Topological Neighborhoods for Spline Curves : Practice & Theory

L. Miller, E. L. F. Moore, T. J. Peters*, A. C. Russell

April 24, 2006

Abstract

The unresolved subtleties of floating point computations in geometric modeling become considerably more difficult in animations and scientific visualizations. Some emerging solutions based upon topological considerations will be presented. A novel geometric seeding algorithm for Newton's method was used in experiments to determine feasible support for these visualization applications.

1 Computing the pipe surface radius

Parametric curves have been shown to have a particular neighborhood whose boundary is non-self-intersecting [5]. It has also been shown that specified movements of the curve within this neighborhood preserve the topology of the curve [9, 8], as is desired in visualization. This neighborhood is defined by a single value, which is the radius of a pipe surface, where that radius depends on curvature and the minimum length over those line segments which are normal to the curve at both endpoints of the line segment [5]. Since computation of curvature is a well-treated problem, the focus of this paper is efficient and accurate floating point techniques to compute the other dependeancy for that radius.

*Department of Computer Science & Engineering, University of Connecticut, Storrs, CT 06269-2155, tpeters@cse.uconn.edu.

Definition 1.1 *For a non-self-intersecting, parametric curve c, where*

$$c : [0, 1] \to \mathbb{R}^3,$$

and for distinct values $s, t \in [0, 1]$, *the line segment* $[c(s), c(t)]$ *is* doubly normal *if it is normal to c at both of the points* $c(s)$ *and* $c(t)$.

To avoid unnecessary complications with computing derivatives, only curves with regular parameterization [3] are considered.

Definition 1.2 *The* global separation *is the minimum over the lengths of all doubly normal segments. (For compact curves, this minimum has been shown in be positive [6].)*

An example cubic B-splines curve is given in Figure 1, with

1. control points: (0.0 0.0 0.0) (-1.0 1.0 0.0) (4.5 5.5 2.0) (5.0 -1.0 8.5) (-1.5 2.5 -4.5) (4.5 6.0 8.5) (3.5 -3.5 0.0) (0.0 0.0 0.0) &
2. knot vector: {0 0 0 0 0.2 0.4 0.6 0.8 1 1 1 1}

For this curve, there exist many doubly normal segments, as shown in Figure 1. The problem is how to efficiently find all these doubly normal seg-

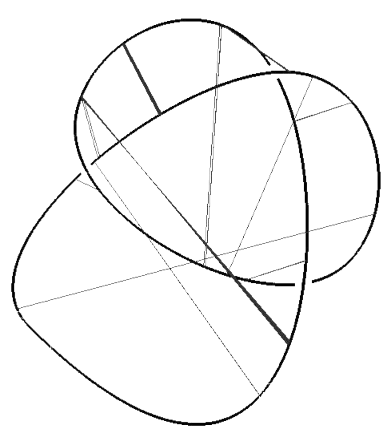

Figure 1: Many doubly normal segments exist on this curve.

ments, and then find the pair which represents the global separation distance, denoted as σ. A pair of distinct points at parametric values s and t on a curve will be endpoints of a doubly normal segment if they satisfy the two equations [5]:

$$[c(s) - c(t)] \cdot c'(s) = 0 \tag{1}$$

$$[c(s) - c(t)] \cdot c'(t) = 0 \tag{2}$$

where, $s, t \in [0, 1]$

In principle, the system given by Equations 1 and 2 could be solved algebraically by writing them in their power basis form, but this approach results in well-known algorithmic difficulties [10]. Hence, alternative techniques will be presented. For the software infrastructure available to these authors, it was convenient to convert the B-spline curve into a composite Bézier curve by the usual technique of increasing the multiplicity of each interior knot to 3. This produces 5 subcurves, with control points, as follows (rounded to 2 decimal places for this exposition):

- (0 0 0) (-1 1 0) (1.75 3.25 1) (3.21 3.29 2.58),
- (3.20 3.29 2.58) (4.67 3.33 4.17) (4.83 1.17 6.33) (3.83 0.67 5.25),
- (3.83 0.67 5.25) (2.83 0.17 4.17) (0.67 1.33 -0.17) (0.58 2.5 -0.17),
- (0.58 2.5 -0.17) (0.5 3.67 -0.17) (2.5 4.83 4.17) (3.25 3.04 4.21),
- (3.25 3.04 4.21) (4 1.25 4.25) (3.5 -3.5 0) (0 0 0).

