Specifying Useful Error Bounds for Geometry Tools: An Intersector Exemplar

C. Mow [a], Thomas J. Peters [b,1], Neil F. Stewart [c]

[a] *Department of Computer Science, University of University of Connecticut, Storrs, CT 06269-3155, c_mow@hotmail.com*

[b] *Department of Computer Science & Engineering, and Department of Mathematics, University of Connecticut, Storrs, CT, 06269-3155, tpeters@cse.uconn.edu*

[c] *Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada, stewart@iro.umontreal.ca*

Abstract

The problem of geometric robustness is pervasive within CAGD. One aspect is to permit convenient user specification of error bounds, so as to ensure the *usefulness* of geometric models. Often, a useful specification requires an additional interface between the user and the geometric tool. As intersections of spline surface patches are fundamental within CAGD, we present a relation between model space and parameter space error bounds for an intersection algorithm as an exemplar of the additional interface needed for practical geometric tools. In particular, we consider the approximation of the intersection curve between two trimmed-surface patches. The Grandine-Klein intersector produces an approximation that is accurate to within a user-specified error bound, where that error bound is specified in parameter space. However, the end user is typically unaware of the details of this parametric domain, so selection of a parametric space error bound often relies upon heuristics. In this note our goal is to demonstrate how a user-specified error bound is made usable in practice through the straightforward application of the mathematical relation between model-space and parameter-space error bounds. The conversion of the model-space tolerance into a parameter-space tolerance is captured in a preprocessing interface to the intersection algorithm. The software implemented has proven to be reliable, efficient and user-friendly. It is based upon an elementary error analysis, which is also presented.

---

[1] All three authors acknowledge, with gratitude, funding in partial support of this work from the National Science Foundation, under grant Numbers DMS-9985802 and DMS-0138098. The research of the third author was also supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada. The views expressed herein are of the authors, not of these sponsors.

1 Introduction

Research into geometric robustness problems dates back over 20 years, as has been documented in the literature [6,9,12]. A central topic under geometric robustness has been intersections [1,8,10]. Geometric-modeling systems with trimmed-surface patches typically store [7] both geometric information (specifying the position of the patch in model space, usually R^3), and topological information (specifying the logical adjacencies of the patches). Unfortunately, tolerance values used in modeling systems to control geometric error are often poorly understood [4].

Grandine and Klein [5] have published an algorithm that computes the two parametric-domain representations of the intersection curve with a rigorously defined parameter-space error bound. The algorithm has been implemented as part of the DT_NURBS Spline Geometry Library [2] and in proprietary code [2]. In this note we discuss our experience in implementing and testing a new interface to the Grandine-Klein (G-K) intersector that relates the error in model space to the parametric error bound. The underlying error analysis, based on standard theorems, is also presented. Our goal is to provide an example of the types of interfaces between users and geometric tools that will be needed to make the most efficient use of CAGD methods.

Following the notation of Grandine and Klein [5, Sec. 3], one parametric surface, denoted F, is parametrized by $(u, v) \in [0, 1]^2$, and the other, denoted G, is parametrized by $(s, t) \in [0, 1]^2$. The surfaces F and G will typically be non-uniform rational B-splines (NURBS). The exact intersection curve is given by a mapping from $[0, 1] \rightarrow [0, 1]^4$ (with components u, v, s and t) such that The G-K intersector creates mappings from $[0, 1] \rightarrow [0, 1]^4$ (with components u, v, s and t) such that

$$F(u(\tau), v(\tau)) = G(s(\tau), t(\tau)).$$

The G-K intersector creates approximations [3] of the intersection curve as $[\tilde{u}(\tau), \tilde{v}(\tau)]$ and $[\tilde{s}(\tau), \tilde{t}(\tau)]$ in the parametric domains; their respective images $F(\tilde{u}(\tau), \tilde{v}(\tau))$ and $G(\tilde{s}(\tau), \tilde{t}(\tau))$ in model space will usually not agree. The G-K algorithm provides bounds on the errors $[u(\tau), v(\tau)] - [\tilde{u}(\tau), \tilde{v}(\tau)]$ and $[s(\tau), t(\tau)] - [\tilde{s}(\tau), \tilde{t}(\tau)]$, but not on either of the errors in model space, given as

$$F(u(\tau), v(\tau)) - F(\tilde{u}(\tau), \tilde{v}(\tau)) \quad \text{and} \quad G(s(\tau), t(\tau)) - G(\tilde{s}(\tau), \tilde{t}(\tau)).$$

[2] A proprietary Boeing implementation was used to generate the experimental software results here.
[3] Such dual approximations are typical.

