Tutorials and Reviews

International Journal of Bifurcation and Chaos, Vol. 12, No. 3 (2002) 451–476

MULTIPLE PARAMETER CONTINUATION: COMPUTING IMPLICITLY DEFINED k-MANIFOLDS

MICHAEL E. HENDERSON
IBM Research Division, T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
mhender@watson.ibm.com

Received January 19, 2001; Revised June 11, 2001

We present a new continuation method for computing implicitly defined manifolds. The manifold is represented as a set of overlapping neighborhoods, and extended by an added neighborhood of a boundary point. The boundary point is found using an expression for the boundary in terms of the vertices of a set of finite, convex polyhedra. The resulting algorithm is quite simple, allows adaptive spacing of the computed points, and deals with the problems of local and global overlap in a natural way. The algorithm is robust (the new points need only be near the boundary), and is well suited to problems with large embedding dimension, and small to moderate dimension.

Keywords: Algorithm; continuation; manifold; nonlinear; numerical; union of balls.

1. Introduction

In this paper we present a new multidimensional continuation method for computing points satisfying a system of nonlinear equations

$$M = \{u \subset \mathbb{R}^n | F(u) = 0, \quad F : \mathbb{R}^n \to \mathbb{R}^{n-k}\}. \quad (1)$$

The function F defining M may be algebraic, but we are primarily interested here in cases where F is an approximation to a system of ordinary or partial differential equations. These problems arise when computing steady states of infinite dimensional dynamical systems, and periodic motions and other special solutions in finite dimensional systems and typically n is quite large, $n \sim 10^2$–10^6. A description of techniques for formulating various structures of interest in dynamical systems in this form, and a discussion of the basic computational issues, may be found in [Doedel, 1997], or [Seydel, 1997]. M is an implicitly defined manifold, and if F is smooth, and the Jacobian of F is full rank $(n-k)$ everywhere on M, it is indeed a manifold in the technical sense. Points on M at which the Jacobian is not full rank are an implicitly defined manifold of dimension $k-1$ (Sard's Lemma).

Continuation methods use an initial solution $u_0 \in M$, and compute that part of M which lies inside some finite region Ω, and is connected to the initial solution

$$M_\Omega(u_0) = \{u | u \in M \cap \Omega, \exists \text{ a path on } M \cap \Omega \text{ between } u \text{ and } u_0\}.$$

This is done by building a neighborhood of the initial point on M, and repeatedly adding neighborhoods of points on M until $M_\Omega(u_0)$ is covered. After m steps $M^m \subset M_\Omega(u_0)$ has been covered, and a point $u_m \in M^m$ has been chosen as the next point. A neighborhood $\mathcal{A}_m(u_m) \subset M$ is constructed about u_m, and merged with M^m to create $M^{m+1} = M^m \cup \mathcal{A}_m(u_m)$. To complete the step, a new point $u_{m+1} \in \Omega$ is chosen from M^{m+1}. If no new point can be found the algorithm terminates.

For the continuation to progress, $\mathcal{A}_{m+1}(u_{m+1})$ must contain some points not in M^{m+1}, so u_{m+1} is usually chosen from the boundary of M^{m+1}.

This is not the usual way of presenting continuation methods, but if we may indulge in a little revisionism, it provides a unified way to present the previous literature. The two main classes of continuation methods — simplicial and predictor–corrector — can be distinguished by the method used to compute the neighborhoods $\mathcal{A}(u)$.

Simplicial continuation is used quite successfully for computing contour lines ($n = 2$, $k = 1$) [Watson, 1992; Jones *et al.*, 1986; Dobkin *et al.*, 1990; Sabin, 1985], and isosurfaces ($n = 3$, $k = 2$) [Lorensen & Cline, 1987; Ning & Bloomenthal, 1993; Bloomenthal, 1988; Zahlten & Jürgens, 1973; Hall & Warren, 1990], (where it is called Marching Cubes). The algorithm in general dimensions [Allgower & Georg, 1980; Allgower & Gnutzmann, 1987; Allgower & Schmidt, 1985], defines the point u_m as the intersection of M and an $(n-k)$-dimensional simplex σ_m^{n-k} in $\mathbb{R}^n$ (the domain of F). The requirement that a point lies on σ_m^{n-k} is k equations, and with the $n-k$ equations defining M this gives a square system. The neighborhood $\mathcal{A}_m(\sigma_m^{n-k})$ is a complex of n-dimensional simplices which has σ_m^{n-k} in its interior. A complex is a general mesh — a set of *compatible* simplices (the intersection of any pair of simplices is empty, or a lower dimensional face shared by both). If M is continuous, the intersection of M and $\mathcal{A}_m(\sigma_m^{n-k})$ is a neighborhood of u_m on M.

To ensure that $\mathcal{A}_m(\sigma_m^{n-k})$ can be merged with M^m (i.e. that the union is a complex), all simplices are selected from a reference decomposition of $\mathbb{R}^n$. So M^{m+1} is M^m and those simplices in $\mathcal{A}_m(\sigma_m^{n-k})$ which are not already in M^m. The next point u_{m+1} (i.e. σ_{m+1}^{n-k}) is any of the $(n-k)$-dimensional faces on the boundary of M^{m+1} which crosses M. If σ_{m+1}^{n-k} is small enough so that it intersects M at most once, a face can be checked for intersection by solving a linear system of size $n-k$ involving the values of F at the vertices of the simplex. There are $\binom{n+1}{n-k+1}$ $(n-k)$-dimensional faces in an n-dimensional simplex, so if the boundary of M^m has already been checked for intersections, checking the boundary of M^{m+1} requires approximately $O(n^k)$ linear system solves.

Predictor–corrector continuation [Garcia & Zangwill, 1981; Govaerts, 2000; Morgan, 1987] has a better asymptotic performance for large n, but requires that F is smooth, and the Jacobian is full rank at u_m. The Implicit Function Theorem states that if $u_m \in M$, and $F_u(u_m)$ is full rank, a mapping $u_m(s)$ from $\mathbb{R}^k$ onto M with $u_m(0) = u_m$ exists in some small neighborhood of the origin. The image of some subset of this neighborhood is $\mathcal{A}_m(u_m)$. If Ω is chosen so that M_Ω contains no singular points, this poses no problem. Singular points can be excluded, for example, by disallowing points of a different degree than u_0. A "branch switching" algorithm is then used to obtain a new point across the singular set.

In one dimension predictor–corrector continuation the neighborhood $\mathcal{A}_m(u_m)$ is the polygonal arc $[u_m(-\Delta s_m), u_m(0), u_m(\Delta s_m)]$, and M^m is a polygonal curve, with vertices on M: $[u_0, \ldots, u_m]$ (i.e. a regular mesh on $\mathbb{R}$ with vertices on M). In one dimension one of the two intervals $[u_m(-\Delta s_m), u_m]$ or $[u_m, u_m(\Delta s_m)]$ is interior to M^m, and the other either lies entirely outside of M^m, or contains the initial point u_0 (if $M_\Omega(u_0)$ is topologically a circle). To merge $\mathcal{A}_m(u_m)$ and M^m the interval of $\mathcal{A}_m(u_m)$ entirely in M^m is removed, and the remaining interval is checked, and shortened if it contains u_0. This gives either $M^{m+1} = [u_0, \ldots, u_m, u_m(\Delta s_m)]$, with $u_{m+1} = u_m(\Delta s_m)$, or $M^{m+1} = [u_0, \ldots, u_m, u_0]$, and the algorithm terminates.

It has proven to be nontrivial to extend predictor–corrector continuation beyond $k = 1$. It is natural to replace the intervals on $\mathbb{R}$ with a simplicial decomposition of $\mathbb{R}^k$ (with vertices on M). However, the merge operation is difficult — trimming intervals is trivial, trimming triangles or higher dimensional simplices is not. Several attempts have been made to avoid generating incompatible simplices, but it will be shown here that for this purpose a better analogy of an interval in higher dimensions is a spherical ball, not a set of simplices.

Perhaps the first attempt to create predictor–corrector methods for higher dimension was [Rheinboldt, 1987, 1988]. Rheinboldt's first algorithm uses a fixed simplicial decomposition of $\mathbb{R}^k$. The point u_m is one of the vertices (s_m) in this mesh, and the neighborhood $\mathcal{A}_m(u_m)$ is the image of the simplices containing s_m under the mapping $u_m(s - s_m)$. M^m is the image of a subset of the mesh, that is, a set of values u_i at vertices s_i. To avoid incompatible simplices when merging the neighborhood with M^m, the parameterization of $u_m(s)$ is carefully chosen so that the vertices s_i of M^m which are in $\mathcal{A}_m(u_m)$ satisfy $u_m(s_i - s_m) = u_i$.

To make the next step u_{m+1} is chosen from the boundary of M^{m+1}.

This algorithm computes a global parameterization of $M_\Omega(u_0)$, and this cannot always be done. The algorithm may reach a point where it is impossible to choose the parameterization of $u_m(s)$. In Rheinboldt's second algorithm, with Brodzik, [Brodzik & Rheinboldt, 1994], and its extension to higher dimensions [Brodzik, 1998], the global mesh was replaced by one which may vary locally. M^m is again represented by a mapping of a complex in $\mathbb{R}^k$ onto M. However, the neighborhood $\mathcal{A}_m(u_m)$ is the image of a fixed triangulation of a spherical ball in $\mathbb{R}^k$. The simplices in M^m which contain u_m are mapped onto the tangent space of M at u_m, giving two sets of simplices, those from M^m, and those from the decomposition of the ball. If the vertices from the sphere which lie inside a circumsphere about any of the k-dimensional simplices from M^m are discarded, and if the complex M^m is Delaunay, then a Delaunay triangulation of the remaining vertices will generate a complex which is compatible with M^m. Eliminating incompatibilities with parts of M^m which do not contain u_m (global self-intersection) would require locating simplices from the other parts of M^m which lie near u_m, and also projecting them onto the tangent space at u_m.

Melville and Mackey [1995] avoided the problem of incompatible complexes for $k = 2$ by representing M^m and $\mathcal{A}_m(u_m)$ as the interior of a set of polygonal curves. The curves are the boundary of M^m and $\mathcal{A}_m(u_m)$, and can be merged by shortening intervals. However, for $k > 2$ the difficulty is again present.

In [Henderson, 1993] the author proposed a boundary representation for $k = 2$, using elliptical arcs in the tangent planes instead of polygonal curves. The algorithm presented here is an extension of that algorithm to $k > 2$, using spherical balls instead of ellipsoids. Since M^m is a list of projected balls, and $\mathcal{A}_m$ is also a projected ball, merging the two is trivial (the challenge becomes finding u_{m+1}). In the rest of this paper we show how to find u_{m+1}. The boundary of a set of overlapping spherical balls will be expressed in terms of the vertices of a set of finite, convex polyhedra in $\mathbb{R}^k$. Then it will be shown that each vertex of these polyhedra that lies outside of the corresponding ball can be associated with a point on the boundary. This is a generalization of the intervals used in one-dimensional predictor–corrector continuation to balls centered at u_m with radius Δs_m.

2. Covering $\Omega \subset \mathbb{R}^k$ with Spherical Balls

We begin with some elementary set algebra. Assuming that M^m is the union of the neighborhoods $\mathcal{A}_i(u_i)$ we derive an expression for the boundary of M^m. Then we show that if M is flat ($n = k$), and if the neighborhoods $\mathcal{A}_i(u_i)$ are spherical balls, there is a very simple algorithm to find a point u_{m+1} on the boundary of M^{m+1}. In Sec. 3 this approach is applied to a general manifold.

2.1. *The boundary of a union of balls in* $\mathbb{R}^k$

Suppose we have a union of neighborhoods $\bigcup_{i=0}^{i<m} \mathcal{A}_i(u_i)$. A point on the boundary of the union must lie on the boundary of one of the individual neighborhoods, but may not be interior to any of the others. For two sets this is simply the set identity $\delta(A \cup B) = (\delta A - \delta A \cap B) \cup (\delta B - \delta B \cap A)$. (Recall that $A - B \equiv A \cap (-B)$, and that δA is the boundary of the set A.) For more than two sets, the boundary can be built up, starting with the boundary of each individual set ($\delta\mathcal{A}_i(u_i)$) and removing pieces ($\delta\mathcal{A}_i(u_i) \cap \mathcal{A}_j(u_j)$) which are interior to the other sets

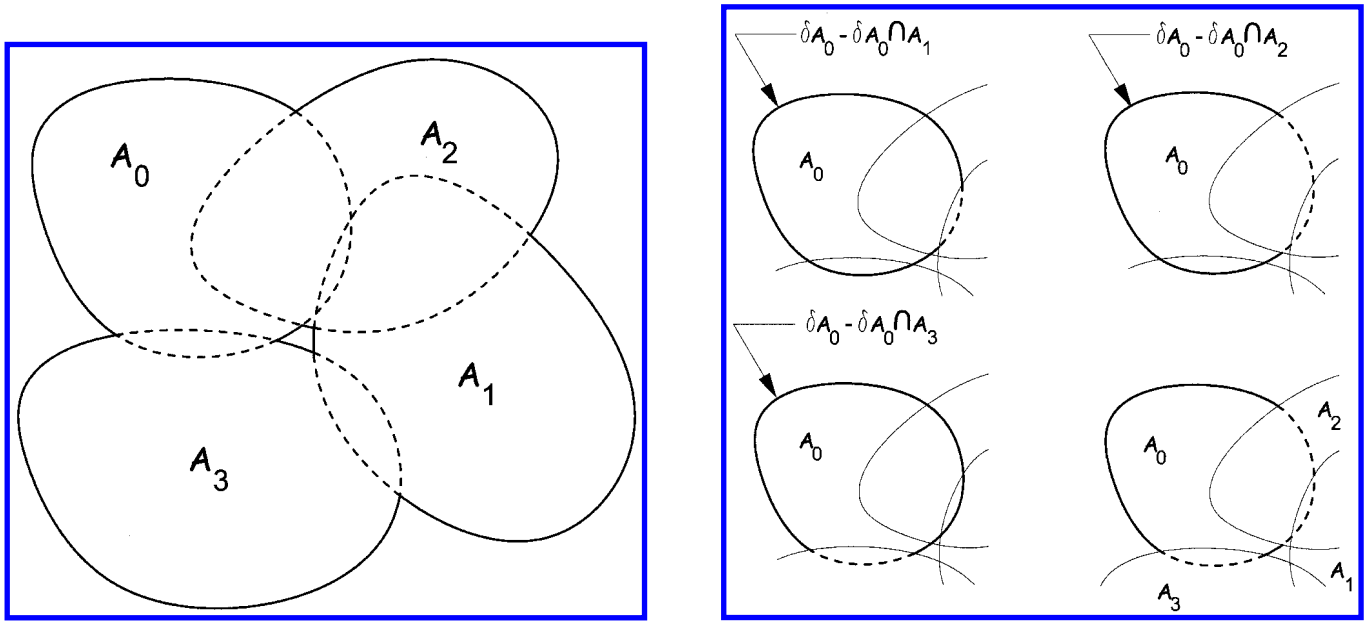
$$\delta\left(\bigcup_{i=0}^{i<m} \mathcal{A}_i(u_i)\right) = \bigcup_{i=0}^{i<m} \left\{ \delta\mathcal{A}_i(u_i) - \bigcup_{j\neq i} \delta\mathcal{A}_i(u_i) \cap \mathcal{A}_j(u_j) \right\}.$$

There are several ways to express the subtractions, for example $A - B - C \equiv (A - B) \cap (A - C)$. Later in this section we will derive a simple expression for the part of $\delta\mathcal{A}_i(u_i)$ not in $\mathcal{A}_j(u_j)$. Using this identity allows us to express the boundary as the intersection of these pairwise subtractions

$$\delta\mathcal{A}_i(u_i) - \bigcup_{j\neq i} \delta\mathcal{A}_i(u_i) \cap \mathcal{A}_j(u_j) = \bigcap_{j\neq i} \{\delta\mathcal{A}_i(u_i) - \delta\mathcal{A}_i(u_i) \cap \mathcal{A}_j(u_j)\},$$

so that

$$\delta\left(\bigcup_{i=0}^{i<m} \mathcal{A}_i(u_i)\right) = \bigcup_{i=0}^{i<m} \bigcap_{j\neq i} \{\delta\mathcal{A}_i(u_i) - \delta\mathcal{A}_i(u_i) \cap \mathcal{A}_j(u_j)\}. \quad (2)$$



(a) (b)

Fig. 1. The boundary of a union of sets. (a) The boundary consists of pieces of the individual boundaries. (b) Each piece may be expressed as the intersection of a set of pairwise subtractions.

