Selective knot insertion for symmetric, non-uniform refine and smooth B-spline subdivision

Thomas J. Cashman[a], Neil A. Dodgson[a], Malcolm A. Sabin[b]

[a] *Computer Laboratory, University of Cambridge, CB3 0FD, England, UK*
[b] *Numerical Geometry Ltd, 19 John Amner Close, Ely, Cambridge CB6 1DT, England, UK*

Notice: this is the author's version of a work that was accepted for publication in Computer Aided Geometric Design. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Computer Aided Geometric Design, [26, 4, 2009] doi:10.1016/j.cagd.2008.11.002

Abstract

NURBS surfaces can be non-uniform and defined for any degree, but existing subdivision surfaces are either uniform or of fixed degree. The resulting incompatibility forms a barrier to the adoption of subdivision for CAD applications. Motivated by the search for NURBS-compatible subdivision schemes, we present a non-uniform subdivision algorithm for B-splines in the spirit of the uniform Lane-Riesenfeld 'refine and smooth' algorithm. In contrast to previous approaches, our algorithm is independent of index direction (symmetric), and also allows a selection of knot intervals to remain unaltered by the subdivision process. B-splines containing multiple knots, an important non-uniform design tool, can therefore be subdivided without increasing knot multiplicity.

Key words: Non-uniform, Subdivision, NURBS, Smoothing, Lane-Riesenfeld, Knot insertion

1. Introduction and prior work

Freeform surfaces are largely designed using one of two representations. The first is NURBS, a set of closed forms offering fine-grained control, which are restricted to a rigid rectangular control grid. The second is subdivision, invented at about the same time as NURBS but with a more recent rise to prominence. Subdivision frees the designer from a rectangular control grid, granting a flexibility which has made it the de facto standard for character animation, among other applications. Designers continue to work with the restrictions of NURBS, however, in applications such as CAD/CAM where designers need the expressive power of non-uniform parameterizations such as multiple knots.

While NURBS surfaces can be non-uniform and defined for any degree, existing subdivision surfaces are either uniform or of fixed degree. The resulting incompatibility forms a barrier to the adoption of subdivision for CAD applications. Motivated by the search for subdivision schemes without these limitations, we present a non-uniform subdivision algorithm for B-splines in the spirit of the uniform Lane-Riesenfeld 'refine and smooth' algorithm. This is a general form of our earlier work (Cashman et al., 2009), which required a knot to be inserted in every existing knot interval. This more general form removes that constraint.

Subdivision works in a series of steps, each of which creates a denser control polygon (or mesh) from a sparser one. Lane and Riesenfeld (1980) observe that a single subdivision step for uniform B-splines can be factorised into a *refine* stage, followed by a sequence of local *smoothing* stages. Prautzsch (1998), in work developed further by Stam (2001), Zorin and Schröder (2001) and Warren and Weimer (2001), uses

Email addresses: tc270@cam.ac.uk (Thomas J. Cashman), nad@cl.cam.ac.uk (Neil A. Dodgson), malcolm@geometry.demon.co.uk (Malcolm A. Sabin)

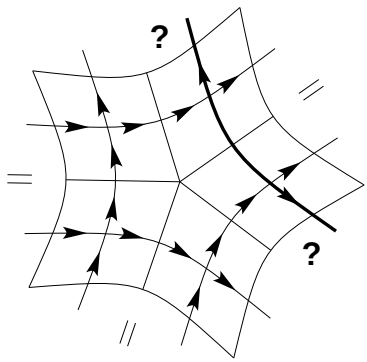


Figure 1: An asymmetric algorithm requires a consistent orientation for mesh edges, but this is impossible to achieve for some vertex valencies. In this example, no orientation for the bold edge is consistent with its parallel edges.

this factorisation to create subdivision schemes for surfaces generalising arbitrary degree uniform B-splines. The refine and smooth mechanism is essential for handling high degrees (above cubic). Without it there are too many special cases to consider for positions of an extraordinary point within a subdivision mask (rising exponentially with degree) and too much computation (rising quadratically for the unfactorised method but linearly for refine and smooth).

Working on a different type of generalisation, Sederberg et al. (1998) consider subdivision schemes extending non-uniform quadratic and cubic B-spline surfaces. These schemes can produce low-degree NURBS in regular regions while also including extraordinary regions where the control mesh is not regular. If the schemes could be extended to general degree, they would be *NURBS-compatible* – able to represent exactly any existing NURBS patch, but also handle extraordinary points. These surfaces might allow the advantages of subdivision to impact on Computer-Aided Design, where NURBS have largely held sway. Unfortunately there was, until recently, no non-uniform analogue of the Lane-Riesenfeld result. Goldman and Warren (1993) present an extension to knots in geometric progression, but do not handle the general case. Gregory and Qu (1996) consider a generalised "corner cutting" procedure, but only look at one such smoothing stage per subdivision step, and do not use their results to generate non-uniform B-splines.

