Intersection of Cylinders

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

Created: November 21, 2000
Last Modified: September 5, 2021

Contents

1 Introduction

This document shows how to determine whether two bounded cylinders intersect. The algorithm uses the method of separating axes. The search for a separating axis for cylinders is more complicated than that for convex polyhedra. The set of potential separating axes for polyhedra is finite; see Intersection of Convex Objects: The Method of Separating Axes. The set of potential separating axes for cylinders is infinite.

2 Representation of a Cylinder

A cylinder has a center point $\boldsymbol{C}$, unit-length axis direction $\boldsymbol{W}$, radius $r > 0$ and height $h > 0$. The enddisks of the cylinder are centered at $\boldsymbol{C} \pm (h/2)\boldsymbol{W}$. Let $\boldsymbol{U}$ and $\boldsymbol{V}$ be any unit-length vectors for which $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$ is a right-handed set of orthonormal vectors. That is, the vectors are unit length, mutually orthogonal and $\boldsymbol{W} = \boldsymbol{U} \times \boldsymbol{V}$. Points in the solid cylinder are parameterized by

$$\boldsymbol{X}(\theta, s, t) = \boldsymbol{C} + (s\cos\theta)\boldsymbol{U} + (s\sin\theta)\boldsymbol{V} + t\boldsymbol{W}, \;\; \theta \in [0, 2\pi), \; 0 \le s \le r, \; |t| \le h/2 \tag{1}$$

The cylinder wall occurs when $s = r$. The enddisks occur when $|t| = h/2$. The choice of $\boldsymbol{U}$ and $\boldsymbol{V}$ is arbitrary, but the inersection queries between cylinders are independent of this choice. A quadratic equation that represents the cylinder wall is $(\boldsymbol{X} - \boldsymbol{C})^{\mathsf{T}}(I - \boldsymbol{W}\boldsymbol{W}^{\mathsf{T}})(\boldsymbol{X} - \boldsymbol{C}) = r^2$ with boundedness specified by $|\boldsymbol{W} \cdot (\boldsymbol{X} - \boldsymbol{C})| \le h/2$. This representation is dependent only on $\boldsymbol{C}$, $\boldsymbol{W}$, r, and h.

3 Nonintersection of Convex Objects by Projection Methods

Consider the problem of determining whether two convex objects in 3D are intersecting. The *test-intersection* geometric query is concerned only about whether the objects intersect, not where they intersect. The latter problem is said to be a *find-intersection* geometric query. This document is about the test-intersection query for two bounded cylinders.

3.1 Separation by Projection onto a Line

A test for nonintersection of two convex objects is simply stated: If there exists a line for which the intervals of projection of the two objects onto that line do not intersect, then the objects do not intersect. Such a line is said to be a *separating line* or, more commonly, a *separating axis*. A direction vector for a separating axis is referred to as a *separating direction*. The translation of a separating axis is also a separating axis, so it suffices to consider lines that contain the origin. Given a line containing the origin and with unit-length direction $\boldsymbol{D}$, the projection of a bounded convex set C onto the line is the interval

$$I = [\lambda_{\min}(\boldsymbol{D}), \lambda_{\max}(\boldsymbol{D})] = [\min\{\boldsymbol{D} \cdot \boldsymbol{X} : \boldsymbol{X} \in C\}, \max\{\boldsymbol{D} \cdot \boldsymbol{X} : \boldsymbol{X} \in C\}] \tag{2}$$

Two compact convex sets C_0 and C_1 are separated if there exists a direction $\boldsymbol{D}$ such that their projection intervals $I_0 = [\lambda^{(0)}_{\min}(\boldsymbol{D}), \lambda^{(0)}_{\max}(\boldsymbol{D})]$ and $I_1 = [\lambda^{(1)}_{\min}(\boldsymbol{D}), \lambda^{(1)}_{\max}(\boldsymbol{D})]$ do not intersect. Specifically they do not intersect when

$$\lambda^{(0)}_{\min}(\boldsymbol{D}) > \lambda^{(1)}_{\max}(\boldsymbol{D}) \text{ or } \lambda^{(0)}_{\max}(\boldsymbol{D}) < \lambda^{(1)}_{\min}(\boldsymbol{D}). \tag{3}$$

The superscripts correspond to the indices of the convex set. Although the comparisons are made for unit-length $\boldsymbol{D}$, the comparisons are invariant to changes in length of the vector. This follows from $\lambda_{\min}(t\boldsymbol{D}) = t\lambda_{\min}(\boldsymbol{D})$ and $\lambda_{\max}(t\boldsymbol{D}) = t\lambda_{\max}(\boldsymbol{D})$ for $t > 0$. The Boolean value of the pair of comparisons is also invariant when $\boldsymbol{D}$ is replaced by the opposite direction $-\boldsymbol{D}$. This follows from $\lambda_{\min}(-\boldsymbol{D}) = -\lambda_{\max}(\boldsymbol{D})$ and $\lambda_{\max}(-\boldsymbol{D}) = -\lambda_{\min}(\boldsymbol{D})$. When $\boldsymbol{D}$ is not unit length, the intervals obtained for the separating axis tests are not the projections of the object onto the line; rather, they are scaled versions of the projection intervals.

In equation (3), allowing equality in any of the four comparisons means that the projection intervals just touch at a point; however, another direction can possibly separate objects. A search of the potential separating axes concludes only when one of those axes satisfies the four strict inequalities, in which case that axis is separating.

3.2 Projection of a Cylinder onto a Line

Let the line be $\lambda\boldsymbol{D}$ where $\boldsymbol{D}$ is a nonzero vector. The projections of cylinder points onto the line are

$$\lambda(\theta, t) = \boldsymbol{D}\cdot\boldsymbol{X}(\theta, t) = \boldsymbol{D}\cdot\boldsymbol{C} + (r\cos\theta)\boldsymbol{D}\cdot\boldsymbol{U} + (r\sin\theta)\boldsymbol{D}\cdot\boldsymbol{V} + t\boldsymbol{D}\cdot\boldsymbol{W} \tag{4}$$

for $\theta \in [0, 2\pi)$ and $|t| \leq h/2$. The interval of projection has endpoints determined by the extreme values of the projection equation. The maximum value occurs when all three terms involving the parameters are as large as possible. The t-term has a maximum of $(h/2)|\boldsymbol{D}\cdot\boldsymbol{W}|$. The θ-terms, not including the radius, can be viewed as a dot product $(\cos\theta, \sin\theta)\cdot(\boldsymbol{D}\cdot\boldsymbol{U}, \boldsymbol{D}\cdot\boldsymbol{V})$. This is maximized when $(\cos\theta, \sin\theta)$ is in the same direction as $(\boldsymbol{D}\cdot\boldsymbol{U}, \boldsymbol{D}\cdot\boldsymbol{V})$. Therefore,

$$(\cos\theta, \sin\theta) = \frac{(\boldsymbol{D}\cdot\boldsymbol{U}, \boldsymbol{D}\cdot\boldsymbol{V})}{\sqrt{(\boldsymbol{D}\cdot\boldsymbol{U})^2 + (\boldsymbol{D}\cdot\boldsymbol{V})^2}} \tag{5}$$

and

$$(\cos\theta)\boldsymbol{D}\cdot\boldsymbol{U} + (\sin\theta)\boldsymbol{D}\cdot\boldsymbol{V} = \sqrt{(\boldsymbol{D}\cdot\boldsymbol{U})^2 + (\boldsymbol{D}\cdot\boldsymbol{V})^2} = \sqrt{|\boldsymbol{D}|^2 - (\boldsymbol{D}\cdot\boldsymbol{W})^2} = |\boldsymbol{D}\times\boldsymbol{W}| \tag{6}$$

A similar argument applies to computing the minimum projection value. The minimum and maximum projection values are

$$\lambda_{\min}(\boldsymbol{D}) = \boldsymbol{D}\cdot\boldsymbol{C} - r|\boldsymbol{D}\times\boldsymbol{W}| - \frac{h}{2}|\boldsymbol{D}\cdot\boldsymbol{W}|, \;\; \lambda_{\max}(\boldsymbol{D}) = \boldsymbol{D}\cdot\boldsymbol{C} + r|\boldsymbol{D}\times\boldsymbol{W}| + \frac{h}{2}|\boldsymbol{D}\cdot\boldsymbol{W}| \tag{7}$$

4 Separating Axis Tests for Two Cylinders

Given two cylinders with centers $\boldsymbol{C}_i$, unit-length axis directions $\boldsymbol{W}_i$, radii r_i and heights h_i, for $i = 0, 1$, the cylinders are separated if there exists a nonzero direction $\boldsymbol{D}$ such that either

$$\lambda^{(0)}_{\min}(\boldsymbol{D}) = \boldsymbol{D}\cdot\boldsymbol{C}_0 - r_0|\boldsymbol{D}\times\boldsymbol{W}_0| - \frac{h_0}{2}|\boldsymbol{D}\cdot\boldsymbol{W}_0| > \boldsymbol{D}\cdot\boldsymbol{C}_1 + r_1|\boldsymbol{D}\times\boldsymbol{W}_1| + \frac{h_1}{2}|\boldsymbol{D}\cdot\boldsymbol{W}_1| = \lambda^{(1)}_{\max}(\boldsymbol{D}) \tag{8}$$

or

$$\lambda^{(0)}_{\max}(\boldsymbol{D}) = \boldsymbol{D}\cdot\boldsymbol{C}_0 + r_0|\boldsymbol{D}\times\boldsymbol{W}_0| + \frac{h_0}{2}|\boldsymbol{D}\cdot\boldsymbol{W}_0| < \boldsymbol{D}\cdot\boldsymbol{C}_1 - r_1|\boldsymbol{D}\times\boldsymbol{W}_1| - \frac{h_1}{2}|\boldsymbol{D}\cdot\boldsymbol{W}_1| = \lambda^{(1)}_{\min}(\boldsymbol{D}) \tag{9}$$

