Blended Elliptic Arc Splines

ALLWYN KUMARESAN AND GANAPATHY SUBRAMANIAN L. RAMACHANDRAN
Department of Aerospace Engineering, SRM Institute of Science and Technology, Chennai 603203, India
Corresponding author: Allwyn Kumaresan (allwynk@srmist.edu.in)

ABSTRACT Arcs of conics are natural looking, smooth and gradual in curvature transition. The method in this paper proposes piecewise interpolating splines, which fit elliptic arcs between consecutive points in a given set of planar data points. Tangents of the curve at the data points are determined by a local procedure inspired by Akima’s formulation. Elliptic arcs that suitably fit the orientation of these tangents are fit at all segments between control points, giving a $G^1$-continuous curve fit. In another option, the curve segments on both sides of the control points are made curvature continuous by blending with circular arcs; resulting in a $G^2$-continuous curve. These curves avoid unwanted inflections and undulations; and are capable of recognizing and generating straight-line paths and sharp corners; and also have an option to specify tension of curve at individual segments. Moreover, this is computationally less expensive. Having these desirable properties, this method can be used as an effective curve fitting tool for multiple-valued functional approximation, motion path tracing, 2D geometric modeling (including typography), etc.

INDEX TERMS Blended elliptic arc (BEA) splines, curve fitting, geometric modeling, motion path tracing, typography.

I. INTRODUCTION Interpolating curves find their application in wide range of fields. However, the requirements of every field are different. In functional approximation applications, the curve used should be intuitive; capable of recognizing straight lines and curves, without any additional user input than the data points. Sometimes, $G^1$-continuous curves perform better than $G^2$-continuous curves in this application [1]–[4]. Geometric modeling is another completely different field where interpolating curves are applied. For modeling of vector images (including fonts), sampling of real world data (e.g. sculpture); the ultimate requirement is an aesthetically pleasing curve [5] (irrespective of its degree of the curve continuity), capable of producing wide range of shapes — ranging from circles to straight lines — with ease. However, design of vehicles (especially aerodynamic shapes) and machine parts expect some degree of geometric continuity. For example, the curve used in airfoil design should have at least $G^2$-continuity; which ensures the flow over it is streamlined with no sudden change in angular acceleration. Motion tracking (e.g., tracking movement of animals, birds, etc., from discrete position data) is another application where $G^2$-continuity is preferred; assuming that the moving object would not make sudden changes in linear and angular acceleration. Since the moving object controls its velocity and acceleration itself, parametric continuity of curve is unnecessary. However, in motion trajectory planning (e.g., robotic arm trajectories, camera paths, key-frame animation etc.) applications, the interpolating curve is required to have, a parametric continuity (preferably at least $C^2$-continuity that ensures smooth transition of velocity and acceleration). Moreover, the curves should avoid sharp corners [6] (inflections and undulations are allowed for this purpose). Smoothness, being the common requirement of all applications, however, is defined by different parameters like geometric continuity, parametric continuity, minimum energy, and minimum variation of curvature; as per the requirements of the application. Moreover, the curves are expected to be invariant under translation, scaling and rotation (rotation property is not required for single-valued functional approximation).

Polynomial curves are mostly used as interpolating curves; however, the curve fitting technique used varies. Cubic Hermite’s interpolation is popular in functional approximations for it is more interactive [7]. The shape preserving curve formulation [3], [4] and Akima’s formulation [1] are some methods of this type, which produce good results. However, these formulations are not useful in geometric modeling and motion path applications. Bézier curves [8], [9], on other hand, are widely used in animation and vector graphics (including fonts); for the option of easy shape modifications. The continuity of these curves is controlled globally;
hence, a change at one control point will force a recalculation of curve slopes at all the points. Although there is an option for inputting slope and tension at individual control points, doing so will affect the continuity of the curve [7]. B-splines [10], [11] are curves that overcome this limitation and are able to present any n degree of smoothness. However, they are computationally expensive compared to many other options [7]. There are some more sophisticated global curve optimization options like minimum energy curves [12] and minimum variation of curvature [13], which can be applied for more demanding cases. Biarc curves [14] and circle splines [6], [15], [16] are other options, which does not use polynomial curves; however, capable of producing smooth curves, especially circles (which the polynomial curves fail to describe perfectly).

Applications are many and the available interpolating curves for them are also many. However, there is still scope for new curve definitions, which have advantages over the other counterparts, at least in few aspects. Here we present one such interpolating curve formulation called Blended Elliptic Arc (BEA) splines, which fits elliptic arcs at individual segments. Tangents at control points are determined by a new local procedure (inspired by Akima’s formulation [1]); which is presented in section II. Formulations for a $G^1$-continuous curve fit option and an improved version with $G^2$-continuity with a little more complexity are presented in section V and VI respectively. Applicability of this formulation in multiple-valued function approximation, geometric modeling and motion path path tracing are discussed in section VIII.

II. EVALUATION OF TANGENTS OF BEA AT DATA POINTS

In most cases, the direction of curve at any data point is dependent on its neighboring points, rather than on the entire set of points. Based on this, Akima [1] formulated a local procedure to find slope of the interpolating curve at any given point, using the slopes of two segments on both sides of the point as follows:

If $\delta_i$ is the slope of the segment between the points $p_i = (x_i, y_i)$ and $p_{i+1} = (x_{i+1}, y_{i+1})$, then the slope of tangent of the interpolating curve at $p_i$ is given by

$$t_i = \frac{w_0 \delta_{i-1} + w_1 \delta_i}{w_0 + w_1}$$

where, $w_0 = |\delta_{i+1} - \delta_i|$ and $w_1 = |\delta_{i-1} - \delta_{i-2}|$.

