Int J Adv Manuf Technol (2006) 29: 151–158
DOI 10.1007/s00170-004-2484-x

ORIGINAL ARTICLE

S.-H. Frank Chuang · J.-L. Shih

A novel approach for computing C^2-continuous offset of NURBS curves

Received: 23 July 2004 / Accepted: 4 November 2004 / Published online: 25 January 2006

Abstract Computing offset curves is an important geometric operation in areas of CAD/CAM, robotics, cam design and many industrial applications. In this paper, an algorithm for computing offsets of NURBS curves using C^2-continuous B-spline curves is presented. The progenitor curve in database is initially approximated by a line-fitting curve, and then the exact offset of this line-fitting curve is introduced as an initial offset. Based on the initial offset and a set of selected knots, an intended C^2-continuous B-spline curve is subsequently constructed. The method uses a new error-measuring scheme, which is based on the convex hull property of Bézier curves and the idea of cumulative errors, to calculate the global error bound of offset approximation. The method obtains offset curves with C^2 continuity and guarantees that the actual error bound is precisely within the prescribed tolerance. In addition, it also allows one to selectively parametrize the offset curve.

Keywords CAD/CAM · NURBS curves · Offsetting · C^2 continuity · Piecewise Bézier curves

1 Introduction

Offset curves are widely used in many engineering areas such as CAD/CAM, tolerance zone definition, tool path generation for machining, and robot-path planning [3, 10, 11]. Given a parametric curve $\mathbf{C}(t)$, the offset curve $\mathbf{C}^0(t)$ with an offset distance d is defined by

$$\mathbf{C}^o(t) = \mathbf{C}(t) + d\mathbf{N}(t) \tag{1}$$

where $\mathbf{N}(t)$ is the unit normal vector to the base curve computed at t [9]. For planar curves, $\mathbf{N}(t)$ is expressed as

$$\mathbf{N}(t) = \frac{(-y'(t), x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}} \tag{2}$$

Due to the square root term in the denominator of $\mathbf{N}(t)$, the exact offset curves in general are neither polynomial nor rational [7, 8]. It is difficult to obtain simple forms of such offsettings and approximation methods are needed.

Although using approximation technique to compute offset of freeform curves is inevitable, a simple but robust error measurement mechanism is still not completely established in both academic study and industrial applications. Most of the methods available in the literature always attempt to calculate the global error by using finite sampling points along the progenitor curve [4, 12, 13, 15, 19]. Those approaches are generally regarded as unreliable since the number of discrete samples required for computing the true global error bound is not clear. However, in many practical engineering applications of curve offsetting, e.g. automatic NC tool path generation, it is firmly believed that finding an offset curve with controlled precision is more important than saving the size of output data. Additionally, approximating offset curves with at least C^2 continuity are preferred for visual pleasure of design work or certain engineering requirements, such as movement smoothness in robot trajectory or automatic machines.

Surprisingly, to the best of our knowledge, there are only few published methods that can handle the approximation error with global preciseness [5, 14]. Those approaches do not give much concern on the problem of continuity for offset curve segments. Elber and Cohen propose a symbolic method to compute the global error of offset approximation using a difference function with square distances [5]. However, this approach are not suitable for practical applications since extreme point searching for high degree curve is needed, and the computing becomes difficult and unstable when the progenitor curves are rational. Lee et al. suggested another method to measure the approxi-

S.-H.F. Chuang (✉) · J.-L. Shih
Department of Mechanical Engineering,
National Chung-Hsing University,
250 Kuo-kuang Road, Taichung 402, Taiwan
E-mail: fzngf@nchu.edu.tw
Tel: +886-4-2285-1348
Fax: +886-4-2287-7170

mation error with improved precision based on circle approximation [14]. The main disadvantage of their method is that the degrees of generated offset curves are very high and only G^1-continuity can be obtained for the resulting offset curve segments.

Piegl and Tiller, and followed by Ravi Kumar et al., lately presented some algorithms for approximating offsets of NURBS curves and surfaces [17, 18]. The main idea of the method relies on point sampling based on derivatives, interpolation and knot removal. The method is characterized as having fewest control points generated by then, however, the problem of error control with global assurance is still not solved since their approach is also discrete in essence and the obvious problem of error stacking seems to be overlooked.

This paper proposes a cumulative error based method, which is a novel attempt to estimate the maximal deviation of offset by using the convex hull property of Bézier (or rational Bézier) curves and considering the inherent difference existing between the constructed composite Bézier polygons and the corresponding B-spline polygon. In this method the progenitor NURBS curve in database is initially approximated by a line-fitting curve, and then an exactly offset line-fitting curve is introduced with the prescribed normal direction and an offset distance. Finally, based on the offset approximating line segments and a set of selected knots, a C^2-continuous B-spline curve is constructed as a desired offset.

Compared with previous methods, the proposed approach provides a simpler mechanism sure and safe on the approximation error estimation, and easily elevates the continuity of the offset curve to a practically useful level of C^2. The parametrization of the offset curve is also selective. Only linear geometric calculations are involved in this method, this offsetting algorithm is very stable and efficient. Some experimental tests and a practical example of NC tool-path generation for profiling are given in Sect. 5 to demonstrate its effectiveness and usefulness.

