Detecting and Calculating Conic Sections in The Intersection of Two Natural Quadric Surfaces
Part II: Geometric Constructions for Detection and Calculation

James R. Miller, University of Kansas
Ronald N. Goldman, Rice University

Abstract

One of the most challenging aspects of the surface-surface intersection problem is the proper disposition of degenerate configurations. Even in the domain of quadric surfaces, this problem has proven to be quite difficult. In a companion paper [2] we presented a case by case algebraic analysis of intersections between pairs of natural quadric surfaces; the result was a complete characterization of all configurations in which the intersection is planar. Here we present details of the geometric algorithms which detect the degeneracies and compute the resulting planar intersections. We also discuss the other degeneracy which can arise in the intersection of two natural quadric surfaces, namely the decomposition of the intersection into a line and a space cubic. Readers whose interest in the topic of planar intersections is purely theoretical will likely only be interested in the companion paper [2]. Readers who instead are looking only for robust and efficient algorithms, but who do not wish to examine the derivations and proofs of correctness and completeness need only study this paper.

1.0 Introduction

Robust detection and processing of degeneracies in surface-surface intersection algorithms continues to be a challenging problem. In a companion paper [2] we used algebraic geometry to enumerate all conditions under which two natural quadric surfaces[1] intersect in planar curves. By Bezout's Theorem, two quadric surfaces always intersect in a degree four curve in complex projective space. Degenerate intersections are those in which the degree four curve actually splits into two or more lower degree components. We are concerned here primarily with those configurations which lead to a pair of degree two curves (i.e., a pair of conic sections).

The analysis in [2] was long and tedious, but the final results (summarized in Tables III and IV below) were quite simple, albeit not entirely intuitive. In this paper we show how to use the results presented in [2] to construct robust geometric procedures for detecting and calculating planar curves in the intersection of two natural quadric surfaces.

In [2] we discussed our primary motivations for this work, and we summarized the work of others in this area. We will not repeat that material here; rather we proceed immediately to the details of calculating these planar intersections. In Section 2 we review the basic geometric descriptions of conic curves and natural quadric surfaces. We also introduce the notation and fundamental geometric tools which we shall use throughout the paper. In Section 3 we begin the calculation of degenerate intersections by showing how to find isolated tangent points. Section 4 is the bulk of the paper. Here we present all the algorithms for calculating conics in the intersection of pairs of natural quadric surfaces. In some cases we can write explicit expressions for the parameters defining the conic sections. However, often the simplest most direct method for computing planar intersections involving pairs of natural quadric surfaces is to find the plane(s) containing the intersection, and then to intersect one of the two original surfaces with the plane(s). We adopt this approach for several of the cases in Section 4, assuming that the reader has access to plane-quadric intersection algorithms. Algorithms for plane-natural quadric intersections based on geometric constructions are described in [7]. In Section 5

[1] The natural quadrics [2] are the sphere, the right circular cylinder, and the right circular cone. In this paper we will use the terms "cylinder" for "right circular cylinder" and "cone" for "right circular cone".

we discuss another type of degeneracy which can arise in the intersection of natural quadric surfaces. When at least one of the two natural quadrics is a cone, the intersection may degenerate into a degree one curve (i.e., a straight line) and an irreducible degree three curve (i.e., a space cubic). We investigate the relative configurations which give rise to this degeneracy in Section 5. Finally, in Section 6 we make some concluding observations.

2.0 Geometric Notation and Tools

As a prelude to discussing the geometric approach to the calculation of planar intersections, we need to review our notation and representation of the data. Geometric representations of conics and quadrics are generally characterized by a local coordinate system with associated scalar parameters. The local coordinate system is defined by (i) three mutually perpendicular unit vectors (**u**,**v**,**w**) which describe the orientation of the conic or quadric and (ii) a base point *O* which fixes the position of the curve or surface in space[2]. The scalar parameters determine the size of the conic or quadric.

We shall adopt certain conventions on the use of the vectors (**u**,**v**,**w**). The vectors (**u**,**v**), for example, are used to specify two-dimensional orientations in a plane (e.g., the major and minor axis vectors for an ellipse). The vector **w** is used for line directions, plane normals, and cylinder and cone axis vectors.

Obviously there is redundant information in the complete coordinate system, and only portions of it need be specified to determine uniquely the position and orientation of a particular conic or quadric. To define a circle, for example, we need to specify only the normal vector **w**; the **u** and **v** directions are arbitrary and are needed only to support parameterizations. On the other hand, an ellipse has a unique orientation in the plane which must be captured in its geometric definition. We therefore require its **u** and **v** directions, but omit the **w** vector since it can be computed as **w** = **u** x **v**.

The geometric parameters which uniquely define lines and conics are summarized in Table I and illustrated in Figure 1. Those defining planes and natural quadrics are listed in Table II and illustrated in Figure 2. Throughout this paper, we shall assume that the vectors associated with the geometric representations are unit vectors.

Using vector techniques it is easy to derive both parametric and implicit representations of these second degree curves and surfaces from their geometric representations. One advantage of the geometric representation is our ability to derive simply and robustly these two other common representations. Moreover, all the data in the geometric representation has a clear cut physical meaning. This is not true for, say, the coefficients of the implicit polynomial representation.

Table I
Geometric Descriptions of Lines and Conics

Curve	Description of Geometric Parameters	Notation
Line	(base point , direction vector)	$(B,\mathbf{w})$
Circle	(center , normal to the plane containing the circle , radius)	$(C,\mathbf{w},r)$
Ellipse	(center , major axis , minor axis , major radius , minor radius)	$(C,\mathbf{u},\mathbf{v},r_u,r_v)$
Parabola	(vertex , directrix vector , focus vector , focal length)	$(V,\mathbf{u},\mathbf{v},f)$
Hyperbola	(center , major axis , minor axis , major radius , minor radius)	$(C,\mathbf{u},\mathbf{v},r_u,r_v)$

[2] Points will be written using upper case italic letters (e.g., *C*, *O*, *Q*) or, in pseudo-code, with italicized names whose first letter is capitalized. Vectors will be written using lower case bold letters (e.g., **u**, **v**, **w**) or, in pseudo-code, with bold-faced lower case names. Scalars will be written using lower case italic (e.g., *r*, *d*). Function names will be written using lower case plain text.

Table II
Geometric Descriptions of Planes and Natural Quadric Surfaces

Surface	Description of Geometric Parameters	Notation
Plane	(base point , normal vector)	$(B,\mathbf{w})$
Sphere	(center , radius)	(C,r)
Cylinder	(base point , axis vector , radius)	$(B,\mathbf{w},r)$
Cone	(vertex , axis vector , half-angle)	$(V,\mathbf{w},\alpha)$

Certain primitive functions creating and manipulating scalars, points, vectors, curves, and surfaces are assumed in the sequel. Some of the main ones are briefly summarized below.

The function `line`(Q,**v**) returns a line whose base point and unit direction vector are as specified. Similarly, the function `plane`(Q,**n**) returns a plane whose base point and unit normal vector are given by the parameters. The function `normalize`(**v**) returns a unit vector whose direction is the same as that of **v**.

The function `distance` returns the distance between its two parameters which may be any combination of points, lines, and planes. The function `signed_distance_along_line`(Q,L) assumes that the point Q is on the line L and calculates the signed distance from the base point of L to Q as:

$$((Q\text{-L.}B)\cdot\text{L.}\mathbf{w}).$$

Based on these functions, we assume that we can test reliably (i.e., to within some prespecified tolerance) whether a point is on a plane or line, or whether two points are identical.

Various tests involving vectors will be applied. Most notably we need to test for parallelism and perpendicularity of vectors. We shall only require such tests on the vectors within the geometric representations, that is, on unit vectors. These tests can be performed reliably since two unit vectors are parallel if and only if their dot product is ±1, and perpendicular if and only if their dot product is zero (again to within some prespecified tolerance).

When intersecting cylinders and cones, we shall sometimes need to intersect their axis lines. General algorithms for intersecting two 3D lines can be found in a variety of sources (e.g., [1, 6]). It is also necessary to test whether the axis lines are skew. The method described in [1] actually generates the parameter values on each of the two lines for the points of closest approach. If these two points are identical, then they describe the point at which the lines intersect; if they are different, the lines are skew.

Finally we shall need tests to determine if a point lies on a cylinder or a cone. We can use the geometric form of the implicit surface equations. A point Q lies on a cylinder C if and only if [5]:

$$(Q-\text{C.}B)\cdot(Q-\text{C.}B)-((Q-\text{C.}B)\cdot\text{C.}\mathbf{w})^2-\text{C.}r^2=0.$$

A point Q lies on a cone C if and only if [5]:

$$((Q-\text{C.}V)\cdot\text{C.}\mathbf{w})^2-\cos^2\alpha(Q-\text{C.}V)\cdot(Q-\text{C.}V)=0.$$

3.0 Isolated Tangent Points

We showed in [2] that two point tangencies could arise between pairs of natural quadrics only under the situations enumerated in Table III. This is the same as Table III in [2]. The section, case, and figure numbers refer to those in [2]. We describe below a general scheme for detecting the presence and computing the position of these isolated points of tangency. This procedure is best incorporated into the algorithm for computing nonplanar intersections between two natural quadrics as described in [5]. This approach detects single points of tangency as well, even when they occur in conjunction with other nonplanar curves of intersection.

Before describing the method, a couple of observations are in order. As we stated in [2], we currently do not know how to characterize all configurations leading to single points of tangency between pairs of natural quadric surfaces. Yet these configurations are just as important as two point tangencies. Whether it is critical to detect such configurations is not clear; the answer is almost certainly application-dependent. In our opinion it is not worth the extra effort to test explicitly the conditions listed in Table III, and, if they are satisfied, to implement individual geometric constructions to compute the pairs of tangent points. This philosophy is based in large part on the fact that we have a single common algorithm, described below, which finds all two point tangencies as well as all single point tangencies..

