ELSEVIER

Computer Aided Geometric Design 15 (1998) 869–877

COMPUTER AIDED GEOMETRIC DESIGN

Global reparametrization for curve approximation

Thomas Speer [a,1], Markus Kuppe [b,2], Josef Hoschek [c,*]

[a] *Motoren- und Turbinen Union, München GmbH, Postfach 50 06 40, D-80995 München, Germany*

[b] *Dev. Co 2, SAP AG, Neurottstrasse 16, D-69190 Walldorf, Germany*

[c] *Department of Mathematics, Darmstadt University of Technology, Schlossgartenstrasse 7, D-64289 Darmstadt, Germany*

Received November 1997; revised March 1998

Abstract

For approximation of a set of points $\mathbf{P}_j \in \mathbb{R}^d$ by a parametric curve $\mathbf{X}(t)$ the choice of parameters t_j is essential. None of the parametrization strategies is optimal. To obtain good approximation results, reparametrization (parameter correction) of the points is necessary. In general, these reparametrization methods work only locally. We present a global reparametrization method which leads to dramatically better results. © 1998 Elsevier Science B.V.

Dedicated to Professor Roland Bulirsch on the occasion of his 65th birthday

1. Formulation of the problem

The given (ordered) points $\mathbf{P}_j \in \mathbb{R}^d$ $(j = 1, \ldots, N)$ are to be approximated by the parametrized curve $\mathbf{X}(t)$. Here, we will use B-spline curves

$$\mathbf{X}(t) = \sum_{i=1}^{n} \mathbf{d}_i N_{ik}(t) \tag{1}$$

with $\mathbf{d}_i \in \mathbb{R}^d$ as (unknown) control points and N_{ik} as B-spline basis functions of order k (degree $k-1$). The B-spline basis functions may be defined over the knot sequence $(u_1, \ldots, u_n)$, where multiple knots are allowed.

We want to minimize the sum of squared distances $\mathbf{r}_j = \mathbf{P}_j - \mathbf{X}(t_j) \in \mathbb{R}^d$, which leads to the *nonlinear* least squares problem

* Corresponding author. E-mail: hoschek@mathematik.tu-darmstadt.de.

[1] E-mail: thomas.speer@muc.mtu.dasa.de.

[2] E-mail: markus.kuppe@sap-ag.de.

PII S0167-8396(98)00024-7

$$\min \sum_{j=1}^{N} \left(\mathbf{P}_j - \sum_{i=1}^{n} \mathbf{d}_i N_{ik}(t_j) \right)^2 \quad \text{with respect to } \mathbf{d}_i \in \mathbb{R}^d,\ i = 1, \ldots, n,$$
$$\text{and } t_j \in [u_1, u_n],\ j = 1, \ldots, N \text{ (with } n \ll N\text{)}. \quad (2)$$

2. Alternating method

The solution of problem (2) requires the search for those parameters t_j that minimize the curve's distance to the given points even though we are not primarily interested in them but only in (the shape of) the curve itself.

As within the framework of optimization theory problem (2) fits into the class of separable problems, several authors (Hoschek et al., 1993; Rogers et al., 1989) use the following approach which breaks down the original problem into smaller problems that can be implemented easily in an alternating manner:

Choosing a suitable parametrization method (Hoschek et al., 1993, p. 201 f) we obtain the parameter values $t_j \in [u_1, u_n]$ for the points $\mathbf{P}_j$. Holding fixed the parameters t_j, the coefficients (x_i, y_i, z_i) of the control points $\mathbf{d}_i$ remain as unknowns which enter *linearly* into problem (2). We solve the linear least squares problem (2) separately for each component, i.e., for n unknowns.

Because of the arbitrary parametrization the errors $\mathbf{r}_j$ do not necessarily give the shortest distances of the given points $\mathbf{P}_j$ to the approximation curve. To minimize the distances, a lot of iterative methods have been proposed (Hoschek et al., 1993, p. 201 f) to move the parameter values t_j of $\mathbf{P}_j$ to values t_j^* such that the vectors $\mathbf{r}_i^* = \mathbf{r}_i(t_i^*)$ are orthogonal to the approximating curve.

