Acta Numerica (2007), pp. 1–65
DOI: 10.1017/S0962492902
© Cambridge University Press, 2007
Printed in the United Kingdom

Semi-Analytic Geometry with R-Functions

Vadim Shapiro
Mechanical Engineering & Computer Sciences,
University of Wisconsin - Madison, 1513 University Avenue,
Madison, Wisconsin 53706, USA
E-mail: vshapiro@engr.wisc.edu

To V. L. Rvachev (1926 – 2005)

V. L. Rvachev called R-functions "logically-charged functions" because they encode complete logical information within the standard setting of real analysis. He invented them in the 1960s as a means for unifying logic, geometry, and analysis within a common computational framework – in an effort to develop a new computationally effective language for modeling and solving boundary value problems. Over the last forty years, R-functions have been accepted as a valuable tool in computer graphics, geometric modeling, computational physics, and in many areas of engineering design, analysis, and optimization. Yet, many elements of the theory of R-functions continue to be rediscovered in different application areas and special situations. The purpose of this survey is to expose the key ideas and concepts behind the theory of R-functions, explain the utility of R-functions in a broad range of applications, and to discuss selected algorithmic issues arising in connection with their use.

CONTENTS

1 From Descartes to Rvachev 2
2 Functions for shapes with corners 3
3 R-functions 8
4 From inequalities to normalized functions 20
5 The inverse problem of analytic geometry 25
6 Geometric modeling 38
7 Boundary value problems 46
8 Conclusions 56
References 58

1. From Descartes to Rvachev

Descartes (1637) is usually credited with conceiving the coordinate method that allows *investigating* geometric objects by algebraic means. Centuries of remarkable progress in understanding and classifying local and global properties of analytic and algebraic representations followed, but the need for systematic *construction* of such representations did not materialize until the middle of the twentieth century. The arrival of the computer created the need to represent and manipulate sets of points, particularly (but not exclusively) in Euclidean three-dimensional space, for the purposes of visualization, animation, geometric design, analysis, simulation, and so on. How does one represent a set of points on a computer using their coordinates? There are really only two methods: by providing a rule for *generating* points in the set, or by designing a method for *testing* a point $\mathbf{p} \in R^n$ with known coordinates against some predicate that distinguishes the points in the set from other points. The first method assumes an ability to *parametrize* the point set (for example, as a spline, or a triangulation, subdivision scheme, or some other procedural definition); see (Farin, Hoschek and Kim 2002) for a recent update on progress in parametric modeling. The second method implies an ability to represent a geometric object Ω *implicitly* by a predicate $S(\mathbf{p})$, as

$$\Omega = \{\mathbf{p} : S(\mathbf{p}) \text{ is } true\}. \tag{1.1}$$

Parametrizations and predicates are well known for simple objects, such as lines, conic sections, and quadric surfaces, but not for the majority of useful geometric objects arising in the physical world and non-trivial modeling situations. Such objects require constructions to be carried out in a *piecewise* fashion, leading to non-trivial data structures and algorithms in geometric modeling and computational geometry. In this survey, we will not deal with parametric representations, but focus instead on the construction of implicit representations. The primitive form of such a representation is an equation $f(\mathbf{p}) = 0$ or an inequality $f(\mathbf{p}) \geq 0$. In this case, the predicate $S(\mathbf{p})$ is defined by the sign of $f(\mathbf{p})$, for example, to be true if $f(\mathbf{p}) = 0$ or $f(\mathbf{p}) \geq 0$, and false otherwise. For convenience of notation, we will use the predicates $(f(\mathbf{p}) = 0)$ or $(f(\mathbf{p}) \geq 0)$, respectively, to denote the set Ω of points for which the predicate $S(\mathbf{p})$ is true.

But how does one write an equation for a rectangle? V. L. Rvachev struggled with this question in the 1960s while using the method of Kantorovich (Kantorovich and Krylov 1958) to solve a boundary value problem of contact mechanics on a square domain. He finally came up with the equation for the boundary of a rectangle with sides $2a$ and $2b$, respectively, as:

$$a^2 + b^2 - x^2 - y^2 - \sqrt{(a^2 - x^2)^2 + (b^2 - y^2)^2} = 0. \tag{1.2}$$

However, he could not explain his own constructions using the classical

methods of analytic and algebraic geometry that focus on *direct* problems of investigating given equations and inequalities. In contrast, Rvachev wanted to devise a methodology for solving what he termed the *inverse problem of analytic geometry:* constructing equations and inequalities for given geometric objects. This quest resulted in his seminal publication, (Rvachev 1963), followed by the comprehensive *theory of R-functions* that has been developed over the last forty-plus years.[1] In a nutshell, R-functions operate on real-valued inequalities as differentiable logic operations; the resulting theory solves the inverse problem of analytic geometry and has a wide range of applications, with particular emphasis on solutions of boundary value problems. As of 2001, the bibliography on the theory of R-functions included more than fifteen monographs and over five hundred technical articles coauthored by Rvachev and his followers (Matsevitiy 2001). R-functions were introduced into the Western literature by the author (Shapiro 1988), and are now widely used in geometric modeling, computer graphics, robotics, engineering analysis, and other computational applications.

The goal of this paper is to expose the key concepts in the theory of R-functions, without trying to be comprehensive. This subject will take us through Section 4. Additional references in English are now accessible, notably the review by Rvachev and Sheiko (1995). Among the references in Russian, the monograph by Rvachev (1982) continues to serve as an encyclopedic source of many key ideas and results. The utility of R-functions cannot be fully appreciated without some discussion of how they may be used algorithmically to solve inverse problem of analytic geometry (Section 5). Sections 6 and 7 are devoted to applications of R-functions and derived constructions to computational tasks in geometric modeling and boundary value problems, respectively.

2. Functions for shapes with corners

2.1. Many equations of a rectangle

How *does* one write an equation for a rectangle? Serendipitously, it is not that difficult to come up with several methods, though none of them would directly yield expression 1.2. For example, recalling the definition of the L_p norm, we know that the equation $1 - |x|^n - |y|^n = 0$ describes a family of shapes for $n = 2, 3, \ldots$, that vary from the circle when $n = 2$ to the unit square as $n \to \infty$. Scaling x by $\alpha = 1/a$ and y by $\beta = 1/b$, we obtain the

[1] The origin of the term R-function is not entirely clear. The use of Roman (not Cyrillic!) italicized symbol R suggests that it is a mathematical symbol corresponding to the set of reals. A conflicting personal account by Rvachev (1996) suggests that R does stand for Rvachev, but that it was coined by his sister, also a noted Ukrainian mathematician, supposedly in memory of their father.

widely used equation of a superellipse:

$$1 - |\alpha x|^n - |\beta y|^n = 0 \tag{2.1}$$

For even n, this is even a polynomial equation since we can then drop the absolute value sign. Of course, this is only an approximation of the rectangle's boundary, since we can never quite get into its corners. Alternatively, if we take the limit, we end up with the L_∞ norm and the corresponding equation

$$1 - \max(|\alpha x|, |\beta y|) = 0, \tag{2.2}$$

which defines the exact desired boundary of the rectangle, even if we no longer have the nice analytic properties of a polynomial.

But what if we want equations for other shapes, say a polygon? A more general method for constructing such equations and inequalities would be needed, and it should also work for the rectangle of course. So let us consider the rectangle once more, and try to compose its equation from simpler primitive equations. Even for the rectangle, there are at least two distinct ways to do this.

First observe that the rectangle is the *intersection* of the vertical stripe $\Omega_1 = (a^2 - x^2 \geq 0)$ and horizontal stripe $\Omega_2 = (b^2 - y^2 \geq 0)$ defined by two inequalities. The corresponding equations define their respective boundaries (pairs of vertical or horizontal lines). Then the rectangle is defined by

$$(a^2 - x^2 \geq 0) \cap (b^2 - y^2 \geq 0), \quad \text{or} \quad (f = \min(a^2 - x^2,\ b^2 - y^2) \geq 0). \tag{2.3}$$

Furthermore, the equality $f = 0$ defines the boundary of the rectangle. Once again, the equation is not polynomial and we may not like the differential properties of the min function, but the construction itself is perfectly valid and generalizes to other shapes that can be defined by intersections of simpler shapes. In fact, we are only one step away from obtaining the expression in 1.2.

Alternatively, we observe that the *boundary* of the rectangle is the *union* of four line segments. *If* we could write an equation $f_i(x, y) = 0,\ i = 1, 2, 3, 4$, for each of the line segments, then the equation of the rectangle is obtained by a simple product as

$$f_1 f_2 f_3 f_4 = 0. \tag{2.4}$$

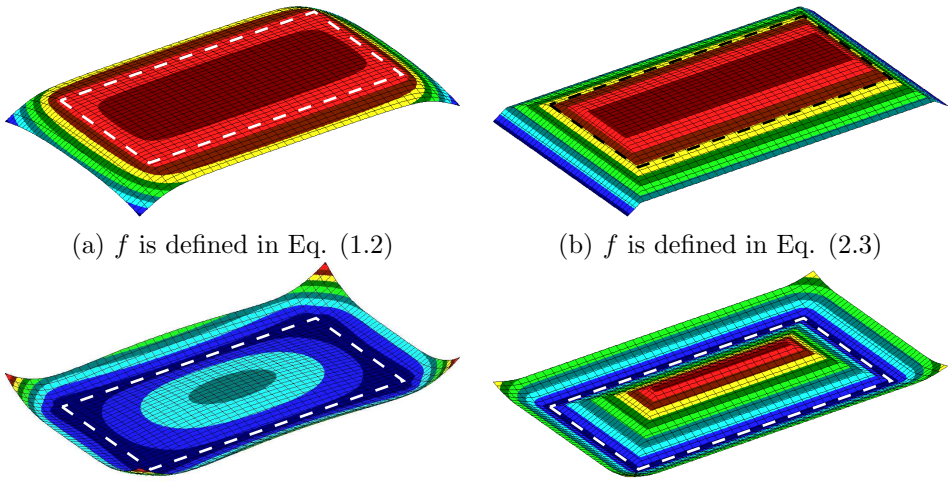
This construction generalizes to arbitrary polygonal boundaries, and requires only multiplication – provided, of course, that we find a method for writing an equation for each line segment. Perhaps, we could try to represent each line segment as the *intersection* of a line and a circular disk, but this would require using again the min operation, and so on.

Apparently there are many different ways to construct an equation $f(x, y) = 0$ for the rectangle's boundary. Four such functions corresponding to some of

the above constructions are plotted in Figure 2.1. The theory of R-functions explains, systematizes, and expands the above constructions to a virtually unlimited variety of shapes and functions. But for now, the above observations naturally raise two interrelated questions:

- What other *useful* point sets (shapes) admit similar solutions to the inverse problem of analytic geometry?
- Which types of functions are *possible* and *preferable* for such shapes?

The answers to these questions depend on what is meant by 'useful' and 'preferable', as we discuss next.



(a) f is defined in Eq. (1.2)

(b) f is defined in Eq. (2.3)

(c) f is defined in Eq. (2.4)

(d) f is the Euclidean distance

Figure 2.1. Four different implicit representations ($f = 0$) of the rectangle.

2.2. Useful shapes

If one is mostly interested in describing domains of boundary value problems (as Rvachev was), then it is reasonable to assume that the point sets in question should include their boundaries, i.e., be closed subsets of $\mathbb{R}^d$. For any such closed set Ω, there exists a suitable continuous function, namely the Euclidean distance function

$$d(\mathbf{p}) \equiv \inf_{\mathbf{x}\in\partial\mathbf{\Omega}} \|\mathbf{p}-\mathbf{x}\| \tag{2.5}$$

and, conversely, for every continuous function f, the equality $f(\mathbf{p}) = 0$ represents a closed subset of $\mathbb{R}^d$. For the rectangle, this function is plotted

in Figure 2.1(d). Furthermore, according to Whitney (1934), whenever it is possible to construct such a continuous function f, it is also possible to construct a C^∞ function that vanishes on the set of points. Specific constructions may be found in (Rvachev and Rvachev 1979), but this would take us well beyond the scope of the present survey; however, we shall see that constructing C^n functions is fairly straightforward with R-functions. We should not expect to do better than that in general, because arguments based on Taylor series expansion easily show that no such function can be analytic in the neighborhood of a corner of the square (Rvachev 1982). This limitation also applies to a wide variety of shapes in science and engineering whose boundaries are only piecewise smooth, i.e., composed in a piecewise manner from smooth curves and surfaces.

We now recognize that such shapes belong to the class of *semi-analytic sets*, defined as those sets that can be constructed as a finite Boolean combination (i.e., a finite composition of unions, intersections, and complements) of sets $(f_i \geq 0)$, where f_i are real analytic functions. Originally proposed by Lojasiewicz (1964) as a natural generalization of semi-algebraic sets, semi-analytic sets are now widely adopted as a proper setting for most geometric modeling applications (Requicha 1977, Requicha 1980, Shapiro 1994, Shapiro 2002), though semi-algebraic sets continue to dominate most practical implementations. Semi-analytic sets clearly include all algebraic and analytic varieties of the form $(f = 0)$, but our goal is to be able to construct functional representations for any closed semi-analytic set, without restrictions on its dimension, codimension, or other topological properties. Furthermore, the distinction between sets defined by inequalities $(f \geq 0)$ and equalities $(f = 0)$ is artificial. A point set Ω that is represented as $(f = 0)$ is also represented by $(-f^2 \geq 0)$; and whenever Ω is represented by $(g \geq 0)$, it is also represented by $(g - |g| = 0)$.

The characterization of useful shapes as semi-analytic sets points the way towards the constructive solution of the problem of inverse analytic geometry. Given any shape Ω, we may subdivide it into *primitive* 'pieces' for which the inverse problem is either trivial or has already been solved. The corners usually provide a good hint on where the shape should be subdivided. We can then combine these primitive solutions using the logical operations of $\wedge, \vee, \neg$ into a single predicate $S(\mathbf{p})$ that represents Ω. Three of the rectangle's equations were constructed using this idea: using intersection of vertical and horizontal stripes, using union of the line segments, and the very first equation (1.2), though the latter may not yet be obvious. On the other hand, introduction of logical (or equivalently set-theoretic) operations also leads to a conceptual difficulty. We no longer have a single real-valued inequality $f \geq 0$, but a logical predicate defining the set of points in Ω. In writing the equations for the square, we carefully translated the logical predicates into the corresponding real-valued function f, but to what end?

2.3. Preferable functions

If the function f is used only as a characteristic function that distinguishes points in the set Ω from all other points, then it really does not matter what f is, as long as it can be evaluated in a reasonably efficient manner. What else would we use the function f for?

Rvachev's original goal was to extend the method of Kantorovich for solving boundary value problems. Briefly, the method is a technique for constructing the coordinate basis functions satisfying the Dirichlet boundary conditions $u_{|\partial\Omega} = \varphi$. The idea of the method is based on the observation that in this case, the solution of a differential equation can be represented in the form

$$u = \varphi + \omega\Psi, \tag{2.6}$$

where $\omega : \mathbb{R}^n \to \mathbb{R}$ is a known function that takes on zero value on the boundary $\partial\Omega$ of the domain and is positive in the interior of Ω; Ψ is some function to be determined. Representing $\Psi = \sum_{i=1}^{n} C_i\chi_i$, as a linear combination of basis functions from some sufficiently complete space (polynomial, splines, etc.) reduces the original boundary value problem to that of determining the coefficients C_i that solve the corresponding variational problem. But it is not obvious how this method may be extended to solve boundary value problems with other types of boundary conditions or what the function ω should be in general. Rvachev recognized expression (2.6) as the beginning of a Taylor series expansion. In one dimension, expression (2.6) becomes

$$u = u(x_0) + (x - x_0)\Psi. \tag{2.7}$$

In other words, the function ω appears to play the role of the *distance* to the boundary, at least in the vicinity of the boundary $\partial\Omega$, where the boundary conditions are prescribed. Thus, the Kantorovich method may be viewed as a power series expansion of a field function u in powers of the distance ω to the boundary $\partial\Omega$; higher-order boundary conditions should be associated with powers of ω, and ω should be sufficiently smooth (Rvachev, Sheiko, Shapiro and Tsukanov 2000).

But there is another problem. If different boundary conditions are prescribed on different portions $\partial\Omega_i$ of the boundary, they must be somehow interpolated. In one dimension, the interpolation problem is well understood. If function values are specified at n points $x_1, \ldots, x_n$, the key to interpolation is to construct weight functions W_i that take on the value of 1 at x_i and are 0 at all other points $x_j, j \neq i$. For example, W_i can be taken as the Lagrange basis polynomial L_i written in a general form as

$$L_i(x) = k_i \prod_{j \neq i} (x - x_j), \tag{2.8}$$

where k_i is chosen so that $L_i(x_i) = 1$ (Lancaster and Salkauskas 1986).

Once again, we see that the key to constructing the weight appears to be the *distances* to the points x_j where the data is prescribed. It takes a bit of imagination to realize that the same technique should work in higher dimensions to *transfinitely* interpolate values prescribed on boundaries $\partial\Omega_i$, if every term $(x-x_j)$ is replaced by a function ω_j that measures the distance to the boundary portion $\partial\Omega_j$ (Rvachev, Sheiko, Shapiro and Tsukanov 2001).

Implicit representations and distance functions are also used widely for computer shape and solid modeling (Shapiro 1994, Pasko, Adzhiev, Sourin and Savchenko 1995, Bloomenthal 1997). In these applications, it is often assumed that $(\omega \geq 0)$ is a solid shape Ω, whose boundary $\partial\Omega$ is $(\omega = 0)$, and whose interior $\mathbf{i}\Omega$ is $(\omega > 0)$. The functions used in equations (1.2), (2.2), and (2.3) for the rectangle exhibit this property, but it clearly does not hold in general. For example, the function in equation (2.4) is strictly positive everywhere except the rectangle's boundary and does not distinguish between the rectangle's interior and exterior. Furthermore, if $(\omega \geq 0)$ is a point set, then $(\omega g \geq 0)$ is the same set for any function g such that $(g \geq 0) \subset (f \geq 0)$, but $(\omega g = 0)$ is not necessarily the set's boundary. These simple examples illustrate that topological properties of implicitly defined point sets vary widely, depending on application, construction procedures, and type of functions f. We will discuss limited results related to solid modeling in Section 5, but we also note that topological properties of semi-analytic and semi-algebraic sets have been studied extensively, for example, in (Whitney 1957, Whitney 1965, Andradas, Bröcker and Ruiz 1996, Bochnak, Coste and Roy 1998).

To summarize, the useful properties of the functions sought include (1) the identification of the function's sign with the membership in the set, (2) some degree of smoothness, and (3) distance-like properties. When the geometry of the point set is encoded using such a function, many otherwise difficult problems involving geometry of the point set become amenable to standard techniques from classical one-dimensional functional and numerical analysis. For boundary value problems, this implies an ability to construct bases of coordinate functions satisfying any and all types of boundary conditions. In retrospect, smooth distance-like functions may be constructed for most shapes by a variety of approximate techniques (Freytag, Shapiro and Tsukanov 2006), but to construct them *exactly* everywhere, including the corners, we need R-functions.

3. R-functions

The main utility of the theory of R-functions is to replace the logical and set-theoretic constructions with the corresponding real-valued functions, yielding an implicit representation $\omega(\mathbf{p}) \geq 0$ for any semi-analytic set Ω. Based on the above discussion, this task would be impossible with algebraic or an-

alytic functions ω. So we must look for additional operations, and it turns out that all we really need is the square root. The material in this section follows roughly (Shapiro 1988), but is essentially an annotated and distilled version of thorough expositions by Rvachev in (Rvachev 1967, Rvachev 1974, Rvachev 1982).

3.1. Logically charged real functions

This is what Rvachev called real-valued functions of real variables having the property that their signs are completely determined by the signs of their arguments, and are independent of the magnitude of the arguments. For example, consider the following functions:

$$\begin{aligned} W_1 &= xyz, \\ W_2 &= x + y + \sqrt{xy + x^2 + y^2}, \\ W_3 &= 2 + x^2 + y^2 + z^2, \\ W_4 &= x + y + z - \sqrt{x^2 + y^2} - \sqrt{x^2 + z^2} - \sqrt{y^2 + z^2} + \sqrt{x^2 + y^2 + z^2}, \\ W_5 &= xy + z + |z - yx|. \end{aligned}$$

Table 3.1 shows how the signs of these functions depend on the signs of their arguments. In contrast, here are some functions whose sign depends

x	y	z	W_1	W_2	W_3	W_4	W_5
−	−	−	−	−	+	−	+
−	−	+	+	−	+	−	+
−	+	−	+	+	+	−	−
−	+	+	−	+	+	−	+
+	−	−	+	+	+	−	−
+	−	+	−	+	+	−	+
+	+	−	−	+	+	−	+
+	+	+	+	+	+	+	+

Table 3.1. The signs of real functions $W_1, \ldots, W_5$ depend only on the signs of their arguments x, y, and z

not only on the sign of the arguments but also on their magnitude:

$$W_6 = xyz + 1, \qquad W_7 = \sin xy, \qquad W_8 = x + y + z - \sqrt{x^2 + y^2},$$

and so on. Specifying distributions of signs for the arguments of functions $W_1, \ldots, W_5$ completely determines the corresponding sign distribution of the functions; functions W_6, W_7 and W_8 do not behave in this fashion.