As a step towards a NURBS-compatible scheme, we previously demonstrated (Cashman et al., 2009) a univariate algorithm which uses a refine and smooth factorisation, like Lane-Riesenfeld, but handles non-uniform knot intervals. Schaefer's univariate algorithm (Schaefer and Goldman, 2009) also fits these criteria, but is *asymmetric*; smoothing stages are dependent on the direction in which control points are indexed. To operate directly on a bivariate mesh, Schaefer's algorithm therefore requires each edge to have an orientation which is consistent with parallel edges. Figure 1 shows this is not possible to achieve for a mesh with general connectivity. Our algorithm, however, is *symmetric* (independent of orientation), and is therefore unaffected by the orientation problem.

Both algorithms, like Lane-Riesenfeld before them, require a new knot to be inserted in every existing knot interval. Unfortunately where knots are multiple, the subdivision process therefore increases multiplicity, which is undesirable. The algorithm described in this paper, in contrast, allows a selection of knot intervals to remain unchanged in the subdivided B-spline. In the special case where one knot is inserted in every interval, the new algorithm reduces to our earlier formulation (Cashman et al., 2009).

2. Problem statement

Subdivision is knot insertion, which has a standard theory. The new element here is a local refine and smooth factorisation of one of the existing knot insertion operations (for example, Boehm's (1980) or the Oslo (Cohen et al., 1980) algorithm).

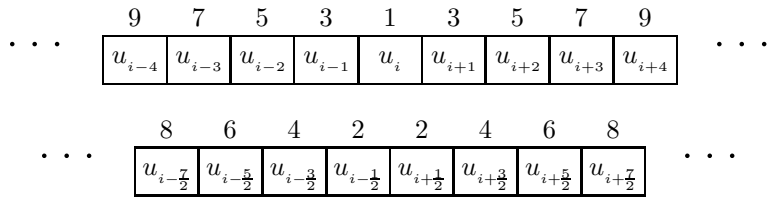


Figure 2: As λ increases, the polar arguments of a point P_i^λ include all the knots in an increasing region of $\mathbf{u}$. This is summarised above for d odd (top) and d even (bottom). Boxes represent knots u_k, where a new knot ($k \notin Y$) is inserted when λ is the value above box k. Figure 5 shows this process for an example where d is odd.

We have a B-spline B, of degree d that we wish to subdivide. The B-spline is defined by its knot vector, $\mathbf{t}$, and its control points. We must specify a new knot vector for the subdivided version. The knot insertion algorithm then determines the location of the new control points.

$\mathbf{u} = (u_j), j \in Z$, is a non-decreasing sequence that contains the knot vector for the subdivided B-spline plus any extra knots at each end required to handle end conditions. For convenience, we index it using Z, an uninterrupted interval on $\mathbb{Z}$. The original knot vector, $\mathbf{t}$, is a subset of $\mathbf{u}$: all of the original knots also appear in $\mathbf{u}$. We want the freedom to insert either zero or one new knot between adjacent original knots. There is therefore no way to index the original knots other than to explicitly list them. Thus we define Y, the indexing set for $\mathbf{t}$, where $Y \subset Z$ and, since $\mathbf{t}$ is a subsequence of $\mathbf{u}$; $\mathbf{t} = (u_j), j \in Y$. For example, if we have five knots and wish to insert knots in every interval, then $Z = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $Y = \{1, 3, 5, 7, 9\}$. If, however, we did not want a new knot in the third of the four intervals, then $Z = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $Y = \{1, 3, 5, 6, 8\}$.

This allows us to subdivide B with multiple knots in $\mathbf{t}$ without increasing knot multiplicity. There are, however, restrictions on Y in order to ensure that this works correctly, and there are end conditions to consider, discussion of which we defer to Section 7.

The restrictions on Y are:

- Y has the same bounds as Z (i.e. Y and Z are either unbounded below, or share a common minimum, and likewise for the upper bound),
- at most one new knot is inserted between adjacent old knots (i.e. $\forall j \in Z \setminus Y$, $j+1 \in Y$ and $j-1 \in Y$).

This general setup allows for several specific scenarios:

- to insert a knot into every non-zero knot interval, $j, j+1 \in Y$ if and only if $u_j = u_{j+1}$,
- to increase multiplicity, $u_j = u_k$ where $j \in Y$ and $k \notin Y$,
- to retain a non-zero knot interval, $j, j+1 \in Y$ for $u_j \neq u_{j+1}$.

We can insert more than one knot in an interval by using more than one complete subdivision step.

3. The algorithm

Our algorithm is best expressed in terms of *blossoming* (Ramshaw, 1987) or, equivalently, the mathematics of *polar forms* (Ramshaw, 1989). The key idea is a recipe for the polar arguments to the blossom of a B-spline at every control point P_i^λ of every level λ of a refine and smooth algorithm. λ grows by steps of 2, so for a degree d algorithm there will be $\lceil d/2 \rceil$ levels. The recipe states that the arguments should include only the new knots that fall in a region of $\mathbf{u}$ of size λ, centred at i (see Figure 2). The polar arguments outside this region must belong to $\mathbf{t}$. As λ grows to d, this region grows to include d consecutive values in the new knot vector, and so the points P_i^d are control points on the subdivided B-spline. For example,

for even degree, at $\lambda = 0$, all knots are in $\mathbf{t}$, the original knot vector. At each level, zero, one or two new knots are added by the process described below, until at the final level ($\lambda = d$), all polar arguments are uninterrupted sequences on the new knot vector, $\mathbf{u}$.

