1984

I. Circumspheres in Hilbert space, II. Automatic handling of finite-dimensional, nonassociative algebras

Ronald Kenneth Smith
Iowa State University

INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106

Smith, Ronald Kenneth

I. CIRCUMSPHERES IN HILBERT SPACE. II. AUTOMATIC HANDLING OF FINITE-DIMENSIONAL, NONASSOCIATIVE ALGEBRAS

Iowa State University PH.D. 1984

I. Circumspheres in Hilbert space
II. Automatic handling of finite-dimensional, nonassociative algebras

by

Ronald Kenneth Smith

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major: Mathematics

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1984

TABLE OF CONTENTS

GENERAL INTRODUCTION

This work consists of two papers, each dealing with a separate problem. The main connection between them is that both studies began as a search for computer algorithms to solve specific problems with inherent combinatorial difficulties.

The first problem is to find the smallest sphere enclosing an arbitrary bounded set of points in R^n. The search for a practical algorithm led to a general development of the theory of circumspheres in Hilbert spaces as well as a new algorithm in R^n. This is the content of the first paper.

The second problem is to find identities in a finite-dimensional, nonassociative algebra. Here, the search for a practical algorithm led to the development of a method and a computer package for finding and checking multilinear identities in any such algebra. The package was then used to find all the identities of degree six containing two occurrences of each of three generators in the free alternative algebra on three generators. In the process, an alternative algebra of dimension 307 was constructed. The method and the computer package are the subjects of the second paper.

Explanation of Thesis/Dissertation Format

Each of the two papers is self-contained. Both are the sole work of this author. In each paper, the results, including lemmas, theorems, and corollaries are numbered sequentially, beginning with 1. For example, the first three results of the first paper are labeled lemma 1, lemma 2, and theorem 3. Proofs end with a "□" symbol. References in each paper refer only to results within that paper. Definitions, equations, examples, and figures are also numbered sequentially from 1 in each paper for referencing within that paper.

I. CIRCUMSPHERES IN HILBERT SPACE

by

Ronald Kenneth Smith

Graceland College

Lamoni, IA

50140

Introduction

In any metric space, a circumsphere of a bounded set X is defined to be a sphere of smallest radius enclosing X. In this paper, we present several results about circumspheres in complete inner product spaces. We begin with a new proof of the existence and uniqueness of the circumsphere of any nonempty, bounded set in a Hilbert space. Next, we show that if r is the radius of the circumsphere of X and d is its diameter ($d = \sup \|x-y\|$ over all $x,y \in X$), then $r \leq d/\sqrt{2}$. Finally, we give a new characterization of circumspheres in R^n in terms of nonredundant sets. A set X is called nonredundant if each point $x \in X$ lies outside the circumsphere of the set $X-\{x\}$. We show that a nonredundant set X never has more than n+1 points, and that its convex hull contains a unique point which is equidistant from each point of X. The circumsphere of the set X is precisely the sphere centered at this point which contains X on its boundary. Furthermore, the closure of every nonempty, bounded set in R^n contains a nonredundant subset whose circumsphere is also the circumsphere of the original set. This characterization leads directly to some algorithms for finding circumspheres in R^n.

Notation

Throughout this paper, we will be working in an inner product space H with inner product $(\cdot,\cdot)$ and norm $\|\cdot\|$. We will use R for the real numbers, and R^n denotes Euclidean n space.

For any $c \in H$ and $r \geq 0$, we will denote the sphere $\{x: \|x-c\| = r\}$ by $S[c,r]$, and the closed ball $\{x: \|x-c\| \leq r\}$ by $B[c,r]$. When we say that a sphere contains X, we mean $X \subset S[c,r]$. When we say that a sphere encloses X, we mean $X \subset B[c,r]$.

If X is a set, then the closure of X is CL X. The flat generated by X, denoted by FLAT X, is the smallest translate of a subspace

containing X. It consists of all points of the form $\sum_{i=0}^{k} t_i c_i$ where $\sum_{i=0}^{k} t_i = 1$, $t_i \in R$, and $c_i \in X$. The convex hull of X, denoted by CONV X is the smallest convex set containing X. It consists of all points of FLAT X with coefficients $t_i \geq 0$ for all i.

A Little Lemma

Let $C = \{c_0, c_1 \ldots c_n\}$ be a set in an inner product space. The distance $\|x-c\|$ between any point x and an arbitrary point $c \in$ FLAT C is completely determined by the distances $\|x-c_i\|$, $\|c_i-c_j\|$, and the coefficients t_i of c. This is the substance of our first lemma, upon which all the results of this paper rest.

Lemma 1. Let X be an inner product space. Fix $c_0, c_1 \ldots c_n \in X$. Choose any $c = \sum_{i=0}^{n} t_i c_i$ where $\sum_{i=0}^{n} t_i = 1$ and $t_i \in R$. Then for $x \in X$,

$$\|x-c\|^2 = \sum_i t_i \|x-c_i\|^2 - \sum_{i<j} t_i t_j \|c_i-c_j\|^2. \tag{1}$$

In the case of only two points, we write

$$c = c(t) = (1-t)c_0 + tc_1 \quad \text{and}$$

$$\|x-c(t)\|^2 = (t^2-t)\|c_0-c_1\|^2 + (1-t)\|x-c_0\|^2 + t\|x-c_1\|^2. \tag{2}$$

Proof: First, suppose a_{ij} and b_{ij} are numbers with $b_{ij} = a_{ij} + a_{ji}$ for $0 \leq i,j \leq n$. Then $\sum_{i \neq j} a_{ij} = \sum_{i<j} (a_{ij}+a_{ji}) = \sum_{i<j} b_{ij}$. In particular, set $a_{ij} = t_i t_j[\|x-c_i\|^2-(x-c_i, x-c_j)]$ and $b_{ij} = t_i t_j \|c_i-c_j\|^2$. Then,

$$\sum_{i \neq j} t_i t_j[\|x-c_i\|^2 - (x-c_i, x-c_j)] = \sum_{i<j} t_i t_j \|c_i-c_j\|^2.$$

Now $\sum_{i=0}^{n} t_i = 1$ so that $\|x-c\|^2 = (\sum_i t_i(x-c_i), \sum_j t_j(x-c_j))$

$$= \sum_i t_i^2 \|x-c_i\|^2 + \sum_{i \neq j} t_i t_j (x-c_i, x-c_j)$$

$$= \sum_i t_i(1-\sum_{j\neq i} t_j)\|x-c_i\|^2 + \sum_{i\neq j} t_i t_j (x-c_i, x-c_j)$$

$$= \sum_i t_i \|x-c_i\|^2 - \sum_{i\neq j} t_i t_j [\|x-c_i\|^2 - (x-c_i, x-c_j)]$$

$$= \sum_i t_i \|x-c_i\|^2 - \sum_{i<j} t_i t_j \|c_i-c_j\|^2 \text{ as required.} \quad \square$$

We can use lemma 1 to explicitly construct a very useful family of spheres from any two spheres that enclose a common point.

Lemma 2. Let $S[c_0, r_0]$ and $S[c_1, r_1]$ be two spheres such that $B[c_0, r_0] \cap B[c_1, r_1]$ is nonempty. Let $S(t)$ be the sphere $S[c(t), r(t)]$ where $c(t)$ and $r(t)$ are given by

$$c(t) = (1-t)c_0 + tc_1 \text{ and} \tag{3}$$

$$r^2(t) = (t^2-t)\|c_0-c_1\|^2 + (1-t)r_0^2 + tr_1^2. \tag{4}$$

If $S[c_0, r_0] \cap S[c_1, r_1]$ is nonempty, then $r(t)$ is defined and $S(t)$ encloses $S[c_0, r_0] \cap S[c_1, r_1]$ for all $t \in R$. In any case, $r(t)$ is defined and $S(t)$ encloses $B[c_0, r_0] \cap B[c_1, r_1]$ for all $t \in [0,1]$.

Proof: Choose $x \in S[c_0, r_0] \cap S[c_1, r_1]$. Then, $\|x-c_0\| = r_0$, and $\|x-c_1\| = r_1$. Substitution into (2) gives $\|x-c(t)\|^2 = r^2(t)$. It follows that $S(t)$ contains $S[c_0, r_0] \cap S[c_1, r_1]$ for all t. If $x \in B[c_0, r_0] \cap B[c_1, r_1]$ and $t \in [0,1]$, then $(1-t)\|x-c_0\|^2 \leq (1-t)r_0^2$, and $t\|x-c_1\|^2 \leq tr_1^2$. Substitution into (2) gives $\|x-c(t)\|^2 \leq r^2(t)$. It follows that $r(t)$ is defined and that $S(t)$ encloses $B[c_0, r_0] \cap B[c_1, r_1]$ for $t \in [0,1]$. $\square$

Definition 1. The sphere $S(t)$ is said to be in the family generated by $S[c_0, r_0]$ and $S[c_1, r_1]$ if $S(t) = S[c(t), r(t)]$ where $c(t)$ and $r(t)$ are given by (3) and (4). Note that the parameterization has been chosen so that $S(0) = S[c_0, r_0]$ and $S(1) = S[c_1, r_1]$. If

$t \in [0,1]$, we say that $S(t)$ is between $S[c_0,r_0]$ and $S[c_1,r_1]$.

Circumspheres

Now we are ready to prove the existence and uniqueness of circumspheres in Hilbert space. Once we have shown this, we can easily show some important classes of sets which have the same circumsphere as a set X. We conclude this section with a proof that the center of the circumsphere of a set always lies in the closure of the convex hull of that set, and we give a sufficient condition for finding the circumsphere of a set. While none of these results are really new, most authors deal only with finite dimensions, and may only mention that some results hold in general. These are the preliminaries necessary to extend Jung's theorem into Hilbert space in the next section.

Theorem 3. Let X be a nonempty bounded subset of a complete inner product space H. There exists a unique sphere $S[c(X),r(X)]$ with minimal radius enclosing X.

Proof: Let $r(X) = \inf\{r \in R: S[c,r] \text{ encloses } X\}$. Choose any sequence of spheres $S[c_i,r_i]$ enclosing X such that the sequence of radii $\{r_i\}$ converges to $r(X)$. For any $\epsilon > 0$, choose N so that $r_n^2 < r^2(X) + \epsilon^2/4$ whenever $n > N$. If $n,m > N$, the sphere $S(1/2)$ between $S[c_n,r_n]$ and $S[c_m,r_m]$ encloses X, so that $r^2(X) \leq r^2(1/2) = -1/4\|c_n-c_m\|^2 + 1/2(r_n^2 + r_m^2) \leq -1/4\|c_n-c_m\|^2 + r^2(X) + \epsilon^2/4$. It follows that $\|c_n-c_m\| \leq \epsilon$, so the sequence of centers, $\{c_i\}$ is a Cauchy sequence. Let $c(X)$ be the limit of this sequence. To see that X is enclosed by $S[c(X),r(X)]$, take n large enough so that $r_n < r(X) + \epsilon/2$ and $\|c_n-c(X)\| < \epsilon/2$. Then, for any $x \in X$, $\|x-c(X)\| \leq \|x-c_n\| + \|c_n-c(x)\| < r(X) + \epsilon$. Since this holds for every $\epsilon > 0$, $\|x-c(X)\| \leq r(X)$. Since this is true for every $x \in X$, $S[c(X),r(X)]$ encloses X. For uniqueness, suppose X is enclosed by $S[c,r(x)]$ as

well. Then, X must also be enclosed by S(1/2) between them (lemma 2). But then, $r^2(X) \leq r^2(1/2) = -1/4 \|c-c(X)\|^{-2} + r^2(X)$, so $c = c(X)$. □

Definition 2. Let X be a nonempty, bounded set in a Hilbert space. Denote the circumsphere of X by S(X). Then $S(X) = S[c(X),r(X)]$. c(X) and r(X) are called the circumcenter and circumradius of X, respectively.

