Fit a Cone to Ellipse and Points

David Eberly, Geometric Tools, Redmond WA 98052
https://www.geometrictools.com/

Created: February 21, 2022

Contents

1 Introduction

This document describes an algorithm for computing the equation of an infinite single-sided cone from an elliptical cross section of the cone and points not on the ellipse.

1.1 Definition of a Cone

An infinite single-sided cone has a vertex $\boldsymbol{K}$, an axis direction $\boldsymbol{D}$ that is unit length, and on acute cone angle $\theta \in (0, \pi/2)$. A point $\boldsymbol{X}$ is inside the cone when it is the vertex or when it is not the vertex and the angle between $\boldsymbol{D}$ and $\boldsymbol{X} - \boldsymbol{K}$ is θ. An algebraic equation for the cone is

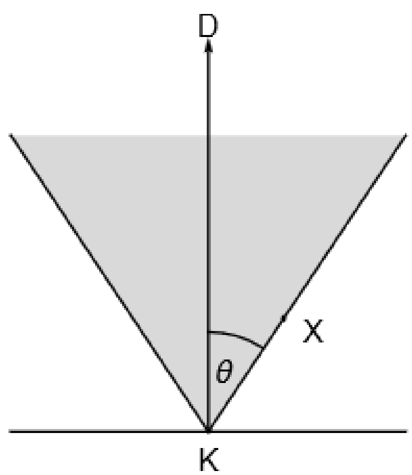
$$\boldsymbol{D} \cdot \frac{(\boldsymbol{X} - \boldsymbol{V})}{|\boldsymbol{X} - \boldsymbol{V}|} = \cos(\theta) \tag{1}$$

when $\boldsymbol{X} \neq \boldsymbol{K}$. Equivalently, the containment is defined by the quadratic equation

$$(\boldsymbol{X} - \boldsymbol{K})^{\mathsf{T}} \left(\boldsymbol{D}\boldsymbol{D}^{\mathsf{T}} - \left(\cos^2 \theta\right) I\right) (\boldsymbol{X} - \boldsymbol{K}) = 0 \tag{2}$$

where I is the identity matrix and $\boldsymbol{D} \cdot (\boldsymbol{X} - \boldsymbol{K}) \geq 0$. Figure 1 shows a 2D view of an infinite single-sided cone, which is sufficient to illustrate the quantities in 3D.

Figure 1. A 2D view of an infinite single-sided cone. The vertex is $\boldsymbol{K}$, the axis unit-length direction is $\boldsymbol{D}$, the angle is $\theta \in (0, \pi/2)$, and $\boldsymbol{X}$ is on the cone. The figure was generated using Mathematica [1].



In this document, I will refer to the infinite single-sided cone simply as the cone.

1.2 Definition of a 3D Ellipse

An ellipse in 3D is defined by

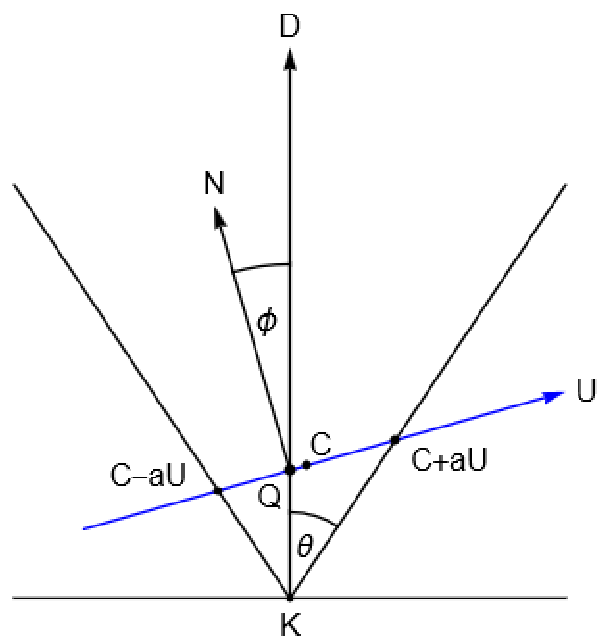
$$\boldsymbol{X} = \boldsymbol{C} + \mu \boldsymbol{U} + \nu \boldsymbol{V}, \quad (\mu/a)^2 + (\nu/b)^2 = 1 \tag{3}$$

where $\boldsymbol{C}$ is the center of the ellipse, $\boldsymbol{U}$ is the direction of the major axis with extent $a > 0$, $\boldsymbol{V}$ is the direction of the minor axis with extent $0 < b \leq a$ and $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{N}\}$ is a right-handed orthonormal basis. The plane containing the ellipse is $\boldsymbol{N} \cdot (\boldsymbol{X} - \boldsymbol{C}) = 0$. To be an ellipse, it is required that $(\mu/a)^2 + (\nu/b)^2 = 1$, in which case the ellipse has the parameterization $\mu(t) = a\cos(t)$ and $\nu(t) = b\sin(t)$ for $t \in [0, 2\pi)$.

