



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

Identification and "reverse engineering" of Pythagorean-hodograph curves

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Identification and "reverse engineering" of Pythagorean-hodograph curves / Farouki Rida T.; Giannelli Carlotta; Sestini Alessandra. - In: COMPUTER AIDED GEOMETRIC DESIGN. - ISSN 0167-8396. - STAMPA. - 34:(2015), pp. 21-36. [10.1016/j.cagd.2015.04.001]

Availability:

This version is available at: 2158/1003421 since: 2021-04-12T10:26:04Z

Published version:

DOI: 10.1016/j.cagd.2015.04.001

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

(Article begins on next page)

Identification and “reverse engineering” of Pythagorean–hodograph curves

Rida T. Farouki

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

Carlotta Giannelli

Istituto Nazionale di Alta Matematica,
Unità di Ricerca di Firenze c/o DiMaI “U. Dini,”
Università di Firenze, Viale Morgagni 67/A, 50134 Firenze, ITALY.

Alessandra Sestini

Dipartimento di Matematica e Informatica “U. Dini,”
Università degli Studi di Firenze,
Viale Morgagni 67a, 50134 Firenze, ITALY.

Abstract

Methods are developed to identify whether or not a given polynomial curve, specified by Bézier control points, is a Pythagorean–hodograph (PH) curve — and, if so, to reconstruct the internal algebraic structure that allows one to exploit the advantageous properties of PH curves. Two approaches to identification of PH curves are proposed. The first is based on the satisfaction of a system of algebraic constraints by the control–polygon legs, and the second uses the fact that numerical quadrature rules that are exact for polynomials of a certain maximum degree generate arc length estimates for PH curves exhibiting a sharp saturation as the number of sample points is increased. These methods are equally applicable to planar and spatial PH curves, and are fully elaborated for cubic and quintic PH curves. The reverse engineering problem involves computing the complex or quaternion coefficients of the pre–image polynomials generating planar or spatial Pythagorean

hodographs, respectively, from prescribed Bézier control points. In the planar case, a simple closed-form solution is possible, but for spatial PH curves the reverse engineering problem is much more involved.

Keywords: Bézier control points; Pythagorean-hodograph curves; arc length; parametric speed; numerical quadrature; reverse engineering.

e-mail: farouki@ucdavis.edu, carlotta.giannelli@unifi.it, alessandra.sestini@unifi.it

1 Introduction

The phrase “reverse engineering” typically refers to the recovery of lost design specifications for an artifact, from its physical realization. Since this endeavor inevitably incurs some loss of fidelity to the original design intent, it is always preferable to invoke original plans or CAD models when possible. There are, nevertheless, circumstances in which this is impractical, and one must appeal to a reverse engineering process to reconstruct — as faithfully as possible — the original underlying design intent of a given physical artifact.

Most modern CAD systems rely on the Bézier/B-spline representation to specify “free-form” curves and surfaces. Although this provides considerable design flexibility, and compatibility with commonly-used “simple” shapes, it necessitates the use of approximations for many basic geometrical functions, such as computation of arc lengths, offset curves, and certain rational frames on space curves. The Pythagorean-hodograph (PH) curves circumvent these problems by incorporating a special algebraic structure [6], while remaining fully compatible with the standard Bézier/B-spline representation. However, to exploit the advantageous features of PH curves, their “internal structure” variables (which are not directly available from their Bézier/B-spline forms) must be known. Specifically, planar and spatial Pythagorean hodographs are constructed by quadratic mappings of complex and quaternion polynomials, respectively, and the coefficients of these pre-image polynomials are required for the exact determination of various PH curve properties.

Many methods for the construction of planar and spatial PH curves are available [8, 12, 14, 20, 21, 22, 23, 24, 25, 27, 29, 30]. The goal of this study is to facilitate their importation into commercial CAD systems through existing CAD data formats, by developing algorithms that (i) identify whether or not specified Bézier/B-spline data define a PH curve; and (ii) if so, reconstruct its “internal structure” variables. Since the polynomial PH curves are a proper subset of all polynomial curves, a randomly-chosen polynomial curve has a negligible probability of being a PH curve: the proposed methods make sense only if one has *a priori* reason to suspect that given data define a PH curve. Moreover, for use in “real” applications, the identification and reconstruction of PH curves must be formulated in floating-point arithmetic, and can thus be solved only within a given tolerance (which in practice can be very small). For purely academic investigations, one may consider only curves defined by rational coefficients, and invoke symbolic computation methods.

The algorithms described herein may be categorized as *identification* and

reconstruction schemes, the latter being invoked only when the former yields an affirmative outcome. The focus will be on the polynomial PH curves, since rational PH curves lack some features of polynomial PH curves required in these methods. Also, for brevity, the input for the algorithms is assumed to be Bézier control points for a single segment (which may be readily extracted [10] from the B-spline representation of a multi-segment spline).

The remainder of this paper is organized as follows. After a brief review of some basic properties of PH curves in Section 2, two different schemes for determining whether or not a given set of Bézier control points specify a PH curve are described in Section 3 — the method appropriate to a particular application context may be selected on the basis of computational efficiency and any desired additional information concerning an identified PH curve. The identification algorithms do not distinguish between planar and spatial PH curves, but the “reverse engineering” algorithms described in Section 4 require individual treatment of these two cases. Specifically, the hodographs for planar and spatial PH curves are generated from complex-number and quaternion “pre-image” polynomials, respectively, whose coefficients must be determined. Finally, Section 5 summarizes the key results obtained herein, and concludes with some comments on their practical use.

2 Pythagorean-hodograph curves

A degree- n polynomial curve $\mathbf{r}(t)$, $t \in [0, 1]$ specified in terms of its Bézier control points $\mathbf{p}_0, \dots, \mathbf{p}_n$ as

$$\mathbf{r}(t) = \sum_{k=0}^n \mathbf{p}_k b_k^n(t), \quad b_k^n(t) = \binom{n}{k} (1-t)^{n-k} t^k, \quad (1)$$

is a Pythagorean-hodograph (PH) curve if and only if its derivative satisfies

$$|\mathbf{r}'(t)|^2 = \sigma^2(t) \quad (2)$$

for some polynomial $\sigma(t)$ of degree $n-1$. We consider here only PH curves of *odd* degree, since *even* degree curves may exhibit singular points, satisfying $\sigma(t) = 0$ for real values of t . The non-negative function

$$\sigma(t) = |\mathbf{r}'(t)| = \frac{ds}{dt}, \quad (3)$$

defining the rate of change of arc length s with respect to the parameter t , is called the *parametric speed* of the curve $\mathbf{r}(t)$. Thus, the distinctive property of a PH curve is that its parametric speed is a polynomial in t .

For planar PH curves, the satisfaction of (2) is typically achieved through a complex-variable model [3], in which plane coordinates (x, y) are regarded as the real and imaginary parts of a complex variable $\mathbf{z} = x + iy$. Then a sufficient-and-necessary condition for satisfaction of (2) is that the derivative should be of the form

$$\mathbf{r}'(t) = \mathbf{w}^2(t) \quad (4)$$

for some complex polynomial $\mathbf{w}(t) = u(t) + iv(t)$, and the parametric speed is $\sigma(t) = |\mathbf{w}(t)|^2$. For spatial PH curves, on the other hand, a quaternion model is employed [1, 11] with the derivative expressed in the form

$$\mathbf{r}'(t) = \mathcal{A}(t) \mathbf{i} \mathcal{A}^*(t), \quad (5)$$

in terms of a quaternion polynomial¹ $\mathcal{A}(t) = u(t) + v(t) \mathbf{i} + p(t) \mathbf{j} + q(t) \mathbf{k}$ and its conjugate $\mathcal{A}^*(t) = u(t) - v(t) \mathbf{i} - p(t) \mathbf{j} - q(t) \mathbf{k}$. The parametric speed of the PH curve defined by (5) is $\sigma(t) = |\mathcal{A}(t)|^2 = u^2(t) + v^2(t) + p^2(t) + q^2(t)$. If $\mathbf{w}(t)$ and $\mathcal{A}(t)$ in expressions (4) and (5) are of degree m , their integration yields planar and spatial PH curves of degree $n = 2m + 1$. Complete details on these formulations, and the properties of PH curves, may be found in [6].

In the present context, we assume that a Bézier curve of the form (1) is given (i.e., its control points $\mathbf{p}_0, \dots, \mathbf{p}_n$ are specified) and we wish to ascertain whether this curve results from an integration of (4) or (5) — and, if so, to reconstruct the complex polynomial $\mathbf{w}(t)$ or quaternion polynomial $\mathcal{A}(t)$ that generates $\mathbf{r}'(t)$. Since the derivative of $\mathbf{r}(t)$ can be expressed [2] as

$$\mathbf{r}'(t) = \sum_{k=0}^{n-1} \mathbf{d}_k b_k^{n-1}(t), \quad \mathbf{d}_k = n(\mathbf{p}_{k+1} - \mathbf{p}_k), \quad (6)$$

we have

$$|\mathbf{r}'(t)|^2 = \sum_{k=0}^{2(n-1)} a_k b_k^{2(n-1)}(t), \quad (7)$$

¹Calligraphic characters such as \mathcal{A} denote quaternions, which are regarded as consisting of a scalar part a and vector part \mathbf{a} , so that $\mathcal{A} = (a, \mathbf{a})$ and $a = \text{scal}(\mathcal{A})$, $\mathbf{a} = \text{vect}(\mathcal{A})$.

where, using the arithmetic procedures for Bernstein-form polynomials [16], we define

$$a_k = \sum_{j=\max(0,k-n+1)}^{\min(n-1,k)} \frac{\binom{n-1}{j} \binom{n-1}{k-j}}{\binom{2n-2}{k}} \mathbf{d}_j \cdot \mathbf{d}_{k-j}. \quad (8)$$

3 PH curve identification schemes

In the context of exact symbolic computation, given a polynomial curve $\mathbf{r}(t)$ with rational coefficients, the factorization of $|\mathbf{r}'(t)|^2$ into a perfect square can be used to determine whether it has a Pythagorean hodograph. For regular curves with $|\mathbf{r}'(t)| \neq 0$ for all real t , such a factorization is equivalent to the property that the roots of the polynomial $|\mathbf{r}'(t)|^2$ occur as complex-conjugate root pairs of even multiplicity. However, methods to compute multiple roots are typically slow to converge and sensitive to rounding errors. Experiments with numerical root-finding algorithms indicate that attempting to compute the root structure of $|\mathbf{r}'(t)|^2$ is not a viable identification scheme, compared to the methods proposed below, in terms of efficiency and robustness.

The emphasis here is on practical methods, amenable to fast and reliable identification of PH curves in floating-point arithmetic, based on the Bézier control points $\mathbf{p}_0, \dots, \mathbf{p}_n$ in (1) as input. Two methods are described below, that require only arithmetic operations on the control points and have been found to yield reliable identifications of PH quintics with a relative precision of 10^{-15} to 10^{-14} in standard double-precision arithmetic. The first is based on evaluating certain expressions in the control polygon legs, and the second exploits the fact that numerical quadrature arc-length estimates of increasing order exhibit a sharp saturation when expression (1) defines a PH curve.