These simple examples illustrate the key idea. We view the set of reals $\mathbb{R}$ as consisting of two subsets: $\Delta = \{(-\infty, 0], [0, +\infty)\}$, and then seek the set $R(\Delta)$ of those functions $f : \mathbb{R}^n \to \mathbb{R}$ that predictably inherit the membership in each of the two subsets of Δ. For the time being, we assume that zero is always signed: $+0$, or -0; this allows us to determine whether it belongs to the set of positive or negative numbers.[2] Summarizing the inheritance properties of such functions in a sign table, such as Table 3.1, immediately suggests the connection between functions in $R(\Delta)$ and the binary logic functions. The connection is made precise by using the (Heaviside) characteristic function $S_2 : \mathbb{R} \to \mathbb{B} \equiv \{0, 1\}$ of the interval $[0+, \infty)$:

$$S_2(x) = \begin{cases} 0 & \text{if } x \leq -0, \\ 1 & \text{if } x \geq +0. \end{cases}$$

Definition 1. A function $f_\Phi : \mathbb{R}^n \to \mathbb{R}$ is an **R-function** if there exists a (binary) logic function $\Phi : \mathbb{B}^n \to \mathbb{B}$ satisfying the commutative diagram:

$$\begin{array}{ccc} \mathbb{R}^n & \xrightarrow{f_\Phi} & \mathbb{R} \\ S_2^n \big\downarrow & & \big\downarrow S_2 \\ \mathbb{B}^n & \xrightarrow[\Phi]{} & \mathbb{B} \end{array} \tag{3.1}$$

It is well known that the logic functions form a Boolean algebra with truth value 1 and false value 0. Such functions are usually defined using logic operations $\wedge$ (and), $\vee$ (or), and $\neg$ (negation) on n logic variables. The logic function Φ in the above definition is called the *companion* function of the R-function f_Φ. The commutative diagram implies that

$$S_2(f_\Phi(x_1, x_2, \ldots, x_n)) = \Phi(S_2(x_1), S_2(x_2), \ldots, S_2(x_n)). \tag{3.2}$$

Informally, a real function f_Φ is an R-function if it can change its property (sign) only when some of its arguments change the same property (sign). We will adopt the above definition of R-functions for the purposes of this survey. But in fact, the notion of R-functions is a special case of a more general concept of R-mapping that is associated with qualitative k-partitions of arbitrary domains and multi-valued logic functions (Rvachev 1982). We will touch briefly on this subject in Section 8.

It follows that every logic function Φ is a companion to infinitely many R-functions. For example, the companion logic function for R-function $w_1 = xy$ is $X \Leftrightarrow Y$ (X is equivalent to Y). Just check to see that

$$S_2(xy) = (S_2(x) \Leftrightarrow S_2(y)).$$

[2] Including zero in both intervals may seem strange, and we will revisit this issue in section 3.7.

But the logical equivalence is also a companion function for R-functions such as

$$w_2 = xy(1 + x^2 + y^2), \qquad w_3 = (1 - 2^{-x})(3^y - 1),$$

and so on. The set of all R-functions that have the same logic companion function is called a *branch* of the set of R-functions. Since the number of distinct logic functions of n arguments is 2^{2^n}, there are also 2^{2^n} distinct branches of R-functions of n arguments.

3.2. Branches of R-functions.

The set of R-functions is infinite. However, for applications, it is not necessary to know all R-functions; we only need to be able to construct R-functions that belong to a specified branch. The recipes for such constructions are implied by the general properties of R-functions that follow almost immediately from their definition. Complete proofs, as well as many additional properties, can be found in the references, notably in (Rvachev 1967) and (Rvachev 1982).

1 The set of R-functions is closed under composition. In other words, any function obtained by composition of R-functions is also an R-function.
2 If a continuous function $f(x_1, \ldots, x_n)$ has zeros only on coordinate hyperplanes (i.e. $f = 0$ implies that one or more $x_j = 0, j = 1, 2, \ldots, n$), then f is an R-function.
3 The product of R-functions is an R-function (because the logical companion of the product is equivalence). If the R-function $f(x_1, \ldots, x_n)$ belongs to some branch, and $g(x_1, \ldots, x_n) > 0$ is an arbitrary function, then the function fg also belongs to the same branch.
4 If f_1 and f_2 are R-functions from the same branch, then the sum $f_1 + f_2$ is an R-function belonging to the same branch.
5 If f_Φ is an R-function whose logic companion function is Φ, and C is some constant, then Cf_Φ is also an R-function. The logic companion function of Cf is Φ if $C > 0$, or $\neg\Phi$ if $C < 0$.
6 If $f_\Phi(x_1, \ldots, x_n)$ is an R-function whose logic companion function is $\Phi(X_1, \ldots, X_n)$ and f can be integrated with respect to x_i, then the function $\int_0^{x_i} f(x_1, \ldots, x_n)dx_i$ is an R-function whose logic companion function is $X_i \Leftrightarrow \Phi(X_1, \ldots, X_n)$.

The above list of properties is not exhaustive, but it is enough to suggest that more complex R-functions may be constructed from simpler functions. In particular, the closure under composition, leads to the notion of *sufficiently complete* systems of R-functions, i.e., collections of R-functions that can be composed in order to obtain an R-function from any branch.

Theorem 1. (Rvachev 1967) Let H be some system of R-functions, and G be the corresponding system of the logic companion functions. The system H is sufficiently complete, if the system G is complete.

The criteria for completeness of a system of Boolean logic functions are well understood. For example, take $G = \{1, \neg X, X_1 \wedge X_2, X_1 \vee X_2\}$. It is well known that all logic functions can be constructed using these basic functions; in other words, G is complete. It is neither unique nor minimal, since the same functions can also be constructed using only conjunction and negation, or disjunction and negation. Furthermore, all logic functions can be constructed using only one operation, the so-called Sheffer's stroke (Sheffer 1913), popularly known as the *nand* ('not and') operation. For geometric applications, the logical operations $\vee$ and $\wedge$ are both convenient and intuitive, because they define the set operations of union and intersection respectively. Thus, we adopt the system G as the primary system of companion functions and, following Theorem 1, seek the R-functions from the corresponding branches. We will refer to these functions respectively as R-negation, R-disjunction, and R-conjunction.

3.3. Sufficiently complete systems of R-functions.

It is fairly easy to come up with any number of sufficiently complete systems of R-functions. For example, it is easy to check that the following functions are R-functions (their logic companion function in parentheses):

$$\begin{array}{rll} C & \equiv \text{const}; & \text{(logical 1)} \\ \overline{x} & \equiv -x; & \text{(logical negation } \neg) \\ x_1 \wedge_1 x_2 & \equiv \min(x_1, x_2); & \text{(logical conjunction } \wedge) \\ x_1 \vee_1 x_2 & \equiv \max(x_1, x_2); & \text{(logical disjunction } \vee) \end{array}$$

Theorem 1 states that an R-function from any branch can be defined using composition of just these functions. We shall see that this system of R-functions, which we will call $R_1(\Delta)$, has a number of attractive properties, but the resulting R-functions are not differentiable. For applications where differentiability is important, for example in solutions of boundary value problems, we need another system. Below we compare several such systems in terms of simplicity, differential properties, and convenience of use. We will adopt a constant function and R-negation as above for all systems of R-functions, so that the differences between various systems amount to choosing only two operations: R-conjunction and R-disjunction.

A particularly elegant method for deriving a simple but powerful sufficiently complete system of R-functions relies on the triangle inequality. Suppose that we want to construct an R-conjunction. We are looking for a function f of two arguments x_1 and x_2, whose sign is positive if and only if both x_1 and x_2 are positive. Consider a triangle with two sides of length x_1

and x_2. The square of the third side is determined by the law of cosines as $x_1^2 + x_2^2 - 2\alpha x_1 x_2$, where α is the cosine of the angle between the two sides. It is easy to see that the function

$$f = x_1 + x_2 - \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}$$

satisfies the desired properties. When both x_1 and x_2 are positive, their sum must exceed the length of the third side of the triangle, and therefore $f > 0$. If either x_1 or x_2 are negative, then by the same argument, $f < 0$. In other words, f is the R-conjunction for any value of $-1 < \alpha < 1$. A similar argument leads immediately to the conclusion that the function

$$g = x_1 + x_2 + \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}$$

is the corresponding R-disjunction of the two real variables x_1, x_2. Together with R-negation, these R-functions constitute a sufficiently complete system, which allows construction of all other R-functions by composition. In fact, we are only one step away from what is considered to be the principal system of R-functions.

Based on our observations above, we define a system of R-functions as

$$R_\alpha(\Delta): \quad \frac{1}{1+\alpha}\left(x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}\right), \tag{3.3}$$

with $(+)$ defining R-disjunction $x_1 \vee_\alpha x_2$ and $(-)$ defining R-conjunction $x_1 \wedge_\alpha x_2$ respectively. The scalar factor $\frac{1}{1+\alpha}$ remains positive and does not affect our derivation above. It will prove useful for other distance-related properties of R-functions. But is the system R_α better than the system R_1?

It may seem that we have not improved all that much, because the two systems are closely related. Observe that $\min(x_1, x_2)$ and $\max(x_1, x_2)$ are the smallest and the largest root respectively of the equation

$$z^2 - (x_1 + x_2)z + x_1 x_2 = 0,$$

because $\min(x_1, x_2) + \max(x_1, x_2) = x_1 + x_2$, and $\min(x_1, x_2)\max(x_1, x_2) = x_1 x_2$. Solving this equation for z, we get two roots:

$$\frac{1}{2}\big[x_1 + x_2 \pm \sqrt{(x_1 - x_2)^2}\big],$$

which are simply $\max(x_1, x_2)$ and $\min(x_1, x_2)$, depending on whether we choose $(+)$ or $(-)$ respectively. Apparently, the system of R-functions $R_1(\Delta)$ is indeed the system $R_\alpha(\Delta)$ with $\alpha = 1$, which we previously excluded because it corresponds to a singular triangle with a zero angle. Whenever the expression under the square root vanishes, the resulting R_α-functions become non-differentiable; thus, R_1-functions are not differentiable whenever

$x_1 = x_2$. But let us choose $\alpha = 0$. The system $R_\alpha(\Delta)$ becomes

$$R_0(\Delta): \quad x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2}. \tag{3.4}$$

If we think of x_1, x_2 as sides of a triangle, then $\alpha = 0$ implies that this triangle is right, and the R_0-functions are based on the Pythagoras theorem! A remarkable property of these R_0-functions is that they are analytic everywhere, except at the origin where $x_1 = x_2 = 0$.

We can improve further, at least theoretically, by upgrading R_0-functions to the class of C^m functions defined as:

$$R_0^m(\Delta): \quad (x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2})(x_1^2 + x_2^2)^{\frac{m}{2}} \tag{3.5}$$

The additional factor of $(x_1^2 + x_2^2)^{\frac{m}{2}}$ makes these functions differentiable at the origin as as well, but with vanishing derivatives. At all other points, this factor stays positive, and hence does not affect the logical properties of the R_0-function. Another useful generalization of the R_0 system comes from restating the triangle inequality with the L_p norm. The resulting system of $R_p(\Delta)$ functions becomes[3]

$$R_p(\Delta): \quad x_1 + x_2 \pm (x_1^p + x_2^p)^{\frac{1}{p}}, \tag{3.6}$$

for any even positive integer p.

One may wonder whether the above R-functions are as simple as possible. For example, can we find a sufficiently complete system of R-functions among polynomials? The answer is no. It was shown in (Rvachev 1967) that a sufficiently complete system of R-functions cannot be constructed using addition and multiplication alone. On the other hand, a sufficiently complete system does not have to use the root operation, and other systems of R-functions, including those constructed in a piecewise manner, are discussed in (Rvachev 1982). We will use the notation $R^*(\Delta)$ to refer to a generic sufficiently complete system of R-functions, and the corresponding R^*-functions as $\wedge^*, \vee^*$, etc. It is convenient to compare the different systems of R-functions by plotting their level sets on the x_1x_2 plane (see Figure 3.2). From the definition, all R-functions from the same branch have identical signs in every quadrant. Thus, all R-conjunctions are positive in the first quadrant and negative in the other three. Similarly, all R-disjunctions are positive in all quadrants except the third quadrant where both x_1 and x_2 are negative. It is also clear that similarities between different systems of R-functions end somewhere in the neighborhoods of the coordinate axes, and the differences may become more pronounced as we go away from the coordinate axes.

[3] An unfortunate consequence of this notation, preserved from (Rvachev 1982), is that $R_p(\Delta)$ with $p = 2$ is identical to $R_\alpha(\Delta)$ with $\alpha = 0$.

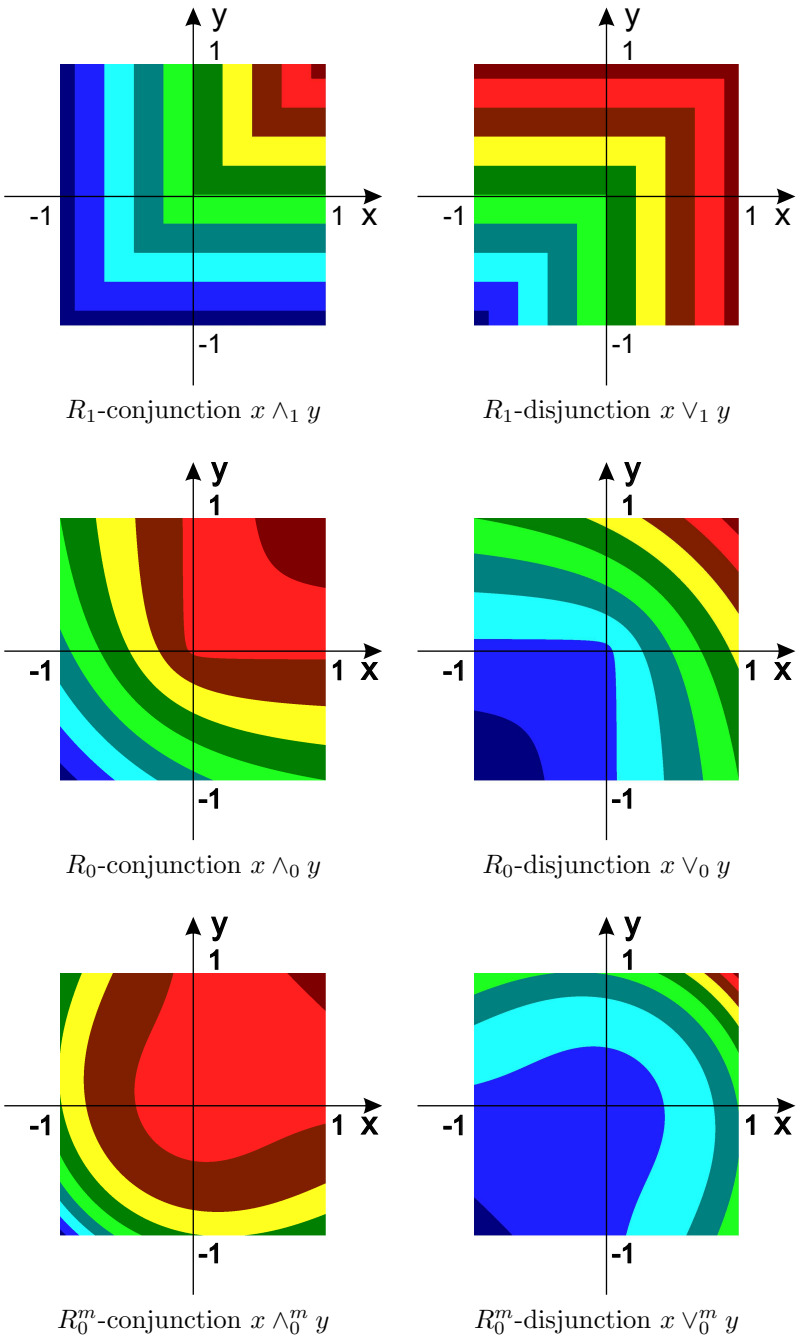


Figure 3.2. R-conjunctions (left) and R-disjunctions (right) for the three popular systems of R-functions: $R_1(\Delta)$, $R_0(\Delta)$, and $R_0^m(\Delta)$.

3.4. Composite and direct R-functions

Given any such sufficiently complete system, composition may be used to construct R-functions for any branch specified by a logic companion function. Suppose $\Phi = (\neg X_1 \wedge X_2) \vee (X_1 \wedge \neg X_2)$. The corresponding R_0-function f is obtained by composition as:

$$\begin{aligned} f &= (\overline{x}_1 \wedge_0 x_2) \vee_0 (x_1 \wedge_0 \overline{x}_2) \\ &= -x_1 + x_2 - \sqrt{x_1^2 + x_2^2} + x_1 - x_2 - \sqrt{x_1^2 + x_2^2} \\ &\quad + \left(\left(-x_1 + x_2 - \sqrt{x_1^2 + x_2^2} \right)^2 + \left(x_1 - x_2 - \sqrt{x_1^2 + x_2^2} \right)^2 \right)^{\frac{1}{2}} \\ &= 2 \left(\sqrt{x_1^2 + x_2^2 - x_1 x_2} - \sqrt{x_1^2 + x_2^2} \right), \end{aligned}$$

and, of course, the factor of 2 may be dropped as well, because it does not affect the logical properties of the composite R-function. Each of the three forms of the R-function above suggests a different use. The first expression suggests that a composite R-function may be represented and evaluated procedurally as any other expression, given its logic companion function. The second form is obtained by syntactic substitution, if such an explicit expression is desired. The last expression is much more efficient, but it could not be obtained without direct analysis and symbolic optimization of the composite R-function. From now on we will use R-conjunctions and R-disjunctions as elementary functions, i.e. we will just write $x_1 \wedge_\alpha^m x_2$, $x_1 \vee_1 x_2$, etc. We know how to evaluate and differentiate these functions, and the notation becomes much simpler.

In many special situations, it may be advantageous to construct R-functions directly, based on desired logic properties, and/or additional assumptions about the arguments. Thus, in the above example, the logic function Φ simplifies to the exclusive 'or' function, which is the negation of the equivalence. The simplest R-function in the corresponding branch is $-xy$, which may or may not be preferable to the functions in the above example, depending on other desired properties of R-functions. The general questions of optimization of composite R-functions according to some criteria may lead to challenging problems (Rvachev 1982, page 127). For example, direct constructions of R_1-conjunction and R_1-disjunction have been generalized to n-ary logical operations, but similar generalizations of R_0-functions, that are analytic almost everywhere, have been established only up to 5 arguments.

We will discuss several other direct constructions in the context of applications in Sections 6 and 7. A particularly useful concept is that of a function $f : \mathbb{R}^n \to \mathbb{R}$ which is an R-function only on some subdomain $G \subset \mathbb{R}^n$. Such a function f is called a *conditional* R-function in (Rvachev 1982). For

example, consider the function $f : \mathbb{R}^4 \to \mathbb{R}$,

$$f = x_1 + x_2 + s_1\sqrt{x_1^2 + x_2^3 + s_2}, \tag{3.7}$$

which is generally not an R-function. However when $s_2 = 0$ and $s_1 \in \{-1, 1\}$, f is an R-function of x_1, x_2: R_0-disjunction if $s_1 = 1$ and R_0-conjunction if $s_1 = -1$. Conditional R-functions may also be defined over finite intervals of its arguments, for example over the interval $x_i \in [-1, 1]$.

3.5. Logic properties of R-functions.

Since R-functions mimic the corresponding companion logic functions, one might expect that they should also inherit some properties of the Boolean logic algebra. In particular, we should be able to rely on the laws of Boolean algebra to transform R-functions without affecting their logical properties. For example, from the definition, the R-functions

$$x_1 \wedge^* (x_2 \wedge^* x_3) \quad \text{and} \quad (x_1 \wedge^* x_2) \wedge^* x_3$$

belong to the same branch, by the associative law, as do the R-functions

$$x_1 \wedge^* (x_2 \vee^* x_3) \quad \text{and} \quad (x_1 \wedge^* x_2) \vee^* (x_1 \wedge^* x_3)$$

by the distributive law, and so on.

Unfortunately, we made one important assumption that directly contradicts the properties of the Boolean algebra. Because zero is included with both positive and negative numbers, the range of $x \wedge^* \overline{x}$ includes both negative numbers and 0. This statement is at odds with the requirement of Boolean algebra that $X \wedge \neg X = \emptyset$. Thus, strictly speaking, the logical properties of R-functions are driven by the properties of a distributive lattice, and not those of the Boolean algebra. As a consequence, properties of R-functions that involve the negation operation have to be considered case by case. For example, it is easy to show that R-functions do satisfy the usual De Morgan's laws. We shall also see in Section 5 that inclusion of zero in both positive and negative numbers causes non-trivial complications in solving the inverse problem of analytic geometry. However, observe that the same assumption about the partition Δ is crucial for the following result.

Theorem 2. (Rvachev 1974, page 62) Every branch of R-functions contains at least one continuous R-function.

Since the R-conjunction and the R-negation as defined above are both continuous functions, this result follows directly from Theorem 1. If instead the real axis were subdivided into two intervals $(-\infty, 0)$ and $[0, +\infty)$, and 0 were considered a positive number, then Theorem 2 would not be true. To see this, observe that any R-negation would have to be discontinuous at $x = 0$ (Rvachev 1974, page 58).

As *real-valued* functions, R-functions may possess a number of additional logic-related properties. For example, it is easy to check that

$$\overline{\overline{x}} = x; \qquad \overline{x_1 \wedge^* x_2} = \overline{x}_1 \vee^* \overline{x}_2; \qquad \overline{x_1 \vee^* x_2} = \overline{x}_1 \wedge^* \overline{x}_2.$$

Since R-disjunctions and R-conjunctions are symmetric with respect to the two arguments x_1, x_2, we have the usual commutative laws

$$x_1 \wedge^* x_2 = x_2 \wedge^* x_1; \qquad x_1 \vee^* x_2 = x_2 \vee^* x_1.$$

Other properties may be derived for specific systems of R-functions and used in their construction and simplification. However, most systems of R-functions do *not* obey the associative or distributive laws. A notable exception to this rule is the system $R_1(\Delta)$, where min and max operations are clearly associative and distribute over each other. These properties and other computational considerations make R_1-functions preferable to other systems whenever differentiability of the functions constructed is not required.