Figure 1 illustrates this formula for $m = 4$.

When the neighborhoods are spherical balls ($n = k$ and $\mathcal{A}_i(u_i) = B_{R_i}(u_i) = \{u||u - u_i| \le R_i\}$), the inner terms

$$\delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j)$$

are particularly simple. Each is the set of points u such that

$$|u - u_i|^2 = R_i^2,$$

and

$$|u - u_j|^2 > R_j^2.$$

Because of the symmetry about the axis containing the two centers, we introduce coordinates $(\alpha, \mathbf{a})$, where $\mathbf{a}$ is any vector orthogonal to $u_j - u_i$, and write

$$u = u_i + \alpha(u_j - u_i) + |u_j - u_i|\mathbf{a}.$$

Since $\alpha = (u - u_i) \cdot (u_j - u_i)/|u_j - u_i|^2$, it is clear that the set of points $\alpha = 0$ is a plane containing u_i, and that $\alpha = 1$ is a plane containing u_j. Using this representation of u, we have that $u \in \delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j)$ if and only if

$$\alpha^2 + |\mathbf{a}|^2 = R_i^2/|u_j - u_i|^2 \equiv r_i^2,$$
$$(\alpha - 1)^2 + |\mathbf{a}|^2 > R_j^2/|u_j - u_i|^2 \equiv r_j^2.$$

Eliminating $|\mathbf{a}|$ gives

$$\alpha < \frac{1}{2}(1 + (r_i - r_j)(r_i + r_j)) \equiv \alpha_{ij}. \qquad (3)$$

Since $|\mathbf{a}|^2 = r_i^2 - \alpha^2$ we must also have $|\alpha| < r_i$, and this constraint divides the (r_i, r_j) plane into four regions, delineated by the lines $|\alpha_{ij}| = r_i$ ($|r_i - r_j| = 1$, and $r_i + r_j = 1$). The four regions (see Fig. 2) are

A. The sphere is completely outside the ball ($|u_j - u_i| > R_i + R_j$).
$$\delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j) = \delta B_{R_i}(u_i)\,.$$

B. The ball is completely inside the sphere ($|u_j - u_i| < R_i - R_j$)
$$\delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j) = \delta B_{R_i}(u_i)\,.$$

C. The sphere is completely inside the ball ($|u_j - u_i| < R_j - R_i$).
$$\delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j) = \emptyset\,.$$

D. The sphere and ball intersect ($|R_j - R_i| \le |u_j - u_i| \le R_j + R_i$.).
$$\delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j) = \delta B_{R_i}(u_i) \cap \{u|(u - u_i) \cdot (u_j - u_i)/|u_j - u_i|^2 \le \alpha_{ij}\}.$$

So, each term in the outer union of Eq. (2) (the part of the ith sphere which lies on the boundary)

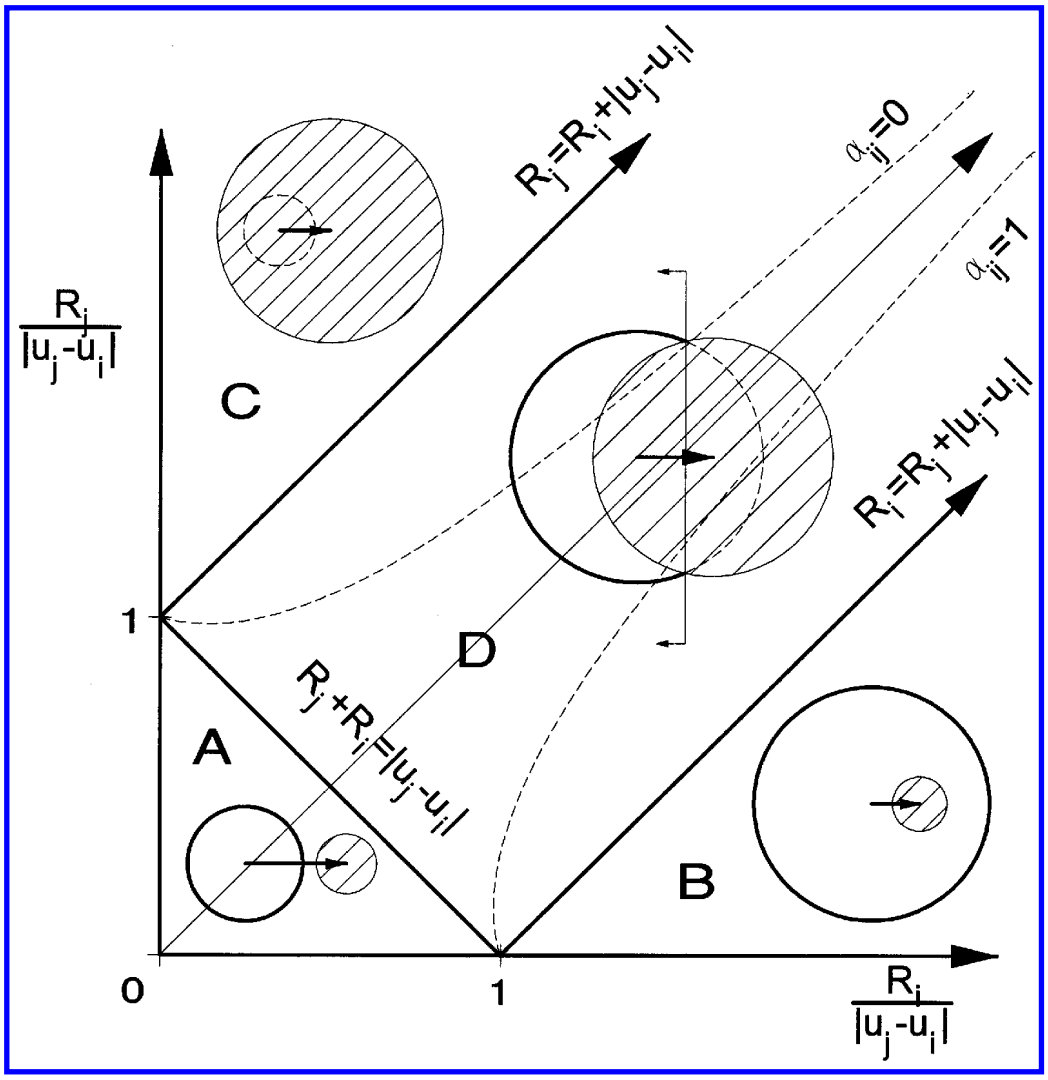


Fig. 2. The part of a sphere which lies outside a spherical ball $\delta B_{R_i}(u_i) - \delta B_{R_i}(u_i) \cap B_{R_j}(u_j)$.

is either empty (if Case C arises for any other ball), the entire sphere $\delta B_{R_i}(u_i)$ (if Cases C and D do not occur), or the part of the sphere which is interior to a convex polyhedron defined by the intersection of the half spaces from Case D. This polyhedron (which may not be finite) can be replaced by a finite polyhedron by first intersecting with any finite, convex polyhedron containing the ball. One choice is the cube

$$C_{R_i}(u_i) = \{u | |u - u_i|_\infty \leq R_i\}\,.$$

Since the intersection of this cube with $B_{R_i}(u_i)$ is the entire ball, this does not effect the result.

With $J_i^m = \{j < m | j \neq i,\ |u_j - u_i| \leq R_i + R_j\}$, (the set of balls which intersect $B_{R_i}(u_i)$) let

$$P_i^m \equiv \begin{cases} \emptyset & \text{if for any } j \in J_i^m,\ |u_j - u_i| < R_j - R_i\,. \\ C_{R_i}(u_i) \bigcap\limits_{j \in J_i^m} HS_{ij} & \text{otherwise} \end{cases}$$

$$HS_{ij} = \{u | (u - u_i) \cdot (u_j - u_i) \leq \frac{1}{2}(R_i^2 - R_j^2 + |u_j - u_i|^2)\}\,.$$

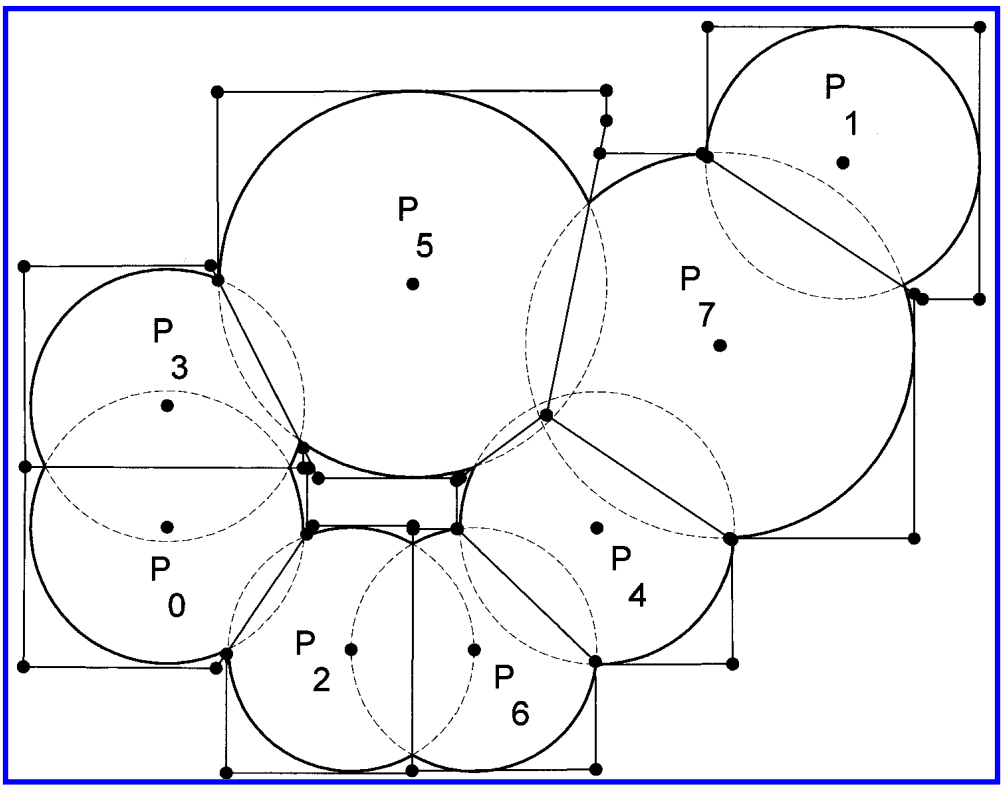


Fig. 3. The set of polyhedra generated by the union of a set of spherical balls.

The boundary of the collection of balls $\mathcal{C}_m = \bigcup_{i=0}^{i<m} B_{R_i}(u_i)$ is then

$$\delta\mathcal{C}_m = \bigcup_{i=0}^{i<m} \delta B_{R_i}(u_i) \cap P_i^m. \tag{4}$$

The relationship between the balls, polyhedra (polygons) and boundary in two dimensions is shown in Fig. 3. Each $P_i^m \subset \mathbb{R}^k$ is a finite, convex polyhedron, and the set of polyhedra covers $\mathcal{C}_m \subseteq \bigcup P_i^m$. P_i^m are disjoint in the sense that

$$\text{interior}(P_i^m \cap B_{R_i}(u_i)) \cap \text{interior}(P_j^m \cap B_{R_j}(u_j)) = \emptyset, \quad i \neq j\,.$$

This result is known, [Aurenhammer, 1988; Imai *et al.*, 1985], and has been used by Edelsbrunner [1995] in three dimensions to compute the topology of molecules. The intersection of the polyhedra with the ball is a subset of the Laguerre Voronoi regions of the centers of the balls. Although we have used spherical balls, there is a generalization to general geometries [Klein, 1987], where the balls are defined by

$$\text{dist}(u, u_i) \leq R_i\,.$$

In this case the faces of the polyhedron are defined by $\text{dist}(u, u_i) = \text{dist}(u, u_j)$.

2.2. *Finding a point on the boundary*

To find a point on $\delta\mathcal{C}_m$, each ball will be checked in turn until one is found for which the term $\delta B_{R_i}(u_i) \cap P_i^m$ in Eq. (4) is not empty. A point in this set is found, and becomes the center of the next ball.

