\ell-1} \left(\frac{1}{2} - p\right).$$

Again, the convergence of the remainder term $o(\Delta^k)$ is uniform with respect to t because, by regularity of $\mathbf{f}$, b_0 is bounded from below by a positive constant. Substituting in the definition of Δ and collecting together powers of h^2, we obtain the expansion

$$|\mathbf{d}_h|/h = \sum_{r=0}^{k} c_r h^{2r} + o(h^{2k})$$

with coefficients

$$c_0 := b_0^{1/2} = |\mathbf{f}'|, \quad c_1 := \frac{b_1}{2|\mathbf{f}'|} = \frac{\mathbf{f}' \cdot \mathbf{f}'''}{24\,|\mathbf{f}'|},$$

and in general, for $r \geq 1$,

$$c_r := \sum_{\ell=1}^{r} \gamma_\ell \, |\mathbf{f}'|^{1-2\ell} \sum_{\substack{j_1+\cdots+j_\ell=r \\ j_p \geq 1}} b_{j_1} \cdots b_{j_\ell}. \tag{8}$$

By (6) and (7), we have $b_j \in C^{2k-2j}[0,1]$. Hence, the least smooth term appearing in the definition of c_r is b_r, and thus, $c_r \in C^{2k-2r}[0,1]$. □

Now, we are prepared to derive an expansion of the difference between the chord length approximation $L_h(\mathbf{f})$ according to (1) and the true length $L(\mathbf{f})$ of a curve $\mathbf{f}$.

Theorem 1 *Let $\mathbf{f} : [0,1] \to \mathbb{R}^d$, $d \geq 2$, be a regular parametric curve with $\mathbf{f} \in C^{2k+1}[0,1]$. Then there exist coefficients d_ℓ independent of h such that*

$$L(\mathbf{f}) - L_h(\mathbf{f}) = \sum_{\ell=1}^{k} d_\ell h^{2\ell} + o(h^{2k}). \tag{9}$$

In particular,

$$d_1 = \frac{1}{24} \int_0^1 |\boldsymbol{\kappa}(t)|^2 |\mathbf{f}'(t)|^3 \, dt. \tag{10}$$

In [10], results closely resembling the theorem above were shown for $k = 1$ and $k = 2$, requiring only $\mathbf{f} \in C^2$ and $\mathbf{f} \in C^4$, respectively. We conjecture that in Theorem 1, the smoothness requirement may be relaxed for all k to $\mathbf{f} \in C^{2k}$. This conjecture is supported by the following observation: One can show that the coefficients d_ℓ depend only on derivatives of $\mathbf{f}$ up to order 2ℓ. This is remarkable since the corresponding coefficients c_ℓ appearing in the expansion for a single segment involve also derivatives of order $2\ell + 1$. Hence, the sum $\sum_{\ell=1}^{k} d_\ell h^{2\ell}$ in (9) is well defined for $\mathbf{f} \in C^{2k}$.

Proof. We apply (4) with $t = u_i := t_i + h/2$ and sum over i to obtain

$$L_h(\mathbf{f}) = \sum_{i=0}^{n-1} |\mathbf{d}(u_i)| = \sum_{j=0}^{k} h^{2j+1} \sum_{i=0}^{n-1} c_j(u_i) + o(h^{2k}). \tag{11}$$

where, as above, $n = 1/h$. In order to convert the terms in this expansion into integrals, let us briefly review the second Euler-Maclaurin expansion concerning quadrature by the midpoint rule. We start from the more familiar first Euler-Maclaurin expansion concerning quadrature by the trapezoidal rule. From Theorem 2.2 of [1], p. 64, it can be shown that if $\phi \in C^{2m}[0,1]$, then

$$h\left(\frac{1}{2}\phi(0) + \sum_{i=1}^{n-1} \phi(t_i) + \frac{1}{2}\phi(1)\right) = \sum_{\ell=0}^{m} \frac{B_{2\ell}}{(2\ell)!} h^{2\ell} \int_0^1 \phi^{(2\ell)}(t)\, dt + o(h^{2m}),$$

where the coefficients B_{2k} are the Bernoulli numbers

$$B_0 = 1, \quad B_2 = \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \quad B_8 = -\frac{1}{30}, \ldots.$$

