Distance from Line to Rectangle in 3D

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

Created: June 5, 2018

Contents

1 Parameterizations of the Objects

The line is parameterized by $\mathbf{P} + t\mathbf{D}$ where $\mathbf{D}$ is a unit-length vector and $t \in \mathbb{R}$. The rectangle has four vertices $\mathbf{V}_i$ for $0 \leq i \leq 3$; they are ordered either clockwise or counterclockwise. Define $\mathbf{E}_0 = \mathbf{V}_1 - \mathbf{V}_0$ and $\mathbf{E}_1 = \mathbf{V}_3 - \mathbf{V}_0$ to be the directions of the edges emanating from the vertex $\mathbf{V}_0$. The edges for a rectangle must be perpendicular, so $\mathbf{E}_0 \cdot \mathbf{E}_1 = 0$. The remaining vertex is $\mathbf{V}_2 = \mathbf{V}_1 + \mathbf{E}_1 = \mathbf{V}_3 + \mathbf{E}_0$. The rectangle is parameterized by $\mathbf{V}_0 + u\mathbf{E}_0 + v\mathbf{E}_1$ where $0 \leq u \leq 1$ and $0 \leq v \leq 1$.

2 Distance Computed by Minimizing a Quadratic Function

Define $\mathbf{\Delta} = \mathbf{V}_0 - \mathbf{P}$. Half the squared distance between two points, one point on the rectangle and one point on the line, is the quadratic function

$$\begin{aligned} Q(u,v,t) &= \tfrac{1}{2}\left|(\mathbf{V}_0 + u\mathbf{E}_0 + v\mathbf{E}_1) - (\mathbf{P} + t\mathbf{D})\right|^2 \\ &= \tfrac{1}{2}\left|u\mathbf{E}_0 + v\mathbf{E}_1 - t\mathbf{D} + \mathbf{\Delta}\right|^2 \\ &= \tfrac{1}{2}\mathbf{x}^{\mathsf{T}} A\mathbf{x} + \mathbf{b}^{\mathsf{T}}\mathbf{x} + c \end{aligned} \tag{1}$$

where the last equality defines

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ t \end{bmatrix}, \quad A = \begin{bmatrix} \mathbf{E}_0 \cdot \mathbf{E}_0 & 0 & -\mathbf{E}_0 \cdot \mathbf{D} \\ 0 & \mathbf{E}_1 \cdot \mathbf{E}_1 & -\mathbf{E}_1 \cdot \mathbf{D} \\ -\mathbf{E}_0 \cdot \mathbf{D} & -\mathbf{E}_1 \cdot \mathbf{D} & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{E}_0 \cdot \mathbf{\Delta} \\ \mathbf{E}_1 \cdot \mathbf{\Delta} \\ -\mathbf{D} \cdot \mathbf{\Delta} \end{bmatrix}, \quad c = \frac{1}{2}|\mathbf{\Delta}|^2 \tag{2}$$

The vector $\mathbf{D}$ is unit length, so the lower-right entry of A is $\mathbf{D} \cdot \mathbf{D} = 1$. The two zero-valued entries of A correspond to $\mathbf{E}_0 \cdot \mathbf{E}_1 = 0$. The variables are constrained by $0 \leq u \leq 1$ and $0 \leq v \leq 1$. The t-variable is unconstrained. For the sake of indexing, let $A = [a_{ij}]$ and $\mathbf{b} = [b_j]$.

For a chosen $(u,v) \in [0,1]^2$, Q is a quadratic function for $t \in \mathbb{R}$. The minimum Q for the chosen (u,v) must occur when

$$0 = \frac{\partial Q}{\partial t} = a_{02}u + a_{12}v + t + b_2 \tag{3}$$

Solving for t, we have

$$t = -(a_{02}u + a_{12}v + b_2) \tag{4}$$

Substituting this into equation (1) leads to

$$\tilde{Q}(u,v) = Q(u,v,t(u,v)) = \frac{1}{2}\tilde{\mathbf{x}}^{\mathsf{T}}\tilde{A}\tilde{\mathbf{x}} + \tilde{\mathbf{b}}^{\mathsf{T}}\tilde{\mathbf{x}} + \tilde{c} \tag{5}$$

where the last equality defines

$$\tilde{\mathbf{x}} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} a_{00} - a_{02}^2 & -a_{02}a_{12} \\ -a_{02}a_{12} & a_{11} - a_{12}^2 \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} b_0 - a_{02}b_2 \\ b_1 - a_{12}b_2 \end{bmatrix}, \quad \tilde{c} = c - \frac{1}{2}b_2^2 \tag{6}$$

Equation (5) is a convex quadratic programming problem with constraints $\mathbf{0} \leq \tilde{\mathbf{x}} \leq \mathbf{1}$. For the sake of indexing, let $\tilde{A} = [\tilde{a}_{ij}]$ and $\tilde{\mathbf{b}} = [\tilde{b}_j]$.

Observe that

$$
\begin{aligned}
\tilde{a}_{00} &= a_{00} - a_{02}^2 = \mathbf{E}_0 \cdot \mathbf{E}_0 - (\mathbf{E}_0 \cdot \mathbf{D})^2 = |\mathbf{E}_0 \times \mathbf{D}|^2 \geq 0 \\
\tilde{a}_{11} &= a_{11} - a_{12}^2 = \mathbf{E}_1 \cdot \mathbf{E}_1 - (\mathbf{E}_1 \cdot \mathbf{D})^2 = |\mathbf{E}_1 \times \mathbf{D}|^2 \geq 0 \\
\tilde{a}_{01} &= -a_{02}a_{12} = -(-\mathbf{E}_0 \cdot \mathbf{D})(-\mathbf{E}_1 \cdot \mathbf{D}) = (\mathbf{E}_0 \times \mathbf{D}) \cdot (\mathbf{E}_1 \times \mathbf{D}) \\
\tilde{b}_0 &= b_0 - a_{02}b_2 = \mathbf{E}_0 \cdot \boldsymbol{\Delta} - (-\mathbf{E}_0 \cdot \mathbf{D})(-\boldsymbol{\Delta} \cdot \mathbf{D}) = (\mathbf{E}_0 \times \mathbf{D}) \cdot (\boldsymbol{\Delta} \times \mathbf{D}) \\
\tilde{b}_1 &= b_1 - a_{12}b_2 = \mathbf{E}_1 \cdot \boldsymbol{\Delta} - (-\mathbf{E}_1 \cdot \mathbf{D})(-\boldsymbol{\Delta} \cdot \mathbf{D}) = (\mathbf{E}_1 \times \mathbf{D}) \cdot (\boldsymbol{\Delta} \times \mathbf{D}) \\
\tilde{c} &= c - \tfrac{1}{2} b_2^2 = \tfrac{1}{2}(|\boldsymbol{\Delta}|^2 - (\boldsymbol{\Delta} \cdot \mathbf{D})^2) = \tfrac{1}{2} \left|\boldsymbol{\Delta} \times \mathbf{D}\right|^2
\end{aligned}
\tag{7}
$$

These can be verified using the identity

$$
(\mathbf{W}_0 \times \mathbf{W}_1) \cdot (\mathbf{W}_2 \times \mathbf{W}_3) = (\mathbf{W}_0 \cdot \mathbf{W}_2)(\mathbf{W}_1 \cdot \mathbf{W}_3) - (\mathbf{W}_0 \cdot \mathbf{W}_3)(\mathbf{W}_1 \cdot \mathbf{W}_2) \tag{8}
$$

and the known conditions $\mathbf{D} \cdot \mathbf{D} = 1$ and $\mathbf{E}_0 \cdot \mathbf{E}_1 = 0$.

Also observe that

$$
\begin{aligned}
\det(\tilde{A}) &= \tilde{a}_{00}\tilde{a}_{11} - \tilde{a}_{01}^2 \\
&= |\mathbf{E}_0 \times \mathbf{D}|^2 |\mathbf{E}_1 \times \mathbf{D}|^2 - ((\mathbf{E}_0 \times \mathbf{D}) \cdot (\mathbf{E}_1 \times \mathbf{D}))^2 \\
&= |(\mathbf{E}_0 \times \mathbf{D}) \times (\mathbf{E}_1 \times \mathbf{D})|^2 \\
&= |(\mathbf{D} \cdot \mathbf{E}_0 \times \mathbf{E}_1)\mathbf{D}|^2 \\
&= |\mathbf{D} \cdot \mathbf{E}_0 \times \mathbf{E}_1|^2
\end{aligned}
\tag{9}
$$

The third equality uses the identities

$$
(\mathbf{W}_0 \times \mathbf{W}_1) \times (\mathbf{W}_2 \times \mathbf{W}_3) = (\mathbf{W}_0 \times \mathbf{W}_1 \cdot \mathbf{W}_3)\mathbf{W}_2 - (\mathbf{W}_0 \times \mathbf{W}_1 \cdot \mathbf{W}_2)\mathbf{W}_3 \tag{10}
$$

and $\mathbf{W}_0 \cdot \mathbf{W}_1 \times \mathbf{W}_2 = -\mathbf{W}_1 \cdot \mathbf{W}_0 \times \mathbf{W}_2$. The matrix is invertible when $\det(\tilde{A}) > 0$ or not invertible when $\det(\tilde{A}) = 0$. The determinant cannot be negative. When $\det(\tilde{A}) > 0$, the line direction $\mathbf{D}$ has a nonzero component in the direction $\mathbf{E}_0 \times \mathbf{E}_1$, which implies the line and rectangle are not parallel. When $\det(\tilde{A}) = 0$, the line direction is a linear combination of $\mathbf{E}_0$ and $\mathbf{E}_1$, which implies the line and rectangle are parallel.

