A Method for Surface/Surface Intersection

Hyung-Bae Jung*

(*Received April 27, 1994*)

A method for solving the problem of surface/surface intersection in suggested. This method uses an adaptive subdivision and a facetted model based on triangles. A bounding box and a different dividing method are newly developed. Instead of the calculation of the characteristic points, a new linking technique is developed. This method solves the intersection problem between bicubic or higher grade surfaces. The desired caculation-precision is specified by the input parameter.

Key Words : Bounding Box, Characteristic Point, Singular Point, Cusp, Linking

1. Introduction

The calculation of intersection curves between surfaces is an important problem in the geometric modeling and in CAD/CAM. Intersection curves are used not only for graphic displays but also with the finite element method and in the production technique, specially for the control of roboters or CNC machines.

For the surface we use two representations :

$$\text{implicit} : F(x, y, z)=0 \quad (1)$$

$$\text{parametric} : G=G(u, v). \quad (2)$$

From these two representations three combinations are possible :

$$\text{implicit+implicit} : F_1(x, y, z)=0, \quad F_2(x, y, z)=0 \quad (3)$$

$$\text{implicit+parametric} : F(x, y, z)=0, \quad G=G(u, v) \quad (4)$$

$$\text{parametric+parametric} : G_1=G_1(u, v), \quad G_2=G_2(s, t). \quad (5)$$

In each case the intersection curves are the simultaneous solution of the two surface representations. Many scientific publications presently discuss this problem. For intersection curves between biquadric surfaces the solution is already well known (Miller, 1987 ; Levin, 1979 ; Sarraga, 1983 ; Pfeiler, 1985 ; Sabin, 1976 ; Pilz, 1989 ; Beyer, 1989 ; Farouki, 1987a). But the calculation of intersection curves between two surfaces of higher degrees than biquadric is not yet solved fully, in spite of that they are used preferentially by engineer.

There are many attempts to solve this problem in the world, but no perfect algorithm is introduced. Each algorithm has several problem, specially with the characteristic points. We can classify following four main categories for these attempts.

1.1 Algebraic method

The initial popular method is the algebraic. Through the use of elimination method one can achieve the classical result. With the problem implicit+parametric one can put the parametric function $P=P(u, v)$ in the implicit function :

$$F(x(u, v), y(u, v), z(u, v))=H(u, v)=0. \quad (6)$$

This equation define the common intersection curve. One can solve algebraic implicit function with different methods, for example Sylvester's resultant, Bezout-resultant etc. Or they are treated numerically for example by a Newton-Raphson method, that calculate the zero position of the nonlinear equation.

If one can convert the implicit+implicit or parametric+parametric in the implicit+par-

* Department of Mechanical Engineering, Mokpo Natinal University, 61, Dorim-Ri, Chyngge-Myun, Muan-Gun, Cheonnam 534-729, Korea

ametric, the geometric modeling will be more simple, because then the select of representation method is open for modeller. But these conversions are possible with only lower polygon degree. For the implicit+implicit it has been tried to transform implicit representation in the parametric representation(Sederberg, 1984a ; Sederberg, 1984b ; Sederberg, 1985a ; Sederberg, 1985b ; Sederbrg 1986a ; Sederberg, 1987 ; Goldman, 1985 ; Abhyankar, 1987a ; Abhyanhar, 1987b). But the general transformation methods are not known until now. To know the transformation-possibility, Katz has defined the "genus" with the connection of the degree of function and the singular point(Katz, 1988). Sederberg has presented two parametrization methods through the use of baryzentric coordinate for cubic algebraic surface(Sederberg, 1987). But the transformation in the parametric transformation is possible in general only for the polynom to second degree.

The transformation of parametric+parametric to parametic+implicit is more simple. Sederberg has developed a general method through the use of analytical mathematics(Sederberg, 1984a ; Sederberg, 1984b ; Sederberg, 1985b). This method make it possible to transform all rational-polynomial types. For the special case between two curves of lower degrees this method is attractive and even faster than subdivision for determining the intersection points(Sederberg, 1986b). But the algorithm for the practical implementation of this method has not yet been developed. The difficulties lie in the geometrical complex. The order of the implicit representation of a parametric surface is 2mn in general. The order of the resulting intersection curve between two parametric surfaces is $4m_1m_2n_1n_2$, for example two bicubic surfaces yield a polynomial of degree 324(Sederberg, 1985a ; Chandru, 1987). A numerical handling of this method is in general great problematical except some special cases. Two new algebraic method are recently developed for the problem implicit+implicit. Garity has introduced the mapping approachs(Garrity, 1989). This method minimize the number of the variables as far as possible, but raise the algebraic degree. In contrast to it, Hoffmann has tried the "geometrical approach"(Hoffmann, 1990). This method minimize the algebric degree, but raise the number of the variables.

1.2 Lattice evaluation

The lattice method evaluate a(or two) surface function on a parameter-mesh, creating a matrix of points P_{ij}. The found points define the edges or polygons with the connection of each other. The intersection problem surface+surface is changed by it in a series of intersection problem plane/plane or plane/surface. The calculation of intersection points is now linear and relative fast. The calculated intersection points are refined by different methods as for example Newton-Raphson iteration. Lattice evaluation is often used for the calculation contouring and surface-rendering (Griffiths, 1975 ; Evans, 1987 ; Mclain, 1974 ; Petersen, 1984 ; Petersen, 1987 ; Petrie, 1987 ; Satterfield, 1985 ; Shantz, 1988 ; Hartwig, 1983), or as preprocessor for the marching-and subdivision-algorithm. The parameter evaluation fix one of four surface parameter in regular or irregular step(Rossignac, 1987). Therewith three nonlinear equations with three unknowns exist for each this value. The calculation is executed through numerical methods, for example through Newton-Raphson iteration, special algebraic methods(Hoschek, 1987) or through conversion of polynombasis(Hoschek, 1989 ; Hoschek, 1990). The parameter evaluation is often used as preprocessor for marching.

Complete intersection curve is generated through the linking and interpolation between calculated points. This lattice evaluation does not require so good start point as marching, but the completeness of the solution depends on the evaluation step. A gross step can cause the miss of the small close intersection curves or contact points. But the small step has efficiency trouble due to gross data volume. There are some adaptive lattice evaluation methods for the reduce of necessary data volume(Shantz, 1988 ; Hoschek, 1989 ; Hoschek, 1990).

1.3 Marching

The marching method develops the point series of intersection curve step by step. Each intersection curve require start points and the local direction-vector. For the local direction-vector curvature anlysis(Faux, 1981 ; Chen, 1988), local explicit power-series(de Montaudouin, 1986 ; Farouki, 1987b ; Hoffmann, 1987 ; Bajaj, 1988), and a orthogonal projection(Kriezis, 1990b) have been used. Someones have developed the local differential geometry for the problem implicit + implicit(Phillips, 1984 ; Asteasu, 1988). Cheng has used a vector-field for the tracing(Cheng, 1988). A difficulty with this method is it, to find out start points as good as possible. Sometimes lattice evaluation is used for that and after that a Newton-Raphson iteration is used. Another difficulty is it, to select the proper step length. Unproper step length can cause endless loop with tightly locating intersection curves(Geisow, 1983). An important disadvantage of marching method appears on the singular points. On a singular point there are three possibility for the further tracing. Bajaj has proposed a desingularization method based on birational transformation(Bajaj, 1988). This method solves singular problem in case of implicit + parametric. Barnhill has developed a general marching-method(Barnhill, 1987 ; Barnhill, 1990). This method need only one procedure that can evaluate the coordinate-value and the local gradient for a given parameter. For the start points he has used lattice evaluation, subdivision and Newton-Rapson iteration at the same time. For the tracing he has used the character of intersection curve, that the tangent-vectors of two surfaces are same on the intersection curve. But this method converges slowly at singular points(Muellenheim, 1990).

