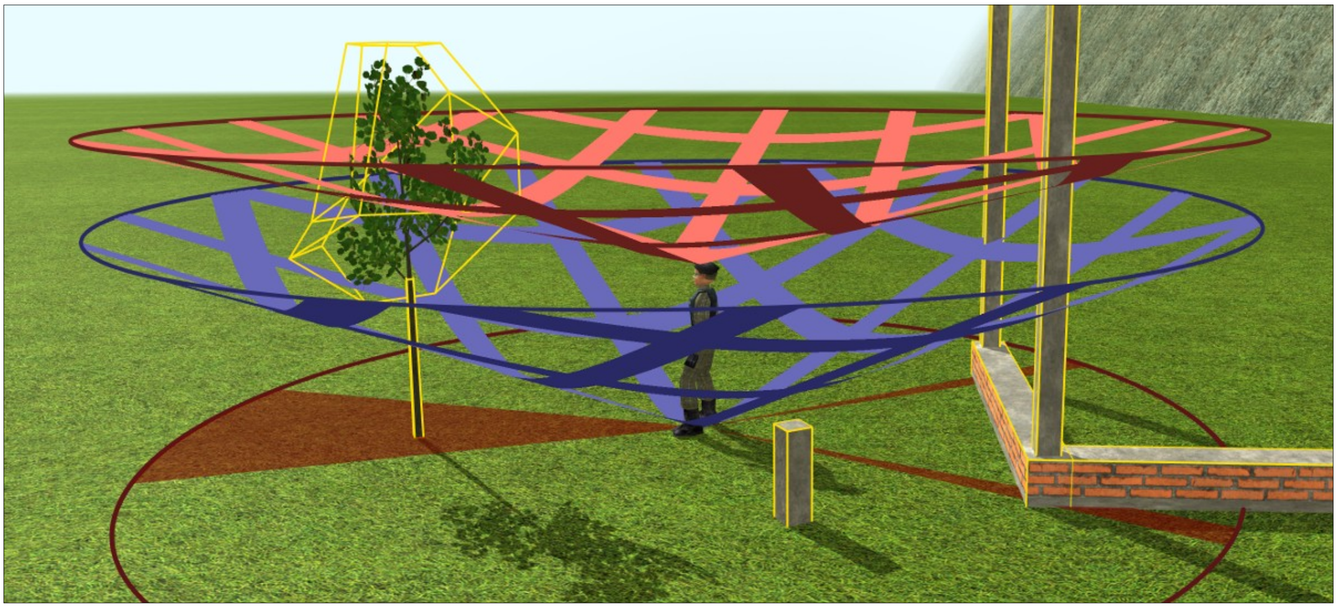
Visibility In 3D Among Polyhedra

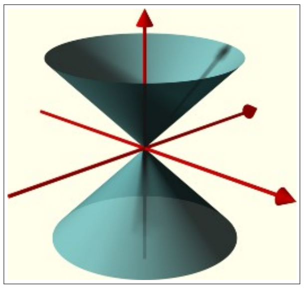


There is an object in the center of the image (h_{item}=1.8). The task is to find all possible locations for camera placement such that:

- Distance camera to item measured on the plane surface is given (d=5.5);
- Vertical distance camera to the plane is given (h_{camera}=3);
- Item must be visible from the camera. Obstacles are represented as polyhedra. Author defines "practically closed polyhedron" as a closed polyhedron with an exception that faces never visible from a camera may be missing. This includes faces obscured by other faces in all practical circumstances.
- Camera has some dimensions. Half height of the camera is given (h_{delta}=0.4).

A set of line segments such that one endpoint is common for all line segments and the other endpoint is off by horizontal distance d and vertical distance h_{camera} is a part of circular cone. Following equiation is used:

$$x^2 + z^2 - b^2 y^2 = 0 \ . \qquad (1)$$



Infinitely extending circular cone (image from Wikipedia).

Here, axis of the cone coincidents with axis Y (which is directed up in the application). Equation (1) defines an infinite cone of two halves. Additional property `halfSgn` is included in cone object, it defines if upper(1), lower(-1) or both(0) halves are considered.

