Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A new extension algorithm for cubic B-splines based on minimal strain energy

MO Guo-liang†, ZHAO Ya-nan

(*Department of Information and Computational Science, Zhejiang University City College, Hangzhou 310015, China*)

†E-mail: mogl@zucc.edu.cn

Received Mar. 28, 2006; revision accepted July 3, 2006

Abstract: Extension of a B-spline curve or surface is a useful function in a CAD system. This paper presents an algorithm for extending cubic B-spline curves or surfaces to one or more target points. To keep the extension curve segment GC^2-continuous with the original one, a family of cubic polynomial interpolation curves can be constructed. One curve is chosen as the solution from a sub-class of such a family by setting one GC^2 parameter to be zero and determining the second GC^2 parameter by minimizing the strain energy. To simplify the final curve representation, the extension segment is reparameterized to achieve C^2-continuity with the given B-spline curve, and then knot removal from the curve is done. As a result, a sub-optimized solution subject to the given constraints and criteria is obtained. Additionally, new control points of the extension B-spline segment can be determined by solving lower triangular linear equations. Some computing examples for comparing our method and other methods are given.

Key words: GC^2-continuous, Extension, Minimal strain energy, Knot removal, Reparametrization
doi:10.1631/jzus.2006.A2043 **Document code:** A **CLC number:** TP391.72

INTRODUCTION

B-spline curves and surfaces have been widely used in Computer Graphics (CG) and Computer Aided Design (CAD) (Hoschek and Lasser, 1993; Piegl and Tiller, 1997). Many practical algorithms, such as those for position and derivatives evaluation, knot insertion, knot deletion and degree elevation, are usually implemented in a CAD system that uses B-spline as a shape design tool.

In curve and surface design, a given B-spline curve or surface usually needs to be extended in order to meet some geometric shape conditions or engineer requirements. Extending a B-spline curve to a target point is to generate a curve segment interpolating a prescribed point. Then the generated curve is represented as a B-spline form. There are many methods for curve and surface interpolation problem (Hoschek and Lasser, 1993; Shi, 2001). In some CAD systems, a common solution is to add a Bézier curve with GC^1-continuity at the end of the B-spline curve, and then convert the entire curve into B-spline form. Shetty and White (1991) presented a straight algorithm for the extension of rational B-spline curves and surfaces without modifying the shape and the parametrization of the original curves and surfaces. The extended B-spline curve is represented with knot vector in the form of $\{0, 0, 0, 0, \ldots, 1, 1, 1, 1, s, s, s, s\}$, $s>1$. However, a B-spline curve with knot vector in this form is not favored in CAD applications. Hu *et al.*(2002) proposed a new extension algorithm for B-splines by curve unclamping. With their method, some control points of the extended B-spline curve need to be re-adjusted many times when extending to multiple target points. Also the knot vectors are computed according to accumulated chord length and the generated curves are not good fairness in some large torsion cases (Farin and Sapidis, 1989; Su and Liu, 1982).

In this paper, the shape of the extended curve is determined by minimizing strain energy (Tai *et al.*, 2003). The knot value is determined by the conditions that the original curve and the extended one meet GC^2-continuity at the joint point. A cubic polynomial interpolation curve is constructed and is converted into a B-spline form. The new control points are determined by knot removal algorithm or solving lower triangular linear equations.

The paper is organized as follows. The definition of the B-spline curve and Boehm knot insertion algorithm are described in Section 2. Section 3 presents the interpolation method for extension of B-spline curves. Some computing examples are given and experiments for comparing our method with other methods are also included. Conclusion is given in Section 4.