1.4 Subdivision

The basic idea with the subdivision is the dividing of original problems, until a simple method exists to solve it(Kriezis, 1990a ; Kriezis, 1990b ; Lane, 1980 ; Catmull, 1978 ; Sabin, 1978 ; Koparkar, 1983 ; Koparkar, 1986 ; Peng, 1984 ; Dokken, 1985 ; Houghton, 1985 ; Lasser, 1986 ; Li, 1988 ; Lee, 1984 ; Carson, 1982 ; Casale, 1987 ; Casale, 1989). Recursive subdivision-technique for the intersection curves is based on the paradigm of "divide-and-conquer"(Pratt, 1986). A divide-and-conquer algorithm for surfaces divides the boundary of each surface in appropriate surface-segments, eliminates then the segments except in a possible boundary and the remaining segments are divided once more. This procedure is continued until only segments within necessary tolerence are left. They describe the intersection curve exact enough. Generally a bounding box is used for the elimination, such as fat arcs(Sederberg, 1989), min max bounding box and retangular bounding box(Houghton, 1985), convex hull.

Subdivision algorithms are mainly applid to calculate the intersection curve between two parametric surfaces(Peng, 1984 ; Dokken, 1985 ; Houghton, 1985 ; Lasser, 1986 ; Lee, 1984 ; Carson, 1982 ; Casale, 1987). But they are also extended for the implicit + implicit(Owen, 1987) and implicit + parametric(Patrikalarkis, 1990 ; Prakash, 1988 ; Kriezis, 1990b). Sometimes this extension is possible through convert of representation of algebraic curve in a Bernstein-basis within a retangular domain(Geisow, 1983 ; Sederberg, 1984c ; Patrikalakis, 1989). A technique for adaptive step was introduced for the irregular refinement(Lyche, 1985). Finally a linking procedure follows it, in order to construct the complete intersection curve from the points found.

1.5 Additional remark

In this work a method is introduced with which we can find the intersection curves between two parametric surfaces(parametric + parametric), such as B-spline surfaces or Bezier surfaces, of arbitrary degrees. This method solves intersection problem fully. This method uses an adaptive subdivision and a facetted model based on triangles.

A refinement of the parameter is used in order to achieve a global precision of the curves. A bounding box and a different dividing are newly developed. Instead of the calculation of the characteristic points a new linking technique is developed.

We can see that each category has several self-problems. Therefore many hybrid algorithms are tried without sucess. There is no general example to show intersection lines between surfaces of higher degrees than biquadric, specially with the characteristic points. For the first time this algorithm shows that.

2. Problem Statements

Suppose, $x=(x, y, z)$ is a vector of coordinates in three dimensional space and $u=(u, v)$ is a vector in the two dimensional parameter space. Then $x=x(u)$ defines a two dimensional geometric object in three dimensional space : each of the three components of x is a function of the two parameters of u.

Two parametric surfaces are defined :

$$P=P(u, v) \text{ and } Q=Q(s, t). \tag{7}$$

Then the intersection curves are defined :

$$F(u, v, s, t)=P(u, v)-Q(s, t)=0. \tag{8}$$

We can derive three equations, each based on one coordinate :

$$P_x(u, v)-Q_x(s, t)=0 \tag{9}$$
$$P_y(u, v)-Q_y(s, t)=0 \tag{10}$$
$$P_z(u, v)-Q_z(s, t)=0. \tag{11}$$

One can derive the solution for the four parameters with these three equations. It remains an independent parameter which defines the curves, because the curves posess one degree of freedom. But it is difficult, sometimes impossible, to eliminate three parameters. An analytic form for the intersection curve can be determined only in case of a low grade.

3. Solution Method

3.1 Strategy

If we consider a calculated intersection segment, there is always a minimal distance from the each point on the calculated intersection curve to the ideal curve, and vice versa. Ideal curve means here the curve that is mathematically exactly calculated, if we can. The maximum of this minimal distances is the maximal distance between two curves, we call here "global precision".

With the minimal distance from the ideal curve

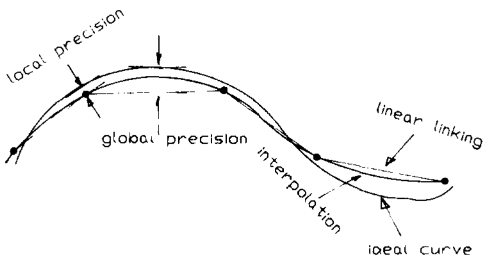


Fig. 1 Local and global precision

to the evaluated point, we call here "local precision". But in this case the distance from the other points to the ideal curve are ignored. Figure 1 explains the difference between local and global precision.

Usually subdivision algorithms calculate relatively few intersection points. An improvement process such as the Newton-Raphson method follows to obtain the precision of each intersection point. The entire intersection curves are then created through linear linking or interpolation between intersection points. But this approach can guarantee only the local precision.

In this work a subdivision algorithm is developed guaranteeing the global precision up to a specified limit. The global precision is very important for industrial applications. In order to reach the global precision a lot of intersection points are necessary so that the entire intersection curve can be determined within a specified limit. A fine refinement approach used in this work is reliable for that goal.

3.2 Model

As input for this intersection method two parametric surface representations and their models are required.The parametric surface representation is a mathematical description of points, which make up the surfaces. A facetted model, in which the relation between points and polygons is saved, is used here. Evaluated lattice points are termed "explicit points". The points, which are the components of surfaces, but are not discretized for the present, are called "implicit points". These points will be calculated later with requirements from the parametric surface representation.

For this intersection method two quadrangle

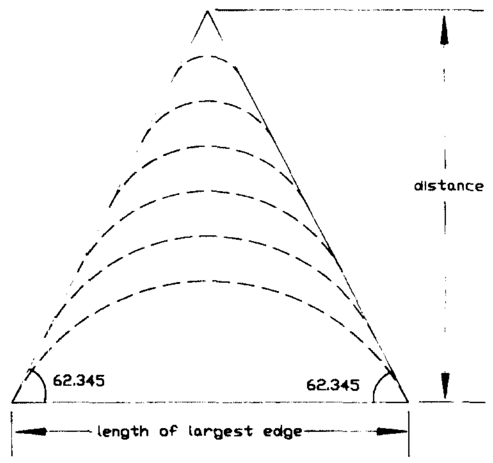


Fig. 2 Maximal angle and distance

grids of explicit points are necessary in the input phase. But triangles are used internal which can be generated easily from the quadrangles.

In general the difficulty with the calculation of intersection curves between two free form surfaces is caused from that the free-form surfaces sometimes have many geometric odd form. If the user has the possibility that informs the geometric specialties of his odd objects to the algorithm, the calculaton will be very flexible. Among the many geometric specialties the developed algorithm uses one criterion, curvature, through the use of following model:

At least one explicit point must be discretized in the area of critical curvature, if the angle between any gradient on a curve of the object and the approximating triangle exceeds 62 degree.

This model gives us the interesting characteristic that the length of the larger edge of a triangle with any further refinement is always larger than the distance between the approximating triangle and the real surface. Figure 2 explains this. If the distance is larger than the largest edge, the angle must be larger than 62.345(arctan(2.) = 62.345°).

The user determines himself the approximating model of his objects. This gives him the possibility to inform the special curvature of his objects to the algorithm. In normal there is no special curvature with the grater than 62.345° in free-form surface. A constructor who generates the parametric surface for a object, knows the geometric form of this object exactly. Even if anyone uses the generated parametric surfaces, he has to have a certain conception on the geometric form of these surfaces. Therefore it is not difficult for him to discretize his object for the upper model. If one neglect to satisfy upper model requirement, this method could miss small intersection loop.