3.1 Identification by control polygon

If the control points in (1) specify a PH curve, the expression defined by (7) and (8) must coincide with the perfect square of some polynomial

$$\sigma(t) = \sum_{k=0}^{n-1} \sigma_k b_k^{n-1}(t) \quad (9)$$

of degree $n-1$. Then $\sigma^2(t)$ is a polynomial of degree $2(n-1)$, whose Bernstein coefficients are given by expression (8), with $\mathbf{d}_j \cdot \mathbf{d}_{k-j}$ replaced by $\sigma_j \sigma_{k-j}$. Hence, satisfaction of the PH condition (2) is equivalent to the existence of real values $\sigma_0, \dots, \sigma_{n-1}$ such that $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ satisfy the system of equations defined for $k = 0, \dots, 2n-2$ by

$$\sum_{j=\max(0, k-n+1)}^{\min(n-1, k)} \frac{\binom{n-1}{j} \binom{n-1}{k-j}}{\binom{2n-2}{k}} (\mathbf{d}_j \cdot \mathbf{d}_{k-j} - \sigma_j \sigma_{k-j}) = 0. \quad (10)$$

We assume $\mathbf{r}(t)$ has a proper parameterization, i.e., $\sigma(t) > 0$ for $t \in [0, 1]$ — this means, in particular, that $\sigma(0) = \sigma_0 > 0$ and $\sigma(1) = \sigma_{n-1} > 0$.

The coefficients \mathbf{d}_k of the derivative $\mathbf{r}'(t)$ differ from the control-polygon legs $\Delta \mathbf{p}_k = \mathbf{p}_{k+1} - \mathbf{p}_k$ only by the factor n , so for brevity we often simply refer to $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ as the control-polygon legs. The conditions on $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ identifying PH curves derived below are homogeneous, and remain valid if we substitute $\Delta \mathbf{p}_0, \dots, \Delta \mathbf{p}_{n-1}$. By eliminating the n indeterminate quantities $\sigma_0, \dots, \sigma_{n-1}$ from the system (10) of $2n-1$ equations, one obtains a system of $n-1$ constraints on the control-polygon legs $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ of a degree- n Bézier curve $\mathbf{r}(t)$, whose satisfaction identifies it as a PH curve. Note that, whereas planar and spatial PH curve *constructions* employ distinct algebraic models, this *identification* scheme for PH curves is independent of the dimension of $\mathbf{r}(t)$. We now consider the reduction of the system (10) in the context of PH cubics and quintics, setting $d_k = |\mathbf{d}_k|$ for brevity.

Proposition 1. *A planar or spatial cubic Bézier curve with control polygon legs $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2$ has a Pythagorean hodograph if and only if it satisfies*

$$d_0 \mathbf{d}_1 \cdot \mathbf{d}_2 = d_2 \mathbf{d}_0 \cdot \mathbf{d}_1, \quad (11)$$

$$2(\mathbf{d}_0 \cdot \mathbf{d}_1)(\mathbf{d}_1 \cdot \mathbf{d}_2) = d_0 d_2 (\mathbf{d}_0 \cdot \mathbf{d}_2 - d_0 d_2 + 2 d_1^2). \quad (12)$$

Proof : For a cubic curve, the conditions (10) become

$$\begin{aligned} d_0^2 &= \sigma_0^2, & \mathbf{d}_0 \cdot \mathbf{d}_1 &= \sigma_0 \sigma_1, \\ \mathbf{d}_0 \cdot \mathbf{d}_2 + 2 d_1^2 &= \sigma_0 \sigma_2 + 2 \sigma_1^2, \\ \mathbf{d}_1 \cdot \mathbf{d}_2 &= \sigma_1 \sigma_2, & d_2^2 &= \sigma_2^2. \end{aligned} \quad (13)$$

By eliminating $\sigma_0, \sigma_1, \sigma_2$ from these five equations, the condition for a cubic to be a PH curve is reducible to two scalar constraints on the control-polygon legs $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2$. Since $\sigma_0, \sigma_2 > 0$ for a regular curve, we must have $\sigma_0 = d_0 \neq 0$ and $\sigma_2 = d_2 \neq 0$ from the first and last of equations (13). We also assume that $d_1 \neq 0$, since setting $\mathbf{d}_1 = \mathbf{0}$ in (13) gives $\mathbf{d}_0 \cdot \mathbf{d}_2 = d_0 d_2$, so $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2$ are linearly dependent and the curve degenerates to a straight line. With $\sigma_0 = d_0$ and $\sigma_2 = d_2$, the second and fourth equations give $\sigma_1 = \mathbf{d}_0 \cdot \mathbf{d}_1 / d_0 = \mathbf{d}_1 \cdot \mathbf{d}_2 / d_2$, and these expressions are consistent if and only if (11) holds. Substituting for $\sigma_0, \sigma_1, \sigma_2$ into the third equation then gives condition (12). ■

The conditions of Proposition 1 are consistent with the characterizations for planar and spatial PH cubics, in terms of their Bézier control polygons, derived in [17] and [18]. For the planar case, it was shown in [17] that a cubic is a PH curve if and only if the lengths d_0, d_1, d_2 of the control-polygon legs and the angles θ_1, θ_2 at the interior vertices (see Figure 1) satisfy

$$d_1 = \sqrt{d_0 d_2} \quad \text{and} \quad \theta_1 = \theta_2. \quad (14)$$

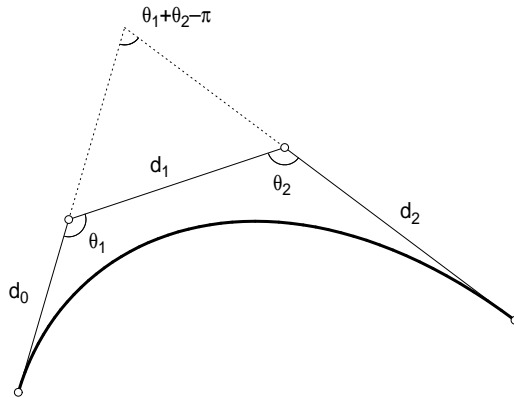


Figure 1: The geometrical parameters d_0, d_1, d_2 and θ_1, θ_2 that characterize the Bézier control polygon of a planar PH cubic through the conditions (14).

Now by the definition of the angles θ_1, θ_2 we have $\mathbf{d}_0 \cdot \mathbf{d}_1 = -d_0 d_1 \cos \theta_1$ and $\mathbf{d}_1 \cdot \mathbf{d}_2 = -d_1 d_2 \cos \theta_2$. Substituting into (11) then implies that $\cos \theta_2 = \cos \theta_1$ and hence $\sin \theta_2 = \pm \sin \theta_1$, so we have either $\theta_2 = \theta_1$ or $\theta_2 = 2\pi - \theta_1$. Since (the extensions of) \mathbf{d}_0 and \mathbf{d}_2 make an interior angle (see Figure 1) of

$\theta_1 + \theta_2 - \pi$, we obtain

$$\begin{aligned}\mathbf{d}_0 \cdot \mathbf{d}_2 &= -d_0 d_2 \cos(\theta_1 + \theta_2 - \pi) = d_0 d_2 \cos(\theta_1 + \theta_2) \\ &= d_0 d_2 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2).\end{aligned}$$

Substituting for $\mathbf{d}_0 \cdot \mathbf{d}_1$, $\mathbf{d}_1 \cdot \mathbf{d}_2$, $\mathbf{d}_0 \cdot \mathbf{d}_2$ into (12) and simplifying then gives

$$2d_1^2(\cos \theta_1 \cos \theta_2 - 1) = d_0 d_2 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 - 1).$$

Now if $\theta_2 = \theta_1$, this reduces to $d_1^2 = d_0 d_2$, so the conditions (14) are satisfied. On the other hand, when $\theta_2 = 2\pi - \theta_1$ it reduces to $2d_1^2 \sin^2 \theta_1 = 0$, implying that (i) $d_1 = 0$, or (ii) θ_1, θ_2 are integer multiples of π . As noted above, case (i) identifies degeneration to a straight line, and this is clearly also true for case (ii). Hence, for a proper PH cubic, the conditions (13) are satisfied.

For the spatial case, it was shown in [18] that a twisted cubic is a PH curve if and only if \mathbf{d}_0 and \mathbf{d}_2 lie on a right-circular cone of some half-angle ϑ about \mathbf{d}_1 as axis, with azimuthal separation φ on this cone given by

$$\cos \varphi = 1 - \frac{2d_1^2}{d_0 d_2}. \quad (15)$$

In other words, in a coordinate system in which \mathbf{d}_1 is parallel to the positive z direction and \mathbf{d}_0 is parallel to the (z, x) plane, we must have

$$\begin{aligned}\mathbf{d}_0 &= d_0(\sin \vartheta, 0, \cos \vartheta), & \mathbf{d}_1 &= d_1(0, 0, 1), \\ \mathbf{d}_2 &= d_2(\sin \vartheta \cos \varphi, \sin \vartheta \sin \varphi, \cos \vartheta).\end{aligned} \quad (16)$$

Now writing $\mathbf{d}_0 \cdot \mathbf{d}_1 = d_0 d_1 \cos \theta_{01}$ and $\mathbf{d}_1 \cdot \mathbf{d}_2 = d_1 d_2 \cos \theta_{12}$, the condition (11) implies that $\cos \theta_{01} = \cos \theta_{12}$ ($= \cos \vartheta$, say) for non-zero d_0, d_1, d_2 . This means that $\mathbf{d}_0, \mathbf{d}_2$ lie on a cone of half-angle ϑ with \mathbf{d}_1 as axis. If φ is their azimuthal separation on this cone, substituting from (16) into (12) gives

$$2d_0 d_1^2 d_2 \cos^2 \vartheta = d_0 d_2 [d_0 d_2 (\sin^2 \vartheta \cos \varphi + \cos^2 \vartheta) - d_0 d_2 + 2d_1^2],$$

and upon simplification this becomes

$$d_0 d_2 \sin^2 \vartheta \cos \varphi = d_0 d_2 \sin^2 \vartheta - 2d_1^2 \sin^2 \vartheta.$$

Noting that d_1 and d_2 are non-zero, and assuming that $\sin \vartheta \neq 0$ (since this identifies the degenerate case in which the control-polygon legs are collinear), this implies that the condition (15) is satisfied.