Now, following Davis and Rabinowitz [4], pp. 108–109 and 327–331, we subtract this formula from twice the formula obtained by replacing h by $h/2$ to get an expansion for the midpoint rule,

$$h \sum_{i=0}^{n-1} \phi(u_i) = -\sum_{\ell=0}^{m} C_\ell h^{2\ell} \int_0^1 \phi^{(2\ell)}(t)\, dt + o(h^{2m}), \tag{12}$$

where $C_\ell := (1 - 2^{1-2\ell}) B_{2\ell}/(2\ell)!$. Applying the expansion (12) to the function $c_j \in C^{2k-2j}[0,1]$ and setting $m = k - j$ gives

$$h \sum_{i=0}^{n-1} c_j(u_i) = -\sum_{\ell=0}^{k-j} C_\ell h^{2\ell} \int_0^1 c_j^{(2\ell)}(t)\, dt + o(h^{2k-2j}),$$

and substituting this into (11) yields

$$L_h(\mathbf{f}) = -\sum_{\ell=0}^{k} d_\ell h^{2\ell} - o(h^{2k}), \quad d_\ell := \int_0^1 \sum_{j=0}^{\ell} C_j c_{\ell-j}^{(2j)}(t)\, dt. \tag{13}$$

Since $c_0(t) = |\mathbf{f}'(t)|$, we have

$$d_0 = -\int_0^1 |\mathbf{f}'(t)|\, dt = -L(\mathbf{f}),$$

and therefore

$$L(\mathbf{f}) - L_h(\mathbf{f}) = \sum_{\ell=1}^{k} d_\ell h^{2\ell} + o(h^{2k}),$$

which is the required expansion. Finally, we compute d_1. Since

$$d_1 = \int_0^1 \left(-c_1(t) + \frac{1}{24} c_0''(t) \right) dt,$$

and

$$c_1 = \frac{1}{24} \frac{\mathbf{f}' \cdot \mathbf{f}'''}{|\mathbf{f}'|}, \qquad c_0'' = \frac{\mathbf{f}' \cdot \mathbf{f}''' + |\mathbf{f}''|^2}{|\mathbf{f}'|} - \frac{(\mathbf{f}' \cdot \mathbf{f}'')^2}{|\mathbf{f}'|^3},$$

we find that

$$d_1 = \frac{1}{24} \int_0^1 \left(\frac{|\mathbf{f}''(t)|^2}{|\mathbf{f}'(t)|} - \frac{(\mathbf{f}'(t) \cdot \mathbf{f}''(t))^2}{|\mathbf{f}'(t)|^3} \right) dt.$$

As noticed above, the terms involving third derivatives of $\mathbf{f}$ are canceled, and the formula (10) follows easily using the definition (3) of the curvature vector. □

3 Numerical Methods and Examples

Due to Theorem 1, we may now obtain numerical approximations to the curve length of any order, assuming regularity and sufficient smoothness of the curves. The basic underlying insight is, that if we may write the error of a method as a function of h, then by computing with different values of h we obtain data from which we may extrapolate to $h = 0$ by canceling the leading term in the error expansion. We give one way to construct such approximations, based on halving the interval length h for every iteration. Alternative ways of choosing h exist, and may be more efficient, but halving h is easy to analyze and program.

Assume that an approximation scheme $L_{h,j}$ of order $2j$ is given which for curves in C^{2k+1} provides the expansion

$$L(\mathbf{f}) - L_{h,j}(\mathbf{f}) = \sum_{\ell=j}^{k} d_{\ell,j} h^{2\ell} + o(h^{2k})$$

with some coefficients $d_{\ell,j}$. Define the approximation $L_{h,j+1}(\mathbf{f})$ by

$$L_{h,j+1}(\mathbf{f}) := \alpha_j L_{h,j}(\mathbf{f}) + \beta_j L_{h/2,j+1}(\mathbf{f}),$$