All these methods for reparametrization are Newton-like algorithms for each of the N one-dimensional minimization problems

$$\min \left\| \mathbf{P}_j - \mathbf{X}(t) \right\| \text{ w.r.t. } t \quad (3)$$

and work iteratively in a purely local manner.

As this approach alternates between the solution of linear least squares problems and one-dimensional minimization, we call it "the alternating method".

3. Global method

The first author of this communication had the idea to use a global method instead of this alternating process and solve (2) directly with Moré's implementation of Powell's variant of the Levenberg–Marquardt method (Moré et al., 1980) which is incorporated within the MINPACK software package (available from netlib). It uses the linearization of the global problem (2) with respect to all $N + dn$ unknowns. (A thorough modern analysis of the Levenberg–Marquardt method is given by (Spellucci, 1993).) It is a trust region variant of Gauss–Newton methods, and as other least square methods which update both sets of unknowns (parameters t_j and control points $\mathbf{d}_i$) simultaneously, this method requires the

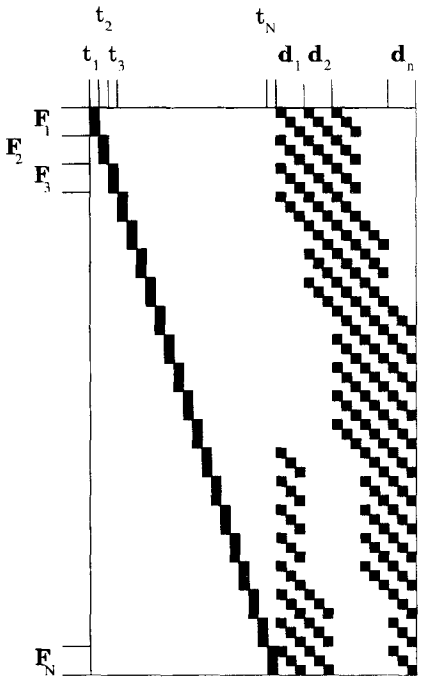

Fig. 1. The sparsity pattern of a typical Jacobian matrix ($N = 20$, $n = 5$, $k = 3$, 3-dimensional).

repeated solution of linear least squares problems

$$\min \|\mathbf{J}\mathbf{p} - \mathbf{F}\| \tag{4}$$

to compute the joint correction $\mathbf{p} = (\Delta t, \Delta \mathbf{d})$ (Kuppe, 1997).

The Jacobian matrix $\mathbf{J}$ consists of the partial derivatives of the total residual vector $\mathbf{F} = (r_1^{\mathrm{T}}, \ldots, r_N^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{dN}$ with respect to all unknowns and has the dimension $dN \times (N + dn)$, where d denotes the dimension of the underlying physical space, i.e., 2 or 3. A typical two-dimensional turbine blade profile approximation problem involves $N = 140$ data points and $n = 12$ control points which result in a $2N \times (N + 2n) = 280 \times 164 -$ matrix.

As many linear least squares methods the Levenberg–Marquardt algorithm uses a QR decomposition of the system matrix: We look for an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{dN \times dN}$ and an upper triangular matrix $\mathbf{R}$ such that

$$\mathbf{J} = \mathbf{Q}\begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}.$$

From $\mathbf{Q}$ and $\mathbf{R}$ the solution of (4) can be calculated easily. (Concerning the QR decomposition and its use in least squares problems see (Golub and van Loan, 1989; Björck, 1996) or any other textbook on numerical linear algebra.)

Because of its magnitude, the QR decomposition of the matrix $\mathbf{J}$ cannot be achieved efficiently if we handle it as a dense matrix and ignore its sparsity structure.