Table III
Summary of Conditions Giving Rise to Two Point Tangencies Between Pairs of Natural Quadric Surfaces

Section	Surface Pair	Case	Geometric Conditions	Figure
4.3	sphere/cone	2	Center of sphere in plane $(V,\mathbf{w})$ at distance $d=r/\cos\alpha$ from vertex	1
4.5	cylinder/cone	3	Skew axes; distance between axes: $d = \frac{r\sin\theta}{\sqrt{\sin^2\theta - \sin^2\alpha}}$	3
4.6	cone/cone	2b(i)	Perpendicular axes; V_2 in plane of $(V_1,\mathbf{w}_1)$; distance, μ, from V_1 to axis line $(V_2,\mathbf{w}_2)$ and distance, υ, from V_1 to plane $(V_2,\mathbf{w}_2)$ are related as: $\upsilon = \sqrt{1-\tan^2\alpha_1\tan^2\alpha_2}\cos\alpha_1\cot\alpha_2\mu$	7
		2b(ii)	Same as 2b(i) with roles of the cones reversed	7
		2d(ii)	Skew axes with constraints (7) and (9) (See [2].)	8

Consider two quadrics Q_1 and Q_2 and suppose that Q_1 is a ruled quadric to be used as a parameterization surface for a nonplanar intersection curve as described in [5]. The (s,t) parameters on Q_1 have the following geometric interpretation. The parameter t selects a ruling ($-\pi \le t \le +\pi$), and s is a signed distance along the ruling. Recall that nonplanar intersections between two such quadrics can be described using an implicit equation in the parameter space of Q_1 as:

$$a(t)s^2 + b(t)s + c(t) = 0. \qquad (1)$$

The functions a, b, and c are rational quadratic polynomials which depend upon the types of quadrics involved. Ordered sets of points can be generated along the intersection curve by selecting successive rulings on Q_1 (i.e., by selecting successive values of t in the range $-\pi$ to $+\pi$) and then solving the resulting quadratic equation (1) in s. In general, only the subsets of the entire $-\pi$ to $+\pi$ range where the discriminant of (1) is nonnegative correspond to real portions of the intersection curve. The ranges of parameter space which satisfy this inequality are delimited by the zeros of this discriminant, i.e., by those values of t satisfying:

$$b(t)^2 - 4a(t)c(t) = 0. \tag{2}$$

Equation (2) can be expressed equivalently as a rational quartic polynomial equation [4, 5]; thus there are up to four real roots which correspond to the t values of rulings on Q_1 which are tangent to Q_2. All other values of t correspond to rulings on Q_1 which either have no real intersection with Q_2 (if the left hand side of (2) is negative) or which intersect Q_2 in two distinct real points (if the left hand side of (2) is positive). Points of tangency between Q_1 and Q_2 determine unique rulings on Q_1 which are tangent to Q_2, and each ruling on Q_1 which is tangent to Q_2 corresponds to a real root of (2). Thus if Q_1 and Q_2 are tangent at one or two points, those points can be computed by solving (1) using the appropriate roots of (2). The appropriate roots of (2) are those where the t values in intervals on either side yield a negative discriminant. In [5] we describe robust geometric methods to find all such roots without explicitly solving any equation of degree greater than two. We will therefore not consider isolated tangent points further in this paper.

4.0 Pairwise Analysis of Surface Intersections

We are now ready to examine in turn each of the six possible combinations of natural quadric surfaces. We first review the geometric tests for planar intersections. We then show how the planar curves can actually be computed.

Table IV
Summary of Conditions Giving Rise to Planar Intersection Curves Between Pairs of Natural Quadric Surfaces

Section	Surface Pair	Case	Geometric Conditions	Results	Figure
4.1	sphere/sphere		All	empty; one tangent point; or one circle	
4.2	sphere/cylinder		Center of sphere on axis of cylinder	empty; one tangent circle; or two circles	
4.3	sphere/cone	1	Center of sphere on axis of cone	empty; one tangent circle; one circle + vertex; or two circles	
4.4	cylinder/cylinder	2	Parallel axes	empty; one tangent line; or two lines	
		3	Intersecting axes & equal radii	two ellipses	
4.5	cylinder/cone	2a	Coincident axes	two circles	
		2b	Axes intersect in a point at distance $d=r/\sin\alpha$ from the vertex of the cone	two ellipses (same or opposite halves of the cone); or one ellipse & tangent line	2
4.6	cone/cone	1a(i)	Parallel axes, same half angle	ellipse; shared tangential ruling; or hyperbola	4
		1a(ii)	Coincident axes	two circles or single vertex	
		1b	Axes intersect at point I such that $d_1\sin\alpha_1=d_2\sin\alpha_2$ where d_i is the distance from vertex i to I. (This includes the case where the vertices coincide; i.e., $d_1=d_2=0$.)	various combinations of pairs of conics *or* a tangent line plus a conic (1-4 lines if the vertices coincide)	6

Table IV summarizes the configurations of natural quadrics which yield planar intersection curves. This table is the same as Table II in [2]. The section, case, and figure numbers refer to those in [2]. The remainder of this section is devoted to a discussion of how the planar curves in the intersection of two natural quadrics can be computed.

We shall proceed as follows. For each of the configurations summarized in Table IV, we shall show how to compute the geometric parameters defining the planar intersection curves presented in Table I from those of the given natural quadrics indicated in Table II. We shall work with the natural quadrics in general position and orientation, and we will not employ coordinate system transformations of any sort. The analysis for the sphere-sphere, sphere-cylinder, sphere-cone, and most of the cylinder-cylinder cases is fairly straightforward. The implementation of cylinder-cone, cone-cone, and some cylinder-cylinder intersections, however, is more complex. We shall find it convenient in these cases to extend the algebraic analysis of [2] to determine the plane or planes containing the conics. That is, we shall reduce the problem to a pair of plane-cone or plane-cylinder intersections. This will not compromise our overall geometric approach since this algebraic analysis is again purely a "paper analysis" used only to derive invariant geometric representations of the planes containing the conics. The algorithms we present shall compute directly the geometric parameters defining the planes. Detailed procedures for intersecting planes and natural quadrics based on geometric constructions are described in [7].

There are alternative geometric approaches which do not require this recourse to algebra. However we prefer this approach because of its elegance and simplicity: we need not try to distinguish geometrically what types of conics will arise from a given configuration of natural quadrics and design algorithms accordingly. Instead we simply construct geometrically the planes containing the conics and push these sorts of considerations down to the plane-quadric intersection algorithms where they can be handled easily, more reliably, and with fewer special case considerations.

Our general approach will be to write the algebraic equation of the planes containing the conics as determined by the appropriate pencil matrix $Q(\lambda)$ and the corresponding constraints derived in [2]. We then manipulate the equation into a form which can be easily factored into two linear terms. These terms are the plane equations from which the normal vectors can immediately be found. We then find a point common to the two planes, either from some a priori knowledge (as in the cylinder-cylinder case) or by solving the plane equations simultaneously (as in the cylinder-cone and cone-cone cases).

The equations from [2] which we manipulate were derived assuming the quadrics were in canonical and relative canonical position. The resulting expressions for the normal vectors and common point are therefore initially stated algebraically in terms of this canonical coordinate system. To eliminate this coordinate dependence, we rewrite these algebraic expressions for the normals and points in vector form using the base points and axis vectors of the quadrics. Vector equations are invariant under coordinate system transformations, and by construction we know these vector expressions are satisfied when the quadrics are in relative canonical position. Therefore we know they are valid for natural quadrics in general position.

Table V
Review of Algebraic Notation for Natural Quadrics in Canonical and Relative Canonical Position

Quadric	Canonical Position	Relative Canonical Position
Sphere	$C=(0,0,0)$	$C=(\mu,\upsilon,\omega)$
Cylinder	$B=(0,0,0)$, $\mathbf{w}=(0,0,1)$	$B=(\mu,\upsilon,\omega)$, $\mathbf{w}=(0,s,c)$
Cone	$V=(0,0,0)$, $\mathbf{w}=(0,0,1)$	$V=(\mu,\upsilon,\omega)$, $\mathbf{w}=(0,s,c)$

In Table V we review the notation used in the algebraic analysis. We denote the angle between the axis vectors in cylinder-cone, cylinder-cylinder, and cone-cone intersections as θ. We then use $s=\sin\theta$ and $c=\cos\theta$. (Note $s^2+c^2=1$.) In intersections involving cones, we write $F=\sec\alpha$ and $E=\tan\alpha$ where α is the half-angle of a cone. (Note $F^2=E^2+1$.)

Finally we observe that we never need to solve any nonlinear equations in the implementation of these algorithms. This demonstrates that finding planar intersections between pairs of natural quadric surfaces is inherently a linear problem.

4.1 Sphere-Sphere

We showed in [2] that the intersection of two spheres is always planar. The nature of the intersection is determined entirely by the distance between the centers of the spheres. Without loss of generality we assume that S1.$r \geq$S2.r. For convenience we also assume that the spheres are not identical.

We shall use the Law of Cosines to compute cosß (see Figure 3), but a remark on numerical reliability is in order. Referring to Figure 3, observe that

$$\cos\beta = \frac{\mathrm{S1}.r^2 + d^2 - \mathrm{S2}.r^2}{2 * \mathrm{S1}.r * d}.$$

To compute the center of the circle, we need to multiply cosß by S1.r to get the distance, f, from S1.C to the center of the circle. If we compute f in this fashion, we multiply and divide by S1.r. Assuming that we would first compute cosß and then compute f, we are likely to get a less accurate result than if we simply calculated f directly. The use of the Law of Cosines in this fashion is quite common. (In fact, we shall see an analogous situation in Section 4.4.1.) We therefore advocate the use of the following auxiliary procedure.

```
procedure LawOfCosines(input d1,d2,d3: real; output cosß, h: real);
begin
   h    := (d1*d1 + d2*d2 - d3*d3) / (2*d2);
   cosß := h / d1
end;
```

The following algorithm can be used to intersect the two spheres. See Figure 3.

```
input: S1, S2: sphere
d := distance(S1.C,S2.C)
c1_to_c2 := normalize(S2.C - S1.C)
if d > (S1.r + S2.r) then
   output: no intersection
else if d < (S1.r - S2.r) then
   output: no intersection
else if (d = (S1.r + S2.r)) or (d = (S1.r - S2.r)) then
   output: tangent point:  S1.C + S1.r*c1_to_c2
else { intersection is a circle }
   LawOfCosines(S1.r,d,S2.r,cosß,f) { See Figure 3 }
   sinß := sqrt(1-cosß*cosß)
   output: circle:  C := S1.C + f*c1_to_c2
                    w := c1_to_c2
                    r := S1.r*sinß
```

To compute the circle of intersection, we could have computed its plane (i.e., just C and w) and then performed a plane-sphere intersection. Clearly the method shown is preferable.