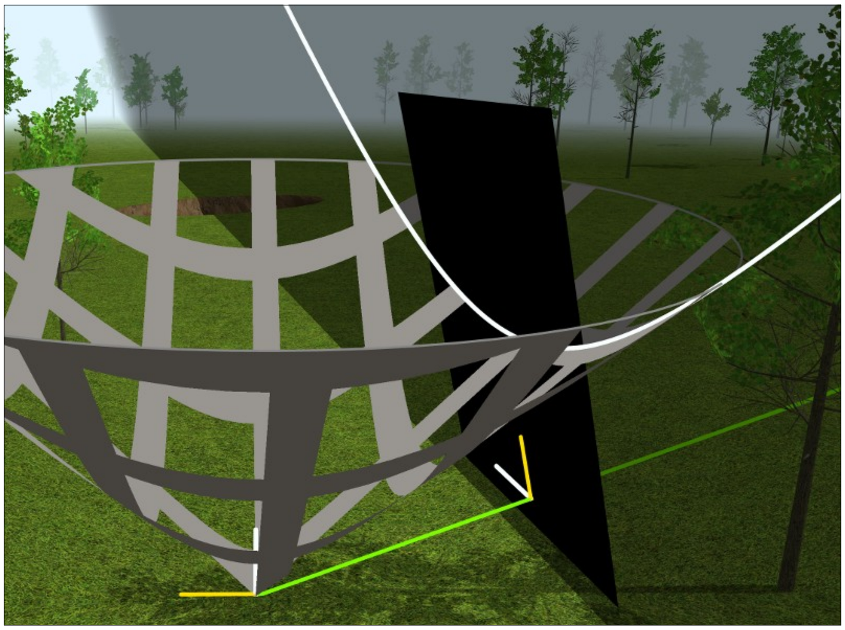
To construct a cone from the apex $\vec{A}$ and a point on conical surface $\vec{P}$, it requires to find parameter b:

$$\vec{V} = \vec{P} - \vec{A}; \quad b = \sqrt{(V_x^2 + V_z^2) \ / \ V_y^2} \ . \qquad (2)$$

Overview of the algorithm

- Created are 3 surfaces: 2 cones and a cylinder. Their axes are coincident with local axis Y. Parts of polyhedral faces inside the volume bounded by 3 coaxial shapes are determined as blocking the sight.
- Polyhedra in the neighboring area are processed. Each polyhedron is given as a base polyhedron with vertex coordinates in local space and transformation matrix which defines placement in 3D world. After a rejection test based on bounding sphere, per-face processing is performed.
- Per-face processing adds intervals to the list of obscured radial intervals (shown as red areas on the ground). Intervals being added may extend existing intervals or get absorbed.

Per-face processing



A coordinate system is developed for each planar face. See image to the right.

In the right part of the image, there's black face and corresponding semi-transparent plane.

Cone apex is the origin in cone's local coordinate system (x, y, z). To create a coordinate system on the plane it requires 3 orthogonal unit vectors (**u**, **v**, **n**) and the origin point.

- Planes are operated in Hessian normal form. **n** is set equal to plane normal (long green straight line in the image, it is directed to the right);
- Origin is located on the plane (n=0). Also it is set such that if x=0 and y=0 then u=0 and v=0;
- Where possible **u** is chosen such that it is coplanar to axis Y in cone's local space;
- **v** = **u** × **n**. This results in left-handed orthonormal base.

Next it creates a matrix for coordinate transformation of a point from (x, y, z) to (u, v, n), and inverse matrix (where n_c is the plane constant):

$$M = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & n_c \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad \overline{M} = \begin{bmatrix} u_x & v_x & n_x & -n_x n_c \\ u_y & v_y & n_y & -n_y n_c \\ u_z & v_z & n_z & -n_z n_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

While coordinate transformation of points, vectors is a usual operation in 3D computer graphics, doing it with a cone is coordinate transformation of a function.

First, equations for transformation of a point from (u, v, n) coordinates to (x, y, z) are written down:

$$\begin{cases} x = u_x u + v_x v + n_x n - n_x n_c \\ y = u_y u + v_y v + n_y n - n_y n_c \\ z = u_z u + v_z v + n_z n - n_z n_c \end{cases} \tag{4}$$

To obtain planar curve, n is set equal to zero.

Next, equations (4) are plugged into the equation of a cone (1):

$$(u_x u + v_x v - n_x n_c)^2 + (u_z u + v_z v - n_z n_c)^2 - b^2 (u_y u + v_y v - n_y n_c)^2 = 0. \tag{5}$$

After squaring terms in parentheses then grouping terms by powers of u, v the result is:

$$\begin{aligned} & u^2(u_x^2 + u_z^2 - b^2 u_y^2) + 2uv(u_x v_x + u_z v_z - b^2 u_y v_y) + v^2(v_x^2 + v_z^2 - b^2 v_y^2) \\ & + 2u(-u_x n_x n_c - u_z n_z n_c + b^2 u_y n_y n_c) + 2v(-v_x n_x n_c - v_z n_z n_c + b^2 v_y n_y n_c) \\ & + n_x^2 n_c^2 + n_z^2 n_c^2 - b^2 n_y^2 n_c^2 = 0. \end{aligned} \tag{6}$$

