Parametric Curves for Graphic Design Systems

P. K. Ghosh* and S. P. Mudur

National Centre for Software Development and Computing Techniques, Tata Institute of Fundamental Research, Bombay 400 005, India

One important requirement of a graphic design system is to be able to draw a visually pleasing curve passing through a sequence of discrete points in the Euclidean plane. These points are normally provided as input by the graphic designer or the graphic artist. Another important requirement is to be able to provide a modification technique for these curves which is simple and has control parameters which are conceptually natural to the designer or artist. This paper first reviews the curve definition as used in the METAFONT alphabet design system, discusses the drawbacks of the METAFONT curve form and then suggests a number of mathematical formulations for such curves and also the parameters for modification of these curves, which have been built into a graphic design system called Palatino. Palatino has been designed and implemented at the National Centre for Software Development and Computing Techniques of the Tata Institute of Fundamental Research in Bombay, India.

THE CURVE DESIGN PROBLEM

Given a sequence of points $(P_1, P_2, \ldots, P_n)$, arbitrarily positioned in a 2D Euclidean plane, the graphic designer or artist will draw with ease a smooth visually pleasing curve passing through all the given points in the specified order. This therefore is one of the prime requirements of a computer aided graphic design system. What is needed is a mathematical representation of a visually pleasing curve passing through a given sequence of points in a 2D plane. The mathematical representation should be computationally convenient and economical in storage.

The central difficulty of the above problem is that no definite (mathematical) criteria exist as to what constitutes a visually pleasing curve. Unfortunately continuity in the mathematical sense, though necessary, is not a sufficient condition for generating visually pleasing curves. Moreover visual pleasantness is very subjective and it is not likely that one can define mathematical criteria which will always result in such curves. However, what one would certainly like is that the mathematical formulation be such that, most of the time, a visually pleasant curve is generated. Additionally the artist or designer should be provided with the freedom to modify the parameters in the curve formulation so that the shape is locally altered. Knuth[1,2] was the first to formulate the definition of a 'most pleasing curve' in the METAFONT alphabet design system.† Before we discuss the criteria postulated by Knuth for generating 'most pleasing curves' let us briefly look at the basic mathematical form of the curve definition as used by Knuth and also by us.

The curve from P_1 to P_n passing through $P_2, P_3, \ldots, P_{n-1}$ is defined as a smooth piecewise cubic spline, each piece defined between P_k and P_{k+1}, where k ranges from 1 to $n-1$. The piecewise smoothness condition can be guaranteed by the zero-order (position) and the first-order (slope) continuity at the joints $(P_2, P_3, \ldots, P_{n-1})$. The parametric form of representation is chosen. A vector valued cubic spline $Q_k(t)$ is defined for each interval $P_k - P_{k+1}$, where t is a scalar variable ranging from to 0 to 1. $Q_k(t)$ has the following properties:

(i) $Q_k(t)$ is a polynomial of degree 3 or less and $P_k \le Q_k(t) \le P_{k+1}$

(ii) $Q_k(t=0) \equiv Q_k(0) \equiv$ position co-ordinate vector at P_k

$Q_k(t=1) \equiv Q_k(1) \equiv$ position co-ordinate vector at P_{k+1}

(iii) The tangent vectors of $Q_k(t)$ at both end points, that is $\dot{Q}_k(0)$ and $\dot{Q}_k(1)$ are specified.

After applying suitable scaling, rotation and translation the canonical form of $Q_k(t)$ can be described in the complex plane by the following expression:

$$Q_k(t) = -2t^3 + 3t^2 + \dot{Q}_k(0)t(1-t)^2 - \dot{Q}_k(1)t^2(1-t) \quad (1)$$

Note that we are linearly mapping the Cartesian co-ordinates of P_k and P_{k+1} such that P_k transforms into (0, 0) and P_{k+1} into (1, 0). Thus points on the actual curve are obtained by applying a suitable transformation to points on the canonical curve.

Let the entrant tangent make an angle θ_k and the exit one $-\phi_k$ with the real axis, so that

$$\dot{Q}_k(0) = r_k e^{i\theta_k} \quad \text{and} \quad \dot{Q}_k(1) = s_k e^{-i\phi_k} \quad (2)$$

where r_k and s_k represent the magnitudes of the tangent vectors at the start and end points of the kth segment, respectively. Since $Q_k(t)$ is in the complex plane it can also be written as

$$Q_k(t) = X_k(t) + iY_k(t) \quad (3)$$

Position continuity is guaranteed because $Q_k(1)$ and $Q_{k+1}(0)$ are both transformed into the same actual point, P_{k+1}. In order to ensure slope continuity it is essential ϕ_k be equal to θ_{k+1}. In order to see what other possibilities are open to us to get suitable criteria for generating visually pleasing curves let us study the vector equation

* Present address: C/o. Prof. D. E. Knuth, Fletcher Jones Professor, Dept. of Computer Science, Stanford University, Stanford, California 94305–2085, USA.

† The problem of curve shape representation such that a 'fair' curve is generated is one that has been worked on for a long time as part of engineering design systems.[3,4] For an extensive bibliography on curves and surfaces in computer-aided geometric design see Ref. 5. However the criteria for graphic artwork generation are different from those for engineering design.

