COMPUTING THE DEGREE OF MAPS AND A GENERALIZED METHOD OF BISECTION

by

Ralph Baker Kearfott

A dissertation submitted to the faculty of the
University of Utah in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Mathematics

University of Utah

June 1977

ACKNOWLEDGEMENTS

I wish to acknowledge my advisor, Professor Frank Stenger, for his continuing confidence in my abilities. I further wish to acknowledge my parents for the financial assistance they gave me at crucial times during my college education. Professor Grant B. Gustafson and Professor R. Eliot Chamberlin also deserve mention here, for the value of their advice and encouragement is inestimable.

I wish to acknowledge my wife, Ruth, for the preliminary typing of this dissertation, despite numerous frustrations. The several typists who put in long hours to get this dissertation on time also deserve mention.

TABLE OF CONTENTS

LIST OF FIGURES

Figure | Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Description of Content

Suppose that $F:\mathcal{D} \subset R^n \rightarrow R^n$ is continuously differentiable on the bounded, open domain $\mathcal{D}$ and $J(F)$, the Jacobian of F, is nonzero at the points $X \in \mathcal{D}$ for which $F(X) = Y$. Then $d(F,\mathcal{D},Y)$, the degree of F at Y with respect to $\mathcal{D}$, is defined to be the number of points $X \in \mathcal{D}$ with $F(X) = Y$ and $J(F)(X) > 0$, minus the number of such X for which $J(F)(X) < 0$, provided $F(X) \neq Y$ for X on the boundary of $\mathcal{D}$.

This concept can be generalized ([1] Ch. XII for the original scheme or [26] pp. 147-165) so that $d(F,\mathcal{D},Y)$ is defined for F which are only continuous on $\mathcal{D}$ and $F(X) \neq Y$, X on the boundary. For such F, the Kronecker existence theorem ([26] or [1]) states that, if $d(F,\mathcal{D},Y) \neq 0$, there exists at least one $X \in \mathcal{D}$ with $F(X) = Y$. For this reason, computation of $d(F,\mathcal{D},Y)$ is of interest in solving systems of non-linear equations (see [8] for a discussion of usefulness).

Methods of computing $d(F,\mathcal{D},Y)$ include quadrature schemes, for the degree can be expressed under suitable assumptions in terms of the Heinz integral [26] or the Kronecker integral ([1] or [14]). The method most similar to the one first developed in this dissertation is related to the Kronecker integral. For that method, Stenger [30] characterized the degree in terms of certain determinants, then calculated it by evaluating those determinants. An advantage of that method (ibid.) over some quadrature schemes was that F needed to be evaluated only to sufficient accuracy to determine the algebraic sign.

The first method developed in this dissertation comes from ideas in [30]. Required notation and several well-known characterizations of $d(F,\mathcal{D},Y)$ are mentioned in Chapter II. In Chapter III, characterizations to be used for machine computation appear. One of these, due to Stenger (ibid.), expresses the degree in terms of determinants of matrices with entries ± 1, while the other gives the degree directly in terms of matrices with entries ± 1. This latter characterization is proven in detail.

Chapter IV deals with an algorithm for computing $d(F,P,Y)$, where P is a polygon. The theory is developed to allow one to economize on both time and computer storage space. An example is then worked using this algorithm.

In Chapter V the algorithm for computing $d(F,P,Y)$ is employed in a method of finding the roots of a system of nonlinear equations. This method, embodied in Algorithm 5.3.1, does not require F to be differentiable.

In Chapter VI a relationship between the scheme for calculation of $d(F,P,Y)$ based on Section 3.2 (the second characterization of $d(F,P,Y)$) and a labeling scheme to define Sperner simplexes is presented. Other search routines for Sperner simplexes are briefly described and compared. Also, a search routine for a Sperner simplex which uses concepts about a labeling scheme for nodes in binary trees (presented in Chapter IV) is outlined.

In Chapter VII applications are considered. These include finding stationary points (optimization), calculating linking numbers (see [1] Ch. XII), and solving the hidden line problem (the hidden line problem can be stated as: "does a given curve pass through a surface in R^3" or "does a curve or point lie in front of or behind a surface in R^3"). Calculating $d(F,\mathcal{D},Y)$, where $\mathcal{D}$ is not necessarily a polygon, is also discussed.

1.2 Background for the Work

Various other topological methods for finding roots of systems of nonlinear equations or fixed points of maps in R^n have been investigated. Such procedures, including the method in this work, may be the only procedures to give meaningful approximations when a system is highly nonlinear, or may be the only ones to converge to fixed points other than "attractive" ones (see [3], esp. p. 2, [26], p. 185, and [26], Ch. 10). Scarf [27] presented one of the most famous with techniques similar to applications of Sperner's lemma ([6], pp. 417-421). The exchange algorithm [27] may perform efficiently when $\mathcal{D}$ is an n-simplex and F is given in terms of barycentric coordinates. If $\mathcal{D}$ can be transformed to an n-cube, then methods from Allgower and Keller ([29] with [21]) may be used. Variants and related methods are under investigation [9].

For the above methods, one usually assumes that $F: \mathcal{D} \to \mathcal{D}$ in order to find a fixed point. It is often sufficient, however, to know that F - I, where I is the identity map, has a root in $\mathcal{D}$. The existence of such roots can be determined with a desired probability by Algorithm 4.4.1 (the degree computation algorithm).

There is a close relationship between labelings for Sperner simplexes, the degree computation algorithm, Algorithm 5.3.1 (the generalized method of bisection), and the methods in [27] and [29]. This relationship will be described in Chapter VI. Algorithm 6.3.1, involving both Sperner simplexes and theory from Section 4.2 (dealing with labeling schemes for elements of simplicial subdivisions) also appears.

1.3 Problems Encountered

Problems encountered with some or all of the topological methods mentioned above are similar. For the method in [27], points on a grid are chosen a priori; other schemes may have similar limitations. In all of the methods, including those of this work, true fixed points may be "missed."

The generalized method of bisection may require more memory than the method in [27] or [29], but uses the additional technique of calculating $d(F,\mathcal{D},\theta_n)$. For this reason, it may be more powerful in some instances than other methods.

Nonetheless, Algorithm 4.4.1 (to compute $d(F,P,\theta_n)$) cannot be used to calculate $d(F,\mathcal{D},\theta_n)$ with absolute certainty. When the characterization in terms of determinants (in lieu of Theorem 3.2.10. the other characterization) provides the basis for an algorithm analogous to Algorithm 4.4.1, a higher degree of certainty is attained at the expense of computation time. Independently of the above modification, parameters in the degree computation algorithm can also be chosen so that more computation is done, but the computed value of $d(F,\mathcal{D},\theta_n)$ has a better chance of being correct. Such additional computations may then be used later in Algorithm 5.3.1 (the generalized method of bisection).

A method for quickly calculating the special determinants in the determinant characterization (Theorem 3.3.1) would improve the degree computation (Algorithm 4.4.1) and Algorithm 5.3.1 considerably.

CHAPTER II

NOTATION FOR THE REGION AND THE CONCEPT OF DEGREE

In the first section of this chapter, the notation used throughout the rest of this work is introduced. In the second section, the degree of a map with respect to a polygonal region is defined, its significance is discussed, and useful characterizations are given.

2.1 The Region

The notation in this section will describe the region over which the degree is defined.

2.1.1 Definition. Suppose that $\{X_0, X_1, \ldots, X_\mu\}$, $\mu \leq n$, is a set of of points in R^n, so that $\{X_k - X_0\}_{k=1}^{\mu}$ is a linearly independent set of n-vectors. Then the μ-simplex $S = \langle X_0, X_1, \ldots, X_\mu \rangle$ spanned by $\{X_0, \ldots, X_\mu\}$ is the closed convex hull of these points, i.e., it is the set $\Sigma_{k=0}^{\mu} \lambda_k X_k : \lambda_k \geq 0$ and $\Sigma_{k=0}^{\mu} \lambda_k = 1$.

2.1.2 Definition. The points $X_k, k = 0, \ldots \mu$ are called the extreme points or vertices of S.

In R^2, a 2-simplex is simply a triangle while in R^3, a 3-simplex is a tetrahedron. A 1-simplex $\langle A, B \rangle$ denotes a line segment whose endpoints are A and B.

We next define orientations with the following two definitions.

<u>2.1.3 Definition</u>. If $Z_1,\ldots,Z_n$ are row vectors in R^n, then $\Delta_n(Z_1,\ldots,Z_n)$ is the determinant of the matrix whose k-th row is Z_k.

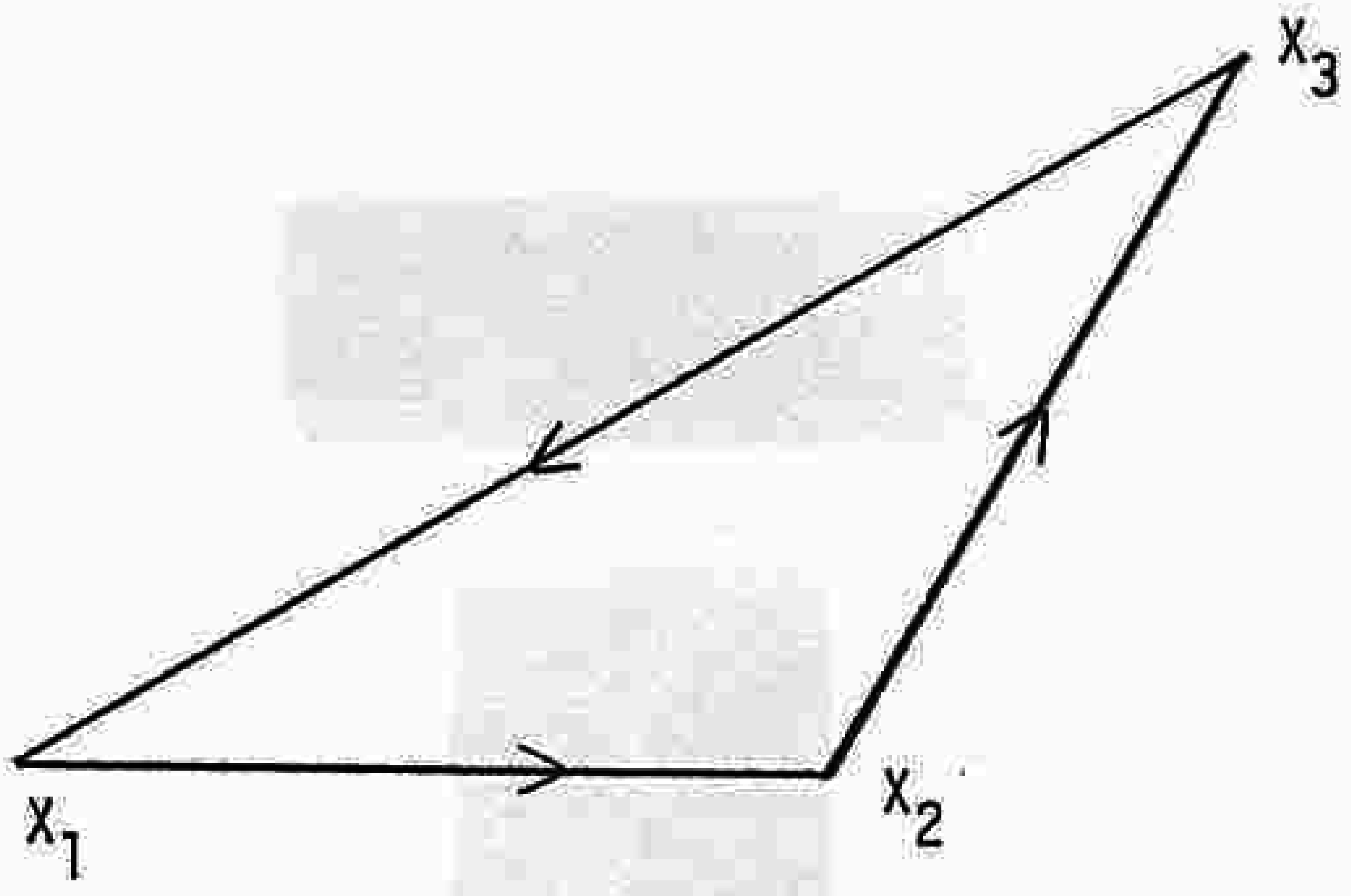
<u>2.1.4 Definition</u>. If $S = \langle X_0,\ldots,X_n\rangle$ is an n-simplex in R^n, then one assigns a positive or negative orientation to S according to whether the determinant $\Delta_n((1,X_0)\ldots,(1,X_n))$ is a positive or negative (see [1] Ch. 4, §2, #1 and [1], Anhang II).

If the k-th and m-th points $(m \neq k)$ of $S = \langle X_0,\ldots,X_k,\ldots, X_m,\ldots,X_n\rangle$ are interchanged, then the orientation of S is reversed. One writes: $\langle X_0,\ldots,X_m,\ldots,X_k,\ldots,X_n\rangle = -\langle X_0,\ldots,X_k,\ldots,X_m,\ldots,X_n\rangle$.

Opposite orientations of a triangle in R^2 are illustrated in Fig. 2.1 and Fig. 2.2.

In general, an odd permutation of the list of points of S changes the orientation, whereas an even permutation does not.

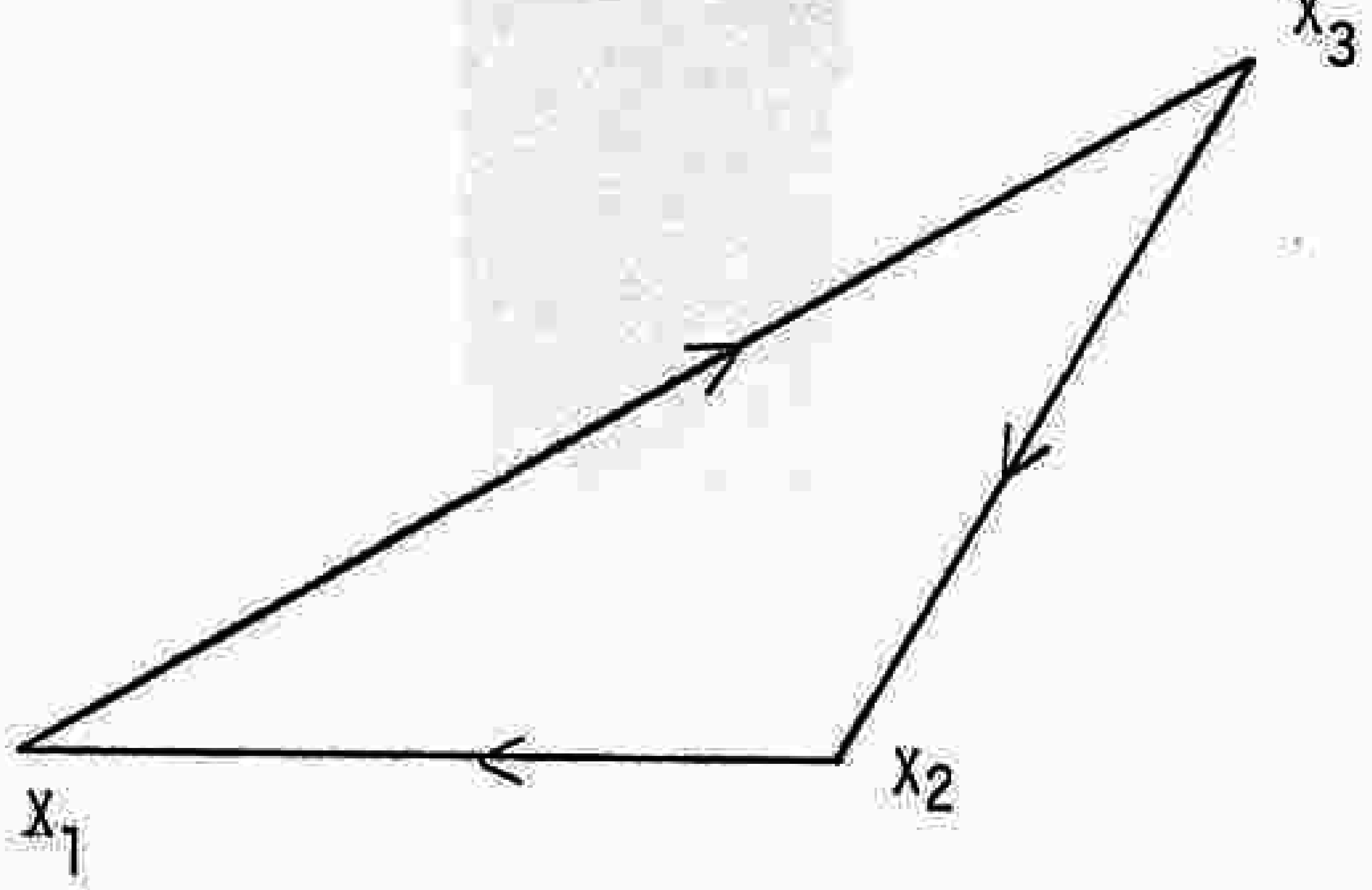
If P is a union of μ-dimensional polygons in R^n, then P can be written as the union of a finite number of μ-simplexes in R^n with the property that the intersection of any two of these simplexes has empty interior. (We define a μ-dimensional polygon as a connected union of μ-simplexes). If these simplexes are $S_1,\ldots,S_\kappa$, then one writes $P = \Sigma_{k=1}^{\kappa} S_k$.



$S = \langle X_1, X_2, X_3 \rangle$

FIGURE 2.1

POSITIVE ORIENTATION



$T = \langle X_1, X_3, X_2 \rangle = -S$

FIGURE 2.2

NEGATIVE ORIENTATION

An example is given in Fig. 2.3. There, the topological boundary of P can be thought of as being "traversed" in the direction depicted. When the boundaries of S_1 and S_2 are also traversed in this manner, the segment $\langle A,C\rangle$ in the interior of P is covered in opposite directions.

As does the concept of the triangle itself, the above characterization of the boundary generalizes to higher dimensions.

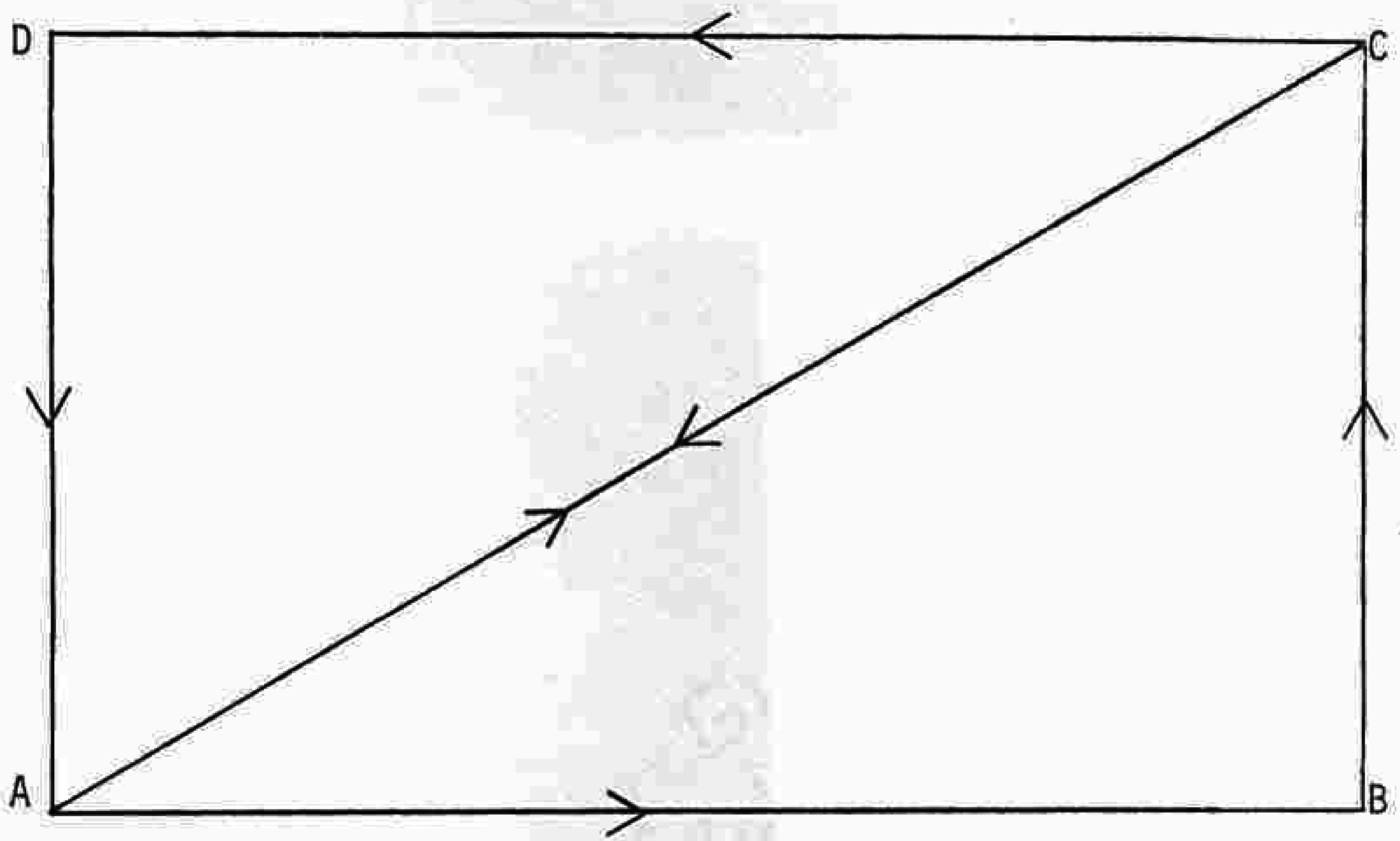
<u>2.1.5 Definition</u>. If $S = \langle X_0,\ldots,X_\mu\rangle$ is a μ-simplex in $R^n (n \geq \mu)$, then the <u>algebraic boundary</u> of S is formally defined by:

$$\text{(i) } b(S) = \sum_{k=0}^{\mu}(-1)^{k-1}\langle X_0,\ldots,\hat{X}_k,\ldots,X_\mu\rangle$$

In this formula, $\langle X_0,\ldots,\hat{X}_k,\ldots,X_\mu\rangle$ is the $(\mu - 1)$-simplex in R^n obtained by deleting X_k from the ordered list of vertices for S.

<u>2.1.6 Definition</u>. If S is as in Def. 2.1.5, then the $(\mu - 1)$-simplex $\langle X_0,\ldots,\hat{X}_k,\ldots,X_\mu\rangle$ will be called the <u>k-th facet of S</u>.

The formal sum in the right member of (i) is interpreted geometrically as an oriented union of $(\mu - 1)$-simplexes appearing within the summand. In a general sum of the form $\sum_{k=0}^{\kappa} S_k$, where the S_k are all oriented μ-simplexes, a similar interpretation is given; there is cancellation when simplexes with similar vertices but opposite orientations appear, and an integer weight is assigned to S if S



$P = ABCD$

$S_1 = \langle A,B,C\rangle;\ S_2 = \langle A,C,D\rangle$

$P = S_1 + S_2$

FIGURE 2.3

A SUM OF SIMPLEXES

appears more than once with the same orientation. (See [1], III or [19], I, #3).

For example, in Fig. 1.3,

$$\begin{aligned} b(P) &= b(S_1) + b(S_2) = [\langle B,C\rangle - \langle A,C\rangle + \langle A,B\rangle] \\ &\quad + [\langle C,D\rangle - \langle A,D\rangle + \langle A,C\rangle] = \langle B,C\rangle + \langle A,B\rangle + \langle C,D\rangle \\ &\quad - \langle A,D\rangle + [\langle A,C\rangle - \langle A,C\rangle] = \langle B,C\rangle + \langle A,B\rangle + \langle C,D\rangle - \langle A,D\rangle \\ &\quad = \langle A,B\rangle + \langle B,C\rangle + \langle C,D\rangle + \langle D,A\rangle \end{aligned}$$

as one would wish.

<u>2.1.7 Remark</u>. The boundary operator, $b(\cdot)$, is linear.

<u>2.1.8 Remark</u>. If one makes cancellations, ignores the orientation of the component simplexes and interprets "Σ" as "$\cup$" one obtains the topological boundary of P.

In most of the theorems and algorithms to follow, it is assumed that P is a simplex. However, the theorems are still true for arbitrary polygons, and the algorithms can be made to work for polygonal regions other than simplexes.

For a discussion of the above concepts (in a slightly more general form), see [8], pp. 2-7.

2.2 The Brouwer Degree

Suppose that P is an n-dimensional polygon in R^n and that $G: b(P) \to S^n$ is continuously differentiable, where S^n denotes the surface of the unit n-disk in R^n. Now let us produce a sequence, $\left\{ \{S_i^j\}_{i=1}^{\kappa_j} \right\}_{j=1}^{\infty}$ of sets of (n - 1)-simplexes so that $b(P) = \sum_{i=1}^{\kappa} S_i^j$ and if j is large, then the diameter of S_i^j is small for $i = 1, \ldots, \kappa_j$. This done, let us approximate G by the sequence, $\{L_j\}_{j=1}^{\infty}$, of maps which interpolate G at the vertices of $S_i^j (i = 1, \ldots, k_j)$ and which are affine in the interiors of the S_i^j (See [1] pp. 341-342). Suppose now that $X \in b(P)$, and that the Jacobian of G at P is nonsingular at X. Then, for j large enough, the images of the S_i^j containing X have the same orientation (since the Jacobian of G has one sign in a neighborhood of X). Furthermore, for large j, no S_i^j containing X contains any other point, X_1, with $G(X_1) = G(X)$. With this in mind, the following definition is made.

<u>2.2.1 Definition</u>. Suppose that P and G are as above, that j is "large enough" (as above) and fixed, and that Y is any point on S^n. Then the <u>rotation of G</u> is defined to be the number of S_i^j containing a point of $G^{-1}(Y)$ which have a positive orientation, minus the number of such S_i^j with a negative orientation.

It can be proven that the rotation of G is independent of the choice of Y ([25], p. 28), so that Def. 2.2.1 makes sense. (See also [1], [20], or [8], pp. 2-25, and esp. [8], Theorem 2.7).

If G is assumed only to be continuous on b(P), then G can be approximated by maps which are C^1 (See [26], p. 15 , and [5], or [8]). Thus, the rotation of such G can be defined, [8].

Now suppose that $F:P \to R^n$ is an arbitrary continuous map, $A \in R^n$, and $F(X) \neq A$ for $X \in b(P)$. Then, by restricting F to b(P), the rotation of $(F - A)/\|F - A\|$ is defined:

<u>2.2.2 Definition</u>. If P, A, and F are as above, then d(F,P,A), <u>the degree of F at A with respect to P</u>, is defined to be the rotation of $(F - A)/\|F - A\|$.

<u>2.2.3 Remark</u>. Note that $d(F,P,A) = d(F - A,P,\Theta_n)$, where Θ_n is the n-dimensional zero vector. Hence, without loss of generality, one need only consider the degree of F at Θ_n to deal with d(F,P,A), where A is any point in R^n.

Very often, $d(F,P,\Theta_n)$ is defined in terms of the Heinz integral[16]. With that definition, important properties of $d(F,P,\Theta_n)$ can be proven with relatively simple analytic tools [26]. Nonetheless, it can be shown that the definition in terms of the Heinz integral is equivalent to Def. 2.2.2 above [5]. The main results in this paper stem most directly from Def. 2.2.2.

The reason for interest in $d(F,P,\Theta_n)$ here is its relationship with the solutions of $F(X) = \Theta_n$ within the interior of P. Namely, the following theorem holds.

<u>2.2.4 Theorem</u>. (The Kronecker existence theorem; [1], pp. 467-468 [26], p. 16 or [8], p. 32). If $d(F,P,\theta_n) \neq 0$, then $F(X) = \theta_n$ has at least one solution in the interior of P.

<u>2.2.5 Remark</u>. When F is differentiable in P and $J(F)(X) = 0$ at each $X \in P$ with $F(X) = \theta_n$, then Theorem 2.2.4 can be strengthened to read: $d(F,P,\theta_n)$ is equal to the number of points $X \in P$ at which $F(X) = \theta_n$ and $J(F)(X) > 0$, minus the number of such points at which $J(F)(X) < 0$, where $J(F)$ is the Jacobian of F([23], pp. 33-38 and in particular, Theorem 1).

The above characterization is often taken to be the definition of $d(F,P,\theta_n)$ ([26], [23], etc.). This definition is then meaningfully generalized via the Heinz integral [26].

There are various approaches to computing $d(F,P,\theta_n)$. Some are quadrature schemes of which the Heinz integral [26] may form the basis. Also, $d(F,P,\theta_n)$ is represented by and numerically approximated via the Kronecker integral [25]; the Kronecker integral ([1], pp. 415-437) has close ties with facts used later.

<u>2.2.6 Formula</u>. (The Kronecker integral: [14]):

$$d(F,P,\theta_n) = \frac{1}{\Omega_n} \int_{X(u) \in b(P)} \frac{1}{\|F(X)\|^n} \Delta_n\left(F(X), \frac{\partial F}{\partial u_1}(X), \ldots, \frac{\partial F}{\partial u_{n-1}}(X)\right) du_1 \ldots du_{n-1}$$

Here Ω_n is the surface area of the unit n-sphere, and $U = (u_1,\ldots,u_{n-1})$, where $u_1,\ldots,u_{n-1}$ are the parameters for any continuous, one-to-one parametrization of b(P).

The following formula complements Formula 2.2.6.

2.2.7 Formula. In R^1, in view of Remark 2.2.5, the following formula is used to characterize $d(f,P,0)$, where $P = [a,b]$ ([30]):

$$d(f,P,0) = \frac{f(b)}{|f(b)|} - \frac{f(a)}{|f(a)|} = \frac{\operatorname{sgn}(f(b)) - \operatorname{sgn}(f(a))}{2} .$$

2.2.8 Remark. If the n-dimensional polygon P is embedded in a higher-dimensional region with a one-to-one parametrization π (i.e., $P = \pi\tilde{P}$, where $\tilde{P} \subset R^n$), and $F:P \to R^n$, then $d(F,P,\theta_n)$ is defined by:

$$d(F,P,\theta_n) = d(F\circ\pi,\ \pi^{-1}(P),\theta_n) \text{ provided } \theta_n \notin F(b(P)) .$$

2.2.9 Remark. The following two properties of $d(F,P,\theta_n)$ are useful:

(1) $d(F,P,\theta_n)$ depends only on values F assumes on $b(P)$.

(2) [14] If $\mathcal{D} = \bigcup_{i=1}^{k} P_i$, where each P_i is a union of polygons, the P_i are disjoint, and $F(X) \neq \theta_n$ for any $X \in b(P_i)$, then $d(F,\mathcal{D},\theta_n)$ can be defined by:

$$d(F,\mathcal{D},\theta_n) = \sum_{i=1}^{k} d(F,P_i,\theta_n) .$$

CHAPTER III

TWO CHARACTERIZATIONS OF $d(F,P,\Theta_n)$

The object of this chapter is to present characterizations of $d(F,P,\Theta_n)$ which can easily be used for machine computations. Such characterizations, given in terms of simplicial subdivisions of $b(P)$, hold when the simplexes in these subdivisions have small diameters. These smallness conditions are clarified in Section 3.1.

A recursion formula [30] is also derived in Section 3.1. This formula relates $d(F,P,\Theta_n)$, where P is n-dimensional, to $d(F_1,\beta^i_{n-1},\Theta_{n-1})$, where β^i_{n-1} are (n-1)-dimensional regions on the boundary of P, and F_1 has (n-1) components, all of which are restrictions of components of F to $b(P)$. In Section 3.2 and Section 3.3, this recursion formula is used to derive two characterizations of $d(F,P,\Theta_n)$. One of these (Theorem 3.2.10) expresses $d(F,P,\Theta_n)$ in terms of certain matrices whose entries are ±1, while the other (Theorem 3.3.1) gives $d(F,P,\Theta_n)$ in terms of determinants of these same matrices. The first characterization, proven in detail, will be seen in Chapter VI to be related to labeling schemes to determine Sperner simplexes.

Either characterization can be used to compute $d(F,P,\Theta_n)$. Because determinants are not involved in Theorem 3.2.10, the computations based on it should usually be the most efficient. However, Theorem 3.3.1 provides an additional method of determining the probability of correct results. This is discussed in Section 3.3.

3.1 A Recursion Relation

In all following work in this chapter, it is assumed that the domain of F is an n-dimensional polygon P or more generally that b(P) can be written as a sum of (n - 1)-simplexes, each of whose diameters is "small" (see [30] pp. 26-28). Just how small is small enough is indicated in the following definitions (ibid.).

3.1.1 Definition. If $P = \langle A,B\rangle \subset R$ and $f:R \to R$ is continuous, then $b(P) = B - A$ is sufficiently refined relative to f if $f(B) \neq 0$ and $f(A) \neq 0$.

3.1.2 Definition. Suppose $\mu > 1$, P_μ is a μ-dimensional polygon, and $F = (f_1,\ldots,f_\mu):P_\mu \to R^n$. Then $b(P_\mu)$ is sufficiently refined relative to F if and only if $b(P_\mu)$ is written as the union of a finite number of $(\mu - 1)$-dimensional regions $\beta^1_{\mu-1},\ldots,\beta^{\kappa_\mu}_{\mu-1}$, each consisting of unions of oriented simplexes and having the following properties:

(a) The interiors of the $\beta^i_{\mu-1}$ are disjoint and each $\beta^i_{\mu-1}$ is connected.

(b) At least one of the components of F, say f_{r_i}, does not vanish on $\beta^i_{\mu-1}$.

(c) If $f_{r_i} \neq 0$ on $\beta^i_{\mu-1}$, then $b(\beta^i_{\mu-1})$ is sufficiently refined relato F_{r_i}, where $F_{r_i} = (f_1,\ldots,f_{r_{i-1}},f_{r_i}+1,\ldots,f_\mu):b(P_\mu) \to R^{n-1}$.

Since properties (a), (b), and (c) of Definition 3.1.2 depend only on the signs of the components of F on b(P), we will say b(P) is sufficiently refined relative to Sgn (F) (infra. Definition 3.2.1) to mean b(P) is sufficiently refined relative to F.

It should be noted that the above assumptions, though somewhat weaker than the ones given by Stenger [30], are all that are necessary in his work (ibid.) and the theorems to follow here (cf. [21]).

It can be shown ([30] pp. 30-32) under mild assumptions on F that, if the diameters of the simplexes comprising $b(P_\mu)$ are sufficiently small, then $b(P_\mu)$ is sufficiently refined relative to Sgn (F).

Methods of subdividing $b(P_\mu)$ in order to produce sufficient refinement will be discussed in Chapter IV. Here, the following definition and theorem set down a recursion relation ([30] pp. 32-33) for computing $d(F,P_\mu,\Theta_\mu)$ provided $b(P_\mu)$ is sufficiently refined relative to F.

<u>3.1.3 Definition</u>. Let $\beta^i_{\mu-1}$ be as in Definition 3.1.2. Then corresponding to an integer $r \in [1,n]$ let $J^+(r)(J^-(r))$ consist of those integers $i \in [1,\kappa_\mu]$ for which $f_r > 0$ on $\beta^i_{\mu-1}(f_r < 0$ on $\beta^i_{\mu-1})$.

<u>3.1.4 Theorem ([16] p. 33) A Recursion Formula</u>. Suppose P_μ is a polygon in R^μ, $F = (f_1,\ldots,f_\mu):P_\mu \to R^\mu$ and $b(P_\mu)$ is sufficiently refined relative to F, with regions $\beta^1_{\mu-1},\ldots,\beta^\kappa_{\mu-1}$ as in Definition 3.1.6.