Newton's method for two variables [7] was applied to Equations 1 and 2. The numerical experiments reported on prototype code suggest that this approach could be sufficiently rapid to support scientific visualization. These experiments were performed on a 64-bit AMD processor with Red Hat Linux Fedora Core 2 and OpenGL with double buffering. As always, the integration with a specific graphics subsystem is highly dependant upon the underlying architecture, and incorporation of this code on any platform would require further development and experimentation.

As is typical, the 'art' required for the successful use of Newton's method is highly dependant upon the determination of reasonable initial estimates, within the following standard formulation

$$\begin{bmatrix} s_{n+1} \\ t_{n+1} \end{bmatrix} = \begin{bmatrix} s_n \\ t_n \end{bmatrix} - J^{-1}(s_n, t_n) \begin{bmatrix} f(s_n, t_n) \\ g(s_n, t_n) \end{bmatrix}, n = 0, 1, ... \tag{3}$$

until $|J^{-1}(s_n, t_n)[f(s_n, t_n)\ g(s_n, t_n)]^T|$ is less than some $\epsilon > 0$, where $J^{-1}(s_n, t_n)$ is the inverse Jacobian matrix.

A viable approach to this art is presented and verified on an illustrative example. The general idea is to take finitely many points on each subcurve and consider all line segments between each pair of points as a candidate for being doubly normal. Many of these segments can be excluded from further consideration by an easy culling technique based upon a lack of normality at one end point or the other.

Let $< c(s), c(t) >$ denote the vector of unit length, formed by taking the vector between $c(s)$ and $c(t)$ and dividing that vector by its norm. Let $c'(s)$ and $c'(t)$ denote the unit tangent vectors at points $c(s)$ and $c(t)$, respectively. Let ϵ_1 and ϵ_2 be positive. The following modifications of Equations 1 and 2 are used

$$< c(s), c(t) > \cdot\ c'(s) < \epsilon_1, \tag{4}$$

and

$$< c(s), c(t) > \cdot\ c'(t) < \epsilon_2. \tag{5}$$

If the result of the preceding comparison is false, then this segment is rejected. Otherwise, it is sufficiently close to being doubly normal to serve as an initial estimate for Newton's method.

These candidate double normal points are shown graphically in Figure 2 with line segments connecting the pairs of candidate points for each pair of Bézier segments. Pairs of Bézier segments that do not have a connecting line segment, mean that no candidate doubly normal points were found. Pairs of Bézier segments that have only one connecting line segment, mean that Newton's method did not converge for those particular points. Pairs of Bézier segments that have two pairs of connecting line segments mean that Newton's method did converge, and the resulting pair of minimum double normal points is one of the two line segments from each pair. Typically, convergence with $\epsilon = 0.0001$ occured after 3 or 4 iterations. Note that Figure 2 depicts the same curve as in Figure 1, but now the curve is rotated about the y-axis to get a better view of doubly normal points, with σ illustrated in the zoomed-in section of Figure refnewtons-method-color.

Table 1 summarizes experimental work completed. Tests 1 - 3 report on a naive, direct approach. This relies purely upon the limiting notion that sufficiently many approximation pairs will produce a list that contains a reasonable estimate for σ. This produces the reliable estimates shown in both

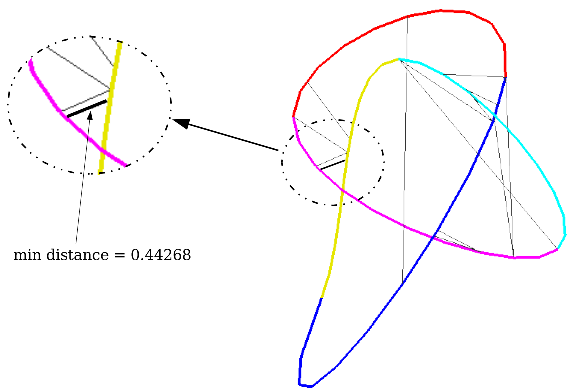

Figure 2: Newton's method.

