Non-uniform Subdivision for B-splines of Arbitrary Degree

S. Schaefer, R. Goldman

We present an efficient algorithm for subdividing non-uniform B-splines of arbitrary degree in a manner similar to the Lane-Riesenfeld subdivision algorithm for uniform B-splines of arbitrary degree. Our algorithm consists of doubling the control points followed by d rounds of non-uniform averaging similar to the d rounds of uniform averaging in the Lane-Riesenfeld algorithm for uniform B-splines of degree d. However, unlike the Lane-Riesenfeld algorithm which follows most directly from the continuous convolution formula for the uniform B-spline basis functions, our algorithm follows naturally from blossoming. We show that our knot insertion method is simpler and more efficient than previous knot insertion algorithms for non-uniform B-splines.

1. Introduction

Subdivision curves and surfaces are ubiquitous in Computer Graphics and Geometric Modeling. Subdivision recursively refines simple polygonal shapes which, if the subdivision rules are chosen correctly, converge in the limit to smooth shapes. Because subdivision can create smooth curves and surfaces of arbitrary topology, subdivision is now an integral part of Geometric Modeling [5].

The Lane-Riesenfeld subdivision algorithm [6] is the most commonly used subdivision algorithm for uniform B-splines of arbitrary degree. To subdivide a B-spline curve of degree d, the algorithm proceeds in two phases. The first step doubles each control point. This step is followed by d rounds of mid-point averaging, where each edge is replaced by a vertex located at the mid-point of that edge. Iterating this subdivision process generates a sequence of piecewise linear curves that in the limit converges to the uniform B-spline defined by the original control points.

We can easily visualize this algorithm as the pyramid shown in Figure 1. In this figure, arrows correspond to taking linear combinations of the points at the base of each arrow with the weights specified on each edge. This repeated averaging paradigm is quite powerful and lies at the foundation of many surface subdivision schemes based on uniform B-splines of arbitrary degree [7,10,12,13].

One of the disadvantages of the Lane-Riesenfeld algorithm is that this algorithm only operates on B-splines with uniform knot spacing. Generally, however, B-splines can have non-uniformly spaced knots. Non-uniform knot spacing forms the basis for NURBS curves and surfaces commonly used in Computer Aided Design. Therefore, here we consider the question of whether a subdivision algorithm, similar to Lane-Riesenfeld subdivision, can be constructed for non-uniform B-splines. In particular, given a B-spline curve of degree d with a monotonically increasing set of knots $t_1, t_2, \ldots t_n$ we would like to double the number of control points by inserting knots u_i such that $t_i \leq u_i \leq t_{i+1}$; that is, by

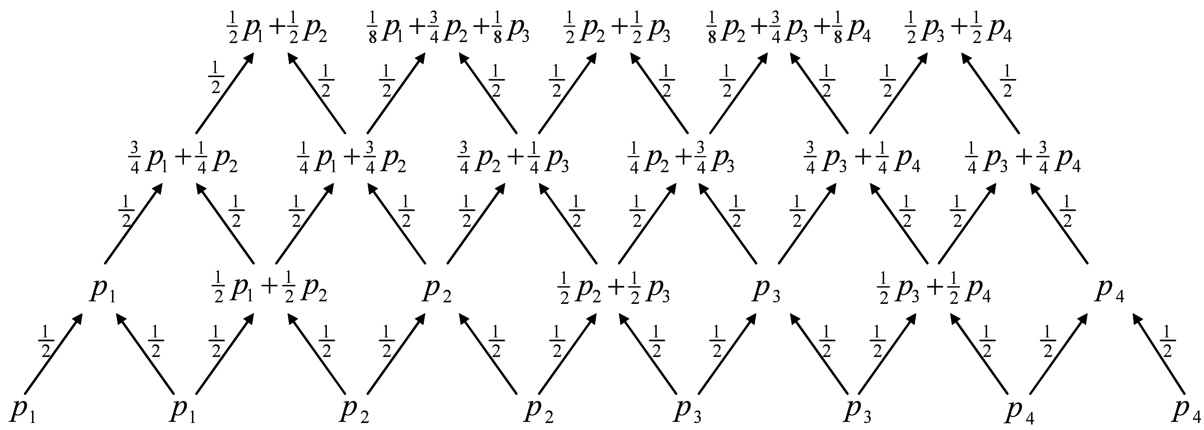


Figure 1. The Lane-Riesenfeld algorithm: the control points are doubled (bottom of the pyramid) followed by d rounds of midpoint averaging. Here we illustrate the cubic case.

inserting one new knot in each parameter interval.