where $\alpha_j := 1 - \beta_j$ and $\beta_j := 4^j/(4^j - 1)$. Then

$$\begin{aligned} L(\mathbf{f}) - L_{h,j+1} &= (\alpha_j + \beta_j)L(\mathbf{f}) - \alpha_j L_{h,j}(\mathbf{f}) - \beta_j L_{h/2,j}(\mathbf{f}) \\ &= \sum_{\ell=j}^{k} (\alpha_j + \beta_j/4^\ell) d_{\ell,j} h^{2\ell} + o(h^{2k}) = \sum_{\ell=j+1}^{k} d_{\ell,j+1} h^{2\ell} + o(h^{2k}), \end{aligned}$$

where $d_{\ell,j+1} := (\alpha_j + \beta_j/4^\ell)d_\ell$. So $L_{h,j+1}$ is an approximation scheme of order $2(j+1)$. Clearly the expansion for $L - L_{h,j+1}$ has the same form as the one for $L - L_{h,j}$. So we may iterate the process until obtaining $L_{h,k}$ with an error of order $2k$.

3.1 The Romberg table

We will now provide an algorithm for approximating the arc length of a curve which is very similar to Romberg integration in the sense that it uses the same extrapolation technique. This is possible since in both cases the underlying basic approximation schemes (composite chord length and composite trapezoidal rule, respectively) yield an error expansion in even powers of h. Of course, our method is best suited for C^∞-curves, such as Bézier curves, or piecewise C^∞-curves, such as spline curves. In the latter case, the lengths of the smooth segments can be computed separately and are then added up. For curves of finite smoothness the size of the Romberg table has to be limited appropriately.

We will compute a table of the form

$$\begin{array}{ccccc}
R(0,0) & & & & \\
R(1,0) & R(1,1) & & & \\
R(2,0) & R(2,1) & R(2,2) & & \\
\vdots & \vdots & \vdots & \ddots & \\
R(N,0) & R(N,1) & R(N,2) & \cdots & R(N,N)
\end{array} \tag{14}$$

where $R(i,0)$ is the composite chordal approximation over 2^i intervals. The other table entries are given by

$$R(i,j) = R(i,j-1) + \frac{1}{4^j - 1}\big(R(i,j-1) - R(i-1,j-1)\big). \tag{15}$$

This formula is the same as for Romberg integration, only the computation of the leftmost column is different. There is an essential difference in the storage requirements, though. In Romberg integration, there is no need to store the function evaluations – $R(i,0)$ can be computed from $R(i-1,0)$ and a sum of new function evaluations. For chord-lengths however, we need to store all evaluated points in order to compute their distances to new points.

The entry $R(i,j)$ corresponds to $L_{h,j+1}$ with $h = 2^{j-i}$ that was described earlier. That is, we expect a scheme of order $2(j+1)$ in the column with index j.

3.2 The Romberg algorithm

In the pseudocode following, we have assumed the following conventions: Arrays are indexed from 0. The notation **for** $k = k_1$ **to** k_2 is to be interpreted as letting k take on all integer values in $[k_1, k_2]$, including in particular the value k_2.

Algorithm 3.1 takes an array P of points and returns the accumulated chord-length.

Algorithm 3.2 takes an array P of points and returns a new array of size $2 \cdot \text{ArrayLength}(P) - 1$. The new array includes (at its even indices) all the points of P, and (at its odd indices) points evaluated half-way between the original points.

Algorithm 3.3 computes an approximation to the arc-length of the parametric curve f on the interval $[0,1]$. The maximum number of Romberg table rows computed is $N+1$, like in (14). The optional tolerance parameter τ together with the code block marked as optional give a way to terminate the iteration once the error estimate $e := |R(i,i) - R(i-1,i-1)|$ is less than τ. In fact, e estimates the error of $R(i-1,i-1)$ since, presumably, $R(i,i)$ is much more accurate than this value. For the same reason, e should over-estimate the error of $R(i,i)$ so that it makes sense to return $R(i,i)$ together with e. It should be noted that e is not an error *bound*. In pathological cases, it can be far smaller that the actual error

– one of the examples will show that quite clearly. Of course, the given basic implementation can (and should) be improved in various ways. In particular, if error estimates indicate that the values in the first rows of the table are far from the true length, those values should be discarded so that the extrapolation starts from a sufficiently good approximation, for which the asymptotic expansion makes sense. Also adaptive variants are conceivable: If for a given value of N the error estimate indicates insufficient accuracy, then the curve is split at the interval midpoint and the two parts are treated separately. For instance, the resulting recursive algorithm can avoid excessive global refinement of curves with few isolated fine features.