After substituting $x=u$, $y=v$ it results in the equation of order 2 algebraic planar curve:

$$\begin{aligned} & Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad \text{where} \\ & A = u_x^2 + u_z^2 - b^2 u_y^2, \quad B = 2(u_x v_x + u_z v_z - b^2 u_y v_y), \quad C = v_x^2 + v_z^2 - b^2 v_y^2, \\ & D = 2(-u_x n_x n_c - u_z n_z n_c + b^2 u_y n_y n_c), \quad E = 2(-v_x n_x n_c - v_z n_z n_c + b^2 v_y n_y n_c), \\ & F = n_x^2 n_c^2 + n_z^2 n_c^2 - b^2 n_y^2 n_c^2. \end{aligned} \tag{7}$$

These curves are often called conic sections. For a cylinder, derivation is similar.

Evasion of corner cases, curve classification

Subsequent computations require to know a kind of the curve: is it an ellipse, a hyperbola etc. It is possible to classify a curve by computing discriminant and determinant from its coefficients.

Author found it would be less expensive and more reliable to classify a curve before its coefficients (7) are computed, out of cone and plane properties, at the stage of evasion of corner cases. It is as follows:

1. If difference $b - n_y / \sqrt{n_x^2 + n_z^2}$ is less than zero then the resulting curve would be an ellipse, otherwise a hyperbola. If absolute value $\left| b - n_y / \sqrt{n_x^2 + n_z^2} \right|$ is below some threshold (e.g. $3 \cdot 10^{-5}$) then it would be a parabola or an ellipse of extreme axis length which would cause numeric imprecision or overflow. Parabola is treated as a degenerate curve in the sense that only one quadratic coefficient is present. In such case plane is rotated slightly.
2. If the origin of either one of two cones is on the plane then it would result in a pair of intersecting lines or a point. These cases are avoided by slightly moving the plane in the direction of plane normal.
3. When sectioning a cylinder, check the case where its axis is nearly parallel to the plane. This case is frequent in an application with enough vertical surfaces. It should result in a pair of parallel lines or imaginary parallel lines. Often vertical planes deviate from being strictly vertical because of roundoff appearing during placement and movement in 3D space. It results in ellipses of extreme axis lengths. In such cases plane is set strictly vertical.
4. When sectioning a cylinder, check if its axis is parallel to the plane. If plane constant is greater than cylinder radius ($|n_c| > r$) then there is no intersection cylinder plane. If $r - |n_c| < \varepsilon$ then resulting parallel lines are coincident or the distance between lines is very small. In such case plane is moved slightly in the direction of plane normal.

Center of quadratic curve

For ellipse, hyperbola and intersecting lines the following formula is used:

$$(x, y) = (2CD - BE,\ 2AE - BD) / (B^2 - 4AC). \tag{8}$$

For parallel lines which have zero discriminant $B^2 - 4AC$ this is inappropriate. In this work, parallel lines always appear symmetric around the origin so (0, 0) is used for the center.

Branches of hyperbolae

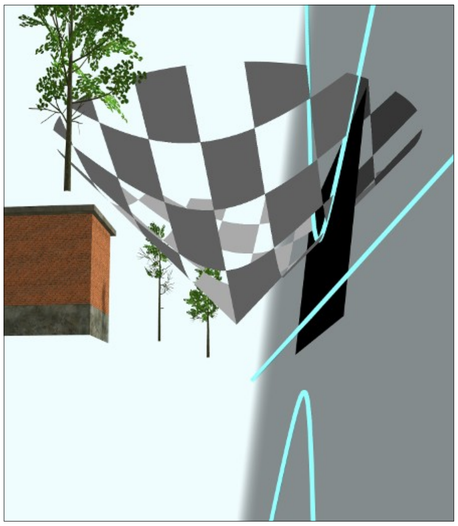
Hyperbola has 2 branches, it is likewise a cone of 2 halves. `halfSgn` property is included in curve object to select only one branch.

For the purpose to determine if a point belongs to the selected branch, it finds conjugate axis. It is stored in Hessian normal form which allows for least expensive computation to find if the point is to the left or to the right of the axis.

To find conjugate axis, first it creates a vector in cone’s local space directed towards the selected half: either (0, 1, 0) or (0, -1, 0).

Next it multiplies upper 3x3 submatrix (3) by the vector thus transforming vector direction to (u, v, n) coordinates, discards n, performs normalization.