2. Previous Work

There are actually several algorithms that perform knot insertion on non-uniform B-splines of arbitrary degree. For example, Boehm's knot insertion algorithm [2] inserts one knot at a time. However, this procedure must be repeated for every knot u_i, leading to a rather inefficient algorithm when compared to the Lane-Riesenfeld algorithm for uniform knot spacing.

The Oslo algorithm [4] is similar to Boehm's algorithm except that the Oslo algorithm simultaneously inserts multiple knots between two of the original consecutive knots. However, when we perform subdivision, we wish to insert only one new knot between each old pair of consecutive knots. In this setting the Oslo algorithm reduces to Boehm's knot insertion algorithm.

Barry et al. [1] provides a bounded depth variant of the Oslo algorithm for inserting multiple knots. Similar to the Oslo algorithm, this algorithm excels when inserting multiple knots between each old pair of consecutive knots whereas we will only be inserting a single new knot in each knot interval.

Sablonniere [9] introduces a tetrahedral algorithm for performing a change of basis between functions defined locally over two different knot sequences. This algorithm can be used to perform knot insertion as well but is seldom used in practice because Sablonniere's algorithm is slower than both Boehm's algorithm and the Oslo algorithm.

Contributions

We shall present an efficient algorithm for subdividing non-uniform B-splines of arbitrary degree in a manner similar to the Lane-Riesenfeld algorithm for uniform B-splines. Our algorithm consists of doubling the control points followed by d rounds of non-uniform averaging for a degree d B-spline curve where the averages depend on the local knot spacing of the curve. This algorithm has a structure similar to the Lane-Riesenfeld algorithm, which is also composed of doubling followed by d rounds of averaging; but in the Lane-Riesenfeld algorithm the averaging is uniform and independent of the local knot spacing. Moreover, unlike the Lane-Riesenfeld algorithm where the standard proof is based on the continuous convolution formula for the uniform B-spline basis functions [12], our algorithm follows easily from blossoming [8].

3. Non-uniform Subdivision

Consider a B-spline curve of degree d with knots $t_1 \leq t_2 \leq \ldots \leq t_n$ and control points $p_0, \ldots, p_{n-d}$. The control point p_i is associated with the knots $t_{i+1}t_{i+2}\ldots t_{i+d}$ and is represented by the polar form or blossom of the B-spline curve evaluated at these knot values [8]. Thus we write $p_i = b[t_{i+1}, \ldots, t_{i+d}]$ where b is the blossom of the B-spline curve.

Our goal is to insert new knots $u_1, u_2, \ldots u_{n-1}$ such that $t_i \leq u_i \leq t_{i+1}$. Knot insertion is akin to subdivision; our knot insertion procedure will produce a more highly refined control polygon relative to the new knot sequence. As the knots become dense, this subdivision procedure will generate a sequence of piecewise linear curves that converge in the limit to the non-uniform B-spline curve for the original control points.

When all of the t_i are uniformly spaced and each $u_i = \frac{t_i + t_{i+1}}{2}$, then the Lane-Riesenfeld algorithm is a simple, effective procedure for performing this knot insertion. However, if the t_i or u_i are non-uniform, then more complex algorithms must be applied. While Boehm's algorithm can always be used to perform knot insertion, Boehm's algorithm is slow and lacks the elegant structure of the Lane-Riesenfeld algorithm. Our algorithm is composed of two steps: doubling followed by d rounds of averaging, where d is the degree of the curve. This structure is identical to the Lane-Riesenfeld algorithm, except that our averaging rounds depend locally on the knot structure corresponding to the surrounding control points.

