

UC Davis

UC Davis Previously Published Works

Title

Accurate Real-time CNC Curve Interpolators Based Upon Richardson Extrapolation

Permalink

<https://escholarship.org/uc/item/40t858d5>

Author

Farouki, Rida T

Publication Date

2021-06-01

DOI

10.1016/j.cad.2021.103005

Peer reviewed

Accurate real-time CNC curve interpolators based upon Richardson extrapolation

Rida T. Farouki

Department of Mechanical and Aerospace Engineering,
University of California, Davis, CA 95616, USA.

Abstract

Real-time CNC interpolators achieving a constant or variable feedrate V along a parametric curve $\mathbf{r}(\xi)$ are usually based on truncated Taylor series expansions defining the time-dependence of the curve parameter ξ . Since the feedrate should be specified as a function of a physically meaningful variable (such as time t , path arc length s , or curvature κ) rather than ξ , successive applications of the differentiation chain rule are necessary to determine Taylor series coefficients beyond the linear term. The closed-form expressions for the higher-order coefficients are increasingly cumbersome to derive and implement, and consequently error-prone. To address this issue, the use of *Richardson extrapolation* as a simple means to compute rapidly convergent approximations to the higher-order coefficients is investigated herein. The methodology is demonstrated in the context of (1) an arc-length-dependent feedrate for cornering motions; (2) direct real-time offset curve interpolation; and (3) a curvature-dependent feedrate. All of these examples admit simple implementations that circumvent the need for tedious symbolic calculations of higher-order coefficients, and are compatible with real-time controllers with millisecond sampling intervals.

Keywords: real-time CNC interpolator, parametric curves, variable feedrate, Taylor series coefficients, Richardson extrapolation, feedrate accuracy.

e-mail address: farouki@ucdavis.edu

1 Introduction

Real-time feedback control of multi-axis computer numerical control (CNC) machines requires accurate determination of the position error (the difference between the commanded and measured machine positions), updated in each sampling interval $\Delta t = 1/f$, where f is the servo system sampling frequency (with typical values $f = 100\text{--}1000$ Hz and $\Delta t = 0.01\text{--}0.001$ sec.). The actual machine position is measured by the encoders on each machine axis, and the commanded position must be computed from the prescribed path geometry and feedrate (path speed) by the *real-time interpolator* algorithm.

Although CNC machines have traditionally employed G code (piecewise-linear/circular) approximations to variable-curvature paths, the advantages of analytic path descriptions — specified by parametric curves — has recently been recognized [2, 3, 14, 16, 18, 19, 24, 25]. Analytic paths, based upon the standard Bézier/B-spline CAD system representations [4], can offer greater geometrical accuracy, smoother high-speed motion, and the ability to achieve continuously-variable feedrate to satisfy requirements such as suppression of chip load variations and satisfaction of axis acceleration limits.

The real-time interpolator for a (non-trivial) parametric curve $\mathbf{r}(\xi)$ must determine, from the specified (constant or variable) feedrate V , the parameter value ξ_i corresponding to sampling time $t_i = i \Delta t$. However, feedrate is the derivative $V = ds/dt$ of arc length s along $\mathbf{r}(\xi)$, and in general there exists no simple function¹ $s(\xi)$ specifying s in terms of ξ . The usual remedy to this problem is to invoke a Taylor series expansion to determine ξ_{i+1} from ξ_i .

Real-time interpolators based on Taylor series possess two inherent shortcomings: (i) they incur *truncation errors* because only a finite (small) number of terms can be employed; and (ii) for variable feedrates, the coefficients of higher-order terms, defined by successive chain-rule differentiation, acquire increasingly cumbersome and computation-intensive expressions.

It should be noted that real-time interpolator inaccuracies incur only *feed error* (lag or lead along the path) — not *contour error* (geometrical deviation from the path) — and linear or quadratic Taylor expansions may suffice for applications involving only constant low feedrates, modest path curvatures, and sufficiently high controller sampling frequencies. However, in *high-speed machining* [15, 21, 22] which incurs severe accelerations, or when pronounced

¹The Pythagorean-hodograph (PH) curves [5], for which $s(\xi)$ is simply a polynomial, are the exception to this rule: they admit accurate real-time interpolators [6, 7, 10, 13, 23] for various feedrate variations dependent on time, arc length, or curvature.

feedrate variations are employed to suppress chip load variations and avoid tool damage, accurate feedrate performance may be paramount.

Motivated by these considerations, the focus of the present study is to develop and analyze a simple, computationally-efficient scheme to determine the higher-order coefficients for Taylor series variable-feedrate interpolators, that bypasses the need to derive their cumbersome and error-prone [24, 25] closed-form expressions. The approach is based on *Richardson extrapolation*, an efficient iterative scheme for computing a sequence of increasingly accurate approximations to higher-order derivatives of a function from sampled values of a known simple expression for a lower-order derivative [1, 20].

The remainder of this paper is organized as follows. First, basic principles governing the design of real-time CNC interpolators for parametric curves based on Taylor series expansions are briefly reviewed in Section 2, and the difficulty in improving their accuracy by incorporating higher-order terms is emphasized. The Richardson extrapolation method for computing accurate estimates of the higher-order coefficients, without deriving their closed-form expressions, is then introduced in Section 3. Sections 4–6 present computed examples illustrating the performance of the methodology for paths specified by smooth analytic curves, in the context of arc-length-dependent feedrates, real-time interpolation of offset curves, and curvature-dependent feedrates. Finally, Section 7 summarizes the principal results of this study, and identifies further possible lines of investigation.

2 Real-time CNC interpolators

Given a prescribed tool path geometry and a (constant or variable) feedrate, the *real-time interpolator* function in the controller of a computer numerical control (CNC) machine with servo sampling frequency f computes a *reference point* (commanded position) within each system sampling interval $\Delta t = 1/f$. The difference between the reference point and the actual machine position, as measured by encoders on the machine axes, defines the instantaneous axis position errors, required for accurate closed-loop control of position.

To generate reference points along a path specified by a parametric curve $\mathbf{r}(\xi)$, the curve parameter value ξ_i corresponding to each sampling instant $t_i = i \Delta t$ must be computed in real time. Since the *parametric speed* $\sigma(\xi) = |\mathbf{r}'(\xi)|$ specifies the derivative $ds/d\xi$ of arc length s along $\mathbf{r}(\xi)$ with respect to the parameter ξ , and the feedrate V is the time derivative ds/dt of the

arc length, we have

$$\frac{d}{dt} = \frac{ds}{dt} \frac{d\xi}{ds} \frac{d}{d\xi} = \frac{V}{\sigma} \frac{d}{d\xi}. \quad (1)$$

In particular, the basic relation that governs the time variation of the curve parameter is

$$\frac{d\xi}{dt} = \frac{V}{\sigma}, \quad (2)$$

where the feedrate V may be constant, or defined as a function of a physically meaningful variable — such as the elapsed time t , path arc length s , or path curvature κ . However, the relation (2) does not (in general) admit integration to obtain a closed-form expression $\xi(t)$ — even in the case $V = \text{constant}$ — since $\sigma(\xi)$ is the square-root of a polynomial/rational function when $\mathbf{r}(\xi)$ is a polynomial/rational curve. The Pythagorean-hodograph (PH) curves [5] are an exception, for which $\sigma(\xi)$ is a polynomial and (2) admits a closed-form integration for various feedrate functions of practical interest.