Then:

$$d(F,P_\mu,\Theta_\mu) = \sum_{i\in J^+(1)} d(F_1,\beta^i_{\mu-1},\Theta_{\mu-1})$$

where

$$F_1 = (f_2,\ldots,f_\mu):b(P_\mu) \to R^{\mu-1} .$$

<u>3.1.5 Proof of Theorem 3.1.4</u>. Without loss of generality it may be assumed that F is differentiable, for otherwise F may be approximated by a map, G, which is differentiable, for which $d(G,P_\mu,\Theta_\mu) = d(F,P_\mu,\Theta_\mu)$, and for which P_μ is sufficiently refined relative to G with the same regions $\beta^i_{\mu-1}$ (infra, also [26]: 6.2.1 and Lemma 3.1.6). Then let $X(u_1,\ldots,u_{\mu-1})$ represent a differentiable parametrization of $b(P_\mu)$ and let W be any point on $b(P_\mu)$ at which $F_1(W) = (f_2(W),\ldots, f_\mu(W)) = \Theta_{\mu-1}$. Then the following formula holds at such points W by expansion of the determinant in the left member along the first row:

$$\text{(i)}\ \Delta_\mu\left(F,\frac{\partial F}{\partial u_1},\ldots,\frac{\partial F}{\partial u_{\mu-1}}\right) = f_1\Delta_{\mu-1}\left(\frac{\partial F_1}{\partial u_1},\ldots,\frac{\partial F_1}{\partial u_{\mu-1}}\right) .$$

With the point Y of Definition 2.2.1 chosen to be the intersection of the positive first coordinate axis of R^μ with the unit μ-sphere, $d(F,P_\mu,\Theta_\mu)$ equals the number of points W as above for which a small neighborhood gets mapped in a one-to-one fashion onto a neighborhood containing Y with a preservation of orientation, minus the number of such points W for which there is a reversal of orientation (see [23]: §5). However, unless the left side of (i) is zero, the argument in [1], p. 466, shows that the sign of the left side of (i) gives the relative orientation of any such small neighborhood on the unit sphere. If the determinant is zero at one or more such W we can replace F by an approximation G so $b(P_\mu)$ is still sufficiently refined relative to G, $d(F,P_\mu,\Theta_\mu) = d(G,P_\mu,\Theta_\mu)$, and $d(F_1,\beta^i_{\mu-1},\Theta_{\mu-1}) = d(G_1,\beta^i_{\mu-1},\Theta_{\mu-1})$, but the determinant $\Delta_\mu(G,\frac{\partial G}{u_1},\ldots,\frac{\partial G}{u_{n-1}})$ does not vanish at such W for the mapping G. Hence, without loss of generality the sum of the sign of the left-hand side of (i) over all W is $d(F,P_\mu,\Theta_\mu)$.

Note, however, that $f_1 > 0 \Rightarrow$ W must lie in some $\beta^i_{\mu-1}$ with $i \in J^+(1)$. Finally observe that, excluding the factor of $f_1(W)$, the right member of (i) is simply the Jacobian of F_1 with respect to the parameters $u_1,\ldots,u_{\mu-1}$, and that the points W are simply those points in $\beta^i_{\mu-1}$ at which F_1 vanishes. Hence, by Remark 2.2.5 (the degree in terms of the Jacobian), the sum over all W of the sign of right-hand side is $\sum_{i\in J^+(1)} d(F_1,\beta^i_{\mu-1},\Theta_{\mu-1})$. This combined with the preceding paragraph establishes the formula.

<u>3.1.6 Lemma</u>. Suppose that $F, \mathcal{D} \subset R^n \to R^n$, is continuous, where $\mathcal{D}$ is some compact set in R^n. Suppose further that $b(\mathcal{D})$ is sufficiently refined relative to F, with regions $\beta^1_{n-1},\ldots,\beta^k_{n-1}$ and

sets $J^{\pm}(s)$, $s = 1,\ldots n$ in the definition of sufficient refinement. Then there exists an ε such that $b(\mathcal{D})$ is sufficiently relative to G if $\|F - G\| < \varepsilon$, where the $\beta_{n-1},\ldots\beta^{\ell}_{n-1}$ and $J^{\pm}(s)$, $s = 1,\ldots,n$ are the same as those for F.

<u>3.1.7 Proof of Lemma 3.1.6</u>. The proof will proceed by induction on n. If $n = 1$ the theorem is trivial. Let us then begin by assuming that the theorem is true for $(n - 1)$-dimensional spaces with $n \geq 2$; further suppose that $F: \mathcal{D} \to R^n$, where $\mathcal{D} \subset R^n$, and $b(\mathcal{D})$ is sufficiently refined relative to F with the sets $\beta^1_{n-1},\ldots\beta^{\kappa}_{n-1}$. Take an arbitrary r_i such that $f_{r_i} \neq 0$ on β^i_{n-1}. From here, choose ε'_{i,r_i} such that $\|F_{r_i} - G\| < \varepsilon'_{i,r_i}$ implies β^i_{n-1} is sufficiently refined relative to G, with the same sets as those for F_{r_i}; set $\varepsilon = \min\limits_{i,r_i} \varepsilon_{i,r_i}$. For each such i and r_i, set $\delta_{i,r_i} = \min\limits_{X\epsilon\beta^i_{n-1}} \|f_{r_i}(X)\|$ and set $\varepsilon'' = 1/2\min\limits_{i,r_i} \delta_{i,r_i}$, so that $\|F - G\| < \varepsilon'' \Rightarrow g_{r_i}(X) > 0$ on β^i_{n-i}. Set $\varepsilon = \min\{\varepsilon',\varepsilon''\}$. Then $\|F - G\| < \varepsilon \Rightarrow \|F - G\| < \varepsilon''$, and $\|F_{r_i} - G_{r_i}\| < \varepsilon'_{i,r_i}$ for every i and r_i. Hence, by definition of sufficient refinement such G are sufficiently refined relative to F. Furthermore, a simple observation shows that the β^i_{n-1} for G can be chosen to be the same as those for F. The $J^{\pm}(S)$ are then the same for both F and G. The theorem is thus proven by induction on n.

3.2 A Characterization of $d(F,P,\theta_n)$ in Terms of Certain Matrices

In this section, we present a characterization in terms of matrices whose entries are ± 1. These matrices are formed by taking the algebraic sign of components of F evaluated at points on b(P). Hence, we begin by defining Sgn(F).

3.2.1 Definition. If $F(X) = (f_1(X),\ldots,f_n(X))$, $P \to R^n$, then Sgn(F) is defined by:

$$\mathrm{Sgn}(F)(X) = (\mathrm{sgn}(f_1(X)),\ldots,\mathrm{sgn}(F_n(X)))$$

where

$$\mathrm{sgn}(y) = \left\{\begin{matrix} 1 \text{ if } y \geq 0 \\ -1 \text{ if } y < 0 \end{matrix}\right\} \text{ for } y \in R.$$

3.2.2 Remark. The fact that sgn y has only two values allows one to economize on computer storage. Also, the results in [30] and this paper are all true with Sgn(F) defined as above.

We now define the main terms appearing in our characterization.

3.2.3 Definition. If $S = \langle X_1,\ldots,X_n\rangle$ is an $(n - 1)$-simplex in R^n and $F:S \to R^n$, then the range simplex associated with S and F, denoted $\mathcal{A}(S,F)$ is the $n \times n$ matrix whose i^{th} row is $\mathrm{Sgn}(F(X_i))$. That is, if $S = \langle X_1,\ldots,X_n\rangle$ is an $(n-1)$-simplex in R^n and $F = (f_1,\ldots,f_n):R^n \to R^n$, then:

$$\mathcal{A}(S,F) = \begin{pmatrix} \mathrm{Sgn}(F(X_1)) \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{Sgn}(F(X_n)) \end{pmatrix}$$

<u>3.2.4 Definition</u>. If S, F, and $\mathcal{A}(S,F)$ are as above, then $\mathcal{A}(S,F)$ is termed useable if one of the following two conditions holds:

(a) $\mathcal{A}(S,F) = (a_{i,j})$ has only $+1$'s on and below the main diagonal and only (-1)'s in the $a_{i,i+1}$ positions, for $i = 1,2,\ldots,n-1$.

(b) $\mathcal{A}(S,F)$ can be put into the form indicated in (a) by a suitable permutation of its rows.

<u>3.2.5 Remark</u>. It is often less cumbersome to formulate the definition of useable $\mathcal{A}(S,F)$ in terms of labelings. To the k^{th} row of $\mathcal{A}(S,F)$ (for $k = 1,\ldots,n$) we assign a label $\ell_k \in [1,n]$ by letting $\ell_k = \ell - 1$ if the first -1 (reading from left to right) in the k^{th} row occurs in the ℓ^{th} column; if there are no -1's in the k^{th} row, we set $\ell_k = n$. It is easy to see that $\mathcal{A}(S,F)$ is useable if and only if $\{\ell_1,\ldots,\ell_n\} = \{1,\ldots,n\}$. Furthermore, if $\mathcal{A}(S,F)$ is useable, then $\mathrm{Par}(\mathcal{A}(S,F)) = 1$ if and only if the permutation required to put the

sequence $\ell_1, \ldots \ell_n$ in natural order is even, and Par $(\mathcal{R}(S,F)) = 1$ if and only if that permutation is odd.

<u>3.2.6 Definition</u>. If $\mathcal{R}(S,F)$ is a range simplex, then $\mathcal{R}(S,F)$ is said to have <u>positive parity</u> or parity +1 if the permuta- required to put $\mathcal{R}(S,F)$ into form (a) of Definition 3.2.2 is even; if that permutation is odd, then $\mathcal{R}(S,F)$ is said to have <u>negative parity</u> or <u>parity -1</u>. If $\mathcal{R}(S,F)$ is not useable, then $\mathcal{R}(S,F)$ has <u>parity 0</u>.

For any range simplex $\mathcal{R}(S,F)$, the parity of $\mathcal{R}(S,F)$ is denoted by Par$(\mathcal{R}(S,F))$.

The characterization in this section deals with the matrices $\mathcal{R}(S,F)$. To enable us to prove our characterization by induction we next define a number associated with submatrices of such $\mathcal{R}(S,F)$.

Suppose that the first column of $\mathcal{R}(S,F)$ has only +1's in it, and set $F_1 = (f_2, \ldots, f_n)$. Also consider $b(S) = \sum_{k=1}^{n} (-1)^{k-1} T_k$, where $T_k = \langle X_1, \ldots, \hat{X}_k, \ldots, X_n \rangle$. Then the range of simplexes $\mathcal{R}(T_k, F_1)$ can be gotten from $\mathcal{R}(S,F)$ by deleting the first row and k^{th} column of $\mathcal{R}(S,F)$. However, T_k occurs in the sum for $b(S)$ with an orientation of $(-1)^{k-1}$, and if the orientation of T is changed, then Par$(\mathcal{R}(T,F_1))$ changes sign. With this in mind we make the following definition.

<u>3.2.7 Definition</u>. The <u>net sum $\sigma(S)$ of the parities of the range simplexes associated with simplexes from $b(S)$ and F_1</u> is given by:

$$\sigma(S) = \sum_{k=1}^{n}(-1)^{k-1}\mathrm{Par}(\mathcal{R}(T_k,F_1)) .$$

For example, if

$$\mathcal{R}(S,F) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

then the net sum is

$$\sigma(S) = (-1)^{1-1}\mathrm{Par}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} + (-1)^{2-1}\mathrm{Par}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$+ (-1)^{3-1}\mathrm{Par}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = 0 + (-1)(-1) + 0 = 1$$

Dealing with the above notions, the following lemma will be used to prove Theorem 3.2.10 (the main characterization).

<u>3.2.8 Lemma</u>. Suppose S,F, $\mathcal{R}(S,F)$, and $\sigma(S)$ are as above, with $n \geq 3$. If the second column of $\mathcal{R}(S,F)$ also contains only +1's, then $\sigma(S) = 0$.

<u>3.2.9 Proof of Lemma 3.2.8</u>. We will think of $\mathrm{Par}(\mathcal{R}(S,F))$ in terms of the labels $\ell_1,\ldots,\ell_n$ assigned to the rows of $\mathcal{R}(S,F)$ (cf Remark 3.2.5).

In order for $\sigma(S)$ to be nonzero, $\mathrm{Par}(\mathcal{R}(T_k,F_1)) \neq 0$ for some T_k, where T_k is as above. But this can happen if and only if $\{2,\ldots,n\} \subset \{\ell_1,\ldots,\ell_n\}$ from the definition of $\ell_1,\ldots,\ell_n$ and the definition of $\mathrm{Par}(\mathcal{R}(T_k,F_1))$, so assume $\{2,\ldots,n\} \subset \{\ell_1,\ldots,\ell_n\}$. But $1 \notin \{\ell_1,\ldots,\ell_n\}$ by the assumption on the second row of $\mathcal{R}(S,F)$. Hence, there are $k,m \in \{1,\ldots,n\}$ such that $\ell_k = \ell_m$ and such that all of the other ℓ_j's are distinct. From this we deduce:

$$\sigma(S) = (-1)^{k-1}\mathrm{Par}(\mathcal{R}(T_k,F_1)) + (-1)^{m-1}\mathrm{Par}(\mathcal{R}(T_m,F_1)) .$$

It will now be shown that $(-1)^{k-1}\mathrm{Par}(\mathcal{R}(T_k,F_1)) + (-1)^{m-1}\mathrm{Par}(\mathcal{R}(T_m,F_1)) = 0$. Consider these two possibilities for k and m:

(a) k and m are both odd or both even.

(b) k and m have opposite parities.

If case (a) occurs, then $(-1)^{k-1} = (-1)^{m-1}$. Suppose without loss of generality that $k < m$ so that $\mathrm{Par}(\mathcal{R}(T_k,F_1))$ is positive if and only if the parity of the permutation required to put $\ell_1,\ldots,\ell_{k-1},\ell_{k+1},\ldots,\ell_m,\ldots,\ell_n$ in the natural order is even, and $\mathrm{Par}(\mathcal{R}(T_m,F_1))$ is positive if and only if the permutation required to put the sequence $\ell_1,\ldots,\ell_k,\ldots,\ell_{m-1},\ell_{m+1},\ldots,\ell_n$ in the natural order is even. Since, $1_k = 1_m$, however, we can get the sequence $\ell_1,\dots,\ell_k,\dots,\ell_{m-1},\ell_{m+1},\dots,\ell_n$ from $\ell_1,\dots,\ell_{k-1},\ell_{k+1},\dots,\ell_m,\dots,\ell_n$ by the (k - m)-cycle: $(\ell_{k+1},\dots,\ell_n)$. If k - m is even, then the parity of this permutation is odd, thus showing that $\mathrm{Par}(\mathcal{R}(T_k,F_1)) = -\mathrm{Par}(\mathcal{R}(T_m,F_1))$. Hence, when k and m have the same parity we have:

$$\sigma(S) = (-1)^{k-1}\,\mathrm{Par}(\mathcal{R}(T_k,F_1)) + \mathrm{Par}(\mathcal{R}(T_m,F_1)) = 0 .$$

If we follow the same argument when k and m have opposite parities, we get:

$$\sigma(S) = \{(-1)^{k-1} + (-1)^{m-1}\}\,\mathrm{Par}(\mathcal{R}(T_k,F_1)) = 0 .$$

Thus Lemma 3.2.8 is proven.///

Suppose now that $P \subset R^n$ is a polygon, and that $F:P \to R^n$ does not vanish on b(P). Then one need only examine the useable simplexes produced from the sufficient refinement of b(P) in order to determine $d(F,P,\theta_n)$. The following theorem (our main characterization) makes this fact explicit.

<u>3.2.10 Theorem (The Parity Theorem)</u>. Suppose P is an n-dimensional polygon contained in R^n for some $n \geq 2$. Suppose further that $\mathcal{S}$ is a finite set of (n - 1)-simplexes such that $\bigcup_{S\in\mathcal{S}} S = b(P)$,

the members of S have disjoint interiors, and the simplexes in $\mathcal{S}$ make b(P) sufficiently refined relative to Sgn(F). Then:

$$d(F,P,\theta_n) = \sum_{S\in\mathcal{S}} \mathrm{Par}(\mathcal{R}(S,F)) .$$

Otherwise stated, $d(F,P,\theta_n)$ equals the number of useable simplexes from members of $\mathcal{S}$ with positive parity minus the number of such useable simplexes with negative parity.

<u>3.2.11 Proof of Theorem 3.2.10</u>. The proof will proceed by induction on n. First assume that P is a 2-dimensional polygon, that $F = (f_1,f_2)\colon P \to R^2$ is continuous, and that $\mathcal{S}$ is a set of 1-simplexes whose union is b(P) such that $\mathcal{S}$ causes b(P) to be sufficiently refined relative to Sgn(F). Then the conclusion of Theorem 3.1.4 holds with $\mu = 2$, so that

$$\text{(i) } d(F,P,\theta_n) = \sum_{i\in J^+(1)} d(f_2,\beta_1^i,0)$$

where the β_1^i are as in Definition 3.1.2 and $J^+(1)$ is as in Definition 3.1.3. However, when n = 2 each β_1^i is a sum of $S_{j,i} \in \mathcal{S}$ where each $S_{j,i}$ is a line segment in R^2 for $i \le j \le m_i$ and $i \in J^+(1)$. We write $S_{j,i} = \langle A_{j,i},B_{j,i}\rangle$ hereafter. Hence, in view of Formula 2.2.7 we have

$$\text{(ii) } d(F_2,\beta_1^i,0) = \sum_{j=1}^{m_i}[\mathrm{sgn}(f(B_{j,i})) - \mathrm{sgn}(f(A_{j,i})] .$$

Combining (i) and (ii) now gives

$$\text{(iii) } d(F,P,\Theta_2) = \sum_{i\in J^+(1)} \sum_{j=1}^{m_i} [\operatorname{sgn}(f(B_{j,i})) - \operatorname{sgn}(f(A_{j,i})]$$

It will be shown that each $[\operatorname{sgn}(f(B_{j,i})) - \operatorname{sgn}(f(A_{j,i})]$ in (iii) can be replaced by $\operatorname{Par}(\mathcal{R}(S_{j,i},F))$ without affecting the sum. Hereafter, unless subscripts are important, S will be written for $S_{j,i}$. First, it will be shown that without loss of generality $S \subseteq \beta_1^i$ for some $i \in J^+(1)$ if and only if the first column of $\mathcal{R}(S,F)$ is $(1,1)^T$. Clearly, if $S \subseteq \beta_1^i$ then the first column of $\mathcal{R}(S,F)$ is $(1,1)^T$. For the converse, suppose that the first column of $\mathcal{R}(S,F)$ is $(1,1)^T$ but $S \nsubseteq \beta_1^i$ for any $i \in J^+(1)$. Then by the sufficient refinement hypothesis, $f_2 > 0$ on S or $f_2 < 0$ on S, so that $[\operatorname{sgn}(f(B_{j,i})) - \operatorname{sgn}(f(A_{j,i})] = \operatorname{Par}(\mathcal{R}(S,F)) = 0$. In this case, S may be included in the sum in (iii).

There are four matices $\mathcal{R}(S,F)$ whose first column is $(1,1)^T$. These are:

(a) $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ (b) $\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$

(c) $\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ (d) $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

These four cases are checked individually to prove the theorem for $n = 2$.

Suppose now that the theorem is true with $(n - 1)$ replacing n, where $n > 2$; also let $P \subset R^n$ be an n-dimensional polygon, let $F = (f_1, \ldots, f_n): P \to R^n$ be continuous, and let $b(P) = \sum_{S \in \mathcal{S}} S$ be sufficiently refined relative to $\mathrm{Sgn}(F)$. Then the conclusion of Theorem 3.1.4 again holds with n replacing μ, so that

$$\text{(iv)}\quad d(F,P,\theta_n) = \sum_{i \in J^+(1)} d(F_1, \beta^i_{n-1}, \theta_{n-1})$$

where $J^+(1)$ is as in Definition 3.1.3 and the β^i_{n-1} are as in Definition 3.1.2.

But, by the induction hypothesis, part (c) of the definition of sufficient refinment, and the definition of $\sigma(S)$, we have

$$\text{(v)}\quad d(F_1, \beta^i_{n-1}, \theta_{n-1}) = \sum_{T \subseteq b(\beta^i_{n-1})} \mathrm{Par}(\mathcal{R}(T,F_1)) = \sum_{S \subseteq \beta^i_{n-1}} \sigma(s).$$

for $i \in J$.

It is first verified (infra; Lemma 3.2.10) that S need be included in the right member of (v) if and only if $\mathcal{R}(S,F)$ is useable. Then we apply the fact that if $\mathcal{R}(S,F)$ is useable, $\sigma(S) = \mathrm{Par}(\mathcal{R}(S,F))$ (infra, Lemma 3.2.12). Hence,

(vi) $\sum_{S \subseteq \beta^i_{n-1}} \sigma(S) = \sum_{S \text{ useable}} \mathrm{Par}(\mathcal{R}(S,F)) = \sum_{S \in \mathcal{S}} \mathrm{Par}(\mathcal{R}(S,F))$

Combining (iv), (v), and (vi) now gives the result of the theorem.///

<u>3.2.12 Lemma</u>. Suppose $b(\mathcal{D})$ is sufficiently refined relative to $\mathrm{Sgn}(F)$, $S \subset b(\mathcal{D})$, and $\mathcal{R}(S,F)$ is useable. Then $S \subseteq \beta^i_{n-1}$ for some $i \in J^+(1)$. Also, suppose $S \subseteq \beta^i_{n-1}$ for some $i \in J^+(1)$ and $\mathcal{R}(S,F)$ is not useable. Then $\sigma(S) = 0$.

<u>3.2.13 Proof of Lemma 3.2.12</u>. If $\mathcal{R}(S,F)$ is useable, then the first column of $\mathcal{R}(S,F)$ is the only column which does not contain both 1's and -1's. But, by the sufficient refinement hypothesis, at least one component of F does not vanish on S. This forces $S \subseteq \beta^i_{n-1}$ for some $i \in J^+(1)$.

Now suppose $S \subseteq \beta^i_{n-1}$ for some $i \in J^+(1)$ and $\mathcal{R}(S,F)$ is not useable. Then there are zero, one, or more than one -1 in the second column of $\mathcal{R}(S,F)$. If there are no -1's in the second column, then $\sigma(S) = 0$ by Lemma 3.2.8.

Finally, if there are one or more zeros in the second column of $\mathcal{R}(S,F)$, and $\mathcal{R}(S,F)$ is not useable, then there is a $j \in \{2, ,\ldots,n\}$ such that $j \in \{\ell_1,\ldots,\ell_n\}$. In that case, $\mathrm{Par}(\mathcal{R}(T_k,F_1)) = 0$ for every $k \in \{1,\ldots,n\}$ so $\sigma(S) = 0$.

<u>3.2.14 Lemma</u>. Suppose $\mathcal{R}(S,F)$ is useable. Then $\sigma(S) = \mathrm{Par}(\mathcal{R}(S,F))$.

<u>3.2.15 Proof of Lemma 3.2.14</u>. Suppose $\mathcal{R}(S,F)$ is useable, and set $b(S) = \Sigma_{k=1}^{n} (-1)^{k-1}T_k$ as in Definition 3.2.7. Then, since T_k would not have any -1's in its first column, if $\mathrm{Par}(\mathcal{R}(T_k,F_1)) \neq 0$, $\mathrm{Par}(\mathcal{R}(T_k,F_1)) = 0$ for all but one k, e.g., k_0. Let P_1 be the permutation on the numbers $1,\ldots,n$ required to put the k_0 th row of $\mathcal{R}(S,F) = \mathcal{R}(\langle X_1,\ldots,X_n\rangle,F)$ into the first position (leaving the other rows fixed relative to each other) and let P_2 be that permutation on $1,2,\ldots,k_0-1,k_0+1,\ldots,n$ required to put $\mathcal{R}(\langle X_1,\ldots,\hat{X}_{k_0},\ldots,X_n\rangle,F_1)$ into form (a) of Definition 3.2.4. Similarly, let P_3 be the permutation on $1,\ldots,n$ required to put $\mathcal{R}(S,F)$ into form (a) of Definition 3.2.4. Then, thinking of P_2 as a permutation on n objects which leaves the first object fixed, we have

$$\text{(i)}\ P_3 = P_2P_1$$

However, $P_1 = (1,2,\ldots,k_0-1,k_0)$ is a k_0-cycle, so its parity is the parity of the integer $k_0 - 1$. Hence:

$$\text{(ii)}\ (-1)^{k_0-1}\mathrm{Par}(\mathcal{R}(T_{k_0},F_1)) = \mathrm{Par}(\mathcal{R}(S,F))$$

However, since $(-1)^{k_0-1}\mathrm{Par}(\mathcal{R}(T_{k_0},F_1))$ was the only nonzero term in the sum for $\sigma(S)$, the conclusion of the Lemma follows from (ii).///

3.3 A Characterization in Terms of Volumes and Determinants, and Analysis of This Characterization

Stenger [30] combines Theorem 3.1.4 with other considerations to produce the characterization in Theorem 3.3.1, presented below.

3.3.1 Theorem. Suppose $\mathcal{D} \subset R^n$ is an n-dimensional polygon, and that F: $\mathcal{D} \to R^n$ is continuous, with $F(X) \neq \theta_n$ for any $X \in b(\mathcal{D})$. Suppose further that $\mathcal{S} = \{S\}_{S \in \mathcal{S}}$ is a finite set of (n - 1)-simplexes such that $b(\mathcal{D}) = \sum_{S \in \mathcal{S}} S$ and that with $\mathcal{S}$, $b(\mathcal{D})$ is sufficiently refined relative to Sgn(F). Then the following formula is true:

$$d(F, \mathcal{D}, \theta_n) = \frac{1}{2^n n!} \sum_{S \in \mathcal{S}} \det(\mathcal{R}(S,F))$$

3.3.2 Remark. In [30], sgn(y) is defined by:

$$\operatorname{sgn}(y) = \left\{ \begin{matrix} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{matrix} \right\}.$$

As was mentioned earlier, however, one can replace this definition by Definition 3.2.1 without altering the truth of Theorem 3.3.1.

Calculating the above determinants with Gaussian elimination requires more computations than deciding whether a range simplex is useable or not (Def. 3.2.4). Nonetheless, for $n = 2$ and $n = 3$, the difference in computation times is not very large. Moreover, when one divides by $2^n n!$, one knows that sufficient refinement has not been attained if the result is not an integer. Furthermore, the determinants in Theorem 3.3.1 have magnitudes of at most $2^{(n^2-n)/2}$, so that this device can be used to guess $d(F, \mathcal{B}, \theta_n)$ with a high probability of correctness, even where sufficient refinement has not been attained.

<u>3.3.3 Theorem</u>. Let $n \geq 2$. If Δ_n is an n-th order determinant, the entries of whose matrix consist of only +1 and -1, then the maximum value of $|\Delta_n|$ is $2^{(n^2-n)/2}$.

<u>3.3.4 Proof of Theorem 3.3.3</u>. Apply the Gaussian elimination process to the rows of the matrix to put it in triangular form. Then no element of the matrix after the k-th stage of the operation has a magnitude larger than 2^k. Since the first through k-th rows remain unchanged during the k-th operation, and since the process triangularizes the matrix in $(n - 1)$ steps, the diagonal element in the k-th row of the triangularized matrix has a magnitude less than or equal to 2^{k-1} for $k = 1, \ldots, n$. Hence

$$|\Delta_n| \leq \Pi_{k=0}^{n-1} 2^k = 2^{\Sigma_{k=0}^{n-1} k} = 2^{(n^2-n)/2}.$$

CHAPTER IV

ALGORITHMS

In the previous chapter, it was seen that $d(F,P,\theta_n)$ could be calculated in terms of simplexes comprising $b(P)$, provided the diameters of those simplexes were "small enough." In order to implement the theorems involved, one desires an iterative method of "subdividing" the $(n-1)$-simplexes of $b(P)$ so that the diameters of the resulting $(n-1)$-simplexes become small.

One such procedure is a generalization of bisection of line segments in R^1. To carry out this procedure on an $(n-1)$-simplex S, we find the longest line segment to be formed by taking the vertices of S two at a time. The midpoint of this line segment is then used to form two new $(n-1)$-simplexes. An illustration for $(n - 1) = 2$ appears in Fig. 4.1. There the longest side of $\langle A,B,C\rangle$ is $\langle A,C\rangle$, which has midpoint, M. The two new triangles $\langle A,B,M\rangle$ and $\langle M,B,C\rangle$ are formed by replacing C by M and A by M, respectively.

When bisections are carried out, too many $(n-1)$-simplexes may be produced to be stored effectively in the machine. Furthermore, a casual approach easily results in repeated evaluation of F at the same points (infra), and may also result in extra work to produce a uniformly fine subdivision of $b(P)$, even though only some of the simplexes in the subdivision need have small diameters to assure

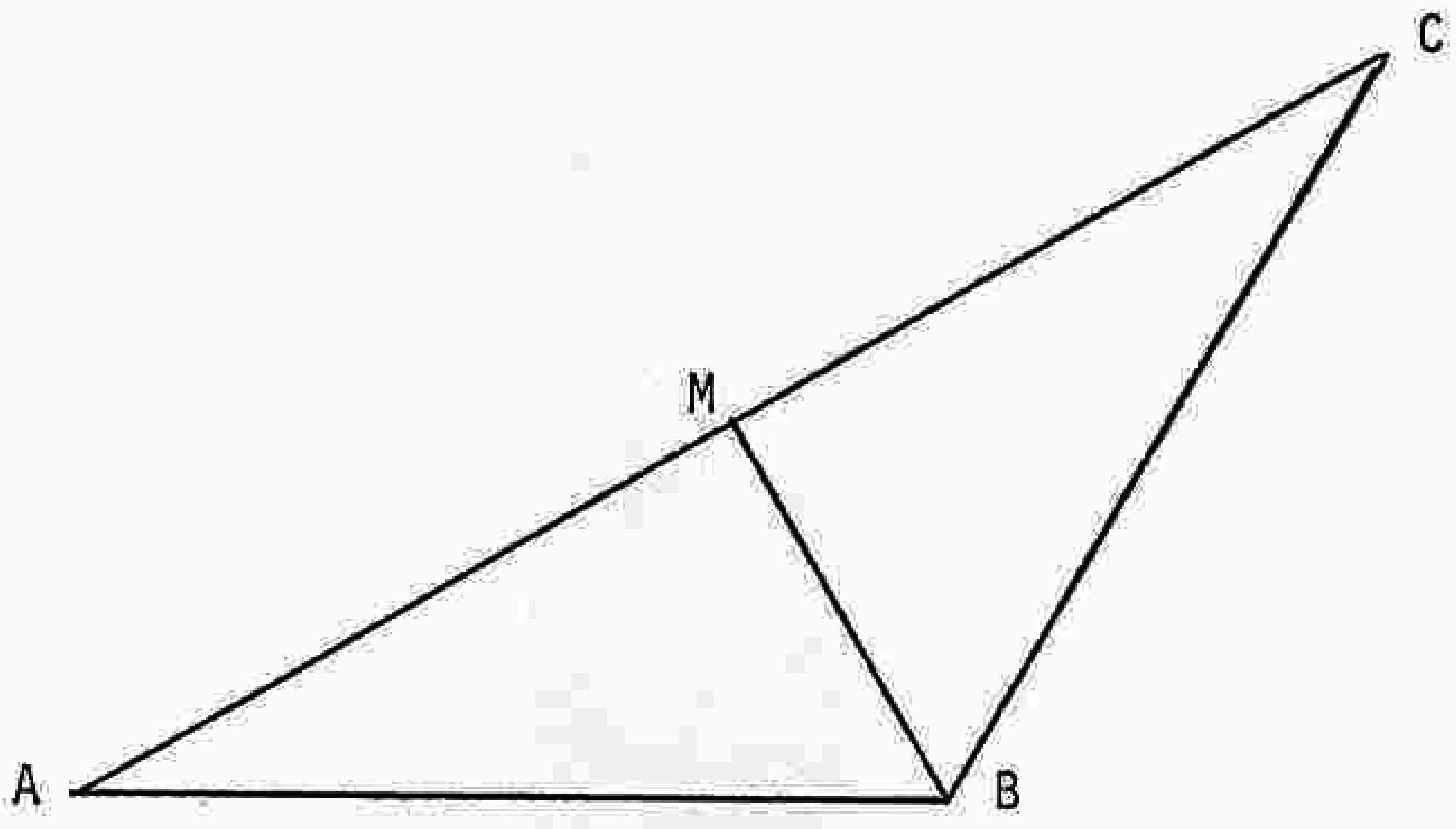


FIGURE 4.1

A SIMPLE ILLUSTRATION OF BISECTION WHEN (n-1)=2 sufficient refinement. For these reasons we will introduce an address scheme for labeling simplexes in subdivisions of b(P) produced by bisection. This address scheme depends on a one-to-one correspondence between the simplexes in the subdivisions and the nodes in a binary search tree, where each node is labeled by an ordered pair of integers.

Necessary concepts about binary trees and the address scheme are given in Section 4.2.

The remainder of the chapter is devoted to presentation and illustration of machine algorithms to compute $d(F,P,\theta_n)$. In Section 4.3, a simple algorithm [30] not requiring use of the address scheme is given. In Section 4.4, modification using the address scheme is explained. In Section 4.5, the action of the algorithm in Section 4.4 is illustrated on a two-dimensional example. In Section 4.6, further modifications of the algorithm in Section 4.4 are given. These modifications will improve the efficiency in special cases, such as when the dimension of the space is 2 or 3, or when the computational results are not going to be used in a generalized method of bisection in R^n. (This generalized method of bisection, used to calculate roots of $F(X) = \theta_n$, will be explained in Chapter V; the generalized method of bisection in R^n is not to be confused with the bisection of the (n-1)-simplexes in order to subdivide b(P), although the two procedures are related.)

4.1 Subdivisions of b(P)

4.1.1 Definition. Suppose $S = \langle X_1, \ldots, X_n \rangle$ is an (n-1)-simplex in R^n. Then a *simple subdivision* of S is any pair of simplexes $\{ S_1, S_2 \}$ such that, for some k and m between 1 and n and some $A \in \langle X_k, X_m \rangle$:

$$S_1 = \langle X_1, \ldots, X_{k-1}, A, X_{k+1}, \ldots, X_m, \ldots, X_n \rangle \text{ and}$$

$$S_2 = \langle X_1, \ldots, X_k, \ldots, X_{m-1}, A, X_{m+1}, \ldots, X_n \rangle .$$

4.1.2 Remark. It follows from Definition 4.1.1 that $S = S_1 + S_2$ if $\{S_1, S_2\}$ is a simple subdivision of S.

4.1.3 Definition. Suppose that $\mathcal{S} = \{S_1\}$, $\mathcal{S}_1$, $\mathcal{S}_2, \ldots, \mathcal{S}_\kappa$ are sets of simplexes such that for each simplex $T \in \mathcal{S}_j$ with $j < \kappa$, either $T \in \mathcal{S}_{j+1}$ or there is a pair of simplexes $\{U, V\} \subseteq \mathcal{S}_{j+1}$ such that $\{U, V\}$ is a simple subdivision of T. Then $\mathcal{S}_{j+1}$ is called a *subdivision* of $\mathcal{S}_i$, for $i < j + 1$. Any process of generating subdivisions of $\mathcal{S}$ is also termed *subdivision of S*.

4.1.4 Definition. If $P = \Sigma_{i=1}^{m} S_i$ is a polygon, then subdivisions of P are defined in the natural way as unions of subdivisions of the component simplexes of P.

<u>4.1.5 Definition</u>. If $\{S_1, S_2\}$ is a simple subdivision of $\{S\} = \{\langle X_1, \ldots, X_n\rangle\}$, then $\{S_1, S_2\}$ is termed a bisection of $\{S\}$, or for brevity, a bisection of S, if and only if the point A of Definition 4.1.1 is the midpoint $(X_k + X_m)/2$ of the longest one-dimensional side $\langle X_k, X_m\rangle$ formed from the vertices of S (here length of $\langle X_i, X_j\rangle$ can be taken to be the square of the ℓ_2 distance $\|X_i - X_j\|_2^2$). Subdivisions in which the component simple subdivisions are only bisections will also be called bisections.

It is fairly easy to compute the vertices of the component simplexes of a bisection of S. Due to the following theorem and the fact that bisections are easy to perform, the algorithms developed in this work use bisection processes.

<u>4.1.6 Theorem</u>. Let S be an (n-1)-simplex. Let $\mathcal{S}_1 = \{S\}$, and for $j > 1$, let $\mathcal{S}_j$ be the set of (n-1)-simplexes forming bisections of elements of $\mathcal{S}_{j-1}$. Then, as $j \to \infty$, the diameters of the elements of $\mathcal{S}_j$ tend to 0.

The proof of Theorem 4.1.6 will appear later.

<u>4.2 Trees</u>

Here, a special type of binary tree is defined, and the address scheme for nodes in such trees and simplexes in a subdivision is presented.

<u>4.2.1 Definition</u>. A simplex tree (binary tree) $\mathcal{J}$ is a finite, partially-ordered set of points (or nodes) with the following properties:

(a) Each point of $\mathcal{J}$ either precedes no points or is the immediate predecessor of precisely two points. If A precedes B, we write $A < B$.

(b) There is a unique point $S_0 \in \mathcal{J}$, called the <u>original point</u> (at which the tree is rooted), such that no point of $\mathcal{J}$ precedes S_0, and all other points of $\mathcal{J}$ follow S_0.

(c) If S_1 and S_2 follow a point S and there are no points T_1 or T_2 such that $S < T_1 < S_1$ or $S < T_2 < S_2$, then the pair $\{S_1, S_2\}$ is linearly ordered. The first element of the pair will be called the <u>lower point from S</u> and the second element will be called the <u>upper point from S</u>.

(d) Each point of $\mathcal{J}$ other than the original point is preceded by precisely one point.

<u>4.2.2 Definition</u>. Points of $\mathcal{J}$ which do not precede any other points of $\mathcal{J}$ are called the <u>leaves of $\mathcal{J}$</u>.

