

Implicit Functions With Guaranteed Differential Properties

Vadim Shapiro Igor Tsukanov
Spatial Automation Laboratory †
University of Wisconsin-Madison

Abstract

Theory of R -functions [12] provides the methodology for constructing exact implicit functions for any semi-analytic set. This paper systematically explores and compares the known constructions in terms of their differential properties and explains how such functions may be constructed automatically from CSG and boundary representations of solids. The constructed functions may be automatically differentiated and integrated and have many important applications in mesh-free engineering analysis, motion planning, and scientific visualization.

1 INTRODUCTION

1.1 Applications of Implicit Functions

It is well known that every solid can be represented by a real-valued function f , such that $f > 0$ for all points in the interior, $f = 0$ for all points on the boundary, and $f < 0$ in the exterior of the represented solid. Such functions were used to approximate geometric shapes in [8, 3], but the theory of R -functions [11, 12, 15] gives an algorithmic method for constructing functions that *exactly* represent virtually any geometric shape of interest in engineering. Specific constructions for solid modeling and computer graphics are described in numerous references, for example see [16, 6, 4].

In its simplest and most commonly used form, a function f serves as a characteristic function for the point set in the sense that its sign can be used to distinguish points belonging to the set from those points that are not in the set. For instance, one can assume that $f \geq 0$ is true for points in the set, and $f < 0$ for all other points. This relaxed representation facilitates development of many popular algorithms in computer graphics and visualization [4], but allowing implicit functions with interior zeros¹ complicates algorithms in solid modeling applications [17].

[†]Complete address: 1513 University Avenue, Madison, WI 53706, USA
vshapiro@engr.wisc.edu, igor@sal-cnc.me.wisc.edu

¹By interior zeros we mean interior points of the set where the function takes on zero values.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Fifth Symposium on Solid Modeling Ann Arbor MI
Copyright ACM 1999 1-58113-080-5/99/06...\$5.00

Implicit function representations of geometric domains also have many other uses in engineering and scientific computations, provided that they possess additional mathematical properties. For example, it may be desirable that the absolute value of $f(p)$ estimates the distance from p to the boundary of the geometric pointset. Such functions have obvious uses in planning a motion that avoids an object's boundaries [5]. By definition, such a function f is not differentiable at all points that are equidistant from the boundaries. A Euclidean distance function f is often called the *normal* function for the object, and the set of points where f is not differentiable corresponds precisely to the points on the medial axis [1] (Figure 2(a)).

Differential properties of the implicit defining functions are critical in many applications. Solutions to many engineering and scientific problems may be formulated in terms of level sets of such differentiable functions [14]. In robotics and vision applications f may be used to define a potential field, which implies that f should be at least twice differentiable everywhere [5, 1]. When used to construct the solutions to boundary value problems, an implicit function may additionally be required to be *normalized* to m -th order. This means that along the direction \bar{v} normal to the boundary

$$\frac{\partial f}{\partial \bar{v}} = 1; \quad \frac{\partial^k f}{\partial \bar{v}^k} = 0; \quad k = 2, 3, \dots, m \quad (1)$$

Intuitively, this condition implies that f approximates the distance function to the m -th order near the boundary of the object. Normalization is required in order to specify Neumann and mixed boundary conditions [12, p.319], but it is also useful in constructing empirical models of various physical phenomena [10]. Finally, many algorithms in computer graphics and visualization rely on differential properties of normalized implicit functions, for example to improve the speed and the quality of rendering algorithms [2, 26].

Ability to automatically construct implicit functions with guaranteed differential properties — from any traditional solid modeling representations — paves the way towards implementing the above techniques within the existing computer-aided design and solid modeling environments.

1.2 Focus on Differential Properties

From the above (necessarily brief) description of applications, it should be apparent that at least four distinct properties of implicit functions require further study: elimination of interior zeros, distance properties, differentiability, and normalization. This paper focuses on the last three issues. We will examine the known properties of R -functions and show that systematic application of R -functions allows constructing implicit functions with desired and

guaranteed differential properties that are suitable for variety of applications. For discussion of automatic elimination of interior zeros the reader is referred to [17].

It should also be clear that not all of the mentioned properties may be achieved simultaneously. Figures 2 and 3 show three distinct functions and their gradients for the same two-dimensional shape in Figure 1. It is apparent that the distance function in Figure 2(a) is not differentiable along the curves of points that are equidistant from the boundaries; by contrast the function in Figure 2(b) is differentiable everywhere, including all the corner points, but it is not normalized; finally the function in Figure 2(c) is normalized everywhere except the corner points and appears to be a good approximation of the distance function near the boundary of domain.

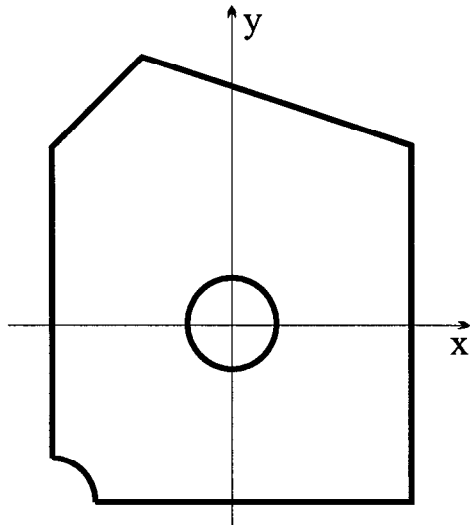


Figure 1: A two-dimensional solid domain

We shall see that such functions may be built automatically for arbitrary (solid and non-solid) semi-analytic sets.

1.3 Outline

Differential properties of several popular systems of R -functions are summarized in Section 2. In Section 3, we show how these properties arise in the context of CSG representations, or more generally, Boolean combinations of halfspaces. Section 4 extends the study of differential properties to implicit functions for trimmed curves and surfaces. Experimental results, including automatic differentiation and comparison of functions constructed from boundary and CSG representations are described in Section 5. The concluding section summarizes the significance of our findings and suggests several possibilities for further investigations.

2 DIFFERENTIAL PROPERTIES of R- FUNCTIONS

2.1 Systems of R -functions

A real valued function $f(x_1, x_2, \dots, x_n)$ is called an R -function if its sign is completely determined by the signs of its arguments x_i . In other words, f works as a Boolean switching function, changing its sign only when its arguments change their signs. For a detailed account of the theory the reader is referred to [12]; a brief

English introduction to R -functions is available as a Cornell technical report[15].

The main utility of the theory comes from a fact that every formal logical sentence has a corresponding class of R -functions, whose signs are essentially determined by the truth table of the logical sentence. Just as any logical function can be written using only three operations \neg , \wedge , and \vee , every R -function may be written as a composition of the three corresponding R -functions. For nonzero arguments, operation \neg is usually accomplished by changing the sign of the R -function. Thus, without loss of generality, we focus our attention on the differential properties of the binary R -functions corresponding to the logical disjunction \vee and conjunction \wedge .

It should be clear from the preceding remarks that R -functions are not unique and exhibit a rich variety of differential properties. The following are some of the more popular systems of R -functions that have been identified by the founder of the theory, V. L. Rvachev[12] and used in various applications:

$$R_\alpha: \frac{1}{1+\alpha} (x + y \pm \sqrt{x^2 + y^2 - 2\alpha xy}),$$

where $\alpha(f_1, f_2)$ is an arbitrary symmetric function such that $-1 < \alpha(f_1, f_2) \leq 1$.

$$R_0^m: (x + y \pm \sqrt{x^2 + y^2})(x^2 + y^2)^{\frac{m}{2}},$$

where m is any even positive integer.

$$R_p: f = x + y \pm (x^p + y^p)^{\frac{1}{p}},$$

for any even positive integer p .

In each case, choosing the (+/-) sign determines the type of R -function: (+) corresponds to R -disjunction and (-) sign gives the corresponding R -conjunction of the real-valued arguments x and y . We now briefly summarize the known differential properties of these R -functions. Additional details can be found in [12, 15].