3.3 Refinement procedure

A subdivision algorithm consists of three parts, namely the refinement procedure, the calculation procedure, and the linking procedure. The first procedure, the refinement procedure, reduces the pending patch sets. For reducing the pending patch set two processes are used. One is dividing, which divides pending boundary in many subpatch sets and the other is elimination, which eliminates the subpatchs with no intersection. Each step of refinement reduces the pending area through the elimination process.

The refinement procedure functions very similary as a searching procedure. In the univariate case there are some searching procedures, such as the halving of the parameter values or gold section search. Here is not one variable but two variables in two surfaces. Therefore not one interval but two surfaces are reduced with each refinement step. The refinement procedure converges to the intersection curve up to specified limit. The ideal case is the reduction of two surfaces until only the simultaneous curves remain. Figure 3 shows an example for that.

A dividing of entire surface will increase the amount of data exponentially. Therefore the adaptive refinement approach that refines the pending boundary step by step is used. After each refinement step the elimination follows. The elimination process makes sure that the areas with no intersection are found and eliminated. Instead of the computation of the intersection between two patches, a check for intersection between two bounding boxes is performed. The use of a bounding box avoides complex computations and improves the efficiency of the computation. A newly developed bounding box allows for an

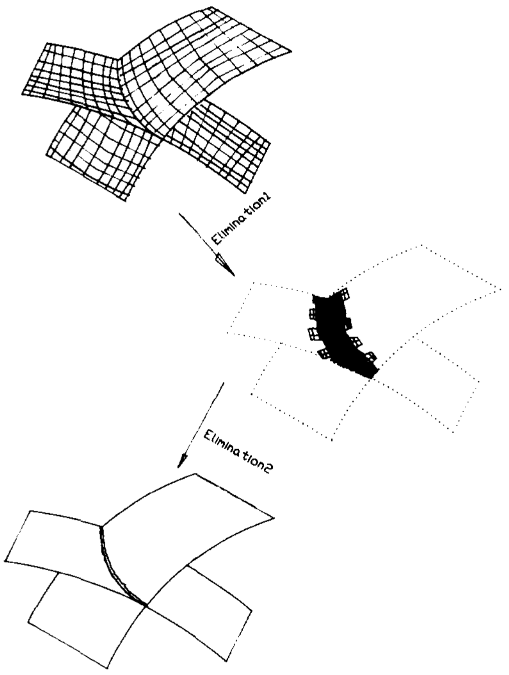


Fig. 3 An example for the refinement procedure

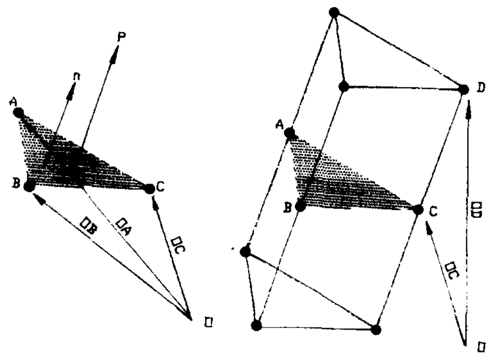


Fig. 4 Bounding box for the surface Q(s, t)

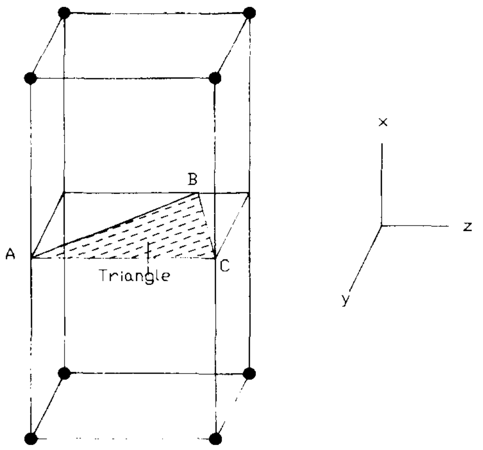


Fig. 5 Bounding box for the surface P(u, v)

adaptive subdivision with less data produced than normally expected.

The refinement steps will continue until finally only subpatches remain within the scope of specified limit. This procedure is controled by the segment length, inputl. The desired global precision, namely the allowable error of the calculation is specified by the inputl. Inputl represents an upper bound for the segment length, namely this procedure is interrupted, if the segment length is smaller than the inputl. The inputl is assumed here as relatively small. With relatively small inputl this method will have more advantages than older methods.

3.4 Bounding box

A bounding box is in general defined such that it can contain the entire concerning subpatch. Here a new bounding box is developed. For its construction directly evaluated points from the representation are used. An expansion, which corresponds to the length of largest edge, follows in both normal directions.

All bounding boxes of $Q(s, t)$, as shown in Fig. 4, are constructed at the beginning of the refinement procedure. But those of $P(u, v)$ are constructed just after transformation. The normal vector for special expansion after the transformation is parallel to a coordinate axis. Therefore the calculation of the normal vector for the bounding box with $P(u, v)$ is not needed. But the bounding box with $P(u, v)$, as shown in Fig. 5, is not a prism but a quader. One can test very simple the intersection possibility with min/max values of three coordinates of a quader, such as the min/max bounding box.

The transformation is used here and simplifies the calculation. The transformation is executed for all triangles of $P(u, v)$. At the beginning of the transformation the coordinate origin is placed at an end point of a triangle. A coordinate axis is put along an edge of the triangle. Finally the coordinate system is rotated around the fixed axis,

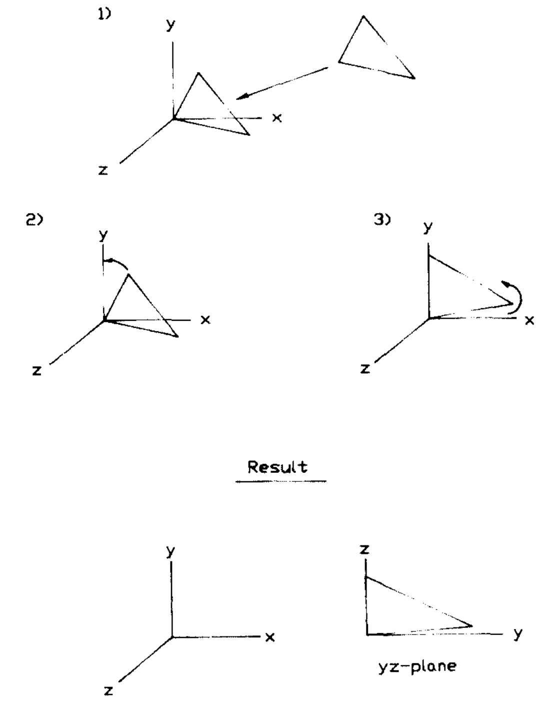


Fig. 6 Transformation process

until a zero-plane of one coordinate reaches the third point. Here the x-coordinate is selected arbitrarily as zero-plane. Figure 6 shows this process. This process is performed simply with help of a 4×4 matrix and homogeneous coordinates.

3.5 Different dividing

To reduce the temporary data size different dividings are used according to the type of critical set. Two critical sets are distinguished: "direct critical sets" and "potential critical sets". The direct critical set are the subpatches, that can be selected as intersection subpatches through comparison with two triangles without the help of bounding boxes. The potential critical set are the subpatches, that can be selected as intersection subpatches additionally considering the each of precision, namely through the use of the bounding box. As explained in Fig. 6, the base triangle lies in the plane x=constant after the transformations.