Definition 3. If $S(Y) = S(X)$, then we say that the set Y supports X. In particular, if Y is contained in X, then Y is a supporting subset of X.

Corollary 4. Let Y be a subset of X. The following are equivalent:

i. X is enclosed by S(Y)

ii. $r(X) \leq r(Y)$

iii. Y is a supporting subset of X.

Proof: (i => ii) If X is enclosed by S(Y), and r(X) is the smallest radius of any sphere enclosing X, then $r(X) \leq r(Y)$. (ii => iii) Since $Y \subset X$, Y is enclosed by S(X). So, $r(Y) \leq r(X)$. If also $r(X) \leq r(Y)$ then $r(X) = r(Y)$. But then S(X) is a circumsphere of Y. By uniqueness, $S(Y) = S(X)$. (iii => i) If $S(Y) = S(X)$ it obviously encloses X. □

Using this corollary, a number of sets related to X can be shown to support one another. For example, since the ball $B[c(X),r(X)]$ is a closed set containing X, it must also contain CL X. But then X is a subset of CL X, and CL X is enclosed by S(X). Hence $S(X) = S(CL\ X)$ by (i => iii). Similarly, since balls are also convex, it follows that $S(X) = S(CONV\ X)$. Combining these also gives $S(X) = S(CL\ CONV\ X) = S(CONV\ CL\ X)$.

An important family of supporting subsets of X is given by those subsets which lie close to the surface of S(X). This is the substance of the next lemma.

Lemma 5. Let X be a nonempty, bounded set. Let $\epsilon > 0$. Let $X_\epsilon = \{x \in X: \|x-c(X)\| \geq r(X) - \epsilon\}$. Then, X_ϵ supports X.

Proof: First, note that X_ϵ is never empty when $\epsilon > 0$, since otherwise $S[c(X),r(X)-\epsilon]$ would enclose X. Thus, $S(X_\epsilon)$ exists. Suppose $c(X) = c(X_\epsilon)$, and let $m = \max\{r(X_\epsilon), r(X) - \epsilon\}$. Then, X is enclosed in $S[c(X),m]$. This gives $r(X) \leq m$ forcing $m = r(X_\epsilon)$. Thus, $r(X) \leq r(X_\epsilon)$, and we are done by corollary 4 (ii => iii). Now suppose that $c(X) \neq c(X_\epsilon)$. Let S(t) be between S(X) and $S(X_\epsilon)$ for each $t \in [0,1]$. We claim that r(t) is a continuous, strictly decreasing function on [0,1]. Since both S(X) and $S(X_\epsilon)$ enclose X_ϵ, so does S(t) for each $t \in [0,1]$ (lemma 2). Since $r(1) = r(X_\epsilon)$ is the smallest radius of any sphere containing X_ϵ, $r^2(t)$ is minimized on this interval at 1. Also, from the form of $r^2(t)$, it is evident that it is an arc of an upward-opening parabola on this interval. It follows that $r^2(t)$ is a continuous, strictly decreasing function here. Hence, so is r(t). Now choose t with $0 < t < \epsilon/[2\|c(X)-c(X_\epsilon)\|]$ with $r(t) > r(X) - \epsilon/2$. Pick any y outside S(t). y cannot be in X_ϵ since S(t) encloses X_ϵ. On the other hand, note that $\|c(t)-c(X)\| = t\|c(X)-c(X_\epsilon)\| < \epsilon/2$. So, $\|y-c(X)\| \geq \|y-c(t)\| - \|c(t)-c(X_\epsilon)\| > r(t) - \epsilon/2 \geq r(X) - \epsilon$. Hence, y cannot be in X, and X is enclosed by S(t). This is impossible, since $r(t) < r(0) = r(X)$. This contradiction shows that c(X) and $c(X_\epsilon)$ must coincide, and we are done. □

Lemma 6. c(X) always belongs to CL CONV X.

Proof: The proof rests on two facts whose proofs we shall defer for a moment. If X is a closed, convex set and y is any point, then there exists a unique point $y' \in X$, called the projection of y on X, which is closer to y than any other point of X. In fact, $\|x-y\| \geq \|x-y'\|$ for every $x \in X$. Assuming these facts, $r(X) \geq \|x-c(X)\| \geq \|x-c(X)'\|$ for every $x \in X$, so X is enclosed by $S[c(X)',r(X)]$. By uniqueness of

circumspheres, $c(X) = c(X)' \in X$. Finally, from the remarks following corollary 4 we have for any set X that $S(X) = S(CL\ CONV\ X)$. Since the closure of a convex set is still convex as well as closed, we are done.

Now we shall proceed to prove our claims about a point and its projection onto a set X which is closed and convex. Fix any y. Set $r = \inf\{\|x-y\|: x \in X\}$. Pick any sequence of spheres $S[c_i,r_i]$ each of which contains y and is centered in X, and so that the sequence of radii converges to r. Since X is convex, the sphere S(1/2) between any two spheres $S[c_m,r_m]$ and $S[c_n,r_n]$ is still centered in X, and it must also contain y, so that $r^2 \leq r^2(1/2) = -1/4\|c_m-c_n\|^2 + 1/2(r_m^2 + r_n^2)$. As in the proof of theorem 3, it follows that the sequence of centers, $\{c_i\}$ must be a Cauchy sequence, and that its limit, which we shall call y', is the unique point in CL X with $\|y-y'\| = r$.

To see that y' is actually closer to each $x \in X$ than y is, fix $x \in X$ and define $f(t) = \|y-c(t)\|^2$ where $c(t) = (1-t)y' + tx$. We have just shown that $\|y-y'\| \leq \|y-c(t)\|$ for $t \in [0,1]$ since $c(t) \in CL\ CONV\ X$ for each such t. Since $c(0) = y'$, we see that f(t) is minimized on [0,1] at 0. From equation 2, we see that $f(t) = (t^2 - t)\|y'-x\|^2 + (1-t)\|y-y'\|^2 + t\|y-x\|^2$. Since this is a parabola whose minimum on [0,1] occurs at 0, its derivative there cannot be negative. We must have $0 \leq f'(0) = -\|y'-x\|^2 - \|y-y'\|^2 + \|y-x\|^2$. Consequently, $\|y-x\| \geq \|y'-x\|$ as required. □

Next, we present a sufficient condition for finding circumspheres. It is far from necessary in general, but we shall see that it almost has a converse in R^n (see theorem 11).

Lemma 7. If there exists a point $c \in CL\ CONV\ X$ and an $r \in R$ such that $\|x-c\| = r$ for all $x \in X$, then $S(X) = S[c,r]$.

Proof: First, we shall show that if $c \in CONV\ X$ and z is any other point, then there is some $x \in X$ closer to c than to z [3, lemma 2].

Let $c = \sum_{i=0}^{n} t_i x_i$ where $\sum_{i=0}^{n} t_i = 1$, $t_i \geq 0$, and $x_i \in X$ for $i = 1,2...n$. Suppose $\|z-x\| \leq \|c-x\|$ for all $x \in X$. By lemma 1,

$$\|z-c\|^2 = \|z-\sum_i t_i x_i\|^2 = \sum_i t_i \|z-x_i\|^2 - \sum_{i<j} t_i t_j \|x_i-x_j\|^2 \leq$$

$$\sum_i t_i \|c-x_i\|^2 - \sum_{i<j} t_i t_j \|x_i-x_j\|^2 = \|c-\sum_i t_i x_i\|^2 = 0.$$

Consequently, either $c = z$ or $\|c-x\| < \|z-x\|$ for some $x \in X$. Now suppose that $c \in \text{CONV } X$ is equidistant from each $x \in X$. Then, $r(X) \leq r$ since X is enclosed by $S[c,r]$. If $c(X) \neq c$, then for some $x \in X$, $r = \|x-c\| < \|x-c(X)\| \leq r(X)$. This is impossible, so $c(X) = c$. Clearly, $r(X) = r$ and $S(X) = S[c,r]$.

If c is a limit point of CONV X but is not in CONV X, then there is a sequence $\{c_i\}$ in CONV X which converges to c but $c_i \neq c(X)$ for any i. Using the first result, this time with c_i for c and $c(X)$ for z, we see that for each i, there is some $x_i \in X$ with $\|c_i-x_i\| < \|c(X)-x_i\|$. Now $|\|c_i-x_i\| - r| = |\|c_i-x_i\| - \|x_i-c\|| \leq \|c_i-c\| \to 0$. Therefore, $r = \lim_i \|c_i-x_i\| \leq \lim_i \inf \|c(X)-x_i\| \leq r(X)$. But $r(X) \leq r$ since X is enclosed in $S[c,r]$. It follows that $r = r(X)$. From the uniqueness of circumspheres, $S(X) = S[c,r]$. □

Jung's Theorem in Hilbert Space

In this section, we give the best bound on the circumradius that can be given when the diameter of the set is known. The analogous bound in R^n is known as Jung's theorem. We will derive it in the next section, where we concentrate on results in finite dimensions. We end this section with an example to show how badly behaved circumspheres can be in general.

Theorem 8. Let $d = \text{diam}(X)$, $r = r(X)$. Then $r \leq d/\sqrt{2}$, and this bound is the best possible.

Proof: First note that when $\sum_{i=1}^{m} t_i = 1$, it is true that $\sum_{i<j} t_i t_j = (\sum_{i,j} t_i t_j - \sum_i t_i^2)/2 = (1 - \sum_i t_i^2)/2 < 1/2$. Now let $\epsilon > 0$ be given. Let $Y = \{x \in X: \|x-c(X)\|^2 \geq r^2 - \epsilon/2\}$. According to lemma 5, any subset of X consisting of all points outside some radius less than r(X) supports X. Certainly Y is such a set, so S(Y) = S(X). Consequently, $c(X) \in$ CL CONV Y by lemma 6. Choose points $x_0, x_1 \ldots x_m \in Y$ with $\|c(X) - \sum_i t_i x_i\|^2 < \epsilon/2$ where $\sum_{i=0}^{m} t_i = 1$, $t_i > 0$ for $i = 0,1\ldots m$. By equation 1, $\sum_i t_i \|c(X)-x_i\|^2 = \sum_{i<j} t_i t_j \|x_i-x_j\|^2 + \|c(X)-\sum_i t_i x_i\|^2$. But then $(\sum_i t_i)(r^2 - \epsilon/2) < (\sum_{i<j} t_i t_j)\, d^2 + \epsilon/2$ so $r^2 < d^2/2 + \epsilon$ for every $\epsilon > 0$. Hence, $r \leq d/\sqrt{2}$. Equality is achieved in example 1. □

Example 1. Consider l^2, the space of square summable sequences. Let $x_1 = (1/2,0,0\ldots)$, $x_2 = (0,2/3,0\ldots)$ . . .

$x_n = (0\ldots0,n/(n+1),0\ldots)$ . . .

Let $X = \{x_i\}$. Clearly, X is enclosed by the unit sphere S[0,1], so $r(X) \leq 1$. On the other hand, if $c(X) = (\alpha_1,\alpha_2,\alpha_3,\ldots)$ then $\alpha_n \to 0$ as n gets large since $\|c(X)\|$ is finite. Thus, $r(X) \geq \sup\|x_n-c(X)\| \geq \lim|n/(n+1) - \alpha_n| = 1$. Hence, r(X) = 1, and by uniqueness of circumspheres, S(X) = S[0,1]. Note also that $\operatorname{diam}(X) = \sup\|x_i-x_j\| = \sup(i^2/(i+1)^2 + j^2/(j+1)^2)^{1/2} = \sqrt{2}$.

