Harmonic Interpolation, Bézier Curves and Trigonometric Interpolation

A. Hardy and W.-H. Steeb

International School for Scientific Computing, Rand Afrikaans University,
Auckland Park 2006, South Africa

Reprint requests to Prof. W.-H. S.; e-mail: WHS@NA.RAU.AC.ZA

Z. Naturforsch. **59a,** 591 – 596 (2004); received May 4, 2004

A harmonic interpolation of a polygon (for odd and even numbers of points forming the polygon) used in computer graphics is derived from the primary permutation matrix using the spectral decomposition of the matrix. This is a technique to draw closed curves. We compare these curves with Bézier curves. We also show situations where the harmonic interpolation of a polygon is a more suitable alternative to Bézier curves. Trigonometric Interpolation is another technique to draw curves. The relationship between trigonometric interpolation and harmonic interpolation is discussed.

Key words: Harmonic Interpolation; Permutation Matrix; Bézier Curves.

1. Introduction

Bézier curves [1] form the basis of most curve drawing operations in computer graphics. Many curves used in computer graphics can be expressed as Bézier curves. Bézier curves allow us to define smooth curves using a number of control points. We are interested in finding a smooth curve (harmonic interpolation) through a number of n-ordered points (a polygon) $\mathbf{X} = (X_0, X_1, \ldots, X_{n-1})^T$ given in the Hilbert space $\mathbf{R}^d$, i. e., $X_j \in \mathbf{R}^d$. In the construction the starting point is the $n \times n$ primary permutation matrix and its spectral decomposition. In this paper we describe the construction of the harmonic interpolation, starting from the primary permutation matrix and the spectral representation of this matrix. We are not only going to study the case with an odd number of points forming a polygon [2], but also the case with an even number of points. We then describe Bézier curves and compare curves produced by harmonic interpolation and Bézier curves. We then show the relationship between trigonometric interpolation [3] and harmonic interpolation, using the discrete Fourier transform.

2. Primary Permutation Matrix and Harmonic Interpolation

The starting point in the derivation of the harmonic interpolation is the $n \times n$ primary permutation matrix U [2, 4, 5], given by

$$U := \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix}. \tag{1}$$

It is an $n \times n$ circulant matrix. An arbitrary $n \times n$ circulant matrix C [4, 5] is given by

$$C = \begin{pmatrix} c_0 & c_1 & c_2 & \ldots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ldots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \ldots & c_{n-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_1 & c_2 & c_3 & \ldots & c_0 \end{pmatrix}. \tag{2}$$

Let us first summarize the properties of the matrix U. The set of matrices $\{U, U^2, \ldots, U^n = I_n\}$ is a subgroup of the group of all $n \times n$ permutation matrices under matrix multiplication. They form a cyclic group of order n. An $n \times n$ matrix D is circulant if and only if $D = UDU^T$, where U is given by (1). Let C be a circulant matrix given by (2) and let

$$f(\lambda) = c_0 + c_1\lambda + \ldots + c_{n-1}\lambda^{n-1}. \tag{3}$$

Then $C = f(U)$, where U is the primary permutation matrix. The matrix C is normal, that is $C^*C = CC^*$. The eigenvalues of C are $f(\omega^k)$, where $k = 0, 1, \ldots, n-1$ and $\omega := \exp(2\pi i/n)$. For the determinant we have $\det(C) = f(\omega^0) f(\omega^1) \ldots f(\omega^{n-1})$.

For our construction of the harmonic interpolation we need the eigenvalues and normalized eigenvectors of U. The eigenvalues are given by $\lambda^0 = 1, \lambda^1, \lambda^2, \ldots, \lambda^{n-1}$ with $\lambda := \exp(2\pi i/n)$. The normalized eigenvectors are given by

$$|\theta_j\rangle = \frac{1}{\sqrt{n}}(1, \exp(-i2\pi j/n), \exp(-i4\pi j/n), \ldots, \exp(-i2\pi(n-1)j/n))^T, \tag{4}$$

where $j = 0, 1, \ldots, n-1$. Thus for the eigenvalue 1 of U we find the normalized eigenvector

$$|\theta_j\rangle = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)^T. \tag{5}$$