There is a case of discontinuity when $\delta_{i+1} = \delta_i$ and $\delta_{i-1} = \delta_{i-2}$, where the slope of the curve through (1) become undefined. At this condition Akima recommends shifting to a different averaging formula $t_i = (\delta_{i-1} + \delta_i)/2$; however, found not continuous with (1) [17]. Moreover, this formulation is applicable only for single-valued functional approximation. Hence, this formulation is not directly applicable for all kinds of curve fitting applications. A modified local procedure, which would overcome these limitations while preserving the same superior slope predicting capability, would be a desirable method.

The basic idea behind Akima’s weighted average formulation (1) is making the curve more inclined towards the flatter side (i.e., the side where the change in orientation is less compared to the other). The problem here is: the change in slope corresponding to a change in orientation is not same in all directions. For example, consider a case where (0,0), (1,0) and (2,1) represent $p_{i-2}$, $p_{i-1}$ and $p_i$ respectively; which would produce $w_1 = 1$ corresponding to a change in orientation of 45°. Now consider another case where (0,0), (1,1) and (1,2) represent $p_{i-2}$, $p_{i-1}$ and $p_i$ respectively; which would produce $w_1 = \infty$ corresponding to a change in orientation of the same 45°. This property is not desirable for applications other than single-valued functional approximation. Moreover, slopes can only define orientations in two quadrants (−90° to 90°). To overcome this problem, we propose a formulation, which uses the change in angular orientation itself to define the weights. Fig. 1 depicts the nomenclature used in this procedure. The procedure is presented in the sequential order of steps in evaluation, for better understanding (also for easy programming).

Let a data point be represented by the position vector $p_i = x_i \mathbf{i} + y_i \mathbf{j}$. The line segment joining two consecutive points $p_i$ and $p_{i+1}$ is represented by

$$l_i = l_{0i} \mathbf{i} + l_{1i} \mathbf{j} = p_{i+1} - p_i$$

The orientation of this vector is identified in angular form in the 2D space using the four-quadrant inverse tangent function as

$$\eta_i = \arctan 2(l_y, l_x)$$

Then, the change in orientation from $l_{i-1}$ to $l_i$ (i.e., change in orientation happening at point $p_i$) is given by

$$O_i = \arctan 2 (\sin(\eta_i - \eta_{i-1}), \cos(\eta_i - \eta_{i-1}))$$

At this step, one might wonder, why such a calculation required for something which could be easily identified using dot product of the unit vectors. However, this is required to define whether the change in orientation of $l_i$ from $l_{i-1}$ is
clockwise or counterclockwise. If \( O_i \) through (4) — gets a negative sign, it says the change in orientation is clockwise and vice versa (accordingly, in Fig. 1 \( O_{i-1} \), \( O_i \) and \( O_{i+1} \) will have negative signs).

The angles \( \alpha_i \) or \( \beta_{i-1} \) can define the direction of the tangent of the curve at \( p_i \), since the directions of \( I_i \) and \( I_{i-1} \) are already known (see Fig. 1). Hence, the formulation is expected to determine the percentage of \( O_i \) taken up by \( \alpha_i \) or \( \beta_{i-1} \). From the knowledge obtained from Akima’s formulation, the ratio of \( \alpha_i \) to \( \beta_{i-1} \) should reduce as the ratio of \( O_{i+1} \) to \( O_{i-1} \) reduces and vice versa. However, the ratios need not be equal, for the scheme is required to make \( \alpha_i \) approach zero as \( O_{i+1} \) approach zero, regardless of the value of \( O_{i-1} \). In a simple statement, the curve should be more oriented towards the flatter side and should be a straight line on the side, where no change in orientation is identified. Considering these requirements, \( \alpha_i \) and \( \beta_{i-1} \) are defined as

\[
\alpha_i = -w_{\alpha_i} O_i \quad \beta_{i-1} = -w_{\beta_{i-1}} O_i
\]

where,

\[
w_{\alpha_i} = \frac{\tan \left( \frac{\pi - |O_{i-1}|}{2} \right)}{\sqrt{\tan \left( \frac{\pi - |O_{i-1}|}{2} \right) + \tan \left( \frac{\pi - |O_{i+1}|}{2} \right)}}
\]

\[
w_{\beta_{i-1}} = \frac{\tan \left( \frac{\pi - |O_{i+1}|}{2} \right)}{\sqrt{\tan \left( \frac{\pi - |O_{i-1}|}{2} \right) + \tan \left( \frac{\pi - |O_{i+1}|}{2} \right)}}
\]

provided \( O_{i+1} \) and \( O_{i-1} \) are substituted in units of radian.

The common sign convention followed for \( \alpha_i \) and \( \beta_i \) is, a positive value specify that the angle is above the vector \( I_i \) (i.e., in the side of the counterclockwise normal to \( I_i \) and vice versa. This convention will be proved useful during the curve fit and for this purpose we put a negative sign in \( \alpha_i \) and \( \beta_{i-1} \) formulae (5). Again, the unconventional definitions of the weight functions \( w_{\alpha_i} \) and \( w_{\beta_{i-1}} \), have peculiar purposes. Firstly, if the change in orientation at one side of \( p_i \) is zero, the weights should enable a straight line fit to that side (for example, \( \alpha_i = 0 \) should be obtained if \( O_{i+1} = 0 \), regardless of the value of change in orientation at the other side. Secondly, if the change in orientation at both sides of \( p_i \) is zero, the values of \( w_{\alpha_i} \) and \( w_{\beta_{i-1}} \) should not become undefined. In this case, according to (6), \( w_{\alpha_i} = w_{\beta_{i-1}} = \tan(\pi/2)/2\tan(\pi/2) = 1/2 \). Hence, the discontinuity case in Akima’s formulation is overcome. However, this is case of an edge, where one would expect straight lines to be fit on both sides of \( p_i \). With \( w_{\alpha_i} = w_{\beta_{i-1}} = 1/2 \), it may seem, the above expectation could not be met with this formulation; however, this will be taken care by the elliptic arc curve fit formulation (defined in section III). It should be noted that, the result of this formulation becomes undefined, when change in orientation on both sides of \( p_i \) are 180° (i.e., \( O_{i-1} = O_{i+1} = \pi \)); however, this case is nearly impossible for any set of data points.

