INTERSECTIONS WITH VALIDATED ERROR BOUNDS FOR BUILDING INTERVAL SOLID MODELS

Harish Mukundan
Kwang Hee Ko**
Nicholas M. Patrikalakis*

Department of Mechanical Engineering
Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA

**Design and Manufacturing Institute
Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030, USA

ABSTRACT

Interval arithmetic has been considered as a step forward to counter numerical robustness problem in geometric and solid modeling. The interval arithmetic boundary representation (B-rep) scheme was developed to tackle this problem. In constructing an interval B-rep solid, robust and efficient computation of intersections between the bounding surfaces of the solid is a critical issue. To address this problem, a marching method based on a validated interval ordinary differential equation (ODE) solver was proposed, motivated by its potential for the interval B-rep model construction. In this paper, we concentrate on the issue of error control in model space using the validated ODE solver, and further explain that the validated ODE solver can be used in the construction of an interval B-rep solid model using such an error control.

INTRODUCTION

Solid modeling is one of the key computational tools aiding design and manufacturing, and has been successfully used in computer graphics, CAD/CAM and CAE. In pursuit of a stable solid modeling environment, which allows us to represent the de-

sign shape in a topologically valid and consistent manner, several approaches have been proposed. Especially, the Boundary Representation (B-rep) is a frequently used technique to represent such a model. The mathematical theory behind B-rep is well established [1,2] but in practice having an *ideal* B-rep solid model still remains beyond the reach of current computational technology primarily due to the inherent limitations present in computer arithmetic and curve approximations. These limitations, such as finite precision (round-off error), arise due to the use of finite set of floating point numbers to represent the infinite set of real numbers with limited number of bits. This may cause pathological behavior of geometric modeling algorithms resulting in significant computational errors, leading to break down of geometric computations and adversely affecting productivity [3].

To mitigate the effects of such limitations, one might use geometric modeling systems based on non-conventional arithmetic techniques. Typical examples are integer and rational arithmetic, and interval and lazy arithmetic [4–12]. Among them, computation techniques based on interval arithmetic have shown the potential to remedy robustness issues in numerical computation [13]. Interval methods can compute bounds in which the correct answer is guaranteed to be enclosed. For example, solution methods of an ordinary differential equation (ODE) based on interval arithmetic can take into account three sources of errors in

*Address all correspondence to this author. Email: nmp@mit.edu; Tel.: (617)253-4555; MIT Room 5-428, Cambridge, MA 02139-4307, USA

the numerical computation of the solution; propagation of *error in initial data*, *truncation error* caused by truncating infinite series after a finite number of terms and *round-off errors* inherent to computation in a floating point environment [14]. As a first attempt to make the most use of interval arithmetic in the robust B-rep method, Hu *et al.* [9, 10] developed a data structure and Boolean operations for manifold and non-manifold interval boundary models. Later, the use of interval boundary models in boundary representation was topologically justified by Sakkalis *et al.* [15].

Interval arithmetic B-rep relies on efficient evaluation of intersection of surfaces and generation of boxes (validated error bounds in 3D model space). Let us assume the existence of a B-rep solid model M , as a well-defined conceptual object. The difficulty then is capturing this in an approximated numerical computational instantiation. An interval boundary representation model is generated from the solid M with *face*, *edge* and *vertex boxes* [15]. The face boxes cover the interior of the faces of M , whereas the edge boxes cover the boundary curves and the vertex boxes corner points of the faces of M . Note that face boxes do not intersect edges and vertices [15]. Please see Figure 1. Face boxes can be constructed by evaluating interval points

Figure 1. Depicts pre-image of face boxes and the pre-image of edge boxes on the parametric space of a surface $\mathbf{P}(\sigma, t)$. Please note that the edge boxes cover the boundary curves.

in the interior of a region bounded by a collection of rectangles (edge boxes) in the parametric space (σ, t) of the underlying surface say $\mathbf{P}(\sigma, t)$. On the other hand, edge boxes in general result from the intersection of surfaces which is a complicated and challenging task. Therefore, computation of intersection is the key step in achieving the interval-based boundary representation. Though papers [9, 10, 16, 17] talk about the attempts made to obtain intersection along with validated error bounds, the issue of efficiency in computing these boxes and validity of the boxes obtained is not addressed. The methods described in the above papers [9, 10, 16, 17] rely on the *Interval Projected Polyhedron algorithm* (IPP) [18, 19]. The IPP algorithm [17], which is based

on subdivision techniques coupled with interval arithmetic, can find the solution to a nonlinear system of polynomial equations robustly and has been applied to intersection of surfaces. However, a *topology resolution* of the intersection based on adjacency information is complicated [17]. 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 [16], an inherent problem associated with any subdivision algorithm. To avoid such problems of the IPP algorithm for intersection computation, a first attempt was made to apply validated ODE solvers to surface-to-surface intersection problems [20]. Because of its robustness and efficiency in solving ODEs, it is shown to be a promising way for intersection computation. Both *transversal* and *tangential* intersections can be formulated as a system of ODEs [21] which are solved using the validated ODE solver producing enclosures of the exact solution [20].