3.3 Numerical examples

In order to investigate convergence properties of our scheme, we first consider Pythagorean hodograph curves (PH-curves) [6]. These are polynomial curves $\mathbf{f}: \mathbb{R} \to \mathbb{R}^d$ with the property that their speed functions $|\mathbf{f}'(t)|$ are polynomials. So their arc length is easy to compute exactly. For $\alpha \in \mathbb{R}$,

$$\mathbf{f}_\alpha(t) = \left(\frac{1}{3}(t+\alpha)^3 - \frac{1}{5}(t+\alpha)^5, \frac{1}{2}(t+\alpha)^4\right), \quad t \in [0,1] \tag{16}$$

is a PH-curve with speed function $|\mathbf{f}'_\alpha| = (t+\alpha)^2 + (t+\alpha)^4$. We obtain, for instance, $L(\mathbf{f}_0) = 8/15$ and $L(\mathbf{f}_1) = 128/15$.

Let us consider the case $\alpha = 1$. Figure 1, left, shows the actual error obtained for the Romberg table generated by Algorithm 3.3 with $N = 5$. Remarkably, the true length $L(\mathbf{f}_1) = 128/15$ is approximated with an error of less than 3e-13 using only 33 function evaluations. In Table 1 we show the experimental rates of convergence (ERC) for the schemes found in the columns of the Romberg table. At least in the first four columns, the theoretical prediction $R(i,j)/R(i+1,j) \approx 4^{j+1}$ is matched quite well. The ratios of the actual errors of $R(i,i)$ on the diagonal and its estimates $e = |R(i+1,i+1) - R(i,i)|$ all lie in the interval $[1, 1.02]$ indicating that, in this case, e is a reliable estimate. Of course, the actual accuracy $L(\mathbf{f}_1) - R(4,4) \approx$ 3e-13 of the returned value is much higher than the final estimate $e \approx$ 1e-09.

On the right hand side, Figure 1 and Table 1 show the according results for the parameter value $\alpha = 0$. We see that the ERCs in the third and fourth column are much smaller than before. The values suggest that in this case convergence is limited to the order h^5. This behavior is due to the fact that $\mathbf{f}_0$ is *not* a regular curve since $\mathbf{f}'_0(0) = 0$.

As a second example, let us consider the helix

$$\mathbf{g}(t) = \big(\cos(50\,t),\ \sin(50\,t), t\big), \quad t \in [0,1].$$

It has approximately 8 windings and length $L(\mathbf{g}) = \sqrt{2501} \approx 50.01$. Calling Algorithm 3.3 with the optional tolerance parameter $\tau =$ 1e-10, the iteration is

Algorithm 3.1: $\textsc{Chordal}(P)$

$n = \textsc{Arraylength}(P) - 1$
$l = 0$
for $i = 1$ **to** n
  do $l = l + |P(i) - P(i-1)|$
return (l)

Algorithm 3.2: $\textsc{Newpoints}(P, f)$

$n = \textsc{Arraylength}(P) - 1$
for $i = 0$ **to** $n - 1$
  do $\begin{cases} Q(2i) = P(i) \\ Q(2i+1) = f((2i+1)/(2n)) \end{cases}$
$Q(2n) = P(n)$
return (Q)

