ACM Symposium on Solid Modeling and Applications (2004)
G. Elber, N. Patrikalakis, P. Brunet (Editors)

Tracing Surface Intersections with Validated ODE System Solver

H. Mukundan, K. H. Ko, T. Maekawa, T. Sakkalis, N. M. Patrikalakis

Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA

Abstract

This paper presents a robust method for tracing intersection curve segments between continuous rational parametric surfaces, typically rational polynomial parametric surface patches. The tracing procedure is based on a validated ordinary differential equation (ODE) system solver which can be applied, without substantial overhead, for transversal as well as tangential intersections. Application of the validated ODE solver in the context of eliminating the phenomenon of straying and looping is discussed. In addition, we develop a method to fulfill the condition of a continuous gap-free boundary with a definite numerically verified upper bound for the intersection curve error in parameter space and is further mapped to an upper bound for the intersection curve error in 3D model space, which assists in defining well-formed boundary representation models of complex 3D solids.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Solid Modeling]: CAD, CAGD, tangency, rounded interval arithmetic, robustness, rigorous error bounds, straying and looping, boundary representation, gap-free boundaries

1. Introduction

Intersection is a fundamental process needed to build and interrogate complex CAD models. It is needed in representing complex objects, in finite element discretization, computer animation, feature recognition and manufacturing simulation. It is also used in scientific visualization for implicitly defined objects and for contouring multivariate functions that represent some properties of a system.

There has been extensive research to solve the surface intersection problem. Solution methods can be broadly classified into lattice methods, marching methods and subdivision based methods [PM02b, PM02a, HL93]. The *Interval Projected Polyhedron* (IPP) algorithm [HMPY97] using subdivision techniques coupled with interval arithmetic can exhaustively and robustly find all roots. However, a *topology resolution* of the roots based on adjacency information is complicated [HMPY97]. In addition, the algorithm tends to be extremely time consuming in the case of tangential, as well as higher order intersections. There is also no guarantee that intervals generated do contain a root [HMSP96], an inherent problem associated with any subdivision algorithm.

It is well known that the problem of surface intersection can be reduced to solving an initial value problem for ordinary differential equations (ODE). Conventional algorithms for solving a system of ODEs, like the Runge-Kutta method or the Adams-Bashforth method [PTVF88], compute an estimate for a solution and its error. The user cannot find out how accurate the estimated answer is without extensive error analysis. Moreover there are cases where completely catastrophic answers are returned without any warning on using the conventional numerical methods. This lack of robustness in numerical computation can cause undesirable changes of the topology of intersection as shown by Hu *et al.* [HPY96]. Grandine and Klein [GK97] formulate the intersection problem as a differential algebraic equation which can be solved as a *boundary value problem* instead of an *initial value problem*. The tracing of the intersection curve is based on approximation methods. Moreover the algorithm has a hard time dealing with tangential intersections.

Schemes based on interval arithmetic take into account three sources of errors in the numerical computation of solution to ODEs: (1) propagation of *error in initial data*, (2) *truncation error* caused by truncating infinite sequences of arithmetic operations after a finite number of steps and (3) *round off errors* inherent to computation in floating point arithmetic [Han69]. When correctly used, interval methods can compute bounds in which the correct answer is guaranteed to be enclosed [CR96].

Robust tracing of an intersection curve segment can be done once we identify each component and further obtain starting and ending points. The focus of the paper is to investigate a robust marching method which will produce a continuous guaranteed bound on the error at each point on the intersection using a *validated ODE solving scheme*. Thus we assume that we are given an intersection curve segment, with strict bounds on the starting point. We also apply the interval versions of the ODEs to robustly trace a tangential intersection problem. We also relate the phenomenon of *straying* and *looping* to the criterion of a step size control based on the validation procedure in the method. The key contribution of this paper is to obtain strict model space bounds for the intersection curve segment for both transversal and tangential intersection cases.

The paper is structured as follows: In Section 2 we obtain the governing interval ODEs for the various intersection cases. The limitations of conventional nonlinear ODE solvers are discussed in Section 3 and we describe and introduce the concept of a *validated ODE solver* [Eij81, NJC99], its application in tracing surface intersection and discuss its use in preventing straying or looping and in resolving singularities. Section 4 deals with the calculation of the model space error bound. We perform various examples and tests using a prototype implementation of the algorithm in Section 5. Section 6 concludes this paper with a review of the possible applications and issues.