The eigenstates $\{|\theta_j\rangle : j = 0, 1, \ldots, n-1\}$ and the standard basis $\{|\mathbf{e}_j\rangle\}$ $(j = 0, 1, \ldots, n-1)$ in $\mathbf{C}^n$ are connected by the discrete Fourier transform

$$\begin{aligned} |\theta_j\rangle &= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \exp(-ik\theta_j)|\mathbf{e}_k\rangle, \\ |\mathbf{e}_k\rangle &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \exp(-ik\theta_j)|\theta_j\rangle, \end{aligned} \tag{6}$$

where $\theta_j := 2\pi j/n$. Thus the spectral representation of U can be written as

$$U = \sum_{j=0}^{n-1} \lambda^j P_j \equiv \sum_{j=0}^{n-1} \lambda^j |\theta_j\rangle\langle\theta_j|. \tag{7}$$

The projection matrix $P_j = |\theta_j\rangle\langle\theta_j|$ can be expressed using U^k. Since $P_j P_k = 0$ for $j \neq k$ and $P_j^2 = P_j$ we find

$$U^k = \sum_{j=0}^{n-1} \lambda^{jk} P_j, \quad k = 1, 2, \ldots, n. \tag{8}$$

Thus we calculated the Fourier transform of the projection matrices. For $k = n$ we have $U^n = I_n$ and the completeness relation

$$I_n = \sum_{j=0}^{n-1} P_j. \tag{9}$$

The inverse of the matrix (λ^{jk}) is given by $\frac{1}{n}(\lambda^{-jk})$. Thus

$$P_j = \frac{1}{n} \sum_{k=0}^{n-1} \lambda^{-jk} U^k, \quad j = 0, 1, \ldots, n-1. \tag{10}$$

For the construction of the harmonic interpolation in computer graphics we proceed as follows. We embed the matrices U^k into a real Lie group [6]. Thus we have to consider the cases $n = 2m+1$ and $n = 2m$, where m is a positive integer.

Furthermore we define $P_{-j} := P_{n-j}$. The projection matrices are therefore real. We consider first the case $n = 2m+1$ [2]. We obtain by replacing k with t

$$\begin{aligned} U(t) &= P_0 + \sum_{j=1}^{m} (\lambda^{jt} P_j + \lambda^{-jt} P_{-j}) \\ &\equiv P_0 + \sum_{j=1}^{m} (e^{2\pi i jt/n} P_j + e^{-2\pi i jt/n} P_j). \end{aligned} \tag{11}$$

The unitary matrices $U(t)$ are n-periodic, i.e., $U(t+n) = U(t)$. We replace t by nt, and therefore we reduce the period to 1. Thus $U(t)$ takes the form

$$\tilde{U}(t) = P_0 + \sum_{j=1}^{m} (e^{2\pi i jt} P_j + e^{-2\pi i jt} P_{-j}), \quad t \in \mathbf{R}. \tag{12}$$

Thus $\tilde{U}(t+1) = \tilde{U}(t)$ and $\tilde{U}(k/n) = U^k$ for $k = 0, 1, \ldots, n-1$. The $n \times n$ matrices $\tilde{U}(t)$ form a one-dimensional abelian unitary group (Lie group) with $\tilde{U}(0) = I_n$ and $\tilde{U}(t+\tau) = \tilde{U}(t)\tilde{U}(\tau)$. Owing to (10) we can write $\tilde{U}(t)$ as

$$\tilde{U}(t) = \frac{1}{n} \sum_{k=0}^{n-1} \left(1 + \sum_{j=1}^{m} \left(e^{2\pi i j(t-k/n)} + e^{-2\pi i j(t-k/n)}\right)\right) U^k. \tag{13}$$

We define

$$\begin{aligned} \sigma_k(t) &\equiv \sigma(t - k/n) \\ &:= \frac{1}{n} \left(1 + \sum_{j=1}^{m} \left(e^{2\pi i j(t-k/n)} + e^{-2\pi i j(t-k/n)}\right)\right). \end{aligned} \tag{14}$$

Using $\exp(i\alpha) \equiv \cos(\alpha) + i\sin(\alpha)$,

$$1 + 2\sum_{j=1}^{m} \cos(j\alpha) \equiv \frac{\sin((m+1/2)\alpha)}{\sin(\alpha/2)}, \tag{15}$$

and with $n = 2m+1$ we find

$$\sigma_k(t) = \frac{\sin(\pi n(t-k/n))}{n\sin(\pi(t-k/n))}, \quad k = 0, 1, \ldots, n-1. \tag{16}$$

