ELSEVIER

0010-4485(95)00044-5

Computer-Aided Design, Vol. 28, No. 1, pp. 67-72, 1996

0010-4485/96 $15.00 + 0.00

Technical note

Partial derivatives of Bézier surfaces

Klaus Spitzmüller

In manufacturing Bézier surfaces, it is often useful to know some of their geometric invariants. In planning milling pathes for example, the curvature should be known in some points of the Bézier surface. From differential geometry it follows, that the partial derivatives up to second order must be computed. How this computation can be done effectively, is the content of this article. Using the algorithm of de Casteljau in a different way, the calculation of a point with his partial derivatives up to second order can be accomplished with little more computational cost than the calculation of the point only.

Keywords: Bézier curves, Bézier patches, de Casteljau algorithm, derivatives

INTRODUCTION

In CAD systems freeform curves and freeform surfaces are important tools[1, 2]. In most cases, a free form curve c is defined by control points $\mathbf{B}_i$ and a set of suitable blending functions f_i $(i = 1, \ldots, N)$ in the form

$$c: \vec{x}(u) = \sum_{i=1}^{N} \vec{\mathbf{b}}_i f_i(u), \; u \in [0,1]$$

where $\vec{\mathbf{b}}_i$ denote the geometric vectors of the control points $\mathbf{B}_i$. If on the curve c a point $X(u_0)$ — its geometric vector $\vec{x}(u_0)$ — has to be determined, one can compute $f_i(u_0)$ for $i = 1, \ldots, N$ and then solve this equation. But generally, this is not the best way. For numerical reasons, in most cases subdivision algorithms are more effective. These algorithms determine the point $X(u_0)$ by iteratively subdividing the polygon formed by the control points. If c is a Bézier curve, the subdividing algorithm is the well-known algorithm of de Casteljau[3].

A similar approach is possible, if points of a freeform surface have to be determined. Generally a freeform surface Φ is given in tensor product form. In this case, a point on Φ can be determined by subdivision too. So, points of a Bézier surface can be computed by the repeated use of the algorithm of de Casteljau[4].

As an additional advantage, the algorithm of de Casteljau gives the possibility to compute the derivatives at a point of a Bézier curve without additional costs[3, 4]. Analogously, it is important to know a simple way for the determination of the partial derivatives of a Bézier surface. This holds, as most of the geometrical invariants of a surface which are essential for the shape of such a surface, depend on the partial derivatives, usually up to second order. An example may illustrate the importance of this question. Milling a surface by a ball end cutter with radius r, the maximum curvature in concave regions of the surface is limited by $1/r$ to avoid surface gouging[5]. So, it is helpful to know the principal curvatures of the surface which has to be milled. To get these curvatures, the partial derivatives of the surface up to second order have to be computed[6]. Some other examples are treated in the final section of this paper.

Using the algorithm of de Casteljau, the partial derivatives up to second order of a Bézier surface can be determined very effectively. A suitable algorithm is presented in this paper, which is organized as follows. After an introductory section about Bézier curves and surfaces, the calculation of partial derivatives of a Bézier surface is explained. Next, the algorithm based on these results is given. Finally, some applications of the presented algorithm are listed.

BÉZIER CURVES AND SURFACES

With $\mathbf{B}_0, \ldots, \mathbf{B}_N \in \mathbf{E}^3$ and the Bernstein polynomials B_i^N, $i = 0, \ldots, N$ the curve

$$c: \vec{\mathbf{b}}_0^N := \vec{x}(u) = \sum_{i=0}^{N} \vec{\mathbf{b}}_i B_i^N(u), \quad u \in [0, 1] \tag{1}$$

is called the *Bézier curve* (of degree N) belonging to the Bézier points $\mathbf{B}_0, \ldots, \mathbf{B}_N$. Another very elegant representation, which is due to Hosaka and Kimura[7], makes use of the shift operator E, which is defined by $E(b_i) = b_{i+1}$, $i = 0, \ldots, N-1$.