Fig. 1 shows a typical sparsity pattern of $\mathbf{J}$. Exploiting this regular structure by a corresponding modification of the QR decomposition algorithm reduces the operation

count by at least two orders of magnitude. With the aid of an appropriate textbook, an available QR decomposition code can be modified in a straight forward manner. The sparsity pattern of $\mathbf{J}$ and a suitable sparse algorithm reflect the inherent separability of the underlying problem (2) which is utilized for the purpose of the efficient and (up to roundoff) exact solution of the linear problem (4) only; exploiting this sparsity has no other influence upon the overall solution process.

We used the Levenberg–Marquardt method as a method at hand which has stood the test of time, but the results might carry over to other nonlinear least squares methods if the sparsity pattern of $\mathbf{J}$ is taken into account within the solution process of their linear subproblems.

4. Numerical examples and comparisons

4.1. For demonstration of the global method's power we will approximate 140 measurement points taken from a cross section of a turbine blade (see Fig. 2). The resulting curve should be a closed quintic B-spline with 4 segments, and it should be C^2-continuous so it can be used later in a simulation program to test its aerodynamics. Therefore we choose triple knots.

We will start the approximation with a chordal parametrization of the points assuming an interval of length 1. As the curve shall be closed, the parametrization is periodic.

Experience shows that the knots should be situated close to the critical areas of the data points. However, if the point with the smallest x coordinate has the parameter value t_v and

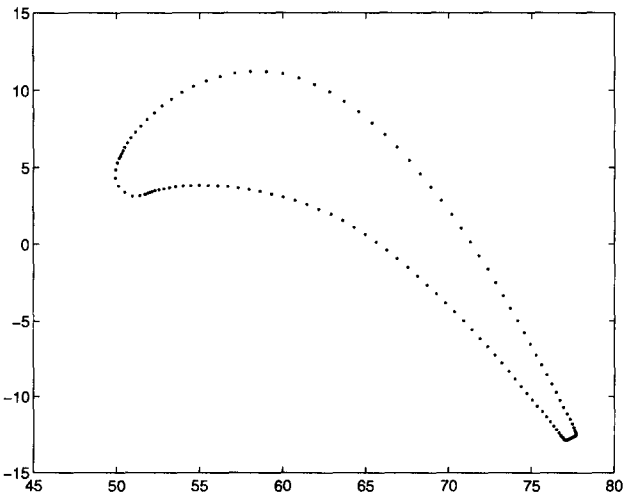

Fig. 2. Given points of a turbine blade's cross section.

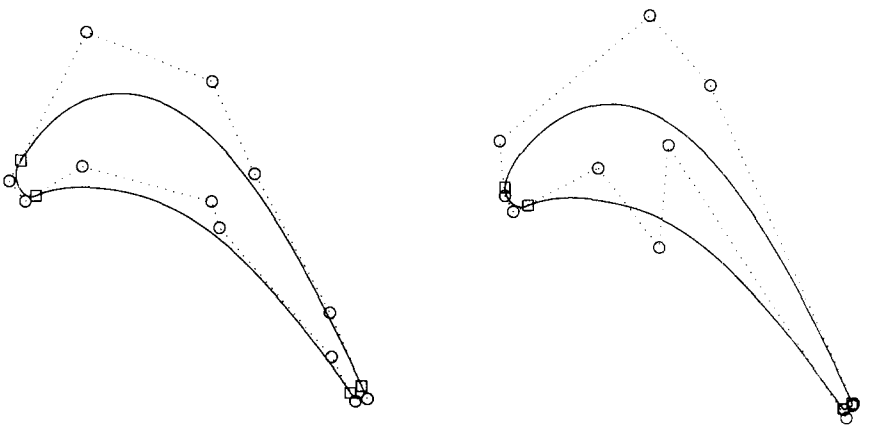

Fig. 3. Approximation curves of the point set from Fig. 2 with control polygons and knots (marked by boxes). The right hand side shows the result of the global method, the left one contains the approximation with the alternating method.