These are just a restatement of equation (3) for bounded cylinders. Defining $\boldsymbol{\Delta} = \boldsymbol{C}_1 - \boldsymbol{C}_0$, the tests of equations (8) and (9) can be combined into a single expression

$$f(\boldsymbol{D}) = r_0|\boldsymbol{D} \times \boldsymbol{W}_0| + r_1|\boldsymbol{D} \times \boldsymbol{W}_1| + \frac{h_0}{2}|\boldsymbol{D} \cdot \boldsymbol{W}_0| + \frac{h_1}{2}|\boldsymbol{D} \cdot \boldsymbol{W}_1| - |\boldsymbol{D} \cdot \boldsymbol{\Delta}| < 0 \tag{10}$$

If $\boldsymbol{\Delta} = 0$, then $f > 0$, which is geometrically obvious because two cylinders with the same center always intersect. The remainder of the discussion assumes $\boldsymbol{\Delta} \neq \boldsymbol{0}$. If $\boldsymbol{D}$ is perpendicular to $\boldsymbol{\Delta}$, then $f(\boldsymbol{D}) > 0$. Geometrically, any line perpendicular to the segment containing the cylinder centers can never be a separating axis. To see this, the segment connecting the cylinder centers is $\boldsymbol{C}_0 + s\boldsymbol{\Delta}$ for $s \in [0, 1]$. If you project the two cylinders onto the plane $\boldsymbol{\Delta} \cdot (\boldsymbol{X} - \boldsymbol{C}_0) = 0$, both regions of projection overlap. Any line in this plane that contains $\boldsymbol{C}_0$ must intersect both projection regions.

It is simple to compute $f(\boldsymbol{D})$ for various known vectors and exit early when one of the f-values is negative. These vectors include $\boldsymbol{\Delta}$, $\boldsymbol{W}_0$, $\boldsymbol{W}_1$ and $\boldsymbol{W}_0 \times \boldsymbol{W}_1$.

5 Separating Axis Tests for Parallel Cylinder Directions

When the cylinder unit-length axis directions are parallel and the cylinders are separated, they must be separated either in height (projection onto a line with direction $\boldsymbol{W}_0$) or radially (projection onto a plane perpendicular to $\boldsymbol{W}_0$). The cylinders are parallel when $\boldsymbol{W}_0 \times \boldsymbol{W}_1 = \boldsymbol{0}$, where $\boldsymbol{W}_1$ is equal to either $\boldsymbol{W}_0$ or $-\boldsymbol{W}_0$.

The separation test in height involves testing the sign of $f(\boldsymbol{W}_0)$,

$$\begin{aligned} f(\boldsymbol{W}_0) &= r_0|\boldsymbol{W}_0 \times \boldsymbol{W}_0| + r_1|\boldsymbol{W}_0 \times \boldsymbol{W}_1| + \tfrac{h_0}{2}|\boldsymbol{W}_0 \cdot \boldsymbol{W}_0| + \tfrac{h_1}{2}|\boldsymbol{W}_0 \cdot \boldsymbol{W}_1| - |\boldsymbol{W}_0 \cdot \boldsymbol{\Delta}| \\ &= \tfrac{h_0+h_1}{2} - |\boldsymbol{W}_0 \cdot \boldsymbol{\Delta}| \end{aligned} \tag{11}$$

The cylinders are separated in the height direction when $(h_0 + h_1)/2 < |\boldsymbol{W}_0 \cdot \boldsymbol{\Delta}|$. The projection of the first cylinder onto its axis with origin $\boldsymbol{C}_0$ is the interval $[-h_0/2, h_0/2]$. The projection of the second cylinder onto the same axis is the interval $[\boldsymbol{W}_0 \cdot \boldsymbol{\Delta} - h_1/2, \boldsymbol{W}_0 \cdot \boldsymbol{\Delta} + h_1/2]$. The intervals are separated when $\boldsymbol{W}_0 \cdot \boldsymbol{\Delta} - h_1/2 > h_0/2$ or $\boldsymbol{W}_0 \cdot \boldsymbol{\Delta} + h_1/2 < -h_0/2$. These combine to the single test $(h_0 + h_1)/2 < |\boldsymbol{W}_0 \cdot \boldsymbol{\Delta}|$, which is equivalent to $f(\boldsymbol{W}_0) < 0$.

The cylinders are separated in the radial direction when the distance between the cylinder axes is larger than the sum of the cylinder radii. The distance between axes is the length of the projection of $\boldsymbol{\Delta}$ onto a plane perpendicular to the cylinder axis. Specifically, the projection is $\boldsymbol{P} = \boldsymbol{\Delta} - (\boldsymbol{W}_0 \cdot \boldsymbol{\Delta})\boldsymbol{W}_0$ and the distance between cylinder axes is $|\boldsymbol{\Delta} - (\boldsymbol{W}_0 \cdot \boldsymbol{\Delta})\boldsymbol{W}_0|$. The cylinders are separated when $|\boldsymbol{\Delta} - (\boldsymbol{W}_0 \cdot \boldsymbol{\Delta})\boldsymbol{W}_0| > r_0 + r_1$. The separating test in radius involves testing the sign of $f(\boldsymbol{P})$,

$$\begin{aligned} f(\boldsymbol{P}) &= r_0|\boldsymbol{P} \times \boldsymbol{W}_0| + r_1|\boldsymbol{P} \times \boldsymbol{W}_1| + \tfrac{h_0}{2}|\boldsymbol{P} \cdot \boldsymbol{W}_0| + \tfrac{h_1}{2}|\boldsymbol{P} \cdot \boldsymbol{W}_1| - |\boldsymbol{P} \cdot \boldsymbol{\Delta}| \\ &= |\boldsymbol{\Delta} \times \boldsymbol{W}_0| \left(r_0 + r_1 - |\boldsymbol{\Delta} \times \boldsymbol{W}_0|\right) \end{aligned} \tag{12}$$

The cylinders are separated when $r_0 + r_1 < |\boldsymbol{\Delta} \times \boldsymbol{W}_0| = |\boldsymbol{\Delta} - (\boldsymbol{W}_0 \cdot \boldsymbol{\Delta})\boldsymbol{W}_0|$, which is what the geometric argument proved.

In summary, when $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ are parallel, the cylinders are separated when

$$\frac{h_0 + h_1}{2} < |\boldsymbol{\Delta} \cdot \boldsymbol{W}_0| \text{ or } r_0 + r_1 < |\boldsymbol{\Delta} \times \boldsymbol{W}_0| \tag{13}$$

6 Separating Axis Tests for Nonparallel Cylinder Directions

If $f(\boldsymbol{D}) < 0$ then $\boldsymbol{D}$ is a separating direction. As noted previously, $-\boldsymbol{D}$ is also a separating direction. It is sufficient to consider unit-length directions $\boldsymbol{D}$ on a hemisphere. Based on the analysis of the previous paragraph, select the hemisphere with north pole at $\boldsymbol{N} = \boldsymbol{\Delta}/|\boldsymbol{\Delta}|$. The hemisphere equatorial directions are $\boldsymbol{E}$ for which $\boldsymbol{N} \cdot \boldsymbol{E} = 0$; they satisfy $f(\boldsymbol{E}) > 0$. Let $\boldsymbol{U}$ and $\boldsymbol{V}$ be vectors for which $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{N}\}$ is a right-handed orthonormal basis.

Equivalently search for separating directions on the plane tangent to the north pole, $\boldsymbol{N} \cdot \boldsymbol{D} = 1$. In this document, the tangent plane is called the *pole plane*. The search in the pole plane is conceptualy simpler than the search over a hemisphere, avoiding both unit-length $\boldsymbol{D}$ and having to compute sine and cosine of angles. The search cannot reach the equatorial directions $\boldsymbol{E}$ because $f(\boldsymbol{E}) > 0$ for those directions. The equatorial directions map to infinity on the pole plane.

For cylinders with nonparallel axes, it must be that $\boldsymbol{W}_0 \times \boldsymbol{W}_1 \neq \boldsymbol{0}$. The pole plane has points

$$\boldsymbol{D} = x\boldsymbol{U} + y\boldsymbol{V} + \boldsymbol{N} \tag{14}$$

The cylinder axis directions are transformed to the same coordinate system,

$$\boldsymbol{W}_i = w_{i0}\boldsymbol{U} + w_{i1}\boldsymbol{V} + w_{i2}\boldsymbol{N} \tag{15}$$

for $i = 0, 1$. In the remainder of the document, the vectors are considered to be 3-tuples representations with respect to the $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{N}\}$ basis. That is, $\boldsymbol{D}(x, y) = (x, y, 1)$ and $\boldsymbol{W}_i = (w_{i0}, w_{i1}, w_{i2})$. As a function of (x, y), equation (10) is

$$f(x,y) = r_0|\boldsymbol{D}(x,y) \times \boldsymbol{W}_0| + r_1|\boldsymbol{D}(x,y) \times \boldsymbol{W}_1| + \frac{h_0}{2}|\boldsymbol{D}(x,y) \cdot \boldsymbol{W}_0| + \frac{h_1}{2}|\boldsymbol{D}(x,y) \cdot \boldsymbol{W}_1| - |\boldsymbol{\Delta}| \tag{16}$$

The function is continuous for all (x, y). The existence of a separating direction is equivalent to $f(x, y)$ having a negative minimum. The numerical algorithm for minimization involves an analysis of the first-order and second-order derivatives of f.