2 Initial line approximation and offsetting

In this research, it is assumed that all curves are represented in NURBS form. The details of mathematical description for such curves can be found in literature [16]. Here, for explanatory convenience, the NURBS curve is introduced briefly. A pth-degree NURBS curve is defined by

$$\mathbf{C}(t) = \frac{\sum\limits_{i=0}^{n} N_{i,p}(t)\mathbf{P}_i w_i}{\sum\limits_{i=0}^{n} N_{i,p}(t) w_i} \tag{3}$$

where $\mathbf{P}_i$ are the control points, w_i are the weights, and $N_{i,p}(t)$ are the normalized B-splines defined on the knot vector for non-periodical conditions

$$T = \left\{ \underbrace{t_0 = \cdots = t_0}_{p+1}, t_{p+1}, \cdots, t_n, \underbrace{t_{n+1} = \cdots = t_{n+p+1}}_{p+1} \right\} \tag{4}$$

Using homogeneous coordinates, an equivalent representation is

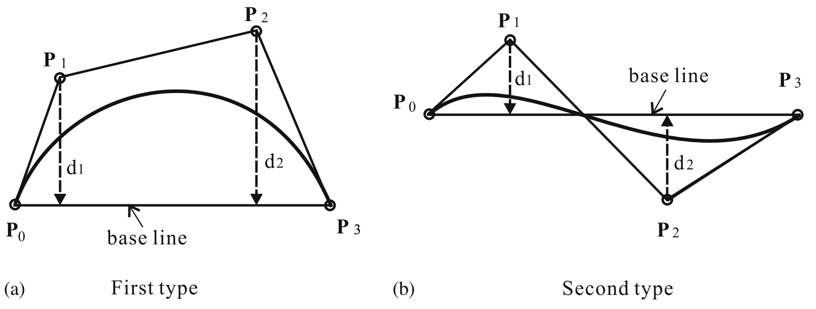
$$\mathbf{C}^w(t) = \sum_{i=0}^{n} N_{i,p}(t)\mathbf{P}_i^w \tag{5}$$

where $\mathbf{P}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$ are the weighted control points and $\mathbf{C}^w(t)$ is a polynomial B-spline curve in homogeneous space.

Before offsetting a NURBS curve, an approximation with straight line fitting is introduced. The knot insertion method is used to decompose a NURBS curve into piecewise Bézier segments [2]. Based on the convex hull property of Bézier curves, it is easy to check if a given curve is near linearity within tolerance [3]. As shown in Fig. 1, for any one of the two representative Bézier curves, it is tested whether all the interior control points, $\mathbf{P}_1$ and $\mathbf{P}_2$, lie within the tolerance range deviating from the straight line joining the first and final end control points, $\mathbf{P}_0$ and $\mathbf{P}_3$. If the required tolerance is not met, the curve is subdivided at the middle of the parameter domain repeatedly using the classical de Casteljau algorithm [6]. For example, a Bézier curve defined by $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ is shown in Fig. 2, one time subdivision generates two sets of control points $\mathbf{P}_0$, $\mathbf{P}_{1,0}$, $\mathbf{P}_{2,0}$, $\mathbf{P}_{3,0}$, and $\mathbf{P}_{3,0}$, $\mathbf{P}_{2,1}$, $\mathbf{P}_{1,2}$, $\mathbf{P}_3$. Therefore, smaller convex hulls $\mathbf{P}_0$, $\mathbf{P}_{1,0}$, $\mathbf{P}_{3,0}$, $\mathbf{P}_{2,0}$, $\mathbf{P}_0$ and $\mathbf{P}_{3,0}$, $\mathbf{P}_{2,1}$, $\mathbf{P}_{1,2}$, $\mathbf{P}_3$, $\mathbf{P}_{3,0}$ then can approximate the original Bézier curve closer than the original convex hull $\mathbf{P}_0$, $\mathbf{P}_1, \mathbf{P}_3, \mathbf{P}_2, \mathbf{P}_0$. Since the subdivision process converges relatively rapidly, these computations are quite efficient.

When a Bézier curve is near linearity within tolerance, it is declared flat and is approximated simply by a straight line segment joining end control points of the curve, which is called

Fig. 1. Testing for near linearity of Bézier curves (upper bound error = Max (d_1, d_2))



(a) First type

(b) Second type

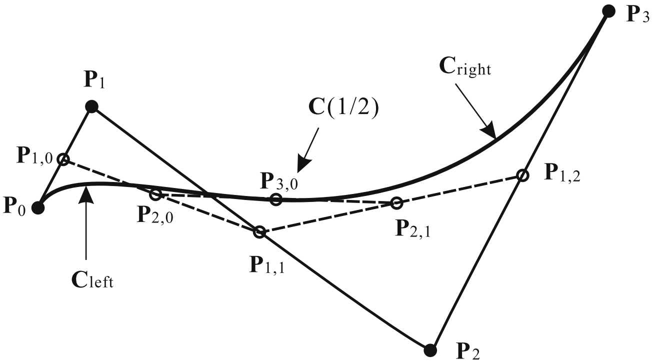


Fig. 2. Subdivision of Bézier curves at $t = 1/2$

a *base line*, or by line segments composing the control polygon. Assuming the decomposed Bézier curves and/or the subdivided Bézier segments all have been flattened within required tolerance, the approximation of a NURBS curve with straight line segments is obtained as follows:

- If the Bézier curve locates at the first or the last position of the list, approximate it with its control polygon;
- Otherwise approximate each Bézier curve with its base line; and
- Collect the approximating straight line segments.

The reason why the NURBS curve are approximated by collecting line segments with different criteria is explained below: (1) using the control polygon to approximate a Bézier curve so that the tangent direction at an endpoint of the curve can be parallel with the first or the final leg of control polygon, i.e., the tangent of the endpoint of the progenitor curve; and (2) if no constraint is imposed, it tends to reduce the output data size using the base line to approximate a Bézier curve for constructing an offset curve.