Resulting 2D unit vector and the center of the curve (8) are used to define conjugate axis. Points in the selected branch have positive signed distance to the line.



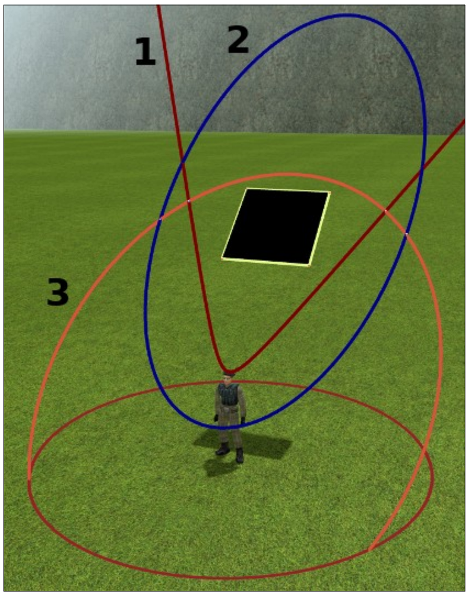
Two branches and conjugate axis of a hyperbola. A view from below the ground (-5m). Only upper half of the cone is shown.

Notion of a point being "inside" of a curve

For polygons in the plane even-odd rule is used. In the application polyhedral faces are never self-intersecting and the requirement for the rule is satisfied.

Hyperbola and parallel lines divide points in the plane into 3 sets.
For an arbitrary curve given by general eqn. (7) it's not obvious where it would be "inside". Author defines a point is inside if values of a function $Ax^2 + Bxy + Cy^2 + Dx + Ey + F$ at a point and at the center of the curve (8) have opposite signs in case of a hyperbola, or same signs in case of an ellipse or parallel lines.

Note: using the above definition, a point is inside a curve if and only if it is inside respective 3D object (a cone or a cylinder).



1 – hyperbola from upper cone; **2** – ellipse from lower cone; **3** – ellipse from a cylinder. All face vertices are inside of all curves.

Parameter along a curve

At every intersection it records parameter along a curve *t*. It serves the following purpose. Given e.g. 3 points on curve P_1, P_2, P_3 this is the method to know: if one moves along the curve from P_1 in given direction which point P_2 or P_3 it would encounter first. Curves composed of two disconnected sets such as hyperbolae and parallel lines are viewed as if they were connected at infinity.

For polygons, *t* at a point on edge is defined as integer edge number (increasing counterclockwise) plus parameter along an edge (where 0 is starting vertex and 1 is ending vertex).

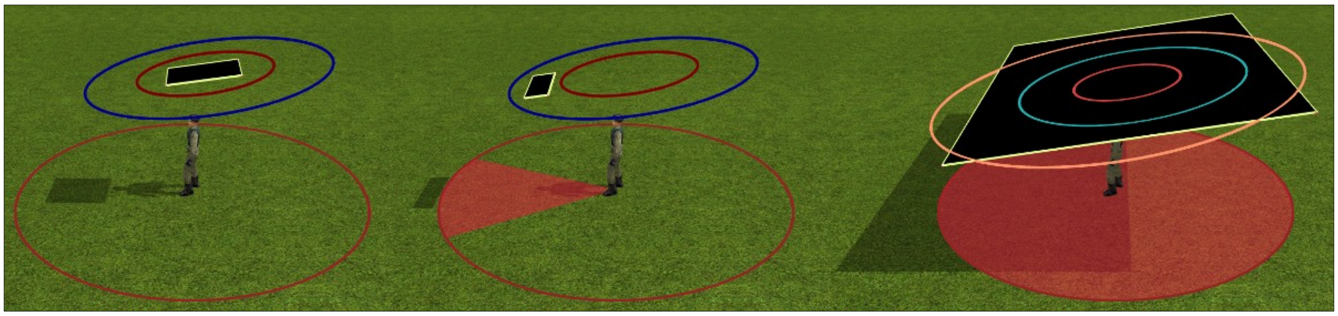
For order 2 algebraic curves, *t* at a point on curve is defined to be an angle between X+ and a ray from the center (8) through given point. It works for curves where any ray from the center intersects the curve in at most one point and for any two distinct points on curve, rays from the center through these points are also distinct. Curves appearing in this algorithm satisfy the above criterion, counterexample is intersecting lines.

Creating intersection lists

After curves were established on the plane containing the face, it creates intersection lists:

- For each polygon edge, intersect the edge and each of 3 curves. See Appendix A. Intersection line segment quadratic curve. If intersection appears outside of the volume bounded by 3 coaxial shapes then it's not recorded.
- Intersect the curve from a cylinder and each of 2 curves from cones. See Appendix B. Intersecting two quadratic curves.