If the data set of \( n \) points correspond to an open curve, then, this formulation additionally requires \( p_{n-2}, p_{n-1}, p_{n+1} \) and \( p_{n+2} \) for evaluating the curve slopes at the end point and the next point on both ends of the curve. These extra points can be approximated by quadratic extrapolation as proposed by Akima [2] a

\[
\begin{bmatrix}
    p_{n-2} & p_{n-1} & p_n \\
    p_{n+1} & p_{n+2} & p_{n+3}
\end{bmatrix} =
\begin{bmatrix}
    1 & 3 & 3 \\
    -3 & -8 & 6
\end{bmatrix}
\]

where, any \( p_i = [x_i \ y_i]^T \).

III. ELLIPTIC ARC CURVE FIT IN A SEGMENT WHEN \( \alpha \) AND \( \beta \) HAVE SAME SIGN

In this section, we define how an elliptic arc is fit between two control points \( p_i \) and \( p_{i+1} \), satisfying the angles \( \alpha_i \) and \( \beta_i \) at its ends, if \( \alpha_i \) and \( \beta_i \) have the same sign (Zero can be considered positive or negative; here, if any of the angles is zero, it will assume the sign of the other). From different ellipses of different sizes and eccentricities, it is possible to get an arc which satisfies the same set of \( \alpha \) and \( \beta \). However, a common methodology is required to ensure the continuity and programmability of the curve fit. A rational Bézier curve of degree two — which are capable of producing arcs of conic section — can be used for this purpose. However, it cannot fit elliptic arc in a segment if \( \alpha_i \) and/or \( \beta_i \) are obtuse. After trying many ways, we finally found one concept which fits in all the possible combinations of \( \alpha \) and \( \beta \). As shown in Fig. 2a, an ellipse is chosen with an eccentricity satisfying the tension parameter (will be defined shortly in (8)) required. Arc segments of the ellipse formed by chords which are parallel to the X-axis are only chosen and skewed to obtain the required \( \alpha \) and \( \beta \) (see Fig. 2).

The elliptic arc curve fit formulation is presented in the sequential order of evaluation as follows:

Tension parameter \( T \) is defined by

\[
\frac{a_i}{b_i} = \sqrt{T_i}
\]

where, \( a \) and \( b \) are the semi-major axis and semi-minor axis respectively of the base ellipse. If \( T = 1 \), it is understood that the base ellipse is a circle. Mostly, the elliptic arcs formed by skewing a base circle (i.e., with \( T = 1 \)), produces good results. However, according to the requirements of the user, \( T \) can be varied to take any value between 0 to \( \infty \). As the value of \( T \) increases from 1, the curve becomes tighter; and as the value decreases from 1, the curve becomes looser. Once the base ellipse is fixed, the skew angle can be obtained by

\[
\varphi_i = \tan^{-1} \left( \frac{\cot |\beta_i| - \cot |\alpha_i|}{2} \right)
\]
The parallel chord above which we get the required arc is defined by the angle \( \theta \) as shown in Fig. 2a and is obtained as

\[
\theta_i = \begin{cases} 
\tan^{-1} \left( \frac{\sin \varphi_i (\tan |\alpha_i| + \tan |\beta_i|)}{\sqrt{T_i}} \right) & \text{if } \alpha_i \neq \beta_i \\
\tan^{-1} \left( \frac{1}{\sqrt{T_i} \tan |\alpha_i|} \right) & \text{if } \alpha_i = \beta_i 
\end{cases}
\]

Then the dimensions of the base ellipse are obtained as

\[
a_i = \frac{l_i}{2 \cos \theta_i} \\
b_i = \frac{a_i}{\sqrt{T_i}}
\]

(10)

where, \( l_i = |l_i| \). Then the elliptic arc in a local coordinate system with \( X_o \) axis along \( l_i \) with the origin at \( p_i \), is defined as

\[
x_{e_i} = a_i \left[ \cos ((1-t)(\pi - \theta_i) + t(\theta_i)) + \cos \theta_i \right] \\
y_{e_i} = \text{sgn}(\alpha_i) b_i \cos \varphi_i \left[ \sin ((1-t)(\pi - \theta_i) + t(\theta_i)) - \sin \theta_i \right]
\]

(12)

where, \( t \) is a local variable which starts from 0 (at initial point of \( l_i \)) and ends at 1 (at terminal point of \( l_i \)).

There is a special case where this formulation would not work. As many would guess, it is the case when \( \alpha_i = \beta_i = 0 \). It is the case which would require a straight line to be fit in that segment. This can be solved either by assigning a very small value for \( \alpha_i \) and \( \beta_i \) like \( \alpha_i = \beta_i = 10^{-10} \) (which worked fine in our MATLAB program), or simply defining a straight-line fit (i.e. \( y_{e_i} = 0 \)), when this special case arises.

FIGURE 2. Skewing a base ellipse to obtain the desired elliptic arc satisfying \( \alpha_i \) and \( \beta_i \) at its ends. (a) base ellipse, (b) skewed ellipse resulted by skewing the base ellipse to an angle \( \varphi \).

