www.cadanda.com
Computer-Aided Design
AND APPLICATIONS

Torus/Torus Intersection

Xiao-Ming Liu[1,2,3,4], Chang-Yuan Liu[1,3,4], Jun-Hai Yong[1,3,4] and Jean-Claude Paul[1,3,5]

[1]Tsinghua University, Beijing, PR China
[2]Command and Engineering College of Chemical Defense, Beijing, PR China
[3]Key Laboratory for Information System Security, Ministry of Education, Beijing, PR China
[4]Tsinghua National Laboratory for Information Science and Technology, Beijing, PR China
[5]INRIA, France

ABSTRACT

This paper presents a new algorithm for torus/torus intersection. The pre-image of the intersection in the parametric space of one torus is represented by an implicit equation. The pre-image is divided into one-valued function curve segments by characteristic points. The topological feature of these characteristic points is analyzed to obtain the structure of the pre-image. Intersection curves satisfying required precision are found by a self-adaptive refinement method. Experiment results are presented to illustrate the stability and efficiency of the method.

Keywords: surface/surface intersection, torus, topology resolution.
DOI: 10.3722/cadaps.2011.465-477

1 INTRODUCTION

Surface/surface intersection is a fundamental problem in CAD/CAM applications, such as trimming and Boolean operations [13]. The torus is one of the CSG primitives in solid modeling systems [8] and has several important applications, such as producing rounds and fillets for objects with sharp edges and the shape of cutter heads of numerically controlled machine tools [12]. We often need to compute the intersection curves of two tori in CAD/CAM systems.

General intersection problems are analyzed in [10]. Tracing methods [1-4, 16, 17] are widely used in computing the intersection curves. They can efficiently generate sequences of points of an intersection curve branch by stepping from a given point on the required curve in a direction prescribed by the local differential geometry. However, such methods require starting points for every branch of the solution and it is not a trivial problem when small loops exist. Step size selection is also complex and too large a step size may lead to straying or looping [10].

Lattice methods [9, 10, 13] reduce the dimensionality of surface/surface intersection by computing intersections of a number of iso-parametric curves of one surface with the other surface followed by connection of the resulting discrete points to form intersection curves. This method avoids solving a large number of uncertain non-linear equations, but it is difficult to find out the intersection curves of all small loops and singular points.

The initial idea of this paper comes from the lattice method. The iso-parametric curves of a torus are a set of circles and the problem of circle/torus intersection has closed-form solutions. The difficulty is determining the structure of the intersection curves, which is also important for finding all small loops and avoiding unnecessary computation of the circle/torus intersection. Grandine and Klein [6] proposed the topology resolution of B-Spline surface intersections. They determine the structure of the intersection curves including close loops prior to numerical tracing. In this paper, their method is developed to obtain the structure of the torus/torus intersections. After the structure of the intersection curves is fixed, the intersection curves are refined by a self- adaptive method until satisfying the required precision.

The rest of the paper is organized as follows. Sect. 2 develops the topology resolution for the torus/torus intersections. Sect. 3 gives intersection curves refinement method. Sect 4 finds the circles of the torus/torus intersections. The experimental results are given in Sect. 5. Finally, Sect. 6 concludes the paper.

2 TOPOLOGY RESOLUTION OF TORUS/TORUS INTERSECTIONS

2.1 The Pre-image Curves of the Intersections in the Parametric Space of a Torus

Suppose that we need to compute the intersections of two tori, $\mathbf{T}_1$ and $\mathbf{T}_2$. By means of translation and rotation of 3-D coordinate system, the major circle of $\mathbf{T}_2$ is transformed into a standard circle in XOY plane and the axis direction of $\mathbf{T}_1$ is $\mathbf{N}(N_x,0,N_z)$. Suppose that the center of $\mathbf{T}_1$ is (x_0,y_0,z_0) and its major radius and minor radius are R_1 and r_1, respectively. The parametric equation of $\mathbf{T}_1$ is

$$\begin{cases} x = x_0 + N_z(R_1 + r_1\cos(v))\cos(u) + N_x r_1 \sin(v) \\ y = y_0 + (R_1 + r_1\cos(v))\sin(u) \\ z = z_0 - N_x(R_1 + r_1\cos(v))\cos(u) + N_z r_1 \sin(v) \end{cases}, \tag{2.1}$$

where the parameters u,v belongs to $[0,2\pi]$. Suppose that the major radius and minor radius of $\mathbf{T}_2$ is R_2 and r_2, respectively, and the implicit equation of $\mathbf{T}_2$ is

$$4(x^2+y^2){R_2}^2 - (x^2+y^2+z^2+{R_2}^2-{r_2}^2)^2 = 0\,. \tag{2.2}$$

Substituting Eqn. (2.1) into Eqn. (2.2) yields an implicit equation in u,v:

$$\begin{aligned} F(u,v) = {} & 4((N_z(R_1+r_1cos(v))cos(u)+N_x r_1 sin(v)+x_0)^2+((R_1+r_1cos(v))sin(u)+y_0)^2){R_2}^2 \\ & -((N_z(R_1+r_1cos(v))cos(u)+N_x r_1 sin(v)+x_0)^2+((R_1+r_1cos(v))sin(u)+y_0)^2 \\ & +(-N_x(R_1+r_1cos(v))cos(u)+N_z r_1 sin(v)+z_0)^2+{R_2}^2-{r_2}^2)^2 = 0 \end{aligned}\,. \tag{2.3}$$

It is the pre-image of the intersections in the parametric space of $\mathbf{T}_1$. Thus the problem of torus/torus intersections is transformed into computing the pre-image curves $F(u,v)=0$.