If points are drawn in columns in R^2 and connected by lines, and if the order is specified by letting points to the left precede points to the right, then one can draw simplex trees. Examples are given in Fig. 4.2 and Fig. 4.3.

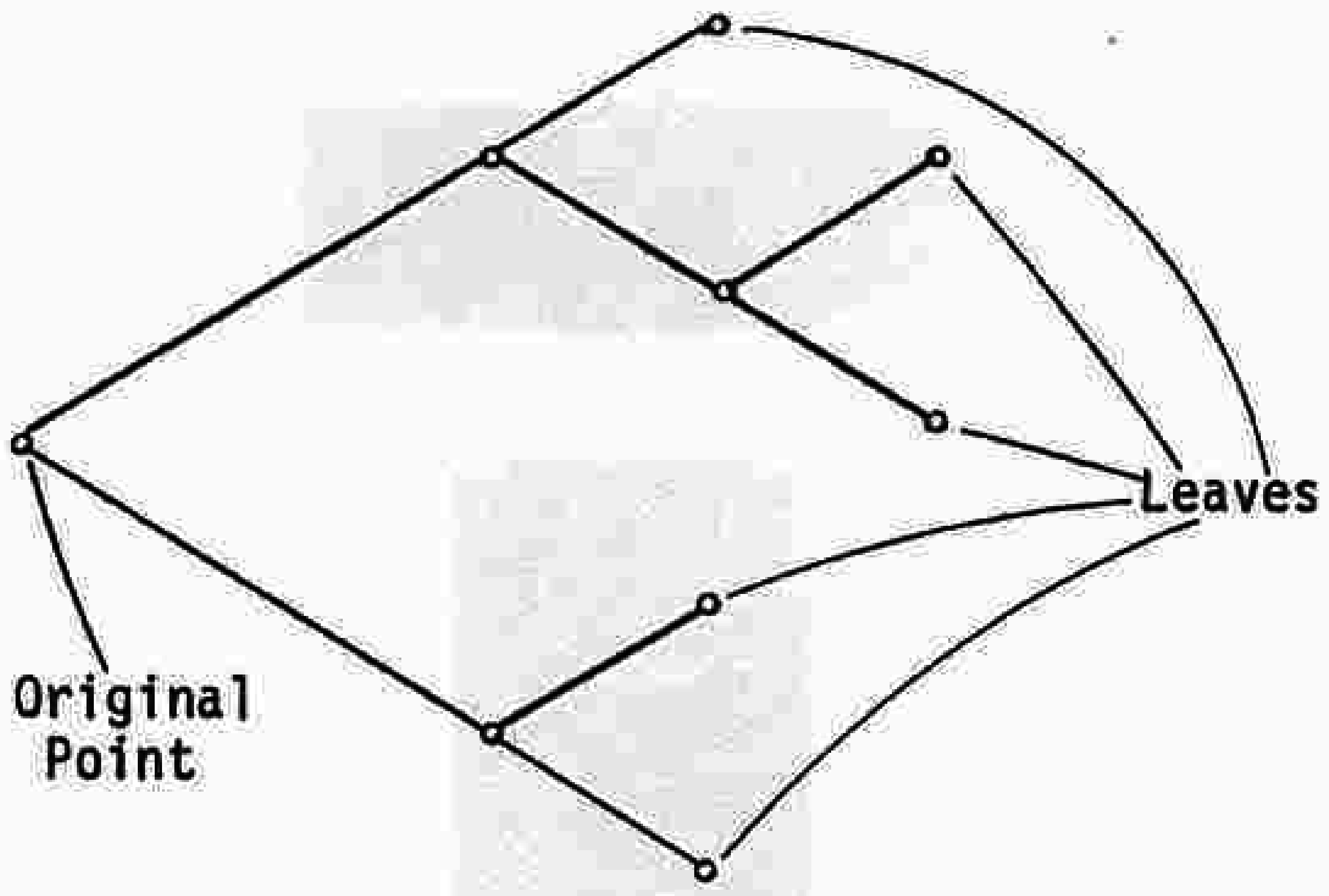


FIGURE 4.2

A 9-POINT TREE

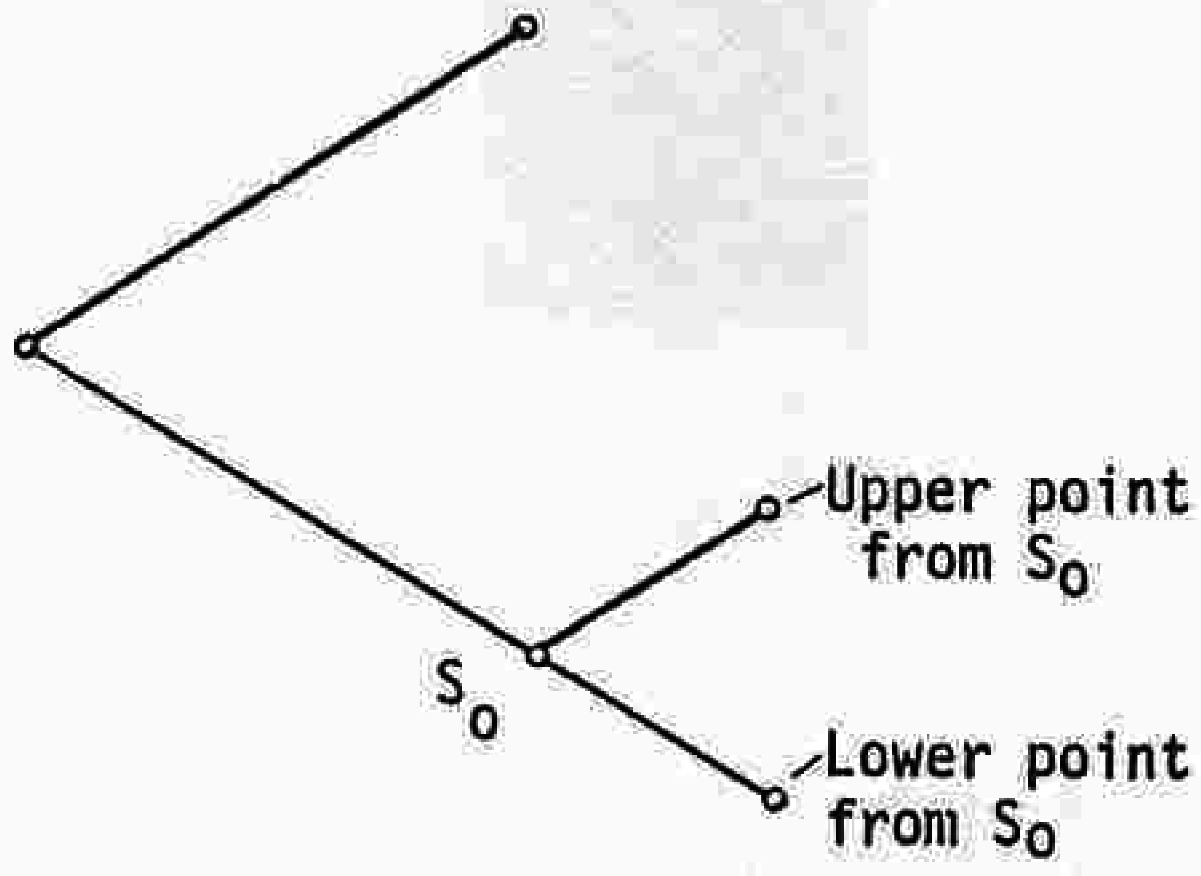


FIGURE 4.3

A 5-POINT TREE

<u>4.2.3 Definition</u>. A bough of a simplex tree is any maximal linearly-ordered subset of that tree.

With these definitions we can explore convenient methods of labeling the nodes of simplex trees.

If $\mathcal{T} = \{S_1, \ldots, S_k\}$ is a simplex tree, then the location of each $S_j \in \mathcal{T}$ with respect to the original point of $\mathcal{T}$ can be labeled by an ordered pair of integers (n_1, n_2) as follows: Let n_2 be the number of points in the linearly-ordered set $\mathcal{B} = \{S \in \mathcal{T},\ S < S_j\}$, i.e., let n_2 be the number of points of $\mathcal{T}$ which precede S_j. Observe that $\mathcal{B} \cup \{S_j\}$ consists of all points on the path in between the original point and S_j; reindex the elements of $\mathcal{B} \cup \{S_j\}$ so that S_0 is the original point in this path, S_1 is the first point to the right of the original point, S_i is the i^{th} point to the right of the original point, and S_{n_2} is the point previously called S_j. Then, for $i \in \{1, \ldots, n_2\}$, S_i is either an upper point or lower point in $\mathcal{T}$. Let us form the unique integer n_1 from this information by setting the i^{th} digit in the binary expansion of n_1 (e.g., counting from the right) equal to 1 if S_i is an upper simplex and otherwise setting the i^{th} digit in the binary expansion of n_1 equal to zero. (It is assumed that n_1 has only n_2 digits in its binary expansion, thus assuring the uniqueness of n_1.)

With the above definitions of n_1 and n_2, each point in $\mathcal{T}$ is uniquely labeled by the ordered pair (n_1, n_2). Fig. 4.4 on the next page illustrates this labeling scheme, and later examples will further clarify the concept (cf. Figs. 4.5-4.8 and Sec. 4.4).

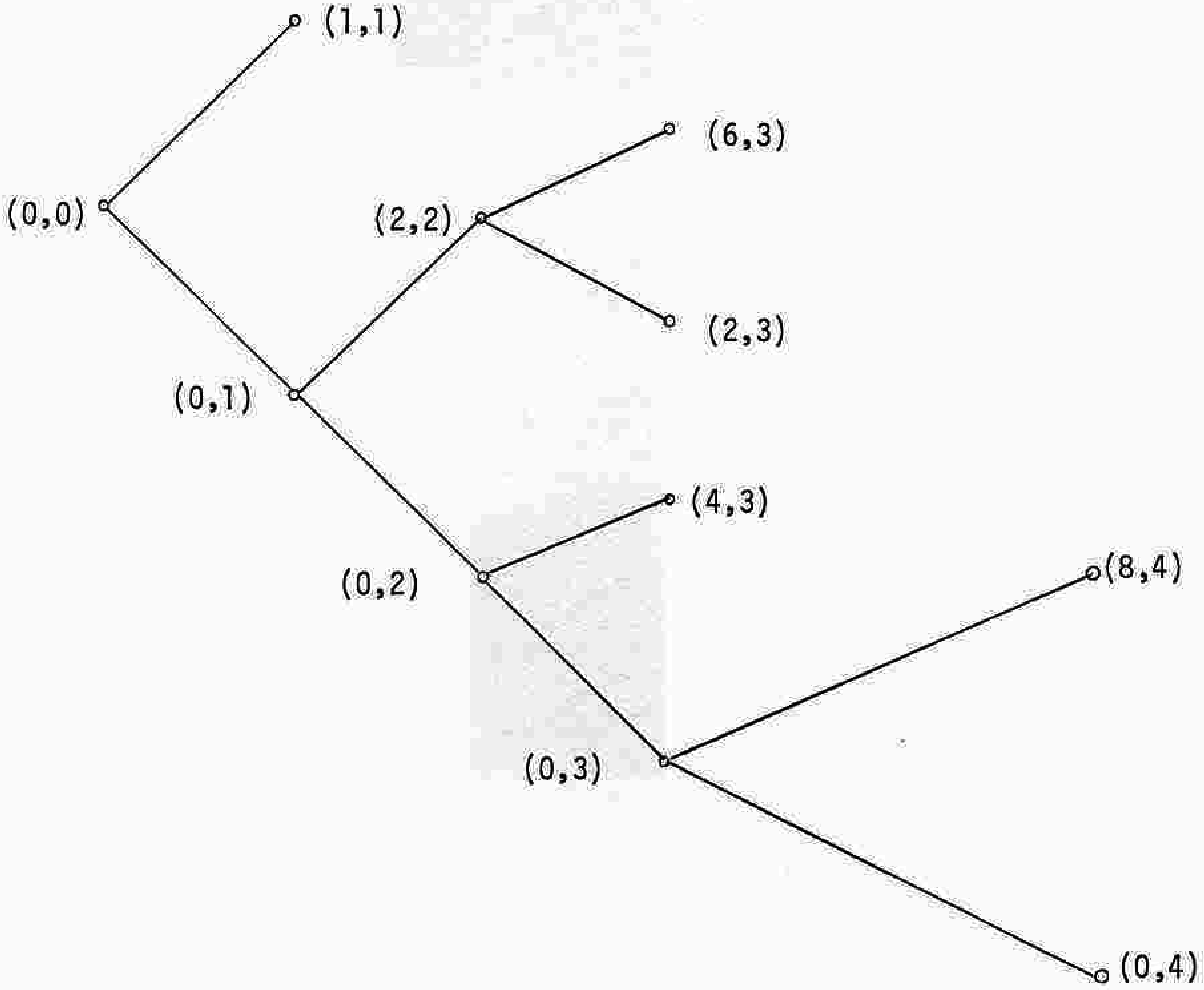


FIGURE 4.4

THE LABELING SCHEME

4.2.4 Definition. The numbers n_1 and n_2 above will be called location numbers, and the pair (n_1, n_2) will be called a location pair.

If $\{\mathcal{S}_i\}_{i=1}^{\kappa}$ is a sequence of sets of simplexes such that $\mathcal{S}_{i+1}$ is a subdivision of $\mathcal{S}_i$ for $i < \kappa$ and $\mathcal{S}_1 = \{S_0\}$, we can order the elements of $\bigcup_{i=1}^{\kappa} \mathcal{S}_i$ in a simplex tree as indicated in the following definition.

4.2.5 Definition. Let k, m, A, S, S_1, and S_2 be as in the definition of simple subdivision. We will say $S < S_1$ and $S < S_2$; we will furthermore distinguish S_1 and S_2 by calling S_1 the lower simplex and calling S_2 the upper simplex of the subdivision $\{S_1, S_2\}$ of S.

Figs. 4.5, 4.6 and 4.7 show successive bisections of $S_0 = \langle A,B,C \rangle$, and Fig. 4.8 shows the corresponding simplex tree. Note that $S_1 = \langle A,B,D \rangle$, $S_2 = \langle D,B,C \rangle$, $S_3 = \langle D,B,E \rangle$, and $S_4 = \langle D,E,C \rangle$. The indices of the vertices of these simplexes are written next to the vertex inside the simplex. For example, the index of B in S_2 is 2, while the index of C in S_2 is 3; thus, S_3 is the upper simplex and S_4 is the lower simplex following S_2.

In Fig. 4.8, the location numbers of the points in the simplex tree are also given, illustrating the method of labeling the simplexes in subdivision of S_0.

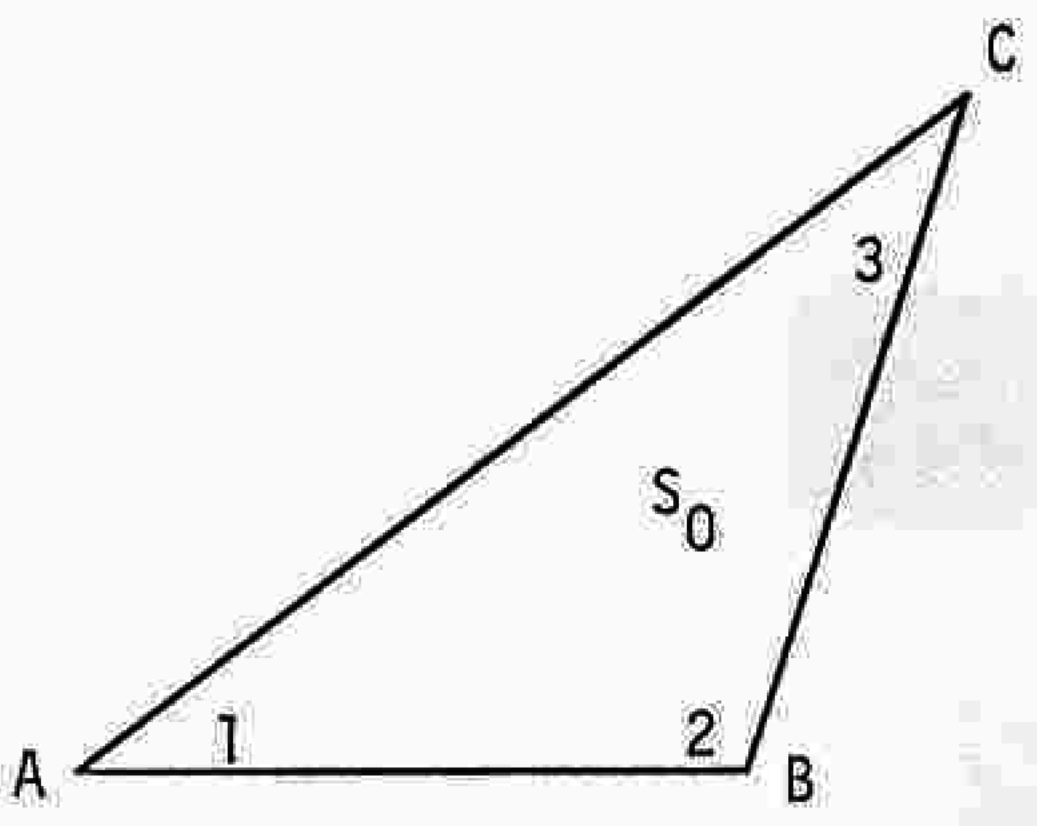


FIGURE 4.5

THE ORIGINAL SIMPLEX

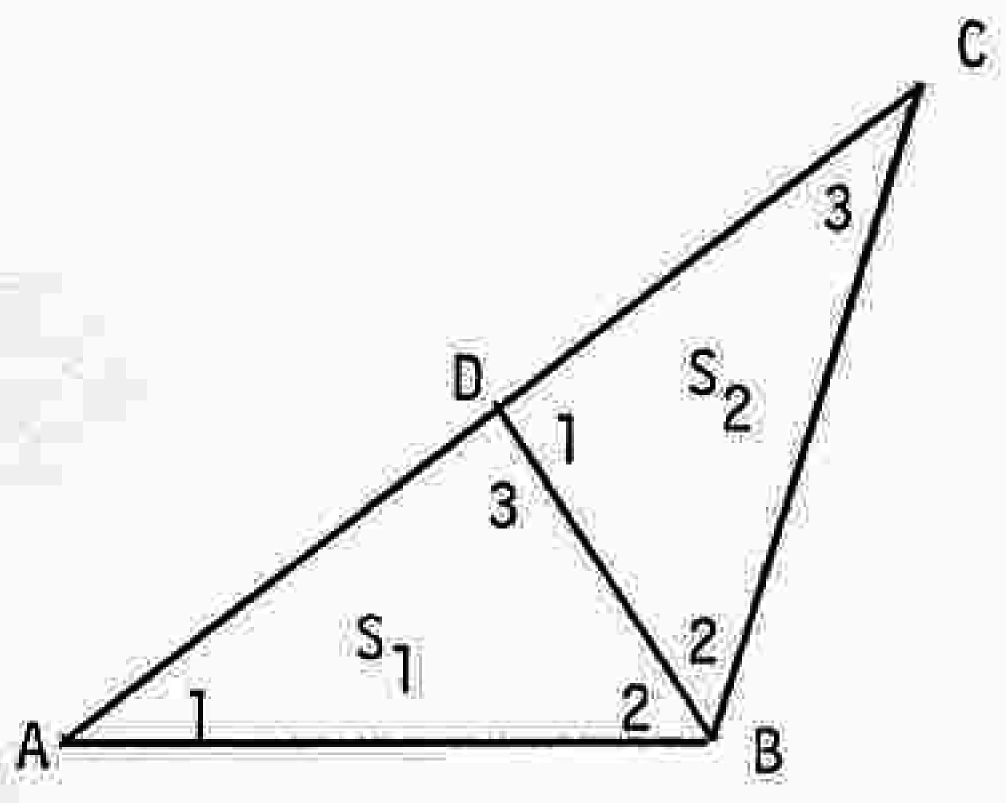


FIGURE 4.6

THE FIRST SUBDIVISION

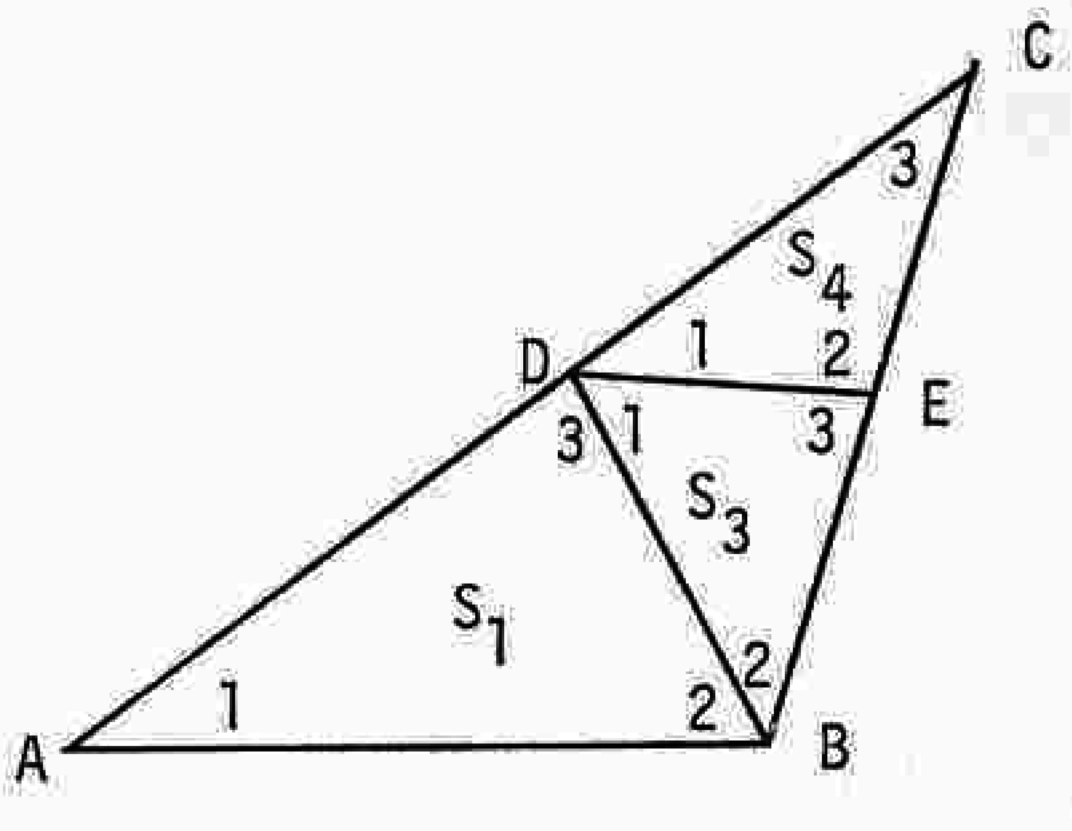


FIGURE 4.7

THE SECOND SUBDIVISION

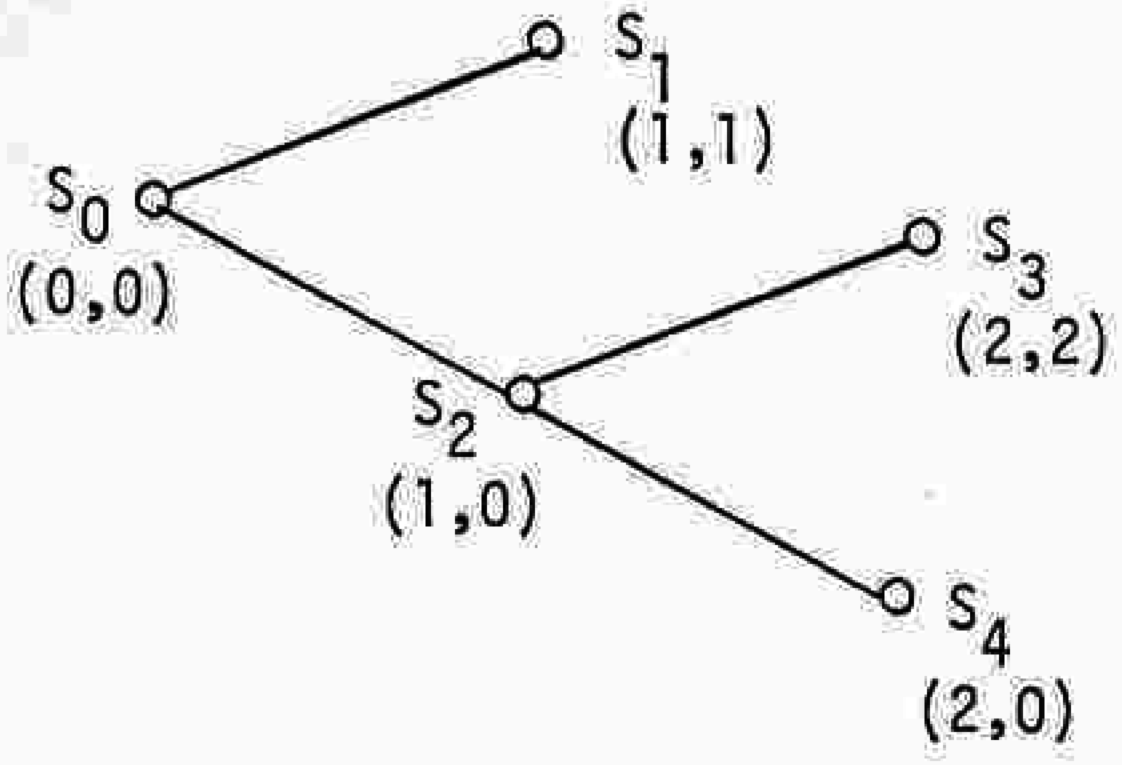


FIGURE 4.8

THE TREE

4.3 A Simple Bisection Algorithm

Stenger ([30] p. 29) presents the following outline for bisection of the (n-1)-simplexes comprising $b(P)$ and calculation of $d(F,P,\theta_n)$.

4.3.1 Algorithm. (1) Let p be a fixed positive integer.

(2) Read in $S_1 = \langle Y_1^1,\ldots,Y_n^1\rangle,\ldots,S_q = \langle Y_1^q,\ldots,Y_n^q\rangle$, the component simplexes of $b(P)$.

(3) Locate the longest segment $\langle Y_k^j, Y_m^j\rangle$ to be formed from the vertices of S_j for j = 1 to q, and calculate each midpoint $A_j = (Y_k^j + Y_m^j)/2$.

(4) Replace each S_j by the two simplexes:

$$\langle Y_i^j,\ldots,Y_{k-1}^j,A,Y_{k+1}^j,\ldots,Y_m^j,\ldots,Y_n^j\rangle \text{ and}$$

$$\langle Y_1^j,\ldots,Y_k^j,\ldots,Y_{m-1}^j,A,Y_{m+1}^j,\ldots,Y_n^j\rangle,$$

thus storing 2q simplexes.

(5) Replace q by 2q.

(6) Calculate the sum appearing in Theorem 3.3.1; call this sum δ.

(7) If δ equals the previous sum, e, which moreover is an integer, then to go step 8. Otherwise, set $e \leftarrow \delta$, and return to step 3.

(8) $p \leftarrow p - 1$. If $p = 0$, then print q and δ and stop.

The principal drawback of the above algorithm for higher dimensions is that after i iterations, $q \cdot 2^i$ simplexes must be stored. To illustrate the difficulty involved, take $n = 5$. Here, one may need to do 20 or more iterations, depending on F. If P were a simplex, then $b(P)$ would consist of six 4-simplexes, so that $6 \cdot 2^{20} = 2{,}097{,}152$ simplexes would be in storage after 20 iterations. Since each simplex has 5 vertices, this amounts to 10,485,760 words of storage. (Many large machines at present have only 50,000 to 200,000 words of fast memory core.) Furthermore, the number of iterations required increases as n increases for functions whose components have the same degree of smoothness.

There are several avenues for modification of the above. One is to not bisect each simplex each time, but to choose a single simplex at each iteration, so that no more than one bough of the simplex tree (in the example above, a bough would have 21 simplexes in it) is ever in storage at one time. With such methods, redundancy in the functional evaluations to get successive $\mathcal{R}(S,F)$ can also be eliminated.

Another possible modification is to allow the lengths of the bough in the simplex tree to vary (here, the length of a bough is the number of points in that bough).

Algorithm 4.4.1, in the next section, both keeps only one bough in storage and allows the bough length to vary.

4.4 Modifications of Algorithm 4.3.1

As was mentioned, to modify Algorithm 4.3.1 it is convenient to produce a tree for only one facet of b(P) at a time. From there, one would do calculations for one bough of the tree for that facet at a time, but keep track of which boughs have already been done. Such a method should proceed so that information from the current bough which will be in common with the next bough considered is stored, but that the quantity of such stored information will not exceed that for one bough.

The bough information consists of three arrays of matrices, to be labeled SA, RA, and MA. The array SA contains the simplexes corresponding to points in the bough currently being considered, the array RA contains the corresponding range simplexes, and the array MA contains information used to decide when the leaf of the bough has been obtained (i.e., when the bough is of a sufficient size to determine sufficient refinement).

Basically, a single iteration of the algorithm will consist of the following steps:

(1) Bisect the current simplex S into two simplexes S_1 and S_2.

(2) Decide whether the lower simplex S_1 has already been examined. If it has not, then $S \leftarrow S_1$. Otherwise, $S \leftarrow S_2$.

(3) Calculate $\mathcal{R}(S,F) = (r_{i,j})$, and compare each $r_{i,j}$ to the previous $r_{i,j}, 1 \le i, j \le n$. Update the "agreement matrix" $M = (m_{i,j})$ by $m_{i,j} \leftarrow m_{i,j} + 1$ if the two $r_{i,j}$'s coincide, and $m_{i,j} \leftarrow 0$ otherwise. Roughly, the information in the i-th column of the matrix M gives how many iterations (i.e., how many subdivisions) have been carried out since a sign change in the i-th component of F has been discovered.

(4) Determine whether the leaf has been obtained; simultaneously determine the parity of $\mathcal{R}(S,F)$ by examining both the entries of M and the entries of $\mathcal{R}(S,F)$.

The information indicating whether the lower simplex has already been examined is stored in an integer, n_1, in the following sense: If the simplex S of step (1) is n_2 simplexes from the original simplex (i.e., if n_2 simplexes precede S) then the first n_2 binary digits of n_1' are the first location number for S in the tree. (The second location number, of course, is n_2 itself.) Hence, S_1 of step (2) will already have been examined if the n_2-th digit of n_1' is 1.

On the first iteration, n_1' is set equal to 0, causing the bough containing only lower simplexes to be considered.

After a finite number of bisections, a leaf is always obtained. The location pair (n_1, n_2) is then stored in conjunction with the parity of $\mathcal{R}(S,F)$. After this, the algorithm backtracks one unit (i.e., $n_2 \leftarrow n_2 - 1$), sets the n_2-th binary digit of n_1' equal to 1, and sets subsequent digits of n_1' equal to 0. The n_2-th entries of the arrays SA, MA, and RA are retrieved and stored in S, M, and the current $\mathcal{R}(S,F)$ matrix, respectively. Iteration of the algorithm is then continued.

After the first bough (the one consisting only of lower simplexes) has already been considered, it may be necessary to backtrack more than one step when a leaf is obtained. If the n_2-th digit of n_1' is equal to 1, then after $n_2 \leftarrow n_2 - 1$, the parities of the lower and upper simplexes following the n_2-th simplex are added together and stored along with the location pair of the simplex in question. This "backtrack" procedure (this is not a backtrack algorithm as described in [6], p. 7) is then repeated until a digit of n_1' is found which is equal to 0. The appropriate matrices from SA, MA, and RA are then retrieved, and bisection iterations are continued.

The algorithm ends when it backtracks to $n_2 = 0$ (this corresponds to the original facet) and finds that both the upper and lower simplexes from this facet have been considered. This happens when the first digit of n_1' is 1. Then, the sum of the parity contributions of both the upper and lower point are added, giving the total parity contribution to the sum in Theorem 3.2.10 (the Parity Theorem).

The following algorithm formalizes the procedures described above.

<u>4.4.1 Algorithm</u>. (1) Let p be a fixed positive integer, let max be the maximum allowed bough length, and let ns be the maximum number of points allowed in the tree.

(2) Read in the original simplex and store it in S_1.

(3) Set all of the entries of the matrix $M = \{m_{i,j}\}_{i,j=1}^{n}$ equal to zero.

(4) $n_1' \leftarrow 0$, $n_2 \leftarrow 0$, $K \leftarrow 1$.

(5) Calculate $\mathcal{R}(S_1,F)$ and store in R_2. ($\mathcal{R}(S_1,F)$ can be stored in a vector with only n words in it by storing the i-th row of $\mathcal{R}(S_1,F)$ in the i-th word; set the j-th bit of the i-th word equal to 0 if $r_{i,j} = -1$, and set this bit equal to 1 if $r_{i,j} = +1$.)

(6) $RA_1 \leftarrow R_2$, $SA_1 \leftarrow S_1$, $MA_1 \leftarrow \{0\}_{i,j=1}^{n}$.

(7) $n_2 \leftarrow n_2 + 1$.

(8) If $n_2 > \max$, then stop.

(9) Find the bisection of S_1. Store the lower simplex back in S_1, and store the upper simplex in S_2. Store k and m, where $\langle X_k, X_m\rangle$ was the longest side of the old S_1.

(10) Examine the n_2-th binary digit of n_1'. If this digit equals zero, then go directly to step (11). Otherwise, $S_T \leftarrow S_1$, $S_1 \leftarrow S_2$, $S_2 \leftarrow S_T$, $nt \leftarrow m$, $m \leftarrow k$, $k \leftarrow mt$.

(11) $R_1 \leftarrow R_2$.

(12) Calculate $\mathcal{R}(S_1,F)$ and store in R_2.

(13) Compare the (i,j)-th entry of R_1 with the (i,j)-th entry of R_2. If they are the same, then $m_{i,j} \leftarrow m_{i,j}+1$. Otherwise $m_{i,j} \leftarrow 0$. Do this step for $i,j = 1,\ldots,n$.

(14) Determine whether a leaf has been obtained and if so, determine the parity of R_2. If a leaf has indeed been obtained then go to step (17). Otherwise, continue to step (15).

(15) $RA_{n_2+1} \leftarrow R_2$, $SA_{n_2+1} \leftarrow S_2$, $MA_{n_2+1} \leftarrow M$.

(16) Go back to step (7).

(17) Store the parity of R_2 in D_k; store the location pair ([first n_2 binary digits of n_1'], n_2) in LOC_k (the array D has length ns, as does the array of pairs of numbers LOC; the values in LOC are used to identify which points of the tree the values in D belong to).

(18) $K \leftarrow K + 1$.

(19) If the n_2-th binary digit of n_1' is 0, then go to step (24). Otherwise, continue to step (20).

(20) $n_2 \leftarrow n_2 - 1$.

(21) If $n_2 = 0$ then go to step (28).

(22) Search through LOC_ℓ, $\ell = 1$ to $k - 1$, until the location numbers for the upper and lower simplexes from the simplex at ([first n_2 digits of n_1], n_2) are found, say in the j_1-st and j_2-nd positions. Then add D_{j_1} and D_{j_2}, and store the sum in D_K. Store ([the first $n_2 - 1$ digits of n_1'], n_2) in LOC_K, then $K \leftarrow K + 1$.

(23) If the n_2-th digit of n_1' is 1, then return to step (20). Otherwise, continue to step (24).

(24) Set the n_2-th digit of n_1' equal to 1, and set all subsequent digits of n_1' equal to zero.

(25) $R_2 \leftarrow RA_{n_2}$, $M \leftarrow MA_{n_2}$, $S_1 \leftarrow SA_{n_2}$.

(26) $n_2 \leftarrow n_2 - 1$.

(27) Return to step (7).

(28) Find the location pairs (0,1), and (1,1), and add the corresponding D's together. This sum is the total contribution of the original facet to the sum of the parities of the simplexes comprising b(P).

The above algorithm is repeated until all of the facets of b(P) have been considered. The total contributions are then added to get $d(F,P,\theta_n)$.

A flowchart for Algorithm 4.4.1 will appear in some additional work.

Determining whether a leaf has been obtained and determining the value of Par($\mathcal{R}$(S,F)) are important parts of Algorithm 4.4.1. Determining when a leaf is obtained and determining Par($\mathcal{R}$(S,F)) can be done simultaneously by examining in sequence either the rows or columns of M and $\mathcal{R}$(S,F). Since the columns of $\mathcal{R}$(S,F) represent the behavior of the individual components of F, examining the columns may lead to a more efficient version of Algorithm 4.4.1, but examining the rows is simpler. The following algorithm accomplishes the task by examination of rows.

<u>4.4.2 Algorithm</u>. (1) Choose, a priori, a parameter p.