4.2 Sphere-Cylinder

In [2] we showed that the intersection of a sphere and a cylinder is planar only if the center of the sphere lies on the axis line of the cylinder. Once we know this to be the case, the nature of the intersection is determined entirely by the radii of the sphere and cylinder. The following algorithm suffices.

```
input: S: sphere; C: cylinder
if S.r < C.r then
   output: no intersection
else if S.r = C.r then
   output:   tangent circle: C := S.C
                             w := C.w
                             r := S.r
else { intersection is a pair of circles; see Figure 4 }
   offset_distance := √(S.r² - C.r²)
   output:   circle 1: C := S.C + offset_distance*C.w
                       w := C.w
                       r := C.r
             circle 2: C := S.C - offset_distance*C.w
                       w := C.w
                       r := C.r
```

As with sphere-sphere intersections, the planes of the intersection circles could have been computed and used to intersect either the sphere or the cylinder. Again, the approach shown is preferable.

4.3 Sphere-Cone

In [2] we showed that the intersection of a sphere and a cone is a planar curve only if the center of the sphere lies on the axis line of the cone. When this is the case, the intersection will be one of the following: empty, one (tangent) circle (Figure 5), a circle plus an isolated point (the vertex; Figure 6), or two circles (Figure 7). If it is two circles, they may be on the same or opposite halves of the cone. The progression through these cases can be visualized by imagining a sphere of some fixed radius moving along the cone axis. When it is infinitely far from the vertex, there is no intersection. As the sphere moves towards the vertex, it first intersects the cone when its center is at a distance *tangent_distance* from the vertex. (See below and Figure 5.) The intersection is then a tangent circle. As it moves closer to the vertex, it transitions to a double circle intersection (both circles on the same half of the cone), then to an intersection consisting of the vertex plus a single circle. As the center moves further towards the vertex, the intersection becomes two circles, one on either cone half. This sequence then repeats in the reverse order as the center moves towards infinity on the other half of the cone axis.

The following algorithm can be used to determine the specific intersection.

```
input: S: sphere; C: cone
tangent_distance := S.r / sin(C.α) { See Figure 5 }
d := signed_distance_along_line(S.C, line(C.V,C.w) )
if abs(d) > tangent_distance then
   output: no intersection
else if abs(d) = tangent_distance then
   { intersection is a tangent circle; see Figure 5 }
   h := √(tangent_distance² - S.r²)
   output:   tangent circle: C := C.V + sign(d)*h*cos(C.α)*C.w
```

```
                                         w := C.w
                                         r := h*sin(C.α)
      else if abs(d) = S.r then
         { intersection is vertex + circle; see Figure 6 }
         h := 2*S.r*cos(C.α)
         output:     circle: C := C.V + sign(d)*h*cos(C.α)*C.w
                             w := C.w
                             r := h*sin(C.α)
                     point:  C.V

      else
         {
         Intersection is 2 circles. They may be on the same or opposite
         cone halves. The analysis is identical for either case. See Figure
         7.
         }
         h1 := d*cos(C.α) { Note that h1 inherits the sign of d }
         h2 := √(S.r² - (d*sin(C.α))²)
         output:     circle 1: C := C.V + (h1 + h2)*cos(C.α)*C.w
                               w := C.w
                               r := abs(h1 + h2)*sin(C.α)
                     circle 2: C := C.V + (h1 - h2)*cos(C.α)*C.w
                               w := C.w
                               r := abs(h1 - h2)*sin(C.α)
```

Once again, the planes containing the circles could have been computed and the result determined by applying plane-sphere or plane-cone intersections. However, the method shown is again preferable.

4.4 Cylinder-Cylinder Intersections

We showed in [2] that the intersection of two cylinders is planar if and only if (i) the axes are parallel, or (ii) the axes intersect and the cylinders have the same radius.

4.4.1 Parallel Axes

We begin by considering the case of parallel axes. The intersection will either be empty or will consist of one tangent or two parallel lines. The distance between the axes (equivalently, the distance between the base point of one cylinder and the axis of the other) is the determining factor. Without loss of generality we assume that C1.$r \geq$C2.r. For convenience we also assume that the cylinders are not identical.

```
      input: C1, C2: cylinder
      d := distance(C1.B,line(C2.B,C2.w))
      if d > (C1.r + C2.r) then
         output: no intersection
      else if d < (C1.r - C2.r) then
         output: no intersection
      else { Intersection is one or two lines. See Figure 8. }
         {
         Compute a unit vector pointing from axis 1 towards axis 2 (Figure 8a).
         }
         b1_to_b2 := C2.B - C1.B
         b1_to_axis2 := normalize(b1_to_b2 - (b1_to_b2·C1.w)*C1.w)
         if (d = (C1.r + C2.r)) or (d = (C1.r - C2.r)) then
            { intersection is a single tangent line; see Figure 8b }
```

```
      output: line:   B := C1.B + C1.r*b1_to_axis2
                      w := C1.w
   else { intersection is two lines; see Figure 8c }
      LawOfCosines(C1.r,C2.r,d,cosß,f)
      Q  := C1.B + f*b1_to_axis2
      sinß := sqrt(1-cosß*cosß)
      h := C1.r*sin ß
      u  := C1.w x b1_to_axis2
      output:    line 1:   B := Q + h*u
                           w := C1.w
                 line 2:   B := Q - h*u
                           w := C1.w
```

4.4.2 Intersecting Axes

We next consider the case of two cylinders with equal radii whose axes intersect. The intersection curve is a pair of ellipses which intersect each other at the two points at the extremes of their minor axes. We shall proceed by computing the planes which contain the two ellipses. Once we have them, we could intersect each with one of the cylinders to obtain the result. As we shall point out, however, significant computation can be saved by explicitly computing the ellipse parameters since many of their defining parameters are shared and need only be computed once. This direct computation is quite straightforward.

The intersection of a plane and a cylinder is an ellipse whose center lies on the cylinder axis [7]. Since each ellipse lies on both cylinders, it follows that the point I of intersection between the two axes is the common center of the two ellipses. Therefore it is also a point common to the two planes containing the two ellipses. To complete the specification of the two planes, we need only find their normal vectors. It can be shown in a variety of ways that the two planes are perpendicular to each other. We choose to demonstrate this fact by extending the analysis of cylinder-cylinder intersections from Section 4.4 of [2] since this technique will be needed for the cylinder-cone and cone-cone cases where the results are not so obvious from purely geometric considerations.

From [2], the matrix representing the pair of planes for cylinder-cylinder intersections in the case of intersecting axes is:

$$Q(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -s^2 & -sc & sc\omega \\ 0 & -sc & s^2 & -\omega s^2 \\ 0 & sc\omega & -\omega s^2 & s^2\omega^2 \end{pmatrix}.$$

From this matrix, we can write down the polynomial representing the two planes:

$$-s^2y^2 - 2scyz + s^2z^2 + 2sc\omega y - 2\omega s^2 z + s^2\omega^2 = 0.$$

Multiplying both sides by -1, and replacing ($-s^2$) with (c^2-1) in the coefficients for z^2 and ω^2, we see that this quadratic polynomial can be factored into linear factors as:

$$(sy + (c+1)z - \omega(c+1))(sy + (c-1)z - \omega(c-1)) = 0. \tag{3}$$

These linear factors represent the planes of the two ellipses. From the implicit equation of a plane:

$$a_1 x + a_2 y + a_3 z + a_4 = 0$$

we can form a vector parallel to the unit normal vector as:

$$\mathbf{n} = (a_1, a_2, a_3).$$

Therefore the normals to the two planes in (3) are

$$\mathbf{n}_1 = (0,s,c+1) = (0,s,c) + (0,0,1) = \text{C1.w} + \text{C2.w}$$

and

$$\mathbf{n}_2 = (0,s,c-1) = (0,s,c) - (0,0,1) = \text{C1.w} - \text{C2.w}.$$

Since for any two unit vectors $\mathbf{u}_1$ and $\mathbf{u}_2$:

$$(\mathbf{u}_1 + \mathbf{u}_2)\cdot(\mathbf{u}_1 - \mathbf{u}_2) = \mathbf{u}_1\cdot\mathbf{u}_1 - \mathbf{u}_1\cdot\mathbf{u}_2 + \mathbf{u}_2\cdot\mathbf{u}_1 - \mathbf{u}_2\cdot\mathbf{u}_2 = 1 - 1 = 0,$$

$(\mathbf{u}_1+\mathbf{u}_2)\perp(\mathbf{u}_1-\mathbf{u}_2)$. Therefore it follows that the planes of the two ellipses are perpendicular to each other.

Clearly the ellipses could be computed by intersecting one of the cylinders with each of the planes: `plane`(I , `normalize`(C1.w+C2.w)) and `plane`(I , `normalize`(C1.w-C2.w)). We present a more efficient algorithm here which exploits the observations made above and which uses the following additional facts.

- The minor radius of each ellipse is the common cylinder radius.
- The major radius of each ellipse is the common cylinder radius divided by cosθ, where θ is the angle between the plane normal and the cylinder axis vector.
- The common minor axis is `normalize`(C1.w x C2.w)≡`normalize`($\mathbf{n}_1$) x `normalize`($\mathbf{n}_2$).
- The normal to the plane of one ellipse is the major axis vector of the other ellipse.

