SAL 2000 - 1

Transfinite Interpolation Over Implicitly Defined Sets

V.L.Rvachev, T.I.Sheiko, V. Shapiro, I. Tsukanov

Transfinite Interpolation Over Implicitly Defined Sets

V.L.Rvachev T.I.Sheiko
Institute for Problems in Machinery
of the National Academy of Sciences
2/10 Pozharsky Str.
Kharkov, 61046
Ukraine
root@rvachev.kharkov.ua

V.Shapiro* I.Tsukanov
Spatial Automation Laboratory
University of Wisconsin-Madison
1513 University Avenue
Madison, WI, 53706
USA
vshapiro@engr.wisc.edu

PRELIMINARY DRAFT of February 21, 2000

Abstract

In a general setting, the transfinite interpolation problem requires constructing a single function $f(\mathbf{x})$ that takes on the prescribed values and/or derivatives on some collection of point sets. The sets of points may contain isolated points, bounded or unbounded curves, as well as surfaces and regions of arbitrary topology. All such closed semi-analytic sets may be represented implicitly by real valued functions with guaranteed differential properties. Furthermore, such functions may be constructed automatically using the theory of R-functions. We show that such implicit representations may be used to solve the general transfinite interpolation problem using a generalization of the classical inverse distance weighting interpolation for scattered data. The constructed interpolants may be used to approximate boundary value and smoothing problems in a meshfree manner.

1 Introduction

1.1 General Problem Of Transfinite Interpolation

The term "transfinite interpolation" has been often used to describe the problem of constructing a surface that passes through a given collection of curves, i.e. the surface must interpolate infinitely many points[15]. In a more general setting, the interpolation problem requires constructing a single function $f(\mathbf{x})$ that takes on prescribed values and/or derivatives on some collection of point sets. In this sense, transfinite interpolation is a special type of a boundary value problem. The sets of points may contain isolated points, bounded or unbounded curves, as well as surfaces and regions of arbitrary topology. [1]

Interpolation of functions in one-dimensional space over a finite set of points is a classical problem that has been extensively treated by many mathematicians. Numerous methods have been developed to solve this problem: interpolation by polynomials (Lagrange, Hermite, Newton, Tschebyshev), interpolation by trigonometric series, interpolation by piecewise polynomials (spline interpolation), and many others. Broadly, all such methods may be divided into those that interpolate only function values (Lagrange type interpolation) and those that interpolate both values and derivatives (Hermite type problems).

The problem of transfinite interpolation of curves by a surface has received much attention in computer-aided geometric design because it arises in numerous applications (see [3] for examples). Most published techniques assume that both the given curves and the constructed surfaces are given in parametric form. Such techniques are significantly restricted by the topology of the network formed by the initial curves and may require great care in Hermite type problems to assure desired degree of smoothness between the adjacent surfaces.

* Corresponding author

[1] In this sense the term "interlocation" may be more appropriate than "interpolation" since functions and their derivatives may be prescribed over arbitrary loci.

In contrast, implicit representation of a point set by the zeros of some real valued function $\omega(\mathbf{x}) = 0$ is not constrained by the topology of the represented set, which can be heterogeneous in dimension, disconnected, and contain any number of voids. Such functions may be constructed for virtually any point sets of interest in engineering using theory of R-functions, may be normalized to behave as smooth approximations to distance functions, and may be constructed automatically from other geometric representations [20]. This paper shows that when point sets are represented implicitly, the general transfinite interpolation problem for both Lagrange and Hermite type problems may be solved simply by using a technique that is a straightforward generalization of the inverse distance weighting method for scattered data interpolation.

1.2 Inverse Distance Weighting

The main difficulty in applying the classical methods of interpolation to points scattered in higher dimensional space has to do with absence of a natural ordering among the points. Numerous interpolation schemes have been developed that require preprocessing the data and arranging it according to some mesh, network, or grid, which largely determines the properties of the resulting interpolating function. Such techniques include bilinear and bicubic interpolation [4], repeated application of Lagrange method [6], and many others [3]. Most of these techniques require constructing and maintaining complex data structures to handle point location and incidence information, are limited in their ability to handle continuity of interpolation across the neighboring cells in the mesh, and do not extend easily to more general problems of transfinite interpolation.

A fairly old technique to perform interpolation of scattered data, known as inverse distance weighting (sometimes also called Shepard's method) stands out because it interpolates the data using global functions without using any mesh or adjacency information between the given points. Shepard used the method in [22] for interpolation of meteorological and geographical/geological data in 1968, but Watson [26] cites much earlier applications of the same technique dating as far back as 1920's; Rvachev proposed a similar method for interpolating functions in 1967 [11]. In all cases, the interpolating function is constructed as a linear combination of function's values f_i at points $\mathbf{x}$ with weight functions W_i:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i W_i(\mathbf{x}). \tag{1}$$

where each weight function W_i is inversely proportional to the distance from the point $\mathbf{x}_i$ where the value f_i is prescribed. This expression can be considered as a representation of the function f in a basis formed by the functions $W_i(\mathbf{x})$. Therefore we also expect that the weight functions W_i, $i = 1, ..., n$, should be positive continuous functions satisfying the interpolation condition $W_i(\mathbf{x}_j) = \delta_{ij}$ and forming a partition of unity, i.e. $\sum W_i(\mathbf{x}) = 1$.

Using the inverse distance weighting, the functions W_i, $(i = 1, ..., n)$ are constructed by normalizing each inverse distance:

$$W_i(\mathbf{x}) = \frac{d_i^{-\mu_i}(\mathbf{x})}{\sum\limits_{j=1}^{n} d_j^{-\mu_j}(\mathbf{x})}, \tag{2}$$

where $d_i(\mathbf{x})$ is the Euclidean distance from point $\mathbf{x}$ to the node $\mathbf{x}_i$. It is well known [3] that the exponents μ_i control behavior of the interpolating function at the nodes: when $0 < \mu_i \leq 1$ the interpolant is not differentiable at the i-th node; values of $\mu_i > 1$ assure that the interpolant is differentiable $\mu_i - 1$ times at the i-th node, but it has a flat spot there. This formulation may not be very useful when the interpolant must take into account the prescribed values of derivatives at the i-the node. This can be achieved by a slight modification of the inverse weighting method (for example, see [3]) that involves setting $\mu = 2$ and constructing the interpolant as:

$$f(\mathbf{x}) = \sum_{i=1}^{n} W_i(\mathbf{x}) \left(f_i + f_{x_i}(x - x_i) + f_{y_i}(y - y_i) \right), \tag{3}$$

where f_{x_i} and f_{y_i} are respectively partial derivatives of the interpolated function with respect to x and y. Similar techniques may be used to interpolate higher order terms in the Taylor series expansion of function $f(x)$ at point $\mathbf{x}_i$. When the values of derivatives are not known, they may also be determined according to some assumed variational principle. For example, partial derivatives are approximated by polynomials in [7] and by trigonometric series in [8].

Many additional properties and variations of inverse distance weighting are discussed extensively in the literature [3, 26, 16]. We conclude this necessarily brief exposition of the method by noticing that numerical evaluation of the weight function W_i in the form (2) would lead to $\frac{\infty}{\infty}$ in the neighborhood of the i-th node. The associated numerical problems are easily avoided by rewriting the weight functions in the equivalent but numerically stable form:

$$W_i(\mathbf{x}) = \frac{\prod_{j=1;j\neq i}^{n} d_j^{\mu_j}(\mathbf{x})}{\sum_{k=1}^{n} \prod_{j=1;j\neq k}^{n} d_j^{\mu_j}(\mathbf{x})} \tag{4}$$

1.3 Transfinite Interpolation With Implicit Functions

The inverse distance weighting method has been used to interpolate functions over a discrete set of scattered points. In his 1967 book [11], Rvachev observed that similar weighting functions may be constructed for boundaries of more general sets. The main contribution of this paper is in extending the inverse distance weighting interpolation to arbitrary collections of point sets. Our technique is particularly appealing for transfinite interpolation problems because it inherits the main advantage of the inverse distance weighting method: it does not place any restriction on incidence, regularity, or the topology of the sets being interpolated.

Note that each weighting function W_i is constructed using the finite collection of distance functions $d_j(\mathbf{x})$ that are positive everywhere, except at the given nodal points $\mathbf{x}_i$ where they vanish. Numerous variations of the inverse distance method have been devised by employing other types of functions, including exponential functions of the form $e^{-d(\mathbf{x})}$, cubic splines, and various combinations of functions with local support [2, 4]. We can view all such functions as *implicitly defining* the data points by equations $d_i(\mathbf{x}) = 0$.

Suppose now that we can construct similar implicit representations (functions) ω_j for other types of geometric objects. Then replacing functions d_j with ω_j in expressions (2) and (4) would immediately yield the corresponding weighting functions, and their linear combination would interpolate the functions f_j defined over the domains represented by $(\omega_j(\mathbf{x}) = 0)$.

Constructing distance functions for complex geometric shapes is a non-trivial task, and, furthermore, such functions in general are not differentiable at points that are equidistant from the given set of points. However, the theory of R-functions [12] provides means for systematically constructing smooth approximations to distance functions for any closed semi-analytic set. Furthermore, such functions may be constructed automatically for a wide class of geometrical objects including curves, surfaces, and solids [20].

1.4 Outline

The rest of the paper is organized as follows. Section 2 briefly summarizes the main ideas of the theory of R-functions and explains how it can be used to construct implicit functions with guaranteed differential properties. Section 3 shows how using such functions with inverse distance weighting method allows transfinite interpolation over implicitly defined sets for both Lagrange and Hermite types of interpolation. The latter relies on the existence of a generalized Taylor series expansion of the interpolant in terms of powers of the distance function ω_j in the neighborhood of points defined by $\omega_j(\mathbf{x}) = 0$.

An interpolation problem without additional constraints clearly has infinitely many solutions. In the concluding section, we explain how a particular transfinite interpolant with desired properties may be constructed according to some minimization or variational principle, by combining the initial interpolant with an additional set of basis functions on a uniform non-conforming mesh. Two-dimensional examples are used throughout the paper for illustration purposes, but the described techniques and results generalize to higher dimensions in a straightforward fashion.