CCC–0010–4620/83/0026–0312$04.00

(1) carefully. $\dot{Q}_k(0)$ and $\dot{Q}_k(1)$ can be chosen for generating the most pleasing curve. From equation (2) this means choosing r, s, θ and ϕ and at the same time satisfying the condition $\theta_k = \phi_{k-1}$ and $\phi_k = \theta_{k+1}$. Since there exists no mathematical definition for the condition that a curve be visually pleasing, all other criteria that r, s, θ and ϕ will be made to satisfy have a certain element of arbitrariness. Knuth suggests the following:

If not explicitly specified by the artist or designer, θ_k and ϕ_k are computed as follows:

(i) ϕ_1 is the slope of the line joining P_1 to P_2.
(ii) $\phi_k(1 \leq k \leq n-2)$ is the tangent direction (at P_{k+1}) to the circle passing through P_k, P_{k+1} and P_{k+2}, provided they do not lie on a straight line. ϕ_{n-1} is the tangent direction (at P_n) to the circle passing through P_{n-2}, P_{n-1} and P_n, provided they do not lie on a straight line.
(iii) $\theta_k(2 \leq k \leq n-1)$ is equal to ϕ_{k-1}.

If the designer or artist is not pleased with the resulting curve then he has the option of specifying additional points and/or the slopes at the joints. In fact Knuth himself says 'When drawing curves it is almost always desirable to specify beginning and ending angles, otherwise METAFONT will be forced to choose directions that have little probability of success'.

r and s are automatically computed as some functions of θ and ϕ satisfying the following conditions (the subscript k has been omitted without any ambiguity):

(i) If $\theta = \phi$ then the curve should approximate a circular arc which leads to the condition:

$$r = 2/(1 + \cos\theta), \qquad s = 2/(1 + \cos\phi)$$

(ii) If $\theta + \phi = 90°$ then the best fit could be a quadrant of an ellipse intercepted between the axes, and this leads to the condition:

$$r = 2\cos\theta/((1 + \cos 45°)\cos 45°),$$
$$s = 2\cos\phi/((1 + \cos 45°)\cos 45°)$$

Knuth then intuitively obtains the general expressions for any arbitrary θ and ϕ as

$$r = \left|\frac{2\sin\phi}{[1 + \cos(\theta+\phi)/2]\sin(\theta+\phi)/2}\right|$$
$$s = \left|\frac{2\sin\theta}{[1 + \cos(\theta+\phi)/2]\sin(\theta+\phi)/2}\right| \qquad (4)$$

which automatically satisfy the above stated conditions.

Since small values of r and s usually make the curve turn too sharply near the end points, and large values make it wander erratically, METAFONT's standard mode of operation ensures the following

$$0.5 \leq r, \quad s \leq 4.0$$

Figure 1 shows typical curves drawn according to the METAFONT prescription.

METAFONT produces quite pleasant curves, but under some conditions only. There are many cases when it fails to do so. The only solution offered by METAFONT in such cases is to increase the number of defining points, and if necessary specify the slope at each of the joints. Our experience has been that artists do not seem to like the idea of having to specify an extra large number of points in some cases only. Nor is the idea of slope angle

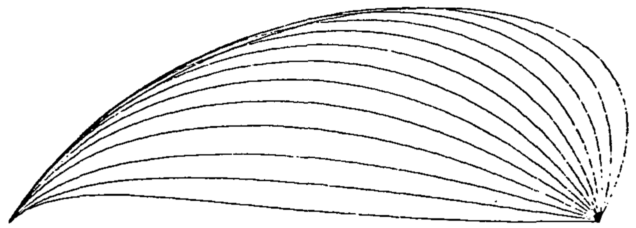

Figure 1. METAFONT curve: $\theta = 60°$, $\phi = \langle 0°, 10°, \ldots, 120° \rangle$, $0.5 \leq r, s \leq 4$.

at each of the joints a natural form of shape description or modification for these artists. We shall shortly discuss the main drawbacks of the METAFONT curve. It is however worth noting here that the design domain of METAFONT was primarily the letters of the Latin script. Fortunately these letters are graphically simple as compared to letters of other scripts, say Indian, or other graphic design tasks like artwork generation.

Palatino[6] is a graphic design system currently under development at the National Centre for Software Development and Computing Techniques at the Tata Institute of Fundamental Research in Bombay. The system has been named after Giovanbattista Palatino Cittadino Romano, who was one of the best known calligraphers of sixteenth century Italy; a renaissance man who will be remembered also for his effort to combine mathematics in constructing type founts. Palatino has been used successfully to design letters of the Indian scripts as well as artwork for posters, covers, logos, greeting cards etc. The METAFONT cubic spline which was first built into Palatino proved inadequate for many of the tasks listed above. Therefore new curve forms and modification techniques had to be built in. It does not lie within the scope of this paper to describe the design facilities of the Palatino system. In the rest of this paper we shall briefly discuss the drawbacks of the METAFONT curve and then discuss the curve forms and modification parameters as built into Palatino.