The usual approach is to expand the relation (2) as a power series in Δt . Namely, the reference-point parameter value ξ_{i+1} at time $t_{i+1} = (i+1)\Delta t$ is obtained from the preceding value as $\xi_{i+1} = \xi_i + \Delta\xi_i$, the increment $\Delta\xi_i$ being computed from the Taylor series

$$\Delta\xi_i = \dot{\xi}_i \Delta t + \frac{1}{2} \ddot{\xi}_i (\Delta t)^2 + \frac{1}{6} \dddot{\xi}_i (\Delta t)^3 + \dots \quad (3)$$

where $\dot{\xi}_i, \ddot{\xi}_i, \dddot{\xi}_i$, etc., denote successive time derivatives² of $\xi(t)$, evaluated at $t_i = i\Delta t$. Now by successive application of the differential operator (1) we may express the derivatives in (3) as

$$\dot{\xi} = \frac{V}{\sigma}, \quad \ddot{\xi} = \frac{\sigma V' - \sigma' V}{\sigma^2} \dot{\xi}, \quad \dddot{\xi} = \frac{\sigma V'' - 3\sigma' V'}{\sigma^2} \ddot{\xi} + \frac{\sigma V''' - \sigma'' V''}{\sigma^2} \dot{\xi}^2, \quad (4)$$

etc., the derivatives of the parametric speed $\sigma(\xi) = |\mathbf{r}'(\xi)|$ being given by

$$\sigma' = \frac{\mathbf{r}' \cdot \mathbf{r}''}{\sigma}, \quad \sigma'' = \frac{\mathbf{r}' \cdot \mathbf{r}''' + |\mathbf{r}''|^2 - \sigma'^2}{\sigma}, \quad \text{etc.} \quad (5)$$

When the feedrate V is specified as a function of time t , arc length s , or path curvature κ , further applications of the differentiation chain rule are required to convert the derivatives with respect to these variables into the parametric

²Henceforth dots are used to denote derivatives with respect to time t , while primes denote derivatives with respect to the curve parameter ξ .

derivatives V' , V'' , etc., appearing in (4). The derivation and implementation of the above expressions is tedious (and potentially error-prone), resulting in very cumbersome formulae [11] for the higher-order time derivatives³ of ξ .

To address this issue, we consider here the use of *Richardson extrapolation* to compute the higher-order coefficients in (3), without the need to explicitly derive their exact closed-form expressions. Richardson extrapolation offers a simple means to obtain a convergent sequence of estimates for higher-order derivatives of an analytic function at a given point, based on function values at a successively augmented set of discrete points. The accuracy improves as additional points are introduced, and with appropriate implementation the computational cost increases only in proportion to their number.

The test results described in Sections 4–6 indicate that relative accuracies of $\lesssim 10^{-10}$ in the second- and third-order terms of (3) can be achieved in a small fraction of the typical sampling interval $\Delta t = 0.001$ s. The primary motivation in using Richardson extrapolation is not computational efficiency, but circumventing the need to derive and implement the complicated closed-form expressions for the higher-order coefficients in (3), or determining their numerical values in cases where such derivations are not feasible.

In typical applications, only the first few terms in the expansion (3) may prove sufficiently accurate. However, there are many circumstances in which higher-order terms may be necessary — for example:

- a prescribed feedrate V that is very high, or exhibits a strong variation along the path $\mathbf{r}(\xi)$;
- a feedrate V that is constant (or only mildly varying) along a path $\mathbf{r}(\xi)$ with a strongly-varying parametric speed $\sigma(\xi)$;
- feed deceleration/acceleration for accurate execution of sharp junctures between smooth curvilinear path segments;
- the use of a curvature-dependent feedrate for a path $\mathbf{r}(\xi)$ along which the curvature $\kappa(\xi)$ exhibits strong variation;
- acceleration management in rapid machine starts or stops, by means of a time-dependent feedrate along $\mathbf{r}(\xi)$;

³Note also that the derivation of these expressions has been susceptible to elementary mistakes by prior authors, as observed in [11].

- the use of controllers with relatively low sampling frequencies (i.e., long sampling intervals Δt).

3 Richardson extrapolation

Consider the determination of estimates for the second derivative $\ddot{\xi}_i$ from a known expression for the first derivative. For brevity write $h = \Delta t$, and let $F_1(h)$ be a function such that $F_1(0)$ coincides with the desired derivative $\dot{\xi}_i$. For any $h > 0$, the deviation of $F_1(h)$ from $\dot{\xi}_i$ can be expressed as

$$F_1(h) - \dot{\xi}_i := c_1 h + O(h^2). \quad (6)$$

Substituting ηh for h in (6), with $0 < \eta < 1$, gives

$$F_1(\eta h) - \dot{\xi}_i = c_1 \eta h + O(h^2). \quad (7)$$

Multiplying (6) by η and subtracting from (7) then furnishes an estimate for $\ddot{\xi}_i$, accurate to second order in h , as

$$\ddot{\xi}_i = \frac{F_1(\eta h) - \eta F_1(h)}{1 - \eta} + O(h^2). \quad (8)$$

The coefficient c_1 in (6) can also be estimated, to second order in h , as

$$c_1 = \frac{F_1(h) - F_1(\eta h)}{(1 - \eta)h} + O(h^2).$$

Equation (8) specifies a *Richardson extrapolation* estimate for $\ddot{\xi}_i$, of accuracy $O(h^2)$, based on a prescribed function $F_1(h)$ of accuracy $O(h)$.

The relation (8) can be written as

$$\ddot{\xi}_i = F_2(h) + O(h^2),$$

where we define the function

$$F_2(h) := \frac{F_1(\eta h) - \eta F_1(h)}{1 - \eta}.$$

This function can now be invoked as a new point of departure for equation (6), re-formulated as

$$F_2(h) - \ddot{\xi}_i := c_2 h^2 + O(h^3), \quad (9)$$

since $F_2(h)$ is of accuracy $O(h^2)$ as an estimate for $\ddot{\xi}_i$. Again replacing h by ηh , we have

$$F_2(\eta h) - \ddot{\xi}_i = c_2 \eta^2 h^2 + O(h^3). \quad (10)$$

Multiplying (9) by η^2 and subtracting from (10) gives a new estimate for $\ddot{\xi}_i$, accurate to third order in h , as

$$\ddot{\xi}_i = F_3(h) + O(h^3), \quad (11)$$

where

$$F_3(h) := \frac{F_2(\eta h) - \eta^2 F_2(h)}{1 - \eta^2}.$$

The above scheme, repeated indefinitely, is called *extrapolation to the limit*. It can be succinctly expressed in the form

$$\ddot{\xi}_i = F_k(h) + O(h^k),$$

where the functions $F_k(h)$ are defined recursively by

$$F_{k+1}(h) = \frac{F_k(\eta h) - \eta^k F_k(h)}{1 - \eta^k}, \quad k = 1, 2, 3, \dots \quad (12)$$