2 Implicit Functions

Our method of transfinite interpolation is critically dependent on the existence of implicit representations for geometric objects in the form $\omega(\mathbf{x}) = 0$, where ω is a real-valued function. Such functions are often called *implicit functions* for brevity [1], and have been used extensively in computer graphics, image processing, and robotics. In principle, any of the known constructions can be used in conjunction with the proposed interpolation method. However, in the most general setting, we are interested not only in making sure that interpolant assumes the correct prescribed values, but also that it is differentiable and has an "intuitive" behavior. For example, it is often desirable that the influence of the values prescribed at a point $\mathbf{x}$ is correlated to the distance to $\mathbf{x}$. Thus, differential and qualitative properties of the implicit functions can become important in many interpolation problems.

A general algorithmic method for constructing implicit functions with guaranteed differential properties is based on the theory of R-functions. The theory of R-functions was developed in Ukraine by Rvachev and his students [12]. A complete list of references through 1987 can be found in [23]. A brief English summary of the theory of R-functions is available as a technical report [17]. Shapiro observed that implicit functions for any solid may be constructed using R-functions [18]. Numerous applications of R-functions are beginning to appear in English literature [25, 9, 24, 5, 13, 10, 1, 21], and the recent work reported in [20] indicates that implicit functions with guaranteed differential properties may be constructed automatically for a wide variety of point sets in geometric modeling.

2.1 R-functions

An R-function is a real-valued function whose sign is completely determined by the signs of its arguments. Treating the sign of a function as its logical property, we can assume that negative values of a function correspond to logical *false*, and positive ones to logical *true*. In this sense, an R-function works as a Boolean switching function, changing its sign only when its arguments change their signs; they can be regarded as "on" or "off" depending on the values of the input variables. Just as Boolean functions, R-functions are closed under composition. This means that all R-functions may be easily constructed using only a small number of primitive R-functions that are not unique.

For example, it is well known that $\min(x_1, x_2)$ is an R-function whose companion Boolean function is logical "and" ($\wedge$), and $\max(x_1, x_2)$ is an R-function whose companion Boolean function is logical "or" ($\vee$). But the same Boolean function corresponds to many other R-functions, e.g.

$$\begin{aligned} x_1 \wedge_\alpha x_2 &\equiv \tfrac{1}{1+\alpha}\left(x_1 + x_2 - \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}\right); \\ x_1 \vee_\alpha x_2 &\equiv \tfrac{1}{1+\alpha}\left(x_1 + x_2 + \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}\right), \end{aligned} \tag{5}$$

where $\alpha(x_1, x_2)$ is an arbitrary function such that $-1 < \alpha(x_1, x_2) \leq 1$. In fact, setting $\alpha \equiv 1$ yields the functions $\min$ and $\max$ respectively, but setting $\alpha \equiv 0$ results in much nicer functions $\vee_0$ and $\wedge_0$ that are *analytic* everywhere except when $x_1 = x_2 = 0$. Similarly, R-functions

$$x_1 \wedge_\alpha^m x_2 \equiv (x_1 \wedge_\alpha x_2)(x_1^2 + x_2^2)^{\frac{m}{2}}; \qquad x_1 \vee_\alpha^m x_2 \equiv (x_1 \vee_\alpha x_2)(x_1^2 + x_2^2)^{\frac{m}{2}} \tag{6}$$

are analytic everywhere except the origin ($x_1 = x_2 = 0$), where they are m times differentiable. Many other systems of R-functions are studied in [12], and the more popular systems are examined closely in [20]. The choice of an appropriate system of R-functions is dictated by many considerations including simplicity, continuity, differential properties, and computational convenience.

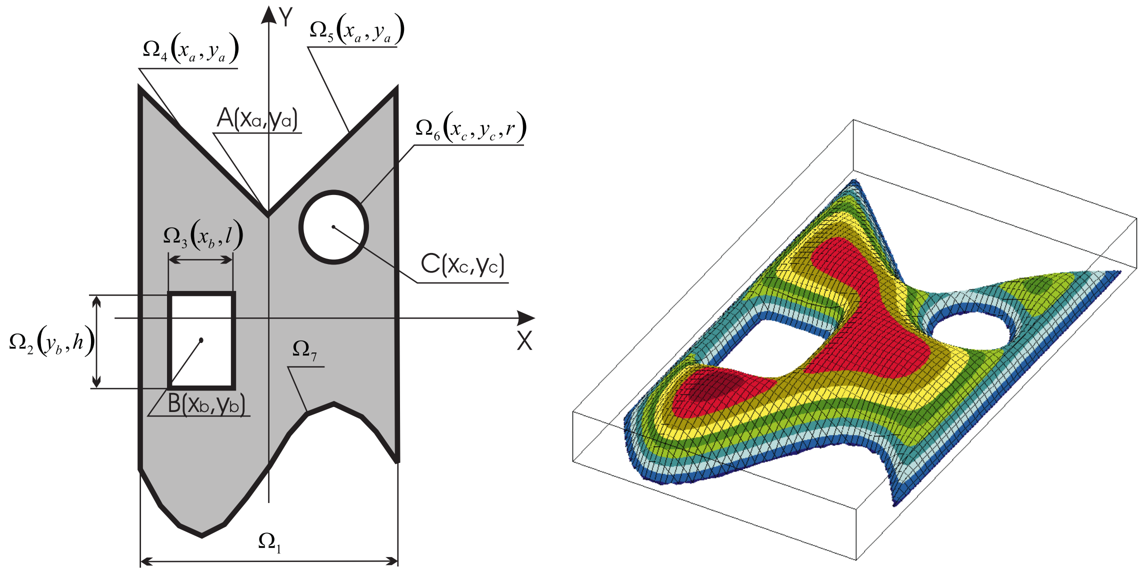
2.2 From R-functions To Implicit Functions

The main result of the theory of R-functions states that any Boolean predicate on systems of inequalities of the same type[2] may be represented by a *single* inequality of the same type. In the context of geometric modeling, this means that any object defined by a predicate on "primitive" geometric regions (e.g. regions defined by a system of inequalities) can now also be represented by a single implicit function ω. The latter can be evaluated, differentiated, and possesses many other useful properties.

The theory of R-functions provides the connection between logical and set operations in geometric modeling and analytic constructions. For *every* logical or set-theoretic construction, there is a corresponding implicit real-valued function with the above properties. Furthermore, the translation from logical and set-theoretic descriptions is a matter of simple syntactic substitution that does not require expensive symbolic computations. We explain below that such set-theoretic constructions include boundary representations (b-reps) and heterogeneous cell complexes, while the relationship to the Boolean set representations is already widely known and is particularly popular.

[2] In other words, all inequalities must be either in a form of $f \geq 0$, or $f \leq 0$, or $f > 0$, or $f < 0$, but these cannot be mixed.

Specifically, suppose that set Ω is represented by a set-theoretic expression $\Phi(h_1, h_2, \ldots, h_n)$ constructed using the standard Boolean set operations on halfspace primitives $h_i \equiv (f_i(\mathbf{x}) \geq 0)$. Then syntactically replacing set operations with the corresponding R-functions and symbols h_i with the corresponding functions f_i yields a single real-valued function of the form $F(f_1, f_2, \ldots, f_n)$. According to the theory of R-functions, the inequality $F(\mathbf{x}) \geq 0$ defines the closure of the set S. Note that points where $F = 0$ do not always lie on the boundary $\partial\Omega$ of Ω, and in some cases may correspond to the "dangling" portions of the boundary. In general, additional properties of the set representations are required to guarantee that S is a regular set and its boundary is defined by all zero points of F [19].[3]



(a) (b)

Figure 1: (a) A geometric domain and (b) a function which is positive inside the domain and is zero on the boundary

For example, the Boolean set expression

$$\Omega = \Omega_1 \cap (\Omega_2 \cup \Omega_3) \cap (\Omega_4 \cup \Omega_5) \cap \Omega_6 \cap \Omega_7 \tag{7}$$

defines the two-dimensional domain shown in Figure 1(a). The primitive halfspaces Ω_i are represented by inequalities $\Omega_i = (\omega_i(x, y) \geq 0)$, where

$$\omega_1 = \frac{4 - x^2}{4}, \quad \omega_2 = -\frac{(h/2)^2 - (y - y_b)^2}{h}, \quad \omega_3 = -\frac{(l/2)^2 - (x - x_b)^2}{l},$$

are functions defining the vertical strip Ω_1, the complement of the horizontal strip Ω_2, and the complement of the vertical strip Ω_3 respectively;

$$\omega_4 = \frac{(y-3)(x_a - 2) - (x - 2)(y_a - 3)}{\sqrt{(x_a - 2)^2 + (y_a - 3)^2}}, \quad \omega_5 = -\frac{(y - y_a)(x_a + 2) - (x - x_a)(3 - y_a)}{\sqrt{(x_a + 2)^2 + (3 - y_a)^2}}$$

define the two linear halfspaces Ω_4, Ω_5;

$$\omega_6 = \frac{1}{2r}\left((x - x_c)^2 + (y - y_c)^2 - r^2\right)$$

[3]Boolean set operations are often used synonymously with Constructive Solid Geometry (CSG). However, CSG representations additionally rely on the topological operation of regularization that assures dimensional homogeneity of the represented set.