Tests 1 and 2, but at prohibitively slow performance for visualization applications. Furthermore, Test 3 shows that further coarsening on the partition results in both poor estimates for σ *and* unacceptable performance. Alternatively, Test 4 shows that Newton's method produces a reliable estimate of σ with acceptable performance over a very coarse partition [1].

Test #	Method	Partition Size, n	Time(s)	σ
1	Direct	10,000	85	0.44268
2	Direct	2,000	6	0.44268
3	Direct	1,000	2	0.91921
4	Newton	10	N	0.44268

Table 1: Estimating σ

2 Guaranteeing a Lower Bound

The estimate of σ produced by Newton's Method can be done quickly, but it could easily be an overestimate of σ. To be conservative in the use of any estimate of σ it would be desirable to have a guaranteed lower bound. Hence, the theory to produce such a lower bound is presented, though it should be

[1] As a verification of the Newton's code implemented, the value of σ for this experimental curve was corobborated by an independently created code [1].

clear that the partitioning becomes prohibitively expensive, so this algorithm has not been presented. This analysis is consistent with the presentation and discussion of Table 1.

2.1 Partitioning by Taylor's Theorem

The algorithm presented in this section will proceed in terms of a parameter δ. One algorithmic criterion will be to create a PL approximation of c that is within $\delta/2$ of c. A lower bound will be provided for σ, and an *a priori* bound can be provided upon the number of subdivision iterations needed to achieve that approximation. Hence, for a curve c, approximation of σ relies on the following two conditions.

1. The distance between a PL approximant and c must be less than $\delta/2$.
2. The edges of the PL approximant must be good local approximations to the tangents of c.

To do so, the rest of the analysis presented will be to find a partition of the parametric interval $[0, 1]$ by the increasing sequence of points

$$0 = s_0, s_1, \ldots, s_\ell = 1.$$

Then a PL approximation to c is created by connecting the interpolant points

$$c(s_0), c(s_1), \ldots c(s_\ell).$$

Both conditions can be met by invoking Taylor's Theorem [2]. Taylor's Theorem is stated as follows. For $f : \mathbb{R} \to \mathbb{R}$ and $n > 0$, suppose that $f^{(n+1)}$ exists for each x in a non-empty open interval $I \subset \mathbb{R}$ containing a. For each $x \neq a$ in I there exists t_x strictly between a and x such that

$$f(x) = f(a) + f'(a)(x - a) + \ldots + \frac{f^n(a)}{n!}(x - a)^n + r_n(x)$$

where

$$r_n(x) = \frac{f^{(n+1)}(t_x)}{(n+1)!}(x - a)^{n+1}$$

Note that this statement of Taylor's Theorem is for the univariate case into $\mathbb{R}$, whereas the present application is to map $c : [0, 1] \to \mathbb{R}^3$, a univariate function into $\mathbb{R}^3$. However, the x, y and z components can be treated independently as functions into $\mathbb{R}$.

Condition 1. PL Approximation within $\delta/2$: This part discusses the creation of a PL approximant of c that is within $\delta/2$ of c. Since the value for δ is unknown, it is initially set to $\delta = 1$. The resulting estimate for σ is then tested for validity (See algorithm in Section 2.2, below.), and failure results in halving δ, repeating the iterations until a valid estimate for σ is obtained. In this way, the overall algorithm is logarithmic in $1/\sigma$.

Since only continuous functions defined on the compact set $[0, 1]$ are considered, there is a maximum positive value for $\|c'(t)\|$, denoted as M_0. Recall that $c'(t)$ is non-zero from the regularity of c. Then for any $t \in [t_0, t_1]$, (when $|t_1 - t_0|$ is sufficiently small), a straightforward application of Taylor's Theorem to the x component of $c(t)$, denoted as $c_x(t)$ would give,

$$c_x(t) = c_x(t_0) + E_x(t^*)$$

for some $t^* \in [t_0, t]$, where

$$E_x(t^*) = (t - t_0)c_x'(t^*),$$

with $E_x(t^*)$ playing the role of $r_1(x)$ above. Clearly, this can be done in each component. Then, since the final intent is to use the Euclidean norm on the vector-valued c, denoted as $\|c(t) - c(t_0)\|$, an elementary algebraic argument shows that the component-wise inequalities can be combined to yield