2. Tracing a Surface to Surface Intersection

The intersection of two interval parametric surfaces $[\mathbf{P}](\sigma,t)$ and $[\mathbf{Q}](u,v)$ can be described as an interval vector equation given by,

$$[\mathbf{P}](\sigma,t) = [\mathbf{Q}](u,v). \tag{1}$$

We can formulate it as a system of *ordinary differential equations*(ODE) which are arc length parametrized. Our approach is to use a marching scheme to find out the curve of intersection by solving this system of ODEs as discussed by Hu *et al.* [HMPY97],

$$\sigma' = \frac{d\sigma}{ds} = \frac{Det([\mathbf{c}],[\mathbf{P}_t],[\mathbf{N^P}])}{[\mathbf{N^P}]\cdot[\mathbf{N^P}]},\; t' = \frac{dt}{ds} = \frac{Det([\mathbf{P}_\sigma],[\mathbf{c}],[\mathbf{N^P}])}{[\mathbf{N^P}]\cdot[\mathbf{N^P}]},$$
$$u' = \frac{du}{ds} = \frac{Det([\mathbf{c}],[\mathbf{Q}_v],[\mathbf{N^Q}])}{[\mathbf{N^Q}]\cdot[\mathbf{N^Q}]},\; v' = \frac{dv}{ds} = \frac{Det([\mathbf{Q}_u],[\mathbf{c}],[\mathbf{N^Q}])}{[\mathbf{N^Q}]\cdot[\mathbf{N^Q}]}, \tag{2}$$

where *Det* denotes the determinant and,

$$[\mathbf{N^P}] = [\mathbf{P}_\sigma]\times[\mathbf{P}_t], \qquad [\mathbf{N^Q}] = [\mathbf{Q}_u]\times[\mathbf{Q}_v],$$

are the normal vectors of $[\mathbf{P}]$ and $[\mathbf{Q}]$ respectively. $[\mathbf{c}]$ is the marching direction and is obtained for various cases as shown below and s is the arc length parameter.

Equations (2) are true for any surface-surface intersection involving parametrically defined surfaces, provided we correctly represent the marching direction (tangent to the intersection curve), and the surfaces. Based on the intersection type the marching direction has to be computed differently.

2.1. Transversal Intersection

For a transversal intersection, the direction of marching $[\mathbf{c}]$, is perpendicular to the normal vectors of both surfaces, thus this direction can be obtained as follows [HMPY97]:

$$[\mathbf{c}] = \pm\frac{[\mathbf{N^P}]\times[\mathbf{N^Q}]}{|[\mathbf{N^P}]\times[\mathbf{N^Q}]|}. \tag{3}$$

2.2. Tangential Intersection

The method to obtain the marching direction for tangential intersection is based on Ye and Maekawa [YM99] and obtaining an interval version of it is further discussed by Mukundan *et al.* [MKM*03]. Note that we cannot use equation (3) to obtain the marching direction because normals to both surfaces are parallel. From the condition of equal normal curvatures of both the surfaces a point on the intersection, we obtain a quadratic equation of the form,

$$[b_{11}](\sigma')^2 + 2[b_{12}](\sigma')(t') + [b_{22}](t')^2 = 0, \tag{4}$$

where the interval coefficients $[b_{11}]$, $[b_{12}]$ and $[b_{22}]$ are the functions of the first and second fundamental form coefficients of the given surfaces. Details are given in [MKM*03]. There are four distinct cases to the solution of (4) depending upon the discriminant $([d] = [b_{12}]^2 - [b_{11}][b_{22}])$.

- $(\overline{d}$, the upper bound of $d < 0)$: The surfaces have an isolated tangential contact point.
- $(\underline{d}$, the lower bound of $d > 0)$: We have the phenomenon of branching, i.e. $[\mathbf{c}]$ is not uniquely defined.
- $(0 \in [d]$ and $0 \in [b_{11}],[b_{12}],[b_{22}])$: The intersection of surfaces $[\mathbf{P}]$ and $[\mathbf{Q}]$ cannot be evaluated by this method or they have a contact of at least second order (i.e., curvature continuous).
- $(0 \in [d]$ and $0 \notin [b_{11}]^2 + [b_{12}]^2 + [b_{22}]^2)$: The marching direction vector is defined. Thus, $[\mathbf{P}]$ and $[\mathbf{Q}]$ are said to intersect tangentially at the neighborhood.