3.6. Differential properties of the elementary R-functions.

Differential properties of R-functions are determined by the properties of the chosen system of R-functions and vary considerably. The smoothness of functions and magnitudes of their derivatives are important in geometric applications described in Section 6, and are critical for correctness of solutions to boundary value problems discussed in Section 7.

Directly differentiating R-functions in the system $R_p(\Delta)$ yields

$$\frac{\partial f}{\partial x_i} = 1 \pm \frac{x_i^{p-1}}{(x_1^p + x_2^p)^{\frac{p-1}{p}}}, \quad i = 1, 2, \tag{3.8}$$

where f is either R-conjunction or R-disjunction, depending on the choice of the sign. We observe that these R-functions are analytic everywhere, except at $x_1 = x_2 = 0$ where the derivative values change with the direction of approach but remain bounded. On the coordinate axes where one of the variables x_i is zero and the R-function changes its sign, the derivative with respect to this variable is 1, and the derivative with respect to the other variable is 0. At the same points, all higher order derivatives up to order $p-1$ vanish. In other words, the R_p-functions behave as the $(p-1)$th order approximation to the distance functions in the vicinity of the coordinate axes where the R-functions change their sign. This behavior is clearly visible in the plots of R-functions in Figure 3.2(b). Recall that the R_0-system is a special case of the R_p system with $p = 2$.

A similar analysis of R_1-functions reveals that min and max behave as exact distances near the same coordinate axes, but these functions are also not differentiable along the line $x_1 = x_2$. Both of these facts are clearly

visible in Figure 3.2(a). Finally, the R_0^m-functions are m times differentiable everywhere, including at the origin where the first m derivatives are 0. The significant drawback of these R-functions is that they no longer possess the distance properties of the other systems (see Figure 3.2(c)). As we shall see in section 6, this severely limits their usefulness in most applications. For additional detailed analysis of differential properties of the popular systems of R-functions, the reader is referred to (Shapiro and Tsukanov 1999*a*).

3.7. Other partitions of the real axis.

We have chosen the sign of a real number as the criterion for partitioning the real axis $\mathbb{R}$, but it is not obvious that choice of the partition Δ was "correct". For example, we could also choose other partitions, such as :

$$\Delta_2 = \{(-\infty, 0),\ [0, +\infty)\}; \qquad \Delta_3 = \{(-\infty, 0),\ 0,\ (0, +\infty)\}.$$

All three partitions distinguish between the positive and the negative real numbers. Note that the Δ_3 partitions the real axis into three intervals, not two. This forces us to redefine the notion of R-function in terms of companion functions of 3-valued logic, as opposed to the Boolean functions, and generalizations to multi-valued logic are briefly discussed in Section 8. The three partitions are different in the handling of zero, and we have already observed the importance of choosing the Δ partition as opposed to Δ_2 in ensuring continuity of the associated R-functions. Observe that the sets $R(\Delta_2)$, $R(\Delta)$ and $R(\Delta_3)$ intersect (Rvachev 1974, page 57). For example, the function $x_1 + x_2 - |x_1 - x_2|$ is an R-function for each of the above partitions. At the same time, the function $x_1 x_2$ is in $R(\Delta_3)$ and $R(\Delta)$ but is not in $R(\Delta_2)$, and function $x_1^2 x_2^2 (1 - x_1)^2$ is in $R(\Delta_2)$ but is not in $R(\Delta_3)$, and so on.

One may wonder why we would bother with Δ_3 to begin with. Such a partition allows one to distinguish the zero value from all other values, which is not possible with either Δ_2 or Δ and may be important for some applications. On the other hand, $R(\Delta_3)$ contains some R-functions with "undesirable" properties and 3-valued logic brings complications of its own (Rvachev 1982). To make a long story short, it turns out that all *continuous* functions in $R(\Delta_3)$ are also in $R(\Delta)$. Thus, we rely on Boolean algebra and use only $R(\Delta)$-functions, but occasionally treat them as $R(\Delta_3)$-functions. This allows us to use 3-valued logic in order to identify and *rule out* any situations where zero values may cause anomalies or ambiguities.

These and and other partitions are formally studied in (Rvachev 1982), (Rvachev and Rvachev 1979), and (Rvachev 1974). The partition Δ was used originally in (Rvachev 1967), while Δ_2 was employed in (Rvachev, Kurpa, Sklepus and Uchishvili 1973) and (Rvachev and Slesarenko 1976) which are more concerned with applications.

4. From inequalities to normalized functions

In this section, we explain what is considered the main result of the theory of R-functions. R-functions allow to construct a smooth distance-like real-valued function for any point set described by a logical predicate on a collection of inequalities. The construction is essentially a syntactic substitution.

4.1. Inequalities from logical predicates

A composition of R-functions involves applying an R-function to other R-functions. But consider what happens if the arguments of an R-function are some other arbitrary functions (that are not necessarily R-functions). Consider a function $f \equiv \phi_1 \wedge^* \phi_2$, where ϕ_1, ϕ_2 are any real-valued functions. By definition, the composite function f is positive if and only if both ϕ_1 and ϕ_2 are positive. In other words,

$$(\phi_1 \wedge^* \phi_2) \geq 0 \iff (\phi_1 \geq 0) \wedge (\phi_2 \geq 0), \tag{4.1}$$

which means that a logical conjunction of two inequalities on the right-hand side can be replaced by an equivalent single inequality on the left-hand side. For example, recall that a rectangle is an intersection of two primitive sets $(a^2 - x^2 \geq 0)$ and $(b^2 - y^2 \geq 0)$. Substituting these into (4.1), and using R_0-conjunction, we have

$$(a^2 - x^2) \wedge_0 (b^2 - y^2) \geq 0 \iff (a^2 - x^2 \geq 0) \wedge (b^2 - y^2 \geq 0).$$

The function in the inequality on the left-hand side is identical to the function in equation (1.2). By construction, the function is zero only on the points of the rectangle's boundary, and positive inside; furthermore the constructed function is analytic everywhere except at the corners of the rectangle, where both primitive functions are zero.

The above reasoning generalizes in a straightforward fashion to arbitrary predicates on sets. Let real-function inequalities $\omega_i(x_1, \ldots, x_n) \geq 0,\ i = 1, \ldots, m$ define the primitive geometric point sets $\Omega_i \subseteq \mathbb{E}^n$, and let $\Phi : \mathbb{B}^m \to \mathbb{B}$ be a predicate constructed using logical functions $\wedge, \vee, \neg$. Then the statement

$$\Phi(S_2(\omega_1), \ldots, S_2(\omega_m)) = 1 \tag{4.2}$$

represents a set $\Omega \subseteq \mathbb{E}^n$ of points where the predicate is true. The logic function Φ defines the corresponding set-valued function $\mathbf{\Phi} : \mathbb{E}^{nm} \to \mathbb{E}^n$, constructed with set operations $\cap, \cup, -$ respectively so that

$$\Omega = \mathbf{\Phi}(\Omega_1, \ldots, \Omega_m). \tag{4.3}$$

We seek a single real-function inequality $f(x_1, \ldots, x_n) \geq 0$ that defines the composite object Ω, which is readily obtained following the general result in (Rvachev 1974):

Theorem 3. Suppose the logic function $\Phi(X_1, \ldots, X_m)$ is the companion of a continuous R-function $f_\Phi(x_1, \ldots, x_m)$ and the corresponding set function $\mathbf{\Phi}$ maps closed sets into closed sets. If the closed set $\Omega \subset \mathbb{E}^n$ is represented as in Eq. (4.2), then it is also represented by the inequality

$$f_\Phi(\omega_1, \ldots, \omega_m) \geq 0. \tag{4.4}$$

In other words, to obtain a real function inequality $f \geq 0$ defining the set Ω constructed from primitive sets $(\omega_i \geq 0)$, it suffices to construct an appropriate R-function and substitute for its arguments the real functions ω_i defining the primitive sets Ω_i. The proof of the theorem follows immediately from the definition of R-functions, as expressed by equation (3.2), where membership of a point in a set is identified by the non-negative sign of the corresponding defining function evaluated at this point.

The restriction to closed sets in Theorem 3 is awkward. On one hand, it would make sense to restrict our attention to the lattice of the closed sets with operations of $\cap$, $\cup$. On the other hand, we do want the complement operation, partly for convenience, but also because it is the companion to the R-negation operation that behaves more like the *pseudo-complement* (defined as the closure of the complement) than the usual complement. For example, technically speaking, the theorem cannot be used with set difference operation $\Omega_1 \setminus \Omega_2$, because

$$(\omega_1 \wedge^* \bar{\omega_2}) \geq 0 \iff (\omega_1 \geq 0) \wedge \neg(\omega_2 \geq 0). \tag{4.5}$$

We can get around this difficulty whenever the *closure* of $\neg(\omega_2 \geq 0)$ is $(-\omega_2 \geq 0)$, since R-negation is defined as $\bar{\omega_2} = -\omega_2$. So in this particular case, we can obtain a *pseudo-difference* operation by using $-\omega_2$ on both sides of the equivalence statement (4.5) in place of R-negation on the left, and logical $\neg$ on the right-hand side. For the time being, we will rely on such case by case analysis to construct equations and inequalities for curves, surfaces, and regions in Euclidean space, but such difficulties need to be accounted for when discussing the algorithmic construction of the Boolean companion functions in Section 5.

4.2. Examples

Example 4.1. The three-dimensional model of a chess pawn, shown in Figure 4.3(a), can be constructed as a set expression adopted from (Rvachev 1967):

$$\Omega = (\Omega_1 \cap \Omega_2 \cap \Omega_3) \cup \Omega_4 \cup \Omega_5,$$

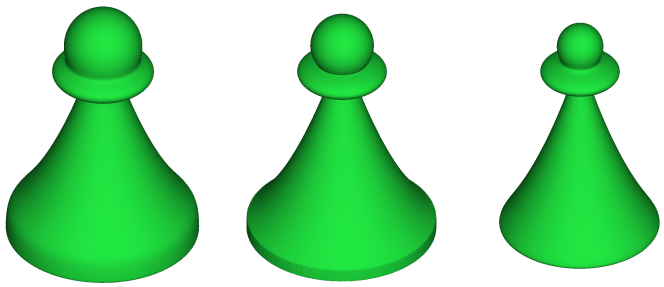
where the primitive regions $\Omega_i = (\omega_i(x, y, z) \geq 0)$ are halfspaces defined as shown below.

Primitive Ω_i	Function ω_i
Ω_1 Solid of revolution	$\omega_1 = -z + \frac{7}{16}\left(\sqrt{x^2+y^2} - 4.0\right)^2$
Ω_2 Cylinder	$\omega_2 = 9.0 - x^2 - y^2$
Ω_3 Horizontal slab	$\omega_3 = z\,(7-z)$
Ω_4 Sphere	$\omega_4 = 1 - x^2 - y^2 - (7-z)^2$
Ω_5 Ellipsoid	$\omega_5 = 2 - x^2 - y^2 - 9\,(6-z)^2$

Following Theorem 3, a single inequality $(\omega \geq 0)$ defining the same point set Ω is obtained by syntactic substitution:

$$\omega = (\omega_1 \wedge^* \omega_2 \wedge^* \omega_3) \vee^* \omega_4 \vee^* \omega_5 \geq 0. \tag{4.6}$$

Figure 4.3(b) shows the isosurface $\omega = 0$ constructed with R_0-functions and computed by polygonization, while Figures 4.3(a) and (c) show isosurfaces $\omega = 0.5$ and $\omega = -0.5$, respectively.



(a) $\omega = 0.5$; (b) $\omega = 0$ (c) $\omega = -0.5$

Figure 4.3. Isosurfaces $\omega = c$ of function ω. The pawn is defined implicitly by $(\omega \geq 0)$ and its boundary by $(\omega = 0)$.

Example 4.2. In this case, the goal is to construct a function ω that vanishes on the dotted line boundary of a "flag" shape shown in Figure 4.4. The boundary of the flag itself consists of four segments: circular on the right, sinusoidal on the top, linear on the left and the bottom of the flag. In addition, the flag's "handle" is the dangling line segment that does not bound any interior. Define four primitives halfspaces $\Omega_i = (\omega_i \geq 0)$ using the following functions:

Primitive Ω_i	Function ω_i
Ω_1 Circular	$\omega_1 = 4.5^2 - (x+2)^2 - y^2$
Ω_2 Sinusoidal	$\omega_2 = 1 + 0.25\sin(\pi x) - y$
Ω_3 Vertical Linear	$\omega_3 = x + 2$
Ω_4 Horizontal Linear	$\omega_4 = y + 1$

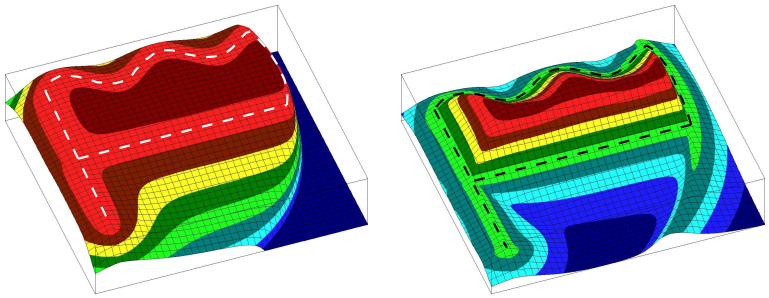
It is easy to see that the bounded flag itself is simply $\bigcap_{i=1}^{4} \Omega_i$, but the handle is more problematic. First of all it is a *segment* of the *line*, not a halfspace. But we can always write the line as a halfspace $(-|\omega_3| \geq 0)$, and intersect it with some region Ω_5, say a unit circular disk $\Omega_5 = (1-(x-2)^2-(y-2)^2 \geq 0)$, to select the required segment. The union of the flag and the handle gives:

$$\Omega = (\Omega_1 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4) \cup ((-|\omega_3| \geq 0) \cap \Omega_5)$$

Applying Theorem 3, yields a single inequality constructed with R-functions as:

$$\omega = (\omega_1 \wedge^* \omega_2 \wedge^* \omega_3 \wedge^* \omega_4) \vee^* (-|\omega_3| \wedge^* \omega_5) \geq 0 \tag{4.7}$$

The function ω constructed with R_0-functions is plotted in Figure 4.4(a).



(a) (b)

Figure 4.4. Implicit representation ($\omega = 0$) of the one-dimensional planar shape shown as a dotted line. (a) Plot of the function ω defined in (4.7); (b) Normalized function ω_1 constructed from ω using (4.9).

4.3. Distance and normalized functions

Given a closed point set Ω, the Euclidean distance function $d : \mathbb{E}^n \to \mathbb{R}$ defined by (2.5) gives for every point of space the shortest distance to the boundary $\partial\Omega$. As we discussed in Section 2, distances play an important role in many applied computational problems. Traditionally, the distance functions are written in a closed form for simple geometric shapes, for example a line, a plane, a circle, a sphere, but more complex shapes usually

involve procedural definitions, numerical computations, and/or approximations. Furthermore, the distance function is not differentiable at all points that are equidistant[4] from two or more points of $\partial\Omega$, making them unsuitable for use in any application where smoothness is required.

Both difficulties are bypassed by replacing the exact distance functions with their mth order approximations in the neighborhood of $\partial\Omega$. Let ν be the unit vector at point $\mathbf{p} \in \partial\Omega$ pointing away from $\partial\Omega$ towards the points that are closer to $\mathbf{p}$ than to any other point in $\partial\Omega$. In other words, ν is the unit normal on regular (smooth) points of the boundary $\partial\Omega$, but it is also well defined in neighborhoods of all other points, including sharp corners. A suitable mth order approximation of the distance function d is a function ω whose derivatives agree with d up to the order m in all normal directions ν. In other words, we say that the function ω is *normalized* up to the mth order if its directional derivatives D_ν^k in the direction ν near $\partial\Omega$ satisfy:

$$D_\nu \omega = 1, \qquad D_\nu^k \omega = 0, \ k = 2, 3 \ldots, m. \tag{4.8}$$

The exact distance function d is normalized to any order, and equations of the form $d = 0$ are often called normal.

Without additional assumptions, there is no reason to expect that a function f constructed using the theory of R-functions should possess any distance-related properties. But suppose that we constructed a function $\omega \in C^m$ such that $\omega(\mathbf{p}) = 0$ and $D_\nu^1 \omega(\mathbf{p}) \neq 0$ on all points $\mathbf{p} \in \partial\Omega$. Then the scaled version of this function

$$\omega_1 \equiv f(f^2 + \|\nabla f\|^2)^{-\frac{1}{2}} \ \in C^{m-1} \tag{4.9}$$

is normalized to the first order. Straightforward differentiation confirms that $D_\nu \omega_1 = \|\nabla \omega_1\| = 1$ on all regular points of $\partial\Omega$. Figure 4.4(b) shows the plot of the function ω_1, normalized to the first order by applying (4.9) to the function ω defined in (4.7) and plotted in Figure 4.4(a). Furthermore, if ω_1 is normalized to the first order, then the function ω_m normalized to the mth order may be constructed by recursively subtracting the non-zero contribution of the higher order terms:

$$\omega_m = \omega_{m-1} - \frac{1}{m!} \omega_1^m D_\nu^m \omega_{m-1}. \tag{4.10}$$

This method of normalization is particularly effective as an analytical tool, or when the initial function ω is relatively simple. However, if the function $\omega = f(\omega_1, \ldots, \omega_m)$ is constructed as an R-function on a large number of primitive functions ω_i in (4.4), it is not likely to satisfy the required smoothness conditions, and the method becomes impractical.

A more constructive approach to normalization is to start with analytic

[4] This includes all points on the medial axis, or on the Voronoi diagram of $\partial\Omega$.

or sufficiently smooth primitive functions ω_i that are already normalized, and then choose the R-function f so that it preserves the normalization at all regular points of the boundary $\partial\Omega$. In particular, we already observed in Section 3.6 that many of the R-functions themselves are normalized near their zero level sets, as is evident from expression (3.8). This is all that is required to establish the following result.

Theorem 4. Suppose that the argument x_i appears in the R_p-function $f(x_1, x_2, \ldots, x_n)$ only once and has an inversion degree[5] of r. Let the functions $\omega_1, \omega_2, \ldots, \omega_n$ be in C^s and the boundary $\partial\Omega = (f(\omega_1, \omega_2, \ldots, \omega_n) = 0)$. Then, at every regular point $\mathbf{p} \in \partial\Omega$ where

$$\omega_i(\mathbf{p}) = 0, \quad \omega_j(\mathbf{p}) \neq 0, j = 1, \ldots, n, j \neq i,$$

for every direction μ and $k \leq s < p$,

$$D^k_\mu f(\omega_1, \omega_2, \ldots, \omega_n) = (-1)^r D^k_\mu \omega_i.$$

Thus, the normalization of functions constructed with R_p-functions, following Theorem 3, comes at no extra cost, provided that the primitives functions ω_i are themselves normalized to the required order. Furthermore, if a point $\mathbf{p} \in \partial\Omega$ belongs to the boundary of exactly one primitive ($\omega_i(\mathbf{p}) = 0$), then all differential properties of the composite function f at $\mathbf{p}$ are completely determined by the differential properties of $\omega_i(\mathbf{p})$. Rvachev (1982) derives sufficient conditions for Theorem 4 to hold with any system of R-functions, and proves specific results for other popular choices of R-functions.

5. The inverse problem of analytic geometry

5.1. The general problem

We now appear to have all the ingredients needed to solve the general problem of inverse analytic geometry. For any closed semi-analytic set Ω, the problem is solved in two steps: first represent Ω by a logical predicate Φ on analytic primitives ($\omega_i \geq 0$), then translate this logical predicate into the corresponding inequality ($\omega_\Phi \geq 0$) by syntactic substitution, as prescribed by Theorem 3. If this inequality represents a set Ω, then every point $\mathbf{p}$ on the boundary $\partial\Omega$ has the property that $\omega_\Phi(\mathbf{p}) = 0$. If there are no other points $\mathbf{p} \notin \partial\Omega$ where ω_Φ vanishes, this translation solves the inverse problem of analytic geometry. We also saw that R-functions may be chosen to

[5] Inversion degree of the argument x is the number of times subexpressions with x are negated during evaluation of f. For example, in $\overline{x_1 \wedge_p \overline{(x_2 \vee_p \overline{x_3})}}$ the inversion degree of x_1 is 1, the inversion degree of x_2 is 2, and the inversion degree of x_3 is 3.

preserve the distance properties of the primitive functions ω_i at the regular points of the boundary $\partial\Omega$ to any desired order.

The above observations may be put to immediate practical use with a new generation of geometric languages that describe complex shapes by recursively combining simpler shapes using various set operations. An example of such a language is Constructive Solid Geometry (CSG) representation (Requicha and Voelcker 1977) that was particularly popular in the early days of solid modeling. Rvachev and Manko (1983) developed a similar language for describing domains of boundary value problems by overloading the usual logic operations with R-functions that combine basic primitives ($\omega_i \geq 0$) of several common types. The R-function constructions now appear in the core of modern computer graphics languages that produce implicit representations $\omega = 0$ of shapes and scenes, for example, in (Pasko et al. 1995, Wyvill, Guy and Galin 1999).

However, in many practical situations, point sets are given not in the required predicate form, but are more naturally described by their *boundaries*. Engineers and scientists tend to sketch or sculpt the shapes of interest, and digitally acquired shapes are often defined by reconstructed boundaries. Manually constructing predicates Φ or, equivalently, set-valued expressions $\mathbf{\Phi}$ for such shapes is often a non-trivial proposition. We do not usually think of a rectangle as the intersection of two unbounded strips, and even the simple predicate expressions for the point sets in the last section are not obvious or unique. Thus, the general problem of the inverse analytic geometry may be formulated as follows.