Proposition 2. *A planar or spatial quintic Bézier curve with control polygon legs $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4$ has a Pythagorean hodograph if and only if it satisfies*

$$3d_0^2d_4^2(d_4\mathbf{d}_0 - d_0\mathbf{d}_4) \cdot \mathbf{d}_2 = 4d_0^3|\mathbf{d}_3 \times \mathbf{d}_4|^2 - 4d_4^3|\mathbf{d}_0 \times \mathbf{d}_1|^2, \quad (17)$$

$$d_0^4(d_4\mathbf{d}_0 - d_0\mathbf{d}_4) \cdot \mathbf{d}_3 + 6d_0^2d_4(\mathbf{d}_0 \times \mathbf{d}_1) \cdot (\mathbf{d}_0 \times \mathbf{d}_2) = 8d_4\mathbf{d}_0 \cdot \mathbf{d}_1|\mathbf{d}_0 \times \mathbf{d}_1|^2, \quad (18)$$

$$d_4^4(d_0\mathbf{d}_4 - d_4\mathbf{d}_0) \cdot \mathbf{d}_1 + 6d_4^2d_0(\mathbf{d}_2 \times \mathbf{d}_4) \cdot (\mathbf{d}_3 \times \mathbf{d}_4) = 8d_0\mathbf{d}_3 \cdot \mathbf{d}_4|\mathbf{d}_3 \times \mathbf{d}_4|^2, \quad (19)$$

$$\begin{aligned} & d_0^3d_4^3(\mathbf{d}_0 \cdot \mathbf{d}_4 - d_0d_4 + 18d_2^2) + 16d_0^2d_4^2[d_0d_4\mathbf{d}_1 \cdot \mathbf{d}_3 - (\mathbf{d}_0 \cdot \mathbf{d}_1)(\mathbf{d}_3 \cdot \mathbf{d}_4)] \\ & = 2[3d_0^2\mathbf{d}_0 \cdot \mathbf{d}_2 + 4|\mathbf{d}_0 \times \mathbf{d}_1|^2][3d_4^2\mathbf{d}_2 \cdot \mathbf{d}_4 + 4|\mathbf{d}_3 \times \mathbf{d}_4|^2]. \end{aligned} \quad (20)$$

Proof. For a quintic curve, the conditions (10) become

$$\begin{aligned} d_0^2 &= \sigma_0^2, & \mathbf{d}_0 \cdot \mathbf{d}_1 &= \sigma_0\sigma_1, \\ 3\mathbf{d}_0 \cdot \mathbf{d}_2 + 4d_1^2 &= 3\sigma_0\sigma_2 + 4\sigma_1^2, \\ \mathbf{d}_0 \cdot \mathbf{d}_3 + 6\mathbf{d}_1 \cdot \mathbf{d}_2 &= \sigma_0\sigma_3 + 6\sigma_1\sigma_2, \\ \mathbf{d}_0 \cdot \mathbf{d}_4 + 16\mathbf{d}_1 \cdot \mathbf{d}_3 + 18d_2^2 &= \sigma_0\sigma_4 + 16\sigma_1\sigma_3 + 18\sigma_2^2, \\ \mathbf{d}_1 \cdot \mathbf{d}_4 + 6\mathbf{d}_2 \cdot \mathbf{d}_3 &= \sigma_1\sigma_4 + 6\sigma_2\sigma_3, \\ 3\mathbf{d}_2 \cdot \mathbf{d}_4 + 4d_3^2 &= 3\sigma_2\sigma_4 + 4\sigma_3^2, \\ \mathbf{d}_3 \cdot \mathbf{d}_4 &= \sigma_3\sigma_4, & d_4^2 &= \sigma_4^2. \end{aligned} \quad (21)$$

Since $\sigma_0, \sigma_4 > 0$ for a regular curve, we have $\sigma_0 = d_0 \neq 0$ and $\sigma_4 = d_4 \neq 0$ from the first and ninth equations, and from the second and eighth we obtain $\sigma_1 = \mathbf{d}_0 \cdot \mathbf{d}_1/d_0$ and $\sigma_3 = \mathbf{d}_3 \cdot \mathbf{d}_4/d_4$. Using $|\mathbf{a} \times \mathbf{b}|^2 = |\mathbf{a}|^2|\mathbf{b}|^2 - (\mathbf{a} \cdot \mathbf{b})^2$, the third and seventh equations then give

$$\sigma_2 = \frac{\mathbf{d}_0 \cdot \mathbf{d}_2}{d_0} + \frac{4}{3} \frac{|\mathbf{d}_0 \times \mathbf{d}_1|^2}{d_0^3}, \quad \sigma_2 = \frac{\mathbf{d}_2 \cdot \mathbf{d}_4}{d_4} + \frac{4}{3} \frac{|\mathbf{d}_3 \times \mathbf{d}_4|^2}{d_4^3}.$$

Consistency of these expressions is equivalent to condition (17). Substituting for $\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4$ into the fourth and sixth equations, and using the relation $(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c})$, we obtain (18) and (19), while the fifth equation gives (20). ■

We emphasize again that the conditions (17)–(20) apply to both planar and spatial PH quintics. No simple and intuitive constraints on the geometry of the control polygon, analogous to those mentioned above for cubics, are

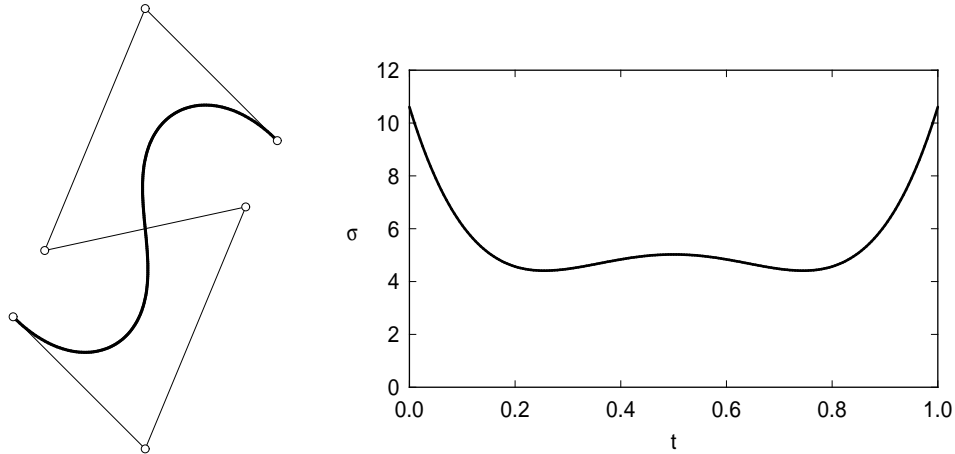


Figure 2: Left: the quintic PH curve in Example 1, together with its Bézier control polygon. Right: variation of the parametric speed $\sigma(t)$ for this curve.

available to guide the construction of PH quintics. Some partial results were given in [3], but their geometrical meaning is rather opaque — as is the case with the constraints in equations (17)–(20). Instead, PH quintics are usually specified as interpolants to first-order Hermite data.

Remark 1. In testing the conditions (11)–(12) and (17)–(20), it is advisable to first normalize the vectors \mathbf{d}_i by dividing each with their mean magnitude. This gives scale-free conditions in which both sides of (11)–(12) and (17)–(20) are of order unity, allowing direct comparison with the *machine unit* for the floating-point number system in use: for double-precision binary arithmetic with 53-bit mantissa and rounding, this is $\eta = 2^{-53} \approx 1.11 \times 10^{-16}$.

Example 1. Consider the planar PH quintic constructed as the interpolant [14] to first-order Hermite data specified by the initial and final control point pairs $\mathbf{p}_0 = (1.0, 1.0)$, $\mathbf{p}_1 = (2.5, -0.5)$ and $\mathbf{p}_4 = (2.5, 4.5)$, $\mathbf{p}_5 = (4.0, 3.0)$. To full precision, the interior two control points (computed by the interpolation algorithm) are found to be

$$\begin{aligned} \mathbf{p}_2 &= (3.6408217899592117, 2.2476669682249213), \\ \mathbf{p}_3 &= (1.3591782100407905, 1.7523330317750787). \end{aligned}$$

This curve is shown in Figure 2, together with the variation of its parametric speed $\sigma(t)$. The differences between the right and left hand sides of equations (17)–(20), computed in double precision arithmetic, are as follows²

$$5 \times 10^{-16}, \quad 4 \times 10^{-16}, \quad 4 \times 10^{-16}, \quad 18 \times 10^{-16}.$$

These are within two orders of magnitude of the machine unit η . Thus, a robust PH curve identification is obtained by choosing a relative tolerance ϵ for the satisfaction of (17)–(20) that is 10^2 – 10^3 larger than η . To illustrate the ability of the method to reject “ordinary” quintic curves, an instance of the latter is defined by leaving $\mathbf{p}_0, \mathbf{p}_1$ and $\mathbf{p}_4, \mathbf{p}_5$ unchanged, and perturbing the inner control points to $\mathbf{p}_2 = (3.6, 2.2)$, $\mathbf{p}_3 = (1.4, 1.8)$. The perturbed curve is barely distinguishable from Figure 2, but it is no longer a PH curve. In this case, the residuals obtained from equations (17)–(20) are

$$2.4 \times 10^{-15}, \quad -9.1 \times 10^{-2}, \quad -9.1 \times 10^{-2}, \quad -9.2 \times 10^{-1}.$$

We observe that equation (17) continues to yield a small residual. This is a consequence of the symmetry³ of the curve. Specifically, $d_0 = d_4$, $|\mathbf{d}_3 \times \mathbf{d}_4| = |\mathbf{d}_0 \times \mathbf{d}_1|$, and $d_4 \mathbf{d}_0 - d_0 \mathbf{d}_4 = \mathbf{0}$, and hence both sides of (17) vanish. However, equations (18)–(20) have residuals much larger than η (commensurate with the perturbations of $\mathbf{p}_2, \mathbf{p}_3$) so the data are inconsistent with a PH curve.

Example 2. To verify the reliability of the method in extreme cases, it was tested on the curve shown in Figure 3, a PH quintic with a severe curvature variation. This curve is specified by the initial and final control point pairs $\mathbf{p}_0 = (4.0, 4.0)$, $\mathbf{p}_1 = (10.0, 9.0)$ and $\mathbf{p}_4 = (6.0, 11.0)$, $\mathbf{p}_5 = (11.0, 5.0)$ with the middle pair determined by the Hermite interpolation algorithm as

$$\begin{aligned} \mathbf{p}_2 &= (5.2662184461825108, 9.1034234921021326), \\ \mathbf{p}_3 &= (9.2741575847607258, 7.5795795100404524). \end{aligned}$$

To emphasize the severity of the parametric speed variation, it is shown in Figure 3 as a function of fractional arc length along the curve, rather than the parameter t . In this case, the differences between the right and left hand sides of equations (17)–(20) are

$$27 \times 10^{-16}, \quad 18 \times 10^{-16}, \quad 5 \times 10^{-16}, \quad 0 \times 10^{-16}.$$

²Note that the normalization proposed in Remark 1 is always used.

³This symmetry also explains the equality of the residuals for equations (18) and (19).

Again, a robust PH curve identification is indicated by the fact that all these values are within two orders of magnitude of the machine unit.