$$\|c(t) - c(t_0)\| \leq (t_1 - t_0)\sqrt{3}M_0.$$

It remains to show that $(t_1 - t_0)\sqrt{3}M_0 < \delta/2$. Note that t_0 and t_1 are values in the parametric domain of c, corresponding to the parametric points picked for uniform subdivison – namely each sub-interval will be halved at each iteration. Their maximum distance apart will span at most two sub-intervals, if chosen corresponding to sub-intervals over consecutive but different curve subsegments. Hence, choose ℓ_0 such that

$$2^{-\ell_0+1}\sqrt{3}M_0 < \delta/2.$$

or,

$$\ell_0 = \lceil \log_2(\frac{\sqrt{3}M_0}{\delta}) + 2\rceil. \quad (6)$$

Note that this analysis only applies to a single curve, and recall that a curve c can be composed of many Bézier sub-curves. Suppose there are j many sub-curves. Then, the Taylor's theorem analysis must be applied to each of the j-many sub-curves.

Condition 2. Guaranteeing Sufficiently Good Local Tangent Approximations: This is analagous to the preceding argument. Suppose the curvature is positive somewhere. If not, the curve is the trivial case of a straight line. Let M_1 denote the maximum value of $\|c''(t)\|$, and let μ_0 denote the minimum value of $\|c'(t)\|$. A similar application of Taylor's Theorem yields,

$$\|c'(t) - c'(t_0)\| \leq |t_1 - t_0| \|c''(t^*)\| \leq (t_1 - t_0) M_1.$$

Let θ_t denote the angle between $c'(t_0)$ and $c'(t)$. Then,

$$|sin(\theta_t)| \leq \frac{\|c'(t) - c'(t_0)\|}{\|c'(t)\|}.$$

For a sufficiently small value of ϵ chosen to be greater than 0, the arcsine function is monotonically increasing on $[-\epsilon/4, \epsilon/4]$. Therefore, showing that $sin(\theta_t) < sin(\epsilon/4)$ over that interval is sufficient to have $|\theta_t| < \epsilon/4$. The previous application of Taylor's Theorem gives

$$|sin(\theta_t)| \leq \frac{\|c'(t) - c'(t_0)\|}{\|c'(t)\|} \leq (t_1 - t_0) \frac{M_1}{\mu_0}.$$

Hence, it remains to show that $(t_1 - t_0) M_1/\mu_0 < sin(\epsilon/4)$. Therefore, choose ℓ_1 such that

$$2^{-\ell_1 + 1} \frac{M_1}{\mu_0} < sin(\epsilon/4).$$

or,

$$\ell_1 = \lceil \log_2(\frac{M_1}{\mu_0 sin(\epsilon/4)}) + 1 \rceil. \tag{7}$$

The above argument keeps track of the angle that the first derivative makes relative to the vector $c(t_0)$. If this angle for any two points on the curve whose parametric values are from consecutive subintervals is strictly less than $\pi/2$ radians, then this small angular measure precludes the possibility of there being a segment joining those points which are normal to the curve at both points on the curve. With these two conditions in place, the algorithm for estimating σ is now given in the next subsection.

Figure 3: The points $c(\tilde{s}_\sigma)$ and $c(\tilde{t}_\sigma)$ are ϵ-nearly doubly normal.

2.2 Lower bound algorithm for σ

The introduction, here, of the terminology "ϵ-nearly doubly normal" is similar to the conditions previously set forth for the seeds for Newton's Method, as expressed in Equations 1 and 2 in Section 1.

Let $c(s_\sigma)$ and $c(t_\sigma)$ be two distinct points of c such that $d(c(s_\sigma), c(t_\sigma)) = \sigma$. Consider those circumstances, where for sufficiently small positive ϵ there exist $\tilde{s}_\sigma, \tilde{t}_\sigma \in [0, 1]$ such that the lines L_1 and L_2 containing the normals at $c(\tilde{s}_\sigma)$ and $c(\tilde{t}_\sigma)$ intersect at a point ν near to the segment connecting $c(s_\sigma)$ and $c(t_\sigma)$ whenever the angle ϕ formed between L_1 and L_2 is between $\pi - \epsilon$ and π. An illustration is shown in Figure 3, where a and b denote the lengths along the indicated line segments.