Mathematisches Institut II, Universität Karlsruhe, D-76128 Karlsruhe, Germany
Paper received: 5 May 1994. Revised: 1 May 1995

From the binomial theorem it follows that the Bézier curve, Equation 1, is given by

$$c: \vec{\mathbf{b}}_0^N(u) = \vec{\mathbf{x}}(u) = (1-u+uE)^N \vec{\mathbf{b}}_0, \quad u \in [0, 1] \tag{2}$$

With $\vec{\mathbf{b}}_i^k := (1-u+uE)^k \vec{\mathbf{b}}_i$, $0 \leq i+k \leq N$ and the recursion $\vec{\mathbf{b}}_i^k = (1-u)\,\vec{\mathbf{b}}_i^{k-1} + u\vec{\mathbf{b}}_{i+1}^{k-1}$ one gets the algorithm of de Casteljau:

$$\begin{array}{lllll}
\vec{\mathbf{b}}_0 & =: & \vec{\mathbf{b}}_0^0 & & \\
\vec{\mathbf{b}}_1 & =: & \vec{\mathbf{b}}_1^0 & \searrow\!\!\to \vec{\mathbf{b}}_0^1 & \\
\vdots & & \vdots & \vdots & \\
\vec{\mathbf{b}}_{N-1} & =: & \vec{\mathbf{b}}_{N-1}^0 \to \vec{\mathbf{b}}_{N-2}^1 \to \vec{\mathbf{b}}_{N-3}^2 \ldots \vec{\mathbf{b}}_0^{N-1} & & \\
\vec{\mathbf{b}}_N & =: & \vec{\mathbf{b}}_N^0 \to \vec{\mathbf{b}}_{N-1}^1 \to \vec{\mathbf{b}}_{N-2}^2 \ldots \vec{\mathbf{b}}_1^{N-1} \to \vec{\mathbf{b}}_0^N & &
\end{array}$$

The vector $\vec{\mathbf{b}}_0^N$, which is constructed in this way, is the geometric vector of the point on the Bézier curve, which belongs to the parameter u.

The representation (2) of a Bézier curve can be used to determine its derivatives.

Lemma 1

For the kth derivative of a Bézier curve of degree N, $0 \leq k \leq N$, it holds

$$\vec{\mathbf{x}}^{(k)}(u) = \frac{N!}{(N-k)!} \cdot (1-u+uE)^{N-k}(E-1)^k \vec{\mathbf{b}}_0 \tag{3}$$

It follows for the first and second order derivatives

$$\vec{\mathbf{x}}^{(1)}(u) = N \cdot \left(\vec{\mathbf{b}}_1^{N-1} - \vec{\mathbf{b}}_0^{N-1}\right)$$

$$\vec{\mathbf{x}}^{(2)}(u) = N \cdot (N-1) \cdot \left(\vec{\mathbf{b}}_2^{N-2} - 2\vec{\mathbf{b}}_1^{N-2} + \vec{\mathbf{b}}_0^{N-2}\right)$$

Because the vectors $\vec{\mathbf{b}}_i^k = (1-u+uE)^k \vec{\mathbf{b}}_i$ can be found in the table of de Casteljau, the derivatives of a Bézier curve can be computed from this scheme. The first derivative is N times the difference of the vectors in the last but one column. The second derivative is a linear combination of vectors in the column before. Generally the vectors in the $(N+1-k)$th column of the table of de Casteljau are needed to calculate the kth derivatives of a Bézier curve.

With $\mathbf{B}_{ij} \in \mathbf{E}^3$ $(i = 0, \ldots, M;\ j = 0, \ldots, N)$ the surface

$$\Phi: \vec{\mathbf{b}}_{0,0}^{M,N} := \vec{\mathbf{x}}(u, v) = \sum_{i=0}^{M} \sum_{j=0}^{N} \vec{\mathbf{b}}_{ij} B_i^M(u) B_j^N(v),$$

$$(u, v) \in [0, 1] \times [0, 1] \tag{4}$$

is called the *Bézier surface* (of degree (M, N)) belonging to the Bézier points $\mathbf{B}_{ij}$.

An example of a Bézier surface is shown in *Figure 1*. For better visibility the net of control points has been lifted. Some obvious properties of Bézier surfaces can be found in the following list:

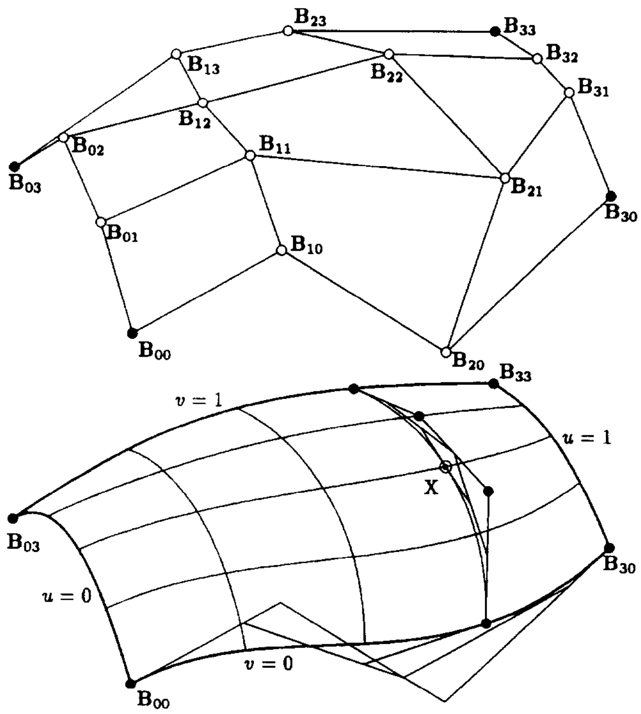

Figure 1 Bézier surface and the algorithm of de Casteljau (1st way)

(a) $\vec{\mathbf{x}}(0, 0) = \vec{\mathbf{b}}_{00}$, $\vec{\mathbf{x}}(0, 1) = \vec{\mathbf{b}}_{0N}$, $\vec{\mathbf{x}}(1, 0) = \vec{\mathbf{b}}_{M0}$, $\vec{\mathbf{x}}(1, 1) = \vec{\mathbf{b}}_{MN}$,

(b) The u-line $v = v_0$ is a Bézier curve belonging to the Bézier points $\mathbf{B}_i(v_0)$ with the geometric vectors

$$\vec{\mathbf{b}}_i(v_0) := \sum_{j=0}^{N} \vec{\mathbf{b}}_{ij} B_j^N(v_0), \quad i = 0, \ldots, M$$

(c) The v-line $u = u_0$ is a Bézier curve belonging to the Bézier points $\mathbf{B}_j(u_0)$ with the geometric vectors

$$\vec{\mathbf{b}}_j(u_0) := \sum_{i=0}^{M} \vec{\mathbf{b}}_{ij} B_i^M(u_0), \quad j = 0, \ldots, N$$

In Figure 1 the v-line $u = 0.75$ and its control points — marked by • — have been drawn.

There also exist shift operators E and F for Bézier surfaces, which are defined by $E(\vec{\mathbf{b}}_{i,j}) = \vec{\mathbf{b}}_{i+1,j}$ and $F(\vec{\mathbf{b}}_{ij}) = \vec{\mathbf{b}}_{i,j+1}$. They are commutative: $E(F(\vec{\mathbf{b}}_{i,j})) = F(E(\vec{\mathbf{b}}_{ij}))$. The Bézier surface can be represented by these operators as

$$\vec{\mathbf{x}}(u, v) = (1-u+uE)^M (1-v+vF)^N \vec{\mathbf{b}}_{00}$$

$$= (1-v+vF)^N (1-u+uE)^M \vec{\mathbf{b}}_{00}$$

It follows

$$\vec{\mathbf{b}}_{0,0}^{MN}(u, v) = (1-u+uE)(1-u+uE)^{M-1}$$

$$(1-v+vF)^N \vec{\mathbf{b}}_{00}$$

$$= (1-u)\vec{\mathbf{b}}_{0,0}^{M-1,N} + u\vec{\mathbf{b}}_{1,0}^{M-1,N} \tag{5}$$

or

$$\vec{\mathbf{b}}_{0,0}^{M,N}(u,v) = (1-v+vF)(1-u+uE)^M$$
$$(1-v+vF)^{N-1}\vec{\mathbf{b}}_{00}$$
$$= (1-v)\vec{\mathbf{b}}_{0,0}^{M,N-1} + v\vec{\mathbf{b}}_{0,1}^{M,N-1} \quad (6)$$

Clearly the algorithm of de Casteljau can be used for the u- as well as for the v-direction. With the notation $\vec{\mathbf{b}}_{ij}^{00} := \vec{\mathbf{b}}_{ij}$ there are two obvious ways to compute a point $X(u_0, v_0)$ on the Bézier surface.