Experiments indicate that, in the present context, the accuracy achieved is not very sensitive to the choice of η . We focus here on the particular case $\eta = \frac{1}{2}$, for which we have

$$F_2(h) := 2 F_1(\tfrac{1}{2}h) - F_1(h), \quad F_3(h) := \frac{4 F_2(\tfrac{1}{2}h) - F_2(h)}{3},$$

$$F_4(h) := \frac{8 F_3(\tfrac{1}{2}h) - F_3(h)}{7}, \quad F_5(h) := \frac{16 F_3(\tfrac{1}{2}h) - F_4(h)}{15}, \quad \text{etc.}$$

For $k \geq 3$, the functions $F_k(h)$ can be expressed in terms of just $F_1(h)$ as

$$F_3(h) = \frac{8 F_1(\tfrac{1}{4}h) - 6 F_1(\tfrac{1}{2}h) + F_1(h)}{3},$$

$$F_4(h) = \frac{64 F_1(\tfrac{1}{8}h) - 56 F_1(\tfrac{1}{4}h) + 14 F_1(\tfrac{1}{2}h) - F_1(h)}{21},$$

$$F_5(h) = \frac{1024 F_1(\tfrac{1}{16}h) - 960 F_1(\tfrac{1}{8}h) + 280 F_1(\tfrac{1}{4}h) - 30 F_1(\tfrac{1}{2}h) + F_1(h)}{315},$$

etc. The function $F_1(h)$ must be defined in order to initiate the recursion. We do this by invoking the known closed-form expression (2) for $\dot{\xi}(t)$ and using the forward difference formula

$$F_1(h) = \frac{\dot{\xi}(t_i + h) - \dot{\xi}(t_i)}{h}, \quad (13)$$

which satisfies $\ddot{\xi}_i - F_1(h) = c_1 h + O(h^2)$.

Note that, if $P_k(x) = a_0 + a_1 x + \dots + a_k x^k$ is the polynomial of degree k that has the values $F_1(h), F_1(\frac{1}{2}h), \dots, F_1(2^{-k}h)$ at $x = h, \frac{1}{2}h, \dots, 2^{-k}h$, then $F_{k+1}(h) = P_k(0)$ — i.e., the quantities generated through the recursion (12) are the initial values of a sequence of polynomials, with increasing degree k , that interpolate the values of $F_1(x)$ at the geometric progression of sample points $x = h, \frac{1}{2}h, \dots, 2^{-k}h$ with the limit point 0.

The preceding discussion was couched in terms of computing estimates of the second derivative from a known expression for the first derivative, but the same principles also apply to computing estimates of the third derivative from a known second derivative expression. If no second derivative expression is known, extrapolated values from the first derivative may be used, although this may influence the efficiency and convergence rate.

To ensure an efficient implementation, it is useful to express the successive estimates for $\ddot{\xi}(t_i)$ explicitly in terms of the sequence of sampled $\dot{\xi}(t)$ values. Upon substituting (13) into the expressions for $F_2(h), F_3(h), \dots$ in terms of $F_1(h)$, and setting $t_k = t_i + (\frac{1}{2})^k h$, we obtain the following sequence of second derivative estimates

$$\begin{aligned} \ddot{\xi}_1(t_i) &= \frac{\dot{\xi}(t_1) - \dot{\xi}(t_i)}{h}, \\ \ddot{\xi}_2(t_i) &= \frac{-\dot{\xi}(t_1) + 4\dot{\xi}(t_2) - 3\dot{\xi}(t_i)}{h}, \\ \ddot{\xi}_3(t_i) &= \frac{\dot{\xi}(t_1) - 12\dot{\xi}(t_2) + 32\dot{\xi}(t_3) - 21\dot{\xi}(t_i)}{3h}, \\ \ddot{\xi}_4(t_i) &= \frac{-\dot{\xi}(t_1) + 28\dot{\xi}(t_2) - 224\dot{\xi}(t_3) + 512\dot{\xi}(t_4) - 315\dot{\xi}(t_i)}{21h}, \\ \ddot{\xi}_5(t_i) &= \frac{\dot{\xi}(t_1) - 60\dot{\xi}(t_2) + 1120\dot{\xi}(t_3) - 7680\dot{\xi}(t_4) + 16384\dot{\xi}(t_5) - 9765\dot{\xi}(t_i)}{315h}, \end{aligned}$$

etc., in terms of $\dot{\xi}(t)$ evaluated at t_1, t_2, \dots and t_i . From the above expressions it is clear that, once $\dot{\xi}_k(t)$ has been computed, evaluation of $\dot{\xi}_{k+1}(t)$ requires

only the one additional derivative value $\dot{\xi}(t_{k+1})$, so the computational cost grows only linearly with the order k of the approximation $\ddot{\xi}_k(t_i)$.

The same procedure can be employed to compute a sequence of estimates

$$\ddot{\xi}_1(t_i), \ddot{\xi}_2(t_i), \ddot{\xi}_3(t_i), \dots$$

for the third derivative $\ddot{\xi}(t_i)$ from a known second derivative expression $\ddot{\xi}(t)$ evaluated at t_1, t_2, t_3, \dots and t_i , and likewise for higher-order derivatives.

Once the derivatives $\dot{\xi}_i, \ddot{\xi}_i, \ddot{\xi}_i, \dots$ at time $t = i \Delta t$ have been determined up to some desired order, the expression (3) can be evaluated to obtain the curve parameter value $\xi_{i+1} = \xi_i + \Delta \xi_i$ at time $t_{i+1} = (i + 1) \Delta t$. The new reference point $\mathbf{r}(\xi_{i+1})$ is then generated, and the derivatives are re-computed at the new parameter value ξ_{i+1} in order to execute the next time step.

For brevity, the examples provided in the following sections are based on planar curves. However, the method is also applicable to spatial curves, and may be even more advantageous for space curves, since the exact expressions for higher-order Taylor coefficients will depend on the torsion and its arc-length derivatives in addition to the curvature. The Richardson scheme can also be readily generalized from single curve segments (as in the examples) to more complicated paths defined by multi-segment spline curves.

4 Arc-length-dependent feedrate

We illustrate the Richardson extrapolation method in terms of an arc-length-dependent feedrate, suitable for use in high-speed cornering motions. Sharp corners in engineering parts are typically smoothed by fillet curves to ensure safe handling, reduce stress concentration effects, and to circumvent the need to reduce the cutting tool feedrate to zero for exact execution of a sharp tool-path corner. For continuous acceleration along a toolpath, the fillet curves must exhibit G^2 continuity with the linear segments they connect.