Any point inside a convex polyhedra can be written as a convex combination of the vertices [Grunbaum, 1967]

$$u \in P_i^m \to u = \sum_{v(P_i^m)} \theta_l v_l\,, \quad \theta_l \geq 0\,, \quad \sum \theta_l = 1$$

if every vertex of P_i^m is inside $B_{R_i}(u_i)$, $|v_l - u_i| < R_i$, and

$$|u - u_i| \leq \sum \theta_l |v_l - u_i| < R_i\,.$$

Therefore

Lemma. *If all vertices of P_i^m are inside $B_{R_i}(u_i)$, $B_{R_i}(u_i)$ makes no contribution to the boundary of*

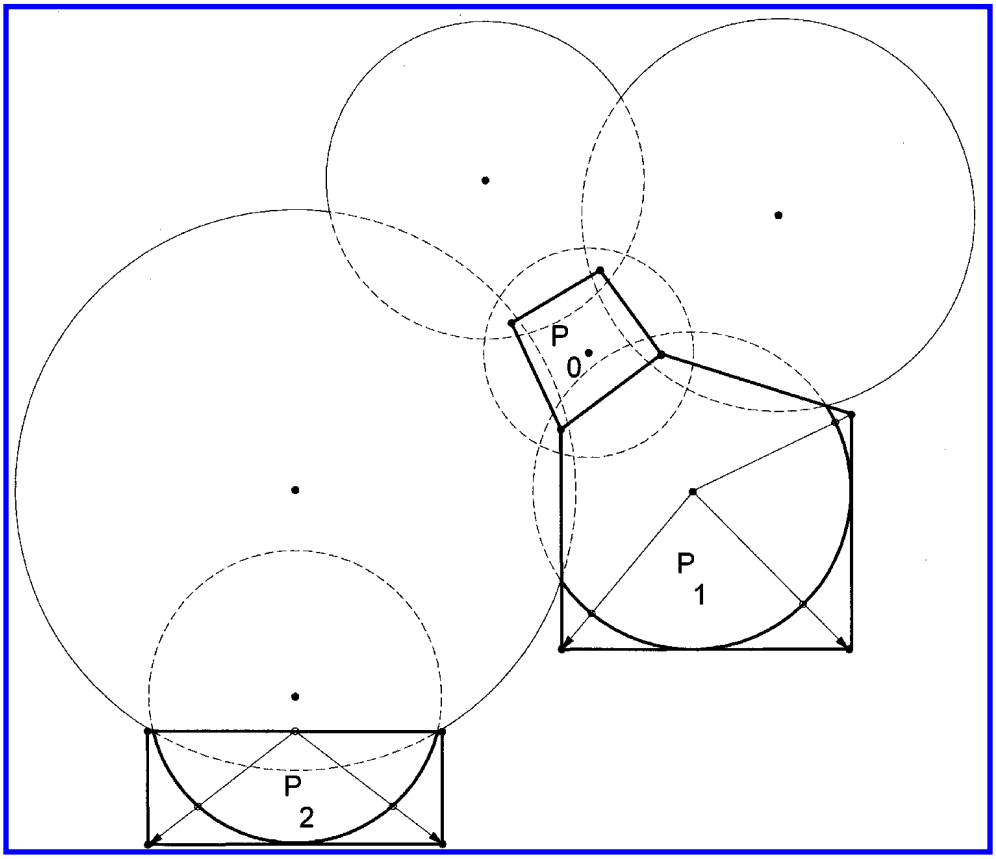


Fig. 4. Finding a point on the boundary of the union of a set of spherical balls (shown for $k = 2$). P_0 has no vertices exterior to $B_{R_0}(s_0)$. P_1 has vertices exterior to $B_{R_1}(s_1)$, and all α's nonnegative. P_2 has vertices exterior to $B_{R_2}(s_2)$, and one negative α.

the union, i.e. $\delta B_{R_i}(u_i) \cap P_i^m = \emptyset$. (*See* P_0 *in Fig. 4.*)

If P_i^m does have a vertex v_e outside $B_{R_i}(u_i)$, we want to find a point in $\delta B_{R_i}(u_i) \cap P_i^m$. P_i^m is convex, so the line joining v_e and any other point u^*, in P_i^m lies in P_i^m. If we can find such a point u^* that is inside $B_{R_i}(u_i)$, the line must cross $\delta B_{R_i}(u_i)$. That is,

Lemma. *If* $v_e \in vertices(P_i^m)$, $v_e \notin B_{R_i}(u_i)$, *and* $u^* \in B_{R_i}(u_i) \cap P_i^m$, *then a line joining* u^* *and* v_e *crosses* $\delta B_{R_i}(u_i)$ *at* $u = u^* + t(v_e - u^*)$, *with* $t \in [0, 1]$ *a root of the quadratic*

$$|v_e - u^*|^2 t^2 + 2(v_e - u^*) \cdot (u^* - u_i)t + |u^* - u_i|^2 - R_i^2 = 0\,.$$

A general approach to finding a point u^* is to solve the positive definite quadratic programming problem (e.g. using the algorithms described in [Gill *et al.*, 1981])

minimize $|\sigma - u_i|^2$

subject to:

$$(\sigma - u_i) \cdot (u_j - u_i) \le \tfrac{1}{2}(R_i^2 - R_j^2 + |u_j - u_i|^2),$$
$$\text{for each } j \in J_i^m\,.$$

The point σ is the closest point in the polyhedron to the center of the ball, so if $|\sigma| > R_i$, the polyhedron is entirely outside the ball ($B_{R_i}(u_i) \cap P_i^m = \emptyset$), and no u^* exists. Otherwise, $u^* = \sigma$.

It is not usually necessary to solve the quadratic program. Any vertex of δP_i^m that is inside the ball can be used for u^*. Also, the center of the ball u_i is expected to lie inside the polyhedron. Recall that P_i^m is defined by the set of half spaces $\alpha \le \alpha_{ij}$, with α_{ij} defined in Eq. (3), and that when $\alpha_{ij} > 0$, u_i ($\alpha = 0$, $|\mathbf{a}| = 0$) is contained in the halfspace, and when $\alpha_{ij} < 1$, u_j ($\alpha = 1$, $|\mathbf{a}| = 0$) is not. So,

Lemma. *If for fixed* i *all* α_{ij} *are nonnegative,* $u_i \in B_{R_i}(u_i) \cap P_i^m$. (*See* P_1 *in Fig. 4.*)

Figure 2 shows the lines $\alpha_{ij} = 0$ and $\alpha_{ij} = 1$. If the local length scale $R(u)$ varies smoothly the ratio R_i/R_j is close to one, and α_{ij} lies between these curves.

Lemma. *If* $|R_i - R_j| \leq |u_j - u_i| \leq R_i + R_j$, *and* $1/\sqrt{2}R_i \leq R_j \leq \sqrt{2}R_i$, *then* $\alpha_{ij} \geq 0$ *and* $\alpha_{ji} = 1 - \alpha_{ij} \geq 0$. (*This follows directly from the expression for* α_{ij}.)

The same result holds if u_i is not inside any of the other balls.

Lemma. *If* $|u_j - u_i| \geq R_j$, *then* $\alpha_{ij}|u_j - u_i|^2 \geq R_i^2/2 \geq 0$.

So, if the ratio of the radius between overlapping balls is close to one, or if the center u_m is outside the other balls, the line from the center to any exterior vertex of P_i^m crosses the sphere, and that point lies on $\delta\mathcal{C}_m$.

This reduces the problem of locating a point on the boundary of a union of balls to that of finding the vertices of a set of convex, finite polyhedra, each defined by the intersection of a finite set of halfspaces. This is straightforward, for example [Chen *et al.*, 1991]. That algorithm starts with vertex and edge lists for the cube $C_{R_i}(u_i)$, and updates them as half spaces are removed.

2.3. *Summary*

When M is flat ($n = k$), and the neighborhoods $\mathcal{A}_m(u_m)$ are spherical balls, the next point in the continuation u_{m+1} can be found by finding a ball whose polyhedron P_i^m has a vertex outside the ball, and then finding a point which is in both the ball and the polyhedron. In most cases the center of the ball can be used. The smallest positive root of a quadratic then gives the point on the boundary.

Merging $\mathcal{A}_m(u_m)$ and M^m requires that complementary halfspaces be removed from P_m^m (which begins as a cube containing $\mathcal{A}_m(u_m)$), and each ball which overlaps $\mathcal{A}_m(u_m)$. If vertex and edge lists are kept for each polyhedron, they can be easily updated using the algorithm of Chen, Hansen and Jaumard [Chen *et al.*, 1991].

3. Covering a Manifold

When M is not flat, the neighborhoods cannot be k-dimensional spherical balls. Instead we will use the projection of spherical balls in the tangent space onto M. We will show that if the balls are small relative to the curvature of M, the projection of a neighborhood into the tangent space at a nearby point can be well approximated by another ball. This allows the results of the previous section to be used in the tangent space to construct an algorithm for covering a general implicitly defined manifold.

3.1. *Projecting from the tangent space onto M*

To project a point from the tangent space onto M, an orthonormal basis Φ_i, for the tangent space at each point u_i is needed. The $n \times k$ matrix Φ_i whose columns are the basis, satisfies

$$\begin{pmatrix} F_u(u_i) \\ \Phi_i^T \end{pmatrix} \Phi_i = \begin{pmatrix} 0 \\ I \end{pmatrix}.$$

The point on M corresponding to a point s in the tangent space at u_i is the solution of the nonsingular nonlinear system

$$\begin{aligned} F(u_i(s)) &= 0 \\ \Phi_i^T(u_i(s) - (u_i + \Phi_i s)) &= 0\,, \end{aligned} \tag{5}$$

which is the projection of the point $u_i + \Phi_i s$ onto the manifold normal to the tangent space. The inverse is simply

$$u_i^{-1}(u) = \Phi_i^T(u - u_i)\,.$$

3.2. *The boundary of neighborhoods on M*

The neighborhoods we use are $\mathcal{A}_i(u_i) = u_i(B_{R_i}(0))$, where $u_i(s)$ is the mapping from the tangent space at u_i onto M. The subset of M covered by a set of these neighborhoods is

$$M^m = \bigcup_{i=0}^{i<m} u_i(B_{R_i}(0))\,.$$

The boundary of this union [Eq. (2)] is

$$\begin{aligned} \delta M^m = \bigcup_{i=0}^{i<m} \bigcap_{j\neq i} \{ & u_i(\delta B_{R_i}(0)) \\ & - u_i(\delta B_{R_i}(0)) \cap u_j(B_{R_j}(0))\}\,. \end{aligned} \tag{6}$$

If u_i and u_j are the centers of two charts that do not overlap, the corresponding term in the inner

union of Eq. (6) is empty. If the charts do overlap, the subtraction can be performed in the tangent space at u_i, (using the inverse of the chart map). With

$$J_i^m = \{j \neq i, j < m | u_i(B_{R_i}(0)) \cap u_j(B_{R_j}(0)) \neq \emptyset\},$$

Eq. (6) becomes

$$\delta M^m = \bigcup_{i=0}^{i<m} u_i \left(\bigcap_{j \in J_i^m} \delta B_{R_i}(0) - \delta B_{R_i}(0) \cap (u_i^{-1} \circ u_j)(B_{R_j}(0)) \right). \quad (7)$$

If the radii are small relative to the curvature of the manifold at the centers of the balls, that is, for some small ε,

$$\forall s \in B_{R_i}(0) \, |u_i(s) - (u_i + \Phi_i s)| \leq \varepsilon \,,$$

then for $s \in B_{R_j}(0)$ we have

$$(u_i^{-1} \circ u_j)(s) \sim \Phi_i^T (u_j - u_i) + \Phi_i^T \Phi_j s + O(\varepsilon) \,,$$

where $|\Phi_i^T \Phi_j| = 1 + O(\varepsilon)$. Which is to say, to leading order, $(u_i^{-1} \circ u_j)(s)$ is a ball $B_{R_j}(\Phi_i^T(u_j - u_i))$. This follows from the Taylor series for $u(s)$,

$$u(s) = u + \Phi s + \frac{1}{2} s^T A s + \cdots \,,$$

where the curvature A satisfies

$$F_u(u_i) A = -F_{uu}(u_i) \Phi_i \Phi_i$$
$$\Phi_i^T A = 0 \,.$$

The assumption on the radius can be approximated as

$$\frac{1}{2} |A| R^2 + O(R^3) < \varepsilon$$

or $R < \sqrt{2\varepsilon / |A|}$.

With this approximation the boundary is

$$\delta M^m \sim \delta \tilde{M}^m = \bigcup_{i=0}^{i<m} u_i \left(\bigcap_{J_i^m} \delta B_{R_i}(0) - \delta B_{R_i}(0) \cap B_{R_j}(\Phi_i^T (u_j - u_i)) \right),$$

which can be written as

$$\delta \tilde{M}^m = \bigcup_{i=0}^{i<m} u_i (\delta B_{R_i}(0) \cap P_i^m) \,.$$

Now,

$$P_i^m = \begin{cases} \emptyset & \text{if } |\Phi_i^T (u_j - u_i)| + R_j > R_i, \text{ for any } j \in J_i^m \\ C_{R_i}(0) \bigcap_{j \in J_i^m} HS_{ij} \,, & \text{otherwise} \end{cases}$$

$$HS_{ij} = \left\{ s | s \cdot \Phi_i^T (u_j - u_i) \leq \frac{1}{2} (R_i^2 - R_j^2 + |\Phi_i^T (u_j - u_i)|^2) \right\}.$$

The effect of the approximation (see Fig. 5) is that (a) a new point may be selected from the approximate boundary that is interior to M^m, or (b) the approximate boundary may be empty when the true boundary is not empty. In the first case, if $\varepsilon \ll \min(R_i, R_j)$ the new point is a distance of order ε inside the image of one of the existing charts, and each step of the algorithm will continue to add volume to the computed part of $M_\Omega(u_0)$. In the second case, the algorithm may terminate when the true boundary still contains pieces of size ε. Neither of these seriously affects the algorithm.

3.3. *Summary*

With a slight modification (projecting the center of overlapping balls into the tangent space before computing the halfspaces), the result of the previous section, that expresses point on the boundary

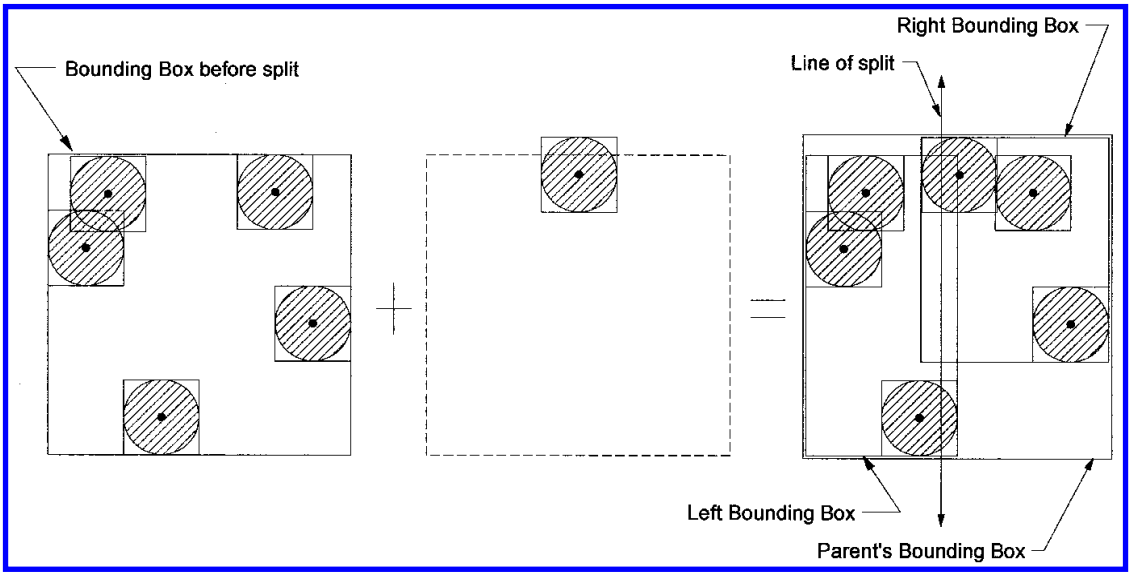


Fig. 5. Adding a ball to the leaf of a set of nested bounding boxes.

of M^m in terms of the vertices of the polyhedra P_i^m holds for manifolds.