- 1st way:
  By $(M+1)$ applications of the algorithm of de Casteljau for the v-direction (operator F) the geometric vectors $\vec{\mathbf{b}}_{i,0}^{0,N}$ of the Bézier points of the u-line $v = v_0$ are calculated. In our notation this can be seen in an augmentation of the second superscript. In *Figure 1* this results in a row of $M+1 = 4$ points, which are marked by •. Then by a single application of the algorithm of de Casteljau for this u-line the geometric vector $\vec{\mathbf{b}}_{0,0}^{M,N}$ is computed (Operator E), which results in the change of the first superscript. The last step for this first way of computation is shown in Equation 5. In *Figure 1* this computation can be seen too. It results in a point X of the surface, which is marked as ⊕
- 2nd way:
  By $(N+1)$ applications of the algorithm of de Casteljau for the u-direction the geometric vectors $\vec{\mathbf{b}}_{0,j}^{M,0}$ of the Bézier points of the v-line $u = u_0$ are calculated. After that the geometric vector $\vec{\mathbf{b}}_{0,0}^{M,N}$ is computed by a single application of the algorithm of de Casteljau for the v-line. The last step for the second way of computation is shown in Equation 6.

Which of these two ways is more effective depends on the degree (M, N) of the Bézier surface. Applying the algorithm of de Casteljau on $(N+1)$ control points $1/2N(N+1)$ convex combinations are computed. So in the lst way $(M+1)$ applications of an $O(N^2)$-algorithm and one application of an $O(M^2)$-algorithm are necessary. All things considered $1/2((M+1)N((N+1)+M(M+1)) = 1/2(MN^2+M^2+MN+N^2+M+N)$ convex combinations have to be computed. Analogously, in the 2nd way $1/2((N+1)M(M+1)+N(N+1)) = 1/2(NM^2+M^2+MN+N^2+M+N)$ convex combinations have to be computed. Therefore the 1st way will in general need less computation time for $N<M$. In the case $N>M$, the 2nd should be faster.

PARTIAL DERIVATIVES OF BÉZIER SURFACES

As for Bézier curves the derivatives of a Bézier surface can be given by a formula.

Lemma 2

The partial derivatives of a Bézier surface are for $0 \le k \le M$ and $0 \le l \le N$ given by:

$$\frac{\partial^{k+l}\vec{\mathbf{x}}}{\partial u^k \partial v^l}(u,v) = \frac{M!}{(M-k)!}\cdot\frac{N!}{(N-l)!}\cdot(1-u+uE)^{M-k}(E-1)^k (1-v+vF)^{N-l}(F-1)^l\cdot\vec{b}_{00}$$
$$= \frac{M!}{(M-k)!}\cdot\frac{N!}{(N-l)!}\cdot(E-1)^k\cdot(F-1)^l\cdot b_{0,0}^{M-k,N-l}. \quad (7)$$

The 1st way to get a point $X(u_0, v_0)$ on the Bézier surface starts with the calculation of the u-line $v = v_0$ and computes finally the point $X(u_0, v = v_0)$ on this curve. Therefore the partial derivatives $\vec{\mathbf{x}}_u$, $\vec{\mathbf{x}}_{uu}$,... with respect to the first parameter u of the surface can be computed as tangent vectors to this u-line as described in the second section of this article. It is as easy to determine the partial derivatives with respect to the second parameter of the surface $\vec{\mathbf{x}}_v$, $\vec{\mathbf{x}}_{vv}$,..., if the 2nd way is used. But to get both sorts of derivatives, it should not be necessary to calculate a point on the Bézier surface twice. Mixed partial derivatives, $\vec{\mathbf{x}}_{uv}$ for example, cannot be computed in this way. So the two ways, to calculate points on Bézier surfaces, shall be modified. To hold this article clear, only the derivatives up to second order are computed. So our task is:

For (u_0, v_0) the point $X(u_0, v_0)$ and the first and second order derivatives in this point shall be computed:

$$\vec{\mathbf{x}} = \mathbf{b}_{0,0}^{M,N} \quad (8a)$$

$$\vec{\mathbf{x}}_u = M\cdot(E-1)\cdot\vec{\mathbf{b}}_{0,0}^{M-1,N} = M\cdot\left(\vec{\mathbf{b}}_{1,0}^{M-1,N} - \vec{\mathbf{b}}_{0,0}^{M-1,N}\right) \quad (8b)$$

$$\vec{\mathbf{x}}_v = N\cdot\left(\vec{\mathbf{b}}_{0,1}^{M,N-1} - \vec{\mathbf{b}}_{0,0}^{M,N-1}\right) \quad (8c)$$

$$\vec{\mathbf{x}}_{uu} = M(M-1)\cdot(E-1)^2\cdot\vec{\mathbf{b}}_{0,0}^{M-2,N} = M(M-1)\cdot\left(\vec{\mathbf{b}}_{2,0}^{M-2,N} - 2\vec{\mathbf{b}}_{1,0}^{M-2,N} + \vec{\mathbf{b}}_{0,0}^{M-2,N}\right) \quad (8d)$$