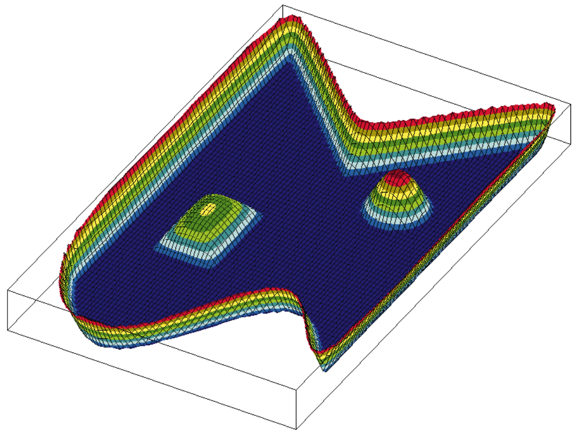


Figure 2: A function taking on zero values inside the domain shown in Figure 1(a) and approximating the distance function outside the domain

defines the exterior of the circular disk Ω_6;

$$\omega_7 = \frac{3 + y - \sin\left(\frac{\pi x}{2}\right)}{\sqrt{1 + \frac{\pi^2}{4}\cos^2\left(\frac{\pi x}{2}\right) + \left(3 + y - \sin\left(\frac{\pi x}{2}\right)\right)^2}}$$

is the function defining the sinusoidal curve bounding halfspace Ω_7. Replacing symbols Ω_i by the implicit functions ω_i and the Boolean operations $\cap$, $\cup$ by the corresponding R-functions in the expression (7) immediately gives a function

$$\omega = \omega_1 \wedge_0 (\omega_2 \vee_0 \omega_3) \wedge_0 (\omega_4 \vee_0 \omega_5) \wedge_0 \omega_6 \wedge_0 \omega_7 \tag{8}$$

which is positive inside the domain, negative outside, and takes on zero value on the boundary of the domain. Figure 1 (b) shows a plot of the ω inside the domain.

The inverse distance interpolation interpolates data over sets of points defined by equations of the form $\omega_i = 0$. It is straightforward to construct such equations for k-dimensional subsets $k = 0, 1, \ldots, d$ of a d-dimensional space E^d. Suppose that a d-dimensional region is defined by an inequality $\omega(\mathbf{x}) \geq 0$ (constructed using R-functions or by some other method). Then it is easy to see that the function

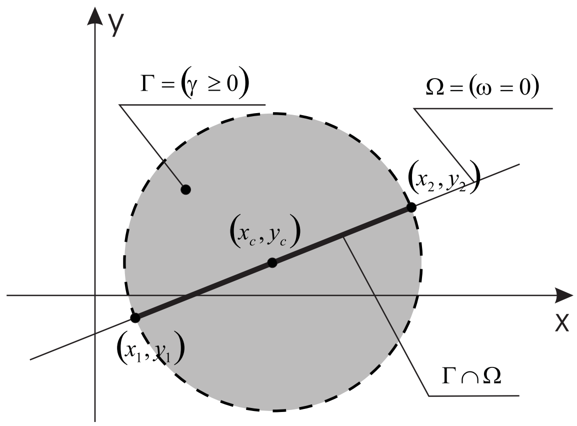
$$\omega^* = \frac{|\omega| - \omega}{2} \tag{9}$$

is identically zero on all points of the same d-dimensional region and is strictly positive on all other points. For example, application of transformation (9) to the function ω given by (8) results in a function shown in Figure 2.

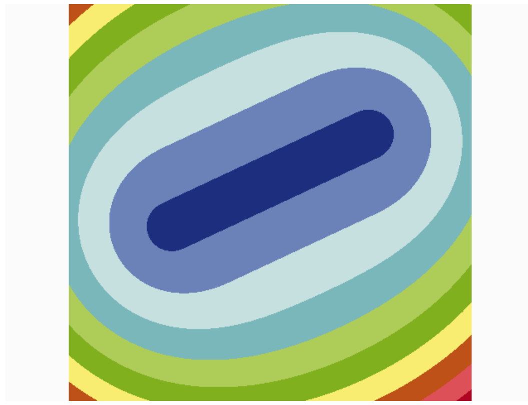
If $\omega(\mathbf{x}) \geq 0$ defines a d-dimensional halfspace, then $\omega(\mathbf{x}) = 0$ defines a $(d-1)$-dimensional hypersurface Ω in E^d (curve in E^2, surface in E^3, etc.). Notice that any such equality may be represented by an inequality as $-\sqrt{\omega^2} \geq 0$. A *trimmed* hypersurface is defined as that portion of Ω that is contained in some trim region Γ and is therefore defined by the intersection $\Omega \cap \Gamma$. Replacing intersection by a corresponding R-functions, for example as $(-\sqrt{\omega^2} \wedge_0 \gamma) \geq 0$, yields a single inequality for the trimmed hypersurface. Since, by construction, the resulting function does not take on positive values anywhere, we actually obtain the equality for the trimmed entity. Several variations of this technique, as well as closer examination of the differential properties of the resulting functions may be found in [20]. These techniques generalize further to lower-dimensional entities by noticing that $(d-2)$-dimensional sets may be written as intersections of the higher-dimensional loci.

For example, to construct an implicit function for a trimmed line segment shown in Figure 3(a), we need the equation of the line Ω going through points (x_1, y_1) and (x_2, y_2):

$$\omega = \frac{(x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}},$$



(a)



(b)

Figure 3: (a) Construction of a trimmed line segment; (b) isolines of the normalized function implicitly defining the line segment

and an inequality for the trim region, which we choose to be a circular disk Γ:

$$\gamma = \frac{1}{2r}\left(r^2 - (x - x_c)^2 - (y - y_c)^2\right),$$

where $x_c = (x_1 + x_2)/2$, $y_c = (y_1 + y_2)/2$ are coordinates of the circle's center and $r = \sqrt{(\frac{x_2 - x_1}{2})^2 + (\frac{y_2 - y_1}{2})^2}$ is the radius. Using the normalized trimming technique described in [20], we obtain a single function whose isolines are shown in Figure 3(b):

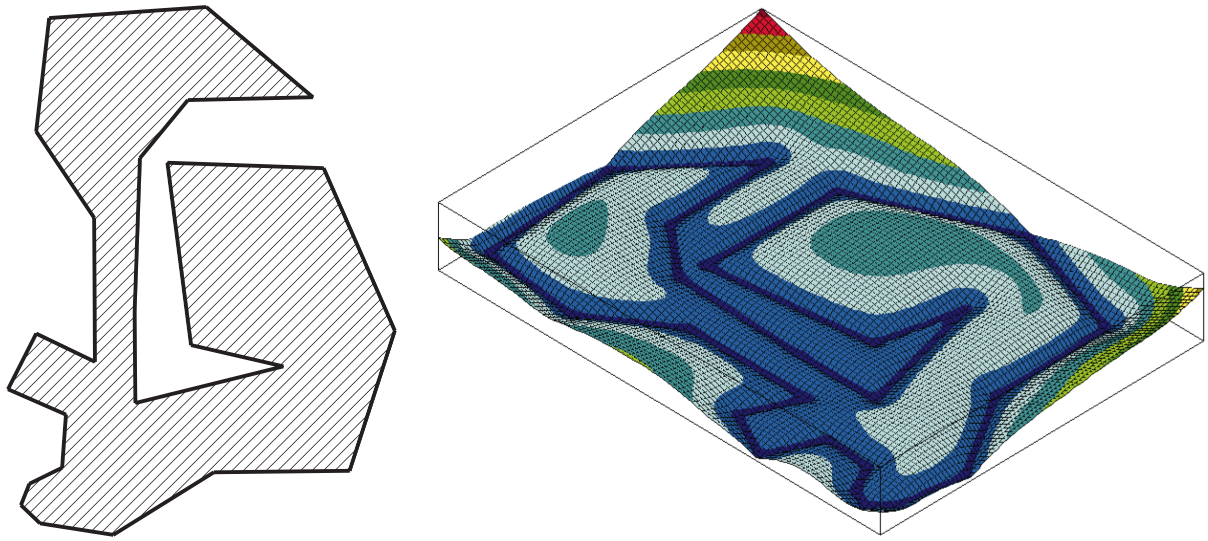
$$f = \sqrt{\omega^2 + \frac{\left(\sqrt{\gamma^4 + \omega^2} - \gamma\right)^2}{4}} \tag{10}$$

The trimmed entities may be assembled via set unions into cell complexes. For example, reference [20] shows that individual edges and faces may be "glued" (united) together using R-functions into complete boundary representations. The zero set of the resulting function will define implicitly the whole boundary of the object. Figure 4(b) shows an implicit function defining the boundary of the polygon in Figure 4(a). Furthermore, the union of trimmed entities does not need to be restricted in dimension or topology. If we construct implicit functions shown in Figure 5(a): polygon's boundary Ω_1, line segment Ω_2, circular sector Ω_3 and triangular region Ω_4, then they are also easily combined together using R-functions into a single implicit function whose zero set is the union $\Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \Omega_4$, as shown in Figure 5(b).

2.3 Smooth Distance-like Functions

The reader may have noticed that some of the constructed functions are signed. For example, implicit functions constructed from CSG representations will take on non-negative values on all interior points and negative values outside of the represented set. In contrast, the implicit functions for trimmed entities and their unions (such as boundary representations) do not distinguish between inside and outside, because they have the same sign on points away from the represented set.

While the sign properties have been exploited by others for point membership classification [5, 1], they have no significance for our purposes. In fact, it may be more convenient to use the absolute value of the implicit functions in order *not* to distinguish between interior and exterior points, because the main purpose of the constructed implicit functions is to approximate distance from a given point to the locus of points where some data may be prescribed. More specifically, recall that for use with the inverse distance weighting, it is desirable that $\omega_i(\mathbf{x})$ is some approximate



(a) (b)

Figure 4: (a) Polygon; (b) function whose zero set defines the boundary of the polygon

measure of the distance from point $\mathbf{x}$ to the i-th set of points. Formally, the function ω is called *normalized up to m-th order* when the following conditions are satisfied:

$$\omega_{|\partial\Omega} = 0; \qquad \frac{\partial\omega}{\partial\nu}_{|\partial\Omega} = 1; \qquad \frac{\partial^k\omega}{\partial\nu^k}_{|\partial\Omega} = 0; \qquad k = 2, 3, ..., m, \tag{11}$$

where ν is the normal to the boundary $\partial\Omega$ which is a subset of the set $\omega = 0$. In other words, the function ω behaves as the m-th order approximation of the distance function near the boundary of the set Ω.