A family of quintic Pythagorean-hodograph (PH) curves that define G^2 fillets of the sharp corners between linear segments was proposed in [9, 17]. These fillets replace portions of length L on the incoming and outgoing line segments at the corner — they agree in position and tangent direction, and have zero curvature, at their junctures with the remaining line segments. For a corner point at the origin, an incoming segment along the x -axis, and an outgoing segment inclined at angle θ to the x -axis, these quintic PH corner

curves are specified [9] by the Bézier control points

$$\mathbf{p}_0 = (-L, 0), \quad \mathbf{p}_1 = \left(\frac{-L}{6 \cos \frac{1}{2}\theta + 1}, 0 \right) = \mathbf{p}_2,$$

$$\mathbf{p}_3 = \left(\frac{L \cos \theta}{6 \cos \frac{1}{2}\theta + 1}, \frac{L \sin \theta}{6 \cos \frac{1}{2}\theta + 1} \right) = \mathbf{p}_4, \quad \mathbf{p}_5 = (L \cos \theta, L \sin \theta).$$

The coincidences $\mathbf{p}_1 = \mathbf{p}_2$ and $\mathbf{p}_3 = \mathbf{p}_4$ ensure zero end-point curvatures. Examples of the G^2 corner curves, corresponding to the choices $L = 0.1$ in and $\theta = 45^\circ, 90^\circ, 135^\circ$, are shown with their curvature profiles in Figure 1.

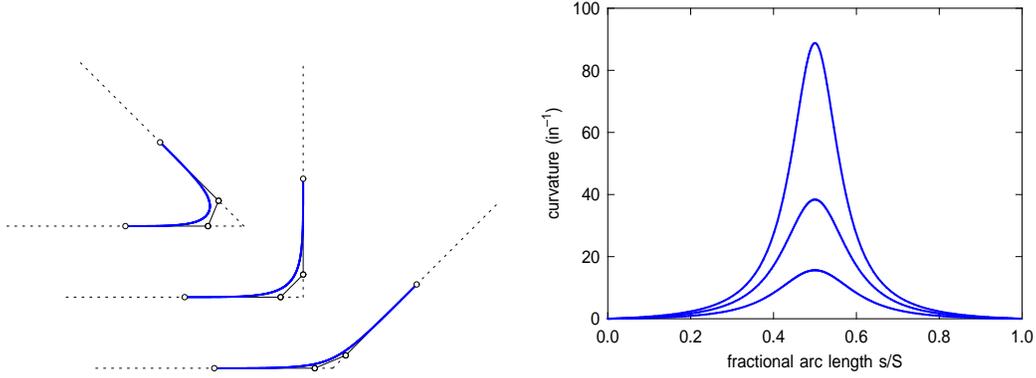


Figure 1: The G^2 PH quintic corner curves that correspond to turning angles $\theta = 45^\circ, 90^\circ, 135^\circ$ (left), together with their curvature distributions (right).

The G^2 quintic PH corner curve specified by the parameters L and θ has a mid-point maximum curvature κ_{\max} and a maximum deviation δ from the sharp corner given [9] by

$$\kappa_{\max} = \frac{32(6 \cos \frac{1}{2}\theta + 1) \tan \frac{1}{2}\theta}{15L(\cos \frac{1}{2}\theta + 1)^2}, \quad \delta = \frac{(3 \cos \frac{1}{2}\theta + 8) |\sin \frac{1}{2}\theta| L}{8(6 \cos \frac{1}{2}\theta + 1)},$$

and the total arc length of the corner curve can be expressed as

$$S = \frac{2L(6 + \cos \frac{1}{2}\theta) \cos \frac{1}{2}\theta}{6 \cos \frac{1}{2}\theta + 1}.$$

Feedrate functions dependent on the curve parameter ξ , curvature κ , or a combination of both, were proposed in [9] to decelerate into and accelerate

out of the quintic PH corner curves $\mathbf{r}(\xi)$. These allow determination of the reference–point parameter value ξ_i at each sampling time $t_i = i\Delta t$ as the unique real roots of monotone analytic functions. However, the derivations are rather complicated, and involve transcendental terms. We consider here instead an arc–length–dependent cornering feedrate function.

Since a PH curve $\mathbf{r}(\xi)$, $\xi \in [0, 1]$ has [5] a polynomial parametric speed $\sigma(\xi) = |\mathbf{r}'(\xi)|$, the cumulative arc length

$$s(\xi) = \int_0^\xi \sigma(u) \, du$$

is a polynomial, and the total arc length $S = s(1)$ can be exactly determined. We consider a cornering feedrate function defined in terms of the fractional arc length $\lambda = s(\xi)/S$ as

$$V(\lambda) = V_0 [1 - 16(1 - f)(1 - \lambda)^2 \lambda^2]. \quad (14)$$

This function embodies the following properties: (1) the entry/exit feedrate has the nominal value V_0 at $\lambda = 0$ and 1; (2) the mid–point feedrate $V(\frac{1}{2}) = f V_0$ is the fraction f of V_0 ; and (3) the entry/exit feed acceleration

$$A = \frac{dV}{dt} = \frac{ds}{dt} \frac{dV}{ds} = \frac{V}{S} \frac{dV}{d\lambda},$$

is zero at $\lambda = 0$ and 1. For a corner curve connecting incoming and outgoing linear segments with G^2 continuity, that are traversed at constant speed V_0 , the feedrate (14) ensures continuity of velocity and acceleration.

To explicitly compute the higher–order derivatives in (3) we observe from (14) that

$$\begin{aligned} \frac{dV}{ds} &= - \frac{32(1-f)V_0}{S} \lambda(1-3\lambda+2\lambda^2), \\ \frac{d^2V}{ds^2} &= - \frac{32(1-f)V_0}{S^2} (1-6\lambda+6\lambda^2), \end{aligned}$$

and these may be converted into derivatives with respect to ξ as

$$V' = \sigma \frac{dV}{ds}, \quad V'' = \sigma' \frac{dV}{ds} + \sigma^2 \frac{d^2V}{ds^2}. \quad (15)$$

These expressions, together with the derivatives (5) of the parametric speed, can then be used to evaluate the closed–form expressions (4) for the second and third time derivatives of ξ required in (3).

Although the above derivations were formulated recursively to obtain a reasonably compact formulation, they remain cumbersome and potentially error prone in implementation. To avoid these issues, we consider the use of Richardson extrapolation to compute the higher-order coefficients in (3).

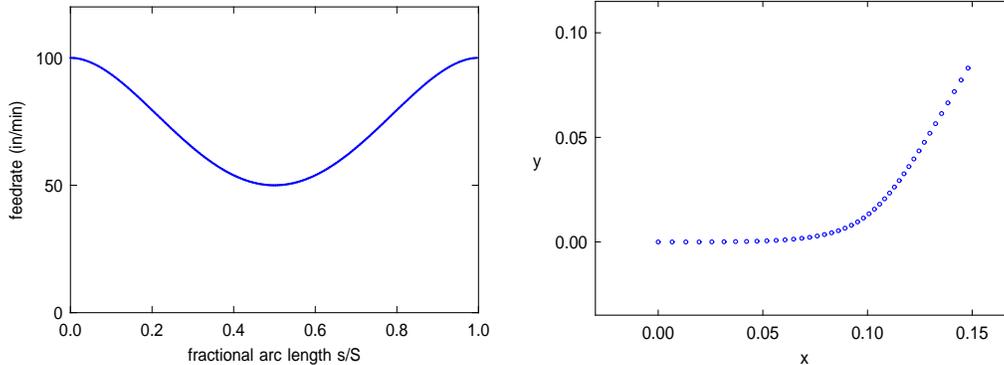


Figure 2: Feedrate variation (left) and subsampling of reference points (right) along a quintic PH corner test curve with the turning angle $\theta = 60^\circ$, to be traversed using the feedrate function (14) with $V_0 = 100$ in/min and $f = 0.5$.