We can formalise this recipe by defining $\text{cen}_i^\lambda(Z)$ to contain the λ indices in Z centred around i. Note that there are two cases here: either d is odd, λ is odd, and i is an element of Z. Or else d is even and λ is even, in which case i must lie between two elements of Z, because the control points of even degree B-splines align with knot *intervals*, not with knots themselves. In either case, we now have that the polar arguments of P_i^λ are

$$u_j, \text{ where } j \in \text{cen}_i^d(Y \cup \text{cen}_i^\lambda(Z)) \tag{1}$$

We now expand on this definition to show how the algorithm can be implemented. In Section 6, we show that (1) is a self-consistent strategy which shares data along a polygon in the correct way.

3.1. Refine

The refine stage operates on the original control points Q and builds the sequence of points P_i^0 (for even d) or P_i^1 (for odd d). The two cases must be handled separately so that subsequent smoothing stages can symmetrically examine two knots at a time.

Even d. When $\lambda = 0$, $\text{cen}_i^\lambda(Z) = \varnothing$ and so from (1), the polar arguments of P_i^0 are u_j for $j \in \text{cen}_i^d(Y)$. But these knots are consecutive in $\mathbf{t}$, and so each point in P^0 is a control point on the original B-spline. Specifically, P_i^0 is equal to the point in Q corresponding to the knot interval which contains the interval indexed by i. After the refine stage, P^0 therefore contains two copies of any control points which correspond to knot intervals where knots are inserted.

Odd d. From (1) when $\lambda = 1$ we have that the polar arguments of each point P_i^1 are the d knots from $\mathbf{t}$ centred around i, but including u_i even if $i \notin Y$. That is, the polar arguments of P_i^1 are

$$u_j, \text{ where } j \in \text{cen}_i^d(Y \cup \{i\}) \tag{2}$$

There are two cases: either $i \in Y$, or $i \notin Y$. If $i \in Y$, then (2) is again a collection of d consecutive knots from $\mathbf{t}$. For convenience, we can index the points Q using the value in Y such that u_i is the corresponding knot. Then we directly obtain $\mathsf{P}_i^1 = \mathsf{Q}_i$.

If $i \notin Y$, we need an affine combination of the two points Q_{i-1} and Q_{i+1} (note that $i-1, i+1 \in Y$ because of our restrictions on Y). If $\alpha = \min(\text{cen}_{i-1}^d(Y))$ and $\delta = \max(\text{cen}_{i+1}^d(Y))$, then

$$\mathsf{P}_i^1 = \frac{u_\delta - u_i}{u_\delta - u_\alpha}\mathsf{Q}_{i-1} + \frac{u_i - u_\alpha}{u_\delta - u_\alpha}\mathsf{Q}_{i+1}. \tag{3}$$

From the multiaffine property of the blossom, and the fact that Q_{i-1} and Q_{i+1} share $d-1$ polar arguments, we have that (3) inserts the knot u_i as a polar argument for P_i^1, as required by (2).

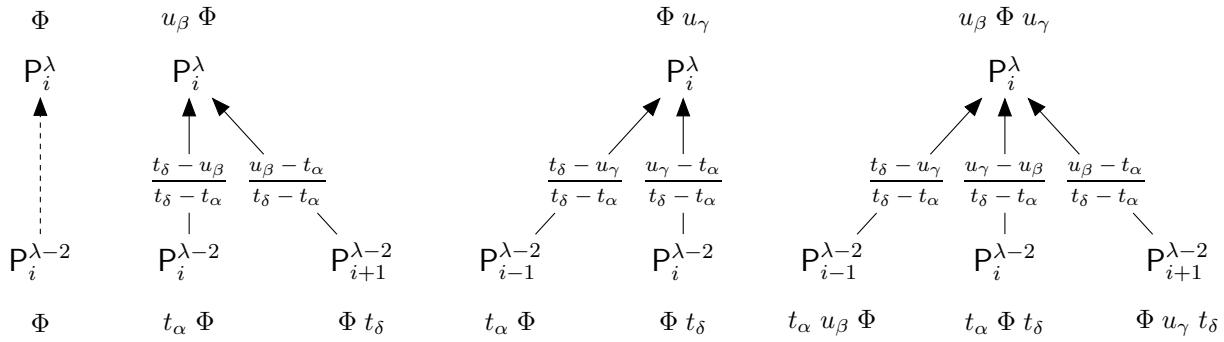
After the refine stage, P^1 contains an additional point on every edge which corresponds to a knot interval where a knot is inserted.