$$\vec{\mathbf{x}}_{uv} = MN\cdot(E-1)(F-1)\cdot\vec{\mathbf{b}}_{0,0}^{M-1,N-1} = MN\cdot\left(\vec{\mathbf{b}}_{1,1}^{M-1,N-1} - \vec{\mathbf{b}}_{1,0}^{M-1,N-1} - \vec{\mathbf{b}}_{0,1}^{M-1,N-1} + \vec{\mathbf{b}}_{0,0}^{M-1,N-1}\right) \quad (8e)$$

$$\vec{\mathbf{x}}_{vv} = N(N-1)\cdot\left(\vec{\mathbf{b}}_{0,2}^{M,N-2} - 2\vec{\mathbf{b}}_{0,1}^{M,N-2} + \vec{\mathbf{b}}_{0,0}^{M,N-2}\right) \quad (8f)$$

The vectors, which occur in these equations, will be found with the help of the algorithm of de Casteljau. To make this clear, we rewrite these, using the shift operators E and F.

$$\vec{\mathbf{x}} = (1-u+uE)^2(1-v+vF)^2 (1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00} \quad (9a)$$

$$\vec{\mathbf{x}}_u = M(E-1)(1-u+uE)^1(1-v+vF)^2 (1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00} \quad (9b)$$

$$\vec{\mathbf{x}}_v = N(1-u+uE)^2(F-1)(1-v+vF)^1$$
$$(1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{b}_{00} \qquad (9c)$$
$$\vec{\mathbf{x}}_{uu} = M(M-1)(E-1)^2(1-v+vF)^2$$
$$(1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00} \qquad (9d)$$
$$\vec{\mathbf{x}}_{uv} = MN(E-1)(1-u+uE)^1(F-1)(1-v+vF)^1$$
$$(1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00} \qquad (9e)$$
$$\vec{\mathbf{x}}_{vv} = N(N-1)(1-u+uE)^2(F-1)^2$$
$$(1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00} \qquad (9f)$$

These formulae show that the desired vectors have the common shape

$$G(1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00}$$

with different operators G. So the approach to compute the vectors effectively is clear. First the common factors

$$(1-u+uE)^{M-2}(1-v+vF)^{N-2}\cdot\vec{\mathbf{b}}_{00}$$

are computed. After that the vectors are computed by the operators G.

THE ALGORITHM

Starting point of all computation of a Bézier surface is the $(M+1)\times(N+1)$-matrix

$$\mathbf{B} = \begin{pmatrix} \vec{\mathbf{b}}_{00}\ldots\vec{\mathbf{b}}_{0j}\ldots\vec{\mathbf{b}}_{0N} \\ \vdots \quad \vdots \quad \vdots \\ \vec{\mathbf{b}}_{i0}\ldots\vec{\mathbf{b}}_{ij}\ldots\vec{\mathbf{b}}_{iN} \\ \vdots \quad \vdots \quad \vdots \\ \vec{\mathbf{b}}_{M0}\ldots\vec{\mathbf{b}}_{Mj}\ldots\vec{\mathbf{b}}_{MN} \end{pmatrix}$$

of geometric vectors of the Bézier points.

Step 1

Apply the algorithm of de Casteljau in v-direction (1st way) on the ith row of B for $i=0,\ldots,M$ and stop the computation in the last but two columns. For fixed i the vectors $\vec{\mathbf{b}}^{0,N-2}_{i,0}$, $\vec{\mathbf{b}}^{0,N-2}_{i,1}$ and $\vec{\mathbf{b}}^{0,N-2}_{i,2}$ are computed from $\vec{\mathbf{b}}^{0,0}_{i,0} := \vec{\mathbf{b}}_{i0},\ldots,\vec{\mathbf{b}}^{0,0}_{i,N} := \vec{\mathbf{b}}_{iN}$. Note the results in the ith row of a new $(M+1)\times 3$-matrix

$$\mathbf{B}' = \begin{pmatrix} \vec{\mathbf{b}}^{0,N-2}_{0,0} & \vec{\mathbf{b}}^{0,N-2}_{0,1} & \vec{\mathbf{b}}^{0,N-2}_{0,2} \\ \vdots & \vdots & \vdots \\ \vec{\mathbf{b}}^{0,N-2}_{i,0} & \vec{\mathbf{b}}^{0,N-2}_{i,1} & \vec{\mathbf{b}}^{0,N-2}_{i,2} \\ \vdots & \vdots & \vdots \\ \vec{\mathbf{b}}^{0,N-2}_{M,0} & \vec{\mathbf{b}}^{0,N-2}_{M,1} & \vec{\mathbf{b}}^{0,N-2}_{M,2} \end{pmatrix}$$

The result of the Step 1 can be seen in *Figure 2*. Instead of producing a single row of 'control points' as in *Figure 1* we have left three rows. With them we now deal in Step 2.