A number of techniques are known for constructing normalized implicit functions [20], including a general method for recursively increasing the order of normalization of any given function [12]. A more popular approach to normalization is based on the observation that many R-functions tend to preserve the normalization properties of their arguments. In fact, the reader may observe that all primitive functions ω_i, $i = 1, 2, \ldots, 7$ used in our earlier example were normalized and combined using R_0-functions that preserved that normalization in the constructed function ω in equation (8). Strictly speaking, normalization is violated at all corner points, where the normal to the boundary of the set is not well-defined. However, the trimming operations and R-functions may be further modified to assure that the constructed functions behave also as a distance function on all regular points as well as in the neighborhoods of the corner points[12, 20]. Thus, function f (10), whose isolines are shown in Figure 3(b), behaves as the distance function in any direction on points of the line segment. Such functions are said to be normalized in a *generalized sense*. The results in [20] suggest that such functions appear to approximate the distance functions globally as well as locally.

3 Transfinite Interpolation

Below, we broadly divide interpolation problems into “Lagrange type” problems where only function values are prescribed over some point sets and “Hermite type” problems requiring interpolating both the function and the derivative data. We now show that both types of problems may be handled using a generalization of the inverse distance weighting interpolation method over implicitly defined sets of points.

3.1 Lagrange Type Interpolation

We already observed that for scattered points, the weight functions $W_i(\mathbf{x})$ were constructed using Euclidean distances $d_j(\mathbf{x})$ between point $\mathbf{x}$ and given points $\mathbf{x}_j$. The Euclidean distance functions d_j are the simplest examples of the

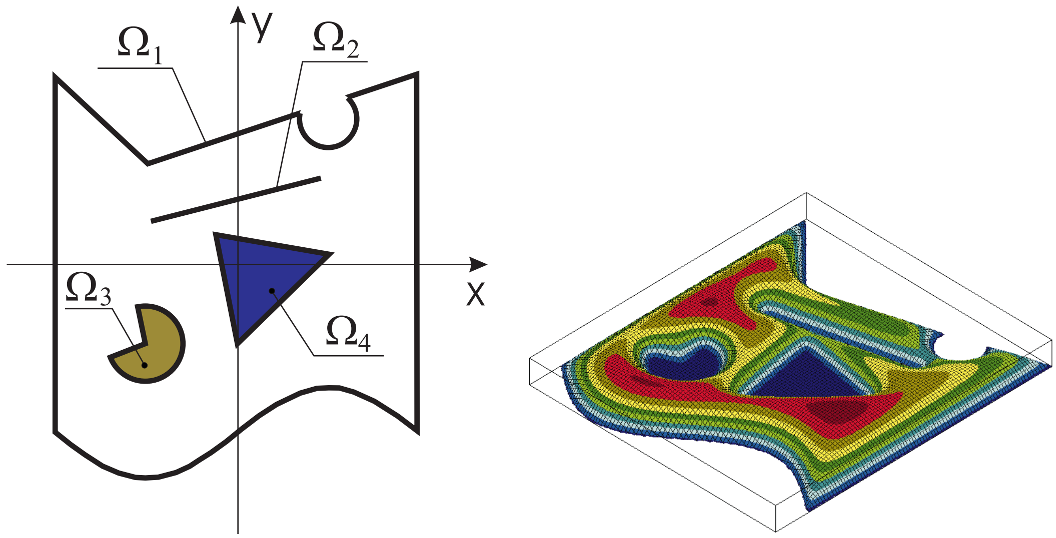


Figure 5: (a) Heterogeneous union of point sets; (b) function implicitly describing the set in Figure 5(a)

implicit functions in the sense that each equation $d_j(\mathbf{x}) = 0$ defines implicitly the position of exactly one point $\mathbf{x}_j$.

Let us now consider the case where the function values are prescribed over arbitrary sets of points $\Omega_1, \Omega_2, \ldots, \Omega_n$ with each set defined implicitly by a normalized non-negative function $\omega_j(\mathbf{x}) = 0$. We claim that replacing distance functions d_i with implicit functions ω_i in expressions (1), (2), and (4) immediately yields a transfinite interpolation technique that is a generalization of the inverse distance weighting interpolation in the sense that it inherits most of its properties. In particular, it is straightforward to check that all properties of the weight functions cited in Section 1.2 still hold:

(a) if the functions $\omega_i(\mathbf{x})$ do not change their signs and $\omega_i(\mathbf{x})$ are continuous functions, then the weight functions $W_i(\mathbf{x})$ are also continuous functions, i.e.

$$W_i(\mathbf{x}) \in C^0$$

Notice that this property does not hold for implicit functions that take on the negative values, for example implicit functions that are constructed from CSG representations and distinguish inside from outside by change of sign. The direct use of such functions in (2) or (4) would lead to weight functions with discontinuities at the points where the denominator is zero and numerator is not equal to zero. On the other hand, such discontinuity problems can be eliminated by always using even values of exponents μ_i in (2,4);

(b) weight functions $W_i(\mathbf{x})$ are strictly positive everywhere except the points that belong to other sets Ω_j, $j = 1, ..., n; j \neq i$, where $W_i(\mathbf{x})$ is identically zero:

$$W_i(\mathbf{x}) \geq 0;$$

(c) in fact, it is easy to check by direct substitution that:

$$W_{i_{|\Omega_l}} = \delta_{il},$$

which means that the weight functions satisfy the interpolation condition;

(d) finally, the weight functions form a partition of unity in the sense that

$$\sum_{i=1}^{n} W_i(\mathbf{x}) = \sum_{i=1}^{n} \frac{\prod\limits_{j=1;j\neq i}^{n} \omega_j^{\mu_j}(\mathbf{x})}{\sum\limits_{k=1}^{n} \prod\limits_{j=1;j\neq k}^{n} \omega_j^{\mu_j}(\mathbf{x})} = \frac{\sum\limits_{i=1}^{n} \prod\limits_{j=1;j\neq i}^{n} \omega_j^{\mu_j}(\mathbf{x})}{\sum\limits_{k=1}^{n} \prod\limits_{j=1;j\neq k}^{n} \omega_j^{\mu_j}(\mathbf{x})} \equiv 1.$$

This, in turn, assures completeness of this system and indicates that any constant function can be approximated exactly at any point.

In addition, we observe the following property that becomes significant only when dealing with sets more general than scattered points:

- The constructed interpolant remains continuous even when the given sets of points Ω_i intersect, provided that the same values of the function are prescribed on points that belong to more than one set Ω_i.

When all Ω_i are disjoint, the denominator of every weight function remains positive. But when two or more sets Ω_i intersect, the denominators of some weight functions go to zero at all those points that are common to more than one Ω_i. To see that this does not cause any serious difficulties, we need to check that the function approaches the same (prescribed) value from any one of the intersection sets Ω_i. For illustration purposes, let us examine the simplest case where values of function F are prescribed over two intersecting sets Ω_1 and Ω_2 represented respectively by the implicit functions ω_1 and ω_2. The resulting interpolant becomes:

$$f = F\frac{\omega_2}{\omega_1 + \omega_2} + F\frac{\omega_1}{\omega_1 + \omega_2}$$

Consider the values of the interpolant in the neighborhood of any intersection point. Let us take a point $\mathbf{x} \in \Omega_1$ and check the value of the interpolant as $\mathbf{x}$ approaches Ω_2:

$$\lim_{\mathbf{x}\in\Omega_1;\mathbf{x}\to\Omega_2} f = \lim_{\mathbf{x}\in\Omega_1;\mathbf{x}\to\Omega_2} F\frac{\omega_2}{\omega_1 + \omega_2} + \lim_{\mathbf{x}\in\Omega_1;\mathbf{x}\to\Omega_2} F\frac{\omega_1}{\omega_1 + \omega_2} = F\frac{\omega_2}{\omega_2} + F\frac{0}{\omega_2} = F.$$

It is easy to see that the situation is symmetric, i.e. that the function's value is also going to approach F if we pick $\mathbf{x}$ in Ω_2 and take the limit as $\mathbf{x}$ approaches Ω_1. For example, let functions

$$u_1 = 1, \qquad u_2 = 1 + \sin\left(\frac{(y - y_A)(x_B - x_A) - (x - x_A)(y_B - y_A)}{\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}}\right)$$

be prescribed on boundaries $\partial\Omega_1$ and $\partial\Omega_2$ shown in Figure 6. Each portion of the boundary is represented implicitly by the functions ω_1 and ω_2 respectively, whose plots are shown in Figures 7(a) and (b). At points A and B, the prescribed functions $u_1 = u_2 = 1$, and therefore the interpolating function $u = \frac{u_1\omega_2 + u_1\omega_2}{\omega_1 + \omega_2}$ preserves continuity at these points as shown in Figure 7(c).

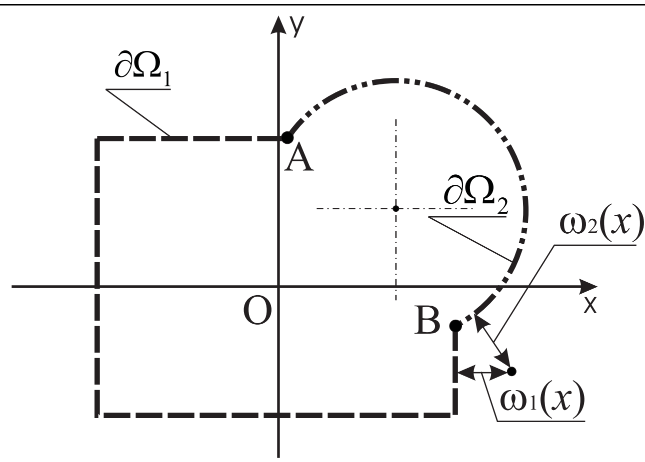


Figure 6: Functions ω_1 and ω_2 define the distance to the boundaries $\partial\Omega_1$ and $\partial\Omega_2$ respectively

In contrast, consider what happens if different values of u_1 and u_2 are prescribed on a point that is common to two intersecting sets Ω_1 and Ω_2. The general form of the interpolant becomes

$$f = u_1\frac{\omega_2}{\omega_1 + \omega_2} + u_2\frac{\omega_1}{\omega_1 + \omega_2}$$