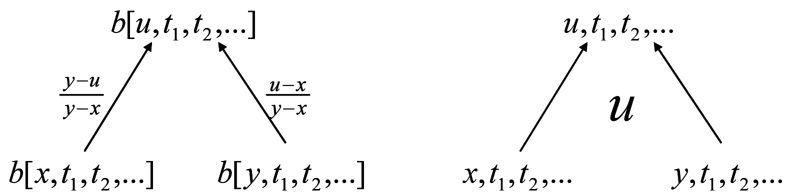


Figure 2. The multi-affine property of the blossom (left) and a short-hand notation indicating the inserted knot where the weights in the affine combination are implicit.

The first step in our algorithm, doubling, is exactly the same as the Lane-Riesenfeld algorithm and we simply double the control points in the curve. We then follow this step with d rounds of averaging. In the k^{th} averaging step where $k = 1, \ldots, d$, we locally insert the knots $u_{k-1}, u_k, u_k, u_{k+1}, u_{k+1}, \ldots$ using the multi-affine property of the blossom. This property states that if $b[x, t_1, t_2, \ldots]$ and $b[y, t_1, t_2, \ldots]$ are control points whose cor-

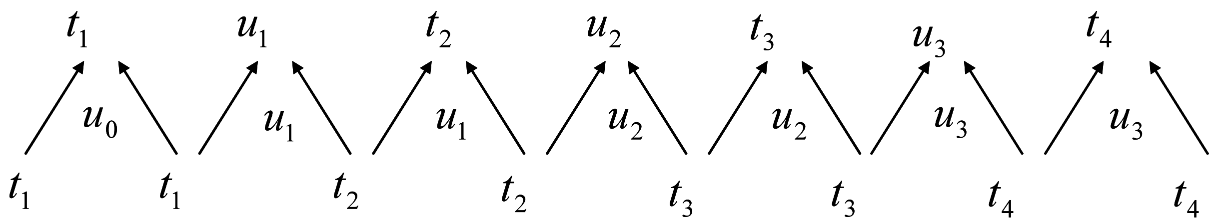


Figure 3. Our non-uniform subdivision algorithm for degree $d = 1$ where control points are referred to by the corresponding knot values. The knot value at the center of each triangle indicates the knot to insert locally using the two control points at the base of the triangle.

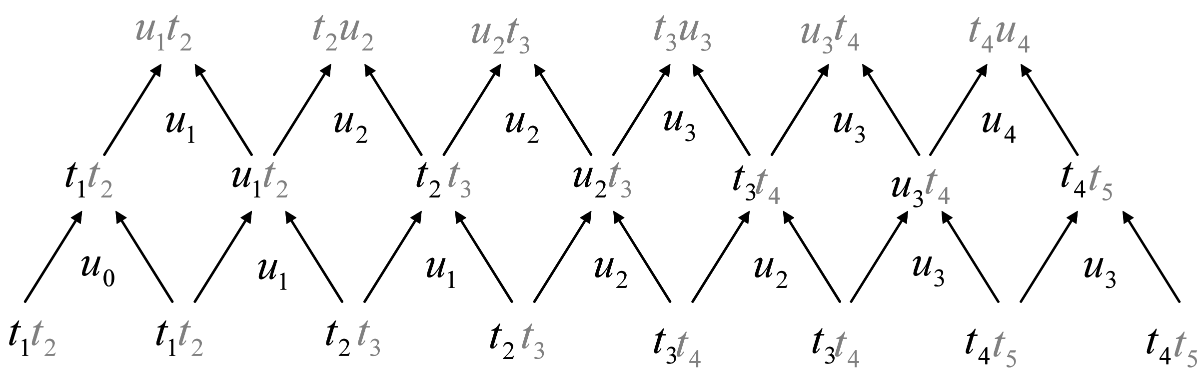


Figure 4. Our non-uniform subdivision algorithm for degree $d = 2$. Notice that the knots inserted at level 1 are identical to those used in Figure 3.

responding knot sequences differ by at most one value (x and y in our example) and the knot value we wish to insert is u, we simply form the affine combination

$$b[u, t_1, t_2, \ldots] = \frac{y - u}{y - x} b[x, t_1, t_2, \ldots] + \frac{u - x}{y - x} b[y, t_1, t_2, \ldots].$$

Figure 2 depicts this property. On the left the arrows indicate affine combinations along with the weights associated with each edge. The right side of the figure shows our equivalent notation that specifies only the inserted knot.