In this paper, we demonstrate how to obtain the two estimates of tolerances parametric space ($\epsilon_{PS_{approx}}$ or $\epsilon_{PS_{cons}}$) from a specified model space tolerance (tol_{MS}) using a conservative relation or an approximate relation. We propose an algorithm to control the parametric space error bounds obtained from a validated ODE solver. Then, we provide the properties of the validated ODE solver which are essential to show that the validated ODE solver can be used to construct consistent interval B-rep models. Please refer Figure 2 for a detailed overview of our method.

This paper is structured as follows: First, given a tolerance in model space, we obtain the corresponding tolerance in parameter space. In the next section we use this tolerance in parametric space, to obtain validated error bounds for the intersection in the parametric spaces. Based on these error bounds we obtain validated error bounds in the model space in the subsequent section, and further suggest a way to reduce these error bounds. We then discuss the properties of the validated ODE solver necessary to construct an interval B-rep solid. The example's section shows how we employ our methods, and we conclude the paper with a summary and identification of applications. Appendix A gives a detailed derivation of our proposed conservative relation for obtaining the tolerance in parametric space from the input model space tolerance.

Preliminaries

Given a very complicated system of space curves, we can find out surfaces, whose intersection results in the system of complicated space curves. But if we limit the class of surfaces, then we are able to predict some characteristics of the intersection curves. Moreover it is much easier to construct and modify some special types of surfaces which are extensively used in the CAD/CAM community like the free-form Rational Polynomial Parametric (RPP) surfaces [21, 22]. We assume at this point that the surfaces we use are *Regular* and *RPP*. RPP surfaces are

Figure 2. Flow chart representing the entire algorithm. The shaded part illustrates a control mechanism for obtaining error bounds in parametric space.

C^∞ continuous [20]. Tracing the intersection (both tangential and transversal) of two RPP surfaces $\mathbf{P}(\sigma, t) = \mathbf{Q}(u, v)$ using a marching method [21] essentially reduces to solving a system of ordinary differential equations (ODEs) which are arc-length(s) parametrized. This system of ODEs (1) with the initial condition represents an initial-value problem (IVP) for ODEs, which can be written in vector form as,

$$\mathbf{y}'(s) = \mathbf{f}(\mathbf{y}(s)), \quad \mathbf{y}(s_0) = \mathbf{y}_0, \quad (1)$$

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

Given initial conditions, we can in principle integrate the system of ODEs to obtain a series of points in the parameter space of each of the surfaces which represent an approximation to the intersection curve segment. The bounds on the initial condition $\mathbf{y}_0 = [[\sigma_0][t_0][u_0][v_0]]^T$ ¹ represented as intervals, uses a

¹[a] is the notation for an interval number in interval arithmetic and $w([a])$ represents the width of the interval [a]. Please refer [23] a detailed treatment of interval arithmetic

validated scheme to trace the system of ODEs (1) in the parametric space of the surfaces. Please refer to the Section on *Validated Error Bounds in Parametric Space*.

ESTIMATION OF TOLERANCE IN PARAMETRIC SPACE

The continuous error bounds which we obtain are essentially the *a priori enclosures* in the parametric space of the surfaces which are further mapped to the model space. The error control based just on a validated interval arithmetic scheme is applicable only to the error at certain definite s_j s. What we require is to develop a method such that given a tolerance in the model space we can limit the size of the enclosures obtained by the validated solution scheme. For this the model space tolerance is transformed into a conservative tolerance in the parametric spaces as shown by the Figure 3. For this we propose two methods.

Figure 3. Depicts how we want to obtain tolerance in parametric space from tolerance in model space.

1. A strictly conservative method in the true spirit of interval arithmetic and,
2. An approximate method, which is adapted from a previous work on interval solids by Shen *et al.* [24, 25].

The method by Grandine and Klein [26] offers to obtain an estimate of the intersection of two surfaces and an associated error. Grandine-Klein algorithm however solves the intersection as a boundary value problem instead of an initial value problem. In this context [27] proposes an algorithm to obtain the tolerance in parametric space as an input to the Grandine-Klein algorithm.

Conservative Relation

This relation is called conservative in the sense that, if we use the relation (2) to obtain the tolerance in parametric space $\epsilon_{PS_{cons}}$, and further input to a validated ODE solver, we obtain model space bounds definitely smaller than the required tolerance in model space tol_{MS} . For our analysis we consider a bicubic RPP surface namely a bicubic Bézier surface $\mathbf{P}(\sigma, t)$ such that

$0 \leq \sigma, t \leq 1$. Suppose we find the uncertainty of any one component (without loss of generality we can assume the x -component be $P(\sigma, t)$). We are to get a bound for width of the intervals generated by the validated ODE solver ($w([\sigma])$ or $w([t])$) such that the evaluated width of the boxes $w(P([\sigma], [t]))$ in the 3D model space of the surface is less than a certain tolerance.