3 Equivalence to the 3D Segment-Segment Distance Query

If both $\mathbf{E}_0 \times \mathbf{D}$ and $\mathbf{E}_1 \times \mathbf{D}$ are not zero, the 3D distance query for line and rectangle is equivalent to the 3D distance query for two line segments. In particular, the segments can be chosen as $\boldsymbol{\Delta} \times \mathbf{D} + u\mathbf{E}_0 \times \mathbf{D}$ and $v(-\mathbf{E}_1 \times \mathbf{D})$. Algorithms for the segment-segment distance query are presented in Robust Computation of Distance Between Line Segments. A couple of algorithms are discussed that are not robust when using floating-point arithmetic. A robust algorithm is discussed that uses a constrained conjugate gradient minimization; this algorithm is reproduced here but using the notation introduced in this document for the 3D distance query of line and rectangle. In terms of the notation of the segment-segment document, the first line segment is $\mathbf{P}_0 + s(\mathbf{P}_1 - \mathbf{P}_0)$ and the second line segment is $\mathbf{Q}_0 + t(\mathbf{Q}_1 - \mathbf{Q}_0)$. The conversion to that formulation is as follows. The parameters (u, v) become the parameters (s, t). Set $\mathbf{P}_0 = \boldsymbol{\Delta} \times \mathbf{D}$ and $\mathbf{P}_1 = \mathbf{P}_0 + \mathbf{E}_0 \times \mathbf{D}$. Set $\mathbf{Q}_0 = \mathbf{0}$ and $\mathbf{Q}_1 = \mathbf{Q}_0 - \mathbf{E}_1 \times \mathbf{D}$.

4 Robust Algorithm for Computing the Distance

As mentioned in Robust Computation of Distance Between Line Segments, the primary failure of an approach that solves $\nabla\tilde{Q}(u,v) = (0,0)$ for (u,v) when using floating-point arithmetic has to do with subtractive cancellation when computing the 2×2 determinants of Cramer's Rule. Divisions of two determinants are required, and any subtractive cancellation in the numerator can be amplified greatly when the denominator is nearly zero. To avoid this problem we may use a conjugate gradient minimization, which is a 2-iteration algorithm for the problem at hand. However, this algorithm must be modified to deal with the constraints that $(u,v) \in [0,1]^2$.

4.1 A Brief Analysis of the Quadratic Function

The conjugate gradient minimization applies to a quadratic function $\tilde{Q}(u,v)$ defined for all $(u,v) \in \mathbb{R}^2$. Our function is specified in equation (5) with the various vectors and matrices defined in equation (6). For the sake of simpler notation, I will omit the tilde symbols over the variables; the quadratic function is $Q(\mathbf{x}) = \mathbf{x}^{\mathsf{T}}A\mathbf{x}/2 + \mathbf{b}^{\mathsf{T}}\mathbf{x} + c$. The minimum of Q occurs when its gradient is the zero vector, $\nabla Q(\mathbf{x}) = A\mathbf{x} + \mathbf{b} = \mathbf{0}$.

4.1.1 The Matrix A is Invertible

When A is invertible, there is a unique solution to the gradient equation and the minimum of Q is attained only at that point. The graph of Q is a paraboloid whose vertex occurs at the solution,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{x} = -A^{-1}\mathbf{b} = \frac{1}{a_{00}a_{11} - a_{01}^2} \begin{bmatrix} a_{01}b_1 - a_{11}b_0 \\ a_{01}b_0 - a_{00}b_1 \end{bmatrix} \tag{11}$$

If the determinant $a_{00}a_{11} - a_{01}^2$ is nearly zero, floating-point rounding errors can lead to subtractive cancellation. If either of the terms $a_{01}b_1 - a_{11}b_0$ or $a_{01}b_0 - a_{00}b_1$ is nearly zero, we can also have subtractive cancellation; that is, we can lose many significant bits of information. The division by the determinant can amplify this to produce grossly incorrect results. The document Robust Computation of Distance Between Line Segments contains an example of catastrophic failure.

4.1.2 The Matrix A is not Invertible

When A is not invertible, the graph of Q is a parabolic cylinder and the minimum of Q occurs at infinitely many points all lying on a line. We know this because $Q(u,v) \geq 0$ guarantees we have at least one global minimum, and the linearity of the gradient equations and the noninvertibility of A guarantees that we have infinitely many minimum points lying on a line. The gradient equations are $a_{00}u + a_{01}v + b_0 = 0$ and $a_{01}u + a_{11}v + b_1 = 0$. To have infinitely many solutions, we need the 3-tuples (a_{00}, a_{01}, b_0) and (a_{01}, a_{11}, b_1) to be linearly dependent; that is $\lambda_0(a_{00}, a_{01}, b_0) + \lambda_1(a_{01}, a_{11}, b_1)$ for some scalars λ_0 and λ_1 that are not both zero. This implies $a_{00}a_{11} - a_{01}^2 = 0$, $a_{00}b_1 - a_{01}b_0 = 0$ and $a_{01}b_1 - a_{11}b_0 = 0$. The quadratic function can be factored symbolically in two ways. Firstly, define $z = a_{00}u + a_{01}v + b_0$; then

$$Q(u,v) = f(z) = \frac{1}{2a_{00}}\,z^2 + c - \frac{b_0^2}{2a_{00}} \tag{12}$$

The minimum of $f(z)$ occurs when $z = 0$, so the (u, v)-line along which Q attains the minimum is $a_{00}u + a_{01}v + b_0 = 0$. Secondly, define $w = a_{01}u + a_{11}v + b_1$; then

$$Q(u, v) = g(w) = \frac{1}{2a_{11}} w^2 + c - \frac{b_1^2}{2a_{11}} \tag{13}$$

The minimum of $g(w)$ occurs when $w = 0$, so the (u, v)-line along which Q attains the minimum is $a_{01}u + a_{11}v + b_1 = 0$. Of course we need either $a_{00} \neq 0$ or $a_{11} \neq 0$ and then select the corresponding equation (12) or (13) to define the line of minimum points and the actual minimum. We know that $a_{00} = |\mathbf{E}_0 \times \mathbf{D}|^2 \geq 0$ and $a_{11} = |\mathbf{E}_1 \times \mathbf{D}|^2 \geq 0$. We also know that $\mathbf{D}$, $\mathbf{E}_0$ and $\mathbf{E}_1$ are coplanar and that $\mathbf{E}_0$ and $\mathbf{E}_1$ are perpendicular. Under these conditions, it is not possible for a_{00} and a_{11} to be simultaneously zero. Moreover, the maximum of a_{00} and a_{11} is bounded away from zero. This is useful to show that when floating-point arithmetic is used, we need not be concerned about the maximum being nearly zero. The proof is presented next.