Case of no intersections



In the right of the image, the curve created from a cylinder (outer ellipse) intersects edges. As intersections appear outside the volume being considered they are not recorded.

Any polygon vertex is selected. Then tests are performed in the following order:

1. The vertex is inside the curve from upper cone (in the left): no addition to obscured intervals;
2. The vertex is inside the curve from lower cone (in the center of the image);
3. The curve from lower cone is an ellipse and its center is inside the polygon (in the right of the image): it fully obscures the view. In all other cases there is no addition to obscured intervals.

Intersection types IN and OUT

When computing curve polygon edge intersection, curve curve intersection it is able to tell if it is type IN or type OUT intersection (thus making intersection functions non-commutative). Type IN means if intersecting `curve1` is viewed along the direction of parameter increase then points on curve are outside of `curve2` before the intersection point and inside of `curve2` after intersection point. “Touching” cases are computed as two intersections with same point coordinates and different types.

Processing intersection lists

It is able to process correctly non-convex faces and handle all possible cases.The result of processing is obscuring intervals added (shown on the ground).

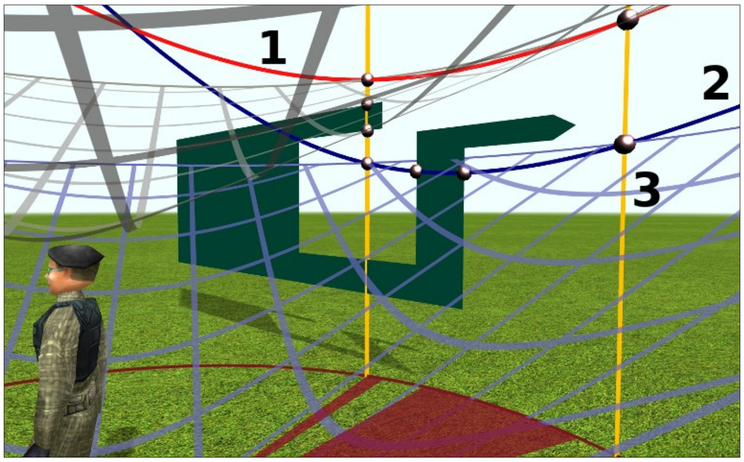
Intersections are sorted by parameter along the curve *t*.

First, it processes intersections of order 2 curves:

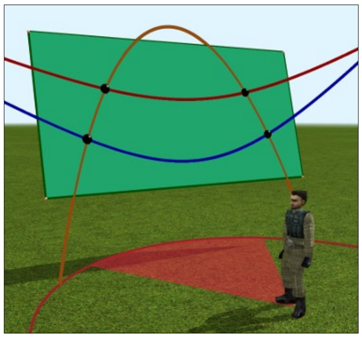
1. Traverse intersection list for given curve, skip intersections until it finds either type IN intersection with polygon edge, or type IN intersection with other curve where intersection point is inside polygon;
2. For this and next intersection, transform points on the plane into cone’s local space by adding 3rd vector component n=0 and multiplying inverse matrix (3) by the vector. Add angular interval to obscuring intervals. Continue with step 1.

Next, it processes edge intersections:

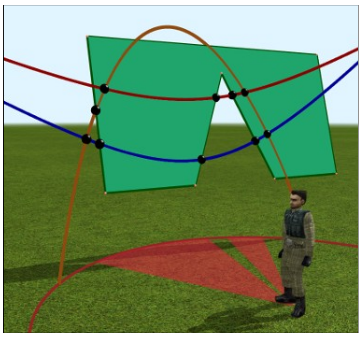
1. Traverse intersections, find one that enters into the volume bounded by 3 coaxial shapes;
2. Take intersection point, next intersection point, vertices in between (if any) and transform points into cone’s local space. Add interval to obscuring intervals.



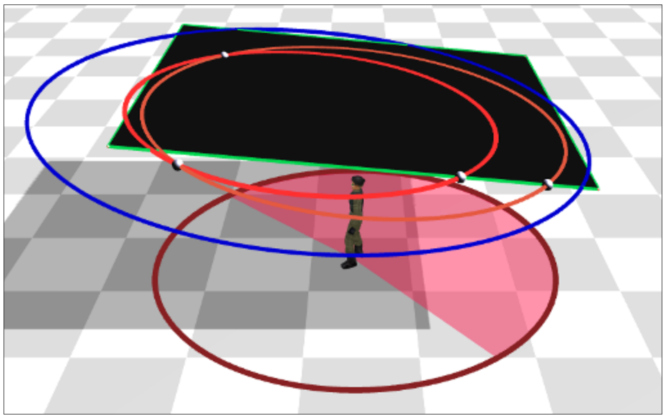
A view from in between cones. Infinite semi-transparent cones are cut with a cylinder. There is one face in the scene, it is non-convex, positioned vertically. **1** – hyperbola from upper cone, **2** – hyperbola from lower cone, **3** – parallel lines from a cylinder.