2.2 Properties of R_α

Setting $\alpha = 1$, we see that this most popular class of R -functions includes the two simplest R -functions: R -conjunction $\min(x, y)$ and R -disjunction $\max(x, y)$. Both are continuous functions, and are fully normalized along the straight lines $x = 0$ and $y = 0$, but unfortunately, neither function is differentiable along the line $x = y$. This significantly limits use of these R -functions, and in fact Ricci functions [8] are constructed as smooth approximations to min and max.

Another interesting special case is $\alpha = 0$. In this case we get a system of R_0 -functions:

$$f = x + y \pm \sqrt{x^2 + y^2} \quad (2)$$

Differentiating, we get

$$\frac{\partial f}{\partial x} = 1 \pm \frac{x}{\sqrt{x^2 + y^2}}; \quad \frac{\partial f}{\partial y} = 1 \pm \frac{y}{\sqrt{x^2 + y^2}}, \quad (3)$$

which indicates that the functions are normalized to the first order whenever either $x = 0$ or $y = 0$. These functions are not differentiable in the corner point $x = y = 0$, but they are in fact analytic at any other point. For clarity, let us rewrite the partial in polar coordinates with $x = r \cos \varphi$ and $y = r \sin \varphi$, and consider the directional derivative with respect to direction \vec{l} in the neighborhood of the corner point $(0, 0)$:

$$\frac{\partial f}{\partial x} = 1 \pm \cos \varphi; \quad \frac{\partial f}{\partial y} = 1 \pm \sin \varphi; \quad (4)$$

$$\frac{\partial f}{\partial l} = \frac{\partial f}{\partial x} \cos \varphi + \frac{\partial f}{\partial y} \sin \varphi = \cos \varphi + \sin \varphi \pm 1$$

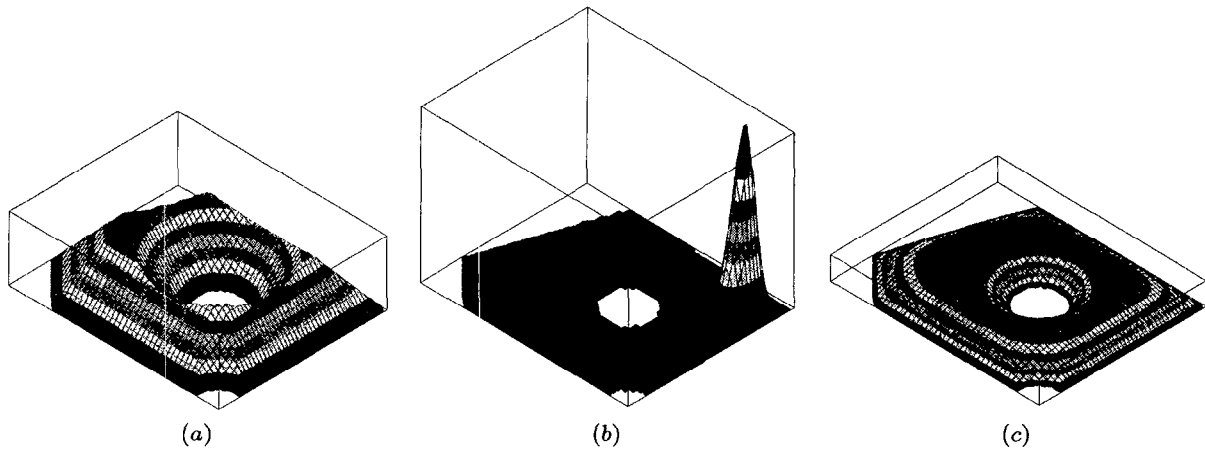


Figure 2: Different types of functions for two-dimensional domain shown in Figure 1: (a) Euclidean distance function; (b) C^2 function ; (c) function normalized to the first order

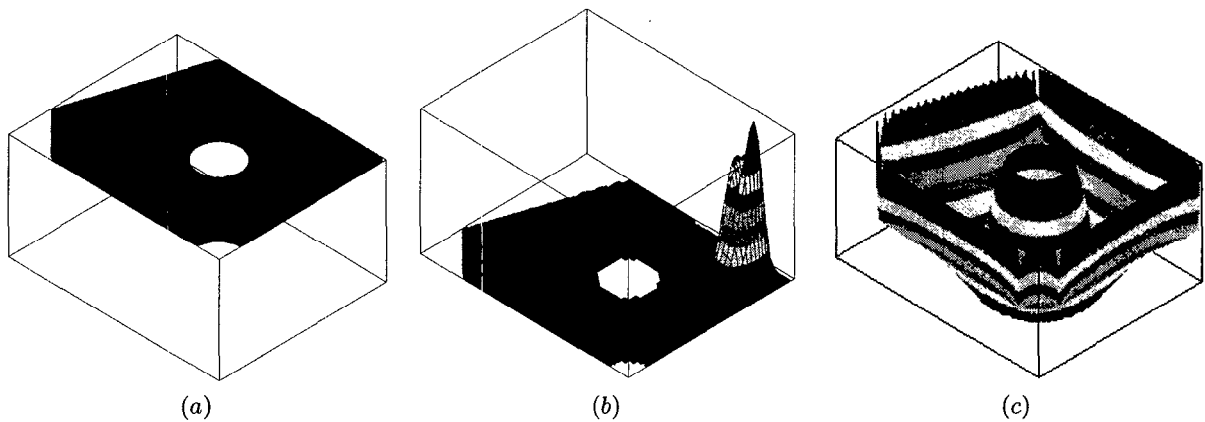


Figure 3: Absolute value of gradient of (a) distance function; (b) two times differentiable function; (c) normalized function

Thus, it is clear that the value of the directional derivatives in the corner depends on the direction φ of the approach to the corner point, but the magnitude of the gradient remains bounded:

$$(\nabla f)^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 = 3 \pm 2(\cos \varphi + \sin \varphi),$$

and is equal to 1 in the directions perpendicular to x and y axes as follows:

$$\begin{aligned} \text{Conjunction : } (\nabla f)^2|_{\varphi=0} &= (\nabla f)^2|_{\varphi=\frac{\pi}{2}} = 1; \\ \text{Disjunction : } (\nabla f)^2|_{\varphi=\pi} &= (\nabla f)^2|_{\varphi=-\frac{\pi}{2}} = 1 \end{aligned} \quad (5)$$

Figure 6(a) shows the computed plots of the gradients for both functions in the neighborhood of the origin.

Sometimes it may be advantageous to choose α to be a function; for some situations described in Section 3.3, Rvachev [12] proposed to choose α in the form

$$\alpha = \frac{1}{1 + x^2 + y^2} \quad (6)$$

It is easy to check that the resulting R_α -conjunction and R_α -disjunction with α given by (6) are also analytic everywhere except

the origin. Unfortunately, gradients of these functions may exhibit rapid oscillations. In particular, the plot in Figure 7 for the gradient of R_α -conjunction suggests a near-singular behavior in the direction of line $x = y$ in the neighborhood of the origin.

2.3 Properties of R_0^m and R_p

The main purpose of the R_m^0 system of R -functions is to remedy the loss of differential properties in the corner points $x = y = 0$. Multiplying R_0 -functions by a factor $(x^2 + y^2)^{\frac{m}{2}}$, we can rewrite them in polar coordinates as

$$f = r^{m+1}(\cos \varphi + \sin \varphi \pm 1)$$

Differentiating, we see that the function is m times differentiable everywhere, including the corner point $x = y = 0$ where all partial derivatives are identically zero.

$$\begin{aligned} \frac{\partial f}{\partial l} &= \frac{\partial f}{\partial r}|_{r=0} = \\ &= (m+1)r^m(\cos \varphi + \sin \varphi \pm 1)|_{r=0} = 0 \end{aligned} \quad (7)$$

The drawback of these R -functions is that, in contrast to the functions R_0 , they are not normalized.