We can obtain a relation based on the expansion of the tensor product bicubic Bézier patch and the application of interval arithmetic identities [23] to obtain the following equation (2). Appendix A describes the derivation in greater detail.

$$w([P(\sigma, t)]) \leq 156R_{max}(w(\sigma) + w(t)). \quad (2)$$

where, R_{max} is the size of the boxes in which the surfaces are enclosed or the maximum dimension of the control points (convex hull). Equation (2) means that if we are given a tolerance in model space i.e. tol_{MS} , we require that:

$$w([P(\sigma, t)]) \leq 156R_{max}(w(\sigma) + w(t)) \leq tol_{MS}.$$

Thus if we assume that we require same tolerance ($\epsilon_{PS_{cons}}$) in the σ and t domain we can write,

$$\epsilon_{PS_{cons}} \leq \frac{tol_{MS}}{312R_{max}}. \quad (3)$$

Evaluation based on this formula depends on the form in which the equation is evaluated and might overestimate the tolerance $\epsilon_{PS_{cons}}$. But it is a *conservative estimate* in the sense that we obtain a tolerance in the parameter space, which will result in a conservative tolerance in the model space. We note that the tolerance in parametric space $\epsilon_{PS_{cons}}$ also depends on the convex hull of the surface.

Approximate Relation

This method for estimates a parameter space bound $\epsilon_{PS_{approx}}$ which would result in a model space bound close to a given tolerance. It is based on a relation obtained by Shen *et al.* [24, 25]. Let us assume a Bézier surface of degrees d_σ and d_t , with each control point having a constant width δ_0 for each of its components (for degenerate surfaces $\delta_0 = 0$). Just as in the conservative relation we obtain the relation for only one component, say R_x is the average of the absolute value of the x -coordinates of the control points (R_i). Similar formula for y and z components can be obtained by replacing R_x with R_y and R_z respectively. We obtain the following estimate:

$$\epsilon_{PS_{approx}} \approx \frac{tol_{MS} - \delta_0}{4(d_\sigma + d_t)(\delta_0 + R_{max})}, \quad (4)$$

$R_{max} = \text{norm}(R_x, R_y, R_z)$, is a factor taking into account the convex hull of the surfaces. Thus the coordinates of control points affect the obtained parametric space tolerance. The width of control points also affects the tolerance needed. This relation can account for surfaces of any order.

Based on the above equations (3) and (4) we are now in a position to obtain a tolerance in the parameter space $\epsilon_{PS_{approx}}$ which corresponds to a given tolerance in the model space.

VALIDATED ERROR BOUNDS IN PARAMETRIC SPACE

Since computing intersection of surfaces is formulated as an IVP of a system of ODEs, it is very important to find the initial values for accurate solution. In the surface to surface intersection problem, a set of characteristic points such as border, turning and singular points of the intersection are identified which are provided as initial values for tracing intersection curve components [19]. This essentially reduces to solving a system of nonlinear polynomial equations. For details, please refer to [19] (pp. 149-150). Robust methods to solve a system of polynomial equations with guaranteed error bounds in the surface to surface intersection context is dealt with in [18, 28]

Standard numerical methods for solving IVPs for ODEs like the Runge Kutta method or Adam's Bashforth method attempt to compute an approximate solution that satisfies a user-specified tolerance at discrete points. The algorithms to control the step size are based on controlling just the error alone [20]. The added advantage of a validated ODE scheme is a robust step size control which also verifies the existence and uniqueness before predicting the step size can prevent the solution from straying from one branch to another within that step as shown by [20, 29]. The validated interval scheme for ODEs [13, 20] also produces a guaranteed error bound on the true solution. Each step in a validated interval solution scheme for solving IVPs for ODEs can guarantee:

1. The existence of the solution: i.e. if solution exists in that step within the enclosure.
2. The uniqueness of the solution: i.e. if we have a unique solution in that step within the enclosure.

Briefly a validated ODE solving scheme is done in two phases. *Algorithm I* verifies the existence and uniqueness of the intersection curve segment, and a successful validation results in a step size and a corresponding *a priori enclosure* $[[\tilde{\sigma}] [\tilde{t}] [\tilde{u}] [\tilde{v}]]^T$. Please refer to Figure 4. *Algorithm II* now tries to propagate the solution by finding a tight estimate, but only at specific value of arc length. This tight estimate also 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 [20]. The result is an intersection curve obtained as a series of connected (gap free) *a priori enclosures* (boxes) in the parameter

space (Figure 4). This series of boxes encloses the exact curve of intersection in the parameter space [20]. For more details on tracing surface intersections using a validated ODE system solver please refer to [20].

Figure 4. This figure illustrates the series of *a priori enclosures* in parametric spaces which enclose the true intersection curve segment.

Controlling A Priori Enclosure in a Validated ODE Solver

We improve the step size control mechanism within the validated ODE solver to incorporate an automatic control of error in the 3D model space. The proposed control mechanism to control the *a priori enclosure* size is obtained using a validated ODE solver within the above tolerance as depicted in the shaded part of Figure 2.