3.2. Smooth

Each smoothing stage produces the points P^λ from the sequence of points $\mathsf{P}^{\lambda-2}$, where the polar arguments of each point $\mathsf{P}_i^{\lambda-2}$ include the $\lambda - 2$ knots surrounding i. In order for each P_i^λ to contain the correct set of polar arguments, we must take affine combinations of points in $\mathsf{P}^{\lambda-2}$ to insert 0, 1, or 2 new knots. We distinguish between four possible actions, shown in Figure 3, where

$$\begin{aligned} \beta &= \min(\text{cen}_i^\lambda(Z)), \\ \gamma &= \max(\text{cen}_i^\lambda(Z)). \end{aligned} \tag{4}$$

Where α and δ appear, they are, respectively, the least and greatest indices of the polar arguments for a point. In an implementation, these values can either be stored for each i, or calculated on-the-fly. The four cases are:



(a) $|\Phi| = d$ (b) $|\Phi| = d - 1$ (c) $|\Phi| = d - 1$ (d) $|\Phi| = d - 2$

Figure 3: The four smoothing actions that form a point P_i^λ; points are shown with their polar arguments. $\beta, \gamma \notin Y$, and t_α and t_δ indicate knots u_α and u_δ with $\alpha, \delta \in Y$. Within each diagram, Φ denotes the same collection of (not necessarily consecutive) knots; typically u_β and u_γ will be placed somewhere in the interior of the knots Φ.

(a) No new knots. This action is illustrated in Figure 3a. If $\beta, \gamma \in Y$, then $\mathsf{P}_i^{\lambda-2}$ already contains all the polar arguments in a λ-sized region of $\mathbf{u}$. We can therefore directly copy $\mathsf{P}_i^{\lambda-2}$ to P_i^λ.

(b) New knot with index less than i. In this scenario we have $\gamma \in Y$ but $\beta \notin Y$. The polar arguments for $\mathsf{P}_i^{\lambda-2}$ therefore already contain u_γ, but we need an affine combination to insert the knot u_β. The coefficients that multiply $\mathsf{P}_i^{\lambda-2}$ and $\mathsf{P}_{i+1}^{\lambda-2}$ are shown in Figure 3b.

Note that in moving from $\mathsf{P}_i^{\lambda-2}$ to P_i^λ, the knot that is displaced from the polar arguments is t_α – the knot with the least index. This property holds for each action in Figure 3; where a knot is inserted, it replaces a knot from $\mathbf{t}$ on the same side of i. The replaced knot is also always further from i than the new knot. From this, we can conclude that inserting all the new knots in a d-sized region of $\mathbf{u}$ (when $\lambda = d$) results in polar arguments that are consecutive in $\mathbf{u}$, which is the condition for P^d to be the control points after subdivision.

(c) New knot with index greater than i. This action is illustrated in Figure 3c, and is completely symmetrical to the previous case. Here we have $\beta \in Y$ but $\gamma \notin Y$, and an affine combination of $\mathsf{P}_{i-1}^{\lambda-2}$ and $\mathsf{P}_i^{\lambda-2}$ inserts the knot u_γ.

(d) New knots on both sides of i. Here we have β and $\gamma \notin Y$, and so the smoothing action must introduce both u_β and u_γ into the polar arguments for P_i^λ. The required affine combination is shown in Figure 3d.

We can factorise action (d) into three affine combinations of two points in order to keep smoothing as local as possible. Instead of

$$\mathsf{P}_i^\lambda = \frac{t_\delta - u_\gamma}{t_\delta - t_\alpha}\mathsf{P}_{i-1}^{\lambda-2} + \frac{u_\gamma - u_\beta}{t_\delta - t_\alpha}\mathsf{P}_i^{\lambda-2} + \frac{u_\beta - t_\alpha}{t_\delta - t_\alpha}\mathsf{P}_{i+1}^{\lambda-2} \tag{5}$$

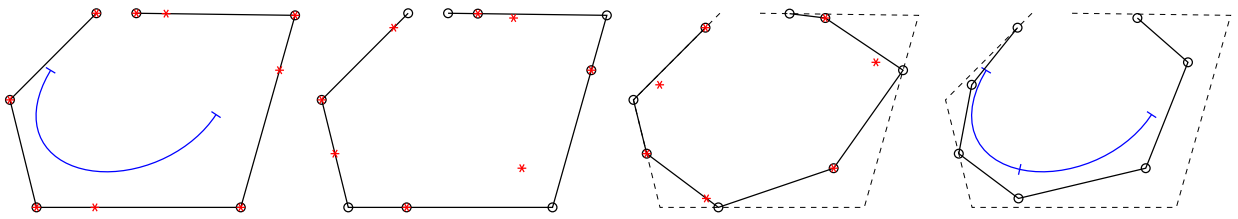
we can choose a pivot, x, and instead calculate

$$\mathsf{D} = \frac{t_\delta - u_\gamma}{t_\delta - x}\mathsf{P}_{i-1}^{\lambda-2} + \frac{u_\gamma - x}{t_\delta - x}\mathsf{P}_i^{\lambda-2} \tag{6}$$

$$\mathsf{E} = \frac{x - u_\beta}{x - t_\alpha}\mathsf{P}_i^{\lambda-2} + \frac{u_\beta - t_\alpha}{x - t_\alpha}\mathsf{P}_{i+1}^{\lambda-2} \tag{7}$$

$$\mathsf{P}_i^\lambda = \frac{t_\delta - x}{t_\delta - t_\alpha}\mathsf{D} + \frac{x - t_\alpha}{t_\delta - t_\alpha}\mathsf{E}. \tag{8}$$

The factorisation is kept as balanced as possible by setting x equal to



(a) Refine stage (b) Two smoothing stages (c) Subdivided B-spline

Figure 4: The example from Section 3.3 applied to the control polygon of a non-uniform quintic B-spline with multiple knots. Points marked with stars are passed to the next stage. A knot is inserted in every non-zero interval: before subdivision, the knot vector is [0, 0, 4, 4, 4, 12, 16, 18, 18, 20] and afterwards, it is [0, 2, 4, 4, 4, 8, 12, 14, 16, 17, 18] (the truncation is explained in Section 7).