Figure 3 illustrates this algorithm for a linear B-spline. For convenience, we refer to the B-spline control points simply by their knot values and omit the function call to the blossom b. The control points sit at the base of the pyramid and are already doubled. The value shown in the center of each pyramid indicates the knot value to insert using the two control points at the base of the pyramid. The resulting control points represented by their knot values are shown at the top of the pyramid. The result of the computation in Figure 3 are the control points for the desired knot sequence after subdividing the curve.

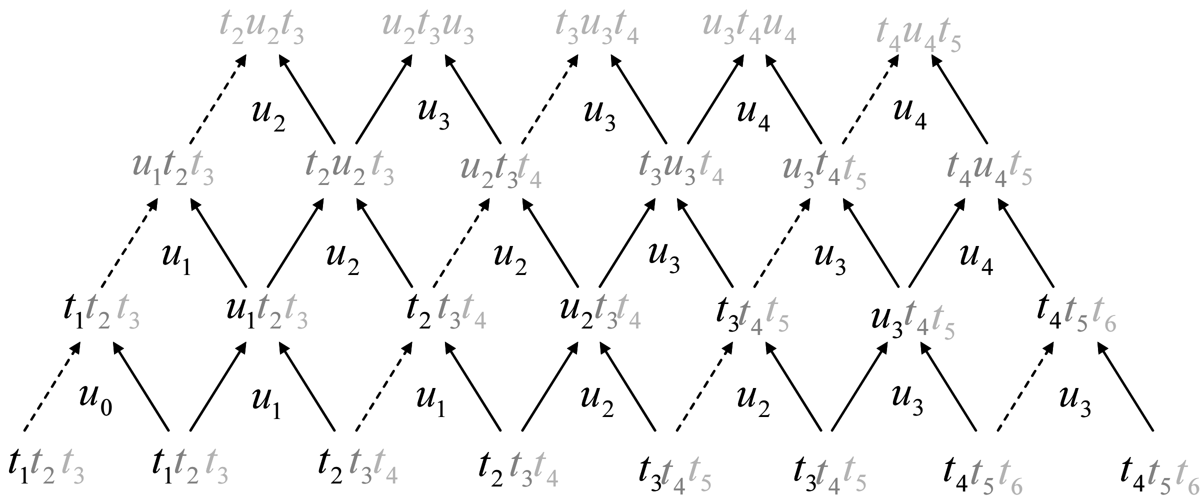


Figure 5. Our non-uniform subdivision algorithm for degree $d = 3$. The dashed edges indicate a weight of zero for the vertex at the base of the edge. Half of the old control points are copied to the next level at each round of averaging.

Figure 4 depicts our algorithm for a quadratic B-spline. The different intensities are designed to illustrate the nesting of the different degrees. If we consider only the black text, the algorithm is identical to the algorithm in Figure 3 for linear B-splines. The gray text indicates the different knots and control points needed for quadratic B-splines.

For quadratic B-splines two rounds of averaging are required to subdivide the curve. At averaging step $k = 1$ for the quadratic algorithm, the local knot values that we insert are identical to those from the linear algorithm. However, even though the knot value is identical, the affine combinations that we use at level $k = 1$ for the quadratic algorithm are not identical to the affine combinations in the linear algorithm due to the different knot sequences associated with the control points. Thus unlike the Lane-Riesenfeld algorithm where the averaging rule is simply to take the midpoint at every level of averaging no matter what the degree of the curve, the averaging rules in our algorithm depend on the degree of the curve. Nevertheless, after two rounds of averaging the knot sequence is exactly that of the subdivided curve.

Figure 5 shows our algorithm for a cubic B-spline curve with the gray-levels designed to show the nesting of the algorithm for different degree B-splines. Notice that every other edge in the pyramid is dashed. For these pyramids the knot inserted is already present in one of the two control points at the bottom of the pyramid. Therefore, these control points are simply promoted to the next level without performing any affine combination.

The fact that these control points are simply copied to the next level illustrates that our algorithm does not reduce to the Lane-Riesenfeld algorithm even when the original knots t_i are uniformly spaced and the new knots $u_i = \frac{t_i + t_{i+1}}{2}$. Notice then that our algorithm performs only half as many multiplications and additions for one round of subdivision as Lane-Riesenfeld subdivision though the later uses only combinations of $\frac{1}{2}$.