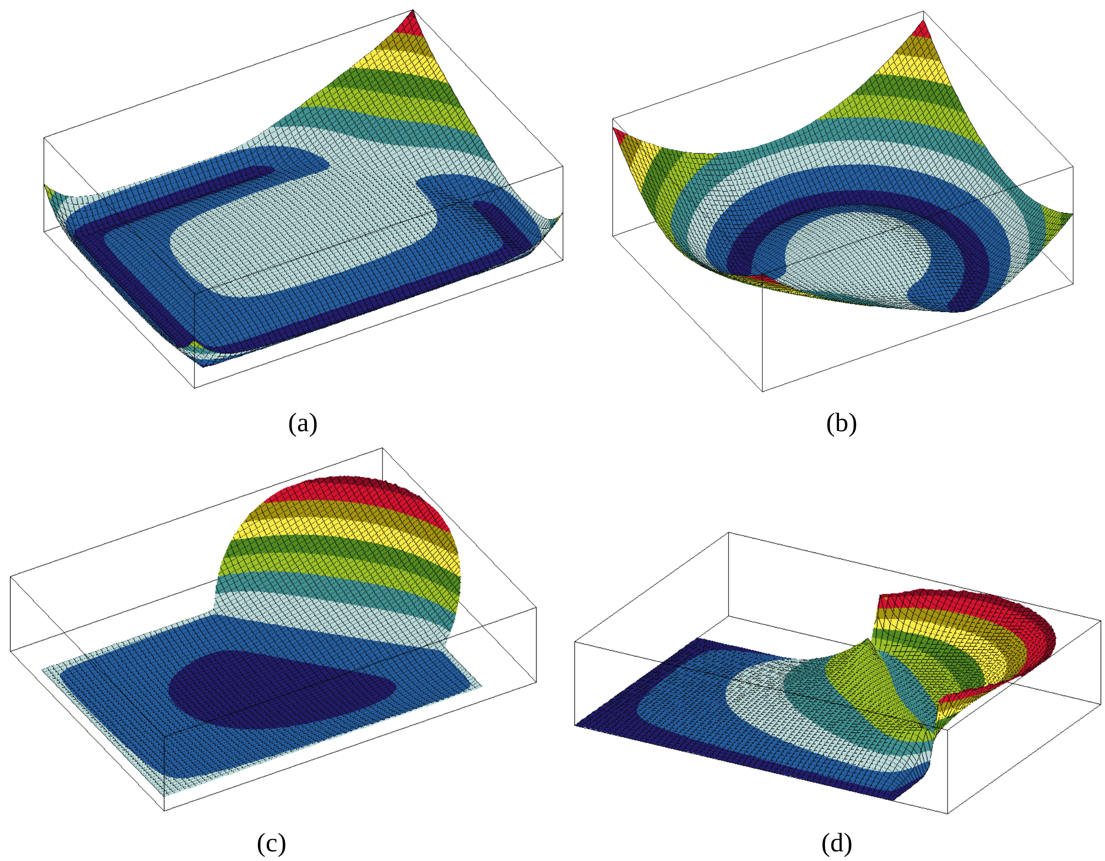


Figure 7: (a) The function implicitly defining the boundary $\partial\Omega_1$ (Figure 6); (b) the function implicitly defining the boundary $\partial\Omega_2$ (Figure 6); (c) the continuous interpolating function; (d) the interpolating function with discontinuities at the points A and B (Figure 6)

Taking the same limit as before, we get

$$\lim_{\mathbf{x}\in\Omega_1;\mathbf{x}\to\Omega_2} f = \lim_{\mathbf{x}\in\Omega_1;\mathbf{x}\to\Omega_2} u_1 \frac{\omega_2}{\omega_1+\omega_2} + \lim_{\mathbf{x}\in\Omega_1;\mathbf{x}\to\Omega_2} u_2 \frac{\omega_1}{\omega_1+\omega_2} = u_1\frac{\omega_2}{\omega_2} + u_2\frac{0}{\omega_2} = u_1;$$

On the other hand, if we pick a neighboring point $\mathbf{x}\in\Omega_2$ and approach the point of intersection that is also in Ω_1, we get:

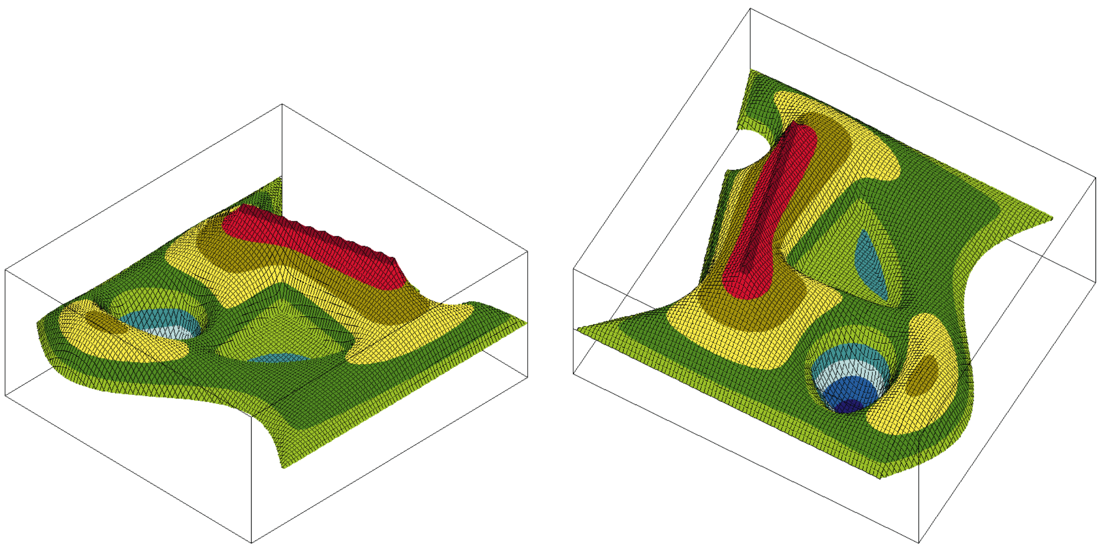
$$\lim_{\mathbf{x}\in\Omega_2;\mathbf{x}\to\Omega_1} f = \lim_{\mathbf{x}\in\Omega_2;\mathbf{x}\to\Omega_1} u_1 \frac{\omega_2}{\omega_1+\omega_2} + \lim_{\mathbf{x}\in\Omega_2;\mathbf{x}\to\Omega_1} u_2 \frac{\omega_1}{\omega_1+\omega_2} = u_1\frac{0}{\omega_1} + u_2\frac{\omega_1}{\omega_1} = u_2.$$

Clearly, f is discontinuous at the point of intersection of Ω_1 and Ω_2, which is to be expected. Notice however that such discontinuities are local phenomena and do not propagate globally. Figure 7(d) shows the function interpolating the constant values $u_1 = 1$ and $u_2 = 3$ prescribed on boundaries $\partial\Omega_1$ and $\partial\Omega_2$ (Figure 6). The discontinuities of the interpolant are clearly visible at the points A and B.

One the major advantages of our interpolation method is that it is not restricted by the dimension, topology, or adjacency of the sets Ω_i where the data is prescribed. And in particular, recall that implicit functions can be constructed for sets of arbitrary dimension and topology that can be further assembled into (heterogeneous) cell complexes. The same identical form of interpolation may be used in all cases. For example, let us construct a function that interpolates the following data:

$$\begin{aligned} u_{|\Omega_1} &= 0; \\ u_{|\Omega_2} &= 2; \\ u_{|\Omega_3} &= 15\,(x+0.5)^2 + 23\,(y+0.5)^2 - 1.5; \\ u_{|\Omega_4} &= 5x + y \end{aligned} \tag{12}$$

over the sets shown in Figure 5(a), i.e. Ω_1 is the boundary of the domain, and Ω_2 is a line segment, Ω_3 is a two-dimensional circular sector, and Ω_4 is a two-dimensional triangular region. We already showed how to construct implicit definitions ω_i for each of the four sets. Direct application of the expressions (1) and (4) results in the function shown in Figures 8 (a),(b).



(a) (b)

Figure 8: Function interpolating prescribed values over heterogeneous in dimension domains shown in Figure 5(a)

3.2 Hermite Type Problems

3.2.1 Transfinite Interpolation Of Normal Derivatives

We briefly mentioned in the introduction that the inverse distance weighting method has been used to interpolate derivative values at scattered points as well. This involves (1) replacing the function f_i at the ith point by its Taylor series expansion in the neighborhood of the point in terms of powers of $(\mathbf{x} - \mathbf{x}_i)$; and (2) raising the power μ_i of the corresponding distance function to be one greater than the highest order of the derivatives being interpolated at $\mathbf{x}_i$. The classical Taylor formula approximates a function in the neighborhood of the given point x_0. The neighborhood itself is described by the term $x - x_0$ which can be thought of as a function that vanishes at a point x_0. Similarly, the generalized Taylor series expansion represents a function u in the neighborhood of the boundary $\partial\Omega$, which is described by some implicit function ω that vanishes on $\partial\Omega$.