IV. ELLIPTIC ARC CURVE FIT IN A SEGMENT WHEN \( \alpha \) AND \( \beta \) HAVE DIFFERENT SIGNS

If \( \alpha \) and \( \beta \) in a segment between two data points have opposite signs (i.e., one angle is above \( 1 \) and another is below \( 1 \)), the curve in this segment must have an inflection somewhere between the two data points; which is not possible with a single elliptic arc. For solving this, the curve is split into two, by an intermediate control point \( p_{im} \) between \( p_i \) and \( p_{i+1} \); which would act as the inflection point. \( p_{im} \) is placed such a way that \( p_{im} - p_i = l_i \) and \( p_{i+1} - p_{im} = l_i \) (see Fig. 3); where, \( l_{i1} = l_i \hat{l}_1 \) and \( l_{i2} = l_i \hat{l}_2 \); and

\[
l_{i1} = l_i \frac{\sin |\alpha_i|}{\sin |\alpha_i| + \sin |\beta_i|} \\
l_{i2} = l_i \frac{\sin |\beta_i|}{\sin |\alpha_i| + \sin |\beta_i|} = l_i - l_{i1}
\]

(13)

Now, the curve angles on both sides of \( p_{im} \) are designated, as in Fig. 3; and are defined as

\[
\alpha_{i1} = \alpha_i, \\
\beta_{i1} = \text{sgn}(\alpha_i) \sin^2 \left( \tan^{-1} \left( \frac{1}{\sqrt{T_i}} \right) \right) \sin^{-1} \left( \frac{l_{i2}}{l_i} \right) \sin |\alpha_i| + \frac{l_{i1}}{l_i} \sin |\beta_i|, \\
\alpha_{i2} = -\beta_{i1} \\
\beta_{i2} = \text{sgn}(\alpha_i) \sin^2 \left( \tan^{-1} \left( \frac{1}{\sqrt{T_i}} \right) \right) \sin^{-1} \left( \frac{l_{i2}}{l_i} \right) \sin |\alpha_i| - \frac{l_{i1}}{l_i} \sin |\beta_i|
\]

FIGURE 3. Introducing new control point \( p_{im} \) in a segment between \( p_i \) and \( p_{i+1} \) for the case if \( \alpha_i \) and \( \beta_i \) have different signs.

VOLUME 8, 2020 227175
\[ \beta_2 = \beta_i. \quad (14) \]

Once the parameters defined in (13) and (14) are found, the segment between \( p_i \) and \( p_{i+1} \) will be considered two segments (\( P_i \) to \( p_{im} \) and \( p_{im} \) to \( P_{i+1} \)); and the curve in these segments can be evaluated using the same procedure explained in section III. At this point, it should be noted that, every such case of opposite angles, will increase the number of control points (initially equal to number of data points ‘\( n \)’) and the number of segments connecting them (initially equal to \( n-1 \)), by one.

The equations in (13) and (14) were formulated based on the following requirements: Firstly, this should maintain continuity with the formulation of section III. This is enabled by positioning the intermediate control point \( p_{im} \) nearer to the smaller angle. It should be noted, according to (13), that the value of an end angle (\( |\alpha_i| \) or \( |\beta_i| \)) which gets the maximum weight is \( \pi/2 \); and the weight reduces as the angle moves away (more acute or obtuse) from it. Additionally, for maintaining continuity, the intermediate angles \( \beta_i \) and \( \alpha_i \) should have their maximum magnitude if \( p_{im} \) is in the mid position; and should reduce in value as \( p_{im} \) moves away from the mid position towards any of the sides; and should approach zero as \( p_{im} \) approaches \( p_i \) or \( p_{i+1} \). Secondly, the curve should get flatter, as the required tension increases; for this, the intermediate angles should reduce in magnitude as the tension increases.

V. G\(^1\)-CONTINUOUS CURVE FIT (G\(^1\) BEA SPLINE)

This section explains how the elliptic arcs fit to individual segments are assembled together to generate a G\(^1\)-continuous curve called G\(^1\) BEA, which passes through the given set of data points. The position and orientation of individual segments are known through their respective \( p_i \) and \( l_i \). The curve relative to the individual segments is evaluated through \( x_i \) and \( y_i \), as explained in section III. Now, through a simple vector addition operation, the desired curve in the global coordinates can be evaluated. Accordingly, the curve, between any two control points \( p_i \) and \( p_{i+1} \), in the global coordinates is given by

\[
\begin{align*}
    x &= x_i + \hat{l}_i x_{e_i} - \hat{l}_i y_{e_i} \\
    y &= y_i + \hat{l}_i x_{e_i} + \hat{l}_i y_{e_i} \\
\end{align*}
\]

(15)

Every \( \alpha_i \) and the corresponding \( \beta_{i-1} \) are formulated in such a way that the elliptic arcs on both sides of \( p_i \) meet tangentially at \( p_i \) (see (5) and (14)). Hence, evaluating the curves at every segment using (15) would result in a G\(^1\)-continuous smooth curve — G\(^1\) BEA — which passes through all the given data points.

VI. G\(^2\)-CONTINUOUS CURVE FIT (G\(^2\) BEA SPLINE)

The curve generated in section V is slope-continuous at all control points; however, cannot guarantee curvature continuity. Hence, a curvature blending is required to make the curve G\(^2\)-continuous. This is done through a blending circular arc as illustrated in Fig. 4. The blending circular arc at the initial point \( p_i \) of a segment is defined in the local coordinate system of the segment as

\[
\begin{align*}
    x_{c_i} &= \frac{1}{\kappa_{c_i}} \left[ \cos \left( (2t - 1)\theta_{c_i} + 2(1 - t)\theta_{c_{im}} \right) - \cos \theta_{c_i} \right] + l_i \\
\end{align*}
\]

(17)
FIGURE 5. Comparison of curvature distributions of $G^1$ and $G^2$ BEAs generating the symbol infinity with four data points. (a) $G^1$ BEA. (b) $G^2$ BEA. The relative magnitudes of curvature distribution is shown along the curve (using [18]). Blending of curvature across the data points and at inflections can be seen in (b) and not in (a).