(2) $i \leftarrow 1$, $j \leftarrow 1$.

(3) Examine $m_{i,j}$. If $m_{i,j} < p$, do step (4). Otherwise, do step (5).

(4) Return to Algorithm 4.4.1 with instructions to continue bisection.

(5) Check $r_{i,j}$. If $r_{i,j} = -1$, then go to step (6). Otherwise go to step (7).

(6) $k_i \leftarrow j$.

(7) $j \leftarrow j + 1$.

(8) If $j \leq n$, return to step (3). Otherwise, go to step (9).

(9) $j \leftarrow 1$, $i \leftarrow i + 1$.

(10) If $i \leq n$, return to step (3). Otherwise, go to step (11).

(11) Attempt to put the k_i's in order of increasing magnitude. If two k_i's are equal, then $\mathrm{Par}(\mathcal{R}(S,F)) = 0$. Otherwise $\mathrm{Par}(\mathcal{R}(S,F)) = -1$ if the permutation of the k_i's was odd, and $\mathrm{Par}(\mathcal{R}(S,F)) = 1$ if that permutation was even.

<u>4.5 An Example</u>

The following example illustrates the working of Algorithm 4.4.1.

<u>4.5.1 Example</u>. Suppose that $F(X) = (f_1(X), f_2(X))$, where $X = (x,y)$, $f_1(X) = x^2 - y^2 - 1$, and $f_2(X) = x^2 + y^2 - 2$. Suppose also that P is the rectangle $\{0 \leq x \leq 2; 0 \leq y \leq 2\}$. Then compute $d(F,P,\theta_n)$ via Algorithm 4.4.1.

<u>4.5.2 Computations for Example 4.5.1</u>. The polygon P is drawn in Fig. 4.9. To begin the computations we write b(P) as the sum:

$$b(P) = \langle A,B\rangle + \langle B,C\rangle + \langle C,D\rangle + \langle D,A\rangle$$

where $A = (0,0)$, $B = (2,0)$, $C = (2,2)$, and $D = (0,2)$. The number $d(F,P,\theta_n)$ will then be the sum of the parities of the range simplexes formed from the above four simplexes, provided that each is subdivided so that b(P) is sufficiently refined relative to F.

Places where all but one of the components of F vanish on the boundary are in this case simply the points on b(P) where $f_1(x,y) = 0$ or $f_2(x,y) = 0$. These are marked in Fig. 4.9 for clarity. One immediately sees that $\langle B,C\rangle$, $\langle C,D\rangle$, and $\langle D,A\rangle$ need no further subdivisions to satisfy the definition of sufficient refinement. Furthermore, the zeros of f_1 and f_2 on $\langle A,B\rangle$ are simple; hence it is easy to check $\mathrm{Sgn}(F(X))$ for $X \in \langle A,B\rangle$.

The value $p = 1$ will be used to demonstrate the action of Algorithm 4.4.1 on $\langle A,B\rangle$. For each iteration, the following will be given: n_2, the binary and decimal representation of the current n_1',

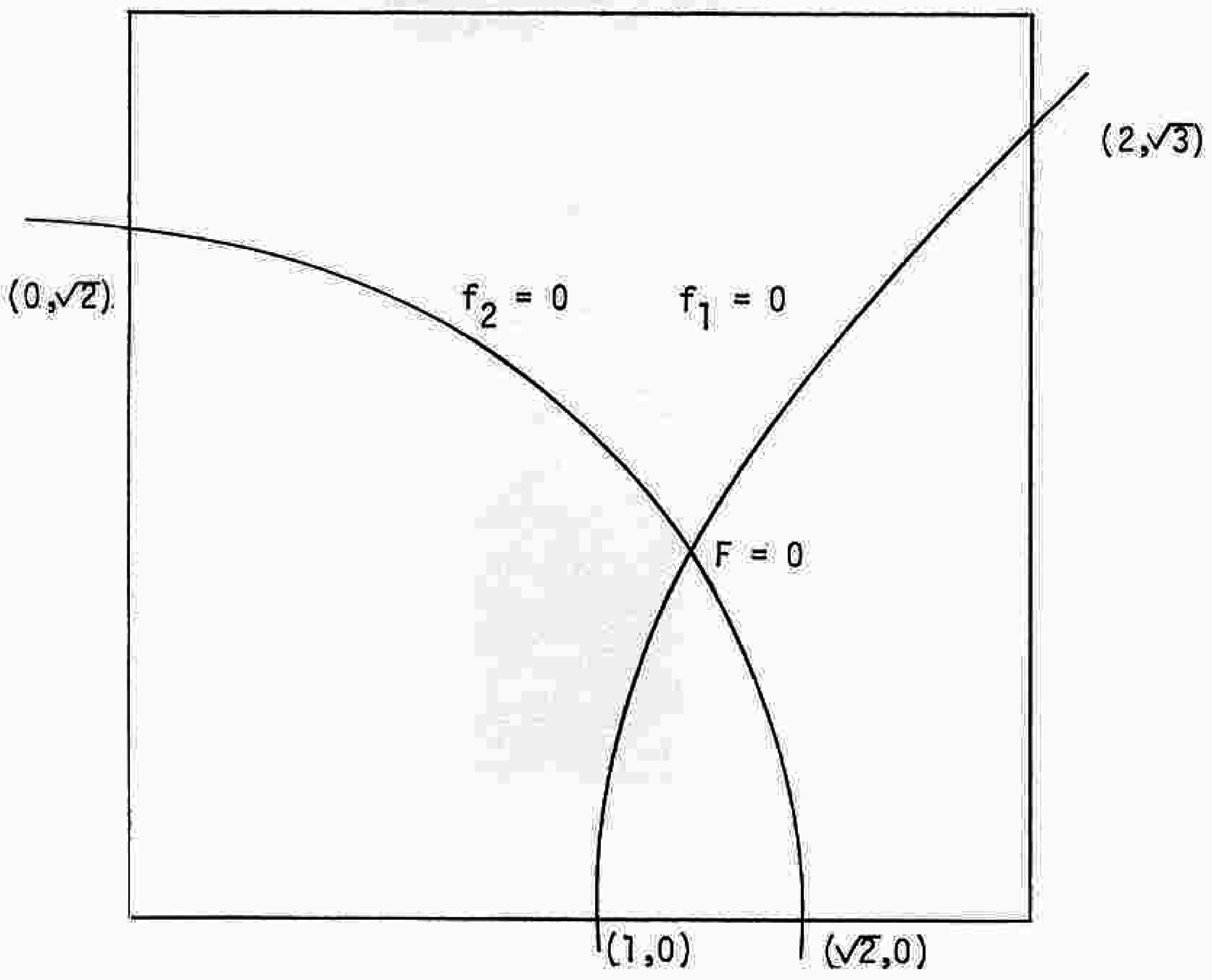


FIGURE 4.9

THE REGION IN THE COMPUTATIONAL EXAMPLE

the lower and upper simplexes S_1 and S_2, whether either S_1 or S_2 is chosen to be bisected further, $\mathcal{R}(S,F)$ and M. It will be assumed that n_1' has five binary digits. When the ℓ-th binary digit of n_1' is referenced, it will refer to the ℓ-th binary digit from the right.

After each leaf is obtained, a table of the parity contributions which have already been stored, a figure of the portion of the tree which has already been examined, and a figure showing the correspondence between the points in that portion of the tree and segments of $\langle A,B\rangle$ will be given.

Simplexes will be listed in matrix form, with the i-th row containing the coordinates of the i-th point of the simplex.

To begin the process, set

$$S_1 = \langle A,B\rangle = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$$

so

$$R_1 = \mathcal{R}(S_1,F) = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} .$$

Also,

$$n_2 \leftarrow 0,\ n_1' \leftarrow (00000)_2 = 0, M = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} ,$$

and

$$RA_1 \leftarrow R_1$$

$$SA_1 \leftarrow S_1$$

$$MA_1 \leftarrow \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Now for the first iteration:

<u>Iteration 1.</u>

$$n_2 \leftarrow n_2 + 1 = 1, \; n_1' = (00000)_2 = 0$$

$$S_1 = \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix} \text{ and } S_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} .$$

Since the first digit of n_1' is 0, $S_0 \leftarrow S_1$. Now

$$R_2 = \mathcal{R}(S_0,F) = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \text{ so } M = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} .$$

Since $m_{1,1} = 0$, it is determined that a leaf has not been obtained.

Hence, $MA_2 \leftarrow M$, $SA_2 \leftarrow S_0$, $RA_2 \leftarrow R_2$, and iteration proceeds.

Iteration 2.

$$n_2 \leftarrow n_2 + 1 = 2,\ n_1' = (00000)_2 = 0$$

$$S_1 = \begin{pmatrix} 1.5 & 0 \\ 0 & 2 \end{pmatrix},\ S_2 = \begin{pmatrix} 1 & 0 \\ 1.5 & 0 \end{pmatrix} .$$

Since the n_2-th (second) digit of n_1' is 0, $S_0 \leftarrow S_1$. Now $R_1 \leftarrow R_2$, and the new R_2 is:

$$R_2 = \mathcal{R}(S_0,F) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} .$$

Comparing R_1 and R_2 again gives

$$M = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix} .$$

It is determined that a leaf has not yet been obtained (one encounters $m_{1,2} = 0$ in the check for the parity of R_2) so bisection is continued.

Iteration 3.

$$n_2 \leftarrow n_2 + 1 = \quad ,\ n_1' = (00000)_2 = 0.$$

$$S_1 = \begin{pmatrix} 1.75 & 0 \\ 2 & 0 \end{pmatrix},\ S_2 = \begin{pmatrix} 1.5 & 0 \\ 1.75 & 0 \end{pmatrix} .$$

Since the third digit of n_1' is 0, $S_0 \leftarrow S_1$. Now $R_2 \leftarrow R_1$ again, and

$$R_2 \leftarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} .$$

Another comparison of R_2 gives:

$$M = \begin{pmatrix} 2 & 1 \\ 3 & 3 \end{pmatrix} .$$

It is now determined that the parity of R_2 is zero, without encountering an entry of M which is less than $p = 1$; hence, the leaf has been obtained. Store the location pair and the parity (zero) in the k-th (1-st) position of the arrays:

$$\text{LOC}(K) \leftarrow (0,3)$$

$$\text{DEG}(K,1) \leftarrow 0$$

Also

$$K \leftarrow K + 1 = 2.$$

The finished part of the tree appears in Fig. 4.10. In Fig. 4.11, the actual subdivision of $\langle A,B \rangle$ is shown and labeled with their corresponding location pairs.

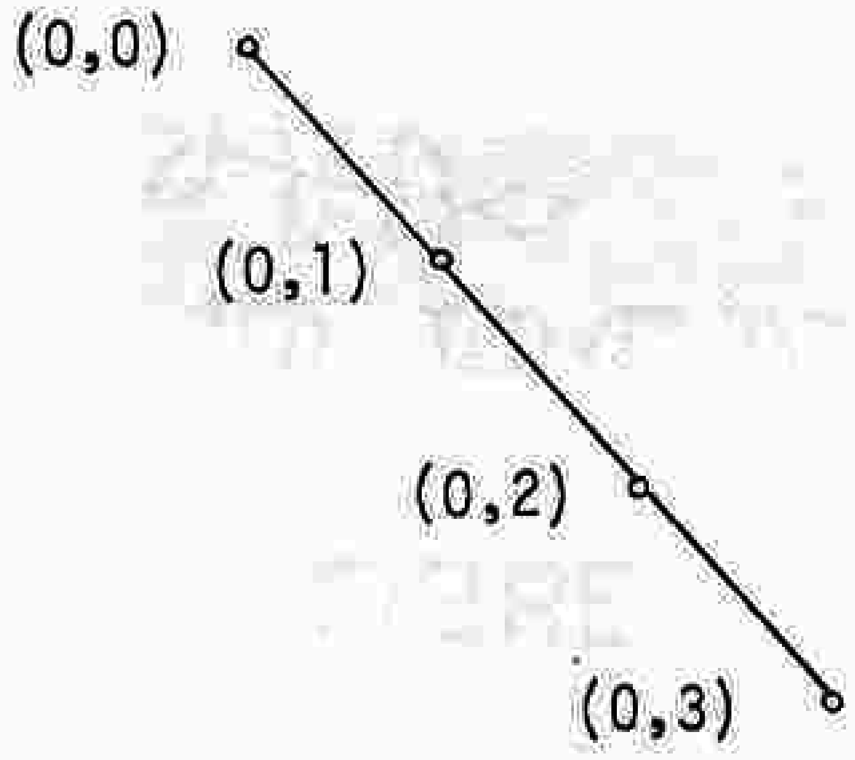


FIGURE 4.10

THE TREE AFTER THREE ITERATIONS

(The Parity Contribution of the point at (0,3) has been stored.)

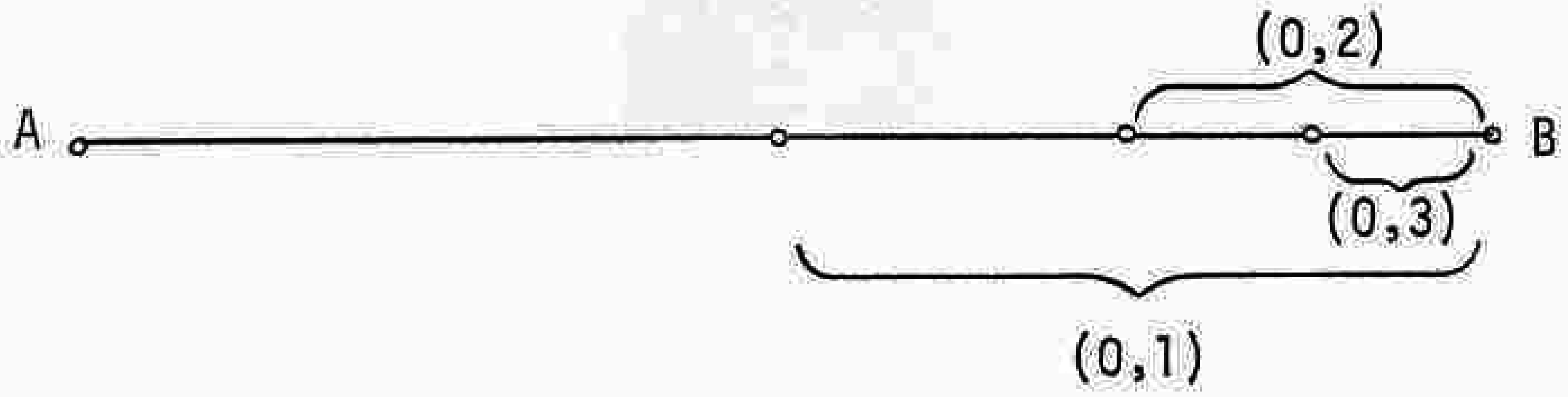


FIGURE 4.11

SUBDIVISION OF ⟨A,B⟩ AFTER THE THIRD ITERATION

In Fig. 4.12, the matrix arrays SA, RA, and MA are given.

The third digit of n_1 is now 0. Hence, the third digit of n_1 is set equal to 1, and all subsequent digits are reset to zero, giving:

$$n_1' = (00100)_2 = 4.$$

Now $n_2 \leftarrow n_2 - 1 = 2$, and the n_2^{th} digit of n_1' is 0. So, $R_2 \leftarrow RA_{n_2+1} = RA(3)$, $S_1 \leftarrow SA(3)$, and $M \leftarrow MA(3)$. Iteration then continues.

<u>Iteration 4</u>.

$$n_2 \leftarrow n_2 + 1 = 3,\ n_1' = (00100)_2 = 4,\ R_1 \leftarrow R_2$$

$$S_1 = \begin{pmatrix} 1.75 & 1 \\ 2 & 0 \end{pmatrix} \ ; \ S_2 = \begin{pmatrix} 1.5 & 0 \\ 1.75 & 0 \end{pmatrix}$$

Since the third digit of n_1' is now 1, $S_0 \leftarrow S_2$, so $R_2 \leftarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, and $M \leftarrow \begin{pmatrix} 2 & 1 \\ 3 & 3 \end{pmatrix}$. Hence, it is determined that the parity of R_2 is zero and a leaf has again been obtained. Consequently, as before: $LOC(K) \leftarrow (n_1, n_2) = (4,3)$

$$DEG(K) \leftarrow 0$$

$$K \leftarrow K + 1 = 3.$$

n	SA	RA	MA
1	$\begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$	$\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
2	$\begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$
3	$\begin{pmatrix} 1.5 & 0 \\ 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix}$
4	N.D.*	N.D.	N.D.
5	N.D.	N.D.	N.D.

*The letters N.D. mean that the corresponding element has not been defined yet.

FIGURE 4.12

THE MATRIX ARRAYS AFTER THE THIRD ITERATION

The third digit of n_1' is now 1, so $n_2 \leftarrow n_2 - 1$; the points in the table of LOC(K) and DEG(K) corresponding to the point at (0,2) (0 is the number represented by the first two digits of n_1') are added together (these happen to be the only two points previously stored in this case) and stored:

$$\mathrm{LOC}(3) \leftarrow (0,2)$$

$$\mathrm{DEG}(3) \leftarrow \mathrm{DEG}(2) + \mathrm{DEG}(1) = 0.$$

The table for the current tree information appears in Fig. 4.13, and the corresponding portion of the tree and subdivision of (A,B) appear in Fig. 4.14 and Fig. 4.15, respectively. The current matrix arrays are still the same as those in Fig. 4.12.

Now $n_2 \leftarrow n_2 - 1 = 2$; and the second digit of n_1' is 0, so that digit is set equal to 1. All subsequent digits of n_1' are set equal to 0 so that $n_1' = (00010)_2 = 2$.

So, $R_2 \leftarrow \mathrm{RA}(2)$, $S_1 \leftarrow \mathrm{SA}(2)$, $M \leftarrow \mathrm{MA}(2)$, and $n_2 \leftarrow n_2 + 1$; iteration is then continued.

Iteration 5.

$$n_2 \leftarrow n_2 + 1 = 2\ ,\ n_1' = (00010)_2 = 2\ ,\ R_1 \leftarrow R_2$$

$$S_1 = \begin{pmatrix} 1.5 & 0 \\ 0 & 2 \end{pmatrix}\ ;\ S_2 = \begin{pmatrix} 1 & 0 \\ 1.5 & 0 \end{pmatrix}$$

K	LOC (K)	DEG (K)
1	(0,3)	0
2	(4,3)	0
3	(0,2)	0

FIGURE 4.13

TABLE OF TREE INFORMATION AFTER ITERATION 4

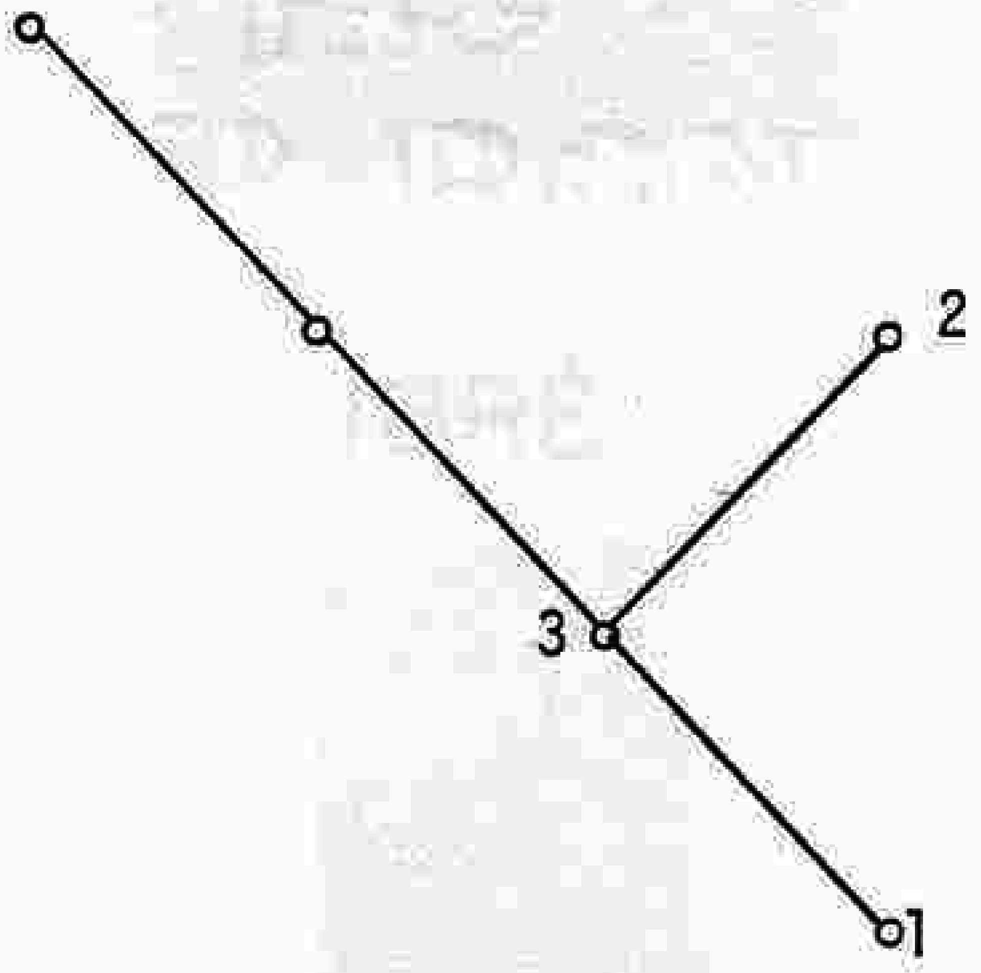


FIGURE 4.14

THE TREE AFTER ITERATION 4

(The points are numbered according to how the tree information appears in the table.)

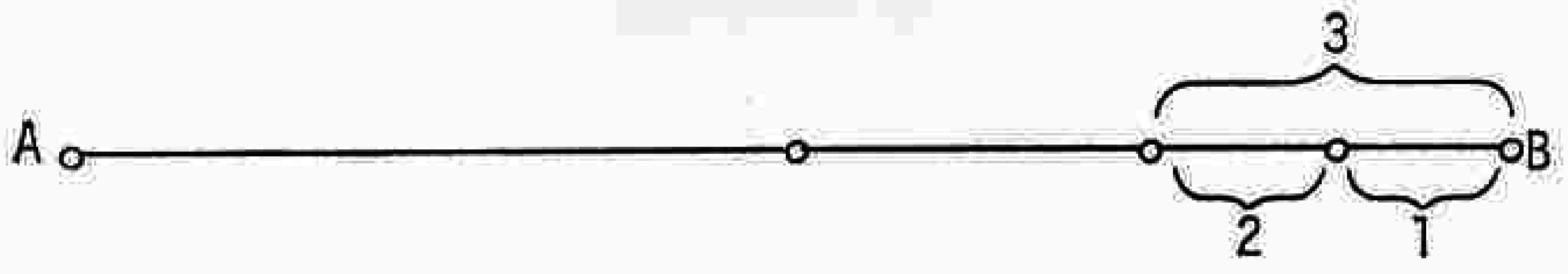


FIGURE 4.15

SUBDIVISION OF ⟨A,B⟩ AFTER ITERATION 4

Since the second digit of n_1' is 1, $S_0 \leftarrow S_2$, and $R_2 \leftarrow \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$. Comparing R_2, R_1 and the previous M now gives

$$M \leftarrow \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix}.$$

It is now concluded that a leaf has been obtained and that the parity of R_2 is 1. Hence,

$$LOC(4) \leftarrow (n_1,n_2) = (2,2)$$

$$DEG(4) \leftarrow 1 .$$

$$K \leftarrow K + 1 = 5$$

The second digit of n_1' is now 1, so $n_2 \leftarrow n_2 - 1$ and the points in the table which follow (0,1) in the partial order are added together. These are points numbers 3 and 4 in the table (the points to be added together are not necessarily the last two points in the table; in fact, they very often are not if $p > 1$), with location numbers (0,2) and (4,2). (The location numbers of the two points immediately following the point (n_1,n_2) can be calculated by the following rule: their common second location number is $n_2 + 1$, while the location number of the lower point is equal to n_1, and the location number of the upper point is equal to $n_1 + 2^{n_2}$.) From this, the total parity contribution of the point at (0,1) is stored:

$\mathrm{LOC}(5) \leftarrow (0,1)$

$\mathrm{DEG}(5) \leftarrow \mathrm{DEG}(4) + \mathrm{DEG}(3) = 1$.

$K \leftarrow K + 1 = 6$

The table for the tree information thus formed appears in Fig. 4.16, and the corresponding tree and subdivision appear in Fig. 4.17 and Fig. 4.18, respectively.

The matrix arrays remain unchanged.

Now $n_2 \leftarrow n_2 - 1 = 1$ again, and the n_2^{th} (first) digit of n_1' is again checked, but found to equal 0. Set this digit equal to 1 and set all subsequent digits equal to 0, so $n_1' = (00001)_2 = 1$. Now $R_2 \leftarrow RA(n_2) = RA(1)$, $S_0 \leftarrow SA(1)$, $M \leftarrow M(1)$, $n_2 \leftarrow n_2 - 1$, and iteration is continued.

<u>Iteration 6</u>.

$n_2 \leftarrow n_2 + 1 = 1$, $n_1' = (00001) = 1$, $R_1 \leftarrow R_2$.

$$S_1 = \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix} ; \; S_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Since the n_2^{th} digit of n_1' is 1, $S_0 \leftarrow S_2$. From this,

$$R_2 \leftarrow \begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix} \text{ and } \quad M \leftarrow \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} .$$

K	LOC (K)	DEG (K)
1	(0,3)	0
2	(4,3)	0
3	(0,2)	0
4	(2,2)	1
5	(0,1)	1

FIGURE 4.16

TABLE OF TREE INFORMATION AFTER ITERATION 5

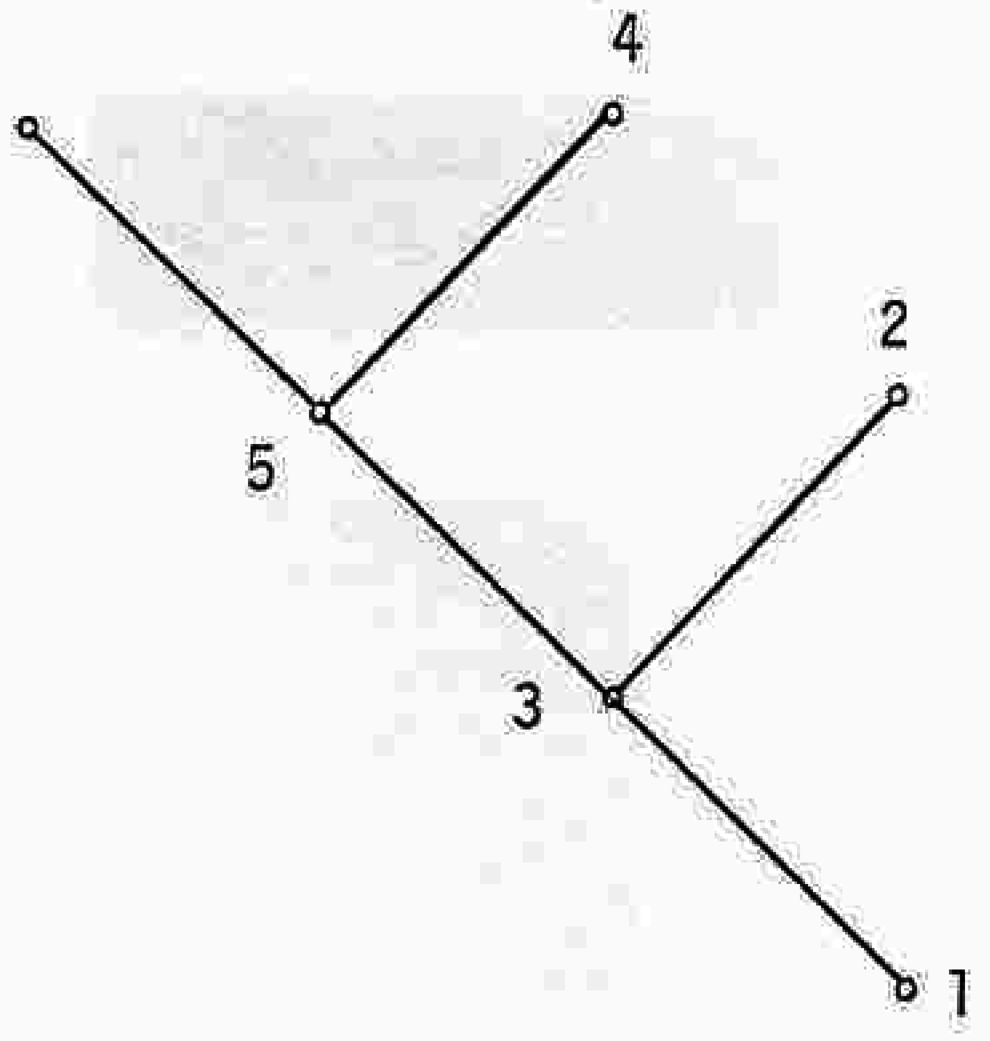


FIGURE 4.17

THE TREE AFTER ITERATION 5

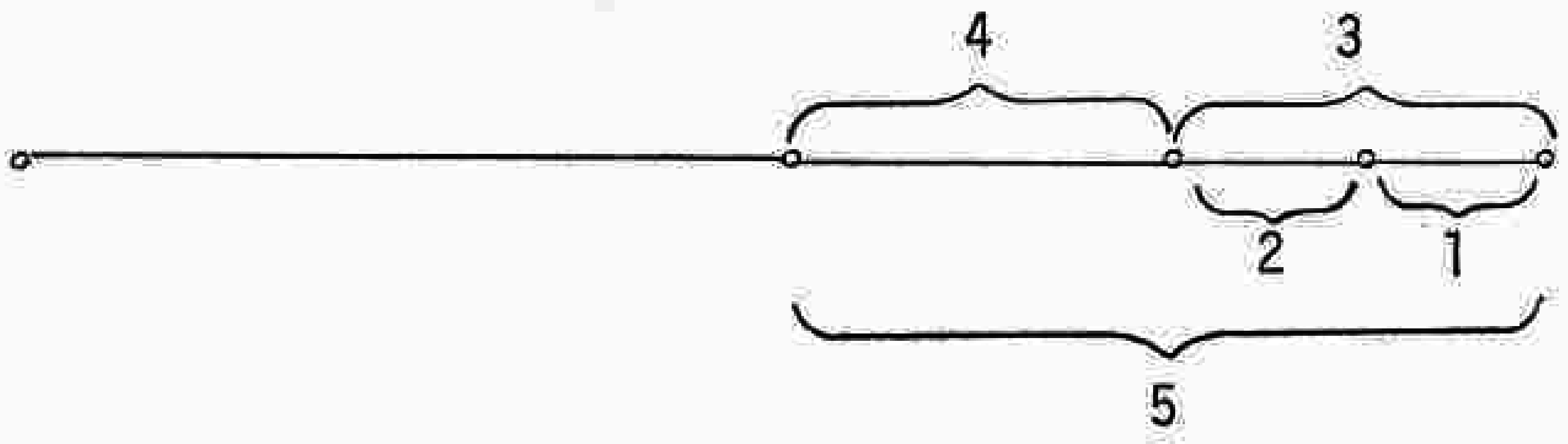


FIGURE 4.18

SUBDIVISION OF ⟨A,B⟩ AFTER ITERATION 5

It is concluded that the leaf has been obtained, and the parity of R_2 is 0.

Hence:

$$LOC(6) \leftarrow (n_1,n_2) = (1,1)$$

$$DEG(6) \leftarrow 0$$

The backtrack-search scheme is initiated, but it is found that $n_2 = 1$ and the first digit of n_1' is 1. This implies that all leaves in the tree have been obtained. The total parity contribution of the side $\langle A,B \rangle$ is found by adding the contributions of the points with location pairs (0,1) and (1,1); this total contribution is found to be 1.

The completed tree information appears in Fig. 4.19. The completed tree and subdivision appear in Fig. 4.20 and Fig. 4.21, respectively. The roots on $\langle A,B \rangle$ of $f_1 = 0$ and $f_2 = 0$ are marked by X's.

The trees for the other three sides of $b(\mathcal{D})$ are even simpler. The total contributions the these are all 0, so $b(F,\mathcal{D},\theta_2) = 1$.

In this case, it is easy to solve the system explicitly for the root within the region, and ascertain that $J(F) > 0$ at that

K	LOC (K)	DEG (K)
1	(0,3)	0
2	(4,3)	0
3	(0,2)	0
4	(2,2)	1
5	(0,1)	1
6	(1,1)	0
Total		1

FIGURE 4.19

THE COMPLETED TABLE OF TREE INFORMATION

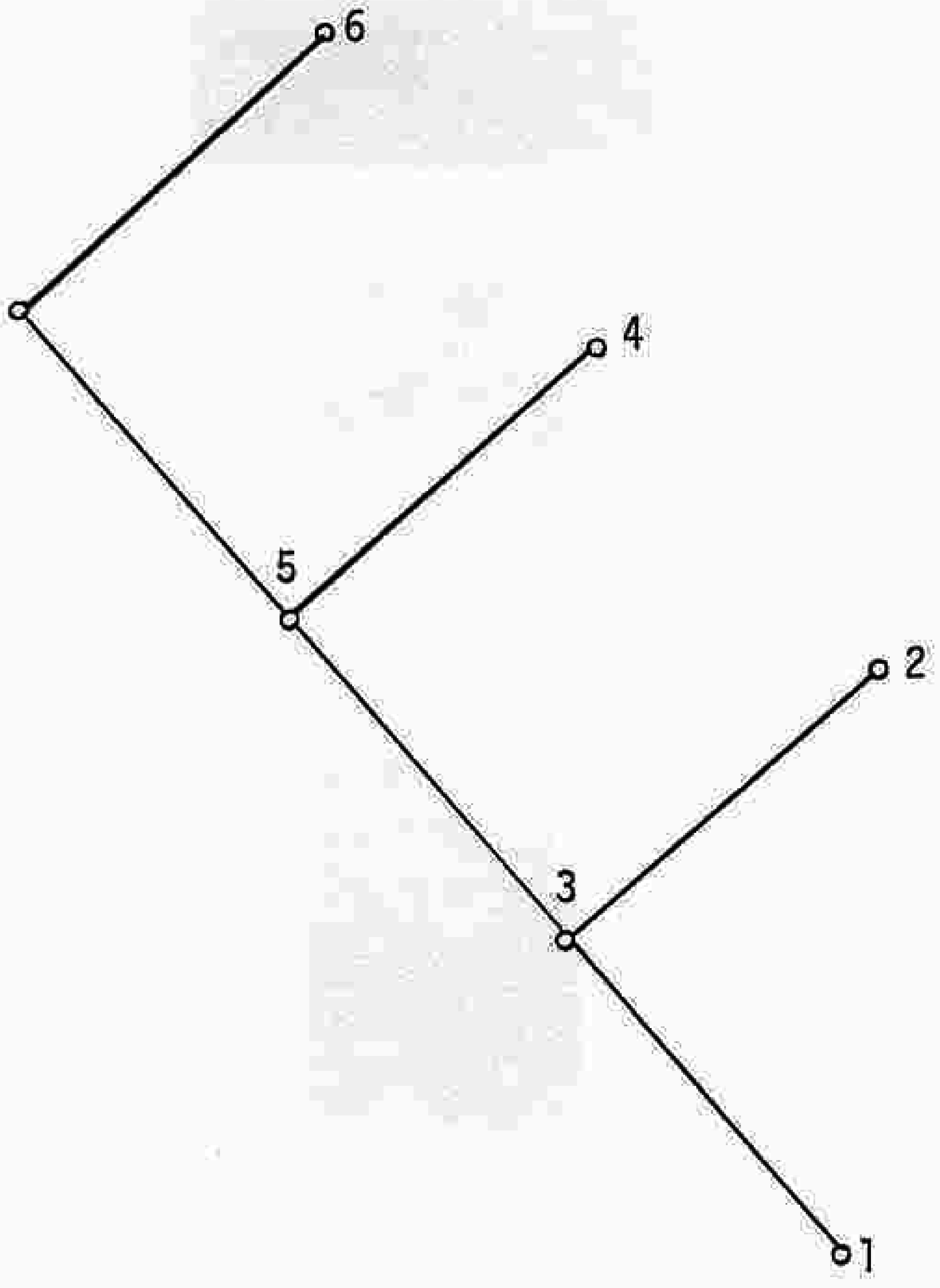


FIGURE 4.20

THE COMPLETED TREE

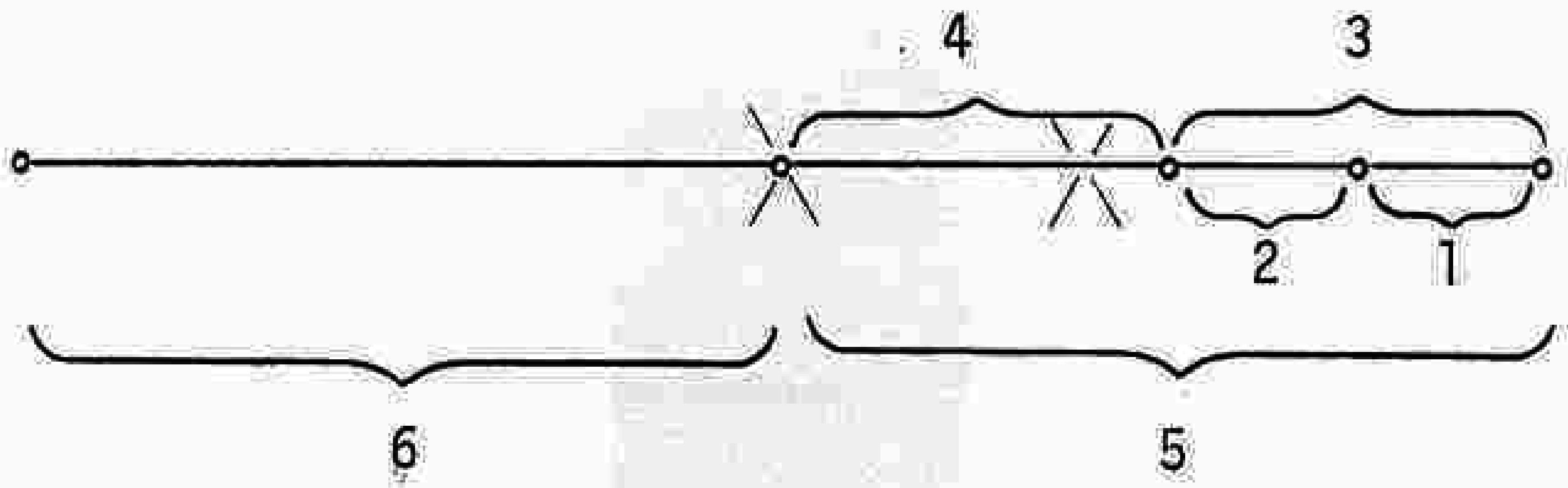


FIGURE 4.21

FINAL SUBDIVISION OF ⟨A,B⟩

root. From Remark 2.2.5, doing so verifies that the calculated degree is correct.