Once the line fitting approximation of NURBS curves is established, a chain of straight line segments is ready for computing offsets. Hence, depending on the normal direction and offset distance, an exact offset polygon is generated quickly, as shown in Fig. 3. Nevertheless, only C^0 continuity is attained at this stage for the offset curve with straight line fitting, further improvement on the degree of continuity is needed. A method for generating C^2 continuous offset curves with B-spline form is proposed in the following section.

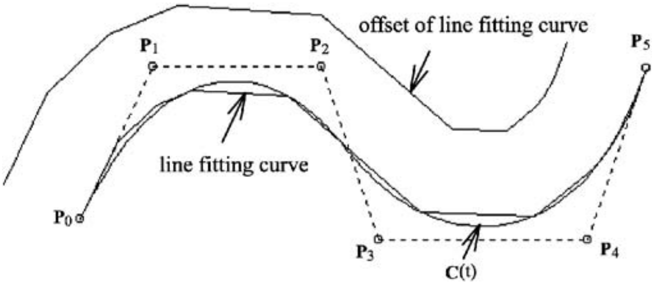


Fig. 3. Offsetting of line fitting curves

3 Constructing C^2 offset curve

The initial offset curve is only C^0 continuous with straight line segment approximation, it may not meet the requirements in some advanced engineering applications. Therefore, the next objective is to reconstruct an offset curve with higher continuity, e.g. C^2 continuity, for more demanding practical purposes. Using the offset line-fitting curve as a cubic B-spline polygon and selecting an adequate knot vector, a C^2 continuous B-spline curve can be formed by piecing its constituent Bézier curve segments together. The main reason for using Bézier pieces as intermediate entities to construct the desired C^2 B-spline offset curve is to facilitate the error estimation of offset approximation. To further clarify the process of constructing C^2 B-spline offset curve, the B-spline representation of piecewise Bézier curves and the C^2-continuity condition are briefly reviewed in Sect. 3.1.

3.1 C^2-continuous piecewise Bézier curve and B-spline representation

A number of Bézier curves can be joined end to end to form a piecewise Bézier curve (or a composite Bézier curve) [9]. Based on the positions of control points for each Bézier curve, any level of continuity, e.g. G^1/C^1 or G^2/C^2, can be achieved. A piecewise cubic Bézier curve which contains two curve segments $\mathbf{C}_0(t)$ and $\mathbf{C}_1(t)$ is shown in Fig. 4. $\mathbf{C}_0(t)$ and $\mathbf{C}_1(t)$ are defined on the intervals $[t_0, t_1]$ and $[t_1, t_2]$ respectively and the common point of the two curve segments is $\mathbf{b}_{0,3}(= \mathbf{b}_{1,0})$. The condition to form a C^2 continuous piecewise Bézier curve for this illustration is explained as follows [6].

Assuming that the two given curve segments $\mathbf{C}_0(t)$ and $\mathbf{C}_1(t)$ have C^1 continuity at parameter t_1, the three Bézier points (control points), $\mathbf{b}_{0,2}$, $\mathbf{b}_{0,3}(= \mathbf{b}_{1,0})$, and $\mathbf{b}_{1,1}$, must be collinear with

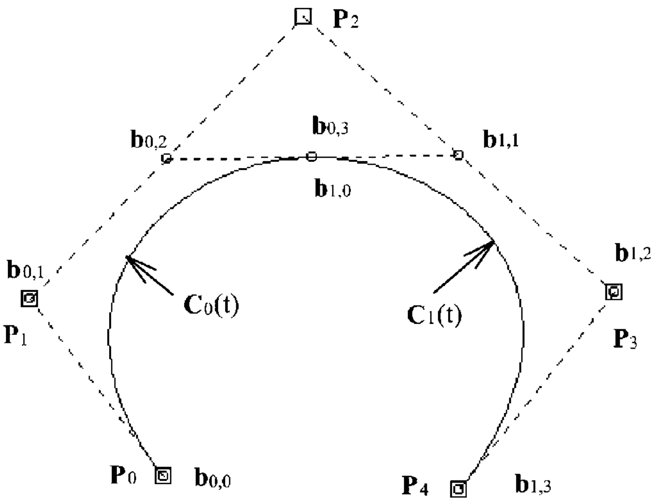


Fig. 4. A C^2-continuous piecewise Bézier curve consisting of two segments

ratio $\Delta_0 : \Delta_1$, that is

$$\mathbf{b}_{0,3} (= \mathbf{b}_{1,0}) = \frac{\Delta_1}{\Delta_0 + \Delta_1}\mathbf{b}_{0,2} + \frac{\Delta_0}{\Delta_0 + \Delta_1}\mathbf{b}_{1.1} \tag{6}$$

where $\Delta_i = t_{i+1} - t_i$, and $i = 0, 1$. Extending to C^2 continuity, an *unique* additional auxiliary point $\mathbf{P}_2$ must exist such that points in each of the two sets of points $\{\mathbf{b}_{0,1},\ \mathbf{b}_{0,2},\ \mathbf{P}_2\}$ and $\{\mathbf{P}_2, \mathbf{b}_{1,1}, \mathbf{b}_{1,2}\}$ are also collinear, respectively, with ratio $\Delta_0 : \Delta_1$. These constraints can be expressed as:

$$\mathbf{b}_{0,2} = \frac{\Delta_1}{\Delta_0 + \Delta_1}\mathbf{b}_{0,1} + \frac{\Delta_0}{\Delta_0 + \Delta_1}\mathbf{P}_2 \tag{7}$$

$$\mathbf{b}_{1,1} = \frac{\Delta_1}{\Delta_0 + \Delta_1}\mathbf{P}_2 + \frac{\Delta_0}{\Delta_0 + \Delta_1}\mathbf{b}_{1,2} \tag{8}$$

It is noteworthy that if a simpler case of uniform parameter spacing is used, i.e. $\Delta_0 = \Delta_1$, then the knot spans in the full parameter domain are all identical. Consequently, the rational coefficients existing at the right hand of Eqs· 6–8 are simplified to $1/2$.