FIGURE 6. $G^1$-continuous blended elliptic arcs applied to generate the alphabets C, S, U and Z through carefully chosen data points.

\[ y_{c\beta_i} = \frac{1}{\kappa_{c\beta_i}} \left[ \sin \left( (2t - 1)\theta_{c\beta_i} + 2(1 - t)\theta_{c\beta_{mi}} \right) - \sin \theta_{c\beta_i} \right] \]  

(21)

FIGURE 7. BEA ($G^1$ or $G^2$) applied to draw a circle and an ellipse. (a) Circle drawn with four equally spaced data points. (b) Ellipse drawn with the end points of the major and minor axes taken as data points. The relative magnitude of curvature distribution is shown along the curve (using [18]).

\[ \theta_{c\beta_i} = \tan^{-1} \left( \frac{1}{\tan |\beta_i|} \right) \]
\[ \theta_{c\beta_{mi}} = \sin^{-1}(\kappa_{c\beta_i} y_{m_i} + \sin \theta_{c\beta_i}) \]  

(22)

where,
\[
\kappa_{c\beta_{i+1}} = \begin{cases} \kappa_{\alpha_{i+1}} & \text{if, } \text{sgn}(\alpha_{i+1}) = \text{sgn}(\beta_i) \\ 0 & \text{if, } \text{sgn}(\alpha_{i+1}) \neq \text{sgn}(\beta_i) \end{cases} 
\]  

(23)

and, $\kappa_{\alpha_{i+1}}$ is the curvature of the blending circular arc at the terminal point and is given by,
\[
\kappa_{\alpha_{i+1}} = \frac{b_{i+1}}{a_{i+1}} \cos \varphi_{i+1} \]  

(24)

This blending circular arc, defined by (21), starts tangentially to the base elliptic arc at its terminal point and extends till it reaches a point where its displacement from $l_{i+1}$ equals $y_{m_{i}}$ (see Fig. 4).
Now the $G^2$-continuous blended arc is defined in the local coordinate system of the segment as

$$
x_b = \begin{cases} 
  g_\alpha x_e + (1-g_\alpha)x_{cai} & \text{for } 0 \leq t \leq 0.5, \text{if } \kappa_{\beta_{i+1}} < \kappa_{\alpha_i} \\
  g_\beta x_e + (1-g_\beta)x_{c_{\beta_i}} & \text{for } 0 \leq t \leq 0.5, \text{if } \kappa_{\beta_{i+1}} \geq \kappa_{\alpha_i} \\
  x_e & \text{for } 0.5 < t \leq 1, \text{if } \kappa_{\alpha_{i+1}} < \kappa_{\beta_i} \\
  y_e & \text{for } 0.5 < t \leq 1, \text{if } \kappa_{\alpha_{i+1}} \geq \kappa_{\beta_i} 
\end{cases}
$$

$$
y_b = \begin{cases} 
  g_\alpha y_e + (1-g_\alpha)y_{cai} & \text{for } 0 \leq t \leq 0.5, \text{if } \kappa_{\beta_{i+1}} < \kappa_{\alpha_i} \\
  g_\beta y_e + (1-g_\beta)y_{c_{\beta_i}} & \text{for } 0 \leq t \leq 0.5, \text{if } \kappa_{\beta_{i+1}} \geq \kappa_{\alpha_i} \\
  y_e & \text{for } 0.5 < t \leq 1, \text{if } \kappa_{\alpha_{i+1}} < \kappa_{\beta_i} \\
  y_e & \text{for } 0.5 < t \leq 1, \text{if } \kappa_{\alpha_{i+1}} \geq \kappa_{\beta_i} 
\end{cases}
$$

where,

$$
g_\alpha = -16\tau^3 + 12\tau^2 
$$

$$
g_\beta = -16(1-\tau)^3 + 12(1-\tau)^2 
$$

Since the base curve and the blending curve are tangential at the control point, third order polynomial blending functions $g_\alpha$ and $g_\beta$ are enough for a curvature continuous blending (see Fig. 5). Usually, a fifth order polynomial blending function is required for a curvature continuous blending, if the two curves are not tangential at the control point.

After obtaining the blended curve in the local coordinate through (25), it can be transferred to the global coordinate as (similar to that in (15)),

$$
x = x_i + \hat{I}_{xi}x_{bi} - \hat{I}_{yi}y_{bi} 
$$

This formulation enables curvature continuity across any control point; by making the curvature across it equal to the lowest curvature among those on its two sides. The curvature blending using blending circular arc is performed only on the side with higher curvature (see (25)) to achieve a visually pleasing smooth $G^2$-continuity. The curvature blending, if performed on the other side with lower curvature, will not produce a visually pleasing curve (this fact can be examined and experienced by the readers of their own interest).

VII. KEY FEATURES OF BEA

BEA curve is formulated to be capable for using in multiple applications. This is a curve which will pass through all the given data points and does not require any additional input than the data points. The property of the curve at any control point is affected by its neighboring points (two on each side). Accordingly, it can make a smooth curve of any curvature and even a straight line. The different features are illustrated in Fig. 6. If the curve finds three consecutive points to be collinear, then it fits a straight line between these three points. In other words, if BEA is expected to draw a straight line, then three collinear points should be given as data points. Transition from straight line to a curve is smooth (see letter U in Fig. 6). An inflection is produced at curve between two consecutive points ($p_i$ and $p_{i+1}$) only if the changes in orientation ($O_i$ and $O_{i+1}$) at those points have opposite sign (see letter S in Fig. 6). Sharp corners can be produced (unlike some...
curves which avoid sharp corners [9], [14], [15]). Unlike polynomial curves, BEA is capable of perfectly producing circular arcs and hence a circle (see Fig. 7a). Moreover, a close fit to ellipse can be produced with only four data points (the end points of the major and minor axes) as shown in Fig. 7b. $G^1$ and $G^2$ BEAs differ very little in their form (see Fig. 5 and Fig. 9). In many cases the difference cannot be visually identified and would require a curvature distribution plot to find it (see Fig. 5). Hence, $G^1$ BEA can be used in applications like vector graphics and multi-valued function approximation, where $G^2$-continuity is not a requirement. Moreover, $G^1$ BEA has a considerable advantage over $G^2$ BEA in terms of the computation time (see Table 1).