Eqn. (2.3) is an equation in two variables u,v. By means of tangent half-angle formula, it can be transformed into a polynomial equation in two variables $\tan(u/2)$ and $\tan(v/2)$ where the degree of each variable is less than 4. If u is a constant value u_0, substituting $u=u_0$ into Eqn. (2.3) will yield a quartic equation in $\tan(v/2)$, which has closed-form solutions. Thus, v can be determined. In a similar way, u can be determined if v is a constant value. In the geometric view, it is equivalent to finding the intersection of $\mathbf{T}_2$ and the iso-parametric circles of $\mathbf{T}_1$, which is also a problem with closed-form solutions.

2.2 Characteristic Points of the Pre-image Curves

The structure of pre-image $F(u,v)=0$ can be determined by finding the following characteristic points [6, 10].

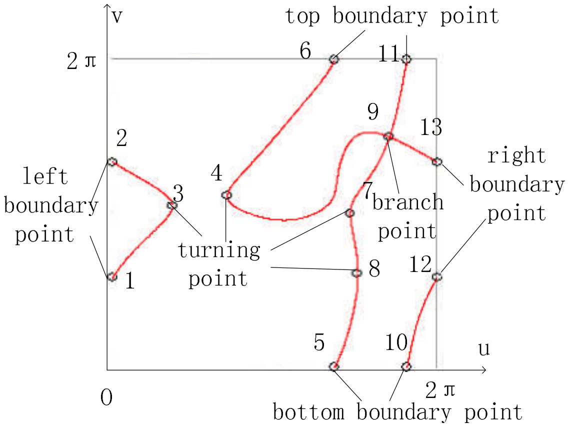

Fig. 1: A pre-image of torus/torus intersections with various characteristic points.

1. Boundary points: The intersections of $F(u,v)=0$ with four boundary edges of the parameter space $[0,2\pi]^2$, including left-right boundary points and top-bottom boundary points, which can be calculated by solving the equations $F(0,v)=0$ and $F(u,0)=0$, respectively. Considering $F(0,v)=F(2\pi,v)$, left boundary points and right boundary points always come in pairs. As illustrated in Fig. 1, points 1 and 12, 2 and 13 are two pairs of left-right boundary points. Similarly top boundary points and bottom boundary points always come in pairs, such as the points 5 and 6, 10 and 11 in Fig. 1.
2. Turning points: The u-tuning points are the points where the tangent of $F(u,v)=0$ is parallel to the $u=0$ axis, which satisfies
$$F_v(u,v)=0\,. \tag{2.4}$$
Similarly, the v-turning points also exist, but it is not necessary to consider it in this paper. So turning points is short for u-turning point in this paper. Eqns. (2.3) and (2.4) form a system of two trigonometric equations in two unknowns. By means of tangent half-angle formula, it can be transformed into a polynomial system of two equations in two variables and each variable has a degree less than 4. Solving this polynomial system by Interval Projected Polyhedron algorithm [10, 14] or other numerical methods [5, 11], we obtain all the points satisfy $F_v(u,v)=0$. A point is a turning point if it satisfies $F_v(u,v)=0$ and $F_u(u,v)\neq 0$ simultaneously. As illustrated in Fig. 1, points 3, 4, 7 and 8 are turning points.
3. Singular points: The points on the curve which satisfy $F_u=F_v=0$ are called singular points. Supposing that the tangent of the curve $F(u,v)=0$ at a singular point (u_0,v_0) is (α,β), we have [15]
$$\alpha^2F_{uu}+2\alpha\beta F_{uv}+\beta^2F_{vv}=0\,. \tag{2.5}$$
We compute the discriminant $\Delta=F_{uv}^2-F_{uu}F_{vv}$. If the discriminant is negative, the singular point (u_0,v_0) is an isolated point, which can be omitted when computing the continuous intersection curves. If the discriminant equals zero, the singular point (u_0,v_0) is a cusp, which is treated as degenerated turning point and handled by perturbation (Sect. 3.4). If the discriminant is positive,

(u_0, v_0) is a branch point. We can calculate the slope $k_{1,2}$ of the curve $F(u,v)=0$ at the branch point by solving β / α from Eqn. (2.5), which has two real roots if $F_{vv} \neq 0$. If F_{vv} equals zero, we get $k_1 = -F_{uu} / 2F_{uv}$ and $k_2 = \infty$.

Following the fundamental principle of algebraic geometry, each branch of $F(u,v)=0$ has characteristic points [15]. Moreover, the following theorem makes it clear that all characteristic points of $F(u,v)=0$ divided it into curve segments of one-valued functions.

Theorem 1. Suppose that $\mathbf{G}: F(u,v)=0$ is the pre-image of intersections of two tori. If we remove all of its characteristic points (including boundary points, turning points and singular points) from it, then

1. $\mathbf{G}: F(u,v)=0$ is divided into n explicit function curve segments $\mathbf{G}_i : v = g_i(u)$, $i = 1,\ldots,n$ with C^1 continuity. Each two curve segments $\mathbf{G}_i$ and $\mathbf{G}_j$ $(i \neq j)$ have no intersections.
2. The endpoints of every curve segment $\mathbf{G}_i$ are characteristic points of $\mathbf{G}$.
3. Let the domain of $\mathbf{G}_i : v = g_i(u)$, $i = 1,\ldots,n$ be $D_i = (u_{i0}, u_{i1})$ .If $\mathbf{G}_i$ and $\mathbf{G}_j$ satisfy $g_i(u_1) > g_j(u_1)$ at some u_1, then $g_i(u) > g_j(u)$ will be satisfied for all $u \in D_i \cap D_j$.