Step 2

Now apply the algorithm of de Casteljau in u-direction (2nd way) on every column $k=0, 1, 2$ of the matrix $\mathbf{B}'$. Stop again the calculation in the last but two steps of the algorithm. In this part of the algorithm for fixed column k the vectors $\vec{\mathbf{b}}^{M-2,N-2}_{0,k}$, $\vec{\mathbf{b}}^{M-2,N-2}_{1,k}$ and $\vec{\mathbf{b}}^{M-2,N-2}_{2,k}$ are computed from the vectors $\vec{\mathbf{b}}^{0,N-2}_{0,k},\ldots,\vec{\mathbf{b}}^{0,N-2}_{M,k}$. These vectors are the common factors and build up for $k=0, 1, 2$ the 1st, 2nd and 3rd column of a 3×3-matrix

$$\mathbf{B}'' = \begin{pmatrix} \vec{\mathbf{b}}^{M-1,N-2}_{0,0} & \vec{\mathbf{b}}^{M-2,N-2}_{0,1} & \vec{\mathbf{b}}^{M-2,N-2}_{0,2} \\ \vec{\mathbf{b}}^{M-2,N-2}_{1,0} & \vec{\mathbf{b}}^{M-2,N-2}_{1,1} & \vec{\mathbf{b}}^{M-2,N-2}_{1,2} \\ \vec{\mathbf{b}}^{M-2,N-2}_{2,0} & \vec{\mathbf{b}}^{M-2,N-2}_{2,1} & \vec{\mathbf{b}}^{M-2,N-2}_{2,2} \end{pmatrix}$$

The results of Step 2 can be seen in *Figure 3*. Instead of computing a single point, as it is the result of the unmodified algorithm of de Casteljau and is shown in *Figure 1*, we computed in three steps from the results of Step 1 (marked by ∘) nine points, which are marked by •. These points contain all the informations we need, to compute the point on the Bézier surface and the desired derivatives. This will be done in the final step.

Step 3

As the operators G show, the reduction of this matrix has to be done now in both directions u and v.

Figure 4 shows the final step. Starting point is the matrix $\mathbf{B}''$, whose elements can be found in the upper-left corner of this table. From left to right the algorithm of de Casteljau is used in u-direction (changing