4.6 Further Modifications

Additional efficiency can often be gained by further modifying Algorithm 4.4.1 to handle special cases. For example, one may not wish to find a root of $F(X) = \theta_n$ by using information gained from calculating $D(F,\mathcal{D},\theta_n)$. A modification eliminating unnecessary storage and complexity to be applied in this case is explained. Also, modifications to eliminate redundancy in evaluations of $F(X)$, better tests of sufficient refinement, modifications to reduce the amount of memory required, and special considerations when n is small are discussed below.

Many of these modifications should be routinely built into any program, and numerical results obtained by including them will appear in later works.

<u>4.6.1 Calculating $d(F,\mathcal{D},\theta_n)$ Only</u>. When no generalized bisection of the n-simplex is to be carried out as in Chapter V, no labels need be attached to any of the elements of DEG (i.e., LOC need not exist). Recall that DEG was the array containing parities or sums of parities, and LOC was the array of addresses of simplexes corresponding to these parities. Also, according to this modification the only elements of DEG to be stored are those corresponding

to the actual parities. When all leaves have been obtained, then all of the elements of DEG are added together to get the total contribution.

With this modification, the amount of memory used is cut by a factor of approximately two, and the routine to search through the elements of the arrays LOC and DEG after completion of a branch is totally eliminated. This can constitute a significant saving of time, especially in higher dimensions where the trees must be large.

<u>4.6.2 Removal of Redundancy</u>. Another modification of Algorithm 4.4.1 is, instead of storing the simplex which was chosen for further bisection in SA_{n_2}, to always store the upper simplex. Also, if $\langle X_k, X_m \rangle$ were the longest one-dimensional segment formed from the points stored in S_1 and A were the midpoint of $\langle X_k, X_m \rangle$, then only Sgn(F(A)) need be evaluated in order to produce the range simplexes for both the upper and lower simplexes from S_1. To get the range simplex corresponding to the lower simplex, replace the k^{th} row of $\mathcal{R}(S_1, F)$ by Sgn(F(A)), and to get the range simplex corresponding to the upper simplex, replace the m^{th} row of $\mathcal{R}(S_1, F)$ by Sgn(F(A)).

Let us assume that, using the above mode of calculation, the upper simplex formed from each bisection has been stored in the array SA, the range simplexes for the upper simplexes have been stored in the array RA, and the matrix M has been computed for each upper simplex and stored in the array MA (cf. Fig. 4.22). We then do not need to backtrack as far as before, and step (25) (the information retrieval step) of Algorithm 4.4.1 becomes:

$$R_2 \leftarrow RA_{n_2+1},\ S_1 \leftarrow SA_{n_2+1},\ M \leftarrow MA_{n_2+1}.$$

Step (26) is then eliminated, and we return to Step (14) (determining if leaf has been obtained) instead of Step (7) (beginning of the bisection process). Also we must reverse the order of Step (14) and Step (15) (determining obtainment of the leaf and storing bough information). In Modification 4.6.5, a modification for the test for sufficient refinement is given for which the matrix M gotten from considering the upper simplex is the same as the corresponding matrix for the lower simplex, so that additional computations are not required.

With the above modification, there is no redundancy in the calculations of bisections or in the functional evaluations; i.e., F(X) is evaluated at most once for each point X. It is seen by examining the possible trees that this modification removes more than 1/3 of the functional evaluations previously required without sacrificing anything other than a slight increase in complexity.

4.6.3 Use of Determinants. The Algorithm 4.4.1 can also be modified so that Theorem 3.3.1 is used instead of Theorem 3.2.10.

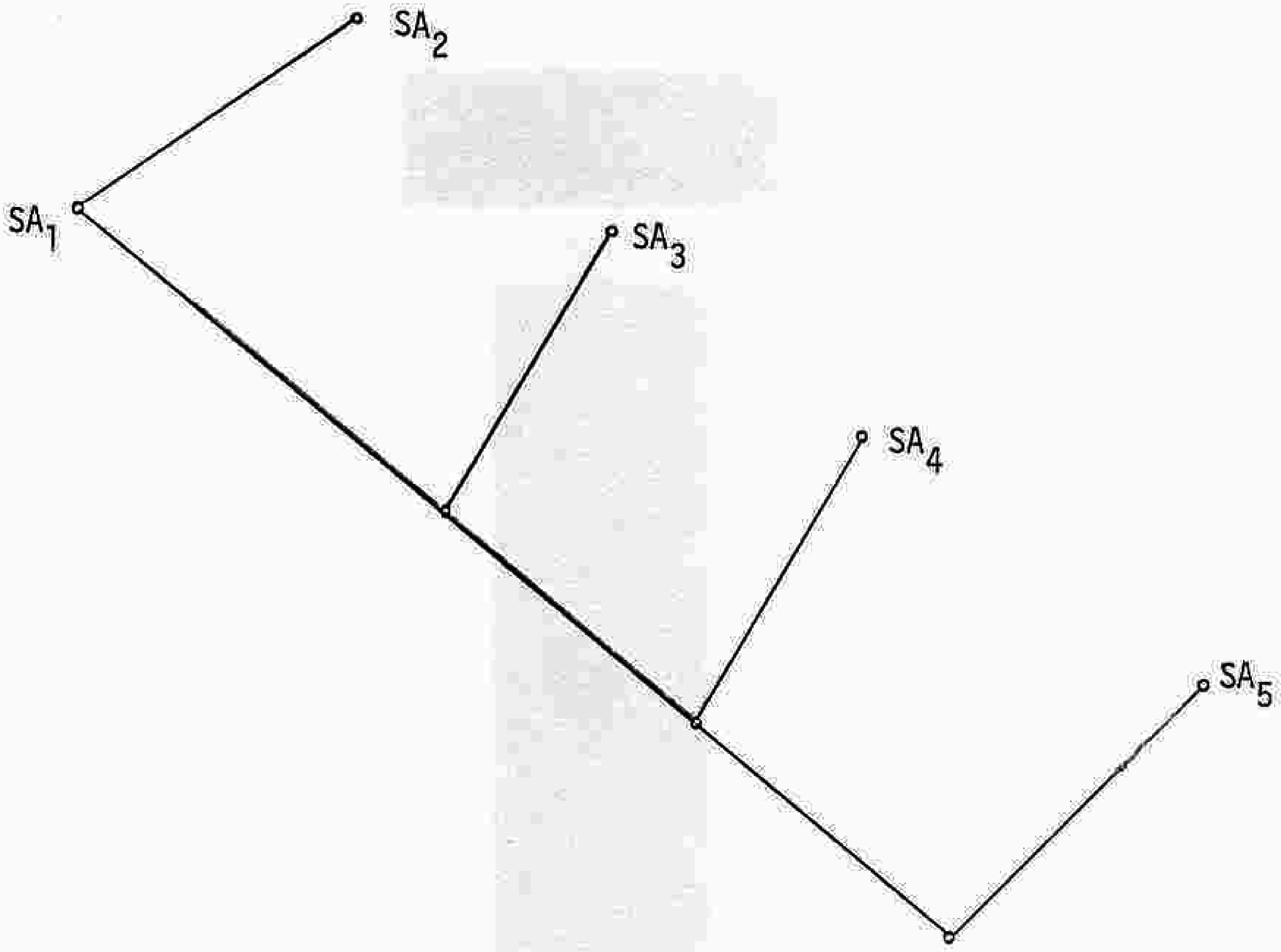


FIGURE 4.22

A POSSIBLE FIRST ITERATION OF ALGORITHM 4.4.1

WITH THE MODIFICATION DESCRIBED IN MODIFICATION 4.6.2

(A branch end was reached when $n_2 = 4$)

As was mentioned, use of Theorem 3.2.10 requires less computations per iteration than Theorem 3.3.1. There is an additional test of the correctness of the result however, when Theorem 3.3.1, requiring calculation of determinants, is used.

To implement Theorem 3.3.1, the determinants corresponding to both $\mathcal{R}(S_1,F)$ and $\mathcal{R}(S_2,F)$ (see Theorem 3.3.1) are evaluated after step (12). The matrix M and the array MA are replaced by an integer m and an array of integers ma. It may now be said that there has been an "agreement" with the previous iteration if either det $(\mathcal{R}(S,F))$ $= 0$ or $\det(\mathcal{R}(S_2,F)) = 0$. Again, a parameter p is specified, a priori, and it is decided that a leaf is obtained when there have been p successive agreements.

When the leaf has been obtained according to this new criterion, $\det(\mathcal{R}(S_1,F))$ and $\det(\mathcal{R}(S_2,F))$ are immediately added together to get the contribution of their predecessor in the simplex tree. This contribution is stored along with the location pair corresponding to the predecessor. The rest of the algorithm proceeds as before.

After the total contributions of the various facets of b(P) are found and added up, however, the result is divided by $2^n n!$ to get $d(F,P,\theta_n)$. If $d(F,P,\theta_n)$ is not an integer, then it is not correct. In that case one may wish to do the calculations over again with a larger value of P.

If the computer has a sufficient amount of storage space, one can store the simplexes and range simplexes corresponding to leaves. In the event an incorrect degree has been calculated, the boughs can then be lengthened without redundency in functional evaluations, although a totally new table of DEG and LOC must be computed. Though requiring much memory, this scheme might still be better than Algorithm 4.3.1, since it allows for nonuniform subdivisions.

4.6.4 Reducing the Amount of Memory Required. Bisecting the simplex may be relatively easy compared to evaluating Sgn(F). One may then wish to store only the range simplexes and M, and calculate the appropriate S after each backtrack instead of retrieving it from memory. Since the range simplexes can be stored in n words, deleting SA and keeping only RA and MA reduces the total storage by a factor of approximately 2, for high dimensions.

4.6.5 Various Test for Sufficient Refinement. There are various ways of testing for sufficient refinement: the definition of the entries of each matrix M can be altered, and (as was seen in Modification 4.6.3), it is not necessary to use a matrix, M. Two additional methods of testing will be mentioned here.

First, suppose $\langle X_k, X_\ell \rangle$ is the longest side of the (n-1)-simplex S, and $P = (X_k + X_\ell)/2$. Then $m \leftarrow m + 1$ if $\mathrm{Sgn}(P) = \mathrm{Sgn}(X_k)$

or Sgn(P) - Sgn(X_ℓ); otherwise $m \leftarrow 0$. A leaf would then be obtained when $m > p$. It is easy to show that $m \leftarrow m + 1$ after each step after sufficient refinement. With this method, no matrix array MA need be stored.

Another test for sufficient refinement involves a change in when we advance the entries of M. We would advance each entry (i.e., $m_{i,j} \leftarrow m_{i,j} + 1$ for $i,j = 1,\ldots,n$) after a given iteration. Then, if $\langle X_k, X_\ell \rangle$ was the side whose midpoint had been found, we would set $m_{k,j} \leftarrow 0$ only if the (k,j)th entry of R_1 equals <u>neither</u> the (k,j)th entry of R_2 <u>nor</u> the (ℓ,j)th entry of R_2, for $j = 1,\ldots,n$.

If $n = 2$, this procedure corresponds to setting $m_{1,1} \leftarrow 0$ if

$$\operatorname{sgn} f_j \left(\frac{A + B}{2}\right)$$

differs from both sgn $f_j(A)$ and sgn $f_j(B)$, for $j = 1,2$. This case is illustrated in Figs. 4.23-4.26. Since it is assumed f_1 is positive in all cases considered, the first column of R_1 and R_2 will be $(1,1)^T$ in all cases; hence, neither $m_{1,1}$ nor $m_{2,2}$ would be set to 0. Similarly,

$$\operatorname{sgn}\left[f_2 \left(\frac{A + B}{2}\right)\right] = \operatorname{sgn}\,[f_2(B)]$$

in Fig. 4.24, and neither $m_{2,1}$ nor $m_{2,2}$ would be set to 0. In

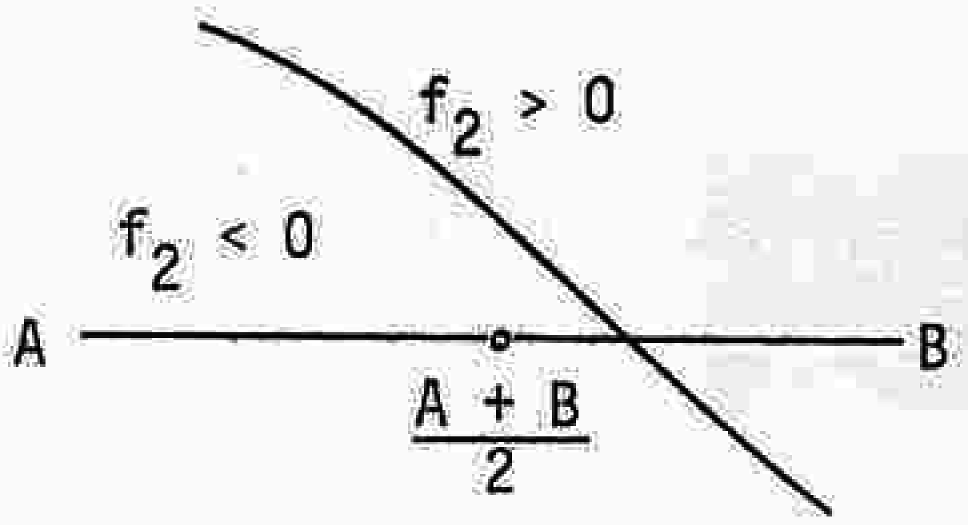


FIGURE 4.23

f_2 CHANGES SIGN

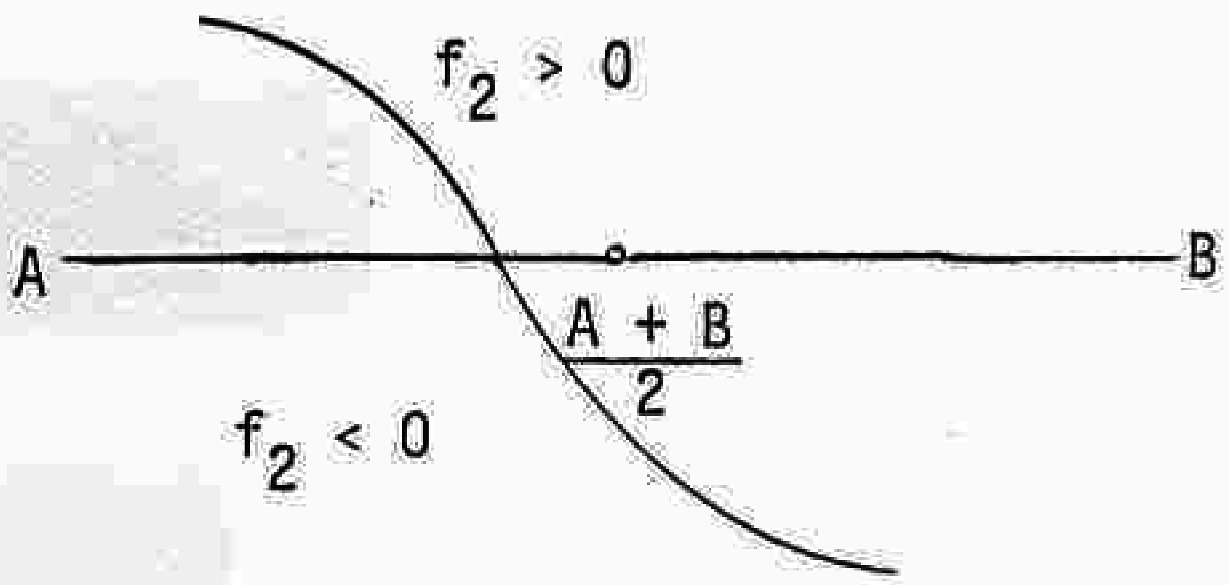


FIGURE 4.24

f_2 CHANGES SIGN

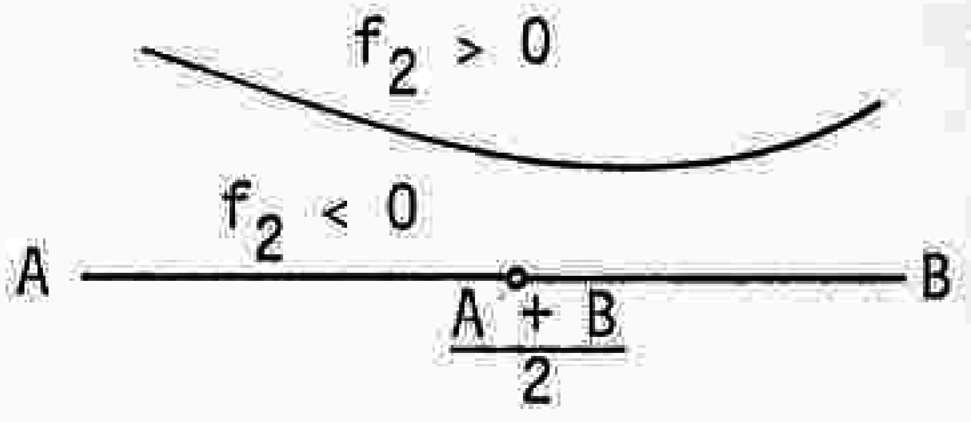


FIGURE 4.25

f_2 DOES NOT CHANGE SIGN

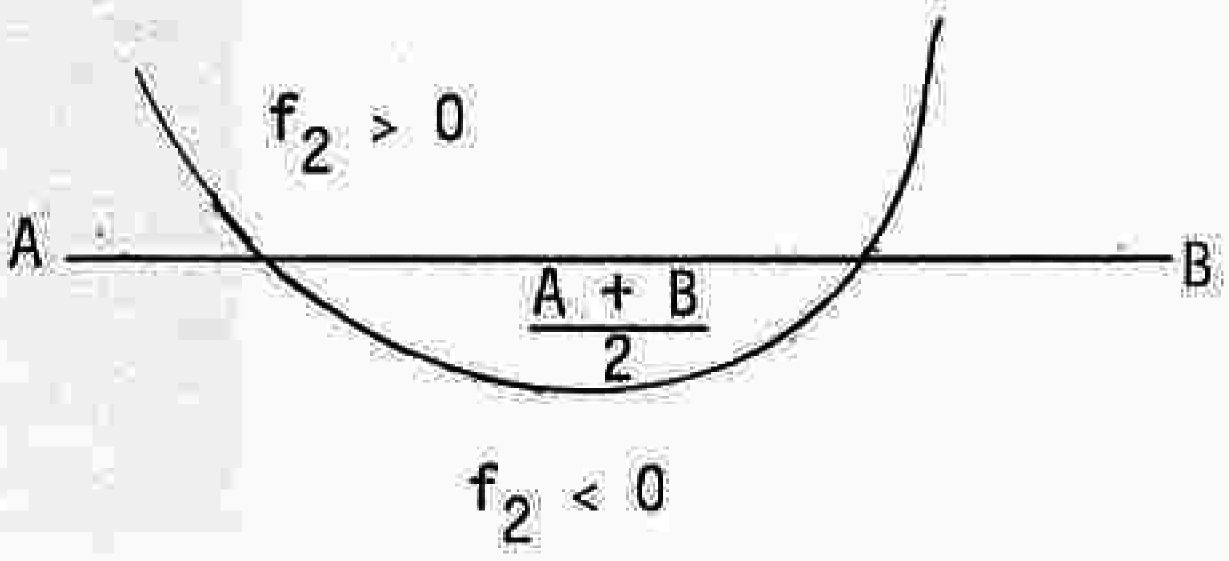


FIGURE 4.26

f_2 CHANGES SIGN TWICE

These figures illustrate the second test for sufficient refinement described in Section 4.2. It is assumed $f_2 > 0$ on $\langle A,B \rangle$ in all four figures.

Fig. 4.25

$$\operatorname{sgn} f_2 \left(\frac{A + B}{2}\right)$$

equals both sgn $f_2(A)$ and sgn $f_2(B)$, and neither $m_{1,2}$ nor $m_{2,2}$ would be set to 0. In Fig. 4.26

$$\operatorname{sgn} f_2 \left(\frac{A + B}{2}\right) = +1,$$

while sgn $f_2(A)$ = sgn $f_2(B) = -1$. Here, $m_{1,2} \leftarrow 0$ if $k = 1$ (i.e., if

$$\left\langle \frac{A + B}{2}, B \right\rangle$$

will be considered in the next iteration), and $m_{2,2} \leftarrow 0$ if $k = 2$ (if

$$\left\langle A, \frac{A + B}{2} \right\rangle$$

will be considered).

This last test seems to be the best to use in most cases. The test reduces to the question: Does f_i have the same sign at the midpoint of $\langle A,B\rangle$ as either $f_i(A)$ or $f_i(B)$? If the answer is no, this means that a sign change of f_i has been detected by including the point $(A + B)/2$, but which would not have been known if only

CHAPTER V

A GENERALIZED METHOD OF BISECTION

The purpose of this chapter is to introduce and analyze an algorithm for obtaining approximate solutions to $F(X) = \theta_n$ where $F:P \subset R^n \to R^n$ is continuous. This algorithm will be based on repeated use of Algorithm 4.4.1. In the first section of the chapter, "bisection" of n-simplexes is defined in a manner analogous to bisection of (n - 1)-simplexes. In the second section, Section 5.2, a relationship between bisection of an n-simplex and bisection of the (n - 1)-simplexes on its boundary is investigated. In Section 5.3, this relationship is used in an algorithm to compute approximate fixed points of F by repeated bisections of a simplex S for which $d(F,S,\theta_n) \neq 0$. Some possible occurrences when using this algorithm are analyzed with examples in Section 5.4.

5.1 Bisecting the n-dimensional Region

Suppose that $S = \langle X_0, X_1, \ldots, X_n \rangle$ is an n-simplex in R^n and $F:S \to R^n$. Suppose further that $d(F,S,\theta_n) \neq 0$ has been computed by Algorithm 4.4.1. Then by Kronecker's Theorem (Theorem 2.2.4), there exists a solution of $F(X) = \theta_n$ in the interior of S. With this situation, one can think of the entire region S as being an approximate solution of $F(X) = \theta_n$.

Such an approximate solution may be estimated more accurately by "bisection" of S. The n-simplex S is bisected in the same manner as described in Section 4.1 for (n - 1)-simplexes (cf. Fig. 5.1). This is formalized in the following definitions.

<u>5.1.1 Definition</u>. Let $S = \langle X_0, \ldots, X_n \rangle$ be an n-simplex, let $\langle X_k, X_m \rangle$ be the longest segment to be formed from the vertices of S and let $A = (X_k + X_m)/2$ be the midpoint of $\langle X_k, X_m \rangle$ (cf. Sec. 4.1). Then if:

$$S_1 = \langle X_0, X_1, \ldots, X_{k-1}, A, X_{k+1}, \ldots, X_m, \ldots, X_n \rangle$$

and

$$S_2 = \langle X_0, X_1, \ldots . X_k, \ldots, X_{m-1}, A, X_{m+1}, \ldots, X_n \rangle$$

one has:

$$S = S_1 + S_2 \ .$$

One says that the n simplex S has been <u>bisected</u>, and that $\{S_1, S_2\}$ is <u>a bisection of S</u>. One further refers to the n-simplex S_1 as the <u>lower simplex</u> and to the n-simplex S_2 as the <u>upper simplex</u> corresponding to the n-simplex S.

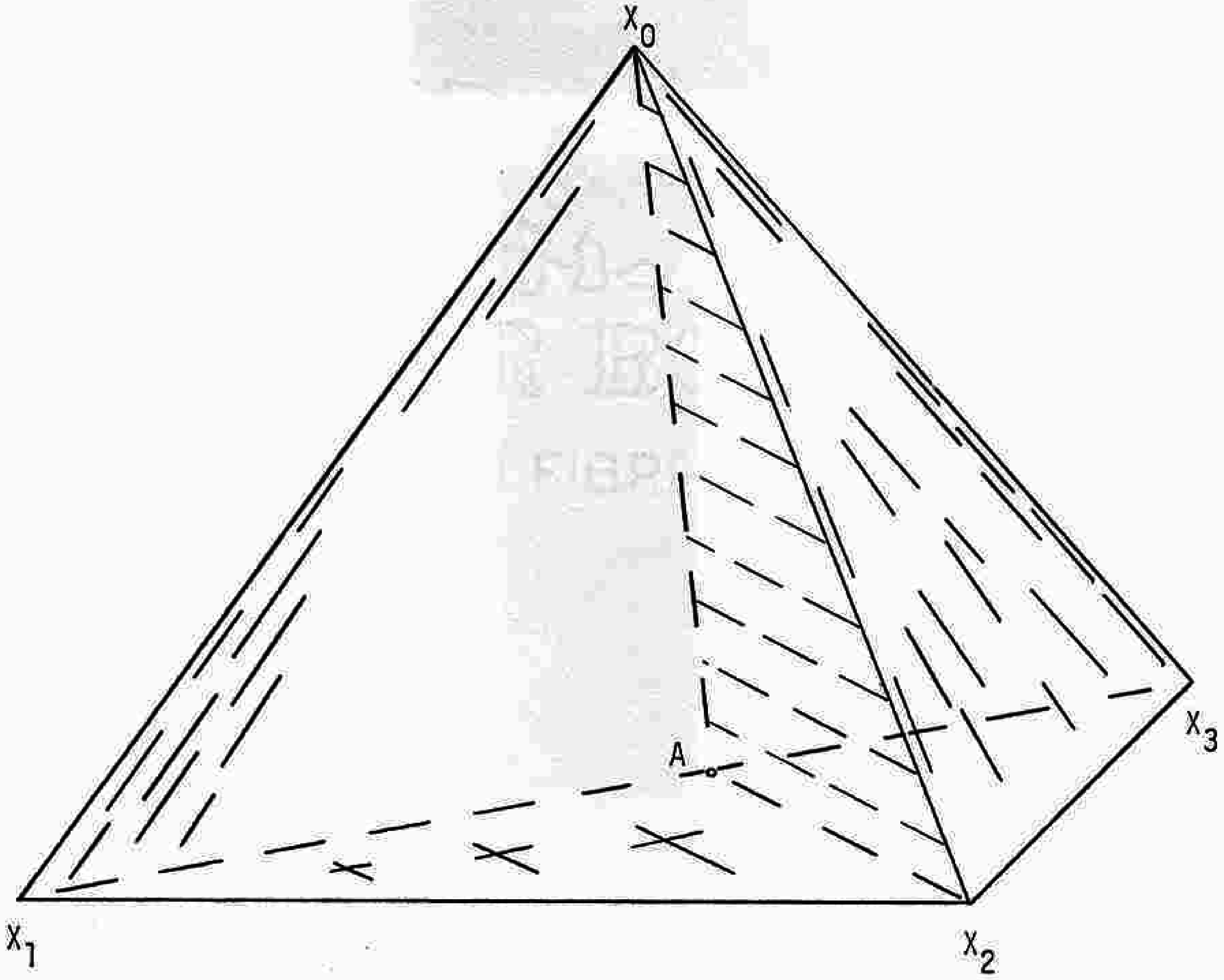


FIGURE 5.1

BISECTION OF n-SIMPLEXES WHEN n = 3

Suppose that the n-simplex S is bisected and that there are no solutions of $F(X) = \theta_n$ on $b(S_1)$ or $b(S_2)$. Then one has:

$$d(F,S,\theta_n) = d(F,S_1,\theta_n) + d(F,S_2,\theta_n) \quad \text{(see [26] p. 158 and [5].}$$

Hence, $d(F,S_i,\theta_n) \neq 0$ for $i = 1$ or $i = 2$ or both $i = 1$ and $i = 2$. If $d(F,S_i,\theta_n)$ is determined to be nonzero, there is a root of $F(X) = \theta_n$ in S_1. The diameter of S_i, however, is likely to be smaller than the diameter of S. The process thus stated can now be continued with S_i replacing S.

If $d(F,S_i,\theta_n)$ were found without using results of previous computations once $d(F,S,\theta_n)$ is calculated, there would be computational redundancy. The following section explains where the redundancy would occur and how it is avoided.

5.2 Using the Previous Tree Information

In this section, Theorem 5.2.1 and Theorem 5.2.3 set up a relationship between tree information for b(S), where S is the original n-simplex, and $b(S_1)$ and $b(S_2)$, where $\{S_1,S_2\}$ is the bisection of S.

5.2.1 Theorem. Suppose that S, $\langle X_k,X_m\rangle$, A, S_1, and S_2 are as in Definition 5.1.1. Then if $i \neq k$ and $i \neq m$, the ith facet (Def. 2.1.6) of S_1 is the lower simplex for the ith facet of S. Likewise, the ith facet of S_2 is the upper simplex for the ith facet of S.

5.2.2 Proof of Theorem 5.2.1. Let T_i, T_i^1, and T_i^2 denote the i^{th} facets of S, S_1, and S_2, respectively. Then:

$$T_i = (-1)^i \langle X_0, \ldots, X_k, \ldots, \hat{X}_i, \ldots, X_m, \ldots, X_n \rangle ,$$

$$T_i^1 = (-1)^i \langle X_0, \ldots, X_{k-1}, A, X_{k+1}, \ldots, \hat{X}_i, \ldots, X_m, \ldots, X_n \rangle ,$$

and

$$T_i^2 = (-1)^i \langle X_0, \ldots, X_k, \ldots, \hat{X}_i, \ldots, X_{m-1}, A, X_{m+1}, \ldots, X \rangle .$$

It is seen that T_i^1 is gotten from T_i by replacing the vertex X_k of T_i by A, and that T_i^2 is gotten similarly by replacing the vertex X_m of T_i by A. However, since the vertices of T_i are all vertices of S and $\langle X_k, X_m \rangle$ was the longest one-dimensional segment of S, a-fortiori, $\langle X_k, X_m \rangle$ is the largest one-dimensional segment of T_i. This shows that the bisection of T_i is $\{T_i^1, T_i^2\}$. Furthermore, X_k and X_m appear in the same order in the list of vertices for T as they did in the list of vertices for S, so T_i^1 is indeed the lower simplex and T_i^2 is the upper simplex.///

The next theorem pinpoints how much new information is needed to calculate $d(F,S_1,\theta_n)$ and $d(F,S_2,\theta_n)$, given the trees for $d(F,S,\theta_n)$.

5.2.3 Theorem. If S, S_1, S_2, and $\langle X_k,X_m\rangle$ are as in Definition 5.1.1, then the kth facet of S_1 is equal to the kth facet of S, whereas the mth facet of S_2 is equal to the mth facet of S. The mth facet of S_1 has no interior points in common with any facets of S. However, the mth facet of S_1 is equal to $-(-1)^{m-k}$ times the kth facet of S_2. Hence, if $U_i^j = (-1)^i T_i^j$, where $b(S_j) = \sum_{i=0}^{n}(-1)^i T_i^j = \sum_{i=0}^{n} U^j$, ($j = 1,2$) then $U_m^1 = -U_k^2$.

Theorem 5.2.3 is illustrated in Fig. 5.1 with $n = 3$, $k = 1$, and $m = 3$. There, $S_1 = \langle X_0,A,X_2,X_3\rangle$ lies to the right and rear, and $S_2 = \langle X_0,X_1,X_2,A\rangle$ lies to the left and front. The kth facet of S_1 is $\langle X_0,X_2,X_3\rangle$, which equals the kth facet of S. The mth facet of S_1 is $\langle X_0,A,X_2\rangle$, which equals the kth facet of S_2. The mth facet of S_2 is $\langle X_0,X_1,X_2\rangle$, which equals the mth facet of S. One sees in the figure that $\langle X_0,A,X_2\rangle$ is, indeed, a "new" face. This new face "cuts" all but the old kth and mth facets into bisections.

Thus, only the mth facet of S_1 or the kth facet of S_2 need be dealt with, once the trees for S have been calculated.

5.2.4 Proof of Theorem 5.2.3. The kth facet of S is:

$$(-1)^k\langle X_0,\ldots,\hat{X}_k,\ldots,X_m,\ldots,X_n\rangle$$

whereas the kth facet of S_1 is simply:

$$(-1)^k\langle X_0,\ldots,X_{k-1},\hat{A},X_{k+1},\ldots,X_m,\ldots.X_n\rangle\ .$$

Since the only difference between S and S_1 was their k^{th} vertex the above two facets are identical. A similar argument shows that the m^{th} facet of S equals the m^{th} facet of S_2.

To prove the remaining assertions of the theorem, consider the m^{th} facet of S_1:

$$T_m^1 = \langle X_0, \ldots, X_{k-1}, A, X_{k+1}, \ldots, X_{m-1}, \hat{X}_m, X_{m+1}, \ldots, X_n \rangle$$

where $A = \langle X_k + X_m \rangle / 2$. Since the vertices of S are linearly independent, it follows that the interior points of T_m^1 are interior points of S and hence, not on any face of S.

To complete the proof observe that

$$U_m^1 = (-1)^m T_m^1 = -(-1)^m (-1)^{m-k} T_k^2$$

$$= -(-1)^m (-1)^{m-k} (-1)^{-k} U_k^2$$

$$= -(-1)^{2(m-k)} U_k^2$$

$$= -U_k^2 \ .$$

This proves the last assertion in Theorem 5.2.3.///

5.3 An Economical Bisection Algorithm

The algorithm in this section is based upon results from Sections 5.1 and 5.2 which relate the facets of $S_i (i = 1,2)$ to the facets of S. The algorithm proceeds roughly as follows: once $d(F,S,\Theta_n)$ has been calculated for the original simplex S, the parity contributions of the ℓ^{th} facets of the lower simplex excluding $\ell = m$ are retrieved from the stored information, while Algorithm 4.4.1 is then applied only to the m^{th} facet of S. This gives all of the information necessary to determine $d(F,S,\Theta_n)$. (To determine $d(F,S_1,\Theta_n)$, the parity contributions for the ℓ^{th} facets of the upper simplex are retrieved from storage.) When S_1 is bisected, the new m^{th} (if $i = 1$) or k^{th} tree (if $i = 2$) replaces the old m^{th} or k^{th} tree, while the other existing trees continue to be used in subsequent iterations.

The following problem can arise with the above scheme. Under certain conditions, more subdivisions of the ℓ^{th} facet are performed in the bisection process than are carried out when Algorithm 4.4.1 was applied to the ℓ^{th} side of the original simplex. This occurrence is illustrated in R^2 in Fig. 5.2. There, the tree for the second side of $\langle X_0,X_1,X_2\rangle$ may not contain information for the segment $\langle A,B\rangle$, i.e., possibly no location pair (2,4) or parity contribution was stored for the segment $\langle A,B\rangle$. In such cases, the parity of the range simplex for the ℓ^{th} facet must be calculated directly and stored.