Any two points $c(s)$ and $c(t)$ are said to be *ϵ-nearly doubly normal* if

$$(c(s) - c(t)) \cdot c'(s) = 0 \quad \& \quad (c(s) - c(t)) \cdot c'(t) = 0,$$

or

$$\pi - \epsilon < \phi < \pi.$$

The triangle inequality gives $d(c(\tilde{s}_\sigma), c(\tilde{t}_\sigma)) \leq a + b$, and that $a + b \leq \sigma$. The algorithm described will estimate the global separation distance using approximations of $d(c(\tilde{s}_\sigma), c(\tilde{t}_\sigma))$. Since $d(c(\tilde{s}_\sigma), c(\tilde{t}_\sigma)) \leq \sigma$, the estimate produced, denoted as $\sigma(\epsilon)$ (defined immedidately, below) will also be shown be no more than σ. The value $\sigma(\epsilon)$ (See Figure 4) is defined over any two ϵ-nearly normal points $c(t), c(s)$ with $t \neq s$,

$$\sigma(\epsilon) = min_{\{c(t),c(s)\}}\{d(c(t), c(s)).\}$$

The transition to providing an estimate of the more conservative value $\sigma(\epsilon)$ rather than trying to directly approximate σ is motivated by the following example. Let α be a planar C^∞ curve α containing an arc of the unit circle with arc-length strictly less than π, but where α has its minimum separation distance being much greater than 2 and found elsewhere on the curve. For any algorithm that attempts to approximate σ by focusing upon pairs of points that were nearly normal within some fixed tolerance, there would always be some input curve like α which would return some value near 2, since this arc-length can be made arbitrarily close to π.

Figure 4: The points $c(s)$ and $c(t)$ on the curve segments inside each cylinder are ϵ-nearly doubly normal, and D is the distance between the PL segments that approximates the curve segments.

Let D be defined as the distance between two PL segments that approximate the curve segments over which $\sigma(\epsilon)$ is realized. Note that $D \leq \sigma(\epsilon) + \delta$, since the radius of the cylinders shown in Figure 4 is at most $\delta/2$, as given previously by Taylor's analysis.

The value $\sigma(\epsilon)$ is now accepted as a good estimate of σ, and the focus shifts to approximating $\sigma(\epsilon)$, recalling that $\sigma(\epsilon) \leq \sigma$. Then, the algorithm below in Figure 5 will return an approximation $A(\epsilon)$ of $\sigma(\epsilon)$, with the following two guarantees:

- $A(\epsilon) \leq \sigma(\epsilon) \leq \sigma$, and
- $A(\epsilon) > (\sigma(\epsilon))/2$.

Note that the previous Taylor anlayses provides for a value ℓ that gives the following three guarantees:

- the length of each cylinder is strictly less than $\delta/2$,
- the radius of each cylinder is strictly less than $\delta/2$, and
- the angular deviation between tangents on the curve segments in each cylinder is strictly less than $\epsilon/4$.

Since the choice of a value for δ is not obviously *a priori*, it is reasonable to first set $\delta = 1$. The resultant estimate $A(\epsilon)$ is then tested for validity (See algorithm in Figure 5), and failure results in halving this value, repeating the

Global Separation Distance Estimate Algorithm

Input: A spline curve c & ϵ.
0. Initialize $\delta = 1$ and ω for upper precision bound.
1. Initialize $A(\epsilon) = 0$.
2. If $\sqrt{3}M_0/\delta > \omega/2$ or $M_1/(\mu_0 sin(\epsilon/4)) > \omega/2$ (Eqs. 6 & 7)
Terminate.
3. Use $1/\ell$ to uniformly partition $[0, 1]$ (Equation **??**).
4. Create PL approximation of c using partition of $[0, 1]$.
5. Find pw-distances, $d(e_i, e_j)$
with points p, q that realize $d(e_i, e_j)$.
6. If p and q are ϵ-nearly doubly normal
retain $d(e_i, e_j)$ for further consideration,
Else discard.
Let $D = min(d(e_i, e_j))$.
7. If $D \geq 4\delta$
$A(\epsilon) = D - 2\delta$
Else $\delta = \delta/2$, and Go to Step **1**.
Output: $A(\epsilon)$ = estimate for global separation distance.

Figure 5: General algorithm for estimating the global separation distance.

iterations until a valid value is obtained. In this way, the overall algorithm is logarithmic in $1/\sigma(\epsilon)$.