In [12] Rvachev observed that if function $\omega(\mathbf{x})$ is normalized up to m-th order and the function $u(\mathbf{x})$ satisfies conditions

$$u\,(\mathbf{x})_{|\partial\Omega} = f_0\,(\mathbf{x})\,, \quad \frac{\partial^k u}{\partial \nu^k}_{|\partial\Omega} = f_k\,(\mathbf{x})\,, \quad (k = 1, 2, ..., m), \tag{13}$$

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

$$u = f_0^* + \sum_{k=1}^{m} \frac{1}{k!} f_k^* \omega^k + O\left(\omega^{m+1}\right) \tag{14}$$

where $f_k^*(\mathbf{x}), k = 0, 1, ..., m$ are *normalizers* of functions $f_k(\mathbf{x}), k = 0, 1, ..., m$ with respect to $\omega(\mathbf{x})$. Functions f_k represent the prescribed values and derivatives of order k on Ω, while their normalizers f_k^* have the same values on the Ω but also behave as constants in the direction normal to the boundary $\partial\Omega$. The latter property qualifies the normalizers to serve as coefficients in the directional Taylor series expansion. Normalizers with respect to function $\omega(\mathbf{x})$ that is normalized to m-th order can be constructed by setting

$$f^*(\mathbf{x}) = f(\mathbf{x} - \omega\nabla\omega) \tag{15}$$

It is straightforward to verify that such normalizers $f^*(\mathbf{x})$ of $f(\mathbf{x})$ with respect to $\omega(\mathbf{x})$ satisfies the conditions [12]:

$$f^*(\mathbf{x})_{|\partial\Omega} = f(\mathbf{x})_{|\partial\Omega}; \quad \frac{\partial^k f^*}{\partial \nu^k}_{|\partial\Omega} = 0; \quad (k = 1, 2, ..., m). \tag{16}$$

Other methods for constructing normalizers are possible, and we will discuss another technique in a more restricted setting below. The generalized Taylor polynomial (14) approximates f near the boundary $\partial\Omega$ described by $\omega(\mathbf{x}) = 0$. The powers of $\omega(\mathbf{x})$ play the same role as the powers of the term $(x - x_0)$ in the classical Taylor formula, and the constant Taylor coefficients are replaced by the normalizers f_k^* of functions f_k. Having constructed the generalized Taylor polynomials u_i for all sets Ω_i, each represented implicitly by a function $\omega_i = 0$, we combine them into a single interpolation function using the usual inverse distance technique:

$$u = \sum_{i=1}^{n} u_i W_i = \sum_{i=1}^{n} \left((f_0^*)_i + \sum_{k=1}^{m_i} \frac{1}{k!} (f_k^*)_i \, \omega_i^k \right) \frac{\prod\limits_{j=1; j\neq i}^{n} \omega_j^{m_j+1}}{\sum\limits_{k=1}^{n} \prod\limits_{j=1; j\neq k}^{n} \omega_j^{m_j+1}}. \tag{17}$$

3.2.2 Example: First Order Normal Interpolation

For illustration, consider a special but common case of Hermitian interpolation that requires interpolating the values of functions and their first order normal derivatives prescribed on portions of the boundary $\partial\Omega_i$:

$$u_{i|\Omega_i} = \varphi_i; \qquad \frac{\partial u_i}{\partial \nu}_{|\partial\Omega_i} = \psi_i, \qquad i = 1, \ldots, n \tag{18}$$

Application of the generalized Taylor series (14) gives:

$$u_i = \varphi_i^* + \omega_i \psi_i^* + O\left(\omega^2\right). \tag{19}$$

We now use a different method,described in [13], for constructing normalizers that works well in the case of the first order derivatives. Instead of applying the coordinate transformation (15) to a given function f, we linearize f in the neighborhood of $\omega = 0$ and subtract the variation in the normal direction, leaving the constant term:

$$f^* = f(\mathbf{x}) - \omega D_1^\omega (f(\mathbf{x})) + O\left(\omega^2\right), \tag{20}$$

where D_1^ω is the differential operator acting in the direction normal to the boundary. Formally, it can be defined as

$$D_1^\omega() = \sum_{i=1}^{dim} \cos\left(\widehat{\nu, \mathbf{x}_i}\right) \frac{\partial}{\partial x_i}(),$$

but for a normalized function ω, the value of $\cos\left(\widehat{\nu, \mathbf{x}_i}\right)$ coincides on the boundary $\partial\Omega$ with the partial derivative of ω with respect to x_i. Therefore, the differential operator D_1^ω can be written in a computationally more convenient form in terms of the partial derivatives of the ω function as

$$D_1^\omega() = \nabla\omega \cdot \nabla() \tag{21}$$

Applying transformation (20) to every prescribed function and substituting in the Taylor series (19), we obtain after simplification:

$$u_i = \varphi_i - \omega_i \left(D_1^{\omega_i} \left(\varphi_i \right) + \psi_i \right) + O\left(\omega_i^2 \right). \tag{22}$$

Now we interpolate the individual Taylor series expansions of the functions u_i using the inverse distance weighting:

$$u = \sum_{i=1}^{n} u_i W_i = \sum_{i=1}^{n} \left(\varphi_i - \omega_i \left(D_1^{\omega_i} \left(\varphi_i \right) + \psi_i \right) \right) \frac{\prod\limits_{j=1;j\neq i}^{n} \omega_j^2}{\sum\limits_{k=1}^{n} \prod\limits_{j=1;j\neq k}^{n} \omega_j^2} \tag{23}$$

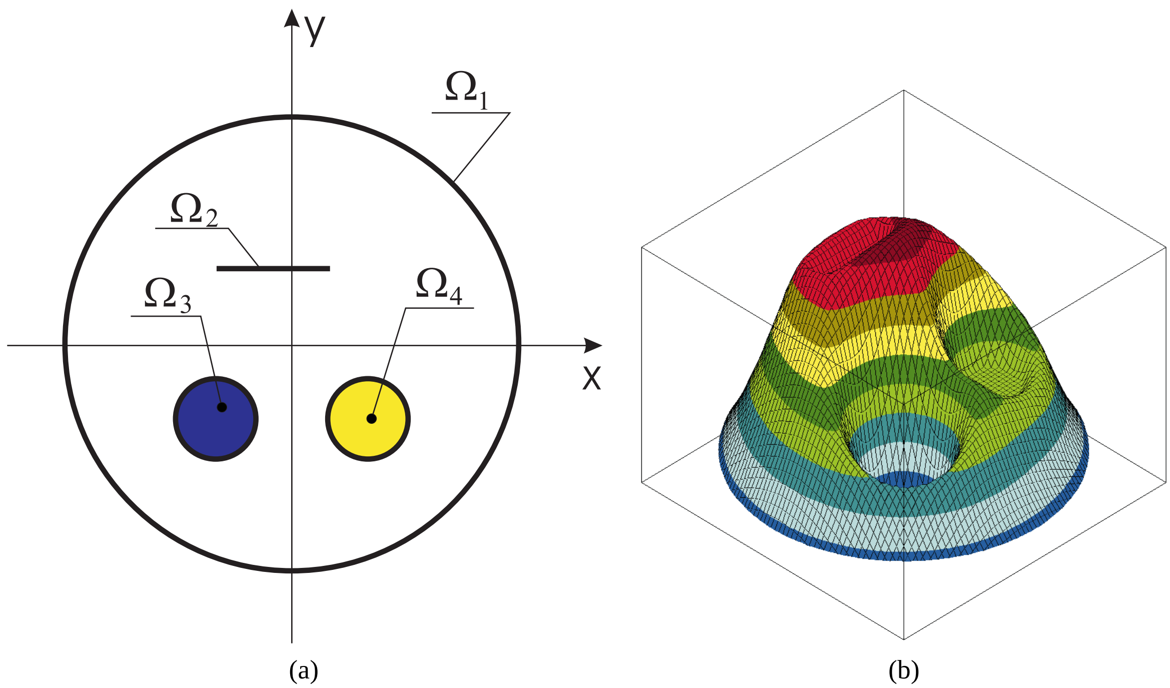


Figure 9: (a) Heterogeneous in dimension regions and (b) function interpolating given values of function and normal derivatives

Example. Figure 9 (b) shows a function that interpolates the values of a function and its normal derivatives prescribed on the outer circle Ω_1, line segment Ω_2, and on (the interior of) the inner circular regions Ω_3 and Ω_4 (Figure 9 (a)):

$$\begin{aligned} & u_{|\Omega_1} = 0; & & \\ & u_{|\Omega_2} = 5; & & \frac{\partial u}{\partial \nu}_{|\partial\Omega_2} = 3; \\ & u_{|\Omega_3} = -1; & & \frac{\partial u}{\partial \nu}_{|\partial\Omega_3} = 5; \\ & u_{|\Omega_4} = 2; & & \frac{\partial u}{\partial \nu}_{|\partial\Omega_4} = 2y+3. \end{aligned} \tag{24}$$

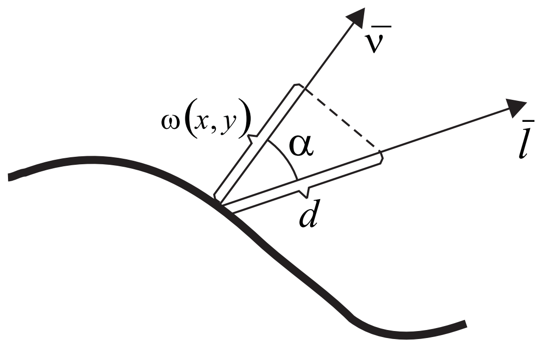


Figure 10: Projection of an arbitrary vector **l** to the normal vector ν

3.2.3 Transfinite Interpolation Of Derivatives In Arbitrary Direction

The transfinite interpolation of the normal derivatives is a special case of a more general problem where derivatives are prescribed in the given direction **l**. To handle this general problem, we once again rely on the Taylor series expansion in powers of the distance function, but now the distance from the boundary is measured along the specified direction **l**. Using a simple geometric argument illustrated in Figure 10, the distance in direction **l** from the boundary described by $\omega = 0$ may be measured as

$$d_l = \frac{\omega}{\cos(\alpha)}, \tag{25}$$

where α is the angle between the normal vector ν and the vector **l** (Figure 10). However, function $\cos\alpha$ is defined only on the boundary, but we need the relationship (25) to also hold in the neighborhood of the boundary. Therefore, we now show how function d_l can be extended on all of the space. Recall that the cosine function may be derived from the dot product of the normal vector ν and the vector **l** as

$$\cos(\alpha) = \frac{\nu \cdot \mathbf{l}}{|\nu||\mathbf{l}|}$$

Since ω is normalized, we have $\nu = \nabla\omega$ on all regular points of the boundary and can replace $\cos\alpha$ by a function that takes the values of $\cos\alpha$ on the boundary points:

$$\nabla\omega \cdot \frac{\mathbf{l}}{|\mathbf{l}|} = \sum_{i=1}^{dim} \frac{\partial\omega}{\partial x_i}\frac{l_i}{|\mathbf{l}|} = \sum_{i=1}^{dim} \frac{\partial\omega}{\partial x_i}\cos\left(\widehat{\mathbf{l}, x_i}\right). \tag{26}$$