Now suppose a piecewise Bézier curve contains m curve segments $\mathbf{C}_i(t)$, $i = 0, \ldots, m-1$. $\mathbf{C}_i(t)$ is defined on the interval $[t_i, t_{i+1}]$, where $t_0 < t_1 < \ldots < t_{m-1} < t_m$. Then the piecewise Bézier curve also can be represented by an integral B-spline curve (or a NURBS curve with all of its weights set to 1) defined on a knot vector in which all the interior knots have multiplicities equal to the degree p as follows:

$$\mathbf{C}(t) = \sum_{i=0}^{n} N_{i,p}(t)\mathbf{P}_i \tag{9}$$

where $\mathbf{P}_i$ are the control points, and $N_{i,p}(t)$ are the p-th degree B-spline basis functions defined over the knot vector

$$T = \left\{ \underbrace{t_0, \cdots, t_0}_{p+1}, \underbrace{t_1, \cdots, t_1}_{p}, \cdots, \underbrace{t_{m-1}, \cdots, t_{m-1}}_{p}, \underbrace{t_m, \cdots, t_m}_{p+1} \right\} \tag{10}$$

However, if the C^2 condition is also satisfied by the piecewise Bézier curve, the multiplicity of each interior knot can be further reduced to (*degree*-2), and a new set of control points of the B-spline curve are generated subsequently with curve shape unchanged. As shown in Fig. 4, the piecewise cubic Bézier curve can be represented by an integral B-spline curve with control points $\mathbf{b}_{0,0}$, $\mathbf{b}_{0,1}$, $\mathbf{b}_{0,2}$, $\mathbf{b}_{0,3}$ (or $\mathbf{b}_{1,0}$), $\mathbf{b}_{1,1}$, $\mathbf{b}_{1,2}$, and $\mathbf{b}_{1,3}$, and is defined on the knot vector $T = \{t_0, t_0, t_0, t_0, t_1, t_1, t_1, t_2, t_2, t_2, t_2\}$. If C^1 condition is met at parameter t_1, then the control point $\mathbf{b}_{0,3}$ $(= \mathbf{b}_{1,0})$ can be dropped and the multiplicity of knot t_1 is reduced by one; and if C^2 condition is met at parameter t_1, a new set of control points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$ is obtained with an associated new knot vector $T = \{t_0, t_0, t_0, t_0, t_1, t_2, t_2, t_2, t_2\}$.

Applying the idea of C^2-continuity to construct offset curves is just a reverse process as mentioned above. In such case, a B-spline polygon and the knot sequence is given in advance, and the C^2-continuous offset curve is constructed accordingly. The detail of this process is described in Sect. 3.2.

3.2 Producing C^2 B-spline offset curve

After the G^0 offset line-fitting curve is obtained, a desired C^2 B-spline fitting curve is computed as follows. Using the straight-line segments of offset line-fitting curve as the B-spline polygon, i.e. the end points of line segments are treated as B-spline control points, and choosing an appropriate knot vector, a C^2-continuous integral B-spline curve can be constructed by invoking its constituent Bézier segments. For example, if the intended B-spline polygon has $(L+3)$ deBoor points (i.e., control points $\mathbf{P}_i$, $i = -1, 0, \ldots, L+1$) and a set of knot sequence $\{t_0, \ldots, t_L\}$ has been assigned, then the C^2 cubic B-spline offset curve is constructed by using L Bézier curve segments with Bézier points as:

$$\mathbf{b}_{3i-1} = \frac{\Delta_i}{\Delta_i + \Delta_{i-1} + \Delta_{i-2}}\mathbf{P}_{i-1} + \frac{\Delta_{i-1} + \Delta_{i-2}}{\Delta_i + \Delta_{i-1} + \Delta_{i-2}}\mathbf{P}_i, \quad i = 2, \ldots, L-1 \tag{11}$$

$$\mathbf{b}_{3i+1} = \frac{\Delta_i + \Delta_{i+1}}{\Delta_{i+1} + \Delta_i + \Delta_{i-1}}\mathbf{P}_i + \frac{\Delta_{i-1}}{\Delta_{i+1} + \Delta_i + \Delta_{i-1}}\mathbf{P}_{i+1}, \quad i = 1, \ldots, L-2 \tag{12}$$

$$\mathbf{b}_{3i} = \frac{\Delta_i}{\Delta_i + \Delta_{i-1}}\mathbf{b}_{3i-1} + \frac{\Delta_{i-1}}{\Delta_i + \Delta_{i-1}}\mathbf{b}_{3i+1}, \quad i = 1, \ldots, L-1 \tag{13}$$

where Δ_i indicates $(t_{i+1} - t_i)$. Different settings are specified according to different end conditions. For the case of control-end conditions, the control points at ends are set as follows [1]:

$$\begin{cases} \mathbf{b}_0 = \mathbf{P}_{-1} \\ \mathbf{b}_1 = \mathbf{P}_0 \\ \mathbf{b}_2 = \frac{\Delta_1}{\Delta_0+\Delta_1}\mathbf{P}_0 + \frac{\Delta_0}{\Delta_0+\Delta_1}\mathbf{P}_1 \end{cases} \tag{14}$$

$$\begin{cases} \mathbf{b}_{3L} = \mathbf{P}_{L+1} \\ \mathbf{b}_{3L-1} = \mathbf{P}_L \\ \mathbf{b}_{3L-2} = \frac{\Delta_{L-1}}{\Delta_{L-1}+\Delta_{L-2}}\mathbf{P}_{L-1} + \frac{\Delta_{L-2}}{\Delta_{L-1}+\Delta_{L-2}}\mathbf{P}_L \end{cases} \tag{15}$$