The marching direction is obtained, depending on $[b_{11}]$, $[b_{12}]$ and $[b_{22}]$, as follows.

If $0 \notin [b_{11}]$, then the marching direction is given by,

$$[\mathbf{c}] = \frac{[\nu][\mathbf{P}_\sigma]+[\mathbf{P}_t]}{|[\nu][\mathbf{P}_\sigma]+[\mathbf{P}_t]|}, \text{where } \frac{\sigma'}{t'} = [\nu] = -\frac{[b_{12}]}{[b_{11}]}. \tag{5}$$

If $0 \in [b_{11}]$ and $0 \notin [b_{22}]$, then the marching direction is given by,

$$[\mathbf{c}] = \frac{[\mathbf{P}_\sigma]+[\mu][\mathbf{P}_t]}{|[\mathbf{P}_\sigma]+[\mu][\mathbf{P}_t]|}, \text{where } \frac{t'}{\sigma'} = [\mu] = -\frac{[b_{12}]}{[b_{22}]}. \tag{6}$$

The system of interval ordinary differential equations (2) represent an autonomous initial-value problem (IVP), which can be rewritten in vector form as,

$$\mathbf{y}'(s) = \begin{bmatrix} \sigma' & t' & u' & v' \end{bmatrix}^T = \mathbf{f}([\mathbf{y}(s)]),$$
$$[\mathbf{y}_0] = \begin{bmatrix} [\sigma_0] & [t_0] & [u_0] & [v_0] \end{bmatrix}^T.$$

We can assume at this point that $\mathbf{f}$ has $k-1$ derivatives, where we define k in Section 3.1.1. The intersection curve segements are computed by solving the initial value problem for a system of interval nonlinear ordinary differential equations (ODE).

It has been shown by Hu *et al.* that for a polynomial surface patch, if there is a tangential contact curve of two polynomials, then it must start from a border and end at a border unless it is a loop. One way to determine if we need to use ODEs for transversal or for tangential intersection is to check if $0 \in |[\mathbf{N^P}]\times[\mathbf{N^Q}]|$. If this criterion is valid then the given branch is a tangential contact curve. At this point we note that such a topological configuration is not the focus of our paper. Our focus is on accurately tracing and finding a validated error bound in 3D model space, given the kind of intersection and a bound on the starting point.

3. Validated ODE Solver in Surface Intersection

3.1. Concept of Validated ODE Solver

A validated interval scheme for ODEs not only produces a guaranteed error bound for the true solution, but also verifies the existence and the uniqueness of the solution for the ODE system within that bound [Moo66, Loh92, Ned99]. Such a scheme is usually performed in two phases [Moo66, NJC99].

3.1.1. Phase I Algorithm

This phase in a validated solving scheme for ODEs involves:

- Choosing an *a priori* bound and a step size based on validation criterion.
- Checking the existence and uniqueness within the *a priori enclosure* for the above step size.

Thus the goal is to compute enclosures $[\tilde{\mathbf{y}}_j]$ on the family of the solutions $\mathbf{y}(s; s_0, [\mathbf{y}_0])$ for the IVP corresponding to the intersection under consideration

$$\mathbf{y}(s; s_j, [\mathbf{y}_j]) \subseteq [\tilde{\mathbf{y}}_j], \;\forall\; s \in [s_j, s_{j+1}],$$

where $\mathbf{y}(s; s_j, [\mathbf{y}_j])$ represents the family of curves passing through $[\mathbf{y}_j]$ satisfying equation (2) and s is the independent variable which in our case is the arc length parameter. We call such a bound $[\tilde{\mathbf{y}}_j]$, an *a priori enclosure*, and try to obtain this bound for the j^{th} step $h_j = (s_{j+1} - s_j)$. For validating the solution for a pair of the step size and an *a priori enclosure*, we can use various methods like the *constant enclosure method* [Eij81], the *polynomial enclosure method* [Loh95] or the *Taylor series method* [CR96]. The Taylor series method is preferred to a constant step size method since it can allow for longer step sizes [Ned99] and can be written as follows:

$$[\tilde{\mathbf{y}}_j(s)] \;\supseteq\; [\mathbf{y}_j] + \sum_{i=1}^{k-1} [\mathbf{y}_j]_i (s - s_j)^i + [\tilde{\mathbf{y}}_j]_k (s - s_j)^k, \qquad (7)$$

where k is the order of the Taylor series used and $[\mathbf{y}_j]_i$ is the i^{th} Taylor coefficient evaluated at $[\mathbf{y}_j]$. We numerically solve for the corrected step size h_j, given an initial guess for an *a priori enclosure* as shown by Nedialkov [Ned99]. At this point we have made an assumption that $\mathbf{f}([\mathbf{y}(s)])$ is well behaved and is C^k continuous.