Equation (15) was used by Lejeune-Dirichlet for his proof of Fourier's formula [7]. Thus we can write

$$\tilde{U}(t) = \sum_{k=0}^{n-1} \sigma_k(t) U^k. \tag{17}$$

We have the properties

$$\sum_{k=0}^{n-1} \sigma_k(t) = 1, \quad \sum_{k=0}^{n-1} \sigma_k^2(t) = 1. \tag{18}$$

Thus the functions $\sigma_k(t)$ and $\sigma_k^2(t)$ provide a partition of unity, and the harmonic interpolation is affine invariant. Now let $\mathbf{X} = (X_0, X_1, \ldots, X_{n-1})^T$ be the vector which describes the polygon. Then

$$\mathbf{X}(t) = \tilde{U}(t)\mathbf{X} = \sum_{k=0}^{n-1} \sigma(t - k/n) U^k \mathbf{X}. \tag{19}$$

The curve which describes our closed smooth curve is given by

$$\begin{aligned} X_l(t) &= \sum_{k=0}^{n-1} \sigma(t - k/n) X_{k+l} \\ &= \sum_{k=0}^{n-1} \sigma(t + l/n - (k+l)/n) X_{k+l} \\ &= X_0(t + l/n), \end{aligned} \tag{20}$$

where $(k+l)$ is calculated mod n. Hence for all l it represents the same curve, and we can write

$$X(t) = \sum_{k=0}^{n-1} \sigma(t - k/n) X_k, \quad t \in [0,1]. \tag{21}$$

Thus this curve interpolates the points of the given polygon smoothly. We consider now the case $n = 2m$. For this case we can write

$$U^k = P_0 + (-1)^k P_m + \sum_{j=1}^{m-1} (\lambda^{jk} P_j + \lambda^{-jk} P_{-j}), \tag{22}$$

where we have $\lambda^0 = 1$ and $\lambda^m = -1$. If we replace the discrete variable k by the real variable t, we find the factor $(-1)^t \equiv e^{\pi i t}$. The other terms are real. Thus the Lie group we would find is not real and therefore cannot be used directly for the harmonic interpolation in computer graphics. Since the function $U(t)$ will be continuous in the complex domain and go through the points of the polygon, the real part will also be continuous and go through the points of the polygon.

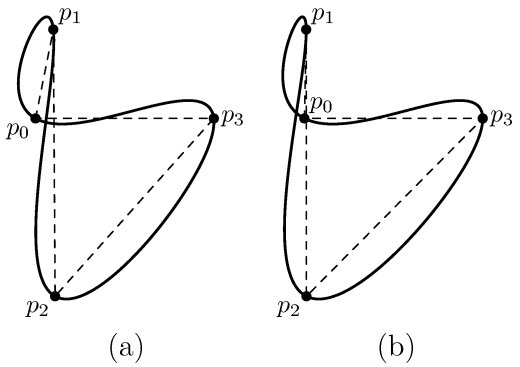

Fig. 1. Stability of harmonic interpolation under small changes.

A similar calculation as described for the case $n = 2m+1$ given above yields ($t \in [0,1]$)

$$\begin{aligned} \tilde{U}(t) = \frac{1}{n} \sum_{k=0}^{n=1} \cos(\pi(nt-k)) U^k \\ + \frac{i}{n} \sum_{k=0}^{n-1} \sin(\pi(nt-k)) U^k + \sum_{k=0}^{n-1} \xi_k(t) U^k, \end{aligned} \tag{23}$$

where

$$\xi(t) := \frac{\sin(\pi t(n-1))}{n \sin(\pi t)}, \quad \xi_k(t) := \xi\left(t - \frac{k}{n}\right). \tag{24}$$

Thus our smooth curve in the even case, which goes through all the points of the polygon, is

$$X(t) = \sum_{k=0}^{n-1} \left(\frac{1}{n} \cos(\pi(nt-k)) + \xi_k(t) \right) X_k. \tag{25}$$

The curves described above are stable in the sense that, making a small change in the position of a point, we only have a small change of the curve in the neighbourhood of this point (Fig. 1). Figure 1 shows how stable the curve is for harmonic interpolation when points are shifted by a small amount. In Fig. 1(a) the points are $\mathbf{p}_0 = (0,0)$, $\mathbf{p}_1 = (0.1, 0.5)$, $\mathbf{p}_2 = (0.11, -1)$ and $\mathbf{p}_3 = (1,0)$ and in Fig. 1(b) the points are $\mathbf{p}_0 = (0,0)$, $\mathbf{p}_1 = (0.01, 0.5)$, $\mathbf{p}_2 = (0.011, -1)$ and $\mathbf{p}_3 = (1,0)$.