B-SPLINE CURVES AND BOEHM KNOT INSERTION ALGORITHM

A B-spline curve of order k (or degree $k-1$) with control points $\boldsymbol{p}_i$ (i=0, 1, …, n) can be defined as:

$$\boldsymbol{p}(t)=\sum_{i=0}^{n} N_{i,k}(t)\boldsymbol{p}_i, \qquad t_{k-1}\le t\le t_{n+1},\ n\ge k,$$

where $N_{i,k}(t)$ are the B-spline basis functions of order k defined over the knot vector $\boldsymbol{T}=\{t_0, \ldots, t_k, \ldots, t_n, \ldots, t_{n+k}\}$ and can be defined by the well-known de Boor-Cox formula (Farin, 1997; Piegl and Tiller, 1997):

$$N_{i,1}(t)=\begin{cases}1, & t\in[t_i,t_{i+1});\\ 0, & \text{otherwise},\end{cases}$$

$$N_{i,k}(t)=\frac{t-t_i}{t_{i+k-1}-t_i}N_{i,k-1}(t)+\frac{t_{i+k}-t}{t_{i+k}-t_{i+1}}N_{i+1,k-1}(t),\ k\ge 2.$$

Boehm knot insertion algorithm is widely used in evaluating points and derivatives on curves and surfaces, subdividing curves and surfaces, adding control points in order to increase flexibility in shape control (Piegl and Tiller, 1997).

Knot vectors are classified as clamped and unclamped (Piegl and Tiller, 1997). Clamped knot vector refers to the first and last knot values that are repeated with multiplicity equal to order (or degree plus one). Otherwise, it is an unclamped knot vector. Many systems today use only clamped B-splines. Repeated insertion end knots into a knot vector can make an unclamped knot vector become a clamped one. In this paper, cubic B-splines are considered. For clamped knot vectors of order 4, it is common to use the following standard form:

$$\boldsymbol{T}_1=\{0, 0, 0, 0, t_4, \ldots, t_n, 1, 1, 1, 1\}.$$

EXTENSION OF B-SPLINE CURVES

Extending a B-spline curve to a target point

For a given cubic B-spline curve $\boldsymbol{p}(t)=\sum_{i=0}^{n} N_{i,4}^{1}(t)\boldsymbol{p}_i$ ($0\le t\le 1$, $n\ge 4$) defined on the knot vector $\boldsymbol{T}_1$, a cubic polynomial interpolation curve $\boldsymbol{r}(u)$ is constructed in order to extend the curve to the target $\boldsymbol{p}_{n+1}$. The extended curve segment and the original $\boldsymbol{p}(t)$ are GC^2-continuous (second geometric continuous). Based on minimizing strain energy, an appropriate parameter is chosen and the extended curve is reparameterized. Then the extended curve and the original $\boldsymbol{p}(t)$ are C^2-continuous (second continuous) at the joint point. The whole curve is represented as a B-spline form on a knot vector with multiplicity of knot values greater than one. Knot removal is feasible and the new curve is simplified.

To interpolate target point $\boldsymbol{p}_{n+1}$, and keep GC^2-continuity conditions with $\boldsymbol{p}(t)$, a cubic polynomial interpolation curve $\boldsymbol{r}(u)$ defined on [0,1] is constructed and satisfies:

$$\begin{cases}\boldsymbol{r}(0)=\boldsymbol{p}(1),\\ \boldsymbol{r}'(0)=\alpha\boldsymbol{p}'(1),\\ \boldsymbol{r}''(0)=\alpha^2\boldsymbol{p}''(1)+\beta\boldsymbol{p}'(1),\end{cases}$$

where $\alpha>0$.

In order that the reparameterized curve $\bar{\boldsymbol{r}}(t)=\boldsymbol{r}(u)$ is still cubic, the coefficient β is set to 0. According to interpolation conditions, $\boldsymbol{r}(u)$ can be represented as:

$$\boldsymbol{r}(u)=\varphi_0(u)\boldsymbol{p}(1)+\varphi_1(u)\alpha\boldsymbol{p}'(1)+\varphi_2(u)\alpha^2\boldsymbol{p}''(1)+\varphi_3(u)\boldsymbol{p}_{n+1},$$

where $\varphi_0(u)=1-u^3$, $\varphi_1(u)=u-u^3$, $\varphi_2(u)=(u^2-u^3)/2$, $\varphi_3(u)=u^3$ are cubic polynomial functions.