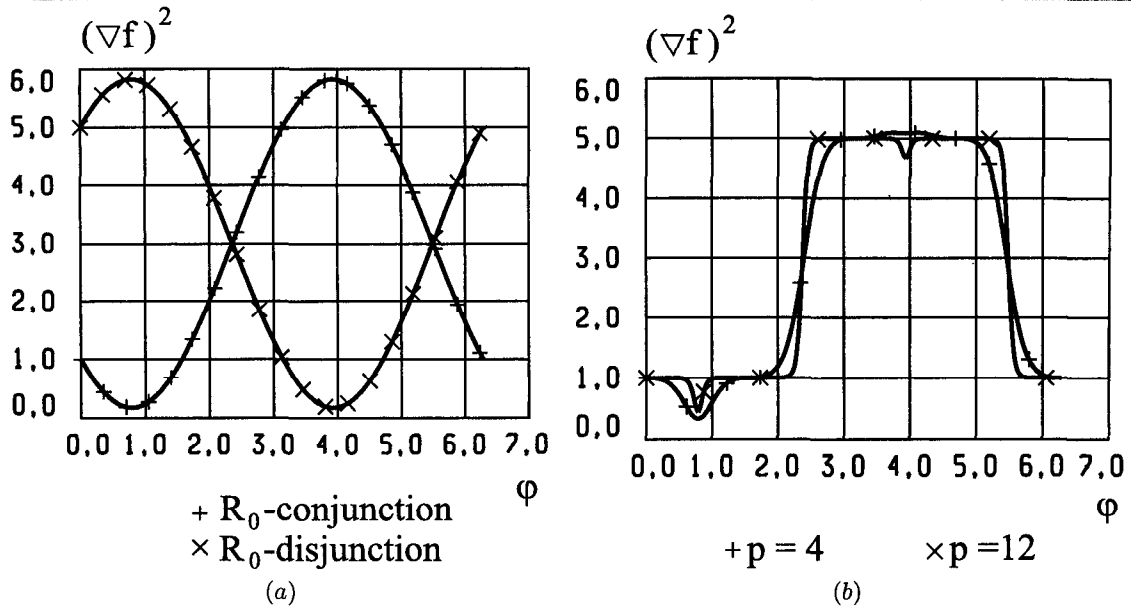


Figure 6: Square of gradient for R -functions in the neighborhood of the origin: (a) R_0 -functions, (b) R_p -conjunction

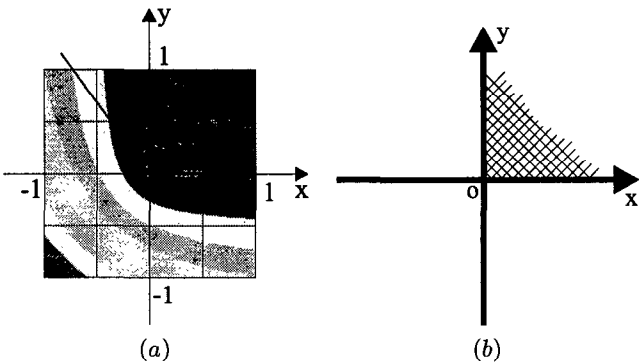


Figure 4: (a) Isolines of R_0 -conjunction; and (b) intersection of two orthogonal linear halfspaces

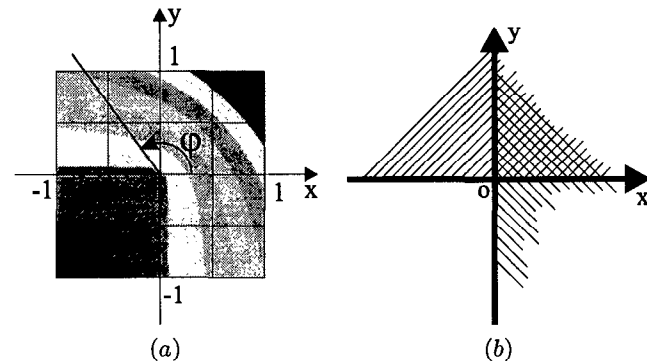


Figure 5: (a) Isolines of R_0 -disjunction; and (b) union of two orthogonal linear halfspaces

A system of R_p -functions is another generalization of R_0 -functions that achieves normalization to order $p - 1$. In this case,

$$\frac{\partial f}{\partial x} = 1 \pm \frac{x^{p-1}}{(x^p + y^p)^{\frac{p-1}{p}}}; \quad (8)$$

$$\frac{\partial f}{\partial y} = 1 \pm \frac{y^{p-1}}{(x^p + y^p)^{\frac{p-1}{p}}}$$

Changing to polar coordinates as before,

$$\begin{aligned} (\nabla f)^2 &= \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \\ &= \left(1 \pm \frac{\cos^{p-1} \varphi}{(\cos^p \varphi + \sin^p \varphi)^{\frac{p-1}{p}}}\right)^2 \\ &\quad + \left(1 \pm \frac{\sin^{p-1} \varphi}{(\cos^p \varphi + \sin^p \varphi)^{\frac{p-1}{p}}}\right)^2 \end{aligned} \quad (9)$$

we see that the magnitude of derivatives in the neighborhood of the corner point remains bounded in all directions, and the gradient has unit magnitude in the same directions indicated by equations (5) (Figure 6(b)). Similar calculations verify that higher order derivatives up to order $p - 1$ vanish whenever either $x = 0$ or $y = 0$. The R_p -functions are differentiable everywhere, except the corner point.

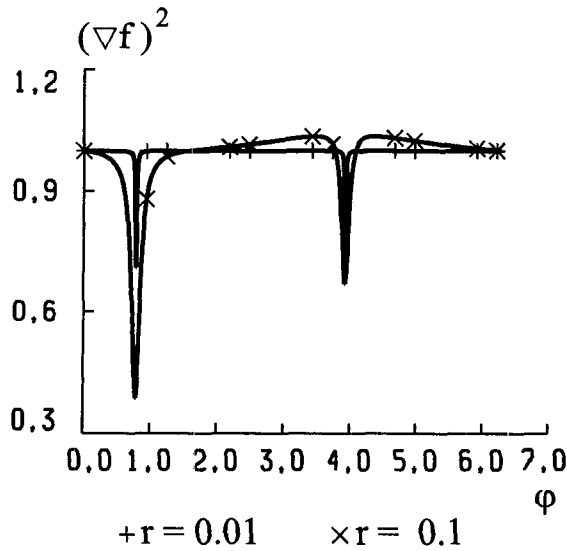


Figure 7: Square of gradient of R_α -conjunction with $\alpha = \frac{1}{1+x^2+y^2}$ in the neighborhood of the origin

3 COMPOSITION OF HALFSACES

3.1 From Set Operations to R -functions

In the context of geometric modeling, logical operations of \wedge and \vee correspond to the standard (non-regularized) set operations on *halfspaces*. If $f_1 \geq 0$ and $f_2 \geq 0$ are two primitive halfspaces, then they can be combined using any R -functions; we will denote then using their logical counterparts (\wedge and \vee) from now on. The inequalities $f_1 \wedge f_2 \geq 0$ and $f_1 \vee f_2 \geq 0$ respectively define the sets of points corresponding to the intersection and union of the two primitive inequalities. Thus, given any Boolean set expression combining primitives defined by inequalities, a simple syntactic substitution results in a single implicit function and inequality defining the composite region.

Strictly speaking, the same procedure does not apply to CSG representations that rely on *regularized* set operations. Regularization has the effect of removing from the set all those points where the composite function is zero but whose neighborhood does not contain any points in the solid's interior. Furthermore, syntactic translation of a CSG into a single inequality as described above may produce zero points in the interior of the solid, which may be unacceptable for some applications [17, 12]. Neither of these problems arise if the CSG expressions are *well-formed* as defined in [17]. Well-formed CSG representations exist for every solid, arise naturally, and may be constructed automatically. We do not deal with the issue of well-formedness in this paper.

Based on the material in [12], differential properties of the composite functions constructed from well-formed CSG expressions are determined by differential properties of CSG primitives and the choice of R -functions.

3.2 Linear Orthogonal Halfspaces

Consider the simplest situation when $f_1 = x$ and $f_2 = y$. Then the inequalities

$$x \wedge y \geq 0; \quad \text{and} \quad x \vee y \geq 0$$

define respectively the regions of intersection and union of the two orthogonal linear halfplanes as illustrated in Figures 4(b) and 5(b). The differential properties of R -functions imply that in both cases the constructed functions are differentiable everywhere in and on the boundary of the two domains, except the corner point. Furthermore, since both $x = 0$ and $y = 0$ are normal equations of the coordinate axis, normalization of the composite function is completely determined by the properties of the R -functions \wedge and \vee .