The control mechanism is implemented within a validated ODE solver. The first step is to obtain the parametric space tolerance ϵ_{PS} from the control points of the surfaces using either of equation (3) or (4). This tolerance in parametric space for each surface is compared with the width of the *a priori enclosure* (RHS of relation (5)) obtained from Algorithm I of the validated ODE solver.

$$[w([\tilde{\sigma}_j]) \ w([\tilde{t}_j]) \ w([\tilde{u}_j]) \ w([\tilde{v}_j])]^T \leq [\epsilon_{PS}^P \ \epsilon_{PS}^P \ \epsilon_{PS}^Q \ \epsilon_{PS}^Q]^T, \quad (5)$$

where ϵ_{PS}^P and ϵ_{PS}^Q are the tolerance in the parametric space of surfaces **P** and **Q**, respectively.

If the width of $[\tilde{\sigma}] \ [\tilde{t}] \ [\tilde{u}] \ [\tilde{v}]^T$ is larger than the tolerance at any step, we use the tolerance as the width of the new *a priori enclosure* and find the corresponding new step size h_j^* for which the validity criterion is satisfied. This condition is depicted in the Figure 5. If the width of $[\tilde{\sigma}] \ [\tilde{t}] \ [\tilde{u}] \ [\tilde{v}]^T$ is smaller than the tolerance, we proceed to the next step. For further verification we might map the obtained enclosure right away to model space and check if the width of the boxes in the model space is less than the given tolerance tol_{MS} .

Figure 5. Mechanism to control the size of the *a priori enclosure* in a validated ODE solver. y represents any of the parameters σ, t, u or v .

VALIDATED ERROR BOUNDS IN 3D MODEL SPACE

Using a validated ODE solver we obtain a series of *a priori enclosures* in the parametric spaces of the surfaces as shown in the Figure 4. This bound in parametric space is continuous because the intersection curve segment is a continuous trajectory in the parametric space. For each of the parameters $\sigma - t$ and $u - v$ corresponding to each surface, the pre-image of the curve of intersection is enclosed in the union of the boxes corresponding to *a priori enclosures* [20]. The series of *a priori enclosures* in the $\sigma - t$ and $u - v$ parametric space of each surface is mapped to the model space as illustrated in Figure 6. According to Mukundan *et al.* [20], the union of the bounds in the 3D model space guarantees to contain the true curve of intersection. This is again justified based on our previous assumption of intersecting surfaces being RPP. This enable us to realize the goal of a continuous gap-free bound on the curve of intersection in the model space, given continuous bounds on its pre-image. At this point we have two series of boxes in the model space each of which enclose the true curve of intersection in the model space. Note that Mukundan *et al.* [20] talks about interval surfaces, but RPP surfaces could be written as interval surfaces with degenerate control points.

Model Space Error Bound Reduction The union of the boxes obtained in model space by mapping the enclosures of the pre-image of the curve of intersection bounds the true curve of intersection. The series of boxes obtained from each of the parametric spaces $\sigma - t$ and $u - v$ contain the true intersection curve segment. We may further show that the true intersection curve segment actually lies in the region obtained by the intersection of the two separate bounds.

Let $[c_P(s)]$ and $[c_Q(s)]$ be the bounds on the curves of intersection in the model space obtained by mapping the pre-images of the bounds to the intersection curve from each of the surface patches. Also let us assume that $c_f(s)$ is the actual curve of intersection of the two surfaces. Then $c_f(s)$ lies in the region in the

Figure 6. Mapping of the pre-image of the intersection curve segment from the parameter space to the model space. Note that the boxes obtained in the parameter space of each of the surface is continuous, gap free and ordered. Also observe that the two sets of boxes corresponding to mapping using each of the surfaces **P** and **Q**

model space obtained by the intersection of $[\mathbf{c_P}(s)]$ and $[\mathbf{c_Q}(s)]$. This is obtained from the definition of interval arithmetic as,

$$\mathbf{c_f}(s) \in [\mathbf{c_P}(s)] \quad \text{and} \quad \mathbf{c_f}(s) \in [\mathbf{c_Q}(s)].$$

Hence we say,

$$\begin{aligned} &\mathbf{c_f}(s) \in [\mathbf{c_P}(s)], \text{ and } \mathbf{c_f}(s) \in [\mathbf{c_Q}(s)], \\ \text{or, } &\mathbf{c_f}(s) \in ([\mathbf{c_P}(s)]) \cap [\mathbf{c_Q}(s)]. \end{aligned}$$

This result implies that, if we obtain the two bounds on the intersection curve from each of the surfaces and intersect those bounds, we are able to reduce the model space error. The reduction in error bound usually depends on the relative orientation of the surfaces close to intersection as well as the relative sizes of the model space bounds obtained from each of the surfaces. Thus if the bounds from the surfaces are relatively of the same size, then the reduction is significant for a transversal intersection case compared to a tangential intersection case.

APPLICATION TO INTERVAL BOUNDARY REPRESENTATION

In [15], it is shown that an interval B-rep solid can be constructed by using an ISI algorithm with a sufficiently tight resolution. In this section, we explain that the properties of the validated ODE solver, which are essential to the mathematical proof that the validated ODE solver can be used for the construction of an interval B-rep solid.