$Q(s, t)$ is transformed on the basis of the model of $P(u, v)$. After each transformation the x-coordinate is zero. If an intersection point is found between two end points i, j of the models of $Q(s, t)$ the following condition must be met:

$$x_i * x_j \leq 0. \qquad (12)$$

This can be tested simply with the signs. Only with different signs or with zero this condition is met. Each triangle of the model of the surface $Q(s, t)$ is tested with two edges for the possiblity of intersection. Two tests are sufficient, to test the possibility of intersection with a triangle. The test of the remaining edge is redundant. In this manner the direct critical set is defined. The remaining area is defined automatically as the potential intersection area. Both intersection areas are divided differently, namely the direct intersection area is divided by four(2×2) and the potential intersection area by nine(3×3). This speeds up the entire process.

3.6 Calculation procedure

After the refinement procedure there are two critical patch sets for two models. They are already refined up to the global precision. The fine refinement approach simplifies the calculation procedure because inspite of linear calculation the precision of intersection curves is guaranteed.

To find the direct intersection area transformations are also used here. After the transformations the x-coordinates of both end points of an edge are observed. If they differ in sign, they lie on different sides of the base triangle. In these cases the intersection point is calculated proportionally. The calculated data for each point, namely three euclidic coordinates and four parameter values, are saved in the data structure. Additionally the numbers of both triangles are saved, in order to help the procedure of linking. Finally it is tested for all intersection points, whether the same points appear frequently, because the same line in two neighbouring triangles constitutes an edge twice.

As a result of the procedure of calculation so many intersection points are obtained that the intersection curves can be represented rather exact. Normally the data size is not so large that

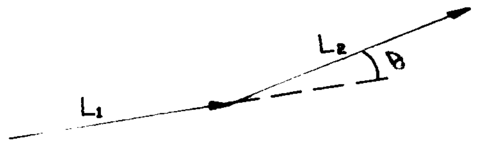


Fig. 7 Connection of connection angle

there is a problem with the data management. For example, with a length of the intersection curve of 10 m one can constitute it with 10000 intersection points in a distance of 1 mm. This process has no problem to current data management system.

3.7 Linking procedure

The intersectionm points are not determined in a continuous sequence. Therefore a linking procedure is necessary for the intersection method. After the linking procedure the intersection points are connected each other direct linear.

The connection angle is the angle between two connecting intersection lines. For example the connection angle in Fig. 7 is :

$$\theta = \arccos(L1 * L2/|L1| * |L2|). \tag{13}$$

This procedure is controlled by two input parameters : input2 and input3. The second step of refinement will begin, if the connection angle is bigger than input2 and the distance from starting point is larger than input3. The second step of refinement ends, if the connection angle is smaller than input2 or the distance is shorter than input3. Here is input3<input1. In order to prevent an endless loop, input3 and input4 are used. Input3 and input4 respectly form a lower bound. If the distance remains below input3, the refinement is stopped. If the distance remains below input4, the linking will follow without the test of the connection angle.

The test of connection angle gives us the possibility, to find out a few characteristic points, as cusp or singular points, because these points are very sensitive with regard to connection angle. This fact is very important for the calculation of exact intersection curve.

3.8 Characteristic points

Here the cases of characteristic points are explained. In this work it is not tried, to find the characteristic points in advance, instead they are found automatically and approximately. This approximation can not reflect the real topology of the intersection curve, but it can approximate them within a given tolerance. In a special area it is refined further up to the tolerance. If nevertheless the linking is not sucessful, a characteristic point is supposed there.

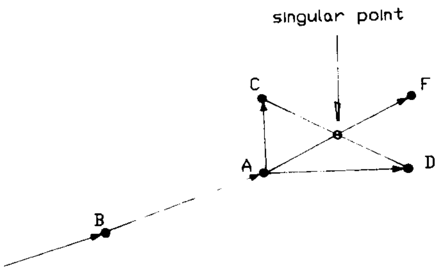


Fig. 8 Linking for singular point

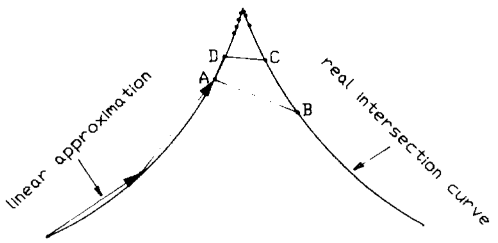


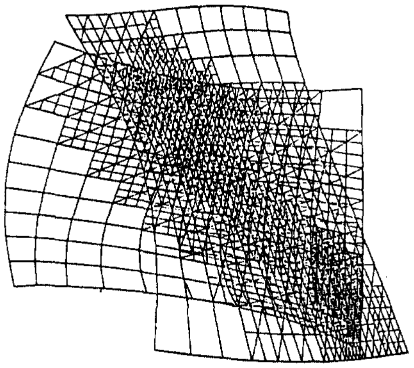
Fig. 9 Linking for cusp

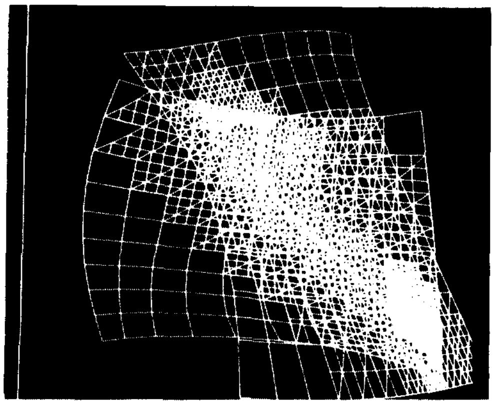
Figure 8 explains the case of singular points. By Eq.(13)

$$\theta_f = BA * AF/|BA| * |AF| \tag{14}$$
$$\theta_c = BA * AC/|BA| * |AC| \tag{15}$$
$$\theta_d = BA * AD/|BA| * |AD|. \tag{16}$$

Suppose, θ_c and θ_d are larger than input2, but θ_f is smaller than input2. The next point from point A is searched. Point C is selected first, because it has a permissble distance according to input1 from point A and lies before point D in the temporary table of intersection points. But the test of the connection angle disqualifies it as a next point. The points are tested in succession. The test of the connection angle remains unsuccessful to D. But finally F is taken as next point, saved and removed from the temporary table. Therefore there is no problem with singular points.

With Fig. 9 the case of cusp is explained. The


(a) After the refinement


(b) After the connection

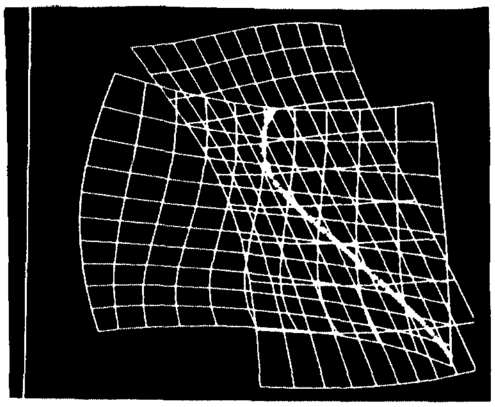

(c) Intersection curve

Fig. 10 Bicubic-bicubic, 95 intersection points

next point from starting point A is searched. Point B is taken as a candidate in the search process, but fails the test of the connection angle. The search process finds then no more candidate. Therefore it is refined further in this area. After successful calculation there are two new intersection points C and D. With the searching of the next point D is taken new intersection point. Now the starting point moves to point D. With point D exactly the same problem as before exist, to search the next point. Then it is refined again. This process is repeated until the segment length reaches input3. This process refines the critical patch set with a cusp exact sufficiently. The case of turning points is similar to the cusp.