2 Intersection of a Cone and a Plane

Consider a plane that intersects the cone in a bounded curve, but the intersection set is not the singleton $\boldsymbol{K}$. Let the plane have unit-length normal $\boldsymbol{N}$. The intersection is a circle when $|\boldsymbol{D} \cdot \boldsymbol{N}| = 1$ or is an ellipse when $|\boldsymbol{D} \cdot \boldsymbol{N}| < 1$. In both cases, I will refer to the curve as an ellipse with the understanding that a circle is a special case of an ellipse. Figure 2 shows a point of view of the intersection so that the observer sees the plane as a line.

Figure 2. The intersection of a cone and a plane. The cone is sliced by a plane containing the cone vertex and is spanned by the plane normal $\boldsymbol{N}$ and the ellipse major axis direction $\boldsymbol{U}$. The point $\boldsymbol{Q}$ is the intersection of the plane and the cone axis. The figure was generated using Mathematica [1].



A 2D ellipse is obtained by substituting equation (3) into equation (2). Define $M = \boldsymbol{D}\boldsymbol{D}^{\mathsf{T}} - (\cos^2\theta)I$ and

$\boldsymbol{\Delta} = \boldsymbol{C} - \boldsymbol{K}$; then

$$\begin{aligned} 0 &= (\mu\boldsymbol{U} + \nu\boldsymbol{V} + \boldsymbol{C} - \boldsymbol{K})^{\mathsf{T}} \left(\boldsymbol{D}\boldsymbol{D}^{\mathsf{T}} - \left(\cos^2\theta\right) I\right) (\mu\boldsymbol{U} + \nu\boldsymbol{V} + \boldsymbol{C} - \boldsymbol{K}) \\ &= (\mu\boldsymbol{U} + \nu\boldsymbol{V} + \boldsymbol{\Delta})^{\mathsf{T}} M \, (\mu\boldsymbol{U} + \nu\boldsymbol{V} + \boldsymbol{\Delta}) \\ &= \left[\begin{array}{cc} \mu & \nu \end{array}\right] \left[\begin{array}{cc} \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{U} & \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{V} \\ \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{U} & \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{V} \end{array}\right] \left[\begin{array}{c} \mu \\ \nu \end{array}\right] + 2 \left[\begin{array}{cc} \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{\Delta} & \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{\Delta} \end{array}\right] \left[\begin{array}{c} \mu \\ \nu \end{array}\right] + \boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta} \end{aligned} \tag{4}$$

Substituting $(\mu, \nu) = (\pm a, 0)$ and $(\mu, \nu) = (0, \pm b)$ into the ellipse equation produces the four equations

$$\begin{aligned} 0 &= a^2 \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{U} \pm 2a \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{\Delta} + \boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta} \\ 0 &= b^2 \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{V} \pm 2b \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{\Delta} + \boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta} \end{aligned} \tag{5}$$

Subtracting the two a-equations and subtracting the two b-equations leads to

$$\boldsymbol{U}^{\mathsf{T}} M \boldsymbol{\Delta} = 0, \;\; \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{\Delta} = 0 \tag{6}$$

The 2D ellipse equation can be factored into

$$1 = \frac{\mu^2}{a^2} + \frac{\nu^2}{b^2} = \frac{\mu^2}{(-\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta} / \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{U})} + \frac{\nu^2}{(-\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta} / \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{V})} \tag{7}$$

We need to compute the quadratic forms $\boldsymbol{U}^{\mathsf{T}} M \boldsymbol{U}$, $\boldsymbol{V}^{\mathsf{T}} M \boldsymbol{V}$ and $\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta}$ to further reduce the ellipse equation to something that allows us to extract cone parameters.