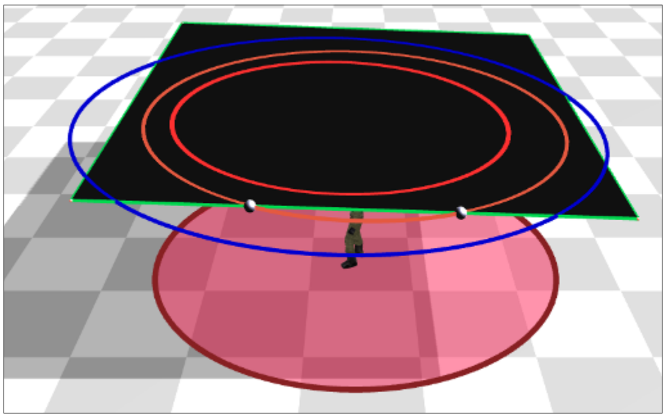
No edge intersections.



Various types of intersections.



Another case.



A case where single interval width exceeds π.

As a result obscuring intervals for the face (if any) were added.

Then it proceeds to the next face.

Appendix A. Intersection line segment quadratic curve

Line segment is given by its endpoints P_1 and P_2. It is directed from P_1 to P_2. It is put into parametric representation:

$$\vec{V} = P_2 - P_1\ ; \quad \begin{cases} x = P_{1x} + V_x t \\ y = P_{1y} + V_y t\ . \end{cases} \tag{9}$$

Quadratic curve is given by coefficients of the equation below.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0\ . \tag{10}$$

From eqn. (9) x, y are plugged into eqn. (10):

$$A(P_{1x}+V_x t)^2 + B(P_{1x}+V_x t)(P_{1y}+V_y t) + C(P_{1y}+V_y t)^2 + D(P_{1x}+V_x t) + E(P_{1y}+V_y t) + F = 0\ . \tag{11}$$

After multiplications and grouping terms by powers of t it results in algebraic equation of order 2:

$$\begin{aligned} &at^2 + bt + c = 0\ , \quad \text{where} \\ &a = AV_x^2 + BV_xV_y + CV_y^2\ , \\ &b = 2(AV_xP_{1x} + CV_yP_{1y}) + B(V_xP_{1y} + V_yP_{1x}) + DV_x + EV_y\ , \\ &c = AP_{1x}^2 + BP_{1x}P_{1y} + CP_{1y}^2 + DP_{1x} + EP_{1y} + F\ . \end{aligned} \tag{12}$$

If equation has roots t_1, t_2 they are ordered in ascending order to represent 1st (along the direction of line) and 2nd intersections. Coordinates of intersection points are:

$$x_n = P_{1x} + t_nV_x, \quad y_n = P_{1y} + t_nV_y, \quad n = 1,2\ . \tag{13}$$

If $t_n < 0 \vee t_n > 1$ then intersection is outside segment endpoints.

If a sequence of polygon edges is processed then an intersection exactly at vertex should not be recorded twice. Skipping $t_n = 1$ protects against such possibility.

Handle intersection with multiplicity 2 ($t_1 = t_2$)

Author distinguishes such intersections if points on curve before and after the intersection keep their “inside” property (see Notion of a point being “inside” of a curve). To check it, function (10) is evaluated at points $[x = P_{1x} + (t_1 - 1)V_x,\ y = P_{1y} + (t_1 - 1)V_y]$ and $[x = P_{1x} + (t_1 + 1)V_x,\ y = P_{1y} + (t_1 + 1)V_y]$.

If function values take same signs then “inside” property is preserved. Examples are: tangent intersection, the curve is a line with multiplicity 2 (e.g. $x^2 = 0$), the curve is a point. These are reported as 2 intersections with different IN and OUT types.

Example of an intersection where “inside” property is not preserved is where a line segment is parallel to hyperbola’s asymptote. It is handled differently, reported as 1 intersection.

Determine intersection type IN or OUT

If it is requested then it has to determine it (see Intersection types IN and OUT). For an ellipse, parallel lines the 1st intersection is always IN and the 2nd is OUT.