Note that no $x \in X$ lies on S(X) even though X is closed and bounded, and that c(X) does not lie in CONV X. Also $\|x-y\| < \operatorname{diam}(X)$ for all $x,y \in X$. Nevertheless, $r(X) = \operatorname{diam}(X)/\sqrt{2}$.

A sufficient condition for equality to occur in theorem 8 is that X contains a sequence $\{x_i\}$ with the property that for any $\epsilon > 0$, there is an N such that $\|x_i-x_j\|^2 > d^2 - \epsilon$ whenever $N < i < j$. Choose

any $m \geq 2$, and let $t_i = 1/m$ for all i. From equation (1), we have that

$$r^2(X) \geq \sum_{i=N+1}^{N+m} \|x_i - c(X)\|^2/m = \|c(X) - \sum_{i=N+1}^{N+m} x_i/m\|^2 + \sum_{\substack{N<i<j \\ \leq N+m}} \|x_i - x_j\|^2/m^2 >$$

$(d^2 - \epsilon)(m-1)/2m$. Since m is arbitrary, $r^2(X) \geq (d^2 - \epsilon)/2$. Since ϵ is arbitrary, $r^2(X) \geq d^2/2$ and by the first part of this proof, equality occurs. As a bonus, we see that in this case, $c(X) = \lim_{N,m} \sum_{i=N+1}^{N+m} x_i/m$.

Circumspheres in R^n

As demonstrated in example 1, even closed and bounded sets in a general Hilbert space can exhibit "bad" behavior with regard to their circumspheres. The situation is much nicer in R^n, however. The main advantage of working in R^n is that we can always work with compact sets. In this section, we show how some of the previous results can be improved upon by assuming compactness or by taking the space to be R^n. Then, we sketch the most commonly found results about circumspheres in R^n.

Lemma 9. (See lemma 5) Let X be a nonempty compact subset of a Hilbert space. Let $X_0 = \{x \in X: \|x - c(X)\| = r(X)\}$. Then, $S(X) = S(X_0)$.

Proof: Because the distance from $c(X)$ is a continuous function defined on X which is compact, the maximum $\|x - c(X)\|$ must be attained at some $x \in X$. But $r(X) = \max_{x \in X} \|x - c(X)\|$, so X_0 is not empty. Suppose $r(X) > r(X_0) + \epsilon$ for some $\epsilon > 0$. Let $Y = \{y \in X: \|y - x\| \leq \epsilon$ for some $x \in X_0\}$. Then, $r(Y) \leq r(X_0) + \epsilon < r(X)$. Consider $S(t)$ between $S(X)$ and $S(Y)$ (definition 1). Then, $r(t) < r(X)$ for $t \in (0,1)$, so $S(t)$ cannot enclose X. Pick $x_n \in X$ but outside $S(1/n)$ for each n. Some subsequence $\{x_{n_i}\}$ converges to $x_0 \in X$ since X is compact. But where is x_0? On the one hand, x_0 cannot be in X_0 since Y is enclosed by $S(1/n)$ for each n,

forcing $\|x_n - x\| > \epsilon$ for any $x \in X_0$. On the other hand, $r(X) \geq \|x_0 - c(X)\| = \lim_i \|x_{n_i} - c(X)\|$ since $x_{n_i} \to x_0 \in X$. For any i, $\|x_{n_i} - c(X)\| \geq \|x_{n_i} - c(1/n_i)\| - \|c(1/n_i) - c(X)\| \geq r(1/n_i) - 1/n_i \|c(X) - c(Y)\|$ since x_{n_i} is outside $S(1/n_i)$ and the second terms are equal. Since r(t) is a continuous function, the limit of this last expression is $r(0) = r(X)$. This forces $\|x_0 - c(X)\| = r(X)$, so that $x_0 \in X_0$. This contradiction shows that $r(X) \leq r(X_0) + \epsilon$ for every $\epsilon > 0$. Hence, $r(X) = r(X_0)$, and by corollary 4, X_0 supports X. □

Lemma 10. (See lemma 6) If X is a nonempty compact subset of R^n, then $c(X) \in$ CONV X.

Proof: We use the fact that the convex hull of a compact set in R^n is compact [4, 3.1] and therefore closed. From lemma 6, $c(X) \in$ CL CONV X = CONV X. □

Theorem 11. (See lemma 7) If X is a nonempty bounded subset of R^n, then CL X contains a subset Y with n+1 or fewer points which supports X (i.e. S(Y) = S(X)), and there is a unique point in CONV Y which is equidistant from each $y \in Y$. This point is c(X).

Proof: Let $X_0 = \{x \in \text{CL } X: \|x - c(X)\| = r(X)\}$. X_0 is a closed subset of the compact set CL X, so X_0 is also compact. By lemma 10, $c(X_0) \in$ CONV X_0. From lemma 9, $S(X_0) = S(\text{CL } X) = S(X)$, so that $c(X) \in$ CONV X_0. By a theorem of Caratheodory [4, p. 103], any point in the convex hull of a set in R^n can be written as a convex combination of n+1 or fewer points of the set. Thus, c(X) is in the convex hull of a set $Y \subset X_0$ which has n+1 or fewer points. But c(X) is equidistant from each point in X_0, and therefore in Y. It follows from lemma 7 that $S(Y) = S(X_0) = S(X)$. □

Theorem 12. (Jung [8], see theorem 8) Let $d = \text{diam}(X)$ and $r = r(X)$. Then, $r \leq d\,[n/(2(n+1))]^{1/2}$ with equality iff CL X contains a regular n-simplex of edge length d.

Proof: From theorem 11, we may choose $x_0, x_1 \ldots x_n \in$ CL X (not necessarily distinct) with $\|x_i - c(X)\| = r(X)$ for $i = 0,1\ldots n$, and $c(X) = \sum_i t_i x_i$ where $t_i \geq 0$ and $\sum_i t_i = 1$. Note that $\sum_{i<j} t_i t_j \leq n/(2(n+1))$ with equality iff $t_i = 1/(n+1)$ for $i = 0,1\ldots n$. This is easily verified using Lagrange multipliers, for example. From equation 1, $0 = \|c(X) - \sum_i t_i x_i\|^2 = r^2(X) - \sum_{\substack{0 \leq i < \\ j \leq n}} t_i t_j \|x_i - x_j\|^2$. It follows that $r^2 \leq d^2 n/(2(n+1))$ with equality iff $\|x_i - x_j\| = d$ for $0 \leq i < j \leq n$, and $t_i = 1/(n+1)$ for $i = 0,1\ldots n$. □

Nonredundant Sets and Circumspheres

Definition 4. If some $x \in X$ is enclosed by $S(X-\{x\})$, then x is said to be redundant in X; otherwise, x is nonredundant in X. The set X is said to be redundant if any of its points is redundant, and nonredundant otherwise. In particular, X is nonredundant iff x lies outside $S(X-\{x\})$ for every $x \in X$.

Theorem 13.

a. The following are equivalent:
   i. x is redundant in X
   ii. $X-\{x\}$ is a supporting subset of X
   iii. $r(X-\{x\}) = r(X)$

b. If x is nonredundant in X, then
   i. x lies on S(X)
   ii. x belongs to every supporting subset of X.

Proof: Let X' denote $X-\{x\}$. It is easy to see that x is enclosed by S(X') iff X is enclosed by S(X') iff $S(X) = S(X')$ iff $r(X') = r(X)$, where the last two equivalences come from corollary 4, and (a) is done.

Suppose x lies outside S(X') but inside (not on) some sphere S[c,r] which encloses X. We claim there is a sphere of the form S(t) between S(X') and S[c,r] which contains x and is smaller than S[c,r]. Since it also encloses X' it must enclose all of X. To find this sphere just set $\|x-c(t)\|^2 = r^2(t)$ and solve for t. Use (2) and (4) and cancel $(t^2-t)\|c(X')-c\|^2$. The resulting linear equation has the solution $t = [1 + (r^2-\|x-c\|^2)/(\|x-c(X')\|^2-r^2(X'))]^{-1}$. Check that $0<t<1$. Since S(X) encloses X and no smaller sphere does, x must lie on S(X). For the second part of (b), let Y be any supporting subset of X, and suppose that x is not in Y. Then, $Y \subset X-\{x\} \subset X$, so $r(Y) \leq r(X-\{x\}) \leq r(X)$. But r(Y) = r(X) by hypothesis, forcing $r(X-\{x\}) = r(X)$. This contradicts part (a). Therefore, $x \in Y$. □

Lemma 14. Any finite set has a nonredundant supporting subset.
Proof: Any set X supports itself. If X is finite, it must have a minimal (with respect to intersection) supporting subset. Call it Y. If x is redundant in Y, then $S(Y-\{x\}) = S(Y) = S(X)$ contradicting the minimality of Y. Therefore, Y is nonredundant. □

Lemma 15. A finite, nonredundant set in R^n has at most n+1 points.
Short proof: Let X be a finite, nonredundant set. Let Y be a supporting subset of X with no more than n+1 points. Since every point in X is nonredundant in X, it must be contained in Y (theorem 13 b). Therefore, X is contained in Y and we are done. □

There is another proof which is longer, but gives more insight into the nature of nonredundant sets, and does not depend on compactness.
Alternate proof: Suppose $X_0 \subsetneq X_1 \subsetneq \ldots \subsetneq X_m$ with X_i nonredundant for each i. We will show that dim H ≥ m. Let $S(X_i) = S[c_i,r_i]$. For $i < j$ we have $X_i \subset X_j \subset S(X_j)$. So $X_i \subset S(X_i) \cap S(X_j) \subset S(t)$ generated by $S(X_i)$ and $S(X_j)$ for all $t \in R$ (theorem 13 b, lemma 2). So, $r(t) \geq r(X_i)$

$= r(0)$. Since $r(0)$ is the smallest possible radius of any sphere containing X_i, $r^2(t)$ is minimized at 0 for all t. Hence, the derivative of $r^2(t)$ at 0 must be 0. That is, $0 = -\|c_i-c_j\|^2 - r_i^2 + r_j^2$. For any $x \in X_i$, $r_i = \|x-c_i\|$. Consequently, $\|x-c_j\|^2 = \|x-c_i\|^2 + \|c_i-c_j\|^2$ for all $x \in X_i$. It follows that $(x-c_i, c_i-c_j) = 0$ for all $x \in X_i$ and $i < j$. For $x \in X_0$, $(c_i-c_{i-1}, c_j-c_{j-1}) = -(x-c_i, c_j-c_{j-1}) + (x-c_{i-1}, c_j-c_{j-1})$ $= (x-c_i, c_i-c_j) - (x-c_i, c_i-c_{j-1}) - (x-c_{i-1}, c_{i-1}-c_j) + (x-c_{i-1}, c_{i-1}-c_{j-1})$ $= 0$. Thus, the set $\{(c_i-c_{i-1})\}_{i=1}^m$ is an orthogonal set, so dim $H \geq m$.