For clarification of using this method, an illustration of a C^2 cubic uniform B-spline curve with 7 deBoor points (i.e. $L = 4$) is shown in Fig. 5. The B-spline polygon formed by $\mathbf{P}_{-1}, \ldots, \mathbf{P}_5$ is just the exactly offset of line-fitting curve and the Bézier points $\mathbf{b}_0, \ldots, \mathbf{b}_{12}$ are computed by Eqs. 11–15 with uniform

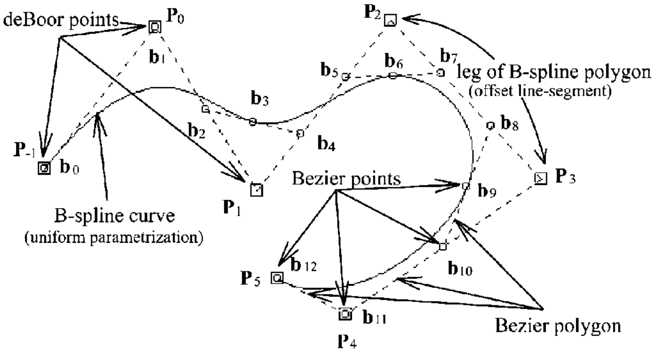


Fig. 5. A C^2 cubic B-spline curve with 7 deBoor points

parametrization. Consequently, the desired C^2 B-spline offset curve is constructed as a piecewise Bézier curve with 4 curve segments.

As shown in the above constructing procedure, the shape of a C^2 offset curve is set down by the following two factors: (1) a set of given B-spline control points; and (2) the parametrization of the B-spline curve. The deBoor points used for constructing the offset curve in this method are inherited from the end points of straight line segments of offset line-fitting curve and deeply affected by the error control mechanism described in next section. Because the parametrizations are selective in constructing the offset curve, the effects for existing parametrization techniques are worth further investigating. Obviously, for offset approximation, a parametrization that results in a smaller approximation error for B-spline curve fitting, i.e. the deviation of B-spline curve from its control polygon, would be preferred. The reason is that the smaller error of B-spline curve fitting, the more loosely the tolerance of original line-fitting curve can be assigned. Then, the number of subdivisions and hence the output data size can also be reduced. Unfortunately, extensive testing has shown that there is no obvious way of parametrization which can generate optimum B-spline curve fitting error.

4 Achieving the required tolerance

The critical issue of the proposed method lies on the error measurement and control. Given a user defined offset tolerance ε, a C^2 continuous offset curve with B-spline form is sought and that such approximation offset does not deviate from the exactly offset curve more than ε everywhere, that is, the error between approximation and exact offset curves should satisfy the following condition in the entire parameter domain:

$$\varepsilon_{t,\max} \le \varepsilon \tag{16}$$

where $\varepsilon_{t,\max}$ represents the upper bound for the total error generated.

In this research, the offset approximation error is generated and cumulated through three stages, which are described in detail as follows. Firstly, in computing the line-fitting curve from the progenitor NURBS curve, an appropriate line-fitting error ε_1, which is a part of ε as figured in Sect. 5, is assigned as a threshold for subdividing. Then, using piecewise Bézier curve segments to construct the offset curve, a second part of error ε_2 is introduced, which is the approximation error of the offset curve to the corresponding Bézier polygons. Finally, a third part of error ε_3 is produced between the offset line-fitting curve (i.e. the B-spline polygon of offset curve) and the piecewise Bézier polygon. Based on the idea of cumulative position error, the total error, ε_t, is defined as the sum of the three parts of approximation errors described above. That is,

$$\varepsilon_t = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \tag{17}$$

And an upper bound for the total error, $\varepsilon_{t,\max}$, is identified as the maximum value of Eq. 17.

In order to estimate the offset approximation error with global preciseness, the following measuring mechanism is established:

- assign the first part error ε_1 and find the second part error ε_2 as the largest distance of any interior control point from the base line based on the convex hull property of rational and non-rational Bézier curves, as shown in Fig. 1; and
- calculate the third part error ε_3 by measuring the height of triangles, which are formed by the B-spline polygon and its constituent Bézier polygons. As shown in Fig. 6, the triangle vertices are deBoor points and a triangle formed by line segments $\mathbf{b}_{11}\mathbf{P}_4$, $\mathbf{P}_4\mathbf{b}_{13}$, and $\mathbf{b}_{13}\mathbf{b}_{11}$ has a height as the perpendicular distance between point $\mathbf{P}_4$ and line segment $\mathbf{b}_{11}\mathbf{b}_{13}$.

Some notes of error measurement using this method are described as follows. Firstly, the third part error for B-spline polygon deviates from the decomposed Bézier polygons is calculated only at finite regions where the legs of Bézier polygons are not consistent with the B-spline polygon. For example, as shown in Fig. 6, there are Bézier polygon legs $\mathbf{b}_2\mathbf{b}_3$, $\mathbf{b}_3\mathbf{b}_4, \dots,$ $\mathbf{b}_{14}\mathbf{b}_{15}$, $\mathbf{b}_{15}\mathbf{b}_{16}$ which are not consistent with the B-spline polygon formed by $\mathbf{P}_{-1}, \dots, \mathbf{P}_7$. Secondly, the chosen error ε_1 is used as a criterion for subdividing, hence the line-fitting curve globally approximate the progenitor NURBS curve without deviating the given error $\varepsilon_{1.}$ Thirdly, the errors ε_2 and ε_3 are not known in advance because they are not generated until the C^2 B-spline offset curve has been constructed. Consequently, an upper bound for the total error in implementation, $\varepsilon_{t,\max}$, will be the summation of the assigned line-fitting error ε_1 with subsequently produced upper error bounds of ε_2 and ε_3.