The bisection algorithm is formalized below as Algorithm 5.3.1. Here, $N1(i)$ denotes the value of the first location numbers of

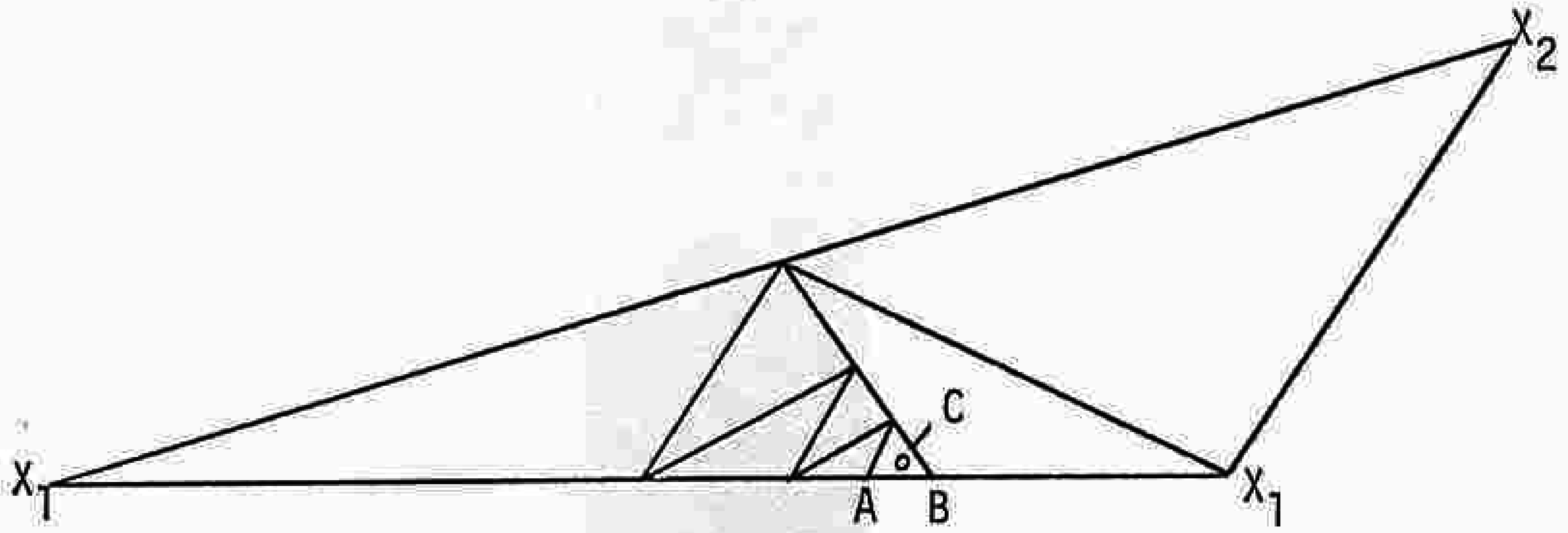


FIGURE 5.2

A ROOT OF $F(X) = \theta$ LIES WITHIN ΔABC

(19) $S_1 \leftarrow S_2$.

(20) Return to step (5).

The following remark simplifies execution of Step (17) of Algorithm 5.3.1.

5.3.2 Remark. The last statement in Theorem 5.2.3 can be used for Step (17) of Algorithm 5.3.1, so that the k^{th} tree for S_2 can be calculated from the m^{th} tree for S_1 without resorting to Algorithm 4.4.1 again. To get the tree for the k^{th} facet of S_2, we simply interchange upper and lower points, and then change the sign of all of the parity contributions. Interchanging the upper and lower points amounts to taking the binary complement of the first n_2 digits of n_1 in each location pair (n_1,n_2).

There are various modifications which can be carried out on Algorithm 5.3.1. The following remark hints at the nature of some of these.

5.3.3 Remark. Although not included in Algorithm 5.3.1, if a value of 0 is returned for $d(F,S,\theta_n)$, then it is clear that p must be larger. The parameter, p, can then be made larger and the parity contribution of the new side can then be recalculated with this larger value for p.

Flowcharts corresponding to Algorithm 5.3.1 will appear elsewhere.

the i^{th} facet of the current n-simplex in the i^{th} tree presently in storage; N2(i) denotes the corresponding second location number.

5.3.1 Algorithm.

(1) Read in the original simplex, S_1, the maximum number of iterations, MAXITR, the square of the error tolerance, E, and the selected parameters for Algorithm 4.4.1.

(3) Initialize: $N2(j) \leftarrow 0$, $N1(j) \leftarrow 0$, for $j = 1$ to $n + 1$. Also, $I \leftarrow 1$.

(3) Calculate $d(F,S_1,\theta_n)$ via Algorithm 4.4.1.

(4) If $d(F,S_1,\theta_n) = 0$, then stop. Otherwise continue to step (5).

(5) $I \leftarrow I + 1$

(6) If $I > MAXITR$, then stop. Otherwise continue to step (7).

(7) Bisect S_1. Store k and m if $\langle X_k,X_m\rangle$ is the longest one-dimensional segment. If the length of $\langle X_k,X_m\rangle$ squared is less than E^2, then print S_1 and stop.

(8) Store the resulting lower simplex from (7) in S_1, and the upper simplex in S_2.

(9) Find the m^{th} facet of S_1 and store in the nxn-matrix, BS_1.

(10) Do Algorithm 4.4.1 for BS_1 to obtain a new m^{th} table of tree information.

(11) Put the new tree information in the m^{th} tree, and store the old m^{th} tree in temporary storage in case $d(F,S_1,\Theta_n) = 0$.

(12) $N2(q) \leftarrow N2(q) + 1$ for $q = 1$ to $(n + 1)$ except $q = k$ and $q = m$.

(13) Set the $N2(q)^{th}$ digit of $N1(q)$ equal to zero unless $N2(q) = 0$, for $q = 1$ to $n + 1$. This will enable the algorithm to take information from the lower simplexes from the trees, which is appropriate for $b(S_1)$.

(14) $N2T \leftarrow N2(m)$, $N2(m) \leftarrow 0$.

(15) For $q = 1$ to $(n + 1)$, look in the table of tree information for the q^{th} facet to find the contribution of the q^{th} facet to the sum in Theorem 3.2.10 or 3.3.1. The location pair for this facet is ({first $N2(q)$ digits of $N1(q)$}, $N2(q)$). If there is no such location pair, then the number of bisections has exceeded the length of the branch previously calculated. In this case, find the q^{th} facet and calculate the parity of its range simplex directly. As the contributions of the q^{th} facet are being looked up or calculated, add them together to get $d(F,S_1,\Theta_n)$.

(16) If $d(F,S_1,\Theta_n) = 0$, then return the old tree information for the m^{th} facet to the permanent storage, and $N2(m) \leftarrow N2T$, $N2(k) \leftarrow 0$.

(17) Calculate a new tree for the k^{th} facet of S_2.

(18) Set the $N2(q)^{th}$ digit of $N1(q)$ equal to 1 unless $N2(q) = 0$, for $q = 1$ to $(n + 1)$. (This indicates that the upper simplex was chosen in this iteration.)

5.4 Applicability of the Method and Amount of Computation Required

If moduli of continuity or similar information is not given, nothing can be said about choosing the parameter p for Algorithm 4.4.1 (which computes contributions of facets). Furthermore, when the diameter of S_1 in Algorithm 5.3.1 is very small, it may take more work to compute the parity contribution of the m^{th} facet of S_1 (Step (10) of Algorithm 5.3.1) than it would if the diameter of S_1 were relatively large. For this reason, Algorithm 5.3.1 may converge slowly, but will still be quite suitable as a starting method for other procedures.

The action of the algorithm depends very much not only on the function, but also on the shape of S_1 and the relative location of the roots of F_1 in S. Notwithstanding, an analysis for the algorithm can often be made in specific cases. Examples in R^2 are illustrated in Figures 5.3, 5.4, and 5.5. In all three figures, the same level curves, $f_1(X) = 0$ and $f_2(X) = 0$, are drawn.

In order to assure a correct result, Algorithm 4.4.1 must subdivide b(S) so that an endpoint of a 1-simplex lies between each 0-set. On $\langle X_0,X_1\rangle(\langle X_0,X_2\rangle)$ in Fig. 5.3, the ratio of the distance between 0-sets to the length of $\langle X_0,X_1\rangle(\langle X_0,X_2\rangle)$ is very small. When this happens, many subdivisions will possibly be required in $\langle X_0,X_1\rangle(\langle X_0,X_2\rangle)$; i.e., a branch may need to be very long.

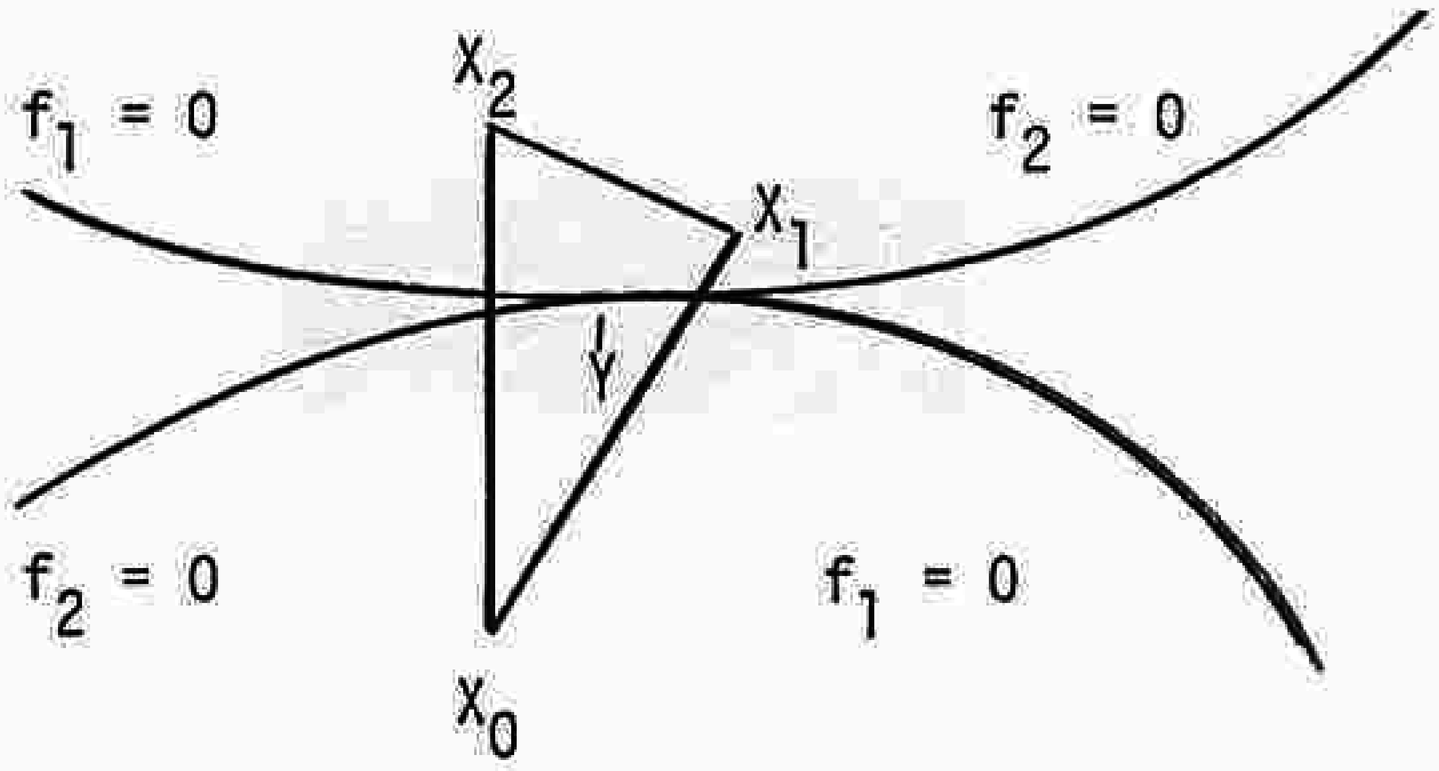


FIGURE 5.3

$S = \langle X_0, X_1, X_2 \rangle$

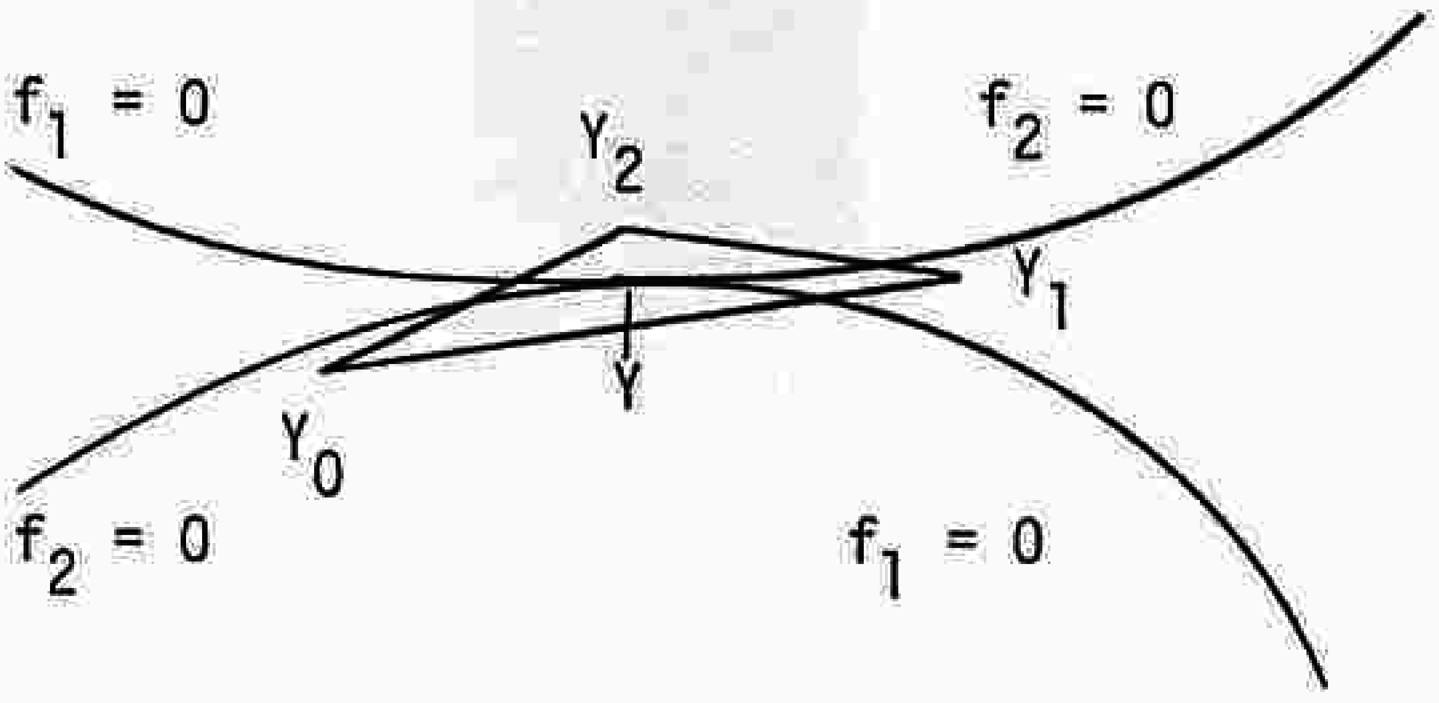


FIGURE 5.4

$S = \langle Y_0, Y_1, Y_2 \rangle$

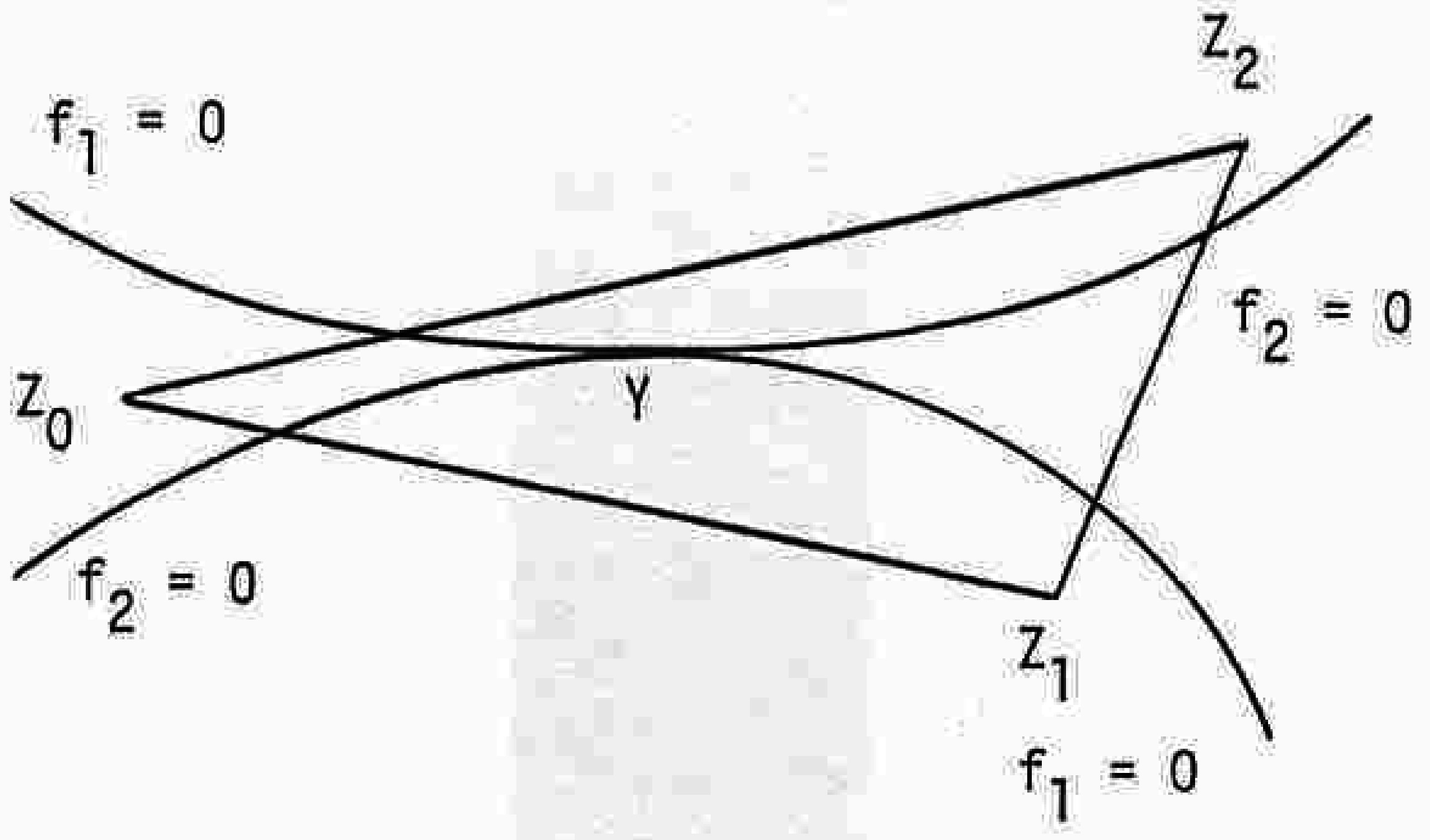


FIGURE 5.5

$S = \langle Z_0, Z_1, Z_2 \rangle$

In Fig. 5.4, the vertices of S are drawn so that there is ample distance between the 0-sets on $\langle Y_0, Y_2\rangle$; also, no further subdivisions need be carried out on $\langle Y_0, Y_1\rangle$ and $\langle Y_0, Y_2\rangle$.

In Fig. 5.5, a simplex with a relatively large diameter has been drawn. In this figure, there should be no problem attaining sufficient refinement (Def. 3.1.6). The following remark may help in such analyses.

5.4.1 Remark. Suppose that $F:S \to R^n$ is differentiable, and that $F(Y) = \theta_n$. Then each level surface $f_i(X) = 0$ $(i = 1,\ldots,n)$ is normal at $X = Y$ to the vector represented by the i^{th} column of $J(F)(Y)$. Suppose also that $|J(F)(Y)|$ is very small or equal to zero, as in Figs. 5.3-5.5: Then consider a sphere of small radius with center at Y. Then the points of intersection of the level surfaces $F_i(X) = 0$ with the surface of that sphere are very close together.

It may or may not be the case that the points of intersection of the level surfaces with the boundary of a given simplex containing Y are also close together (Figs. 5.3-5.5). However, the magnitude of $J(F)(Y)$ can still provide some indication of possible excessive branch lengths.

As a final note, the following property of $d(F,S,\theta_n)$ is considered.

5.4.2 Remark ([1], [16], [26], [30]). $d(F,S,\theta_n)$ depends only on the values F assumes on $b(S)$. Thus F may have pathologies or areas of nonzero measure where $J(F) = 0$ in the interior of S, but a value for $d(F,S,\theta_n)$ can still be computed. Moreover, this value may be given a useful interpretation as in Fig. 5.6. There is a large area within $S = \langle X_0, X_1, X_2 \rangle$ where $J(F)(X)$ vanishes. However, Algorithm 4.4.1 will calculate $d(F,S,\theta_n) = 1$; this indicates that there is a nonempty set $A \subset S$ such that $F(A) = \{F(X): X \in A\} = \{\theta_n\}$.

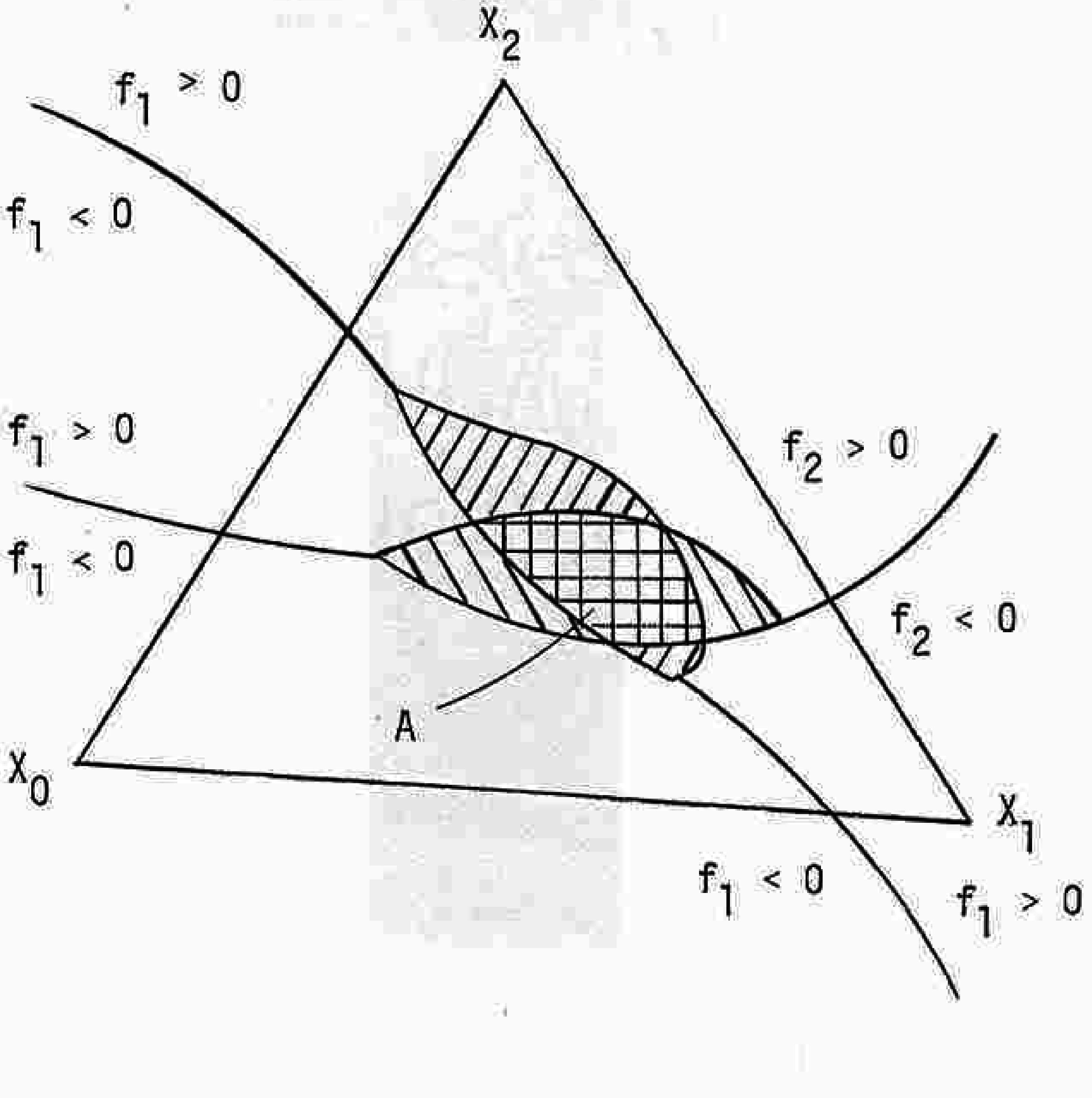


FIGURE 5.6

$(f_1, f_2) = (0,0)$ ON THE SET A

CHAPTER VI

RELATIONSHIP TO OTHER TOPOLOGICAL METHODS AND ALGORITHMS

As mentioned earlier (Sec. 1.2), many problems require solution of systems of equations for which Newton's Method and the related class of algorithms (see [26], Ch. 7) fail due to nonexistence of derivatives or poorly behaved partial derivatives. Also, Newton's method often converges to a solution X_1 of $F(X) = \Theta_n$ almost independently of the initial guess, while $F(X) = \Theta_n$ may have several solutions, all of which are desired for the application [3]. Because of this, various approaches based upon topological fixed-point theorems, and, in particular, methods for finding "Sperner simplexes" (containing approximate fixed points) have been investigated.

As was mentioned in Chapter I, an important contribution to this type of method came from Scarf [27]. The algorithm in [27] is efficient in terms of memory and computations, desirable in that it often detects approximate fixed points (not merely true fixed points), and esthetic in so far that it mimics the famous Simplex Method for linear programming (Dantzig) in the sense of being an exchange algorithm. Nonetheless, to enlarge the variety of problems that can be solved, variations and other algorithms were developed. Allgower and Keller [4] worked with another method of subdivision, applicable on the unit n-cube, and a way of labeling vertices of simplexes; Jeppson [18] extensively tested the method from [4].

B. C. Eaves [10] also investigated the subject. Once an approximate fixed point has been found via the method in [10], the effort expended could be used to obtain a better approximation; the method in [10] was also extended to work for unbounded regions.

Interestingly, all of the above are closely related to Theorem 3.2.10 and Algorithm 4.4.1: A range simplex, $\mathcal{R}(S,F)$, is useable if and only if S has a "complete set of labels" (see [18] and Sec. 6.1 of this chapter).

This chapter is devoted to a brief discussion of the algorithms mentioned above and the relationships existing between them. Background and a definition of Sperner simplexes are presented in Section 6.1, while an elementary discussion of methods appears in Section 6.2. In Section 6.3, the relationship between such methods and Algorithm 4.4.1 is discussed. In Section 6.4, yet another algorithm combining ideas from Section 4.2 and Section 6.1 is set down. Section 6.5 is devoted to comparisons.

6.1 Sperner Simplexes and Sperner's Lemma

This section introduces necessary background for subsequent discussions of algorithms.

<u>6.1.1 Definition</u>. Suppose that $F = (F_{\ },\ldots,F_{\ }):\mathcal{D} \to R^n$ is continuous on the closed, bounded domain $\mathcal{D} \subset R^n$, so that $G = F - I$ is continuous on $\mathcal{D}$, where I is the identity map in R^n. Now, letting $S = \langle X_0,\ldots,X_n\rangle$ be a simplex in $\mathcal{D}$, define a vector $L(X_i)$, by $\underline{L(X_i) = -\mathrm{Sgn}\ (G(X_i))}$, where $\mathrm{Sgn}(\cdot)$ is as in Definition 3.1.4.

<u>6.1.2 Definition</u>. ([18], pp. 122-123 or [3], p. 2, etc.). Assign a label ℓ_i to X_i by letting ℓ_i be the number of ones encountered in the vector $L(X_i)$ to the left of the first -1. This label will be called the <u>label of X_i induced by F</u>.

<u>6.1.3 Assumption</u>. Suppose S is an n-simplex, $F:S \to R^n$, and $\mathcal{S}$ is a subdivision of S. Suppose $\langle X_o,\ldots,X_k\rangle$ is any k-dimensional facet of S_0, where some subdivision of S_0 is contained in $\mathcal{S}$; suppose $A \in \langle X_0,\ldots,X_k\rangle$. Then assume $L(A) = L(X_j)$ for some $j \in \{0,\ldots,k\}$.

Properties of the labeling induced by F needed to define Sperner simplexes and state Sperner's Lemma are now defined.

<u>6.1.4 Definition</u>. If S is an n-simplex in R^n, $\mathcal{S}$ is a subdivision of S and $F:S \to R^n$, then the labeling of the vertices of elements of $\mathcal{S}$ induced by F is proper if and only if the points, A, forming vertices of elements of $\mathcal{S}$ follow assumption 6.1.3.

<u>6.1.5 Definition</u>. The set of labels of the vertices of S is said to be complete if and only if $\{\ell_o,\ldots,\ell_n\} = \{0,\ldots,n\}$.

<u>6.1.6 Definition</u>. Simplexes with complete sets of labels are called Sperner simplexes.

It has been proven that Sperner simplexes will contain approximations to fixed points (infra Theorem 6.2.2 and corresponding references).

The following theorem underlies the working of Sperner simplex algorithms.

<u>6.1.7 Theorem (Sperner's Lemma [6])</u>. Suppose that S is an n-simplex in R^n, that $\mathcal{S}$ is a subdivision of S and that the vertices of S and elements of $\mathcal{S}$ are labeled with a proper labeling. Suppose further that S bears a complete set of labels. Then one of the elements of $\mathcal{S}$ also has a complete set of labels.

Sperner's Lemma is proven in [6].

<u>6.2 Methods Derived from Sperner's Lemma</u>

The methods described in this section are based upon the fact that Sperner simplexes can be considered approximations to fixed points X, such that $F(X) = X$, of maps $F:\mathcal{D}\subset R^n \rightarrow \mathcal{D}$. The problems are frequently set up in one of two anologous settings, depending upon whether the region $\mathcal{D}$ is an n-simplex or an n-cube. These two settings will be considered in the theorems to follow.

Suppose first that $\mathcal{D} = S = \langle X_1,\ldots,X_n\rangle$ is an n-simplex in R^n, so that $F:S \rightarrow S$. Then points in S and the components of F can be given in terms of barycentric coordinates ([13], pp. 36-37): $P \in S \Rightarrow P = (\pi_o, \pi, \ldots, \pi_n)$: $\Sigma_{i=0}^{n} \pi_i X_i = P$, $\Sigma_{i=0}^{n} \pi_i = 1$, $\pi_i \geq 0$ for $i = 1, \ldots, n$; similarly, $F = (f_o, \ldots, f_n)$, $F(P) = \Sigma_{i=0}^{n} f_i(P) X_i$; $\Sigma_{i=0} f_i(P) = 1$, $f_i(P) \geq 0$ for $i = 1, \ldots n$. Then:

<u>6.2.1 Remark ([27], pp. 1328-1329)</u>. Suppose $F: S \to S$ is given as above, and that the labeling of the components of F is given as in Definition 6.1.2. Then S has a complete set of labels.

In this setting the hypotheses of Sperner's Lemma (Theorem 6.1.6) are satisfied. Coupled with the following theorem, this provides a basis for algorithms to find fixed points.

<u>6.2.2 Theorem</u>. Let F and S be as above. Then, given $\varepsilon > 0$, there is an N such that if S' is an element of some subdivision of S with the diameter of S' less than 1/N, and S' is a Sperner simplex, then $\|F(X) - X\| < \varepsilon$ for all $X \in S'$.

Scarf [27] developed an algorithm from this setting. In that algorithm, a clever way to test the points corresponding to vertices of simplexes in the subdivision in order to locate a Sperner simplex was worked out.

A second important setting for Sperner's Lemma is when $\mathscr{D} = C^n$, the unit n-cube. The n-cube is "triangulated" in some standard way ([4], p. 159), and the labels corresponding to the resulting simplexes are analyzed. In this setting, the conclusion of Theorem 6.1.5

must hold for the hypotheses of Sperner's Lemma to be satisfied. However, under mild assumptions, the existence of a Sperner simplex is ascertained ([4]), and an exact analogue of Theorem 6.2.2 holds ([3], Theorem 3).

In each of the above settings, algorithms were developed whereby successive simplexes or component simplexes of a subdivision were checked for a complete set of labels.

The next section deals with a relationship between such methods and Algorithm 4.4.1.

6.3 Relationship between Sperner's Lemma and Useable Simplexes

We first establish a relationship between Par($\mathcal{R}$(S,F)) and labelings for Sperner simplexes. Take the n-simplex $X = \langle X_0, X_1, \ldots, X_n \rangle \subset R^n$, and let $F = (f_1, \ldots, f_n): S \to R^n$ be continuous. Consider any facet T of S, and suppose that $f_1(X) > 0$ for $X \in T$. Then for $X = (x_1, \ldots, x_n) \in T$, define $\check{F}_1: T \to R^{n-1}$ by $\check{F}_1(X) = (f_2(X), \ldots, f_n(X)) + (x_2, \ldots, x_n)$. Then the following theorem holds.

<u>6.3.1 Theorem</u>. $\mathcal{R}(T, F_1)$ is useable if and only if the labeling of T induced by $\check{F}_1$ is complete, i.e., if and only if T is a Sperner simplex for $\check{F}_1$. Furthermore, if $T = \langle X_{j_1}, \ldots X_{j_n} \rangle$, then the label, ℓ_i, corresponding to X_{j_i}, is equal to the number k_i from Algorithm 4.4.2.

<u>6.3.2 Proof of Theorem 6.3.1</u>. Theorem 6.3.1 follows directly from Definition 3.2.2 and the definition of the k_i in Algorithm 4.4.2.

Theorem 6.3.1 provides another way of looking at Theorem 3.2.10. Take the Sperner simplexes T to be approximations to fixed points of $\check{F}$; this is equivalent to taking them to be approximations to points X where $F_1(X) = \theta_{n-1}$. Hence, the Sperner simplexes in Theorem 6.3.1 correspond to approximations to points at which $F/\|F\| = (1,0,\ldots,0)$.

It is evident that useable simplexes with positive parities correspond to coverings of $(1,0,\ldots,0) \in S^n$ by $F/\|F\|$ with a positive orientation (cf. Def. 2.2.1), whereas useable simplexes with negative parities correspond to coverings of $(1,0,\ldots,0)$ with a negative orientation. In particular, we have:

<u>6.3.3 Theorem</u>. Let $F:\mathcal{D}\subset R^n \to R^n$ be as before. Let W be the only point in $b(\mathcal{D})$ with $F(W)/\|F(W)\| = (1,0,\ldots,0)$. Suppose $(1,0,\ldots,0)$ is covered with a positive (negative) orientation. Suppose that $b(\mathcal{D})$ is sufficiently refined relative to F, with a subdivision, $\mathcal{S}$. Let ν be the number of $\mathcal{R}(S,F)$, $S \in \mathcal{S}$, with $\mathrm{Par}(\mathcal{R}(S,F)) = 1(-1)$. Then $\nu = 1 \bmod 2$.

Theorem 6.3.3 is a direct consequence of Definition 2.2.1 and Theorem 3.2.10.

Although Theorem 6.3.1 and Theorem 6.3.3 point to a correspondence between Sperner simplexes for $\check{F}_1$ and useable simplexes, it should be noted that Theorem 3.2.10 would not follow directly from arguments involving Sperner's Lemma. In particular there is no guarantee that there are no more Sperner simplexes than fixed points. Furthermore, the hypotheses of sufficient refinement show from a different angle what is needed for fixed points to exist.

6.4 A Sperner Simplex Search Using Binary Trees

Ideas previously outlined in this chapter suggest that we can search for Sperner simplexes by taking bisections of a n-simplex. A tree can be produced as in Algorithm 4.4.1 to keep track of where Sperner simplexes are located, but no points need be stored. Checking to see if a simplex is a Sperner simplex is similar to determining the useability of a range simplex. The following algorithm results.

6.4.1 Algorithm.

```
(1) Input the maximum branch length, MAX.
(2) Input the original n-simplex in $S_1$.
(3) Input the diameter tolerance, EPS.
(4) $n_1 \leftarrow 0$, $n_2 \leftarrow 0$
(5) $n_2 \leftarrow n_2 + 1$
(6) Bisect $S_1$ and store the lower simplex in $S_1$ and the
upper simplex in $S_2$. Store the diameter of $S_1$ in DIAM.
(7) If the $n_2$th digit of $n_1$ is 1, then switch the values in
the arrays $S_1$ and $S_2$.
```

(8) $SA(n_2) \leftarrow S_1$

(9) If DIAM $\geq$ EPS and $n_2 < MAX$, return to step (5).