Prove:

1. Removing all the characteristic points of $\mathbf{G}: F(u,v)=0$, we can get n C^0 continuous curve segments $\mathbf{G}_i$, $i = 1,\ldots,n$. If $\mathbf{G}_i$ and $\mathbf{G}_j$ $(i \neq j)$ have an intersection point, it must be a self-intersection point of $\mathbf{G}: F(u,v)=0$. The self-intersection point must be singular points, contradict with the fact that all characteristic points have been removed. Therefore, $\mathbf{G}_i$ and $\mathbf{G}_j$ $(i \neq j)$ have no intersections. Similarly, $\mathbf{G}_i$ has no self-intersections. Computing the derivative dv / du of the implicit function $F(u,v)=0$, we have

$$dv / du = -F_u / F_v. \tag{2.6}$$

   Because all points satisfying $F_v = 0$ are removed for they are turning points or singular points, $\mathbf{G}_i$ has first-order derivate dv / du everywhere and it has C^1 continuity. If the line $u = u_0$ and $\mathbf{G}_i$ have two or more intersection points, there must exist points on $\mathbf{G}_i$ satisfying $F_v = 0$ according to Rolle's Theorem. Clearly there are contradictions. So $\mathbf{G}_i$ is one-valued, and can be expressed as an explicit function $v = g_i(u)$.
2. The endpoints of $\mathbf{G}_i$ must be the endpoints of $\mathbf{G}$ or the characteristic points of $\mathbf{G}$. The endpoints of $\mathbf{G}$ are also the characteristic points of $\mathbf{G}$. Therefore, the endpoints of $\mathbf{G}_i$ are characteristic points of $\mathbf{G}$.
3. Suppose that there exists $u_2 \in D_i \cap D_j$ satisfying $g_i(u_2) < g_j(u_2)$. Defining a function $f(u) = g_i(u) - g_j(u)$, we have $f(u_1) > 0$ and $f(u_2) < 0$. Because $f(u)$ is a continuous function, according to intermediate value theorem there must exist u_0 between u_1 and u_2 , which satisfies $f(u_0) = g_i(u_0) - g_j(u_0) = 0$. This means that $\mathbf{G}_i$ and $\mathbf{G}_j$ have intersections and leads to contradiction.

For the equation $F(u_0, v) = 0$ in v, we consider the changes of the number of its roots, when u_0 changes from 0 to 2π . Suppose that D is an open interval not containing u coordinates of the characteristic points. Because the boundary values of D_i must be u coordinates of the characteristic points, D and D_i have two possible relationships: $D \subset D_i$ or $D \cap D_i = \phi$. Supposing that there are m intervals D_i satisfying $D \subset D_i$, the equation $F(u_0, v) = 0$ has always m roots, when u_0 changes in the interval D . Only when u_0 moved from the left side of a u coordinate of a characteristic point to the right side of it, the number of roots may change. According to the changes of the number of roots, the topological features of the characteristic points can be classified as follows.

- The number of roots increases by 1. The characteristic point is the starting point of a curve segment and we call it "single starting point (SSP)". As illustrated in Fig. 1, points 1, 2, 5, 10 are SSPs.
- The number of roots decreases by 1. The characteristic point is the end point of a curve segment and is called "single end point (SEP)". As shown in Fig. 1, points 6, 11, 12, 13 are SEPs.
- The number of roots increases by 2. The characteristic point is the starting point of two curve segments and is called "double starting point (DSP)". As show in Fig. 1, points 4, 7 are DSP s.
- The number of roots decreases by 2. The characteristic point is the end point of two curve segments and is called "double end point (DEP)". As illustrated in Fig. 1, points 3, 8 are DEPs.

If the number of roots does not change, they have the following two types of topological features.

- The characteristic point is the starting point of a curve segments and is also the end point of another curve segment. We call it "single starting&end point (SSEP)".
- The characteristic point is the starting point of two curve segments and is also the end point of two other curve segments. We call it "double starting&end point (DSEP)". As illustrated in Fig. 1, the point 9 is a DSEP.

2.3 Topological Features of Characteristic Points

Let us consider the topological features of four kinds of simple characteristic points at first.