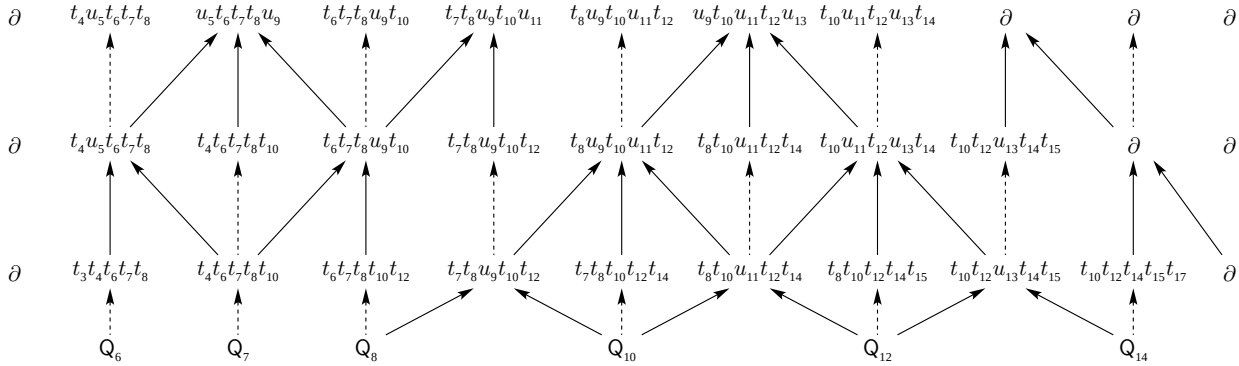


Figure 5: The example from Section 3.3 in full; see Figure 4 for an example application. At the bottom are the input points Q, above which are P^1, P^3, and finally the output P^5. Points are represented as a list of arguments to the polar form b of B.

- the mean of the knot interval indexed by i, when d is even,
- u_i, when d is odd.

The latter is the pivot used in our previous paper (Cashman et al., 2009) to insert a knot in every interval, since in this case β and $\gamma \notin Y$ never occurs for even degree.

3.3. An example

To illustrate the above algorithm, consider the example shown in Figure 4, where $d = 5$. We write $b(\cdot)$ for the polar form of B, and use $Z = \{3, \ldots, 17\}$, with

$Y =$	3	4		6	7	8		10		12		14	15		17
$Z \setminus Y =$			5				9		11		13			16	
$\mathbf{u} =$	0	0	2	4	4	4	8	12	14	16	17	18	18	19	20

Since d is odd, the refine stage produces P^1. Figure 5 shows the algorithm in full, and we consider two example points. $8 \in Y$, so $\mathsf{P}^1_8 = \mathsf{Q}_8 = b(u_6, u_7, u_8, u_{10}, u_{12})$. On the other hand, $11 \notin Y$, so P^1_{11} is an affine combination of Q_{10} and Q_{12}:

$$\begin{aligned} \mathsf{P}^1_{11} &= \frac{u_{15} - u_{11}}{u_{15} - u_7}\mathsf{Q}_{10} + \frac{u_{11} - u_7}{u_{15} - u_7}\mathsf{Q}_{12} \\ &= b(u_8, u_{10}, u_{11}, u_{12}, u_{14}) \end{aligned} \tag{9}$$

As these two points progress through the smoothing stages, we find that $\mathsf{P}_8^3 = b(u_6, u_7, u_8, u_9, u_{10})$, inserting u_9 using the affine combination in Figure 3c. Since P_8^3 has polar arguments consecutive in Z, we know that it is already a control point on the subdivided B-spline, and so inevitably $\mathsf{P}_8^5 = \mathsf{P}_8^3$ (as in Figure 3a). For the point centred on u_{11}, we have $\mathsf{P}_{11}^3 = \mathsf{P}_{11}^1 = b(u_8, u_{10}, u_{11}, u_{12}, u_{14})$ since $10, 12 \in Y$ (also using the action from Figure 3a). For P_{11}^5, however, we find that both 9 and $13 \notin Y$, so $\mathsf{P}_{11}^5 = b(u_9, u_{10}, u_{11}, u_{12}, u_{13})$ using the action in Figure 3d. This also results in polar arguments with indices that are consecutive in Z, as required.