We have assumed that a basis for the tangent space can be computed, that the nonlinear system defining the projection from the tangent space to M can be solved, and that a radius for the chart can be found which keeps the distance between the tangent space and M less than some threshold. The first two are the same operations usually performed in one-dimensional continuation (with k arclength constraints instead of one). However, the *a priori* stepsize control is difficult in practice. It requires second derivatives of F, and the solution of a linear system with k^2 right-hand sides. Most one-dimensional continuation methods instead use an *a postiori* control [Govaerts, 2000], which uses an educated guess at the radius when the chart is created, then reduces or increases the radius after a point is projected.

If the estimated radius is too large, either δ, the distance between the tangent plane and the manifold, will be larger than ε, the projection will fail, or the tangent spaces will vary too much between the charts, e.g. $\| \Phi_i^T \Phi_j - I \| > \varepsilon$. In this case the radius used to generate the boundary point (not the radius of the ball itself), may be reduced by some factor, e.g. $u_{m+1} = u_i(0.8 R_i s_e/|s_e|)$, and the projection is attempted again. Conversely, if δ is too small, say less than $\varepsilon/2$, the radius may be increased by a factor of 1.2.

The same technique can be applied to detect bifurcations. If the degree of the center u_i and the projected point u_{m+1} are different, a bisection on the line $u_{m+1}(\theta) = u_i(\theta R_i s_e/|s_e|)$ can be used to locate the point where the degree changes. If the exterior vertices of the polyhedra of balls with centers on the singular set are not used to generate new points the continuation will not cross the singular set.

This stepsize control results in a point which is either inside the boundary (if the stepsize was decreased) or on the boundary. As long as there are exterior points in the neighborhood of the new point there is no adverse effect on the continuation. The alternative is to reduce the radius of the neighborhood, and recompute the polyhedra. This is expensive, and in our experience does not work as well.

The norm used in the tangent space to define the balls does not have to be Euclidean, or even independent of the tangent space. Thus, the balls may be ellipsoidal if a weighted two-norm is used on the tangent space. What is critical is that the projection of the "sphere" in one tangent space be almost "spherical" in the norm used in its neighbors. So if ellipsoids are used the principle axes must not vary too much between neighboring tangent spaces.

4. Implementation

The algorithm described above can be implemented using simple lists. The output is an *atlas*, which is a list of *charts*, each chart having a center u_i, a tangent space Φ_i, a radius R_i, and a polyhedron P_i^m.

The polyhedron consists of two lists, vertices and edges. Each vertex and edge has an index, which is a list of the $(k-1)$-faces on which it lies. In addition, edges have two endpoints, which are vertices.

There are two operations that are potentially linear in m (per step, for a quadratic overall complexity) — the selection of a polyhedron with an exterior vertex, and the construction of J_i^m, the list of balls which overlap the ball being added. These can be reduced to constant and $\log m$ by introducing a simple list of polyhedra with exterior vertices, and a binary tree of nested bounding boxes for the balls.

4.1. *A list of polyhedra with exterior vertices*

Locating a polyhedron with an exterior vertex could require that each of the balls be checked at each step. In fact, the radius of the polyhedron (the radius of a circumsphere with center at the origin) does not increase as additional halfspaces are removed; once a polyhedron has no exterior vertices, it no longer has to be checked. By keeping a list of these balls

$$\mathcal{B}_{\text{list}}^m = \{i | \exists v \in \text{ vertex}(P_i^m) \text{ s.t. } |v| > R_i\}$$

the worst case $O(m)$ time needed to select a boundary point is reduced to constant time.

The first ball in $\mathcal{B}_{\text{list}}^m$ is chosen, and the first exterior vertex s_e is used to generate a boundary point, e.g. if $0 \in P_i^m$, $s = R_i s_e/|s_e|$. If $u_i + \Phi_i s \notin \Omega$, the next exterior vertex is tested, continuing on to the next ball in $\mathcal{B}_{\text{list}}^m$ if necessary. Newly added balls are added to $\mathcal{B}_{\text{list}}^m$, and as balls in the list are checked, those not yielding points on the boundary are removed from the list. Unfortunately, a polyhedron with no exterior vertices in Ω may be trimmed to move a vertex inside Ω, so in principle only balls with no exterior vertices should be removed from the list. In practice, balls with no exterior vertices in Ω should at least be moved to the end of the list. Removing them will result in small regions of Ω near $\delta\Omega$ remaining uncovered, which may be acceptable.

The order in which balls are added can be controlled by sorting the boundary list. For example, sorting according to increasing distance in $\mathbb{R}^n$ from the point u_0, and choosing the first exterior vertex on the most distant ball generates a pattern with tendrils that start with a rough covering of M and then fills in the gap, while choosing the closest ball to u_0 keeps M^m roughly spherical. Other criteria can be imagined — all can be implemented by insertion into a sorted list.

4.2. *A set of nested hierarchical bounding boxes for constructing J_i^m*

Constructing J_i^m, the set of balls that overlap $u_i(B_{R_i}(0))$, could require checking each ball for overlap. Introducing a balanced binary tree [Aho & Ullman, 1992; Tarjan, 1983] of nested bounding boxes for the balls reduces the brute force $O(m)$ algorithm (checking all balls for overlap) to $O(\log\, m)$, provided that the number of overlapping balls is small compared to m. To each node of the basic binary tree a bounding box is added. Terminal nodes store a list of balls, nonterminal nodes divide the bounding box of the parent perpendicular to a coordinate axis, cycling between the coordinate directions from parent to child.

A ball is added to the tree by locating the terminal node which contains the center of the ball. If the number of balls in that node is larger than a threshold, the terminal node is split, so that roughly the same number of balls are in each part. Otherwise the ball is added to the list. The split is done using the next coordinate direction (lexicographically) from the parent. Bounding boxes are updated while locating the terminal node. The balancing is not dynamic, and the nesting is between parent and child, siblings may overlap (see Fig. 6).

To build J_i^m, the bounding box of each node is checked recursively for overlap with $B_{R_i}(u_i)$ (starting with the root node). Since the bounding boxes are nested, if a node's bounding box does not intersect the ball, none of the children will intersect. If the bounding box does overlap, and the node is a terminal node, each of the balls in that node must be checked. If the node is not a terminal node, each of the two children is checked.

It is not necessary to build the tree in $\mathbb{R}^n$, since the manifold is essentially a k-dimensional structure. The centers of the balls should be projected onto some k-dimensional subspace of $\mathbb{R}^n$ before being placed into the tree (this might be the tangent space at u_0).

4.3. *Summary*

For each chart, the center u_i, tangent space Φ_i, radius R_i and polyhedron P_i^m are needed. The set

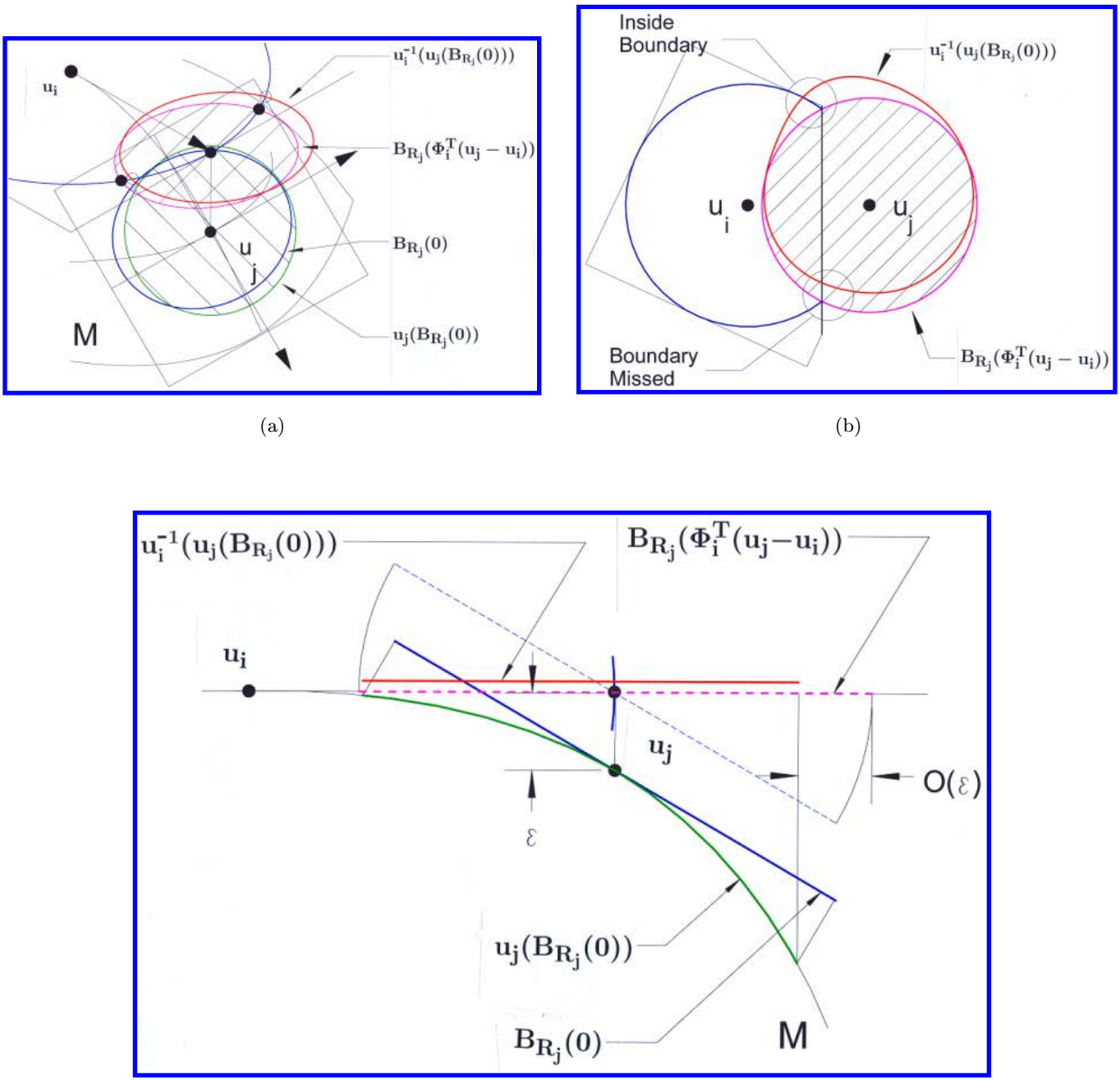


Fig. 6. The effect of the approximation. The sphere in the tangent space at u_j is projected onto M, then onto the tangent space at u_i. It is then approximated by a sphere, which may incorrectly expose or cover parts of the boundary of the sphere at u_i which are of size $O(\varepsilon)$.

J_i^m is constructed using a balanced binary tree with hierarchically nested bounding boxes. A simple list $\mathcal{B}_{\text{list}}^m$ is kept of those neighborhoods on the boundary.

With

$$\begin{aligned} J_i^m &= \{j \neq i, j < m | u_i(B_{R_i}(0)) \\ &\qquad \cap u_j(B_{R_j}(0)) \neq \emptyset\}, \\ HS_{ij} &= \{s | s \cdot \Phi_i^T(u_j - u_i) < (R_i^2 - R_j^2 \\ &\qquad + |\Phi_i^T(u_j - u_i)|^2)/2\}\,. \end{aligned}$$

Initially:

$$\begin{aligned} M^0 &= u_0(B_{R_0}(0)) \\ P_0^0 &= C_{R_0}(0) \\ \mathcal{B}_{\text{list}}^0 &= \{0\}\,. \end{aligned}$$

To add ball $u_m(B_{R_m}(0))$ to M^m:

$$\begin{aligned} M^{m+1} &= M^m \cup u_m(B_{R_m}(0)) \\ P_m^{m+1} &= C_{R_m}(0) \cap_{j \in J_i^m} HS_{mj} \\ P_j^{m+1} &= P_j^m \cap HS_{jm}, \text{ for } j \in J_i^m. \\ \mathcal{B}_{\text{list}}^m &= \mathcal{B}_{\text{list}}^{m-1} \cup \{m\}. \end{aligned}$$

To remove a halfspace from a polyhedron, each vertex s is labeled with the value $s \cdot \Phi_i^T(u_j - u_i) - (R_i^2 - R_j^2 + |\Phi_i^T(u_j - u_i)|^2)$. Edges which connect vertices with values of opposite signs cross the plane, and are shortened, introducing a new vertex. Vertices with positive values are removed, and edges connecting vertices introduced by shortened edges are found [Chen *et al.*, 1991].

To find a point $u_m \in \delta M^m$:

- Locate the first polyhedron P_i^m in $\mathcal{B}_{\text{list}}^m$ with an exterior vertex s_e for which $u_i + \Phi_i s_e \in \Omega$. Remove the charts from the beginning of $\mathcal{B}_{\text{list}}^m$ that have no such exterior vertices.
- If $\alpha_{ij} \geq 0$ for all $j \in J_i^m$,

$$u_m = u_i(R_i s_e / |s_e|)\,.$$

4.4. *Complexity*

This algorithm is roughly equivalent to an incremental computation of the power diagram of a set of balls, [Aurenhammer, 1991; Aurenhammer & Edelsbrunner, 1984; Melhorn *et al.*, 1990]. That algorithm is worst case $O(m^2)$ for $k = 2$ and $O(m^4)$ for $k = 3$ [Aurenhammer, 1991]. This is due to the worst case number of intersections between the inserted region and existing regions. The present algorithm computes the intersection of the Voronoi regions with a cube of side $2R_i$. The number of Voronoi regions that a new region intersects is therefore $|J_i^m|$, significantly less than the worst case for the entire Voronoi regions.

If the number of balls in J_i^m is bounded by some number $F(k)$, and $E(F(k))$ and $V(F(k))$ are the number of edges and vertices in a k-dimensional polyhedron with $F(k)$ $(k-1)$-faces, the complexity of updating the polyhedra is $O(E(F(k))+V(F(k)))$. The number of edges is potentially large for large k. For example, a simplex with $k+1$ faces has $k+1$ vertices and $k(k+1)$ edges. It has been conjectured [Grunbaum, 1967] that

$$E(k) \geq k(k-1)F(k) - 1\,,$$

and we expect that $V(k) \sim F(k)$.

$F(k)$ can be roughly estimated using the following argument. We have a point on a sphere $s \in \delta B_R(0)$, and want to know how many k-spheres $\delta B_R(s_i)$ are needed to guarantee that for some $0 \leq i < F(k)$, $s \in B_R(s_i)$. Clearly this is a covering problem. $\delta B_R(0) \cap B_R(s_i)$ is a spherical cap of radius $\sqrt{3}R/2$, and the question is how many caps are needed to cover $\delta B_R(0)$.

The spherical cap is the projection of a ball of radius $\sqrt{3}R/2$ in the tangent space of $\delta B_R(0)$ at s_i. This means that $F(k)$ is the number of charts of radius $\sqrt{3}R/2$ needed to cover the implicitly defined $(k-1)$-manifold embedded in k-space, $|s|^2 = R^2$. The continuation algorithm itself can be used to estimate $F(k)$. A closely related notion is the Kissing Number τ (the number of spheres in a close packing that touch one sphere). From [Conway & Sloan, 1993]

$$2^{(1-0.5\log_2 3)k(1+o(k))} \leq \tau \leq 2^{0.401k(1+o(1))}$$

The results of computing the number of neighbors, as well as the Kissing number, are given in Table 1.