The Bézier form of $\boldsymbol{r}(u)$ is

$$\boldsymbol{r}(u)=\sum_{j=0}^{3}B_j^3(u)\boldsymbol{r}_j,$$

where $B_j^3(u)=\binom{3}{j}u^{3-j}(1-u)^j\ (j=0,1,2,3)$ are classical Bernstein basis functions on [0,1]. The control points of $\boldsymbol{r}(u)$ are determined by

$$\begin{pmatrix}\boldsymbol{r}_0\\ \boldsymbol{r}_1\\ \boldsymbol{r}_2\\ \boldsymbol{r}_3\end{pmatrix}=\begin{pmatrix}1&0&0&0\\ 1&1/3&0&0\\ 1&2/3&1/6&0\\ 0&0&0&1\end{pmatrix}\begin{pmatrix}\boldsymbol{p}(1)\\ \alpha\boldsymbol{p}'(1)\\ \alpha^2\boldsymbol{p}''(1)\\ \boldsymbol{p}_{n+1}\end{pmatrix}.$$

The strain energy (Su and Liu, 1982) function of $\boldsymbol{r}(u)$ is expressed as

$$EG=\int_0^1\left\|\boldsymbol{r}''(u)\right\|^2\mathrm{d}u.$$

Generally speaking, if a curve has minimal strain energy, it looks more fairing and natural, and it is smoother. The minimum of EG is our goal. It is a variation problem (Farin and Sapidis, 1989), that is,

$$\min(EG)=\int_0^1\left\|\boldsymbol{r}''(u)\right\|^2\mathrm{d}u.$$

According to variation principle, EG is differentiated with respect to α:

$$\frac{\mathrm{d}(EG)}{\mathrm{d}\alpha}=0,\ \alpha>0,$$

that is,

$$\begin{aligned}&2\left(\boldsymbol{p}''(1)\cdot\boldsymbol{p}''(1)\right)\alpha^3+9\left(\boldsymbol{p}'(1)\cdot\boldsymbol{p}''(1)\right)\alpha^2\\ &+6\left(\boldsymbol{p}(1)\cdot\boldsymbol{p}''(1)+2\boldsymbol{p}'(1)\cdot\boldsymbol{p}'(1)-\boldsymbol{p}''(1)\cdot\boldsymbol{p}_{n+1}\right)\alpha\\ &+12\left(\boldsymbol{p}(1)\cdot\boldsymbol{p}'(1)-\boldsymbol{p}'(1)\cdot\boldsymbol{p}_{n+1}\right)=0,\end{aligned}$$

where $\boldsymbol{p}(1)\cdot\boldsymbol{p}'(1)$ denotes the dot product of two vectors $\boldsymbol{p}(1)$ and $\boldsymbol{p}'(1)$.

Solve the above algebraic equation to yield a positive α. Let $u=(t-1)/\alpha$, and then $\boldsymbol{r}(u)$ is reparameterized as $\bar{\boldsymbol{r}}(t)=\boldsymbol{r}\left(\frac{t-1}{\alpha}\right)$, $t\in[1,1+\alpha]$. $\bar{\boldsymbol{r}}(t)$ and $\boldsymbol{p}(t)$ are C^2 continuous at t=1. Then a new curve is constructed as follows:

$$\bar{\boldsymbol{p}}(t)=\begin{cases}\boldsymbol{p}(t), & t\in[0,1];\\ \bar{\boldsymbol{r}}(t), & t\in(1,1+\alpha].\end{cases}$$

Theorem 1 $\bar{\boldsymbol{p}}(t)$ is a C^2 continuous curve that keeps the original shape of $\boldsymbol{p}(t)$ and interpolates the target point $\boldsymbol{p}_{n+1}$. $\bar{\boldsymbol{p}}(t)$ is a B-spline curve defined on [0, 1+α] with knot vector

$\boldsymbol{T}_2$={0, 0, 0, 0, t_4, …, t_n, 1, 1, 1, 1+α, 1+α, 1+α, 1+α}

with multiplicity of 1 equal to 3 and control points {$\boldsymbol{p}_0$, $\boldsymbol{p}_1$, …, $\boldsymbol{p}_n$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, $\boldsymbol{p}_{n+1}$}.

Proof The results are from the construction of $\bar{\boldsymbol{p}}(t)$.