VIII. RESULTS AND DISCUSSION

A. APPLICATION OF BEA IN MULTIPLE-VALUED FUNCTION APPROXIMATION

BEA cannot be used for single-valued function approximation; since it is formulated for a smooth curve fit in a plane and not for a single valued nature. However, for multiple-valued functions, it can produce very good functional approximation. As an example, a curve fit for a circle involute is shown in Fig. 8. Since circle involute is a curve of continually decreasing curvature, $G^2$ BEA is used (however, $G^1$ BEA will produce a similar result). BEA produces an effectively close curve fit to the actual curve; except at the first and last two segments, where a slight deviation is noticed. For making sure that the deviation happens only at free ends, in Fig. 8a, the involute is drawn till $t = 3\pi$ and in Fig. 8b the same involute is extended till $t = 4\pi$. In both cases, the deviation happens only near the free ends. This deviation near the open ends happens because of the unavailability of actual data after the free ends and a polynomial extrapolation approximation (through (7)) is used to define the curve near the free ends (refer section II). However, this limitation is possessed by most other curve fits also.

B. APPLICATION IN GEOMETRIC MODELING

Bézier curves, for their interactivity and versatility, are extensively used in vector graphics including typography. BEA also is versatile by possessing the ability to create straight lines, sharp corners and smooth curves. Hence, it was checked for its ability in typography (see Fig. 9 and Fig. 10). Firstly, aiming to draw the letters ‘A’ and ‘a’, data points are chosen considering its properties (refer section VII). Then, $G^1$ and $G^2$ BEAs were called in for the curve fit. After some fine adjustments, they produced a good fit resembling the letters. Next, three other curve fits (Biarc spline, Hobby spline and Cubic spline) were tested with the same set of points. They did not produce the intended result. The commonality of these three splines is: they failed to produce straight lines and sharp corners. This may have happened because of insufficient data for those splines. However, this clearly indicates that, BEA is capable of giving good curve fit with lesser input data.

In addition to the ability to model the geometry, the curve fit used in vector graphics should allow some transformation features. Fig. 10 shows the various transformation capabilities of BEA. Initially the data points used in Fig. 9 were taken as base points. Then, these base points were multiplied with the corresponding transformation matrix to get the data points of the required transformation feature. Then, BEA was applied to the new transformed set of data points and

---

1Hobby spline has the option for inputting individual slopes and tension of curve at the data points. However, here, only its basic spline configuration is used.
checked whether it produces the required transformed image of the original image. It is evident from Fig. 10 that BEA is successful in creating the transformed image from the transformed data points. Moreover, it maintains invariance of shape under translation, scaling and rotation; which is the necessary property of a curve used in geometric modeling. Additionally, mirroring and skewing are performed with no difficulty.

The next example shown in Fig. 11 is modeling of NACA-8412 airfoil; for demonstrating the abilities to model aerodynamic shapes. The basic requirements of a low-speed airfoil like NACA-8412 are having a well-rounded leading edge and a sharp trailing edge. It is a well-known fact that elliptic arcs can produce well rounded curve and smooth curvature change than polynomials. This can be noticed in Fig. 11, as BEA produced better leading-edge shape than other options. Moreover, it produced sharp trailing edge with ease as compared to biarc spline and Hobby spline (which have the nature of avoiding sharp corner). This discussion is meant only to demonstrate the ability of BEA

FIGURE 10. Various vector graphics transformation abilities of BEA. Both $G^1$ and $G^2$ BEAs showed similar output. Letters in dashed lines represent the base image.
A. Kumaresan, G. S. L. Ramachandran: Blended Elliptic Arc Splines

FIGURE 11. Comparison of different curve fitting methods applied for generating NACA-8412 Airfoil from 20 points on the surface. These airfoil coordinate points — in cosine spacing from leading edge to trailing edge — were obtained from [20].

to produce the required curve fit with lesser input data; and not to falsely declare that the other methods are incapable of producing these shapes. However, they require more data or different kind of data other than the data points like tension and slope data. For example, Bézier curves are capable of reproducing the image in Fig. 9 with lesser number of data points; however, require proper curve handles defined at the data points. Hence, the claim of this paper is, BEA is a simpler yet effective solution for a decent curve fit.

Another important requirement for a curve fitting method used in geometric modeling is faster computation of the curve. For this purpose, the time — to process the curve fit corresponding to Fig. 9 and Fig. 10— taken by Biarc spline, Hobby spline, Cubic spline, $G^1$ BEA and $G^2$ BEA were compared and shown in Table 1. The testing was performed using MATLAB codes run on a Windows 10 personal computer with AMD Ryzen 5 2400G processor with base frequency of 3.60 GHz and 16 GB RAM. The computation time was measured with the internal timer using tic toc function. Every method was programmed to generate 101 curve points in a segment between any two consecutive data points. The experiment for every method consisted of three trials. For every trial, the code for each method was run, such that it computes the curve data repeatedly for 5000 times using a for loop; and the computation time was noted. Hence, at the end of an experiment, the computation time for 15000 times computing the same curve fit was obtained. Then, the average times for one-time computation were calculated and tabulated (Table 1). The first experiment was conducted for curve fit corresponding to Fig. 9. Then the test of curve fit for NACA-8412 Airfoil was conducted as three experiments with 20, 80 and 200 coordinate points obtained from [20] as data points.

From Table 1 it can be observed that, $G^1$ BEA is the fastest among the 5 methods. $G^2$ BEA takes less time than Hobby spline; however, takes more time than the remaining three methods. Moreover, the additional computation time, taken by $G^2$ BEA, increases as the number of data points increases. This can be explained by comparing the computation times of $G^1$ and $G^2$ BEAs. The additional time taken by $G^2$ BEA is because of the curvature blending process. Hence, since $G^1$ and $G^2$ BEAs produce visually similar results, $G^1$ BEA is recommended in all applications which does not strictly require $G^2$-continuity; for example, in functional approximations, vector graphics etc. In $G^2$-continuity demanding applications like aerodynamic shape design and motion path tracing, $G^2$ BEA curve can be used. Whereas, the additional time cost is acceptable considering the quality of the curve fit obtained.