Termination and Satisfactory Value: The algorithm will terminate when $2\delta < \sigma(\epsilon)$, which will be logarithmic in $1/\sigma(\epsilon)$. Several applications of the triangle inequality in Figure 4 show that $D \geq \sigma(\epsilon) - 2\delta$, or equivalently $D + 2\delta \geq \sigma(\epsilon)$, yielding

$$\frac{D}{\sigma(\epsilon)} \geq \frac{D}{D + 2\delta} = \frac{1}{1 + \frac{2\delta}{D}} \geq \frac{1}{1 + \frac{2\delta}{4\delta}} = 2/3.$$

Hence, $D \geq (2/3)\sigma(\epsilon)$ and $D \leq \sigma(\epsilon) \leq \sigma$, as desired.

The global separation distance algorithm in Figure 5 assumes the existence of a geometric distance predicate $d(e_i, e_j)$ between two line segments, e_i and e_j, which returns:

- the distance $d(e_i, e_j)$ between the two line segments, and
- the points p and q on e_i and e_j, respectively, where that distance is realized.

Asymptotic Time Bound: The upper time bound for the global separation distance estimate algorithm is given quadratically in the bounds derived earlier for the Taylor's analysis. The final *a priori* bound is expressed in terms on the total number of subdivisions required, namely ℓ, as is standard practice [4]. For a curve c with j-many sub-curves, this entails $(j2^{2\ell})^2$ distance computations. However, since ℓ is logarithmic, this yields a tractable bound of $O(max\{\ j^2{M_0}^2/\delta^2,\ j^2{M_1}^2/{m_0}^2\ \})$. The asymptotic upper time bound for this algorithm is expressed as

$$O(max\{\ (\frac{jM_0}{\delta})^2\ log(1/\sigma(\epsilon)),\quad (\frac{jM_1}{m_0})^2\ log(1/\sigma(\epsilon))\ \}).$$

2.3 Example Analysis

For the curve already used, the values for the indicated parameters, above, were computed using the Maple symbolic software as

- $M_0 = 14.9$,
- $\mu = 3.4$,
- $M_1 = 21.9$.

These lead to a partition size of 4.9×10^5, when $\epsilon = 0.1$. This appears to be consistent with the experiments reported in Table 1, where approximately 2000 iterations produced an acceptable approximation, but without the additional numer bounds guaranteed here.

Acknowledgements: The authors were partially supported by NSF grants DMS-9985802, DMS-0138098, CCR-022654, CCR-0429477 and by an IBM Faculty Award. All statements here are the responsibility of the authors, *not* of the National Science Foundation *nor* of IBM. Both authors thank the Dagstuhl Seminar organizers and the Dagstuhl staff for providing the intellectually stimulating environment for refinement of these ideas, which were initially based upon the dissertation of E. L. F. Moore.

References

[1] J. Bisceglio. Personal communication. justin.bisceglio@gmail.com, October 2005. UConn department of CS&E alumni, Computer graphics engineer at Blue Sky Studios.

[2] R. Ellis and D. Gullick. *Calculus with Analytic Geometry*. Harcourt Brace Jovanovich, 3rd edition, 1986.

[3] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practicle Guide*. Academic Press, San Diego, CA, 2 edition, 1990.

[4] D. Lutterkort and J. Peters. Linear envelopes for uniform B–spline curves. In *Curves and Surfaces, St Malo*, pages 239–246, 2000. acceptance rate ca 75%.

[5] T. Maekawa and N. M. Patrikalakis. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer, New York, 2002.

[6] T. Maekawa, N. M. Patrikalakis, T. Sakkalis, and G. Yu. Analysis and applications of pipe surfaces. *Computer Aided Geometric Design*, 15:437–458, 1998.

[7] J. H. Mathews. *Numerical Methods for Computer Science, Engineering and Mathematics*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[8] E. L. F. Moore. *Computational Topology of Spline Curves for Geometric and Molecular Approximations*. PhD thesis, The University of Connecticut, 2006.

[9] E. L. F. Moore, T. J. Peters, and J. A. Roulier. Preserving computatational topology by subdivision of quadratic and cubic bézier curves. *dagstuhl*, to appear.

[10] L. Piegl and W. Tiller. *The NURBS Book*. Springer, 2nd edition, 1997.