First, the validated ODE solver generates boxes in 3D space, which guarantee to contain the exact solution. The intersection of two surfaces is formulated as a system of nonlinear ODE with

bounds on the starting points [20]. This property ensures that the exact intersection solution is contained in the boxes produced by the validated ODE solver. One of the reasons that the ISI algorithm can be used in the construction of an interval solid is its capability of controlling the size of boxes containing the exact intersection values. The validated ODE solver is also provided with such a mechanism realized by the method of this paper. Namely, a user can control the size of boxes in 3D space obtained by the validated ODE solver without compromising the first property above. Therefore, we can make the validated ODE solver produce sufficiently tight boxes.

Using these two properties of the validated ODE solver, we can prove that the validated ODE solver can be used for the construction of an interval B-rep solid by following the similar steps given in [15].

In summary, based on the results of [15] what we would like to emphasize here is that under a tight error bound which can be controlled by our procedure, we can construct an interval solid using face, and edge and vertex boxes that are obtained from the validated ODE solver which has more advantages over the ISI algorithm, and such an interval solid is approximately equal to the exact underlying solid.

EXAMPLE

Let us say our aim is to generate a solid model which looks similar to an hour-glass shape as shown in the Figure 7, using the boundary representation. Generation of the solid requires the intersection of a hyperbolic surface $\mathbf{P_1}(\sigma, t)$ and $\mathbf{Q_1}(u, v)$ a plane and subsequent intersections by two mutually parallel planes $\mathbf{R_1}$ and $\mathbf{R_2}$ and finally trimming the dangling surfaces to obtain the solid. We are specifically interested in the intersection of $\mathbf{P_1}$ with $\mathbf{Q_1}$ as this offers significant computational challenges as we observe. The control points of the hyperbolic surface $\mathbf{P_1}$ are given

below.

$$\mathbf{P}_1(\sigma, t) = \begin{bmatrix} \text{Control points for} \\ (-50, 0, 0) & (0, 0, 100) & (50, 0, 0) \\ (-50, 100, 0) & (0, 100, 0) & (50, 100, 0) \\ (-50, 200, 0) & (0, 200, 100) & (50, 200, 0) \end{bmatrix}$$

The first step would be to obtain the allowed parametric space tolerances ϵ_{PS}^P and ϵ_{PS}^Q given tol_{MS} . We can use either of the formulae (3) or (4) for this purpose. We tabulate the comparison of the two methods in Table 1. As was shown in this paper, the parametric space tolerances ϵ_{PS}^P and ϵ_{PS}^Q are used to restrict the size of enclosures in the parametric space, which in turn maintains the size of boxes in the model space within the given tolerance. Note that the conservative relation produces smaller tolerance in parametric space than the approximate relation. This is because, if we input a smaller tolerance to the validated ODE solver, we get much smaller error bounds and such error bounds hence obtained, when mapped to model space are definitely smaller than input tolerance in model space.

Case I The plane $\mathbf{Q}_1$ is arranged such that it touches the hyperbolic point of the surface $\mathbf{P}_1$. When this happens the normals to the surfaces are perfectly aligned there is a singular point in the surface intersection curve segment as we can see from the Figure 7. Tracing the intersection now involves separately tracing the four intersection curve segments and obtaining the bounds on them, given appropriate starting points. Thus we obtain the edge boxes for the intersection of $\mathbf{P}_1$ and $\mathbf{Q}_1$ in the model space within a tolerance tol_{MS} . The error bounds on each of the four starting points are obtained and are listed below.

$$\begin{aligned} [\sigma_0] [t_0] [u_0] [v_0] &= [0.85355339, 0.85355340] [0, 0] [0.85355339, 0.85355340] [0, 0]^T, \\ [\sigma_0] [t_0] [u_0] [v_0] &= [0.14644660, 0.14644661] [1, 1] [0.14644660, 0.14644661] [1, 1]^T, \\ [\sigma_0] [t_0] [u_0] [v_0] &= [0.14644660, 0.14644661] [0, 0] [0.14644660, 0.14644661] [0, 0]^T, \\ [\sigma_0] [t_0] [u_0] [v_0] &= [0.85355339, 0.85355340] [1, 1] [0.85355339, 0.85355340] [1, 1]^T. \end{aligned}$$

The model hence formed comprises of two separate parts joined at a point. This does not represent a manifold solid as its boundary is not a 2-manifold surface.

Case II (a) Now consider the case when, the plane $\mathbf{Q}_1$ is moved along the common normal near the hyperbolic point of $\mathbf{P}_1$ by a small amount. The intersection curve has a completely different behavior. Based on the direction of the perturbation (z-direction) this behavior changes. A positive perturbation in z-direction will lead to the branching of the curve which will lead to a case where we actually have two separate solids generated. Refer Figure 8 (a).

Case II (b) Now consider the case (Please refer Figure 8 (b)) when, one of the surfaces, the plane $\mathbf{Q}_1$ perturbed in the negative z-direction which leads to a different branching of the intersection curve. This will lead to a case where we have only

Figure 7. Development of a solid model which resembles an hour glass. Transversal intersection of a hyperbolic surface (shown as the red boxes in model space) and a plane involves tracing four separate intersection curve segments.