Theorem 2 The multiplicity of 1 in the knot vector $\boldsymbol{T}_2$ can be reduced to 1, that is, 1 can be removed from $\boldsymbol{T}_2$ twice (Tiller, 1992). $\bar{\boldsymbol{p}}(t)$ can be expressed as

$$\bar{\boldsymbol{p}}(t)=\sum_{i=0}^{n+1}N_{i,4}^3(t)\bar{\boldsymbol{p}}_i,\quad 0\le t\le 1+\alpha,\ n\ge 4,$$

with knot vector

$\boldsymbol{T}_3$={0, 0, 0, 0, t_4, …, t_n, 1, 1+α, 1+α, 1+α, 1+α}

and control points

$$\bar{\boldsymbol{p}}_i=\begin{cases}\boldsymbol{p}_i, & i=0,1,\cdots,n-2;\\ \dfrac{(1+\alpha-t_{n-1})\boldsymbol{p}_{n-1}-\alpha\boldsymbol{p}_{n-2}}{1-t_{n-1}}, & i=n-1;\\ \boldsymbol{r}_2, & i=n;\\ \boldsymbol{p}_{n+1}, & i=n+1.\end{cases}$$

Proof The knot insertion process is displayed in the following formulae and Fig.1.

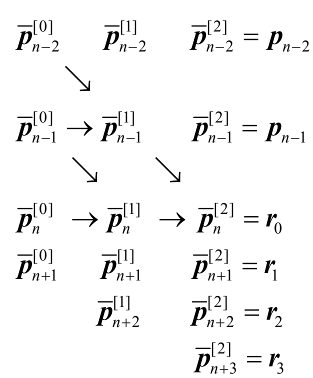


Fig.1 Knot insertion process for the knot 1

Let $t'=1$, $\bar{\boldsymbol{p}}_i^{[0]}=\bar{\boldsymbol{p}}_i$ $(i=0,1,\cdots,n+1)$ and run the following procedure:

$$\bar{\boldsymbol{p}}_i^{[r]}=\left(1-\lambda_i^{[r]}\right)\bar{\boldsymbol{p}}_{i-1}^{[r-1]}+\lambda_i^{[r]}\bar{\boldsymbol{p}}_i^{[r-1]},$$

$$\lambda_i^{[r]}=\begin{cases}1, & i\le n+r-3;\\ \dfrac{t'-t_i}{t_{i+3-r}-t_i} & n+r-2\le i\le n;\\ 0, & n+1\le i\le n+r+1,\end{cases}$$

$$r=1,\ 2.$$

The knot vector $\boldsymbol{T}_3$ is converted into $\boldsymbol{T}_2$ to obtain the corresponding control points $\bar{\boldsymbol{p}}_i^{[2]}$ (i=0, 1, …, n+3). By inverse knot insertion, the two control points are computed as

$$\bar{\boldsymbol{p}}_{n-1}=\frac{(1+\alpha-t_{n-1})\boldsymbol{p}_{n-1}-\alpha\boldsymbol{p}_{n-2}}{1-t_{n-1}},$$

$$\bar{\boldsymbol{p}}_n=\frac{1}{(1-t_{n-1})(1-t_n)^2}\Big\{(1-t_{n-1})(1+\alpha-t_n)^2\boldsymbol{p}_n -\alpha\left[2(1+\alpha)-(2+\alpha)(t_n+t_{n-1})+2t_nt_{n-1}\right]\boldsymbol{p}_{n-1} +\alpha^2(1-t_n)\boldsymbol{p}_{n-2}\Big\}.$$

Then according to knot insertion formula again, $\bar{\boldsymbol{p}}_{n+j}^{[2]}=\boldsymbol{r}_j$ (j=0,1,2,3) can be drawn. It is shown that the B-spline $\bar{\boldsymbol{p}}(t)=\sum_{i=n}^{n+3}N_{i,4}^2(t)\bar{\boldsymbol{p}}_i^{[2]}$ $(1\le t\le 1+\alpha)$ with knot vector $\boldsymbol{T}_2$ is just the cubic Bézier curve $\bar{\boldsymbol{r}}(t)$. Knot removal is the inverse process of knot insertion. The knot 1 can be removed twice from $\boldsymbol{T}_2$ indeed because the knot 1 is inserted into $\boldsymbol{T}_3$ twice.

The knot removal process is depicted in Fig.2.

