1992

On the Separability Problem of Real Functions and its Significance in Solid Modeling

Christoph M. Hoffmann
Purdue University, cmh@cs.purdue.edu

Report Number:
92-041

ON THE SEPARABILITY PROBLEM OF REAL FUNCTIONS AND ITS SIGNIFICANCE IN SOLID MODELING

Christoph M. Hoffmann

CSD-TR 92-041
July 1992

On the Separability Problem of Real Functions and its Significance in Solid Modeling*

Christoph M. Hoffmann
Computer Science Department
Purdue University
West Lafayette, Indiana 47907

July 11, 1992

Abstract

Given a set F of real-valued functions from $\mathbf{R}^3$ to $\mathbf{R}$, the separability problem asks for a set of auxiliary functions H that distinguishes the connected components of the cells of the sign-invariant partition of $\mathbf{R}^3$ induced by F. We explain the problem and some of the ways in which it comes up in solid modeling. We also summarize what is known about it in the solid modeling community.

1 Introduction

Given a set of n real-valued functions $F = \{f_1, ..., f_n\}$, each from $\mathbf{R}^3$ to $\mathbf{R}$, call two points p and q of $\mathbf{R}^3$ *equivalent under* F, denoted $p \equiv_F q$, if the sign vectors $(\mathrm{sgn}(f_1(p)), \mathrm{sgn}(f_2(p)), ..., \mathrm{sgn}(f_n(p)))$ and $(\mathrm{sgn}(f_1(q)), \mathrm{sgn}(f_2(q)), ..., \mathrm{sgn}(f_n(q)))$ are equal.[1] We consider the following

> *Separability Problem:*
> Given a set of n real-valued functions $F = \{f_1, ..., f_n\}$, each from $\mathbf{R}^3$ to $\mathbf{R}$, consider the sign-invariant partition induced by F on $\mathbf{R}^3$ whose cells are the equivalence classes under $\equiv_F$. Find a set of m *auxiliary* real-valued functions $H = \{h_1, ..., h_m\}$, also from $\mathbf{R}^3$ to $\mathbf{R}$, such that every connected component of the 3-dimensional equivalence classes in the F-partition is the union of certain cells of the sign-invariant partition induced on $\mathbf{R}^3$ by the set $F \cup H$.

*Work supported in part by ONR Contract N00014-90-J-1599, by NSF Grant CCR 86-19817, and by NSF Grant ECD 88-03017.

[1] We define $\mathrm{sgn}(x) = 1$ if x is positive, $\mathrm{sgn}(0) = 0$, and $\mathrm{sgn}(x) = -1$ if x is negative.

We discuss the origins of this problem in geometric modeling and what work has been done to solve it.

2 Problem Origins in Geometric Modeling

The objective of geometric modeling is to devise theories and algorithms that permit representing, manipulating, and analyzing geometric shapes by computer. In particular, solid modeling restricts attention to solid geometric objects in 3-space.

2.1 Solid Representations

Two major styles of representing 3-dimensional solid objects have emerged in geometric modeling, namely the surface-based *boundary representation* (Brep), and the volume-based *constructive solid geometry* (CSG). For an introduction see, e.g., [5]. A solid S must satisfy the

> *Boundary Separation Property:*
> Let S be a solid in $\mathbf{R}^3$. Then the boundary of S, written ∂S, is such that its complement consists of two disconnected sets called the *interior* and the *exterior* of S.

Note that neither the interior nor the exterior of S are required to be connected.

In Brep, the surface of a solid is described by a data structure that specifies *vertices*, *edges*, and *faces*, as well as their adjacencies and incidences. Roughly speaking, a face is a bounded area on a surface, usually connected; an edge is a (connected) segment of a line or a space curve; and a vertex is a point in 3-space. Breps are preferred when rendering the solid.

In CSG, the solid is specified as a (finite) expression formed from certain *primitives*, for example blocks, cylinders, spheres and cones, parameterized by size, radius, etc., using operations for positioning the primitives with respect to each other and the *regularized Boolean operations* of union, intersection, and difference. In contrast to the set-theoretic Boolean operations of union, intersection and difference, the regularized Boolean operations result in strictly homogeneous 3-dimensional objects. Regularization can be thought of as the closure of the interior. For example, the regularized union of solids A and B, written $A \cup^* B$, is obtained as the closure of the interior of the set $A \cup B$. Note that solids need not be connected sets.