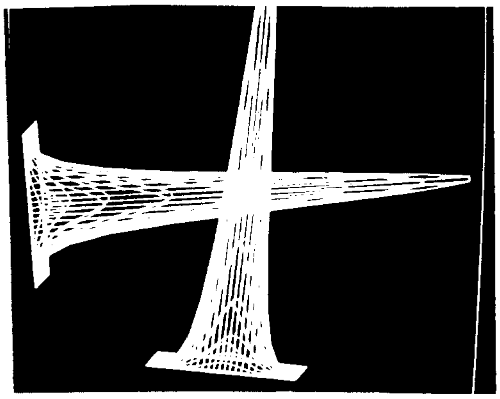
4. Results and Discussion

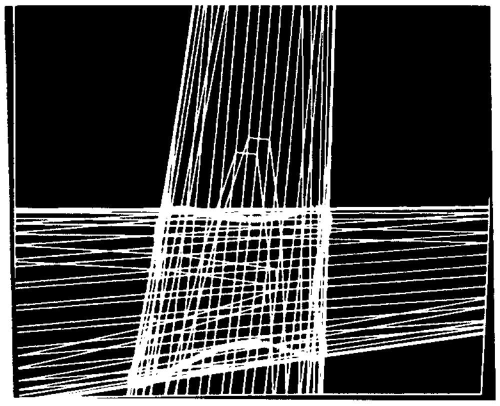
Seven examples(Fig. (10)~(16)) are presented here. They show the application of this algorithm for the intersection problem between bicubic or higher grade surfaces. Here nonuniform rational B-spline surfaces(NURBS) are used. Intersection points are marked with "+".

The pictures of Fig. 10 represent a simple normal case of intersection between two bicubic parametric surfaces. Figure 10(a) shows the polygons after the refinemet. The area that the polygons are concentrated means that many refinement steps are executed. Normally the polygons are concentrated along the intersection line. Figure 10(b) adds the intersection line to Fig. 10(a) 95 intersection points are calculated and linear connected. Figure 10(c) shows only the intersection line.

The pictures of Fig.11 show the more complicated case of intersection between two bicubic parametric surfaces. Two conic figures are selected, because the intersection problem between them give us special difficulties so as the turning points, singular points and cusps. 456 intersection points are calculated and the grate number of intersection points means indirectly that many refinement steps are executed. Figure 11(a) shows the intersection line, Fig. 11(b) the enlargement and Fig. 11(c) the enlargement of another sight.

The pictures of Fig. 12 show the general case of intersection between two biquintic parametric surfaces. Two intersection lines are calculated with 251 intersection points. Two views of different sights are shown.


(a) Intersection curve


(b) Enlargement : view 1

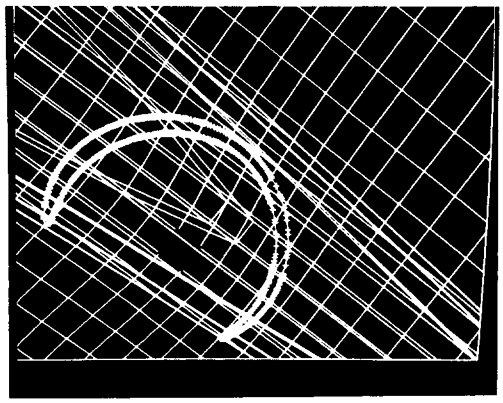
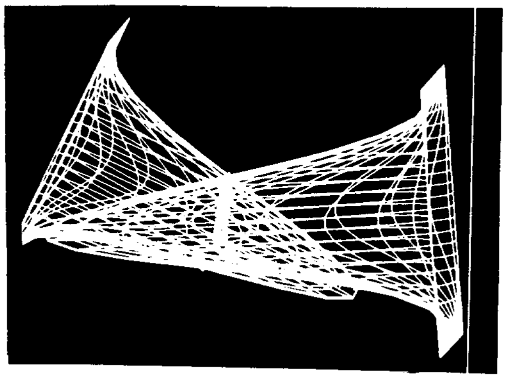

(c) Enlargement : view 2

Fig. 11 Bicubic-bicubic, 456 intersection points


(a) View 1

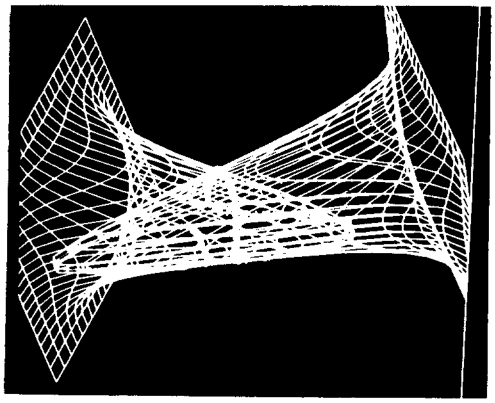
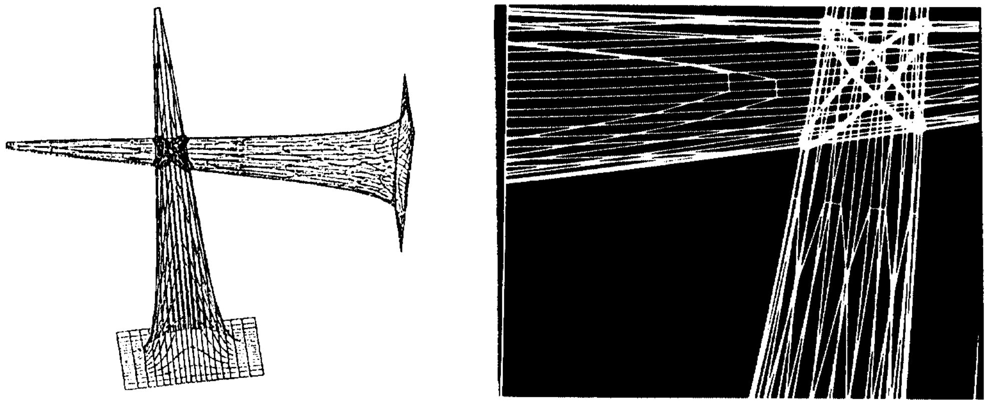
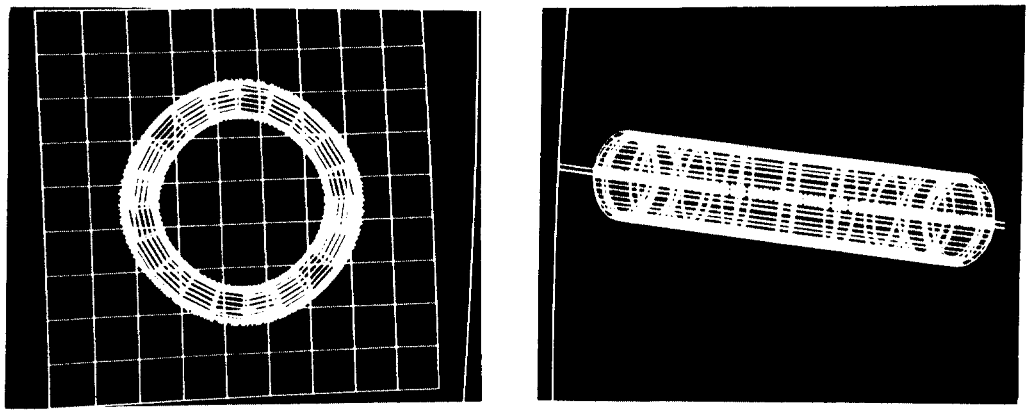

(b) View 2

Fig. 12 Bicubic-biquintic, 251 intersection points



(a) Intersection curve (b) Enlargement

Fig. 13 Bicubic-bicubic, 616 intersection points



(a) View 1 (b) View 2

Fig. 14 Torus-plain, bicubic-bicubic, truning points, 501 intersection points

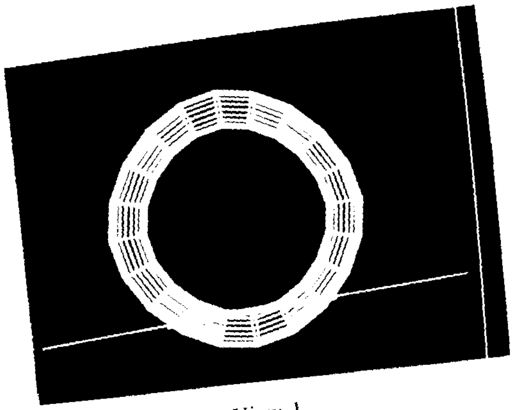
The pictures of Fig. 13 show two intersection lines with two singular points between two conic figures. Two bicubic parameter surfaces are used and 616 intersection points are calculated.