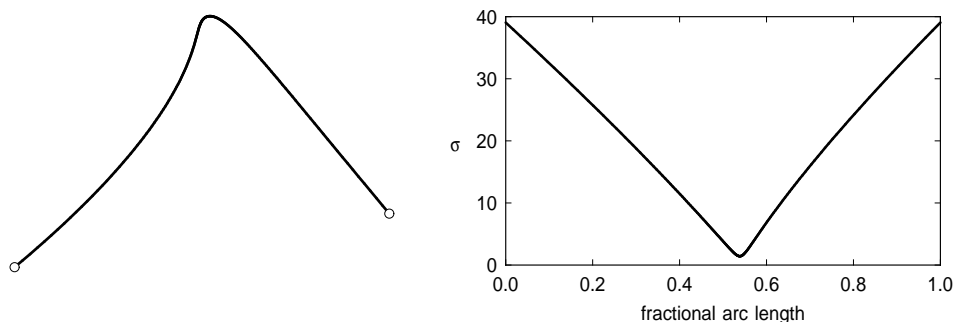


Figure 3: The PH quintic in Example 2 and its parametric speed variation.

In principle, the methods described above for PH cubics and quintics can be extended to higher-order PH curves — although the system of constraints, specified by eliminating the $\sigma_0, \dots, \sigma_{n-1}$ variables from (10), obviously grows correspondingly in number and degree. Since quintic PH curves are generally recognized to have sufficient shape flexibility for most applications, we do not pursue the extension to higher-order PH curves here.

3.2 Identification by arc length

A *quadrature rule* estimates the integral of a function $f(t)$ over an interval $t \in [a, b]$ as a weighted sum $w_1 f(t_1) + \dots + w_m f(t_m)$ of m function values, sampled at nodes $t_1, \dots, t_m \in [a, b]$. When the nodes and weights are suitably chosen, the estimate is *exact* (modulo round-off error) if $f(t)$ is a polynomial of a certain maximum degree, dependent on m . A quadrature rule has *precision* n if it yields exact integrals for polynomials of degree $\leq n$, but not $> n$.

The cumulative arc length $s(t)$ of a parametric curve is the integral of the parametric speed (3). For a general polynomial curve of degree $n \geq 3$ this integral has no closed-form reduction, but for a degree- n PH curve with the polynomial parametric speed (9) it is also a polynomial of degree n , namely

$$s(t) = \sum_{k=1}^n s_k b_k^n(t), \quad s_k = \frac{1}{n} \sum_{j=0}^{k-1} \sigma_j, \quad k = 1, \dots, n.$$

In particular, the total arc length for the parameter interval $[0, 1]$ is

$$S = s(1) = \frac{\sigma_0 + \sigma_1 + \cdots + \sigma_{n-1}}{n}. \quad (22)$$

The exactitude of a quadrature rule for sufficiently high m , when $f(t)$ is a polynomial, may be exploited to identify PH curves — if $\mathbf{r}(t)$ is a degree- n PH curve, $\sigma(t)$ is a polynomial of degree $n - 1$, and there will be no change (modulo round-off error) in the quadrature estimates of S with increasing m once it exceeds a certain threshold value. However, if $\mathbf{r}(t)$ is not a PH curve, the quadrature values will exhibit steady convergence with increasing m , but no “saturation” as with a PH curve. This approach is illustrated using two well-known quadratures, the *Gauss–Legendre* and *Newton–Cotes* rules [26].

The exactitude of a quadrature rule for sufficiently high m , when $f(t)$ is a polynomial, may be exploited to identify PH curves — if $\mathbf{r}(t)$ is a degree- n PH curve, $\sigma(t)$ is a polynomial of degree $n - 1$, and there will be no change (modulo round-off error) in the quadrature estimates of S with increasing m once it exceeds a certain threshold value. On the other hand, if $\mathbf{r}(t)$ is not a PH curve, the computed S values will exhibit steady but slow convergence with increasing m — there is no “saturation” of these values, as with a PH curve. This approach is illustrated below in the context of two well-known quadratures, the *Gauss–Legendre* and *Newton–Cotes* rules [26].

An important consideration is the numerical stability of the quadrature rule. The parametric speed $\sigma(t)$ is always a non-negative function. In Gauss–Legendre quadrature, the weights are all positive, so S is estimated as a sum of non-negative terms, and no amplification of floating-point error through the cancellation of leading digits can occur. For Newton–Cotes quadrature, on the other hand, negative weights arise for $m \geq 9$ in the closed case, and $m \geq 3$ in the open case. To preclude numerical instability, the closed rule is used only for curves of degree ≤ 7 , and the open rule is avoided altogether.

3.2.1 Gauss–Legendre quadrature

The nodes and weights for *Gauss–Legendre quadrature* are usually specified [26] for the interval $t \in [-1, +1]$. The nodes t_1, \dots, t_m are the roots of the degree- m Legendre polynomial, defined by the Rodrigues formula

$$P_m(t) = \frac{1}{2^m m!} \frac{d^m}{dt^m} (t^2 - 1)^m,$$

and the corresponding weights w_1, \dots, w_m are given by

$$w_k = \frac{2}{(1 - t_k^2) [P'_m(t_k)]^2}, \quad k = 1, \dots, m.$$

The nodes and weights for $m \leq 5$ admit closed-form expressions, as listed⁴ in Table 1. For larger m , they must be determined numerically — values for $m = 6, \dots, 10, 12$, and 16 are tabulated to 15 decimal places in [26]. The quadrature error may be formulated [26] as

$$e_m = \frac{2^{2m+1}(m!)^4}{(2m+1)[(2m)!]^3} f^{(2m)}(\xi), \quad -1 < \xi < +1.$$

Since $f^{(2m)}(t) \equiv 0$ for any polynomial of degree $d \leq 2m - 1$, the quadrature is *exact* for a polynomial integrand $f(t)$ of degree n whenever $m \geq \lceil \frac{1}{2}(n+1) \rceil$. In the present context, we adapt the Gauss–Legendre quadrature rule to the domain $[0, 1]$ by the change of variables $t \rightarrow \frac{1}{2}(t+1)$, to obtain

$$S = \int_0^1 \sigma(t) dt \approx \frac{1}{2} \sum_{k=1}^m w_k \sigma(\tfrac{1}{2}(t_k+1)), \quad (23)$$

with parametric speed values sampled at the points $\frac{1}{2}(1+t_k)$ for $k = 1, \dots, m$ using successive m values, saturation of S being observed when $m \geq \lceil \frac{1}{2}n \rceil$. With a quintic curve, for example, the quadrature arc length values S should remain essentially unchanged for $m \geq 3$, if it is a PH curve. Note that this method is again independent of whether $\mathbf{r}(t)$ is a planar or spatial curve.

Example 3. Consider the quintic PH curve constructed in Example 1, shown together with the variation of its parametric speed in Figure 2. To compare the performance of Gauss–Legendre quadrature in computing the arc length of this curve with that for an “ordinary” quintic, we again define the latter by moving the inner control points to $\mathbf{p}_2 = (3.6, 2.2)$, $\mathbf{p}_3 = (1.4, 1.8)$. For both curves, the parametric speed values required in the numerical quadrature are computed from $\sigma(t) = \sqrt{x'^2(t) + y'^2(t)}$. Table 2 compares arc lengths for the two curves, computed in double-precision arithmetic by Gauss–Legendre quadrature with $m = 1, \dots, 5$. For the PH quintic, the quadrature values

⁴Note that, in the case $m = 5$, the weight corresponding to the node $x_3 = 0$ is incorrectly quoted as 64/225 in [26].

m	nodes	weights
1	0	2
2	$\pm \sqrt{1/3}$	1
3	$0, \pm \sqrt{3/5}$	8/9, 5/9
4	$\pm \sqrt{\frac{3-2\sqrt{6/5}}{7}}, \pm \sqrt{\frac{3+2\sqrt{6/5}}{7}}$	$\frac{18+\sqrt{30}}{36}, \frac{18-\sqrt{30}}{36}$
5	$0, \pm \frac{\sqrt{5-2\sqrt{10/7}}}{3}, \pm \frac{\sqrt{5+2\sqrt{10/7}}}{3}$	$\frac{128}{225}, \frac{322+13\sqrt{70}}{900}, \frac{322-13\sqrt{70}}{900}$

Table 1: Nodes and weights for Gauss–Legendre quadrature with $1 \leq m \leq 5$.

agree with the exact arc length⁵ to a relative accuracy of $\sim 10^{-15}$ for $m \geq 3$. For the “ordinary” quintic, however, the quadrature values have an accuracy of only $\sim 10^{-3}$ at $m = 5$, although it is “very close” to the PH quintic.

m	computed arc length	m	computed arc length
1	5.026711675008204	1	5.081369156044461
2	4.507171181637951	2	4.472998552356430
3	5.458972718024720	3	5.462598411370442
4	5.458972718024721	4	5.469779178678197
5	5.458972718024720	5	5.460633553605954
exact	5.458972718024720	exact	_____

Table 2: Comparison of arc lengths computed by Gauss–Legendre quadrature using an increasing number n of sample points for (left) a quintic PH curve, and (right) an “ordinary” quintic Bézier curve — as described in Example 3.

Let S_1, S_2, \dots be arc lengths of a polynomial curve, computed in floating-point arithmetic by Gauss–Legendre quadrature with $m = 1, 2, \dots$. To identify a PH curve by agreement of these values for all $m \geq l$, it is necessary to choose a suitable convergence tolerance ϵ , such that

$$\frac{|S_m - S_l|}{|S_l|} \leq \epsilon \quad \text{for } m > l.$$

⁵For the PH curve, the exact value is given by (22), but no exact value is possible for the “ordinary” Bézier curve.

Clearly, ϵ should be larger than the machine unit η for double-precision arithmetic. Since the nodes and weights (see Table 1) are irrational numbers, and evaluating the sum (23) incurs some floating-point error, it is advisable to choose ϵ a few orders of magnitude larger than η , but at a level of accuracy that numerical quadrature cannot achieve for integration of non-polynomial functions. In practice, the value⁶ $\epsilon = 10^{-14}$ appears to work well.

If the numerical quadrature is restricted to instances with $m \leq 5$, so as to use only the nodes and weights with closed-form expressions (see Table 1), the maximum degree of PH curves that can be identified using this approach is $n = 7$, since two consecutive arc lengths that agree in value are required. The nodes and weights listed numerically to 15 decimal places in [26], up to $m = 10$, can be employed if higher-order PH curves are of interest.

Example 4. As seen in Figure 3, the curve of Example 2 exhibits a severe parametric speed variation, providing a stringent test of the method. Table 3 lists arc lengths computed in double-precision arithmetic by Gauss-Legendre quadrature. Convergence to an accuracy of $\sim 10^{-15}$ is again seen for $m \geq 3$.

m	Example 4 PH quintic	Example 5 PH cubic
1	1.553608834708754	1.2500000000000000
2	9.099750036509274	1.3333333333333333
3	11.080978828432336	1.3333333333333333
4	11.080978828432333	1.3333333333333333
5	11.080978828432333	1.3333333333333333
exact	11.080978828432336	1.3333333333333333

Table 3: Gauss-Legendre quadrature arc length values for the PH quintic in Example 4 (Figure 3) and degree-elevated PH cubic in Example 5 (Figure 4).