As another example of harmonic interpolation, we illustrate how a unit circle centered at the origin can be parameterized using harmonic interpolation. We use four control points: $\mathbf{X}_0 = (1,0)$, $\mathbf{X}_1 = (0,1)$, $\mathbf{X}_2 = (-1,0)$ and $\mathbf{X}_3 = (0,-1)$. Then we obtain $\mathbf{X}(t) = (\cos(2\pi t), \sin(2\pi t))$.

A Java application and Java applet which draw the harmonic interpolation curves is available from the author.

3. Bézier Curves

A Bézier curve $\mathbf{B}(t)$ [1] of degree n is defined by a set of control points $\mathbf{p}_j$ for $j = 0, 1, \ldots, n$ as follows

$$\mathbf{B}(t) = \sum_{j=0}^{n} \mathbf{p}_j B_{j,n}(t), \tag{26}$$

where $B_{j,n}$ are the Bernstein polynomials

$$B_{j,n}(t) := \binom{n}{j} t^j (1-t)^{n-j}. \tag{27}$$

The sum of the Bernstein polynomials is a partition of unity

$$\sum_{j=0}^{n} B_{j,n}(t) = 1, \tag{28}$$

where $B_{j,n}(t) \geq 0$ for every $0 \leq t \leq 1$. These properties imply that Bézier curves are affine invariant, and that the curve lies entirely in the convex hull of the control points defining the curve. We find that $\mathbf{B}(0) = \mathbf{p}_0$ and that $\mathbf{B}(1) = \mathbf{p}_n$. Thus the Bézier curve interpolates the end points. If we calculate the derivative with respect to t at the endpoints, we find that $\frac{d\mathbf{B}}{dt}(0) = n(\mathbf{p}_1 - \mathbf{p}_0)$, and $\frac{d\mathbf{B}}{dt}(1) = n(\mathbf{p}_n - \mathbf{p}_{n-1})$. Thus the tangents at the end points of the curve are easily computed. This allows us to construct piecewise smooth C^1 or G^1 curves out of Bézier curves by setting $n_p(\mathbf{p}_1 - \mathbf{p}_0) = cn_q(\mathbf{q}_n - \mathbf{q}_{n-1})$ for a curve

$$\mathbf{Q}(t) = \sum_{j=0}^{n_q} \mathbf{q}_j B_{j,n}(t), \tag{29}$$

followed by a curve

$$\mathbf{P}(t) = \sum_{j=0}^{n_p} \mathbf{p}_j B_{j,n}(t). \tag{30}$$

The De Casteljau representation of Bézier curves [8] generates points on the curve by repeated linear interpolation. If the control points of the curve are $\mathbf{p}_j$ for $j = 0, 1, \ldots, n$, which defines a Bézier curve of degree n, then we define

$$\mathbf{p}_j^r(t) := (1-t)\mathbf{p}_j^{r-1}(t) + t\mathbf{p}_{j+1}^{r-1}(t) \tag{31}$$

with $r = 1, \ldots, n$, $j = 0, \ldots, n-r$, and $\mathbf{p}_j^0(t) = \mathbf{p}_j$, so that $\mathbf{B}(t) = \mathbf{p}_0^n(t)$. The repeated linear interpolation as demonstrated in Fig. 2 is an efficient technique to draw Bézier curves. Dashed lines show the control polygon, and dark dashed lines connect the points between which repeated linear interpolation takes place.