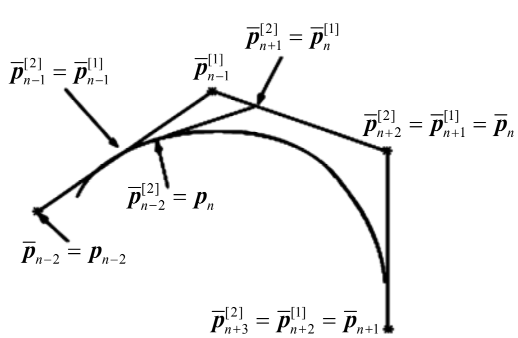


Fig.2 Knot removal process for knot 1

Algorithm 1: Knot removal and computing control points

Let t'=1, j=n, $\bar{\boldsymbol{p}}_i^{[2]}=\bar{\boldsymbol{p}}_i$ $(i=j-2,j-1,j)$.

$$\begin{cases}\bar{\boldsymbol{p}}_i^{[r-1]}=\bar{\boldsymbol{p}}_i^{[r]}, & i=j-r,j-1;\\ \bar{\boldsymbol{p}}_i^{[r-1]}=\dfrac{\bar{\boldsymbol{p}}_i^{[r]}-\left(1-\lambda_i^{[r]}\right)\bar{\boldsymbol{p}}_{i-1}^{[r-1]}}{\lambda_i^{[r]}}, & i=j+r-2,j,\end{cases}$$

where $\lambda_i^{[r]}=\dfrac{t'-t_i}{t_{i+3-r}-t_i}$, r=2,1;

$$\bar{\boldsymbol{p}}_i=\bar{\boldsymbol{p}}_i^{[0]},\quad i=j-2,j-1,j.$$

The control points $\bar{\boldsymbol{p}}_i$ $(i=0,1,\ldots,n+1)$ of the new cubic B-spline curve $\bar{\boldsymbol{p}}(t)$ are obtained by Algorithm 1.

Knot vector $\boldsymbol{T}_3$ is converted into

$$\boldsymbol{T}_4=\{0,0,0,0,t_4/(1+\alpha),\ldots,t_n/(1+\alpha),1/(1+\alpha),1,1,1,1\},$$

$$\bar{\boldsymbol{p}}(t)=\sum_{i=0}^{n+1}N_{i,4}^4(t)\bar{\boldsymbol{p}}_i,\quad 0\le t\le 1,\ n\ge 4,$$

where $N_{i,4}^4(t)$ are B-spline basis functions defined on standard form $\boldsymbol{T}_4$. Fig.3a is an example for extending a B-spline curve to a prescribed point.

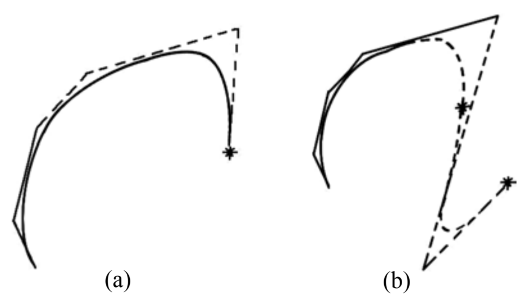


Fig.3 Extension of B-spline curves with one target point and two points. (a) One target points; (b) Two target points

Extending a B-spline curve to two target points

For a given B-spline curve $\boldsymbol{p}(t)=\sum_{i=0}^{n}N_{i,4}^1(t)\boldsymbol{p}_i$ $(0\le t\le 1,\ n\ge 4)$ defined on $\boldsymbol{T}_1$, two cubic polynomial interpolation curves $\boldsymbol{r}(u)$ and $\boldsymbol{s}(u)$ are constructed in order to extend the curve to $\boldsymbol{p}_{n+1}$ and $\boldsymbol{p}_{n+2}$.