CSG representations allow conceptually simple algorithms for the point-solid query, that is, for testing whether a point in 3-space is inside, on the surface of, or outside a solid.

Although Breps have become the dominant representation, it remains a problem of interest to convert between CSG and Brep representations, in part because the point-solid query is simpler in a CSG representation than in a Brep.

Conversion between the representations is also of foundational interest, and the algorithmic structure of the conversion from Brep to CSG is related to the search-tree approaches to the point-solid query.

2.2 Geometric Coverage

The surfaces that are allowed for the faces, and the permitted curves for the edges, are referred to as *geometric coverage*. In the simplest case, faces must be planar and edges must be line segments. The resulting solids are then polyhedra. More sophisticated modeling systems allow quadratic surfaces and their intersection curves. Other systems allow integral and rational parametric surfaces and their intersections; e.g., [4]; as well as implicit algebraic surfaces and their intersection, without degree restrictions; [1].

There are also procedural surfaces by which is meant a computational process for evaluating whether a point is on the surface, or a computational procedure for generating the surface, [3].

2.3 Two Geometric Modeling Problems

We restrict attention to solid modeling and assume that the faces are bounded, smooth areas on (irreducible) implicit algebraic surfaces. It is then clear that there are algorithms that can, in principle, convert any CSG representation into an equivalent Brep. The opposite algorithmic conversion problem, converting from Brep to CSG, is solvable on theoretical grounds using cylindrical algebraic decomposition; [2, 11]. In practice it is an unsolved problem that is closely related to the separability problem. We formulate it as

> *Problem 1:*
> Given a Brep solid, possibly with restricted geometric coverage, find an equivalent CSG representation.

As we will see, the point-solid query for Breps, formulated here as Problem 2, can be approached in a way that also makes it closely related to the separability problem and to Problem 1.

> *Problem 2:*
> Given a Brep solid and a point in 3-space, decide whether the point is inside, on the surface, or outside the solid.

Note that Problem 2 is customarily approached in the following way [14]: Consider a ray from the given point, into a random direction. Intersect the ray with the boundary of the solid, and determine the winding number of the point, properly counting nontransversal intersection. Then the classification of the given point is determined by the winding number. Note that the winding-number approach to Problem 2 does not establish a relationship with Problem 1. We explain a different approach that does in the next section.

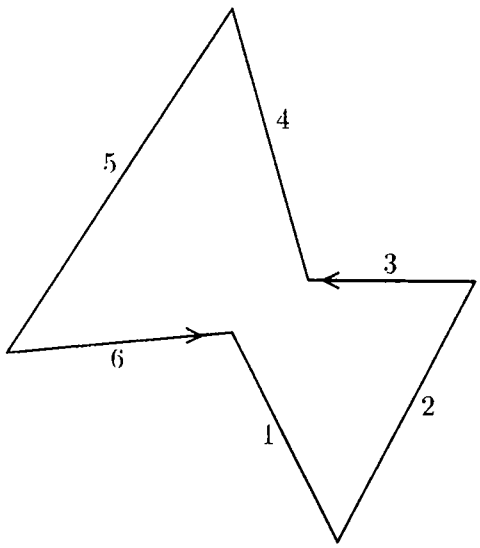

Figure 1: Polygon with oriented edges

3 The Linear Case

We introduce the BSP tree, a data structure developed in computer graphics, that can be used for solving Problem 2 in the case of polyhedra, [7, 8, 9]. Using the BSP tree, we will give a solution for Problems 1 and 2, for polyhedra. The idea is best explained in the 2-dimensional version of Problem 2, testing whether a point is in a polygon.

Consider the polygon shown in Figure 1. Its six edges have been oriented so that the interior of the polygon, the finite region delimited by the edges, is locally to the left. We describe a BSP tree associated with this polygon and how it is used to solve Problem 2. The tree is not unique and depends on the way in which the edges are chosen when the tree is constructed.