As a test case, we employ the quintic PH corner curve with turning angle $\theta = 60^\circ$ and side length $L = 0.1$ in, to be traversed with a nominal feedrate $V_0 = 100$ in/min and feedrate reduction factor $f = 0.5$ (see Figure 2). The corner curve has arc length $S = 0.1919$ in, maximum deviation $\delta = 0.0107$ in, and extremum curvature $\kappa_{\max} = 21.9172$ in $^{-1}$.

Figure 3 shows that the Richardson extrapolation estimates for $\ddot{\xi}_k(t_i)$ and $\ddot{\xi}_k(t_i)$ can achieve high relative accuracies for modest k values ($\lesssim 10^{-5}$ with $k = 3$ and $\lesssim 10^{-10}$ with $k = 5$). To assess the computational cost, the corner curve was executed 1000 times, with each execution involving 170 reference-point computations for a sampling interval $\Delta t = 0.001$ sec. The average times required to compute the derivatives $\ddot{\xi}(t_i)$ and $\ddot{\xi}(t_i)$ for each reference point, based on both the closed-form expressions and the Richardson extrapolations $\ddot{\xi}_3(t_i)$ and $\ddot{\xi}_5(t_i)$, using a modest 1.1 GHz cpu, are listed in Table 1.

It is apparent from Table 1 that, since Richardson extrapolation involves more function evaluations, it is somewhat slower than using the closed-form derivative expressions. However, the primary motivation for the Richardson method is to circumvent the extensive (and sometimes impractical) symbolic computations required to derive exact closed-form derivatives. The data in Table 1 appear consistent with a linear growth of computation time with the

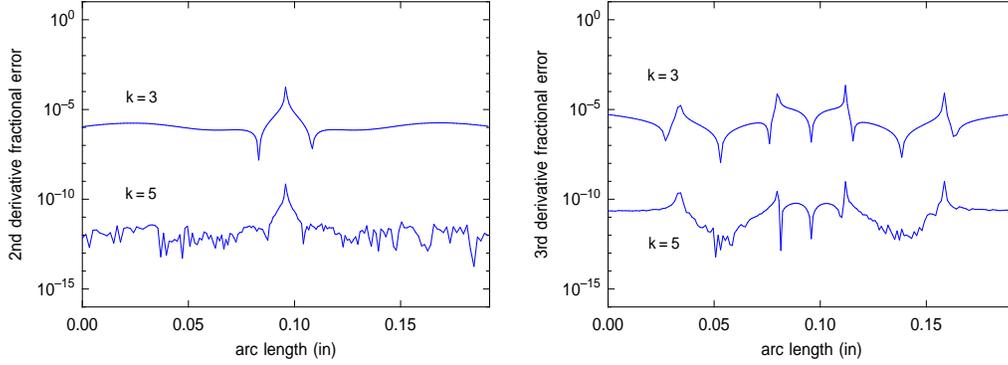


Figure 3: Relative errors in the second derivative estimates $\ddot{\xi}_3, \ddot{\xi}_5$ (left) and third derivative estimates $\dddot{\xi}_3, \dddot{\xi}_5$ (right) in the traversal of a quintic PH corner curve with the parameters $L = 0.1$ in, $\theta = 60^\circ$, $V_0 = 100$ in/min, and $f = 0.5$.

	closed form	Richardson $k = 3$	Richardson $k = 5$
$\ddot{\xi}(t_i)$	1.30×10^{-6} s	3.94×10^{-6} s	6.04×10^{-6} s
$\dddot{\xi}(t_i)$	1.61×10^{-6} s	5.17×10^{-6} s	8.04×10^{-6} s

Table 1: Observed computation times for the derivatives $\ddot{\xi}(t_i)$ and $\dddot{\xi}(t_i)$ at each reference point along a 60° PH quintic corner curve traversed with the feedrate function (14), employing the closed-form derivative expressions and Richardson extrapolations of order $k = 3$ and 5 (based on a 1.1 GHz cpu).

extrapolation order k , as noted in Section 3, and the computation times for $k \leq 8$ are $\lesssim 1\%$ of the adopted sampling interval $\Delta t = 0.001$ s, and are thus eminently compatible with real-time implementation.

5 Offset curve traversal

To machine a planar shape specified by a parametric curve $\mathbf{r}(\xi)$, $\xi \in [0, 1]$ with a cylindrical tool of radius d , the tool center must follow the offset path

$$\mathbf{r}_d(\xi) = \mathbf{r}(\xi) + d \mathbf{n}(\xi), \quad (16)$$

where $\mathbf{n}(\xi)$ is the unit normal to $\mathbf{r}(\xi)$. This offset path has [8] the derivative $\mathbf{r}'_d(\xi) = [1 + \kappa(\xi) d] \mathbf{r}'(\xi)$, with the curvature of $\mathbf{r}(\xi)$ being defined by

$$\kappa(\xi) = \frac{[\mathbf{r}'(\xi) \times \mathbf{r}''(\xi)] \cdot \mathbf{z}}{\sigma^3(\xi)}, \quad (17)$$

where \mathbf{z} is a unit vector orthogonal to the plane of $\mathbf{r}(\xi)$. The unit tangent $\mathbf{t}_d(\xi)$ of the offset curve (16) is related to the unit tangent $\mathbf{t}(\xi)$ of $\mathbf{r}(\xi)$ by

$$\mathbf{t}_d(\xi) = \frac{1 + \kappa(\xi) d}{|1 + \kappa(\xi) d|} \mathbf{t}(\xi), \quad (18)$$

The expression (16) defines, for unrestricted ξ , the “untrimmed” offset curve — which is locally (but not globally) at distance d from $\mathbf{r}(\xi)$. Equation (18) indicates that points where $1 + \kappa(\xi) d$ changes sign will incur sudden tangent reversals — or *cusps* — on the untrimmed offset curve. Such points lie within segments of the untrimmed offset at distance $< d$ from $\mathbf{r}(\xi)$, which must be deleted to obtain the *trimmed* offset curve, that will ensure gouge-free machining of the desired shape $\mathbf{r}(\xi)$. On each remaining trimmed offset segment, the expression $1 + \kappa(\xi) d$ is of constant sign.