But the expression (26) is simply the first partial derivative of ω in the given direction **l**, which we will denote $D_1^l(\omega)$. We could now use $D_1^l(\omega)$ in place of the cosine function in expression (25), resulting in a function that is continuous everywhere and is easy to compute. Unfortunately, $D_1^l(\omega)$ can vanish in the direction tangent to the boundary and possibly other points. Instead we define the distance d_l (25) as a product of ω and q:

$$d_l = \omega q,$$

where

$$q = \frac{D_1^l(\omega)}{\left(D_1^l(\omega)\right)^2 + \omega^2}$$

is a function that has no singularities and takes on the value of $\frac{1}{D_1^l(\omega)}$ on the boundary points where $\omega = 0$. With this definition of the distance we now rewrite the generalized Taylor series in any direction **l** as

$$u = f_0^{**} + \sum_{k=1}^{m} \frac{1}{k!}(\omega q)^k f_k^{**} + O\left(\omega^{m+1}\right), \tag{27}$$

where $f_k^{**}(\mathbf{x})$ must behave as constants analogously to (16) but now in the prescribed direction $\mathbf{l}$. This can be achieved, for example, by setting

$$f_k^{**}(\mathbf{x}) = f_k\left(\mathbf{x} - \frac{\omega}{\cos(\alpha)}\mathbf{l}\right)$$

for any given functions f_k, but other techniques may be used as well. Finally, if we use the Taylor series (27) for individual functions u_i whose values and derivative may be prescribed on the boundary $\partial\Omega_i$, then we can construct the global interpolating function using the previously described inverse distance technique as

$$u = \sum_{i=1}^{n} u_i W_i = \sum_{i=1}^{n}\left((f_0^{**})_i + \sum_{k=1}^{m}\frac{1}{k!}(\omega_i q_i)^k (f_k^{**})_i\right)\frac{\prod_{j=1;j\neq i}^{n}\omega_j^{m_j+1}}{\sum_{k=1}^{n}\prod_{j=1;j\neq k}^{n}\omega_j^{m_j+1}} \tag{28}$$

3.2.4 Example: First Order Directional Interpolation

Consider the special case similar to the earlier interpolation example for the normal derivatives shown in Figure 9. Once again we want to interpolate the values of functions and their first derivatives, but this time each derivative is prescribed in some given direction. Suppose that on each set Ω_i values of functions u_i and their partial derivatives in the given direction $\mathbf{l}_i$ are specified:

$$u_{i|\Omega_i} = \varphi_i; \qquad \frac{\partial u_i}{\partial \mathbf{l}_i}_{|\partial\Omega_i} = \psi_i \tag{29}$$

For each boundary portion, we construct the generalized Taylor series locally approximating the prescribed functions:

$$u_i = \varphi_i^{**} + \psi_i^{**}(\omega_i q_i) + O\left(\omega_i^2\right), \tag{30}$$

Constructing and substituting the normalizers using the same procedure as in the case of normal derivatives yields after simplification:

$$u_i = \varphi_i - \omega_i q_i\left(D_1^{l_i}(\varphi_i) + \psi_i\right). \tag{31}$$

Application of the inverse distance interpolation combines individual functions u_i (31) into a single global function u:

$$u = \sum_{i=1}^{n} u_i W_i = \sum_{i=1}^{n}\left(\varphi_i - \omega_i q_i\left(D_1^{l_i}(\varphi_i) + \psi_i\right)\right)\frac{\prod_{j=1;j\neq i}^{n}\omega_j^2}{\sum_{k=1}^{n}\prod_{j=1;j\neq k}^{n}\omega_j^2} \tag{32}$$

Example. Let us once more interpolate functions specified over point sets Ω_1, Ω_2, Ω_3, and Ω_4 (Figure 9 (a)) as follows:

$$\begin{aligned}
&u_{|\Omega_1} = 0; && && \\
&u_{|\Omega_2} = 5; && \frac{\partial u}{\partial \mathbf{l}_2}_{|\Omega_2} = 3; && \mathbf{l}_2 = (0,-1) \\
&u_{|\Omega_3} = -1; && \frac{\partial u}{\partial \mathbf{l}_3}_{|\Omega_3} = 5; && \mathbf{l}_3 = (0,1) \\
&u_{|\Omega_4} = 2; && \frac{\partial u}{\partial \mathbf{l}_4}_{|\Omega_4} = 2y+3; && \mathbf{l}_4 = (-1,1).
\end{aligned} \tag{33}$$

The reader may notice that we used the same expressions for functions and derivatives as we used in the earlier example (24). But now in addition we also indicated that derivatives on Ω_2, Ω_3, and Ω_4 are prescribed in the directions $\mathbf{l}_2$, $\mathbf{l}_3$, and $\mathbf{l}_4$ respectively. Figure 11 shows the resulting interpolating function that satisfies all prescribed conditions (33) exactly.

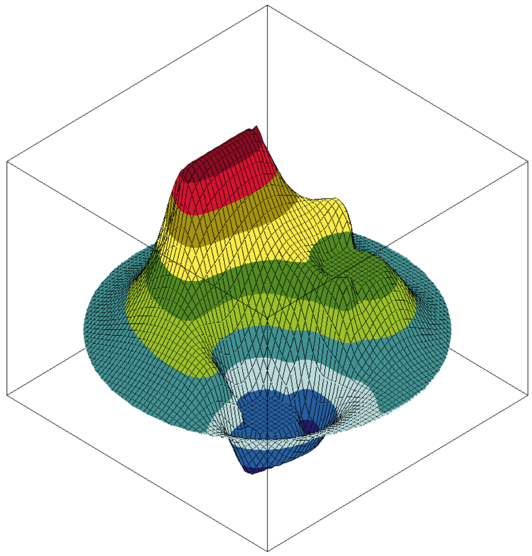


Figure 11: Function interpolating given values and directional derivatives over heterogeneous in dimension objects shown in Figure 9(a)

4 Conclusions

4.1 Summary

The method of transfinite interpolation described in this paper has numerous applications and advantages, when compared to other methods of interpolation. In particular, because the method places no restrictions on topology, adjacency, and dimension of the interpolated sets, it eliminates the need for preprocessing or meshing and allows effortless modification and updates of the input geometry and data, which is particularly useful in modeling time-varying boundary conditions and geometry [21]. The original idea is due to Rvachev [11, 12] who first suggested the technique (albeit in a somewhat restricted setting) as part of the theory of R-functions and quite independently of the inverse distance weighting technique. The present paper is intended to show the full power and generality of the method and its application to transfinite interpolation of both functions and derivatives over the implicitly defined sets of points.

The fact that our method may be viewed as a generalization of the classical inverse distance method is revealing both in terms of power and limitations of the technique. For example, it should be clear that weight functions are globally dependent on all geometric data, and that this dependency may be reduced or eliminated by modifying the implicit functions to have local support.

An apparent limitation of the presented techniques is that it requires associating all geometric data with implicit equations of the form $\omega = 0$. Such representations are already popular in computer graphics. Theory of R-functions offers a general algorithmic approach to constructing implicit functions for a variety of geometric point sets that are normally created using set operations on implicit primitives or represented by boundary representations and other heterogeneous cell complexes. The process of constructing such equations for free-form parametric curves and surfaces is often called implicitization and is known to be practically intractable for even moderately high degree parameterizations. However, because we use such implicit functions to represent the interpolant and not the geometry *per se* (which is given in some other form), it is likely that reasonable approximate implicitization methods may suffice. It should also be clear that properties of the interpolant will depend strongly on the properties of the constructed implicit functions. Differential and normalization properties of implicit functions constructed algorithmically using R-functions are reasonably well understood [20].

The interpolation problem arises in numerous computer-aided design and analysis applications, and the proposed techniques generalize to higher dimensions in a straightforward manner. The problems of constructing a surface interpolating a given set of curves, or a curve interpolating a finite set of points are probably the more familiar but very restricted special cases of transfinite interpolation. In many practical situations only some values of functions and/or their derivatives may be known at the certain locations; other are unknown and have to be determined based on the appropriate physical or mathematical principles. In this case, these unknowns must also appear in the Taylor polynomial u in expressions (14) and (27). For example, in [7, 8] unknown derivatives at scattered points are approximated by different types of polynomials. Similar transfinite interpolation techniques may be used to approximate functions and their derivatives over more general implicitly defined sets [14]. More generally, the interpolation of prescribed values and derivatives is only a first step in solving problems of engineering analysis and mathematical physics. In fact, the described interpolation method was developed precisely for that purpose [11]. We conclude with a brief explanation of the connection between the transfinite interpolation and boundary value problems.

4.2 Fairing/Smoothing And Boundary Value Problems

The interpolation problem has no unique solution in the sense that there are infinitely many functions interpolating any given data. A suitable function can be chosen by putting additional constraints on the interpolant. Often such constraints appear as a requirement of minimization of some quantity. For example, the well known Lagrange interpolation minimizes the degree of the interpolating polynomial. Many interpolation schemes use minimization of potential energy of tension or bending [3] as a means for controlling the shape of the interpolant. Interpolating cubic splines minimize potential energy of a bending beam fixed at data points. In the case of functions of two independent variables, minimization of potential energy of membrane or a thin plate can be used. In all such cases, the interpolation problem is a special case of some boundary value problem.

On the other hand, mathematical formulation of any boundary value problem consists of a differential equation (or equivalent variational statement) constraining the distribution of physical field inside domain and boundary conditions specifying the interaction of the field with the external environment. In this case, interpolation of the prescribed boundary conditions describes the behavior of the solution in the vicinity of the boundaries; the behavior of the interpolant away from the boundaries is quite arbitrary. Thus, the solution of a boundary value problem can be viewed as an interpolating function that extends the boundary conditions into the domain, with differential equation playing the role of a constraining or smoothing operator.