The method is to evaluate function (10) at point $[x = P_{1x} + (t_1 - 1)V_x,\ y = P_{1y} + (t_1 - 1)V_y]$ (before the 1st intersection), and to compare sign of the value and sign of function value in the center of the curve.

The 2nd intersection always has type other than the 1st one.

If the curve is a hyperbola and only one branch is relevant then branch check is performed (see Branches of hyperbolae).

Appendix B. Intersecting two quadratic curves

According to Bézout's theorem, two planar curves of degrees *n* and *m* have at most *nm* intersections. It follows two curves of degree 2 have at most 4 intersections.

Quadratic curves are given by coefficients of their equiations:

$$\begin{cases} A_1 x^2 + B_1 xy + C_1 y^2 + D_1 x + E_1 y + F_1 = 0 \\ A_2 x^2 + B_2 xy + C_2 y^2 + D_2 x + E_2 y + F_2 = 0 \, . \end{cases} \tag{14}$$

In this work, it attempts to perform transformation of coordinates the way such that in most cases, curve's axis appear coincident with axis X. It means B=0 and E=0. Intersecting such curves is much simpler task than arbitrary curves.

Case $B_{1,2}$=0 and $E_{1,2}$=0

Remove B, E terms from equations (14):

$$\begin{cases} A_1 x^2 + C_1 y^2 + D_1 x + F_1 = 0 \\ A_2 x^2 + C_2 y^2 + D_2 x + F_2 = 0 \, . \end{cases} \tag{15}$$

Combine equations (15) to eliminate y^2 term. To do it, multiply 1st equation by C_2 then subtract 2nd equation multiplied by C_1:

$$x^2 (A_1 C_2 - A_2 C_1) + x(D_1 C_2 - D_2 C_1) + F_1 C_2 - F_2 C_1 = 0 \, . \tag{16}$$

Solve resulting quadratic equation, obtain roots $x_{1,2}$. If there are no roots then there are no intersections.

For each $x_{1,2}$ it requires to find corresponding y-coordinates of points. Express any one of equations (15) as a function of y (the one having greater absolute value of C is preferable):

$$\begin{aligned} & Ax^2 + Dx + F = -Cy^2 \, ; \\ & y^2 = -(Ax^2 + Dx + F)/C \, ; \\ & y_{1,2} = \pm\sqrt{-(Ax^2 + Dx + F)/C} \, . \end{aligned} \tag{17}$$

If the radicand is non-negative then two points for given x exist and they are symmetric with respect to axis X.

If the curve is a hyperbola and only one branch is relevant then branch check is performed (see Branches of hyperbolae).

Determine intersection type IN or OUT

See Intersection types IN and OUT.

It requires "tangent vectors in the direction of increase of Parameter along a curve" at the intersection point.

For that, find gradient vectors of curves at the point:

$$\nabla f = \left(\frac{\partial x}{\partial f}, \frac{\partial y}{\partial f} \right) = (2Ax + By + D, \; 2Cy + Bx + E) \, . \tag{18}$$

Note for two coincident curves (e.g. $x^2 + y^2 - 1 = 0$ and $-x^2 - y^2 + 1 = 0$) directions of gradient vectors may be the same or opposite.

"Tangent vector in the direction of parameter increase" is perpendicular to gradient vector multiplied by scalar sign of function value at the center of the curve (8).

Next, it takes the sign of perp product of "tangent vectors in the direction of parameter increase" `curve1` and `curve2`. Type IN or OUT for both curves depends on the sign of perp product and types of curves.

General case

Implemented is the method as outlined in Wikipedia article “Conic Section”, in the section “Intersecting two conics“.

Find degenerate conics

A set of conics which pass through given 4 points (including imaginary ones) is a pencil of conics. It contains at least one degenerate conic: a pair of straight lines.

A pencil of conics is defined by a linear combination of 2 conics:

$$\begin{aligned}&\left(A_1 x^2 + B_1 xy + C_1 y^2 + D_1 x + E_1 y + F_1\right)\\ &+ \ \mu\left(A_2 x^2 + B_2 xy + C_2 y^2 + D_2 x + E_2 y + F_2\right).\end{aligned} \tag{19}$$

Below is matrix representation of conics and an expression for determinant. For a degenerate conic determinant equals zero.