In the case of intersection, the resulting region is convex, and the gradient in the neighborhood of the convex corner point does not exceed unity; whereas in the case of the union, the gradient is significantly higher in the concave neighborhood defined by the reflex angle (Figure 3).

3.3 General Smooth Halfspaces

Differential properties of functions of the form $f(h_1, h_2)$, where f is any R -function and $h_1(x, y)$ and $h_2(x, y)$ are arbitrary smooth normalized functions, can be observed through a simple coordinate transformation:

$$x \rightarrow h_1(x, y); \quad y \rightarrow h_2(x, y);$$

Then the partial derivatives of the composite function are given by the Jacobian transformation:

$$\begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_2}{\partial x} \\ \frac{\partial h_1}{\partial y} & \frac{\partial h_2}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial h_1} \\ \frac{\partial f}{\partial h_2} \end{pmatrix} \quad (10)$$

where the partial derivatives $\frac{\partial f}{\partial h_i}$ of R -function f depends on the chosen system of R -functions. Thus, the magnitude of the gradient of $f(x, y)$ at any point (x, y) is a scalar multiple of the gradient of the corresponding R -function $f(h_1, h_2)$. The transformation (10) preserves normalization of the function f on boundaries of halfspaces $h_1 = 0$ and $h_2 = 0$. For example, if $h_1 = 0$, then we have $\frac{\partial f}{\partial h_1} = 1$, $\frac{\partial f}{\partial h_2} = 0$, and consequently $(\nabla f)_{|h_1=0}^2 = \left(\frac{\partial f}{\partial h_1} \frac{\partial h_1}{\partial x} + \frac{\partial f}{\partial h_2} \frac{\partial h_2}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial h_1} \frac{\partial h_1}{\partial y} + \frac{\partial f}{\partial h_2} \frac{\partial h_2}{\partial y}\right)^2 = \left(\frac{\partial h_1}{\partial x}\right)^2 + \left(\frac{\partial h_1}{\partial y}\right)^2$. Because h_1 is normalized on the line $h_1 = 0$, we get $(\nabla f)_{|h_1=0}^2 = 1$. Results are similar for $h_2 = 0$.

Other differential properties of R -functions are also inherited by the composite functions. In particular, if f is an R_0 -function or R_p -function, then the composite function is analytic everywhere except the corner points, and the normalization of h_1 and h_2 is preserved to the corresponding order on all points where only one of $h_i = 0$. The normalization property of R_0 -functions breaks down when the two halfspaces are identical, i.e. $h_1 \equiv h_2$, but is preserved by R_α -functions with α defined by equation (6). Similarly, if f is an R_0^m -function, then the composite function is m times differentiable everywhere, including all corner points, but is not normalized.

In a more general situation, the number of halfspaces h_i need not be limited to two. Suppose the boundaries of n halfspaces $\{h_i\}$ intersect at a single point p and are combined by some R -function $f(h_1, h_2, \dots, h_n)$ to define a planar region S . If S is homogeneous in dimension in the neighborhood of p , then some even number of boundaries ∂h_i form the boundary of S , while the rest of them do not contribute (see Figure 8). Then the properties of R -functions imply that those halfspaces that do *not* contribute to the boundaries of the region S have no effect on the differential properties and magnitudes of derivatives at regular points of the boundary of S [12, pp.128-137].

The generalization to higher dimensions is straightforward. When a number of three-dimensional halfspaces $\{h_i\}$ are combined by an R -function, then the smoothness and normalization

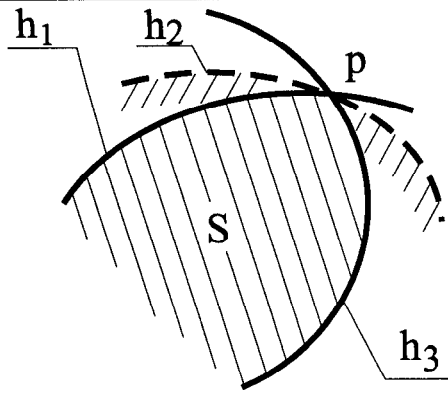


Figure 8: Halfspace h_2 does not contribute to the boundary of the domain and does not affect the differential properties of the composite function

properties are assured by the choice of the corresponding R -function on all regular points of the halfspace boundaries. If two or more surfaces (boundaries of halfspaces) intersect along some curve C , then taking a normal planar section at every point of C yields the two-dimensional situation described above. The behavior of the composite function is naturally extended to the vertex v defined by intersection of three or more surfaces, where it is either m differentiable in the case of an R_0^m -function, or in the case of R_0 - and R_p -functions, has discontinuities in derivatives with bounded magnitude depending on the direction in the neighbourhood of v .

Example. The solid in Figure 9 is defined as the intersection of five halfspaces: two cylindrical, one conical, and two planar. To construct a normalized function for the solid, we apply the R_0 intersection to the normalized functions for each primitive halfspace:

- Cylindrical:

$$f_1 = \frac{1 - x^2 - y^2}{2}; \quad (11)$$

$$f_2 = -\frac{0.25^2 - x^2 - y^2}{0.5}; \quad (12)$$

- Planar:

$$f_3 = z; \quad (13)$$

$$f_4 = \frac{-2(x-1) + 2(y-1) - 4(z-1)}{2\sqrt{6}} \quad (14)$$

- Conical:

$$f_5 = -\frac{z - \sqrt{x^2 + y^2}}{\sqrt{2}} \quad (15)$$

The plots for the composite function R_0 -operations:

$$\omega = f_1 \wedge_0 f_2 \wedge_0 f_3 \wedge_0 f_4 \wedge_0 f_5 \quad (16)$$

sectioned at $z = 0.5$ and $x = 0$ are shown in Figures 9(b) and (c) respectively.

4 TRIMMED CURVES AND SURFACES

4.1 Simple Trimming

Every trimmed curve or surface can be constructed by some sequence of set operations on unbounded primitives, which immediately suggests that they too can be described by implicit functions using the theory of R -functions. First we need to describe an infinite curve or surface. The traditional representation by the equality $h(p) = 0$ is not acceptable, because only inequalities of the same type are suitable arguments for R -functions. If curve C is a boundary of halfspace $h(x, y) \geq 0$, then it can also be defined as

$$(h \geq 0) \cap (-h \geq 0), \quad \text{or} \quad (h \wedge -h) \geq 0.$$

However, examination of the resulting expressions for derivatives shows that *any* choice of R -function \wedge (except R_0^m) will result in a function that is not differentiable at *all* points of the curve. We can also rewrite the curve as an inequality $-h^2(p) \geq 0$, but this function is not normalized, even if h is. We can achieve normalization by defining the curve as $-\sqrt{h^2} \geq 0$ but again this function is not differentiable on all points of the unbounded curve.

The above difficulties are bypassed by first using R -functions to construct the inequality for the *trimmed* entity, and then normalizing it. Consider the construction for a semi-infinite curve as an unbounded curve $h(x, y) = 0$ trimmed by a halfspace $x \geq 0$:

$$(-h^2 \geq 0) \cap (x \geq 0) \quad \text{or} \quad f = (-h^2) \wedge x \geq 0. \quad (17)$$

The composite function f is negative everywhere except the points on the trimmed curve where it is zero. For systems R_1 , R_0 , and R_p , it is analytic everywhere except the origin, where the magnitudes and directional properties of discontinuities of derivatives can be estimated using the methods described in Sections 2 and 3. The function is not normalized, and its normal derivatives are zero on all points along the curve. However, when h is normalized, it is shown in [12, p.195] and [24, p.206] that the normalized composite function f_1 can be obtained as:

$$f_1 = \sqrt{-f} = \sqrt{-((-h^2) \wedge x)}. \quad (18)$$

It should be clear that $-f_1 \geq 0$ defines the same set of points because it preserves the zero and the negative points of f . The function defined by (18) is not differentiable on the points of the trimmed curve, but is analytic everywhere else. Furthermore, f_1 is normalized on both sides of the curve along the opposite normal vectors \bar{n} and \bar{v} (Figure 10).