The line direction $\mathbf{D}$ is in the plane of the edge directions $\mathbf{E}_0$ and $\mathbf{E}_1$, so $\mathbf{D} = k_0\mathbf{E}_0 + k_1\mathbf{E}_1$ for some scalars k_0 and k_1. The length of $\mathbf{D}$ is 1 and the edge vectors are perpendicular, so dotting the equation with itself leads to

$$1 = \mathbf{D} \cdot \mathbf{D} = (k_0\mathbf{E}_0 + k_1\mathbf{E}_1) \cdot (k_0\mathbf{E}_0 + k_1\mathbf{E}_1) = k_0^2|\mathbf{E}_0|^2 + k_1^2|\mathbf{E}_1|^2 \tag{14}$$

Applying cross products with the edge vectors,

$$\mathbf{E}_0 \times \mathbf{D} = k_1\mathbf{E}_0 \times \mathbf{E}_1, \;\; \mathbf{E}_1 \times \mathbf{D} = -k_0\mathbf{E}_0 \times \mathbf{E}_1 \tag{15}$$

which imply

$$a_{00} = |\mathbf{E}_0 \times \mathbf{D}|^2 = k_1^2|\mathbf{E}_0 \times \mathbf{E}_1|^2 = k_1^2|\mathbf{E}_0|^2|\mathbf{E}_1|^2, \;\; a_{11} = |\mathbf{E}_1 \times \mathbf{D}|^2 = k_0^2|\mathbf{E}_0 \times \mathbf{E}_1|^2 = k_0^2|\mathbf{E}_0|^2|\mathbf{E}_1|^2 \tag{16}$$

Solving these for k_0^2 and k_1^2 and substituting into equation (14) leads to

$$\frac{a_{00}}{|\mathbf{E}_0|^2} + \frac{a_{11}}{|\mathbf{E}_1|^2} = 1 \tag{17}$$

The equality of a_{00} and a_{11} in equation (17) leads to a minimum value for the maximum,

$$\max\{a_{00}, a_{11}\} \geq \frac{1}{\frac{1}{|\mathbf{E}_0|^2} + \frac{1}{|\mathbf{E}_1|^2}} = \frac{|\mathbf{E}_0|^2|\mathbf{E}_1|^2}{|\mathbf{E}_0|^2 + |\mathbf{E}_1|^2} \tag{18}$$

We still need to be concerned about the possibility that both a_{00} and a_{11} are nearly zero, but this can happen only when the edge lengths $|\mathbf{E}_0|$ and $|\mathbf{E}_1|$ are nearly zero, in which case nearly all the floating-point operations are of concern because we might not have enough precision near zero to handle such small numbers. The only way around this is to increase the precision of the floating-point numbers; for example, you might not have enough precision with 32-bit floating-point numbers but 64-bit floating-point numbers suffice. If 64-bit numbers are not precise enough, you will need either a package that supports more bits of precision or switch to a package that allows arbitrary precision arithmetic.

4.2 Conjugate Gradient Method

Given a 2-dimensional quadratic function $Q(\mathbf{x}) = \mathbf{x}^{\mathsf{T}}A\mathbf{x}/2 + \mathbf{b}^{\mathsf{T}}\mathbf{x} + c$ where A is positive semidefinite, the conjugate gradient minimization is a 2-iteration algorithm that locates the minimum point when A is invertible or the minimum line when A is not invertible.

Choose an initial guess $\mathbf{x}_0$ and an initial direction vector $\mathbf{d}_0$ for which $\mathbf{d}_0^\mathsf{T} A\mathbf{d}_0 > 0$. Define $f(r) = Q(\mathbf{x}_0 + r\mathbf{d}_0)$, which is a quadratic function of $r \in \mathbb{R}$. Compute the minimum of $f(r)$ by solving $0 = f'(r) = \mathbf{d}_0 \cdot \nabla Q(\mathbf{x}_0 + r\mathbf{d}_0) = \mathbf{d}_0^\mathsf{T} A(\mathbf{x}_0 + r\mathbf{d}_0)$ for r, namely, $r = -\mathbf{d}_0^\mathsf{T}(A\mathbf{x}_0 + \mathbf{b})/\mathbf{d}_0^\mathsf{T} A\mathbf{d}_0$. Set $\mathbf{x}_1 = \mathbf{x}_0 - (\mathbf{d}_0^\mathsf{T}(A\mathbf{x}_0 + \mathbf{b})/\mathbf{d}_0^\mathsf{T} A\mathbf{d}_0)\mathbf{d}_0$.

A nonzero direction $\mathbf{d}_1$ is *conjugate* to $\mathbf{d}_0$ when $\mathbf{d}_1^\mathsf{T} A\mathbf{d}_0 = 0$. The minimum of Q must occur on the line $\mathbf{x}_1 + s\mathbf{d}_1$. Define $g(s) = Q(\mathbf{x}_1 + s\mathbf{d}_1)$. When A is invertible, $g(s)$ is a quadratic function of $s \in \mathbb{R}$ and $\mathbf{d}_1^\mathsf{T} A\mathbf{d}_1 > 0$. Compute the minimum of $g(s)$ by solving $0 = g'(s) = \mathbf{d}_1 \cdot \nabla Q(\mathbf{x}_1 + s\mathbf{d}_1) = \mathbf{d}_1^\mathsf{T} A(\mathbf{x}_1 + s\mathbf{d}_1)$ for s, namely, $s = -\mathbf{d}_1^\mathsf{T}(A\mathbf{x}_1 + \mathbf{b})/\mathbf{d}_1^\mathsf{T} A\mathbf{d}_1$. Set $\mathbf{x}_2 = \mathbf{x}_1 - \mathbf{d}_1^\mathsf{T}(A\mathbf{x}_1 + \mathbf{b})/\mathbf{d}_1^\mathsf{T} A\mathbf{d}_1$. It must be that $\nabla Q(\mathbf{x}_2) = \mathbf{0}$ in which case $\mathbf{x}_2$ is the point at which Q is a minimum. When A is not invertible, it must be that $\mathbf{d}_1^\mathsf{T} A\mathbf{d}_1 = 0$ and $g(s)$ is identically constant; specifically, $g(s) = Q(\mathbf{x}_1 + s\mathbf{d}_1) = \mathbf{x}_1^\mathsf{T} A\mathbf{x}_1/2 + \mathbf{b}_1^\mathsf{T}\mathbf{x}_1 + c$. The line of minimum points is $\mathbf{x}_2(s) = \mathbf{x}_1 + s\mathbf{d}_1$ and $\nabla Q(\mathbf{x}_2(s)) = \mathbf{0}$ for all s.

4.3 Constrained Conjugate Gradient Algorithm

The conjugate gradient algorithm for minimizing $Q(u, v)$ over all $(u, v) \in \mathbb{R}^2$ must be modified to handle the constraints of the distance query, namely, $(u, v) \in [0, 1]$.

4.3.1 Case $a_{00} = \max\{a_{00}, a_{11}\} > 0$

Let us assume that $a_{00} \geq a_{11}$. A similar argument may be derived for the case when $a_{11} > a_{00}$ and is summarized in the next section.

Firstly, compute the minimum of $Q(u, v)$ along the line $v = 0$. The solution is $\hat{u}_0 = -b_0/a_{00}$. Secondly, compute the parameter point $(\hat{u}_1, 1)$ that minimizes $Q(u, v)$ along the line $v = 1$. The solution is $\hat{u}_1 = -(b_0 + a_{01})/a_{00}$. The computations of both roots are robust when using floating-point arithmetic. The difference of these points is

$$(\hat{u}_1, 1) - (\hat{u}_0, 0) = (-a_{01}/a_{00}, 1) \tag{19}$$

which happens to be a conjugate direction for $(1, 0)$. The minimum for $Q(u, v)$ for all $(u, v) \in \mathbb{R}^2$ must lie on the line through the two computed points. We may compute the intersection of this line with the parameter domain $[0, 1]^2$. If there is no intersection or the intersection is a single point, the minimum of Q occurs at a corner of the domain. If there is a segment of intersection, the minimum of Q must occur on that segment.

When there is a segment of intersection, let its endpoints be $\mathbf{e}_0$ and $\mathbf{e}_1$. For $w \in [0, 1]$, define

$$G(w) = Q((1 - w)\mathbf{e}_0 + w\mathbf{e}_1) \tag{20}$$

which is a quadratic function of a single variable w. Its derivative is

$$\begin{aligned} G'(w) &= (\mathbf{e}_1 - \mathbf{e}_0) \cdot \nabla Q((1 - w)\mathbf{e}_0 + w\mathbf{e}_1) \\ &= (1 - w)(\mathbf{e}_1 - \mathbf{e}_0) \cdot \nabla Q(\mathbf{e}_0) + w(\mathbf{e}_1 - \mathbf{e}_0) \cdot \nabla Q(\mathbf{e}_1) \\ &= (1 - w)G'(0) + wG'(1) \\ &= G'(0) + (G'(1) - G'(0))w \end{aligned} \tag{21}$$

If $G'(0) \leq 0$, the minimum of $G(w)$ on $[0, 1]$ occurs at $w = 0$. If $G'(1) \geq 0$, the minimum of $G(w)$ on $[0, 1]$ occurs at $w = 1$. Otherwise, $G'(0) > 0$ and $G'(1) < 0$ and the minimum of $G(w)$ on $[0, 1]$ occurs at $w = G'(0)/(G'(0) - G'(1))$. The approach is robust in the sense that if numerical errors cause the computed

w to be outside $(0, 1)$, clamp the result to the interval $[0, 1]$. Because we are in a 1-dimensional situation, the numerical errors cannot steer us away from the line segment that contains the minimum. This is better than in the 2-dimensional situation where the subtractive cancellation moves you away from the true minimum in an unknown direction.