3 Extracting the Cone Parameters

In figure 2, the angle ϕ between $\boldsymbol{D}$ and $\boldsymbol{N}$ is shown to be acute and the angle $\pi/2 - \phi$ between $\boldsymbol{D}$ and $\boldsymbol{U}$ is shown to be acute. It must be that $\cos\phi \geq 0$ and $\sin\phi \geq 0$. If $\boldsymbol{N}$ were between $\boldsymbol{D}$ and $\boldsymbol{U}$, then $\pi/2 - \phi$ is obtuse in which case $\sin\phi < 0$. It is also possible that the angle between $\boldsymbol{D}$ and $\boldsymbol{N}$ is obtuse, in which case $\cos\phi < 0$. And in this case, $\sin\phi$ can be positive, negative or zero. In any event, we can represent

$$\boldsymbol{D} = (\cos\phi)\boldsymbol{N} + (\sin\phi)\boldsymbol{U} \tag{8}$$

In the configuration of figure 2, observe that if the plane is rotated about the axis $\boldsymbol{Q} + s\boldsymbol{V}$ so that $\boldsymbol{U}$ is parallel to a boundary ray of the cone, the plane-cone intersection is a parabola, which violates the assumption the plane and cone intersect in an ellipses. This places additional constraints on the angle ϕ.

Two of the quadratic forms can now be computed in terms of θ and ϕ,

$$\begin{aligned} \boldsymbol{U}^{\mathsf{T}} M \boldsymbol{U} &= (\boldsymbol{D} \cdot \boldsymbol{U})^{\mathsf{T}} - \cos^2\theta |\boldsymbol{U}|^2 = \sin^2\phi - \cos^2\theta \\ \boldsymbol{V}^{\mathsf{T}} M \boldsymbol{V} &= (\boldsymbol{D} \cdot \boldsymbol{V})^{\mathsf{T}} - \cos^2\theta |\boldsymbol{V}|^2 = -\cos^2\theta \end{aligned} \tag{9}$$

These forms are independent of the sign of $\cos\phi$. It remains to compute $\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta}$ in terms of θ and ϕ.

The intersection of the plane and the cone axis is $\boldsymbol{Q} = \boldsymbol{K} + h\boldsymbol{D}$ for some $h > 0$. The line $\boldsymbol{Q} + t\boldsymbol{U}$ intersects the cone boundary rays for two distinct t-values that are roots to a quadratic equation obtained by substituting $\boldsymbol{X} = \boldsymbol{Q} + t\boldsymbol{U}$ into equation (2),

$$\xi_2 t^2 + 2\xi_1 t + \xi_0 = (\sin^2\phi - \cos^2\theta)t^2 + (2h\sin\phi\sin^2\theta)t + h^2\sin^2\theta = 0 \tag{10}$$

The discriminant is

$$\xi_1^2 - \xi_0\xi_2 = h^2\sin^2\theta\cos^2\theta\cos^2\phi \geq 0 \tag{11}$$

Given a cone angle $\theta \in (0, \pi/2)$, the only possibilities for the discriminant to be zero is when $\cos\phi = \pm\pi/2$. In these cases, the line $\boldsymbol{Q} + t\boldsymbol{U}$ intersects only the cone vertex, but in this case the intersection of the plane and cone is not an ellipse. Therefore, it is guaranteed that the roots of the quadratic equation are real-valued and distinct.

The center point $\boldsymbol{C}$ of the ellipse occurs at the average of the two intersection points, which occurs at the t-value that is the average of the quadratic roots. This t-value is $-\xi_1/\xi_2$ from which it follows

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{Q} + \left(\tfrac{h\sin\phi\sin^2\theta}{\cos^2\theta - \sin^2\phi}\right)\boldsymbol{U} \\ &= \boldsymbol{K} + h\boldsymbol{D} + \left(\tfrac{h\sin\phi\sin^2\theta}{\cos^2\theta - \sin^2\phi}\right)\boldsymbol{U} \\ &= \boldsymbol{K} + h(\cos\phi)\boldsymbol{N} + h(\sin\phi)\left(1 + \tfrac{\sin^2\theta}{\cos^2\theta - \sin^2\phi}\right)\boldsymbol{U} \\ &= \boldsymbol{K} + h(\cos\phi)\boldsymbol{N} + h(\sin\phi)\left(\tfrac{\cos^2\phi}{\cos^2\theta - \sin^2\phi}\right)\boldsymbol{U} \end{aligned} \tag{12}$$