Figure 8. Our approach of a validated solution scheme for surface intersection is able to obtain validated boxes even for critical scenarios. Figure (a) shows the plane ($\mathbf{Q}_1$) 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.

one valid solid. We have shown that our scheme for obtaining validated bounds for surface intersection can resolve such cases.

Test No.	tol_{MS}	$\epsilon_{PS_{cons}}$	$\epsilon_{PS_{approx}}$
1	10^{-2}	3.125×10^{-6}	1.60256×10^{-7}
2	10^{-4}	3.125×10^{-8}	1.60256×10^{-9}
3	10^{-6}	3.125×10^{-10}	1.60256×10^{-11}
4	10^{-8}	3.125×10^{-12}	1.60256×10^{-13}

Table 1. The tolerance in parametric space of surface $\mathbf{P}_1$ obtained from the given tolerance in model space.

CONCLUSIONS

Geometric modeling systems used in variety of design aspect in engineering, analysis and development requires a solid modeling environment. Typical environments for solid modeling is based on boundary representation which require validated surface intersection techniques. This paper presents a method to address this problem in the context of numerical inconsistency issues in geometric modeling community.

An algorithm based on a validated surface intersection scheme has been proposed. By using the validated ODE solver, we can improve efficiency of finding intersections (eliminating the problems of the IPP), resulting in improvement in B-rep model construction. This method has advantages over available IPP method in speed, automation, validity and further ease of use. Using the error control capability in the model space, we can impose a user provided tolerance in creating a model conforming to the tolerance. Success has been achieved in devising a way to reduce this error bounds in 3D model space.

Under conditions where the resolution of the intersection is not necessarily easy for other methods, our method is shown to work well. The integration of this validated error bounds on edge boxes to interval solid modeling is further demonstrated.

Future directions for research involve increased automation and the development of a solid modeling system which can be used in industry context.

ACKNOWLEDGMENT

We thank T. Maekawa, N. S. Nedialkov and T. Sakkalis for useful discussions and their comments. The authors acknowledge the valuable suggestions and comments from the reviewers. This work was funded in part by the NSF (grants No. DMS-0138098 and CCR-0231511).

REFERENCES

[1] Hoffmann, C. M., 1989. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers, Inc., San Mateo, California.

- [2] Sakkalis, T., Shen, G., and Patrikalakis, N. M., 2000. "Representational validity of boundary representation models". *Computer-Aided Design*, **32**(12), pp. 719–726.
- [3] Hoffmann, C. M., 2001. "Robustness in geometric computations". *Journal of Computing and Information Science in Engineering*, **1**(2), June, pp. 105–204.
- [4] Milenkovic, V., 1993. "Robust polygon modelling". *Computer-Aided Design*, **25**(9), September, pp. 546–566.
- [5] Fortune, S., 1997. "Polyhedral modeling with multiprecision integer arithmetic". *Computer-Aided Design*, **29**(2), pp. 123–133.
- [6] Keyser, J., Krishnan, S., and Manocha, D., 1999. "Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: I-representations". *Computer Aided Geometric Design*, **16**(9), October, pp. 841–859.
- [7] Keyser, J., Krishnan, S., and Manocha, D., 1999. "Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: II-computation". *Computer Aided Geometric Design*, **16**(9), October, pp. 861–882.
- [8] Salesin, D. H., 1991. "Epsilon geometry: Building robust algorithms from imprecise computations". PhD thesis, Stanford University, March.
- [9] Hu, C. Y., Patrikalakis, N. M., and Ye, X., 1996. "Robust interval solid modeling: Part I, Representations". *Computer-Aided Design*, **28**(10), October, pp. 807–817.
- [10] Hu, C. Y., Patrikalakis, N. M., and Ye, X., 1996. "Robust interval solid modeling: Part II, Boundary evaluation". *Computer-Aided Design*, **28**(10), October, pp. 819–830.
- [11] Benouamer, M., Michelucci, D., and Peroche, B., 1993. "Error-free boundary evaluation based on a lazy rational arithmetic: A detailed implementation". *Computer-Aided Design*, **26**(6), June, pp. 403–415.
- [12] Agrawal, A., 1995. General approach to the design of robust algorithms for geometric modeling. Technical Report IRIS-95-339, Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA.
- [13] Nedialkov, N. S., 1999. "Computing the rigorous bounds on the solution of an initial value problem for an ordinary differential equation". PhD thesis, University of Toronto, Toronto, Canada.
- [14] Hansen, E., ed., 1969. *Topics in Interval Analysis*. Oxford University Press.
- [15] Sakkalis, T., Shen, G., and Patrikalakis, N. M., 2001. "Topological and geometric properties of interval solid models". *Graphical Models*, **63**(3), pp. 163–175.
- [16] Hu, C. Y., Maekawa, T., Sherbrooke, E. C., and Patrikalakis, N. M., 1996. "Robust interval algorithm for curve intersections". *Computer-Aided Design*, **28**(6/7), June/July, pp. 495–506.
- [17] Hu, C. Y., Maekawa, T., Patrikalakis, N. M., and Ye, X.,