4. Efficiency

Boehm (1985) shows that inserting n knots into a degree d B-spline uses, at best, dn affine combinations of two points. It is interesting that our algorithm, when d is even and a knot is inserted in every interval, calculates points P_i^λ at $2n$ values of i and $d/2$ values of λ. The total number of affine combinations for this case is therefore $2n(d/2) = dn$ (ignoring end conditions). The odd degree case has the complication of affine combinations of three points, but is comparable in efficiency.

On the other hand, consider inserting just one new knot. Every point P_i^λ will be a copy of $\mathsf{P}_i^{\lambda-2}$, except in the region of the new knot. Here, there will be d affine combinations of two points (one of which is used to form P^1 if d is odd). So here, too, the algorithm achieves Boehm's lower bound. There is a hidden cost here, of course, of establishing that every other point should be copied, but this observation is still interesting from a theoretical point of view.

Goldman (1990) shows that the Oslo algorithm (Cohen et al., 1980) can also achieve this lower bound, by overlapping the pyramids used to construct new points as much as possible. Goldman treats the Oslo algorithm as a *local* knot insertion algorithm (inserting into only one interval at a time), while refine and smooth algorithms are *global* (acting on all knot intervals at once). There may be a parallel result, however, showing in a similar way that the refine and smooth structure of this algorithm overlaps computations as much as possible. We consider that this algorithm's limit of at most one knot per interval is likely to be important in this sort of result, since Goldman provides a strong argument for the maximum amount of overlap that is possible in the general case. A search for such a result is left to future work.

5. Variants

There are various ways we can modify the algorithm from its description in Section 3. We have implemented variants that

- use a constant amount of memory by storing P_i^λ in the same location for every value of λ.
- have a unified refine stage, producing points in P^0 for both odd and even degree, accompanied by a special first smoothing stage when d is odd.
- use extra memory to improve efficiency, by storing the values of α and δ discussed in Section 3.2 for each index.
- use the factorisation of Figure 3d discussed in Section 3.2.
- use a different, 'eager' knot insertion process. If $\beta \in Y$ and $\gamma \in Y$, this variant searches outwards for a knot to insert by finding a $\lambda \leq d$ such that $\min(\mathrm{cen}_i^\lambda(Z)) \notin Y$ or $\max(\mathrm{cen}_i^\lambda(Z)) \notin Y$ (or both). At the following level, the search resumes from the value of λ that was used for the last affine combination. There is one special case; when both $\min(\mathrm{cen}_i^\lambda(Z)) \notin Y$ and $\max(\mathrm{cen}_i^\lambda(Z)) \notin Y$, the algorithm must first copy (as in Figure 3a) before inserting both knots (as in Figure 3d) at the next level. If there is no suitable $\lambda \leq d$, then the point is already a control point on the subdivided B-spline, and therefore is also simply copied to the next level. This variant inserts knots 'as soon as possible', and should have a similar proof to the algorithm described in Section 3. If an application needs the points surrounding unchanged knot intervals to stabilise as early as possible, this variant might be a good candidate. We have no reason here, however, to prefer the eager variant to the conceptually simpler version in Section 3, and so we do not consider it any further.

6. Proofs of correctness

In Section 3, we considered a point P_i^λ as λ increases to d, and showed that the refine and smooth stages result in a control point on the subdivided B-spline. So far, however, we have not justified the polar arguments for $\mathsf{P}_{i-1}^{\lambda-2}$ and $\mathsf{P}_{i+1}^{\lambda-2}$ shown in Figure 3. In this section we address the omission, thereby proving that the actions described in Section 3 mesh together correctly as i varies along a polygon.

To do so, we assume that the polar arguments of all points $\mathsf{P}_i^{\lambda-2}$ contain the $\lambda-2$ knots surrounding i. Section 3 showed that this property holds for P^λ, with a region of size λ, as long as we can show that the polar arguments take the form shown in Figure 3. There is nothing to prove for Figure 3a, and we will consider just the cases in Figure 3b and 3d. The proof for Figure 3c is completely symmetrical to that for 3b.

6.1. New knot with index less than i

First, note that the polar arguments for $\mathsf{P}_i^{\lambda-2}$ contain all the knots with indices in the range $\mathrm{cen}_i^{\lambda-2}(Z)$. The equivalent range for $\mathsf{P}_{i+1}^{\lambda-2}$ is $\mathrm{cen}_{i+1}^{\lambda-2}(Z)$. These ranges overlap on $\lambda-3$ values, and the polar arguments of both points must contain the knots indexed by the overlap. At the ends of these ranges are $\min(\mathrm{cen}_i^{\lambda-2}(Z))$ and $\max(\mathrm{cen}_{i+1}^{\lambda-2}(Z))$, and if either of these indices is not in Y (i.e. indexes a new knot), then that knot will not appear in the polar arguments of one of the points. On the other hand, if both $\min(\mathrm{cen}_i^{\lambda-2}(Z))$ and $\max(\mathrm{cen}_{i+1}^{\lambda-2}(Z)) \in Y$, then the knots indexed by these values will appear in both sets of polar arguments.