We can also write the operations in our algorithm compactly as a recurrence involving the control points and the knot values. Let p_i be the control points of the B-spline curve associated with the knots $t_{i+1}t_{i+2}\dots t_{i+d}$, where d is the degree of the curve. The points p_i^k at the k^{th} level of our algorithm are then given by

$$\begin{aligned}
p_{2i}^0 &= p_i \\
p_{2i+1}^0 &= p_i \\
p_{2i}^k &= p_{2i+1}^{k-1} \\
p_{2i+1}^k &= \begin{cases} \frac{t_{i+d+1}-u_{i+k}}{t_{i+d+1}-t_{i+(k+1)/2}} p_{2i+1}^{k-1} + \frac{u_{i+k}-t_{i+(k+1)/2}}{t_{i+d+1}-t_{i+(k+1)/2}} p_{2(i+1)}^{k-1} & k \text{ is odd} \\ \frac{t_{i+d+1}-u_{i+k}}{t_{i+d+1}-u_{i+k/2}} p_{2i+1}^{k-1} + \frac{u_{i+k}-u_{i+k/2}}{t_{i+d+1}-u_{i+k/2}} p_{2(i+1)}^{k-1} & k \text{ is even} \end{cases}
\end{aligned}$$

for $k = 1, \dots, d$. The points p_i^d represent the result of one round of subdivision.

Finally, notice that our subdivision scheme will produce a sequence of piecewise linear curves that converge in the limit to the non-uniform B-spline curve represented by the original control points and knots provided that the new knots u_i inserted at each level of subdivision lead to a dense covering of parameter space.

Figure 6 shows an example of subdivision for a non-uniform B-spline basis function of degree 3 using our algorithm where the knot spacing is uniform on both the left and right sides of the basis function, but the knots are twice as dense on the left as on the right. In this example, we simply insert the new knots u_i at the midpoint of each consecutive pair of old knots.

4. Conclusions and Future Work

Our algorithm is a simple, fast method for performing subdivision on non-uniform B-splines of arbitrary degree. Similar to the Lane-Riesenfeld algorithm, our technique is composed of doubling the control points followed by d non-uniform averaging steps where d is the degree of the curve. Our algorithm does not reduce to the Lane-Riesenfeld algorithm when the original knots are uniformly spaced and the new knots lie at the midpoints of the old knots. However, our method performs only half as many computations as the Lane-Riesenfeld algorithm. Moreover our method is easier to derive than the Lane-Riesenfeld algorithm. The standard proof of the Lane-Riesenfeld algorithm is based on the observation that the basis functions for uniform B-splines can be generated by continuous convolution [12]. In contrast, the proof of our algorithm is immediate from the dual functional and multiaffine properties of the blossom. Blossoming proofs of the Lane-Riesenfeld algorithm are also possible, but these proofs are not so simple [11].

In the future, we would like to explore whether or not we can construct an algorithm that mimics the Lane-Riesenfeld algorithm even closer. Our current technique requires that we know the degree of the curve before subdivision, since the degree influences the affine combinations used in the local knot insertion step. Ideally, we could perform these averaging passes without knowing the degree of the curve ahead of time, though current experiments indicate that this type of algorithm may not be possible in the fully non-uniform setting [3]. Nevertheless, if we place restrictions on the original knots t_i or the new knots u_i, such an algorithm may exist. We also believe that there exists an algorithm for each particular degree that reduces to the Lane-Riesenfeld algorithm without any

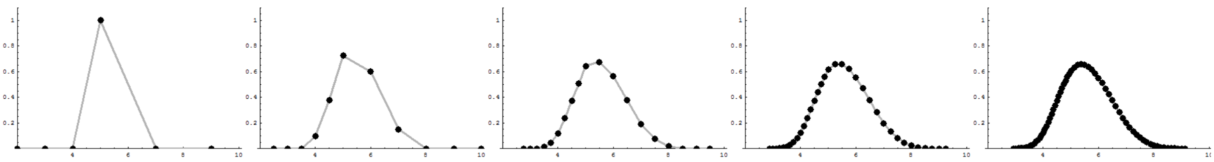


Figure 6. Subdivision of a non-uniform cubic B-spline basis function using our technique.

restrictions on the knots. However, these algorithms are special in each degree; the rules from one degree to another are completely different and, hence, do not nest like our algorithm or the Lane-Riesenfeld algorithm. Appendix A provides two examples of non-uniform knot insertion algorithms for cubic B-splines that reduce to the Lane-Riesenfeld algorithm for cubic B-splines when the knots are evenly spaced.