Let $\boldsymbol{E}_0 = (1, 0, 0)$, $\boldsymbol{E}_1 = (0, 1, 0)$ and $\boldsymbol{E}_2 = (0, 0, 1)$. The direction is $\boldsymbol{D} = x\boldsymbol{E}_0 + y\boldsymbol{E}_1 + \boldsymbol{E}_2$. The first-order partial derivatives of $\boldsymbol{D}$ are $\boldsymbol{D}_x = \boldsymbol{E}_0$ and $\boldsymbol{D}_y = \boldsymbol{E}_1$. The first-order partial derivatives of $f(x, y)$ are

$$\begin{aligned} f_x &= r_0 \tfrac{\mathbf{D}\times\mathbf{W}_0\cdot\mathbf{E}_0\times\mathbf{W}_0}{|\mathbf{D}\times\mathbf{W}_0|} + r_1 \tfrac{\mathbf{D}\times\mathbf{W}_1\cdot\mathbf{E}_0\times\mathbf{W}_1}{|\mathbf{D}\times\mathbf{W}_1|} + \tfrac{h_0}{2}(\mathbf{E}_0\cdot\mathbf{W}_0)\sigma(\mathbf{D}\cdot\mathbf{W}_0) + \tfrac{h_1}{2}(\mathbf{E}_0\cdot\mathbf{W}_1)\sigma(\mathbf{D}\cdot\mathbf{W}_1) \\ f_y &= r_0 \tfrac{\mathbf{D}\times\mathbf{W}_0\cdot\mathbf{E}_1\times\mathbf{W}_0}{|\mathbf{D}\times\mathbf{W}_0|} + r_1 \tfrac{\mathbf{D}\times\mathbf{W}_1\cdot\mathbf{E}_1\times\mathbf{W}_1}{|\mathbf{D}\times\mathbf{W}_1|} + \tfrac{h_0}{2}(\mathbf{E}_1\cdot\mathbf{W}_0)\sigma(\mathbf{D}\cdot\mathbf{W}_0) + \tfrac{h_1}{2}(\mathbf{E}_1\cdot\mathbf{W}_1)\sigma(\mathbf{D}\cdot\mathbf{W}_1) \end{aligned} \tag{17}$$

where the sign function is

$$\sigma(z) = \begin{cases} +1, & z > 0 \\ -1 & z < 0 \\ \text{undefined}, & z = 0 \end{cases} \tag{18}$$

The second-order derivatives are

$$\begin{aligned} f_{xx} &= r_0 \tfrac{(\mathbf{D}\times\mathbf{W}_0\cdot\mathbf{E}_0)^2}{|\mathbf{D}\times\mathbf{W}_0|^3} + r_1 \tfrac{(\mathbf{D}\times\mathbf{W}_1\cdot\mathbf{E}_0)^2}{|\mathbf{D}\times\mathbf{W}_1|^3} \\ f_{yy} &= r_0 \tfrac{(\mathbf{D}\times\mathbf{W}_0\cdot\mathbf{E}_1)^2}{|\mathbf{D}\times\mathbf{W}_0|^3} + r_1 \tfrac{(\mathbf{D}\times\mathbf{W}_1\cdot\mathbf{E}_1)^2}{|\mathbf{D}\times\mathbf{W}_1|^3} \\ f_{xy} &= r_0 \tfrac{(\mathbf{D}\times\mathbf{W}_0\cdot\mathbf{E}_0)(\mathbf{D}\times\mathbf{W}_0\cdot\mathbf{E}_1)}{|\mathbf{D}\times\mathbf{W}_0|^3} + r_1 \tfrac{(\mathbf{D}\times\mathbf{W}_1\cdot\mathbf{E}_0)(\mathbf{D}\times\mathbf{W}_1\cdot\mathbf{E}_1)}{|\mathbf{D}\times\mathbf{W}_1|^3} \end{aligned} \tag{19}$$

with the understanding that the second-order derivatives have the same discontinuities as the first-order derivatives: $\boldsymbol{D}\cdot\boldsymbol{W}_i = 0$ and $\boldsymbol{D}\times\boldsymbol{W}_i = \boldsymbol{0}$ for $i = 0, 1$. Observe that,

$$f_{xx}f_{yy} - f_{xy}^2 \quad = \quad r_0 r_1 \tfrac{(\mathbf{D}\cdot\mathbf{W}_0\times\mathbf{W}_1)^2}{|\mathbf{D}\times\mathbf{W}_0|^3|\mathbf{D}\times\mathbf{W}_1|^3} \tag{20}$$

It is clear that $f_{xx} \geq 0$, $f_{yy} \geq 0$ and $f_{xx}f_{yy} - f_{xy}^2 \geq 0$, which imply that the function $f(x, y)$ is convex on each subdomain that contains no points of discontinuity. Generally, local convexity does not imply global convexity. For example, the function $g(t)$ with values $(t+1)^2$ for $t < 0$, values $(t-1)^2$ for $t > 0$, and $g(0) = 1$ is continuous. Its first-order derivative $g'(t)$ is $2(t+1)$ for $t < 0$, $2(t-1)$ for $t > 0$, and has a discontinuity at $t = 0$. The second-order derivative is $g''(t) = 1$ for $t \neq 0$ and is discontinuous at $t = 0$. Although $g''(t) > 0$ for all $t \in (-\infty, 0) \cup (0, +\infty)$, the graph of $g(t)$ consists of two parabolas, one for $t < 0$ and one for $t > 0$. The function $g(t)$ is not globally convex. As will be seen later for the problem at hand, the restrictions of $f(x, y)$ to lines in the pole plane are globally convex with the implication that $f(x, y)$ is globally convex.

7 Pseudocode for the Algorithm

The search for the minimum value of $f(x, y)$ is accomplished by restricting the function to lines in the pole plane, computing the minimum along each line and then selecting the minimum of the line minima. The search has an early exit when a line has a negative minimum, in which case a separating direction has been found.

A finite set of line samples is selected for the search. If the minima at the samples are all nonnegative, it is possible that a separating direction is missed because it is not one of the samples. If the number of samples is large, an application might consider this a sufficient conclusion–that there is no separating direction. However, more computing can be used to try to locate a separating direction. The line samples are generated by angles $\theta_i \in [0, \pi)$. The directions are $(\cos\theta_i, \sin\theta_i, 0)$. The idea is to search the sample lines for the one leading to smallest g-value, say, $g_i = g(t; \theta_i)$. The triple $\langle g_{i-1}, g_i, g_{i+1}\rangle$ is a bracket for a minimum of $g(t)$. A 1-dimensional minimizer can be used, one that does not require derivative evaluations (in θ). One such minimizer uses successive parabolic interpolation, where the number of bisections will be a user-specified parameter.

7.1 Analysis of the Derivative Discontinuities

The lines are chosen to have a common origin, $\boldsymbol{P} = (p_0, p_1, 1)$, which is on both lines of discontinuity: $\boldsymbol{P}\cdot\boldsymbol{W}_0 = 0$ and $\boldsymbol{P}\cdot\boldsymbol{W}_1 = 0$. This pair of linear equations has a unique solution because $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ are not parallel. Geometricaly, $\boldsymbol{P}$ is perpendicular to both cylinder axis directions, so it is parallel to $\boldsymbol{W}_0\times\boldsymbol{W}_1$. The solution is

$$\boldsymbol{P} = \frac{\boldsymbol{W}_0\times\boldsymbol{W}_1}{\boldsymbol{E}_2\cdot\boldsymbol{W}_0\times\boldsymbol{W}_1} \tag{21}$$

The line directions are chosen to be $\boldsymbol{L}(\theta) = (\cos\theta, \sin\theta, 0)$ for $\theta \in [0, \pi)$. The search lines are $\boldsymbol{D}(t, \theta) = (x(t, \theta), y(t, \theta), 1) = \boldsymbol{P} + t\boldsymbol{L}(\theta)$.

For a specified θ, the restriction of $f(x, y)$ to the line is defined by $g(t) = f(x(t), y(t))$, where the dependence on θ is omitted for readability,

$$g(t) = r_0|\boldsymbol{D}(t)\times\boldsymbol{W}_0| + r_1|\boldsymbol{D}(t)\times\boldsymbol{W}_1| + \frac{h_0}{2}\,|\boldsymbol{D}(t)\cdot\boldsymbol{W}_0| + \frac{h_1}{2}\,|\boldsymbol{D}(t)\cdot\boldsymbol{W}_1| - |\boldsymbol{\Delta}| \tag{22}$$