Example 5. A useful side-benefit of the quadrature identification procedure is that it signals, when a PH curve is detected, whether its true degree is less than the nominal degree indicated by its Bézier control-point representation. As is well-known [2], a Bézier curve of true degree n always has a non-trivial representation in the Bernstein basis of any degree $> n$. Consider the quintic

⁶More precise estimates of the (worst case) relative errors incurred in computing S by numerical quadrature can be obtained by the method of *running error analysis* [28].

Bézier curve specified by the control points

$$\mathbf{p}_0 = (0, 0), \quad \mathbf{p}_1 = (0, \frac{2}{3}), \quad \mathbf{p}_2 = (\frac{1}{10}, \frac{7}{10}),$$

$$\mathbf{p}_3 = (\frac{4}{15}, \frac{9}{10}), \quad \mathbf{p}_4 = (\frac{7}{15}, 1), \quad \mathbf{p}_5 = (\frac{2}{3}, 1).$$

Applying the Gauss–Legendre quadrature yields the arc–length values shown in Table 3. For a PH quintic, these values should agree for $m \geq 3$, but in the present case they actually agree for $m \geq 2$. This indicates that the curve is actually a degree–elevated PH cubic, rather than a true PH quintic. Figure 4 illustrates this curve, together with its Bézier control polygons of degree 5 and 3, the latter being defined by the control points

$$\mathbf{p}_0 = (0, 0), \quad \mathbf{p}_1 = (0, \frac{2}{3}), \quad \mathbf{p}_2 = (\frac{1}{3}, 1), \quad \mathbf{p}_3 = (\frac{2}{3}, 1).$$

The parametric speed has the quadratic Bernstein coefficients $(\sigma_0, \sigma_1, \sigma_2) = (2, 1, 1)$ and hence the exact arc length is $S = (2 + 1 + 1)/3 = 4/3$.

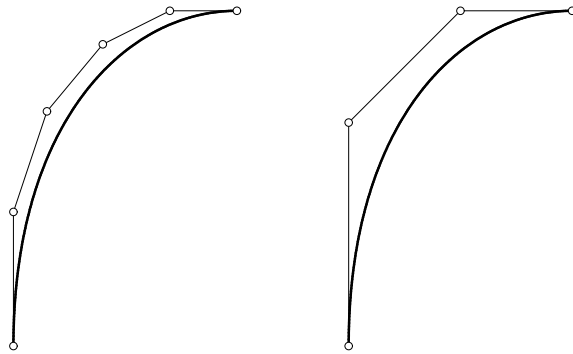


Figure 4: The PH quintic in Example 5 (left) is actually a degree–elevated PH cubic (right). The PH curve identification procedure detects this through the fact the quadrature arc lengths are in agreement for $m \geq 2$ (see Table 3).

3.2.2 Newton–Cotes quadrature

The (closed) *Newton–Cotes quadrature rule* is based on Lagrange polynomials [26] with the uniform nodes $t_k = (k - 1)/(m - 1)$, $k = 1, \dots, m$ on $t \in [0, 1]$, and involves only positive weights when $m \leq 8$ — see Table 4. The remainder term [26] is proportional to $f^{(p)}(\xi)$ for some $\xi \in [0, 1]$, where $p = 2 \lfloor \frac{1}{2}(m+1) \rfloor$.

Hence, to obtain the exact arc length of a degree- n PH curve, m must satisfy $2 \lfloor \frac{1}{2}(m+1) \rfloor \geq n$ since $\sigma(t)$ is of degree $n-1$. Thus, for a PH quintic, the arc length values saturate for $m \geq 5$ nodes with Newton-Cotes quadrature (as compared to $m \geq 3$ for Gauss-Legendre quadrature) — this is borne out by the values obtained for the curves in Examples 1 and 2, listed in Table 5.

m	weights
2	$\frac{1}{2}$
3	$\frac{1}{6}, \frac{4}{6}$
4	$\frac{1}{8}, \frac{3}{8}$
5	$\frac{7}{90}, \frac{32}{90}, \frac{12}{90}$
6	$\frac{19}{288}, \frac{75}{288}, \frac{50}{288}$
7	$\frac{41}{840}, \frac{216}{840}, \frac{27}{840}, \frac{272}{840}$
8	$\frac{751}{17280}, \frac{3577}{17280}, \frac{1323}{17280}, \frac{2989}{17280}$

Table 4: Weights w_k for closed Newton-Cotes quadrature with $m = 2, \dots, 8$ nodes. Since $w_{m-k+1} = w_k$, only the values for $k \leq \lceil \frac{1}{2}m \rceil$ are tabulated here.

Comparing the Newton-Cotes and Gauss-Legendre rules, the former has the advantage of rational nodes and weights for all m , while the latter incurs irrational nodes and weights with closed-form expressions only if $m \leq 5$. On the other hand, Gauss-Legendre quadrature permits PH curve identification with fewer nodes. Since consecutive identical arc length values are required, the minimum number of nodes to identify a PH quintic is $m = 4$ with Gauss-Legendre quadrature, but $m = 6$ with Newton-Cotes quadrature.

Remark 2. Although much simpler than the control-polygon constraints, the arc length quadrature method should be regarded as providing a tentative PH curve identification. Since it employs discrete sampling of the parametric speed, it may be possible for non-PH curves to exhibit successive coincident quadrature arc length values (although in practice this is highly improbable). To validate the identification, one should ensure the existence of consistent solutions to the over-determined systems of equations that characterize the reverse-engineering problem, as described below.

m	Example 1 PH quintic	Example 2 PH quintic
2	10.606601717798217	39.051248379533270
3	6.886675022604875	14.052822016316927
4	6.093507075615900	12.401798023047711
5	5.458972718024720	11.080978828432334
6	5.458972718024721	11.080978828432336
7	5.458972718024720	11.080978828432334
8	5.458972718024719	11.080978828432334
exact	5.458972718024720	11.080978828432336

Table 5: Newton–Cotes arc length quadrature estimates for Examples 1 & 2.

4 Reverse engineering of PH curves

Once the curve (1) specified by its Bézier control points has been identified as a PH curve, through the procedures described above, it will be necessary to “reverse engineer” it, in order to fully exploit its advantageous computational properties. This amounts to a determination of the coefficients of the complex polynomial $\mathbf{w}(t)$ in (4) for a planar PH curve, or the quaternion polynomial $\mathcal{A}(t)$ in (5) for a spatial PH curve. Since the identification schemes do not distinguish these two cases, the first step is to ascertain which applies. This is accomplished by forming the components of $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ into a $3 \times n$ array: the curve is planar or spatial if it is of rank 2 or 3, respectively. In the former case, the control points are transformed so as to lie in the (x, y) plane.

4.1 Planar PH curves

For an identified planar PH curve, the reverse engineering problem consists of determining the complex pre-image polynomial that generates its hodograph through (4). If $\mathbf{w}(t)$ is of degree m , and specified in Bernstein form as

$$\mathbf{w}(t) = \sum_{k=0}^m \mathbf{w}_k b_k^m(t),$$

the complex coefficients $\mathbf{w}_0, \dots, \mathbf{w}_m$ are to be determined from $\mathbf{p}_0, \dots, \mathbf{p}_n$. To achieve this, the control points are regarded as complex values $\mathbf{p}_k = x_k + i y_k$, the quantities in (6) being $\mathbf{d}_k = n [(x_{k+1} - x_k) + i(y_{k+1} - y_k)]$.

Equating (4) with (6) then gives a system of $2m + 1$ equations for the $m + 1$ complex unknowns $\mathbf{w}_0, \dots, \mathbf{w}_m$. Specifically, for PH cubics, we obtain

$$\mathbf{w}_0^2 = \mathbf{d}_0, \quad \mathbf{w}_0\mathbf{w}_1 = \mathbf{d}_1, \quad \mathbf{w}_1^2 = \mathbf{d}_2,$$

while for PH quintics the equations are

$$\mathbf{w}_0^2 = \mathbf{d}_0, \quad \mathbf{w}_0\mathbf{w}_1 = \mathbf{d}_1, \quad 2\mathbf{w}_1^2 + \mathbf{w}_0\mathbf{w}_2 = 3\mathbf{d}_2, \quad \mathbf{w}_1\mathbf{w}_2 = \mathbf{d}_3, \quad \mathbf{w}_2^2 = \mathbf{d}_4.$$

Although there are more equations than unknowns, the fact that $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ are known to define a PH curve ensures that the equations are consistent, and it suffices to consider only the first (or last) $m + 1$ equations to determine $\mathbf{w}_0, \dots, \mathbf{w}_m$. From the first two equations in the case of PH cubics, we obtain

$$\mathbf{w}_0 = \sqrt{\mathbf{d}_0}, \quad \mathbf{w}_1 = \frac{\mathbf{d}_1}{\sqrt{\mathbf{d}_0}},$$

while the first three equations for PH quintics yield

$$\mathbf{w}_0 = \sqrt{\mathbf{d}_0}, \quad \mathbf{w}_1 = \frac{\mathbf{d}_1}{\sqrt{\mathbf{d}_0}}, \quad \mathbf{w}_2 = \frac{1}{\sqrt{\mathbf{d}_0}} \left[3\mathbf{d}_2 - \frac{2\mathbf{d}_1^2}{\mathbf{d}_0} \right], \quad (24)$$

where $\mathbf{d}_0 \neq 0$ for a regular curve. In general, $\sqrt{\mathbf{d}_0}$ admits two complex values, and the same value must be used in these expressions — the value chosen is immaterial, since they differ only in sign, and the complex polynomial $\mathbf{w}(t)$ is squared in (4). Alternatively, one may use

$$\mathbf{w}_1 = \sqrt{\mathbf{d}_2}, \quad \mathbf{w}_0 = \frac{\mathbf{d}_1}{\sqrt{\mathbf{d}_2}},$$

for PH cubics (where $\mathbf{d}_2 \neq 0$ for a regular curve), and

$$\mathbf{w}_2 = \sqrt{\mathbf{d}_4}, \quad \mathbf{w}_1 = \frac{\mathbf{d}_3}{\sqrt{\mathbf{d}_4}}, \quad \mathbf{w}_0 = \frac{1}{\sqrt{\mathbf{d}_4}} \left[3\mathbf{d}_2 - \frac{2\mathbf{d}_3^2}{\mathbf{d}_4} \right]. \quad (25)$$

for PH quintics (where $\mathbf{d}_4 \neq 0$ for a regular curve). The equivalence of these alternative forms arises from the fact that $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ are not independent for a PH curve. For PH cubics, their dependence can be compactly expressed [3] by the complex constraint⁷ $\mathbf{d}_1^2 = \mathbf{d}_0\mathbf{d}_2$. For PH quintics, the situation is more complicated [3], since a number of special cases must be accommodated.

⁷This can be viewed as an alternative to conditions (11)–(12) — in the case of planar PH cubics — facilitated by use of the complex representation.