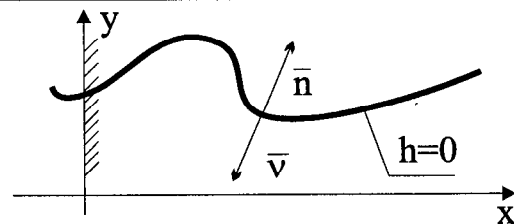


Figure 10: Opposite normal vectors of trimmed curve

4.2 Normalized Trimming

Because trimming relies on R -conjunction, any resulting function will exhibit rapid changes in gradient magnitude near the end point. Moreover, the square root in expression (18) implies that $\frac{\partial f_1}{\partial x} \rightarrow \infty$

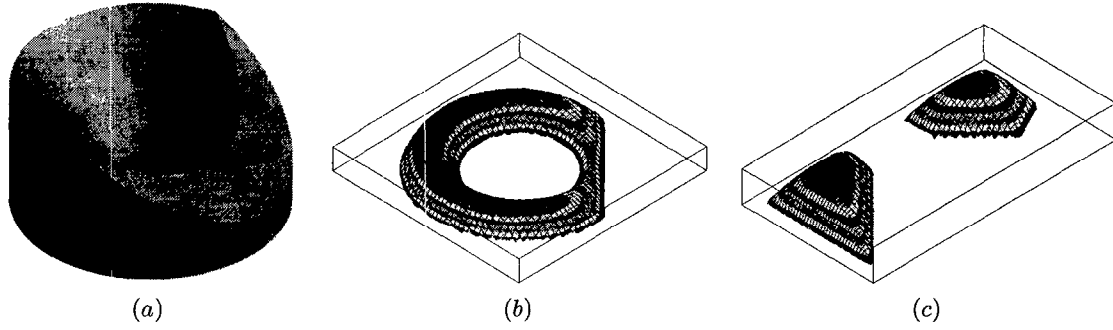


Figure 9: Solid object(a) and composite function $f(x, y, z)$ plots in (b) section $z = 0.5$, (c) section $x = 0$

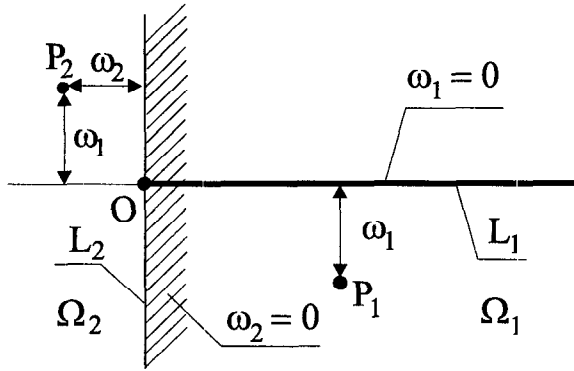


Figure 11: Normalized trimming of a curve

as $x \rightarrow 0$ when approaching the end point from outside of the trimming halfspace. This type of anomaly can be observed in isolines — see Figure 12(a) — of a function f_1 defining the positive portion of the x -axis (i.e. $h = y$). Rvachev proposed a technique that normalizes the resulting function in all normal² directions near the end point, which we will call 'normalized trimming'. Consider two functions ω_1 and ω_2 in the neighborhood of the point, such that ω_1 gives distance from any point P to the line L_1 ; ω_2 is identically zero in region Ω_1 and behaves as a distance function to the line L_2 in Ω_2 . Here, L_2 is the local boundary of the trimming halfspace and L_1 is the curve being trimmed. Then the function

$$\omega = \sqrt{\omega_1^2 + \omega_2^2} \quad (19)$$

defines the Euclidean distance from any point to the trimmed halfline [12]. If functions ω_1 and ω_2 are normalized to the first order then the function defined by (19) is normalized to the first order. If a function ω_2^* that describes the trimming region by the inequality ($\omega_2^* \geq 0$) is constructed using R -functions, then it is easy to construct the desired function ω_2 :

$$\omega_2 = \frac{|\omega_2^*| - \omega_2^*}{2}$$

Substituting this expression into (19) we obtain:

$$\omega = \sqrt{\omega_1^2 + \frac{(|\omega_2^*| - \omega_2^*)^2}{4}} \quad (20)$$

²For a point $x \in S$ direction \vec{v} is normal if there exists a point $p \notin S$ such that p is closer to x than other points of S [12].

The isolines of ω defining the positive x -axis, with $\omega_1 = y$ and $\omega_2^* = x$ are shown in Figure 12(b). Though not visible in the plot, this construction leads to discontinuous second derivatives on the line where $\omega_2^* = 0$. This can be remedied by further modifying the construction for ω_2 as suggested in [12]:

$$\omega_2 = \frac{\sqrt{(\omega_2^*)^2 + \omega_1^4} - \omega_2^*}{2}$$

Then expression (19) can be rewritten as follows:

$$\omega = \sqrt{\omega_1^2 + \frac{\left(\sqrt{(\omega_2^*)^2 + \omega_1^4} - \omega_2^*\right)^2}{4}} \quad (21)$$

which gives the implicit function for the trimmed curve that is normalized in any normal direction in the neighborhood of the end point of the curve. The isolines of this function are shown in Figure 12(c).

Normalized trimming guarantees the normalization only if the lines L_1 and L_2 are orthogonal at their intersection point; otherwise expression (19) no longer gives the Euclidean distance to the point O (Figure 11). For example, Figure 13(a) shows the loss of normalization at the end point when the x -axis is trimmed by a linear halfspace inclined at 45° . When normalization is important, it can be restored at non-orthogonal intersections by orthogonalizing of lines L_1 and L_2 at their intersection point. For example, replacing function ω_2^* with

$$\omega_2^{**} = \omega_2^* - \omega_1 \nabla \omega_2^* \cdot \nabla \omega_1 \quad (22)$$

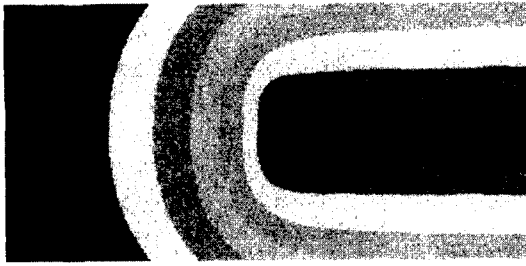
guarantees normalization even for non-orthogonal intersections [22, 12], as is illustrated in Figure 13(b).

4.3 General Trimming

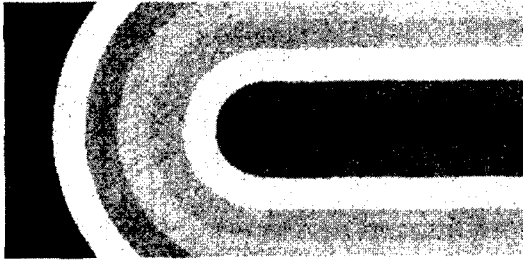
All of the above constructions generalize in a straightforward fashion. For example, let $h(p) = 0$ be an equality describing an unbounded entity (curve or surface), and let $\phi(p) \geq 0$ define the "trimming" region containing some portion of the unbounded entity. Then the trimmed entity is given by the intersection of the unbounded entity and the trimming region, and is defined by the inequality

$$f = -\sqrt{-((-h^2) \wedge \phi)} \geq 0. \quad (23)$$

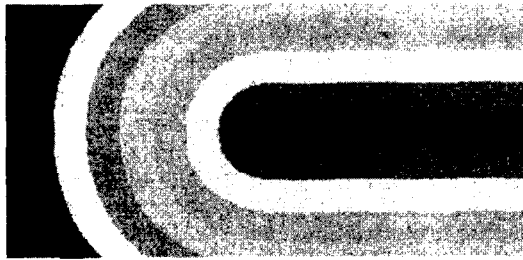
The composite function f is zero on points of the trimmed entity, and is strictly negative everywhere else. Normalized trimming (possibly with orthogonalization) may be also applied.