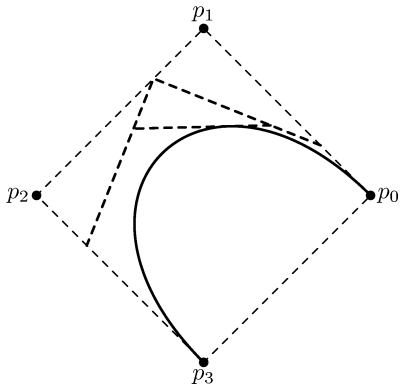

Fig. 2. De Casteljau's algorithm to render a point on a Bézier curve ($t = 0.3$).

4. Trigonometric Interpolation

Trigonometric interpolation is based on the discrete Fourier transform. Given a set of data

$$\{x_j, y_j = f(x_j)\}, \tag{32}$$

where $j = 0, 1, \ldots, N-1$, suppose that y_j can be complex and that the node points are equally spaced in the interval $[0, 2\pi]$, i. e., $x_j = 2\pi j/N$ for $j = 0, 1, \ldots, N$. Then we have to find

$$\begin{aligned} p(x) &= \sum_{k-0}^{N-1} c_k e^{ikx} \\ &= c_0 + c_1 e^{ix} + c_2 e^{2ix} + \ldots + c_{N-1} e^{(N-1)ix} \end{aligned} \tag{33}$$

such that

$$p(x_j) = f(x_j). \tag{34}$$

This interpolation is called trigonometric interpolation. Since

$$e^{i(x+2\pi)} = e^{ix} e^{2i\pi} = e^{ix}, \tag{35}$$

the interpolation is periodic, i. e., $p(x+2\pi) = p(x)$. Let $w^j = e^{ix_j}$, where $w := e^{2\pi i/N}$. Thus $w^N = 1$, so that w is the Nth root of unity. We obtain

$$\begin{aligned} f(x_j) &= p(x_j) \\ &= c_0 + c_1 w^j + c_2 w^{2j} + \cdots + c_{N-1} w^{(N-1)j}. \end{aligned} \tag{36}$$

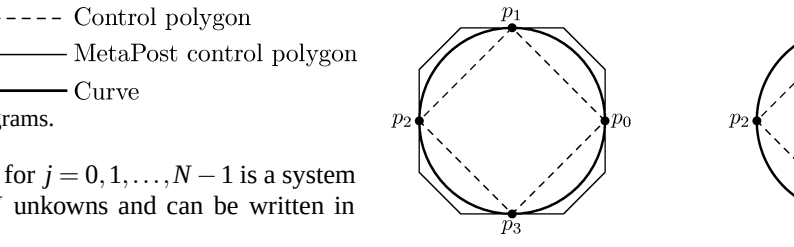

Fig. 3. Key for the diagrams.

This set of equations for $j = 0, 1, \ldots, N-1$ is a system of N equations in N unkowns and can be written in matrix form $A\mathbf{c} = \mathbf{y}$:

$$\begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & w & w^2 & \ldots & w^{N-1} \\ 1 & w^2 & w^4 & \ldots & w^{2(N-1)} \\ \vdots & & & & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \ldots & w^{(N-1)^2} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{pmatrix}. \quad (37)$$

This is the discrete Fourier transform. The matrix A on the left-hand side has an inverse with

$$(A^{-1})_{jk} = \frac{1}{N} w^{-jk}. \quad (38)$$

Thus $\mathbf{c} = A^{-1}\mathbf{y}$. It follows that

$$c_k = \frac{1}{N} \sum_{m=0}^{N-1} w^{-km} y_m, \quad (39)$$

where $k = 0, 1, \ldots, N-1$. Neglecting sums, to compute each coefficient c_k we do N products. This means that discrete Fourier transforms require N^2 operations. Hence for the calculation we should use the fast Fourier transform. Consider the data set ($N = 4$) $(x_0 = 0, y_0 = 0)$, $(x_1 = \frac{\pi}{2}, y_1 = \frac{\sqrt{3}}{2}\pi)$, $(x_2 = \pi, y_2 = \pi)$, $(x_3 = \frac{3}{2}\pi, y_3 = \frac{\sqrt{3}}{2}\pi)$. We find

$$\begin{aligned} c_0 &= \frac{\pi}{4}(1+\sqrt{3}), \quad c_1 = -\frac{\pi}{4}, \\ c_2 &= \frac{\pi}{4}(1-\sqrt{3}), \quad c_3 = -\frac{\pi}{4}. \end{aligned} \quad (40)$$

Thus

$$p(x) = \frac{\pi}{4}(1+\sqrt{3} - e^{ix} + (1-\sqrt{3})e^{2ix} - e^{3ix}). \quad (41)$$

The real part is

$$\begin{aligned} pr(x) = \frac{\pi}{4}(1+\sqrt{3} - \cos(x) \\ + (1-\sqrt{3})\cos(2x) - \cos(3x)), \end{aligned} \quad (42)$$

which passes through the given points. However the curve shows an oscillating behaviour. On the other hand, harmonic interpolation gives a smooth curve without oscillating behaviour.