1997. “Robust interval algorithm for surface intersections”. *Computer-Aided Design*, **29**(9), September, pp. 617–627.
- [18] Sherbrooke, E. C., and Patrikalakis, N. M., 1993. “Computation of the solutions of nonlinear polynomial systems”. *Computer Aided Geometric Design*, **10**(5), October, pp. 379–405.
- [19] Patrikalakis, N. M., and Maekawa, T., 2002. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Heidelberg, February.
- [20] Mukundan, H., Ko, K. H., Maekawa, T., Sakkalis, T., and Patrikalakis, N. M., 2004. “Tracing surface intersections with a validated ODE system solver”. In Proceedings of the Ninth EG/ACM Symposium on Solid Modeling and Applications, G. Elber and G. Taubin, eds., Eurographics Press.
- [21] Patrikalakis, N. M., and Maekawa, T., 2002. “Intersection problems”. In Handbook of Computer Aided Geometric Design, Chapter 25, G. Farin, J. Hoschek, and M. S. Kim, eds., pp. 623–650.
- [22] Struik, D. J., 1961. *Lectures on Classical Differential Geometry*, second ed. Dover.
- [23] Moore, R. E., 1966. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ.
- [24] Shen, G., Sakkalis, T., and Patrikalakis, N. M., 2000. “Interval methods for B-Rep model verification and rectification”. In Proceedings of the ASME 26th Design Automation Conference, pp. 140 and CD-ROM.
- [25] Shen, G., Sakkalis, T., and Patrikalakis, N. M. “Boundary representation model rectification”. *Graphical Models*, **63**(3), pp. 177–195. Also in: *Proceedings of the Sixth ACM Solid Modeling Symposium*. D. Anderson and K. Lee, editors. Ann Arbor, Michigan, June 2001. NY: ACM, 2001.
- [26] Grandine, T. A., and Klein, F. W., 1997. “A new approach to the surface intersection problem”. *Computer Aided Geometric Design*, **14**(2), pp. 111–134.
- [27] Mow, C., Peters, T. J., and Stewart, N. F., 2003. “Specifying useful error bounds for geometry tools: An intersector exemplar”. *Computer Aided Geometric Design*, **20**(5), pp. 247–251.
- [28] Ko, K. H., Sakkalis, T., and Patrikalakis, N. M., 2004. “Nonlinear polynomial systems: Multiple roots and their multiplicities”. In Shape Modeling International Conference, SMI 2004, F. Giannini and A. Pasko, eds., IEEE Computer Society Press, Los Alamitos, CA.
- [29] Patrikalakis, N. M., Maekawa, T., Ko, K. H., and Mukundan, H., 2004. “Surface to surface intersection”. *Computer-Aided Design and Applications*, **1**(1-4), pp. 449–458.

Appendix A: Derivation of Conservative Relationship

Consider an interval bi-cubic Bézier surface $[P](\sigma, t)$ such that $0 \leq \sigma, t \leq 1$. Suppose we find the uncertainty of any one

component (without loss of generality we can assume the x -component be $P(\sigma, t)$). We are to get a bound for $w([\sigma])$ or $w([t])$ such that, $w([P]([\sigma], [t])) \leq g(w([\sigma]), w([t]))$.

For a bi-cubic Bézier patch we can write:

$$P(\sigma, t) = \begin{bmatrix} (1-\sigma)^3 \\ 3\sigma(1-\sigma)^2 \\ 3\sigma^2(1-\sigma) \\ \sigma^3 \end{bmatrix}^T \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}, \quad (6)$$

where b_{ij} 's are the control points of the x -component. For the simplicity of analysis we assume that the control points are degenerate intervals (real numbers). We expand the above relation as,

$$P(\sigma, t) = [\beta_1 \ \beta_2 \ \beta_3 \ \beta_4] \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}, \quad (7)$$

where,

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} b_{11}(1-\sigma)^3 + 3b_{21}\sigma(1-\sigma)^2 + 3b_{31}\sigma^2(1-\sigma) + b_{41}\sigma^3 \\ b_{12}(1-\sigma)^3 + 3b_{22}\sigma(1-\sigma)^2 + 3b_{32}\sigma^2(1-\sigma) + b_{42}\sigma^3 \\ b_{13}(1-\sigma)^3 + 3b_{23}\sigma(1-\sigma)^2 + 3b_{33}\sigma^2(1-\sigma) + b_{43}\sigma^3 \\ b_{14}(1-\sigma)^3 + 3b_{24}\sigma(1-\sigma)^2 + 3b_{34}\sigma^2(1-\sigma) + b_{44}\sigma^3 \end{bmatrix}. \quad (8)$$

Thus we can write $P(\sigma, t)$ in a form $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$.