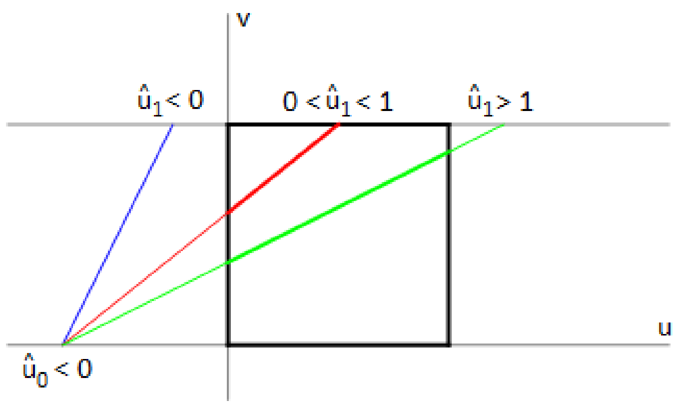
In each of the 9 cases mentioned next, I will make it clear on which line segment the minimum occurs, so the aforementioned framework applies to that segment. In these cases define the quantities

$$\tilde{v}_0 = -b_0/a_{01}, \quad \tilde{v}_1 = -(b_0 + a_{00})/a_{01} \tag{22}$$

which are related to the intersection points of conjugate lines with the edges $u = 0$ and $u = 1$ of the domain. The figures shown in the document were generated by [1] and manually edited with text labels.

Figure 1 illustrates the case when $\hat{u}_0 < 0$. The three colored line segments represent the cases for $\hat{u}_1$.

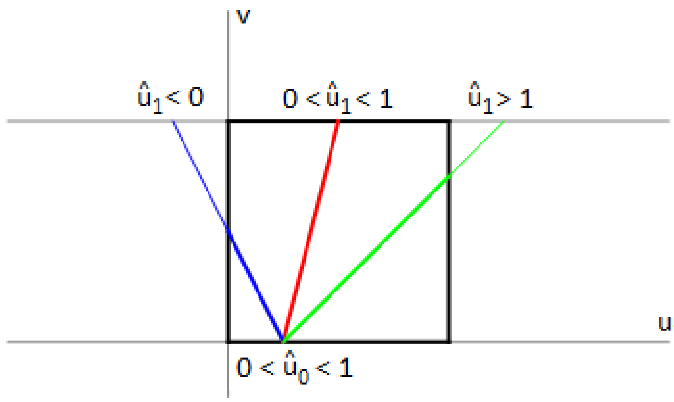
Figure 1. The case $\hat{u}_0 < 0$. The blue conjugate line corresponds to $\hat{u}_1 < 0$, the red conjugate line corresponds to $\hat{u}_1 \in (0, 1)$ and the green conjugate line corresponds to $\hat{u}_1 > 1$.



The blue line does not intersect the domain. Any point to the right of the line has the property that the directional derivative $(1, 0) \cdot \nabla Q(u, v) > 0$. This forces the minimum of $Q(u, v)$ on the domain to be on the edge $u = 0$ for some $v \in [0, 1]$. The endpoints are $\mathbf{e}_0 = (0, 0)$ and $\mathbf{e}_1 = (0, 1)$. The red line intersects the domain at $\mathbf{e}_0 = (0, \tilde{v}_0)$ and $\mathbf{e}_1 = (\hat{u}_1, 1)$. The green line intersects the domain at $\mathbf{e}_0 = (0, \tilde{v}_0)$ and $\mathbf{e}_1 = (1, \tilde{v}_1)$.

Figure 2 illustrates the case when $\hat{u}_0 \in (0, 1)$. The three colored line segments represent the cases for $\hat{u}_1$.

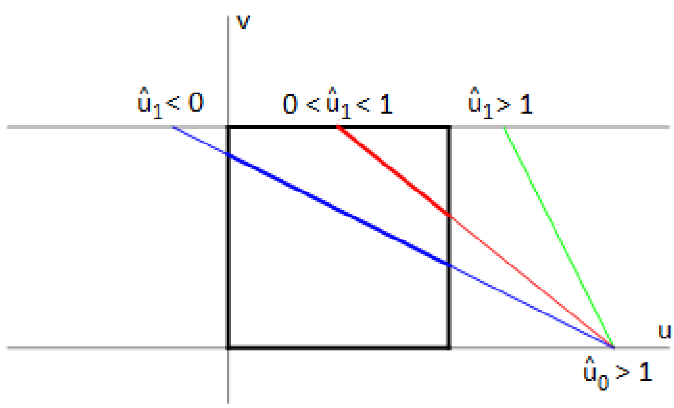
Figure 2. The case $\hat{u}_0 \in (0,1)$. The blue conjugate line corresponds to $\hat{u}_1 < 0$, the red conjugate line corresponds to $\hat{u}_1 \in (0,1)$ and the green conjugate line corresponds to $\hat{u}_1 > 1$.



The blue line intersects the domain at $\mathbf{e}_0 = (\hat{u}_0, 0)$ and $\mathbf{e}_1 = (0, \tilde{v}_0)$. The red line intersects the domain at $\mathbf{e}_0 = (\hat{u}_0, 0)$ and $\mathbf{e}_1 = (\hat{u}_1, 1)$. The green line intersects the domain at $\mathbf{e}_0 = (\hat{u}_0, 0)$ and $\mathbf{e}_1 = (1, \tilde{v}_1)$.

Figure 3 illustrates the case when $\hat{u}_0 > 1$. The three colored line segments represent the cases for $\hat{u}_1$.

Figure 3. The case $\hat{u}_0 > 1$. The blue conjugate line corresponds to $\hat{u}_1 < 0$, the red conjugate line corresponds to $\hat{u}_1 \in (0,1)$ and the green conjugate line corresponds to $\hat{u}_1 > 1$.



The blue line intersects the domain at $\mathbf{e}_0 = (0, \tilde{v}_0)$ and $\mathbf{e}_1 = (1, \tilde{v}_1)$. The red line intersects the domain at $\mathbf{e}_0 = (1, \tilde{v}_0)$ and $\mathbf{e}_1 = (\hat{u}_1, 1)$. The green line does not intersect the domain. Any point to the left of the line has the property that the directional derivative $(-1, 0) \cdot \nabla Q(u, v) > 0$. This forces the minimum of $Q(u, v)$ on the domain to be on the edge $u = 1$ for some $v \in [0, 1]$. The endpoints are $\mathbf{e}_0 = (1, 0)$ and $\mathbf{e}_1 = (1, 1)$.

The derivative $G'(w)$ shown in equation (21) can be factored symbolically as $G'(w) = \lambda h(w) = \lambda(h_0 + \sigma w)$, where $\lambda > 0$ and $\sigma > 0$. Rather than computing $G'(0)$ and $G'(1)$ and testing their signs, we can instead compute h_0 and σ and determine whether the quadratic minimum occurs at $w = 0$, $w = 1$ or $w = -h_0/\sigma$. Elimination of the λ factor leads to a reduced number of computations and to avoiding numerical issues related to the computation of $G'(0)$ and $G'(1)$. The cases are summarized in Table 1.

Table 1. The endpoints for the segments to be processed when searching for a minimum along a conjugate line. The endpoint components are $\hat{u}_0 = -b_0/a_{00}$, $\hat{u}_1 = -(a_{01}+b_0)/a_{00}$, $\tilde{v}_0 = -b_0/a_{01}$ and $\tilde{v}_1 = -(a_{00}+b_0)/a_{01}$. The derivatives use the quantities $s_0 = a_{00}+b_0$, $s_1 = a_{01}+b_0$, $s_2 = a_{00}+a_{01}+b_0$, $d_0 = a_{00}a_{11} - a_{01}^2$, $d_1 = a_{01}b_0 - a_{00}b_1$ and $d_2 = a_{01}b_1 - a_{11}b_0$. Recall that $G'(w) = \lambda[h_0 + \sigma w]$ where $\lambda > 0$ and $\sigma > 0$. The equations were generated using Mathematica [1].