Consider the (consistently oriented) line L_1 on which edge 1 lies. L_1 is said to *support* edge 1. Let $f_1 = 0$ be the implicit equation of L_1, adjusted such that $f_1(p) > 0$ for all points p that lie to the right of L_1. We consider a tree whose root is labeled with the equation f_1. If p is inside the polygon and $f_1(p) > 0$, then it must lie in the triangle T_1 delimited by L_1, the edge 6, and a portion of edge 5.

We now consider the line L_6 supporting edge 6, with implicit equation $f_6 = 0$

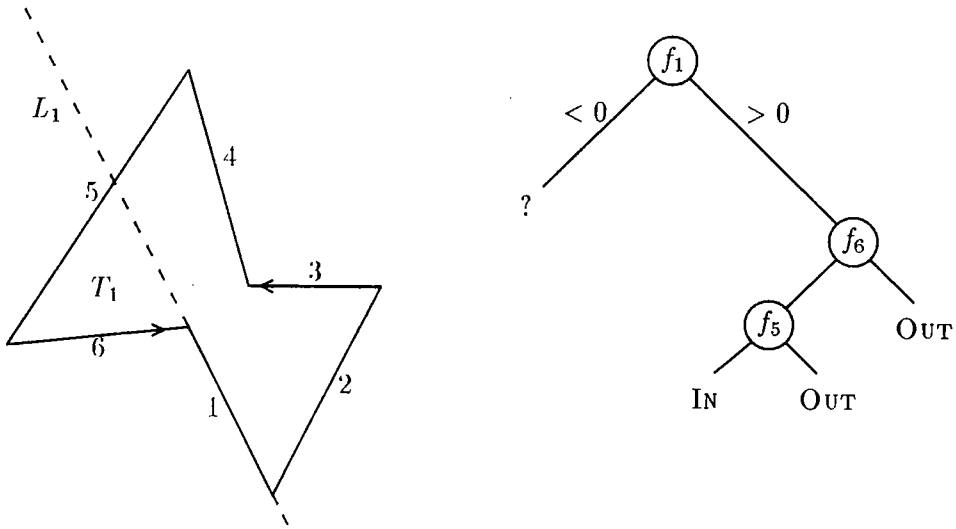

Figure 2: Partial BSP tree for the polygon of figure 1

adjusted in the same way as before, and thereafter consider the line L_5 supporting edge 5 with implicit equation $f_5 = 0$. If $f_1(p) > 0$ and $f_6(p) > 0$, then p is not in the polygon. If $f_1(p) > 0$ and $f_6(p) < 0$, then p is in the polygon interior iff $f_5(p) < 0$. This fact can be recorded by the partial tree shown in Figure 2. In this tree, the nodes are labeled with the functions f_i of the supporting lines. We descend down the right branch if the value of the function, evaluated at the point, is positive. The leaf labels shown indicate whether the point is ultimately outside or inside the polygon.

If $f_1(p) < 0$, then an evaluation of the equation of L_3 determines if the point should be searched in the triangle T_2 or in the area T_3, as shown in Figure 3. In case $f_3(p) > 0$, we have to determine the sign of $f_2(p)$. In case $f_3(p) < 0$ we must determine the sign of $f_4(p)$ and, if needed, the sign of $f_5(p)$. The final tree is shown in Figure 4.

We have sketched how to use the BSP tree of a polygon for testing whether a point is inside or outside the polygon. We have not discussed how to react when a value $f_i(p)$ is zero, e.g., when p is on the "surface" of the polygon. For these details see [6, 8].

Before proceeding, note that the implicit functions f_i induce a partition of the plane and, in particular, of the polygon. It is incidental that the cells of this partition are convex, but it is important that the cells are sign-invariant with respect to the f_i supporting the edges. In fact, the sign-invariant partition induced by the linear functions f_i consists only of connected cells, a rather special case.