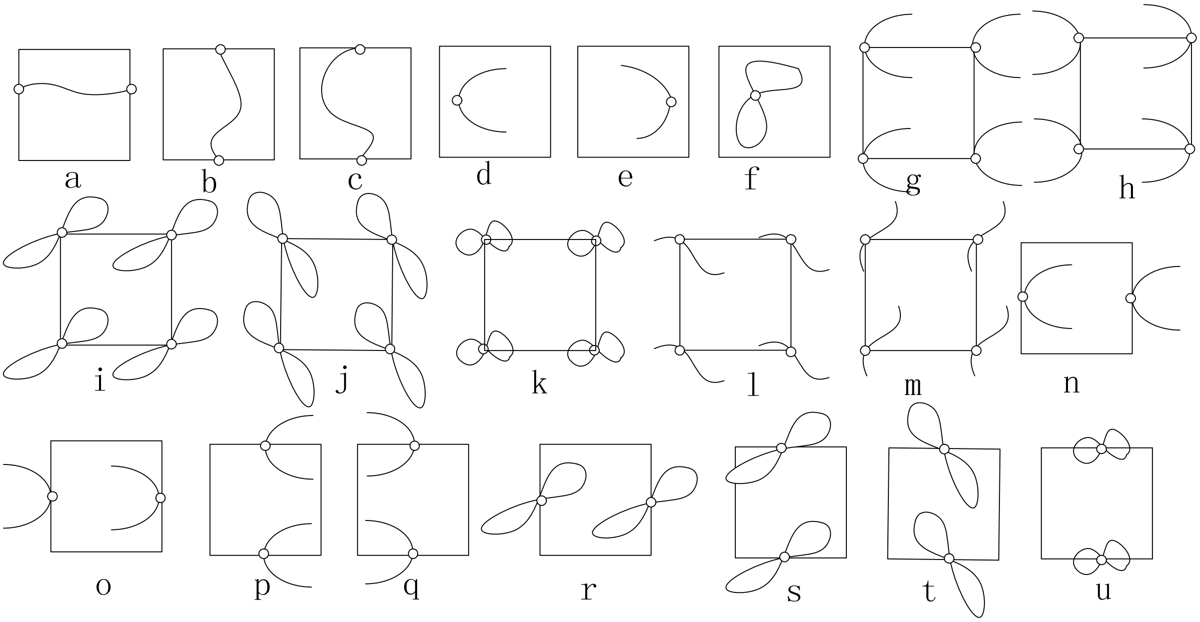

Fig. 2 Topological features of characteristic points: left-right boundary points (a); top-bottom boundary points with a negative slope (b) or a positive slope (c); turning point with right open side (d) or left open side (e); branch point (f); turning points at corner with right open side (g) or left open side (g); branch points at corner with two positive slopes (i) or two negative slopes (j) or two slopes of opposite

signs (k); corner points with a negative slope (l) or a positive slope (m); turning points at left-right boundaries with right open side (n) or left open side (o); turning points at top-bottom boundaries with right open side (p) or left open side (q); branch points at left-right boundaries (r); branch points at top-bottom boundaries with two positive slopes (s) or two negative slopes (t) or two slopes of opposite signs (u).

1. Left-right boundary points. As illustrated in Fig. 2-a, they come in pairs and the left boundary point is a SSP and the right boundary point is a SEP.
2. Top-bottom boundary points. They come in pairs, and we calculate their slope according to Eqn. (2.6) firstly. If the slope is negative, the top boundary point is a SSP and the bottom boundary point is a SEP, as illustrated in Fig. 2-b. If the slope is positive, the top boundary point is a SEP and the bottom boundary point is a SSP, as show in Fig. 2-c. The cases that the slope equals zero or ∞ will be discussed in Sect 2.4.
3. Turning points. Compute the second order derivative:

$$d^2u\,/\,dv^2 = -\left(\mathrm{F}_{\mathrm{vv}}\mathrm{F}_{\mathrm{u}}{}^{2} - 2\mathrm{F}_{\mathrm{v}}\mathrm{F}_{\mathrm{u}}\mathrm{F}_{\mathrm{uv}} + \mathrm{F}_{\mathrm{uu}}\mathrm{F}_{\mathrm{v}}{}^{2}\right)/\,\mathrm{F}_{\mathrm{u}}{}^{3}. \tag{2.7}$$

   If the derivative is positive, the open side of $\mathbf{G}$ there is directed to right, as illustrated in Fig. 2-d, and the turning point is a DSP. If the derivative is negative, there the open side of $\mathbf{G}$ is directed to right, as shown in Fig. 2-e, and the turning point is a DEP. The cases that the derivative equals zero will be discussed in Sect 2.4.
4. Branch points. They are DSEPs, as illustrated in Fig. 2-f.

When various kinds of characteristic points coincide in position, we called them multiple characteristic points. For example, when a left-right boundary point is also a turning point, we called it as a “left-right boundary point + turning point”. Treating cusps as degenerated turning points, there are 7 kinds of multiple characteristic points.

1. Left-right boundary point + top-bottom boundary point + turning point. The turning point is at the corner of the parametric domain. Firstly, compute the second order derivative $d^2u\,/\,dv^2$ according to Eqn. (2.7). If it is positive, the open side of $\mathbf{G}$ is directed to right and $\mathbf{G}$ is divided into two pieces by the top-bottom boundaries there, as illustrated in Fig. 2-g. Therefore, two SSPs are set at the left-top and the left-bottom corners, respectively. If the derivative is negative, the open side of $\mathbf{G}$ is directed to right and $\mathbf{G}$ is divided into two pieces by the top-bottom boundaries there, as illustrated in Fig. 2-h. Two SEPs are set at the right-top and the right-bottom corners. The cases that the derivative equals zero will be discussed in Sect 2.4.
2. Left-right boundary point + top-bottom boundary point + branch point. The branch point is at the corner of the parametric domain. Solve the slopes k_1 and k_2 of two branches from Eqn. (2.5) at first.
   - If k_1 and k_2 are both positive, two branches are at first and third quartiles, as illustrated in Fig. 2-i. A DSP and a DEP is set at the left-bottom corner and the right-top corner, respectively.
   - If k_1 and k_2 are both negative, two branches are at second and fourth quartiles, as illustrated in Fig. 2-j. A DSP and a DEP is set at the left-top corner and the right-bottom corner, respectively.
   - If k_1 and k_2 have opposite sign, two branches are located in different quartiles, as illustrated in Fig. 2-k. Two SSPs are set at the left-top corner and the left-bottom corner, and two SEPs are set at the right-top corner and the right-bottom corner.
   - The cases that the slope equals zero or ∞ will be discussed in Sect 2.4.
3. Left-right boundary point + top-bottom boundary point. It is a normal corner point and the slope can be calculated by Eqn. (2-6). If the slope is negative, the curve segment is at second and fourth quartiles, as illustrated in Fig. 2-l, and a SSP and a SEP are set at the left-top and right-bottom corners, respectively. If the slope is positive, the curve segment is at first and third quartiles, as illustrated in Fig. 2-m, and a SSP and a SEP are set at the left-bottom and right-top corners, respectively. The cases that the slope equals zero will be discussed in Sect 2.4.
4. Left-right boundary point + turning point. Compute the second order derivative $d^2u\,/\,dv^2$ according to Eqn. (2.7). If it is positive, the open side of $\mathbf{G}$ is directed to right there, as illustrated in Fig. 2-n, and a DSP is set at the left boundary. If it is negative, the open side of $\mathbf{G}$ is directed to left there,

as illustrated in Fig. 2-o, and a DEP is set at the right boundary. The cases that the derivative equals zero will be discussed in Sect 2.4.