(10) Print the integer represented by first n_2 binary digits of n_1 and n_2.

(11) Check to see if S_1 is a Sperner simplex. If it is, then print S_1.

(12) If $n_2 > MAX$, print out a message.

(13) If n_2th of n_1 is 0, go to step (16).

(14) $n_2 \leftarrow n_2 - 1$.

(15) If $n_2 = 0$ then stop. Otherwise, return to step (13).

(16) Set the n_2th digit of n_1 equal to 1 and all subsequent digits equal to zero.

(17) $S_1 \leftarrow SA(n_2)$

(18) $n_2 \leftarrow n_2 - 1$

(19) Return to step (5).

The above algorithm can be modified further; $d(F,S_1,\theta_n)$ can be calculated at certain stages, etc.

6.5 Comparisons

All of the methods mentioned in this chapter require that F be evaluated only to sufficient accuracy to determine Sgn(F(X)) or Sgn(F(X) - X), but demand varying amounts of memory and are applicable to different types of problems. Except for Algorithm 5.3.1, the differences occur mainly in the method of subdivision or formation of the simplexes or the method of carrying out the search.

The algorithm [27] requires an a priori selection of a set of points to be vertices. Essentially, only those points need be stored as the algorithm progresses, yet no two simplexes are ever considered twice. The algorithm terminates when a Sperner simplex S is found. The diameter of such S can be made small by choosing the initial points close together so that S is a Sperner simplex $\Rightarrow \exists\ X \in S$ (ibid.), such that $\|F(X) - X\| < \varepsilon$ (Theorem 6.2.2). In the algorithm as first stated (ibid.), however, S may not contain a true fixed point; finding an X with $\|F(X) - X\| < \varepsilon$ would involve choosing a new set of initial points and completely repeating the algorithm.

The algorithm [4] is set up to find an approximate fixed point in the unit n-cube. Only the "mesh size" need be specified at the start, for the algorithm systematically subdivides the n-cube into smaller cubes of the specified size. The algorithm can be set up to check all cubes for fixed points, so more than one approximate fixed point may be found. Since the algorithm proceeds according to a fixed pattern, little memory is required. If the mesh width is not small enough etc., however, some fixed points may not correspond to Sperner simplexes, and some Sperner simplexes may not correspond to fixed points. One would then specify a smaller mesh size and carry out the algorithm again.

Other approaches (in particular [10]) have also been tried.

In the method of bisection, neither an initial grid of points nor a mesh width is specified. Instead, the parameter p (Algorithm 4.4.1) is chosen in an attempt to vary the "mesh width" according to how smooth F is. Information from previous calculations is stored in a binary tree. The diameters of S_1 and S_2 (Algorithm 5.3.1) may become small very slowly if n is large, so that p must be large (it is recommended that $p \geq n$ for most applications), and, hence, the amount of information in the tree must be large. It was found that for the 5-dimensional example in Appendix A.1, one of the trees needed approximately 400 points (albeit a rudimentary test for sufficient refinement was used). If bisection is carried out, $n + 1$ trees must be stored. However, if $d(F, \mathcal{D}, \theta_n)$ only is desired, storage for less than one tree is needed (cf. Section 4.6). At present, no rigorous way of determining p or finding when a correct result has been obtained has been employed; we have experimented by varying p and varying the vertices of the initial simplex slightly.

One does not assume that F maps the initial region into itself in order to apply Algorithm 5.3.1. Furthermore, Algorithm 5.3.1 can be applied directly to any simplex contained in the domain of F.

At the writing of this work, there seems to be an explosion in the activity of development and testing of such algorithms and the comparisons in this paper barely hint at the results which have been obtained. Results of further testing and comparison will be indeed interesting.

CHAPTER VII

FURTHER APPLICATIONS AND EXAMPLES

The problems described below can be solved via Algorithm 4.4.1 and Algorithm 5.3.1.

7.1 Minimization Problems

Suppose that $S \subset \mathbb{R}^n$ is an n-simplex, and $F:S \to \mathbb{R}$ is continuous. Then knowledge of points in S at which F takes on maximal or minimal values is often desired.

A standard method of obtaining this knowledge is to compute the stationary points of F in S, then test them to determine whether they are local maximae or local minimae.

Two alternative cases immediately come to mind for computing such stationary points via degree theory:

(i) F has continuous derivatives, and these derivatives can be calculated formally. In this case, one obtains $F' = (F_1,\ldots,F_n):S \to \mathbb{R}^n$ is continuous, where F_i is the ith partial of F. One could then apply Algorithms 4.4.1 and 5.3.1 directly to F'.

(ii) It is inconvenient to formally differentiate F. Finite differences of F are then used for Algorithms 4.4.1 and 5.3.1. This second alternative is explained below.

Note that only Sgn(F') is required to be correct in Algorithm 4.4.1. Hence it should not be too difficult to obtain sufficiently accurate approximations (less than one significant digit is required) to derivatives via central differences. In addition, average rates of change may thus be computed and used even when F' is not continuous.

The following algorithm is suggested to replace evaluation of $\mathrm{sgn}(\partial F/\partial x_i)$. It uses a scheme similar to the one in Algorithm 4.3.1 which determined when to stop bisection.

In Algorithm 7.1.1, the notation $E_i = (0,0,\ldots,0,1,0,\ldots,0) \in R^n$ where the "1" occurs in the i^{th} position, will be used.

<u>7.1.1 Algorithm</u>.

(1) Choose, a priori, an integer parameter q, a small number, δ; and a large number, N.

(2) Input i and the point $X = (x_1,\ldots,x_n) \in R^n$ at which $\mathrm{sgn}(\partial F/\partial x_i)(X)$ will be approximated.

(3) $j \leftarrow 0$

(4) $J_2 \leftarrow \mathrm{sgn}(F(X + \delta E_i) - F(X - \delta E_i))$

(5) $J_1 \leftarrow J_2$

(6) $\delta \leftarrow \delta/M$

(7) $J_2 \leftarrow \mathrm{Sgn}(F(X + \delta E_i) - F(X - \delta E_i))$

(8) If $J_1 = J_2$, then $j \leftarrow j + 1$. Otherwise, $j \leftarrow 0$.

(9) If $j \geq q$, then return the value of J_1 for $\mathrm{sgn}(\partial F/\partial x_i)(X)$. Otherwise go back to step (5).

In the above algorithm, when X is a vertex of $S = \langle X_0,\ldots,X_n\rangle$, it is assumed that F is defined in a neighborhood of S. If this assumption is inconvenient, then, for each fixed vertex X_k take directional derivatives along the directions, $V_{0,k},\ldots,V_{n,k}$, of the 1-dimensional sides, $\langle X_o,X_k\rangle,\langle X_1,X_k\rangle,\ldots,\langle X_{k-1},X_k\rangle;\ \langle X_{k+1},X_k\rangle,\ldots,\langle X_n,X_k\rangle$. A method of doing this is explained below. First, however, a property which these directions must have is discussed.

<u>7.1.2 Remark</u>. Consider the linear transformations, $L_0,\ldots,L_n$, defining $n(n + 1)$ direction vectors $\{V_{i,j}\}$, $j = 0,\ldots,n$, $i = 1,\ldots,n$, in terms of $E_1,\ldots,E_n$ at each of the points $X_0,\ldots,X_n$ by $V_{i,j} = L_j(E_i)$. Then it will be assumed that there is a continuously parametrized family of linear transformations L_X such that $L_{X_i} = L_i$ for $i = 0,\ldots,n$, and such that L_X is nonsingular for $X \in S$.

For, otherwise,

$$\text{(i)}\quad \begin{pmatrix} F_{V_1(X)}(X) \\ \\ \\ F_{V_n(X)}(X) \end{pmatrix} = L_X \begin{pmatrix} F_{E_1}(X) \\ \cdot \\ \cdot \\ \cdot \\ F_{E_n}(X) \end{pmatrix} = \theta_n$$

might have a solution in S for which not all of the partials, F_{E_i} are zero.

A scheme for producing a set of direction vectors with this property is set forth below.

7.1.3 Definition. Choose $V_{i,j} = V_i(X_j)$ by:

$$V_i(X_j) = \left\{ \begin{array}{ll} X_j - X_{i-1} & i \leq j \\ & \\ X_i - X_j & k > j \end{array} \right\}$$

When S is then bisected by substituting, e.g., the point $A = (X_k + X_m)/2$ for, e.g., X_k then only the direction vectors involving X_k are changed, excepting $X_m - X_k$ (assuming $m > k$) or $X_k - X_m$ (assuming $k > m$). Now, $X \leftarrow A$, and direction vectors for the changed directions are then recalculated via the same formula.

7.1.4 Remark. The conditions in Remark 7.1.2 are satisfied if and only if $\{V_i(X_j)\}_{i=1}^{n}$ defines the same orientation of R^n for all j. It is easy to show that the above scheme for choosing $V_i(X_j)$ gives this condition and that the orientations at the points of either component simplex in any bisection are equal to the common original orientation.

7.1.5 Remark. It can be shown that the degree of the mapping $F : R^n \rightarrow R^n$ with partial derivatives taken in the coordinate directions is plus or minus the degree of the mapping represented by the left member of (i) in Remark 7.1.2. This is because L_X is homotopic to either the identity matrix or minus the identity matrix in $GL_n(R)$, so that $G(X) = F(X)$ and $H(X) = L_X F(X)$ are homotopic via a homotopy whose range does not include θ_n (see [26], p. 156, Theorem 6.22). Furthermore, $d(G,S,\theta_n) = \eta d(H,S,\theta_n)$, where $\eta = \operatorname{sgn}\det(L_{X_i})$ [15].

7.1.6 Remark. It should be noted that, under additional assumptions on F, it may be possible to find the critical points of F in a more direct manner. For example, if $F:\mathcal{D} \to F(\mathcal{D})$ maps $b(\mathcal{D})$ into $b(F(\mathcal{D}))$, F has a critical point within $\mathcal{D}$ if $d(F,\mathcal{D}\,\theta_n) \neq \pm 1$. This results from the fact that mappings of the n - disk, D^n, to D^n which map the surface of D^n to the surface of D^n must have at least one critical point if $n \geq 2$([3],

7.2 The Hidden Line Problem

For various applications, one often wishes to know whether a given line or curve in R^3 passes through a surface, or whether a given point $A \in R^3$ lies inside or outside of a closed surface (such as the surface of a sphere) in R^3. These problems, just as their generalizations to R^n, (See [1], pp. 497-498) can sometimes be solved with Algorithm 4.4.1.

If the region $\mathcal{D}$ enclosed by the surface is an n-simplex or a polygon, then the second problem can be solved by simply calculating $d(I - A, \mathcal{D}, \theta_n)$, where I is the identity map on $\mathcal{D}$. The case in which the surface enclosed a curved region $\mathcal{D}$ (an n-manifold) can also be done simply (curved regions are considered in Section 7.3).

To consider the problem of the intersecting line and surface, suppose that the line or curve is parameterized by:

$$\left\{\begin{array}{ll} x = f_1(t) & \\ y = f_2(t) & a \leq t \leq b \\ z = f_3(t) & \end{array}\right\}$$

and that the surface has the parametrization:

$$\left\{\begin{array}{ll} x = g_1(u,v) & \\ y = g_2(u,v) & c_1 \leq u \leq d_1 \\ & c_2 \leq v \leq d_2 \\ z = g_3(u,b) & \end{array}\right\}$$

Then define a function $F{:}P = [a,b]x[c_1,d_1]x[c_2,d_2] \to R^3$ by:

$$F(t,u,v) = (f_1(t) - g_1(u,v),\ f_2(t) - g_2(u,v),\ f_3(t) - g_3(u,v))$$

In this case, the results can be checked by solving the system directly. Equating each of the three components of F to zero gives:

$$\left\{\begin{array}{l} t - u - v = 0 \\ 2t - u^2 - v^2 = 0 \\ v = 0 \end{array}\right\} \quad \text{where} \quad \left\{\begin{array}{l} t = u \\ u^2 = 2t \end{array}\right\}$$

whence $4(u - 2) = 0$. Hence, $(2,2,0)$ is a solution. Since

$$J(F)(t,u,v) = 2(1 - u),\ J(F)(2,2,0) = -2 < 0,$$

Comparison shows that the degree calculations were correct.

The curve so parameterized now passes through the surface if and only if there is a point $(t_o,u_o,v_o) \in P$ such that $F(t_o,u_o,v_o) = \theta_3$. The following example will illustrate the calculation of $d(F,P,\theta_3)$ to ascertain the existence of such a solution.

7.2.1 Example. Show that the line L given by:

$$\left\{\begin{array}{l} x = t \\ y = 0 \\ z = 2t \end{array}\right\}$$

intersects the surface M given by:

$$\left\{\begin{array}{l} x = u \\ y = v \\ z = u^2 + v^2 \end{array}\right\}$$

for some t: $1 \leq t \leq 4$.

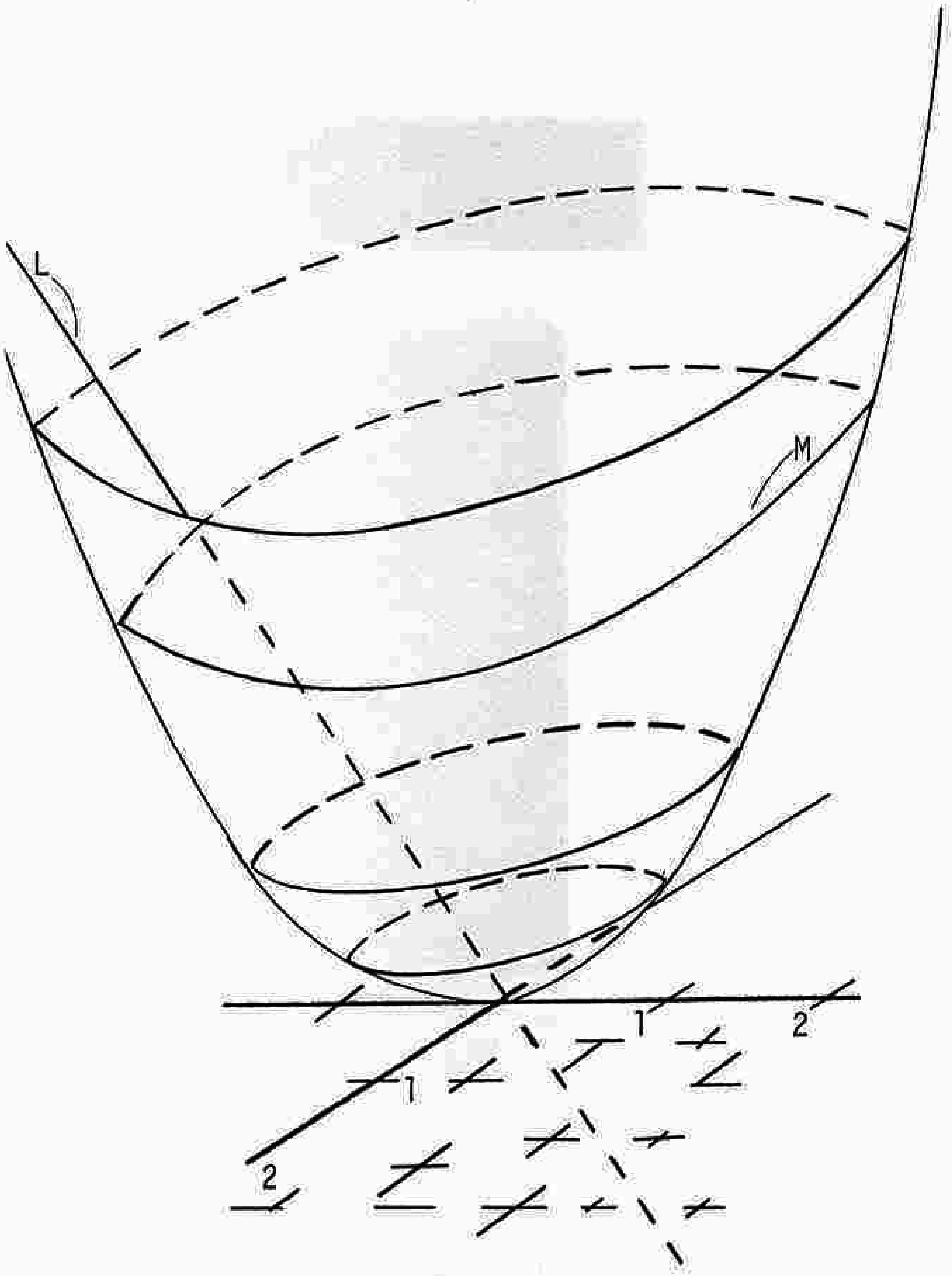


FIGURE 7.1

THE LINE AND THE SURFACE IN EXAMPLE 7.2.1

7.2.2 Calculations for Example 7.2.1. Set $F(t,u,v) = (t - u, -v, 2t - u^2 - v^2)$, and apply Algorithm 4.4.1 to F. To do this, one must choose, a priori, a simplex $S = \langle X_1, X_2, X_3, X_4 \rangle$ in $t - u - v$ space. If care is taken that the determinant $\Delta_4((1,X_1),(1,X_2),(1,X_3),(1,X_4))$ is not small, the volume of S will be large, and there will be less chance that a solution of F lies outside S. One may choose

$$S = \langle (3,0,0),(4,4,0),(1,3,-1),(3,3,2) \rangle.$$

It can be verified (see Figs. 7.2 and 7.3) that this S contains a considerable portion of the surface: $\{(t,u,v): 1 \leq t \leq 4\}$ and that, for all $(t,u,v) \in S$, $1 \leq t \leq 4$.

The parameter p of Algorithm 4.4.1 was chosen to be 2. The resulting trees produced are drawn in Appendix A.2. The degree of F was found to be -1.

7.3 The Degree with Respect to Regions Other than Polygons

Remark 2.2.8 introduces the fact that $d(F,\mathcal{P},\theta_n)$ is defined for straight-line regions other than polygons. To calculate $d(F,\mathcal{P},\theta_n)$, simply find all of the facets of:

$$b(\mathcal{P}) = b(P_1) + \ldots + b(P_n)$$

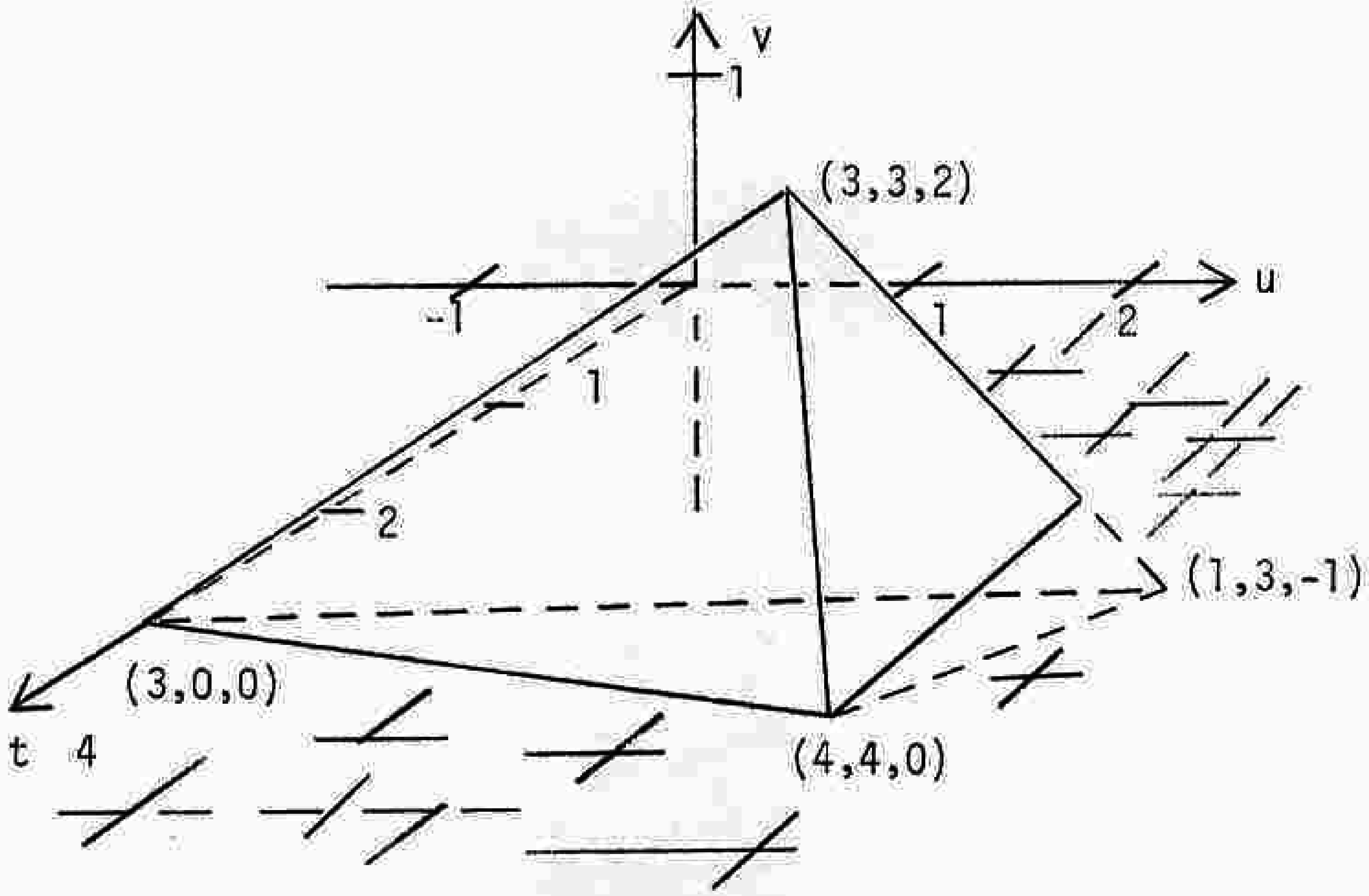


FIGURE 7.2

THE PARAMETER SPACE

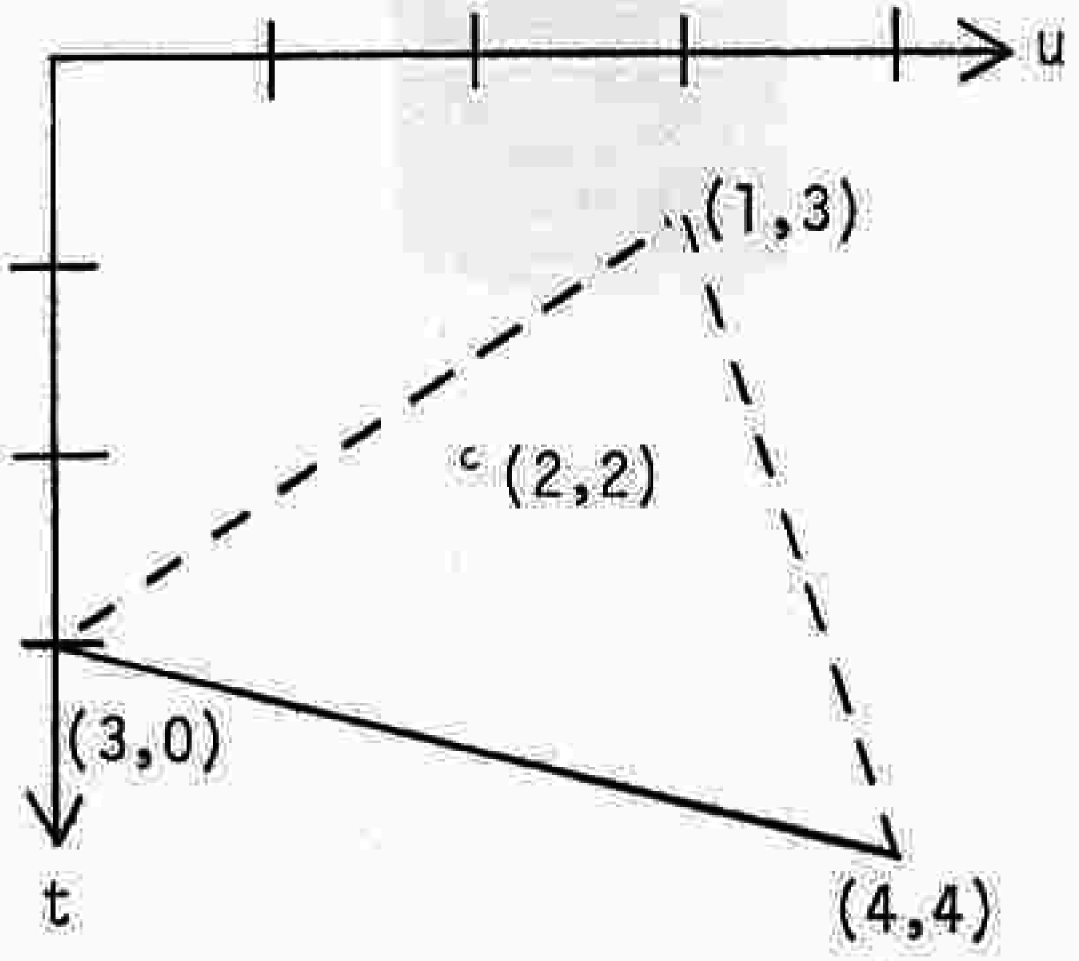


FIGURE 7.3

PROJECTION OF THE SIMPLEX IN FIGURE 7.2 ONTO THE t-u PLANE

$$\mathcal{D} = \sum_{i=1}^{n} P_i .$$

Then apply Algorithm 4.4.1 to each of these facets. The sums of the parity contributions of these are then added to get $d(F,\mathcal{D},\theta_n)$.

There are two approaches for handling curved regions in R^n, as indicated below.

<u>7.3.1 Approximation Approach</u>. Replace $b(\mathcal{D})$ by a sum $\mathcal{J}$ of (n-1)-simplexes each of whose vertices lie on $b(\mathcal{D})$. If $F: \mathcal{D} \to R^n$ is continuous, then, for all such $\mathcal{J}$ whose simplexes have small diameters, the sum of the parities of the range simplexes associated with F and elements of $\mathcal{J}$ is constant. From this, one can define $d(F,\mathcal{D},\theta_n)$ to be this sum of parities. The Kronecker existence theorem still holds when one defines $d(F,\mathcal{D},\theta_n)$ in this manner [1].

<u>7.3.2 A Change of Variable Approach</u>. Parametrize $\mathcal{D}$ so that the parameter space P is a polygon or union of polygons in R^n. Suppose X(U) denotes the mapping from P to $\mathcal{D}$, and let $\tilde{F}(U) = F(X(U))$. Then, provided that X is one-to-one, $F = \theta_n$ will have a solution in $\mathcal{D}$ if and only if $\tilde{F} = \theta_n$ has a solution in P. Algorithm 4.4.1 could then be applied to $\tilde{F}$ on P.

<u>7.3.3 Remark</u>. Approach 7.3.2 indicates that $\mathcal{D}$ often need only be an n-dimensional manifold with boundary (cf. [23]).

7.3.4 Remark. Note that, with Approach 7.3.2, curvilinear regions (i.e., n-manifolds of smoothness C^0) can be "bisected" into unions of curvilinear regions.

7.3.5 Remark. When composing maps of regions, one should keep in mind product theorems concerning the degree of compositions of maps (cf. [23], p. 52, problem 1 and [8], p. 36).

For an example of Approach 7.3.2, consider the following:

7.3.6 Example. Let $\mathcal{D}$ be the region in R^2 defined by:

$$\mathcal{D} = \{(x,y): 0 \leq y \leq 2 - \sqrt{x},\ x \geq 0\},$$

as shown in Fig. 7.4, and let F be the function in Example 4.5.1. Then calculate $d(F, \mathcal{D}, \theta_2)$.

7.3.7 Solution of Example 7.3.6. Parametrize $\mathcal{D}$ by setting $x = 4t$, $y = 2u(1 - \sqrt{2})$, where $0 \leq t \leq 1$ and $0 \leq u \leq 1$. Then:

$$\hat{F}(t,u) = (16t^2 - 4u^2(1 - \sqrt{t})^2 - 1,$$

$$16t^2 + 4u^2(1 - \sqrt{t})^2 - 2),$$

and the new region (in terms of t and u) is the rectangle:

$$P = \langle(0,0),(1,0)\rangle + \langle(1,0),(1,1)\rangle + \langle(1,1),(0,1)\rangle + \langle(0,1),(0,0)\rangle.$$

In this example, the parametrization is not one-to-one on P, and $\tilde{F}(t,u)$ is constant on $\langle(1,0),(1,1)\rangle$. Inclusion of this side does not affect the computations, however.

An orientation is assigned to $\mathcal{B}$ through the orientation of P and the parametrization. The induced orientations of $b(\mathcal{B})$ and $b(P)$ are shown in Fig. 7.4.

Algorithm 4.4.1 was applied to $\tilde{F}$ over P, and correctly gave $d(F,\mathcal{B},\theta_2) = 1$. The computations are listed in Appendix A.3.

7.4 Computing the Linking Number

7.4.1 Definition. [19] Suppose that C is a union of k-dimensional polygons in R^n (or the image of such a union as in Approach 7.3.2). Then C is called a k-cycle (or a cycle of dimension k) if and only if $b(C) = 0$.

Consider two cycles C_1 and C_2 in R^n, the sum of whose dimensions n_1 and n_2 is $n_1 + n_2 = n - 1$. Also assume that the first cycle C_1 is the boundary of some (n_1+1)-polygon (or sum of polygons) in R^n. In this setting one can deal with the concept of whether C_1 and C_2 "pass through" each other, illustrated in Fig. 7.5-Fig. 7.8. If C_1 and C_2 pass through each other as links in a chain, then one says that C_1 and C_2 are "linked" (cf. Figs. 7.5-7.7, and also [1] ch. XI, §1, #6).

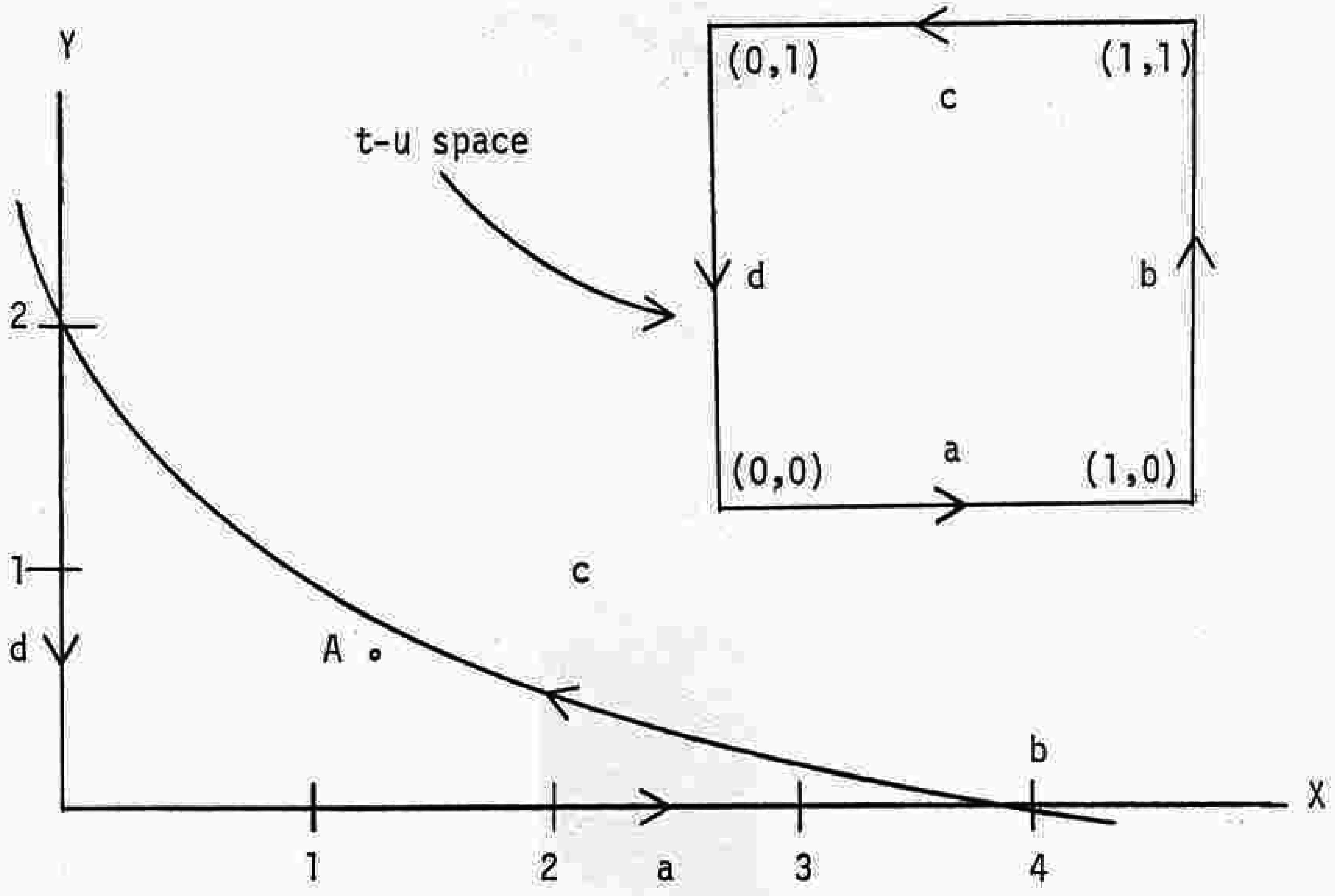


FIGURE 7.4

THE ROOT OF $F(x,y) = (0,0)$ OCCURS AT $A \doteq (1.224, .707)$, AND $J(F)(A) > 0$

The amount of "linkage" is expressed by the linking number (Verschlingungszahl; ibid.), which here will be denoted $v(C_1,C_2)$. The geometric interpretation of various values of $v(C_1,C_2)$ is illustrated in Figs. 7.5-7.8.

The following statements give a formal definition of $v(C_1,C_2)$.

<u>7.4.2 Definition</u>. (cf [1], pp. 493-495) Suppose that K is any region whose boundary is C_1. Then parametrize K and C_2 with parameter variable $U = (u_1,u_2,\ldots,u_n)$ so that, if $X \in K$ then $X = f(u_1,\ldots,u_{n_1+1})$, and if $Y \in C_2$ then $Y = g(u_{n_1+2},\ldots,u_n)$, for some continuous one-to-one mappings f and g. Let the space of parameter variables for the parametrizations f and g be $\mathcal{D}$. Then $v(C_1,C_2)$ is defined by:

$$v(C_1,C_2) = (-1)^{n+1}d(g - f,\mathcal{D},\theta_n) .$$

As is illustrated in Figs. 7.5-7.8, the interpretation of $v(C_1,C_2)$ from Definition 7.4.2 corresponds to the geometric concept for $n_1 = n_2 = 1$ and $n = 3$.