case	$\mathbf{e}_0$	$\mathbf{e}_1$	$G'(w)$
$\hat{u}_0 \le 0,\ \hat{u}_1 \le 0$	$(0,0)$	$(0,1)$	$1[(b_1) + (a_{11})w]$
$\hat{u}_0 \le 0,\ \hat{u}_1 \in (0,1)$	$(0,\tilde{v}_0)$	$(\hat{u}_1,1)$	$\frac{-s_1}{a_{00}a_{01}^2}[(-a_{00}d_2) + (-s_1 d_0)w]$
$\hat{u}_0 \le 0,\ \hat{u}_1 \ge 1$	$(0,\tilde{v}_0)$	$(1,\tilde{v}_1)$	$\frac{a_{00}}{a_{01}^2}[(-d_2) + (d_0)w]$
$\hat{u}_0 \in (0,1),\ \hat{u}_1 \le 0$	$(\hat{u}_0,0)$	$(0,\tilde{v}_0)$	$\frac{-b_0}{a_{00}a_{01}^2}[(-a_{01}d_1) + (-b_0 d_0)w]$
$\hat{u}_0 \in (0,1),\ \hat{u}_1 \in (0,1)$	$(\hat{u}_0,0)$	$(\hat{u}_1,1)$	$\frac{1}{a_{00}}[(-d_1) + (d_0)w]$
$\hat{u}_0 \in (0,1),\ \hat{u}_1 \ge 1$	$(\hat{u}_0,0)$	$(1,\tilde{v}_1)$	$\frac{s_0}{a_{00}a_{01}^2}[(a_{01}d_1) + (s_0 d_0)w]$
$\hat{u}_0 \ge 1,\ \hat{u}_1 \le 0$	$(0,\tilde{v}_0)$	$(1,\tilde{v}_1)$	$\frac{a_{00}}{a_{01}^2}[(-d_2) + (d_0)w]$
$\hat{u}_0 \ge 1,\ \hat{u}_1 \in (0,1)$	$(\hat{u}_1,1)$	$(1,\tilde{v}_1)$	$\frac{s_2}{a_{00}a_{01}^2}[(-a_{01}(d_0 - d_1)) + (s_2 d_0)w]$
$\hat{u}_0 \ge 1,\ \hat{u}_1 \ge 1$	$(1,0)$	$(1,1)$	$1[(a_{01} + b_1) + (a_{11})w]$

The factorization with proper signs is based on the equivalence conditions

$$\left\{ \begin{array}{lcl} \hat{u}_0 \le 0 & \iff & b_0 \ge 0 \\ \hat{u}_0 \in (0,1) & \iff & b_0 < 0,\ s_0 > 0 \\ \hat{u}_0 \ge 1 & \iff & b_0 < 0,\ s_0 \le 0 \end{array} \right\},\ \left\{ \begin{array}{lcl} \hat{u}_1 \le 0 & \iff & s_1 \ge 0 \\ \hat{u}_1 \in (0,1) & \iff & s_1 < 0,\ s_2 > 0 \\ \hat{u}_1 \ge 1 & \iff & s_1 < 0,\ s_2 \le 0 \end{array} \right\} \tag{23}$$

4.3.2 Case $a_{11} = \max\{a_{00}, a_{11}\} > 0$

Now assume $a_{11} > a_{00}$. Compute the minimum of $Q(u,v)$ along the line $u = 0$, say, $\hat{v}_0 = -b_1/a_{11}$. Compute the minimum of $Q(u,v)$ along the line $u = 1$, say, $\hat{v}_1 = -(b_1 + a_{01})/a_{11}$. The difference of the points is

$$(1, \hat{v}_1) - (0, \hat{v}_0) = (1, -a_{01}/a_{11}) \tag{24}$$

which happens to be a conjugate direction for $(0,1)$. The minimum for $Q(u,v)$ for all $(u,v) \in \mathbb{R}^2$ must lie on the line through the two computed points. We may compute the intersection of this line with the parameter domain $[0,1]^2$. If there is no intersection or the intersection is a single point, the minimum of Q occurs at a corner of the domain. If there is a segment of intersection, the minimum of Q must occur on that segment. Defining $\mathbf{e}_0$, $\mathbf{e}_1$, $\phi(w)$ and $H(w)$ as before, the argument presented in the previous section applies to locating the minimum of $Q(u,v)$ along the line segment.

Define

$$\tilde{u}_0 = -b_1/a_{01}, \quad \tilde{u}_1 = -(b_1 + a_{11})/a_{01} \tag{25}$$

Table 2 lists the cases and their endpoints. As in the construction of Table 1, we need only construct $h(w)$ rather than $G'(w)$ for determining the location of the quadratic minimum.

Table 2. The endpoints for the segments to be processed when searching for a minimum along a conjugate line. The endpoint components are $\hat{v}_0 = -b_1/a_{11}$, $\hat{v}_1 = -(a_{01}+b_1)/a_{11}$, $\tilde{u}_0 = -b_1/a_{01}$ and $\tilde{u}_1 = -(a_{11}+b_1)/a_{01}$. The derivatives use the quantities $t_0 = a_{11}+b_1$, $t_1 = a_{01}+b_1$, $t_2 = a_{01}+a_{11}+b_1$, $d_0 = a_{00}a_{11} - a_{01}^2$, $d_1 = a_{01}b_0 - a_{00}b_1$ and $d_2 = a_{01}b_1 - a_{11}b_0$. The equations were generated using Mathematica [1].

case	$\mathbf{e}_0$	$\mathbf{e}_1$	$G'(w)$
$\hat{v}_0 \le 0,\ \hat{v}_1 \le 0$	$(0,0)$	$(1,0)$	$1[(b_0) + (a_{00})w]$
$\hat{v}_0 \le 0,\ \hat{v}_1 \in (0,1)$	$(\tilde{u}_0, 0)$	$(1, \hat{v}_1)$	$\frac{-t_1}{a_{11}a_{01}^2}[(-a_{11}d_1) + (-t_1 d_0)w]$
$\hat{v}_0 \le 0,\ \hat{v}_1 \ge 1$	$(\tilde{u}_0, 0)$	$(\tilde{u}_1, 1)$	$\frac{a_{11}}{a_{01}^2}[(-d_1) + (d_0)w]$
$\hat{v}_0 \in (0,1),\ \hat{v}_1 \le 0$	$(0, \hat{v}_0)$	$(\tilde{u}_0, 0)$	$\frac{-b_1}{a_{11}a_{01}^2}[(-a_{01}d_2) + (-b_1 d_0)w]$
$\hat{v}_0 \in (0,1),\ \hat{v}_1 \in (0,1)$	$(0, \hat{v}_0)$	$(1, \hat{v}_1)$	$\frac{1}{a_{11}}[(-d_2) + (d_0)w]$
$\hat{v}_0 \in (0,1),\ \hat{v}_1 \ge 1$	$(0, \hat{v}_0)$	$(\tilde{u}_1, 1)$	$\frac{t_0}{a_{11}a_{01}^2}[(a_{01}d_2) + (t_0 d_0)w]$
$\hat{v}_0 \ge 1,\ \hat{v}_1 \le 0$	$(\tilde{u}_0, 0)$	$(\tilde{u}_1, 1)$	$\frac{a_{11}}{a_{01}^2}[(-d_1) + (d_0)w]$
$\hat{v}_0 \ge 1,\ \hat{v}_1 \in (0,1)$	$(1, \hat{v}_1)$	$(\tilde{u}_1, 1)$	$\frac{t_2}{a_{11}a_{01}^2}[(-a_{01}(d_0 - d_2)) + (t_2 d_0)w]$
$\hat{v}_0 \ge 1,\ \hat{v}_1 \ge 1$	$(0,1)$	$(1,1)$	$1[(a_{01} + b_0) + (a_{00})w]$

The factorization with proper signs is based on the equivalence conditions

$$\left\{ \begin{array}{lcl} \hat{v}_0 \le 0 & \iff & b_1 \ge 0 \\ \hat{v}_0 \in (0,1) & \iff & b_1 < 0,\ t_0 > 0 \\ \hat{v}_0 \ge 1 & \iff & b_1 < 0,\ t_0 \le 0 \end{array} \right\}, \quad \left\{ \begin{array}{lcl} \hat{v}_1 \le 0 & \iff & t_1 \ge 0 \\ \hat{v}_1 \in (0,1) & \iff & t_1 < 0,\ t_2 > 0 \\ \hat{v}_1 \ge 1 & \iff & t_1 < 0,\ t_2 \le 0 \end{array} \right\} \tag{26}$$

Observe the symmetry between Tables 1 and 2, effectively a swap between variable names u and v. The second table is obtained from the first table by swapping a_{00} and a_{11} and by swapping b_0 and b_1. Using the definitions for d_i, s_i and t_i, this has the side effect of swapping d_1 and d_2 and replacing s_i with t_i for $0 \le i \le 2$. This observation is used for code sharing.

4.4 An Implementation

The top-level pseudocode is shown in Listing 1.