Example 6. Consider the planar PH quintic of Example 1 shown in Figure 2. For this curve, we obtain (to full precision) the coefficients

$$\begin{aligned}\mathbf{w}_0 &= 3.0088703625944260 - 1.2463149116090630i, \\ \mathbf{w}_1 &= 0.0038308962625464 + 4.5675312287005045i, \\ \mathbf{w}_2 &= 3.0088703625944269 - 1.2463149116090637i,\end{aligned}$$

from the expressions (24) based on $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2$ and

$$\begin{aligned}\mathbf{w}_0 &= 3.0088703625944211 - 1.2463149116090662i, \\ \mathbf{w}_1 &= 0.0038308962625461 + 4.5675312287005028i, \\ \mathbf{w}_2 &= 3.0088703625944255 - 1.2463149116090635i,\end{aligned}$$

from the expressions (25) based on $\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4$. These two sets of values agree to at least 14 decimal places, consistent with the tolerance arising from the determination of its Bézier control points as a PH quintic interpolant to first-order Hermite data (the fact that \mathbf{w}_0 and \mathbf{w}_2 essentially identical reflects the symmetry of the curve). Analogous results were observed for the asymmetric PH quintic of Example 2, illustrated in Figure 3.

The methods described above, for planar PH cubics and quintics, can be extended in a relatively straightforward manner to higher-order PH curves.

4.2 Spatial PH curves

The reverse engineering of a spatial PH curve entails a determination of the quaternion polynomial $\mathcal{A}(t)$ in (5), given the curve control points. Namely, if $\mathcal{A}(t)$ is specified in the Bernstein as

$$\mathcal{A}(t) = \sum_{k=0}^m \mathcal{A}_k b_k^m(t), \quad (26)$$

the coefficients $\mathcal{A}_0, \dots, \mathcal{A}_m$ must be computed. Equating (5) and (6) gives a system of $2m + 1$ vector equations for the $m + 1$ quaternions $\mathcal{A}_0, \dots, \mathcal{A}_m$. As in the planar case, there are more equations than unknowns, but the fact that $\mathbf{d}_0, \dots, \mathbf{d}_{n-1}$ are known to define a PH curve ensures the consistency of these equations. Moreover, for spatial PH curves, the quaternion polynomial

(26) that generates a particular hodograph $\mathbf{r}'(t)$ through expression (5) is not unique [11]. Namely, if $\mathcal{A}(t)$ generates $\mathbf{r}'(t)$, then for any ϕ the polynomials

$$\tilde{\mathcal{A}}(t) = \mathcal{A}(t) \exp(\phi \mathbf{i}), \quad (27)$$

where $\exp(\phi \mathbf{i}) = (\cos \phi, \sin \phi \mathbf{i})$, also generates $\mathbf{r}'(t)$. This indeterminacy is required for rotation invariance of the form (5), i.e., $\mathbf{r}'(t)$ must be expressible in the form (5), for a suitable $\mathcal{A}(t)$, in a coordinate system of any orientation [11]. We focus below on reverse engineering of spatial PH cubics and quintics.

4.2.1 Spatial PH cubics

Equating (5) with (6) in the case of spatial PH cubics ($m = 1$) yields the system of three vector equations

$$\mathcal{A}_0 \mathbf{i} \mathcal{A}_0^* = \mathbf{d}_0, \quad \text{vect}(\mathcal{A}_0 \mathbf{i} \mathcal{A}_1^*) = \mathbf{d}_1, \quad \mathcal{A}_1 \mathbf{i} \mathcal{A}_1^* = \mathbf{d}_2, \quad (28)$$

for the two quaternion coefficients $\mathcal{A}_0, \mathcal{A}_1$. Setting $d_0 = |\mathbf{d}_0|, d_2 = |\mathbf{d}_2|$ (where $d_0, d_2 \neq 0$ for a regular curve) and $\boldsymbol{\delta}_0 = \mathbf{d}_0/d_0, \boldsymbol{\delta}_2 = \mathbf{d}_2/d_2$ the first and third equations in (28) can be used [8] to express \mathcal{A}_0 and \mathcal{A}_1 in the form

$$\mathcal{A}_0 = \sqrt{d_0} \mathbf{n}_0 \exp(\phi_0 \mathbf{i}) \quad \text{and} \quad \mathcal{A}_1 = \sqrt{d_2} \mathbf{n}_1 \exp(\phi_1 \mathbf{i}), \quad (29)$$

where ϕ_0, ϕ_1 are free parameters, and

$$\mathbf{n}_0 = \frac{\boldsymbol{\delta}_0 + \mathbf{i}}{|\boldsymbol{\delta}_0 + \mathbf{i}|} \quad \text{and} \quad \mathbf{n}_1 = \frac{\boldsymbol{\delta}_2 + \mathbf{i}}{|\boldsymbol{\delta}_2 + \mathbf{i}|}$$

are the unit bisectors of $\boldsymbol{\delta}_0, \boldsymbol{\delta}_2$ with \mathbf{i} (for brevity, we discount the degenerate cases where $\boldsymbol{\delta}_0 = -\mathbf{i}$ or $\boldsymbol{\delta}_2 = -\mathbf{i}$, which require separate treatment).

Substituting from (29) into the second of equations (28) and setting $\Delta\phi = \phi_1 - \phi_0$ then gives the vector condition

$$\sqrt{d_0 d_2} (\mathbf{u} \cos \Delta\phi + \mathbf{v} \sin \Delta\phi) = \mathbf{d}_1,$$

where

$$\mathbf{u} = (\mathbf{i} \cdot \mathbf{n}_1) \mathbf{n}_0 + (\mathbf{i} \cdot \mathbf{n}_0) \mathbf{n}_1 - (\mathbf{n}_0 \cdot \mathbf{n}_1) \mathbf{i}, \quad \mathbf{v} = \mathbf{n}_1 \times \mathbf{n}_0. \quad (30)$$

The value of $\Delta\phi$ must be chosen so as to ensure satisfaction of this condition. For generic data with $\mathbf{i} \cdot (\mathbf{n}_0 \times \mathbf{n}_1) \neq 0$, which is equivalent to $\mathbf{i} \cdot (\boldsymbol{\delta}_0 \times \boldsymbol{\delta}_2) \neq 0$, the dot product of this equation with $\mathbf{n}_0, \mathbf{n}_1, \mathbf{i}$ gives the scalar conditions

$$(\mathbf{i} \cdot \mathbf{n}_1) \cos \Delta\phi = \frac{\mathbf{n}_0 \cdot \mathbf{d}_1}{\sqrt{d_0 d_2}}, \quad (\mathbf{i} \cdot \mathbf{n}_0) \cos \Delta\phi = \frac{\mathbf{n}_1 \cdot \mathbf{d}_1}{\sqrt{d_0 d_2}}, \quad (31)$$

$$[2(\mathbf{i} \cdot \mathbf{n}_0)(\mathbf{i} \cdot \mathbf{n}_1) - \mathbf{n}_0 \cdot \mathbf{n}_1] \cos \Delta\phi - \mathbf{i} \cdot (\mathbf{n}_0 \times \mathbf{n}_1) \sin \Delta\phi = \frac{\mathbf{i} \cdot \mathbf{d}_1}{\sqrt{d_0 d_2}}. \quad (32)$$

If $\boldsymbol{\delta}_0 \neq -\mathbf{i}$ and $\boldsymbol{\delta}_2 \neq -\mathbf{i}$, we have $\mathbf{i} \cdot \mathbf{n}_0 \neq 0$ and $\mathbf{i} \cdot \mathbf{n}_1 \neq 0$, so either of the two equations (31) can be used to determine $\cos \Delta\phi$ (the equivalence of these equations can be seen from the “canonical form” (16) of a spatial PH cubic). In general, $\cos \Delta\phi$ determines two values $\Delta\phi \in [0, 2\pi)$ and the appropriate value must be determined by checking the satisfaction of (32). Once $\Delta\phi$ has been computed, ϕ_0 can be freely specified, and then with $\phi_1 = \phi_0 + \Delta\phi$ the desired quaternion coefficients are determined from (29).

Example 7. Consider the spatial PH cubic defined [18] by the control points

$$\mathbf{p}_0 = (0, 0, 0), \quad \mathbf{p}_1 = (2\sqrt{3}, 0, 2), \quad \mathbf{p}_2 = (2\sqrt{3}, 0, 8), \quad \mathbf{p}_3 = (-\sqrt{3}, 9, 14).$$

One can verify that this curve has the polynomial parametric speed $\sigma(t) = 10t^2 - 2t + 4$. To full precision, the quaternion coefficients computed by the above method (choosing $\phi_0 = 0$) are

$$\begin{aligned} \mathcal{A}_0 &= 0.0000000000000000 + 3.3460652149512313 \mathbf{i} \\ &\quad + 0.0000000000000000 \mathbf{j} + 0.8965754721680534 \mathbf{k}, \\ \mathcal{A}_1 &= 2.8977774788672042 + 1.3448632082520799 \mathbf{i} \\ &\quad - 0.7764571353075622 \mathbf{j} + 5.0190978224268470 \mathbf{k}. \end{aligned}$$

When the control points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are re-computed from these coefficients, through the expressions

$$\mathbf{p}_1 = \mathbf{p}_0 + \frac{1}{3} \mathcal{A}_0 \mathbf{i} \mathcal{A}_0, \quad \mathbf{p}_2 = \mathbf{p}_1 + \frac{1}{3} \text{vect}(\mathcal{A}_0 \mathbf{i} \mathcal{A}_1), \quad \mathbf{p}_3 = \mathbf{p}_2 + \frac{1}{3} \mathcal{A}_1 \mathbf{i} \mathcal{A}_1,$$

their errors relative to the originally-specified points are found to be

$$6 \times 10^{-16}, \quad 35 \times 10^{-16}, \quad 46 \times 10^{-16}.$$

Furthermore, it is verified from the implementation that the relative accuracy of $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ is insensitive to the choice of ϕ_0 .