Now, considering the proposed error measure and control mechanism, the major steps of the algorithm for approximating freeform offsets by C^2 B-spline curve are summarized as follows:

1. Input a progenitor NURBS curve and the user prescribed offset approximation error ε.
2. Decompose the NURBS entity into Bézier pieces.

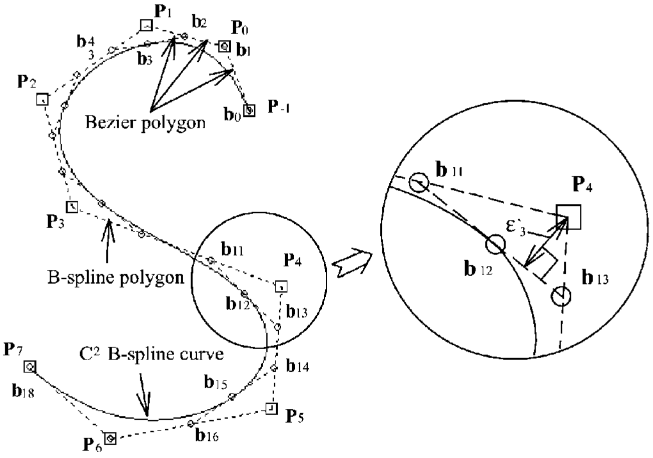


Fig. 6. Error between B-spline polygon and the decomposed Bézier polygons

3. Check whether all of the decomposed Bézier curves are within an initial suggested linear approximation tolerance ε_1, $\varepsilon_1 < \varepsilon$, or not:
   3.1. If it is yes, go to step 4;
   3.2. Else, recursively subdivide each of the Bézier curves, which has convex hull size greater than the allowance ε_1, until the required error ε_1 is achieved.
4. Construct the NURBS line-fitting curve using all of the flattened Bézier curves.
5. Compute the exact offset of the line-fitting curve.
6. Replace the offset line-fitting curve with a C^2 continuous B-spline curve. That is, to use the offset line-fitting curve as a B-spline polygon and to choose a set of adequate knot sequence, the approximated offset curve of B-spline form is obtained by constructing its constituent Bézier curve segments with C^2 continuity.
7. Estimate the B-spline curve fitting error, which is the sum of the maximum error existing between the constructed piecewise Bézier curve and the corresponding Bézier polygons ($(\varepsilon_2)_{\max}$) plus the maximum error caused by the deviation of the decomposed Bézier polygons from the B-spline polygon ($(\varepsilon_3)_{\max}$).
8. Compute the generated total error bound, $\varepsilon_{t,\max}$, as a summation of the chosen line-fitting error ε_1 and the obtained B-spline curve fitting error, i.e., $\varepsilon_{t,\max} = \varepsilon_1 + (\varepsilon_2)_{\max} + (\varepsilon_3)_{\max}$.
9. Check if the cumulative error bound, $\varepsilon_{t,\max}$, is within the required tolerance ε or not:
   9.1. If $\varepsilon_{t,\max} <= \varepsilon$, the procedure is terminated;
   9.2. Else, subdivide the input curve based on a tighter error, $\varepsilon_1 = \varepsilon_1 - \Delta\varepsilon$ (e.g., put $\Delta\varepsilon = 5\%\varepsilon$), and then go back to step 4.

In the method, only linear geometric calculations and the measure of cumulative position tolerance are used for error estimation, the offset approximation error can be computed easily with global accuracy. Unlike the methods using discrete and finite points sampling, which offer no certainty on error bound, this method always guarantee that the error is within the designated tolerance. The above procedure of error control indicates that a good assignment of the initial line-fitting error provides a better computational efficiency. As a result of extensive testing, a value of 30 percent of the prescribed tolerance ε is recommended for the initial ε_1. Some examples are given in Sect. 5.

5 Implementation and examples

The method described above has been implemented to offset cubic NURBS curves for demonstrating its effectiveness and usefulness. In the following experimental and practical examples, handling on error measure and control are based on the proposed mechanism as described in Sect. 4, and the generated cubic offset curves are C^2 continuous with uniform parametrization.

The progenitor curve in Fig. 7 is a uniform cubic B-spline curve with 6 control points: (345, 380), (472, 339), (342, 261), (504, 210), (383, 98), and (289, 187). An offset distance 20 is used for the example. In order to understand the effects of generated total approximation error bound ($\varepsilon_{t,\max}$) influenced by the selected initial line-fitting error (ε_1), an empirical study is performed. The result is shown in Table 1 with the prescribed tolerance (ε) in the range of engineering design, i.e. $10^{-1} \sim 10^{-5}$. Mark "×" appearing in Table 1 denotes that the selected ε_1 induces a cumulative error greater than the tolerance. The maximal value of $\varepsilon_1/\varepsilon$, which can be reached without exceeding the prescribed tolerance, is the best value for this offsetting with the least control points while satisfying prescribed tolerance.