Fig. 4. Circle drawn from four control points; (a) Bézier curves (MetaPost); (b) harmonic interpolation.

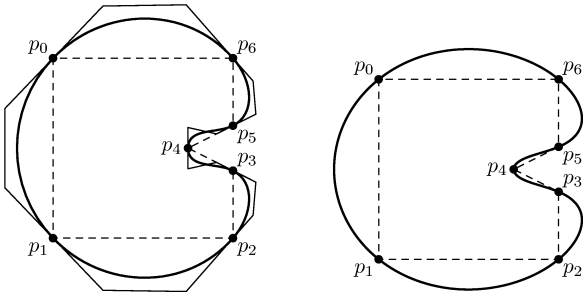

Fig. 5. Seven control points; (a) Bézier curves (MetaPost); (b) harmonic interpolation.

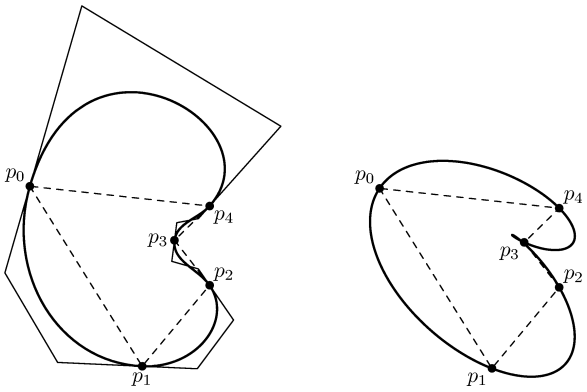

Fig. 6. Five control points; (a) Bézier curves (MetaPost); (b) harmonic interpolation.

5. Comparison

Bézier curves interpolate only the end points of the curve and are not necessarily closed. Harmonic interpolation defines curves that are closed and interpolate all the provided points. To provide a more meaningful comparison, we will compare piecewise smooth Bézier curves as produced by MetaPost with those produced with harmonic interpolation. The key for the figures is displayed in Figure 3.

Figure 4 shows a circle drawn with harmonic interpolation and Bézier curves.

The following figures illustrate how the curves compare when more control points are used. Figure 5 shows seven control points for harmonic interpolation and Bézier curves going through the same points. Figure 6 compares Bézier curves with harmonic interpolation for points that are assymetric and not equidistant.

The significant difference between the piecewise smooth Bézier curves and the curve produced by harmonic interpolation is that the piecewise smooth Bézier curves have C^1 or G^1 continuity while the curve produced by harmonic interpolation has C^∞ continuity.

It should be noted that Bézier curves cannot describe a circle while harmonic interpolation can. It should also be noted that harmonic interpolation always generates a closed curve, whereas the other techniques do not. MetaPost has been instructed to close the curve by repeating the first control point.

[1] P. Bézier, Numerical Control: Mathematics and Applications, Wiley, Chichester 1972, UK.
[2] W. Schuster, Math. Semesterber. **48**, 1 (2001).
[3] K. L. Nielsen, Methods in Numerical Analysis, second edition, MACMILLAN, New York 1964.
[4] W.-H. Steeb, Matrix Calculus and Kronecker Product with Applications and C++ Programs, World Scientific, Singapore 1997.
[5] W.-H. Steeb and Y. Hardy, Problems and Solutions in Quantum Computing and Quantum Information, World Scientific Publishing, Singapore 2004.
[6] W.-H. Steeb, Continuous Symmetries, Lie algebras, Differential Equations and Computer Algebra, World Scientific, Singapore 1996.
[7] P. Lejeune-Dirichlet, J. Reine Angew. Math. **4**, 157 (1829).
[8] A. Watt and M. Watt, Advanced Animation and Rendering Techniques, Addison Wesley, New York 1992.