For large n, the cost of computing a chart can be significant. A direct solver, assuming the Jacobian is banded, and of bandwidth b, is $O(bn^2)$. One factorization and $k + k^2$ backsolves will be needed (k for the tangent, and k^2 for the curvature).

If NLS is the cost of solving the *NonLinear System* Eq. (5), and LS is the cost of solving the

Table 1. Estimates for the size of $|J_i^m|$.

k	Kissing Number	Computed
2	6	6
3	12	13
4	24	25
5	40	42
6	72	78

Linear System which is the Jacobian of Eq. (5), the complexity of computing m charts is

$$O(m \log m) + O(mnk + mE(F(k)) + mV(F(k))) + mNLS + mk(k+1)LS$$

Computing manifolds of large dimension k is expensive. The volume of the manifold, and therefore m, the number of charts needed to cover it, grows exponentially with k, as does the cost of updating the polyhedra. The exact growth rates depend on the numerology of polyhedra (the number of edges in a polyhedra with a certain number of faces), a number similar to the Kissing number. For large n and small k the computation is dominated by the cost of computing a chart, typically at least an n^2 operation.

5. Examples

Below we show how the algorithm performs in two and three dimensions. Several tests can be made to verify that the algorithm is performing correctly.

If the center of chart m is on $\delta\mathcal{C}_m$, then

$$R_j \le |u_j - u_m| \le R_j + R_m, \quad \text{for } j \in J_m^m .$$

If the distribution of $(|u_j - u_i| - R_j)/R_i$ with $j < i$, $j \in J_i^m$ is plotted, all points should lie in [0, 1].

The volume of the polyhedra in the tangent spaces can be computed directly (e.g. [Cohen & Hickey, 1979]). This volume must increase. When M is flat ($N = k$), the polyhedra cover Ω, so the computed volume will be greater than the volume of Ω.

The number of balls overlapping each ball (the number of $(k-1)$-faces in the P_i^m) gives an indication of how well Ω is covered. As was mentioned previously, this is related to the Kissing number, which behaves as 6 for 2-d, then 12, 24 and 40.

The performance of the bounding boxes can be gauged by comparing the size of J_i^m from the nested bounding boxes to the actual number of charts which overlap.

5.1. *Covering* $\mathbb{R}^k$

A computation of the interior of the square $|s| \le 1$ in $\mathbb{R}^2$, is shown in Fig. 7, and the performance measures described above are given in Fig. 8. The balls are discs of radius 0.1, and the square has side of length 2, so the area of the covering should be between 4 and 4.82 ($= 2.2^2$).

The result of applying the algorithm to the interior of the cube $|s| < 1$ is shown in Fig. 9. The balls are the interior of spheres, with radius 0.1, so the volume contained in the polyhedra should be between 8 and 10.65. The final volume is 10.23. The measures of performance are given in Fig. 10.

5.2. *Manifolds*

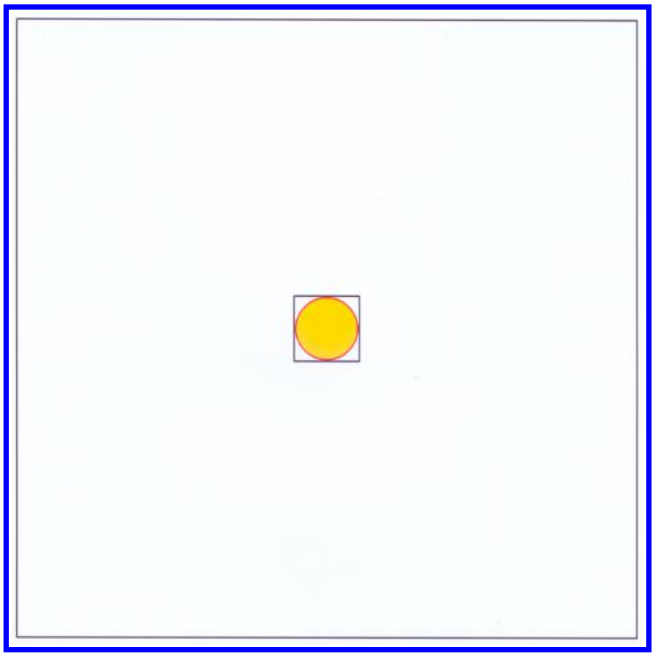
The computation of a circle (a 1-manifold embedded in 2-space) is shown in Fig. 11. The balls in this case are intervals on the tangent line, and the polyhedra are also intervals. This example shows the correspondence with the more usual predictor–corrector algorithms. Apart from the path being extended in both directions, it is identical to pseudo-arclength continuation [Keller, 1977].

In Fig. 12 the algorithm has been applied to compute the surface of a torus.

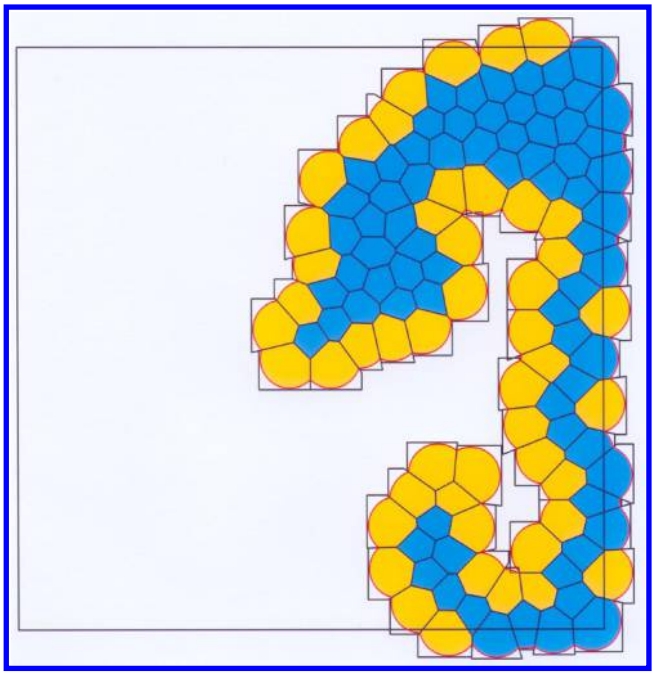
$$(\sqrt{x^2 + y^2} - R_1)^2 + z^2 - R_2^2 = 0 ,$$

with $R_1 = 0.8$, $R_2 = 0.5$. The performance is shown in Fig. 13. This example uses a nonconstant radius, corresponding to $\varepsilon = 0.01$. The surface area of the torus is $4\pi^2 R_1 R_2 = 15.79$ [Zwillinger, 1996], and the computed volume of the polyhedra was 15.70. This is measured in the tangent space, which should slightly underestimate the volume on the surface. The balls in this case are discs, and the polyhedra are polygons. Circles that make up part of the boundary correspond to the lightly shaded discs.

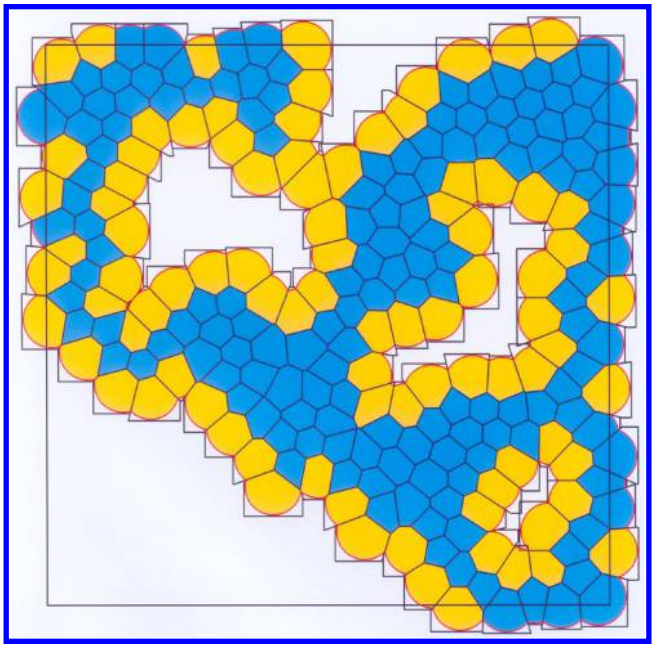
As an example of a manifold embedded in higher dimensions, consider the periodic motions of a pair of coupled pendula [Henderson *et al.*, 1991; Aronson *et al.*, 1991], which can be formulated as the solution of a two point boundary value problem. The two pendula are at angles ϕ_0 and ϕ_1 with respect to the vertical, and constant torques I_0 and I_1 are applied to the pendula. The spring connecting



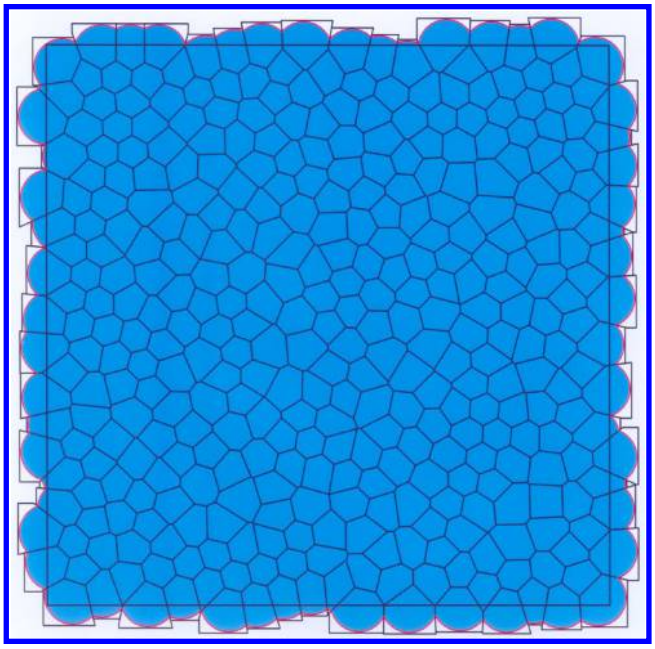
$m = 1$



$m = 121$

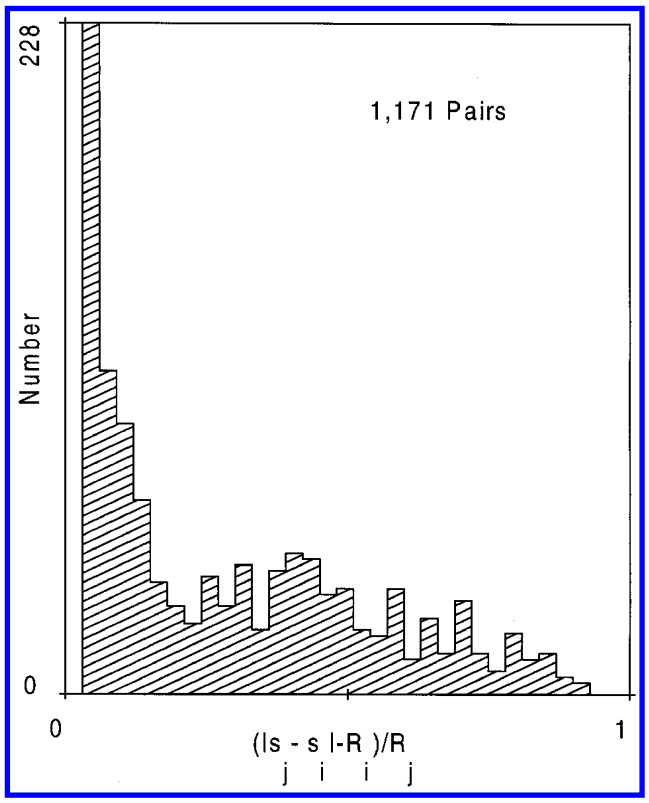


$m = 241$

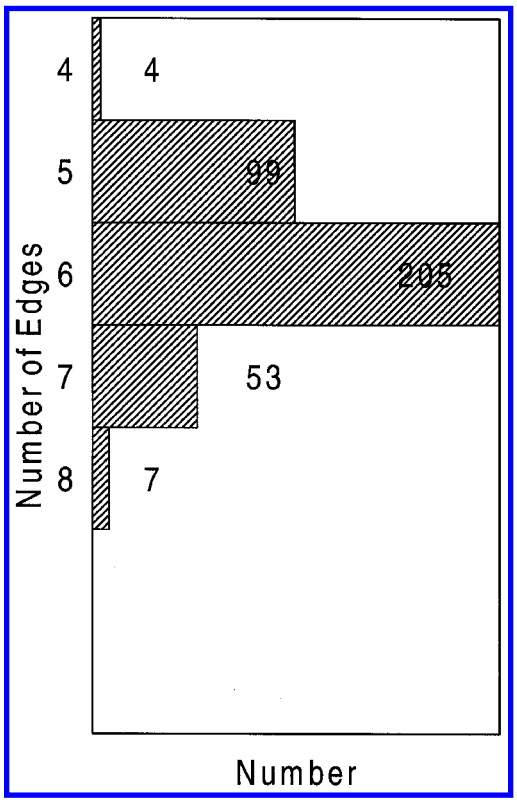


$m = 368$

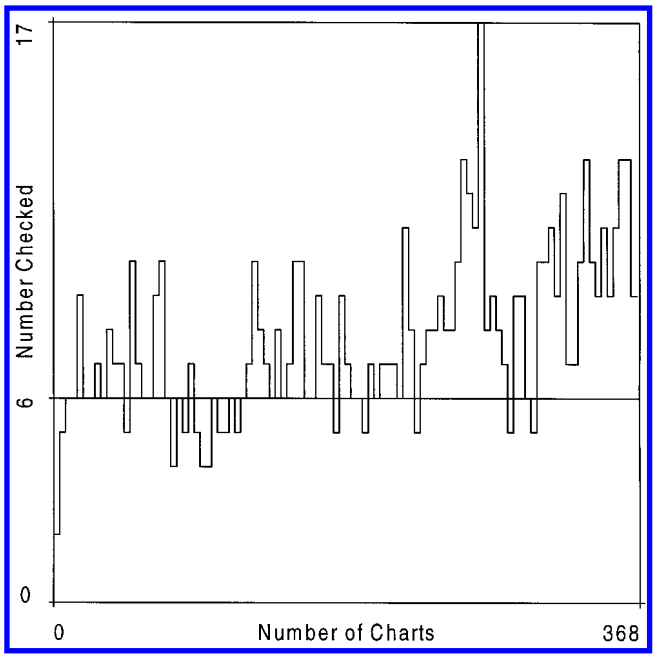
Fig. 7. The interior of a square $[-1, 1] \times [-1, 1]$, with balls of radius 0.1. $k = 2$, $n = 2$. The balls are disks, and the polyhedra are polygons. The result consists of 368 charts.