(a)



(b)



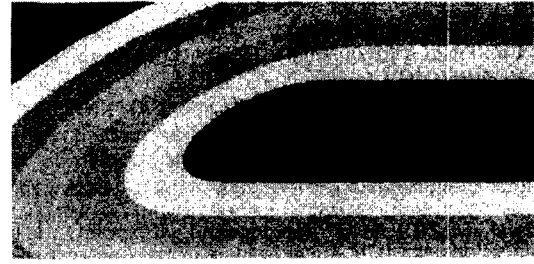
(c)

Figure 12: Isolines of functions for the trimmed halfline shown in Figure 11: (a) simple trimming; (b) normalized trimming with 2nd order discontinuity; (c) normalized smooth trimming

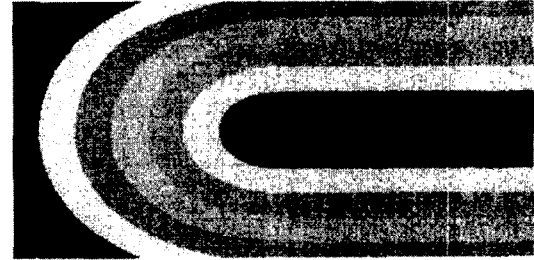
The differential properties of the composite function are determined by the differential properties of h and ϕ and the trimming method. In particular, if h and ϕ are both smooth and normalized, then f is also smooth everywhere and normalized on all points of the trimmed entity, except the "border" points lying on the boundary of the trimming region. The notion of a trimming region is not new; for example [9] uses a similar concept in order to construct Boolean expressions for faces in the boundary representations. Notice that a Boolean representation of the trimming region can be trivially translated into a single inequality $\phi \geq 0$ by syntactic substitution of R -functions for the Boolean set operations. The differential properties of ϕ may be determined as described in Section 3.

Example. Let us construct an implicit function for the cylindrical face of the solid in Figure 9. The unbounded cylindrical surface is given by equation (11). The trimming region in this case is the intersection of the halfspaces (two planar and one conical); it is unbounded, but its restriction to a box that contains the solid is shown in Figure 15. The function defining the trimming region is constructed as

$$\phi = f_3 \wedge f_4 \wedge f_5, \quad (24)$$



(a)



(b)

Figure 13: (a) The influence of non-orthogonality of lines L_1 and L_2 at point O (Figure 11); (b) The result of orthogonalization (22)

and the implicit function for the trimmed cylindrical face (Figure 16(a)) is obtained by applying the expression (23) to function f_1 defined in (11) and function ϕ defined in (24). Figures 16(b) and (c) plot the resulting composite function in sections $z = 0.5$, and $x = 0$.

5 Experimental Results

We implemented all of the above construction techniques in the SAGE (Semi-Analytic Geometry Engine) computational environment that is being developed by the authors at the University of Wisconsin, and we experimented with variety of composite functions automatically constructed from boundary and CSG representations.

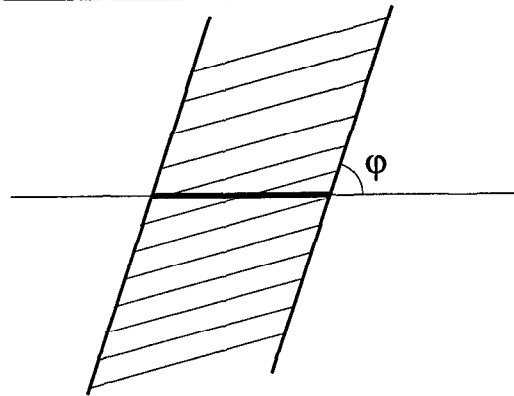


Figure 14: Angle between boundary of trimming region and trimmed line.

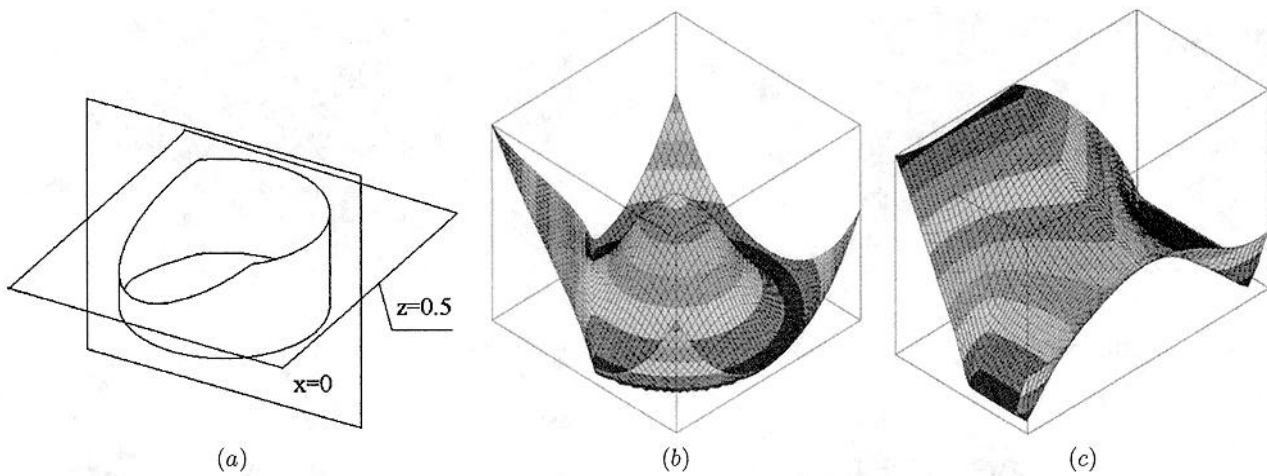


Figure 16: Trimmed cylindrical face (a) and plots of composite function in sections (b) $z = 0.5$, (c) $x = 0$

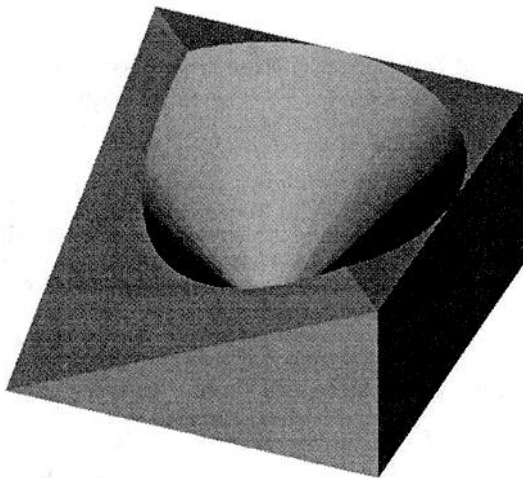


Figure 15: The solid trim region for the cylindrical face in Figure 16(a) restricted to a box enclosing the solid.

5.1 Boundary Representation vs CSG

Given a well-formed CSG representation, the corresponding implicit function is obtained by a syntactic substitution of the desired R-functions for the set operations in the CSG expression. The resulting function will inherit the properties of the CSG primitives and the R-functions. In particular, using RF-functions will ensure that the constructed functions are differentiable everywhere, including edges and vertices. If all primitives are represented by normalized functions and combined using R_p - or $\&$ -functions, the resulting composite function will also be normalized and analytic everywhere except the edges and vertices of the three-dimensional solid. Figure 17(a) shows the isolines of the composite function for Washington Island (Door County, Wisconsin) whose boundary consists of 613 linear edges. A well-formed CSG representation for this polygon was constructed automatically using the decreasing convex hull algorithm [17] with one linear halfspace for every edge in the boundary, and R_0 functions were applied in place of

the CSG operations. It took 232 seconds³ to construct this plot, by sampling the function on a 90×90 regular grid and bilinearly interpolating the sampled values.

A boundary representation can be converted to a CSG representation using the techniques described in [19, 20, 21]. But the desired implicit functions may be also constructed directly from the boundary representation. We already described the procedure for constructing an implicit function for an individual face as intersection of an unbounded surface and a trimming region. Since the boundary representation is a finite union of faces, the implicit function for the boundary of the solid may be constructed by “gluing” the individual face functions together, using R-functions. Differential, normalization, and other properties of the composite function may be further affected by this construction [10, 12], but can be controlled using methods summarized in this paper. For example, if the i th face of the b-rep is defined by an implicit function $\omega_i = 0$, $i = 1, 2, \dots, n$, then the function $w = \omega_1 \wedge \omega_2 \wedge \dots \wedge \omega_n$ is the implicit function defining the boundary of the solid. The resulting function will be zero everywhere on the boundary of the solid and strictly positive everywhere else; it does not distinguish between the interior and exterior of the solid, but this information is available from the boundary representation. In other words, if a PMC function $\xi(p)$ returns 1, 0, or -1 depending on whether point p is in, on, or out of the solid respectively, then the function $f = \xi w$ is the usual solid defining implicit function [16]. Figures 17(b) and (c) show two functions for the same polygon constructed directly from the boundary representation of Washington Island: using the simple trimming in Figure 17(b) and using the normalized trimming in Figure 17(c). By comparison, the functions constructed from the boundary representation are composed of twice as many primitives (because every linear edge was trimmed by a circular halfspace) and the total evaluation time took 397 (Figure 17(b)) and 367 (Figure 17(c)) seconds, including point membership classification for all points on the grid.