the point with the largest x coordinate has the parameter value t_h, we choose as a (periodic) knot vector

$$\begin{aligned} U = (\, & t_h - \varepsilon, && t_h - \varepsilon, && t_h - \varepsilon, \\ & t_h + \varepsilon, && t_h + \varepsilon, && t_h + \varepsilon, \\ & t_v - \varepsilon, && t_v - \varepsilon, && t_v - \varepsilon, \\ & t_v + \varepsilon, && t_v + \varepsilon, && t_v + \varepsilon, \\ & t_h - \varepsilon + 1\,). \end{aligned}$$

For the following example we arbitrarily choose $\varepsilon = 0.02$. Of course this choice influences the result of the approximation, but the efficiency of both methods remains comparable for other values of ε. Fig. 3 gives the approximation curve using the alternating method (left hand side) and using the global method (right hand side).

One can observe that the knots (marked by boxes) are moved in the neighbourhood of the critical area of the given set of points. The reason of these motions are the larger changes of the parametrization during the global process. Thus, the distribution of knots is better adapted to the problem. Most obviously the difference in approximation quality can be seen by zooming the critical region (see Fig. 4).

As a maximal distance error $\Delta_{\max} = \max |\mathbf{r}_j|$ we obtain for the global method $\Delta_{\max} = 0.025$ mm and for the alternating method $\Delta_{\max} = 0.075$ mm (CPU time 3 seconds, extension of the object 26×24 mm).

The ratio of the root mean squared errors (measured in mm) versus the CPU time (measured in seconds) is shown in Fig. 5. After 3 seconds of CPU time we achieve a root mean squared error $5.15 \cdot 10^{-2}$ mm by using the alternating method and $2.70 \cdot 10^{-2}$ mm by the global method. To reach a comparably small error with the alternating method it takes about 858 seconds.

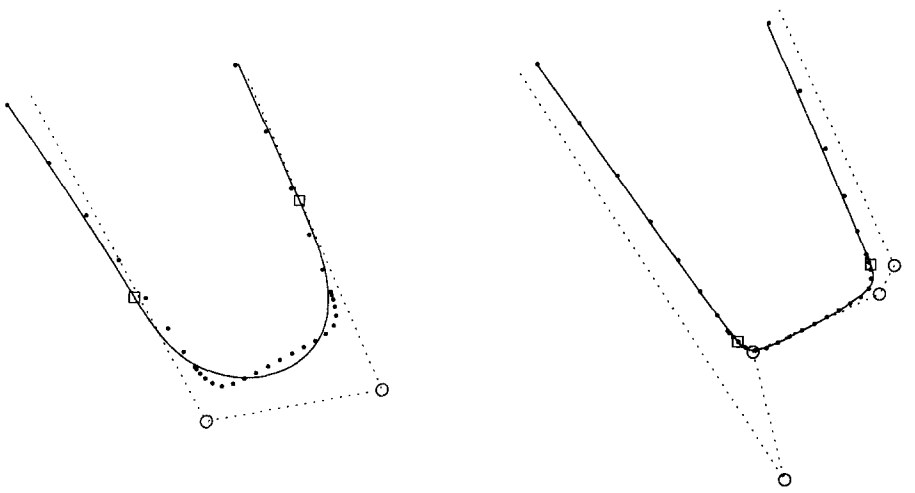

Fig. 4. Zooming of the critical region in Fig. 3 (left hand side alternating method, right hand side global method) with the data points and the approximation curves in that region with the control polygons and knots (boxes).