DRAWBACKS OF THE METAFONT CURVE

(1) METAFONT fails to generate a pleasant curve automatically when the curve follows a zig-zag path, that is when the curve changes its sign of curvature frequently from clockwise to anticlockwise and vice versa. Except for the letter S the other letters of the Latin script do not

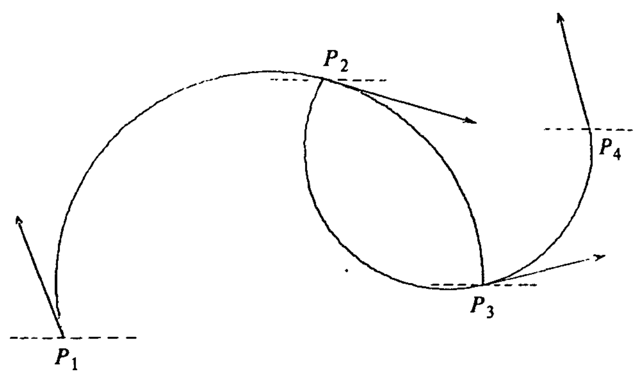

Figure 2. Gradient calculation in METAFONT.

have curves having frequent changes in the sign of curvature. This however cannot be said to be true of the letters of the Indian script, or of other graphic arts tasks.

The problem is the method used to calculate θ and ϕ, as can be seen in Fig. 2. Knuth himself states that 'Most of these curves are rather arbitrary, so that you are taking a chance if you expect METAFONT to change directions drastically'.

(2) The METAFONT curve does not behave very well when θ and ϕ are of the opposite sign. Using equations (3) and (4) it can be shown that $Y(t) = 0$ at $t = \frac{1}{2}$. That is the curve always intersects the chord joining the end points nearly in the ratio 1:3 when $|\theta| > |\phi|$ or 3:1 when $|\theta| < |\phi|$, irrespective of the magnitudes of θ and ϕ. Figure 3(a) shows several such curves that have $\theta = +60°$ and several negative values of ϕ. ($\phi = -60°$ is not drawn, however, as this leads us to the case of the curve singularity which we shall discuss later.)

Because of the preordained intercept at $t = \frac{1}{2}$, METAFONT does not always have its predictable pleasantness. Consider the example of Fig. 3(b) where the METAFONT curve segment (solid) passes through the points P_{k-1}, P_k, P_{k+1}, P_{k+2}. If the point P_{k+2} is shifted to P'_{k+2} then, since the curve has to intercept the chord $P_k - P_{k+1}$ at the same point, some odd kinks are introduced, as shown by the dotted curve. A more pleasant curve can be generated only if the intercept point can be shifted more towards P_{k+1}. If the ideal case of this shift occurring automatically cannot be built into the formulation then at least the option should be available to the designer. Unfortunately METAFONT does not provide this freedom.

(3) When θ and ϕ are of opposite signs and are very nearly equal, that is $\theta + \phi \simeq 0$, the curve has a singularity which results in an oscillation of infinite magnitude because both r and s become infinitely large (see equation (4)). The METAFONT solution for this problem is to arbitrarily set the values of r and s to 2. Once again no choice is given to the user to modify the curve, say by changing the amplitude.

CHOOSING TANGENT VECTORS

As already stated, except for the requirement that $\theta_k = \phi_{k-1}$ at the joints, there is a lot of scope for experimenting with the values of θ, ϕ, r and s of equation (2). Here we shall suggest some alternative choices for these and then show their effect in some typical problem situations. These techniques have been built into Palatino. All the curves shown in this paper have been produced using the Palatino system.

Choosing gradients

The reason why METAFONT does not automatically produce good curves when the defining points follow a zig-zag pattern is because the choice of the slope angle at the joints depends only on a single point on either side of the joint. This means that the zig-zag nature of the curve is not recognized at all. The alternative we suggest is as follows:

(i) θ_1 is the tangent direction (at P_1) to the circle passing through P_1, P_2, P_3, provided they do not lie on a straight line.

(ii) $\phi_k (1 \le k \le n-3)$ is the arithmetic mean of the tangent direction (at P_{k+1}) to the circle passing through P_k, P_{k+1}, P_{k+2} and the tangent direction (at P_{k+1}) to the circle passing through P_{k+1}, P_{k+2}, P_{k+3}, provided they do not lie on straight lines. ϕ_{n-2} and ϕ_{n-1} are the tangent directions (at P_{n-1} and P_n, respectively) to the circle passing through P_{n-2}, P_{n-1}, P_n, provided they do not lie on a straight line.

(iii) $\theta_k (2 \le k \le n-1)$ is equal to ϕ_{k-1}.

Figure 4 compares the METAFONT curve (dotted) with the Palatino curve (solid) drawn using the above

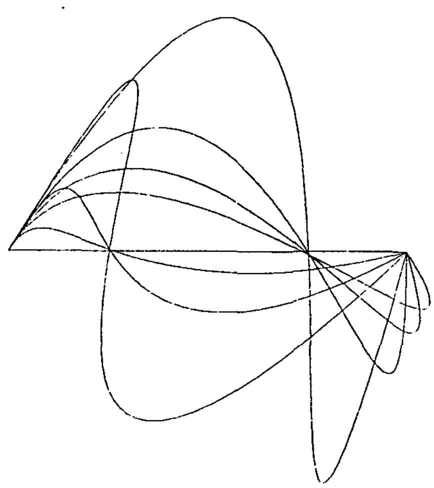

Figure 3(a). METAFONT curve: $\theta = 60°$, $\phi = \langle 0°, -15°, \ldots -120° \rangle$ ($\phi = -60°$ not shown).