The first-order derivative is

$$
\begin{aligned}
g'(t) &= r_0\frac{\mathbf{D}(t)\times\mathbf{W}_0\cdot\mathbf{D}'(t)\times\mathbf{W}_0}{|\mathbf{D}(t)\times\mathbf{W}_0|} + r_1\frac{\mathbf{D}(t)\times\mathbf{W}_1\cdot\mathbf{D}'(t)\times\mathbf{W}_1}{|\mathbf{D}(t)\times\mathbf{W}_1|} + \frac{h_0}{2}\frac{(\mathbf{D}(t)\cdot\mathbf{W}_0)(\mathbf{D}'(t)\cdot\mathbf{W}_0)}{|\mathbf{D}(t)\cdot\mathbf{W}_0|} + \frac{h_1}{2}\frac{(\mathbf{D}(t)\cdot\mathbf{W}_1)(\mathbf{D}'(t)\cdot\mathbf{W}_1)}{|\mathbf{D}(t)\cdot\mathbf{W}_1|} \\
&= r_0\frac{(t\mathbf{L}\times\mathbf{W}_0+\mathbf{P}\times\mathbf{W}_0)\cdot\mathbf{L}\times\mathbf{W}_0}{|t\mathbf{L}\times\mathbf{W}_0+\mathbf{P}\times\mathbf{W}_0|} + r_1\frac{(t\mathbf{L}\times\mathbf{W}_1+\mathbf{P}\times\mathbf{W}_1)\cdot\mathbf{L}\times\mathbf{W}_1}{|t\mathbf{L}\times\mathbf{W}_1+\mathbf{P}\times\mathbf{W}_1|} + \frac{h_0}{2}\frac{(t\mathbf{L}\cdot\mathbf{W}_0)(\mathbf{L}\cdot\mathbf{W}_0)}{|t\mathbf{L}\cdot\mathbf{W}_0|} + \frac{h_1}{2}\frac{(t\mathbf{L}\cdot\mathbf{W}_1)(\mathbf{L}\cdot\mathbf{W}_1)}{|t\mathbf{L}\cdot\mathbf{W}_1|} \\
&= r_0\frac{(t\mathbf{L}\times\mathbf{W}_0+\mathbf{P}\times\mathbf{W}_0)\cdot\mathbf{L}\times\mathbf{W}_0}{|t\mathbf{L}\times\mathbf{W}_0+\mathbf{P}\times\mathbf{W}_0|} + r_1\frac{(t\mathbf{L}\times\mathbf{W}_1+\mathbf{P}\times\mathbf{W}_1)\cdot\mathbf{L}\times\mathbf{W}_1}{|t\mathbf{L}\times\mathbf{W}_1+\mathbf{P}\times\mathbf{W}_1|} + \sigma(t)\left(\frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|\right)
\end{aligned}
\tag{23}
$$

where $\sigma(t)$ is the sign function defined in equation (18). The evaluation of $g(t)$ on the discontinuity line $\boldsymbol{D}\cdot\boldsymbol{W}_0 = 0$ effectively uses only the terms with coefficients r_0, r_1 and h_1. The h_0-term of $g'(t)$ is undefined because $\boldsymbol{L}\cdot\boldsymbol{W}_0 = 0$. If instead that term is simplified to $(h_0/2)|\boldsymbol{L}\cdot\boldsymbol{W}_0|$, then evaluation of $g(t)$ effectively uses only the terms with coefficients r_0, r_1 and h_1. A similar argument applies for the discontinuity line $\boldsymbol{D}\cdot\boldsymbol{W}_1 = 0$ where the evaluations effectively use only the terms with coefficients r_0, r_1 and h_0. Therefore, equation (23) is valid for all lines with origin $\boldsymbol{P}$.

Using equation (23), the one-sided limits of $g'(t)$ at $t = \pm\infty$ are

$$
\begin{aligned}
&g'(+\infty) = \lim_{t\to+\infty} g'(t) = r_0|\mathbf{L}\times\mathbf{W}_0| + r_1|\mathbf{L}\times\mathbf{W}_1| + \left(\frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|\right) \\
&g'(-\infty) = -g'(+\infty)
\end{aligned}
\tag{24}
$$

Using equation (23), the one-sided limits of $g'(t)$ at $t = 0$ are

$$
\begin{aligned}
&g'(0^-) = \lim_{t\to 0^-} g'(t) = r_0\frac{\mathbf{P}\times\mathbf{W}_0\cdot\mathbf{L}\times\mathbf{W}_0}{|\mathbf{P}\times\mathbf{W}_0|} + r_1\frac{\mathbf{P}\times\mathbf{W}_1\cdot\mathbf{L}\times\mathbf{W}_1}{|\mathbf{P}\times\mathbf{W}_1|} - \left(\frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|\right) \\
&g'(0^+) = \lim_{t\to 0^+} g'(t) = r_0\frac{\mathbf{P}\times\mathbf{W}_0\cdot\mathbf{L}\times\mathbf{W}_0}{|\mathbf{P}\times\mathbf{W}_0|} + r_1\frac{\mathbf{P}\times\mathbf{W}_1\cdot\mathbf{L}\times\mathbf{W}_1}{|\mathbf{P}\times\mathbf{W}_1|} + \left(\frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|\right)
\end{aligned}
\tag{25}
$$

The jump in the derivative at $t = 0$ is $h_0|\boldsymbol{L}\cdot\boldsymbol{W}_0| + h_1|\boldsymbol{L}\cdot\boldsymbol{W}_1| > 0$, so $g(t)$ is convex in a full neighborhood of $t = 0$.

Let $\boldsymbol{Q}_0$ be the pole-plane point that solves $\boldsymbol{Q}_0\times\boldsymbol{W}_0 = \mathbf{0}$, in which case $\boldsymbol{Q}_0$ is parallel to $\boldsymbol{W}_0$. If the sample line contains the discontinuity point $\boldsymbol{Q}_0$, then choose the parameterization $\boldsymbol{D}(t) = \boldsymbol{Q}_0 + (t-1)\boldsymbol{L}$, where $\boldsymbol{L} = \boldsymbol{Q_0} - \boldsymbol{P}$. It follows that $\boldsymbol{D}(t)\times\boldsymbol{W}_0 = (t-1)\boldsymbol{L}\times\boldsymbol{W}_0$ and

$$
g'(t) = r_0\sigma(t-1)|\mathbf{L}\times\mathbf{W}_0| + r_1\frac{(t\mathbf{L}\times\mathbf{W}_1+\mathbf{P}\times\mathbf{W}_1)\cdot\mathbf{L}\times\mathbf{W}_1}{|t\mathbf{L}\times\mathbf{W}_1+\mathbf{P}\times\mathbf{W}_1|} + \sigma(t)\left(\frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|\right)
\tag{26}
$$

The one-sided limits of $g'(t)$ at $t = 1$ are

$$
\begin{aligned}
&g'(1^-) = \lim_{t\to 1^-} g'(t) = -r_0|\mathbf{L}\times\mathbf{W}_0| + r_1\frac{\mathbf{Q}_0\times\mathbf{W}_1\cdot\mathbf{L}\times\mathbf{W}_1}{|\mathbf{Q}_0\times\mathbf{W}_1|} + \frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1| \\
&g'(1^+) = \lim_{t\to 1^+} g'(t) = +r_0|\mathbf{L}\times\mathbf{W}_0| + r_1\frac{\mathbf{Q}_0\times\mathbf{W}_1\cdot\mathbf{L}\times\mathbf{W}_1}{|\mathbf{Q}_0\times\mathbf{W}_1|} + \frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|
\end{aligned}
\tag{27}
$$

The jump in the derivative at $t = 1$ is $2r_0|\mathbf{L}\times\mathbf{W}_0| > 0$, so $g(t)$ is convex in a full neighborhood of $t = 1$.

Let $\boldsymbol{Q}_1$ be the pole-plane point that solves $\boldsymbol{Q}_1\times\boldsymbol{W}_1 = \mathbf{0}$, in which case $\boldsymbol{Q}_1$ is parallel to $\boldsymbol{W}_1$. If the sample line contains the discontinuity point $\boldsymbol{Q}_1$, then choose the parameterization $\boldsymbol{D}(t) = \boldsymbol{Q}_1 + (t-1)\boldsymbol{L}$, where $\boldsymbol{L} = \boldsymbol{Q_1} - \boldsymbol{P}$. It follows that $\boldsymbol{D}(t)\times\boldsymbol{W}_1 = (t-1)\boldsymbol{L}\times\boldsymbol{W}_1$ and

$$
g'(t) = r_0\frac{(t\mathbf{L}\times\mathbf{W}_0+\mathbf{P}\times\mathbf{W}_0)\cdot\mathbf{L}\times\mathbf{W}_0}{|t\mathbf{L}\times\mathbf{W}_0+\mathbf{P}\times\mathbf{W}_0|} + r_1\sigma(t-1)|\mathbf{L}\times\mathbf{W}_1| + \sigma(t)\left(\frac{h_0}{2}|\mathbf{L}\cdot\mathbf{W}_0| + \frac{h_1}{2}|\mathbf{L}\cdot\mathbf{W}_1|\right)
\tag{28}
$$

The one-sided limits of $g'(t)$ at $t = 1$ are

$$\begin{array}{l} g'(1^-) = \lim_{t \rightarrow 1^-} g'(t) = r_0 \frac{\mathbf{Q}_1 \times \mathbf{W}_0 \cdot \mathbf{L} \times \mathbf{W}_0}{|\mathbf{Q}_1 \times \mathbf{W}_0|} - r_1 |\mathbf{L} \times \mathbf{W}_1| + \frac{h_0}{2} |\mathbf{L} \cdot \mathbf{W}_0| + \frac{h_1}{2} |\mathbf{L} \cdot \mathbf{W}_1| \\ g'(1^+) = \lim_{t \rightarrow 1^+} g'(t) = r_0 \frac{\mathbf{Q}_1 \times \mathbf{W}_0 \cdot \mathbf{L} \times \mathbf{W}_0}{|\mathbf{Q}_1 \times \mathbf{W}_0|} + r_1 |\mathbf{L} \times \mathbf{W}_1| + \frac{h_0}{2} |\mathbf{L} \cdot \mathbf{W}_0| + \frac{h_1}{2} |\mathbf{L} \cdot \mathbf{W}_1| \end{array} \tag{29}$$

The jump in the derivative at $t = 1$ is $2r_1|\mathbf{L} \times \mathbf{W}_1| > 0$, so $g(t)$ is convex in a full neighborhood of $t = 1$.

7.2 Pseudocode for Evaluation of $g(t)$ and $g'(t)$

Pseudocode for computing $g(t)$, $g'(t)$ and the one-sided limits is shown in listing 1.