The connection between transfinite interpolation and the boundary values problems is explicit in the R-function method (RFM) for solving boundary value problems. The key concept of RFM is that of a *solution structure* for a boundary value problem – an analytical expression that satisfies all given boundary conditions exactly [13, 12, 14] and is complete in the sense of the Weierstrass theorem. A solution structure can be written in its most general form as

$$f = \sum_{i=1}^{n} f_i W_i + \Phi \prod_{i=1}^{n} \omega_i^{\mu_i} \tag{34}$$

where the first term interpolates all given boundary conditions; ω_i, $i = 1, \ldots, n$, are implicit functions associated with point sets where the boundary conditions of order $\mu_i - 1$ are prescribed; and Φ is a linear combination of basis functions $\{\chi_i\}_{i=1}^{N}$:

$$\Phi = \sum_{i=1}^{N} C_i \chi_i. \tag{35}$$

Note that the second term in the expression (34) is independent of the interpolant and therefore does not affect behavior of f at the boundaries. On the other hand, the behavior of f everywhere else is determined by the choice of the coefficients C_i in (35), that are usually chosen so that expression (34) approximates some differential equations or minimizes some functional. It is shown in [14] that the term $\Phi \prod_{i=1}^{n} \omega_i^{\mu_i}$ in the solution structure (34) also assures completeness of the approximation of f in the sense of the Weierstrass theorem.

For illustration purposes, consider the problem of smoothing (or fairing) that often arises in computer-aided geometric design. Depending on a particular applications, the designed surfaces are often chosen to minimize one of several functionals [3]:

- Potential energy of tension of thin membrane:

$$I_1 = \int_\Omega (\nabla f)^2 \, d\Omega. \tag{36}$$

- Potential energy of bending of thin plate:

$$I_2 = \int_\Omega \left((\nabla^2 f) \right)^2 \, d\Omega. \tag{37}$$

- Energy of thin plate in tension analogy, which is a basically a linear combination of the two previous functionals:

$$I_3 = (1-\alpha) \, I_1 + \alpha I_2, \qquad 0 \le \alpha \le 1. \tag{38}$$

  with parameter α controlling the amount influence of the individual functionals I_1 and I_2 on the interpolant.

To obtain the desired approximation for f, the solution structure (34) is differentiated and integrated as required to construct and solve a system of linear equations for coefficients C_i. Figure 12 shows the results of computations for the boundary value problem specified by

$$\begin{aligned} u_{|\Omega_1} &= 0; \\ u_{|\Omega_2} &= 5; \\ u_{|\Omega_3} &= -1; \\ u_{|\Omega_4} &= 2; \end{aligned} \tag{39}$$

on circle Ω_1, line segment Ω_2, and on circular regions Ω_3 and Ω_4 shown earlier in Figure 9(a). In all cases, the resulting surface represents approximation of the solution structure (34) with bicubic B-splines on a uniform rectangular 40×40 grid chosen as basis functions $\{\chi_i\}_{i=1}^N$. The coefficients C_i of B-splines were computed to minimize the functional (38) with varying values of the parameter α. Figure 12(a) shows only the interpolant term of the solution structure; Figure 12 (b) shows computed f for $\alpha = 0$, corresponding to the membrane functional (36). Notice the discontinuity of the first derivatives along the line segment Ω_2. This discontinuity of the interpolant can be smoothed out by setting $\alpha = 1$, corresponding to the thin plate functional (37). The resulting function is shown in Figure 12(f). Functions in Figures 12 (c),(d), and (e) were computed for intermediate values of $\alpha = 0.05$, $\alpha = 0.25$, and $\alpha = 0.5$ respectively.

Acknowledgments

This work was supported in part by the National Science Foundation grants DMI-9502728, DMI-9900171, CCR-9813507, and the NATO Linkage grant PST.CLG.976192. V. L. Rvachev was also supported by grant SPU 071055 of the International Soros Science Education Program (ISSEP).

References

[1] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, 1997.

[2] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

[3] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A K Peters, 1993.

[4] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting: An Introduction*. Academic Press, Ltd, 1986.

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

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[7] R. J. Renka. CSHEP2D: Cubic Shepard method for bivariate interpolation of scattered data. *ACM Transactions on Mathematical Software*, 25(1):70–73, 1999.

[8] R. J. Renka and R. Brown. TSHEP2D: Cosine series Shepard method for bivariate interpolation of scattered data. *ACM Transactions on Mathematical Software*, 25(1):74–77, 1999.

[9] E. Rimon and D. E. Koditschek. Exact robot navigation in geometrically complicated but topologically simple environment. In *IEEE International Conference on Robotics and Automation*, May 13-18 1990.

[10] V. Rvachev, T. Sheiko, V. Shapiro, and J. Uicker. Implicit function modeling of solidification in metal casting. *Transaction of ASME, Journal of Mechanical Design*, 119:466–473, December 1997.

[11] V. L. Rvachev. *Geometric Applications of Logic Algebra*. Naukova Dumka, 1967. In Russian.

[12] V. L. Rvachev. *Theory of R-functions and Some Applications*. Naukova Dumka, 1982. In Russian.

[13] V. L. Rvachev and T. I. Sheiko. R-functions in boundary value problems in mechanics. *Applied Mechanics Reviews*, 48(4):151–188, 1995.

[14] V. L. Rvachev, T. I. Sheiko, V. Shapiro, and I. Tsukanov. On completeness of RFM solution structures. *Computational Mechanics*, 1999. Accepted for publication in the special issue on meshfree methods.

[15] M. A. Sabin. Transfinite surface interpolation. In Glen Mullineux, editor, *Proceedings of a sixth conference on "Mathematics of Surfaces"*. Oxford University Press, 1996.

[16] N. S. Sapidis. *Designing fair curves and surfaces: shape quality in geometric modeling and computer-aided design*. SIAM, 1994.

[17] V. Shapiro. Theory of R-functions and applications: A primer. Tech. Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY, 1991.

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

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

[20] V. Shapiro and I. Tsukanov. Implicit functions with guaranteed differential properties. In *Fifth ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, 1999.

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

[22] D. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *Proceedings 23rd ACM National Conference*, pages 517–524, 1968.

[23] A. K. Shidlovsky, editor. *Vladimir Logvinovich Rvachev*. Biobibliography of scientists of UkSSR. Naukova Dumka, 1988. In Russian.

[24] A. Sourin and A. Pasko. Function representation for sweeping by a moving solid. In *Proc. Third Symposium on Solid Modeling Foundations and CAD/CAM applications*, May 17-19 1995.

[25] A. Waberski. Vibration statistics of thin plates with complex form. *AIAA J*, 16(8):788–794, 1978.

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

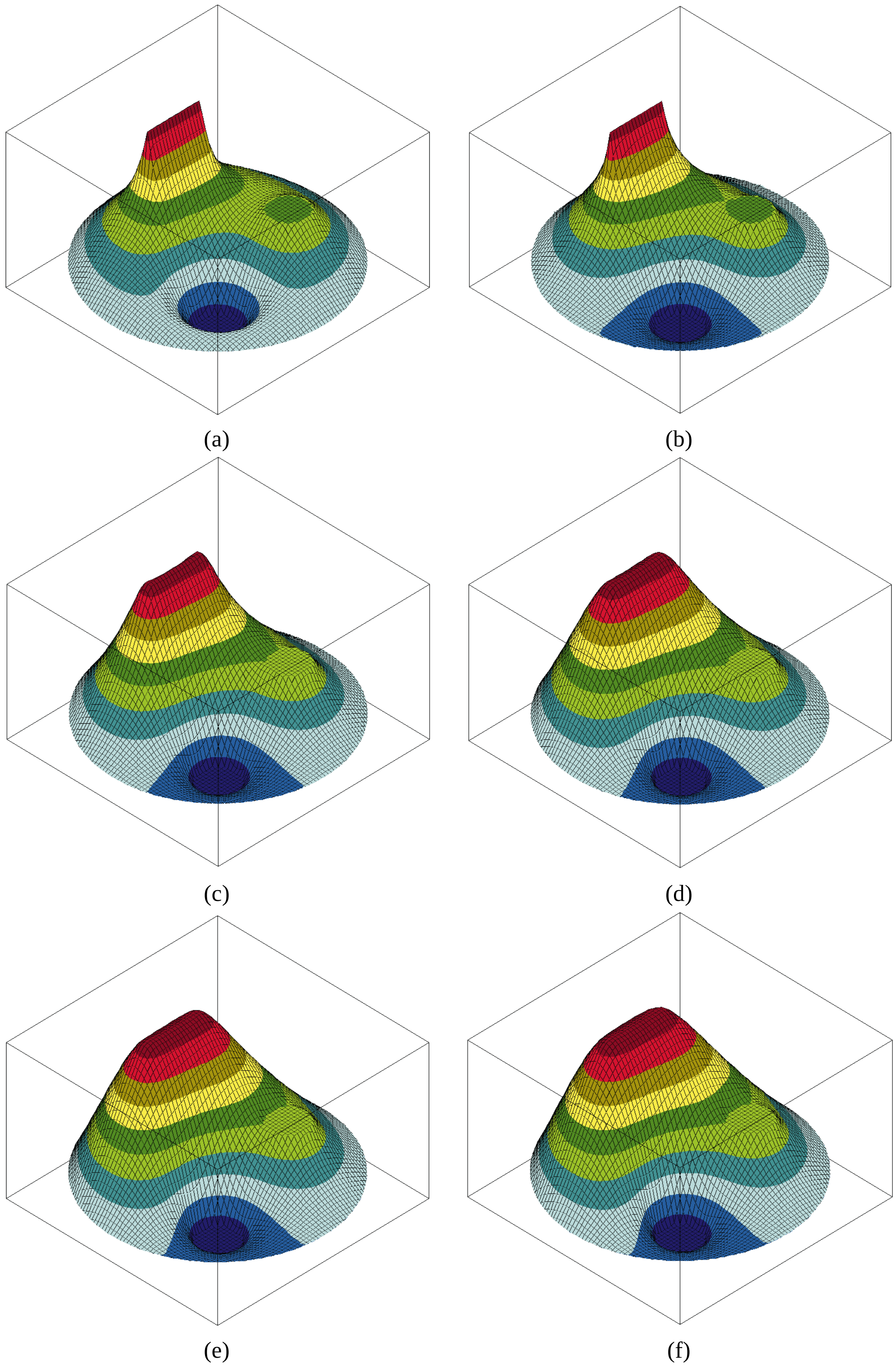


Figure 12: (a) Transfinite interpolant; results of minimization of the functional (38) with (b) $\alpha = 0$; (c) $\alpha = 0.05$; (d) $\alpha = 0.25$; (e) $\alpha = 0.5$; (f) $\alpha = 1$