To finish the proof, we will show that for any nonredundant set X with m elements, there is a nested chain $X_1 \subset X_2 \subset \ldots \subset X_m = X$ of nonredundant sets with $|X_i| = i$. To show this, pick any $x_0 \in X$ which maximizes $r(X-\{x\})$ for $x \in X$. Let $X' = X - \{x_0\}$. We claim that X' is nonredundant. Suppose y is redundant in X'. Then $S(X-\{x_0,y\}) = S(X')$. Also, $r(X-\{x_0,y\}) \leq r(X-\{y\}) \leq r(X') = r(X-\{x_0,y\})$ where the second inequality comes from the maximality of $r(X')$. Since $r(X-\{y\}) = r(X-\{x_0,y\})$, the uniqueness of circumspheres gives $S(X-\{y\}) = S(X-\{x_0,y\}) = S(X')$. But x_0 is enclosed by $S(X-\{y\}) = S(X')$, contradicting the nonredundance of X. Therefore, X' is nonredundant.

Thus, given a nonredundant set X_i with i elements, we can construct X_{i-1} by removing any x_i that maximizes $r(X_i-\{x\})$. This can be continued until we end up with a singleton set X_1. The existence of this chain implies that the original set X_i is contained in a space with dimension at least i-1. It follows that in R^n, the largest finite nonredundant set has at most n+1 points. □

Theorem 16. Let X be a bounded, nonempty set in R^n. Then CL X contains a nonredundant, supporting subset Y with n+1 or fewer points. Short proof: Let Z be any supporting subset of CL X that has no

more than n+1 points (theorem 11). Z has a nonredundant supporting subset Y. So $S(Y) = S(Z) = S(X)$, and $|Y| \leq |Z| \leq n+1$. □

There is a longer proof that depends only on lemma 14 to get the number n+1. However, it gives more enlightenment about compactness than about nonredundance. The idea is to use compactness to approximate X with finite subsets X_i. For each i, use lemma 14 to get a nonredundant subset $Y_i \subset X_i$ with no more than n+1 points. Then, a subsequence Y_{i_j} of these nonredundant sets can be chosen so that there are no more than n+1 limit points in $\bigcup_j Y_{i_j}$. This set of limit points can be shown to support X. It must also contain a nonredundant, supporting subset Y. This last subset is the one required by the theorem.

Algorithms

Identifying the circumsphere S(X) of an arbitrary bounded set X in a Hilbert space may be difficult since its center, c(X), need not lie in CONV CL X, nor does S(X) necessarily contain any points of CL X (example 1). If X should contain a subset Y whose circumsphere S(Y) can be found (e.g. if Y is nonredundant) and S(Y) encloses X, then S(X) = S(Y) (corollary 4). In R^n, CL X always contains a nonredundant subset Y (with no more than n+1 points) such that S(Y) = S(X) (theorem 16). S(Y) is then easy to find, since c(Y) is the unique point in CONV Y which is equidistant from each $y \in Y$ (theorem 11). Thus, the problem of finding S(X) is reduced to identifying nonredundant subsets of CL X, finding their circumspheres, and checking to see whether or not they enclose X. This approach leads directly to a simple, recursive algorithm for finding circumspheres of finite sets in R^n.

We will assume that the following functions and procedures are available. MIN(a,b) returns the minimum value of a and b. CHOOSE(X) returns any point of the set X. CIRCUMSCRIBE(X,c,r) accepts a nonredundant set X and returns c and r so that S[c,r] = S(X). The variables X, NR, and Y refer to sets, x and c are points, and r is a real number.

In the following algorithm, X is passed by value, while NR, c, and r are passed as variables.

```
Algorithm CIRCUMSPHERE0(X,NR,c,r);
    IF |X| ≤ 2 THEN NR ← X; CIRCUMSCRIBE(NR,c,r);
    ELSE begin
        M ← MIN(|X|,n+1); Y ← {};
        REPEAT
            x ← CHOOSE(X-Y);
            CIRCUMSPHERE0(X-{x},NR,c,r);
            IF "x-c" > r THEN Y ← Y ∪ {x}
        UNTIL "x-c" ≤ r OR |Y| = M;
        IF |Y| = M THEN NR ← Y; CIRCUMSCRIBE(NR,c,r);
    end;
```

We claim that CIRCUMSPHERE0(X,NR,c,r) terminates with NR equal to a nonredundant, supporting subset of X, and with S[c,r] = S(NR) = S(X) for any nonempty finite set X.

Proof: We proceed by induction on |X|. If |X| = 1 or 2, then X itself is nonredundant, and the algorithm terminates correctly after executing only the first line. Suppose the algorithm terminates correctly whenever |X| = m ≥ 2, and let |X| = m+1. For any x ∈ X, CIRCUMSPHERE0(X-{x},NR,c,r) terminates with NR a nonredundant subset of X-{x}, and S[c,r] = S(NR) = S(X-{x}). If "x-c" ≤ r, then x is redundant in X and S(X) = S(X-{x}) (theorem 13 a). The algorithm terminates with no further changes to NR, c, or r. In case "x-c" > r, x is nonredundant in X. It is accumulated in the set Y, the set of all points known to be nonredundant in X. Let Z be any nonredundant, supporting subset of X. Then Y ⊂ Z (theorem 13 b). Now M = MIN(|X|,n+1) is the largest possible size of any nonredundant subset of X (lemma 14), so |Z| ≤ M. It follows that if |Y| = M, then Y = Z. That is, Y is a nonredundant, supporting subset of X, and the algorithm terminates correctly with NR = Y. □

Because the algorithm above is recursive, it can take lots of time if the number of points in X is large. In the following algorithm,

this difficulty is overcome for small n by guaranteeing that CIRCUMSPHERE0 is never used on a set with more than n+1 points. The variables X, X', NR, and NR' are sets, c and c' are points, and r and r' are real numbers. All the parameters are passed as variables.

```
Algorithm CIRCUMSPHERE1(X,NR,c,r);
   IF |X| ≤ n+1 THEN CIRCUMSPHERE0(X',NR',c',r')
   ELSE begin
      r ← 0;
      FOR each X' ⊂ X with |X'| = n+1 do
         CIRCUMSPHERE0(X',NR',c',r');
         IF r < r' THEN NR ← NR'; c ← c'; r ← r';
   end;
```

We claim that given any finite, nonempty set X, that CIRCUMSPHERE1 terminates with NR a nonredundant, supporting subset of X, and S[c,r] = S(NR) = S(X).

Proof: It is easy to see that $r(X) = \max_{Y \subset X} r(Y) = \max_{\substack{Y \subset X \\ |Y|=n+1}} r(Y)$, since X must contain a supporting subset with no more than n+1 points. Thus, whenever CIRCUMSPHERE0 returns an r' which maximizes r(X') for all X' ⊂ X with |X'| = n+1, it must be that NR' is a nonredundant, supporting subset of X', and so S[c',r'] = S(NR') = S(X). □

It is easy to find the order of this algorithm. If m = |X|, then there are m choose n+1 sets to search through. Since CIRCUMSPHERE0 only operates on sets with n+1 points, it is independent of m. In terms of m, then, the execution time of this algorithm is $O(m^{n+1})$.

It would be nice to be able to determine whether or not a nonredundant subset of X is also a supporting subset of X when it is found rather than continuing the search through all the subsets to find larger circumradii. Doing this requires checking to see whether or not X is enclosed by each circumsphere with a larger radius. While such schemes may be faster in some cases, they may also have larger worst case orders. Our final algorithm is one in which we trade some possible worst case order for very nice behavior in many respects.

Here CIRCUMSPHERE0 is never used on a set with more than n+2 points. For this algorithm, we need the following Boolean function: FINDOUT(X,x,c,r) cycles through the points of the set X, beginning with x, until a point is found which lies outside S[c,r]. If found, 'true' is returned, and x is the point outside S[c,r]. Otherwise, 'false' is returned. The variables X and NR are sets, c and x are points, and r is real. In addition, FOUND is boolean. All parameters are passed as variables.

```
Algorithm CIRCUMSPHERE2(X,NR,c,r);
   REPEAT
      FOUND ← FINDOUT(X,x,c,r);
      IF FOUND THEN CIRCUMSPHERE0(NR ∪ {x},NR,c,r);
   UNTIL NOT FOUND;
```

We claim that if CIRCUMSPHERE2 is called with NR a nonredundant subset of X, and S(NR) = S[c,r], then it will terminate with NR being a nonredundant, supporting subset of X with S(NR) = S[c,r] = S(X). Proof: Suppose that NR is indeed a nonredundant subset of X with S(NR) = S[c,r]. If no x lies outside S[c,r], then S[c,r] = S(NR) = S(X) (Corollary 4). In this case, FOUND will be false, and the algorithm terminates correctly. Otherwise, a point x lying outside S(NR) will be found, and NR is replaced by a nonredundant, supporting subset of NR ∪ {x} via CIRCUMSPHERE0. Now r(NR ∪ {x}) > r(NR) since x lies outside S(NR). Thus, the old NR can never be tried again. Since the number of nonredundant subsets is finite, the algorithm terminates. □

It is difficult to pin down the execution time of this algorithm precisely. Surely it is bounded by the number of possible nonredundant subsets of X times the number of points in X. If |X| = m, this is $O(m^{n+2})$. On the other hand, the algorithm may only cycle through the points of X once. This occurs when the first points found happen to form a nested sequence of nonredundant supporting sets leading up to a supporting subset of X. Such a sequence exists in any set (theorem 15). In this case, the rest of the points of X simply have to be checked to find out that they lie within S(X).

It may happen that a point x is nonredundant in every subset of X that contains it. Such a point obviously belongs to every nonredundant, supporting subset of X. It is also easy to verify that once such a point is tried, it must remain in NR from then on. Thus, all such points must belong to NR after the first cycle through X.

We have seen two ways in which CIRCUMSPHERE2 behaves very nicely. In practice, it seems to converge very rapidly.

Notes

The problem of constructing circumspheres in n space is over one hundred years old. In 1857, Sylvester proposed the problem in the plane [10]. He gave a solution due to Pierce [11]. Chrystal gave a similar solution in 1885, and indicated how one might construct the circumsphere of four points in space [2].

Chrystal's paper actually contains two different geometric algorithms for constructing the circumsphere of a finite set in the plane. His first algorithm is motivated by the idea of "continuously diminishing the radius" of a circle which encloses the set while keeping two points on its boundary fixed--a geometric notion corresponding to our family of spheres between two spheres. The circle is diminished until it contains the vertices of an acute-angled triangle or has the two fixed points on a diameter--corresponding to our notion of nonredundance in the plane. While his algorithm actually has order $O(m^3)$, with a little modification it can be made to have order $O(m^2)$. Unfortunately, the modification does not extend to 3 space. Chrystal's second algorithm is based on the fact that a circumsphere in the plane is the largest circle containing a nonredundant subset of the original set. Its worst case order is $O(m^3)$.

Another algorithm is given by Rademacher and Toepelitz [9, p. 103]. This algorithm is based on the fact that a circumsphere in the plane is the smallest circle containing two or three points of a set and which also encloses the set. Its order is $O(m^4)$.

Another approach is given by Franklin [6, p. 237] in which the problem of finding circumspheres is reduced to a problem in quadratic programming. I do not know the order of this approach.

After Jung's paper in 1901 [8], the focus of the papers involving circumspheres turned toward finding better proofs of the existence of circumspheres, and the relation between the radius of the circumsphere and diameter of a set in n space. A notable effort in this direction is Blumenthal and Wahlin [1] where Helley's theorem is used to reduce the general case of an arbitrary compact set to the case of a finite set. A different approach is taken in Eggleston [5, p. 76] where Blaschke's selection theorem is used to prove the existence of circumspheres directly. Probably the most compact proof of Jung's theorem is given by Gustin [7] who reformulated the proof given by Verblunsky [12]. An overview of these and related results can be found in [4]. All of these methods, however, are strictly finite-dimensional.

