On a special interpolation problem of functional B-spline curves

Miklós Hoffmann (Eger, Hungary)

Dedicated to the memory of Professor Péter Kiss

Abstract. A multistep interpolation method is presented in this paper by which one can compute a planar functional B-spline curve from its gradient function with special endconditions. The method is applied in turbine blade section curve design where the curve consists of two interpolating B-spline arcs connecting two predefined circular arcs.

AMS Classification Number: 68U05

1. Introduction and problem statement

Interpolation or approximation of a set of data by B-spline curves is a well-studied area of computer aided design and manufacture. One can find numerous methods solving the problem for different types of data, see e.g. [3] and references therein. All of these classical methods, however, require direct geometrical data (points and/or tangent lines) of the curve.

In the problem presented here, the curve has to be created from the angles of its tangent lines, while the exact position of the curve is determined by endpoints and tangential information in them. A special, multistep method will be described in the following section to solve this problem. Application of this method for turbine blade section curve design can be found in Section 3. Finally conclusion and directions of future research close the paper.

To clarify the notations here we present the definition of the B-spline curve.

Definition. The curve \( s(u) \) defined by

\[
s(u) = \sum_{i=0}^{n} N^k_i(u) d_i, \quad u \in [u_{k-1}, u_{n+1}]
\]

is called B-spline curve of order \( k \) (degree \( k-1 \)), where \( N^k_i(u) \) is the \( i^{th} \) normalized B-spline basis function, for the evaluation of which the knots \( u_0, u_1, \ldots, u_{n+k} \) are

This research was sponsored by the Hungarian Scientific Research Foundation (OTKA No. F032679), the Hungarian Ministry of Education (FKFP No.0027/2001) and the European Research Training Network MINGLE (HPRN-1999-00117).
necessary. The points $d_i$ are called control points, while the polygon formed by
these points is called control polygon.

The turbine blade section curve (or simply profile curve) consists of two
circular arcs and two B-spline arcs connecting them. In profile curve design the
connecting B-spline curves are smooth, almost linear pieces along the $x$
direction, hence throughout this paper one can consider them as functions of $x$, which is
necessarily not the general case.

2. Multistep interpolation

Let a continuously differentiable real function $\alpha = f(x), x \in [a, b]$ be given.
We would like to create an arc (function) over $[a, b]$ with the following property:
at every $x_0 \in [a, b]$ the angle of the $x$ axis and the tangent line of the arc at this
point is $\alpha_0 = f(x_0)$. Thus the function $\alpha = f(x)$ can be considered as the gradient
function of the arc we are looking for. Moreover, the arcs have to be defined as a
B-spline curves, because this type of curves is the standard description method in
CAGD. Thus let this arc be denoted by $s(u), u \in [a, b]$.

At first a sample set of pairs $(x_i, \alpha_i), i = 1, ..., n$ is chosen, where $n$ depends
on the desired error bound. The $x_i$'s can be uniformly distributed over $[a, b]$. We
will create a cubic B-spline curve $s(u)$ in a way that at every point of the curve
coresponding to the coordinate values $x_i$ the angle between the tangent line of
the curve and the $x$ axis is $\alpha_i$. Denote these corresponding points of the curve by
der $x_i, y_i$). Thus

$$s(u_i) = q_i(x_i, y_i), \quad i = 1, ..., n$$

where the parameters $u_i$ and the coordinates $y_i$ are the unknowns. Let $e_i(\cos \alpha_i,
\sin \alpha_i)$ denote the unit vector at the direction of the $i^{th}$ tangent line. This yields
the equations

$$\hat{s}(u_i) = \lambda_i e_i, \quad i = 1, ..., n$$

where the values $\lambda_i$ are unknowns.

Denote the coordinates of the future control points $d_j$ by $(x^d_j, y^d_j), j = 1, ..., n$
and the coordinate functions of the B-spline curve by $s(u) = (x^s(u), y^s(u))$. Note,
that the number of control points is the same as the number of samples. Thus from
(1) and (2) we obtain the following system of equations:

$$x^s(u_i) = x_i,$n
$$\hat{x}^s(u_i) = \lambda_i \cos \alpha_i, \quad i = 1, ..., n.$$
\[
\hat{y}^s(u_i) = \lambda_i \sin \alpha_i,
\]
By the definition of the B-spline curve this system can be written as

\[ \sum_{j=0}^{n} N_j^4(u_i)x_j^d = x_i, \]
\[ \sum_{j=0}^{n} N_j^4(u_i)x_j^d = \lambda_i \cos \alpha_i, \quad i = 1, \ldots, n. \]
\[ \sum_{j=0}^{n} N_j^4(u_i)y_j^d = \lambda_i \sin \alpha_i. \]

(3)

Here each row represents \( n \) equations and all subsystems are linear. If we fix the parameter values \( u_i \) then we can compute the B-spline basis functions \( N_j^4(u_i) \) and their derivatives \( \dot{N}_j^4(u_i) \). There are several methods for the choice of \( u_i \) (uniform, cumulative chord length, centripetal model, c.f. [1]), but our future curve is quite simple, smooth, and the samples \( x_i \) are uniformly distributed, thus we can apply the uniform model as

\[ u_i = \frac{i - 1}{n - 1}. \]

Now the first subsystem of (3) —containing the first \( n \) equations— can be solved for \( x_j^d \), then the second subsystem for \( \lambda_i \), finally the third \( n \) equations for \( y_j^d \). The first and the third systems of equations have a banded matrix where the bandwidth is 4, while the solution of the second one is even more straightforward, containing one unknown per equation. After this process we will obtain the control points \( d_j \) \((x_j^d, y_j^d)\), with the help of which one can compute the desired B-spline curve. In case of non-uniform curve the knot vector can be the same as the sequence of the parameter values.

**2. An application: turbine blade section curve design**

The design process of turbine blades generally contains several steps. At first designers create planar curves which will serve as section curves of the future surface. A typical section curve can be seen in Fig.1.
Figure 1

Focusing on this problem a designer may prefer the following scenario: he can define two circular arcs by their centres, radii and endpoints. The system connects them by two B-spline arcs which can be modified by direct manipulation of their predefined gradient functions. The gradient functions, of course, always satisfy the endconditions defined by the circular arcs.

Figure 2

These two B-spline arcs can be calculated by the interpolation methods described in the previous section. To estimate the correctness of this technique we considered an arc of an existing section curve and picked 350 points of them as samples.

Figure 3

Creating the new B-spline arc two types of error measured at these sample points: the distance of the two curve and the difference between the tangential angles of the two curve. The distance is measured by the following way: at each sample point $p$ of the original curve a perpendicular straight line is dropped. This line intersects the computed B-spline curve at $\hat{p}$ and the error at this point is defined as the length of $\hat{pp}$. The graph of this error function (connecting the discrete points for better visualization) can be seen in Fig.2.
The other type of error is measured by the difference between the angles of the tangent lines of the original and the computed curve at the sample points. This graph can be seen in Fig. 3.

4. Conclusion

A special interpolation method has been presented in this paper with the help of which one can create and modify a planar B-spline curve based on its gradient function. Using this method a turbine blade section curve design technique has been developed: two circular arcs are defined which are connected by B-spline arcs computed by the new method.

References


Miklós Hoffmann
Department of Mathematics
Károly Eszterházy College
H-3301 Eger, P. O. Box 43.
Hungary
E-mail: hofi@ektf.hu