Since the $f_i = 0$ are the "surfaces" supporting the "faces" of the polygon, the

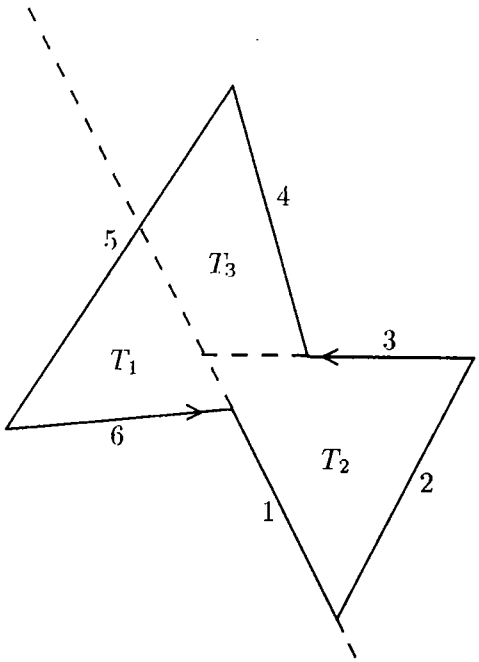

Figure 3: Polygon areas

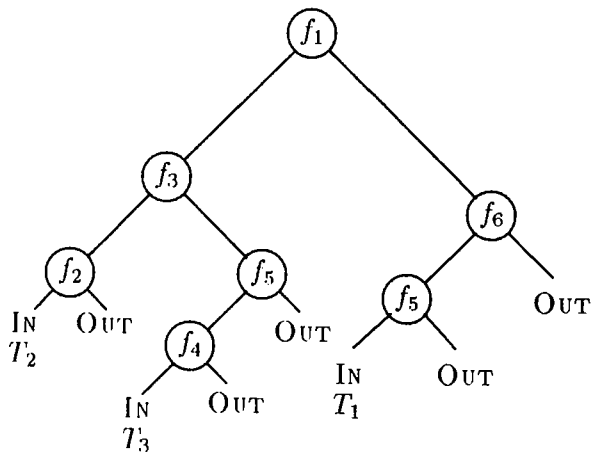

Figure 4: BSP tree of the polygon of figure 1

cell partition is compatible with the polygon, that is, the polygon is the union of the closure of certain cells of the sign-invariant partition, and this fact implies that we can give a CSG expression representing the polygon. To construct it, we read out the labels along each path in the BSP tree leading to a leaf labeled IN, negating the node label if the path continues to the right.

In abuse of notation, we denote by f_i the point set $\{p|f_i(p) \leq 0\}$. Similarly, $\neg f_i$ denotes the point set $\{p|f_i(p) \geq 0\}$. The CSG expression is then

$$T_1 \cup T_2 \cup T_3 \quad \text{where}$$

$$\begin{aligned} T_1 &= \neg f_1 \cap f_6 \cap f_5 \\ T_2 &= f_1 \cap f_3 \cap f_2 \\ T_3 &= f_1 \cap \neg f_3 \cap f_5 \cap f_4 \end{aligned}$$

Note that the expressions T_1, T_2 and T_3 denote closed sets.

It remains to discuss how the BSP tree looks for polyhedra, and how it is constructed. In the case of polyhedra in 3-space, the interior tree nodes are labeled with the implicit equation of the planes supporting the faces of the polyhedron. The leaves are labeled either IN or OUT.

Sketching the intuition, the tree is determined recursively as follows. A face is picked, and the intersection of the support plane with the polyhedral surface is constructed. The support plane equation labels the tree root, with the left and right subtrees denoting the halfspaces below and above the plane, assuming the faces and support planes follow standard conventions. The surface to each side of the plane is then considered separately, each giving rise to a subtree of the root. For details see [8, 9]. A derivative of the BSP tree, called the Brep index, has been used very effectively as the underlying data structure for a collision test in a dynamic simulator; [15].

We summarize this section as follows, connecting the material to the separation problem stated at the outset of the paper:

> *Theorem*
> The planes supporting the faces of a polyhedron, given in boundary representation, induce a cell partition of $\mathbf{R}^3$ that has the property that every 3-dimensional cell of the partition is connected. There is an algorithm that constructs, from the Brep and the implicit equations of the planes, a search tree structure that allows efficient solutions to both Problems 1 and 2.