The first two statements are proved in [7]; the third can be derived easily from the presentation given there. The final result is true because $\mathbf{n}_2$ is perpendicular to $\mathbf{n}_1$ and to the minor axis; similarly $\mathbf{n}_1$ is perpendicular to $\mathbf{n}_2$ and the minor axis.

```
input: C1, C2: cylinder
{ Compute the point of intersection of the cylinder axes }
I := intersect(line(C1.B,C1.w),line(C2.B,C2.w))
unit_n1  := normalize(C1.w + C2.w)
unit_n2  := normalize(C1.w - C2.w)
common_minor_axis  := unit_n1 x unit_n2
output:   ellipse 1:  C := I
                      u  := unit_n2
                      v  := common_minor_axis
                      ru  := C1.r / (unit_n1·C1.w)
                      rv  := C1.r
          ellipse 2:  C := I
                      u  := unit_n1
                      v  := common_minor_axis
                      ru  := C1.r / (unit_n2·C1.w)
                      rv  := C1.r
```

Figure 9 illustrates the results of applying this algorithm.

4.5 Cylinder-Cone Intersections

In [2] we showed that the intersection of a cylinder and a cone is a planar curve if and only if either (i) the axes coincide, or (ii) the axes intersect at a distance $\omega = r/\sin\alpha$ from the cone vertex.

4.5.1 Coincident Axes

We consider first the case of identical axes. The intersection is two circles lying on opposite cone halves at a distance $d = r/\tan\alpha$ from the vertex (see Figure 10):

```
input: cyl: cylinder; con: cone
d := cyl.r/tan(con.α)
output:     circle 1: C  := con.V + d*con.w
                      w  := con.w
                      r  := cyl.r
            circle 2: C  := con.V - d*con.w
                      w  := con.w
                      r  := cyl.r
```

4.5.2 Intersecting Axes

If the axes intersect at a point I whose distance from the vertex is $r/\sin\alpha$, then the intersection is either one ellipse plus a tangent line or a pair of ellipses. The former results if the acute angle between the axis lines is the same as the cone half angle (i.e., if $\theta = \alpha$). Our approach will be to find the pair of planes containing the intersection using the algebraic method illustrated in Section 4.4.2. As we shall see, when the intersection is a tangent line plus an ellipse, one of the planes we compute is tangent to the cylinder and the cone along the shared ruling. While in principle, therefore, we need not distinguish between the line-plus-ellipse and two-ellipse cases, it is desirable to do so since detecting tangentially shared rulings is quite delicate numerically. The test in terms of the parameters of the given cylinder and cone is straightforward as we shall see. If instead we were to compute a plane, and then rely on the plane-cylinder algorithm to detect a tangent line of intersection, there would be a much greater probability that, due to small numerical errors, the plane might be judged to intersect the cylinder in two parallel lines, or a long skinny ellipse, or not at all. We therefore advocate detecting this special case and then computing directly the defining parameters of the tangent line of intersection.

In Section 4.5 of [2], we derived the following matrix representation for the pair of planes containing the conics:

$$Q(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -s^2 & -sc & sc\omega \\ 0 & -sc & s^2+E^2 & -\omega s^2 \\ 0 & sc\omega & -\omega s^2 & s^2\omega^2 - r^2 \end{pmatrix}.$$

Multiplying this matrix by -1 and writing the equation in implicit form, we obtain:

$$s^2y^2 + 2scyz - \left(s^2+E^2\right)z^2 - 2sc\omega y + 2\omega s^2 z + \left(r^2 - s^2\omega^2\right) = 0. \qquad (4)$$

We then observe:

$$-(s^2 + E^2) = -((1 - c^2) + (F^2 - 1)) = c^2 - F^2. \tag{5}$$

Using the constraint $r = \omega\sin\alpha$, we can rewrite the constant term as:

$$\omega^2 \sin^2 \alpha - s^2\omega^2 = \omega^2(\sin^2 \alpha - \sin^2 \theta) = \omega^2((1 - \cos^2 \alpha) - (1 - \cos^2 \theta)) = \omega^2(c^2 - 1/F^2). \tag{6}$$

Substituting (5) and (6) into (4), we see that the pair of planes can be expressed as:

$$s^2y^2 + 2scyz + (c^2 - F^2)z^2 - 2sc\omega y + 2\omega s^2 z + \omega^2(c^2 - 1/F^2) = 0.$$

This can be factored as:

$$(sy + (c + F)z - \omega(c + 1/F))(sy + (c - F)z - \omega(c - 1/F)) = 0. \tag{7}$$

Again, we can read off the normals to these two planes as:

$$\mathbf{n}_1 = (0, s, c + F) = (0, s, c) + F(0,0,1) = \text{cyl.}\mathbf{w} + F\text{con.}\mathbf{w}$$

and

$$\mathbf{n}_2 = (0, s, c - F) = (0, s, c) - F(0,0,1) = \text{cyl.}\mathbf{w} - F\text{con.}\mathbf{w}.$$

To complete the specification of the planes containing the conic curves, we must find a point on each plane. We choose to find a single point common to both planes by solving simultaneously the two plane equations in (7):

$$sy + (c + F)z - \omega(c + 1/F) = 0$$
$$sy + (c - F)z - \omega(c - 1/F) = 0.$$

Replacing the first equation by half the sum of the two equations and replacing the second by half their difference, we obtain:

$$sy + cz - c\omega = 0$$
$$Fz - \omega/F = 0.$$

Clearly the solution is then:

$$x = \text{arbitrary}$$
$$y = \frac{c\omega}{s}\left(1 - \frac{1}{F^2}\right) = \frac{c\omega}{s}\sin^2 \alpha$$
$$z = \frac{\omega}{F^2} = \omega\cos^2 \alpha.$$

Choosing x to be 0, we derive a geometrically invariant characterization of the common point Q as:

$$Q = V + \frac{c\omega}{s^2}\sin^2\alpha(0,s,c) + \left(\omega\cos^2\alpha - \frac{c^2\omega\sin^2\alpha}{s^2}\right)(0,0,1)$$
$$= V + \frac{c\omega\sin^2\alpha}{s^2}\text{cyl.w} + \left(\omega\cos^2\alpha - \frac{c^2\omega\sin^2\alpha}{s^2}\right)\text{con.w}$$
$$= V + \frac{c\omega\sin^2\alpha}{s^2}\text{cyl.w} + \frac{\omega}{s^2}\left(s^2\cos^2\alpha - c^2\sin^2\alpha\right)\text{con.w}.$$

Using $\cos^2\alpha = 1 - \sin^2\alpha$, this simplifies to:

$$Q = V + \frac{c\omega\sin^2\alpha}{s^2}\text{cyl.w} + \frac{\omega}{s^2}\left(s^2 - s^2\sin^2\alpha - c^2\sin^2\alpha\right)\text{con.w}$$
$$= V + \frac{c\omega\sin^2\alpha}{s^2}\text{cyl.w} + \frac{\omega}{s^2}\left(s^2 - \sin^2\alpha\right)\text{con.w}$$
$$= V + \frac{c\omega\sin^2\alpha}{s^2}\text{cyl.w} + \left(\omega - \frac{\omega\sin^2\alpha}{s^2}\right)\text{con.w}.$$

It is easy to see that Q is actually the point of intersection of the major axis lines of the two ellipses (or, when $\theta=\alpha$, the intersection of the tangent line with the major axis line of the ellipse). Since the cylinder axis and cone axis intersect and the form of Q is ($V + a_1$*cyl.w + a_2*con.w), the point Q clearly lies in the plane of the two quadric axes. By symmetry, this plane contains the major axis lines of the two ellipses. By construction, Q also lies in each of the two planes containing these ellipses. Therefore Q is the single point common to these three distinct planes and hence is the point of intersection of the two major axis lines.

As noted above, the intersection is a tangent line and an ellipse when $\theta=\alpha$. We now show that `plane`(Q,$\mathbf{n}_2$) is tangent to the cylinder and cone in this case. Since $\theta=\alpha$, $s=\sin\theta=\sin\alpha$, and the expression for Q simplifies to:

$$Q = V + \frac{c\omega s^2}{s^2}\text{cyl.w} + \left(\omega - \frac{\omega}{s^2}s^2\right)\text{con.w}$$
$$= V + c\omega\text{cyl.w}.$$

Since cyl.w·con.w=$\cos\theta=\cos\alpha$, cyl.w is parallel to a cone ruling. Therefore in this case, Q is a point on the cone. Since the vertex V must also be on the cylinder when the two share a ruling, it follows that Q is also a point on the cylinder. Now consider normal vector $\mathbf{n}_2$:

$$\text{cyl.w}\cdot\mathbf{n}_2 = \text{cyl.w}\cdot(\text{cyl.w} - F\text{con.w}) = 1 - \frac{\cos\theta}{\cos\alpha} = 1 - 1 = 0.$$

The vector $\mathbf{n}_2$ is therefore perpendicular to cyl.w. By our earlier observations $\mathbf{n}_2$ is therefore perpendicular to both the cylinder and the cone along their respective rulings through the point Q. Thus `plane`(Q,$\mathbf{n}_2$) is tangent to both surfaces along a ruling. We shall use this fact in the algorithm below. When we determine that $\theta=\alpha$, we shall only intersect the cylinder with `plane`(Q,$\mathbf{n}_1$); we will express the tangent line directly.

Finally recall that the magnitude of ω is the distance from the cone vertex to the point I at which the axes intersect. However ω is a signed distance, since it is the z coordinate of I in the canonical coordinate system of the cone. We therefore compute ω in a geometrically invariant manner as the signed distance of I along the cone axis line. For simplicity we show I as a parameter to this routine. This is reasonable since I would have been computed previously while testing to see if the intersection is planar.

We summarize these results in the following pseudo-code.

```
input: cyl: cylinder; con: cone; I: point
cos_theta := cyl.w·con.w
sin_sqr_theta := 1 - cos_theta*cos_theta
cos_alpha := cos(con.α)
F := 1 / cos_alpha
sin_sqr_alpha := 1 - cos_alpha*cos_alpha
ω := signed_distance_along_line(I , line(con.V,con.w))
n1 := normalize(cyl.w + F*con.w)
t := ω*sin_sqr_alpha/sin_sqr_theta
Q := con.V + cos_theta*t*cyl.w + (ω-t)con.w
conic_1 := intersect(cyl , plane(Q,n1))
if abs(cos_theta) = cos_alpha then
     conic_2 := line(con.V,cyl.w)
else
     n2 := normalize (cyl.w - F*con.w)
     conic_2 := intersect(cyl , plane(Q,n2))
```

Figure 11 illustrates the three possible results: (a) two intersecting ellipses on the same half of the cone, (b) an ellipse plus a tangentially shared ruling, and (c) two ellipses on opposite halves of the cone.