Suppose that $\boldsymbol{r}(u)$ is constructed in the same as that for the above method, and α_1 is obtained. Then $\boldsymbol{s}(u)$ is constructed as follows:

$$\boldsymbol{s}(u) = \varphi_0(u)\boldsymbol{r}(1) + \varphi_1(u)\alpha_2\boldsymbol{r}'(1) + \varphi_2(u)\alpha_2^2\boldsymbol{r}''(1) + \varphi_3(u)\boldsymbol{p}_{n+2} = \sum_{j=0}^{3} B_j^3(u)\boldsymbol{s}_j.$$

According to minimal strain energy, α_2 can be obtained also. Then $\boldsymbol{r}(u)$ and $\boldsymbol{s}(u)$ are reparameterized as

$$\overline{\boldsymbol{r}}(t) = \boldsymbol{r}\left(\frac{t-1}{\alpha_1}\right), \quad t \in (1, 1+\alpha_1],$$
$$\overline{\boldsymbol{s}}(t) = \boldsymbol{s}\left(\frac{t-1-\alpha_1}{\alpha_1\alpha_2}\right), \quad t \in (1+\alpha_1, 1+\alpha_1+\alpha_2].$$

$\overline{\boldsymbol{p}}(t)$ is constructed as follows:

$$\overline{\boldsymbol{p}}(t) = \begin{cases} \boldsymbol{p}(t), & t \in [0,1]; \\ \overline{\boldsymbol{r}}(t), & t \in (1, 1+\alpha_1]; \\ \overline{\boldsymbol{s}}(t). & t \in (1+\alpha_1, 1+\alpha_1+\alpha_2]. \end{cases}$$

Theorem 3 The newly constructed curve $\overline{\boldsymbol{p}}(t)$ is composed of three segments: the original one, $\boldsymbol{r}(u)$ and $\boldsymbol{s}(u)$, and interpolates $\boldsymbol{p}_{n+1}$ and $\boldsymbol{p}_{n+2}$. $\overline{\boldsymbol{p}}(t)$ is a C^2 continuous B-spline curve with knot vector

$\boldsymbol{T}_5=\{0, 0, 0, 0, t_4, \ldots, t_n, 1, 1, 1, 1+\alpha_1, 1+\alpha_1, 1+\alpha_1, 1+\alpha_1+\alpha_2, 1+\alpha_1+\alpha_2, 1+\alpha_1+\alpha_2, 1+\alpha_1+\alpha_2\}$,

and control points

$\{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{p}_{n+1}, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{p}_{n+2}\}$.

The multiplicity of knot 1 and knot $1+\alpha_1$ can be reduced to 1.

Proof This can be processed according to Algorithm 1: first removing knot 1 twice and then knot $1+\alpha_1$ twice.

Then the $\overline{\boldsymbol{p}}(t)$ can be converted into $\overline{\boldsymbol{p}}(t) = \sum_{i=0}^{n+2} N_{i,4}^6(t)\overline{\boldsymbol{p}}_i$ $(0 \le t \le 1+\alpha_1+\alpha_2,\ n \ge 4)$ with the knot vector

$\boldsymbol{T}_6=\{0, 0, 0, 0, t_4, \ldots, t_n, 1, 1+\alpha_1, 1+\alpha_1+\alpha_2, 1+\alpha_1+\alpha_2, 1+\alpha_1+\alpha_2, 1+\alpha_1+\alpha_2\}$.

The knot vector $\boldsymbol{T}_6$ then is standardized and converted into

$$\boldsymbol{T}_7 = \left\{0,0,0,0,\frac{t_4}{1+\alpha_1+\alpha_2},\cdots,\frac{t_n}{1+\alpha_1+\alpha_2}, \frac{1}{1+\alpha_1+\alpha_2},\frac{1+\alpha_1}{1+\alpha_1+\alpha_2},1,1,1,1\right\}.$$

$\overline{\boldsymbol{p}}(t) = \sum_{i=0}^{n+2} N_{i,4}^7(t)\overline{\boldsymbol{p}}_i$ $(0 \le t \le 1,\ n \ge 4)$, $N_{i,4}^7(t)$ are B-spline basis functions defined on $\boldsymbol{T}_7$. Fig.3b is an example for extending a B-spline curve to two appointed points.

Extension to more than 2 target points is similar and multiple knots must be repeatedly removed.

The crux of the extension problem is to find new control points of the new B-spline curve. It also can be done by solving linear equations.