3.1.2. Phase II Algorithm

Phase II of a validated solution scheme for ODEs involves:

- Propagation of the solution.
- Reducing the phenomenon of wrapping.

Using the *a priori enclosure* $[\tilde{\mathbf{y}}_j]$ from *phase I algorithm*, *phase II algorithm* computes a tighter enclosure $[\mathbf{y}_{j+1}]$ at s_{j+1} using an interval version of the Taylor series formula coupled with the application of the mean-value theorem [NJC99]. Taylor series coefficients and their Jacobian are robustly computed with a technique called *automatic differentiation* [Moo66, Sta97].

The main difficulty we face in *phase II algorithm* is the *wrapping effect* [KLF01]. The most promising solution to the wrapping effect is a *QR factorization method* developed by Löhner [Loh92]. By limiting wrapping we prevent the exponential growth in the width of the interval solution at s_{j+1}.

3.2. Formulation Based on Validated ODE Solver

We solve the ODEs given by the equation (2) using a validated ODE solver given initial conditions. Our use of rational polynomial parametric surfaces which are C^∞ continuous makes sure that $\mathbf{f}([\mathbf{y}(s)])$ is well behaved and is at least C^k continuous.

Phase I algorithm verifies the existence and uniqueness of the intersection curve segment, and a successful validation results in a step size h_j and a corresponding *a priori enclosure* $[\tilde{\mathbf{y}}_j]$, which in the context of surface intersection is,

$$[\tilde{\mathbf{y}}_j] \equiv \begin{bmatrix} [\tilde{\sigma}] & [\tilde{t}] & [\tilde{u}] & [\tilde{v}] \end{bmatrix}^T.$$

Phase II algorithm now finds a tight estimate of the bound on the parameter for a specific s_{j+1},

$$[\mathbf{y}_{j+1}] \equiv \begin{bmatrix} [\sigma_{j+1}] & [t_{j+1}] & [u_{j+1}] & [v_{j+1}] \end{bmatrix}^T.$$

This tighter bound acts as the initial condition for the next step, and hence helps in marching along the intersection curve, without significant increase of the error in the evaluation of the intersection curve segment. The intersection curve is obtained as a series of connected *a priori enclosures* (boxes) in the parameter space, which encloses the exact curve of intersection in the parameter space as shown in Figure 2 [MKM*03].

3.3. Resolving Singularities and Preventing Straying or Looping

Any numerical scheme, yielding a solution for a physical system represented by an IVP should first check for the *existence* and the *uniqueness* of the solution before returning an approximation, or a bound for it [Kre94]. This, however is not a common practice in the conventional solution schemes for IVPs. A typical solution procedure is to use an approximate, point based algorithm [PTVF88] like Runge-Kutta method, Taylor series method or Adams-Bashforth technique for solving the ODEs corresponding to the surface to surface intersection problem at discrete values of the arc length parameter s as mentioned in [PM02b, PM02a]. These methods are usually robust and reliable for most applications, but it is easy to find examples for which they return inaccurate results [PMKM04], especially in the presence of closely spaced features as shown in Figure 1. This is because the algorithms to control the step size are based on controlling just the error alone. As mentioned previously,

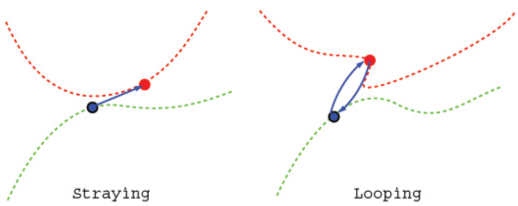


Figure 1: *Phenomenon of straying or looping.*

a validated ODE solver verifies the *existence* and *uniqueness* of the solution for the ODE system within the *a priori* bound before determining the step size. We employ this idea to successfully resolve the cases involving singularities in parameter space where the criterion of existence and uniqueness is not satisfied. Also validation before tracing can prevent the phenomenon of straying or looping even when intersection curve segments come quite close together, thereby tracing the correct intersection curve segment.