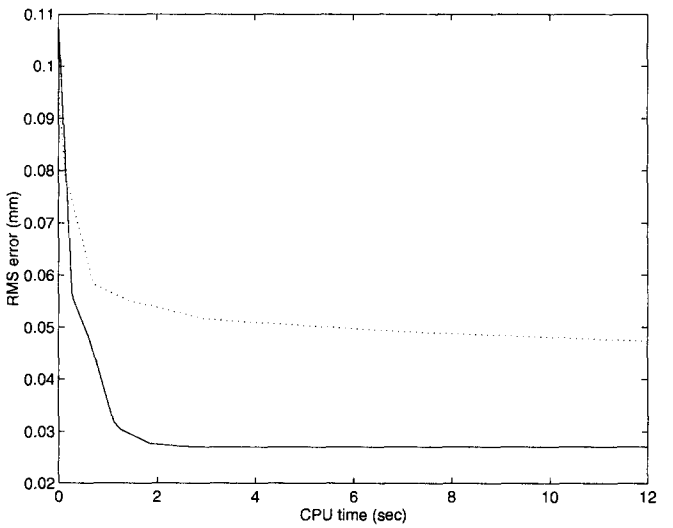

Fig. 5. Root mean squared errors versus the CPU time for the alternating method (dashed) and for the global method.

4.2. Now we will consider a second example: we take 200 points from a punch line (space curve) of a binder for an engine bonnet of a car body. We choose as an approximation curve a C^2-continuous cubic B-spline with 16 segments and (this time) a uniform knot vector with single knots. Again we start with chordal parametrization. Without any reparametrization we obtain an approximation curve as plotted in Fig. 6.

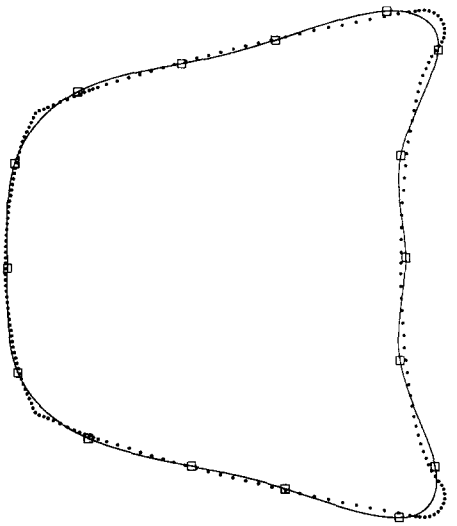

Fig. 6. Given set of points of a punch line (space curve) and cubic approximation curve without reparametrization (knots are marked by boxes).

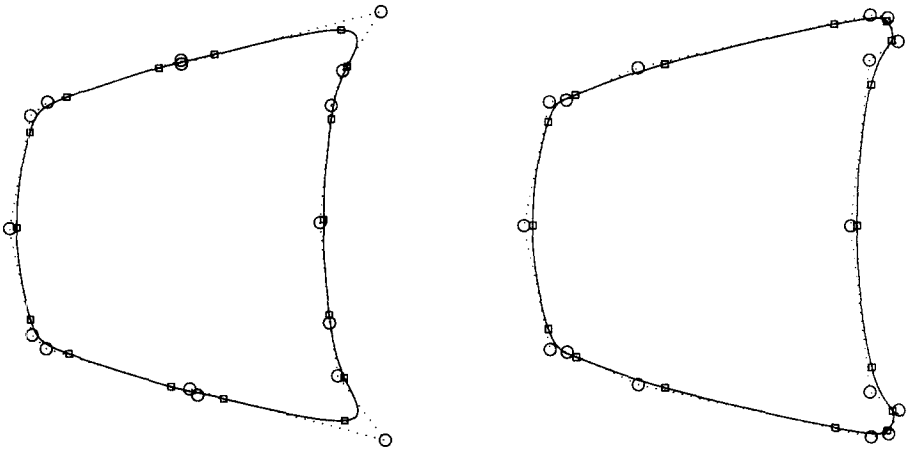

Fig. 7. Approximation curve of the point set in Fig. 6 with the control polygon and knots (marked by boxes) after reparametrization. (Left hand side alternating method, right hand side global method.)