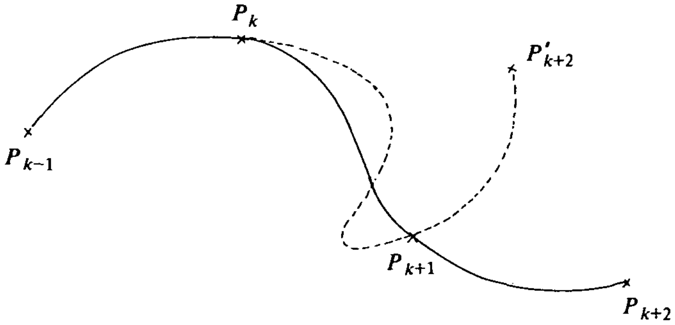

Figure 3(b). The introduction of unpleasant kinks in the METAFONT curve.

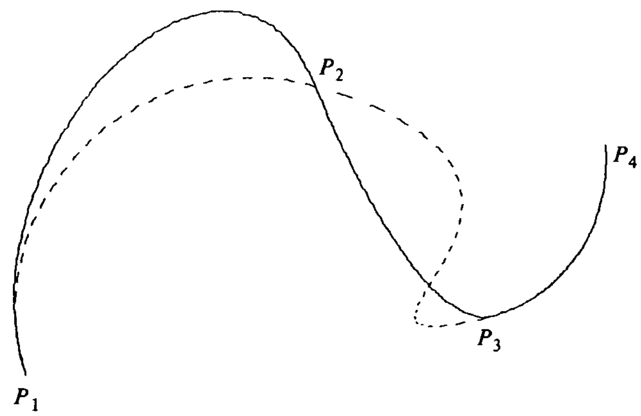

Figure 4. Palatino curve using the average gradient method (solid) METAFONT curve (dotted).

averaging mechanism for computing the gradients at the joints.

Choosing tangent vector magnitudes

The neo-spline. METAFONTS's criteria that the curve be a part of a circle when $\theta = \phi$ and a quadrant of an ellipse when $\theta + \phi = 90°$ are reasonably good. While satisfying the above two criteria, we have chosen the following expressions for r and s,

$$r = \left|\frac{2}{(1 + \cos\psi/2)}\sqrt{\left(\frac{2\cos\theta\sin\phi}{\sin\psi}\right)}\right|;$$

$$s = \left|\frac{2}{(1 + \cos\psi/2)}\sqrt{\left(\frac{2\sin\theta\cos\phi}{\sin\psi}\right)}\right| \quad (5)$$

where $\psi = \theta + \phi$.

Restricting the r and s values between $\frac{1}{2}$ and 3, this neo-spline conforms with the METAFONT spline in most cases (Fig. 5(a)). However the r and s values increase at a fast rate when ψ approaches 180° (Fig. 5(b)). A remarkable superiority of neo-spline over the METAFONT spline is when θ and ϕ are of opposite signs (Fig. 5(c)). This is basically due to the term $\sin\psi$ in the denominator of equation (5) under the radical sign (the METAFONT spline uses a $\sin\psi/2$ term instead).

The change-ringing spline. It is reasonable to assume that r and s should be proportional to the vectors P_0X and XP_1,

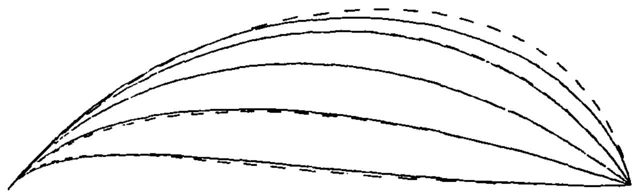

Figure 5(a). Neo-spline (solid): $0.5 \le r, s \le 3$. METAFONT curve (dotted): $0.5 \le r, s \le 4$. $\theta = 50°$, $\phi = \langle 0°, 20°, \ldots, 80° \rangle$.

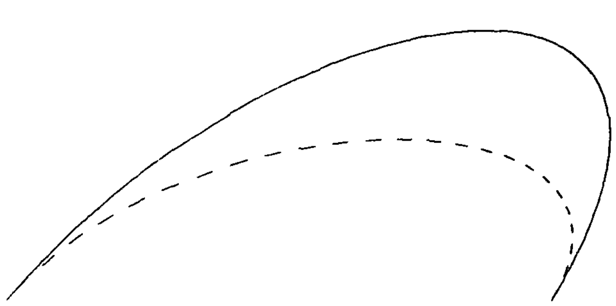

Figure 5(b). Neo-spline (solid) vs METAFONT curve (dotted): ψ approaching 180°, $\theta = 50°$, $\phi = 120°$.

Figure 5(c). Neo-spline (solid) vs METAFONT curve (dotted): $\theta = 30°$, $\phi = -60°$.