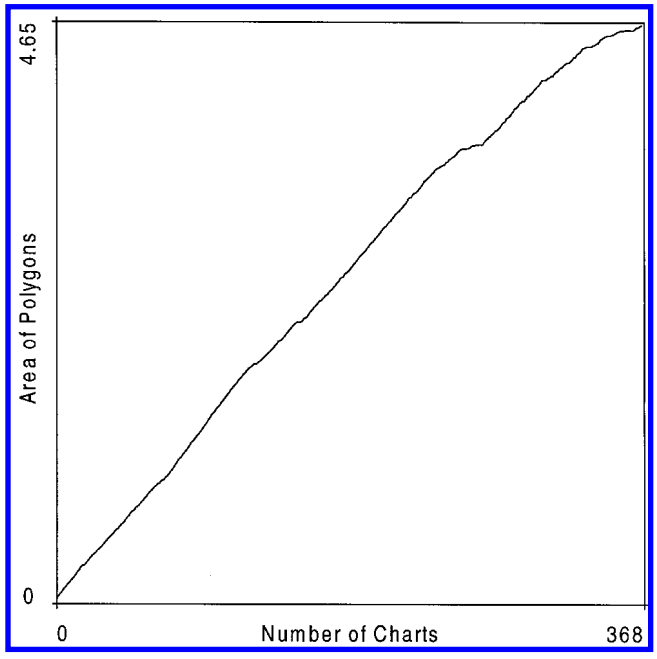
(a)



(b)

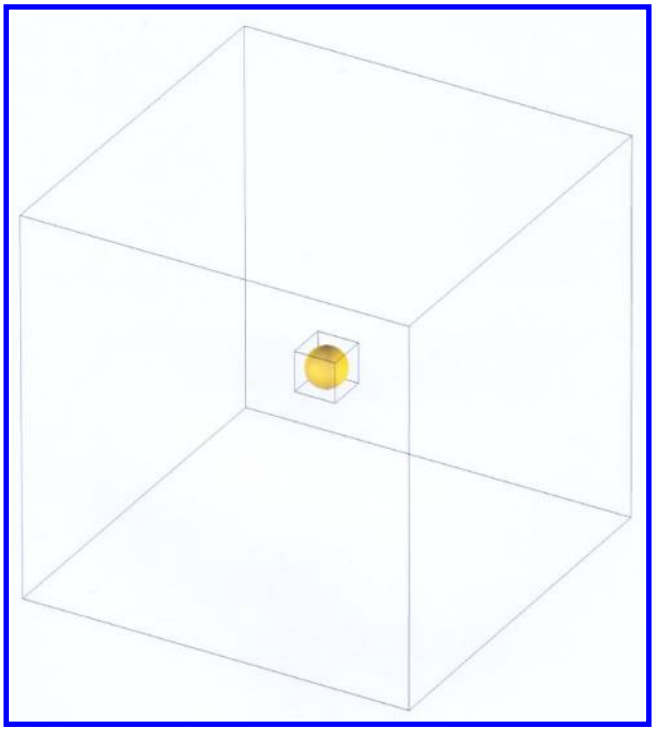


(c)

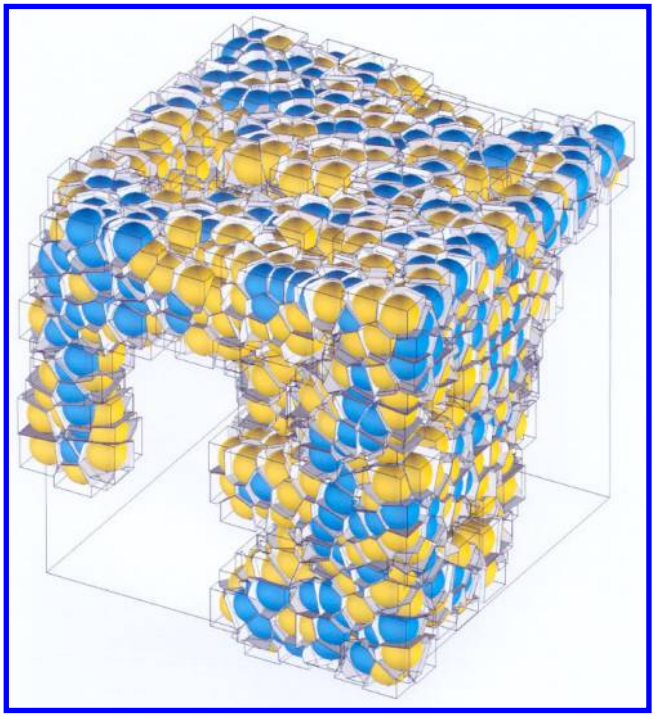


(d)

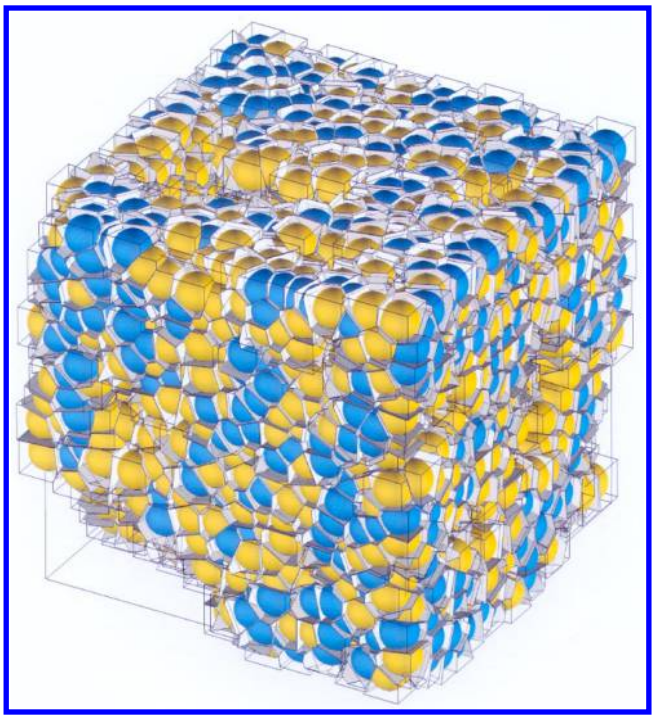
Fig. 8. Performance measures for the interior of a square $[-1, 1] \times [-1, 1]$, with balls of radius 0.1. (a) The distribution of normalized distance between centers. (b) The number of edges in the polygons. (c) The number of charts checked for overlap. (d) $\sum_{i<m} \text{vol}(P_i^m)$, as a function of m.



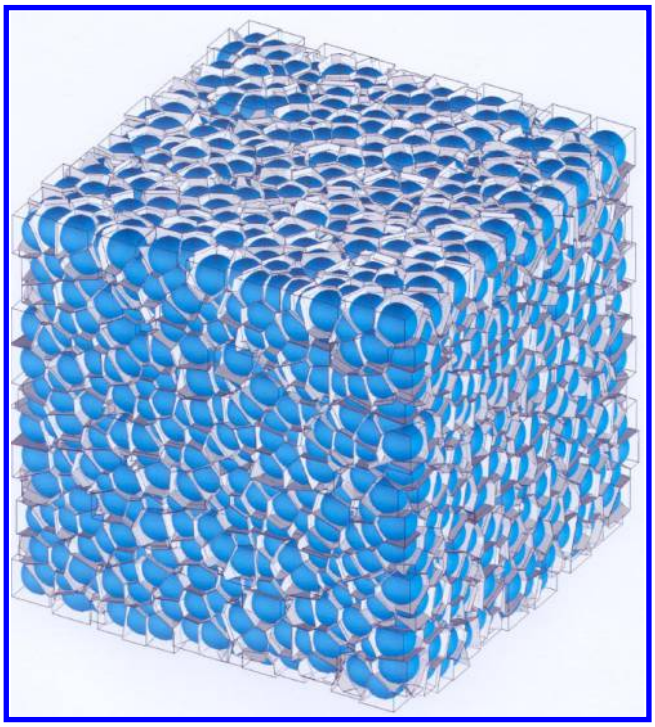
$m = 1$



$m = 2501$

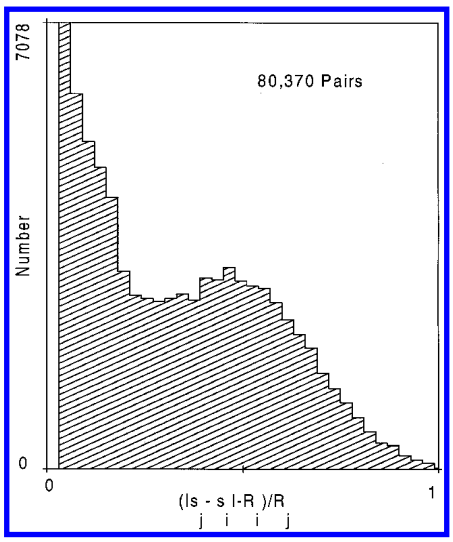


$m = 5001$

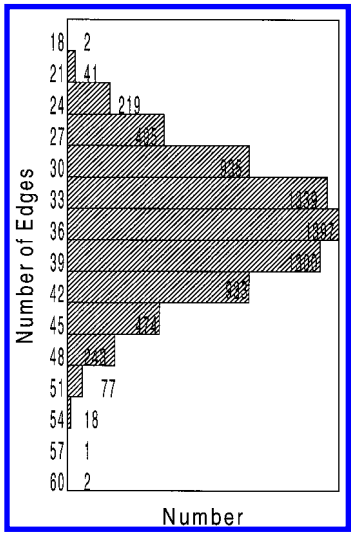


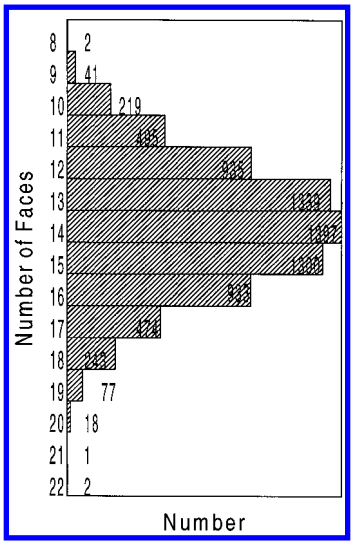
$m = 7476$

Fig. 9. The interior of a cube $[-1, 1] \times [-1, 1] \times [-1, 1]$, with balls of radius 0.1. $k = 3$, $n = 3$. The balls are spheres, and the polyhedra are polyhedra. The result consists of 7,476 charts.

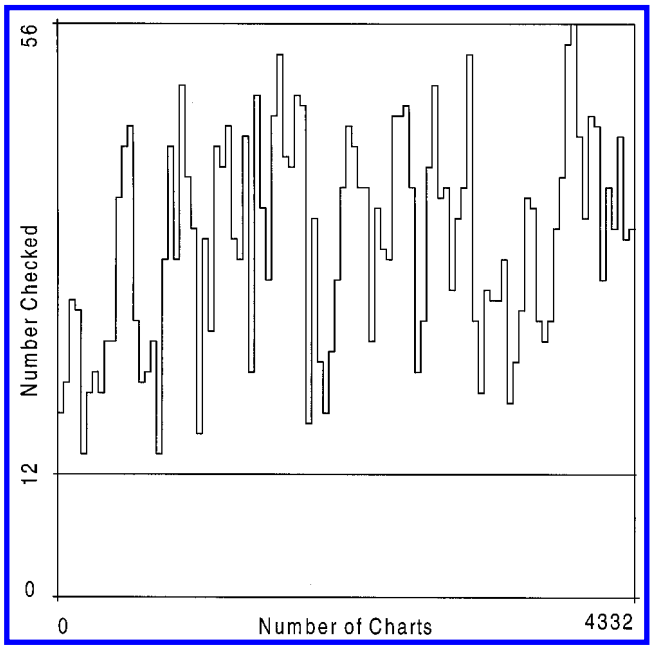


(a)

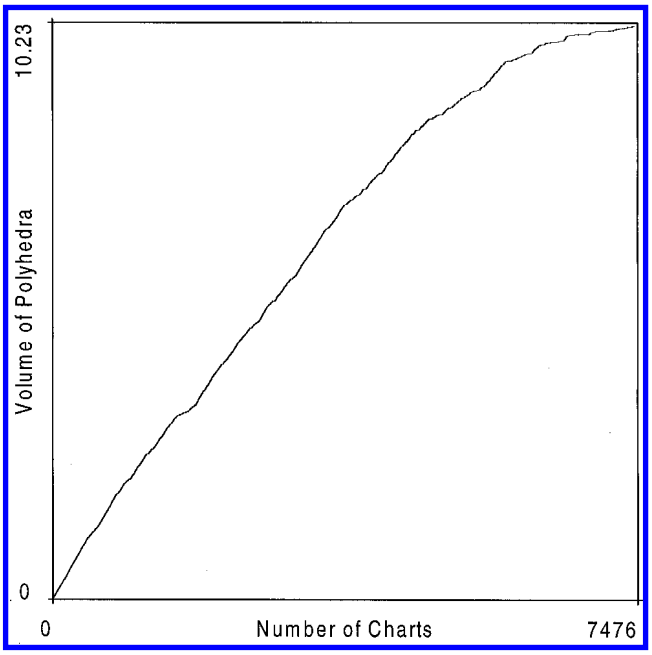




(b)

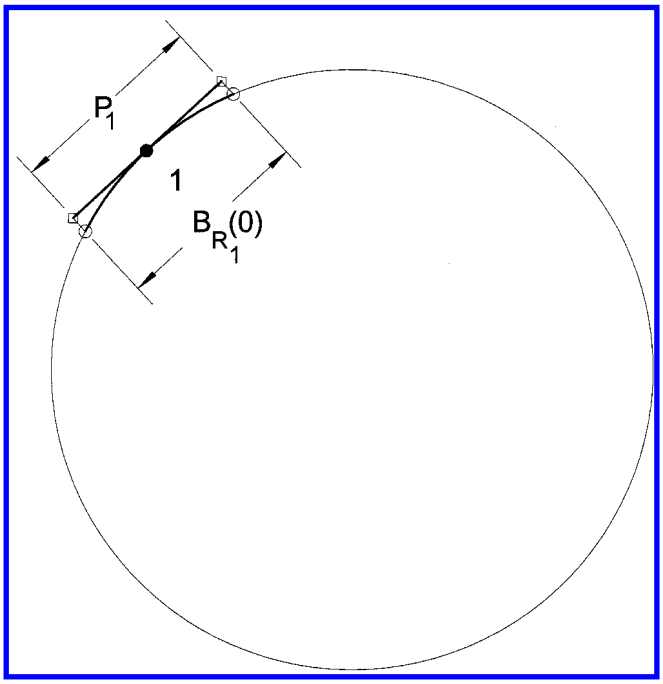


(c)

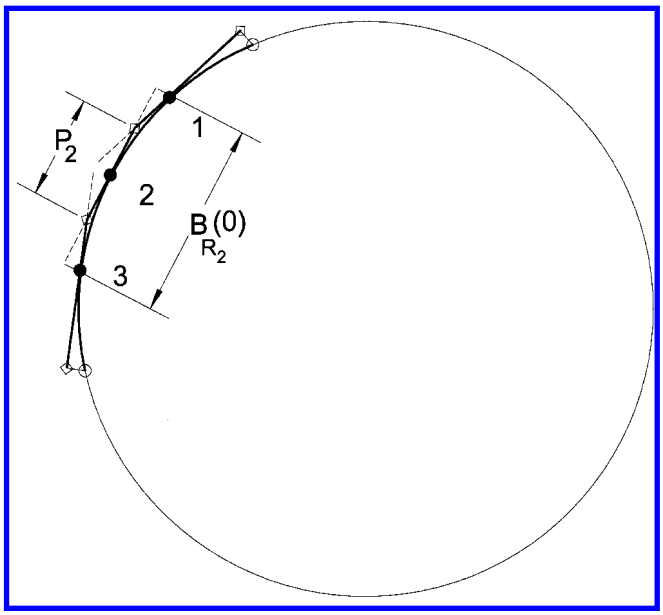


(d)