Now note that, since we are performing the action in Figure 3b, we already know that $\gamma \in Y$ and $\gamma = \max(\mathrm{cen}_i^\lambda(Z)) = \max(\mathrm{cen}_{i+1}^{\lambda-2}(Z))$. We also know that $\beta = \min(\mathrm{cen}_i^\lambda(Z)) \notin Y$ and so our restrictions on Y enforce that $\beta+1 \in Y$ and $\beta+1 = \min(\mathrm{cen}_i^{\lambda-2}(Z))$. Therefore the polar arguments of $\mathsf{P}_i^{\lambda-2}$ and $\mathsf{P}_{i+1}^{\lambda-2}$ overlap on $d-1$ knots Φ, differing only in the knot with lowest index t_α, and the knot with the highest index t_δ. So the polar arguments do, in fact, take the form shown in Figure 3b.

6.2. New knots on both sides of i

Here we have that $\beta \notin Y$ and $\gamma \notin Y$. $\beta = \min(\mathrm{cen}_i^\lambda(Z)) \in \mathrm{cen}_{i-1}^{\lambda-2}(Z)$, so u_β is certainly one of the polar arguments for $\mathsf{P}_{i-1}^{\lambda-2}$. The same argument shows that u_γ is a polar argument for $\mathsf{P}_{i+1}^{\lambda-2}$. If we can show that the polar argument with minimum index is the same for $\mathsf{P}_{i-1}^{\lambda-2}$ and $\mathsf{P}_i^{\lambda-2}$, and that the polar argument with maximum index is the same for $\mathsf{P}_i^{\lambda-2}$ and $\mathsf{P}_{i+1}^{\lambda-2}$, then it will necessarily follow that all three sets of polar arguments share $d-2$ knots Φ.

We know from (1) that we can find the index of the smallest polar argument in $\mathsf{P}_i^{\lambda-2}$ by selecting the $1+(d-\lambda)/2$ largest value in Y which is smaller than $\min(\mathrm{cen}_i^{\lambda-2}(Z))$. Furthermore, $\beta = \min(\mathrm{cen}_{i-1}^{\lambda-2}(Z)) \notin Y$. So the sets $\{y \in Y : y < \beta+1\}$ and $\{y \in Y : y < \beta\}$ are identical. Therefore the $1+(d-\lambda)/2$ largest value smaller than $\min(\mathrm{cen}_{i-1}^{\lambda-2}(Z))$ is the same as that less than $\min(\mathrm{cen}_i^{\lambda-2}(Z))$. The polar arguments for $\mathsf{P}_{i-1}^{\lambda-2}$ and $\mathsf{P}_i^{\lambda-2}$ therefore share a minimum-index knot t_α. Naturally, we can apply a symmetrical argument to show a common maximum-index knot t_δ for $\mathsf{P}_i^{\lambda-2}$ and $\mathsf{P}_{i+1}^{\lambda-2}$.

We can therefore justify the polar arguments in Figure 3, which validates the affine combinations we use in Section 3 and proves that the algorithm works correctly if we are far enough from any bounds of Z. It remains to show that the algorithm handles end conditions correctly, which we address in Section 7.

7. End conditions

The bounds on the domain of the B-spline (where they exist) are u_a and u_b, where a is the dth smallest index in Y and b is the dth largest. Therefore any point that has polar arguments with indices entirely below a, or above b, should be discarded so that the limit curve remains invariant. The knot vector of B after subdivision can therefore be shorter than $\mathbf{u}$. With $\mathbf{u}$ as in Section 3.3, for example, we have $a = 8$ and $b = 10$, with the knot vector after subdivision $\{u_4, \ldots, u_{14}\} \subset \mathbf{u}$.

The domain of the B-spline also allows us to specify the number of points which must be computed in the refine stage. For even degree, the refine stage builds P_i^0, where i starts by indexing the interval which

begins at the value $d/2$ smallest in Y. Similarly, the final i indexes the interval which ends at the value $d/2$ largest in Y. This is the largest possible range which maintains a polar argument with index in $[a\ b]$ for all points in P^0. In the same way, when d is odd, i ranges from the $\lceil d/2 \rceil$th smallest index in Y to the $\lceil d/2 \rceil$th largest.

Although these definitions guarantee that points are valid in P^1 or P^0, we now need to establish which points are discarded after smoothing. Each smoothing stage may, potentially, result in another point at each end with polar arguments indexed outside of the valid range. Fortunately there is a simple implementation which handles this complication automatically. We start by defining a boundary marker point, ∂, and append it to the results of the refine stage. For the example from Section 3.3, the result of the refine stage is then

$$\{\partial \quad \mathsf{P}^1_6 \quad \mathsf{P}^1_7 \quad \dots \quad \mathsf{P}^1_{13} \quad \mathsf{P}^1_{14} \quad \partial\}.$$

Now we simply define an affine combination of points, $(1-\mu)P_1 + \mu P_2$ to be equal to ∂ whenever $P_1 = \partial$ or $P_2 = \partial$. For our running example, $\mathsf{P}^3_{14} = \partial$ (see Figure 5), because the affine combination used to form P^3_{14} takes a contribution from $\mathsf{P}^1_{15} = \partial$. Note that this action (shown in Figure 3b) would otherwise displace t_{10} from this point's polar arguments, which would subsequently be indexed entirely above $b = 10$ and so P^3_{14} cannot be a control point on the subdivided B-spline. After the next smoothing stage, we similarly have $\mathsf{P}^5_{13} = \partial$, and also note that P^5_{13} is not one of the required control points.