### C. APPLICATION IN MOTION PATH TRACING

In motion capture technology, when an event is captured using camera, particularly when the frame rate is limited, curve fitting technology can be used for the motion path tracing from the available captured data. As an example, wing kinematics data of a pigeon presented in [21] was taken for analysis. The pigeon flapped its wings at around 6 Hz; and the motion of the wing was captured by a camera operated at 60 fields per second. The locations of the wing tip and the wrist at every captured frame of the camera were produced as dots and were shown connected using straight lines. However, this data is insufficient for simulating the same wing
FIGURE 12. Comparison of different curve fitting methods applied for a pigeon wing’s motion path tracing. The path data points were taken from [21]. The trajectory curve fits present are only for qualitative comparison. The relative sizes of different trajectories are not in scale. Since motion paths require at least $G^2$-continuity, the Biarc spline and $G^1$ BEA methods were not considered for this comparison.
motion artificially for kinematic or aerodynamic analyses. This difficulty can be overcome with a closed curve fit which would smoothly connect the available data points to produce an approximate trajectory of the wing motion; then it can be used for the aforesaid analyses.

Wing motion captured data points at 8 m/s and 16 m/s flight velocities were taken for analysis. G2 BEA, along with cubic spline and Hobby spline were applied for approximating the wing trajectory and the results are shown in Fig. 12. The following features of BEA can be observed from Fig. 12: It avoids unwanted undulations; which Hobby spline produced in some cases. It produces well rounded smooth turns as opposed to Cubic spline which produced sharp turns in some cases. It should be admitted that Cubic spline produced smoother curve than BEA, in some cases. Overall, it can be said that, BEA is a good option as a curve fitting method for motion path tracing applications; especially when well-rounded turns and minimal undulations are required.

D. CURVE CONTROL WITH TENSION PARAMETER 'T'

The tension parameter T of this formulation was defined in (8); and all other parameters were defined considering it. However, all the results presented in the previous sections were generated with BEA considering T = 1. Although it produces good curve fit in most cases, in some cases, the curve can be further fine-tuned by changing the value of T, for obtaining better results. BEA has an option for applying different values of T for every individual segment between data points; without losing the required geometric continuity (G1 or G2). Figure 13 illustrates a case where curve control using T is required. It corresponds to wing wrist kinematic data of pigeon’s flight at 20 m/s taken from [21]. In Fig. 13a, G2 BEA is fit to the data with T = 1 in all segments. The curve is smooth. However, in the segment between points 9 and 1, the curve overshoot produced is to an unnecessary value. This is an undesirable property of BEA to produce higher overshoots when αi and βi have higher magnitudes. This can be controlled either by carefully specifying the values of T at individual sections, or by defining an automatic tension predicting formula as given below.

\[ T_i = \frac{\sqrt{\tan \left( \frac{\alpha_i}{2} \right)} + \sqrt{\tan \left( \frac{\beta_i}{2} \right)}}{2 \times \sqrt{\tan \left( \frac{\beta_i}{8} \right)}} \]  

(28)

This formula tightens the curve as the magnitudes of αi and βi increases; besides preserving the ability of BEA to produce perfect circle as illustrated in Fig. 7. In Fig. 13b, it can be noticed that, after application of the auto-tension formula (28), BEA produced a reasonably good curve fit.

E. LIMITATIONS OF BEA

Besides having superior smooth curve fitting and shape preserving abilities, BEA splines do have some limitations. BEA splines allow shape modification to curve corresponding to a given set of data points, however, not as much as a Bézier curve would allow. Bézier curves enable shape modification using curve handles, which is not possible in BEA splines. This fact makes BEA curves no better competitor for Bézier curves in field of vector graphics; although it is able to produce the same shape with fewer data than that required by Bézier curves, since it requires careful manipulation of data points.

G1 BEA is faster in computation time than other methods (see Table 1). However, for cases where G2 continuity is mandatory, G2 BEA is not the fastest. Its performance in computational time is faster than a G2 continuous Bézier curve, however, slower than cubic spline. This is because of the G2 blending process which increase the computation time, especially when data points are more in number.

Conic sections are undoubtedly natural in shape. However, each conic section is best suitable for different applications. Ellipses are meant for returning loops like orbital motion of satellites. Hence, elliptic arcs are best applicable for closed paths and internal sections of open curves; however, not for open ends. This fact is apparent in Fig. 8. In nature, open ends mean an escape path like escape trajectory of a space object, which are generally hyperbolic. Elliptic arcs with some modifications can be made represent straight line, circular and parabolic arcs; however, cannot represent a hyperbolic arc. Compared to elliptic arcs, hyperbolic arcs are better suitable in certain places in motion path applications like free ends of open curves representing escape path from a constrained curve, transition from curved path to straight line.

By principle, any G2 curve can enable a C2 movement through it. However, reparameterization of G2 BEA requires more effort since it involves elliptic functions; which do not have a direct integral solution, where elliptical integrals or numerical methods should be applied. Hence, for trajectory planning applications, it is not advisable to apply BEA curves.
directly. Instead, an initial fit by BEA for generating more data points, followed by an overlapping polynomial fit would enable achieving a parametric continuous curve with the smooth fit abilities of BEA.

IX. CONCLUSION

BEA proves to be a good option for interpolation and smooth curve fit in a 2D plane. It provides a simple solution for curve fit in multiple-valued functional approximation, geometric modeling and motion path tracing applications. This method can be applied for a set of not less than three data points in a 2D plane. It is capable of identifying straight line and curved paths from the arrangement of consecutive data points alone. The curve passes through all the given data points and in the order of the given data; it does not do any modifications to the given data. The curve solution is straight forward and does not require iterations to solve for the curve data. This method is applicable for both open and closed curves. BEA open curves applied to functional approximation provides a reasonably good interpolation; however, may produce a slight deviation near the end points depending on the actual nature of the function involved.

$G^1$ and $G^2$ BEAs give provision for modifying the tension of the curve at every individual segment, yet will preserve the $G^1$ and $G^2$ continuities respectively. BEA has the property of increasing overshoot, as the orientations of the curve tangents, at the control points, increase from the base line segment. However, this can be controlled by proper manipulation of tensions at individual segments; either manually or defining an auto-tension evaluation formula as given in (28).