Algorithm 2: Solving linear equations and computing control points

The new curve $\overline{\boldsymbol{p}}(t)$ and the original curve $\boldsymbol{p}(t)$ have the same values at the same knot values respectively.

$$\begin{cases} \sum_{j=n-4}^{n-1} N_{i,4}^3(t_{n-1})\overline{\boldsymbol{p}}_i = \boldsymbol{p}(t_{n-1}), \\ \sum_{j=n-3}^{n} N_{i,4}^3(t_n)\overline{\boldsymbol{p}}_i = \boldsymbol{p}(t_n), \\ \sum_{j=n-2}^{n+1} N_{i,4}^3(1)\overline{\boldsymbol{p}}_i = \boldsymbol{p}(1), \\ \sum_{j=n-1}^{n+2} N_{i,4}^3(1+\alpha_1)\overline{\boldsymbol{p}}_i = \boldsymbol{p}_{n+1}, \end{cases}$$

or

$$\begin{cases} \sum_{j=n-4}^{n-1} N_{i,4}^6(t_{n-1})\overline{\boldsymbol{p}}_i = \boldsymbol{p}(t_{n-1}), \\ \sum_{j=n-3}^{n} N_{i,4}^6(t_n)\overline{\boldsymbol{p}}_i = \boldsymbol{p}(t_n), \\ \sum_{j=n-2}^{n+1} N_{i,4}^6(1)\overline{\boldsymbol{p}}_i = \boldsymbol{p}(1), \\ \sum_{j=n-1}^{n+2} N_{i,4}^6(1+\alpha_1)\overline{\boldsymbol{p}}_i = \boldsymbol{p}_{n+1}, \\ \sum_{j=n}^{n+3} N_{i,4}^6(1+\alpha_1+\alpha_2)\overline{\boldsymbol{p}}_i = \boldsymbol{p}_{n+2}. \end{cases}$$

If the curve is extended to more target points, some more linear equations are added in. The equations are lower triangular matrix equations and easy to solve by linear equations theory. The algorithm can produce new control points of the extended curve once. There is iterative computation in the knot removal algorithm and a control point needs to be re-adjusted every time if a point is extended to another point. If on extends to m points, there are $m-1$ points to be repeatedly computed.

SOME COMPUTING EXAMPLES

In this section, two examples are given for comparing the accumulated chord length parameterization method (Hu *et al.*, 2002) and our method. The strain energy and rotation number (Do Carmo, 1976) of curves are two indexes for experiment results. Rotation number is an index of global geometric properties in differential geometry. It describes how tangent vectors or normal vectors rotate when they move on the curve. For a planar curve $\boldsymbol{p}(t)$ ($0\le t\le 1$) with its curvature $k(t)$, its absolute number is defined as (Do Carmo, 1976):

$$Rot=\frac{1}{2\pi}\int_0^1 |k(t)|\,\|\boldsymbol{p}'(t)\|\,\mathrm{d}t.$$

From the computing results, the new curves generated by minimal strain energy have less strain energy and less rotation number than the ones by accumulated chord length parameterization, as Table 1 shows. From Fig.4 and Fig.5, the fairness of the two curves generated by our method is greatly improved and the geometric shape becomes more natural.

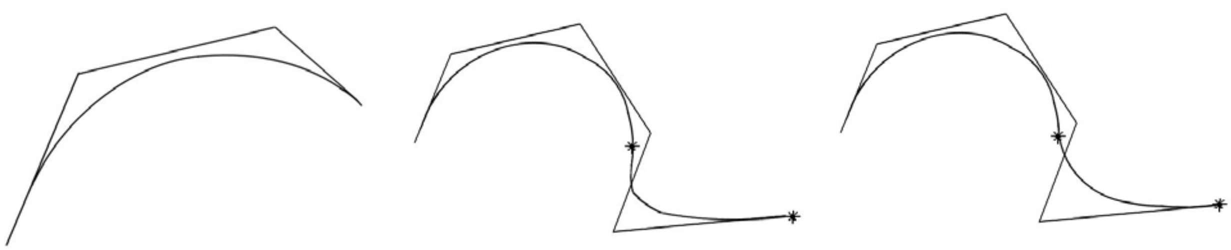
Table 1 Comparison of different computing results