4.2.2 Spatial PH quintics

Given the Bézier control points $\mathbf{p}_0, \dots, \mathbf{p}_5$ of a spatial PH quintic, the reverse engineering problem consists of determining the three quaternion coefficients

$\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ from the five vector equations

$$\begin{aligned}\mathcal{A}_0 \mathbf{i} \mathcal{A}_0^* &= \mathbf{d}_0, & \text{vect}(\mathcal{A}_0 \mathbf{i} \mathcal{A}_1^*) &= \mathbf{d}_1, \\ 2 \mathcal{A}_1 \mathbf{i} \mathcal{A}_1^* + \text{vect}(\mathcal{A}_0 \mathbf{i} \mathcal{A}_2^*) &= 3 \mathbf{d}_2, \\ \text{vect}(\mathcal{A}_1 \mathbf{i} \mathcal{A}_2^*) &= \mathbf{d}_3, & \mathcal{A}_2 \mathbf{i} \mathcal{A}_2^* &= \mathbf{d}_4,\end{aligned}\tag{33}$$

with $\mathbf{d}_i = 5(\mathbf{p}_{i+1} - \mathbf{p}_i)$ for $i = 0, \dots, 4$. Setting $d_i = |\mathbf{d}_i|$ (where $d_0, d_4 \neq 0$ for a regular curve) and $\boldsymbol{\delta}_0 = \mathbf{d}_0/d_0$, $\boldsymbol{\delta}_4 = \mathbf{d}_4/d_4$, we consider again only the generic case $\mathbf{i} \cdot (\boldsymbol{\delta}_0 \times \boldsymbol{\delta}_4) \neq 0$. Then the first and fifth equations in (33) yield

$$\mathcal{A}_0 = \sqrt{d_0} \mathbf{n}_0 \exp(\phi_0 \mathbf{i}) \quad \text{and} \quad \mathcal{A}_2 = \sqrt{d_4} \mathbf{n}_1 \exp(\phi_2 \mathbf{i}),\tag{34}$$

where ϕ_0, ϕ_2 are free parameters, and

$$\mathbf{n}_0 = \frac{\boldsymbol{\delta}_0 + \mathbf{i}}{|\boldsymbol{\delta}_0 + \mathbf{i}|} \quad \text{and} \quad \mathbf{n}_1 = \frac{\boldsymbol{\delta}_4 + \mathbf{i}}{|\boldsymbol{\delta}_4 + \mathbf{i}|}$$

are the unit bisectors of $\boldsymbol{\delta}_0, \boldsymbol{\delta}_4$ with \mathbf{i} . Substituting from (34) into the second of equations (33), one can deduce the expression⁸

$$\mathcal{A}_1 = - \frac{(\zeta, \mathbf{d}_1) \mathbf{n}_0 \exp(\phi_0 \mathbf{i}) \mathbf{i}}{\sqrt{d_0}},\tag{35}$$

where ζ is a free real parameter. Forming the product $\mathcal{A}_1 \mathbf{i} \mathcal{A}_1^*$ from (35), and noting that $2(\mathbf{i} \cdot \mathbf{n}_0) \mathbf{n}_0 - \mathbf{i} = \boldsymbol{\delta}_0$, after some simplification we obtain

$$\mathcal{A}_1 \mathbf{i} \mathcal{A}_1^* = \frac{\zeta^2 \boldsymbol{\delta}_0 - 2 \zeta d_1 \boldsymbol{\delta}_0 \times \boldsymbol{\delta}_1 + 2 d_1^2 (\boldsymbol{\delta}_0 \cdot \boldsymbol{\delta}_1) \boldsymbol{\delta}_1 - d_1^2 \boldsymbol{\delta}_0}{d_0}.\tag{36}$$

Also, using the expressions (34) gives

$$\text{vect}(\mathcal{A}_0 \mathbf{i} \mathcal{A}_2^*) = \sqrt{d_0 d_4} (\mathbf{u} \cos \Delta\phi + \mathbf{v} \sin \Delta\phi),\tag{37}$$

where $\Delta\phi = \phi_2 - \phi_0$, and \mathbf{u}, \mathbf{v} are as defined in (30).

Substituting (36) and (37) into the third of equations (33) then yields the vector condition

$$\begin{aligned}2 \zeta^2 \boldsymbol{\delta}_0 - 4 \zeta d_1 \boldsymbol{\delta}_0 \times \boldsymbol{\delta}_1 + d_0 (\mathbf{u} \cos \Delta\phi + \mathbf{v} \sin \Delta\phi) \\ = 3 d_0 \mathbf{d}_2 + 2 d_1^2 \boldsymbol{\delta}_0 - 4 d_1^2 (\boldsymbol{\delta}_0 \cdot \boldsymbol{\delta}_1) \boldsymbol{\delta}_1.\end{aligned}\tag{38}$$

⁸An analogous expression can also be obtained from the fourth of equations (33).

This defines an over-determined system of three scalar conditions on the two parameters ζ and $\Delta\phi$, but the system is consistent if the given data define a spatial PH quintic. Now the cross product of the vectors (30) simplifies to

$$\mathbf{u} \times \mathbf{v} = \frac{1}{2}d_0d_4(\boldsymbol{\delta}_4 - \boldsymbol{\delta}_0),$$

and is non-zero under the stated assumption $\mathbf{i} \cdot (\boldsymbol{\delta}_0 \times \boldsymbol{\delta}_4) \neq 0$. The variable $\Delta\phi$ can thus be eliminated from (38) by taking its dot product with $\mathbf{u} \times \mathbf{v}$ to obtain a quadratic equation

$$a\zeta^2 + b\zeta + c = 0, \quad (39)$$

in ζ with the coefficients $a = 2(\boldsymbol{\delta}_0 \cdot \boldsymbol{\delta}_4 - 1)$, $b = -4d_1(\boldsymbol{\delta}_0 \times \boldsymbol{\delta}_1) \cdot \boldsymbol{\delta}_4$, and $c = (\boldsymbol{\delta}_0 - \boldsymbol{\delta}_4) \cdot (3d_0\mathbf{d}_2 + 2d_1^2\boldsymbol{\delta}_0 - 4d_1^2(\boldsymbol{\delta}_0 \cdot \boldsymbol{\delta}_1)\boldsymbol{\delta}_1)$. The discriminant of (39) must satisfy $b^2 - 4ac \geq 0$ — this is a (non-obvious) consequence of the fact that $\mathbf{d}_0, \dots, \mathbf{d}_4$ define a PH quintic. For each root ζ of (39), corresponding values for $\Delta\phi$ must be computed. This is achieved by taking the dot product of equation (38) with \mathbf{i} , thereby reducing it to

$$\cos(\Delta\phi - \psi) = \frac{r(\zeta)}{\sqrt{p^2 + q^2}}, \quad (40)$$

where we set $p = d_0\mathbf{i} \cdot \mathbf{u}$, $q = d_0\mathbf{i} \cdot \mathbf{v}$,

$$r(\zeta) = \mathbf{i} \cdot [3d_0\mathbf{d}_2 + 2d_1^2\boldsymbol{\delta}_0 - 4d_1^2(\boldsymbol{\delta}_0 \cdot \boldsymbol{\delta}_1)\boldsymbol{\delta}_1 - 2\zeta^2\boldsymbol{\delta}_0 + 4\zeta d_1\boldsymbol{\delta}_0 \times \boldsymbol{\delta}_1],$$

and ψ is uniquely determined by

$$\cos \psi = \frac{p}{\sqrt{p^2 + q^2}}, \quad \sin \psi = \frac{q}{\sqrt{p^2 + q^2}}. \quad (41)$$

If a root ζ is such that the condition

$$|r(\zeta)| \leq \sqrt{p^2 + q^2} \quad (42)$$

is not satisfied, it must be rejected. If ζ is such that (42) is satisfied, equation (40) identifies, in general, two corresponding values $\Delta\phi \in [0, 2\pi]$.

Up to four distinct solutions $(\zeta, \Delta\phi)$ are determined in this manner. For each solution, with (say) ϕ_0 freely chosen and $\phi_2 = \phi_0 + \Delta\phi$, corresponding quaternions coefficients $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ may be computed by means of (34)–(35). In general, only one set of coefficients generates (within a given tolerance) the

prescribed vectors $\mathbf{d}_0, \dots, \mathbf{d}_4$ using (33), and the others must be discarded. The coefficients are unique only modulo a common factor $\exp(\phi \mathbf{i})$ for any ϕ , reflecting the free choice of ϕ_0 (or ϕ_2) — as noted in expression (27). Once $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ have been determined, the control points can be computed from them (and a given initial point \mathbf{p}_0) through the expressions

$$\begin{aligned}
\mathbf{p}_1 &= \mathbf{p}_0 + \frac{1}{5} \mathcal{A}_0 \mathbf{i} \mathcal{A}_0^*, \\
\mathbf{p}_2 &= \mathbf{p}_1 + \frac{1}{10} (\mathcal{A}_0 \mathbf{i} \mathcal{A}_1^* + \mathcal{A}_1 \mathbf{i} \mathcal{A}_0^*), \\
\mathbf{p}_3 &= \mathbf{p}_2 + \frac{1}{30} (\mathcal{A}_0 \mathbf{i} \mathcal{A}_2^* + 4 \mathcal{A}_1 \mathbf{i} \mathcal{A}_1^* + \mathcal{A}_2 \mathbf{i} \mathcal{A}_0^*), \\
\mathbf{p}_4 &= \mathbf{p}_3 + \frac{1}{10} (\mathcal{A}_1 \mathbf{i} \mathcal{A}_2^* + \mathcal{A}_2 \mathbf{i} \mathcal{A}_1^*), \\
\mathbf{p}_5 &= \mathbf{p}_4 + \frac{1}{5} \mathcal{A}_2 \mathbf{i} \mathcal{A}_2^*.
\end{aligned} \tag{43}$$

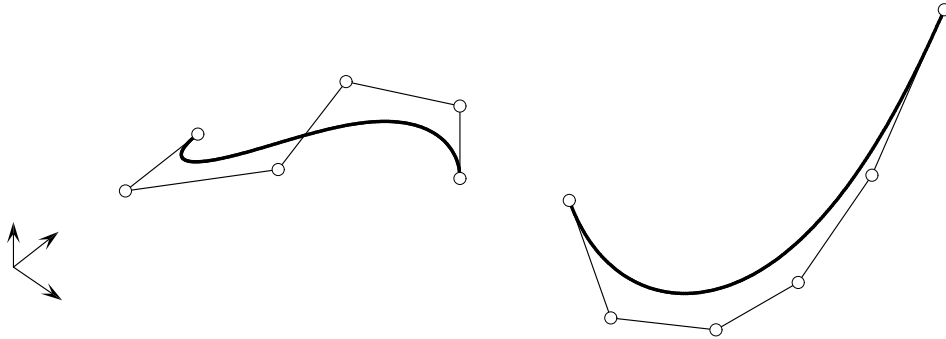


Figure 5: The two spatial PH quintic curves that are reverse-engineered in Example 8 (left) and Example 9 (right), with their Bézier control polygons.

Example 8. Consider the quintic space curve (see Figure 5) specified by the Bézier control points

$$\begin{aligned}
\mathbf{p}_0 &= (0, 0, 0), \quad \mathbf{p}_1 = (0, -\frac{2}{5}, 0), \quad \mathbf{p}_2 = (\frac{3}{5}, -\frac{1}{5}, \frac{2}{5}), \\
\mathbf{p}_3 &= (\frac{4}{15}, \frac{8}{15}, \frac{1}{15}), \quad \mathbf{p}_4 = (\frac{2}{3}, \frac{11}{15}, \frac{1}{15}), \quad \mathbf{p}_5 = (\frac{2}{3}, \frac{11}{15}, -\frac{1}{3}).
\end{aligned}$$

By the methods of Section 3 it is identified as a PH curve, with parametric speed is $\sigma(t) = 2(19t^4 - 40t^3 + 27t^2 - 6t + 1)$. Equation (39) has the roots $\zeta = 0$ and 6, and since $p = 2$ and $q = -2$ in (41), we have $\psi = -\frac{1}{4}\pi$. For $\zeta = 6$ condition (42) is violated, so this root must be discarded, but with $\zeta = 0$ we have $r(\zeta)/\sqrt{p^2 + q^2} = 1/\sqrt{2}$, and $\Delta\phi - \psi = \frac{1}{4}\pi$ or $\frac{7}{4}\pi$.