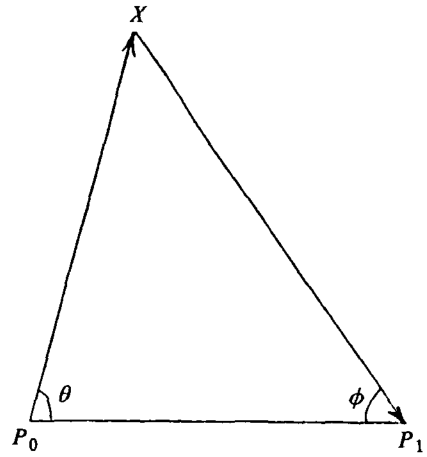

Figure 6. r, s calculation for the change-ringing spline.

respectively, where X is the intersection point of the two tangents drawn at an angle of θ and ϕ, respectively, with the joining chord (Fig. 6), because the tangent vector magnitude determines how far the curve follows the direction of the tangent vector before turning to the direction of the other tangent vector. Since $P_0P_1 = 1$,

$$P_0X = \left|\frac{\sin\phi}{\sin(\theta + \phi)}\right| e^{i\theta}; \qquad XP_1 = \left|\frac{\sin\theta}{\sin(\theta + \phi)}\right| e^{-i\phi}$$

we choose

$$r = \alpha\left|P_0X\right| = \alpha\left|\frac{\sin\phi}{\sin(\theta + \phi)}\right|$$

and

$$s = \beta\left|XP_1\right| = \beta\left|\frac{\sin\theta}{\sin(\theta + \phi)}\right|$$

where α and β are two positive scalar quantities. Again assuming that $r = s$ when $\theta = \phi$, which requires $\alpha = \beta$, the final forms of r and s become

$$r = \alpha\left|\frac{\sin\phi}{\sin(\theta + \phi)}\right|; \qquad s = \alpha\left|\frac{\sin\theta}{\sin(\theta + \phi)}\right| \quad (6)$$

It is interesting to note the following points:

1. If $\alpha = 2$, the curve represents a parabolic arc.
2. The curvature of the spline is away from the joining chord if $\alpha > 3$.
3. No loop occurs if $\alpha \le 6$.

This type of spline curve with a specific α value (here in this example 1.5) is shown in Fig. 7(a).

The main advantage of this curve over the METAFONT curve is the added degree of freedom α. Its value can be easily adjusted to produce pleasant curves of varying degrees (Fig. 7(b)). Since the art of producing every possible variation in the ringing of a peal of bells is called 'change-ringing' this curve has been designated as the change-ringing spline. Figure 7(c) compares the use of this curve with the METAFONT curve.

It is important to note that the ratio r/s is $|\sin\phi/\sin\theta|$ for both the METAFONT spline and the change-ringing spline. Therefore METAFONT curves can be very closely reproduced by properly choosing the value of α. For example $\alpha = 1.31$ approximately reproduces a METAFONT curve for $\theta = \phi = 60°$.

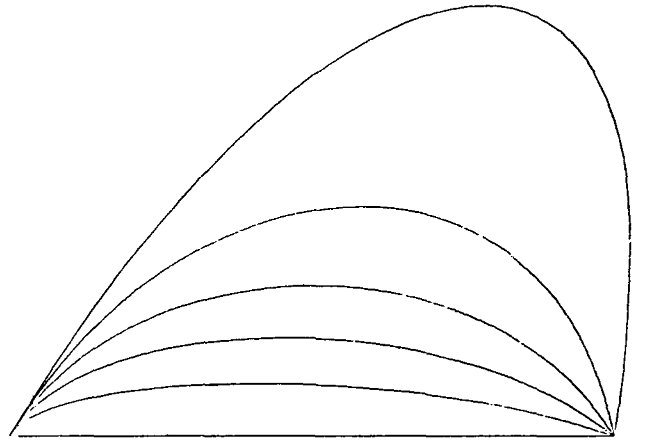

Figure 7(a). Change-ringing spline $\alpha = 1.2$, $\theta = 60°$, $\phi = \langle 0°, 20°, \ldots, 100° \rangle$.

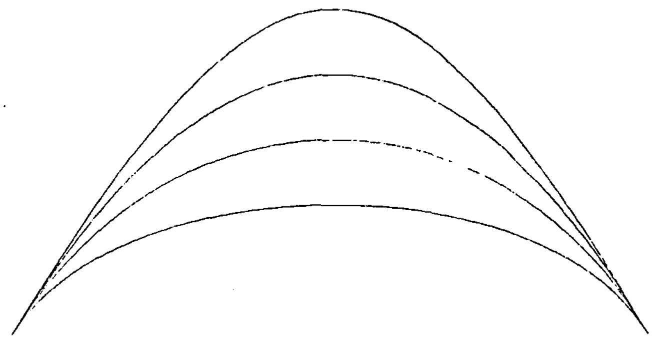

Figure 7(b). Change-ringing spline ($\theta = \phi = 60°$) and varying α values. $\alpha = \langle 0.5, 1.0, 1.5, 2 \rangle$.

Figure 7(c). The letter A using the METAFONT curve without specifying explicit slopes (left) and the same letter using the change-ringing spline of Palatino (right).

The xeno-spline. When $\theta + \phi$ is close to zero, rather than set $r = s = 2$ as in METAFONT, we find, a sinusoidal curve serves as a better alternative. (It is, in fact, seen that some artists, when asked to draw a pleasing curve with the angles $\theta = -\phi$, draw curves which look almost like sinusoidal curves.) For satisfying the design criteria at the end points, one may choose an expression of the form:

$$Q(t) = t + \mathrm{i}\frac{1}{2\pi}(-1)^n (\tan\{|\theta| + (|\phi| - |\theta|)\, t\} \sin 2\pi t) \quad (7)$$

n being 0 or even for starting above the joining chord and odd otherwise.