If the solution exists and is unique for a given step size h_j and an *a priori enclosure* $[\tilde{\mathbf{y}}_j]$, the criterion (7) based on Taylor series holds [Ned99]. Without loss of generality, we consider the case of $k = 1$, namely, the *constant enclosure method* [Ned99]:

$$[\tilde{\mathbf{y}}_j(s)] \;\supseteq\; [\mathbf{y}_j] + \mathbf{f}([\tilde{\mathbf{y}}_j]) h_j. \qquad (8)$$

Let us assume that the surfaces $[\mathbf{P}](\sigma,t)$ and $[\mathbf{Q}](u,v)$ intersect transversally in such a way that they have two distinct branches and that these branches lie close to each other in a given region in the parameter space. For such regions the denominator of equation (3) $|[\mathbf{N^P}]\times[\mathbf{N^Q}]|$ gives a value close to 0 and in the event of both curve segments intersecting each other, it contains 0. The evaluation of $\mathbf{f}([\tilde{\mathbf{y}}_j])$ based on the equation (2) blows up, returning a smaller and smaller step size and correspondingly smaller $[\tilde{\mathbf{y}}_j]$ to satisfy the criterion (8). In the event of $0 \in |[\mathbf{N^P}]\times[\mathbf{N^Q}]|$, the criterion is never satisfied, and this condition is reported as a singularity.

This validated step size strategy can hence prevent straying or looping and successfully resolve the singularities of intersection curve segments. Also note that when the step size obtained from the algorithm falls below a pre-specified h_{min}, the algorithm warns that the two curves are closer than they can possibly be resolved. An example that fully demonstrates this capability of the validated ODE solver is performed in Section 5.

4. Model Space Error Bound

The significance of the *a priori enclosure* in interval analysis has been limited as a way to enclose the truncation error term in the Taylor formula for obtaining each successive step, thus providing a method for obtaining a bound for the solution to the ODE system at s_{j+1}. We realize that the *a priori enclosure* $[\tilde{\mathbf{y}}_j]$ actually bounds the solution $\mathbf{y}(s;s_j,[\mathbf{y}_j])$. The series of *a priori enclosures* in the parameter space is mapped to the 3D model space to enclose the true curve of intersection and to provide a *model space error bound*.

To prove the existence of a gap-free or continuous bound enclosing the true curve of intersection in the 3D model space, we develop the following theorem.

Theorem 1 Let $[\sigma(s)]$ and $[t(s)]$ be mappings defined by,

$$([\sigma(s)],[t(s)]) : \mathbf{IR} \rightarrow \mathbf{IR}^2,$$

such that they are continuous in $s \in [s_0, s_{end}]$. If $[\mathbf{P}](\sigma(s),t(s))$ is a *continuous rational interval function* defined by,

$$[\mathbf{P}](\sigma,t) : \mathbf{IR^2} \rightarrow \mathbf{IR^3},$$

then the mapping $[\mathbf{P}](s) = [\mathbf{P}](\sigma(s),t(s)) : \mathbf{IR} \rightarrow \mathbf{IR^3}$ is continuous in $\mathbf{IR^3}$ for $s \in [s_0, s_{end}]$. Here $\mathbf{IR}$ denotes a set of interval numbers.

A similar theorem can be developed for $[\mathbf{Q}](u(s),v(s))$.

The proof directly follows from the continuity of *rational interval functions* proved by Moore [Moo66]. Majority of mapping in CAD practice including polynomials is continuous and rational and hence we realize the goal of a *continuous gap-free bound* on the curve of intersection in 3D model space, given continuous bounds on its pre-image.

5. Examples

We performed examples using the component packages [Pro, Fad, Vno] and the Design Laboratory interval library at MIT. All computation was performed on a PC running at 1.4 GHz with 512MB of RAM under Linux.