We would also like to generalize our method to create a quadrilateral subdivision scheme for non-uniform B-spline surfaces similar to the way the Lane-Riesenfeld algorithm is used to create surface subdivision schemes [7,10,12,13]. The tensor-product case is relatively straightforward and does not pose a challenge. However, extraordinary vertices complicate the algorithm. One of the biggest problems is simply encoding the knot vectors for the vertices in the presence of extraordinary vertices.

REFERENCES

1. Phillip Barry and Rui-Feng Zhu. Another knot insertion algorithm for B-spline curves. *Computer Aided Geometric Design*, 9(3):175–183, 1992.
2. W. Boehm. Inserting new knots into B-spline curves. *Computer Aided Design*, 12(4):199–201, 1980.
3. Thomas J. Cashman, Neil A. Dodgson, and Malcolm A. Sabin. Non-uniform B-Spline subdivision using refine and smooth. In *IMA Conference on the Mathematics of Surfaces*, pages 121–137, 2007.
4. E. Cohen, T. Lyche, and R. Riesenfeld. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14:87–111, 1980.
5. Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94, 1998.
6. J. Lane and R. Riesenfeld. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:35–46, 1980.
7. Hartmut Prautzsch. Smoothness of subdivision surfaces at extraordinary points. Adv. in Comp. Math., 9: 377-390, 1998.
8. L. Ramshaw. Blossoms are polar forms. *Computer Aided Geometric Design*, 6:323–358, 1989.
9. P. Sablonniere. Spline and Bezier polygons associated with a polynomial spline curve. *Computer Aided Design*, 10(4):257–261, 1978.