It should be clear that computing time grows linearly with the size of the constructed function. For example, it takes 11,400 seconds (3 hours and 10 minutes) to compute the composite function from the boundary representation of the entire state of Wisconsin (20,748 edges) on the same uniform rectangular grid 90×90 (Figure 18).

³All times were measured on a computer with dual 400 MHz Pentium II processors.

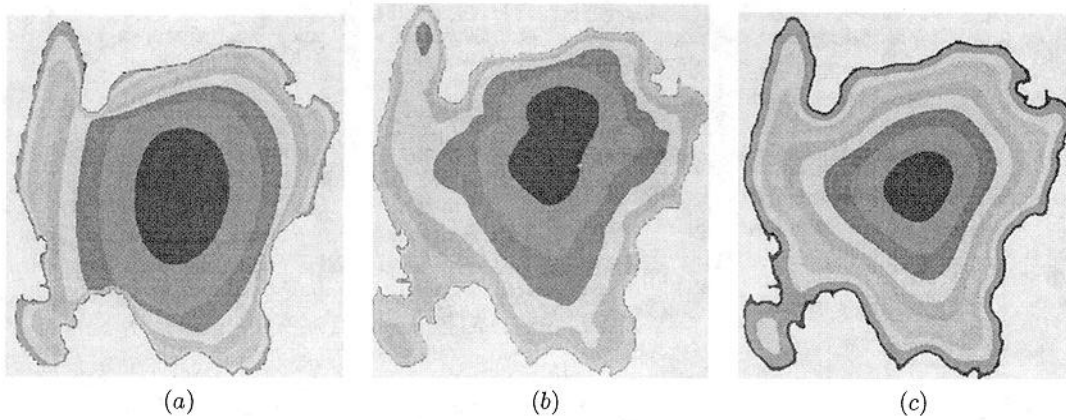


Figure 17: (a) Isolines of function derived from CSG; (b) isolines of function constructed from boundary representation using simple trimming for each edge; (c) isolines of function derived from boundary representation using normalized trimming for each edge

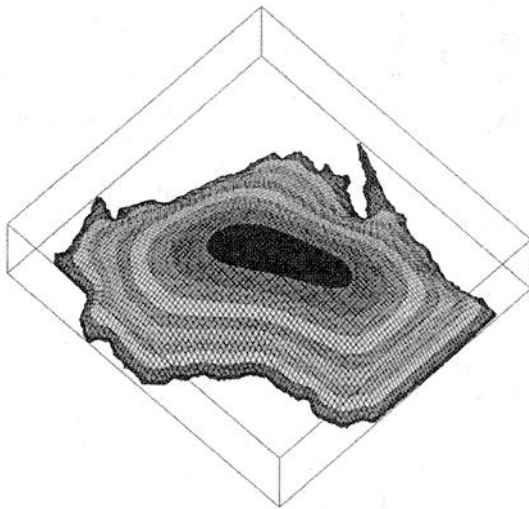


Figure 18: The composite function constructed from the boundary representation of the state of Wisconsin with 20,748 edges

5.2 Normalization and Global Properties

Notice that the functions constructed from a boundary representation appear to be qualitatively better than the corresponding functions constructed from CSG. This is explained by the fact that construction from boundary representation relies on R-conjunction applied to nonnegative arguments; recall that the gradient of R-conjunction fluctuates significantly in the neighborhood of the corner points in the zone where both arguments are negative. These local fluctuations tend to compound and propagate, particularly in the presence of small edges or faces. A similar qualitative difference is apparent in comparing the functions constructed from the boundary representation. Normalized trimming in Figure 17(c) results in a function that is smoother and is closer to the true distance function than the function in Figure 17(b) constructed using simple trimming.

More generally, the higher degree of normalization near the solid's boundary, the better the resulting function will approximate the distance function. Recall that $\&$ -functions can assure normalization to order $p - 1$. We observe that as p increases, the constructed composite functions approach the true distance function

near the regular points on the solid's boundary, and begin to resemble the distance function globally. For example, compare the function constructed using R_p -functions with $p = 12$ on normalized primitives in Figure 19 to the distance function in Figure 2(a); this smooth function is not quite the distance function, but there is an apparent correspondence between the set of points where functions have the maximum curvature and the medial axis for the shown domain.

Figure 19: The function constructed using R_p operations for $p = 12$

For another example of how normalization on the boundary may affect the global behavior, let us compare the simple and normalized trimming described in Section 4.3. Figures 20(a), (b), and (c) show the effect of changing the angle φ between the boundary of the trimming region and the line for every one of the four line segments in the boundary representation of the rectangle in the case of the simple trimming using equation (18). The test case geometry is illustrated in Figure 14. Recall that the purpose of normalized trimming is to assure that the function is normalized in all directions near the corner points. Figures 20(d), (e), and (f) show the corresponding functions for the same geometry using normalized (but not orthogonalized) trimming. Computations were performed for three values of φ : $\varphi = \pi/10$ (Figures 20(a) and (d)); $\varphi = \pi/4$ (Figures 20(b) and (e)); $\varphi = \pi/2$ (Figures 20(c) and (f)). As the angle φ increases, it should be obvious that normalized trimming contains local defects in normalization much better than the simple trimming method that could lead to functions such as in Figure 20(a).

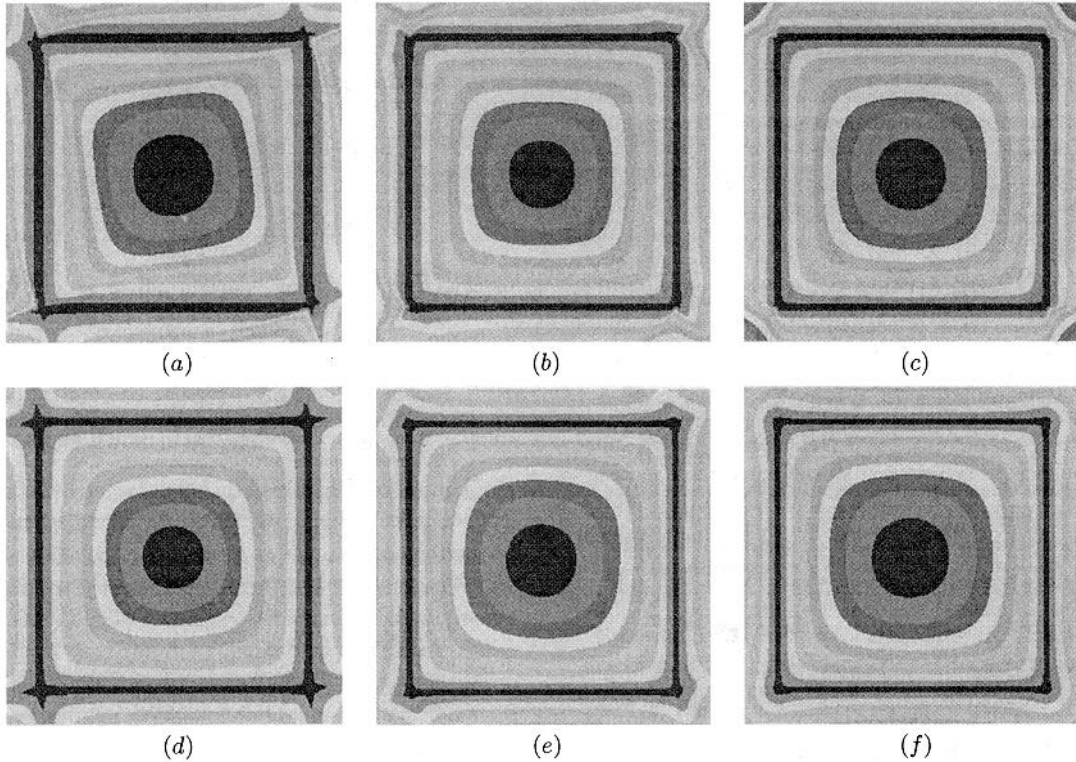


Figure 20: Influence of non-orthogonality of trimming region's boundary and trimmed line on composite function for union of trimmed lines

5.3 Automatic Differentiation

Differential properties of the constructed functions may be useful indirectly, for example to guarantee convergence or correctness of sampling algorithms. But many more important applications (solutions of differential equations, sensitivity analysis, control systems) require explicit computation of derivatives at various points in space. Several differentiation techniques are known:

Symbolic differentiation may be used when we have closed-form expressions for the function. Commercially available software can produce symbolic expressions for partial derivatives of a function with respect to independent variables. Once these expressions are computed, they may be evaluated at any point of interest. However, the symbolic computations are fairly intensive, and the symbolic expressions quickly become unmanageable in their size and complexity.