We are interested in finding the maximum width $w([P(\sigma, t)])$ within which the evaluated width would lie. We use interval arithmetic relation, $w(a+b) = w(a) + w(b)$ where a and b are any two intervals. Applying this formula we obtain,

$$w([P(\sigma, t)]) = w(\alpha_1) + w(\alpha_2) + w(\alpha_3) + w(\alpha_4). \quad (9)$$

The RHS represents the thickness of the interval surface. Further using the interval inequalities,

$$w(a.b) \leq \|a\| w(b) + \|b\| w(a), \quad \text{and} \\ \|abc\| = \|a\| \|b\| \|c\|. \quad (10)$$

We can evaluate the widths of α_i s as:

$$\begin{aligned} w(\alpha_1) &= w((1-t)^3 \beta_1) \leq w((1-t)^3) \|\beta_1\| + w(\beta_1) \|(1-t)^3\|, \\ w(\alpha_2) &= w(3t(1-t)^2 \beta_2) \leq w(3t(1-t)^2) \|\beta_2\| + w(\beta_2) \|(3t(1-t)^2)\|, \\ w(\alpha_3) &= w(3t^2(1-t) \beta_3) \leq w(3t^2(1-t)) \|\beta_3\| + w(\beta_3) \|(3t^2(1-t))\|, \\ w(\alpha_4) &= w(t^3 \beta_4) \leq w(t^3) \|\beta_4\| + w(\beta_4) \|t^3\|. \end{aligned} \quad (11)$$

We can also assume that $0 \leq \sigma, t \leq 1$. After further simplification we can obtain the inequalities²:

$$\|(1-t)\| \leq 1,$$

²The maximum of the functions over an interval is obtained from differential calculus.

$$\begin{aligned}
\|(1-t)^3\| &\leq 1, \\
\|t^3\| &\leq 1, \\
\|3t(1-t)^2\| &\leq \frac{4}{9}, \\
\|3t^2(1-t)\| &\leq \frac{4}{9}, \\
w(1-t) &= w(1) + w(t) = w(t), \\
w((1-t)^3) &\leq 12w(t), \\
w(t^3) &\leq 3w(t), \\
w(3t(1-t)^2) &\leq 24w(t), \\
w(3t^2(1-t)) &\leq 15w(t). \tag{12}
\end{aligned}$$

Based on the assumption that the surfaces are represented by degenerate intervals i.e. $w(b_{ij}) = 0$, we can write

$$\begin{aligned}
w(\beta_1) &\leq w(\sigma) (12 \|b_{11}\| + 24 \|b_{21}\| + 15 \|b_{31}\| + 3 \|b_{41}\|), \\
w(\beta_2) &\leq w(\sigma) (12 \|b_{12}\| + 24 \|b_{22}\| + 15 \|b_{32}\| + 3 \|b_{42}\|), \\
w(\beta_3) &\leq w(\sigma) (12 \|b_{13}\| + 24 \|b_{23}\| + 15 \|b_{33}\| + 3 \|b_{43}\|), \\
w(\beta_4) &\leq w(\sigma) (12 \|b_{14}\| + 24 \|b_{24}\| + 15 \|b_{34}\| + 3 \|b_{44}\|). \tag{13}
\end{aligned}$$

To obtain $\|\beta_i\|$ s we use the triangular inequality,

$$\|a + b\| \leq \|a\| + \|b\|. \tag{14}$$

We obtain $\|\beta_i\|$ s as:

$$\begin{aligned}
\|\beta_1\| &\leq \|b_{11}\| + \frac{4}{9} \|b_{21}\| + \frac{4}{9} \|b_{31}\| + \|b_{41}\|, \\
\|\beta_2\| &\leq \|b_{12}\| + \frac{4}{9} \|b_{22}\| + \frac{4}{9} \|b_{32}\| + \|b_{42}\|, \\
\|\beta_3\| &\leq \|b_{13}\| + \frac{4}{9} \|b_{23}\| + \frac{4}{9} \|b_{33}\| + \|b_{43}\|, \\
\|\beta_4\| &\leq \|b_{14}\| + \frac{4}{9} \|b_{24}\| + \frac{4}{9} \|b_{34}\| + \|b_{44}\|. \tag{15}
\end{aligned}$$

If we assume that $\|b_{ij}\| \leq R_{max}$, i.e. the size of the boxes in which the surfaces are enclosed or the maximum dimension of the control points(convex hull), we can write,

$$\|\beta_1\| = \|\beta_2\| = \|\beta_3\| = \|\beta_4\| \leq \frac{26}{9} R_{max}. \tag{16}$$

Substituting into equation (13) and using the inequalities (12) we obtain,

$$\begin{aligned}
w(\beta_1) &\leq w(\sigma) 54 R_{max}, \\
w(\beta_2) &\leq w(\sigma) 54 R_{max}, \\
w(\beta_3) &\leq w(\sigma) 54 R_{max}, \\
w(\beta_4) &\leq w(\sigma) 54 R_{max}. \tag{17}
\end{aligned}$$

Applying the above relations to equation (9) we finally obtain the relationship,

$$w([P(\sigma, t)]) \leq 156 R_{max} (w(\sigma) + w(t)). \tag{18}$$