The equation makes intuitive sense. As h increases, the plane moves farther away from $\boldsymbol{K}$ and $\boldsymbol{C}$ must move farther from $\boldsymbol{Q}$. As ϕ becomes small, $\boldsymbol{N}$ rotates towards $\boldsymbol{D}$ and $\boldsymbol{C}$ must move closer to $\boldsymbol{Q}$. As the cone angle becomes larger, $\cos\theta$ becomes smaller and $\boldsymbol{C}$ must move farther away from $\boldsymbol{Q}$. The difference $\boldsymbol{\Delta} = \boldsymbol{C} - \boldsymbol{K}$ is

$$\boldsymbol{\Delta} = h\left((\cos\phi)\boldsymbol{N} + (\sin\phi)\left(\frac{\cos^2\phi}{\cos^2\theta - \sin^2\phi}\right)\boldsymbol{U}\right) \tag{13}$$

and we can compute the final quadratic form,

$$\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta} = (\boldsymbol{D}\cdot\boldsymbol{\Delta})^2 - \cos^2\theta|\boldsymbol{\Delta}|^2 = h^2\left(\frac{\cos^2\phi\cos^2\theta\sin^2\theta}{\cos^2\theta - \sin^2\phi}\right) \tag{14}$$

The last equality was verified using Mathematica [1]. In the equation for $\boldsymbol{\Delta}$, observe that the right-hand side is dependent on the signs of $\cos\phi$ and $\sin\phi$. However, the quadratic form $\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta}$ is independent of those signs.

The eccentricity $e \in [0, 1)$ of an ellipse with $a \geq b$ is defined to be $e^2 = 1 - (b/a)^2$. Using equations (7) and (9), we have

$$e^2 = 1 - \frac{b^2}{a^2} = 1 - \left(\frac{-\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta}/\boldsymbol{V}^{\mathsf{T}} M \boldsymbol{V}}{-\boldsymbol{\Delta}^{\mathsf{T}} M \boldsymbol{\Delta}/\boldsymbol{U}^{\mathsf{T}} M \boldsymbol{U}}\right) = 1 - \frac{\sin^2\phi - \cos^2\theta}{-\cos^2\theta} = \frac{\sin^2\phi}{\cos^2\theta} \tag{15}$$

This implies

$$\sin^2\phi = e^2\cos^2\theta, \;\; \cos^2\phi = 1 - e^2\cos^2\theta, \;\; \sin\phi = \sigma_0 e\cos\theta, \;\; \cos\phi = \sigma_1\sqrt{1 - e^2\cos^2\theta} \tag{16}$$

with $|\sigma_0| = 1$ and $|\sigma_1| = 1$. It is important to carry along the signs σ_0 and σ_1 in the computations. We can solve for

$$a = \frac{h\sin\theta\sqrt{1 - e^2\cos^2\theta}}{(1 - e^2)\cos\theta}, \;\; b = \frac{h\sin\theta\sqrt{1 - e^2\cos^2\theta}}{\sqrt{1 - e^2}\cos\theta} \tag{17}$$

The construction here shows that the 3D ellipse of intersection is not enough to determine the cone. The angle θ and the signs σ_0 and σ_1 are independent parameters. Given inputs $\boldsymbol{C}$, $\boldsymbol{N}$, $\boldsymbol{U}$, $\boldsymbol{V}$, a and b, compute $e = \sqrt{1-(b/a)^2}$. For a specified θ compute

$$\begin{aligned} \sin\phi &= \sigma_0 e \cos\theta \\ \cos\phi &= \sigma_1 \sqrt{1-e^2\cos^2\theta} \\ h &= a(1-e^2)\cos\theta/(\sin\theta\sqrt{1-e^2\cos^2\theta}) \\ \boldsymbol{D} &= (\cos\phi)\boldsymbol{N} + (\sin\phi)\boldsymbol{U} \\ \boldsymbol{Q} &= \boldsymbol{C} - \left(h\sin\phi\sin^2\theta/(\cos^2\theta - \sin^2\phi)\right)\boldsymbol{U} \\ \boldsymbol{K} &= \boldsymbol{Q} - h\boldsymbol{D} \end{aligned} \tag{18}$$

4 Determining the Cone Angle

Additional information is required to determine the cone angle θ. In particular, more points must be provided that are approximately on the purported cone.

The motivating problem for this document was to fit a cone to a collection of points for which subsets of the points are approximately on ellipses of intersection of planes and a cone. Let $\mathcal{E}$ be the subset of points that are approximately on an ellipse of intersection of a plane and a cone. Let $\mathcal{A}$ be additional points approximately on the same cone. The complete set of points are $\mathcal{P} = \mathcal{E} \cup \mathcal{A}$.