4 The Nonlinear Case

Sign partitions induced by sets of linear functions always have connected cells. In the nonlinear case this is no longer true. Consider Figure 5. The points in the

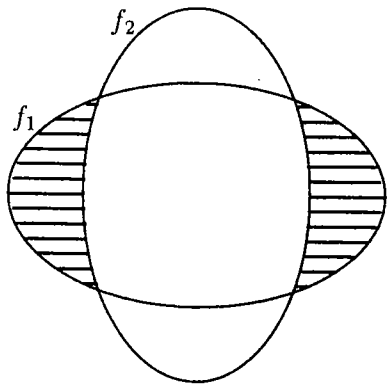

Figure 5: A disconnected cell $f_1 \cap \neg f_2$ of the sign-invariant partition

two shaded regions have the same sign vector, $(-1, +1)$, and therefore belong to the same equivalence class. They can be separated by an auxiliary line h_1, as shown in Figure 6. Now the connected components of $f_1 \cap \neg f_2$ are separately described by $f_1 \cap \neg f_2 \cap h_1$ and by $f_1 \cap \neg f_2 \cap \neg h_1$. Note that the cell $\neg f_1 \cap f_2$ is also disconnected. That cell can be separated by another line h_2 or by placing h_1 as shown in Figure 7.

The separability problem arises in solid modeling when considering Problem 1, as seen in Figure 8 The solid in the figure has a boundary that is delimited by an arc and four lines. It cannot be described solely in terms of the four lines bounding the rectangle and the circle, because the solid is not composed of complete equivalence classes induced by the corrsponding implicit functions.

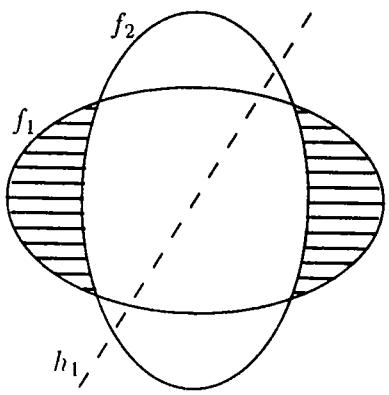

Figure 6: Line h_1 separates the components of $f_1 \cap \neg f_2$

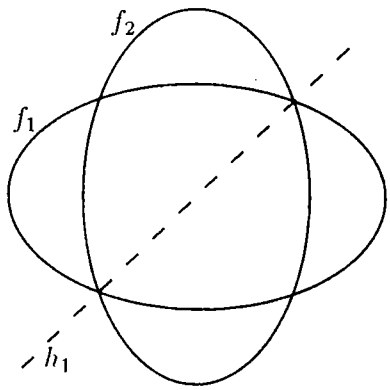

Figure 7: Line h_1 separates the components of both $f_1 \cap \neg f_2$ and of $\neg f_1 \cap f_2$

At least one additional function is needed, and the dashed line of Figure 9 will do. This line is a chord, and it can be proved that in the 2-dimensional version of the separability problem chordal lines suffice [12, 10].

In 3-space, the following theorem from [10] is easy to prove:

> *Theorem*
> Let S be a solid in $\mathbf{R}^3$ whose faces are supported by the implicit functions $F = \{f_1, ..., f_n\}$. If every face Q_i of ∂S is separated from the rest of the supporting surface $f_i = 0$ by a set of planes defined by the functions $G_i = \{g_1, g_2, ..., g_k\}$, then the exterior of S is separated from the interior of S by the functions of $F \cup (\cup_i G_i)$.

Intuitively, consider a line segment connecting an exterior point a and an interior point b. Then this line segment intersects ∂S an odd number of times, by the boundary separation properties of solids. So there is a face Q_i that is intersected an odd number of times. If both a and b are in the same cell of the sign-invariant partition induced by F, then the line segment $\overline{(a, b)}$ intersects the surface $f_i = 0$ an even number of times. So, we can find a point a' that intersects $f_i = 0$ but