For brevity we focus henceforth on a trimmed offset segment for which $1 + \kappa(\xi) d > 0$. The parametric speed $\sigma_d(\xi)$ of $\mathbf{r}_d(\xi)$ and its first and second derivatives can then be expressed as

$$\sigma_d = (1 + \kappa d) \sigma, \quad \sigma'_d = (1 + \kappa d) \sigma' + \sigma^2 d \frac{d\kappa}{ds}, \quad (19)$$

$$\sigma''_d = (1 + \kappa d) \sigma'' + 3\sigma\sigma' d \frac{d\kappa}{ds} + \sigma^3 d \frac{d^2\kappa}{ds^2}, \quad (20)$$

where the arc-length derivatives of the curvature are given by

$$\frac{d\kappa}{ds} = \frac{(\mathbf{r}' \times \mathbf{r}''') \cdot \mathbf{z} - 3\sigma^2\sigma'\kappa}{\sigma^4}, \quad (21)$$

$$\frac{d^2\kappa}{ds^2} = \frac{(\mathbf{r}'' \times \mathbf{r}''' + \mathbf{r}' \times \mathbf{r}'''') \cdot \mathbf{z} - 3\sigma(2\sigma'^2 + \sigma\sigma'')\kappa - 7\sigma^3\sigma'(d\kappa/ds)}{\sigma^5}. \quad (22)$$

To obtain the derivatives $\dot{\xi}$, $\ddot{\xi}$, $\dddot{\xi}$ appropriate to traversal of the offset curve at feedrate V , the above expressions for $\sigma_d, \sigma'_d, \sigma''_d$ must be substituted in lieu of $\sigma, \sigma', \sigma''$ in (4). Moreover, if V is not constant, the appropriate expressions for V', V'' must be obtained — equations (15), for example, in the case of an arc-length-dependent feedrate $V(s)$.

Figure 4 illustrates a sub-sampling of reference points along the offset at distance $d = 0.5$ in to a quintic curve $\mathbf{r}(\xi)$. A primary challenge for the real-time interpolator in executing the offset path $\mathbf{r}_d(\xi)$ is the extreme variation of its parametric speed $\sigma_d(\xi) = [1 + \kappa(\xi)d]\sigma(\xi)$, as can be seen in Figure 4. To emphasize this aspect, we consider a traversal at the constant feedrate $V_0 = 400$ in/min, with a sampling interval $\Delta t = 0.001$ s.

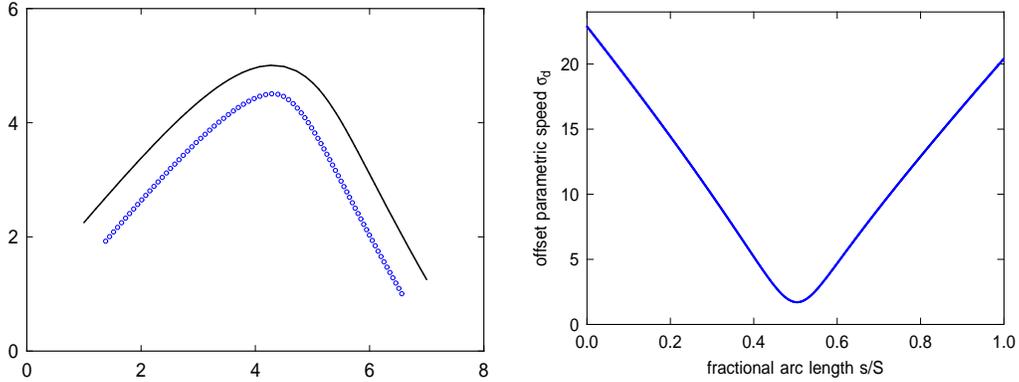


Figure 4: Left: sub-sampled distribution of reference points along the offset (16) to a quintic test curve with $d = 0.5$ in and feedrate $V_0 = 400$ in/min. Right: extreme variation of the parametric speed $\sigma_d(\xi)$ along the offset curve.

Along the offset path, Figure 5 shows the relative errors in the Richardson extrapolation estimates for $\dot{\xi}_k(t_i)$ and $\ddot{\xi}_k(t_i)$ for the orders $k = 3$ and $k = 5$. With $k = 3$, the difficulty of achieving high accuracy near the “sharp turn” in the offset path is evident. With $k = 5$, however, this problem is largely subdued and an accuracy of $\lesssim 10^{-10}$ is obtained along the entire offset curve.

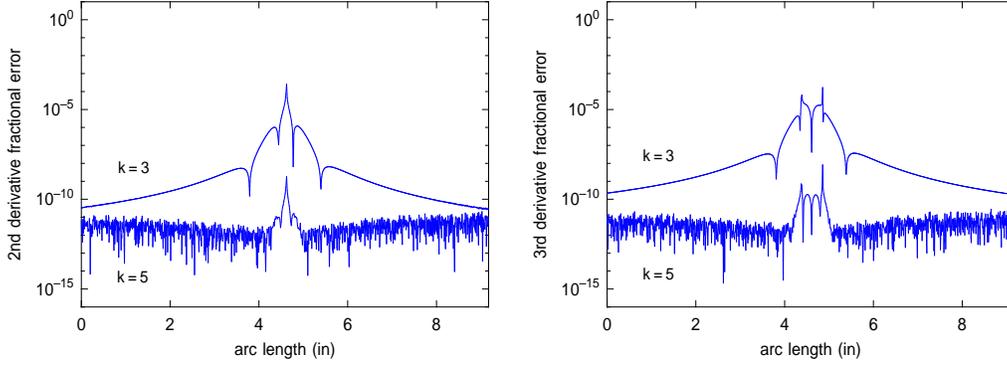


Figure 5: Relative errors in the second derivative estimates $\ddot{\xi}_3, \ddot{\xi}_5$ (left) and third derivative estimates $\dddot{\xi}_3, \dddot{\xi}_5$ (right) in the traversal of the offset to the quintic curve shown in Figure 4 with $d = 0.5$ in and feedrate $V_0 = 400$ in/min.

	closed form	Richardson $k = 3$	Richardson $k = 5$
$\ddot{\xi}(t_i)$	2.22×10^{-6} s	6.21×10^{-6} s	9.36×10^{-6} s
$\dddot{\xi}(t_i)$	2.57×10^{-6} s	8.52×10^{-6} s	13.05×10^{-6} s

Table 2: Observed computation times for the derivatives $\ddot{\xi}(t_i)$ and $\dddot{\xi}(t_i)$ at each reference point along the offset curve in Figure 4 traversed with constant feedrate $V = 400$ in/min, employing the closed-form derivative expressions and Richardson extrapolations of order $k = 3$ and 5 (with a 1.1 GHz cpu).

To estimate computational cost, the offset curve was executed 100 times, with each execution requiring 1232 reference–point for a sampling interval $\Delta t = 0.001$ s. The average times required to compute the derivatives $\ddot{\xi}(t_i)$ and $\ddot{\xi}(t_i)$ for each reference point, employing both the closed–form expressions and the Richardson extrapolations $\ddot{\xi}_3(t_i)$ and $\ddot{\xi}_5(t_i)$, using a modest 1.1 GHz cpu, are indicated in Table 2. The results are in close agreement with those in Table 1 for the corner curve, and are consistent with a linear dependence of the computation time on the extrapolation order k (they are $\lesssim 1\%$ of the adopted sampling interval $\Delta t = 0.001$ s). Although Richardson extrapolation is somewhat more expensive, it avoids the need for the extensive derivations embodied in equations (4), (15), and (19)–(22).