5. Top-bottom boundary point + turning point. Compute the second order derivative d^2u/dv^2 according to Eqn. (2.7). If it is positive, the open side of $\mathbf{G}$ is directed to right there, as illustrated in Fig. 2-p, and two SSPs are set at the top and bottom boundaries, respectively. If the derivative is negative, the open side of $\mathbf{G}$ is directed to left there, as illustrated in Fig. 2-q, and two SEPs are set at the top and bottom boundaries, respectively. The cases that the derivative equals zero will be discussed in Sect 2.4.
6. Left-right boundary point + branch point. As illustrated in Fig. 2-r, a DSP and a DEP are set at the left boundary and the right boundary, respectively.
7. Top-bottom boundary point + branch point. Solve the slopes k_1 and k_2 of two branches from Eqn. (2.5).
   - If k_1 and k_2 are both positive, as illustrated in Fig. 2-s, a DSP and a DEP is set at the bottom boundary and the top boundary, respectively.
   - If k_1 and k_2 are both negative, as illustrated in Fig. 2-t, a DSP and a DEP is set at the top boundary and the bottom boundary, respectively.
   - If k_1 and k_2 have opposite sign, as illustrated in Fig. 2-u, two SSEPs are set at the top boundary and the bottom boundary, respectively.
   - The cases that the slopes equal zero or ∞ will be discussed in Sect 2.4.

2.4 Deal with Degeneration by Method of Perturbation

In previous section, the topological features of characteristics points are often determined by the signs of slopes and second order derivatives. However, if these values vanish, such differential method will fail. Moreover, the topological features of cusps are not determined by the differential method. [7] proposed method of perturbation dealing with the degeneration of topology resolution of [6]. Perturbation methods are applied in this paper as follows. In this section, δ is a small perturbation value.

1. If the slope dv/du at (u_0,v_0) vanishes when computing it according to Eqn. (2-6), we perturb u_0 bilaterally and search the roots of the equations $F(u_0-\delta,v)=0$ and $F(u_0+\delta,v)=0$ near v_0, respectively. If the roots near v_0 we get are v_1 and v_2, respectively, we treat the sign of v_2-v_1 as the sign of dv/du.
2. If the second order derivative d^2u/dv^2 at (u_0,v_0) vanishes when computing it according to Eqn. (2-7), or if the characteristic point is a cusp, we perturb u_0 bilaterally and search the roots of the equations $F(u_0-\delta,v)=0$ and $F(u_0+\delta,v)=0$ near v_0, respectively. If $F(u_0-\delta,v)=0$ has two roots near v_0 and $F(u_0+\delta,v)=0$ has no root near v_0, we treat d^2u/dv^2 as negative. If $F(u_0+\delta,v)=0$ has two roots near v_0 and $F(u_0-\delta,v)=0$ has no root near v_0, we treat d^2u/dv^2 as positive. If $F(u_0-\delta,v)=0$ and $F(u_0+\delta,v)=0$ have roots v_1 and v_2 near v_0, respectively, (u_0,v_0) is a negligible characteristics point and we omit it when it is at the inner of the domain. And if it is at the boundary, we treat the sign of v_2-v_1 as the sign of dv/du.
3. When we calculating the slopes k_1 and k_2 of two branches by solving Eqn. (2-5), if F_{uu} vanishes, we will get $k_1=0$. We perturb u_0 bilaterally and search the nearest roots of the equations $F(u_0-\delta,v)=0$ and $F(u_0+\delta,v)=0$ to v_0, respectively. If the nearest roots near v_0 are v_1 and v_2, respectively, we treat the sign of v_2-v_1 as the sign of k_1.

4. When we calculating the slopes k_1 and k_2 of two branches by solving Eqn. (2-5), if F_{vv} vanishes, we will get $k_2 = \infty$. We perturb v_0 bilaterally and search the nearest roots of the equations $F(u, v_0 - \delta) = 0$ and $F(u, v_0 + \delta) = 0$ to u_0 , respectively. If the nearest roots near u_0 are u_1 and u_2 , respectively, we treat the sign of $u_2 - u_1$ as the sign of k_2 .

When the characteristic point (u_0, v_0) is at or near the boundary of the domain, the intersection points of the line $u = u_0 \pm \delta$ (or $v = v_0 \pm \delta$) and **G** may be out of the domain $[0, 2\pi]^2$. Therefore, we need to search the roots of the perturbation equations near v_0 (or u_0) at the entire real field $(-\infty, +\infty)$.

2.5 Construct the Pre-image Curves

After obtaining all the characteristic points and their topological features, we sorted them ascending by the u coordinates (if the u coordinates equal, sort them by v coordinates). Suppose that the sorted characteristic points are $\mathbf{P}_i(u_i, v_i), i = 1, \ldots, n$. Moreover, the vertical sequence number (VSN) of every $\mathbf{P}_i(u_i, v_i)$ needs to be calculated where VSN is the sequence number of v_i in the increasing roots sequence of $F(u_i, v) = 0$, zero based. Thus the VSN of bottom boundary points is zero, and the VSN of top boundary points is $m - 1$, if $F(u_i, v) = 0$ has m real roots.