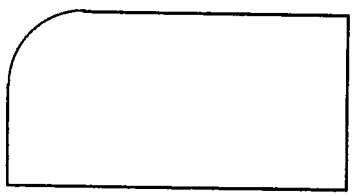

Figure 8: A solid requiring auxiliary functions

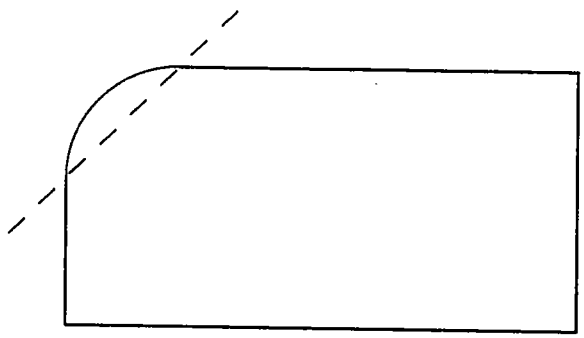

Figure 9: An auxiliary chord to describe the solid

not Q_i and a point b' that intersects Q_i such that the line segment $\overline{(a', b')}$ does not intersect $f_i = 0$ anywhere else. By assumption, a' and b' are separated by the planes in G_i. Since a line and a plane intersect in at most one point, the planes also separate a and b.

In particular, if S is a solid with quadratic faces and only planar edges, then it can be proved that sets G_i always exist, and that they can be found algorithmically [13]. Based on this insight, algorithms can be devised for Problems 1 and 2 and the separability problem can be solved algorithmically.

5 Summary of Known Results

We have explained the solution of the separability problem in the linear case and a partial solution of the problem in the nonlinear case. Moreover, it should be clear that an efficient solution of the separability problem also provides us with an efficient solution for Problems 1 and 2, two important problems in solid modeling whose current solutions are incomplete and leave room for improvement. Table 1, reproduced from [13], summarizes the known results on the separability problem.

References

[1] A. Bowyer, J. H. Davenport, D. A. Lavender, A geometric algebra system. In D. Kapur, editor, *Integration of Symbolic and Numeric Methods.* MIT Press, 1991.

[2] G. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conf. Automata Theory and Formal Languages*, Springer Lecture Notes in CS, 33, 1975.

Halfspace Domain	Halfspace Convexity	Halfspace connected?	Edge type	Linear separators?	Construction known?
linear	convex	yes	line segments	none required	—
quadrics	convex	yes	planar	yes	yes
			nonplanar	yes	no
	nonconvex	one sheet	planar	yes	yes
			nonplanar	yes	no
		two sheets	planar	yes	yes
			nonplanar	yes	no
general	convex	yes	planar	yes	yes
			nonplanar	no	no
	nonconvex	no	planar	yes	limited
			nonplanar	no	no

Table 1: Known Results in Separation, [13]

[3] N. Dyn. Subdivision schemes in CAGD. In W. Light, editor, *Advances in Numerical Analysis*, pages 36–104. Oxford University Press, 1991.

[4] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, 1988.

[5] C. M. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, Cal., 1989.

[6] C. M. Hoffmann and G. Vaněček. Fundamental techniques for geometric and solid modeling. In C. T. Leondes, editor, *Advances in Control and Dynamics, 48*, pages 101–165. Academic Press, 1991.

[7] B. Naylor. *A Priori Based Techniques for Determining Visibility Priority for 3-D Scenes.* PhD thesis, University of Texas at Dallas, 1981.

[8] B. Naylor. Binary space partitioning trees as an alternative representation of polytopes. *Computer Aided Design*, 22, 1990.

[9] B. Naylor, J. Amanatides, and W. Thibault. Merging BSP trees yields polyhedral set operations. *ACM Computer Graphics*, 24:115–124, 1990.

[10] V. Shapiro. *Representations of Semialgebraic Sets.* PhD thesis, Cornell University, Department of Mechanical Engineering, 1991.

[11] V. Shapiro and D. Vossler. Brep to csg conversion i: representations. Technical Report CPA89-3a, Cornell University, Mechanical and Aerospace Engineering, 1990.

[12] V. Shapiro and D. Vossler. Brep to csg conversion ii: solids. Technical Report CPA89-4a, Cornell University, Mechanical and Aerospace Engineering, 1990.

[13] V. Shapiro and D. Vossler. Boundary-based separation for Brep to CSG conversion. Manuscript, 1991.

[14] R. B. Tilove. Set membership classification: A unified approach to geometric intersection problems. *IEEE Transactions on Computers*, C-29:874–883, 1980.

[15] G. Vaněček Jr. A data structure for analyzing collisions of moving objects, 1991.