Fig. 7 contains the approximation curves and their control polygons after reparametrization: for the figure at the left hand side the alternating method is used, for the right hand sided figure the global method. The knots are marked by boxes. One can observe again that by using the global method the knots are moved in the neighbourhood of the critical regions of the curve (see also Fig. 8).

With the alternating method we obtain a maximum error distance of $\Delta_{\max} = 5.92$ mm and the root mean squared error 0.386 mm (after 10 seconds of CPU time), whilst

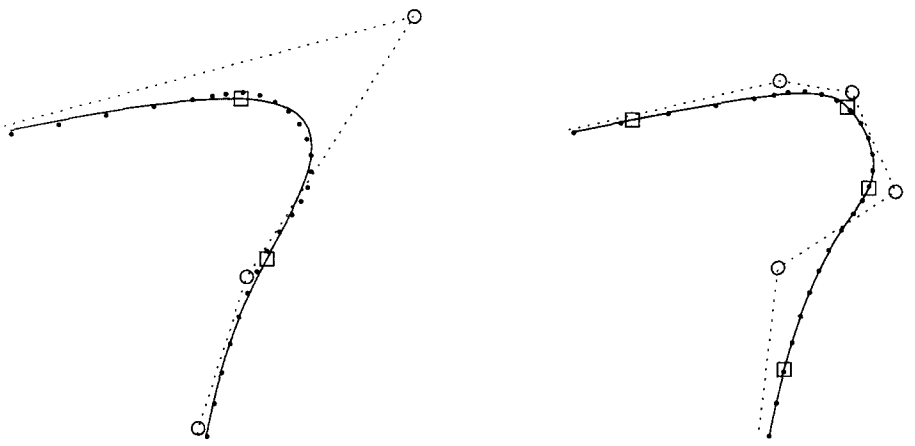

Fig. 8. Shows a zooming of one of the critical regions of Fig. 7 (left hand side alternating method, right hand side global method) with the data points and the approximation curves in that region with the control polygons and knots (boxes).

the global method leads to a maximum distance error of $\Delta_{\max} = 3.53$ mm and a root mean squared error 0.258 mm (after 7 seconds CPU-time). The curve covers an area of 1440×1645 mm. The root mean squared errors of the example in Fig. 7 have the same behaviour as demonstrated in Fig. 5.

5. Final remarks

Updating all unknowns simultaneously as in Gauss–Newton methods asymptotically leads to quadratic convergence rates (for the zero residuum case), while alternately minimizing $F(t, \mathbf{d})$ over $\mathbf{d}$ and the components of t always converges only linearly (Björck, 1996, p. 351). Neglecting the interaction of t and $\mathbf{d}$ slows down convergence and even leads to an oscillating behaviour that prevents the process from meeting the desired accuracy in a reasonable time.

Björck gives a variable projection algorithm for separable problems like the present one that reaches the same asymptotic convergence rate as Gauss–Newton algorithms by an alternative sequence of two different linear least squares problems. This could be an alternative to our global method that can be implemented more easily and still yields its fast convergence. It is anticipated that upon surface fitting problems the method reveals an even greater beneficial impact.

References

Björck, Å. (1996), *Numerical Methods for Least Squares Problems*, SIAM.

Golub, G.H. and van Loan, D.F. (1989), *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore.

Hoschek, J. and Lasser, D. (1993), *Fundamentals of Computer Aided Geometric Design*, A.K. Peters.

Kuppe, M. (1997), Zwei Methoden zur Approximation mit B-splines und ein Ansatz zur Übertragung auf geometrische Splines, Master Thesis in Mathematics, Department of Mathematics, University of Technology Darmstadt.

Moré, J., Garbow, B.S. and Hillstrom, K.E. (1980), *Users' Guide for MINPACK-1*, Argonne Nat. Lab.

Rogers, D.F. and Fog, N.G. (1989), Constrained B-spline curve and surface fitting, Computer-Aided Design 21, 641–648.

Spellucci, P. (1993), *Numerische Verfahren der nichtlinearen Optimierung*, Birkhäuser, Basel.