Listing 1. The query for computing the distance between a line and a rectangle in 3D, including the pair of closest points (when there is a unique solution) or a pair of closest points (when there are infinitely many solutions).

```
struct QueryResult<Real>
{
    Vector3<Real> lineClosest, rectClosest;
    Real t;  // line parameter
    Real u, v;  // rectangle parameters

    // The squared distance is computed to allow for the type Real to be an exact-arithmetic type.
    Real sqrDistance;
};

void DCPQuery(Line3<Real> line, Rectangle3<Real> rect, QueryResult<Real>& result)
{
    Vector3<Real> delta = rect.V0 - line.P;
    Vector3<Real> E0xE1 = Cross(rect.E0, rect.E1);
    Vector3<Real> E0xD = Cross(rect.E0, line.D);
    Vector3<Real> E1xD = Cross(rect.E1, line.D);
    Vector3<Real> E0xDelta = Cross(rect.E0, delta);
    Vector3<Real> E1xDelta = Cross(rect.E1, delta);
    Vector3<Real> DeltaxD = Cross(delta, line.D);
    Real DdE0xE1 = Dot(line.D, E0xE1);
    Real E0dD = Dot(rect.E0, line.D);
    Real E1dD = Dot(rect.E1, line.D);
    Real DeltadD = Dot(delta, line.D);

    Real a00 = Dot(E0xD, E0xD);
    Real a01 = Dot(E0xD, E1xD);
    Real a11 = Dot(E1xD, E1xD);
    Real b0 = Dot(E0xD, DeltaxD);
    Real b1 = Dot(E1xD, DeltaxD);
    Real d0 = DdE0xE1 * DdE0xE1;

    LocateMinimum(a00, a01, a11, b0, b1, d0, result.u, result.v);
    result.t = E0dD * result.u + E1dD * result.v + DeltadD;
    result.lineClosest = line.P + result.t * line.D;
    result.rectClosest = rect.V0 + result.u * rect.E0 + result.v * rect.E1;
    Vector3<Real> diff = result.lineClosest - result.rectClosest;
    result.sqrDistance = Dot(diff, diff);
}
```

The function LocateMinimum has an implementation based on the discussion of the previous sections of the document. It uses a function GetMinimumLocation to locate the $w \in [0,1]$ that minimizes the quadratic function on the interval. Based on the previous discussion, it is sufficient to use $h(w) = h_0 + \sigma w$ as a proxy for the derivative of the quadratic function. If $h(0) \geq 0$, the minimum is located at $w = 0$, else if $h(1) \leq 0$, the minimum is located at $w = 1$, else the minimum occurs when $h(w) = 0$ which produces $w = -h_0/\sigma$. Listing 2 contains pseudocode for locating the minimum.

Listing 2. Pseudocode for the minimization of $Q(u,v) = (a_{00}u^2 + 2a_{01}uv + a_{11}v^2)/2 + (b_0u + b_1v)$ for $(u,v) \in [0,1]$. The preconditions are that $a_{00} \geq 0$, $a_{11} \geq 0$, $\max\{a_{00}, a_{11}\} > 0$ and $d_0 = a_{00}a_{11} - a_{01}^2 \geq 0$.

```
Real GetMinimumLocation(Real h0, Real sigma)
{
    if (h0 >= 0) { return 0; } else if (h0 + sigma <= 0) { return 1; } else { return -h0 / sigma; }
```

```
}

void LocateMinimum(Real a00, Real a01, Real a11, Real d0, Real b0, Real b1, Real& u, Real& v)
{
    if (a00 >= a11)
    {
        LocateShared(a00, a01, a11, d0, b0, b1, u, v);
    }
    else
    {
        LocateShared(a11, a01, a00, d0, b1, b0, v, u);
    }
}

void LocateShared(Real a00, Real a01, Real a11, Real d0, Real b0, Real b1, Real& u, Real& v)
{
    Real d1 = a01 * b0 - a00 * b1, d2 = a01 * b1 - a11 * b0;
    Real s0 = a00 + b0, s1 = a01 + b0, s2 = a00 + s1;
    Real w, e0[2], e1[2];

    if (b0 >= 0)  // û0 ≤ 0
    {
        if (s1 >= 0)  // û1 ≤ 0
        {
            w = GetMinimumLocation(a11, b1);
            e0 = { 0, 0 };
            e1 = { 0, 1 };
        }
        else if (s2 > 0)  // û1 ∈ (0,1)
        {
            w = GetMinimumLocation(-a00 * d2, -s1 * d0);
            e0 = { 0, -b0 / a01 };
            e1 = { -s1 / a00, 1 };
        }
        else  // û1 ≥ 1
        {
            w = GetMinimumLocation(-d2, d0);
            e0 = { 0, -b0 / a01 };
            e1 = { 1, -s0 / a01 };
        }
    }
    else if (s0 > 0)  // û0 ∈ (0,1)
    {
        if (s1 >= 0)  // û1 ≤ 0
        {
            w = GetMinimumLocation(-a01 * d1, -b0 * d0);
            e0 = { -b0 / a00, 0 };
            e1 = { 0, -b0 / a01 };
        }
        else if (s2 > 0)  // û1 ∈ (0,1)
        {
            w = GetMinimumLocation(-d1, d0);
            e0 = { -b0 / a00, 0 };
            e1 = { -s1 / a00, 1 };
        }
        else  // û1 ≥ 1
        {
            w = GetMinimumLocation(a01 * d1, s0 * d0);
            e0 = { -b0 / a00, 0 };
            e1 = { 1, -s0 / a01 };
        }
    }
    else  // û0 ≥ 1
    {
        if (s1 >= 0)  // û1 ≤ 0
        {
            w = GetMinimumLocation(-d2, d0);
            e0 = { 0, -b0 / a01 };
            e1 = { 1, -s0 / a01 };
        }
        else if (s2 > 0)  // û1 ∈ (0,1)
        {
```

```
            w = GetMinimumLocation(-a01 * (d0 - d1), s2 * d0);
            e0 = { -s1 / a00, 1 };
            e1 = { 1, -s0 / a01 };
        }
        else  // û₁ ≥ 1
        {
            w = GetMinimumLocation(a01 + b1, a11);
            e0 = { 1, 0 };
            e1 = { 1, 1 };
        }
    }

    Real omw = 1 - w;
    u = omw * e0[0] + w * e0[1];
    v = omw * e1[0] + w * e1[1];
}
```

---

5 Distance from Ray or Segment to Rectangle in 3D

Computing the distance from a ray or a segment to a rectangle can be formulated in the same manner as distance from a line to a rectangle. An alternative approach uses the line-rectangle distance query and makes an appeal to *convexity* of the squared distance function when dealing with rays or segments.

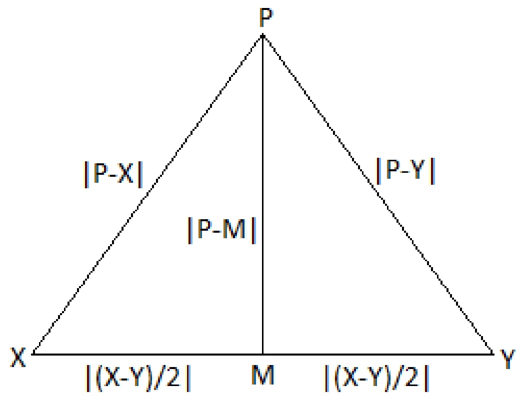
5.1 Distance between a Point and a Convex Object

A convex set S in $\mathbb{R}^n$ is a set of points such that if $\mathbf{X}$ and $\mathbf{Y}$ are in S, then the segment connecting them is also in the set. Let $\mathbf{P}$ be any point in $\mathbb{R}^n$. The distance from $\mathbf{P}$ to S is defined as

$$\text{distance}(\mathbf{P}, S) = \min_{\mathbf{X} \in S} |\mathbf{X} - \mathbf{P}| \tag{27}$$

If $\mathbf{P} \in S$, then the distance and 0 and $\mathbf{P}$ is the point of S closest to $\mathbf{P}$. If $\mathbf{P} \notin S$, we wish to compute also a point $\mathbf{X} \in S$ that is closest to $\mathbf{P}$. It turns out that the closest point is unique. The proof is by contradiction. Suppose that $\mathbf{X}$ and $\mathbf{Y}$ are two distinct points for which $|\mathbf{P} - \mathbf{X}| = |\mathbf{P} - \mathbf{Y}| = \text{distance}(\mathbf{P}, S)$. Because S is convex, the segment connecting $\mathbf{X}$ and $\mathbf{Y}$ is in S. In particular, the midpoint $\mathbf{M} = (\mathbf{X} + \mathbf{Y})/2$ is in S. Moreover, the segments $\langle \mathbf{X}, \mathbf{Y} \rangle$ and $\langle \mathbf{P}, \mathbf{M} \rangle$ are perpendicular. The hypotenuse of the triangle $\langle \mathbf{P}, \mathbf{X}, \mathbf{M} \rangle$ has length strictly larger than the length of either leg, so $|\mathbf{P} - \mathbf{X}| > |\mathbf{P} - \mathbf{M}|$ which contradicts $|\mathbf{P} - \mathbf{X}|$ being the minimum distance. Figure 4 illustrates the triangles involved.