From Sect. 2.2, we know that the number of roots of $F(u_0, v) = 0$ is always m , when u_0 changes in an open interval D not containing u coordinates of characteristics points. The number m is the number of pre-image curve segments that have definition on the open interval D . An active list C of the pre-image curve segments is constructed and maintained to record the curve segments in vertical order, which have intersections with the line $u = u_0$. Considering the change of the active list C when the line $u = u_0$ passes through every characteristic point, we get the following algorithm for torus/torus intersection.

Algorithm 1: computing the intersections of two tori.
Input: two tori;
Output: all of the pre-image curve segments of the intersections of the two tori.

1. Compute all characteristic points. Determine their topological features and VSN. Sort them and save them into a list L of characteristic points.
2. Initialize the active list A of characteristic points, the active list C and the resultant list R of pre-image curve segments.
3. If L is empty, goto 14.
4. Take out characteristic points with same u coordinates from the head of L and save them into A . Let u_0 be their u coordinate.
5. Initialize and empty a list S of the pre-image curve segments, which need to be refined.
6. Take out the first characteristic point **P** from A . Let i be its VSN.
7. If **P** is a SEP or a SSEP, take out the i th curve segment B from C , push **P** into B , push B into R , record the last segment of B in S .
8. If **P** is a DEP or a DSEP, take out the i th and $(i+1)$ th curve segments B_1 and B_2 from C , push **P** into B_1 and B_2 , push B_1 and B_2 into R , record the last segment of B_1 and B_2 in S .
9. If **P** is a SSP or a SSEP, create one new curve segment B with **P** as its head, insert B into C before the i th curve segment of C .

```
10. If $\mathbf{P}$ is a DSP or a DSEP, create two new curve segments $B_1$ and $B_2$ with $\mathbf{P}$ as their head, insert $B_1$ and $B_2$ into $C$ before the $i$ th curve segment of $C$ .
11. If $A$ is not empty, goto 6.
12. Solve the equation $F(u_0, v) = 0$ . For each curve segment $B$ in $C$ , if the $u$ coordinate of $B$ 's last point is not $u_0$ , push a new point into $B$ according to the roots of the equation and record the last segment of $B$ in $S$ .
13. Refine the list $S$ of the pre-image curve segments. Goto 3.
14. Output $R$
```

The refinement of the pre-image curve segments will be introduced in Sect.3.

3 REFINE THE INTERSECTION CURVES

After getting the structure of the intersection curves, in order to ensure that the error of the intersections we get is less than a prescribed precision, we apply self-adapting method to refine the intersection curves.

Suppose that the 3-D line segment $\mathbf{AB}$ is a segment of the intersection curve we get and the point $\mathbf{A}$ and the point $\mathbf{B}$ are exactly on the real intersection curves, and their parameters in the parametric space of the torus $\mathbf{T}_1$ are (u_1, v_1) and (u_2, v_2), respectively. Suppose $\overset{\frown}{\mathbf{AB}}$ be the real intersection segment. The exact error of the intersection curve segment is the Hausdorff distance between $\mathbf{AB}$ and $\overset{\frown}{\mathbf{AB}}$, but it is expensive to compute the Hausdorff distance. In this paper, we propose a method to estimate an approximate error. Firstly, solve the equation $F((u_1 + u_2)/2, v) = 0$, and obtain a point $\mathbf{C}$ exactly on $\overset{\frown}{\mathbf{AB}}$ with the parameter $((u_1 + u_2)/2, v_0)$. After that, compute the minimal distance d between the point $\mathbf{C}$ and the line segment $\mathbf{AB}$, and regard d as the approximate error. With the analyses of lots of experimental data, we find that the real error is 22% larger than the approximate error we estimate. Therefore, if the required precision is ε, it is feasible to set the tolerance to be $\varepsilon / 1.22$.

Algorithm 2: refine the intersection curves

Input: the required precision ε; the list S of m intersection curve segments $E_i, i = 1, \ldots, m$; the left and right endpoints of E_i are $\mathbf{L}_i$ and $\mathbf{R}_i$, respectively; all the u coordinates of $\mathbf{L}_i$ and $\mathbf{R}_i$ equal u_1 and u_2, respectively.

Output: intersection segments satisfying required precision.

```
1   Let $u_0 := (u_1 + u_2)$ . Solve the equation $F(u_0, v) = 0$ and get an increasing roots sequence $v_i, i = 0, 1, \ldots$ ;
2   Initialize and empty two list $S_l$ and $S_r$ of intersection curve segments.
3   for (each intersection curve segment $E_i$ in $S$ )
    3.1   Let $j$ be the VSN of $E_i$ ;
    3.2   Compute the point $\mathbf{C}$ on the torus $\mathbf{T}_1$ by the parameter $(u_0, v_j)$ ;
    3.3   Compute the minimal distance $d$ between $\mathbf{C}$ and the line segment $\mathbf{L}_i\mathbf{R}_i$ ;
    3.4   If $d$ is larger than $\varepsilon / 1.22$ , insert $\mathbf{C}$ into the intersection curve between $\mathbf{L}_i$ and $\mathbf{R}_i$ , push $\mathbf{L}_i\mathbf{C}$ and $\mathbf{CR}_i$ into $S_l$ and $S_r$ , respectively.
4   If $S_l$ is not empty, refine $S_l$ and $S_r$ by calling this algorithm recursively.
5   Output the intersection curve segments refined.
```

4 COMPUTE INTERSECTION CIRCLES

The intersection curves of two tori may contain a circle. If the intersection circle is the minor circle of the torus $\mathbf{T}_1$, the pre-image curves contain a line segment $u = u_0$ and all points on the line-segment satisfy Eqns. (2.3) and (2.4). At that time, it is difficult to solve this system. Before applying topology resolution, it is necessary to compute the minor circle of the torus $\mathbf{T}_1$ if it coincides with a circle of $\mathbf{T}_2$.