Algorithm 3.3: $\textsc{Romberg chord-length}(f, N, [\tau])$

$P = [f(0),\ f(1)]$
$R(0,0) = \textsc{Chordal}(P)$
for $i = 1$ **to** N

$$\textbf{do} \begin{cases} P = \textsc{Newpoints}(P, f) \\ R(i,0) = \textsc{Chordal}(P) \\ \textbf{for } j = 1 \textbf{ to } i \\ \quad \textbf{do} \begin{cases} r = \big(R(i,j-1) - R(i-1,j-1)\big)/(4^j - 1) \\ R(i,j) = R(i,j-1) + r \end{cases} \\ \text{optional} \begin{cases} e = |R(i,i) - R(i-1,i-1)| \\ \textbf{if } e < \tau \\ \quad \textbf{then return } (R(i,i), e) \end{cases} \end{cases}$$

return $(R(N,N), [e])$

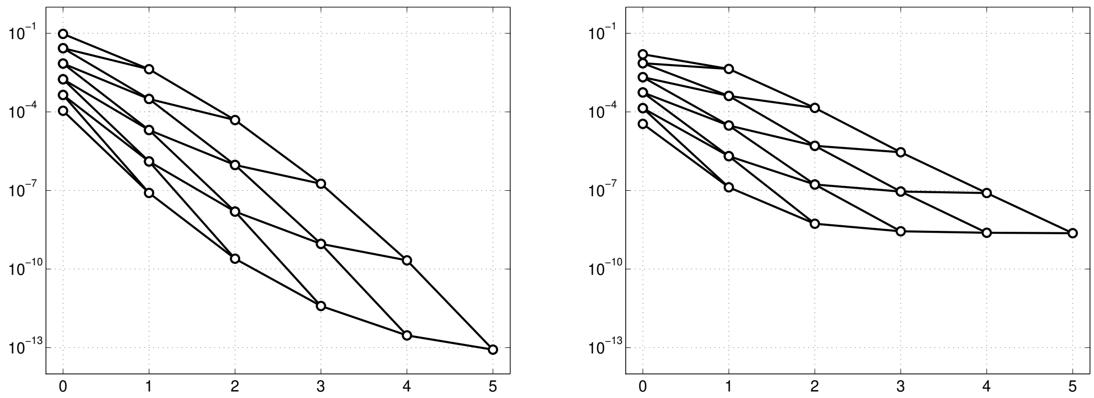

Figure 1: Error of Romberg table for PH-curves $\mathbf{f}_1$ *(left)*, and $\mathbf{f}_0$ *(right)*.

0	1	2	3	4
3.53				
3.87	13.65			
3.97	15.31	51.92		
3.99	15.82	60.29	196.38	
4.00	15.95	63.01	236.74	731.70

0	1	2	3	4
2.19				
3.42	10.75			
3.84	13.48	28.02		
3.96	14.76	30.05	31.84	
3.99	15.39	31.47	32.84	32.99

Table 1: ERC of Romberg table for PH-curves $\mathbf{f}_1$ *(left)*, and $\mathbf{f}_0$ *(right)*.

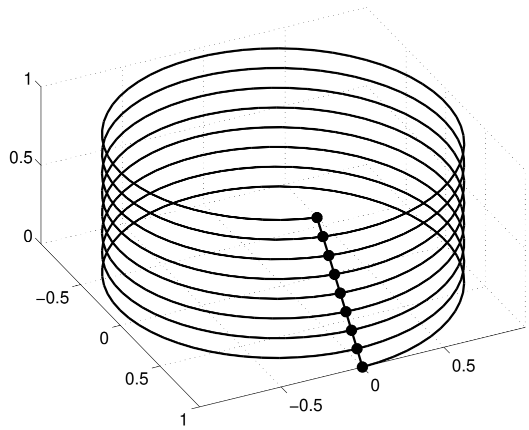

Figure 2: Helix with shortcut.

stopped after only three steps, and the return value is $R(3,3) \approx 1.03$. This is completely off the target since the error estimate fails spectacularly here. The point is that, incidentally, the 9 points used to compute the approximations up to the third row are all very close to a straight line, see Figure 2. This shortcut is resolved only if more points are included, disregard the far too optimistic error estimate. With $N = 8$, we obtain the true length with error < 2e-09. As always, the phenomenon of false error estimation cannot be excluded for a discrete method unless further information on the considered curve is available.