Listing 1. Pseudocode for computing $g(t)$ and $g'(t)$ at nonsingular values and for computing the one-sided limits at $t = 0$ and $t = 1$. The one-sided limits $g'(0^-)$ and $g'(0^+)$ are represented by dgdt0n and dgdt0p, respectively. The one-side limits $g'(1^-)$ and $g'(1^+)$ are represented by dgdt1n and dgdt1p, respectively. The one-sided limit $g'(+\infty)$ is represented by dgdtInfinity. It is always true that $g'(-\infty) = -g'(+\infty)$. The pseudocode functions are assumed to have access to the cylinder information as global state.

```
// Compute one-sided limits at t = 0 for any line.
void LimitsDGDTZero(Real& dgdt0n, Real& dgdt0p)
{
    Real r0Term = r0 * Dot(Cross(P, W0), Cross(L, W0)) / Length(Cross(P, W0));
    Real r1Term = r1 * Dot(Cross(P, W1), Cross(L, W1)) / Length(Cross(P, W1));
    Real h0Term = (h0/2) * Abs(Dot(L, W0));
    Real h1Term = (h1/2) * Abs(Dot(L, W1));
    dgdt0n = r0Term + r1Term - (h0Term + h1Term);
    dgdt0p = r0Term + r1Term + (h0Term + h1Term);
}

// Compute one-sided limits at t = 1 for line P + t(Q0 - P) where Q0 x W0 = 0 with Q0 last component 1.
void LimitsDGDTOneQ0(Real& dgdt1n, Real& dgdt1p)
{
    Real r0Term = r0 * Length(Cross(L, W0));
    Real r1Term = r1 * Dot(Cross(Q0, W1), Cross(L, W1)) / Length(Cross(Q0, W1));
    Real h0Term = (h0/2) * Abs(Dot(L, W0));
    Real h1Term = (h1/2) * Abs(Dot(L, W1));
    dgdt1n = -r0Term + r1Term + h0Term + h1Term;
    dgdt1p = +r0Term + r1Term + h0Term + h1Term;
}

// Compute one-sided limits at t = 1 for line P + t(Q1 - P) where Q1 x W1 = 0 with Q1 last component 1.
void LimitsDGDTOneQ1(Real& dgdt1n, Real& dgdt1p)
{
    Real r0Term = r0 * Dot(Cross(Q1, W0), Cross(L, W0)) / Length(Cross(Q1, W0));
    Real r1Term = r1 * Length(Cross(L, W1));
    Real h0Term = (h0/2) * Abs(Dot(L, W0));
    Real h1Term = (h1/2) * Abs(Dot(L, W1));
    dgdt1n = r0Term - r1Term + h0Term + h1Term;
    dgdt1p = r0Term + r1Term + h0Term + h1Term;
}

// Compute one-sided limit g'(+inf) with g'(-inf) = -g'(+inf).
void LimitsDGDTInfinity(Real& dgdtInfinity)
{
    Real r0Term = r0 * Length(Cross(L, W0));
    Real r1Term = r1 * Length(Cross(L, W1));
    Real h0Term = (h0/2) * Abs(dotLW0);
    Real h1Term = (h1/2) * Abs(dotLW1);
    dgdtInfinity = r0Term + r1Term + h0Term + h1Term;
}
```

```
// Evaluation of g(t) for all t.
Real G(Real t)
{
    Vector3<Real> D = P + L * t;
    Real r0Term = r0 * Length(Cross(D, W0));
    Real r1Term = r1 * Length(Cross(D, W1));
    Real h0Term = (h0/2) * Abs(Dot(D, W0));
    Real h1Term = (h1/2) * Abs(Dot(D, W1));
    return r0Term + r1Term + h0Term + h1Term - Length(delta);
}

// Evaluation of g'(t) except at singularity t = 0 and potential singularity t = 1.
Real DGDT(Real t)
{
    Vector3<Real> D = P + L * t;
    Real r0Term = r0 * Dot(Cross(D, W0), Cross(L, W0)) / Length(Cross(D, W0));
    Real r1Term = r1 * Dot(Cross(D, W1), Cross(L, W1)) / Length(Cross(D, W1));
    Real sgn = sign(t);
    Real h0Term = (h0/2) * Abs(Dot(L, W0)) * sgn;
    Real h1Term = (h1/2) * Abs(Dot(L, W1)) * sgn;
    return r0Term + r1Term + h0Term + h1Term;
}
```

7.3 Pseudocode for Computing the Minimum of $g(t)$

The convexity of $g(t)$ on a line allows for a robust floating-point minimizer. The lines all have a common origin at $\boldsymbol{P}$ specified in equation (21), and that origin occurs at $t = 0$. For a line with a discontinuity only at $t = 0$, the minimum of $g(t)$ occurs either at $\bar{t} \in (-\infty, 0)$ or at $\bar{t} \in (0, +\infty)$ where $g'(\bar{t}) = 0$ or at the discontinuity $\bar{t} = 0$. Listing 2 contains pseudocode for locating the number $\bar{t}$ and computing $g(\bar{t})$.

Listing 2. Pseudocode for computing the minimum of $g(t)$. The one-sided limits $g'(0^-)$ and $g'(0^+)$ are represented by dgdt0n and dgdt0p, respectively. The local variables imax and maxBisections have values based on whether Real is float or double. The pseudocode functions are assumed to have access to the cylinder information as global state.

```
// Compute the minimum of g(t), which occurs at t̄. The output tMin is t̄ and the output gMin is g(t̄).
void ComputeMinimumSingularZero(Real dgdt0n, Real dgdt0p, Real& tMin, Real& gMin)
{
    if (dgdt0n > 0)
    {
        // The root of g'(t) occurs on (-∞, 0). Locate t0 for which g'(t0) < 0.
        Real t0 = -1, t1 = 0, dgdtT0, dgdtTMin;
        for (int i = 0; i < imax; ++i)
        {
            dgdtT0 = g'(t0);
            if (dgdtT0 < 0)
            {
                break;
            }
            t0 *= 2;
        }

        Bisection<Real> bisector(maxBisections);
        bisector(DGDT, t0, t1, dgdtT0, dgdt0n, tMin, dgdtTMin);
    }
    else if (dgdt0p < 0)
    {
        // The root of g'(t) occurs on (0, +∞).. Locate t1 for which g'(t1) > 0.
        Real t0 = 0, t1 = 1, dgdtT1, dgdtTMin;
```

```
        for (int i = 0; i < imax; ++i)
        {
            dgdtT1 = g'(t1);
            if (dgdtAtT1 > 0)
            {
                break;
            }
            t1 *= 2;
        }

        Bisection<Real> bisector(maxBisections);
        bisector(DGDT, t0, t1, dgdt0p, dgdtT1, tMin, dgdtTMin);
    }
    else
    {
        // At this time, g'(0^-) <= 0 <= g'(0^+). The minimum of g(t) occurs at g(0).
        tMin = 0;
    }

    gMin = G(tMin);
}
```

For a line with discontinuities at $t = 0$ and $t = 1$, of which there are at most two, the minimum of $g(t)$ occurs either at $\bar{t} \in (-\infty, 0) \cup (0, 1) \cup (1, +\infty)$ where $g'(\bar{t}) = 0$ or at one of the discontinuities $\bar{t} \in \{0, 1\}$. Listing 3 contains pseudocode for locating the number $\bar{t}$ and computing $g(\bar{t})$.

Listing 3. Pseudocode for computing the minimum of $g(t)$. The one-sided limits $g'(0^-)$ and $g'(0^+)$ are represented by dgdt0n and dgdt0p, respectively. The one-sided limits $g'(1^-)$ and $g'(1^+)$ are represented by dgdt1n and dgdt0p, respectively. The local variables imax and maxBisections have values based on whether Real is float or double. The pseudocode functions are assumed to have access to the cylinder information as global state.

```
// Compute the minimum of g(t), which occurs at t̄. The output tMin is t̄ and the output gMin is g(t̄).
void ComputeMinimumSingularZeroOne(Real dgdt0n, Real dgdt0p, Real dgdt1n, Real dgdt1p, Real& tMin, Real& gMin)
{
    if (dgdt0n > 0)
    {
        // The root of g'(t) occurs on (-inf, 0). Locate t_0 for which g'(t_0) < 0.
        Real t0 = -1, t1 = 0, dgdtT0, dgdtTMin;
        for (int i = 0; i < imax; ++i)
        {
            dgdtT0 = g'(t0);
            if (dgdtT0 < 0)
            {
                break;
            }
            t0 *= 2;
        }

        Bisection<T> bisector(maxBisections);
        bisector(DGDT, t0, t1, dgdtT0, dgdt0n, tMin, dgdtTMin);
    }
    else if (dgdt1p < 0)
    {
        // The root of g'(t) occurs on (1, +inf). Locate t_1 for which g'(t_1) > 0.
        Real t0 = 1, t1 = 2, dgdtT1, dgdtTMin;
        for (int i = 0; i < imax; ++i)
        {
            dgdtT1 = g'(t1);
            if (dgdtT1 > 0)
            {
                break;
            }
```

```
            t1 *= 2;
        }

        Bisection<T> bisector(maxBisections);
        bisector(DGDT, t0, t1, dgdt1p, dgdtT1, tMin, dgdtTMin);
    }
    else
    {
        // At this time, g'(0^-) <= 0 <= g'(1^+).
        if (dgdt0p < 0)
        {
            if (dgdt1n > 0)
            {
                // The root of g'(t) occurs on (0,1).
                Real t0 = 0, t1 = 1, dgdtTMin;
                Bisection<T> bisector(maxBisections);
                bisector(DGDT, t0, t1, dgdt0p, dgdt1n, tMin, dgdtTMin);
            }
            else
            {
                // The minimum of g(t) occurs at g(1).
                tMin = 1;
            }
        }
        else
        {
            // The minimum of g(t) occurs at g(0).
            tMin = 0;
        }
    }

    gMin = G(tMin);
}
```

7.4 Pseudocode for Computing a Separating Direction

This section describes the high-level details for the test-intersection query between two bounded cylinders. The cylinders have centers $\boldsymbol{C}_i$, unit-length axis directions $\boldsymbol{W}_i$, radii r_i and heights h_i for $i \in \{0, 1\}$.