Numerical (approximate) differentiation uses either finite difference evaluation of derivatives or differentiation of the approximation of a given function by polynomials or splines. These techniques may be particularly useful when the functions are defined by (interpolation of) a discrete set of points. In some cases, numerical differentiation may become unstable or inaccurate.

Automatic differentiation is a term used to describe the process of exact differentiation that propagates values of derivatives (as opposed to expressions in symbolic differentiation). For a recent survey on automatic differentiation see [7].⁴ Of course, the

numerical automatic differentiation process has to be repeated at every new point of interest. The procedure uses forward application of the chain rule - from independent to dependent variables. Such recursive procedures have been derived for all elementary functions. In each case, given values of functions f_i and all their partial derivatives through order k , the procedure computes the values of all partial derivatives of the composite function that combines f_i . The recursive rules have been derived independently by a number of researchers [7, 13,23,25].

Our particular implementation of automatic differentiation is based on the procedure originally proposed in [13] and the formulation in [23] that places no restrictions on the number of independent variables and order of derivatives. Elementary and arithmetic operations have been defined as friend functions of a class *tuple* and overload the corresponding usual operators in C++. Each tuple contains the values of functions and all of its partial derivatives through the specified order at a single point. Overloading allows preservation of the usual notation for mathematical expressions. Independent variables are generated by function *Argument(tuple &x, int dim, double value)* that sets function's value to *value*, and the first partial derivative with respect to *dim*-th coordinate to unity. Other partial derivatives are zeros.

All pictures of gradients in this paper were produced using this code. The time to compute a tuple at a point grows linearly in the number of operations in the composite function, and exponentially in the number of independent variables and the order of derivatives.

⁴when, as in our case, the functions being differentiated are constructed at run time.

⁴Automatic differentiation is most commonly applied to a computer program and user-specified dependent and independent variables - to produce another program to compute the corresponding partial derivatives. But this technique is not very practical

6 CONCLUSIONS

The main contribution of this paper is the careful and systematic analysis and experimental studies of the constructive means offered by the theory of R -functions in the context of solid modeling representations. We conclude that implicit functions with guaranteed differential properties may be automatically constructed for any solid bounded by polynomial surfaces from the traditional solid modeling representations. It is not clear at this time how the described constructions may be extended to parametric surfaces. Automatic differentiation and integration of such functions can be carried out efficiently and robustly. Our results open new opportunities for scientific and engineering applications of solid modeling that can take advantage of such global functions. For example, we showed in [18] that such functions allow meshfree simulation of engineering problems with deforming domains.

Many other techniques and methods from the theory of R -functions remain to be investigated. For example, normalization of primitives or intermediate results may be difficult or computationally expensive, and it could be advantageous to perform normalization as the last step in the construction process. A technique described in [12] transforms any function f into a function f_1 normalized to the first order as follows:

$$f_1 = \frac{f}{\sqrt{f^2 + (\nabla f)^2}} \quad (25)$$

The computational implications of this procedure require further investigation. The “gluing” of implicit functions for individual faces may be considered as a trivial case of transfinite interpolation. Further generalizations of the known interpolation techniques based on the theory of R -function appear promising and useful in the context of numerous applications.

ACKNOWLEDGMENTS

The authors are grateful to Professors V. L. Rvachev and T. I. Sheiko for numerous discussions on the properties of R -functions. This work was supported in part by the National Science Foundation grants DMI-9502728, DMI-9522806 and CCR-9813507. All computations described in this paper (including generation of all images) were performed in the SAGE (Semi-Analytic Geometry Engine) computational environment.

REFERENCES

- [1] N. Ahuja and J.-H. Chuang. Shape representation using a generalized potential field model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):169–176, 1997.
- [2] C.L. Bajaj. Modelling physical fields for interrogative visualization. In T. Goodman and R. Martin, editors, *The Mathematics of Surfaces VII*. Information Geometers, 1997.
- [3] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.
- [4] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, 1997.
- [5] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [6] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: Concepts, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.
- [7] L. B. Rall and G. F. Corliss. An introduction to automatic differentiation. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation, Procs Second International Workshop on Computational Differentiation*. SIAM, 1996.
- [8] A Ricci. A constructive geometry for computer graphics. *Computer Journal*, 16(3):157–160, May 1973.
- [9] J. R. Rossignac. CSG formulations for identifying and for trimming faces of CSG models. In *CSG'96, Winchester, UK*. Information Geometers Ltd, April 1996.
- [10] V. Rvachev, T. Sheiko, V. Shapiro, and J. Uicker. Implicit function modeling of solidification in metal casting. *Transaction of ASME, Journal of Mechanical Design*, 119:466–473, December 1997.
- [11] V. L. Rvachev. *Geometric Applications of Logic Algebra*. Naukova Dumka, 1967. In Russian.
- [12] V. L. Rvachev. *Theory of R-functions and Some Applications*. Naukova Dumka, 1982. In Russian.
- [13] V. L. Rvachev and G. P. Manko. *Automation of Programming for Boundary Value Problems*. Naukova Dumka, 1983. In Russian.
- [14] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences*. Cambridge University Press, 1996.
- [15] V. Shapiro. Theory of R -functions and applications: A primer. Tech. Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY, 1991.
- [16] V. Shapiro. Real functions for representation of rigid solids. *Computer-Aided Geometric Design*, 11(2):153–175, 1994.
- [17] V. Shapiro. Well-formed set representations of solids. *International Journal on Computational Geometry and Applications*, 1997. accepted for publication.
- [18] V. Shapiro and I. Tsukanov. Meshfree simulation of deforming domains. Technical report sal 1998-2, Spatial Automation Laboratory, University of Wisconsin, June 1998. <http://sal-cnc.mc.wisc.edu/>.
- [19] V. Shapiro and D. L. Vossler. Construction and optimization of CSG representations. *Computer-Aided Design*, 23(1):4–20, January/February 1991.
- [20] V. Shapiro and D. L. Vossler. Efficient CSG representations of two-dimensional solids. *Transactions of ASME, Journal of Mechanical Design*, 113:292–305, September 1991.
- [21] V. Shapiro and D. L. Vossler. Separation for boundary to CSG conversion. *ACM Transactions on Graphics*, 12(1):35–55, January 1993.
- [22] T.I. Sheiko. On handling of singularities at corner points and at junctions of boundary conditions in R -functions method. *Applied Mechanics*, (4):95–102, 1982. In Russian.
- [23] A. N. Shevchenko and V. N. Rokityanska. To the problem of automatic differentiation of functions of several variables. *Cybernetics and System Analysis*, (5):1–20, 1996.
- [24] Protsenko V.S. Stoian, Yu. G. and G.P. Manko. *Theory of R-functions and Actual Problems of Applied Mathematics*. Naukova Dumka, 1986. In Russian.
- [25] G. P. Manko V. L. Rvachev and V. V. Fedko. Computer differentiation technique of functions of several variables. *Cybernetics*, (5):31–33, 1983.
- [26] L. Velho. Adaptive polygonization made simple. In *Proceedings of SIBGRAP'95*, 1995.