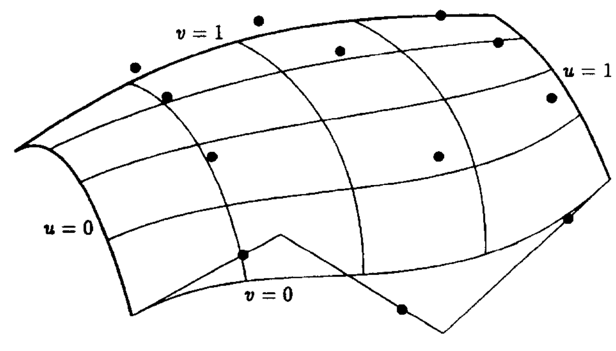

Figure 2 Situation after step 1

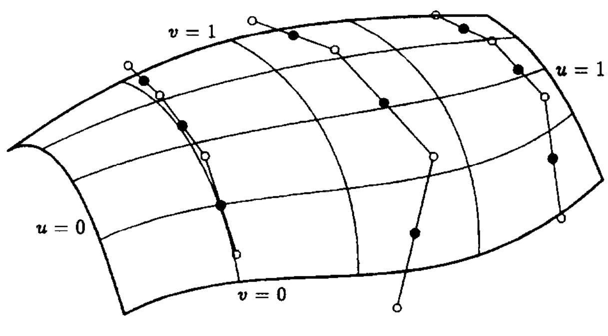

Figure 3 Situation after step 2

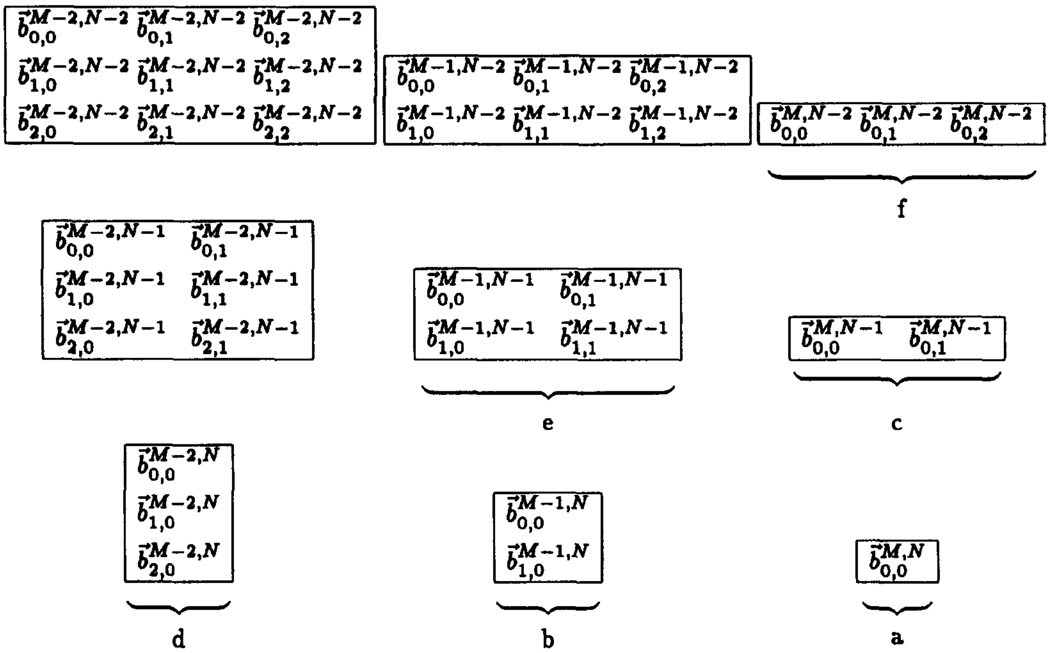

Figure 4 The third and final step

the first superscript). From top to bottom the algorithm of de Casteljau in v-direction is used (raising the second superscript). In the second diagonal of this table and below the vectors $\vec{\mathbf{b}}_{p,r}^{q,s}$ can be found, which result —combined according to Equations 8a–8f — in the partial derivatives of a Bézier surface in the point $X(u, v)$. The letters in *Figure 4* point to the corresponding formulae in the Equations 8a–8f.

It is now easy to see, how partial derivatives up to general order k can be computed. The reductions of the starting matrix in Steps 1 and 2 result in a $(k+1) \times (k+1)$-matrix, to which in the third step the algorithm of de Casteljau must be applied as well in u- as in v-direction. In this table the second diagonal contains all vectors, which are needed to combine the kth derivatives according to Equation 7. In the lower-left end of this diagonal the vectors to combine $\partial^k \vec{\mathbf{x}}/\partial u^k$ can be found, in the upper-right end the vectors for computing $\partial^k \vec{\mathbf{x}}/\partial v^k$. Looking for one partial derivative $\partial^k \vec{\mathbf{x}}/\partial u^j \partial v^{k-j}$ only, the $(k+1) \times (k+1)$-matrix has to be reduced by j steps of the algorithm of de Casteljau in v-direction and by $k-j$ steps in u-direction, to find the vectors to compute this derivative according to Equation 7.

To get the $(k-1)$ derivatives, the vectors below the second diagonal are combined. As in *Figure 4* the geometric vector of the point $X(u, v)$ on the Bézier surface occurs in the lower-right corner of the table.

The final remark concerns the computational costs of the algorithm. In Step 1 $(M+1)$ applications of the reduced algorithm of de Casteljau are necessary. So here $(M+1)[1/2\,N(N+1)-3)]$ convex combinations are computed. In Step 2 the three reduced algorithms in the u-direction compute $3[1/2M(M+1)-3]$ convex combinations. In the final step 27 convex combinations must be calculated. So totally $1/2((M+1)[N(N+1)-6]+3M(M+1)-18+54) = 1/2(MN^2+N^2+MN+M^2+M+N+2M^2-4M+30)$ convex combinations must be computed. Comparing this with the cost of the algorithm of de Casteljau (cf. the end of the section 'Bézier curves and surfaces'), the additional costs are the computation of $M^2-2M+15$ convex combinations. In the case of a Bézier surface of degree $(3,3)$ as in *Figure 1* this means the calculation of 18 additional convex combinations.