Another example for offsetting a typical cubic NURBS curve is illustrated in Fig. 8. In this example, the progenitor curve is given with 10 control points: (443,263), (541, 342), (429, 415), (348, 366), (366, 296), (480, 217), (502,143), (394, 107), (337, 184), and (422, 245). The weight of each control point and the knot vector are as follows:

$$w = \{1, 1, 1.5, 1, 1.2, 1.8, 1, 1, 1, 1\},$$
$$T = \{0, 0, 0, 0, 0.142857, 0.285714, 0.428571, 0.6, 0.714286, 0.857143, 1, 1, 1, 1\}.$$

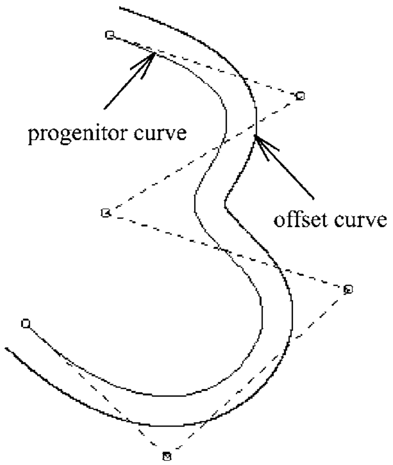
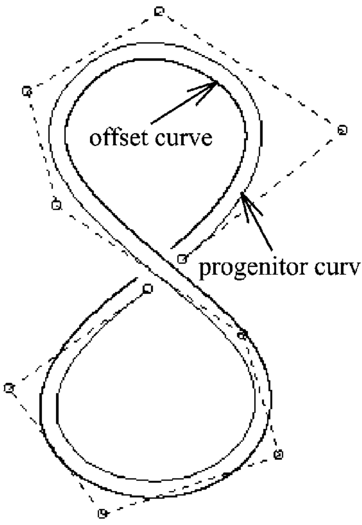


Fig. 7. Offset of a cubic B-spline curve (tolerance $\varepsilon = 10^{-1}$)

Table 1. Cumulative approximation error bound $\varepsilon_{t,\max}(\varepsilon = 10^{-1} \sim 10^{-5})$

$\varepsilon_1/\varepsilon$	5%	10%	15%	20%	25%	30%	35%
$\varepsilon = 10^{-1}$	0.021685	0.035934	0.058260	0.084392	0.094239	0.099239	×
$\varepsilon = 10^{-2}$	0.001814	0.004151	0.005837	0.007229	0.009471	×	
$\varepsilon = 10^{-3}$	0.000208	0.000371	0.000584	0.000828	×		
$\varepsilon = 10^{-4}$	0.000022	0.000040	0.000063	0.000088	0.000093	×	
$\varepsilon = 10^{-5}$	0.000002	0.000004	0.000006	0.000008	×		

Fig. 8. Offset of a cubic NURBS curve (tolerance $\varepsilon = 10^{-1}$)



The offset distance used in this case is 10. The changes of $\varepsilon_{t,\max}$ with respect to the ratio $\varepsilon_1/\varepsilon$ are shown in Table 2.

As the above results indicate, an initial ε_1 must be assigned appropriately in order to compute the offsetting more efficiently. Empirical studies have shown that $\varepsilon_1/\varepsilon = 30\%$ is a good recommended value in the range of engineering tolerances, i.e. $\varepsilon \leq 10^{-1}$. Although a smaller assigned ε_1 will provide a better approximation of line-fitting curve and a better constructed offset curve, the generated total approximation error bound may become excessively smaller than the prescribed tolerance and an extra expense of over subdividing is induced. On the other hand, if an initial bigger ε_1 is used for computing offsets, the total approximation error bound with a larger value will not satisfy the prescribed tolerance, the computational efficiency will be reduced because iterations are requested to achieve the required tolerance. Therefore, a proper initial value for ε_1 may save computations and reduce the output data size.

Based on the above two cases, an additional test is carried out to see the number of control points generated according to different precisions of the prescribed tolerance. Table 3 shows the results with near optimal ε_1 assigned.

At last, NURBS format tool-path generation, which is one of the most important applications for NURBS curve offsetting, is demonstrated with an example as shown in Fig. 9. The tool paths for 2.5D freeform profile machining (profiling) are obtained by repeatedly offsetting the given NURBS boundary curve. As shown in the figure, there are two types of tool-paths are identified. The curve of first offsetting is used as a finish cutting tool path, and other offset curves show the rough cutting tool paths. Generally, to reduce output data size and to increase machining efficiency, prescribed tolerances for the two types of tool paths may be assigned differently. In this demonstration, the tolerances assigned for calculating finish cutting and rough cutting tool paths are 0.1 and 0.2, respectively. The main benefits and features of this machining strategy are further described as follows.

Table 3. Number of generated control points

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Fig. 7	167	620	2197	5321	20459
Fig. 8	264	981	3114	8636	32688

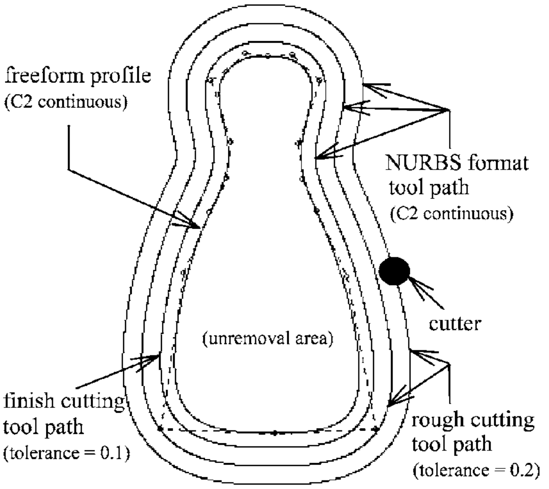


Fig. 9. NURBS-format tool path generation for profiling ($\varepsilon_1/\varepsilon = 20\%$, cutter radius = 10, cutter overlap = 70%)