Figure 3 illustrates the intersection of two interval bicubic Bézier surfaces $[\mathbf{P}_1](\sigma,t)$ and $[\mathbf{Q}_1](u,v)$. The solution is obtained by the validated ODE solver and mapped into 3D model space. Figure

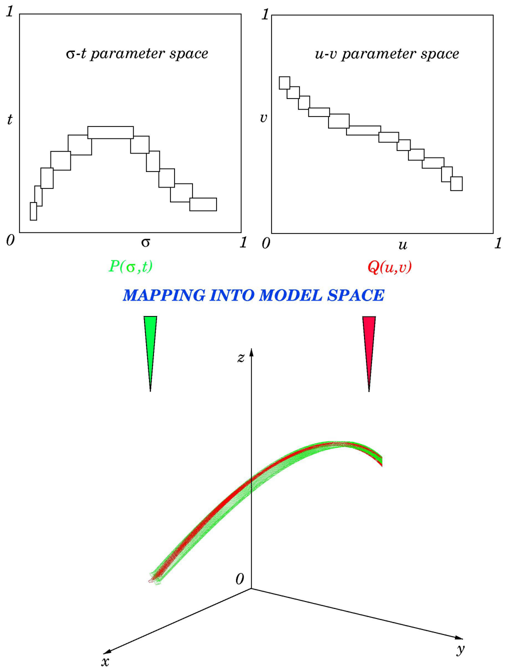


Figure 2: *Mapping of the pre-image of the intersection curve segment from the parameter space to the 3D model space. Note that the boxes obtained in the parameter space of each of the surface is continuous, gap free and ordered.*

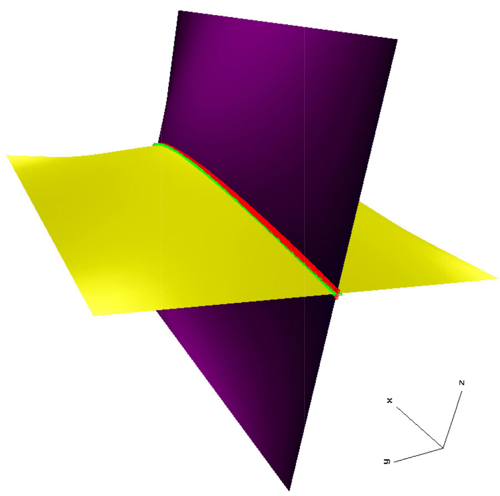


Figure 3: *Transversal intersection of two bicubic-Bézier surfaces corresponding to a maximum relative model space error of 0.00350.*

4 shows the tangential intersection of two interval cubic-quadratic Bézier surfaces $[\mathbf{P}_2](\sigma,t)$ and $[\mathbf{Q}_2](u,v)$. The surfaces are placed such that the intersection is tangential. The pre-image of the curve of intersection is obtained using the validated ODE solver and mapped into 3D model space.

Figure 5 shows an example constructed very similarly to the one

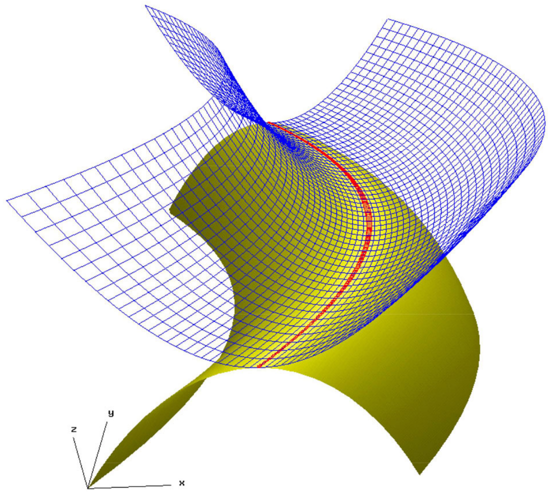
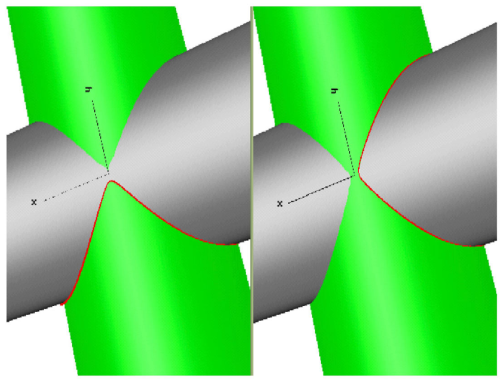