REFERENCES CITED

1. Blumenthal, L. M., and G. E. Wahlin. "On the Spherical Surface of Smallest Radius Enclosing a Bounded Subset of n-dimensional Euclidean Space." Bulletin of the American Mathematical Society, 47 (1941), 771-77.

2. Chrystal, G. "On the Problem to Construct the Minimum Circle Enclosing n Given Points in a Plane." Proceedings of the Edinburg Mathematical Society, 3 (1885), 30-33.

3. Danzer, L. "Über Durchschnittseigenschaften n-dimensionaler Kugelfamilien." Journal für die reine und angewandte Mathematik, 208 (1961), 181-203.

4. Danzer L., B. Grünbaum, and V. Klee. "Helly's Theorem and its Relatives." In Convexity. Proceedings of Symposia in Pure Mathematics, Vol. 7. Providence: American Mathematical Society, 1963, pp. 100-81.

5. Eggleston, H. G. Convexity. Cambridge, England: University Press, 1958.

6. Franklin, J. "Mathematical Methods of Economics." Mathematical Monthly, 90, No. 4 (1983), 229-44.

7. Gustin, W. Rev. of Verblunsky [12]. Mathematical Reviews, 14, 495.

8. Jung, H. W. E. "Ueber die Kleinste Kugel, die eine räumliche Figur einschliesst." Journal für die reine und angewandte Mathematik, 123 (1901), 241-57.

9. Rademacher H., and O. Toepelitz. The Enjoyment of Mathematics. Trans. H. Zuckerman. Princeton: Princeton University Press, 1957.

10. Sylvester, J. J. "A Question in the Geometry of Situation." Quarterly Journal of Pure and Applied Mathematics, 1 (1857), 79.

11. _____. "On Poncelet's Approximate Valuation of Surd Forms." Philosophical Magazine, 20 (1860), 203-22.

12. Verblunsky, S. "On the Circumradius of the Bounded Set." Journal of the London Mathematical Society, 27 (1952), 505-07.

II. AUTOMATIC HANDLING OF FINITE-DIMENSIONAL, NONASSOCIATIVE ALGEBRAS

by

Ronald Kenneth Smith

Graceland College

Lamoni, IA

50140

Introduction

Definition 1. A nonassociative algebra X over a field F is a vector space over F along with a multiplication defined on X which satisfies the following: For every $x,y,z \in X$ and $c \in F$,

i. $c(x*y) = (cx)*y = x*(cy)$

ii. $x*(y+z) = x*y + x*z$

iii. $(x+y)*z = x*z + y*z$

Definition 2. A function $f:X^m \to X$ is called multilinear or m-linear if for each $i = 1,2,\ldots,m$ and all $a,b \in F$, and all $x_1 \cdots x_{i-1}, x, y, x_{i+1} \cdots x_m \in X$, $f(x_1 \cdots x_{i-1}, ax+by, x_{i+1} \cdots x_m) =$

$$af(x_1 \cdots x_{i-1}, x, x_{i+1} \cdots x_m) + bf(x_1 \cdots x_{i-1}, y, x_{i+1} \cdots x_m). \quad (1)$$

Note that multiplication in X is a multilinear function mapping $X^2 \to X$. There are many important multilinear functions defined on X. For example, the function $[x,y] = x*y - y*x$ is called the commutator. If $[x,y] = 0$ for every $x,y \in X$, then the algebra is commutative. Another multilinear function is the associator, which is usually designated by $(x,y,z) = (x*y)*z - x*(y*z)$. If it is identically zero for all possible arguments, the algebra is associative. Let $r(x,y,z) = (x,y,z) + (x,z,y)$, and $l(x,y,z) = (x,y,z) + (y,x,z)$. If both are 0 for all choices of x, y, and z, then X is said to be alternative. (r and l are linearizations of the right and left alternative laws, $x*(y*y) = (x*y)*y$ and $(x*x)*y = x*(x*y)$, respectively. It is easy to check that r and l are identities whenever the alternative laws hold, and the converse is true whenever the field is not of characteristic 2.)

We present here a technique which enables a computer to manipulate finite-dimensional algebras and multilinear functions defined on them. Specifically, we show an effective way to approach the computational difficulties involved in the following three problems:

The first problem is to check a nonassociative algebra for specific identities. There are examples in the literature of nonassociative algebras for which a certain identity or group of identities is claimed to hold. We present a general scheme and a computer program for check-

ing multilinear identities in a finite-dimensional algebra of modest size directly and automatically. The problem here is that when the algebra has dimension n over F and a function is m-linear, then, a direct substitution of basis elements requires an algorithm of order $O(n^m)$ which quickly becomes impractical.

The second problem is to find a basis for the linear span of the union of the ranges of one or more multilinear functions. If a multivariate function f defined on an algebra X is not identically zero, then the elements of the subspace of X spanned by the range of f are dependence relations between the basis elements of X which would have to hold if f were an identity. When f is multilinear, this subspace is spanned by the set of values of f evaluated on basis elements only. Even this set of values is difficult to compute by direct substitution for any but the smallest examples.

The third problem is to determine the effect of a change in the multiplication on the identities of an algebra. Once one or more dependence relations are known, it may be desirable to change the multiplication table and to see the effects without redoing all the calculations. Our method allows some freedom to experiment, even when relatively large tables are involved. Another situation where it is desirable to change the multiplication in an algebra and not have to start from scratch in checking identities occurs when there is bad data. Anyone who has spent the time and effort to input a large multiplication table knows how easily typographical errors can occur. After checking a large table only to find that the alternative law does not hold because of an erroneous subscript in the original copy (e.g. Kleinfeld [5, p. 296]), it is nice to be able to find out if changing that subscript will make the example work without redoing all the calculations.

The major goal of this project was to discover identities of degree 6 in the free alternative algebra on three generators. The elements of this algebra are sums of words with rational coefficients. Each word is a parenthesized string of copies of the generators a, b,

or c. The degree of a word is the number of letters appearing in it. The degree type of a word is a triple (i,j,k) of integers giving the number of occurrences of a, b, and c respectively. For example, the word (((ab)c)b)a has degree 5 and degree type (2,2,1). It is easily seen that the requirement that r and l be identities implies dependence relations among words of the same degree type whose degree is at least 3. We obtained a basis for the linear span of the dependence relations of degree type (2,2,2). The dimension of this subspace is 16. A set of 9 dependence relations can be chosen which, along with the permutations of the letters involved, spans the basis we obtained, and hence accounts for all the dependence relations in alternative algebras of degree type (2,2,2). These are listed in Appendix 2. Incidentally, we obtained an example of an alternative algebra of dimension 307.

In the next section, we will define what we mean by the table of a multilinear function. A table is a generalization of both a multiplication table and the matrix of a linear transformation. The concept of a table for each multilinear function is fundamental to our approach to the problems listed above. Following the discussion of tables, we define operations on multilinear functions, and the corresponding operations on their tables. Next, we discuss the computer implementation of tables and the operations on them, and then return to the three problems given above. After that, we briefly explain the use of the computer system we wrote. Finally, we give some of the details of our particular example, which was done using this system.

Tables

Let X be a finite-dimensional, nonassociative algebra over a field F. Let $V = \{v_1, v_2 \ldots v_n\}$ be a basis of X, and let $f: X^m \to X$ be a multilinear function defined on X. To find $f(x_1, x_2 \ldots x_m)$ where x_i is given by $\sum_{j=1}^{n} a_{ij} v_j$ for $i = 1,2 \ldots m$, it is easily seen that by repeated

application of (1), $f(x_1,x_2\ldots x_m) =$

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_m=1}^{n} a_{1i_1}a_{2i_2}\cdots a_{mi_m}\, f(v_{i_1},v_{i_2}\cdots v_{i_m}). \qquad (2)$$

Thus, the function f is completely determined on X^m by its values on the set V^m. Let $N_n = \{1,2\ldots n\}$. For $I = (i_1,i_2\ldots i_m) \in N_n^m$ and $1 \le k \le n$, let $c_{Ik} \in F$ be the coefficient of v_k in the expansion of $f(v_{i_1},v_{i_2}\ldots v_{i_m})$. That is, $f(v_{i_1},v_{i_2}\ldots v_{i_m}) = \sum_{k=1}^{n} c_{Ik}v_k$. Clearly f uniquely defines each c_{Ik}. Conversely, any choice of c_{Ik} for each $I \in N_n^m$, $1 \le k \le n$ defines a multilinear function f on X^m for a given basis V of X.

If the coefficients c_{Ik} are arranged into an $n^m * n$ array, it looks very much like the matrix of a linear transformation. In fact, by splitting up the evaluation of f into two natural pieces and defining a new vector space, we can exhibit a linear transformation whose matrix is precisely this array.

Definition 3. Let V_m be the vector space over F with basis $VXVX\ldots XV = V^m$. If $I = (i_1,i_2\ldots i_m) \in N_n^m$, denote by V_I the element $(v_{i_1}\ldots v_{i_m}) \in V_m$. Then, V_m is the space of formal sums of the form $\sum_I c_I V_I$ for $I \in N_n^m$, where $c_I \in F$.

Definition 4. Let the map $f_m: X^m \to V_m$ be given by

$$f_m(x_1,x_2\ldots x_m) = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_m=1}^{n} a_{1i_1}a_{2i_2}\cdots a_{mi_m}(v_{i_1},v_{i_2}\ldots v_{i_m}) \text{ where}$$

$x_i = \sum_{j=1}^{n} a_{ij}v_j$ $i = 1,2\ldots m$. We call the action of the map f_m formal multiplication.

Lemma 1. Let $f:X^m \to X$ be multilinear. There is a unique linear transformation $f':V_m \to X$ such that $f = f'f_m$. Its matrix is given by the $n^m * n$ array (c_{Ik}) where $f(V_I) = \sum_{k=1}^{n} c_{Ik}V_k$ for each $I \in N_n^m$.

Proof: Let $f'(\sum_I a_I V_I) = \sum_I a_I f(V_I)$. Then, $f'f_m(x_1 \ldots x_m) = f'(\sum_{i_1=1}^{n} \ldots \sum_{i_m=1}^{n} a_{1i_1} \ldots a_{mi_m} (v_{i_1} \ldots v_{i_m}) = f(x_1 \ldots x_m)$. For uniqueness, note that $f_m(V_I) = V_I$ for all $I \in N_n^m$. Suppose g is any other linear transformation with $gf_m = f$. Then, $f'(V_I) = f'f_m(V_I) = f(V_I) = gf_m(V_I) = g(V_I)$ for each $V_I \in V_m$. Since f' and g agree on basis of V_m, $f' = g$. Finally, $f'(V_I) = f(V_I) = \sum_{k=1}^{n} c_{Ik}v_k$ for each I, so (c_{Ik}) is the matrix of f'.

Definition 5. Let f be m-linear on X. The $n^m * n$ matrix (c_{Ik}) of the transformation $f':V_m \to X$ is called the table of f.

The reason for going to all the trouble to define a table for a function in this way is that there are many different physical forms that a table such as a multiplication table can take. Rather than tying our definition to a particular form, it is helpful to define the table abstractly so that we are free to represent it in more ways than one.

Operations on Tables

There are at least four important operations whereby new multilinear functions can be derived from old ones. Each such operation determines a corresponding operation on the tables of the functions involved. We shall first define these four operations, and then the corresponding operations on tables. Then, we shall show how to implement them.

Linear combinations. Let f and g be m-linear functions on X, and $c \in F$. Define cf and f+g in the usual manner. It is easy to verify that the resulting functions are m-linear.

