Distance from Point to a General Quadratic Curve or a General Quadric Surface

David Eberly
Magic Software, Inc.
http://www.magic-software.com

Created: March 2, 1999

This document describes an algorithm for computing the distance from a point in 2D to a general quadratic curve defined implicitly by a second–degree quadratic equation in two variables or from a point in 3D to a general quadric surface defined implicitly by a second–degree quadratic equation in three variables.

1 The Problem

The general quadratic equation is

$$Q(\vec{x}) = \vec{x}^T A\vec{x} + \vec{b}^T\vec{x} + c = 0$$

where A is a symmetric $N \times N$ matrix ($N = 2$ or $N = 3$ not necessarily invertible, for example in the case of a cylinder or paraboloid), $\vec{b}$ is an $N \times 1$ vector, and c is a scalar. The parameter is $\vec{x}$, a $N \times 1$ vector. Given the surface $Q(\vec{x}) = 0$ and a point $\vec{y}$, find the distance from $\vec{y}$ to the surface and compute a closest point $\vec{x}$.

Geometrically, the closest point $\vec{x}$ on the surface to $\vec{y}$ must satisfy the condition that $\vec{y} - \vec{x}$ is normal to the surface. Since the surface gradient $\nabla Q(\vec{x})$ is normal to the surface, the algebraic condition for the closest point is

$$\vec{y} - \vec{x} = t\nabla Q(\vec{x}) = t(2A\vec{x} + \vec{b})$$

for some scalar t. Therefore,

$$\vec{x} = (I + 2tA)^{-1}(\vec{y} - t\vec{b})$$

where I is the identity matrix. One could replace this equation for $\vec{x}$ into the general quadratic equation to obtain a polynomial in t of at most sixth degree.

2 Reduction to a Polynomial Equation

Instead of immediately replacing $\vec{x}$ in the quadratic equation, we can reduce the problem to something simpler to code. Factor A using an eigendecomposition to obtain $A = RDR^T$ where R is an orthonormal matrix whose columns are eigenvectors of A and where D is a diagonal matrix whose diagonal entries are the eigenvalues of A. Then

$$\begin{aligned}\vec{x} &= (I + 2tA)^{-1}(\vec{y} - t\vec{b}) \\ &= (RR^T + 2tRDR^T)^{-1}(\vec{y} - t\vec{b}) \\ &= [R(I + 2tD)R^T]^{-1}(\vec{y} - t\vec{b}) \\ &= R(I + 2tD)^{-1}R^T(\vec{y} - t\vec{b}) \\ &= R(I + 2tD)^{-1}(\vec{\alpha} - t\vec{\beta})\end{aligned}$$

where the last equation defines $\vec{\alpha}$ and $\vec{\beta}$. Replacing in the quadratic equation and simplifying yields

$$0 = (\vec{\alpha} - t\vec{\beta})^T(I + 2tD)^{-1}D(I + 2tD)^{-1}(\vec{\alpha} - t\vec{\beta}) + \vec{\beta}^T(I + 2tD)^{-1}(\vec{\alpha} - t\vec{\beta}) + c.$$

The inverse diagonal matrix is

$$(I + 2tD)^{-1} = \mathrm{Diag}\{1/(1 + 2td_0), 1/(1 + 2td_1)\}.$$

for 2D or

$$(I + 2tD)^{-1} = \mathrm{Diag}\{1/(1 + 2td_0), 1/(1 + 2td_1), 1/(1 + 2td_2)\}.$$

for 3D. Multiplying through by $((1 + 2td_0)(1 + 2td_1))^2$ in 2D leads to a polynomial of at most fourth degree. Multiplying through by $((1 + 2td_0)(1 + 2td_1)(1 + 2td_2))^2$ in 3D leads to a polynomial equation of at most sixth degree.

The roots of the polynomial are computed and $\vec{x} = (I + 2tA)^{-1}(\vec{y} - t\vec{b})$ is computed for each root t. The distances between $\vec{x}$ and $\vec{y}$ are computed and the minimum distance is returned by the `PointQuadDistanceN` for $N = 2$ or $N = 3$.