6 Curvature-dependent feedrate

Consider the execution of a path specified by a parametric curve $\mathbf{r}(\xi)$ with a feedrate V dependent on the path⁴ curvature (17). Specifically, we use the curvature–dependent feedrate

$$V(\kappa) = \frac{V_0}{1 + (\kappa/\kappa_0)^2}, \quad (23)$$

for which $V = V_0$ represents a nominal feedrate when $\kappa = 0$; $V = \frac{1}{2}V_0$ when $\kappa = \kappa_0$; and V diminishes rapidly for $|\kappa/\kappa_0| \gg 1$. The feedrate (23) can be employed to mitigate large axis accelerations incurred when paths with high localized curvatures are to be executed at nominally rapid feedrates, to ensure that the drive motors can supply the required torque [12].

The feedrate parametric derivatives V', V'', \dots in (4) must be determined in terms of the function (23). This can be achieved by invoking the chain rule to write

$$\frac{d}{d\xi} = \sigma \frac{d\kappa}{ds} \frac{d}{d\kappa}.$$

By successive applications of this differential operator to $V(\kappa)$ we obtain

$$V' = \sigma \frac{d\kappa}{ds} \frac{dV}{d\kappa}, \quad V'' = \sigma' \frac{d\kappa}{ds} \frac{dV}{d\kappa} + \sigma^2 \frac{d^2\kappa}{ds^2} \frac{dV}{d\kappa} + \sigma^2 \left(\frac{d\kappa}{ds} \right)^2 \frac{d^2V}{d\kappa^2}, \quad (24)$$

⁴For brevity, we focus on planar paths. However, the methodology also applies to spatial paths, for which the curvature is defined as the non–negative quantity $\kappa = |\mathbf{r}' \times \mathbf{r}''|/\sigma^3$.

etc., where the arc-length derivatives of the curvature are given by (21)–(22), and for the feedrate function (23) we have

$$\frac{dV}{d\kappa} = -\frac{2V_0\kappa}{\kappa_0^2[1+(\kappa/\kappa_0)^2]^2}, \quad \frac{d^2V}{d\kappa^2} = \frac{2V_0[3(\kappa/\kappa_0)^2-1]}{\kappa_0^2[1+(\kappa/\kappa_0)^2]^3}. \quad (25)$$

Note that $dV/d\kappa$ is opposite in sign to κ , but $d^2V/d\kappa^2$ is positive or negative according to whether $|\kappa/\kappa_0|$ is greater than or less than $1/\sqrt{3}$.

In (3), the quantities (5) and (21)–(22) are evaluated at ξ_i , while (23) and (25) are evaluated at $\kappa(\xi_i)$. Note that these expressions have been formulated recursively, to present them as compactly as possible.

Figure 6 shows a quintic test curve with strong curvature variation, used to illustrate an implementation of the feedrate function (23) with parameters $V = 400$ in/min and $\kappa_0 = 1$ in⁻¹. As can be seen in Figure 7, the Richardson extrapolation estimates for $\ddot{\xi}_k(t_i)$ and $\ddot{\xi}_k(t_i)$ are again of excellent accuracy, with relative errors $\lesssim 10^{-6}$ for $k = 3$ and $\lesssim 10^{-10}$ for $k = 5$.

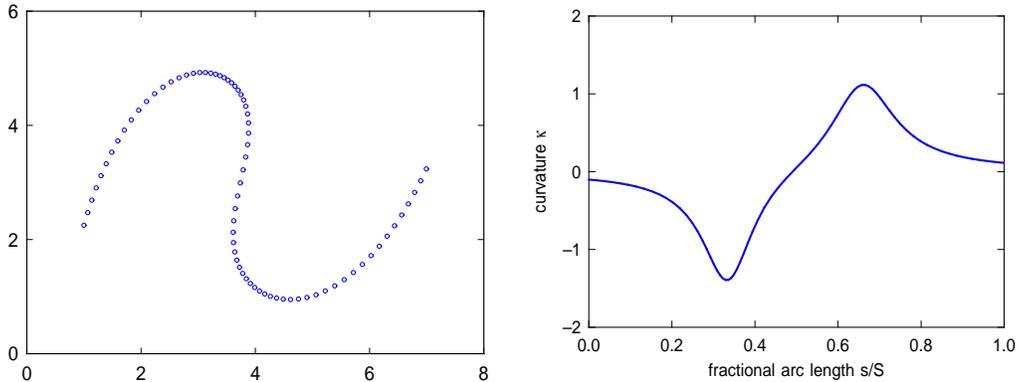


Figure 6: Left: subsampling of reference points along a test curve traversed with the curvature-dependent feedrate function (23) with $V_0 = 400$ in/min and $\kappa_0 = 1$ in⁻¹. Right: variation of the curvature along the test curve.

Table 3 lists the computational cost for the curvature-dependent feedrate (23), determined through 100 curve traversals with 2522 computed reference points each, and a sampling interval $\Delta t = 0.001$ sec. The results are in very good agreement with those enumerated in Tables 1 and 2, indicating that the computational cost of Richardson extrapolation is remarkably consistent for a variety of different feedrate variations.

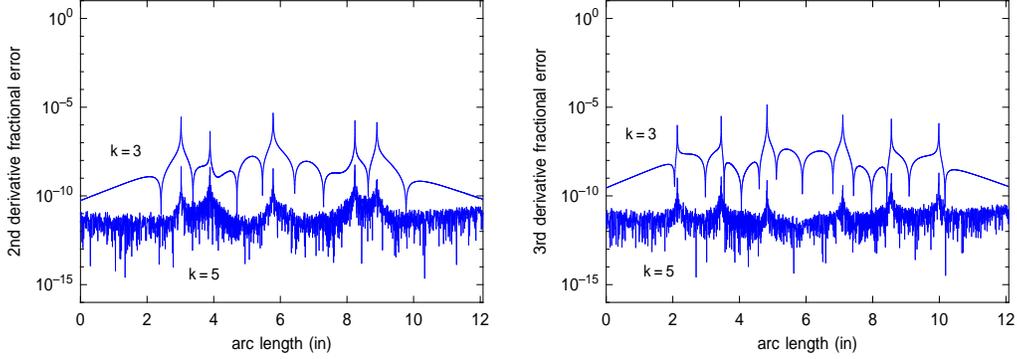


Figure 7: Relative errors in the second derivative estimates $\ddot{\xi}_3, \ddot{\xi}_5$ (left) and third derivative estimates $\dddot{\xi}_3, \dddot{\xi}_5$ (right) in traversing the test curve shown in Figure 6 using the feedrate (23) with $V_0 = 400$ in/min and $\kappa_0 = 1$ in $^{-1}$.

	closed form	Richardson $k = 3$	Richardson $k = 5$
$\ddot{\xi}(t_i)$	2.21×10^{-6} s	7.22×10^{-6} s	12.13×10^{-6} s
$\dddot{\xi}(t_i)$	2.71×10^{-6} s	8.89×10^{-6} s	13.45×10^{-6} s

Table 3: Observed computation times for the derivatives $\ddot{\xi}(t_i)$ and $\dddot{\xi}(t_i)$ at each reference point along the curve in Figure 6 traversed with the curvature-dependent feedrate (23) using both the closed-form derivative expressions, and Richardson extrapolations of order $k = 3$ and 5 (with a 1.1 GHz cpu).