This form of the curve can be seen to be totally different from the spline expression of equation (1) and may be used only near the singularity region, i.e. for $\theta + \phi \simeq 0$. This type of curve we shall designate as a xeno-spline, since the combining form 'xeno-' means a guest, stranger or a foreigner.

MODIFYING THE SHAPE OF THE CURVE

We have already stated that specification of gradients is something that is not very natural to the graphic artist or designer. In our opinion this technique of shape specification and modification is not the most suitable one. In this section we suggest three different methods of shape modification.

Let us recall the form of the curve definition as used in equation (3)

$$Q(t) = X(t) + \mathrm{i}\,Y(t)$$

Shape control

In this section a very simple mechanism to modify the overall shape of a curve segment is presented, regardless of the nature of turns in the entire curve. This mechanism is very similar to the one presented in Refs 7 and 8.

Note that the function

$$f(t) = \sin^2(\pi t/2)$$

satisfies the properties

$$f(0) = 0, \qquad f(1) = 1$$
$$f(0) = 0, \qquad \dot{f}(1) = 0$$

Now suppose that $Q(t)$, $0 \leq t \leq 1$ is any curve segment, and we construct a compound vector function

$$Q_\tau(t) = (1 - \tau)Q(t) + \tau f(t) \quad (8)$$

where τ is some arbitrarily chosen scalar constant. Since $\dot{f}(t)$ is zero at the end points, the spline $Q_\tau(t)$ satisfies all the desired properties of the original spline function, i.e. the specified conditions on $Q(t)$ and $\dot{Q}(t)$ at both ends of the segment do not change.

The quantity τ may be left at designer's control and the shape of the curve can be modified between any two given points. We shall designate τ as the tension factor, as positive values of τ may be said to tighten (increase the tension) the segment whereas negative values of τ may be said to loosen (reduce the tension) the segment. $\tau = 0$ produces the original curve $Q(t)$. Figure 8(a) shows the METAFONT spline modified using the tension

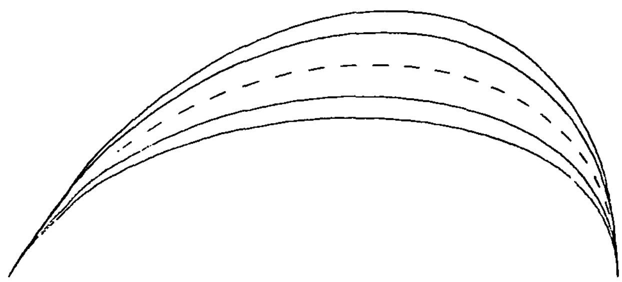

Figure 8(a). Tension factor (τ) modified METAFONT curve (solid), unmodified METAFONT curve, i.e. $\tau = 0$ (dotted): $\tau = \langle 0.25, 0.15, -0.15, -0.25 \rangle$, $\theta = 60°$, $\phi = 90°$.

Figure 8(b). The letter A of Fig. 7(c) modified using shape factor $\tau = 0.15$.

factor for different values of τ. Figure 8(b) shows the letter A of Fig. 7(c) modified using $\tau = 0.15$.

Although, for most of the graphic design curves continuity of the second derivative at the end points is not essential, nevertheless, we would like to point out that the form

$$f(t) = 4\sin^3(\pi t/2) - 3\sin^4(\pi t/2)$$

has its second derivatives also vanishing at both the ends. So using the above function $Q(t)$ would be modified in such a way that the first and the second derivatives at the end-points of $Q_\tau(t)$ would agree with those of $Q(t)$ in direction, though their magnitudes would be multiplied by a factor $(1 - \tau)$.

Amplitude control

In order to introduce a control over the amplitude we suggest the following amplitude function

$$A(t) = (1 + \sin^2 \pi t)^\varepsilon$$

to be multiplied with the Y-part of $Q(t)$ alone, keeping the X-part intact. That is, the new curve is now of the form

$$Q_\varepsilon(t) = X(t) + \mathrm{i}A(t)Y(t) \tag{9}$$

ε is any real number. $A(t)$ is the amplitude function which is always positive and shrinks or stretches the curve in the Y direction (in the canonical representation) depending on whether ε is negative or positive. ε is called the amplitude modulation factor. Note that the end conditions of $Q(t)$ are met as $A(t)$ satisfies the following properties independent of the value of ε, all of which are necessary and sufficient:

$$A(0) = 1, \qquad A(1) = 1$$
$$\dot{A}(0) = 0, \qquad \dot{A}(1) = 0$$

Figure 9 shows the use of this on the xeno-spline as defined in equation (7). The condition $|\varepsilon| \leq 1.5$ results in pleasing curves in most cases.

Intercept control

In order to achieve the freedom of shifting the chord intercept point we introduce a local 'strain' of value $\chi \sin^2(\pi t)$ in the X-part of $Q(t)$, by replacing t by $t + \chi \sin^2(\pi t)$, and our curve now has the form:

$$Q_\chi(t) = X(t + \chi \sin^2(\pi t)) + \mathrm{i}Y(t) \tag{10}$$

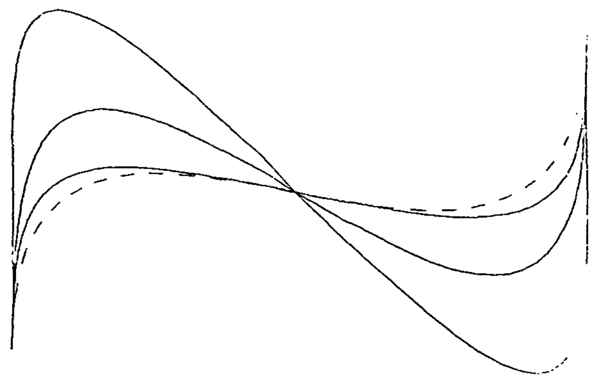

Figure 9. Amplitude modulated xeno-spline (solid), METAFONT curve (dotted): $\varepsilon = \langle -1, 0, 1 \rangle$, $\theta = -\phi = 60°$.