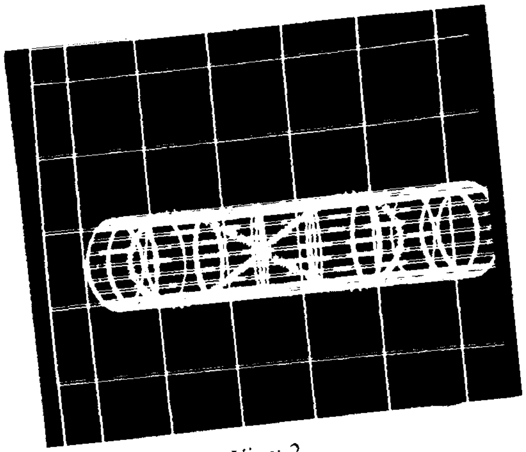
The pictures of Figs. (14)~(16) show the characteristic points. To show the characteristic points evidently the special figures are selected intentionally. The pictures show the intersection between torus and plain, bicubic-bicubic.

Normally the data size is the important problem of subdivision. To reduce data size this algorithm uses the adaptive subdivision and different dividing. Neverthless there are often the same problem with the small precision. Together with the development of computer science this problem will be solved.

5. Conclusions

A method for sloving the problem of surface/surface intersection is suggested. This method can solve the intersection problem between bicubic or higher grade surfaces exactly. This method uses an adptive subdivision and a facetted model based on triangles. A bounding box and a different dividing method are newly developed. Instead of the calculation of the characteristic points, a new linking technique is developed.


(a) View 1


(b) View 2

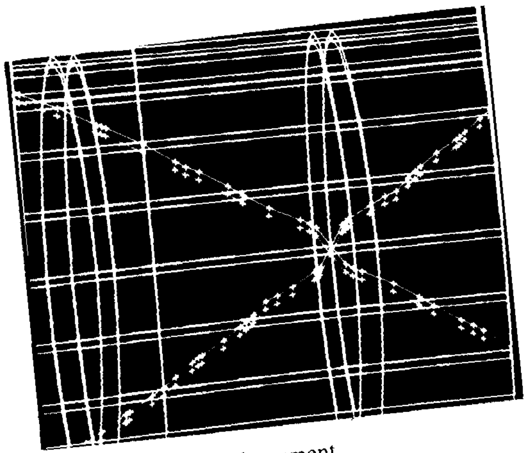
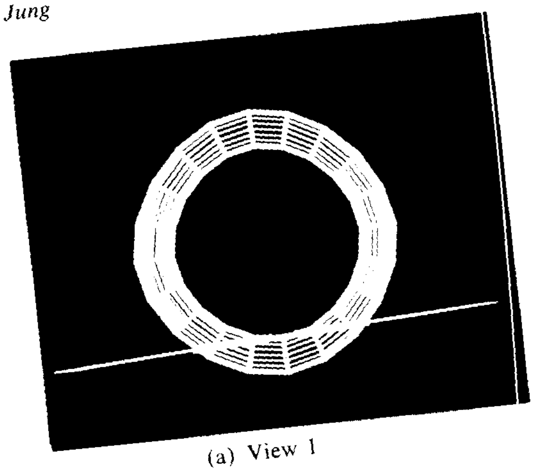
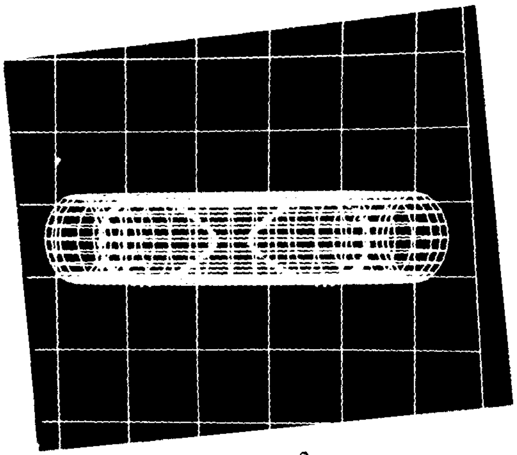

(c) Enlargement

Fig. 15 Torus-plain, bicubic-bicubic, singular points, 214 intersection points


(a) View 1


(b) View 2

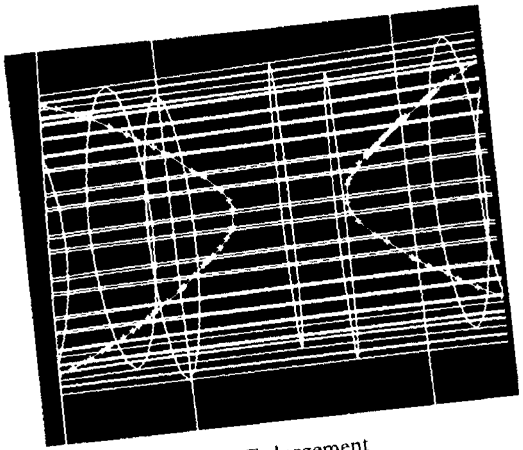

(c) Enlargement

Fig. 16 Torus-plain, bicubic-bicubic, cusp 310 intersection points

Appendix

Algorithm

Following abbreviations are used in algorithm :

PP	: current triangle of $P(u, v)$
PQ	: current triangle of $Q(s, t)$
TS	: temporary table of intersection points
NSP	: the number of intersection point in the *TS*
$V1$	: vector from the last point to starting point
$V2$	: vector from starting point to current point

The algorithm uses following input-and output -parameter :