At the top-most level, pseudocode for the separation function is shown in listing 4.

Listing 4. Pseudocode for the top-most level of the separation function. The inputs are the parameters of the cylinders and the number of line samples to be used for the coarse-level search. The function returns true whenever the cylinders are separated, in which case separatingDirection is valid and separates the cylinder. If the function returns false, separatingDirection is invalid.

```
bool SeparatedCylinders(
    Vector3<Real> C0, Vector3<Real> W0, Real r0, Rea> h0,
    Vector3<Real> C1, Vector3<Real> W1, Real r1, Real h1,
    int numLines,
    Vector3<Real>& separatingDirection)
{
    if (C0 == C1)
    {
        return false;
    }

    Vector3<Real> Delta = C1 - C0;
    Vector3<Real> W0xW1 = Cross(W0, W1);
    Real lenW0xW1 = Length(W0xW1);
```

```
    if (lenW0xW1 > 0)
    {
        // Test for separation by W0.
        if (r1*lenW0xW1 + h0/2 + (h1/2)*Abs(Dot(W0,W1)) - Abs(Dot(W0, Delta)) < 0)
        {
            separatingDirection = W0;
            return true;
        }

        // Test for separation by W1.
        if (r0*lenW0xW1 + (h0/2)*Abs(Dot(W0,W1)) + (h1/2) - Abs(Dot(W1, Delta)) < 0)
        {
            separatingDirection = W1;
            return true;
        }

        // Test for separation by W0 x W1.
        if (rSum*lenW0xW1 - Abs(Dot(W0xW1, Delta)) < 0)
        {
            separatingDirection = W0xW1;
            Normalize(separatingDirection);
            return true;
        }

        // Test for separation by Delta.
        if (r0*Length(Cross(Delta,W0)) + r1*Length(Cross(Delta,W1))) +
            (h0/2) * Abs(Dot(Delta,W0)) + (h1/2)*Abs(Dot(Delta,W1)) - Dot(Delta, Delta)) < 0)
        {
            separatingDirection = mDelta;
            Normalize(separatingDirection);
            return true;
        }

        // Test for separation by other directions. This function implements the minimum search described previously.
        if (SeparatedByOtherDirections(numLines, Delta,W0, r0, h0,W1, r1, h1, separatingDirection))
        {
            return true;
        }
    }
    else
    {
        // Test for separation by height.
        if ((h0 + h1)/2 - Abs(Dot(Delta,W0)) < 0)
        {
            separatingDirection = W0;
            return true;
        }

        // Test for separation radially.
        if ((r0 + r1) - Length(Cross(Delta,W0)) < 0)
        {
            separatingDirection = Delta - Dot(Delta,W0)*W0;
            Normalize(separatingDirection);
            return true;
        }
    }

    separatingDirection = { 0, 0, 0 };
    return false;
}
```

The function SeparatedByOtherDirections implements the minimum search described in this document. It is shown in listing 5.

Listing 5. The minimum search used to locate a separating direction, if one exists.

```
// Global state accessed by other functions is marked by comments.
bool SeparatedByOtherDirections(int numLines, Vector3<Real> Delta, Vector3<Real> W0, Real r0, Real h0,
    Vector3<Real> W1, Real r1, Real h1, Vector3<Real>& separatingDirection)
{
    // r0, r1, h0 and h1 are global state
    Real lengthDelta = Length(Delta); // global state

    // Convert to the coordinate system where N = Δ/|Δ| is the north pole of the hemisphere to be searched.
    Vector3<Real> U, V, N = Delta / lengthDelta; // global state
    ComputeOrthonormalBasis(U, V, N);
    Vector3<Real> W0 = { Dot(U, W0), Dot(V, W0), Dot(N, W0)}; // global state
    Vector3<Real> W1 = { Dot(U, W1), Dot(V, W1), Dot(N, W1)}; // global state
    Vector3<Real> W0xW1 = Cross(W0, W1); // global state

    // The axis directions and their cross product must be in the hemisphere with north pole N.
    if (W0[2] < 0) { W0 = -W0; }
    if (W1[2] < 0) { W1 = -W1; }
    if (W0xW1[2] < 0) { W0xW1 = -W0xW1; }

    // Compute the common origin for the line discontinuities.
    Vector3<Real> P = W0xW1 / W0xW1[2]; // global state

    // Compute the point discontinuities.
    Vector3<Real> Q0 = W0 / W0[2]; // global state
    Vector3<Real> Q1 = W1 / W1[2]; // global state

    // Search the line samples for a separating direction.
    array<array<Real, 2>, numLines> lineMinimum(numLines + 1);   // Each element is (angle,gBar).
    Real tMin, gMin;
    for (int i = 0; i < numLines; ++i)
    {
        // Compute a line direction.
        Real angle = PI * i / numLines;

        // Compute the minimum of g(t) on the line P + tL(θ).
        Vector3<Real> L = { cos(angle), sin(angle), 0 };
        ComputeLineMinimum(L, tMin, gMin);
        lineMinimum[i] = gMin;

        // Exit early when a line minimum is negative.
        if (gMin < 0)
        {
            // Transform to the original coordinate system.
            Vector3<Real> polePoint = P + tMin * L;
            separatingDirection = polePoint[0] * U + polePoint[1] * V + N;
            Normalize(separatingDirection);
            return true;
        }
    }
    lineMinimum[numLines] = lineMinimum[0];

    // The samples did not produce a negative minimum. Use a derivativeless minimizer to refine the search. The
    // pseudocode uses a successive parabolic interpolator.
    int numBisections = internally_defined;    // Known at compile time based on Real of float or double.
    SuccessiveParabolicInterpolator<Real> minimizer(numBisections);

    // Locate a triple of points that bracket the minimum.
    array<int, 3> bracket = { numLines-1, 0, 1 };
    for (int i0 = 0, i1 = 1, i2 = 2; i2 <= numLines i0 = i1, i1 = i2++)
    {
        if (lineMinimum[i1] < lineMinimum[bracket[1]])
        {
```

```
            bracket = { i0, i1, i2 };
        }
    }

    // The minimizer must have access to ComputeLineMinimum and to global state.
    T angle0 = PI * bracket[0] / numLines;
    T angle1 = PI * bracket[1] / numLines;
    T angle2 = PI * bracket[2] / numLines;
    T angleMin;
    minimizer(lineMinimum[bracket[0]], lineMinimum[bracket[1]], lineMinimum[bracket[2]], angleMin, gMin);
    if (gMin < 0)
    {
        // Transform to the original coordinate system.
        Vector3<Real> polePoint = P + tMin * L;
        separatingDirection = polePoint[0] * U + polePoint[1] * V + N;
        Normalize(separatingDirection);
        return true;
    }

    return false;
}

// Compute the minimum of g(t) on the line P + tL(θ). The input angle is θ ∈ [0, π).
void ComputeLineMinimum(Vector3<Real> L, Real& tMin, Real& gMin)
{
    Vector3<Real> LPerp = { L[1], -L[0], 0 };
    Real dgdt0n, dgdt0p, dgdt1n, dgdt1p;

    if (Dot(LPerp, Q0 - P) != 0)
    {
        // Q0 is not on the line P+t*L(angle).
        if (Dot(LPerp, Q1 - P) != 0)
        {
            // Q1 is not on the line P+t*L(angle).
            LimitsDGDTZero(dgdt0n, dgdt0p);
            ComputeMinimumSingularZero(dgdt0n, dgdt0p, tMin, gMin);
        }
        else
        {
            // Q1 is on the line P+t*L(angle).
            LimitsDGDTZero(dgdt0n, dgdt0p);
            LimitsDGDTOneQ1(dgdt1n, dgdt1p);
            ComputeMinimumSingularZeroOne(dgdt0n, dgdt0p, dgdt1n, dgdt1p, tMin, gMin);
        }
    }
    else
    {
        // Q0 is on the line P+t*L(angle).
        LimitsDGDTZero(dgdt0n, dgdt0p);
        LimitsDGDTOneQ0(dgdt1n, dgdt1p);
        ComputeMinimumSingularZeroOne(dgdt0n, dgdt0p, dgdt1n, dgdt1p, tMin, gMin);
    }
}
```

---

8 An Example of the Structure of $f(x, y)$

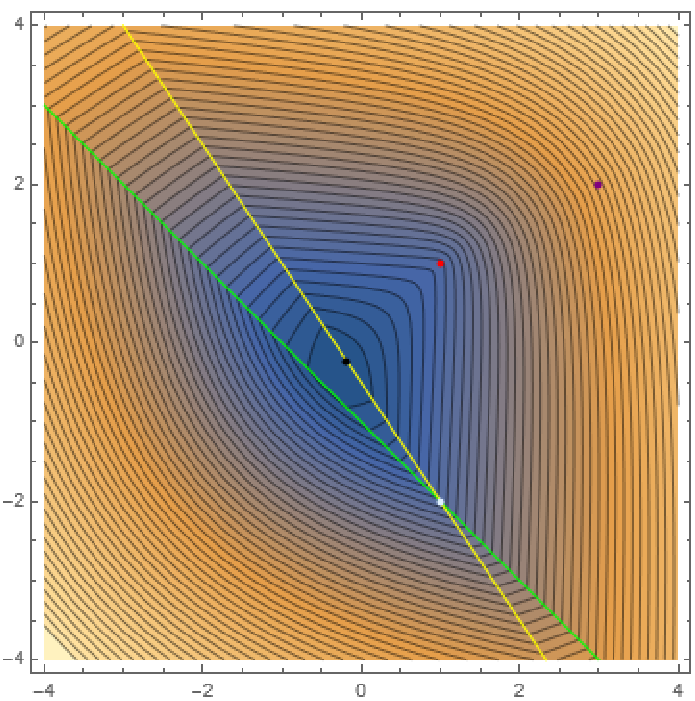
Let the first cylinder have axis direction $\boldsymbol{W}_0 = (1, 1, 1)/\sqrt{3}$, radius $r_0 = 1$ and height $h_0 = 2$. Let the second cylinder have axis direction $\boldsymbol{W}_1 = (3, 2, 1)/\sqrt{14}$, radius $r_1 = 1/8$ and height $h_1 = 1$. For simplicity, the difference of cylinder centers is chosen to be a segment with direction $(1, 0, 0)$ and length L; that is, $\boldsymbol{\Delta} = L(1, 0, 0)$. For various choices of L, the cylinders overlap. For other choices, the cylinders are separated.