4.6 Cone-Cone Intersections

We showed in [2] that the intersection between two cones is a planar curve if and only if one of the following three conditions is satisfied: (i) the axes are distinct but parallel, and the cones have the same half angle; (ii) the axes are coincident; or (iii) the axes intersect at a point I equidistant from the two cones. Condition (iii) is equivalent to $r_1 \sin\alpha_1 = r_2 \sin\alpha_2$ where r_i is the distance from vertex i to I, and α_i is the half angle of cone i. We now consider each of these three cases in turn.

4.6.1 Distinct Parallel Axes, Same Half Angle

When condition (i) is satisfied, Q(1) as derived in [2] simplifies to:

$$Q(1) = \begin{pmatrix} 0 & 0 & 0 & -\mu \\ 0 & 0 & 0 & -\upsilon \\ 0 & 0 & 0 & -\omega(1-F_2^2) \\ -\mu & -\upsilon & -\omega(1-F_2^2) & \mu^2+\upsilon^2+\omega^2-F_2^2\omega^2 \end{pmatrix}.$$

Since this corresponds to the equation of a single plane, the intersection in this case is a single (possibly degenerate) conic. The intersection will be a hyperbola, an ellipse, or a double line if the vertex of one cone is, respectively, outside, inside, or on the other cone. (See Figure 12.) The intersection cannot be a parabola, nor can it be a circle since the axes are distinct. A circle is possible only when the axes coincide. This situation is discussed in Section 4.6.2.

We shall derive an invariant characterization of the plane in terms of the parameters of the two cones which will yield the correct plane for any of these relative vertex locations. In principle, then, we need not distinguish between these cases in the code. As we argued in Section 4.5.2, however, it is prudent to detect the tangent line case since it is especially delicate numerically.

Multiplying $Q(1)$ by -1/2 and writing the corresponding implicit equation, we get:

$$\mu x + \upsilon y + \omega(1-F_2^2)z - (\mu^2 + \upsilon^2 + \omega^2(1-F_2^2))/2 = 0. \qquad (8)$$

The first cone is in canonical position, and the second is in relative canonical position. From Table V, V_1=(0,0,0), $\mathbf{w}_1$=(0,0,1), and V_2=(μ,υ,ω). We can therefore express the vector normal to this plane as:

$$\mathbf{n} = (\mu, \upsilon, \omega(1-F_2^2)) = (\mu, \upsilon, \omega) - F_2^2\omega(0,0,1) = (V_2 - V_1) - F_2^2\omega\mathbf{w}_1$$

Notice that ω is simply the signed distance between V_2 and the plane determined by (V_1,$\mathbf{w}_1$).

Now we need only determine a point Q on the plane $Q(1)$. Substituting x=μ/2 and y=υ/2 into (8) and solving for z, we find:

$$z = \frac{(\mu^2 + \upsilon^2 + \omega^2(1-F_2^2))/2 - \mu^2/2 - \upsilon^2/2}{\omega(1-F_2^2)} = \frac{\omega}{2}$$

Thus a point on the plane is Q=(μ/2,υ/2,ω/2). In general position, this point Q is simply the midpoint of the line segment joining the vertices. Q is also the center of the conic intersection curve when the conic is an ellipse or a hyperbola. The rationale is as follows. The two cones are identical except for position, and hence the conic must be symmetrically located with respect to each cone. Consider the three planes: (i) the locus of points equidistant from V_1 and V_2: `plane`(Q, `normalize`(V_2-V_1)); (ii) the locus of points equidistant from `plane`(V_1,**w**) and `plane`(V_2,**w**) where **w**=$\mathbf{w}_1$=±$\mathbf{w}_2$: plane(Q,**w**); and (iii) the plane containing the axes of the cones: plane(O,**n**) where **n**=`normalize`(**w** x (V_2-V_1)). The point O can clearly be either V_1 or V_2, and hence Q as well. The center of the conic must be on all three of these planes by symmetry. Since all three are clearly distinct, and since Q lies on all three, Q must be the center of the conic.

We summarize these results in the following algorithm. Figure 12 illustrates the possible results.

```
input: C1, C2: cone
v1_to_v2 := C2.V - C1.V
if C2.V on C1 { See Section 2 } then
   conic := line(C1.V , normalize(v1_to_v2))
else
   ω := v1_to_v2·C1.w
   Q := midpoint(C1.V , C2.V)
   F2 := sec(C2.α)
   n := normalize(v1_to_v2 - ω*F2*F2*C1.w)
   conic := intersect(C1 , plane(Q,n))
```

4.6.2 Coincident Axes

We next consider condition (ii), the case of coincident axes. This case is sufficiently simple that geometric analysis suffices; further algebraic analysis is unnecessary. We assume that the vertices are distinct since the cones would otherwise be identical or intersect only at their common vertex. Furthermore we assume without loss of generality that C1.α≥C2.α. In general, the intersection is one or two circles, depending on whether the cone half angles are the same or different. When the half angles are the same, the analysis is simple and is summarized in the pseudo-code below. When the half angles are different, the two circles will lie on the same half of C2. See Figure 13a. We seek the distances *a* and *d* which are, respectively, the distance from C2.*V* to the circle closest to it and the distance from C1.*V* to the other circle. Once we have computed these distances, we can easily construct the center point of the two circles and calculate their radii as summarized in the pseudo-code below. To compute these distances, we consider the distance *h* between the two vertices and write two pairs of equations in two unknowns, one pair to find *a* and one to get *d*:

$$h = a + b$$
$$a \tan\alpha_2 = b \tan\alpha_1$$
$$\therefore a = h \frac{\tan\alpha_1}{\tan\alpha_1 + \tan\alpha_2}$$

$$c = h + d$$
$$c \tan\alpha_2 = d \tan\alpha_1$$
$$\therefore d = h \frac{\tan\alpha_2}{\tan\alpha_1 - \tan\alpha_2}$$

Collecting these results, we can write the following algorithm.

```
input: C1, C2: cone
E1  := tan(C1.α)
E2  := tan(C2.α)
if C1.α = C2.α then { intersection is a circle; refer to Figure 13b }
   output: circle:   C := midpoint(C1.V,C2.V)
                     w := C1.w
                     r := distance(C1.V,C) *E1
else { intersection is two circles; refer to Figure 13a }
   h := distance(C1.V,C2.V)
   v2_to_v1 := normalize(C1.V - C2.V)
   a := h * E1 / (E1 + E2)
   d := h * E2 / (E1 - E2)
   output:    circle 1: C  := C2.V + a*v2_to_v1
                        w  := C2.w
                        r  := a*E2
              circle 2: C  := C1.V + d*v2_to_v1
                        w  := C1.w
                        r  := d*E1
```

4.6.3 Intersecting Axes

Finally we consider the more complex condition (iii). If the vertices of the two cones are identical (i.e.,

$r_1 = r_2 = 0$), the cones intersect in one to four lines, or they intersect only at their common vertex. If the vertices are distinct (i.e., condition (iii) is satisfied with $r_i \neq 0$), the intersection may consist of various pairs of (possibly degenerate) conics depending on the angle between the axes and the cone half angles.

As before, our approach will be to compute the pair of planes containing the conics, and then to perform two plane-cone intersections. When doing so, we need not differentiate between the coincident and non-coincident vertex cases. As we shall see, the common point which we derive for the two planes will be the shared vertex if the vertices are coincident. While the derivation of the planes containing the conics is rather long, the final algorithm is fairly simple.

In [2] we demonstrated that in this subcase, $\mu=0$ and $s \neq 0$. The matrix representing the pair of planes in this case is therefore [2]:

$$Q(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -s^2F_2^2 & -scF_2^2 & sc\omega F_2^2 - \upsilon(1 - s^2F_2^2) \\ 0 & -scF_2^2 & F_1^2 - c^2F_2^2 & sc\upsilon F_2^2 - \omega(1 - c^2F_2^2) \\ 0 & sc\omega F_2^2 - \upsilon(1 - s^2F_2^2) & sc\upsilon F_2^2 - \omega(1 - c^2F_2^2) & \upsilon^2 + \omega^2 - F_2^2(s\upsilon + c\omega)^2 \end{pmatrix}.$$

The final constraint derived for this case was Cofactor $Q_{11}=0$. This led to the following equation [2]:

$$\left[s^2E_1^2F_2^2 + \left(F_2^2 - F_1^2\right)\right]\upsilon^2 + 2scE_1^2F_2^2\upsilon\omega - s^2E_1^2F_2^2\omega^2 = 0. \quad (9)$$

Multiplying the matrix by -1 and writing the polynomial describing the planes, we get:

$$\begin{aligned} &s^2F_2^2y^2 + 2scF_2^2yz + \left(c^2F_2^2 - F_1^2\right)z^2 \\ &+2\left(\left(1 - s^2F_2^2\right)\upsilon - scF_2^2\omega\right)y + 2\left(\left(1 - c^2F_2^2\right)\omega - scF_2^2\upsilon\right)z \\ &+F_2^2(s\upsilon + c\omega)^2 - \upsilon^2 - \omega^2 = 0. \end{aligned} \quad (10)$$

As before, we need to factor this polynomial to determine the two planes. We know that it will factor, so we can proceed in the following fashion. We seek values for a_i and b_i such that the left-hand side of (10) factors as:

$$\left(a_1y + a_2z + a_3\right)\left(b_1y + b_2z + b_3\right). \quad (11)$$

By equating coefficients of terms in (10) with the corresponding ones in the expansion of (11), we note that:

$$s^2F_2^2y^2 + 2scF_2^2yz + \left(c^2F_2^2 - F_1^2\right)z^2 = \left(a_1y + a_2z\right)\left(b_1y + b_2z\right).$$