P1: *Given a piecewise (semi-)analytic boundary* $\Gamma = \bigcup \Gamma_i$ *of a set* Ω*, where* $\Gamma_i \subseteq (\gamma_i = 0)$*, construct a function* ω *such that* $\Gamma = (\omega = 0)$ *and* ω *is normalized to some order* p *on all regular points of the boundary* Γ*.*

Both the boundary Γ and the set Ω are closed semi-analytic sets and, as such, can be represented by set expressions as required by Theorem 3. But these set expressions are neither known nor unique. Accordingly, there are two generic approaches to solving this problem using the theory of R-functions described below: the first one focuses on set representation of Γ and the second on set representation of the set Ω itself. Both approaches are based on the well known fact that every Boolean set expression may be represented in a disjunctive canonical form as a union of intersection terms.

5.2. Normalized functions from boundaries

In many applications, it is understood that $\Gamma = \partial\Omega$ is the boundary of some domain Ω with non-empty interior, and we will consider such situations in Section 5.3. But in general, this need not be so, and when Γ does not bound any interior, we have $\Gamma = \Omega$. The problem of constructing the normalized

function ω such that $\omega = 0|_\Gamma$ is solved by constructing normalized functions ω_i for portions Γ_i of the boundary and then combining them, using R-functions in both steps of the procedure. In fact, we already used this method in constructing the function ω in Example 4.2. The key steps are described in (Rvachev 1982), but for detailed analysis, extensions, and experimental results the reader is referred to (Shapiro and Tsukanov 1999*a*) and (Biswas and Shapiro 2004). We will refer to the two steps in this construction procedure as *trimming* and *joining* respectively.

Trimming

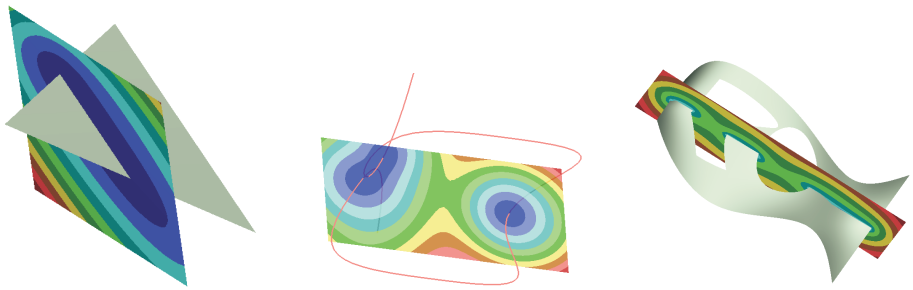
The trimming step of the construction procedure assumes that a portion Γ_i of the boundary Γ may be represented as $(\gamma_i = 0) \cap \Lambda$, where γ_i is normalized to some order and represents an unbounded curve, surface, or hypersurface $(\gamma_i = 0) \supset \Gamma_i$, and $\Lambda \subset \mathbb{E}^d$ is a full-dimensional region that contains its portion Γ_i. If Λ is defined implicitly by $(\lambda \geq 0)$ and λ is also normalized, then this construction translates directly into $\omega_i = -(-|\gamma_i| \wedge^* \lambda)$, with $(\omega_i = 0)$ defining the trimmed portion of Γ_i. This achieves the desired result, unless one is concerned with differential properties of ω_i. It is expected that ω_i is not differentiable on the points of the *trimmed* boundary Γ_i, but the constructed function ω_i is not differentiable on all points where $\gamma_i = 0$. It is also easy to check that ω_i is normalized on all regular points of Γ_i, but not near its end points where $\gamma_i = \lambda = 0$. Several improved alternative approaches to trimming are known. For example, it can be shown (Sheiko 1982) that

$$\omega_i = \sqrt{\gamma_i^2 + \frac{\left(\sqrt{\lambda^2 + \gamma_i^4} - \lambda\right)^2}{4}} \tag{5.1}$$

is normalized on all regular points of Γ, twice differentiable on the boundary of Λ and is analytic on all other points $\mathbf{p} \notin \Gamma_i$. Normalization at the end points of Γ_i depends on the local geometry of the intersection between the sets $(\gamma_i = 0)$ and $(\lambda_i = 0)$ and can be guaranteed via suitable coordinate transformation. (See examples in (Shapiro and Tsukanov 1999*a*).)

The above approach to trimming works well for curve segments, because both planar and space curves can be trimmed by relatively simple trim regions, such as sphere, box, etc., for which normalized implicit representation $(\lambda = 0)$ are easily constructed. Two difficulties arise when $\Gamma_i \subset \mathbb{E}^3$ is a trimmed surface. In this case, the trim region Λ may be described by a complex set-theoretic expression, closely related to Constructive Solid Geometry representations (Rossignac 1996). We will discuss how such expressions may be constructed automatically in Section 5.3.

The second issue relates to the differential properties of the function λ. Suppose that a trim region $\Lambda = \mathbf{\Phi}(\Lambda_1, \ldots, \Lambda_n)$ is described by a set-theoretic



(a) Triangle (b) Polygonized curve (c) Triangulated surface

Figure 5.5. Normalized functions, constructed for various sets Γ by the trim-and-join method, are differentiable on all points not in the set Γ.

expression $\mathbf{\Phi}$, where Λ_j are primitives halfspaces ($\lambda_j \geq 0$). Following Theorem 3, we immediately obtain the implicit representation for $\Lambda = (\lambda \geq 0)$, where $\lambda = f_\Phi(\lambda_1, \ldots, \lambda_n)$ is an R-function corresponding to $\mathbf{\Phi}$. As we discussed earlier, this function may not be differentiable at certain "corner points" where more than one λ_j vanish. These points typically do *not* lie on the hypersurface ($\gamma_i = 0$), but the singularities will be inherited by ω_i in equation (5.1). The difficulty is resolved by noticing that the set-theoretic representation of the trim volume $\mathbf{\Phi}(\Lambda_1, \ldots, \Lambda_n)$ is needed only on the surface ($\gamma_i = 0$) being trimmed. Then constructing $\lambda = f_\Phi$ using the *conditional* R_0-functions (3.7) with $s_2 = a\gamma_i^k$ guarantees that the function λ will be analytic everywhere in $\mathbb{E}^3$ except at the edges of the trimmed surface Γ_i. Parameters a, k can be used to control the overall shape of the trim region away from the surface. Figure 5.5(a) shows a normalized function for a triangle from (Biswas and Shapiro 2004), where this method was proposed. In this case, Λ is the unbounded triangular prism perpendicular to the plane of the triangle. The function λ was constructed using the conditional R-conjunction on the three linear halfspaces bounding the prism. The function shown is twice differentiable on all points away from the triangle.

Joining

Suppose that $\Gamma = \bigcup \Gamma_i$ and we used the trimming operations described above to construct normalized functions ω_i such that $\Gamma_i = (\omega_i = 0)$. We now want to construct a single function ω such that $\Gamma = (\omega = 0)$. By the above construction, ω_i is strictly positive on all points $\mathbf{p} \notin \Gamma_i$. Applying Theorem 3 and De Morgan's law to the union $\bigcup_i \Gamma_i$ gives:

$$\bigcup_i (\omega_i = 0) = \bigcup_i (-\omega_i \geq 0) = \left(\bigvee_i^* (-\omega_i) \geq 0 \right) = \left(\bigwedge_i^* \omega_i \geq 0 \right). \qquad (5.2)$$

In particular, applying the $\wedge_p$ operations to normalized ω_i ensures the normalization of the resulting function ω to the pth order on all regular points of Γ.

Notice, however, that the functions ω_i are non-negative *everywhere*, and we are only interested in the zero set of the constructed function ω. This implies that the union of sets Γ_i is represented by the product of the respective functions ω_i. Since we also want to guarantee that ω is normalized, we replace multiplication with the R_p-equivalence operation defined as

$$x \sim_p y = xy(|x|^p + |y|^p)^{-1/p}, \tag{5.3}$$

and construct the required function ω as

$$\omega = \omega_1 \sim_p \omega_2 \sim_p \cdots \sim_p \omega_n.$$

Once again, the function ω is normalized if each ω_i is normalized. Furthermore, the R_p-equivalence operation is associative, whereas the R_p-conjunction used in equation (5.2) is not. As should be expected, neither joining operation maintains normalization at the corner points where $\Gamma_i \cap \Gamma_j \neq \emptyset$. The reader is referred to (Shapiro and Tsukanov 1999*a*) and (Biswas and Shapiro 2004) for additional discussion of differential properties of ω in the neighborhoods of the corners, as well as possible means for controlling them. See also section 7.3 for a related discussion.

The described trim-and-join technique can be used to construct an an implicit representation ($\omega = 0$) for a variety of point sets Γ, including space curves, polygonized surfaces, polyhedra, piecewise-algebraic boundary representations, dimensionally heterogeneous complexes, and so on. Figure 5.5(b) and (c) show examples of normalized functions from (Biswas and Shapiro 2004) constructed by the described method.

One disadvantage of the above approach is that the constructed function ω is strictly positive everywhere away from the boundary Γ, and thus does not distinguish any interior points of Ω even when they are bounded by Γ. For example, the Jordan-Brouwer separation theorem (which subsumes the Jordan Curve theorem) guarantees that the bounded interior of $\Omega \subset \mathbb{E}^3$ is determined unambiguously whenever Γ is a compact two-dimensional C^0-manifold surface in $\mathbb{E}^3$. In this case, the *signed* function ω may be constructed for Ω, such that $\omega > 0$ for all points in the interior $\mathbf{i}\Omega$, and $\omega < 0$ for all points $\mathbf{p} \notin \Omega$. This can be achieved by multiplying ω by the characteristic function $\xi(\mathbf{p}, \Omega)$ defined to be 1 when $\mathbf{p} \in \Omega$ and -1 otherwise, but algorithms for computing ξ usually require non-trivial data structures to represent $\partial\Omega$ and numerically sensitive algorithms, for example to compute the winding number and/or mod 2 intersection computations (Shapiro 2002, O'Rourke 1998). We now consider an alternative approach

that relies on constructing set-theoretic representation for the Ω (and not just its boundary $\partial\Omega$).

5.3. Signed functions via set expressions

When $\partial\Omega$ bounds a non-empty bounded interior $\mathbf{i}\Omega$, we know that the closed semi-analytic set Ω may be represented explicitly by a set-theoretic expression $\mathbf{\Phi}$ on some set of primitive analytic halfspaces. We also wish for ω_Φ to be a *signed* function for Ω, that is, $\partial\Omega = (\omega_\phi = 0)$ and $\mathbf{i}\Omega = (\omega_\phi > 0)$, but we shall see that this may not always be the case. The general problem P1 may be restated as follows.

P2: *Given the boundary* $\partial\Omega = \bigcup \partial\Omega_i$ *of the set* Ω*, let* $\mathcal{H}$ *be the set of bounding halfspaces* $\Omega_i^+ = (\omega_i \geq 0)$ *and* $\Omega_i^- = (-\omega_i \geq 0)$ *induced from boundary portions* $\partial\Omega_i$*. Construct a set expression* $\mathbf{\Phi}(\Omega_1^\pm, \ldots, \Omega_m^\pm)$ *such that* $\Omega = \mathbf{\Phi}$*, and* ω_Φ *is a signed function for* Ω*.*

When $\partial\Omega$ bounds some non-empty interior, the halfspaces $\Omega_i^+ = (\omega_i \geq 0)$ may be chosen to include some points in the interior $\mathbf{i}\Omega$. Below we consider this problem for three different classes of sets Ω: simple two dimensional polygons, general semi-analytic sets, and manifold solids.

Simple polygons and extensions

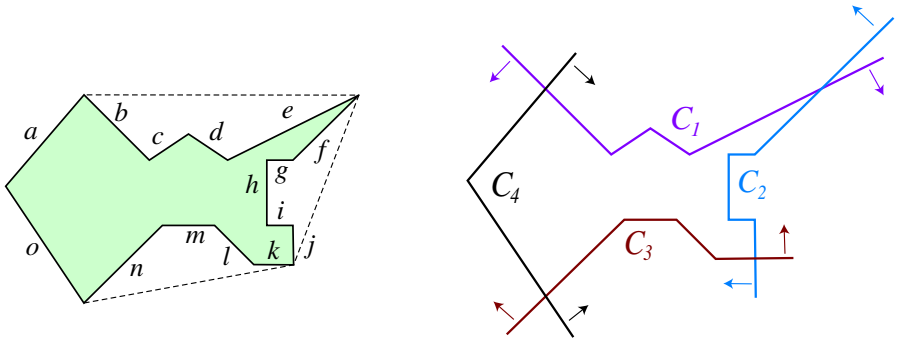
If $\Omega \subset \mathbb{E}^2$ is a simple polygon, its boundary $\partial\Omega$ is a union of n line segments $\partial\Omega_i$. Thus, $(\omega_i = 0)$ is a line, Ω_i^+ is an induced closed linear halfspace, and Ω_i^- is it's pseudo-complement. We will assume that the positive side of ω_i coincides with the interior of the polygon Ω. It is easy to see that Ω may be represented by a set expression using only halfspaces induced from the polygon's edges. Let $\mathcal{A}(L)$, be the linear arrangement of the collection of lines $L = \{(\omega_i = 0)\}$. It consists of all k-cells σ^k formed by nonempty intersections of the induced halfspaces and their pseudo-complements (Edelsbrunner 1987). It should be obvious that Ω is the union of all two-dimensional $\sigma_j^2 \subseteq \Omega$. The resulting set expression has the unique canonical disjunctive form

$$\mathbf{\Phi} = \bigcup_j \bigcap_i S_i, \tag{5.4}$$

where $S_i \in \{\Omega_i^+, \Omega_i^-\}$. Expression (5.4) is inefficient, but can be optimized in a number of ways using Boolean optimization techniques.

Alternatively, the optimal representation of a polygon, with every primitive Ω_i^+ appearing exactly once, may be constructed efficiently using the algorithm described in (Dobkin, Guibas, Hershberger and Snoeynik 1988). Any polygon can be represented as the intersection of two or more polygonal semi-infinite chains as illustrated in Figure 5.6(b). The chains intersect at

the vertices of the polygon's convex hull. Each of the chains can be split recursively into smaller subchains. If the split occurs at a concave vertex of the *original* polygon, then the subchains are combined using set union; the subchains are combined using intersection at the convex vertices of the original polygon.



(a) A polygon and its convex hull

(b) A polygon can be represented as the intersection of four polygonal chains

Figure 5.6. A set representation for any polygon can be constructed using union and intersection on polygonal chains associated with polygon's edges.

Every subexpression of the resulting set expression corresponds to some polygonal chain. For the polygon in Figure 5.6, the resulting expression is

$$(b \cup (c \cap d) \cup e)((f \cap g) \cup h \cup (i \cap j))((k \cap l) \cup m \cup n) \cap (o \cap a),$$

where the literals correspond to the linear halfspaces Ω_i^+ associated with the polygon edges. The expression is the intersection of four subexpressions corresponding to the four chains shown in Figure 5.6. Each chain is either the union or the intersection of its subchains. For example, the chain C_1 is formed by three subchains: b, $c \cap d$, and e meeting at the vertices of the convex hull of C_1; since these vertices are concave, C_1 is represented as a union of the three subchains: $b \cup (c \cap d) \cup e$; and so on.

A similar algorithm was articulated much earlier by Rvachev et al. (1973), who proposed to recursively split the polygonal chains at the vertices of the convex hulls of the *bounded* (trimmed) polygonal chains, connected by the dashed lines in Figure 5.6(a). A counter-example in (Peterson 1986) shows that such an algorithm does not always result in the correct expression. Other related algorithms to construct set expressions for polygons are described by Tor and Middleditch (1984) and Woodwark and Wallis (1982).

The construction algorithms for simple polygons based on the convex hull may be extended to some other point sets. Non-simple polygons are easily represented as a set combination of simple polygons. The approach may also be generalized to a large class of curved polygons (Shapiro 2001). Two-dimensional set representations are often used to construct representations for three-dimensional solids by translational or rotational extrusion of a planar shape in the direction normal to the plane (Woodwark and Wallis 1982, Peterson 1986, Shapiro and Vossler 1991*b*). In this case, linear halfplanes become either linear or quadric halfspaces in $\mathbb{E}^3$. Algorithms based on the convex hull can also be applied directly to some, but not all three-dimensional polyhedra (Woo n.d., Kim and Wilde 1992).

General semi-analytic sets

General theoretical algorithms for constructing set expressions for semi-algebraic (but not semi-analytic) sets are known; for example see (Basu, Pollack and Roy 2003, Basu, Pollack and Roy 2005) and references therein. At the time of writing, none of these algorithms is practical enough to deal with realistic engineering problems, even when the dimension of space, as well as the number and degree of all polynomials, are fixed and imply polynomial complexity. We seek more a intuitive, geometric characterization of the construction problem that can be used in restricted practical situations.

Conceptually, the approach to constructing set expressions for simple polygons using the arrangement of primitives generalizes to arbitrary semi-algebraic and semi-analytic sets (Shapiro 1991, Shapiro 1997). It is based on the observation that a set of $2n$ (semi-)analytic primitives $(\pm\omega_i \geq 0) \subset \mathbb{R}^d$ generate a finite distributive lattice $\mathcal{L}$ of closed subsets of $\mathbb{R}^d$ under operations of $\cap$, $\cup$. This implies that Ω is an element of $\mathcal{L}$ *if and only if* it can be represented in the disjunctive canonical form (5.4). The key difference between the linear arrangement and the general case is in the intersection terms $J_k = \bigcap_i S_i,\ S_i \in \{\Omega_i^+, \Omega_i^-, (\omega_i = 0)\}$. They are no longer convex sets but can be heterogeneous, possibly disconnected, sets of arbitrary dimension. A non-empty intersection term J_k is called a *join-irreducible* element of the lattice $\mathcal{L}$ if

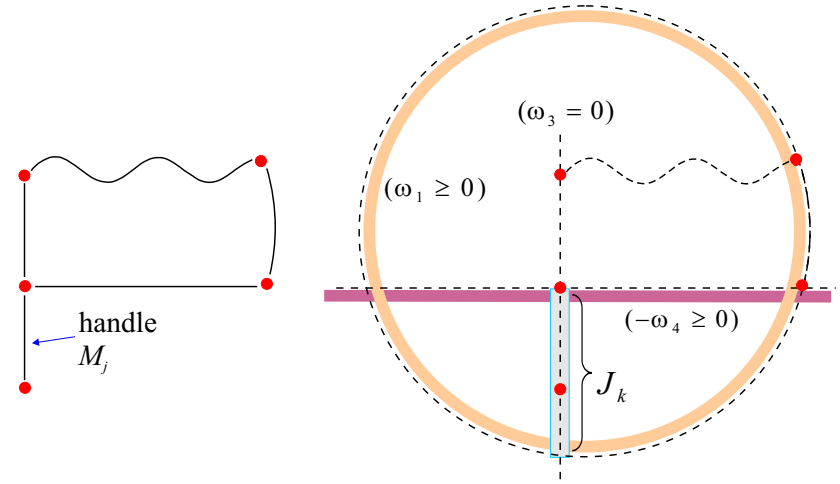
$$A \cup B = J_k \implies (A = J_k) \text{ or } (B = J_k),$$

for any sets $A, B \in \mathcal{L}$. In geometric terms, J_k is either a set that does not contain any other sets of the lattice, or J_k contains other elements as *proper* subsets. In the arrangement $\mathcal{A}\{(\omega_i = 0)\}$ of analytic primitives, the sets J_k play the same role as the k-cells σ^k play in the linear arrangement of lines. It follows that, to construct a set expression $\mathbf{\Phi}(\Omega_1^{\pm}, \ldots, \Omega_2^{\pm})$ for a semi-analytic set Ω, we need to compute the decomposition of $\mathbb{R}^d$ into join-irreducible sets J_k of $\mathcal{L}$; the union of $J_k \subseteq \Omega$ yields the disjunctive canonical form (5.4) for Ω.

A potential problem with this approach is that Ω may not be an element of the lattice $\mathcal{L}$. Consider the shape Ω in the flag example 4.2. Four primitive halfspaces $\omega_i,\ i = 1, 2, 3, 4$ were induced from the given piecewise description of $\partial\Omega$. But the flag's handle is a trimmed portion of the line $(\omega_3 = 0)$ and cannot be represented without introducing an additional (non-unique) halfspace $(\omega_5 \geq 0)$. This happened because the join-irreducible element

$$J_k = (\omega_1 \geq 0) \cap (\omega_3 = 0) \cap (-\omega_4 \geq 0)$$

contains the flag's handle but $J_k \not\subseteq \Omega$. See Figure 5.7. This example illustrates the key difficulty in constructing set representations. The set of halfspaces $\mathcal{H}$ induced from the boundary $\partial\Omega_i$ may not be sufficient for representing Ω, because the decomposition of space into the join-irreducible elements may not be fine enough. In this case, we say that the set Ω is *not describable* by the primitives in $\mathcal{H}$.



(a) Whitney regular stratification of Ω

(b) Join-irreducible element $M_j \subset J_k \not\subset \Omega$

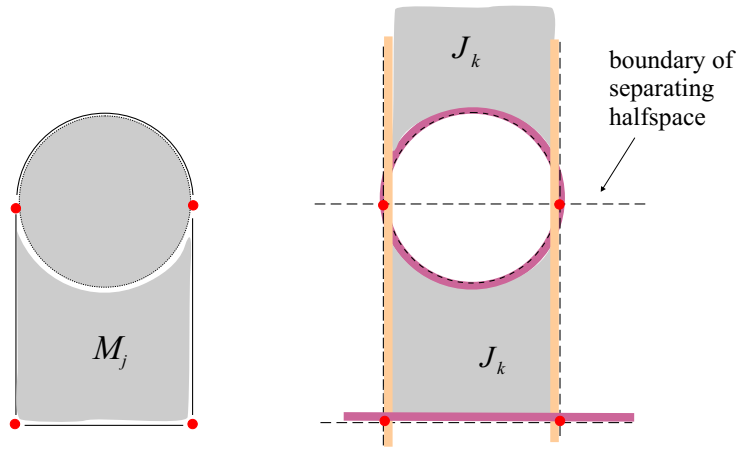
Figure 5.7. The flag shape Ω is not describable by the halfspaces induced from $\partial\Omega$.