Permutation. Let f be m-linear on X, and let $\rho \in S_m$, the symmetric group on m letters. Then, define $f_\rho(x_1 \cdots x_m) = f(x_{\rho(1)} \cdots x_{\rho(m)})$. To see m-linearity, look at the j^{th} slot. $f_\rho(\cdot[ax+by]_j\cdot) = f(\cdot[ax+by]_{\rho(j)}\cdot) = af(\cdot x_{\rho(j)}\cdot) + bf(\cdot y_{\rho(j)}\cdot)$ $= af_\rho(\cdot x_j\cdot) + bf_\rho(\cdot y_j\cdot)$. The second equality holds because f is linear in the $\rho(j)^{th}$ slot.

Composition. Let f be m-linear and g be p-linear on X, and let $1 \le j \le m$. Define $h = f \circ_j g$ by $h(x_1 \cdots x_{m+p-1}) = f(x_1 \cdots x_{j-1}, g(x_j \cdots x_{j+p-1}), x_{j+p} \cdots x_{m+p-1})$. We shall call this operation composition in the j^{th} slot. That h is linear in slots $1 \cdots j-1, p+j-1 \cdots p+m-1$ follows immediately from the linearity of f in slots $1 \cdots j-1, j+1 \cdots m$. For $j \le i \le j+p-1$ we have $h(\cdot(ax+by)\cdot) =$ $f(\cdot g(\cdot(ax+by)\cdot)\cdot) = f(\cdot ag(\cdot x\cdot)+bg(\cdot y\cdot)\cdot)$ by linearity in slot j-i+1 of g. Since f is linear in slot j, this last expression becomes

$$af(\cdot g(\cdot x\cdot)\cdot) + bf(\cdot g(\cdot y\cdot)\cdot) = ah(\cdot x\cdot) + bh(\cdot y\cdot).$$

Example 1. Let $f(x,y) = x*y$, the multiplication in X. Then, f is 2-linear by definition. The commutator $[x,y] = x*y - y*x$ is given by $f - f_\rho$ where ρ is the transposition $(12) \in S_2$. If we denote the associator $(x,y,z) = (x*y)*z - x*(y*z)$ by $a(x,y,z)$, then a is given by $a = f \circ_1 f - f \circ_2 f$, and a is 3-linear. The right alternator function, which we denote by $r(x,y,z) = a(x,y,z) + a(x,z,y)$, is $r = a + a_\rho$ where $\rho = (23) \in S_3$. Many other important multilinear functions are similarly derived via these operations.

Now let us define the corresponding operations on the tables of multilinear functions.

Linear combinations. Scalar multiplication and addition are defined in the usual way for matrices.

Permutations. For $\rho \in S_m$, define $\rho(c_{Ik}) = (c_{\rho(I)k})$, where $\rho(i_1 \ldots i_m) = (i_{\rho(1)} \ldots i_{\rho(m)})$ for each $(i_1 \ldots i_m) \in N_n^m$.

Composition. Let (a_{Ik}) be $n^m * n$, and (b_{Hk}) be $n^p * n$. Define the $n^{m+p-1} * n$ table $(c_{Jk}) = (a_{Ik}) *_j (b_{Hk})$ where for each $J = (i_1 \ldots i_{m+1-1}) \in N_n^{m+p-1}$,

$$c_{Jk} = \sum_{q=1}^{n} b_{(i_j \ldots i_{j+p-1})q} \; a_{(i_1 \ldots i_{j-1}, q, i_{j+p} \ldots i_{m+p-1})k}.$$

The correspondence between the operations on multilinear functions and the table operations defined here is given in the lemma below. First let us look at a special case of composition.

In the special case where f and g map $X \to X$, that is, when $m = p = 1$, we have that formal multiplication from X into V_1 is an isomorphism, and that (a_{ik}) and (b_{ik}) are the usual matrices of these linear transformations. For example, $f(v_i) = \sum_{j=1}^{n} a_{ij} v_j$. In this case, composition must occur in the first slot, since there is only one, and $(a_{ik}) *_1 (b_{ik}) = (\sum_{q=1}^{n} b_{iq} a_{qk})$, which is the usual multiplication of the $n * n$ matrices $(b_{ik}) * (a_{ik})$. This is precisely the matrix for the linear transformation $f \circ g$. Regarding permutation, the identity is the only permutation of one slot, so the corresponding operation is trivial here.

Lemma 2. Let f, g be m-linear and h be p-linear on X, with tables (a_{Ik}), (b_{Ik}), and (c_{Hk}) respectively. Let $d \in F$, $\rho \in S_m$, and $1 \leq j \leq n$. The tables for df, f+g, f_ρ, and $f \circ_j h$ are given by $d(a_{Ik})$, $(a_{Ik}) + (b_{Ik})$, $\rho(a_{Ik})$, and $(a_{Ik}) *_j (c_{Hk})$, respectively.

Proof: $df(V_I) = d \sum_{k=1}^{n} a_{Ik} v_k = \sum_{k=1}^{n} d a_{Ik} v_k$ for every $I \in N_n^m$.

Consequently, the table of df is $d(a_{Ik})$. Similarly, $(f+g)(V_I) =$

$f(V_I) + g(V_I) = \sum_{k=1}^{n} a_{Ik}v_k + \sum_{k=1}^{n} b_{Ik}v_k = \sum_{k=1}^{n} (a_{Ik}+b_{Ik})v_k$. Also,

$f_\rho(V_I) = f(V_{\rho(I)}) = \sum_{k=1}^{n} a_{\rho(I)k}v_k$. Finally, $f \circ_j h(v_{i_1}\cdots v_{i_{m+p-1}}) =$

$f(v_{i_1}\cdots v_{i_{j-1}}, h(v_{i_j}\cdots v_{i_{j+p-1}}), v_{i_{j+p}}\cdots v_{i_{m+p-1}}) =$

$f(v_{i_1}\cdots v_{i_{j-1}}, \sum_{q=1}^{n} c_{(i_j\cdots i_{j+p-1})q}\, v_q, v_{i_{j+p}}\cdots v_{i_{m+p-1}}) =$

$\sum_{q=1}^{n} c_{(i_j\cdots i_{j+p-1})q}\, f(v_{i_1}\cdots v_{i_{j-1}}, v_q, v_{i_{j+p}}\cdots v_{i_{m+p-1}}) =$

$\sum_{k=1}^{n} [\sum_{q=1}^{n} c_{(i_j\cdots i_{j+p-1})q}\, a_{(i_1\cdots i_{j-1},q,i_{j+p}\cdots i_{m+p-1})k}]v_k$

so that the table of $f \circ_j h$ is given by $(a_{Ik}) *_j (c_{Hk})$. □

Implementation

In order to implement efficiently the operations we have defined, it is essential to choose an appropriate structure for the tables. In practice, the tables we can work with are very sparse. That is, most of the entries are 0. For instance, in our main example, the multiplication table that we start with has over 33 million entries, when in fact fewer than 6 thousand are nonzero. In Kleinfeld's example [5] of dimension 107, the multiplication table has over 1 million entries, but fewer than 700 are nonzero. Even small examples, such as the one in Humm and Kleinfeld [4] of dimension 23 quickly get out of hand if entire tables are represented. The table for the function (x*y)*z in [4], for example, has 279,841 entries, of which 147 are nonzero. It is standard procedure to represent such tables using linked lists.

For input and output of tables, we chose to use a singly linked list of the nonzero rows of the table. Each row in turn consisted of the row label along with a singly linked list of nonzero entries in the row. This form is ideal for input and output of tables, but we found that having the tables listed by columns was ideal for carrying out the

operations, particularly composition. When it is helpful to distinguish between the row representation of a table and its column representation, we call the latter a cotable. This is for our convenience only, as both represent the same table. The cotable could be thought of as the transpose of its table, but there is little theoretical advantage in this. For implementation purposes, there is no structural difference between a table and a cotable except for the length of the basis element lists in the primary and secondary lists that are used to represent the table or cotable. (See figure 1.)

Table (by rows)		Cotable (by columns)	
(v_1,v_2) :	$3v_1\ 1v_3$	v_1:	$3(v_1,v_2)\ \ 5(v_2,v_4)$
(v_2,v_3) :	$2v_4$	v_3:	$1(v_1,v_2)$
(v_2,v_4) :	$5v_1\ -1v_4$	v_4:	$2(v_2,v_3)\ -1(v_2,v_4)$

Figure 1. The structure of a table and its associated cotable

Figure 1 illustrates the two representations of a single table that we worked with. We did not link the tables by row and column both, because maintaining both kinds of links would have been very costly in terms of time when operating on the tables. Rather, we chose to do a sort each time it was necessary to change from a list by columns to a list by rows and vice versa.

Now let us consider each of the three operations in turn and how they are implemented. Taking linear combinations of tables amounts to the usual scalar multiplication and addition of matrices. This can be done using either the row representation or the column representation with equal efficiency. One simply multiplies each coefficient in the tables by the appropriate scalar and then merges the two lists, adding coefficients whenever the row labels and the column labels coincide.

To implement the permutation of a table, it is only necessary to permute the subscripts of the basis elements in each row label and then sort the result to get the new table. If the table is listed by rows, then one large sort involving all the rows is required. If the table

is listed by columns, then each column must be sorted. Typically, these sorts are each much smaller than one big sort would be. The actual trade-off depends on the particular table, and we chose arbitrarily to do it by columns because of the advantages of that representation for the operation of composition, which we shall see next.

To obtain the composition $A *_j B = C$ where $A = (a_{Ik})$ is $n^m * n$, $B = (b_{Hk})$ is $n^p * n$, and $C = (c_{Jk})$ is $n^{m+p-1} * n$, we need

$$c_{(i_1 \ldots i_{j-1}, h_1 \ldots h_p, i_{j+1} \ldots i_m)k} = \sum_{q=1}^{n} b_{(h_1 \ldots h_p)q} \, a_{(i_1 \ldots i_{j-1}, q, i_{j+1} \ldots i_m)k}.$$

We proceed by choosing each nonzero a_{Ik} in the k^{th} column of A, and forming a list in the following way. Let q be the j^{th} element of the list I. That is, let $q = i_j$. List all the products $b_{Hq}a_{Ik}$ with nonzero elements from column q of B. To each product, attach the row label J formed by substituting the list H for the element i_j in I, giving $J = (i_1 \ldots i_{j-1}, h_1 \ldots h_p, i_{j+1} \ldots i_m)$. Attach the column label k. These are precisely the row and column labels of c_{Jk} in which the product occurs as a summand. As a result, when all the lists from the k^{th} column of A have been compiled, the elements with identical row labels J will be all the nonzero summands of c_{Jk}. Merging these lists gives all the nonzero entries in the k^{th} column of C. Repeating this for each column of A completes the task.

While the columns of A are searched in a linear fashion in this scheme, the columns of B need to be accessed randomly in order to make this scheme work efficiently. Our solution is to use an index consisting of n pointers, one for each possible column of B. The index is initialized only when this operation is called for. This saves space and time for sparse tables since there is no other space or time devoted to zero entries in the tables by any of the other operations.

Example 2. To form the table of $g = f \circ_1 f$ for the function f whose table is given in figure 1, construct the cotable of f, as in figure 1. Then, write out the products for the cotable of g. Finally, list g in its row representation. (See figure 2.)