Another curvature-dependent feedrate function was introduced in [6] to ensure a constant material removal rate for a fixed depth of cut δ , when the tool radius d and path radius of curvature $\rho = \kappa^{-1}$ are comparable, namely

$$V(\kappa) = \frac{V_0}{1 + \kappa(d - \frac{1}{2}\delta)}.$$

In this case, the derivatives of the feedrate function are

$$\frac{dV}{d\kappa} = -\frac{V_0(d - \frac{1}{2}\delta)}{[1 + \kappa(d - \frac{1}{2}\delta)]^2}, \quad \frac{d^2V}{d\kappa^2} = \frac{2V_0(d - \frac{1}{2}\delta)^2}{[1 + \kappa(d - \frac{1}{2}\delta)]^3},$$

and otherwise the implementation proceeds in the exactly same manner as for the feedrate (23) — the results obtained using the Richardson extrapolation method are closely analogous to those for (23).

7 Closure

The ability to accurately achieve constant or varying feedrates along curved paths plays an important role in diverse applications, such as management of accelerations or chip load variations in CNC machining, or exposure control in laser cutting and laser-based 3D printing processes. However, real-time interpolators based on truncated Taylor series expansions may incur substantial inaccuracies when only the linear term is retained, and the determination of exact closed-form expressions for the coefficients of higher-order terms is an increasingly cumbersome and potentially error-prone process.

The Richardson extrapolation method circumvents the need to explicitly obtain the complicated closed-form expressions for higher-order derivatives, by allowing a rapidly-convergent sequence of estimates for their values to be computed using elementary pre-determined formulae. Tests on a variety of curved path geometries and feedrate variations have shown that it is capable of achieving very high accuracy with a computational cost easily compatible with modern cpu processor speeds and millisecond sampling intervals.

It is expected that the Richardson extrapolation method can prove useful in contexts beyond the simple demonstrative cases described above. Suppose, for example, that it is desired to minimize cutting force fluctuations incurred by variations in workpiece specific cutting energy, depth or width of cut, etc. If the cutting force at regularly-spaced path distances can be determined by

appropriate simulation software, these data can be fit using a suitable (e.g., spline) function, from which the feedrate modulation required to suppress the cutting force fluctuations can be determined. The Richardson extrapolation method then provides a simple means to implement a real-time interpolator that accurately realizes this required feedrate variation.

References

- [1] C. Brezinski and M. Redivo Zaglia (1991), *Extrapolation Methods: Theory and Practice*, North Holland, Amsterdam.
- [2] J.-J. Chou and D. C. H. Yang (1991), Command generation for three-axis CNC machining, *ASME Journal of Engineering for Industry* **113** (August), 305–310.
- [3] J.-J. Chou and D. C. H. Yang (1992), On the generation of coordinated motion of five-axis CNC/CMM machines, *ASME Journal of Engineering for Industry* **114** (February), 15–22.
- [4] G. Farin (1997), *Curves and Surfaces for Computer Aided Geometric Design* (4th Edition), Academic Press, San Diego.
- [5] R. T. Farouki (2008), *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*, Springer, Berlin.
- [6] R. T. Farouki, J. Manjunathaiah, D. Nicholas, G.-F. Yuan, and S. Jee (1998), Variable feedrate CNC interpolators for constant material removal rates along Pythagorean-hodograph curves, *Computer Aided Design* **30**, 631–640.
- [7] R. T. Farouki, J. Manjunathaiah, and G.-F. Yuan (1999), G codes for the specification of Pythagorean-hodograph tool paths and associated feedrate functions on open-architecture CNC machines, *International Journal of Machine Tools and Manufacture* **39**, 123–142.
- [8] R. T. Farouki and C. A. Neff (1990), Analytic properties of plane offset curves, *Computer Aided Geometric Design* **7**, 83–99.
- [9] R. T. Farouki and K. M. Nittler (2016), Efficient high-speed cornering motions based on continuously-variable feedrates I. Real-time

- interpolator algorithms, *International Journal of Advanced Manufacturing Technology* **87**, 3557–3568.
- [10] R. T. Farouki and S. Shah (1996), Real-time CNC interpolators for Pythagorean-hodograph curves, *Computer Aided Geometric Design* **13**, 583–600.
- [11] R. T. Farouki and Y-F. Tsai (2001), Exact Taylor series coefficients for variable-feedrate CNC curve interpolators, *Computer Aided Design* **33**, 155–165.
- [12] R. T. Farouki, Y-F. Tsai, and C. S. Wilson (2000), Physical constraints on feedrates and feed accelerations along curved tool paths, *Computer Aided Geometric Design* **17**, 337–359.
- [13] R. T. Farouki, Y-F. Tsai, and G-F. Yuan (1998), Contour machining of free-form surfaces with real-time PH curve CNC interpolators, *Computer Aided Geometric Design* **16**, 61–76.
- [14] J.-T. Huang and D. C. H. Yang (1992), A generalized interpolator for command generation of parametric curves in computer-controlled machines, Proceedings of the *Japan/USA Symposium on Flexible Automation*, Vol. 1, ASME, 393–399.
- [15] R. Komanduri, K. Subramanian, and B. F. von Turkovich (eds.) (1984), *High Speed Machining*, PED-Vol. 12, ASME, New York.
- [16] R-S. Lin and Y. Koren (1996), Real-time interpolators for multi-axis CNC machine tools, *Manufacturing Systems* **25**, 145–149.
- [17] K. M. Nittler and R. T. Farouki (2017), Efficient high-speed cornering motions based on continuously-variable feedrates II. Experimental performance analysis, *International Journal of Advanced Manufacturing Technology* **88**, 159–174.
- [18] A. Shima, T. Sasaki, T. Ohtsuki, and Y. Wakimoto (1996), 64-bit RISC-based Series 15 NURBS interpolation, *FANUC Technical Review* **9** (1), 23–28.
- [19] M. Shpitalni, Y. Koren, and C. C. Lo (1994), Realtime curve interpolators, *Computer Aided Design* **26**, 832–838.

- [20] A. Sidi (2003), *Practical Extrapolation Methods: Theory and Applications*, Cambridge University Press.
- [21] S. Smith and J. Tlusty (1997), Current trends in high-speed machining, *ASME Journal of Manufacturing Science and Engineering* **119**, 664–666.
- [22] J. Tlusty (1993), High-speed machining, *CIRP Annals* **42**, 733–738.
- [23] Y-F. Tsai, R. T. Farouki, and B. Feldman (2000), Performance analysis of CNC interpolators for time-dependent feedrates along PH curves, *Computer Aided Geometric Design*, submitted.
- [24] D. C. H. Yang and T. Kong (1994), Parametric interpolator versus linear interpolator for precision CNC machining, *Computer Aided Design* **26**, 225–234.
- [25] S-S. Yeh and P-L. Hsu (1999), The speed-controlled interpolator for machining parametric curves, *Computer Aided Design* **31**, 349–357.