Apparently, describability of Ω has something to do with the ability to represent some subsets of Ω, such as in the example above. What are these subsets and can they be enumerated? Any semi-analytic d-dimensional shape Ω can be stratified into k-manifold cells $M_j,\ k = 0, \ldots d$ such that all points in M_j have the same signs with respect to all primitives in $\mathcal{H}$. Because we are concerned with closed sets, we require that the closure of cells $\mathbf{k}M_j \subseteq \Omega$, which in turn requires that the closure of every cell M_j is a union of other cells in the stratification. The coarsest (and therefore minimal) stratification

satisfying these conditions is the sign-invariant Whitney regular stratification into connected strata (Whitney 1965, Shapiro 1997).

Theorem 5. Let $\mathcal{L}$ be a finite distributive lattice generated by a set $\mathcal{H}$ of halfspaces $(\pm\omega_i \geq 0), i = 1, \ldots, n$. Set $\Omega \in \mathcal{L}$ if and only if, for every k-cell $M_j \subset \Omega$ in the connected sign-invariant Whitney regular stratification of Ω, $\mathbf{k}M_j \subseteq J_k \subseteq \Omega$, for some join-irreducible element $J_k \in \mathcal{L}$.

The proof follows from results in (Shapiro 1991). The theorem reaffirms that the set Ω is not describable if it cannot be represented as a union of join-irreducible elements, but it also explains why this happens. In general, when the set $\mathcal{H}$ of primitive halfspaces is induced from the boundary $\partial\Omega$, there is no guarantee that the intersection terms defining the join-irreducible elements J_k satisfy the conditions of the theorem. This could happen when $\mathbf{k}(\omega_i > 0) \neq (\omega_i \geq 0)$, because the latter could contain some additional points. Another common situation is that M_k are connected sets, but the set $J_k \setminus \partial\Omega$ may contain several connected components. See the example in Figure 5.8. In all such cases, the set Ω is not describable by the set of halfs-



(a) A ell M_j in Whitney regular stratification of Ω

(b) The disconnected join-irreducible element $J_k \not\subset \Omega$

Figure 5.8. The shape Ω in (a) is not describable by the halfspaces induced from $\partial\Omega$, shown in (b), because the join-irreducible element J_k containing the cell M_j is separated by the boundary $\partial\Omega$ into two components. Additional separating halfspace is needed for representing Ω.

paces $\mathcal{H}$, and additional *separating* halfspaces must be used in constructing any set expression for Ω. The purpose of the separating halfspaces is to break up the problematic join-irreducible elements into smaller join-irreducible el-

ements satisfying the conditions of Theorem 5. See (Shapiro 1997) for more details.

The above observations suggest that, in order to construct a set expression for Ω, we may need to construct the Whitney regular stratification of Ω, compute the relevant join-irreducible intersection terms J_k, add separating halfspaces as needed, and finally construct the disjunctive canonical form. Theoretically, all these steps are feasible, at least for semi-algebraic sets, but no practical algorithms are available, and this situation is not likely to change in the foreseeable future. The next section outlines a pragmatic and fully implemented approach to solving the describability problem for a limited but useful class of three-dimensional solid shapes.

Set expressions for solids

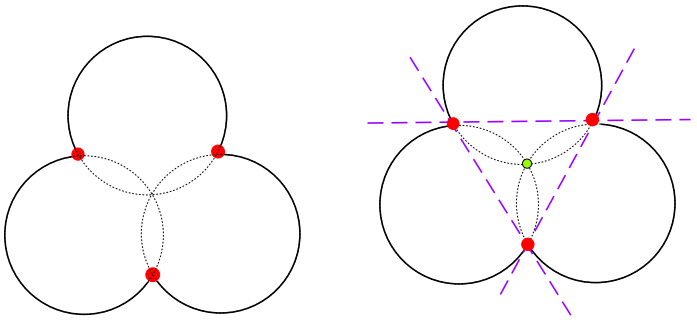
For a recent survey on solid modeling and many additional references, the reader is referred to (Shapiro 2002). A closed bounded semi-analytic set $\Omega \subseteq \mathbb{E}^d$ is called a *solid* if it is *closed regular*, that is, $\mathbf{ki}\Omega = \Omega$ (Requicha 1980). A solid can be represented on a computer using one of many representation schemes, but the most common way to represent a solid model on a computer is by its boundary $\partial\Omega$ stored as a union of faces $\partial\Omega = \bigcup_i \partial\Omega_i$. It is also common to assume that $\partial\Omega$ is a C^0 orientable $(d-1)$-dimensional manifold, and every face $\partial\Omega_i$ is a subset of an analytic or algebraic hypersurface ($\omega_i = 0$). Two-dimensional polygons and three-dimensional polyhedra (curved or linear) are widely recognized examples of solids. We shall assume that ω_i are known, which is the case for polyhedral solids or solids bounded by second-degree surfaces (but may not be true for more general solids bounded by parametric surfaces).

If a solid Ω is not describable by halfspaces in $\mathcal{H}$, there is at least one join-irreducible element J_k that intersects both the interior $\mathbf{i}\Omega$ and the exterior $\mathbf{e}\Omega$. This means that for some points $\mathbf{p} \in \mathbf{i}\Omega$ and $\mathbf{q} \in \mathbf{e}\Omega$, $\omega_i(\mathbf{p})$ and $\omega_i(\mathbf{q})$ have the same sign for *all* primitive functions ω_i used to define the halfspaces in $\mathcal{H}$. On the other hand, because Ω is a solid, such points $\mathbf{p}$ and $\mathbf{q}$ must be separated by the boundary $\partial\Omega$. Figure 5.8 demonstrates the situation for a simple two-dimensional solid shape. These observations suggest that the construction of the additional separating halfspaces $\mathcal{G}$ can be guided by the faces $\partial\Omega_i$ in the boundary representation. For example, the following result is proved in (Shapiro 1991).

Theorem 6. Let Ω be a solid, $\{\partial\Omega_i\}$ a set of faces in the boundary representation, and $\mathcal{H}$ a set of halfspaces induced from the faces. Suppose that the interior of every face $\partial\Omega_i$ is separated from the rest of the surface $(\omega_i = 0) \setminus \partial\Omega_i$ by a family $\mathcal{G}$ of linear halfspaces ($g_k \geq 0$). Then Ω is describable by $\mathcal{H} \cup \mathcal{G}$.

If Ω is a curved polygon with edges that do not change their sign of curva-

ture, Theorem 6 implies that the polygon is describable by the halfplanes $\mathcal{H}$ induced from the polygon's edges and the linear halfplanes $\mathcal{G}$ associated with polygon's chords (Shapiro and Vossler 1991*b*). Figure 5.9 illustrates this result on a simple 2D solid. Notice that in this case, the additional halfspaces are not necessary, because Ω is the union of three circular halfspaces. For



(a) The boundary $\partial\Omega$ is a union of three circular arcs.

(b) Separating halfplanes associated with chords

Figure 5.9. Illustration of Theorem 6. A set of halfspaces associated with the chords is sufficient but not always necessary for describability of Ω and its pseudo-complement $\mathbf{k}(-\Omega)$.

three-dimensional solids, the theorem requires all faces to be bounded by planar curves, but the construction of the linear halfspaces $\mathcal{G}$ may be fully automated in many cases. Linear separators may also suffice when a solid's boundary contains non-planar edges, but in general this is not true. For example, when the functions ω_i are polynomials of degree k, the degree of separating primitives may be $\geq k/2$. Shapiro and Vossler (1993) used the above results to design and implement a fully automated procedure for constructing set representations for solids bounded by second-degree surfaces.

Optimization and signed functions

Once Ω is known to be describable by halfspaces in $\mathcal{H} \cup \mathcal{G}$, it could be represented in the canonical disjunctive form (5.4), but that is probably the most inefficient way to represent Ω. There are at least two different possibilities for optimization. First, the constructed set $\mathcal{G}$ is sufficient, but not all halfspaces in $\mathcal{G}$ are usually necessary for representing Ω. A (non-unique) minimal set $\mathcal{G}$ may be determined by incrementally removing halfspaces from $\mathcal{G}$ until the conditions of Theorem 5 are violated. In the example of Figure 5.9, all chordal halfspaces would be removed. Secondly, standard Boolean optimization techniques (Lawler 1964) may be used to optimize the constructed function $\mathbf{\Phi}$.

Note that one can write down 2^n distinct intersection terms, but there are

only a polynomial number of non-empty join-irreducible elements J_k in the arrangement $\mathcal{A}\{(\omega_i = 0)\}$. Thus, the canonical form (5.4) may be computed as a decomposition of Ω and represented by a set of characteristic points, one point from each J_k. The intersection terms may be optimized by dropping all those halfspaces that do not bound the particular J_k. Containment relationships between the intersection terms may be exploited to obtain a nearly optimal union of intersection expression, which intuitively correspond to computing a minimal convex cover of Ω (O'Rourke 1982). These observations were used to design practical algorithms for optimizing set representations for polygons and solids in (Shapiro and Vossler 1991*a*, Shapiro and Vossler 1991*b*).

A more serious problem is that the function ω_Φ obtained from the optimized expression $\mathbf{\Phi}$ via Theorem 3 may not be properly signed. It is true that $\partial\Omega \subseteq (\omega_\Phi = 0)$, but ω_Φ may *also* be zero at some points in the interior $\mathbf{i}\Omega$. In Figure 5.9, if $\mathbf{\Phi}$ is the union of three circular disks, then $\omega_\Phi = 0$ at the point where all three circles intersect. Recall from section 3 that the root cause of this problem lies in the adopted definition of R-functions that does not distinguish between zero and the positive numbers. On the other hand, notice that ω_ϕ is strictly negative at all points $\mathbf{p} \notin \Omega$. Similarly, if $\mathbf{\Upsilon}$ is a set representation for the pseudo complement of Ω defined as $\mathbf{k}(-\Omega)$, then the function $-\omega_\Upsilon > 0$ on all points $\mathbf{p} \in \mathbf{i}\Omega$. It follows that the function $\omega_\Phi - \omega_\Upsilon$ is the signed normalized function for $\partial\Omega$ solving the problem P2 (Shapiro 1994). Notice that Theorem 6 applies to both Ω and its pseudo-complement. For example, the chordal halfplanes in Figure 5.9(b) are also sufficient for representing $\mathbf{k}(-\Omega)$ and, in this case, they are also necessary.

The above solution is general, but is inelegant and expensive because it requires constructing set representations twice: once for the set Ω and again for its pseudo-complement $\mathbf{k}(-\Omega)$. It is reasonable to ask if this double effort may be avoided and, if so, under what conditions. For example, the function $-\omega_\Upsilon$ may not be properly signed outside of Ω, but it has all the properties required by the Kantorovich's method for solving boundary value problems and its generalizations, as described in Section 7. For solid models, Shapiro (1999) showed that the function ω_Φ is properly signed if and only if the set expression $\mathbf{\Phi}$ and its dual[6] represent closed regular sets. Examples of such set expressions include set expressions for simple polygons computed by the recursive decomposition algorithm in Section 5.3, and monotone set expressions using primitives from a simple arrangement.

[6] A dual of a set expression $\mathbf{\Phi}$ is obtained from $\mathbf{\Phi}$ by complementing all primitives and changing every $\cap$ to $\cup$ and vice versa.

6. Geometric modeling

The solution to the inverse problem of analytic geometry afforded by R-functions encodes the geometric information in terms of sufficiently smooth and normalized real-valued functions. This, in turn, allows reformulating many geometry-intensive computational problems in terms of simpler problems that can be solved using classical tools and algorithms for dealing with such functions. This section briefly surveys several such applications, with particular focus on those areas where significant advances depend specifically on properties of R-functions.

The theory of R-functions remained largely unknown to western researchers until the late 1980s, and even after its initial exposure, many similar concepts have been developed independently in the former Soviet Union and in the west. This brief survey does not attempt to be comprehensive or to establish priority among the proposed ideas. Over the last thirty years, geometric modeling blossomed as a discipline, with implicit shape representations the form $\omega = 0$ becoming increasingly popular and now used widely. Many construction methods for such representations are available as described, for example, by Bloomenthal (1997), Velho, Gomes and de Figueiredo (2002), and others.

The popularity of implicit representations can be attributed to several factors. It is conceptually a very simple representation of a shape that determines point membership via the sign of the defining function ω. It puts no restrictions on the topological properties of the represented sets, but a variety of computational techniques have been developed to parameterize and render the set boundaries when they are manifold. These include the marching cube algorithm (Lorensen and Cline 1987), polygonization (Bloomenthal 1988), and other numerical continuation methods for piecewise-linear approximation of manifolds (Allgower and Georg 1990). Volumetric scan-conversion of such shapes is achieved via ray casting (Roth 1982), computed by intersecting (a grid of) lines with the implicit representation ($\omega = 0$).

6.1. Point membership classification

Computer modeling of complex shapes (point sets) as set combinations of simpler point sets was called "constructive geometry" by Ricci (1973) who suggested that a 3D point set may be represented as $X = f^{-1}(0, 1]$ where f is a non-negative real-valued function. He then observed that the set (pseudo-)complement is defined by $1/f$, the set operations $\cap$ and $\cup$ can be encoded in terms of max and min functions respectively, and proposed their L_p norm and polynomial approximations for construction of increasingly complex shapes and images. Ricci's representations have remained popular in many geometric modeling application; for example, see (Storti, Ganter and Nevrinceanu 1992) and (Blechschmidt and Nagasuru 1990). More gen-

erally, the wide use of Constructive Solid Geometry (CSG) (Requicha and Voelcker 1977) also promoted the use of min/max operators, for example for scan converting CSG representations into a volumetrically defined Euclidean distance map (Breen, Mauch and Whitaker 1998), or for reformulating the boundary evaluation problem as a level-set marching method (Sethian 1996).

Translation of the theory of R-functions into English (Shapiro 1988) led to a widespread adoption of other systems of R-functions for computer graphics and shape modeling applications. Notably, Pasko et al. (1995) developed a powerful geometric language for computer graphics based on R-functions, which inspired others to add R-functions as basic geometric operations for combining functionally (implicitly) defined point sets, for example (Wyvill et al. 1999, Fougerolle, Gribok, Foufou, Truchetet and Abidi 2005). Use of R-functions in solid modeling remains limited, partly because associating R-functions with *regularized* set operations is technically incorrect. The resulting sets ($\omega = 0$) may not be closed regular and may include points in the interior of the solid, as we already discovered in Section 5.3. On the other hand, such non-regular and interior points may be identified from extrema properties of the constructed function ω, and every solid shape may be represented by a sufficiently smooth signed function ω using methods described in section 5.3 (Shapiro 1994).

6.2. Blending

Unions and intersections of smooth shapes create sharp edges and corners, which may be undesirable in many applications. Such sharp features are the source of stress concentrations and other singular behaviors, cannot be manufactured by many manufacturing processes (for example, metal casting, stamping, etc.), and may not be aesthetically pleasing. The procedure for smoothing the sharp edges and corners is commonly known as *blending* (see (Woodwark 1988) for an introductory survey of blending techniques). Ricci's operations and other polynomial approximations of R-functions are examples of *global* blending because the smoothing affects all points of the shape's boundary. In most practical situations, it is desirable to blend sharp features *locally* and in range-controlled fashion, for example with the desired radius as a function of distance from the feature, so that the points some distance away are not affected by the blends.

A general formulation for blending was proposed by Rockwood and Owen (1987). Suppose we want to blend the intersection of two implicitly defined primitive shapes ($\omega_i \geq 0$), $i = 1, 2$. Construct a binary blend function $B_{12} : \mathbb{R}^2 \to \mathbb{R}$ that blends the intersection of two linear halfspaces, say ($x \geq 0$) and ($y \geq 0$). Then the blend of the desired intersection is simply $B_{12}(\omega_1, \omega_2)$. A variety of blending functions B_{12} can be used, including superelliptic, circular, variable-radius, and others. It is also observed that

n-primitive blending functions $B_{12\ldots n}$ can be constructed directly, or if a solid is defined constructively using set operations on implicitly defined primitives, then simply substituting B_{12} for the intersection and $-B_{12}(-\omega_1, -\omega_2)$ for the union blends all sharp features in the resulting shape. Rockwood (1989) explains (and resolves) several difficulties with this approach, including the dependence of the blend on the metric properties of the defining functions and how the blend itself is constructed in different regions of $\mathbb{R}^2$.

The concept of blending may seem to be at odds with R-functions, because the sharp features (corners) is why we needed R-functions in the first place. But in fact, the blending functions B_{12} are simply R-functions modified in the neighborhood of their zero set. This view was made explicit by Pasko, Pasko, Ikeda and Kunii (2002) who proposed to modify R-functions as:

$$B_{12} = x_1 \odot x_2 + d(x_1, x_2), \tag{6.1}$$

where $\odot$ is any binary R-function, and $d(x_1, x_2)$ is a "displacement" function whose behavior defines the actual blend. If d is a non-negative function, it affects the shape globally. If the blend is to be restricted to some neighborhood of the corner, the displacement function $d(x_1, x_2)$ must vanish outside the neighborhood of the origin. Such displacement functions have been proposed in (Pasko et al. 2002) and more complex transition functions that generalize the notion of blending are proposed in (Barthe, Wyvill and De Groot 2004) and in (Fayolle, Pasko, Schmitt and Mirenkov 2006).

Viewing blending and transition operations in terms of R-functions is particularly appealing because R_p-functions behave as approximate distances that are smooth everywhere except at the origin, which corresponds exactly to the sharp feature to be blended. Following Theorem 4, composition of such blends preserves normalization properties of the primitive functions ω_i, providing a convenient mechanism for controlling the shape of the composite blends in the vicinity of sharp features. Furthermore, it is easy to see that (6.1) is an instance of a *conditional* R-function introduced in Section 3.4, under the condition that $d(x_1, x_2) = 0$. Other conditional R-functions can be used to define powerful blending techniques. For example, consider a ρ-blending operation corrected from (Sheiko 1982):

$$B_{12} = x_1 + x_2 + s\sqrt{x_1^2 + x_2^2 + \frac{1}{8\rho^2} s_\rho (s_\rho - |s_\rho|)}, \tag{6.2}$$

with $s_\rho = x_1^2 + x_2^2 - \rho^2$. Comparing (6.2) with (3.7), it should be clear that in this case B_{12} is a conditional R_0-function. Everywhere outside the circular region of radius ρ centered at the origin, the term $s_\rho - |s_\rho|$ vanishes, making B_{12} a disjunction when $s = 1$ and a conjunction when $s = -1$. Figure 6.10 shows the result of normalizing all primitives and replacing all R-functions in example 4.1 by the conditional R-function B_{12} from (6.2) with $\rho = 0.25$.

(a) The original model from 4.3(b) (b) The normalized and blended model

Figure 6.10. The isosurface of the pawn in 4.3(b) after all primitives are normalized and all R-functions are replaced by the conditional R-functions B_{12} with $\rho = 0.25$

Notice that the overall shape of the pawn is virtually unchanged, but all the sharp edges are now replaced with smooth blends.

6.3. Envelopes and projections

Beyond the obvious applications in 3D Euclidean space, R-functions can and have been used in more abstract settings. For example, Pasko et al. (1995) describe a general multi-dimensional framework for geometric modeling that includes a number of advanced operations relying on R-functions. If $\Omega_1 \subset \mathbb{R}^k$ is defined by $f_1 : \mathbb{R}^k \to \mathbb{R}$ and $\Omega_2 \subset \mathbb{R}^m$ is defined by $f_2 : \mathbb{R}^m \to \mathbb{R}$, then the Cartesian product $\Omega_3 \subset \mathbb{R}^{k+m}$ is immediately given by an R-conjunction operation:

$$\Omega_1 \times \Omega_2 = f_1 \wedge^* f_2.$$

If $F : \mathbb{R}^n \to \mathbb{R}$, and $\Omega = (F \geq 0) \subset \mathbb{R}^n$, then a *section* of Ω is obtained by assigning a fixed value K to the ith variable x_i. For m values of $K_{j+1} = K_j + \Delta x_i$ that are spaced apart by some Δx_i, we end up with a stack of sections C_{ij}, $j = 1, \ldots, m$. As $\Delta x_i \to 0$, R-disjunction $\bigvee_j^* C_{ij}$ converges to the *projection* of Ω on the bounded interval of $\mathbb{R}^{n-1}$. Instead of taking the limit, the R-disjunction of m sections may be blended together using one of the blending operations described above. These and other derived functional operators provide a powerful arsenal for modeling point sets in a multi-dimensional setting. For example, the *sweep* of a shape Ω that is being transformed by a one-parameter affine transformation M_t, $t \in [t_0, t_1]$ is defined as $\bigcup_{q \in M_t} \Omega_q$, where Ω_q is a transformed instance of Ω by transformation q. Sweep is one of the fundamental operations in solid modeling, with many applications in mechanical design and manufacturing (Abdel-Malek, Blackmore and Joy 2006). Sourin and

Pasko (1995) show how the general sweep may be formulated and computed using R-functions. If $\Omega = (F(p) \geq 0) \subset \mathbb{R}^n$ is a static shape, then the dynamic shape $(F(p,t) \geq 0) \subset \mathbb{R}^{n+1}$ is obtained by the coordinate substitution $p \mapsto M_t^{-1}(p)$. Computing the sweep amounts to representing the projection on $\mathbb{R}^n$ of the $(n+1)$-dimensional point set defined using R-functions as

$$F(p,t) \wedge^* (t - t_0) \wedge^* (t_1 - t),$$

and evaluating its boundary numerically, based on the requirement that $\partial F/\partial t = 0$. This technique can also be used to compute the boundary of the dual infinite intersection $\bigcap_{q \in M_t} \Omega_q$ operation described by Ilies and Shapiro (1999).