Cotable of f	Cotable of g
v_1: $3(v_1,v_2)$ $5(v_2,v_4)$	v_1: $9(v_1,v_2,v_2)$ $15(v_2,v_4,v_2)$
v_3: $1(v_1,v_2)$	v_3: $3(v_1,v_2,v_2)$ $5(v_2,v_4,v_2)$
v_4: $2(v_2,v_3)$ $-1(v_2,v_4)$	

Table of g

(v_1,v_2,v_2) : $9v_1$ $3v_3$

(v_2,v_4,v_2) : $15v_1$ $5v_3$

FIGURE 2: Composition of functions
$g = f \circ_1 f$ where f is given in figure 1

Note that in this example, the table of g has dimension 64*4, but it only has 4 nonzero entries. This method obtains the composite function g with almost trivial effort compared to trying all 64 combinations of basis vectors.

This basic technique for composition of functions applied to multiplication tables is essentially found in Hentzel and Hogben [3], although the idea of cotables is not found there. The search of the multiplication table for a pair (i,j) so that $f(v_i,v_j)$ has v_k in its expansion is called "factoring" in that paper. The idea is that if v_3 (say) occurs in an expansion of $g(v_i,v_j,v_k) = f(f(v_i,v_j),v_k)$, then it must occur in $f(v_h,v_k)$ for some h and k. The only possibility is $f(v_1,v_2)$, so that v_3 is "factored" into (v_1,v_2). But then v_1 must have come from some expansion $f(v_i,v_j)$. From the table of f, we find that v_1 can be factored in two ways, into either (v_1,v_2) or (v_2,v_4). Hence, $g(v_1,v_2,v_2)$ and $g(v_2,v_4,v_2)$ are exactly the rows of the table of g containing v_3.

Having seen how these operations can be carried out with some degree of efficiency on a computer, let us turn to each of the three tasks posed earlier and explain our approach to them. Checking identities is a very straightforward procedure. Simply generate the table (or cotable) for that function. If it is empty, the identity holds. Otherwise, we have a list of dependence relations which must all be zero if the function is to be an identity.

For example, suppose an algebra is to be checked to see if it is right alternative. Beginning with the table for multiplication, let A be its cotable. From A, generate $B = A *_1 A - A *_2 A$. B is the cotable for the associator function, (x,y,z). Then, let $C = A + (23)A$. C is the cotable for the right alternator function, $(x,y,z) + (x,z,y)$.

Once a table has been constructed, it is a simple matter to put its rows into row echelon form. These rows are a basis for the space spanned by the function acting on basis elements. By multilinearity, this is the whole space spanned by the range of the function.

The next problem is to substitute back into the original multiplication table so that the desired dependence relations hold. Suppose we want $\sum_{i=1}^{n} a_i v_i = 0$, and that $a_1 = 1$. We can eliminate v_1 from the algebra in the following way. If v_1 occurs in the expansion of $v_i * v_j$ with coefficient $c_{(i,j)1}$, then adding $c_{(i,j)1}(-v_1 + \sum_{k=2}^{n} a_k v_k)$ to this expansion expresses the product without v_1. Note that each such pair (i,j) along with the coefficient $c_{(i,j)1}$ is immediately available in the first row of the cotable for the multiplication. Since v_1 is no longer to be considered a basis element, any nonzero product in which it occurs should be deleted from the multiplication table.

Finally, we show how to recheck identities after changes are made in the multiplication table without having to repeat all the work.

Lemma 3. Let f and g be m-linear and h be p-linear on X.

i. $(f+g)_\rho = f_\rho + g_\rho$ for all $\rho \in S_m$.

ii. $(f+g) \circ_j h = f \circ_j h + g \circ_j h$ for $1 \leq j \leq m$.

iii. $h \circ_j (f+g) = h \circ_j f + h \circ_j g$ for $1 \leq j \leq p$.

Proof: i. $(f+g)_\rho(V_I) = (f+g)(V_{\rho(I)}) = f(V_{\rho(I)}) + g(V_{\rho(I)})$ $= f_\rho(V_I) + g_\rho(V_I)$ for each $V_I \in V_m$.

ii. If $1 \leq k \leq j-1$, then $(f+g) \circ_j h(v_{i_1} \cdots v_{i_{m+p-1}}) =$

$$(f+g)(v_{i_1} \cdots v_{i_k} \cdots v_{i_{j-1}}, h(v_{i_j} \cdots v_{i_{j+p-1}}), v_{i_{j+p}} \cdots v_{i_{m+p-1}}) =$$

$$f(v_{i_1} \cdots v_{i_k} \cdots v_{i_{j-1}}, h(v_{i_j} \cdots v_{i_{j+p-1}}), v_{i_{j+p}} \cdots v_{i_{m+p-1}}) +$$

$$g(v_{i_1} \cdots v_{i_k} \cdots v_{i_{j-1}}, h(v_{i_j} \cdots v_{i_{j+p-1}}), v_{i_{j+p}} \cdots v_{i_{m+p-1}}) =$$

$f \circ_j h(v_{i_1} \cdots v_{i_{m+p-1}}) + g \circ_j h(v_{i_1} \cdots v_{i_{m+p-1}})$. The other cases, for $j \leq k \leq j+p-1$ and $j+p \leq k \leq j+m-1$ are similar. The last part is proved in the same way. □

This lemma can save quite a bit of effort when small changes are made in a table. For example, let f, a, and r be the old multiplication, associator, and right alternator functions, and let $f + \Delta f$ represent the new multiplication, where Δf has a relatively small table in comparison to the table of f. Then, $(f + \Delta f) \circ_1 (f + \Delta f) =$ $f \circ_1 f + f \circ_1 \Delta f + \Delta f \circ_1 (f + \Delta f)$. Since Δf has a small table, there is not nearly as much work involved in calculating the table of $f \circ_1 \Delta f + \Delta f \circ_1 (f + \Delta f)$ and adding it to the table of $f \circ_1 f$ as there would be in adding $f + \Delta f$ and then calculating the final table. It is easy to verify that Δa, the change in the associator function, is given by $\Delta a = (f \circ_1 \Delta f - f \circ_2 \Delta f) + [\Delta f \circ_1 (f + \Delta f) - \Delta f \circ_2 (f + \Delta f)]$. The change in the right alternator function is $\Delta r = \Delta a + (23)\Delta a$.

Program Description

A computer program was written to implement these ideas. It is an interpreter for a simple language for manipulating tables and cotables for multilinear functions defined on finite-dimensional nonassociative algebras. Here we give a brief description of this system.

There are 21 commands available to the user. These fall into four categories. There are I/O, OPERATION, SUPERVISORY and MACRO commands.

The I/O commands are ENTER, USE, PRINT, and SAVE. The user may ENTER basis element names and tables directly. Once a table has been defined, either by entering it or via an operation on existing tables, the user can PRINT it. He may also SAVE it on an external file in internal form for quick retrieval. Basis element names and tables in this form are retrieved with the USE command.

The OPERATION commands are INVERT, ADD, SUBTRACT, COMBINE, PERMUTE, FACTOR, REDUCE, and MODS. The INVERT command creates the cotable corresponding to a given table and vice versa. ADD creates the table containing the sum of two other tables. SUBTRACT creates the table with their difference. COMBINE is just like ADD, except that the original tables are physically combined and therefore destroyed. The action of a permutation on a table is accomplished with the PERMUTE command. The composition operation is called FACTOR. REDUCE puts a table into row echelon form, except for dividing to normalize the leading term in each row. MODS is a specialized function that takes a table of dependence relations, solves for a specified variable in each relation, and creates a table of modifications that will eliminate the designated variables from the expansions in a given table.

The SUPERVISORY commands help the operator run the system. They include OPEN, SKIP, DESTROY, LIST, HELP, and EXIT. OPEN causes an external file to be available for the USE and SAVE commands. SKIP allows a portion of an external file to be ignored when tables or names are being read in. DESTROY frees the storage space from a table for allocation to new tables. LIST gives the names of each table defined by the user. HELP lists the commands available to the user and explains the required parameters and formats. Finally, EXIT terminates the whole process.

The last category of commands are MACRO instructions which combine several of the basic commands. These are ASSOCIATE, ALTERNATE, and

DELASS. ASSOCIATE creates the cotable $A *_1 B - A *_2 B$ for any cotables A and B. ALTERNATE creates the cotable $A + \rho A$ for any cotable A and an appropriate permutation ρ. DELASS creates the table $[A *_1 B - A *_2 B] + [B *_1 (A + B) - B *_2 (A + B)]$ and the table $A + B$. This corresponds to the change in the associator if A represents the original multiplication table and B the desired change in the multiplication table.

An Alternative Algebra Example

As we mentioned before, we generated all the dependence relations between words of degree type (2,2,2) and we constructed explicitly the multiplication table for the segment of the free alternative algebra generated by 3 letters and which is spanned by words of degree type (i,j,k) where i, j, and $k \leq 2$. We followed these steps:

We began by choosing a set which spans the desired algebra, but which is not quite a basis. This set consists of all left associated strings involving only a, b, c, or (a,b,c) and which contain no more than two occurrences of each generator, e.g. bb, a(a,b,c), (a(a,b,c))b, ((((ab)c)a)c)b, and (a,b,(a,b,c)). These 323 strings are listed in Appendix 1. To see that this set spans the whole space, we need the following facts. First, in any alternative algebra, it is easy to see that $(x_{\rho(1)},x_{\rho(2)},x_{\rho(3)}) = \mathrm{sgn}(\rho)(x_1,x_2,x_3)$ for each $\rho \in S_3$. This can be found, for example, in [1, p. 879]. Now $x(yz) = (xy)z - (x,y,z)$, so that any right association can be expressed as the sum of a left associated product and an associator. Using the following lemma, we can always reduce an associator involving products to a sum of products involving associators. In this way, we can decompose any product into a linear combination of our near basis elements.

Lemma 4. In any alternative algebra whose field is not of characteristic 3, we have for every w, x, y, and z,

$$(wx,y,z) = 2/3\ x(y,z,w) + 2/3\ (x,y,z)w + 1/3\ w(x,y,z) + 1/3\ (w,x,y)z - 1/3\ z(w,x,y) - 1/3\ (z,w,x)y + 1/3\ y(z,w,x) + 1/3\ (y,z,w)x.$$

Proof: In any ring, by direct substitution, we have

$(wx,y,z) - (w,xy,z) + (w,x,yz) = w(x,y,z) + (w,x,y)z$ [1, p. 879]. Sum the expansions of (wx,y,z), $-(zw,x,y)$, (yz,w,x), and $2(xy,z,w)$. The result follows upon division by 3. □

Armed with these results and a program written by I. R. Hentzel [2] which actually did the expansions, we generated an expansion of each product of near basis elements in terms of this set.

Next, a text editor was used to code the whole table of parenthesized expressions into the names A1 - A3, B1 - B9, C1 - C25, D1 - D60, E1 - E105 and F1 - F121 for ease of handling. This table exists on tape, but because of its size, it is not listed here.

Our next task was to generate the tables of right and left alternators in this algebra. This was done using the program described in this paper. The program was also used to reduce the tables to row echelon form. Sixteen dependence relations resulted from this operation. Nine linear combinations of these rows were then chosen which, along with their permutations, span the same space as the original sixteen rows. To check this we used a text editor to generate the permutations, and then used our program to reduce the new identities to row echelon form. The rows all matched. These nine identities are listed in Appendix 2.

To check that the space spanned by our identities was the whole space of identities in our algebra, we used the original sixteen relations to substitute back into the multiplication table, eliminating 16 of the original spanning set in the manner we previously described. The resulting algebra was then checked to verify alternativity, and indeed turned out to be alternative. This table also resides on tape because of its size.

