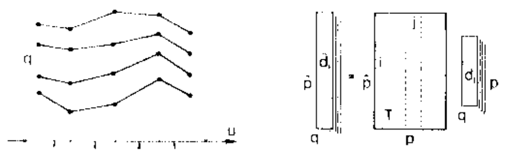

Fig. 2. The knot line insertion problem.

2. Inserting new knot lines

In the case, where the problem is to be solved for several columns, say q, of control points corresponding to the same knot refinement, e.g. as appearing already in 3D curves on in tensor product spline surfaces, one can compute $\hat{p}, p$-matrix T that transforms the d_i into the $\hat{d}_i$, see Fig. 2.

The a priori non-vanishing elements of this matrix can be computed:

(iii) row by row by the triangular schemes of the Oslo I algorithm [Cohen et al. '80], or

(iv) column by column by the insertion algorithm (ii) applied to the p unit columns j, where only one $d_j = 1$, while all others vanish.

The q columnes $[\hat{d}_i]$ can be computed column by column from the columns $[d_i]$:

(v) by multiplication with the matrix T or directly,

(vi) by the simple use of the insertion algorithm (ii) without the use of T.

3. Operation count

The given algorithms chiefly use linear combinations. Their count depends on p, $\hat{p}$, q, and k, this is shown in Table 1, where $\sum k = (k+1)k/2$ concerning triangular schemes: The counts for computing the matrix T do no differ very much. However, for large q this count is insignificant with respect to the count corresponding to the transformation of q columns of surface control points. Besides this, matrix multiplication should use the bandstructure of T, so that the transformation of one column requires the equivalent of $\hat{p}k/2$ linear combinations only. The count shows that the simple insertion (ii) is at least equivalent or superior to any of the other methods also in the case of surfaces.

Some variations may be found e.g. by considering vanishing linear combinations etc. in computing and using the matrix T [Lyche et al. '85], by accelerating the q runs of (ii), or more simple by the parallel use of the simple insertion (ii) [Boehm et al. '85]. However, some of these improvements will complicate the algorithms and raise the costs.

Table 1
Count of linear combinations

	curve	matrix		surface
Oslo	(i) $\hat{p}\sum(k-1)$	(iii) $\hat{p}\sum k$	matrix	(v) $q\hat{p}k/2$
insertion	(ii) $(\hat{p}-p)(k-1)$	(iv) $(\hat{p}-p)k(k-1)$	insertion	(vi) $q(\hat{p}-p)(k-1)$

* This short paper was presented as a spontaneous addition to Lyche's report "Can the Oslo algorithm be made more efficient?" [Lyche et al. '84]

References

Boehm, W. (1980). Inserting new knots into B-spline curves. Computer-Aided Design 12, 99–102.

Boehm, W. and Prautzsch, H. (1984–85). The insertion algorithm. Computer-Aided Design 17, 58–59.

Cohen, E., Lyche, T. and Riesenfeld, R. (1980). Discrete B-splines and subdivision techniques in computer aided geometric design and computer graphics. Computer Graphics and Image Processing 14, 87–111.

Lyche, T., Morken, K. and Cohen, E. (1984). Knot line refinement algorithms for tensor product B-spline surfaces. Computer Aided Geometric Design, in this Volume.

In: CAGD, Vol. 2, No. 1-3, pp. 141-143

Computer Aided Geometric Design 2 (1985) 141–143
North-Holland

On the efficiency of knot insertion algorithms

Wolfgang BOEHM

Technische Universität Braunschweig, D-3300 Braunschweig, Fed. Rep. Germany

Presented at Oberwolfach 15 November 1984*
Revised 5 April 1985

Abstract. The principal differences of knot insertion algorithms are discussed. Their efficiencies are compared.

Keywords. Splines, B-splines, spline curves, spline surfaces, algorithms, subdividing (subdivision), refinements.

1. Inserting new knots

In 1980, Boehm [Boehm '80] and Cohen et al. [Cohen et al. '80] independently found two different algorithms to solve the following problem:

Given p scalar points d_i corresponding to given knots u_i, defining a B-spline curve of order k. Furthermore, given additional knots v_j, together with the u_i defining a 'refinement' of knots $\hat{u}_i$. The problem is thus to find $\hat{p}$ new control points $\hat{d}_i$ corresponding to the refined knots $\hat{u}_i$ and defining the same B-spline curve, see Fig. 1. Note that the number of $\hat{u}_i$ exceeds the number of $\hat{d}_i$ at most by k (see [Boehm et al. '84]).

Two solutions of the insertion problem were found in 1980:

(i) The 'Oslo II algorithm' [Cohen et al. '80] constructs each of the $\hat{d}_i$ by a triangular, i.e. two-dimensional, scheme.

(ii) the 'insertion algorithm' [Boehm '80] constructs the changes of the control points by inserting knot by knot, each by a linear, i.e. one-dimensional, scheme.

Because a linear scheme is a part of a triangular scheme and the Oslo algorithm requires $\hat{p}$ triangular schemes while the insertion algorithm requires only $\hat{p}$-p linear schemes, the insertion algorithm is clearly more efficient.

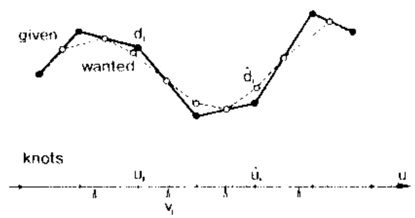

Fig. 1. The knot insertion problem.

0167-8396/85/$3.30 © 1985, Elsevier Science Publishers B.V. (North-Holland)