6.4. Symmetric and periodic coordinate transformations

Coordinate transformations, a standard tool in analytic geometry, can be used effectively with R-functions. If $\omega(x_1, \ldots, x_n)$ is a normalized function defining Ω, then $\omega(\mu_1, \ldots, \mu_n)$ is constructed by applying the coordinate transformations $\mu_i : \mathbb{R}^k \to \mathbb{R}$ to some or all variables x_i. The notion of the dynamic shape $(F(p,t) \geq 0)$ in the sweep operation described above relied on such a coordinate transformation, but a more familiar form involves sweeping a two-dimensional shape $(\omega(x_1, x_2) \geq 0) \subset \mathbb{R}^2$ in $\mathbb{R}^3$ by coordinate substitution $x_i \mapsto \mu_i(x_1, x_2, x_3)$. When $\mu_2 = \sqrt{x_2^2 + x_3^2}$, the inequality $(\omega(x_1, \mu_2) \geq 0)$ defines a body of revolution; when $\mu_i = x_i - \beta_i x_3$, $(\omega(\mu_1, \mu_2) \geq 0)$ is the prismatic body swept in the direction $(\beta_1, \beta_2, 1)$, and a general screw sweep when

$$\begin{aligned} \mu_1 &= x \cos\phi(x_3) + y \sin\phi(x_3) + c_1(x_3); \\ \mu_2 &= x \cos\phi(x_3) - y \sin\phi(x_3) + c_2(x_3), \end{aligned} \tag{6.3}$$

where $\phi(x_3)$ and $c_i(x_3)$ are the parameters of the screw motion around the x_3-axis. Rvachev (1982) studied these and other coordinate transformations that construct 3D shapes from 2D sections, modifying them as needed to ensure differential and normalization properties of the transformed function $\omega(\mu_1, \ldots, \mu_n)$.

When the μ_is specify an isometry, they can be used to make copies of Ω that can be combined with Ω using R-functions. If $\mu_i(x_i)$ are periodic functions, then the shape $(\omega(\mu_1, \ldots, \mu_n) \geq 0)$ inherits the discrete symmetries of μ_i. For example, suppose that Ω is symmetric with respect to the coordinate axis and fits inside a coordinate box of width $2a$. Define the ith coordinate transformation $\mu_i(x_i) = x_i$ on the interval $-a \leq x_i \leq a$ and require it to be a periodic function with period greater than $2a$. Then the set $(\omega(x_1, \ldots, \mu_i(x_i), \ldots x_n) \geq 0)$ has translational symmetry along the x_i-axis, reproducing the base shape Ω at regular intervals. The simplest function μ satisfying this requirement is the 45° saw-tooth pattern, but smooth approx-

imations may be constructed using Fourier series, or by methods described by Rvachev (1974).

Rvachev, Sheiko and Shapiro (1999) observed that more complex coordinate transformation functions μ_i may be constructed as semi-analytic compositions of the above primitive coordinate transformations and R-functions. Such coordinate transformations may prescribe periodic, symmetric, and random properties that explicitly depend on the values of the coordinates x_i and additional external parameters. Assuming the typical situation where the complexity of the constructed normalized function ω significantly exceeds that of the coordinate transformations, this approach leads to constructions that are not only more intuitive, but also more efficient. Maksimenko-Sheiko, Matsevity and Sheiko (2005) also used this technique to construct piecewise functions $\phi(x_3)$ in the coordinate transformations (6.3).

Example 6.1. The 2D portion of the flag in Example 4.2 is defined by $(\omega(\mathbf{p}) = 0)$, where $\omega = \omega_1 \wedge_0 \omega_2 \wedge_0 \omega_3 \wedge_0 \omega_4$, and all functions ω_i have been normalized to the first order. Figure 6.11(b) shows the plot of the normalized function $\omega(M(\mathbf{p}))$, where $M = M_2 \circ M_1$ is a composition of two coordinate transformations:

$$M_2 = (x - 2h) \wedge_1 \mu_2(x) \wedge_1 (l - x); \qquad M_1 = Rot(\mu_1(\theta)).$$

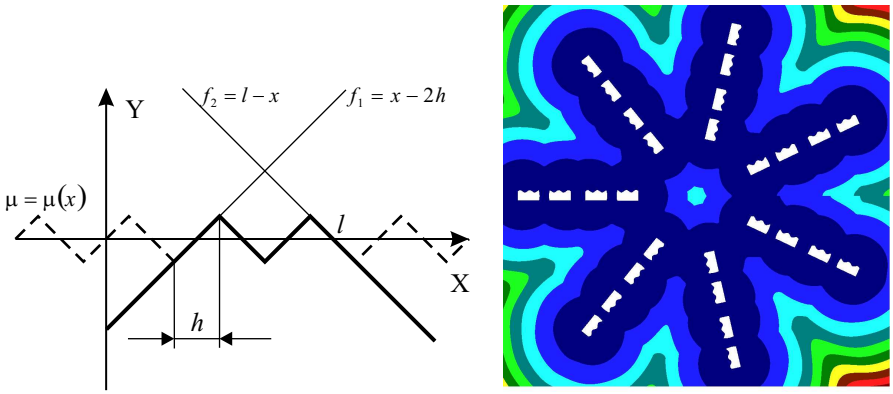
The functions μ_1 and μ_2 are Fourier series approximations of the function μ shown in Figure 6.11(a). In this particular example, the function μ_1 is shifted by $\pi/7$, has a period of $4\pi/7$, and is a composition of rotation and reflection as seen in Figure 6.11(b). Transformation M_2 uses the R_1-conjunction to truncate the periodic transformation $\mu_2 = \mu$ as shown in Figure 6.11(a).

6.5. Planning and design

R-functions and derived constructions, such as those described above and others, have been used extensively in a variety of shape modeling applications, for example in mechanical design (Kutsenko 1990, Ensz, Storti and Ganter 1998), robot motion planning (Shkel 1997, Rimon and Koditschek 1990), hair modeling (Sourin, Pasko and Savchenko 1996), Monte Carlo models of transport phenomena (Altiparmakov and Belicev 1990), and stochastic optimization techniques (Komkov 1989) that require repeated sampling of the domain and/or boundary. In addition to the usual advantages of implicit representations, parametric and differential properties of R-functions significantly expand both the range and the possibilities of implicit representations. Below, we consider two (not mutually exclusive) situations where modeling a region Ω with R-functions is useful: when Ω represents a constraint in physical or abstract space, and when Ω itself is the object of design and optimization.

A typical mathematical programming problem is to find a point $\mathbf{x}^0 \in \Omega \subset \mathbb{R}^n$ where some objective function $f(\mathbf{x})$ attains a maximum or a minimum value. R-functions allow to represent virtually any constraint region Ω by a single inequality $(\omega \geq 0)$. In geometric modeling applications, the constraint $(\omega \geq 0)$ often describes the subset Ω of physical space that must be avoided or included for design or planning purposes. In robot motion planning, Ω usually corresponds to either obstacle space (to be avoided by the robot) or to free space (where the robot can move). For example, Rimon and Koditschek (1992) use R-functions to construct the artificial potential function ω representing free space Ω; the differential properties of ω are critical because the gradient information is used to navigate through Ω.

In more general situations, the constraint region Ω may be specified as a union of N systems of inequalities $\sigma_{ki}(\mathbf{x}) \geq 0, \quad i = 1, \ldots, M$, where each $(\sigma_{ki} \geq 0)$ is a region Σ_{ik}, and $k = 1, \ldots, N$. In other words, the region of interest is given in disjunctive normal form as $\Omega = \bigcup_k \left(\bigcap_i \Sigma_{ki} \right)$ or, using R-functions, by the inequality $\bigvee_k^* \left(\bigwedge_i^* \sigma_{ki} \right) \geq 0$. For example, when a rigid shape $\Omega_i \subset \mathbb{E}^3$ is represented by $(\omega_i \geq 0)$ and is free to move in space, it can be represented in general location by $(\omega_i(\mathbf{x}, \mathbf{p_i}) \geq 0)$, where $\mathbf{x} \in \mathbb{E}^3$, and $\mathbf{p_i}$ is a vector of its location (position and orientation) parameters. The intersection of two such shapes is given by $\sigma_{ij}(\mathbf{x}, \mathbf{p_i}, \mathbf{p_j}) = (\omega_i(\mathbf{x}, \mathbf{p_i}) \wedge^* \omega_j(\mathbf{x}, \mathbf{p_j})) \geq 0$. It follows that the pairwise non-interference condition for the two objects can be formulated as the requirement that $-\max_{\mathbf{x}} \sigma_{ij}(\mathbf{x}, \mathbf{p_i}, \mathbf{p_j}) \geq 0$. The non-interference conditions generalize in an obvious fashion to collections of n rigid shapes and to the unions of rigid shapes, using R-conjunctions and R-disjunctions on pairwise condition functions σ_{ij}. These ideas were first



(a) Periodic transformation μ may be truncated

(b) Function $\omega(M(\mathbf{p}))$ from Example 6.1

Figure 6.11. Coordinate transformations may be constructed using compositions of symmetric and periodic functions with R-functions.

discussed in (Rvachev 1967), and further developed in (Stoian 1975, Stoian and Panasenko 1978) with many applications to problems of optimal placement, blank nesting, and packing.

When it is possible to estimate that the optimal value of the objective function $f(x)$ is $z_0 \in [z_{\min}, z_{\max}]$, R-functions can be used to transform the original constrained problem into a sequence of unconstrained optimization problems. Suppose we are looking for the minimum, and consider the region $Q(z) = \Omega \cap (z - f(\mathbf{x}) \geq 0)$, also defined by the inequality

$$q(\mathbf{x}, z) = \omega(\mathbf{x}) \wedge_\alpha (z - f(\mathbf{x})) \geq 0. \tag{6.4}$$

For a fixed value $z > z_0$, the region $Q(z)$ has interior points, and if $z < z_0$, then $Q(z)$ is an empty set $\emptyset$. Assuming that $Q(z)$ is bounded, and ω, f are continuous functions, it follows that the maximum value of the objective function $f(\mathbf{x})$ is achieved when $q_0(z) = \max_{\mathbf{x}} q(\mathbf{x}, z) = 0$. Note that $q_0(z)$ is monotone in z. Thus, the original problem is transformed into a sequence of optimization problems (each problem is defined by a fixed value of $z \in [z_{\min}, z_{\max}]$) of the function $q(\mathbf{x}, z)$ without any constraints on $\mathbf{x}$. Additional details can be found in (Rvachev 1967, Rvachev 1982).

6.6. Shape optimization

In shape design and optimization problems, Ω is not a constraint, but is the object of study. In this case, the shape is parameterized as $\Omega(\mathbf{b})$, and the challenge is to determine the values of parameters $\mathbf{b} = \{b_1, \ldots, b_k\}$ that will optimize some objective function $F(\Omega(\mathbf{b}))$, for example, volume, energy, stress, etc. Shape optimization involves three tasks: computation of sensitivity dF/db_i, updating the model $\Omega(\mathbf{b})$, and (re)evaluating the objective function F; each of the tasks is simplified using R-functions. Consider a generic case, when the objective function F is defined on Ω as $F = \int_\Omega f d\Omega$. Then, it can be shown (Chen, Freytag and Shapiro 2007) that computation of sensitivity requires calculation of

$$\int_{\partial\Omega} \frac{f}{|\nabla\omega|} \frac{\partial\omega}{\partial b_i} d\Gamma = \sum_{k=1}^{N} \int_{\partial\Omega_k} \frac{f}{|\nabla\omega_k|} \frac{\partial\omega_k}{\partial b_i} d\Gamma, \tag{6.5}$$

where $\partial\Omega = (\omega = 0)$, and $\partial\Omega_k = (\omega_k = 0)$ is the portion of the boundary that belongs to the kth primitive used to construct the domain Ω and is affected by the parameter b_i. This result is implied by the mere *existence* of the R_p-construction for ω in terms of the primitive functions ω_k, and Theorem 4 – even if this construction may not be known explicitly. On the other hand, if the domain is indeed represented by an R-function on n primitives

$$\omega(\omega_1, \ldots, \omega_k(b_i), \ldots \omega_n) \geq 0, \tag{6.6}$$

then updating the geometric model $\Omega(\mathbf{b})$ is simply a matter of syntactically updating the parameter b_i. Furthermore, when the objective function $F(\Omega(\mathbf{b}))$ can be computed directly from the implicit representation $\omega = 0$, it automatically inherits the same parametrization and may be (re)evaluated for any value of b_i. A great advantage of this approach to shape optimization is that it places no artificial constraints on the topology of Ω, which is free to change during the optimization process.

The above techniques work well when the objective function F is volume, mass, surface, and moments of inertia, but they also apply to more general situations where F may depend on the solution of a boundary value problem defined over Ω. Suppose that boundary conditions ϕ_k are prescribed on the portion of boundary Ω_k. In this case, Shapiro and Tsukanov (1999*b*) observed that parametrization of $\Omega(\mathbf{b})$ as above also induces a parametrization of the solution structure to the boundary value problem $u = B(\omega, \omega_k, \phi_k)[\Psi]$, where B can be viewed as an operator and Ψ is a suitable set of basis functions.[7] Such a parametrization supports fully automated (re)evaluation of boundary value problems, which is particularly effective for problems with deforming domains and moving boundary conditions. Chen, Shapiro, Suresh and Tsukanov (2006) show that the computation of sensitivity for a large class of boundary value problems also reduces to boundary integration over primitive boundaries ($\omega_k = 0$). They also showed that the representation of Ω by (6.6) can include primitive halfspaces ($\omega_k \geq 0$) that define free-form boundaries and/or spatial constraints, leading to fully automated procedures for shape and topology optimization with parametric control.

7. Boundary value problems

Perhaps the most significant application of R-functions has been in the area of boundary value problems – the area that motivated Rvachev to invent the concept of R-functions and that he himself has always included under the general auspices of the "theory of R-functions". We summarize the key ideas in this section.

7.1. Generalized Taylor series

The classical Taylor formula approximates a function in the neighborhood of a given point x_0 by a polynomial in $(x - x_0)$. The neighborhood itself is described by the term $x - x_0$ which can be thought of as a one-dimensional distance function that vanishes at the point x_0. The generalized Taylor series expansion introduced by Rvachev (1974) represents a function u in the neighborhood of the boundary $\partial\Omega$, as a polynomial in (powers of) distance

[7] The concept of solution structure is explained in Section 7.

to the boundary $\partial\Omega$. Suppose that $\partial\Omega$ is described by some function ω that vanishes on $\partial\Omega$ and is normalized to order m. Then for a *known* function u:

$$u = u(0) + \sum_{k=1}^{m} \frac{1}{k!} u_k(0)\omega^k + O\left(\omega^{m+1}\right), \tag{7.1}$$

where $u_k = \frac{\partial^k u}{\partial \omega^k} = D_\nu^k u$ (since ω is normalized) are evaluated at the boundary $\partial\Omega$, in the direction ν normal to $\partial\Omega$.

In most applications, the function u is not known, but must be reconstructed from its values and/or derivatives f_k specified at the boundary $\partial\Omega$. In order to be used as coefficients in the generalized Taylor series (7.1), not only must the functions f_k be defined everywhere, but they must also behave as constants in the direction normal to the boundary. This is achieved by conditioning the specified functions f_k through the coordinate transformation

$$f^*(\mathbf{x}) = f(\mathbf{x} - \omega\nabla\omega). \tag{7.2}$$

Since ω is normalized, the result of this coordinate transformation is that the value of f_k^* at any point near $\partial\Omega$ is determined by the closest point on $\partial\Omega$ and

$$f^*(\mathbf{x})_{|\partial\Omega} = f(\mathbf{x})_{|\partial\Omega}; \qquad \frac{\partial^k f^*}{\partial \nu^k}_{|\partial\Omega} = 0; \qquad (k = 1, 2, ..., m). \tag{7.3}$$

The functions f_k^* are called *normalizers* of functions f_k by ω in (Rvachev 1974), and are used as coefficients in the generalized Taylor series expansion. Another technique to construct normalizers is described by Rvachev and Sheiko (1995).

Theorem 7. (Rvachev 1982) If the function $\omega(\mathbf{x})$ is normalized up to the m-th order and a function $u(\mathbf{x})$ satisfies conditions

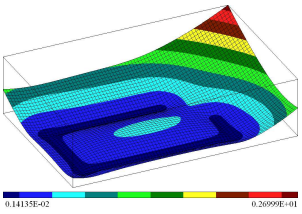
$$u(\mathbf{x})_{|\partial\Omega} = f_0(\mathbf{x}), \quad \frac{\partial^k u}{\partial \nu^k}_{|\partial\Omega} = f_k(\mathbf{x}), \quad (k = 1, 2, ..., m), \tag{7.4}$$

then u can be represented in the neighborhood of the boundary $\partial\Omega$ in the form

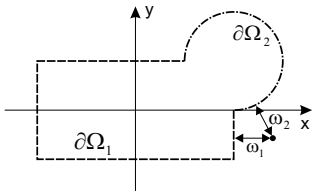
$$u = f_0^* + \sum_{k=1}^{m} \frac{1}{k!} f_k^* \omega^k + O(\omega^{m+1}) \tag{7.5}$$

where $f_k^*(\mathbf{x}), k = 0, 1, ..., m$ are normalizers of the functions $f_k(\mathbf{x}), k = 0, 1, ..., m$ with respect to $\omega(\mathbf{x})$.

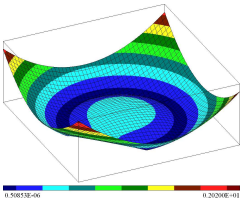
Example 7.1. The boundary of the domain in Figure 7.12 is $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$, where $\partial\Omega_2$ is the portion of the circle. The following boundary conditions are prescribed:



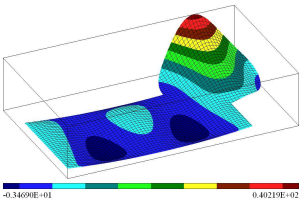
(a) Function ω_1



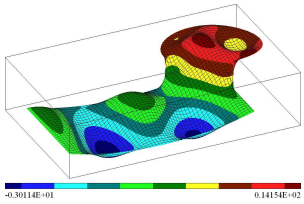
(b) $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$



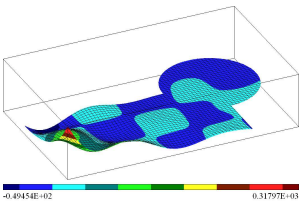
(c) Function ω_2



(d) Taylor polynomial P_1



(e) Function u interpolates P_1 and P_2



(f) Taylor polynomial P_2

Figure 7.12. When portions of the boundary $\partial\Omega_1$ and $\partial\Omega_2$ are represented implicitly by normalized functions $(\omega_1 = 0)$ and $(\omega_2 = 0)$ respectively, interpolating boundary conditions is a matter of syntactic substitution.

$$f|_{\partial\Omega_1} = \underbrace{1 + x^2}_{g_0}, \qquad \left.\frac{\partial f}{\partial \nu}\right|_{\partial\Omega_1} = \underbrace{10\left(\cos\left(\pi x\right) + y\right)}_{g_1}; \tag{7.6}$$

$$f|_{\partial\Omega_2} = \underbrace{10 + (y-2)^2}_{h_0}, \qquad \left.\frac{\partial f}{\partial \nu}\right|_{\partial\Omega_2} = \underbrace{10\sin\left(\pi x\right)\cos\left(\pi y\right)}_{h_1}. \tag{7.7}$$

If the boundaries $\partial\Omega_i$ are represented by the respective normalized functions $(\omega_i = 0)$, extending these boundary conditions into the domain is a matter of syntactic substitution into the truncated Taylor series from (7.5). For $\partial\Omega_1$, we have:

$$\begin{aligned} P_1 &= g_0^* + g_1\omega_1 = 1 + \left(x - \frac{\partial\omega_1}{\partial x}\omega_1\right)^2 + 10\left(\cos\left(\pi x\right) + y\right)\omega_1, \\ P_2 &= h_0^* + h_1\omega_2 = 10 + \left(y - 2 - \frac{\partial\omega_1}{\partial y}\omega_2\right)^2 + 10\sin\left(\pi x\right)\cos\left(\pi y\right)\omega_2. \end{aligned}$$

The two functions are plotted over the domain Ω in Figures 7.12(d) and (f) respectively.

The generalization suggests that many techniques relying on Taylor polynomials in the classical univariate setting may also be applicable in the gen-

eral multi-dimensional setting, once the normalized function ω is constructed for the boundary $\partial\Omega$ of interest. In particular, Rvachev (1967) recognized that Kantorovich's method for representing the solutions to boundary value problems with Dirichlet boundary conditions via (2.6) is a special case of (7.5) with $k = 1$. More generally, in the context of boundary value problems, the prescribed functions f_k correspond to the boundary conditions, while the remainder term $O(\omega^{m+1})$ must be determined to satisfy some additional constraint, for example, a differential equation. Formally, the approach is justified following a generalization of the classical Weierstrass Approximation Theorem, modified from (Kharrik 1963):[8]

Theorem 8. Suppose that Ω is a bounded region of m-dimensional space with boundary $\partial\Omega$. Let $\omega(\mathbf{x}) \in C^s$ be a function defined in an open region that contains Ω and satisfies the following conditions:

1 $\omega(\mathbf{x}) = 0 \iff \mathbf{x} \in \partial\Omega$;
2 derivatives of ω through order s satisfy the Lipschitz condition;
3 $\nabla\omega(\mathbf{x})_{|\mathbf{x}\in\partial\Omega} \neq 0$.