4 Approximation of Surface Area

In this section, we consider the problem of computing the area

$$A(\mathbf{s}) := \int_D |\mathbf{s}_u(u,v) \times \mathbf{s}_v(u,v)|\, du\, dv, \tag{17}$$

of a parametric surface $\mathbf{s} : D \to \mathbb{R}^3$ defined on the unit square $D = [0,1]^2$. We assume that $\mathbf{s}$ is regular, by which we mean that the integrand in (17) is never zero. For any chosen n, let $u_i = (i + 1/2)h$ and $v_j = (j + 1/2)h$, where $h = 1/n$. We then make the approximation

$$A_h(\mathbf{s}) := \frac{1}{2} \sum_{i,j=0}^{n-1} |\delta_{h,-h}\mathbf{s}(u_i, v_j) \times \delta_{h,h}\mathbf{s}(u_i, v_j)| \approx A(\mathbf{s}),$$

where

$$\delta_{\alpha,\beta}\mathbf{s}(u,v) := \mathbf{s}(u + \alpha/2, v + \beta/2) - \mathbf{s}(u - \alpha/2, v - \beta/2).$$

That is, the surface $\mathbf{s}$ is sampled on a square grid, and the area of each quadrilateral facet is approximated by half the norm of the cross product of its two diagonals.

Theorem 2 *Let $\mathbf{s} : D \to \mathbb{R}^3$ be a regular parametric surface with $\mathbf{s} \in C^{2k+1}(D)$. Then there exist coefficients d_p independent of h such that*

$$A(\mathbf{s}) - A_h(\mathbf{s}) = \sum_{p=1}^{k} d_p h^{2p} + o(h^{2k}). \tag{18}$$

Unlike in the curve case, the first coefficient d_1 involves third derivatives of $\mathbf{s}$ as well as first and second ones and we do not write down an explicit expression.

We note that in [15] it was shown that the above method is second order, if the surface $\mathbf{s} \in C^{2,2}$. So as in the curve case, we conjecture that the smoothness requirement above may be relaxed, but make no attempt to prove this. Also, a similar expansion was studied by Lyness in [14] for the purpose of quadrature on surfaces. Applying the basic approximation in that paper to integrating the constant function 1 over a surface reduces to, for every quad, adding the areas of the two triangles obtained by splitting the quad. So the basic approximation is different. In particular, it requires the computation of two cross products per quad, instead of only one cross product in our approach.

Proof. Using four Taylor expansions around the point $\mathbf{s}(u_i, v_j)$ with remainder of order $2k+1$, one easily finds that

$$\delta_{h,-h}\mathbf{s}(u_i, v_j) \times \delta_{h,h}\mathbf{s}(u_i, v_j) = \sum_{p=0}^{k} \mathbf{a}_p(u_i, v_j) h^{2p+2} + o(h^{2k+2})$$

for certain coefficients $\mathbf{a}_p(u, v)$, where $\mathbf{a}_0 := 2(\mathbf{s}_u \times \mathbf{s}_v)$. The dot product of this with itself is

$$|\delta_{h,-h}\mathbf{s}(u_i, v_j) \times \delta_{h,h}\mathbf{s}(u_i, v_j)|^2 = \sum_{p=0}^{k} b_p(u_i, v_j) h^{2p+4} + o(h^{2k+4}),$$

for coefficients $b_p(u, v)$ with $b_0 = 4|\mathbf{s}_u \times \mathbf{s}_v|^2$. Then, analogous to the curve case, since the leading term $h^4 b_0(u_i, v_j)$ is positive by the regularity of $\mathbf{s}$, there is an expansion

$$|\delta_{h,-h}\mathbf{s}(u_i, v_j) \times \delta_{h,h}\mathbf{s}(u_i, v_j)| = \sum_{p=0}^{k} c_p(u_i, v_j) h^{2p+2} + o(h^{2k+2})$$

for some coefficients $c_p(u, v)$ with $c_0 = 2|\mathbf{s}_u \times \mathbf{s}_v|$. Summing over i and j gives

$$\frac{1}{2}\sum_{i,j=0}^{n-1} |\delta_{h,-h}\mathbf{s}(u_i, v_j) \times \delta_{h,h}\mathbf{s}(u_i, v_j)| = \frac{1}{2}\sum_{p=0}^{k} h^{2p+2} \sum_{i,j=0}^{n-1} c_p(u_i, v_j) + o(h^{2k}). \quad (19)$$