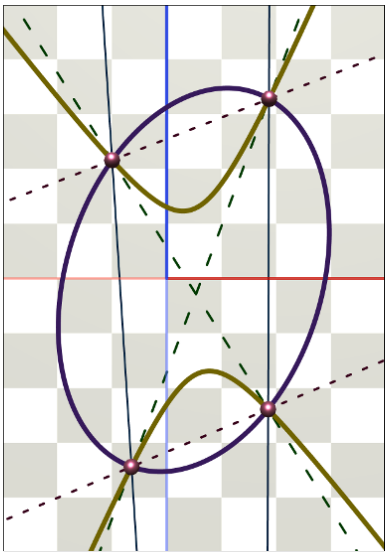
$$\begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix} = ACF - AE^2/4 - B^2F/4 + BDE/4 - D^2C/4\,. \tag{20}$$

To find degenerate conics, plug coefficients from (19) into (20) and set expression equal to zero. After algebraic manipulations, it results in degree 3 equation in variable μ:

$$a\mu^3 + b\mu^2 + c\mu + d = 0\,, \quad \text{where}$$

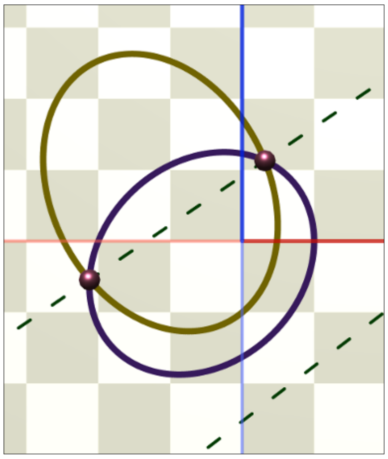
$$\begin{aligned}
a &= A_2C_2F_2 + \left(B_2D_2E_2 - E_2^2A_2 - B_2^2F_2 - D_2^2C_2\right)/4\,,\\
b &= A_1C_2F_2 + A_2C_1F_2 + A_2C_2F_1 \ +\\
&\quad + \left(B_1D_2E_2 + B_2D_1E_2 + B_2D_2E_1 - E_2^2A_1 - 2E_1E_2A_2\right)/4 \ -\\
&\quad - \left(B_2^2F_1 - 2B_1B_2F_2 - D_2^2C_1 - 2D_1D_2C_2\right)/4\,,\\
c &= A_1C_1F_2 + A_1C_2F_1 + A_2C_1F_1 \ +\\
&\quad + \left(B_1D_1E_2 + B_1D_2E_1 + B_2D_1E_1 - 2E_1E_2A_1 - E_1^2A_2\right)/4 \ -\\
&\quad - \left(2B_1B_2F_1 - B_1^2F_2 - 2D_1D_2C_1 - D_1^2C_2\right)/4\,,\\
d &= A_1C_1F_1 - A_1E_1^2/4 - B_1^2F_1/4 + B_1D_1E_1/4 - D_1^2C_1/4\,.
\end{aligned} \tag{21}$$

Solving (21) yields 3 roots, at least 1 of them is real. Substituting $\mu_{1,2,3}$ in (19) result in equations of degenerate conics.



Ellipse $2x^2$-$0.7xy$+y^2-$2x$+$0.4y$-11= =0 and -$3x^2$-xy+y^2+$3x$+y-3=0.

Roots: μ=1042.99 (lines left to right intersecting ~200 units away); μ=0.367228 (lines intersecting near origin); μ=1.25278 (lines top to bottom).



x^2-$0.4xy$+y^2+x+$0.4y$-2=0 and x^2+$0.4xy$+$0.7y^2$+$2x$-$0.5y$-1.2=0. One real root μ=-2.9345.

Decompose degenerate conic into 2 independent lines

Apply a sequence of transformations to remove terms B, D, E:

1. Rotate the curve to align its axes to coordinate axes removing B term. This is described in Wikipedia article “Rotation of axes in two dimensions”.
2. Compute discriminant $B^2 - 4AC$ to determine if it is parallel or intersecting lines.
3. For parallel lines, remove quadratic factor and offset along major axis. Quadratic factor should be close to 0 already. For horizontal lines $Cy^2 + F = 0$ set $A=0, D=0$, for vertical $C=0, E=0$.
4. For intersecting lines remove D and E. This is accomplished using “completing the square” technique.

The result takes the form $Ax^2 + Cy^2 = 0$ (opposite signs A and C) for intersecting lines or either $Ax^2 + F = 0$ or $Cy^2 + F = 0$ for parallel lines. Create 2 independent lines out of result equations.
The above transformations are recorded (using 2x3 matrix), their inverses are applied to created lines.

Intersect each of 2 lines with either one of two original curves (14)

See Appendix A. Intersection line segment quadratic curve. It is prone to numerical instability if it intersects at sharp angle. Solutions to intersect other curve or use line from other root are inapplicable in some cases (e.g. original curves are touching and only 1 root is available).