Once again note that the simple trigonometric function

$$f(t) = \sin^2(\pi t)$$

satisfies the properties:

$$f(0) = 0, \qquad f(1) = 0$$
$$\dot{f}(0) = 0, \qquad \dot{f}(1) = 0$$

Clearly then $Q_\chi(t)$ satisfies all the desired properties of the original spline function, i.e. the specified values of $\dot{Q}(t)$ and $Q(t)$ at both ends of the segment do not change. Because of the effect that the value of χ has on the curve shape it will be called the shear factor. The shear factor can be manipulated for shifting the chord intercept point on either side of $t = \frac{1}{2}$. Note that the parameter χ remains concealed or hidden near the two ends and becomes active only in the intermediate region of the segment.

A positive value of χ shifts the intercept point towards the $t = 1$ end, whereas a negative value shifts it towards the $t = 0$ end; $\chi = 0$ reproduces the original spline. In Fig. 10 we have shown how small values of χ help

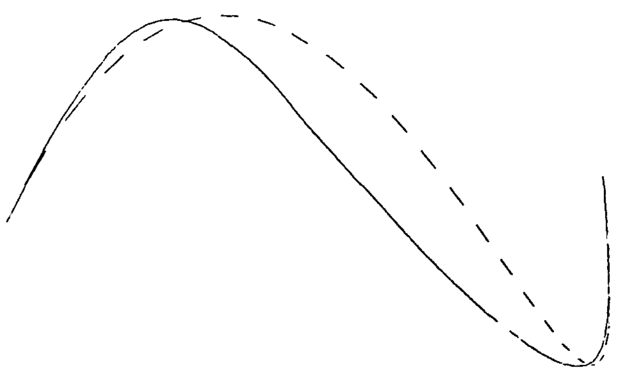

Figure 10(a). Shear factor ($\chi = -0.12$) modified METAFONT curve (solid), unmodified METAFONT curve (dotted): $\theta = 60°$, $\phi = -90°$.

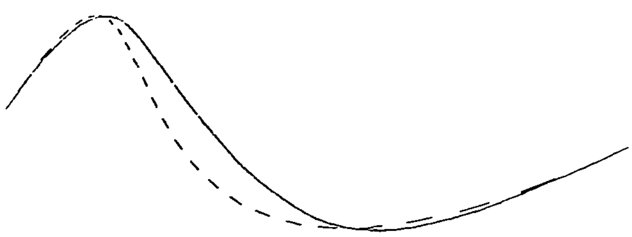

Figure 10(b). Shear factor ($\chi = 0.1$) modified METAFONT curve (solid), unmodified METAFONT curve (dotted): $\theta = 60°$, $\phi = -30°$.

obtaining more pleasant curves than the ones by METAFONT without, of course, incurring any noticeable changes near the ends. If χ is not directly set by the designer, a suitable criterion for the choice of χ has to be sought for as a function of θ and ϕ as well as of the chord lengths P_kP_{k+1} and $P_{k+1}P_{k+2}$. $|\chi| \leq 0.3$ results in pleasant curves in most cases.

CONCLUSIONS

In this paper we have tried to provide satisfactory solutions to two important requirements of graphic design systems:

(i) mathematical representations for visually pleasing curves which match the artist's 'feel' to the way their shapes behave.

(ii) control parameters other than the defining data points, by which the artist or designer can implement his 'desired' modification in shape. The control parameters should be natural to the designer. Too few parameters simplify the designer's task at the expense of not having enough control over the range of varying shapes. Too many parameters cause confusion.