In fact this implementation works because the actions in Figure 3 take a contribution from $\mathsf{P}^{\lambda-2}_{i+1}$ *whenever* the minimum index is increased (potentially above b). If P^λ_i would have polar arguments indexed entirely above b, then the same must be true of $\mathsf{P}^{\lambda-2}_{i+1}$, and so $\mathsf{P}^{\lambda-2}_{i+1} = \partial$ gives $\mathsf{P}^\lambda_i = \partial$. In the same way, the actions take a contribution from $\mathsf{P}^{\lambda-2}_{i-1}$ whenever the maximum index in the polar arguments of P^λ_i potentially drops below a.

Multiple knots are therefore neatly handled by a single framework without any special cases. In particular, this includes Bézier end conditions (where there are knots of multiplicity d terminating $\mathbf{t}$ and $\mathbf{u}$), which are widely used for the boundary of B-spline curves and surfaces.

8. Conclusion

Our algorithm provides the univariate foundations for a NURBS-compatible subdivision scheme. We believe it is the first appearance of a non-uniform, general degree, symmetric refine and smooth process that handles multiple knots without increasing multiplicity.

Unlike the Lane-Riesenfeld algorithm, the affine combinations used by smoothing stages depend on the knot values and may therefore vary both along a polygon and between stages. Like Schaefer's algorithm and our symmetric variant (Cashman et al., 2009), this algorithm also does not produce the control polygons for lower degree B-splines as the result of intermediate smoothing stages. We previously showed (Cashman et al., 2007) that this property limits the possible knot vectors, whereas this new algorithm does not constrain the choice of knots. New knots may be placed at the midpoints of existing intervals, such that the knot vector tends towards uniformity, or using any other criteria. Furthermore, the intervals for insertion may be chosen as well as new knot values.

Since affine combinations overlap as much as possible, the algorithm is efficient. At any time, at most d additional rounds of subdivision can bring every knot up to multiplicity d and hence yield points on the limit curve. The algorithm is trivially extended to a tensor product form that can be applied to meshes. We are working on a generalisation of this tensor product form to meshes containing extraordinary points.

References

Boehm, W., 1980. Inserting new knots into B-spline curves. Computer-Aided Design 12 (4), 199–201.

Boehm, W., 1985. On the efficiency of knot insertion algorithms. Computer Aided Geometric Design 2 (3), 141–143.

Cashman, T. J., Dodgson, N. A., Sabin, M. A., 2007. Non-uniform B-Spline Subdivision Using Refine and Smooth. In: Martin, R. R., Sabin, M. A., Winkler, J. R. (Eds.), 12th IMA Conference on the Mathematics of Surfaces. Vol. 4647 of Lecture Notes in Computer Science. Springer, pp. 121–137.

Cashman, T. J., Dodgson, N. A., Sabin, M. A., 2009. A Symmetric, Non-uniform, Refine and Smooth Subdivision Algorithm for General Degree B-splines. Computer Aided Geometric Design 26 (1), 94–104.

Cohen, E., Lyche, T., Riesenfeld, R., 1980. Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics. Computer Graphics and Image Processing 14 (2), 87–111.

Goldman, R., Warren, J., 1993. An extension of Chaiken's algorithm to B-spline curves with knots in geometric progression. CVGIP: Graphical Models and Image Processing 55 (1), 58–62.

Goldman, R. N., 1990. Blossoming and knot insertion algorithms for B-spline curves. Computer Aided Geometric Design 7 (1-4), 69–81.

Gregory, J. A., Qu, R., 1996. Nonuniform corner cutting. Computer Aided Geometric Design 13 (8), 763–772.

Lane, J. M., Riesenfeld, R. F., 1980. A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (1), 35–46.

Prautzsch, H., 1998. Smoothness of subdivision surfaces at extraordinary points. Advances in Computational Mathematics 9 (3), 377–389.

Ramshaw, L., 1987. Blossoming: A Connect-the-Dots Approach to Splines. Tech. Rep. 19, Digital Systems Research Center.

Ramshaw, L., 1989. Blossoms are polar forms. Computer Aided Geometric Design 6 (4), 323–358.

Schaefer, S., Goldman, R., 2009. Non-uniform Subdivision for B-splines of Arbitrary Degree. Computer Aided Geometric Design 26 (1), 75–81.

Sederberg, T. W., Zheng, J., Sewell, D., Sabin, M., 1998. Non-Uniform Recursive Subdivision Surfaces. Proceedings of SIGGRAPH 98, 387–394.

Stam, J., 2001. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. Computer Aided Geometric Design 18 (5), 383–396.

Warren, J., Weimer, H., 2001. Subdivision Methods for Geometric Design. Morgan Kaufmann.

Zorin, D., Schröder, P., 2001. A unified framework for primal/dual quadrilateral subdivision schemes. Computer Aided Geometric Design 18 (5), 429–454.