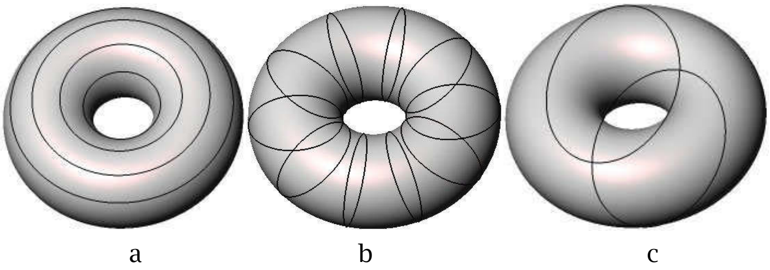

Fig. 3: Circles on a torus: (a) profile circles, (b) cross sectional circles, (c) Yvone-Villarceau circles.

There are three types of circles on a torus: profile circles, cross sectional circles (i.e. minor circles) and Yvone-Villarceau circles [8], as illustrated in Fig.3. A torus contains Yvone-Villarceau circles only if its major radius is larger than its minor radius. The radius of the Yvone-Villarceau circles is the major radius of the torus. Thus, if we assume that the minor radius of the torus $\mathbf{T}_1$ is less than that of the torus $\mathbf{T}_2$, the minor circle of $\mathbf{T}_1$ would never coincide with one Yvone-Villarceau circle of $\mathbf{T}_2$.

4.1 The Minor Circle of $\mathbf{T}_1$ Coincides with a Minor Circle of $\mathbf{T}_2$

The center of a minor circle is on the major circle of the torus. If the minor circles of two tori coincide, it must be satisfied that: 1) the two tori have similar minor radius; 2) the two major circles of two tori have intersection point at which the two major circles have similar tangent direction. Fig. 4-b illustrates the case.

4.2 The Minor Circle of $\mathbf{T}_1$ Coincides with a Profile Circle of $\mathbf{T}_2$

The center of a profile circle of $\mathbf{T}_2$ is on the axis of $\mathbf{T}_2$, and the normal of the plane containing the profile is the axis direction of $\mathbf{T}_2$. The center of a minor circle of $\mathbf{T}_1$ is on the major circle of $\mathbf{T}_1$, and the normal of the plane containing the minor circle is the tangent direction of the major circle there. If a minor circle of $\mathbf{T}_1$ coincides with a profile circle of $\mathbf{T}_2$, it must be satisfied that: 1) the axis of $\mathbf{T}_2$ is coplanar and tangent with the major circle of $\mathbf{T}_1$; 2) the radius of $\mathbf{T}_2$'s profile circle crossing the tangent point equals the minor radius of $\mathbf{T}_1$. Fig. 4-c illustrates the case.

If a minor circle of $\mathbf{T}_1$ whose u coordinate is u_0 coincides with a circle of $\mathbf{T}_2$, the circle is treated as an intersection curve segment and the left side of Eqns. (2.3) and (2.4) are divided by $u - u_0$ before it is solved to compute the characteristics points.

5 EXPERIMENTAL RESULTS

The algorithm described in previous sections has been implemented with VC++ 2005, and the data in this paper are obtained using a personal computer with E5300 2.6G CPU and 3G memory. For comparisons, we also implement the tracing method which tracing the intersection curves with a fixed advance step 0.1 after the structure of them has been determined by the topology resolution.

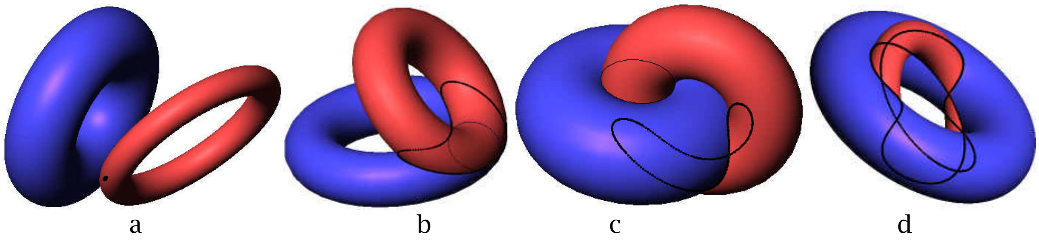

Fig. 4: Examples of torus/torus intersection: (a) The intersection curve is a small loop, (b) Two tori have a common minor circle, (c) A minor circle of $\mathbf{T}_1$ coincide with a profile circle of $\mathbf{T}_2$, and (d) Two tori have 4 tangent points and the intersection curves have 4 corresponding branch points.

Four examples of the torus/torus intersections are illustrated in Fig. 4. Tab. 1 lists the parameters of two tori in Fig. 4, including centers, axis directions, major radii and minor radii. Because the center and the axis of the torus $\mathbf{T}_2$ are always the origin of coordinates and z-axis, respectively, they are not listed in Tab. 1. In addition, Tab. 1 lists the execute time and the number of the points on the intersection curves obtained, when the precision ε is set to be 0.01, 0.001 and 0.0001. The last two columns of Tab. 1 list the experimental data of the tracing method.

Lots of experiment data show that our algorithm can robustly process the tangent intersection and the case that straying or loop missing may take place when applying the tracing method. The points obtained by analytic method are exactly on the real intersections, and the precision of the entire intersection curve is controllable. When the precision is improved by one order of magnitude, the number of points is up about two times. The execute time is nearly in proportion to the number of points. When the precision is set to be 0.001, it takes only several milliseconds to produce enough dense points on the intersection curves to satisfy the demand of normal Boolean operation in geometric modeling. From the data of Tab.1, the tracing method seems slightly faster than our method. From the algorithm 2 we can see that half of the intersection points computed in step 3.2 are discarded because the required precision has already been satisfied. It is the price we pay for the controllable precision. As a matter of fact, the average time on computing one intersection point by our method is much less than that by the tracing method because the analytic method is faster than the numerical method when solving an equation.