It is then straightforward to verify that:

$$\begin{aligned} a_1 &= b_1 = sF_2 \\ a_2 &= \left(cF_2 + F_1\right) \\ b_2 &= \left(cF_2 - F_1\right). \end{aligned}$$

Therefore the normals to the two planes in canonical and general position are:

$$\mathbf{n}_1 = (0, a_1, a_2) = (0, sF_2, cF_2 + F_1) = F_2(0, s, c) + F_1(0,0,1) = F_2\mathbf{w}_2 + F_1\mathbf{w}_1$$
$$\mathbf{n}_2 = (0, b_1, b_2) = (0, sF_2, cF_2 - F_1) = F_2(0, s, c) - F_1(0,0,1) = F_2\mathbf{w}_2 - F_1\mathbf{w}_1.$$

Next equating the terms in (10) and (11) which are linear in y and z, we observe:

$$b_1 a_3 + a_1 b_3 = 2\left(\left(1 - s^2F_2^2\right)\upsilon - scF_2^2\omega\right)$$
$$b_2 a_3 + a_2 b_3 = 2\left(\left(1 - c^2F_2^2\right)\omega - scF_2^2\upsilon\right).$$

Solving these equations for a_3 and b_3 by Cramer's Rule, we find:

$$a_3 = \frac{\det\begin{vmatrix}\left(1 - s^2F_2^2\right)\upsilon - scF_2^2\omega & sF_2 \\ \left(1 - c^2F_2^2\right)\omega - scF_2^2\upsilon & cF_2 + F_1\end{vmatrix}}{sF_1F_2}$$

and

$$b_3 = \frac{\det\begin{vmatrix} sF_2 & \left(1 - s^2F_2^2\right)\upsilon - scF_2^2\omega \\ cF_2 - F_1 & \left(1 - c^2F_2^2\right)\omega - scF_2^2\upsilon\end{vmatrix}}{sF_1F_2}.$$

As a check, note that a_3b_3 must be identically equal to the constant term in (10). This is easy to verify.

$$a_3 = \frac{\left(c\upsilon\left(1 - s^2F_2^2\right)F_2 + \left(1 - s^2F_2^2\right)\upsilon F_1 - sc^2F_2^3\omega - scF_1F_2^2\omega\right) - \left(sF_2\omega - sc^2F_2^3\omega - s^2cF_2^3\upsilon\right)}{sF_1F_2}$$

$$= \frac{c\upsilon F_2 + \upsilon F_1 - s^2\upsilon F_1F_2^2 - scF_1F_2^2\omega - sF_2\omega}{sF_1F_2} \tag{12}$$

$$b_3 = \frac{\left(sF_2\omega - sc^2F_2^3\omega - s^2cF_2^3\upsilon\right) - \left(cF_2\upsilon - cs^2F_2^3\upsilon - F_1\upsilon + s^2F_1F_2^2\upsilon - sc^2F_2^3\omega + scF_1F_2^2\omega\right)}{sF_1F_2}$$

$$= \frac{\left(sF_2\omega - cF_2\upsilon + F_1\upsilon - s^2F_1F_2^2\upsilon - scF_1F_2^2\omega\right)}{sF_1F_2} \tag{13}$$

$$a_3b_3 = \frac{\left(\left(F_1\upsilon - s^2F_1F_2^2\upsilon - scF_1F_2^2\omega\right) + \left(cF_2\upsilon - sF_2\omega\right)\right)\left(\left(F_1\upsilon - s^2F_1F_2^2\upsilon - scF_1F_2^2\omega\right) - \left(cF_2\upsilon - sF_2\omega\right)\right)}{s^2F_1^2F_2^2}$$

$$= [\upsilon^2\left(F_1^2 + s^4F_1^2F_2^4 - 2s^2F_1^2F_2^2 - c^2F_2^2\right) +$$
$$\upsilon\omega\left(-2scF_1^2F_2^2 + 2s^3cF_1^2F_2^4 + 2scF_2^2\right) +$$
$$\omega^2\left(s^2c^2F_1^2F_2^4 - s^2F_2^2\right)]/\left(s^2F_1^2F_2^2\right)$$

Rewriting as the sum of an integral and a rational term, we get:

$$a_3b_3 = \left\{\upsilon^2\left(s^2F_2^2 - 2\right) + \upsilon\omega\left(2scF_2^2\right) + \omega^2\left(c^2F_2^2\right)\right\} + \left\{\frac{\upsilon^2\left(F_1^2 - c^2F_2^2\right) + \upsilon\omega\left(2scF_2^2 - 2scF_1^2F_2^2\right) + \omega^2\left(-s^2F_2^2\right)}{s^2F_1^2F_2^2}\right\}$$

Now the first term in braces is nearly the same as the constant term in (10). We need only add (υ^2-ω^2). Doing so and balancing by subtracting (υ^2-ω^2) from the second term, we get:

$$a_3b_3 = \left\{\upsilon^2\left(s^2F_2^2 - 1\right) + \upsilon\omega\left(2scF_2^2\right) + \omega^2\left(c^2F_2^2 - 1\right)\right\} + \left\{\frac{\upsilon^2\left(F_1^2 - c^2F_2^2 - s^2F_1^2F_2^2\right) + \upsilon\omega\left(2scF_2^2 - 2scF_1^2F_2^2\right) + \omega^2\left(s^2F_1^2F_2^2 - s^2F_2^2\right)}{s^2F_1^2F_2^2}\right\}$$

We now demonstrate that the second term is identically zero by showing that the numerator is zero. Recall that F^2=E^2+1.

$$\begin{aligned}\text{numerator} &= -\upsilon^2\left(s^2F_1^2F_2^2 + \left(F_2^2 - F_1^2\right)\right) - \upsilon\omega\left(2csF_1^2F_2^2\right) - \omega^2\left(-s^2F_1^2F_2^2\right) \\ &\quad + \upsilon^2\left(s^2F_2^2\right) + \upsilon\omega\left(2scF_2^2\right) + \omega^2\left(-s^2F_2^2\right) \\ &= -\upsilon^2\left(s^2E_1^2F_2^2 + \left(F_2^2 - F_1^2\right)\right) - \upsilon\omega\left(2scE_1^2F_2^2\right) - \omega^2\left(-s^2E_1^2F_2^2\right) \\ &\quad + \upsilon^2\left(s^2F_2^2 - s^2F_2^2\right) + \upsilon\omega\left(2scF_2^2 - 2scF_2^2\right) + \omega^2\left(-s^2F_2^2 + s^2F_2^2\right)\end{aligned}$$

The final three terms are identically zero, and our current constraint (equation (9) above) tells us that the first three terms are zero.

All that remains is to find a point common to the two planes. We do so by solving simultaneously the two plane equations determined by the factorization of (10) which we have just derived. The two planes are:

$$\begin{aligned} sF_2y + \left(cF_2 + F_1\right)z + a_3 &= 0 \\ sF_2y + \left(cF_2 - F_1\right)z + b_3 &= 0.\end{aligned}$$

Replacing the first with half their sum and the second with half their difference, we get:

$$\begin{aligned} sF_2y + cF_2z + \left(a_3 + b_3\right)/2 &= 0 \\ F_1z + \left(a_3 - b_3\right)/2 &= 0.\end{aligned}$$

Clearly the solution is:

$$x = \text{anything}$$
$$y = \frac{cF_2(a_3 - b_3) - F_1(a_3 + b_3)}{2sF_1F_2}$$
$$z = \frac{b_3 - a_3}{2F_1}. \tag{14}$$

Choosing x=0, we write a geometrically invariant expression for the common point Q as:

$$Q = \text{C1}.V + \frac{y}{s}(0,s,c) + \left(z - \frac{cy}{s}\right)(0,0,1)$$
$$= \text{C1}.V + \frac{y}{s}\text{C2.w} + \left(z - \frac{cy}{s}\right)\text{C1.w}. \tag{15}$$

The work required to compute Q can be simplified considerably by using the expressions for a_3 and b_3 in (12) and (13) to derive the following quantities:

$$e_1 = b_3 - a_3 = \frac{2sF_2\omega - 2cF_2\upsilon}{sF_1F_2} = \frac{2(s\omega - c\upsilon)}{sF_1}, \tag{16}$$
$$e_2 = b_3 + a_3 = \frac{2F_1\upsilon - 2s^2\upsilon F_1F_2^2 - 2scF_1F_2^2\omega}{sF_1F_2} = \frac{2\left(\upsilon\left(1 - s^2F_2^2\right) - scF_2^2\omega\right)}{sF_2}. \tag{17}$$

Substituting (14), (16), and (17) into (15), we find:

$$Q = \text{C1}.V + \frac{cF_2(-e_1) - F_1e_2}{2s^2F_1F_2}\text{C2.w} + \left(\frac{e_1}{2F_1} - \frac{c^2F_2(-e_1) - cF_1e_2}{2s^2F_1F_2}\right)\text{C1.w}$$
$$= \text{C1}.V - g\text{C2.w} + \left(\frac{e_1}{d_1} + cg\right)\text{C1.w}$$

where

$$g = \frac{y}{s} = \frac{ce_1}{d_2} + \frac{e_2}{d_3}$$
$$d_1 = 2F_1$$
$$d_2 = s^2d_1$$
$$d_3 = 2s^2F_2.$$

By an argument analogous to that in Section 4.5.2, we again observe that Q is the point of intersection of the major axes of the two conics.

Observe that if the vertices are coincident, then $\upsilon=\omega=0$ which means that e_1 and e_2 are also zero. Therefore the common point Q will be C1.V, the common vertex. In this case, the cones may intersect only at their common vertex, or they may share 1-4 lines. If the cones intersect only at their common vertex, then both planes computed by this algorithm will intersect the cones only at their common vertex. If they intersect in one tangential or two distinct non-tangential rulings, then one of the planes will contain the line(s), and the other will intersect the cones only at the vertex. The upshot of these observations is simply that a filter must be applied at the end of this algorithm so that (i) if the cones intersect only at their vertex, the vertex intersection is reported only once, and (ii) if they intersect in one tangential or two distinct non-tangential rulings, then we do not report an additional vertex intersection.