Finally, we apply the expansion for the midpoint rule in (12) in both parameters to get the bivariate midpoint rule expansion

$$h^2 \sum_{i,j=0}^{n-1} \phi(u_i, v_j) = \sum_{p=0}^{m} h^{2p} \sum_{q=0}^{p} C_q C_{p-q} \int_D \partial_u^{2q} \partial_v^{2p-2q} \phi(u, v)\, du\, dv + o(h^{2m})$$

Substitution of this expansion with $\phi = c_p$ and $m = k - p$ into (19) gives

$$\frac{1}{2}\sum_{i,j=0}^{n-1} |\delta_{h,-h}\mathbf{s}(u_i, v_j) \times \delta_{h,h}\mathbf{s}(u_i, v_j)| = -\sum_{q=0}^{k} d_q h^{2q} + o(h^{2k}),$$

where

$$d_0 = -\int_D |\mathbf{s}_u \times \mathbf{s}_v|\, du\, dv.$$

□

References

[1] D. N. Arnold. *A concise introduction to numerical analysis*. Lecture notes, IMA, Minnesota.

[2] G. Casciola and S. Morigi. Reparametrization of nurbs curves. *Int. Journal of Shape Modelling*, 2:103–116, 1996.

[3] P. Constantini, R. T. Farouki, C. Manni, and A. Sestini. Computation of optimal composite re-parametrizations. *Computer Aided Geometric Design*, 18:875–897, 2001.

[4] Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Academic Press, 1975.

[5] R. T. Farouki. Optimal parametrizations. *Computer Aided Geometric Design*, 8:153–168, 1997.

[6] R. T. Farouki. Pythagorean-hodograph curves. In *Handbook of Computer Aided Geometric Design*, pages 405–427, 2002.

[7] R. T. Farouki and T. Sakkalis. Real rational curves are not 'unit speed'. *Computer Aided Geometric Design*, 8:151–157, 1991.

[8] M. S. Floater. Arc length estimation and the convergence of polynomial curve interpolation. *To appear in BIT*.

[9] M. S. Floater. Chordal cubic spline interpolation is fourth order accurate. *To appear in IMA J. Numer. Anal.*, 2005.

[10] M. S. Floater and A. F. Rasmussen. Point-based methods for estimating the length of a parametric curve. *J. Comp. Appl. Math.*, 196:512–522, 2006.

[11] Jens Gravesen. Adaptive subdivision and the length and energy of Bézier curves. *Comput. Geom.*, 8:13–31, 1997.

[12] B. Guenter and R. Parent. Computing the arc length of parametric curves. *IEEE Comp. Graph. and Appl.*, 5:72–78, 1990.

[13] J. Kearney H. Wang and K. Atkinson. Arc-length parameterized spline curves for real-time simulation. In *Curve and Surface Design, Saint-Malo*, pages 387–396, 2002.

[14] J. N. Lyness. Quadrature over curved surfaces by extrapolation. *Math.Comp.*, 63:727–740, 1994.

[15] A. F. Rasmussen and M. S. Floater. A point-based method for estimating surface area. In *Proceedings of the SIAM conference on Geometric Design and Computing, Phoenix 2005*.

[16] R. J. Sharpe and R. W. Thorne. Numerical method for extracting an arc length parameterization from parametric curves. *Computer-Aided Design*, 14(2):79–81, 1982.

[17] S. Vincent and D. Forsey. Fast and accurate parametric curve length computation. *J. Graph. Tools*, 6(4):29–40, 2002.

[18] M. Walter and A. Fournier. Approximate arc length parameterization. In *Proceedings of the 9th Brazilian symposium on computer graphics and image processing*, pages 143–150, 1996.

[19] F.-C. Wang and D. C. H. Yang. Nearly arc-length parameterized quintic-spline interpolation for precision machining. *Computer-Aided Design*, 25(5):281–288, 1993.