APPLICATIONS

A list of applications of differential geometry in computer-aided design was given by Hoschek[8]. The algorithm, presented in this paper, gives an effective way, to compute the invariants of differential geometry, which can be found in appropriate text-books, for example in Reference 7.

Using the partial derivatives $\vec{\mathbf{x}}_u$ and $\vec{\mathbf{x}}_v$, it can be determined, whether a point on the Bézier surface is regular or not. In the regular case the partial derivatives of first order define the tangent plane to the Bézier surface in this point and the normal

$$\vec{\mathbf{n}} = \frac{\vec{\mathbf{x}}_u \times \vec{\mathbf{x}}_v}{\|\vec{\mathbf{x}}_u \times \vec{\mathbf{x}}_v\|}$$

to the Bézier surface. In the tangent plane the coefficients of the first fundamental form

$$g_{11} = \vec{\mathbf{x}}_u \vec{\mathbf{x}}_u, \quad g_{12} = \vec{\mathbf{x}}_u \vec{\mathbf{x}}_v, \quad g_{22} = \vec{\mathbf{x}}_v \vec{\mathbf{x}}_v$$

determine length of tangent vectors as well as angles between two tangent vectors. The first fundamental form is also used, to determine the length of curves on the Bézier surface and the area of parts of the Bézier surface.

With the partial derivatives of second order the behaviour of the surface curvature in this point is known by the second fundamental form. Its coefficients are given by

$$h_{11} = \vec{\mathbf{x}}_{uu} \vec{\mathbf{n}}, \quad h_{12} = \vec{\mathbf{x}}_{uv} \vec{\mathbf{n}}, \quad h_{22} = \vec{\mathbf{x}}_{vv} \vec{n}$$

With this form the normal curvature κ_n in this point of the surface can be calculated for each direction $\vec{a} = a^1\vec{x}_u + a^2\vec{x}_v$ in the tangent plane. In addition to Hoschek's list this may be used in milling for cutter location and collision control.

The extreme values of the normal curvature, the principal curvatures, can be computed by the first and second fundamental form by a well-known formula in differential geometry. The corresponding directions, the principal directions in the given point, can be calculated as the solutions of a 2×2-system of linear equations. The Gaussian curvature and the mean curvature are given as product and average of the principal curvatures. They may be used in addition to Hoschek's list to subdivide a surface in regions of small and big curvature, which can be worked in milling by different strategies.

Beside the normal curvature, the geodesic curvature of a curve on the Bézier surface can also be computed. So the geodesics — curves with zero geodesic curvature — on the surface can be determined. It is well known that the shortest way between two points on the surface is part of a geodesic.

The computation of asymptotic lines can be accomplished by these fundamental forms as well as the determination of umbilic points, of the net of curvature lines and of geodesic parallel curves on the Bézier surface.

REFERENCES

1 Bézier, P *Numerical Control — Mathematics and Applications* Wiley, New York (1972)

2 Faux, I D and Pratt, M J *Computational Geometry for Design and Manufacture* Wiley, New York (1979)

3 Boehm, W, Farin, G and Kahmann, J 'A survey of curve and surface methods in CAGD' *Comput. Aided Geom. Des.* Vol 1 (1984) pp 1–60

4 Aumann, G and Spitzmüller, K *Computerorientierte Geometrie* BI-Wissenschaftsverlag, Mannheim (1993)

5 Choi, B K and Jun, C S 'Ball-end cutter interference avoidance in NC machining of sculptured surfaces' *Comput.-Aided Des.* Vol 21 No 6 (1989) pp 371–378

6 Carmo, M do *Differential Geometry of Curves and Surfaces* Prentice Hall, NJ (1976)

7 Hosaka, M and Kimura F 'Synthesis methods of curves and surfaces in interactive CAD' *in* Valle, G (Ed.) *Proc. Interactive Techniques in Computer Aided Design* Bologna (1978) pp 151–156

8 Hoschek, J 'Applications of differential geometry in computer-aided design' *in* Fuchssteiner, B and Luxemburg, W A J (Eds.) *Analysis and Geometry* BI-Wissenschaftsverlag, Mannheim (1992)

Klaus Spitzmüller is an assistant lecturer of geometry at the University of Karlsruhe, Germany. He was born in Offenburg, Germany, in 1958. He passed his examination for secondary school teachers in mathematics and physics in 1985, and received a PhD in mathematics in 1990. His research interests are in differential geometry and in computer-aided geometric design.