If a function $\gamma \in C^s(\Omega)$ and vanishes on the boundary $\partial\Omega$ together with its derivatives up to order $k < s$, then for any positive ε there exists a polynomial Ψ such that

$$\|\gamma - \omega^{k+1}\Psi\|_{H^s(\Omega)} < \varepsilon. \tag{7.8}$$

We can now rewrite expression (7.5) as

$$u = P + \omega^{k+1}\Psi, \tag{7.9}$$

where P satisfies the boundary conditions $f_i, i = 0, 1, \ldots, k$. If f is the solution to the boundary value problem, the function $\gamma = f - P$ satisfies the conditions of Theorem 8, and approximating the solution to the boundary value problem amounts to choosing the undetermined polynomial Ψ. But before we discuss this task, we deal with yet another challenge: in most practical situations, different boundary conditions are prescribed at different portions $\partial\Omega_i$ of the boundary $\partial\Omega$. These boundary conditions must be somehow interpolated, and once again, the task is greatly simplified using normalized functions ω.

7.2. Transfinite interpolation

For every type of boundary condition specified on a portion of the boundary $\partial\Omega_i$, we can construct a generalized Taylor polynomial P_i of the form (7.5).

[8] Kharrik's original theorem also deals with issues related to the order of approximation and convergence that are outside the scope of this survey.

We would now like to interpolate all of them by a single function $\sum_{i=1}^{n} P_i W_i$. Rvachev (1967) proposed choosing the weights W_i as

$$W_i(\mathbf{x}) = \frac{\omega_i^{-\mu_i}(\mathbf{x})}{\sum_{j=1}^{n} \omega_j^{-\mu_j}(\mathbf{x})} = \frac{\prod_{j=1;j\neq i}^{n} \omega_j^{\mu_j}(\mathbf{x})}{\sum_{k=1}^{n} \prod_{j=1;j\neq k}^{n} \omega_j^{\mu_j}(\mathbf{x})}, \tag{7.10}$$

which can be seen as a form of inverse distance interpolation, where the weights W_i form a partition of unity and are chosen to be inversely proportional to the power μ_i of the approximate distance ω_i from the locus $\partial\Omega_j$. When $\mu_i = 1$, the expression on the right appears to be a form of Lagrange interpolation (2.8). Inverse distance interpolation was used by Shepard (1968) to interpolate scattered point data, but Watson (1992) cites applications of this technique dating back to 1920s. It is well known (Hoschek and Lasser 1993) that the exponents μ_i control the behavior of the interpolating function at the loci $\partial\Omega_i$: when $0 < \mu_i \leq 1$, the interpolant is not differentiable; values of $\mu_1 > 1$ ensure that the interpolant is differentiable $\mu_i - 1$ times at $\partial\Omega_i$.

Rvachev et al. (2001) study the above approach to interpolation and demonstrate its advantages in several applications. Note that (7.10) depends only on the knowledge of the normalized functions ω_i, and places no constraints on the differential or topological properties of the sets $\partial\Omega_i$. Thus, the method may be used without modification to interpolate functions and their derivatives that are specified over arbitrary points, curves, surfaces, or regions – with or without sharp corners – provided they are represented implicitly by normalized functions ω_i. Applications of this technique are found in many areas ranging from geographic information systems to modeling of material properties (Biswas, Shapiro and Tsukanov 2004).

If P_i are generalized Taylor polynomials in the form of (7.5) satisfying the boundary conditions of order k_i on the boundary $\partial\Omega_i$, then they can be interpolated into a single expression using weights (7.10). For example, the two Taylor polynomials P_1 and P_2 in the Example 7.1 are interpolated by

$$u_0 = \frac{P_1\omega_2^2 + P_2\omega_1^2}{\omega_1^2 + \omega_2^2},$$

which is shown in Figure 7.12(e). Theorem 8 may no longer apply because derivatives of different order may be indicated on different portions $\partial\Omega_i$. However, an extension of this result is proved by Rvachev et al. (2000).

Theorem 9. Let Ω be a closed region and $f \in C^s(\Omega)$ be defined in the interior of Ω. Values of the function f and its partial derivatives up to order $k_i < s$ are prescribed on boundaries $\partial\Omega_i \subset \partial\Omega$. Then for any small ε there

exists a polynomial Ψ such that the inequality

$$\|\gamma - \lambda\Psi\|_{H^s(\Omega)} < \varepsilon \tag{7.11}$$

is satisfied, where $\gamma = f - u_0$ is a function that vanishes on $\partial\Omega_i$ together with its partial derivatives up to order k_i, and $\lambda = \prod_{i=1}^{N} \omega_i^{k_i+1}$.

In other words, a general solution to a boundary value problem may be written as

$$u = u_0 + R = \frac{\sum\limits_{i=1}^{n} P_i \omega_i^{-k_i}}{\sum\limits_{i=1}^{n} \omega_i^{-k_i}} + \Psi \prod_{j=1}^{n} \omega_j^{k_j+1}. \tag{7.12}$$

The first term u_0 in the expression satisfies all imposed boundary conditions exactly, while the second term is simply a product of the remainder terms for each individual boundary condition. The power $k_j + 1$ of ω_j indicates that derivatives up to order k_j have been specified on $\partial\Omega_j$. Finding a solution to the boundary value problem reduces to constructing the polynomial Ψ.

7.3. Solution structures of boundary value problems

Theorem 9 suggests a non-traditional approach to solving boundary value problems that generalizes the method of Kantorovich. Given any representation of boundary $\partial\Omega = \bigcup \partial\Omega_i$ and associated boundary conditions, we first construct a normalized function ω_i for each portion of the boundary $\partial\Omega_i$, for example using R-functions and one of the methods described in Section 5. Syntactic substitution into expression (7.5) yields a Taylor polynomial corresponding to the boundary condition on $\partial\Omega_i$. Expression 7.12 interpolates all specified boundary conditions and defines the *solution structure* of the boundary value problem – a space of functions that satisfy all given boundary conditions and differ from each other only in the choice of the undetermined polynomial Ψ. Solving the boundary value problem amounts to choosing Ψ from a sufficiently complete space (multivariate polynomials, B-splines, finite elements, etc.) to approximate the differential equation using least squares, Ritz, or another variational method. For example, the expression

$$u = \underbrace{(\Psi_1 - \omega D_\nu(\Psi_1))}_{f_0^*(0)} + \underbrace{\varphi\omega}_{f_1(0)\omega} + \underbrace{\omega^2\Psi_2}_{O(\omega^2)} \tag{7.13}$$

defines a family of functions that satisfy the Neumann boundary conditions $\frac{\partial u}{\partial \nu}_{|\partial\Omega} = \varphi_0$, where φ is an extension of φ_0 into the interior of domain Ω. The first term f_0 in expression (7.13) represents a value of the function prescribed on the boundary $\partial\Omega$. Since the Neumann boundary condition does not explicitly prescribe the value of u on the boundary, f_0 is represented

by a linear combination Ψ_1 of basis functions with coefficients that will be determined by the numerical solution procedure. Subtraction of $\omega D_\nu(\Psi_1)$ from Ψ_1 ensures that the first normal derivative of u_0 vanishes on the zero set of ω. The second term in expression (7.13) represents a first-order normal derivative of u. This term does not need to be conditioned since no higher order derivatives are prescribed. The remainder term $\omega^2\Psi_2$ guarantees completeness of u.

Consider now a second-order boundary value problem with mixed boundary conditions

$$u_{|\partial\Omega_1} = \varphi_0; \qquad \left(\frac{\partial u}{\partial \nu} + h_0 u\right)_{|\partial\Omega_2} = \psi_0 \tag{7.14}$$

specified on $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$. If $\partial\Omega_i = (\omega_i = 0), i = 1, 2$, and ω_i are normalized functions, the corresponding generalized Taylor series expansions are:

$$\begin{aligned} u_1 &= \varphi + O(\omega_1) = P_1 + O(\omega_1) \\ u_2 &= \Psi_2 - \omega_2 D_{\nu 2}(\Psi_2) - h\omega_2\Psi_2 + \psi\omega_2 + O(\omega_2^2) = P_2 + O(\omega_2^2). \end{aligned}$$

We could now write the corresponding solution structures for u_1 and u_2 and then interpolate them into a common solution structure using weights (7.10). Alternatively, simply substituting into the expression for the general solution structure (7.12) yields

$$\begin{aligned} u &= \frac{P_1\omega_2^2 + P_2\omega_1}{\omega_1 + \omega_2^2} + \Psi\omega_1\omega_2^2 \\ &= \frac{1}{\omega_1 + \omega_2^2}\left(\varphi\omega_2^2 + \omega_1\left(\Psi_2 - \omega_2 D_{\nu 2}(\Psi_2) - h\omega_2\Psi_2 + \psi\omega_2\right)\right) + \Psi\omega_1\omega_2^2. \end{aligned} \tag{7.15}$$

The above procedure for constructing solution structures can be fully automated, but the resulting expressions can often be optimized using special case analysis. Several alternative solution structures for the above second-order boundary value problem with mixed boundary conditions were derived in (Rvachev 1982). Much of the Ukrainian literature on application of R-functions is devoted to the explicit derivation and simplification of solution structures for common boundary value problems. Rvachev and Sheiko (1995) summarize the main results and give a number of illustrative examples.

The transfinite interpolation term in the solution structure (7.12) allows to enforce all imposed boundary conditions in a boundary value problem. But the concept of the solution structure naturally generalizes to include *a priori* known behaviors of the field u by explicitly modifying the remainder term $R = \Psi\prod_{j=1}^{n}\omega_j^{k_j+1}$ in the general solution structure (7.12). Examples

of such behaviors include singularities (often associated with corners and cracks), asymptotic changes (expressed as a function of distance from some boundary), and other empirical and/or postulated principles (such as St. Venant's principle in solid mechanics and Chvorinov's rule in solidification of metal castings). Computationally, it is convenient to associate the construction of the polynomial Ψ with the choice of the basis functions and numerical procedure; on the other hand, the normalized functions ω_i are the natural means for extending the boundary conditions from the boundary into the domain.

A common situation arises at corner points $\mathbf{p}_0 \in (\partial\Omega_i \cap \partial\Omega_j)$, and the field is known to behave as a function τ_j (of distances, angles, etc.) in the neighborhood of the corner points. In this case, modifying the remainder term as $R\prod\tau_j$ captures the singular behavior. This technique was originally applied to torsion problems in (Goncharyuk, Rvachev and Shklyarov 1968), and was revised in the context of solidification in metal casting using (Rvachev, Sheiko, Shapiro and Uicker 1997), using R_p-equivalence operation (5.3) in place of multiplication xy. Suppose that the domain's boundary is decomposed into n smooth segments $\partial\Omega_i$ and "sharp features" contained in $\partial\Omega_i \cap \partial\Omega_j$. Both the smooth segments and the sharp features are represented implicitly by $(\omega_i = 0)$ with normalized functions ω_i. Then, following Theorem 4, the properties of the composite function

$$\omega = \frac{\omega_1}{m_1} \sim_p \frac{\omega_2}{m_2} \sim_p \ldots \sim_p \frac{\omega_n}{m_n}$$

are conveniently controlled by parameters m_i, because R_p-functions preserve the differential properties of the arguments up to order p. If $(\omega_i = 0)$ is a smooth boundary, then m_i defines the gradient of ω at a regular point of $\partial\Omega_i$; if $(\omega_i = 0)$ is a sharp feature, then m_i specifies the behavior of ω as a function of the angle at the ith corner. A given normalized function ω_i may also be scaled

$$\omega_i' = \frac{\omega_i}{1 + (\beta\omega_i)^\alpha}; \quad \text{and/or} \quad \omega_i' = \omega_i \wedge^* \gamma(\mathbf{p})$$

with parameters α, β, γ controlling its magnitude within the domain, while preserving the normalization of ω_i'. Sheiko (1982) and Rvachev (1982) discuss these and several other methods for including *a priori* information in a solution structure for problems with corners, cracks, and interfaces. Completeness of the modified solution structures is considered in (Rvachev and Mikhal' 2001).

Numerous other solution structures for many common boundary value problems have been derived: elasticity (Rvachev and Sinekop 1990), vibration and stability of plates and shells (Rvachev et al. 1973, Rvachev and Kurpa 1987), heat transfer (Rvachev and Slesarenko 1976, Rvachev, Slesarenko and Safonov 1993), fluid dynamics (Tsukanov, Shapiro and Zhang

2003, Maksimenko-Sheiko and Sheiko 2005), thermo-elasticity (Rvachev, Sinekop and Molotkov 1991), contact (Rvachev, Sinekop and Molotkov 1992), diffraction (Gulyayev, Kravchenko, Rvachev and Sizova 1995), heterogeneous media (Tsukanov 2002, Tsukanov and Shapiro 2005), time-varying domains (Shapiro and Tsukanov 1999*b*), and many others.

7.4. Meshfree modeling and analysis with RFM

The approach outlined above was termed the R-Function Method, or RFM, by Rvachev. In retrospect, it seems more appropriate to interpret RFM as 'Rvachev's Function Method', in recognition that it does not directly rely on the use of R-functions since normalized functions may be constructed by other means. RFM offers a number of computational advantages, as described in (Shapiro and Tsukanov 1999*b*, Rvachev et al. 2000), including the ability to satisfy all prescribed boundary conditions exactly (on the zero set of the normalized functions ω_i) without any spatial discretization. This qualifies RFM as essentially a *meshfree* method, even though background meshes may be used for integration and/or visualization purposes.

One of the very first meshfree systems based on RFM was a software system called POLYE (which means "field" in Russian) developed in Ukraine in the 1970s and 1980s, specifically for solving 2D boundary value problems (Rvachev and Shevchenko 1988, Rvachev, Manko and Shevchenko 1986, Rvachev and Manko 1983) using R-functions. In POLYE, the geometric domain, boundary conditions, and solution structure were described in a programming language RL. A typical RL program contained geometric description of the domain in terms of predefined or user-specified analytic primitives ($\omega_i = 0$) and R-functions, explicit declaration of the solution structure, as well as detailed specification of the solution procedure, including the number and type of basis functions to represent Ψ, the numerical procedure (such as Ritz or least square method), integration parameters, and so on. At run time, the solution structure was automatically differentiated and numerically integrated over the background mesh in order to assemble the corresponding linear system. The solution for Ψ was substituted back into the solution structure and visualized. Both integration and visualization algorithms utilized variants of the marching cube algorithms (Shevchenko and Tsukanov 1994, Rvachev, Shevchenko and Veretel'nik 1994).

POLYE served as an early prototype for several more advanced systems. The first fully automated system, SAGE (Shapiro and Tsukanov 1999*a*, Tsukanov and Shapiro 2002), algorithmically constructed all required normalized functions from boundary and/or Constructive Solid Geometry (CSG) representations using R-functions as described in Section 5, and automatically assembled the solution structure implied by the indicated boundary conditions. Greatly improved algorithms for automatic differentiation

(Tsukanov and Hall 2003) and numerical integration resulted in performance that is competitive with mesh-based methods. The architecture of a general purpose meshfree system is described in detail in (Tsukanov and Shapiro 2002), where 3D applications of RFM are also demonstrated. A variety of basis functions may be chosen for approximating the polynomial Ψ, including multivariate polynomials, multi-resolution B-splines, trigonometric polynomials and so on. Fully automated meshfree technology has been applied to a variety of boundary value problems, ranging from thermal conduction, linear elasticity, vibration, and bending, to more challenging problems such as fluid dynamics (Tsukanov et al. 2003) and thermoelasticity in domains with heterogeneous materials (Tsukanov and Shapiro 2005). The single most difficult task in 3D remains the automatic construction of the normalized distance functions, because the practical algorithms based *solely* on R-functions tend to be limited and inefficient. Freytag et al. (2006) recently demonstrated that RFM can be combined with approximate distance fields that are sampled directly from any 3D geometric representation, using R-functions only when set operations are explicitly required.

The fundamental ideas of RFM are also used by other meshfree and meshless methods, as surveyed by Babuška, Banerjee and Osborn (2003). A major challenge for all such methods is imposition of boundary conditions, and Dirichlet boundary conditions in particular, in the absence of a mesh. The so called 'characteristic function method' for satisfying the boundary conditions relies on the Dirichlet solution structure (2.6) used by Kantorovich, and is becoming increasingly popular due to R-functions. Notably, in the WEB-splines method proposed by Höllig (2003), the undetermined polynomial Ψ is constructed using uniform multivariate B-splines that are extended based on their location with respect to the domain's boundary $\partial\Omega$ in order to ensure the stability of numerical computations. The weight function ω is constructed using R-functions on numerically constructed primitive functions of the form

$$\omega_i = 1 - \max(0, 1 - \mathrm{dist}(x, \partial D)/\delta)^\gamma.$$

With the natural neighbour Galerkin method, Laguardia, Cueto and Doblare (2005) rely on the same solution structure, but ω is constructed using R-functions on analytically defined primitives, while Ψ is constructed using Sibson's natural neighbour interpolation on a Voronoi diagram. When ($\omega = 0$) represents the geometry of small internal features (for example, cracks) and discontinuities (for example at interfaces between bonded materials), replacing ω in the Dirichlet solution structure with the Heaviside function $H(\omega)$ and Ψ with standard finite elements Dirichlet solution structure, yields a representation for an "enhanced" solution field with built-in singularities, avoiding the usual difficulties with fine (re)meshing normally required in such problems (**?**).

More generally, Babuška et al. (2003) pointed out that many of the meshless methods can be considered to be special cases of the partition of unity or generalized finite element (GFEM) method. All such methods appear to start with a selection of basis functions (partition of unity) to represent Ψ, which is then multiplied by "local functions" ω_i that enhance the approximating solution space, based on geometric, asymptotic, or empirical information. Considering these methods as special cases of the Dirichlet solution structure suggests a systematic and constructive approach for satisfying the imposed boundary conditions, singularities, and/or other asymptotic conditions, as discussed in Section 7.3. For example, Duarte, Kim and Quaresma (2006) recently proposed a new type of C^m non-convex finite element that is a product of the standard partition of unity Ψ and a C^m weight ω constructed using R_0^m-functions.

The concept of solution structure is also useful for adaptive refinement of the approximations and multi-resolution modeling of boundary value problems. The most obvious approach would be to build the adaptivity and multi-resolution into the undetermined polynomial function Ψ. For example, variable (non-uniform) B-splines are common, and Höllig (2003) proposed using hierarchical B-splines. Both p and h refinements may be supported, but the global problem must be solved for each refinement, and the shape of refinement regions is determined and limited by the type of basis functions used in Ψ. Another approach to refinement in (Tsukanov and Shapiro 2007) advocates representing the refined solution as a series of localized structures, each requiring the solving of a local boundary value problem. Each localization is specified by a refinement window of arbitrary shape, represented implicitly by a normalized window function ($\omega_1 \geq 0$). For example, if the refinement region is contained in the interior of a domain Ω, the Dirichlet solution structure (2.6) is modified as

$$u = \varphi + \omega\Psi + \omega_1^2 H(\omega_1)\Psi_1,$$

where ω_1^2 ensures C^1 continuity of the solution field, $H(\omega_1)$ guarantees that the refined solution does not modify the solution outside the refinement window, and Ψ_1 is a refinement polynomial constructed from a set of additional basis functions. See (Tsukanov and Shapiro 2007) for further details and application to more general boundary value problems and refinement windows.

8. Conclusions

Rvachev hoped that R-functions would eventually be accepted as fundamental operations on par with other elementary functions. His belief was based on the observation that R-functions seem to provide a missing link between the logic and real analysis, and that they tend to streamline and unify many

computational tasks. This survey attempted to explain this link, to convey the key ideas and concepts of the theory, and to connect them to related developments in geometric modeling and engineering analysis. The survey did not attempt an in-depth analysis of any topics, and did not try to be comprehensive. The cited references (albeit many are available only in Russian) contain a wealth of additional results, techniques, open problems, and applications. The focus on semi-analytic sets is justified because their properties are well understood, and because they are assumed in many computational applications. But it should be clear that this restriction is artificial. For example, Martin (1994) proposed using R-functions with fuzzy sets.

Generalizations of R-functions were also evident to Rvachev (1982). He observed that besides the partition of real numbers into positive and negative, there are many other choices for potentially useful partitions. For example, one can partition real numbers into rational and irrational numbers, or, say, into all real numbers in the interval $[0, 1]$, and the rest of the real numbers. It is possible to introduce several or even infinitely many gradations when subdividing the set of real numbers. In general, any such partition Δ of the set of real numbers (based on some criterion) also determines the set $R(\Delta)$ of those real functions that in some sense "inherit" the partition criterion (sign, rationality, membership in $[0, 1]$, etc.). Such functions are a generalization of the concept of R-functions as described in this paper. In fact, Rvachev went a step further and defined a notion of R-mapping $f : \mathbb{X}^n \to \mathbb{X}^m$, where $\mathbb{X}$ is an arbitrary abstract space. The partition Δ of $\mathbb{X}$ into the qualitative equivalence classes is based on some multi-valued logic function $S_k : \mathbb{X} \to \mathbb{B}_k$. Then R-mapping f is identified by the existence of the companion multi-valued logic function $\Phi : \mathbb{B}_k^n \to \mathbb{B}_k^m$, satisfying the following commutative diagram:

$$\begin{array}{ccc} \mathbb{X}^n & \xrightarrow{\ f\ } & \mathbb{X}^m \\ S_k^n \big\downarrow & & \big\downarrow S_k^m \\ \mathbb{B}_k^n & \xrightarrow[\ \Phi\ ]{} & \mathbb{B}_k^m \end{array}$$

This and further generalizations of R-functions described by Rvachev (1982) do not appear to have found many applications so far, most likely because they remain largely unknown to the research community at large.

It is evident that R-functions are becoming more popular and are now widely used in many computational applications. It should be remembered that R-functions were invented when computational technology was in its infancy, and computational geometry and geometric modeling had not been established as disciplines. Today, semi-analytic sets may be represented or approximated by many other methods, and each method has its strength and weaknesses. The wealth of alternative methods in no way diminishes the

intellectual and practical significance of Rvachev's contributions and of the theory of R-functions. Interestingly, representation of point sets by sampled distance fields appears to be growing in popularity, partly driven by advances in image processing and medical imaging (Jones and Bærentzen 2006). Such distance fields may be smoothed using interpolation or fitting techniques (Freytag et al. 2006), resulting in smooth approximations to normalized functions. However, R-functions must be used if such fields need to represent sharp corners and features.