Finally we need to derive invariant expressions for υ and ω. From Table V, we observe that these are simply the y and z coordinates of C2.V in the canonical coordinate system of C1. The z-axis of this coordinate system is given by C1.w, hence we can immediately write:

$$\omega = (\mathrm{C2}.V - \mathrm{C1}.V) \cdot \mathrm{C1.w}.$$

The y-axis of this coordinate system is the unit vector whose direction is given by the component of C2.w perpendicular to C1.w:

$$\mathbf{yaxis} = \frac{(\mathrm{C2.w} - (\mathrm{C2.w} \cdot \mathrm{C1.w})\mathrm{C1.w})}{|\mathrm{C2.w} - (\mathrm{C2.w} \cdot \mathrm{C1.w})\mathrm{C1.w}|}.$$

The denominator simplifies to:

$$\begin{aligned}&|\mathrm{C2.w} - (\mathrm{C2.w} \cdot \mathrm{C1.w})\mathrm{C1.w}| \\ &\quad = \sqrt{\mathrm{C2.w} \cdot \mathrm{C2.w} - 2(\mathrm{C2.w} \cdot \mathrm{C1.w})^2 + (\mathrm{C2.w} \cdot \mathrm{C1.w})^2 \mathrm{C1.w} \cdot \mathrm{C1.w}} \\ &\quad = \sqrt{1 - 2c^2 + c^2} = \sqrt{1 - c^2} = s.\end{aligned}$$

Therefore we compute υ as:

$$\begin{aligned}\upsilon &= \mathbf{yaxis} \cdot (\mathrm{C2}.V - \mathrm{C1}.V) \\ &= \frac{(\mathrm{C2}.V - \mathrm{C1}.V) \cdot \mathrm{C2.w} - (\mathrm{C2.w} \cdot \mathrm{C1.w})(\mathrm{C2}.V - \mathrm{C1}.V) \cdot \mathrm{C1.w}}{s} \\ &= \frac{(\mathrm{C2}.V - \mathrm{C1}.V) \cdot \mathrm{C2.w} - c\omega}{s}.\end{aligned}$$

We summarize the handling of condition (iii) in the following pseudo-code.

```
input: C1, C2: cone
/* Compute trigonometric constants */
c := C1.w·C2.w
c_sqr := c*c
s_sqr := 1.0 - c_sqr
s := sqrt(s_sqr)
F1 := sec(C1.α)
F2 := sec(C2.α)
F2_sqr := F2*F2

/* Compute vectors normal to planes containing conics */
n1 := normalize(F2*C2.w + F1*C1.w)
```

```
n2 := normalize(F2*C2.w - F1*C1.w)

/* Compute point common to two planes */
if C1.V = C2.V then
    Q := C1.V
else
    v1_to_v2 := C2.V - C1.V
    ω := v1_to_v2·C1.w
    υ := (v1_to_v2·C2.w - c*ω)/s
    e1 := 2(s*ω-c*υ)/(s*F1)
    e2 := 2(υ(1-s_sqr*F2_sqr)-s*c*F2_sqr*ω)/(s*F2)
    d1 := 2*F1
    d2 := s_sqr * d1
    d3 := 2 * s_sqr * F2
    g := c*e1/d2  +  e2/d3
    Q := C1.V - g*C2.w + (e1/d1 + c*g)C1.w

/* Compute the two (possibly degenerate) conics by intersecting one of the
cones with the two planes. */
conic_1 := intersect(C1 , plane(Q,n1))
conic_2 := intersect(C1 , plane(Q,n2))

/* Finally, ensure that no extraneous vertex intersections are reported. */
if conic_1 is a single point then
    return only conic_2
else if conic_2 is a single point then
    return only conic_1
else
    return both conic_1 and conic_2
```

Figure 14 illustrates the results of applying this single algorithm to a variety of cone pairs. Shown are coincident vertices yielding two real lines of intersection (Figure 14a), a pair of intersecting ellipses (Figure 14b), an ellipse plus a tangentially shared ruling (Figure 14c), and an ellipse plus a hyperbola (Figure 14d).

Note that if the cone half angles are the same, our constraint tells us that the point of intersection of the axes is equidistant from the two vertices. Moreover, the planes containing the conics are perpendicular since, up to constant multiples, $\mathbf{n}_1$=C2.w+C1.w and $\mathbf{n}_2$=C2.w-C1.w. This is quite similar to the cylinder-cylinder case studied in Section 4.4.2 and may be worth treating as a special case since a considerable amount of computation can be saved.

4.7 On Efficiency, Good Software Engineering, and Numerical Robustness

In presenting the algorithms of the preceding sections, our primary goal was clarity of presentation rather than maximal efficiency of implementation. While these algorithms are reasonably efficient as well, we indicate here some additional low-level observations which could be exploited to improve the efficiency of these algorithms, especially those for cylinder-cone and cone-cone intersections.

A word of caution: oftentimes the goals of good software engineering conflict with those of computational efficiency. Moreover, organizing computations in order to speed up algorithms by saving floating point operations is often not worth the added effort. An average workstation can perform on the order of a million floating point operations per second, and this figure is steadily rising. To see any significant decrease in interactive response time from the use of many of the tricks mentioned here, we would have to invoke the algorithms thousands or perhaps tens of thousands of times in response to an interactive request.

Nevertheless, there is something to be said for minimizing computation, but it has to do not so much with speed as with numerical robustness. Generally speaking, the more calculation required to arrive at a particular value, the less accurate that value can be expected to be. Many of our derivations and algorithmic organizations were motivated by this observation. The primary benefits of the algorithms we have presented here are (i) guaranteed handling of all planar cases involving the natural quadrics, and (ii) numerical reliability due to the exclusive use of geometric approaches in the implementation. The improved numerical robustness is due in part to our ability to assign meaningful tolerances when testing for equality and in part to our ability to reduce required computations. That is, by exploiting our understanding of the geometric meaning of various expressions, we were often able to derive simpler and more compact formulas for certain points and vectors than would otherwise have been possible.

Having said this, we present the following observations and leave it to those who implement these algorithms in production systems to weigh the tradeoffs given the anticipated use of their systems. In our implementation, we have incorporated none of these optimizations.

When the intersection is a pair of conics (hence when two plane normal vectors are to be computed and normalized), the lengths of the two unnormalized vectors can generally be calculated simultaneously. Using these lengths to normalize the vectors directly requires less computation than invoking a general vector normalization routine for each of the two vectors.

In several of the algorithms, we compute trigonometric functions for cone half angles and angles between axis vectors. Since these quantities are often also needed to test the planarity conditions in Table IV, they could certainly be passed to the algorithms presented here, thereby avoiding their recomputation.

As we observed in cylinder-cylinder intersections, certain axis vectors oftentimes are shared between the two conics which result from a given intersection. Our cylinder-cylinder algorithm was structured to exploit this directly since so much of the information is common between the two ellipses. In cylinder-cone and cone-cone intersections, however, we did not do this. While there is less to be gained, certainly some efficiency improvements can be realized by, for example, computing the common minor axis vector once and passing it to the two plane-cone intersection algorithms.

In Section 4.6.1, we noted that the common point Q is actually the center of the conic of intersection. Since much of the work of a plane-cone intersection algorithm is related to finding the center, we could pass this center to the plane-cone algorithm, thereby generating the conic section more efficiently and with greater numerical precision.

Algorithms such as the ones we have presented here are required by the boundary evaluation algorithm in solid modeling. This algorithm is extremely complex and will generally fail if the curve and surface intersection utilities on which it depends give incorrect or, worse, inconsistent answers. Once the curves of intersection between the various surfaces are determined, the curves must be partitioned where they meet other surfaces or curves. A particularly delicate situation relates to the subject of this paper: degenerate intersections. As we have seen, we often get pairs of conics which intersect each other at one or two points. The boundary evaluation algorithm must be able to find these points reliably. Because of numerical fuzz, however, it is possible that a pair of surfaces are determined to intersect in intersecting conics, yet subsequent conic-conic intersection algorithms find no intersection between them. Generally speaking, we know in the algorithms presented here when the conics will intersect each other, and it would be relatively straightforward to modify the algorithms so that we directly compute the points of intersection from the surface data (i.e., without invoking a conic-conic intersection utilitity). These points could then be returned to the caller along with the conics themselves.

Finally recall our earlier discussion of the one-point and two-point tangency configurations. It is certainly straightforward to implement the tests of Table III and to design appropriate geometric constructions, both to test the conditions indicated there as well as to construct geometrically the two tangent points. This approach would most likely be more computationally efficient and probably more numerically reliable than the approach indicated here. We chose not to do this in our implementation since we wanted to detect one-point tangencies as well. We observed that we could detect and calculate one-point and two-point tangencies as described with a fairly small investment in additional code. The amount of additional code is far less than what would be required to test explicitly the conditions in Table III, not to mention what would then be required to actually implement the admittedly straightforward geometric constructions to compute the tangencies. Moreover we would still need another strategy anyway to detect the presence and compute the position of single points of tangency.

5.0 Other Possible Degenerate Intersections

By Bezout's Theorem, two quadric surfaces always intersect in a degree four curve in complex projective space. So far we have been concerned here and in [2] almost exclusively with situations where this degree four curve splits into two (possibly degenerate) degree two curves. The other possible way in which the intersection curve can degenerate is into a line and a non-degenerate degree three space curve. This cannot happen when spheres are involved since there are no straight lines on a sphere. Clearly too it cannot happen in cylinder-cylinder intersections since the cylinder axis vectors must be parallel in order for the intersection to contain a straight line. But when their axes are parallel, two cylinders either have no real intersection, or they intersect in one tangent or two parallel lines. Therefore we need only concern ourselves with cylinder-cone and cone-cone intersections. The following two theorems present necessary and sufficient conditions for the line plus space cubic case to arise in cone-cylinder and cone-cone intersections.

Theorem 1: The intersection of a cylinder and a cone degenerates into a line and a space cubic if and only if all the following hold:
(i) The angle θ between the axis vectors is the same as the cone half angle α.
(ii) The cone vertex lies on the cylinder.
(iii) The axes are skew.