Even though the method is defined with the tangent determination procedure given in section II, the curve definitions explained in section III to VI can be used with any method of tangent evaluation as per the requirement of the user. Even then the method will preserve the qualities of minimal inflections, smooth transition of curvature. This method can be extended for producing 3D curves; however, requires further formulations; which are left for future work.

REFERENCES

[1] H. Akima, “A new method of interpolation and smooth curve fitting based on local procedures,” J. ACM, vol. 17, no. 4, pp. 589–602, Oct. 1970.

[2] H. Akima, “A method of smooth curve fitting,” U.S. Government Printing Office, ESSA, Washington, DC, USA, Tech. Rep. ERL 101-ITS, 1969, vol. 73.

[3] F. N. Fritsch and R. E. Carlson, “Monotone piecewise cubic interpolation,” SIAM J. Numer. Anal., vol. 17, no. 2, pp. 238–246, Apr. 1980.

[4] M. Stenberg, “A simple method for monotonic interpolation in one dimension,” Astron. Astrophys., vol. 239, pp. 443–450, Nov. 1990.

[5] R. L. Levién, “From spiral to spline: Optimal techniques in interactive curve design,” Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep., 2009.

[6] C. H. Séquin, K. Lee, and J. Yen, “Fair, G2-and C2-continuous circle splines for the interpolation of sparse data points,” Comput. Aided Des., vol. 37, no. 2, pp. 201–211, 2005.

[7] D. Salomon, Curves and Surfaces for Computer Graphics. New York, NY, USA: Springer, 2006.

[8] A. R. Forrest, “Interactive interpolation and approximation by Bézier polynomials,” Comput. Des., vol. 22, no. 9, pp. 527–537, Nov. 1990.

[9] J. D. Hobby, “Smooth, easy to compute interpolating splines,” Discrete Comput. Geometry, vol. 1, no. 2, pp. 123–140, Jun. 1986.

[10] I. J. Schoenberg, “Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulas,” Quart. Appl. Math., vol. 4, no. 2, pp. 112–141, 1946.

[11] C. de Boor, “On calculating with B-splines,” J. Approximation Theory, vol. 6, no. 1, pp. 50–62, 1972.

[12] B. K. P. Horn, “The curve of least energy,” ACM Trans. Math. Softw., vol. 9, no. 4, pp. 441–460, Dec. 1983.

[13] H. P. Moreton and C. H. Sequin, “Functional optimization for fair surface design,” Comput. Graph., vol. 26, no. 2, pp. 167–176, 1992.

[14] K. M. Bolton, “Bicubic curves,” Comput. Des., vol. 7, no. 2, pp. 89–92, Apr. 1975.

[15] D. S. Meek and D. J. Walton, “Approximation of discrete data by G1 arc splines,” Comput. Des., vol. 24, no. 6, pp. 301–306, 1992.

[16] H. J. Wenz, “Interpolation of curve data by blended generalized circles,” Comput. Aided Geom. Des., vol. 13, no. 8, pp. 673–680, 1996.

[17] C. Ionita. (2019). Makima Piecewise Cubic Interpolation? Cleve’s Corner: Cleve Moler on Mathematics and Computing MATLAB & Simulink. Accessed: Mar. 10, 2020. [Online]. Available: https://blogs.mathworks.com/cleve/2019/04/29/makima-piecewise-cubic-interpolation/

[18] A. Mjaaavatten. (2020). Curvature of a 2D or 3D Curve. MATLAB Central File Exchange. Accessed: May 8, 2020. [Online]. Available: https://in.mathworks.com/matlabcentral/fileexchange/69452-curvature-of-a-2d-or-3d-curve

[19] W. Robertson. (2020). Smooth 3D Bezier Curves With Implicit Control Points. MATLAB Central File Exchange. Accessed: May 11, 2020. [Online]. Available: https://in.mathworks.com/matlabcentral/fileexchange/42302-smooth-3d-bezier-curves-with-implicit-control-points

[20] (2020). NACA 4 Digit Airfoil Generator. Airfoil Tools. Accessed: May 11, 2020. [Online]. Available: http://airfoiltools.com/airfoil/naca4digit

[21] B. Tobalske and K. Dial, “Flight kinematics of black-billed magpies and pigeons over a wide range of speeds,” J. Exp. Biol., vol. 199, no. 2, pp. 263–280, 1996.

ALLWYN KUMARESAN was born in Nagercoil, India, in 1988. He received the B.E. degree in aeronautical engineering and the M.E. degree in aerospace technology from Anna University, Chennai, India, in 2009 and 2012, respectively. Since 2013, he has been an Assistant Professor with the Department of Aerospace Engineering, SRM Institute of Science and Technology, India. His research interests include flapping wing theoretical analyses and biomechanics of natural flyers. He is also a member of The Institution of Engineers (India), and an Associate Member of the Aeronautical Society of India.

GANAPATHY SUBRAMANIAP. L. RAMACHANDRAN was born in Lalgudi, India, in 1966. He received the B.E. degree in mechanical engineering from the Regional Engineering College, Durgapur, India, in 1988, the M.E. degree in aeronautical engineering from the Madras Institute of Technology, Chennai, India, in 1990, and the Ph.D. degree in mechanical engineering from Anna University, Chennai, in 2008. He joined the SRM Institute of Science and Technology, Chennai, as a Lecturer at the Department of Mechanical Engineering, in 1990. Since then, he has been with the SRM Institute of Science and Technology, where he is currently a Professor and the Head of the Department of Aerospace Engineering. He has published many articles in reputed journals and conferences. Under his guidance, he has produced two Ph.D. graduates. At present six scholars are pursuing their Ph.D. under his guidance. His research interests include environmental engineering, aircraft structures, composite materials and structures, and aerodynamics. He is also a Life Member of the Indian Society for Technical Education and a Fellow of The Institution of Engineers (India).