The points $\mathcal{E}$ are fit with a 3D ellipse using a least-squares algorithm. The output contains $\boldsymbol{C}$, $\boldsymbol{N}$, $\boldsymbol{U}$, $\boldsymbol{V}$, a and b as specified previously in the document. Another least-squares algorithm computes an error function that measures how close the points of $\mathcal{P}$ are to the cone. The input to the error function contains θ, σ_0 and σ_1 as specified previously in the document. A numerical minimizer is applied for each of the four pairs (σ_0, σ_1) and with $\theta \in (\varepsilon, \pi/2 - \varepsilon)$ for some small $\varepsilon \geq 0$. The epsilon value is used to avoid the endpoints of the interval where the error function is effectively zero (which produces false positives for the reported minima). The pair (σ_0, σ_1) that produces the minimum of the 4 minima is used to generate the fitted cone.

Pseudocode is provided in listing 1.

Listing 1. The function FitConeToEllipseAndPoints has two sets of inputs, the points $\mathcal{E}$ that are used to generate an ellipse and the points $\mathcal{A}$ the additional points (that might also be partitioned into subsets, each generating a different ellipse).

```
struct Ellipse3
{
    Vector3 C;      // ellipse center
    Vector3 N;      // unit-length normal to plane containing ellipse
    Vector3 U, V;   // unit-length directions that span the plane
    Real a, b;      // major-axis length and minor-axis length
};

struct Cone3
{
    Vector3 K;      // vertex
    Vector3 D;      // axis direction
    Real theta;     // angle
}
```

```
Cone3 ComputeCone(Real theta, Real sigma0, Real sigma1, Ellipse3 ellipse)
{
    Cone3 cone;
    cone.theta = theta;

    Real bDivA = ellipse.b / ellipse.a;
    Real e = sqrt(1 - bDivA * bDivA);
    Real snTheta = sin(theta);
    Real csTheta = cos(theta);
    Real snPhi = sigma0 * e * csTheta;
    Real csPhi = sigma1 * sqrt(one - snPhi * snPhi));
    cone.D D = csPhi * ellipse.N + snPhi * ellipse.U;

    Real h = ellipse.a * (1 - e * e) * csTheta / (snTheta * abs(csPhi));
    Vector3 Q = ellipse.C - ((h * snPhi * snTheta * snTheta) / (csTheta * csTheta - snPhi * snPhi)) * U;
    cone.K = Q - h * cone.D;

    return cone;
}

// These are global parameters that can be tuned for your datasets.
Real epsilon = 0.001;
Real penalty = 1;

Real Error(Real theta, Real sigma0, Real sigma1, Ellipse3 ellipse, Vector3 PPoints[numPoints])
{
    Cone3 cone = ComputeCone(theta, sigma0, sigma1, ellipse);
    Real cs = cos(cone.theta), csSqr = cs * cs;
    Real error = 0;
    foreach (p in PPoints)
    {
        Vector3 diff = p - cone.K;
        Real dot = Dot(cone.D, diff);
        if (dot >= 0)
        {
            Real sqrLength = Dot(diff, diff);
            Real coneValue = dot * dot - csSqr * sqrLength;
            error += coneValue * coneValue;
        }
        else
        {
            // Points not on the side of the plane containing the single-sided
            // cone are penalized in the error function. This can happen a lot
            // when the (sigma0,sigma1) pair is not the minimizer.
            error += penalty;
        }

        error = sqrt(error) / numPoints;
        return error;
    }
}

Cone3 FitConeToEllipseAndPoints(Vector3 EPoints[numEPoints], Vector3 APoints[numAPoints])
{
    Ellipse3 ellipse = FitEllipseToPoints(EPoints);

    Vector3 PPoints = Union(EPoints, APoints);
    Real sigma[4][2] = { { 1, 1 }, { 1, -1 }, { -1, 1}, { -1, -1 }};
    Real theta0 = epsilon, theta1 = PI/2 - epsilon, thetaMin, errorAtThetaMin;
    Real minError = -1;
    Cone3 cone;

    for (int i = 0; i < 4; ++i)
    {
        Minimize(Error, [theta, sigma[i][0], sigma[i][1], ellipse, PPoints],
            theta0, theta1, thetaMin, errorAtThetaMin);

        if (theta0 < thetaMin and thetaMin < theta1)
        {
            if (minError == -1 || errorAtThetaMin < minError)
            {
```

```
                minError = errorAtThetaMin;
                cone = ComputeCone(thetaMin, sigma[i][0], sigma[i][1], ellipse);
            }
        }
    }

    // ASSERT: minError != -1
    return cone;
}
```

References

[1] Wolfram Research, Inc. *Mathematica 13.0.0*. Wolfram Research, Inc., Champaign, Illinois, 2021.