Proof: The proof proceeds as follows. We first show that the cylinder and cone share a single line if and only if conditions (i) and (ii) hold. We then show that it is shared non-tangentially (hence the remainder of the intersection must be an irreducible degree three curve) if and only if condition (iii) holds. We denote the geometric parameters of the cone by $(V,\mathbf{w}_{con},\alpha)$ and those of the cylinder by $(B,\mathbf{w}_{cyl},r)$.

By condition (i), $\mathbf{w}_{cyl}\cdot\mathbf{w}_{con}=\cos\theta=\cos\alpha$. Therefore there is a ruling on the cone which is parallel to the cylinder axis vector. Since by condition (ii), V lies on the cylinder, it follows that `line`$(V,\mathbf{w}_{cyl})$ lies on both the cylinder and the cone. Since all rulings on the cone intersect at V and since all cylinder rulings are parallel to each other, `line`$(V,\mathbf{w}_{cyl})$ can be the only ruling the two share.

Conversely suppose that the cylinder and cone share a ruling. Clearly V must lie on the cylinder since all cone rulings contain V. The direction vector of all rulings on the cylinder is $\mathbf{w}_{cyl}$, and the direction vector of all rulings on the cone form an acute angle α with $\mathbf{w}_{con}$. By defintion, $\mathbf{w}_{cyl}\cdot\mathbf{w}_{con}=\cos\theta$. Since $\mathbf{w}_{cyl}$ must be the direction vector for a cone ruling, $\mathbf{w}_{cyl}\cdot\mathbf{w}_{con}=\cos\alpha$. Therefore $\cos\theta=\cos\alpha$, and hence $\theta=\alpha$.

Now assume conditions (i) and (ii) hold. We wish to show that the intersection contains a non-degenerate degree three space curve if and only if condition (iii) holds. Suppose the axes are not skew. Since $\theta=\alpha$, the axes cannot be parallel, so they must intersect. But when the axes intersect, the plane determined by them is a mirror symmetry plane for both the cone and the cylinder. Because we know conditions (i) and (ii) hold, the cross section of the geometry as determined by this mirror symmetry plane must be as shown in Figure 15. As is clear from the figure, if the axes intersect while conditions (i) and (ii) hold, then the point I of intersection must lie at a distance $r/\sin\alpha$ from the vertex of the cone. However we showed in [2] that when the axes intersect in such a point, the intersection curve is planar.

Finally, suppose conditions (i) and (ii) hold and the axes are skew. If the remainder of the intersection were planar, then it must contain a conic since the only planar curves which lie on quadrics are conics. But we showed in [2] that conics can never arise in the intersection of a cylinder and a cone unless their axes are either coincident or intersect (see Table IV). Hence the remainder of the intersection cannot be planar; therefore it must be a non-degenerate degree three space curve. *QED.*

As noted in the proof of Theorem 1, the line of intersection is given by `line`$(V,\mathbf{w}_{cyl})$. Calculation of the space cubic is discussed in [5]. The best way to visualize the geometry giving rise to a line and a space cubic is to start with the geometry of Figure 11b. Recall in this case that the cylinder and cone axes intersect, and the intersection curve splits into a tangent line and an ellipse. If we rotate either the cylinder or the cone about their common line, the axes become skew, the ruling is no longer shared tangentially, and the ellipse breaks apart into a space cubic at the point where it intersects the shared ruling. One end of the ellipse then tends towards infinity in one direction, and the other towards infinity in the opposite direction. See Figure 16.

Theorem 2: The intersection of two cones degenerates into a line and a space cubic if and only if all the following hold:
(i) Each vertex lies on the other cone.
(ii) The vertices are distinct.
(iii) The axes are skew.

Proof: The proof proceeds as follows. We first show that if conditions (i) and (ii) hold, the cones share a single line. We then show that if condition (iii) also holds, the remainder of the intersection must be an irreducible degree three space curve. To establish the converse, we also show that if the intersection is a line and an irreducible degree three space curve, then conditions (i), (ii), and (iii) must hold. We call the first cone C_1 and denote its geometric parameters by $(V_1,\mathbf{w}_1,\alpha_1)$. Similarly, we call the second cone C_2 and denote its geometric parameters by $(V_2,\mathbf{w}_2,\alpha_2)$.

By condition (ii), the vector $\mathbf{v}=(V_2-V_1)$ is a non-zero vector. Since by condition (i) V_2 lies on cone C_1, `line`$(V_1,\mathbf{v})$=`line`$(V_2,\mathbf{v})$ must lie on C_1. By a similar argument, this line also lies on C_2. Hence C_1 and C_2 share a line. Because every line on a cone must pass through its vertex, it follows from condition (ii) that this can be the only line the two cones share. Now assume that condition (iii) also holds; we shall show that the remainder of the intersection curve is nonplanar. The only planar curves lying on cones are conics. We proved in [2] that the intersection of two cones contains a conic if and only if one of the following is true: (a) $\mathbf{w}_1 \| \mathbf{w}_2$ and $\alpha_1=\alpha_2$, (b) `line`$(V_1,\mathbf{w}_1)$=`line`$(V_2,\mathbf{w}_2)$, or (c) the axes intersect at a point equidistant from the surface of each cone (see Table IV). But condition (iii) states that the axes are skew; hence none of these conditions can hold. Therefore the remainder of the intersection cannot contain a conic, so it must be an irreducible degree three space curve.

Conversely suppose the intersection of C_1 and C_2 is a line and an irreducible degree three space curve. Clearly condition (i) must hold since all cone rulings pass through the vertex. In [2] we showed that when the vertices are coincident, the intersection is planar. By assumption, the intersection contains an irreducible degree three space curve. Therefore condition (ii) must hold. We now show that condition (iii) must hold by showing that the axes can be neither parallel nor intersecting. Suppose they are parallel. Since we have established already that conditions (i) and (ii) must hold, it follows that the half angles must be the same. But when the axes are parallel and the cone half angles are the same, the intersection is planar [2]. Therefore the axes cannot be parallel. Suppose they intersect in a single point I. The plane of the two axes is then a mirror symmetry plane for both cones, and the cross-section of the geometry as determined by that plane must be as shown in Figure 17. Clearly the distances d_i from I to V_i are related by $d_1\sin\alpha_1=d_2\sin\alpha_2$. But in [2] we showed that when this condition holds, the intersection is planar. Hence the axes cannot intersect; therefore they must be skew. *QED*.

As mentioned in the proof of Theorem 2, the line of intersection can be defined by `line`(V_1, `normalize`(V_2-V_1)). Calculation of the space cubic is discussed in [5]. One can visualize this geometry in a manner analogous to that described for the cylinder-cone case above. Start with the geometry of Figure 14c and twist one of the cones about the common ruling. The result is illustrated in Figure 18.

Given the geometric data for cylinders and cones, it is easy to implement analytic tests for the geometric conditions in Theorems 1 and 2. A procedure for deciding whether two lines are skew is described in Section 2. Formulas for deciding whether a given point lies on a cylinder or cone are also provided at the end of Section 2.

6.0 Conclusions

Detecting the presence of conic sections in quadric surface intersections is important for a variety of reasons including more efficient and reliable representation and analysis [2,5,8] and the possibility of blending with low degree surfaces [9]. Using algebraic geometry, we characterized in [2] all configurations under which the intersection of a given pair of natural quadrics is planar. Here we have developed methods for calculating these planar intersection curves once we have determined that they actually arise. These methods are based solely on the geometric data of natural quadrics in general position and orientation. This combination of algebraic and geometric approaches is ideal since it exploits the rigor of the algebraic method and the numerical reliability of computer implementations based on geometric representations.

All the algorithms described in this paper have been implemented by the first author in a solid modeling system being developed at the University of Kansas and have proven to be quite efficient and highly reliable in practice. The algorithms have been implemented in C under UNIX on a Silicon Graphics IRIS 4D/60 workstation. This machine runs at about 7 MIPS and is capable of approximately 0.7 MFLOPS. We measured the required execution times for the two most computationally intensive examples: the cylinder-cone case of Figure 11c and the cone-cone case of Figure 14d. Measurements indicate that the geometry of Figure 11c can be intersected approximately 480 times per second, while that of Figure 14d can be performed at the rate of about 330 per second.

It would be of interest to know how these results extend to other surface types. The most obvious candidates are other quadrics of revolution. Other possibilities include generalizations to all quadrics, to surfaces of revolution of degree greater than two, or even to other higher degree algebraic surfaces.

References

1. Goldman, R. N., Intersection of Two Lines in Three-Space, in *Graphics Gems,* edited by Andrew Glassner, Academic Press, 1990, p. 304.
2. Goldman, R. N. and Miller, J. R., *Detecting and Calculating Conic Sections in the Intersection of Two Natural Quadric Surfaces, Part I: Theoretical Analysis*, The University of Kansas, Department of Computer Science, Technical Report TR-93-1, January 1993.
3. Hakala, D. G., Hillyard, R. C., Nourse, B. E., and Malraison, P. J., Natural Quadrics in Mechanical Design, *Proceedings Autofact West,* Vol. 1, November, 1980.
4. Levin, J., A Parametric Algorithm for Drawing Pictures of Solid Objects Composed of Quadric Surfaces, *Communications of the ACM,* Vol. 19, No. 10, October 1976, pp. 555-563.
5. Miller, J.R., Geometric Approaches to Nonplanar Quadric Surface Intersection Curves, *ACM Transactions on Graphics,* Vol. 6, No. 4, October 1987, pp. 274-307.
6. Miller, J.R., Analysis of Quadric-Surface-Based Solid Models, *IEEE Computer Graphics and Applications,* Vol. 8, No. 1, January 1988, pp. 28-42.
7. Miller, J. R. and Goldman, R. N., Using Tangent Balls to Find Plane Sections of Natural Quadrics, *IEEE Computer Graphics and Applications*, Vol. 12, No. 2, March 1992, pp. 68-82.
8. Sarraga, R.F., Algebraic Methods for Intersections of Quadric Surfaces in GMSOLID, *Computer Vision, Graphics, and Image Processing,* V22, N2, May 1983.
9. Warren, J., Blending Algebraic Surfaces, *ACM Transactions on Graphics,* Vol. 8, No. 4, October 1989, pp. 263-278.