It is also possible that applications of R-functions may prove to be equally or even more important than the R-functions themselves. In particular, the notion of the RFM solution structure described in Section 7 appears to be extremely useful for modeling and solving boundary value problems. With the exception of RFM itself, many modern meshfree methods appear to be struggling with modeling and approximating the Dirichlet solution structure. Meanwhile, RFM provides a systematic and accurate method for imposing any and all types of boundary conditions, without artificial topological constraints or meshing, and independently of any particular numerical scheme. It remains to be seen whether satisfaction of boundary conditions results in improved numerical properties, but there is no question that RFM provides dramatic improvement in flexibility and programmability of solvers for boundary value problems.

Acknowledgements

I am indebted to my long-time collaborator, Igor Tsukanov, for numerous technical discussions and for his help in creating the examples for this survey, to Carl de Boor for reading an earlier draft of this survey and suggesting numerous improvements, and to T. Sheiko for her help with the example demonstrating use of R-functions in modeling symmetry. I would also like to thank the editor, Arieh Iserles, for his encouragement and (seemingly) infinite patience. This work was supported in part by the National Science Foundation grants DMI-0621116, OCI-0537370, DMI-0323514, and DMI-0500380, and by the Wisconsin I&EDR Program. The responsibility for any errors and omissions lies solely with the author.

REFERENCES

K. Abdel-Malek, D. Blackmore and K. Joy (2006), 'Swept Volumes: Foundations, Perspectives, and Applications', *International Journal of Shape Modeling* **12**(1), 87–127.

E. Allgower and K. Georg (1990), *Numerical continuation methods: an introduction*, Springer-Verlag New York, Inc. New York, NY, USA.

D. Altiparmakov and P. Belicev (1990), 'An efficiency study of the R-function method applied as solid modeler for Monte Carlo calculations', *Progress in Nuclear Energy* **24**, 77.

C. Andradas, L. Bröcker and J. Ruiz (1996), *Constructible sets in real geometry*, Springer New York.

I. Babuška, U. Banerjee and J. Osborn (2003), 'Survey of meshless and generalized finite element methods: A unified approach', *Acta Numerica* **12**, 1–125.

L. Barthe, B. Wyvill and E. De Groot (2004), 'Controllable binary CSG operators for soft objects', *International Journal of Shape Modeling*.

S. Basu, R. Pollack and M. Roy (2003), *Algorithms in Real Algebraic Geometry. Algorithms and Computations in Mathematics, vol. 10*, Springer-Verlag.

S. Basu, R. Pollack and M.-F. Roy (2005), Computing the first Betti number and the connected components of semi-algebraic sets, in *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, ACM Press, New York, NY, USA, pp. 304–312.

A. Biswas and V. Shapiro (2004), 'Approximate distance fields with non-vanishing gradients', *Graphical Models* **66**(3), 133–159.

A. Biswas, V. Shapiro and I. Tsukanov (2004), 'Heterogeneous material modeling with distance fields', *Computer Aided Geometric Design* **21**, 215–242.

J. L. Blechschmidt and D. Nagasuru (1990), The use of algebraic functions as a solid modeling alternative: an investigation, in *16th ASME Design Automation Conference*, Chicago, IL, pp. 33–41.

J. Bloomenthal (1988), 'Polygonization of implicit surfaces', *Computer Aided Geometric Design* **5**(4), 341–355.

J. Bloomenthal (1997), *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers.

J. Bochnak, M. Coste and M. Roy (1998), *Real algebraic geometry*, Springer.

D. Breen, S. Mauch and R. Whitaker (1998), '3D scan conversion of CSG models into distance volumes', *Volume Visualization, 1998. IEEE Symposium on* pp. 7–14.

J. Chen, M. Freytag and V. Shapiro (2007), Shape Sensitivity of Constructive Representations, in *Proceeding of the 2007 ACM Symposium on Solid and Physical Modeling*, Bejing, China.

J. Chen, V. Shapiro, K. Suresh and I. Tsukanov (2006), 'Shape optimization with topological changes and parametric control', *International Journal of Numerical Methods in Engineering*. in press.

R. Descartes (1637), *La Géométrie*.

D. Dobkin, L. Guibas, J. Hershberger and J. Snoeynik (1988), 'An efficient algorithm for finding the csg representation of a simple polygon', *Computer Graphics* **22**(4), 31–40.

C. A. Duarte, D.-J. Kim and D. M. Quaresma (2006), 'Arbitrarily smooth generalized finite element approximations', *Computer Methods in Applied Mechanics and Engineering* **196**, 33–56.

H. Edelsbrunner (1987), *Algorithms in Combinatorial Geometry*, Vol. 10 of *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag.

M. Ensz, D. Storti and M. Ganter (1998), 'Implicit Methods for Geometry Creation', *International Journal of Computational Geometry and Applications* **8**(5/6), 509–536.

G. Farin, J. Hoschek and M. Kim (2002), *Handbook of computer aided geometric design*, Elsevier.

P. Fayolle, A. Pasko, B. Schmitt and N. Mirenkov (2006), 'Constructive Heterogeneous Object Modeling Using Signed Approximate Real Distance Functions', *Journal of Computing and Information Science in Engineering* **6**, 221.

Y. D. Fougerolle, A. Gribok, S. Foufou, F. Truchetet and M. A. Abidi (2005), 'Boolean operations with implicit and parametric representation of primitives using R-functions', *IEEE Transactions on Visualization and Computer Graphics* **11**(5), 529–539.

M. Freytag, V. Shapiro and I. Tsukanov (2006), 'Field modeling with sampled distances', *Computer-Aided Design* **38**(2), 87–100.

I. Goncharyuk, V. Rvachev and L. Shklyarov (1968), 'Torsion in bars of polygonal profile, with allowance for stress-function characteristics', *International Applied Mechanics* **4**(8), 88–94.

Y. Gulyayev, V. Kravchenko, V. Rvachev and N. Sizova (1995), 'R-function method in investigation of elastic waves diffraction on rigid insert of arbitrary shape', *Cybernetics and systems analysis* **344**(4), 457–459.

K. Höllig (2003), *Finite Element Methods with B-Splines*, number 26 *in* 'Frontiers in Applied Mathematics', SIAM.

J. Hoschek and D. Lasser (1993), *Fundamentals of Computer Aided Geometric Design*, A K Peters.

H. Ilies and V. Shapiro (1999), 'Dual of sweep', *Computer Aided Design* **31**(3), 185–201.

M. Jones and J. Bærentzen (2006), '3D Distance Fields: A Survey of Techniques and Applications', *IEEE Transactions on Visualization and Computer Graphics* **12**(4), 581–599.

L. V. Kantorovich and V. I. Krylov (1958), *Approximate Methods of Higher Analysis*, Interscience Publishers.

I. Y. Kharrik (1963), 'On approximation of functions that have zero values and derivatives on domain boundary by special functions', *Siberian mathematical journal* **4**(2), 408–425.

Y. Kim and D. Wilde (1992), 'Local cause of non-convergence in a convex decomposition using convex hulls', *Transactions of ASME, Journal of Mechanical Design* **114**, 468 – 476.

V. Komkov (1989), Stochastic modelling of physical processes and optimization of the domain, in *Design Theory '88: NSF Grantees Workshop on Design Theory and Methodology* (S. L. Newsome, W. R. Spillers and S. Finger, eds), Springer-Verlag, New York.

L. N. Kutsenko (1990), *Computer graphics in the problems of projective nature*, number 8 *in* 'Mathematics and Cybernetics', Znanie, Moscow.

J. Laguardia, E. Cueto and M. Doblare (2005), 'A natural neighbour Galerkin method with quadtree structure', *International Journal for Numerical Methods in Engineering* **63**, 789–812.

P. Lancaster and K. Salkauskas (1986), *Curve And Surface Fitting: An Introduction*, Academic Press.

E. Lawler (1964), 'An Approach to Multilevel Boolean Minimization', *Journal of the Association for Computing Machinery* **11**(3), 283–295.

S. Lojasiewicz (1964), 'Triangulations of semi-algebraic sets', *Annali della Scuola Normale Superiore di Pisa* **18**, 449–474.

W. Lorensen and H. Cline (1987), 'Marching cubes: A high resolution 3D surface construction algorithm', *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* pp. 163–169.

K. Maksimenko-Sheiko and N. Sheiko (2005), 'Simulation of Incompressible Viscous Liquid Motion in Twisted Tubes by R-function Method', *Elektronnoe Modelirovanie* **27**(1), 31–43.

K. B. Maksimenko-Sheiko, A. M. Matsevity and T. N. Sheiko (2005), 'Constructive methods of R-functions for constructing of 3d primitives', *Problemy mashinostroyenia (Problems of machine building)* **8**(1), 59–65.

R. Martin (1994), 'Modelling inexact shapes with fuzzy sets', *CSG 94: Set-theoretic Solid Modelling Techniques and Applications* pp. 73–100.

Y. M. Matsevitiy, ed. (2001), *Vladimir Logvinovich Rvachev*, Bibliography of Ukrainian Scientists, Academy of Sciences of Ukraine.

J. O'Rourke (1982), 'Polygon decomposition and switching function minimization', *Computer Graphics and Image Processing* **18**, 382–391.

J. O'Rourke (1998), *Computational Geometry in C*, Cambridge University Press, Cambridge, United Kingdom.

A. Pasko, V. Adzhiev, A. Sourin and V. Savchenko (1995), 'Function representation in geometric modeling: Concepts, implementation and applications', *The Visual Computer* **11**(8), 429–446.

G. Pasko, A. Pasko, M. Ikeda and T. Kunii (2002), Bounded blending operations, in *SMI '02: Proceedings of the Shape Modeling International 2002 (SMI'02)*, IEEE Computer Society, Washington, DC, USA, p. 95.

D. Peterson (1986), 'Boundary to constructive solid geometry mappings: a focus on 2D issues', *Computer-Aided Design* **18**(1), 3–14.

A. A. G. Requicha (1977), Mathematical models of rigid solid objects, Tech. Memo 28, Production Automation Project, University of Rochester, Rochester, NY.

A. A. G. Requicha (1980), 'Representations for rigid solids: Theory, methods, and systems', *ACM Computing Surveys* **12**(4), 437–464.

A. A. G. Requicha and H. B. Voelcker (1977), Constructive solid geometry, Tech. Memo 25, Production Automation Project, University of Rochester.

A. Ricci (1973), 'A constructive geometry for computer graphics', *Computer Journal* **16**(3), 157–160.

E. Rimon and D. Koditschek (1992), 'Exact robot navigation using artificial potential functions', *Robotics and Automation, IEEE Transactions on* **8**(5), 501–518.

E. Rimon and D. E. Koditschek (1990), Exact robot navigation in geometrically complicated but topologically simple environment, in *IEEE International Conference on Robotics and Automation*.

A. P. Rockwood (1989), 'The displacement method for implicit blending surfaces in solid modeling', *ACM Transactions on Graphics* **8**(4), 279–297.

A. P. Rockwood and J. C. Owen (1987), Blending surfaces in solid modeling, in *Geometric Modeling: Algorithms and New Trends* (G. E. Farin, ed.), SIAM, Philadelphia, pp. 367–383.

J. R. Rossignac (1996), CSG formulations for identifying and for trimming faces of CSG models, in *CSG'96, Winchester, UK*, Information Geometers Ltd.

S. D. Roth (1982), 'Ray casting for modeling solids', *Computer Graphics and Image Processing* **18**, 109–144.

V. Rvachev and E. Mikhal' (2001), 'Completeness of Structural Solutions in Boundary-Value Problems for Domains of Special Form', *Cybernetics and Systems Analysis* **37**(4), 551–561.

V. L. Rvachev (1963), 'On analytical description of some geometric objects', *Reports (Doklady) of Academy of Sciences, USSR* **153**(4), 765–768.

V. L. Rvachev (1967), *Geometric Applications of Logic Algebra*, Naukova Dumka. In Russian.

V. L. Rvachev (1974), *Methods of Logic Algebra in Mathematical Physics*, Naukova Dumka. In Russian.

V. L. Rvachev (1982), *Theory of R-functions and Some Applications*, Naukova Dumka. In Russian.

V. L. Rvachev (1996), 'Lectures at the University of Wisconsin - Madison', Video recording, Spatial Automation Laboratory. http://sal-cnc.me.wisc.edu.

V. L. Rvachev and L. V. Kurpa (1987), *R-functions in Problems of Theory of Plates*, Naukova Dumka. In Russian.

V. L. Rvachev and G. P. Manko (1983), *Automation of Programming for Boundary Value Problems*, Naukova Dumka. In Russian.

V. L. Rvachev and V. A. Rvachev (1979), *Nonclassical Methods of Approximation Theory in Boundary Value Problems*, Naukova Dumka. In Russian.

V. L. Rvachev and T. I. Sheiko (1995), 'R-functions in boundary value problems in mechanics', *Applied Mechanics Reviews* **48**(4), 151–188.

V. L. Rvachev and A. N. Shevchenko (1988), *Problem-oriented languages and systems for engineering computations*, Tekhnika, Kiev. In Russian.

V. L. Rvachev and N. S. Sinekop (1990), *R-functions method in problems of the elasticity and plasticity theory*, Nauk. Dumka, Kiev. In Russian.

V. L. Rvachev and A. P. Slesarenko (1976), *Logic Algebra and Integral Transforms in Boundary Value Problems*, Naukova Dumka. In Russian.

V. L. Rvachev, L. V. Kurpa, N. G. Sklepus and L. A. Uchishvili (1973), *Method of R-functions in Problems on Bending and Vibrations of Plates of Complex Shape*, Naukova Dumka. In Russian.

V. L. Rvachev, G. P. Manko and A. N. Shevchenko (1986), The R-function approach and software for the analysis of physical and mechanical fields, in *Software for Discrete Manufacturing* (J. P. Crestin and J. F. McWaters, eds), North-Holland, Paris.

V. L. Rvachev, T. I. Sheiko, V. Shapiro and I. Tsukanov (2000), 'On completeness of RFM solution structures', *Computational Mechanics* **25**, 305–316.

V. L. Rvachev, T. I. Sheiko, V. Shapiro and I. Tsukanov (2001), 'Transfinite interpolation over implicitly defined sets', *Computer Aided Geometric Design* **18**(4), 195–220.

V. Rvachev, T. Sheiko and V. Shapiro (1999), 'The R-function method in boundary-value problems with geometric and physical symmetry', *Journal of Mathematical Sciences* **97**(1), 3888–3899.

V. Rvachev, T. Sheiko, V. Shapiro and J. Uicker (1997), 'Implicit function modeling of solidification in metal casting', *Transaction of ASME, Journal of Mechanical Design* **119**, 466–473.

V. Rvachev, A. Shevchenko and V. Veretel'nik (1994), 'Numerical integration software for projection and projection-grid methods', *Cybernetics and Systems Analysis* **30**(1), 154–158.

V. Rvachev, N. Sinekop and I. Molotkov (1991), 'R-function method for thermoelasticity problems in finite objects', *Transactions of the Academy of Sciences of USSR (Doklady Academii Nauk SSSR)* **321**(4), 721–725. In Russian.

V. Rvachev, N. Sinekop and I. Molotkov (1992), 'Solution structure for contact problems of thermoelasticity', *Transactions of the Ukrainian National Academy of Sciences(Dopovidi NAN Ukrainy)* **A**(6), 25–29. In Russian.

V. Rvachev, A. Slesarenko and N. Safonov (1993), 'On solution of steady non-linear heat transfer problems', *Transactions of the Ukrainian National Academy of Sciences(Dopovidi NAN Ukrainy)* **A**(11), 1015–1021. In Russian.

J. Sethian (1996), *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences*, Cambridge University Press.

V. Shapiro (1988), Theory of R-functions and applications: A primer, Tech. Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY. Revised 1991.

V. Shapiro (1991), Representations of Semi-Algebraic Sets in Finite Algebras Generated by Space Decompositions, PhD thesis, Cornell University, Cornell Programmable Automation, Ithaca, NY, 14853.

V. Shapiro (1994), 'Real functions for representation of rigid solids', *Computer-Aided Geometric Design* **11**(2), 153–175.

V. Shapiro (1997), 'Maintenance of geometric representations through space decompositions', *International Journal on Computational Geometry and Applications* **7**(4), 383–418.

V. Shapiro (1999), 'Well-formed set representations of solids', *International Journal on Computational Geometry and Applications* **9**(2), 125–150.

V. Shapiro (2001), 'A convex deficiency tree algorithm for curved polygons', *International Journal of Computational Geometry and Applications* **11**(2), 215–238.

V. Shapiro (2002), Solid modeling, in *Handbook of Computer Aided Geometric Design* (G. Farin, J. Hoschek and M.-S. Kim, eds), Elsevier Science.

V. Shapiro and I. Tsukanov (1999*a*), Implicit functions with guaranteed differential properties, in *Fifth ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, pp. 258–269.

V. Shapiro and I. Tsukanov (1999*b*), 'Meshfree simulation of deforming domains', *Computer-Aided Design* **31**(7), 459–471.

V. Shapiro and D. L. Vossler (1991*a*), 'Construction and optimization of CSG representations', *Computer-Aided Design* **23**(1), 4–20.

V. Shapiro and D. L. Vossler (1991*b*), 'Efficient CSG representations of two-dimensional solids', *Transactions of ASME, Journal of Mechanical Design* **113**, 292–305.

V. Shapiro and D. L. Vossler (1993), 'Separation for boundary to CSG conversion', *ACM Transactions on Graphics* **12**(1), 35–55.

H. Sheffer (1913), 'A Set of Five Independent Postulates for Boolean Algebras, with

Application to Logical Constants', *Transactions of the American Mathematical Society* **14**(4), 481–488.

T. Sheiko (1982), 'Taking account of singularities at angular points and juncture points of the boundary conditions in the method of R-functions', *International Applied Mechanics* **18**(4), 365–370.

D. Shepard (1968), A two-dimensional interpolation function for irregularly spaced data, in *Proceedings 23rd ACM National Conference*, pp. 517–524.

A. Shevchenko and I. Tsukanov (1994), 'Software for representing results of field simulation in complex-form domains', *Control Systems and Mechanisms* (4-5), 86–89. ISSN 0130-5395 (in Russian).

A. M. Shkel (1997), Sensor-based motion planning with kinematic and dynamic constraints, Phd thesis, mechanical engineering, University of Wisconsin-Madison.

A. Sourin and A. Pasko (1995), Function representation for sweeping by a moving solid, in *Proc. Third Symposium on Solid Modeling Foundations and CAD/CAM applications*.

A. Sourin, A. Pasko and V. Savchenko (1996), 'Using real functions with application to hair modelling', *Computers & graphics* **20**(1), 11–19.

Y. G. Stoian (1975), *Placement of Geometric Objects*, Naukova Dumka. In Russian.

Y. G. Stoian and A. A. Panasenko (1978), *Periodic Placement of Geometric Objects*, Naukova Dumka. In Russian.

D. Storti, M. Ganter and C. Nevrinceanu (1992), 'A Tutorial on Implicit Solid Modeling', *The Mathematica Journal* **2**(3), 70–78.

S. Tor and A. Middleditch (1984), 'Convex decomposition of simple polygons', *ACM Transactions on Graphics* **3**(4), 244–265.

I. Tsukanov (2002), 'On the question of calculation of temperature fields in piecewise-homogeneous orthotropic media', *Journal of Mathematical Sciences* **109**(1), 1338–1343. ISSN: 1072-3374 (paper), 1573-8795 (online).

I. Tsukanov and M. Hall (2003), 'Data structure and algorithms for fast automatic differentiation', *International Journal for Numerical Methods in Engineering* **56**(13), 1949–1972.

I. Tsukanov and V. Shapiro (2002), 'The architecture of SAGE – a meshfree system based on RFM', *Engineering with Computers* **18**(4), 295–311.

I. Tsukanov and V. Shapiro (2005), 'Meshfree modeling and analysis of physical fields in heterogeneous media', *Advances in Computational Mathematics* **23**(1), 95–124.

I. Tsukanov and V. Shapiro (2007), 'Adaptive multiresolution refinement with distance fields', *International Journal for Numerical Methods in Engineering*. Accepted for publication.

I. Tsukanov, V. Shapiro and S. Zhang (2003), 'A meshfree method for incompressible fluid dynamics problems', *International Journal for Numerical Methods in Engineering* **58**(1), 127–158.

L. Velho, J. Gomes and L. de Figueiredo (2002), *Implicit Objects in Computer Graphics*, Springer.

D. F. Watson (1992), *Contouring: A Guide To The Analysis And Display Of Spatial Data*, Pergamon Press.

H. Whitney (1934), 'Analytic Extensions of Differentiable Functions Defined in Closed Sets', *Transactions of the American Mathematical Society* **36**(1), 63–89.

H. Whitney (1957), 'Elementary Structure of Real Algebraic Varieties', *The Annals of Mathematics* **66**(3), 545–556.

H. Whitney (1965), 'Local properties of analytic varieties', *Differential and Combinatorial Topology* pp. 205–44.

T. C. Woo (n.d.), Feature extraction by volume decomposition, in *Proc. Conference on CAD/CAM Technology in Mechanical Engineering*, Cambridge, MA, p. 76.

J. R. Woodwark (1988), Blends in geometric modelling, in *Proceedings on Mathematics of surfaces II*, Clarendon Press, New York, NY, USA, pp. 255–297.

J. R. Woodwark and A. L. Wallis (1982), Graphical input to a boolean solid modeller, in *Proc. CAD '82*, Brighton, U.K., pp. 681–688.

B. Wyvill, A. Guy and E. Galin (1999), 'Extending the CSG Tree. Warping, Blending and Boolean Operations in an Implicit Surface Modeling System', *Computer Graphics Forum* **18**(2), 149–158.