vertp	: point table of model of $P(u, v)$
vertq	: point table of model of $Q(s, t)$
facep	: polygon table of model of $P(u, v)$
faceq	: polygon table of model of $Q(s, t)$
fdp	: table of coefficients and control points of $P(u, v)$
fdq	: table of coefficients and control points of $Q(s, t)$
vtp	: point table of pending patch set of $P(u, v)$
vtq	: point table of pending patch set of $Q(s, t)$
fcp	: triangle table of pending patch set of $P(u, v)$
fcq	: triangle table of pending patch set of $Q(s, t)$
sch	: *TS*
link	: linking table of intersection curves.

```
Procedure : Refinement(vertp, vertq, facep,
            faceq, fdp, fdq, input1, vtp, vtq, fcp,
            fcq)
Make the triangular grids from facep and faceq
While : refinement request(exist triangles in fcp
        and fcq, whose maximal edge length has
        larger than input1)
  for : all triangles of $Q(s, t)$
      construct bounding box
  endfor
  for : all triangles of $P(u, v)$
    Transform $PP$
    Construct bounding box
    $MKP$=maximal edge length of $PP$
    for : all triangles of $Q(s, t)$
      Transform bounding box of on the basis of
      $PP$
      Compare two bounding boxes of $PP$ and
$PQ$
      if(intersected) then
        $MKQ$=maximal edge length of $PQ$
        if($MKP <$ input1) then
          Save data of $PQ$
        else
          Save the number of $PQ$
        endif
      endif
    endfor
    if(intersected) then
      if($MKP <$ input1) then
        Save data of $PP$
      else
        Save the number of $PP$
      endif
    endif
  endfor
  Test, whether the one triangle appears only one
  times in pending subpatch sets and in this case
  only one of them is remained and the others are
  removed.
  Subdivide the direct critical set with 9 and save
  the generated data
  Subdivide the potential critical set with 4 and
  save the generated data
  Define the new critical(pending) patch sets
endwhile
endprocedure

  Procedure : Calculation(vtp, vtq, fcp, fcq, sch)
for : all triangles of $P(u, v)$
  Transform $PP$
  for : all triangles of $Q(s, t)$
    Transform $PQ$ on the basis of $PP$
    if(direct critical set) then
      calculate the intersection point and save in
$\rightarrow TS$
    endif
  endfor
endfor
```

```
Test, whether the one point appears only one times.
endprocedure

  Procedure : Linking(vtp, vtq, fcp, fcq, sch, fdp, fdq, input1, input2, input3, input4, link)
while : NSP>0
  if(first point) then
    Define new intersection curve
    Select the starting point
    Remove the starting point from the $TS$
  elseif(second point) then
    Select the second point
    Calculate $V2$
    Link to the second point
    Starting point=second point
    $V1=V2$
    second point is removed from $TS$ and saved in the link table
  else
    for : all intersection points in $TS$
    Set default value in 0(m$x$1, m$x$2, n1, n2)
    Define the distance
    if(distance<input1) then
      Calculate $V2$ and connection angle
      if(connection angle<input2) then
        if(first times) then
          m$x$1=distance, n1=current point
        else
          if(distance<m$x$1) then
            m$x$1=distance, n1=current point
          endif
        endif
      else
        if(the first times) then
          m$x$2=ditance, n2=current point
        else
          if(distance<m$x$2) then
            m$x$2=distance, n2=current point
          endif
        endif
      endif
    endif
  endfor
  if(m$x$2<input4) then
    Link to n2
    Starting point=n2
    $V1=V2$
    n2 is removed from the $TS$ and saved in the link table
  endif
  if(n1 exist) then
    Link to n1
    Starting point=n1
    $V1=V2$
    n1 is removed from the $TS$ and saved in the link table
  else
    Subdivide following two areas with n2
      1. Parameter area between two intersection points
      2. The triangles, which the both points(starting point and current point) are produced by the intersection of each other, that is two triangles of $P(u, v)$ and two triangles of $Q(s, t)$
    for : all triangles generated by the subdivision in $P(u, v)$
      $MKP$=maximal edge length of $PP$
      if($MKP$>input3) then
        Transform $PP$
        for : all triangle generated by the subdivision in $Q(s, t)$
          $MKQ$=maximal edge length of $PQ$
          if($MKQ$>input3) then
            Transform $PQ$ on the basis of $PP$
            Calculate the new intersection point in the direct critical set and save in the $TS$
          endif
        endfor
      endif
    endfor
    Test, whether the same point appears two times and in this case one of them is removed.
  endif
  if(no new intersection point are found through the subdivision) then
    Define new intersection curve
  endif
endif
endwhile
endprocedure
```

References

Abhyankar, S. S. and Bajaj, C., 1987, "Automatic Parameterization of Rational Curves and Surfaces I : Conics and Conicides," *CAD*, pp. 11 ~14.

Abhyankar, S. S. and Bajaj, C., 1987, "Automatic Parameterization of Rational Curves and Surfaces II : Curves and Cubicoids," *CAD*, pp. 499~502.

Asteasu, C., 1988, "Intersection of Arbitary Surfaces," *CAD*, pp. 533~538.

Bajaj, C., Hoffmann, C. M., Lynch, R. E. and Hopcroft, J.E.H., 1988, "Tracing Surface Intersection," *CAGD*, pp. 285~307.

Barnhill, R. E., Farin, G., Jordan, M. and Piper, B. R., 1987, "Surface/Surface Intersection," *CAGD*, pp. 3~16.

Barnhill, R. E. and Kersey, S. N., 1990, "A Marching Method for Parametric Surface/Surface Intersection," *CAGD*, pp. 257~280.

Beyer, G., 1989, "Entwurf von Bezierkurven Fuer den Schnitt Zweier Quadriken," Ph. D. Thesis, TU Dresden.

Carlson, W., 1982, "An Algorithm and Data Structure for 3D Object Synthesis Using Surface Path Intersections," *Computer Graphics*, pp. 255 ~259.

Casale, M. S., 1987, "Free Form Solid Modeling with Trimmed Surface Patches," *IEEE Computer Graphics and Applications*, pp. 33~43.

Casale, M. S. and Bobrow, J. E., 1989, "A Set Operation Algorithm for Sculptured Solids Modelled with Trimmed Patches," *CAGD*, pp. 235~247.

Catmull, E. and Chark, J., 1978, "Recursively Generated *B*-Spline Surfaces on Arbitary Topological Meshes," *CAD*, pp. 350~355.

Chandru, V. and Kochar, B. S., 1987, "Analytic Techniques for Geometric Intersection Problems," In Farin G.(ed) : Geometric Modeling, Algorithms and New Trends, SIAM, pp. 305 ~319.

Chen, J. J., 1988, "Predictor-Corrector Type of Intersection Algorithm for C^2 Parametric Surfaces," *CAD*, pp. 347~353.

Cheng, K. P., 1988, "Using Plane Vector Fields to Obtain All the Intersection Curves of Two General Surfaces," In Strasser W.(ed) : Theory and Practice of Geometric Modeling pp. 187 ~203.

De Montaudouin, Y., Tiller, W. and Vold, H., 1986, "Applications of Power Series in Computational Geometry," *CAD*, pp. 514~524.

Dokken, T., 1985, "Finding Intersections of *B*-Spline Represented Geometries Using Recursive Subdivision Techniques," *CAGD*, pp. 189 ~195.

Evans, B. M., 1987, "View from Practice," *CAD*, pp. 203~211.

Farouki, R. T., 1987, "Direct Surface Section Evaluation," In Farin G.(ed) : Geometric Modeling, Algorithms and New Trends, SIAM, pp. 319 ~334.

Farouki, R. T., 1987, "Trimmed-Surface Algorithm for the Evaluation and Interrogation of Solid Boundary Representations," *IBM Jounal Res. Develop.*, Vol. 31 pp. 314~332.

Faux, I. D. and Pratt, M. J., 1981, "Computational Geometry for Design and Manufacture," Ellis Horwood Chichester England.

Garrity, T. and Warren, J., 1989, "On Computing the Intersection of a Pair of Algebraic Surfaces," *CAGD*, pp. 137~153.

Geisow, A., 1983, "Surface Interrogations," Ph. D. Theisis, University of East Angelia, Norwich U. K.

Goldman, R. N., 1985, "The Method of Resolvents : A Technique for the Implicitization, Inversion and Intersection of Non-Planar, Parametric, Rational Cubic Curves," *CAGD*, pp. 237~255.

Griffiths, J. G., 1975, "A Data-Structure for the Elimenation of Hidden Surfaces by Patch Subdivision," *CAD*, pp. 171~178.

Hartwig, R. and Nowacki, H., 1983, "Isolinien und Schnitte in Coonschen Flächen," Informatik-Fachberichte 65, pp. 329~343.

Hoffmann, C. M. and Lynch, R. E., 1987, "Following Space Curves Numerically," Report No. *CSD-TR*-684 Computer Science Department, Purdue Univgersity.