Conventional CNC machines are only capable of linear and circular arc interpolations, the freeform curves and hence the tool paths for profiling must be approximated and represented by straight lines or arcs before NC machining to start running [20]. With the advent of commercial CNC controllers that supports NURBS output formats (e.g., Fanuc and Siemens), a more popular cutting strategy is naturally turned to directly use interpolator's NURBS function to carry out a NURBS-based machining. As a smoother and higher-speed cutting is considered, undoubtedly, NURBS-based machining would be a better choice. Since the contour-parallel method is favored for profiling or pocketing, in which the cutter is guided to remove material using only one

Table 2. Cumulative approximation error bound $\varepsilon_{t,\max}(\varepsilon = 10^{-1} \sim 10^{-5})$

$\varepsilon_1/\varepsilon$	5%	10%	15%	20%	25%	30%	35%
$\varepsilon = 10^{-1}$	0.014835	0.033022	0.052599	0.058581	0.084325	0.095473	×
$\varepsilon = 10^{-2}$	0.001660	0.003447	0.005019	0.006584	0.008415	×	
$\varepsilon = 10^{-3}$	0.000178	0.000347	0.000509	0.000711	0.000867	×	
$\varepsilon = 10^{-4}$	0.000015	0.000034	0.000052	0.000061	0.000087	×	
$\varepsilon = 10^{-5}$	0.000002	0.000003	0.000005	0.000007	×		

consistent milling type (up or down milling), the pattern of tool paths selected in this example is regarded as appropriate [11]. It is hoped that the proposed approach with C^2 continuous NURBS curve offsetting will play a key role to elevate the level of automatic tool-path generation with NURBS-based NC machining.

Since all of the operations involved in computing offsets are linear geometric calculations, i.e., line intersections, point distances, linear interpolations, and vector operations such as dot and cross products, the proposed method is very efficient and robust. It is believed that, in engineering applications, a method with an accurate, efficient and robust offset approximation would be more important than saving a few control points in representing the offset curve. The proposed approach is proven to successfully fulfill these demands.

6 Conclusions

A method is proposed to approximate the offsets of arbitrary NURBS curves using C^2 integral B-spline curves. Based on the convex hull property of Bézier curves and the idea of cumulative position errors, this method can provide an accurate, global and fast measurement on computing the offset approximation error. Since all operations involved in computing offsets are linear geometric calculations, the algorithm is very efficient and robust. The proposed approach is most suitable for engineering applications where the offset curves require higher degree of continuity, for examples, as the need of visual purpose and certain practical requirements such as movement smoothness in robot trajectory or automatic machines.

As a next step, it is worth investigating to further reduce the number of generated control points by assigning the initial line-fitting error ε_1 more properly or by using a newer and better parametrization technique for constructing the C^2 offset curve. Extending this method on surface offsetting is another future direction. It is hoped that all aspects of problems now existing in computing offsets of NURBS curves and surfaces can be solved efficiently and robustly in the near future.

References

1. Boehm W (1977) Cubic B-spline curves and computer aided geometric design. Computing 19(1):29–34
2. Boehm W, Prautzsch H (1985) The insertion algorithm. Comput Aided Des 17(2):58–59
3. Chuang SH, Kao CZ (1999) One-sided arc approximation of B-spline curves for interference-free offsetting. Comput Aided Des 31(2): 111–118
4. Coquillart S (1987) Computing offsets of B-splines curves. Comput Aided Des 19(5):305–309
5. Elber G, Cohen E (1991) Error bounded variable distance offset operator for free form curves and surfaces. Int J Comput Geom Appl 1(1):67–78
6. Farin G (1993) Curves and surfaces for computer aided geometry design, a practical guide, 3rd ed. Academic Press, San Diego, CA
7. Farouki RT, Neff C (1990) Analytic properties of plane offset curves. Comput Aided Geom Des 7:83–100
8. Farouki RT, Neff C (1990) Algebraic properties of plane offset curves. Comput Aided Geom Des 7:101–127
9. Faux ID, Pratt MJ (1979) Computational geometry for design and manufacture. Ellis Horwood, Chinchester, UK
10. Hansen A, Arbab F (1992) An algorithm for generating NC tool paths for arbitrary shaped pockets with islands. ACM Trans Graph 12(2): 152–182
11. Held M (1991) On the computational geometry of pocket machining. Springer-Verlag, Berlin, Germany
12. Hoschek J (1988) Spline approximation of offset curves. Comput Aided Geom Des 5(1):33–40
13. Klass K (1983) An offset spline approximation for plane cubic splines. Comput Aided Des 15(4):297–299
14. Lee IK, Kim Ms, Elber G (1996) Plane curve offset based on circle approximation. Comput Aided Des 28(7):617–630
15. Pham B (1988) Offset approximation of uniform B-splines. Comput Aided Des 20(7):471–474
16. Piegl L, Tiller W (1997) The NURBS book, 2nd ed. Springer, New York
17. Piegl L, Tiller W (1999) Computing offsets of NURBS curves and surfaces. Comput Aided Des 31:147–156
18. Ravi Kumar GVV, Shastry KG, Prakash BG (2002) Computing non-self-intersecting offsets of NURBS surfaces. Comput Aided Des 34(3):209–228
19. Tiller W, Hanson EG (1984) Offsets of two-dimensional profiles. IEEE Comput Graph Appl 4(8):61–69
20. Tseng YG, Chen YD, Liu CC (2001) Numerically controlled machining of freeform curves using biarc approximation. Int J Adv Manuf Tech 17(11):783–790