Hence, $\Delta\phi = 0$ or $\frac{3}{2}\pi$. Choosing $\phi_0 = 0$ and $\phi_2 = \Delta\phi$, and computing $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ from (34) and (35), the former choice yields agreement with the vectors $\mathbf{d}_i = 5(\mathbf{p}_{i+1} - \mathbf{p}_i)$ computed from (33), but the latter does not. Hence, $\zeta = 0$ and $\phi_0 = \phi_2 = 0$ identify the correct solution (any ϕ_0 value can be chosen, provided that $\phi_2 = \phi_0$). When $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ are computed from (34) and (35) in this manner, using double precision arithmetic, they are found to agree with the coefficients $\mathcal{A}_0 = \mathbf{i} - \mathbf{j}$, $\mathcal{A}_1 = 1 + \mathbf{i} + 2\mathbf{j} + \mathbf{k}$, $\mathcal{A}_2 = \mathbf{i} - \mathbf{k}$, from which the curve control points were initially generated using (43), to an accuracy of $\sim 10^{-15}$. The relative errors in $\mathbf{p}_1, \dots, \mathbf{p}_5$ (computed using the reverse-engineered coefficients) are found to be

$$0 \times 10^{-16}, \quad 2 \times 10^{-16}, \quad 13 \times 10^{-16}, \quad 33 \times 10^{-16}, \quad 31 \times 10^{-16}.$$

Example 9. The preceding example was constructed so as to have a simple known pre-image polynomial. As a more general example, we consider the spatial PH quintic interpolant (see Figure 5) to the first-order Hermite data

$$\mathbf{r}(0) = (0, 0, 0), \quad \mathbf{r}'(0) = (2, -1, -1) \quad \text{and} \quad \mathbf{r}(1) = (1, 1, 1), \quad \mathbf{r}'(1) = (0, 2, 3)$$

using $\phi_0 = \phi_2 = 0$ in the interpolation algorithm [8]. For this curve, $\mathbf{p}_0 = \mathbf{r}(0)$, $\mathbf{p}_1 = \mathbf{r}(0) + \frac{1}{5}\mathbf{r}'(0)$ and $\mathbf{p}_4 = \mathbf{r}(1) - \frac{1}{5}\mathbf{r}'(1)$, $\mathbf{p}_5 = \mathbf{r}(1)$, while to full precision the two interior control points obtained from the algorithm are

$$\mathbf{p}_2 = (0.7746664443097209, -0.0209484017535140, -0.1326892881613731),$$

$$\mathbf{p}_3 = (0.9601287216055423, 0.2353431635591809, 0.0635268956967925).$$

When $\mathbf{p}_1, \dots, \mathbf{p}_5$ are re-computed from the reverse-engineered coefficients $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ using expressions (43), their errors relative to the original control points are found to be

$$2 \times 10^{-16}, \quad 1 \times 10^{-16}, \quad 25 \times 10^{-16}, \quad 33 \times 10^{-16}, \quad 23 \times 10^{-16},$$

so the reverse-engineering is also very accurate in this more general context.

Remark 3. In various contexts, the roots of the parametric speed $\sigma(t)$ are required, e.g., in computing the absolute rotation index [14], elastic bending energy [4], and rotation-minimizing frames [5], which require partial-fraction decompositions of rational integrands with $\sigma(t)$ as denominator. The roots can be computed after completing the reverse-engineering process, by setting $\sigma(t) = |\mathbf{w}(t)|^2$ or $|\mathcal{A}(t)|^2$ for planar or spatial PH curves, respectively.

5 Closure

Two simple methods have been presented to identify whether or not a given polynomial curve, specified by its Bézier control points, has a Pythagorean hodograph. The first method checks a system of constraints on the control–polygon legs, enumerated explicitly for the case of cubics and quintics. The second method computes arc length estimates through numerical quadrature rules with polynomial precision on successively augmented node sets.

Both methods apply equally to planar and spatial PH curves. Computed examples indicate that they can both distinguish PH curves from “ordinary” polynomial curves to an accuracy of $\sim 10^{-14}$ or better using double precision arithmetic. The first method employs only the curve control points, but the second method also requires storage of the quadrature nodes and weights. However, the quadrature method is perhaps preferable in terms of efficiency and straightforward extensibility to PH curves of degree 7 or higher.

The proposed methods provide a practical means of importing PH curves into existing CAD systems without modification of prevailing representation schemes. Once the control points have been identified as defining a PH curve, the reverse engineering procedures allow very accurate reconstruction of the complex or quaternion pre–image polynomials (for planar and spatial curves, respectively) in double–precision arithmetic, so the advantageous properties of PH curves can be fully exploited. These include closed–form solutions for arc lengths, offset curves, and elastic bending energy [4, 17]; real–time CNC interpolators for constant or variable feedrates along curved paths [13, 19]; a diverse family of sweep operations yielding rational surfaces [15]; and rational rotation–minimizing frames along space curves [7, 9].

Acknowledgements

This work was supported by the Gruppo Nazionale per il Calcolo Scientifico (GNCS) of the Istituto Nazionale di Alta Matematica Francesco Severi (INdAM) and by the project DREAMS (MIUR Futuro in Ricerca RBFR13FBI3).

References

- [1] H. I. Choi, D. S. Lee, and H. P. Moon (2002), Clifford algebra, spin representation, and rational parameterization of curves and surfaces,

Adv. Comp. Math. **17**, 5–48.

- [2] G. Farin, (1997), *Curves and Surfaces for Computer Aided Geometric Design*, 4th edition, Academic Press, San Diego.
- [3] R. T. Farouki (1994), The conformal map $z \rightarrow z^2$ of the hodograph plane, *Comput. Aided Geom. Design* **11**, 363–390.
- [4] R. T. Farouki (1996), The elastic bending energy of Pythagorean–hodograph curves, *Comput. Aided Geom. Design* **13**, 227–241 (1996)
- [5] R. T. Farouki (2002), Exact rotation–minimizing frames for spatial Pythagorean–hodograph curves, *Graph. Models* **64**, 382–395 (2002)
- [6] R. T. Farouki (2008), *Pythagorean–Hodograph Curves: Algebra and Geometry Inseparable*, Springer, Berlin.
- [7] R. T. Farouki (2010), Quaternion and Hopf map characterizations for the existence of rational rotation–minimizing frames on quintic space curves, *Adv. Comp. Math.* **33**, 331–348.
- [8] R. T. Farouki, C. Giannelli, C. Manni, and A. Sestini (2008), Identification of spatial PH quintic Hermite interpolants with near–optimal shape measures, *Comput. Aided Geom. Design* **25**, 274–297.
- [9] R. T. Farouki, C. Giannelli, C. Manni, and A. Sestini (2012), Design of rational rotation–minimizing rigid body motions by Hermite interpolation, *Math. Comp.* **81**, 879–903.
- [10] R. T. Farouki, C. Giannelli, and A. Sestini (2014), Local modification of Pythagorean–hodograph quintic spline curves using the B–spline form, preprint.
- [11] R. T. Farouki, M. al–Kandari, and T. Sakkalis (2002), Structural invariance of spatial Pythagorean hodographs, *Comput. Aided Geom. Design* **19**, 395–407.
- [12] R. T. Farouki, B. K. Kuspa, C. Manni, and A. Sestini (2001), Efficient solution of the complex quadratic tridiagonal system for C^2 PH quintic splines, *Numer. Algor.* **27**, 35–60.

- [13] 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, *Comput. Aided Design* **30**, 631–640.
- [14] R. T. Farouki and C. A. Neff (1995), Hermite interpolation by Pythagorean-hodograph quintics, *Math. Comp.* **64**, 1589–1609.
- [15] R. T. Farouki and K. M. Nittler (2014), Rational swept surface constructions based on differential and integral sweep curve properties, *Comput. Aided Geom. Design* to appear.
- [16] R. T. Farouki and V. T. Rajan (1988), Algorithms for polynomials in Bernstein form, *Comput. Aided Geom. Design* **5**, 1–26.
- [17] R. T. Farouki and T. Sakkalis (1990), Pythagorean hodographs, *IBM J. Res. Develop.* **34**, 736–752.
- [18] R. T. Farouki and T. Sakkalis (1994), Pythagorean-hodograph space curves, *Adv. Comp. Math.* **2**, 41–66.
- [19] R. T. Farouki and S. Shah (1996), Real-time CNC interpolators for Pythagorean-hodograph curves, *Comput. Aided Geom. Design* **13**, 583–600.
- [20] Z. Habib and M. Sakai (2007), G^2 Pythagorean hodograph quintic transition between two circles with shape control, *Comput. Aided Geom. Design* **24**, 252–266.
- [21] M. Huard, R. T. Farouki, N. Sprynski, and L. Biard (2014), C^2 interpolation of spatial data subject to arc-length constraints using Pythagorean-hodograph quintic splines, *Graph. Models* **76**, 30–42.
- [22] G. Jaklic, J. Kozak, M. Krajnc, V. Vitrih, and E. Zagar (2012), An approach to geometric interpolation by Pythagorean-hodograph curves, *Adv. Comp. Math.* **37**, 123–150.
- [23] B. Jüttler and C. Mäurer (1999), Cubic Pythagorean hodograph spline curves and applications to sweep surface modelling, *Comput. Aided Design* **31**, 73–83.

- [24] B. Jüttler (2001), Hermite interpolation by Pythagorean hodograph curves of degree seven, *Math. Comp.* **70**, 1089–1111.
- [25] G. Klar and G. Valasek (2011), Employing Pythagorean hodograph curves for artistic patterns, *Acta Cybernetica* **20**, 101-110.
- [26] P. K. Kythe and M. R. Schäferkötter (2005), *Handbook of Computational Methods for Integration*, Chapman & Hall/CRC, Boca Raton.
- [27] J. Monterde and F. Ongay (2012), An isoperimetric type problem for primitive Pythagorean hodograph curves, *Comput. Aided Geom. Design* **29**, 626–647.
- [28] G. Peters and J. H. Wilkinson (1971), Practical problems arising in the solution of polynomial equations, *J. Inst. Math. Appl.* **29**, 626–647.
- [29] D. J. Walton and D. S. Meek (1998), G^2 curves composed of planar cubic and Pythagorean–hodograph spirals, *Comput. Aided Geom. Design* **15**, 547–566.
- [30] D. J. Walton and D. S. Meek (2002), Planar G^2 transition with a fair Pythagorean hodograph quintic curve, *J. Comput. Appl. Math.* **138**, 109–126.