Hoffmann, C. M., 1990, "A Dimensionality Paradigm for Surface Interrogation," *CAGD*, pp.

517～532.

Hoschek, J., 1987, "Algebraische Methoden zum Glätten und Schneiden von Splineflächen," *Result in Mathematics* Vol. 12, pp. 119～132.

Hoschek, J. and Schneider, F. J., 1989, "Spline Konversion Fuer Getrimmte Rational Bezier-und B-Splineflächen," *Workshop in Darmstadt.*

Hoschek, J. and Schneider, F. J., 1990, "Spline Conversion for Trimmed Rational B-spline Surfaces," *CAD*, pp. 580～590.

Houghton, E. et al., 1985, "Implementaion of Divide-and-Conquer Method for Intersection of Parametric Surfaces," *CAGD*, pp. 173～183.

Katz, S. and Sederberg, T. W., 1988, "Genus of the Intersection Curve of Two Rational Surface Patches," *CAGD*, pp. 253～258.

Koparkar, P. A. and Mudur, S. P., 1983, "A New Class of Algorithms for the Processing of Parametric Curves," *CAD*, pp. 41～45.

Koparkar, P. A. and Mudur, S. P., 1986, "Generation of Continuous Smooth Curves Resulting from Operation on Parametric Surface Patches," *CAD*, pp. 193～206.

Kriezis, G. A., Prakash, P. V. and Patrikalakis, N. M., 1990a, "Method for Intersecting Algebraic Surfaces with Rational Polynomial Patches," *CAD*.

Kriezis, G. A., 1990, "Algorithms for Rational Spline Surface Intersections." Ph. D. Thesis, MIT.

Lane, J. M. and Riesenfeld, R. F., 1980, "A Theoretical Development for the Computer Display and Generation of Piecewise Polynomial Surfaces," *IEEE Transactions PAMI2*, pp. 35～46.

Lasser, D., 1986, "Intersection of Parametric Surfaces in The Bernstein-Bezier Representation," *CAD*, pp. 186～192.

Lee, R. B. and Fredricks, D. A., 1984, "Intersection of Parametric Surfaces and a Plane," *IEEE CG&A*, pp. 48～51.

Levin, J. Z., 1979, "Mathematical Models for Determining the Intersections of Quadric Surfaces," *Computer Graphics and Image Processing* Vol. 11, pp. 73～87.

Li, L., 1988, "Hidden-Line Algorithm for Curved Surfaces," *CAD*, pp. 446～470.

Lyche, T. and Cohen, E., 1985, "Knot Line Refinement Algorithms for Tensor Product B-Spline Surfaces," *CAGD*, pp. 133～139.

Mclain, D. H., 1974, "Drawing Contours from Arbitary Data Points," *The Computer Jounal* Vol. 17, pp. 328～324.

Miller, J. R., 1987, "Geometric Approaches to Nonplanar Quadric Surface Intersection Curves," *ACM Trans Graphics*, pp. 274～307.

Muellenheim. G., 1990, "Convergence of a Surface/Surface Intersection Algorithm," *CAGD*, pp. 415～423.

Owen, J. C. and Rockwood, A. P., 1987, "Intersection of General Implicit Surfaces," In Farin G. (ed) : Geometric Modeling, Algorithms and New Trends, SIAM, pp. 335～345.

Patrikalakis, N. M. and Kriezis, G. A., 1989, "Representation of Piecewise Continuous Algebraic Surfaces in Terms of B-Splines," *The Visual Computer*, pp. 360～375.

Patrikalakis, N. M. and Prakash, P. V., 1990, "Surface Intersections of Geometric Modeling," *Jounal of Mechanical Design, ASME Transactions*, pp. 100～107.

Peng, Q. S., 1984, "An Algorithm for Finding the Intersection Lines Between Two B-Spline Surfaces," *CAD*, pp. 191～196.

Petersen, C. S., 1984, "Adaptive Contouring of Three-Dimensional Surfaces," *CAGD*, pp. 61～74.

Petersen, C. S., Piper, B. R. and Worsey, A. J., 1987, "Adaptive Contouring of a Trivariate Interpolant," In Farin G.(ed) : Geometric Modeling, Algorithms and New Trends, SIAM, pp. 385～395.

Petrie, G. and Kennie, T. J. M., 1987, "Terrain Modelling in Surveying and Civil Engineering," *CAD*, pp. 171～187.

Pfeiler, H. U., 1985, "Methods Used for Intersecting Geometrical Entities in the GPM Module for Volume Geometry," *CAD*, pp. 311～318.

Phillips, M. B. and Odell, G. M., 1984, "An Algorithm for Locating and Displaying the Intersection of Two Arbitrary Surfaces," *IEEE CG&A*, pp. 49～58.

Pilz, M., Kamel, H. A., 1989, "Creation and Boundary Evaluation of CSG-Models," *Engineering with Computers* 5, pp. 105～116.

Prakash, P. V., 1988, "Surface/Surface Intersections for Geometric Modeling," Ph. D. in MIT.

Pratt, M. J. and Geisow, A. D., 1986, "Surface/Surface Intersection Problems," *Mathematics of surfaces*, pp. 117~142.

Rossignac, J. R. and Requicha, A. A. G., 1987, "Piecewise Circular Curves for Geometric Modeling," *IBM Jounal Res. Devel.*, Vol. 31, pp. 296~312.

Sabin, M. A., 1976, "A Method for Displaying the Intersection Curve of Two Quadric Surfaces," *The Computer Journal*, Vol. 19, pp. 336~338.

Sabin, M. A. and Doo, D., 1978, "Behaviour of Recursive Division Surfaces Near Extraordinary Points," *CAD*, pp. 356~360.

Sarraga, R. F., 1983, "Algebraic Methods for Intersection of Quadric Surfaces in GMSOLID," *Computer Vision, Graphics and Image Processing*, Vol. 22, pp. 222~238.

Satterfield, S. G. and Rogers, D. F., 1985, "A Procedure for Generating Contour Lines from a B-Spline Surface," *IEEE CG&A*, pp. 71~75.

Sederberg, T. W. and Anderson, D. C., 1984a, "Implicit Representation of Parametric Curves and Surfaces," *Computer Version, Graphics and Image Processing*, Vol. 28, pp. 72~84.

Sederberg, T. W. et al., 1984b, "Vector Elimination : A Technique for the Implicitization, Inversion and Intersection of Parametric Rational Polynomial Curves," *CAGD*, pp. 327~355.

Sederberg, T. W., 1984c, "Planar Piecewise Algebraic Curves," *CAGD*, pp. 241~255.

Sederberg, T. W. and Anderson, D. C., 1985a, "Steiner Surface Patches," *IEEE CG&A*, pp. 23~36.

Sederberg, T. W. et al., 1985b, "Implcitization, Inversion and Introduction of Rational Cubic Curves," *Computer Version, Graphics and Image Processing*, pp. 89~102.

Sederberg, T. W. and Goldmann, R. N., 1986a, "Algebraic Geometry for Computer-Aided Geometric Design," *IEEE CG&A*, pp. 52~59.

Sederberg, T. W. and Parry, S. R., 1986b, "Comparison of Three Curve Intersection Algorithms," *CAD*, pp. 58~63.

Sederberg, T. W. and Sniverly, J. P., 1987, "Parameterization of Algebraic Surfaces," *Mathematics of Surfaces*, pp. 299~319.

Sederberg, T. W., White, S. C. and Zundel, A. K., 1989, "Fat Arcs : A Bounding Region with Cubic Convergence," *CAGD*, pp. 205~218.

Shantz, M. and Chang, S. L., 1988, "Rendering Trimmed NURBS with Adaptive Forward Differencing," *Computer Graphics*, pp. 189~197.