	$\mathbf{T}_1$ (red)				$\mathbf{T}_2$ (blue)		Our method						Tracing method	
							$\varepsilon=10^{-2}$		$\varepsilon=10^{-3}$		$\varepsilon=10^{-4}$			
	Center	Axis	R_1	r_1	R_2	r_2	ET	NP	ET	NP	ET	NP	ET	NP
a	(-1,4,5)	(1,0,2)	5	0.86	4	2	0.17	15	0.35	40	0.83	120	0.19	13
b	(3,0,4)	(4,0,3)	5	2	6	2	0.85	85	2.1	269	6.0	799	2.0	240
c	(0,4,2.4)	(1,0,0)	4	2.2	4	3	1.0	93	2.6	259	8.8	923	0.8	92
d	(0,0,0)	(3,0,4)	4	1	5	2	1.8	161	4.4	537	14	1709	3.8	566

Tab.1: Experimental data of the torus/torus intersection in Fig. 4. (ET = Execute Times in milliseconds, NP=Numbers of the Points on the intersection curves).

6 CONCLUSION

Topology resolution has been developed to solve the torus/torus intersection problem. After the structure of the intersection curves has been determined, the self-adapting refinement method is applied to obtain the intersection curves with controllable precision. Compared with the tracing method, our algorithm has the following advantages. 1) Our algorithm can overcome the drawbacks of straying and loop missing of the tracing method. 2) The points on the intersection curves obtained by our algorithm are computed by the analytic method, whose precision is higher than the points obtained by iteration when using the tracing method. 3) The tracing method can only control the precision roughly by estimating the advance step, while our method can control the precision accurately.

ACKNOWLEDGMENT

The research was supported by Chinese 973 Program (2010CB328001) and the National Science Foundation of China (60625202, 90715043). The third author was supported by Tsinghua University Initiative Scientific Research Program (2009THZ0) and the Fok Ying Tung Education Foundation (111070). The fourth author was supported by the ANR-NSFC(60911130368).

REFERENCES

[1] Aziz, N. M.; Bata, R.; Bhat S.: Bezier surface/surface intersection, IEEE Computer Graphics & Applications, 10(1), 1990, 50-58. doi:10.1109/38.45810

[2] Bajaj, C. L.; Hoffmann, C. M.: Tracing surface intersections, Computer Aided Geometric Design, 5(4), 1988, 285-307. doi:10.1016/0167-8396(88)90010-6

[3] Barnhill, R. E.; Farin, G.; Jordan, G. M.; Piper B.R.: Surface/surface intersection, Computer Aided Geometric Design, 4(1-2) , 1987, 3-16. doi:10.1016/0167-8396(87)90020-3

[4] Barnhill, R. E.; Kersey, S. N.: A marching method for parametric surface/surface intersection, Computer Aided Geometric Design, 7(1-4), 1990, 257-280. doi:10.1016/0167-8396(90)90035-P

[5] Elber, G.; Kim, M.-S.: Geometric constraint solver using multivariate rational spline functions, Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, 2001, 1-10. doi:10.1145/376957.376958

[6] Grandine, T. A.; Klein, F. W.: A new approach to the surface intersection problem, Computer Aided Geometric Design, 14(2), 1997, 111-134. doi:10.1016/S0167-8396(96)00024-6

[7] Hur, S.; Oh, M.; Kim, T.: Classification and resolution of critical cases in Grandine and Klein's topology determination using a perturbation method, Computer Aided Geometric Design, 26(2), 2009, 243-258. doi:10.1016/j.cagd.2008.03.003

[8] Kim, K.-J.; Kim, M.-S.: Torus/sphere intersection based on a configuration space approach, Graphical Models and Image Processing, 60(1), 1998, 77-92. doi:10.1.1.98.2638

[9] Limaiem, A; Trochu, F.: Geometric algorithms for the intersection of curves and surfaces. Computers and Graphics 19(3), 1995, 391–403. doi:10.1016/0097-8493(95)00009-2

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

[11] Mourrain, B.; Pavone, J. P.: Subdivision methods for solving polynomial equations, Journal of Symbolic Computation, 44(3), 2009, 292–306. doi:10.1016/j.jsc.2008.04.016

[12] Neff, C. A.: Finding the distance between two circles in three-dimensional space, IBM Journal of Research and Development, 34 (5), 1990, 770-775. doi:10.1147/rd.345.0770

[13] Patrikalakis, N. M.; Surface-to-surface intersections. IEEE Computer Graphics and Applications, 13(1): 1993, 89–95. doi:10.1109/38.180122

[14] Sherbrooke, E. C.; Patrikalakis, N. M.: Computation of the solutions of nonlinear polynomial systems, Computer Aided Geometric Design, 10 (5), 1993, 379-405. doi:10.1016/0167-8396(93)90019-Y

[15] Walker, R. J.: Algebraic Curves, Princeton University Press, Princeton, 1985.

[16] Wu, S. T.; Andrade, L. N.: Marching along a regular surface/surface intersection with circular steps, Computer Aided Geometric Design, 16(4), 1999, 249-268. doi:10.1016/S0167-8396(98)00048-X

[17] Wu, S. T.; Osmar, A.; Costa, S. I. R.: A complete and non-overlapping tracing algorithm for closed loops, Computer Aided Geometric Design, 22(6), 2005, 491-514. doi:10.1016/j.cagd.2005.01.005