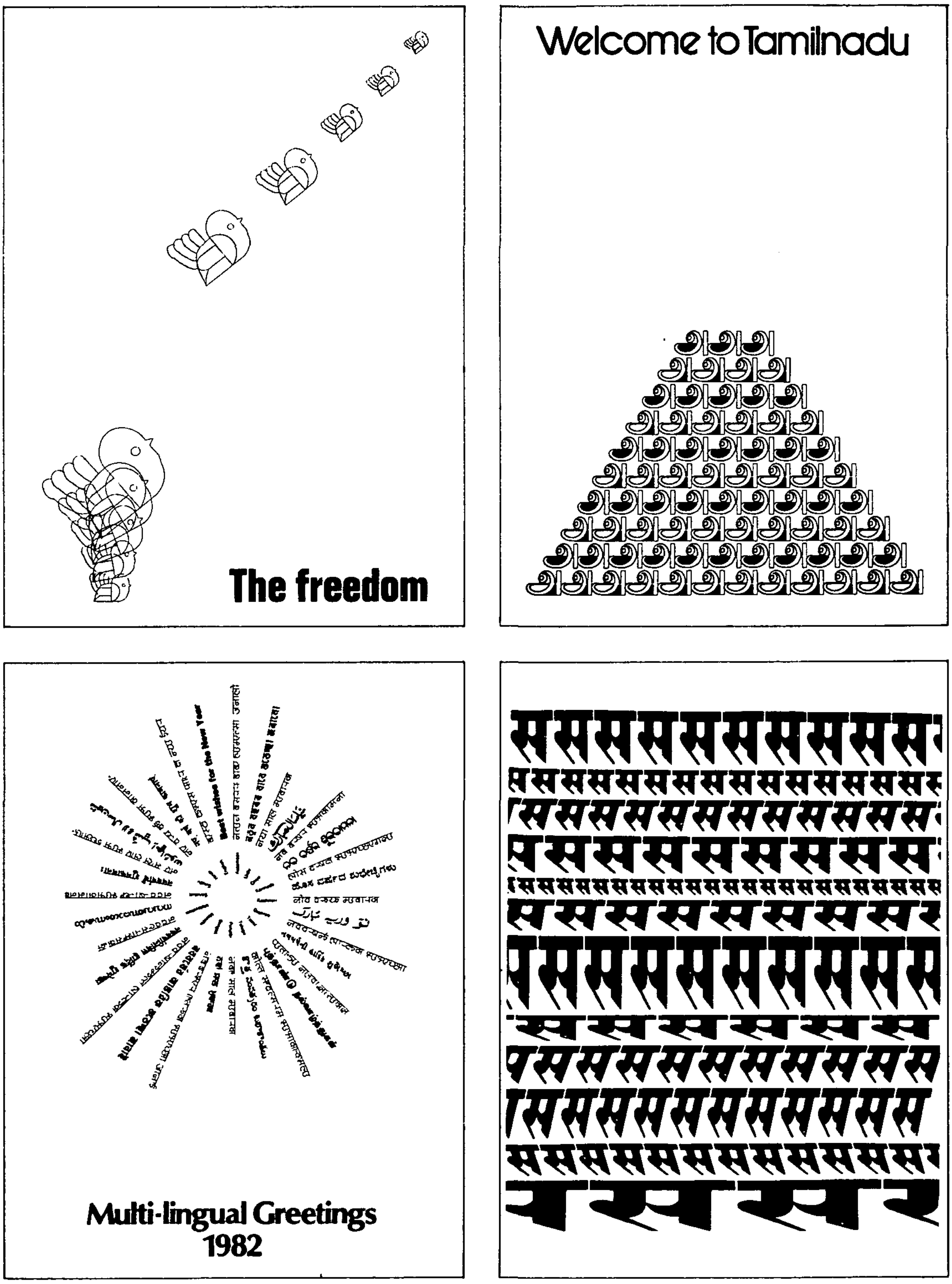

Figure 11. Some graphic artwork examples designed using Palatino (designer: R. K. Joshi).

The drawbacks of the only known curve definition for such tasks, namely the METAFONT spline, have been discussed.

Three new curve forms have been introduced: the neo-spline which, though very similar to the METAFONT spline, behaves remarkably better when the start and end slopes are opposite in direction; the change-ringing spline which has a user controllable multiplicative factor in the expression for the magnitudes of the tangent vectors at the end points and the xeno-spline to take care of the near singular domain of the other curve forms.

Three independently specifiable control parameters have been introduced: the tension factor τ which can be used to 'tighten or loosen' a segment of the curve; the amplitude modulation factor ε for changing the amplitude of the curve in a direction orthogonal to the chord and the shear factor χ for shifting the intercept point along the chord or for generally 'pushing' the curve segment towards either end.

These curve forms and control parameters have been built into a graphics design system called Palatino and have been successfully used for a wide variety of artwork design and production tasks. Figure 11 shows some examples of design using Palatino.

Acknowledgements

It is a great pleasure to acknowledge Prof. R. Narasimhan's keen interest and encouragement in our work. We gratefully thank Shri R. K. Joshi, a professional calligraphist, for using Palatino and providing the very essential feedback without which this work would not have been completed to its current stage. We also wish to thank Dr. P. Ghosh, Mr. P. Ayub and particularly Dr. N. C. Rana for the extensive discussions, suggestions and clarifications which helped crystallize the ideas in this paper.

REFERENCES

1. D. E. Knuth, Mathematical typography, *Bulletin of the American Mathematical Society* **1** (2), 337–372 (1979).
2. D. E. Knuth, METAFONT, a system for alphabet design, part 3 of *TEX and METAFONT: New Directions in Typesetting*, American Mathematical Society and Digital Press (1979).
3. R. E. Barnhill and R. F. Riesenfeld, *Computer Aided Geometric Design*, Academic Press, New York (1974).
4. I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester (1979).
5. B. A. Barsky, Computer-aided geometric design: a bibliography with keywords and classified index, *IEEE Computer Graphics and Applications* **1** (3), 67–109 (1981).
6. S. P. Mudur, P. K. Ghosh and R. Sujatha, Software development for computer typography and type design, *NCSDCT Research Review*, 13–26 (1981).
7. S. A. Coons, Modification of the shape of piecewise curves, *Computer Aided Design* **9** (3), 178–180 (1977).
8. D. G. Schweikert, An interpolation curve using a spline in tension, *J. Math. & Phys.* **45**, 312–317 (1966).

Received September 1982