	Accumulated chord length parameterization		Minimal strain energy	
	1st case	2nd case	1st case	2nd case
EG	0.7628	0.6275	0.3852	0.3126
Rot	0.5239	0.3941	0.5213	0.3462

EG: energy; *Rot*: rotation number

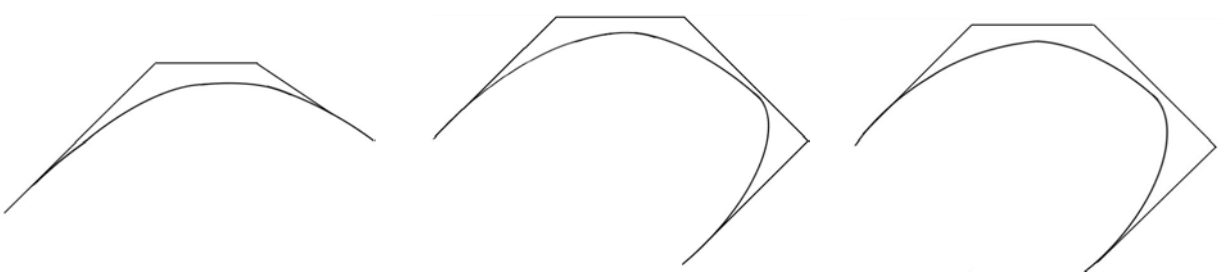
CONCLUSION

In this paper, extension of cubic B-spline curves is discussed. To keep GC^2-continuous at the joint points, cubic polynomial interpolation curves are constructed. Based on minimal strain energy, appropriate knot values are selected. Then the interpolation curves are reparameterized. Some repeated knots are deleted from the knot vector and the knot vector is



(a) (b) (c)

Fig.4 Comparison of different methods. (a) The first original curve; (b) The first new curve by accumulated chord length parameterization; (c) The first new curve by minimal energy method



(a) (b) (c)

Fig.5 Comparison of different methods. (a) The second original curve; (b) The second new curve by accumulated chord length parameterization; (c) The second new curve by minimal energy method

standardized. The curves are represented as a standard B-spline form and interpolate the appointed points. Control points of the new B-spline can also be obtained by solving linear equations, which decreases much repeated computation. At last, some computing examples are presented and compared with other methods. Our method can be generalized to extension of B-spline surfaces.

References

Do Carmo, M.P., 1976. Differential Geometry of Curves and Surfaces. Pearson Education.

Farin, G., 1997. Curves and Surfaces for Computer Aided Geometric Design (4th Ed.). Academic Press, San Diego.

Farin, G., Sapidis, N., 1989. Curvature and the fairness of curves and surfaces. *Computer Graphics and Applications*, *IEEE*, **9**(2):52-57. [doi:10.1109/38.19051]

Hoschek, J., Lasser, D., 1993. Fundamentals of Computer Aided Geometric Design. Wellesley, MA.

Hu, S.M., Tai, C.L., Zhang, S.H., 2002. An extension algorithm for B-splines by curve unclamping. *Computer-Aided Design*, **34**(5):415-419. [doi:10.1016/S0010-4485(01)00108-7]

Piegl, L., Tiller, W., 1997. The NURBS Book. Springer-Verlag, Bernlin, Germany.

Shetty, S., White, P.R., 1991. Curvature-continuous extensions for rational B-spline curves and surfaces. *Computer-Aided Design*, **23**(7):484-491. [doi:10.1016/0010-4485(91)90046-Y]

Shi, F.Z., 2001. Computer Aided Geometric Design and NURBS. Higher Education Press, Beijing, China (in Chinese).

Su, B.Q., Liu, D.Y., 1982. Computational Geometry. Shanghai Science Press, Shanghai, China (in Chinese).

Tai, C.L., Hu, S.M., Huang, Q.X., 2003. Approximate merging of B-spline curves via knot adjustment and constrained optimization. *Computer-Aided Design*, **35**(10):893-899. [doi:10.1016/S0010-4485(02)00176-8]

Tiller, W., 1992. Knot-removal algorithms for NURBS curves and surfaces. *Computer-Aided Design*, **24**(8):445-453. [doi:10.1016/0010-4485(92)90012-Y]