It is instructive to compute the global minimum of $f(x, y)$ without the $|\boldsymbol{\Delta}|$ term, say, $F(x, y)$ defined by

$$\begin{aligned} F(x,y) &= r_0|\boldsymbol{D}\times\boldsymbol{W}_0| + r_1|\boldsymbol{D}\times\boldsymbol{W}_1| + \tfrac{h_0}{2}\,|\boldsymbol{D}\cdot\boldsymbol{W}_0| + \tfrac{h_1}{2}\,|\boldsymbol{D}\cdot\boldsymbol{W}_1| \\ &= \frac{\sqrt{(1+x+y)^2}}{\sqrt{3}} + \frac{\sqrt{(1+3x+2y)^2}}{2\sqrt{14}} + \frac{\sqrt{1-x-y+x^2-xy+y^2}}{\sqrt{3/2}} + \frac{\sqrt{13-6x-4y+5x^2-12xy+10y^2}}{8\sqrt{14}} \end{aligned} \tag{30}$$

Mathematica [1] was used to obtain information about $F(x, y)$. The NMinimize function shows $F_{\min} \doteq 1.45537$ and occurs at $(x, y) \doteq (-0.179602, -0.230596)$. A contour plot of $F(x, y)$ for $(x, y) \in [-4, 4]^2$ is shown in figure 1.

Figure 1. A contour plot for $F(x, y)$ of equation (30) for $(x, y) \in [-4, 4]^2$. The figure was generated with Mathematica [1]. Two lines and four points are overlaid; they are described in the text after this figure.



The green line contains the points for which $\boldsymbol{D}\cdot\boldsymbol{W}_0 = 0$; the line is $1+x+y = 0$. The yellow line contains the points for which $\boldsymbol{D}\cdot\boldsymbol{W}_1 = 0$; the line is $1 + 3x + 2y = 0$. The red point (in the blue region) is $(x, y) = (1, 1)$ and is the solution to $\boldsymbol{D}\times\boldsymbol{W}_0 = \boldsymbol{0}$. The purple dot (in the orange region) is $(x, y) = (3, 2)$ and is the solution to $\boldsymbol{D}\times\boldsymbol{W}_1 = \boldsymbol{0}$. The light blue point is $(1, -2)$ and is the intersection of the green and yellow lines; this corresponds to the point $\boldsymbol{P}$ that solves $\boldsymbol{D}\cdot\boldsymbol{W}_0\times\boldsymbol{W}_1 = \boldsymbol{0}$. The black point is the location of the minimum of $F(x, y)$, occurring at $(x, y) \doteq (-0.179602, -0.230596)$. In this example, the minimum point is on the yellow line $1 + 3x + 2y = 0$.

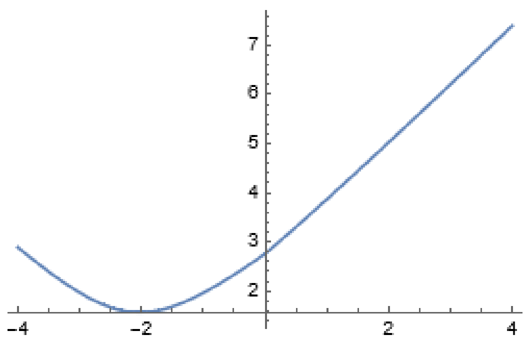
The search for the minimum will include computing local minima of $F(x, y)$ on lines with a common origin (x_0, y_0), where $D(x_0, y_0)$ is the solution to both $\boldsymbol{D}(x, y)\cdot\boldsymbol{W}_0 = 0$ and $\boldsymbol{D}(x, y)\cdot\boldsymbol{W}_1 = 0$. In Figure 1,

the common origin is the light blue point. Knowing that $F(x, y)$ is a convex function, the restriction to a parameterized line $g(t) = F(x(t), y(t))$ is also convex. This implies that $g'(t)$ is an increasing function.

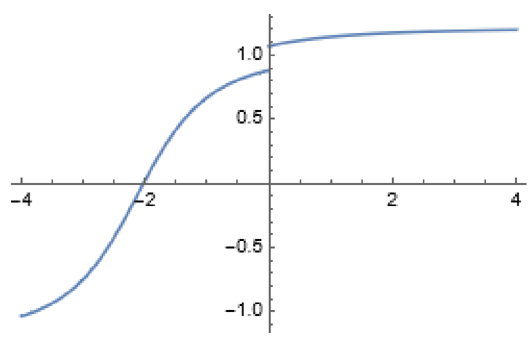
8.1 Analysis of $F(x, y)$ on Line $\boldsymbol{D} \cdot \boldsymbol{W}_0 = 0$

Parameterize the line $1 + x + y = 0$ so that the origin is at the intersection with $1 + 3x + 2y = 0$, say, $(x(t), y(t)) = (1, -2) + t(1, -1)/\sqrt{2}$ where $t \in \mathbb{R}$. The graphs of $g_0(t) = F(x(t), y(t))$ and $g_0'(t)$ for $t \in [-4, 4]$ are shown in figure 2.

Figure 2. The graphs of $g_0(t) = F(x(t), y(t))$ and $g_0'(t)$ for $t \in [-4, 4]$. The figures were generated with Mathematica [1].



The graph of $g_0(t)$.



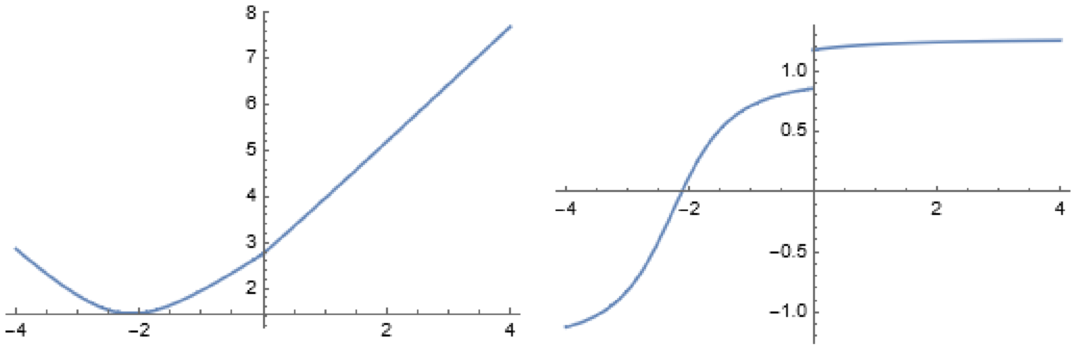
The graph of $g_0'(t)$.

The root of $g_0'(t)$ occurs at $\bar{t} \doteq -2.02716$. The minimum value of $g_0(t)$ is $g_0(\bar{t}) \doteq 1.56579$. The one-sided limits of $g_0'(t)$ at 0 are $g_0'(0^-) \doteq 0.879787$ and $g_0'(0^+) \doteq 1.06877$. The infinite limits are $g_0'(-\infty) \doteq -1.21724$ and $g_0'(+\infty) \doteq 1.21724$.

8.2 Analysis of $F(x, y)$ on Line $\boldsymbol{D} \cdot \boldsymbol{W}_1 = 0$

Parameterize the line $1 + 3x + 2y = 0$ so that the origin is at the intersection with $1 + x + y = 0$, say, $(x(t), y(t)) = (1, -2) + t(2, -3)/\sqrt{13}$ where $t \in \mathbb{R}$. The graphs of $g_1(t) = F(x(t), y(t))$ and $g_1'(t)$ for $t \in [-4, 4]$ are shown in figure 3.

Figure 3. The graphs of $g_1(t) = F(x(t), y(t))$ and $g_1'(t)$ for $t \in [-4, 4]$. The figures were generated with Mathematica [1].



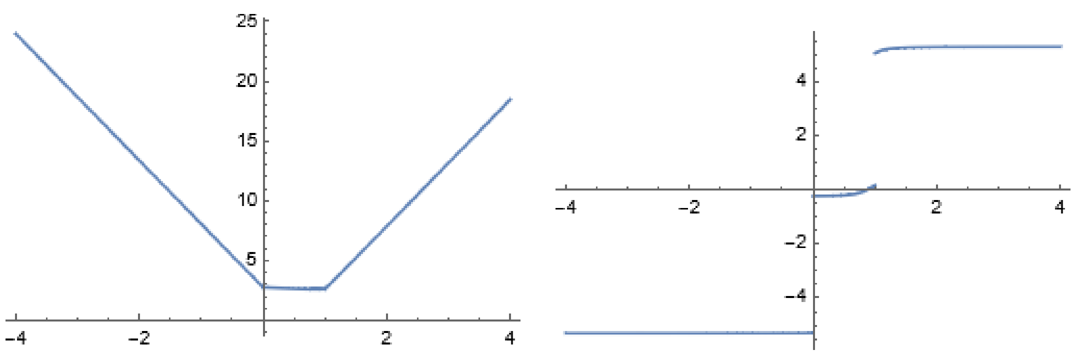
The graph of $g_1(t)$. The graph of $g_1'(t)$.