Figure 4: *Tangential intersection of two cubic-quadratic Bézier patches for a maximum relative model space error = 0.0050.*

Test No.	Perturbation of $[\mathbf{Q}_3]$ in z-direction in model space	Steps needed for resolving singularity
1	*+0.03*	*845*
2	*+0.0003*	*989*
3	*+0.000003*	*1139*
4	*0.0*	*singularity reported*
5	*-0.000003*	*1303*
6	*-0.0003*	*1153*
7	*-0.03*	*1007*

Table 1: *Resolving singularities of the curve of intersection.*



(a) (b)

Figure 5: *Figure (a) shows the surface* $[\mathbf{Q}_3](u,v)$ *perturbed along the positive z-direction, the intersection curve segment is correctly traced by the validated ODE solver. Figure (b) in a similar way illustrates how the validated ODE solver successfully trace the correct intersection curve segment when the perturbation is in the negative z-direction.*

used in Hu *et al.* [HPY96]. The surfaces are modeled such that when an interval bicubic Bézier surface $[\mathbf{P}_3](\sigma,t)$ intersects an interval cubic-quadratic Bézier surface $[\mathbf{Q}_3](u,v)$, then there is a singularity in the intersection curve (curve is connected). The pre-image of the curve of intersection is obtained using a validated ODE solver and mapped into 3D model space.

Marching schemes based on the direct application of floating point arithmetic may cause the curve to have conflicting topological structure from the real curve [HPY96]. These violations are manifested as gaps or as inappropriate intersections. This example illustrates that, a validated ODE solver should be able to resolve the singularity of the intersection curve and report to the user. We have performed experiments where we perturb one of the surfaces ($[\mathbf{Q}_3](u,v)$) in the z-direction so that the curve of intersection is free of singularity. A positive perturbation in the z-direction will lead to the branching of the curve in a sense different from a negative perturbation in the z-direction. Table 1 compares the number of steps needed to resolve a possible candidate for looping or straying, vs. perturbation of the Bézier patches. Note that when the perturbations are small we need more steps for resolving.

6. Conclusions

The condition of a continuous gap-free boundary with a numerically verified upper bound for the intersection curve error is fulfilled. We map this error in parameter space bound to 3D model space bounds conservatively. Thus this definite upper bound for error helps in defining well-formed boundary representation of complex 3D solids. Validated error bounds for surface intersection is essential in the interval boundary representation for consistent solid models as shown by Sakkalis *et. al.* [SSP01]. Also discussed is our ability to resolve *straying* or *looping* by a validated ODE solver which adapts its own step size to verify the existence and uniqueness. This technique is applied to both transversal and tangential intersections.

It was noticed that by following Horner's rule while we input the expression for $[\mathbf{f}(\mathbf{y}(s))]$ we are not only able to improve on the speed but also on the size of the interval. Further the large amount of data we have obtained could be reduced by the use of an approximation scheme proposed by Tuohy *et al.* [TMSP97].

This algorithm might prove to be costlier compared to a conventional algorithms for simple problems. Also unfortunate is the inherent problem of very small but nonzero increase in the width of the interval solutions due to rounding, but the quality of the solution far outweighs the cost factor specially for complicated intersection problems.

Future work on the topic of interval solid modeling could be on how to reduce the width of the *a priori enclosures*, there by reducing the error bounds.

Acknowledgements

We thank C. M. Hoffmann, N. S. Nedialkov, T. J. Peters, N. F. Stewart for useful discussions and their comments and the reviewers for their careful reviews and suggestions. This work was funded in part by the NSF (grants No. DMS-0138098 and CCR-0231511).

References

[CR96] CORLISS G. F., RIHM R.: Validating an a priori enclosure using high-order Taylor series. In *Scientific Computing and Validated Numerics: Proceedings of the International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics - SCAN '95*, Alefeld G., Frommer A.„ Lang B., (Eds.). Akademie Verlag, Berlin, 1996, pp. 228–238. 1, 3

[Eij81] EIJGENRAAM P.: *The Solution of Initial Value Problems Using Interval Arithmetic.* Mathematical Centre Tracts No. 144., Stichting Mathematisch Centrum, Amsterdam, 1981. 2, 3

[Fad] FADBAD/TADIFF, A C++ package for automatic differentiation. *http://www.imm.dtu.dk/fadbad.html/*. 4