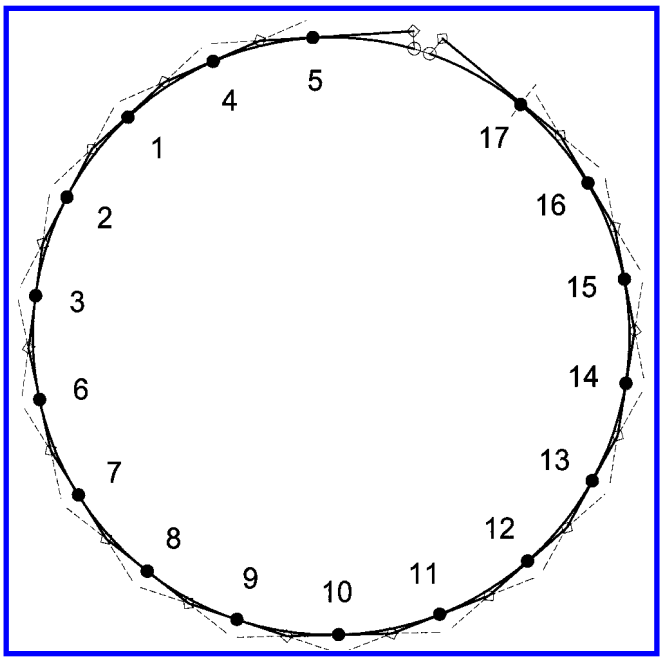
Fig. 10. Performance measures for the interior of a cube $[-1, 1]\times[-1, 1]\times[-1, 1]$, with balls of radius 0.1. (a) The distribution of normalized distance between centers. (b) The number of edges and faces in the polyhedra. (c) The number of charts checked for overlap. (d) $\sum_{i<m} \text{vol}(P_i^m)$, as a function of m.



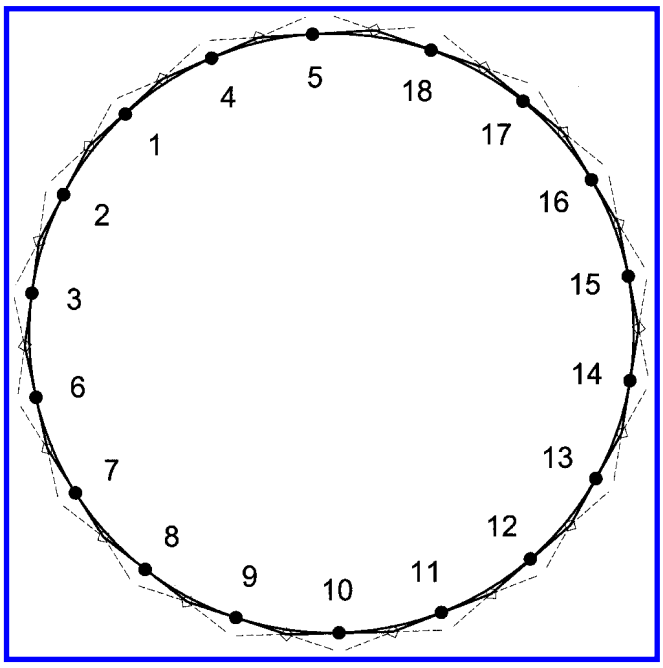
$m = 1$



$m = 3$

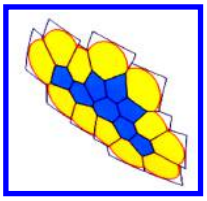


$m = 17$

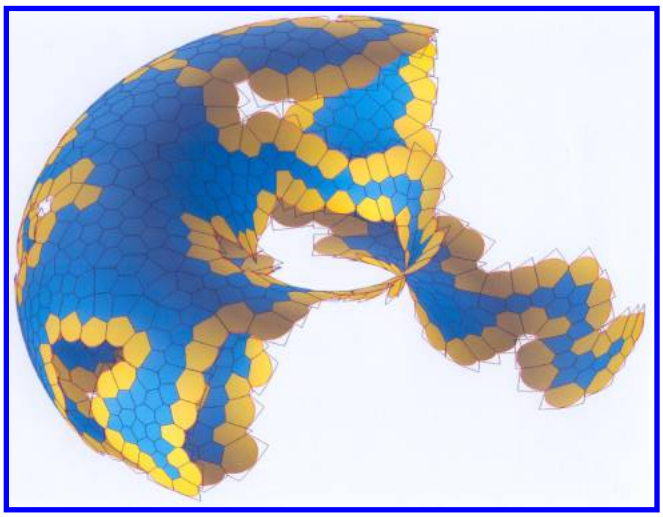


$m = 18$

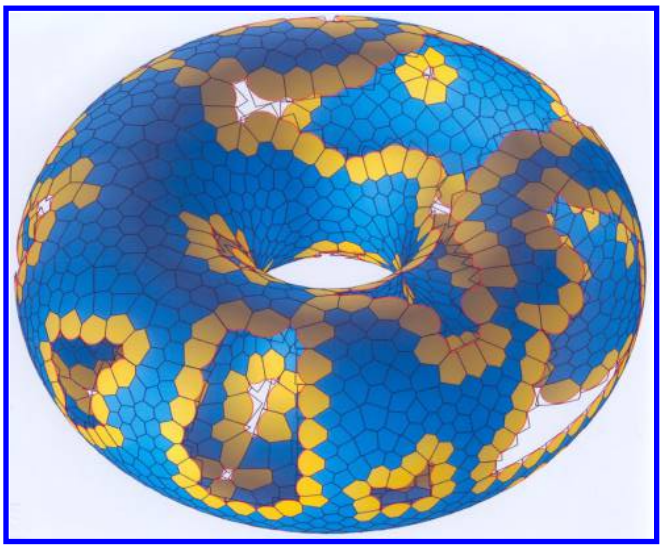
Fig. 11. The perimeter of a circle, $k = 1$, $n = 2$. The balls are intervals, as are the polyhedra. The result consists of 18 charts.



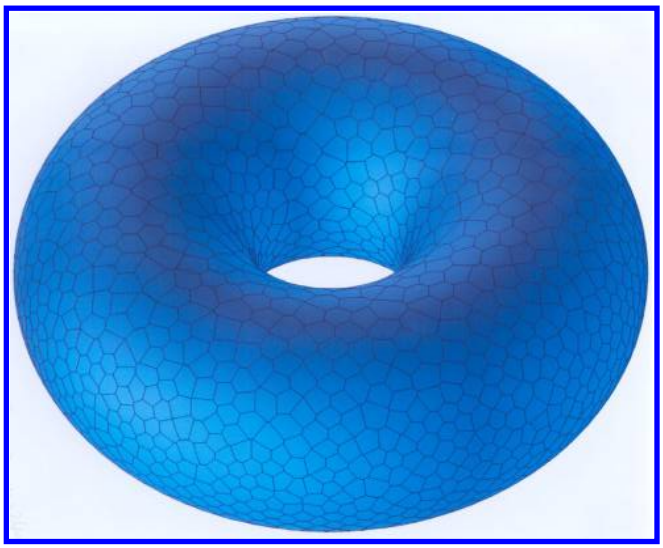
$m = 21$



$m = 701$



$m = 1401$



$m = 2041$

Fig. 12. The surface of a torus, $(\sqrt{x^2 + y^2} - R_1)^2 + z^2 - R_2^2 = 0$, $R_1 = 0.5$, $R_2 = 0.8$, $\varepsilon = 0.01$. $k = 2$, $n = 3$. The balls are circles, and the polyhedra are polygons. The result consists of 2041 charts. The surface area of the torus is 15.79.

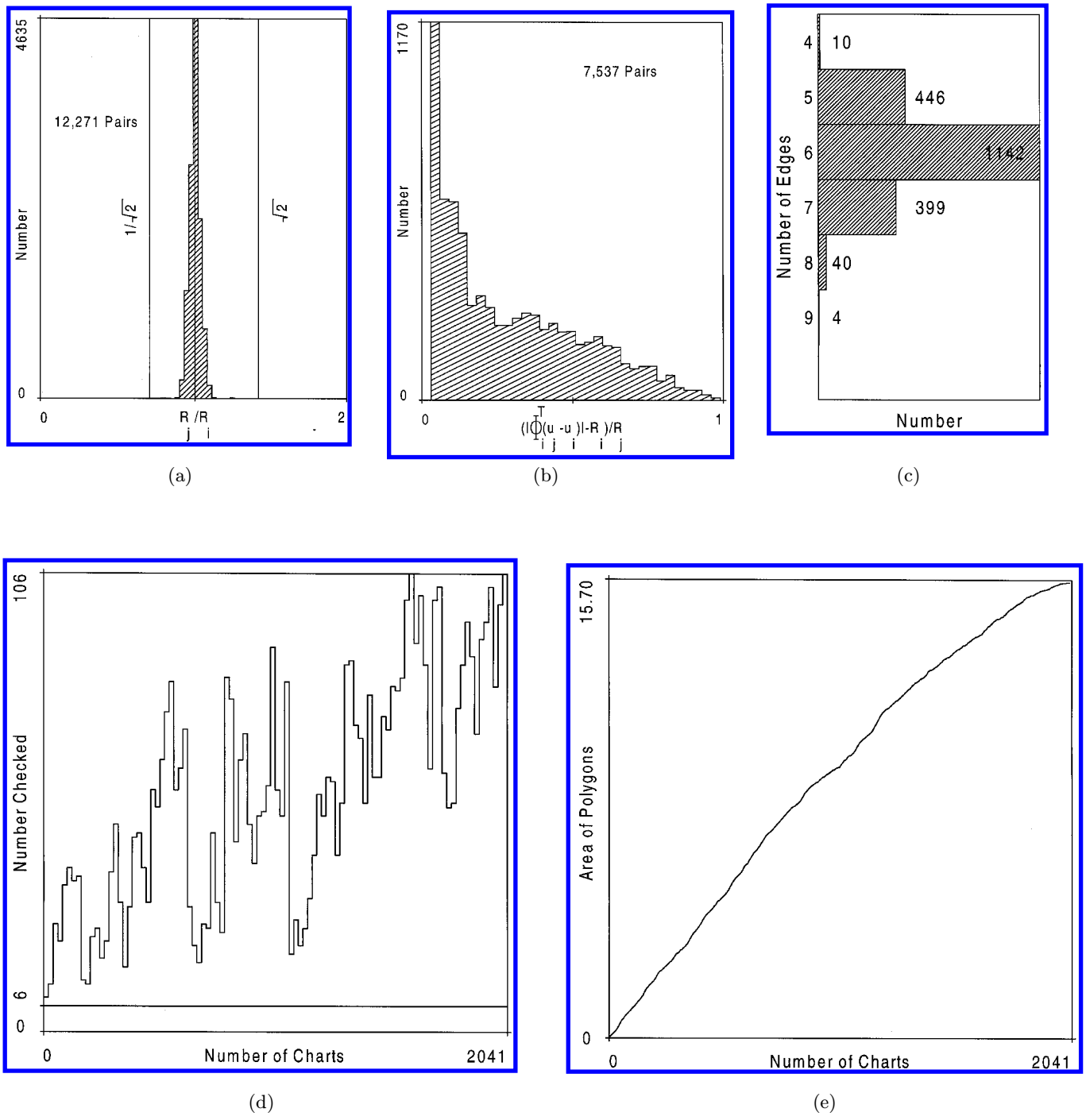


Fig. 13. Performance measures for the surface of a torus, $R_1 = 0.5$, $R_2 = 0.8$, $\varepsilon = 0.01$. (a) The distribution of R_j/R_i for $j \in J_i^m$. (b) The distribution of normalized distance between centers. (c) The number of edges in the polygons. (d) The number of charts checked for overlap. (e) $\sum_{i<m} \text{vol}(P_i^m)$, as a function of m.

the pendula is linear, with spring constant κ, and γ models friction.

Rather than formulate the problem in terms of the individual pendula, we use the half angle between the pendula and the average position of the pendula.

$$\xi = (\phi_0 - \phi_1)/2, \quad \eta = (\phi_0 + \phi_1)/2,$$
$$I = (I_0 + I_1)/2, \quad I_c = (I_0 - I_1)/2$$

This coordinate system is more natural, partly because the spring is linear in the difference, and partly because it is the balance between net torque I and the energy stored in the spring $\sim \kappa\xi^2$ that determines the steady states of the system. The two point boundary value problem is:

$$T^2\eta'' + T\gamma\eta' + \sin\eta\cos\xi = I$$
$$T^2\xi'' + T\gamma\xi' + \sin\xi\cos\eta + 2\kappa\xi = I_c$$
$$\eta(1) = \eta(0) + 2\pi N_w$$
$$\xi(1) = \xi(0)$$
$$\eta(0) = 0$$

These are sometimes called periodic motions of the second kind. The first two boundary conditions are periodicity, the third is a phase constraint. There are four continuous parameters (γ, κ, I, I_c) and one discrete parameter N_w, which is a winding number.

For large average torque I, it can be shown that the only solution is the "synchronous" solution, where the pendula "tumble" or "run" together. For large I, on $I_c = 0$, the family of synchronous solutions is $\xi(t) = 2N_w\pi$, $\eta(t) \sim (I - 2N_w\kappa\pi)t/\gamma T$. This provides the initial solution u_0.

Figure 14 shows the manifold of periodic motions at damping $\gamma = 0.5$, spring constant $\kappa = 0.1$, and winding number $N_w = 1$, over a range of torques. There are two symmetries in the difference of the torques, a reflection about $I_c = 0$, and a translation of $2\kappa\pi$ in I_c. The computation in Fig. 14 shows three copies of the fundamental domain $0 \le I_c \le \kappa\pi$.