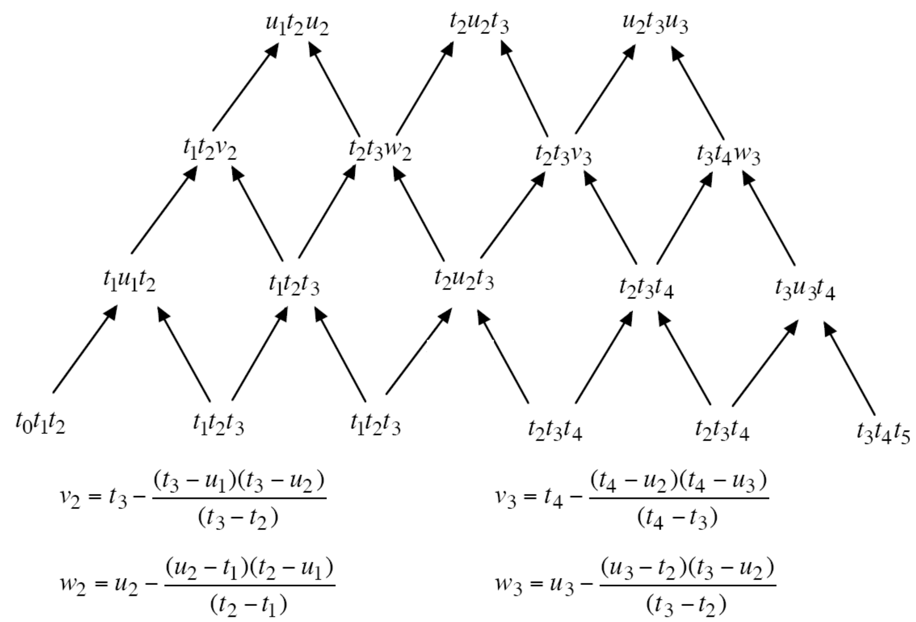


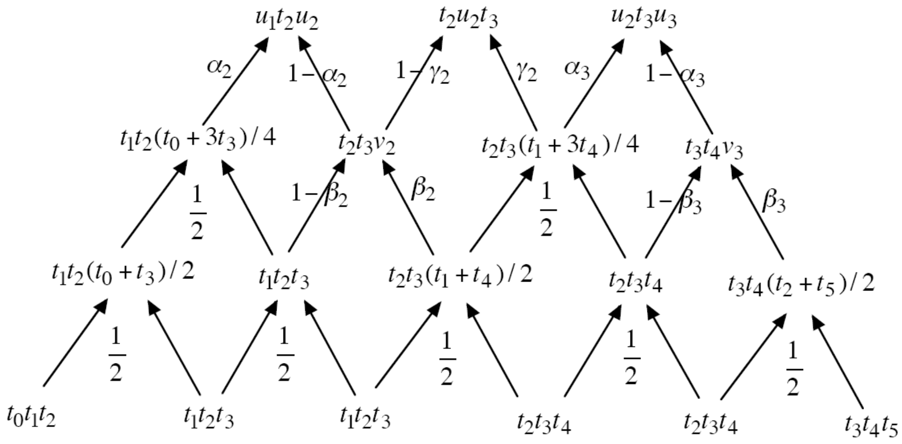
Figure 7. A subdivision algorithm for non-uniform cubic B-splines that reduces to the Lane-Riesenfeld algorithm when the knots are uniformly space.

10. Jos Stam. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. *Computer Aided Geometric Design*, 18(5):383–396, 2001.
11. E. Vouga and R. Goldman. Two blossoming proofs of the Lane-Riesenfeld algorithm. *Computing*, 79:153–162, 2007.
12. Joe Warren and Henrik Weimer. *Subdivision Methods for Geometric Design: A Constructive Approach.* Morgan Kaufmann Publishers Inc., 2001.
13. Denis Zorin and Peter Schröder. A unified framework for primal/dual quadrilateral subdivision schemes. *Computer Aided Geometric Design*, 18(5):429–454, 2001.

A. Non-uniform Subdivision Reducing to the Lane-Riesenfeld Algorithm

Our algorithm for inserting knots into non-uniform B-splines of arbitrary degree does not reduce to the Lane-Riesenfeld algorithm when the knots are uniformly spaced. However, for a particular degree, we believe that it is always possible to construct a knot insertion algorithm that reduces to the Lane-Riesenfeld algorithm when the knots are evenly spaced. Figure 7 shows an example of such an algorithm for cubic B-splines. This algorithm requires the insertion of pseudo-knots that are not part of the original or final knot sequences. However, these knots cause the algorithm to reduce to the Lane-Riesenfeld algorithm when the original knots t_i are uniformly spaced and the new knots $u_i = \frac{t_i + t_{i+1}}{2}$.

The disadvantage of this cubic algorithm is that it does not generalize to B-splines of different degrees. Furthermore, the algorithm itself is not unique; there exists many different pseudo-knots and insertion orders that will produce the same knot sequence after



$$\alpha_2 = \frac{4(t_3 - u_1)(t_3 - u_2)}{(t_3 - t_0)(t_3 - t_1)} \qquad \beta_2 = \frac{2(u_2 - t_1)(u_1 - t_1)}{(t_4 - t_1)(t_3 - t_1)(1 - \alpha)}$$

$$v_2 = t_1 + \frac{\beta_2 (t_4 - t_1)}{2} \qquad \gamma_2 = \frac{4(u_2 - v_2)}{t_1 + 3t_4 - 4v_2}$$

Figure 8. Another subdivision algorithm for non-uniform cubic B-splines that reduces to the Lane-Riesenfeld algorithm when the knots are uniformly space.

subdivision and reduce to the Lane-Riesenfeld algorithm for uniform knots. For example, Figure 8 illustrates another subdivision algorithm for non-uniform cubic B-splines that reduces to the Lane-Riesenfeld algorithm when the knots are equally spaced. Again, this algorithm relies on the insertion of pseudo-knots but does not produce that same intermediate affine combinations. Currently, it is an open problem whether or not there exists an algorithm that reduces to the Lane-Riesenfeld algorithm for uniform knots and has a simple, nested structure relative to the degree of the B-spline curve.