Figure 4. Proof that there cannot be two distinct points $\mathbf{X}$ and $\mathbf{Y}$ in S that are closest to another point $\mathbf{P}$. The midpoint of the segment $\langle \mathbf{X}, \mathbf{Y} \rangle$ is $\mathbf{M} = (\mathbf{X} + \mathbf{Y})/2$.

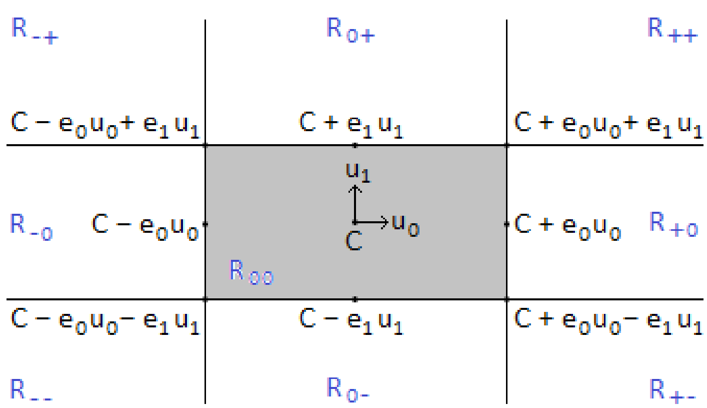


5.2 Distance from Point to Rectangle in 3D

Consider a 3D rectangle with center $\mathbf{C}$, unit-length axis directions $\mathbf{U}_0$ and $\mathbf{U}_1$ that are perpendicular, and extents e_0 and e_1. The points in the solid rectangle are $\mathbf{C} + y_0\mathbf{U}_0 + y_1\mathbf{U}_1$ for $|y_0| \leq e_0$ and $|y_1| \leq e_1$. The rectangle lies in the plane containing $\mathbf{C}$ with unit-length normal vector $\mathbf{U}_2 = \mathbf{U}_0 \times \mathbf{U}_1$. Any 3D point $\mathbf{P}$ can be written in the coordinate system whose origin is the rectangle center and whose direction vectors are the $\mathbf{U}_i$, namely, $\mathbf{P} = \mathbf{C} + x_0\mathbf{U}_0 + x_1\mathbf{U}_1 + x_2\mathbf{U}_2$.

The rectangle point closest to $\mathbf{P}$ is easily computed by observing that the rectangle and its plane normal $\mathbf{U}_2$ partition space into 9 regions, as shown in figure 5.

Figure 5. The rectangle and its plane normal partition space into 9 regions.



The 4 planes that contain the rectangle edges and that are perpendicular to the rectangle plane provided the partitioning. The regions are shown in 2D, but all points vertically above or below those 2D regions are included in the actual 3D region. The subscripts on the R-regions are the signs of the $e_i \mathbf{U}_i$ terms of the corners and edge midpoints marked in the figure. For example, R_{0+} is the region whose only finite-length edge has midpoint $\mathbf{C} + 0 * e_0 \mathbf{U}_0 + e_1 \mathbf{U}_1 = \mathbf{C} + e_1 \mathbf{U}_1$. R_{+-} is the region with no finite-length edges and whose corner is $\mathbf{C} + e_0 \mathbf{U}_0 - e_1 \mathbf{U}_1$.

The algebraic constraints for the regions are listed in Table 3.

Table 3. The regions of the partitioning. Each table entry contains the region name, the algebraic constraints defining the region, the rectangle point closest to $\mathbf{P} = \mathbf{C} + x_0 \mathbf{U}_0 + x_1 \mathbf{U}_1 + x_2 \mathbf{U}_2$ and the squared distance from the rectangle to $\mathbf{P}$.

$R_{-+},\ x_0 < -e_0,\ x_1 > e_1$ $\mathbf{C} - e_0 \mathbf{U}_0 + e_1 \mathbf{U}_1$ $(x_0 + e_0)^2 + (x_1 - e_1)^2 + x_2^2$	$R_{0+},\ \|x_0\| \le e_0,\ x_1 > e_1$ $\mathbf{C} + x_0 \mathbf{U}_0 + e_1 \mathbf{U}_1$ $(x_1 - e_1)^2 + x_2^2$	$R_{++},\ x_0 > e_0,\ x_1 > e_1$ $\mathbf{C} + e_0 \mathbf{U}_0 + e_1 \mathbf{U}_1$ $(x_0 - e_0)^2 + (x_1 - e_1)^2 + x_2^2$
$R_{-0},\ x_0 < -e_0,\ \|x_1\| \le e_1$ $\mathbf{C} - e_0 \mathbf{U}_0 + x_1 \mathbf{U}_1$ $(x_0 + e_0)^2 + x_2^2$	$R_{00},\ \|x_0\| \le e_0,\ \|x_1\| \le e_1$ $\mathbf{C} + x_0 \mathbf{U}_0 + x_1 \mathbf{U}_1$ x_2^2	$R_{+0},\ x_0 > e_0,\ \|x_1\| \le e_1$ $\mathbf{C} + e_0 \mathbf{U}_0 + x_1 \mathbf{U}_1$ $(x_0 - e_0)^2 + x_2^2$
$R_{--},\ x_0 < -e_0,\ x_1 < -e_1$ $\mathbf{C} - e_0 \mathbf{U}_0 - e_1 \mathbf{U}_1$ $(x_0 + e_0)^2 + (x_1 + e_1)^2 + x_2^2$	$R_{0-},\ \|x_0\| \le e_0,\ x_1 < -e_1$ $\mathbf{C} + x_0 \mathbf{U}_0 - e_1 \mathbf{U}_1$ $(x_1 + e_1)^2 + x_2^2$	$R_{+-},\ x_0 > e_0,\ x_1 < -e_1$ $\mathbf{C} + e_0 \mathbf{U}_0 - e_1 \mathbf{U}_1$ $(x_0 - e_0)^2 + (x_1 + e_1)^2 + x_2^2$

The squared distance function is piecewise quadratic. Within the interior of each region, the function is a single quadratic that is necessarily a continuous function. At a point shared by two or more regions, the quadratic function values agree and are equal to the squared distance. For example, a point $\mathbf{P}$ with $x_0 = -e_0$ and $x_1 > e_1$ is shared by the two regions R_{-+} and R_{0+}. The quadratic function for R_{-+} is $Q_{-+}(x_0, x_1, x_2) = (x_0 + e_0)^2 + (x_1 - e_1)^2 + x_2^2$ and the quadratic function for R_{0+} is $Q_{0+}(x_0, x_1, x_2) = (x_1 - e_1)^2 + x_2^2$. It is clear that $Q_{-+}(-e_0, x_1, x_2) = Q_{0+}(-e_0, x_1, x_2)$. Therefore, the squared distance function is continuous for all (x_0, x_1, x_2).

5.3 Line-Rectangle Squared Distance is Convex and C^1

The line is parameterized by $\mathbf{P}(t) = \mathbf{P}_0 + t\mathbf{D}$ for some nonzero direction vector $\mathbf{D}$ and for $t \in \mathbb{R}$. The distance from the line $\mathcal{L}$ to the rectangle $\mathcal{R}$ is the minimum distance for all the point-rectangle distances of the points on the line,

$$\text{distance}(\mathcal{L}, \mathcal{R}) = \min_{t \in \mathbb{R}} \text{distance}(\mathbf{P}(t), \mathcal{R}) \tag{28}$$

Define the squared distance between a line point and the rectangle to be $Q(t) = (\text{distance}(\mathbf{P}(t), \mathcal{R}))^2$. In terms of the line parameter t, let $\mathbf{P}(t) = \mathbf{C} + x_0(t)\mathbf{U}_0 + x_1(t)\mathbf{U}_1 + x_2(t)\mathbf{U}_2$ and define $\mathbf{\Delta} = \mathbf{P}_0 - \mathbf{C}$; then,

$$x_i(t) = \mathbf{U}_i \cdot (\mathbf{\Delta} + t\mathbf{D}) = a_i + b_i t \tag{29}$$

for $i = 0, 1, 2$ where the last equality defines the constants a_i and b_i. Replacing thes in the squared distance entries of Table 3, it is clear that $Q(t)$ is a piecewise quadratic function of t.

We can summarize the squared distance functions as follows. For $i = 0, 1$, define I_i to be 1 if $x_i < -e_i$ and to be 0 otherwise. Define σ_i to be 1 when $x_i < -e_i$, to be -1 when $x_i > e_i$ and to be 0 when $|x_i| \le e_i$. The quadratic function is

$$Q(t) = I_0[x_0(t) + \sigma_0 e_0]^2 + I_1[x_1(t) + \sigma_1 e_1]^2 + x_2(t)^2 \tag{30}$$

The function is continuous for all t. The only potential discontinuities are at t-values where $\mathbf{P}(t)$ is on a shared boundary between R-regions. As we saw previously, the function is indeed continuous at these points.