6. Conclusions

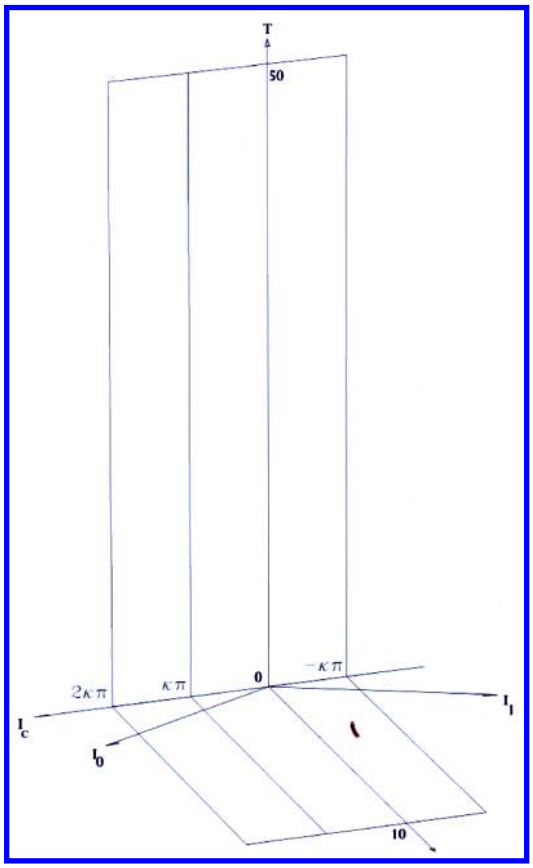
We have presented a new continuation method for computing implicitly defined manifolds. The manifold is represented as a set of overlapping neighborhoods, and extended by an added neighborhood of a boundary point. The boundary point is found using an expression for the boundary in terms of the vertices of a set of finite, convex polyhedra. The resulting algorithm is quite simple, allows adaptive spacing of the computed points, and deals with the problems of local and global overlap in a natural way. The algorithm is robust (the new points need only be near the boundary), and is well suited to problems with large embedding dimension n, and small to moderate values of k. Finally, there is a very transparent parallelism, in that the computational region Ω may be partitioned, and independent computations performed on each piece. Merging the results requires updating polyhedra near the boundary of each piece.

We have given detailed examples for the combinations of n and k up to three, and for a two point boundary value problem with $n = 203$, $k = 2$. We have also used the algorithm to estimate the Kissing number, which is the case $n = k + 1$, for n up to 6.

The algorithm is based on a set of results on the boundary of a union of balls [Edelsbrunner, 1995]. For the flat manifold $\mathbb{R}^3$, the union of the balls is the space filling shape of a molecule, [Brostow *et al.*, 1978; Edelsbrunner, 1995], which places a ball at each of the atoms of a molecule, with a radius related to electronic properties of the atom. This shape determines certain chemical properties of the molecule.

The algorithm resembles the incremental insertion algorithm for computing part of the Voronoi diagram on the surface $F(u) = 0$, with the Laguerre distance [Imai *et al.*, 1985] (also called the Power Diagram [Aurenhammer, 1987, 1988]). However, only that part of the diagram near the sites is computed, and the sites are determined as the algorithm progresses. There are also similarities to advancing front mesh generation on higher dimensional surfaces. The algorithm presented here uses Voronoi

Fig. 14. (*facing page*) The solution manifold for the coupled pendula, $\kappa = 0.1$, $\gamma = 0.5$, computed as a two-manifold. The figure shows I_0 and I_1 on the horizontal axes and the period T on the vertical axis. The computation generated 14,373 charts, the result being a two-manifold in $\mathbb{R}^{203}$. The region Ω is $0 \le I \le 10$, $-1 \le I_c/\kappa\pi \le 2$, and $T < 50$.

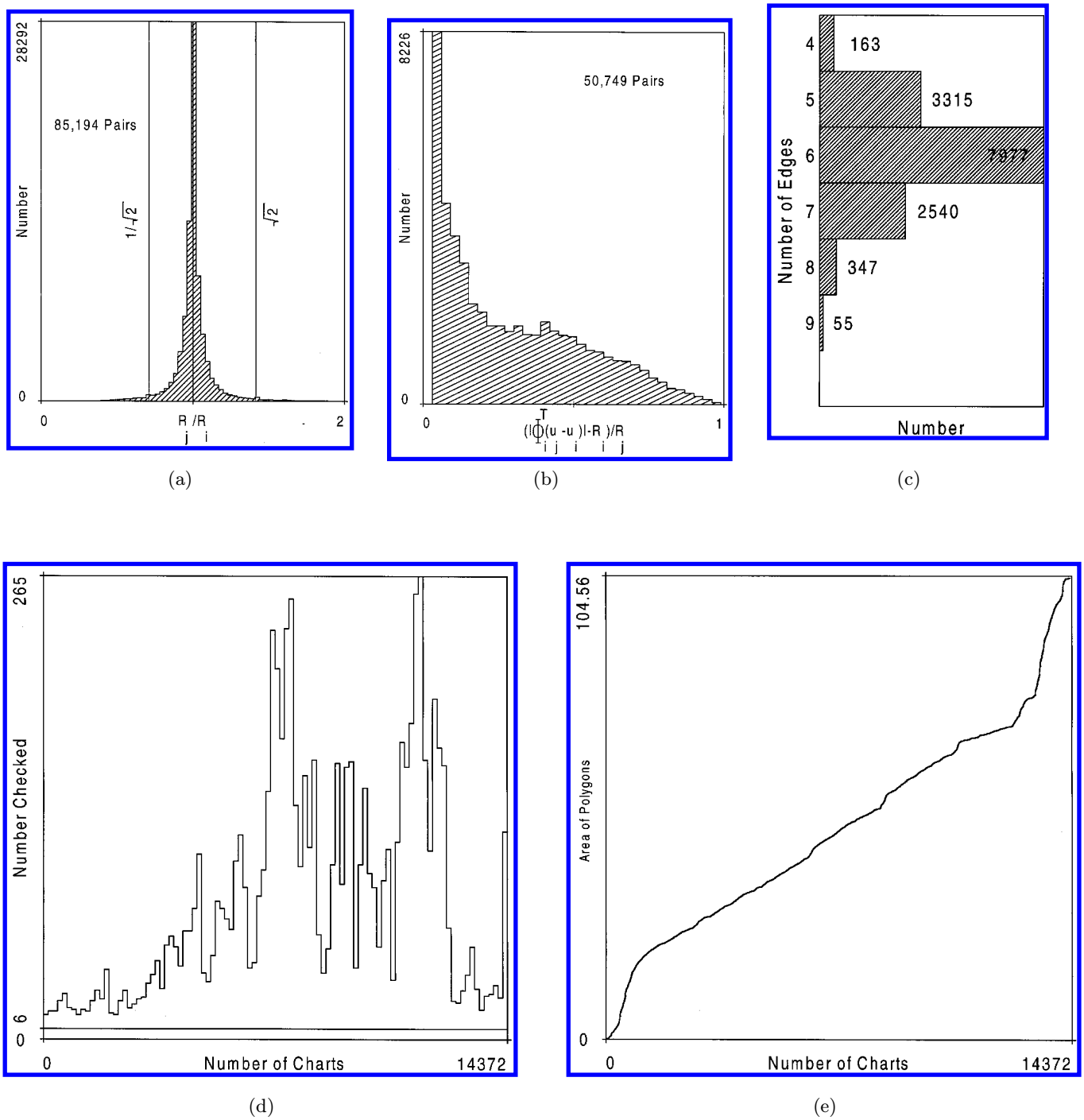


$m = 21$

$m = 5001$

$m = 10001$

$m = 14373$



(a) (b) (c)

(d) (e)

Fig. 15. Performance measures for the periodic motions of the coupled pendula, $\kappa = 0.1$, $\gamma = 0.5$. (a) The distribution of R_j/R_i for $j \in J_i^m$. (b) The distribution of normalized distance between centers. (c) The number of edges in the polygons. (d) The number of charts checked for overlap. (e) $\sum_{i<m} \text{vol}(P_i^m)$, as a function of m.

cells to advance the mesh instead of advancing the boundary of a Delaunay triangulation.

When the radius is constant, and the manifold is flat, the covering we obtain is a Minkowski set of spherical balls (the center of no ball is interior to another) [Tóth, 1999; Betke & Henk, 1998]. Reducing the radius of the balls in a Minkowski set produces a packing (i.e. a nonoverlapping set) of the balls. So the algorithm might also be used to compute packings of spherical balls on manifolds.

Work is underway to incorporate the algorithm described here into AUTO, Doedel's continuation package. It is also being adapted to compute invariant manifolds, and as an advancing front mesh generator.

References

Aho, A. V. & Ullman, J. D. [1992] *Foundations of Computer Science* (Computer Science Press, NY).

Allgower, E. L. & Georg, K. [1980] "Simplicial and continuation methods for approximations, fixed points and solutions to systems of equations," *SIAM Rev.* **22**, 28–85.

Allgower, E. L. & Schmidt, P. H. [1985] "An algorithm for piecewise-linear approximation of an implicitly defined manifold," *SIAM J. Numer. Anal.* **22**(2), 322–346.

Allgower, E. L. & Gnutzmann, S. [1987] "An algorithm for piecewise-linear approximation of implicitly defined two-dimensional surfaces," *SIAM J. Numer. Anal.* **24**(2), 452–469.

Aronson, D. G., Doedel, E. J. & Othmer, H. G. [1991] "The dynamics of coupled current-biased Josephson junctions — Part II," *Int. J. Bifurcation and Chaos* **1**(1), 51–66.

Aurenhammer, F. & Edelsbrunner, H. [1984] "An optimal algorithm for constructing the weighted voronoi diagram in the plane," *Patt. Recogn.* **17**(2), 251–257.

Aurenhammer, F. [1987] "Power diagrams: Properties, algorithms and applications," *SIAM J. Comput.* **16**(1), 78–96.

Aurenhammer, F. [1988] "Improved algorithms for discs and balls using power diagrams," *J. Algorith.* **9**, 151–161.

Aurenhammer, F. [1991] "Voronoi diagrams — a survey of a fundamental geometric data structure," *ACM Comput. Surv.* **23**(3), 345–405.

Betke, U. & Henk, M. [1998] "Finite packings of spheres," *Discret. Comput. Geom.* **19**, 197–227.

Bloomenthal, J. [1988] "Polygonalization of implicit surfaces," *Comput. Aided Geom. Design* **5**(4), 341–355.

Brodzik, M. L. & Rheinboldt, W. C. [1994] "On the computation of simplicial approximations of implicitly defined two-dimensional manifolds," *Comput. Math. Appl.* **28**(9), 9–21.

Brodzik, M. L. [1998] "The computation of simplicial approximations of implicitly defined p-dimensional manifolds," *Comput. Math. Appl.* **36**(6), 93–113.

Brostow, W., Dussault, J.-P. & Fox, B. L. [1978] "Construction of voronoi polyhedra," *J. Comput. Phys.* **29**, 81–92.

Chen, P.-C., Hansen, P. & Jaumard, B. [1991] "On-line and off-line vertex enumeration by adjacency lists," *Operat. Res. Lett.* **10**, 403–409.

Cohen, J. & Hickey, T. [1979] "Two algorithms for determining volumes of convex polyhedra," *J. ACM* **26**(3), 401–414.

Conway, J. H. & Sloan, N. J. A. [1993] *Sphere Packings, Lattices and Groups*, 2nd edition (Springer-Verlag).

Dobkin, D. A., Levy, S. V. F., Thurston, W. P. & Wilks, A. R. [1990] "Contour tracing by piecewise linear approximations," *ACM Trans. Graph.* **9**(4), 389–423.

Doedel, E. [1997] "Nonlinear numerics," *Int. J. Bifurcation and Chaos* **7**(9), 2127–2143.

Edelsbrunner, H. [1995] "The union of balls and its dual shape," *Discret. Comput. Geom.* **13**, 415–440.

Garcia, C. B. & Zangwill, W. I. [1981] *Pathways to Solutions, Fixed Points, and Equilibria* (Prentice-Hall, Englewood Cliffs, NJ).

Gill, P. E., Murray, W. & Wright, M. H. [1981] *Practical Optimization* (Academic Press, NY).

Govaerts, W. J. F. [2000] *Numerical Methods for Bifurcations of Dynamical Equilibria* (SIAM, Philadelphia).

Grunbaum, B. [1967] *Convex Polytopes* (Interscience Publishers, London).

Hall, M. & Warren, J. [1990] "Adaptive polygonalization of implicitly defined surfaces," *IEEE Comput. Graph. Appl.* **10**(6), 33–40.

Henderson, M. E., Levi, M. & Odeh, F. [1991] "The geometry and computation of the dynamics of coupled pendula," *Int. J. Bifurcation and Chaos* **1**(1), 27–50.

Henderson, M. E. [1993] "Computing implicitly defined surfaces: Two parameter continuation," Research Report 18777, IBM Research.

Imai, H., Iri, M. & Murota, K. [1985] "Voronoi diagram in the Laguerre geometry and its applications," *SIAM J. Comput.* **14**(1), 93–105.

Jones, T. A., Hamilton, D. E. & Johnson, C. R. [1986] *Contouring Geologic Surfaces with the Computer* (Van Nostrand Rheinhold).

Keller, H. B. [1977] "Numerical solutions of bifurcation and nonlinear eigenvalue problems," *Applications of Bifurcation Theory*, ed. Rabinowitz, P. (Academic Press, NY), pp. 359–384.

Klein, R. [1987] *Concrete and Abstract Voronoi Diagrams*, Lecture Notes in Computer Science, Vol. 400 (Springer-Verlag, Berlin).

Lorensen, W. & Cline, H. [1987] "Marching cubes: A high resolution 3D surface construction algorithm," *Comput. Graph.* **21**(4), 163–170.

Melhorn, K., Meiser, S. & Ó'Dúnlaing, C. [1990] "On the construction of abstract voronoi diagrams," *STACS 90, 7th Ann. Symp. Theoretical Aspects of Computer Science*, eds. Choffrut, C. & Lengauer, T., Lecture Notes in Computer Science, Vol. 415 (Springer-Verlag), pp. 227–239.

Melville, R. & Mackey, D. S. [1995] "New algorithm for two-dimensional numerical continuation," *Comput. Math. Appl.* **30**(1), 31–46.

Morgan, A. [1987] *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems* (Prentice-Hall, Englewood Cliffs, NJ).

Ning, P. & Bloomenthal, J. [1993] "An evaluation of implicit surface tilers," *IEEE Comput. Graph. Appl.* **13**(6), 33–41.

Rheinboldt, W. C. [1987] "On a moving-frame algorithm and the triangulation of equilibrium manifolds," *ISNM79: Bifurcation: Analysis, Algorithms, Applications*, eds. Kupper, T., Seydel, R. & Troger, H., The University of Dortmund, Basel (Birkhäuser Verlag), pp. 256–267.

Rheinboldt, W. C. [1988] "On the computation of multidimensional solution manifolds of parameterized equations," *Numer. Math.* **53**(1&2), 165–181.

Sabin, M. A. [1985] "Contouring — the state of the art," *Fundamental Algorithms for Computer Graphics*, ed. Earnshaw, R. A. (Springer-Verlag, NY).

Seydel, R. [1997] "Nonlinear computation," *Int. J. Bifurcation and Chaos* **7**(9), 2105–2126.

Tarjan, R. E. [1983] *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Tóth, L. F. [1999] "Minkowski circle packings on the sphere," *Discret. Comput. Geom.* **22**, 161–166.

Watson, D. F. [1992] *Contouring: A Guide to the Analysis and Display of Spatial Data* (Pergamon Press).

Zahlten, C. & Jürgens, H. [1973] "Continuation methods for approximating isovalued complex surfaces," *EUROGRAPHICS '91*, eds. Post, F. & Barth, W. (Elsevier Science Publishers), pp. 5–19.

Zwillinger, D. [1996]. *Standard Mathematical Tables and Formulae*, 30th edition (CRC Press).