The interface supplied by Grandine and Klein may be summarized as follows. Let S_1 and S_2 be the input surfaces and let ϵ be the user-specified error bound in model space. The output is a representation of the intersection set, within parameter spaces, with its error bounded above by ϵ. The new pre-processing interface developed is similar, except that now the user-specified error bound is given in model space by γ. Clearly, the key link is software that converts a given model-space bound γ into its corresponding parameter-space bound ϵ, consistent with the respective definitions of F and G. The error analysis (based upon standard methods) is given in Section 2.

2 The Error Analysis

We now restrict our attention to the surface $F(u, v)$. An exactly analogous analysis applies to $G(s, t)$. For brevity, we write

$$[u_1, v_1] = [u(\tau), v(\tau)] \quad \text{and} \quad [u_0, v_0] = [\tilde{u}(\tau), \tilde{v}(\tau)].$$

Then, Taylor's theorem in two dimensions [3, p. 200] states that each component of $F(u_1, v_1)$ can be written

$$F(u_1, v_1) = F(u_0, v_0) + R(u^*, v^*)$$

where

$$R(u, v) = (u_1 - u_0)\frac{\partial F}{\partial u}(u, v) + (v_1 - v_0)\frac{\partial F}{\partial v}(u, v)$$

is evaluated at some point $[u^*, v^*]$ on the line segment joining $[u_0, v_0]$ and $[u_1, v_1]$. Suppose that, using the bound in parameter space for the G-K intersector, we can write

$$|u_1 - u_0| \leq \epsilon, \qquad |v_1 - v_0| \leq \epsilon.$$

Then it follows, with $\|\cdot\|$ being any convenient vector norm, that

$$\|F(u_1, v_1) - F(u_0, v_0)\| \leq \epsilon M \tag{1}$$

for any M satisfying

$$\|\frac{\partial F}{\partial u}(u^*, v^*)\| + \|\frac{\partial F}{\partial v}(u^*, v^*)\| \leq M. \tag{2}$$

For the given surface, let γ be an upper bound for the acceptable error in model space. In order to guarantee that this error is sufficiently small, we will require that

$$\epsilon M \leq \gamma.$$

Then software to obtain an upper bound for M can be implemented using any standard technique for obtaining the maximums of the functions indicated in the left hand side of Inequality (2).

Using the triangle inequality and Inequality (1), applied to each surface, it is then easy to derive an upper bound for the distance between corresponding points on the approximated boundary curves in model space, as

$$F(\tilde{u}(\tau), \tilde{v}(\tau)) - G(\tilde{s}(\tau), \tilde{t}(\tau))\| \leq \gamma(F) + \gamma(G),$$

where $\gamma(F)$ denotes any upper bound for the left side of Inequality (1) (similarly $\gamma(G)$).

In our implementation we used standard properties of spline basis functions (in particular, "partition of unity" [11]) to bound the norm of each of the two partial derivatives of F. These bounds permit calculation of M by means of $O(mn)$ arithmetic operations, where m and n are the dimensions of the control-point array. In principle, the use of this upper bound to calculate ϵ could result in performance degradation of the intersection algorithm. However, in practice this cost is negligible: in our experiments there was no noticeable performance degradation versus running the G-K intersector directly with its default parameter-space error bound. On the other hand, there is considerable advantage provided by this new interface for obtaining acceptable intersections on the first try, rather than by trial-and-error iteration.