The first-order derivative of $Q(t)$ is

$$\begin{aligned} Q'(t) &= 2I_0[x_0(t) + \sigma_0 e_0]x_0'(t) + 2I_1[x_1(t) + \sigma_1 e_1]x_1'(t) + 2x_2(t)x_2'(t) \\ &= 2I_0[x_0(t) + \sigma_0 e_0]b_0 + 2I_1[x_1(t) + \sigma_1 e_1]b_1 + 2x_2(t)b_2 \end{aligned} \tag{31}$$

This function is continuous except potentially at t-values where $\mathbf{P}(t)$ is on a shared boundary between R-regions. The same argument applies as for $Q(t)$. For example, the derivative of the quadratic function for R_{-+} is $Q'_{-+}(t) = 2(x_0(t) + e_0)x_0'(t) + 2(x_1(t) - e_1)x_1'(t) + 2x_2(t)x_2'(t)$ and the derivative of the quadratic function for R_{0+} is $Q'_{0+}(t) = 2(x_1(t) - e_1)x_1'(t) + 2x_2(t)x_2'(t)$. Suppose the line intersects the plane $x_0 = -e_0$ at $t = T$ with $\mathbf{P}(t)$ on the boundary shared by R_{-+} and R_{0+}. Observe that $Q'_{-+}(T) = 2(x_1(T) - e_1)b_1 + 2x_2(T)b_2 = Q'_{0+}(T)$.

The second-order derivative of $Q(t)$ is

$$Q''(t) = 2I_0[x_0'(t)]^2 + 2I_1[x_1'(t)]^2 + 2[x_2'(t)]^2 = 2I_0 b_0^2 + 2I_1 b_1^2 + 2b_2^2 \ge 0 \tag{32}$$

This function is piecewise constant, but not all constants are the same, so the function is not continuous on the shared boundaries of the R-regions. For example, the second derivative of the quadratic function for R_{-+} is $Q''_{-+}(t) = b_0^2 + b_1^2 + b_2^2$ and the second derivative of the quadratic function for R_{0+} is $Q''_{0+}(t) = b_1^2 + b_2^2$, which are not the same when $b_0 \ne 0$.

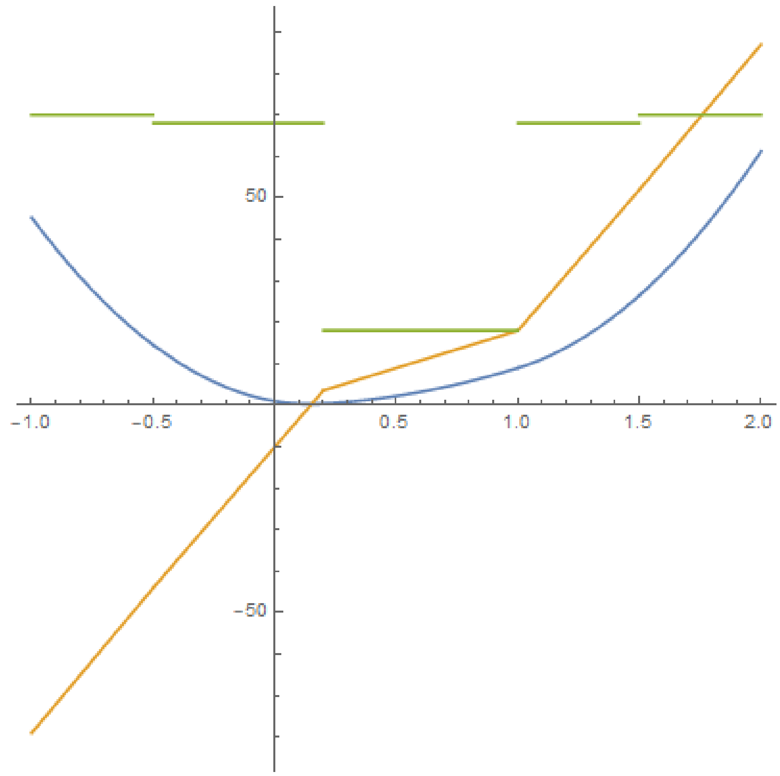
It is the case that $Q''(t) \ge 0$ for all t, so $Q(t)$ is a convex function. In some cases $Q(t)$ is strictly convex in that $Q''(t) > 0$. In some cases it is not strictly convex; for example, when the line is parallel to the rectangle, the minimum distance can be achieved along an entire line segment. When this happens, $Q''(t) = 0$ for all t-values that generate that segment.

As an example, let the rectangle have center $\mathbf{C} = (0, 0, 0)$, axes $\mathbf{U}_0 = (1, 0, 0)$ and $\mathbf{U}_1 = (0, 1, 0)$, and extents $e_0 = 2$ and $e_1 = 1$. Let the line have origin $\mathbf{P}_0 = (-3, -1/2, 0)$ and direction $\mathbf{D} = (5, 1, 3)$. The line intersects regions R_{--}, R_{-0}, R_{00}, R_{+0} and R_{++}, in that order as t increases. The quadratic pieces are

$$\begin{array}{lllllllll} Q_{--}(t) &=& 35t^2 - 9t + 5/4, & Q'_{--}(t) &=& 70t - 9, & Q''_{--}(t) &=& 70, & t \le -1/2 \\ Q_{-0}(t) &=& 34t^2 - 10t + 1, & Q'_{-0}(t) &=& 68t - 10, & Q''_{-0}(t) &=& 68, & -1/2 \le t \le 1/5 \\ Q_{00}(t) &=& 9t^2, & Q'_{00}(t) &=& 18t, & Q''_{00}(t) &=& 18, & 1/5 \le t \le 1 \\ Q_{+0}(t) &=& 34t^2 - 50t + 25, & Q'_{+0}(t) &=& 68t - 50, & Q''_{+0}(t) &=& 68, & 1 \le t \le 3/2 \\ Q_{++}(t) &=& 35t^2 - 53t + 109/4, & Q'_{++}(t) &=& 70t - 53, & Q''_{++}(t) &=& 70, & t > 3/2 \end{array} \tag{33}$$

The graphs of $Q(t)$, $Q'(t)$ and $Q''(t)$ are shown in figure 6.

Figure 6. The graphs of $Q(t)$, $Q'(t)$ and $Q''(t)$. $Q(t)$ is piecewise quadratic and continuous; its graph is drawn in blue. $Q'(t)$ is piecewise linear and continuous; its graph is drawn in yellow. $Q''(t)$ is piecewise constant but discontinuous; its graph is drawn in green.



The minimum of $Q(t)$ occurs at $T = 5/34 \doteq 0.147509$, the value where $Q'(T) = 0$. The minimum is $Q(T) = 9/34 \doteq 0.264706$. The closest point is on the rectangle edge $x_0 = -e_0$.

5.4 Ray-Rectangle and Segment-Rectangle Squared Distance

The line-rectangle squared-distance query is formulated as the minimum of a quadratic function $Q(t)$ for $t \in \mathbb{R}$, where $Q(t)$ is a convex function with continuous derivative $Q'(t)$. It is necessary that the minimum occurs at the value T for which $Q'(T) = 0$.

The ray is $\mathbf{P}(t) = \mathbf{P}_0 + t\mathbf{D}$ for $t \geq 0$. The ray-rectangle squared-distance query is also formulated as the minimum of $Q(t)$ but only for $t \geq 0$. If $T \geq 0$, then the minimum squared distance $Q(T)$ is the same for the line-rectangle query and the ray-rectangle query. If $T < 0$, the convexity of $Q(t)$ guarantees that the minimum squared distance for the ray-rectangle query is $Q(0)$.

The segment is $\mathbf{P}(t) = \mathbf{P}_0 + t\mathbf{D}$ for $t \in [0, 1$, where the segment endpoints are $\mathbf{P}_0$ and $\mathbf{P}_0 + \mathbf{D}$. The segment-rectangle squared-distance query is also formulated as the minimum of $Q(t)$ but only for $t \in [0, 1]$.

If $T \in [0, 1]$, then the minimum squared distance $Q(T)$ is the same for the line-rectangle query and the segment-rectangle query. If $T \notin [0, 1]$, the convexity of $Q(t)$ guarantees that the minimum squared distance for the segment-rectangle query is $Q(0)$ or $Q(1)$. If $T < 0$, the minimum is $Q(0)$. If $T > 1$, the minimum is $Q(1)$.

References

[1] Wolfram Research, Inc. *Mathematica 11.3*. Wolfram Research, Inc., Champaign, Illinois, 2018.