REFERENCES CITED

1. Bruck, R. H., and E. Kleinfeld. "The Structure of Alternative Division Rings." Proceedings of the American Mathematical Society, 2 (1951), 878-90.
2. Hentzel, I. R. "Decompose." A computer program. Mathematics Department, Iowa State University, Ames, Iowa. Unpublished.
3. Hentzel, I. R., and L. Hogben. "Exhaustive Checking of Sparse Algebras." Journal of Algorithms, 2 (1981), 44-47.
4. Humm, M. M., and E. Kleinfeld. "On Free Alternative Rings." Journal of Combinatorial Theory, 2 (1967), 140-44.
5. Kleinfeld, E. "On Centers of Alternative Algebras." Communications in Algebra, 8(3) (1980), 289-97.

APPENDIX 1: NEAR BASIS ELEMENTS

The following list gives the 323 words which span the segment of the free alternative algebra of degree type (i,j,k) where i, j, and k are no larger than 2. Here, X = (A,B,C).

A B C

AB AC BA BC CA CB AA BB CC

(AB)C (AC)B (BA)C (BC)A (CA)B (CB)A (AA)B (AA)C (AB)A (AB)B (AC)A

(AC)C (BA)A (BA)B (BB)A (BB)C (BC)B (BC)C (CA)A (CA)C (CB)B (CB)C

(CC)A (CC)B X

((AA)B)C ((AA)C)B ((AB)A)C ((AB)B)C ((AB)C)A ((AB)C)B ((AB)C)C

((AC)A)B ((AC)B)A ((AC)B)B ((AC)B)C ((AC)C)B ((BA)A)C ((BA)B)C

((BA)C)A ((BA)C)B ((BA)C)C ((BB)A)C ((BB)C)A ((BC)A)A ((BC)A)B

((BC)A)C ((BC)B)A ((BC)C)A ((CA)A)B ((CA)B)A ((CA)B)B ((CA)B)C

((CA)C)B ((CB)A)A ((CB)A)B ((CB)A)C ((CB)B)A ((CB)C)A ((CC)A)B

((CC)B)A ((AA)B)B ((AA)C)C ((AB)A)B ((AB)B)A ((AC)A)C ((AC)C)A

((BA)A)B ((BA)B)A ((BB)A)A ((BB)C)C ((BC)B)C ((BC)C)B ((CA)A)C

((CA)C)A ((CB)B)C ((CB)C)B ((CC)A)A ((CC)B)B AX BX CX XA XB XC

(((AA)B)B)C (((AA)B)C)B (((AA)B)C)C (((AA)C)B)B (((AA)C)B)C

(((AA)C)C)B (((AB)A)B)C (((AB)A)C)B (((AB)A)C)C (((AB)B)A)C

(((AB)B)C)A (((AB)B)C)C (((AB)C)A)B (((AB)C)A)C (((AB)C)B)A

(((AB)C)B)C (((AB)C)C)A (((AB)C)C)B (((AC)A)B)B (((AC)A)B)C

(((AC)A)C)B (((AC)B)A)B (((AC)B)A)C (((AC)B)B)A (((AC)B)B)C

(((AC)B)C)A (((AC)B)C)B (((AC)C)A)B (((AC)C)B)A (((AC)C)B)B

(((BB)C)C)A (((BB)C)A)C (((BB)C)A)A (((BB)A)C)C (((BB)A)C)A

(((BB)A)A)C (((BC)B)C)A (((BC)B)A)C (((BC)B)A)A (((BC)C)B)A

(((BC)C)A)B (((BC)C)A)A (((BC)A)B)C (((BC)A)B)A (((BC)A)C)B

(((BC)A)C)A (((BC)A)A)B (((BC)A)A)C (((BA)B)C)C (((BA)B)C)A

(((BA)B)A)C (((BA)C)B)C (((BA)C)B)A (((BA)C)C)B (((BA)C)C)A

(((BA)C)A)B (((BA)C)A)C (((BA)A)B)C (((BA)A)C)B (((BA)A)C)C

(((CC)A)A)B (((CC)A)B)A (((CC)A)B)B (((CC)B)A)A (((CC)B)A)B

(((CC)B)B)A (((CA)C)A)B (((CA)C)B)A (((CA)C)B)B (((CA)A)C)B

(((CA)A)B)C (((CA)A)B)B (((CA)B)C)A (((CA)B)C)B (((CA)B)A)C

(((CA)B)A)B (((CA)B)B)C (((CA)B)B)A (((CB)C)A)A (((CB)C)A)B

(((CB)C)B)A (((CB)A)C)A (((CB)A)C)B (((CB)A)A)C (((CB)A)A)B
(((CB)A)B)C (((CB)A)B)A (((CB)B)C)A (((CB)B)A)C (((CB)B)A)A
(AB)X (AC)X (BC)X (AX)B (AX)C (BX)A (BX)C (CX)A (CX)B (XA)B
(XA)C (XB)C (A,B,X) (A,C,X) (B,C,X)

((((CA)A)B)B)C ((((CA)A)B)C)B ((((BA)A)B)C)C ((((CA)B)A)B)C
((((CA)B)A)C)B ((((BA)B)A)C)C ((((CA)B)B)A)C ((((CA)B)B)C)A
((((AA)B)B)C)C ((((CA)B)C)A)B ((((BA)B)C)A)C ((((CA)B)C)B)A
((((AA)B)C)B)C ((((BA)B)C)C)A ((((AA)B)C)C)B ((((BA)A)C)C)B
((((BA)A)C)B)C ((((CA)A)C)B)B ((((BA)C)A)C)B ((((BA)C)A)B)C
((((CA)C)A)B)B ((((BA)C)C)A)B ((((BA)C)C)B)A ((((AA)C)C)B)B
((((BA)C)B)A)C ((((CA)C)B)A)B ((((BA)C)B)C)A ((((AA)C)B)C)B
((((CA)C)B)B)A ((((AA)C)B)B)C ((((AB)B)C)C)A ((((AB)B)C)A)C
((((CB)B)C)A)A ((((AB)C)B)C)A ((((AB)C)B)A)C ((((CB)C)B)A)A
((((AB)C)C)B)A ((((AB)C)C)A)B ((((BB)C)C)A)A ((((AB)C)A)B)C
((((CB)C)A)B)A ((((AB)C)A)C)B ((((BB)C)A)C)A ((((CB)C)A)A)B
((((BB)C)A)A)C ((((CB)B)A)A)C ((((CB)B)A)C)A ((((AB)B)A)C)C
((((CB)A)B)A)C ((((CB)A)B)C)A ((((AB)A)B)C)C ((((CB)A)A)B)C
((((CB)A)A)C)B ((((BB)A)A)C)C ((((CB)A)C)B)A ((((AB)A)C)B)C
((((CB)A)C)A)B ((((BB)A)C)A)C ((((AB)A)C)C)B ((((BB)A)C)C)A
((((BC)C)A)A)B ((((BC)C)A)B)A ((((AC)C)A)B)B ((((BC)A)C)A)B
((((BC)A)C)B)A ((((AC)A)C)B)B ((((BC)A)A)C)B ((((BC)A)A)B)C
((((CC)A)A)B)B ((((BC)A)B)C)A ((((AC)A)B)C)B ((((BC)A)B)A)C
((((CC)A)B)A)B ((((AC)A)B)B)C ((((CC)A)B)B)A ((((AC)C)B)B)A
((((AC)C)B)A)B ((((BC)C)B)A)A ((((AC)B)C)B)A ((((AC)B)C)A)B
((((BC)B)C)A)A ((((AC)B)B)C)A ((((AC)B)B)A)C ((((CC)B)B)A)A
((((AC)B)A)C)B ((((BC)B)A)C)A ((((AC)B)A)B)C ((((CC)B)A)B)A
((((BC)B)A)A)C ((((CC)B)A)A)B ((AB)C)X ((AC)B)X ((BA)C)X ((BC)A)X
((CA)B)X ((CB)A)X ((AB)X)C ((AC)X)B ((BA)X)C ((BC)X)A ((CA)X)B
((CB)X)A ((AX)B)C ((AX)C)B ((BX)A)C ((BX)C)A ((CX)A)B ((CX)B)A
((XA)B)C ((XA)C)B ((XB)A)C ((XB)C)A ((XC)A)B ((XC)B)A (A,B,X)C
(A,C,X)B (B,C,X)A A(B,C,X) B(A,C,X) C(A,B,X) XX

APPENDIX 2: IDENTITIES OF DEGREE 6, TYPE (2,2,2)

The following identities are linearly independent. Furthermore, they along with their permuted images have dimension 16, and span the space of dependence relations of type (2,2,2).

Note: In this table, X stands for the associator (A,B,C).

1. $-((AB)X)C + ((BA)X)C + (A,B,X)C$
2. $-((XA)B)C + ((XB)A)C + (A,B,X)C$
3. $((AB)C)X - ((BA)C)X - C(A,B,X) - 2XX$
4. $-((AB)C)X + ((AC)B)X + ((CA)B)X - ((CB)A)X - (A,B,X)C + (B,C,X)A$
5. $((AX)B)C - ((AX)C)B - ((CX)A)B + ((CX)B)A + (A,B,X)C - (B,C,X)A$
6. $((CA)B)X - ((CB)A)X - ((CX)A)B + ((CX)B)A$
7. $((CX)A)B - ((CX)B)A + ((XC)A)B - ((XC)B)A - (A,B,X)C - C(A,B,X) - 2XX$
8. $((XB)A)C - ((XB)C)A + ((XC)A)B - ((XC)B)A - B(A,C,X) - C(A,B,X)$
9. $3((XA)B)C - 3((XA)C)B - (A,B,X)C + (A,C,X)B - (B,C,X)A - 2A(B,C,X) - B(A,C,X) + C(A,B,X)$

Remark: The first two are direct consequences of Kleinfeld's identities [1, p. 881, eq. 2.20]. One simply multiplies on the right by C. The rest of the identities are new as far as we know. It is also interesting to note that the only degree 5 identities were Kleinfeld's.

SUMMARY AND QUESTIONS

In our first paper, we developed the theory of circumspheres in Hilbert space. Our main result is that if r is the circumradius and d the diameter of a nonempty set in Hilbert space, then $r \leq d/\sqrt{2}$. A simple example shows that this is the best possible bound, and a sufficient condition for equality to hold is given. It would be interesting to know whether this condition is also necessary.

In addition, we presented three algorithms for finding circumspheres in R^n. The best overall order that we could prove is $O(m^{n+1})$ where m is the number of points in the set. We gave an algorithm with best case order O(m). We suspect that the order of this algorithm is better than $O(m^{n+1})$, but were unable to either prove it or find an example in which this algorithm takes nearly this long.

In the second paper, we showed how to find and check multilinear identities in finite-dimensional nonassociative algebras using the computer and a very general, exhaustive technique. We found identities of degree 6 in the free alternative algebra on three generators. Two of the nine we found are simple consequences of Kleinfeld's identities of degree 5. It would be nice to independently verify the others through traditional means.

ACKNOWLEDGEMENTS

Special thanks are due to each of the following for their contributions to this effort over the last few years.

Irvin Hentzel,

Thanks for your support, encouragement, long-suffering, and for allowing me to tackle this project in my own disjointed way for all this time.

Di Smith,

Thanks for being my best friend.

Joe Clifton,

Thanks for all the good times in our office, and for showing me that circumspheres exist.

David DeBarthe and the Graceland Computing Services staff,

Thanks for all the computer time and the help you all gave, especially in getting this manuscript printed.

Jon Applequist,

Thanks for the computer time to implement my algorithms for finding circumspheres.

Iowa State Math Department,

Thanks for the computer time for my massive failures as well as my success in finding identities.

Jim Hawley, Bruce Graybill, and the administration of Graceland College,

Thanks for your support.

Sincerely,

Ron Smith