There is, of course, a theoretical risk that the calculated bound M is unduly pessimistic, specifically when the norms of the partial derivatives are relatively large in a part of the parametric domain that is far from the trimmed patch. This problem could be avoided by finding a tighter bound in the neighborhood of the trimmed patch, for example, by surface subdivision, but in our experimentation this was not necessary. Note that both the parameter-space and the model-space error bounds are global over the entire intersection set. Hence, issues of subdivision are appropriately left to the end user and are beyond the scope of the results we present here. A representative test case is summarized in Table 1 for the intersection of a plane with the extrusion of a spiral curve having varying curvature.

Table 1

Upper Bounds and Maximum Measured Values

ϵ	$\gamma(F) + \gamma(G)$	GK_m
10^{-3}	$3.03 * 10^{-1}$	$1.64 * 10^{-2}$
10^{-4}	$3.03 * 10^{-2}$	$2.34 * 10^{-3}$
10^{-5}	$3.03 * 10^{-3}$	$3.31 * 10^{-4}$
10^{-6}	$3.03 * 10^{-4}$	$3.67 * 10^{-5}$

3 Conclusion

We have presented an analysis and a user interface to allow for user specification of model-space error bounds on the output of an intersector algorithm. This approach has an algorithmic conversion of the specified model-space error bound into a corresponding parameter-space error bound, which is already expected as input to the intersection algorithm. However, this automatic conversion then makes the intersector more usable in practice, because the required parameter-space bound varies with input surface characteristics, as this presentation shows. This interface then allows the user to specify the final desired model space bound independently of any *a priori* analysis of the input surfaces. As intersection is fundamental and typical, this note can be used as an exemplar for similar analyses and interfaces for many geometric algorithms. Our reliance upon Taylor's Theorem is basic, fundamental and extensible. Indeed, as the underlying mathematics is elementary, there appears to be little justification for any geometric implementation to fail to incorporate such user-friendly interfaces.

4 Acknowledgements

The authors thank T. A. Grandine, J. Keiter, F. W. Klein IV, D. R. Ferguson, S. W. Lee and anonymous referees for various technical comments. Any errors or omissions are the sole responsibility of the authors.

References

[1] Barnhill, R. E., Farin, G., Jordan, M. and Piper, B. R., *Surface/surface intersection*, *Computer Aided Geometric Design*, 4(1-2):3–16, July 1987.

[2] DT_NURBS library. URL *http://ocean.dt.navy.mil/dtnurbs/*

[3] Buck, R. C. Advanced Calculus, McGraw-Hill, New York, 1956.

[4] Ferguson, D. R., Lucian, M. L., and Seitleman, L. H., PDES Inc. Geometric Accuracy Team, Interim Report, PDES, Inc., SCRA, Charleston, SC, 1996.

[5] Grandine, T. A. and Klein, F. W. A new approach to the surface intersection problem. *Computer Aided Geometric Design* (14), 111-134, 1997.

[6] Hoffmann, C. M., *How solid is solid modeling?*, ed M. Lin and D. Manocha, LNCS State-of-the-Art-Survey, Springer Verlag, 1-8, 1996.

[7] Hoffmann, C. M., Geometric and Solid Modeling, An Introduction, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1989.

[8] Hoschek, J. and Lasser, D., Fundamentals of Computer Aided Geometric Design, A. K. Peters, Wellesley, MA, 1993, Translated by L. L. Schumaker.

[9] Peters, T. J., Rosen, D. W., and Dorney, S. M., *The diversity of topological applications within computer aided geometric design*, Annals of the New York Academy of Science, 728, 198-209, 1994.

[10] Patrikalakis, N. M., *Surface-to-surface intersections*, IEEE Computer Graphics and Applications, 13(1):89–95, January 1993.

[11] Piegl, L. A. and Tiller, W. The NURBS Book, Springer Verlag, 1997.

[12] Requicha, A. A. G., *Representations for rigid solids: theory, method and systems*, Lecture Notes in Computer Science, ACM Computing Surveys, 12(4), Springer-Verlag, 437 - 464, 1980.