[GK97] GRANDINE T. A., KLEIN F. W.: A new approach to the surface intersection problem. *Computer Aided Geometric Design 14*, 2 (1997), 111–134. 1

[Han69] HANSEN E. (Ed.): *Topics in Interval Analysis*. Oxford University Press, 1969. 1

[HL93] HOSCHEK J., LASSER D.: *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Wellesley, MA, 1993. Translated by L. L. Schumaker. 1

[HMPY97] HU C. Y., MAEKAWA T., PATRIKALAKIS N. M., YE X.: Robust interval algorithm for surface intersections. *Computer-Aided Design 29*, 9 (September 1997), 617–627. 1, 2

[HMSP96] HU C. Y., MAEKAWA T., SHERBROOKE E. C., PATRIKALAKIS N. M.: Robust interval algorithm for curve intersections. *Computer-Aided Design 28*, 6/7 (June/July 1996), 495–506. 1

[HPY96] HU C. Y., PATRIKALAKIS N. M., YE X.: Robust interval solid modeling: Part I, Representations. *Computer-Aided Design 28*, 10 (October 1996), 807–817. 1, 5

[KLF01] KULISCH U., LOHNER R. J., FACIUS A.: *Prespectives on Enclosure Methods*. Springer, New York, 2001. 3

[Kre94] KREYSZIG E.: *Advanced Engineering Mathematics*. Wiley Eastern Edition, New Age International Limited, 1994. 3

[Loh92] LOHNER R. J.: Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In *Computational Ordinary Differential Equations*, Cash J., Gladwell I., (Eds.). Clarendon Press, Oxford, 1992, pp. 425–435. 2, 3

[Loh95] LOHNER R. J.: Step size and order control in the verified solution of IVP with ODEs. In *SciCADE'95 International conference on scientific computation and differential equations* (Stanford, C.A., 1995). 3

[MKM*03] MUKUNDAN H., KO K. H., MAEKAWA T., SAKKALIS T., PATRIKALAKIS N. M.: *Surface Intersections with Validated Error Bounds*. Tech. Rep. 2003-6, Design Laboratory, MIT, 2003. 2, 3

[Moo66] MOORE R. E.: *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966. 2, 3, 4

[Ned99] NEDIALKOV N. S.: *Computing the Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation*. PhD thesis, University of Toronto, Toronto, Canada, 1999. 2, 3

[NJC99] NEDIALKOV N. S., JACKSON K. R., CORLISS G. F.: Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation 105*, 1 (1999), 21–68. 2, 3

[PM02a] PATRIKALAKIS N. M., MAEKAWA T.: Intersection problems. In *Handbook of Computer Aided Geometric Design, Chapter 25* (Amsterdam, July 2002), Farin G., Hoschek J.„ Kim M. S., (Eds.), Elsevier, pp. 623–650. 1, 3

[PM02b] PATRIKALAKIS N. M., MAEKAWA T.: *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Heidelberg, 2002. 1, 3

[PMKM04] PATRIKALAKIS N. M., MAEKAWA T., KO K. H., MUKUNDAN H.: Surface to surface intersection. In *International CAD Conference and Exhibition, CAD'04* (Thailand, May 2004), Piegl L., (Ed.). 3

[Pro] PROFIL/BIAS, Interval Arithmetic Subroutines. *http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html*. 4

[PTVF88] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical Recipes in C*. Cambridge University Press, 1988. 1, 3

[SSP01] SAKKALIS T., SHEN G., PATRIKALAKIS N. M.: Topological and geometric properties of interval solid models. *Graphical Models 63*, 3 (2001), 163–175. 5

[Sta97] STAUNING O.: *Automatic Validation of Numerical Solutions*. PhD thesis, Technical University of Denmark, Lyngby, Denmark, 1997. 3

[TMSP97] TUOHY S. T., MAEKAWA T., SHEN G., PATRIKALAKIS N. M.: Approximation of measured data with interval B-splines. *Computer-Aided Design 29*, 11 (November 1997), 791–799. 5

[Vno] VNODE, Validated non linear ODE solver. *www.cas.mcmaster.ca/nedialk/Software/VNODE/VNODE.shtml*. 4

[YM99] YE X., MAEKAWA T.: Differential geometry of intersection curves of two surfaces. *Computer Aided Geometric Design 16*, 8 (Sept. 1999), 767–788. 2