The root of $g_1'(t)$ occurs at $\bar{t} \doteq -2.12656$. The minimum value of $g_1(t)$ is $g_1(\bar{t}) \doteq 1.45537$. The one-sided limits of $g_1'(t)$ at 0 are $g_1'(0^-) \doteq 0.858921$ and $g_1'(0^+) \doteq 1.17918$. The infinite limits are $g_0'(-\infty) \doteq -1.27222$ and $g_0'(+\infty) \doteq 1.27222$.

8.3 Analysis of $F(x, y)$ on Line Containing Solution to $\boldsymbol{D} \times \boldsymbol{W}_0 = \mathbf{0}$

Parameterize the line with origin $(1, -2)$ and containing $(1, 1)$, the latter point the solution to $\boldsymbol{D} \times \boldsymbol{W}_0 = \mathbf{0}$, say, $(x(t), y(t)) = (1, -2) + t(0, 3)$ where $t \in \mathbb{R}$. Define $g_2(t) = F(x(t), y(t))$ with derivatives $g_2'(t)$ and $g_2''(t)$. The graphs of $g_2(t)$ and $g_2'(t)$ for $t \in [-4, 4]$ are shown in figure 4.

Figure 4. The graphs of $g_2(t) = F(x(t), y(t))$ and $g_2'(t)$ for $t \in [-4, 4]$. The figures were generated with Mathematica [1].



The graph of $g_2(t)$. The graph of $g_2'(t)$.

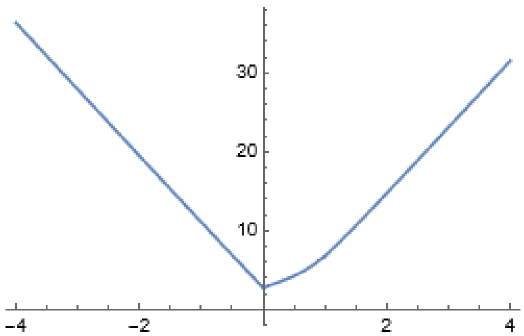
The root of $g_2'(t)$ occurs at $\bar{t} \doteq 0.864466$. The minimum value of $g_2(t)$ is $g_2(\bar{t}) \doteq 2.60442$. The one-sided

limits of $g_2'(t)$ at 0 are $g_2'(0^-) \doteq -5.28951$ and $g_2'(0^+) \doteq -0.221841$. The one-sided limits of $g_2'(t)$ at 1 are $g_2'(1^-) \doteq 0.166176$ and $g_2'(1^+) \doteq 5.06516$. The infinite limits are $g_2'(-\infty) \doteq -5.30026$ and $g_2'(+\infty) \doteq 5.30026$.

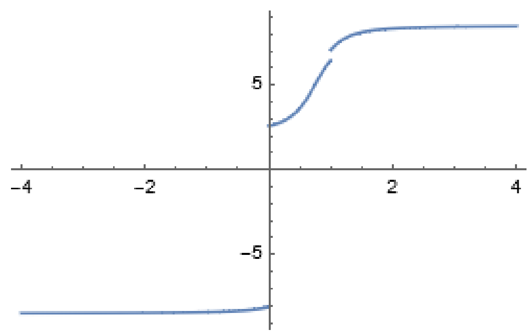
8.4 Analysis of $F(x, y)$ on Line Containing Solution to $\boldsymbol{D} \times \boldsymbol{W}_1 = \boldsymbol{0}$

Parameterize the line with origin $(1, -2)$ and containing $(3, 2)$, the latter point the solution to $\boldsymbol{D} \times \boldsymbol{W}_1 = \boldsymbol{0}$, say, $(x(t), y(t)) = (1, -2) + t(2, 4)$ where $t \in \mathbb{R}$. Define $g_3(t) = F(x(t), y(t))$ with derivatives $g_3'(t)$ and $g_3''(t)$. The graphs of $g_3(t)$ and $g_3'(t)$ for $t \in [-4, 4]$ are shown in figure 5.

Figure 5. The graphs of $g_3(t) = F(x(t), y(t))$ and $g_3'(t)$ for $t \in [-4, 4]$. The figures were generated with Mathematica [1].



The graph of $g_3(t)$.



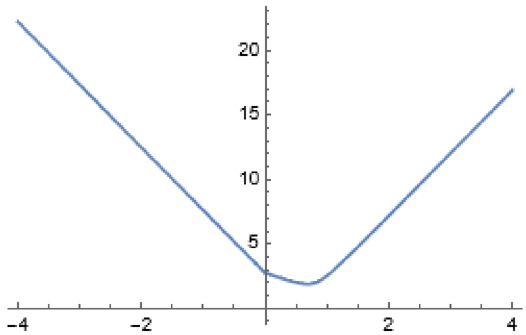
The graph of $g_3'(t)$.

The minimum of $g_3(t)$ occurs at the derivative discontinuity $t = 0$. The minimum value is $g_3(0) \doteq 2.75568$. The one-sided limits of $g_3'(t)$ at 0 are $g_3'(0^-) \doteq -8.09061$ and $g_3'(0^+) \doteq 2.57925$. The one-sided limits of $g_3'(t)$ at 1 are $g_3'(1^-) \doteq 6.44296$ and $g_3'(1^+) \doteq 7.05533$. The infinite limits are $g_3'(-\infty) \doteq -8.46954$ and $g_3'(+\infty) \doteq 8.46954$.

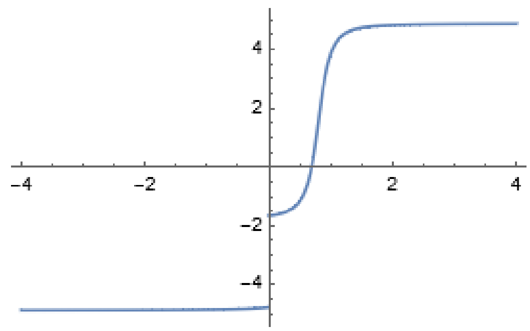
8.5 Analysis of $F(x, y)$ on Other Lines

For lines other than the 4 discussed previously, the only derivative discontinuity is at the common origin $(1, -2)$. For example, consider the line with origin $(1, -2)$ and containing $(0, 1)$. A parameterization is $(x(t), y(t)) = (1, -2) + t(-1, 3)$ where $t \in \mathbb{R}$. Define $g_4(t) = F(x(t), y(t))$ with derivatives $g_4'(t)$ and $g_4''(t)$. The graphs of $g_4(t)$ and $g_4'(t)$ for $t \in [-4, 4]$ are shown in figure 6.

Figure 6. The graphs of $g_4(t) = F(x(t), y(t))$ and $g_4'(t)$ for $t \in [-4, 4]$. The figures were generated with Mathematica [1].



The graph of $g_4(t)$.

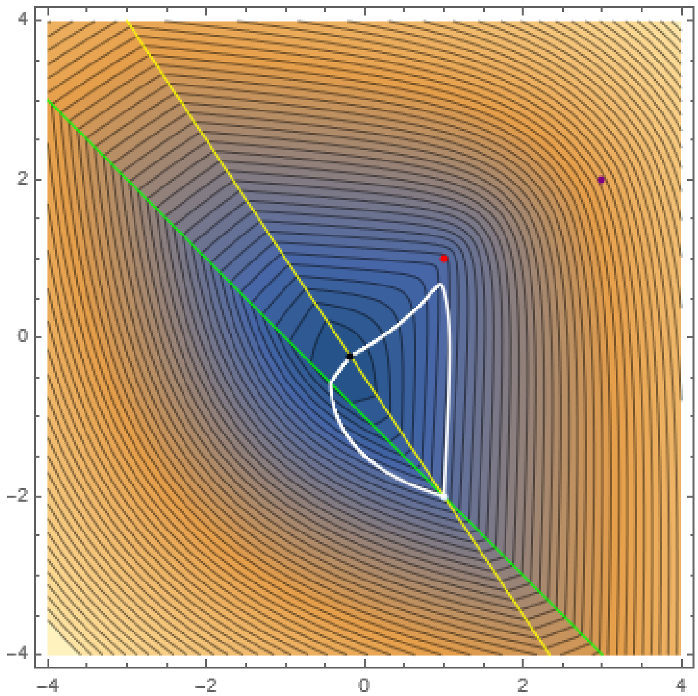


The graph of $g_4'(t)$.

The unique root of $g_4'(t)$ occurs at $\bar{t} \doteq 0.691351$ with $g_4(\bar{t}) \doteq 1.86579$. Listing 2 provides the pseudocode for computing the minimum of $g_4(t)$.

The minimum search of listing 5 produces a collection of line minima. Figure 7 shows the contour plot and discontinuity points overlaid with a polyline connecting the ordered minimizers.

Figure 7. The image of figure 1 overlaid with a white-colored polyline connecting the ordered minimizers of the lines with a common origin. The figure was generated with Mathematica [1].



The global minimum of $F(x, y)$ occurs at the black point shown in figure 7. The point happens to be located on the discontinuity line $\boldsymbol{D} \times \boldsymbol{W}_1 = 0$.

The tangent lines to the curve of minima at $\boldsymbol{P}$ form a wedge in the pole plane that contains the curve. Let the angles of the lines in the wedge be $[\theta_0, \theta_1]$. For $\theta \in (\theta_0, \theta_1)$, the lines with common origin $\boldsymbol{P}$ have minimizers occurring for $t > 0$. For $\theta \in [0, \theta_0] \cup [\theta_1, \pi)$, the lines with common origin $\boldsymbol{P}$ all have minimizers at $\boldsymbol{P}$; the minima occur when $t = 0$.

References

[1] Wolfram Research, Inc. *Mathematica 12.3.1.* Wolfram Research, Inc., Champaign, Illinois, 2021.