Since $\mathcal{D}$ is a product region, calculation of $v(C_1,C_2)$ can be simplified (ibid.). The domain $\mathcal{D}$ is the Cartesian product of the regions:

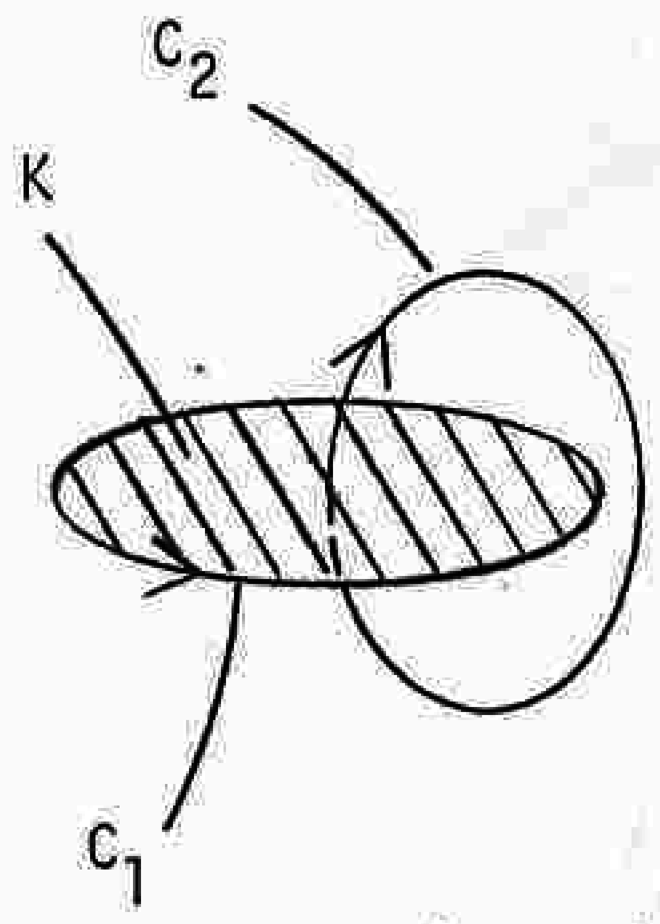


FIGURE 7.5 $v(C_1,C_2) = 1$

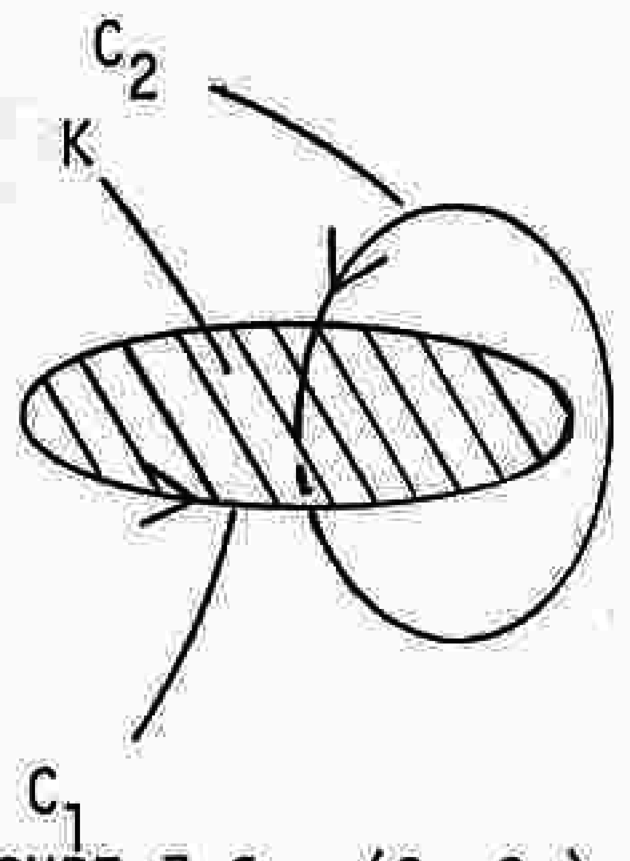


FIGURE 7.6 $v(C_1,C_2) = -1$

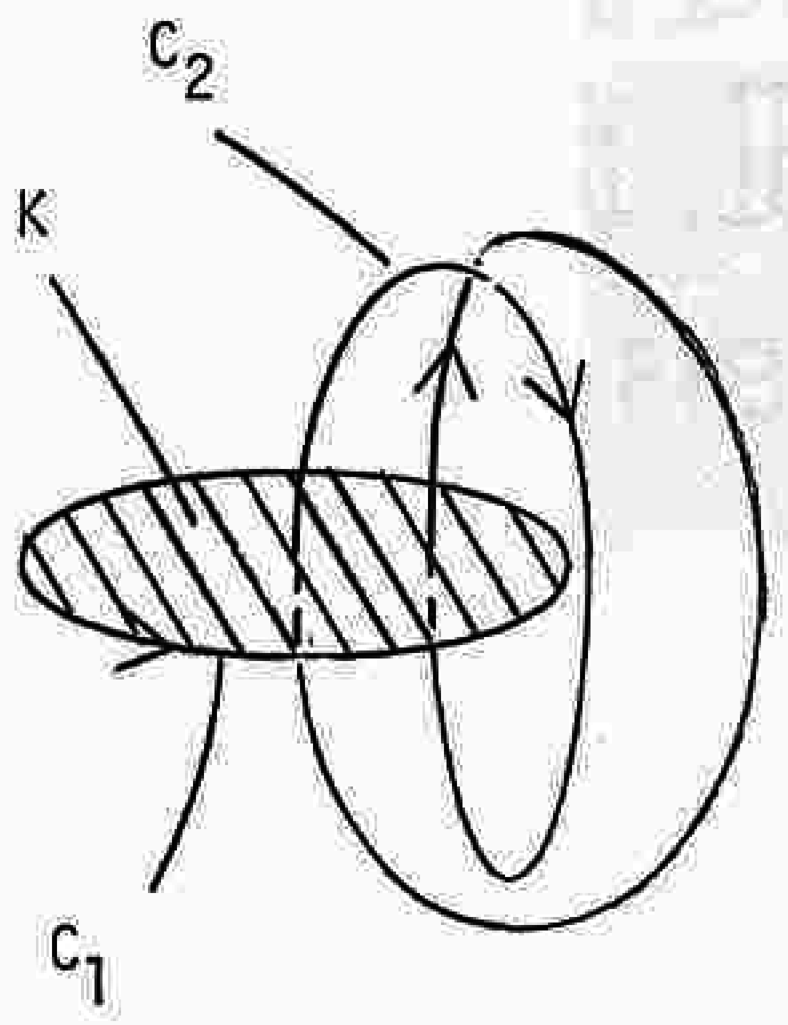


FIGURE 7.7 $v(C_1,C_2) = 2$

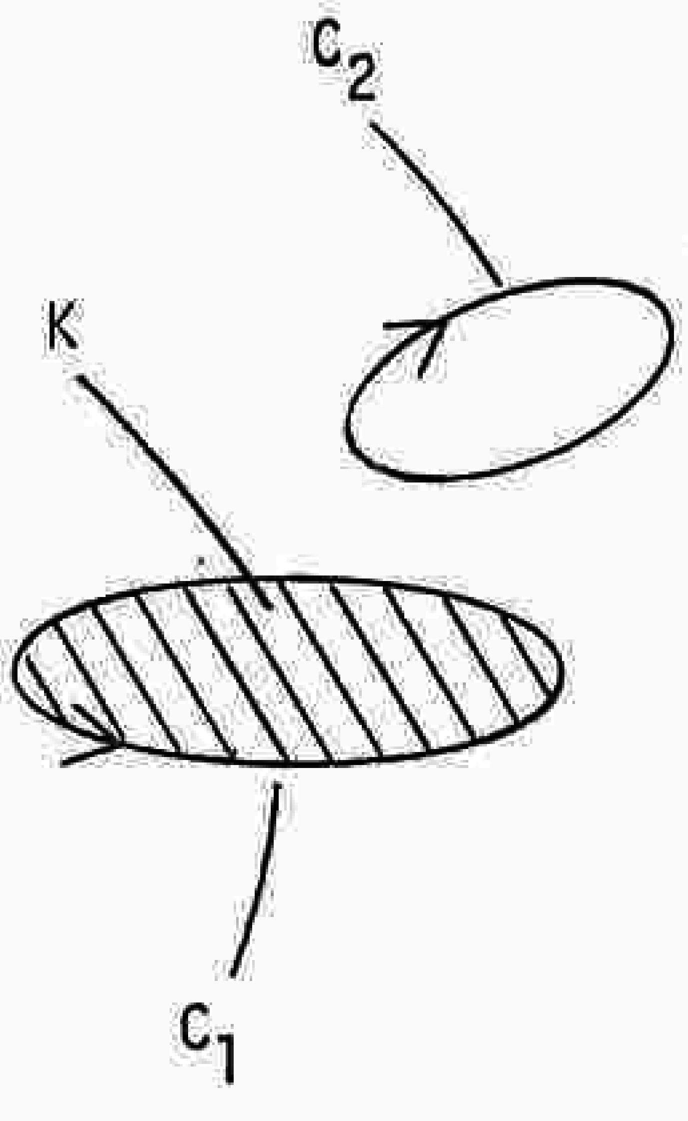


FIGURE 7.8 $v(C_1,C_2) = 0$

$$\mathcal{D}_1 = \{(u_1,\ldots,u_{n_1+1}):(u_1,\ldots,u_n) \in \mathcal{D}\}, \text{ and}$$

$$\mathcal{D}_2 = \{(u_{n_1+2},\ldots,u_n):(u_1,\ldots,u_n) \in \mathcal{D}\},$$

so that: $b(\mathcal{D}) = b(\mathcal{D}_1 \otimes \mathcal{D}_2) = b(\mathcal{D}_1) \otimes \mathcal{D}_2 + \mathcal{D}_1 \otimes b(\mathcal{D}_2)$. However, since C_2 is a cycle, $b(\mathcal{D}_2) = 0$, so that:

$$b(\mathcal{D}) = b(\mathcal{D}_1) \otimes \mathcal{D}_2$$

Also, $b(\mathcal{D}_1)$ is the inverse image under f of the cycle C_1. This fact can be used to obtain a simple representation for $b(\mathcal{D})$ (ibid.).

The above analysis can be carried out directly from a glance at the Gauss integral ([1] p. 497).

7.4.3 Approach. To compute $v(C_1,C_2)$, parametrize C_1 over an n_1-dimensional region P_1, parametrize C_2 over an n_2-dimensional region P_2, and apply Algorithm 4.4.1 to each of the simplexes comprising $P_1 \otimes P_2$.

7.4.4 Example. Determine whether the circles in R^3 given by

$$C_1 = \{(x,y,z);\ x^2 + y^2 = 1;\ z = 0\} \text{ and}$$

$$C_2 = \{(y - 1)^2 + z^2 = 1;\ x = 0\}$$

are linked.

<u>7.4.5 Computations for Example 7.4.4.</u> It is clear from visualization that the above two circles are linked (cf. Fig. 7.9). To test this fact with Algorithm 4.4.1, we will set $P_1 = P_2 = [0,1]$. We will now parametrize C_1 by:

$$\left\{\begin{array}{l} x(t) = \cos 2\pi t \\ y(t) = \sin 2\pi t \\ z(t) = 0 \end{array}\right\} \qquad t \in P_1$$

and parametrize C_2 by:

$$\left\{\begin{array}{l} x(u) = 0 \\ y(u) = \cos 2\pi u + 1 \\ z(u) = \sin 2\pi u \end{array}\right\} \qquad u \in P_2$$

Then $f(t) = (\cos 2\pi t, \sin 2\pi t, 0)$, and $g(u) = (0, \cos 2\pi u + 1, \sin 2\pi u)$, so:

$$g(u) - f(t) = (-\cos 2\pi t,\ \cos 2\pi tu + 1 - \sin 2\pi t,\ \sin 2\pi u)$$

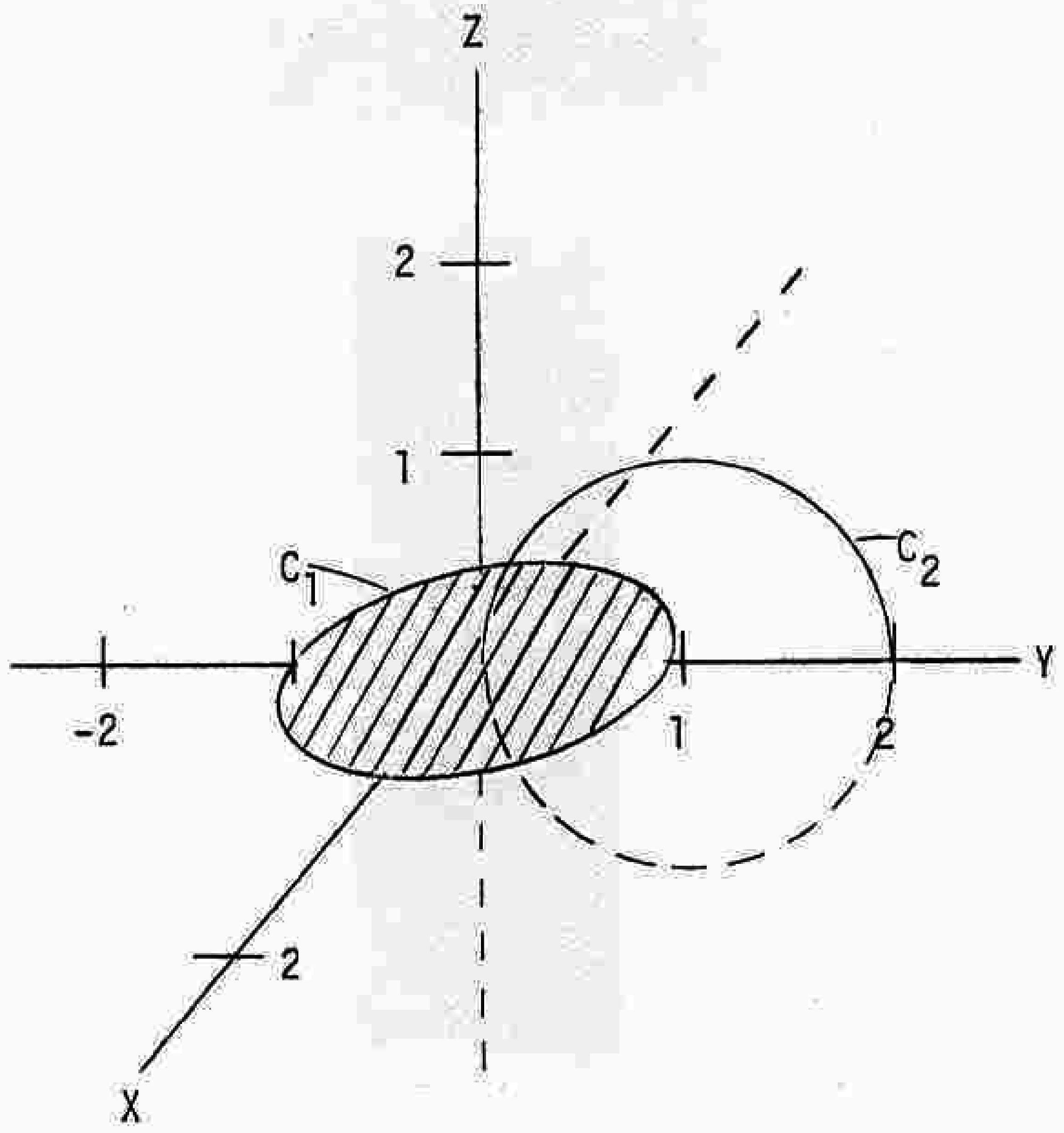


FIGURE 7.9

THE CYCLES C_1 AND C_2

In this example $P_1 \otimes P_2$ can be written as the sum:

$$\langle (0,0),(1,0),(1,1)\rangle + \langle (0,0),(1,1),(0,1)\rangle$$

We will apply Aogorithm 4.4.1 to each of the above simplexes, treating them as boundaries of 3-simplexes. The computational results are listed in Appendix A.4. It was found that $v(C_1,C_2) = +1$.

For further application of Algorithm 4.4.1 and Algorithm 5.3.1, see [3], etc.

APPENDIX

COMPUTER OUTPUT FOR SELECTED EXAMPLES

Computer output for selected examples from the text is listed in this appendix. Binary trees corresponding to some of this output are also drawn. Central processing unit times are listed after the output. Unless otherwise stated, the programs were run on a Univac 1108 with a cycle time of 750 nanoseconds. These programs were not fully optimized, and the best stopping tests were not used.

A.1 A Five-Dimensional Example

The following example was used to test the feasibility of calculating $d(F,\mathcal{D},\theta_n)$ in 5 dimensions. Here, $X = (x_1,x_2,x_3,x_4,x_5)$, and $F(X) = (f_1(X)),f_2(X),f_3(X),f_4(X),f_5(X))$ where:

$$\left\{\begin{aligned} f_1(X) &= x_1^2 - x_2 \\ f_2(X) &= x_2^2 - x_3 \\ f_3(X) &= x_3^2 - x_4 \\ f_4(X) &= x_4^2 - x_5 \\ f_5(X) &= x_5^2 - x_1 \end{aligned}\right\}$$

The domain $\mathcal{D}$ was chosen to be:

$$S = \langle (1,0,0,0,0),(0,1,0,0,0),(0,0,1,0,0),(0,0,0,1,0),(0,0,0,0,1),$$

$$(-1,-1,-1,-1,-1)\rangle = \langle X_0,X_1,X_2,X_3,X_4,X_5\rangle.$$

Now:

$$\begin{vmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 6 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} = -6 < 0,$$

so the points of S are linearly independent, and the orientation of S is negative. Now, $F(0,0,0,0,0) = \theta_5$, and:

$$\theta_5 = 1/6(0,0,0,0,0) + 1/6(0,1,0,0,0) + (1/6(0,0,1,0,0)$$

$$+ 1/6(0,0,0,1,0) + 1/6(0,0,0,0,1) + 1/6(-1,-1,-1,-1,-1),$$

so $(0,0,0,0,0) \in S$. Note also that $(1,1,1,1,1) \notin S$. Also:

$$J(F(X)) = \begin{vmatrix} 2x_1 & -1 & 0 & 0 & 0 \\ 0 & 2x_2 & -1 & 0 & 0 \\ 0 & 0 & 2x_3 & -1 & 0 \\ 0 & 0 & 0 & 2x_4 & -1 \\ -1 & 0 & 0 & 0 & 2x_5 \end{vmatrix} \begin{aligned} &= 32\, x_1x_2x_3x_4x_5 - 1 \\ &= -1 \text{ at } (0,0,0,0,0), \end{aligned}$$

so $d(F,S,\theta_5) = -1$.

The parameter p was taken to be 3 in the calculations. There were 243 points in the tree for the facet:

$$\langle (1,0,0,0,0),(0,0,1,0,0),(0,0,0,1,0),(0,0,0,0,1),(-1,-1,-1,-1,-1)\rangle.$$

The sum of the parity contributions of the facets was calculated to be +1. Since the simplex had a negative orientation, this gives a correct value for $d(F,S,\theta_5)$.

The tables of tree information for the six facets are listed on the following pages. These tables are listed in order, with the one for:

$$\langle X_0,X_1,X_2,X_3,X_4,X_5\rangle = \langle (0,1,0,0,0),(0,0,1,0,0),(0,0,0,1,0),$$

$$(0,0,0,0,1),(-1,-1,-1,-1,-1)\rangle$$

appearing first, then the one for:

$$\langle X_0, X_1, X_2, X_3, X_4, X_5 \rangle \text{ , etc.}$$

It should be noted that two or more iterations of the bisection process would take much less than double the amount of time for the computation of the six trees.

A.2 Computation for the Hidden Line Problem

Graphs of the trees produced for the problem of Example 7.2.1 are given here. If the sum of the parity contributions at a given point in the tree is nonzero, this sum is given next to the point. The total execution time was 2.861 seconds.

It should be recalled that the original simplex was:

$$S = \langle (3,0,0);\ (4,4,0);(1,3,-1);(3,3,2)\rangle,$$

and that the function was given by:

$$F(t,u,v) = (t - u, -v, 2t - u^2 - v^2).$$

A.3 The Degree with Respect to a Curved Region

Computations for Example 7.3.6 are given in this section in the form of tables of tree information.

TABLE A.1.1

TREE INFORMATION FOR

$\langle (0,1,0,0,0), (0,0,1,0,0), (0,0,0,1,0), (0,0,0,0,1), (-1,-1,-1,-1,-1) \rangle$

(Total Parity Contribution: 0; Execution Time: .217 seconds)

Number	n_1	n_2	Parity
1	0	3	0
2	4	3	0
3	0	2	0
4	2	3	0
5	6	3	0
6	2	2	0
7	0	1	0
8	1	3	0
9	5	3	0
10	1	2	0
11	3	3	0
12	7	3	0
13	3	2	0
14	1	1	0

TABLE A.1.2

TREE INFORMATION FOR

$\langle (1,0,0,0,0),\ (0,0,1,0,0),\ (0,0,0,1,0),\ (0,0,0,0,1),\ (-1,-1,-1,-1,-1) \rangle$

(Total Parity Contribution: -1, Execution Time: 1.304 seconds)

Number	n_1	n_2	Parity	Number	n_1	n_2	Parity
1	0	4	0	52	181	8	0
2	8	4	0	53	53	7	0
3	0	3	0	54	117	8	0
4	4	4	0	55	245	9	0
5	12	4	0	56	501	9	0
6	4	3	0	57	245	8	0
7	0	2	0	58	117	7	0
8	2	4	0	59	53	6	0
9	10	4	0	60	21	5	0
10	2	3	0	61	5	4	0
11	6	4	0	62	13	6	0
12	14	4	0	63	45	10	0
13	6	3	0	64	557	10	0
14	2	2	0	65	45	9	0
15	0	1	0	66	301	10	0
16	1	3	-1	67	813	10	0
17	5	7	0	68	301	9	0
18	69	7	0	69	45	8	0
19	5	6	0	70	173	13	0
20	37	13	0	71	4,269	13	0
21	4,133	13	0	72	173	12	0
22	37	12	0	73	2,221	13	0
23	2,085	13	0	74	6,317	13	0
24	6,181	13	0	75	2,221	12	0
25	2,085	12	0	76	173	11	0
26	37	11	0	77	1,197	13	0
27	1,061	11	0	78	5,293	13	0
28	37	10	0	79	1,197	12	0
29	549	10	0	80	3,245	15	0
30	37	9	0	81	19,629	15	0
31	293	10	0	82	3,245	14	0
32	805	11	0	83	11,437	15	0
33	1,829	11	0	84	27,821	15	0
34	805	10	0	85	11,437	14	0
35	293	9	0	86	3,245	13	0
36	37	8	0	87	7,341	13	0
37	165	9	0	88	3,245	12	0
38	421	9	0	89	1,197	11	0
39	165	8	0	90	173	10	0
40	37	7	0	91	685	10	0
41	101	7	0	92	173	9	0
42	37	6	0	93	429	10	0
43	5	5	0	94	941	10	0
44	21	7	0	95	429	9	0
45	85	7	0	96	173	8	0
46	21	6	0	97	45	7	0
47	53	9	0	98	109	14	0
48	309	9	0	99	8,301	14	0
49	53	8	0	100	109	13	0
50	181	9	0	101	4,205	14	0
51	437	9	0	102	12,397	14	0

TABLE A.1.2 con't

Number	n_1	n_2	Parity	Number	n_1	n_2	Parity
103	4,205	13	0	155	1,565	12	0
104	109	12	0	156	3,613	12	0
105	2,157	14	0	157	1,565	11	0
106	10,349	14	0	158	541	10	0
107	2,157	13	0	159	29	9	0
108	6,253	14	0	160	285	9	0
109	14,445	14	0	161	29	8	0
110	6,253	13	0	162	157	9	0
111	2,157	12	0	163	413	9	0
112	109	11	0	164	157	8	0
113	1,133	12	0	165	29	7	0
114	3,181	12	0	166	93	9	0
115	1,133	11	0	167	349	9	0
116	109	10	0	168	93	8	0
117	621	12	0	169	221	14	0
118	2,669	12	0	170	8,413	14	0
119	621	11	0	171	221	13	0
120	1,645	11	0	172	4,317	14	0
121	621	10	0	173	12,509	14	0
122	109	9	0	174	4,317	13	0
123	365	9	0	175	221	12	0
124	109	8	0	176	2,269	14	0
125	237	12	0	177	10,461	14	0
126	2,285	12	0	178	2,269	13	0
127	237	11	0	179	6,365	14	0
128	1,261	12	0	180	14,557	14	0
129	3,309	12	0	181	6,365	13	0
130	1,261	11	0	182	2,269	12	0
131	237	10	0	183	221	11	0
132	749	12	0	184	1,245	12	0
133	2,797	12	0	185	3,923	12	0
134	749	11	0	186	1,245	11	0
135	1,773	12	0	187	221	10	0
136	3,821	12	0	188	733	14	0
137	1,773	11	0	189	8,925	14	0
138	749	10	0	190	733	13	0
139	237	9	0	191	4,829	14	0
140	493	9	0	192	13,021	14	0
141	237	8	0	193	4,829	13	0
142	109	7	0	194	733	12	0
143	45	6	0	195	2,781	14	0
144	13	5	0	196	10,973	14	0
145	29	12	0	197	2,781	13	0
146	2,077	12	0	198	6,877	14	0
147	29	11	0	199	15,069	14	0
148	1,053	12	0	200	6,877	13	0
149	3,101	12	0	201	2,781	12	0
150	1,053	11	0	202	733	11	0
151	29	10	0	203	1,757	12	0
152	541	12	0	204	3,805	12	0
153	2,581	12	0	205	1,757	11	0
154	541	11	0				

TABLE A.1.3

TREE INFORMATION FOR

$\langle(1,0,0,0,0), (0,1,0,0,0), (0,0,0,1,0), (0,0,0,0,1), (-1,-1,-1,-1,-1)\rangle$

(Total Parity Contribution: 0; Execution Time: .219 seconds)

Number	n_1	n_2	Parity
1	0	3	0
2	4	3	0
3	0	2	0
4	2	3	0
5	6	3	0
6	2	2	0
7	0	1	0
8	1	3	0
9	5	3	0
10	1	2	0
11	3	3	0
12	7	3	0
13	3	2	0
14	1	1	0

TABLE A.1.4

TREE INFORMATION FOR

$\langle(1,0,0,0,0), (0,1,0,0,0), (0,0,1,0,0), (0,0,0,0,1), (-1,-1,-1,-1,-1)\rangle$

Total Parity Contribution: 0; Execution Time: .242 seconds)

Number	n_1	n_2	Parity
1	0	3	0
2	4	3	0
3	0	2	0
4	2	3	0
5	6	3	0
6	2	2	0
7	0	1	0
8	1	3	0
9	5	3	0
10	1	2	0
11	3	3	0
12	7	3	0
13	3	2	0
14	1	1	0

TABLE A.1.5

TREE INFORMATION FOR

$\langle(1,0,0,0,0), (0,1,0,0,0), (0,0,1,0,0), (0,0,0,1,0), (-1,-1,-1,-1,-1)\rangle$

(Total Parity Contribution: 0; Execution Time: .216 seconds)

Number	n_1	n_2	Parity
1	0	3	0
2	4	3	0
3	0	2	0
4	2	3	0
5	6	3	0
6	2	2	0
7	0	1	0
8	1	3	0
9	5	3	0
10	1	2	0
11	33	3	0
12	7	3	0
13	3	2	0
14	1	1	0

TABLE A.1.6

TREE INFORMATION FOR

$\langle(1,0,0,0,0), (0,1,0,0,0), (0,0,1,0,0), (0,0,0,1,0), (0,0,0,0,1)\rangle$

(Total Parity Contribution: 0; Execution Time: .261 seconds)

Number	n_1	n_2	Parity
1	0	4	0
2	8	4	0
3	0	3	0
4	44	4	0
5	12	4	0
6	4	3	0
7	0	2	0
8	2	4	0
9	10	4	0
10	2	3	0
11	6	4	0
12	14	4	0
13	6	3	0
14	2	2	0
15	0	1	0
16	1	3	0
17	5	3	0
18	1	2	0
19	3	3	0
20	7	3	0
21	3	2	0
22	1	1	0

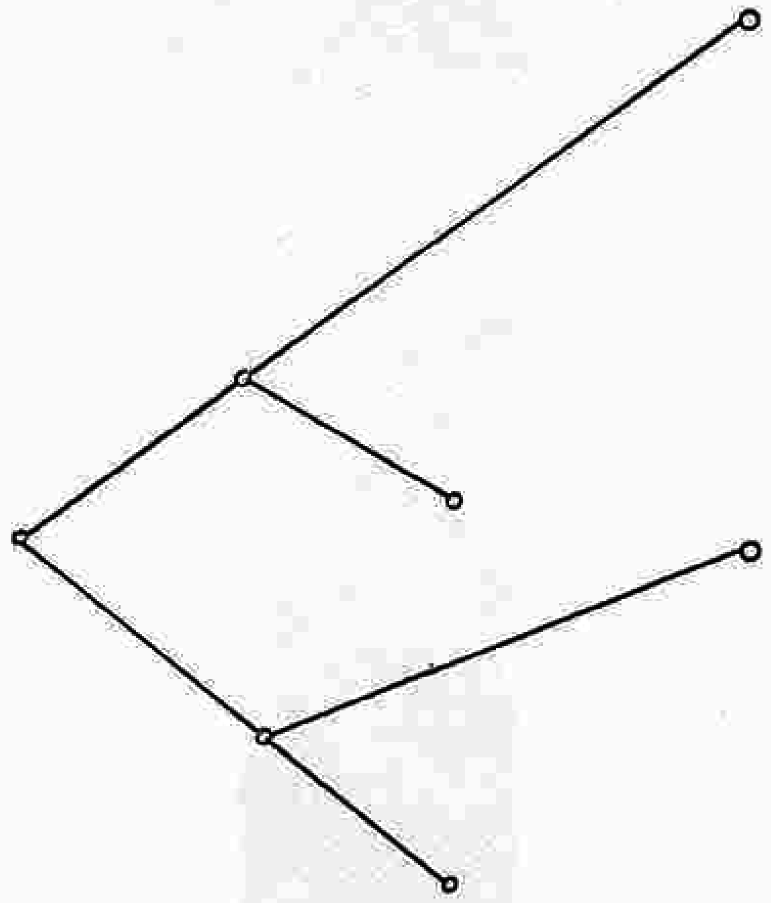


FIGURE A.2.1

TREE PRODUCED FOR $\langle(4,4,0), (1,3,-1), (3,3,2)\rangle$

$p = 2; n = 3$

7 points

(This is the minimal tree for $p = 2$.)

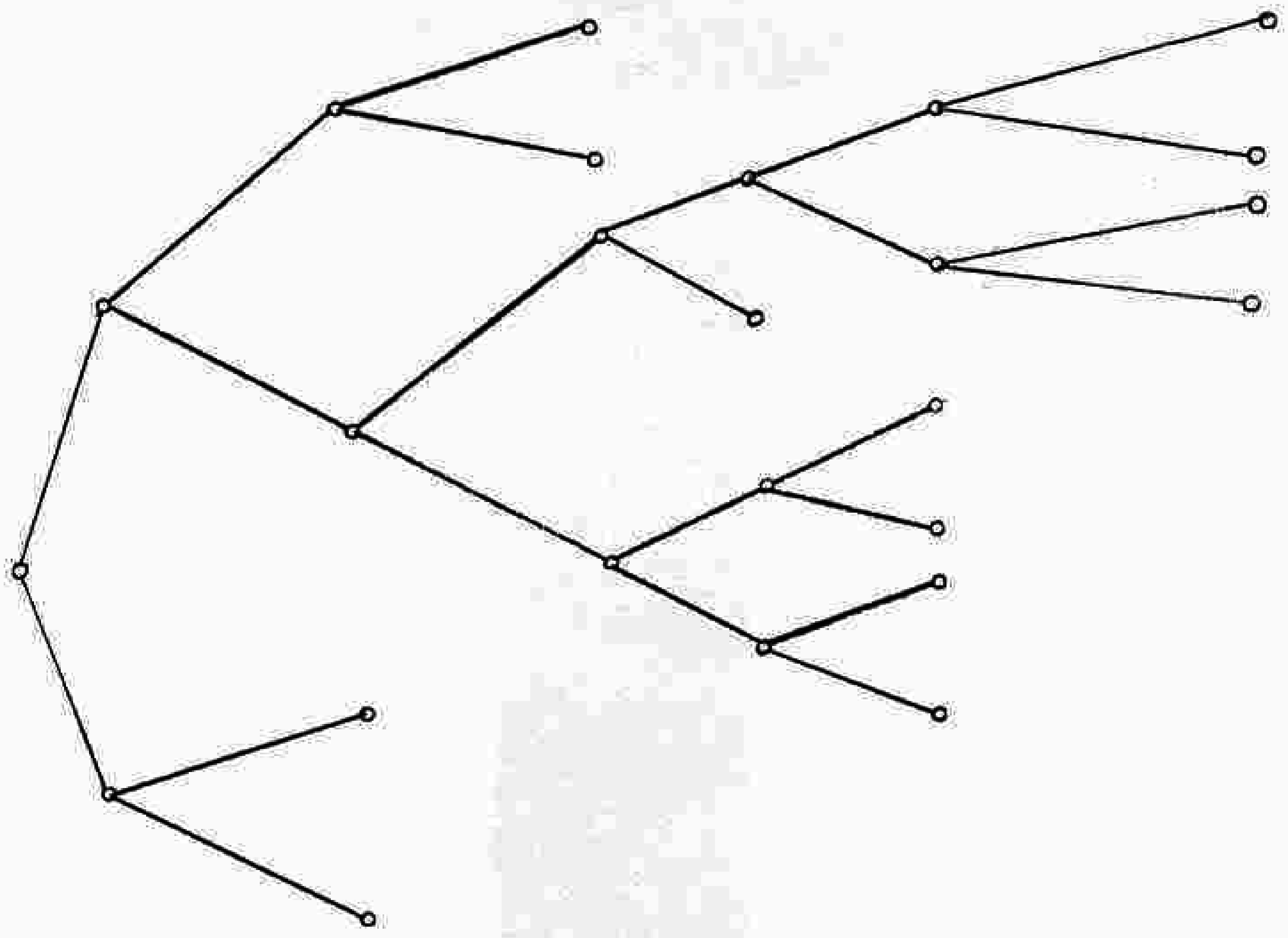


FIGURE A.2.2

TREE PRODUCED FOR $\langle(3,0,0), (1,3,-1), (3,3,2)\rangle$

$p = 2$; $n = 3$

25 points

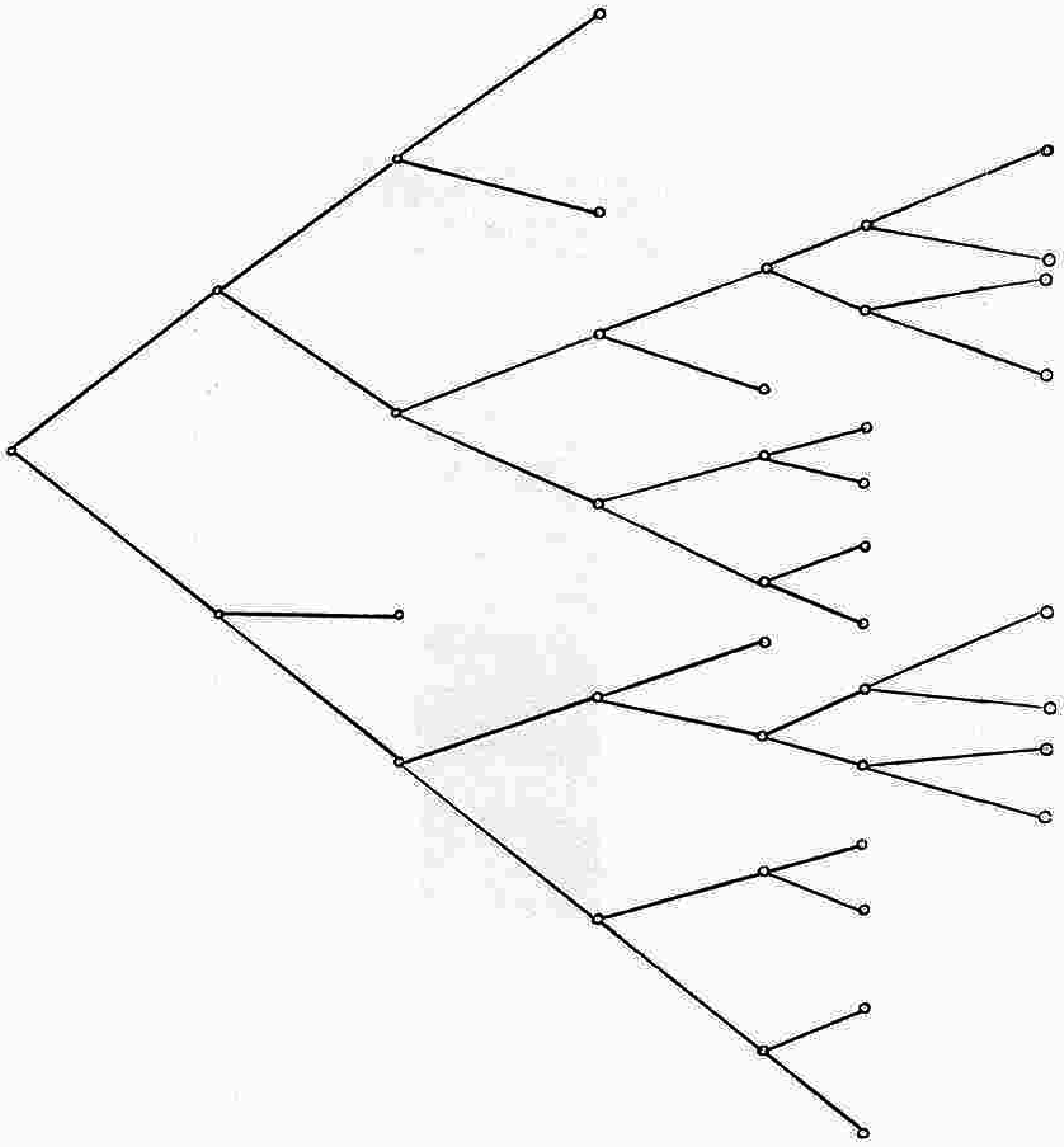


FIGURE A.2.3

TREE PRODUCED FOR $\langle(3,0,0), (4,4,0), (3,3,2)\rangle$

$p = 2$; $n = 3$

41 points

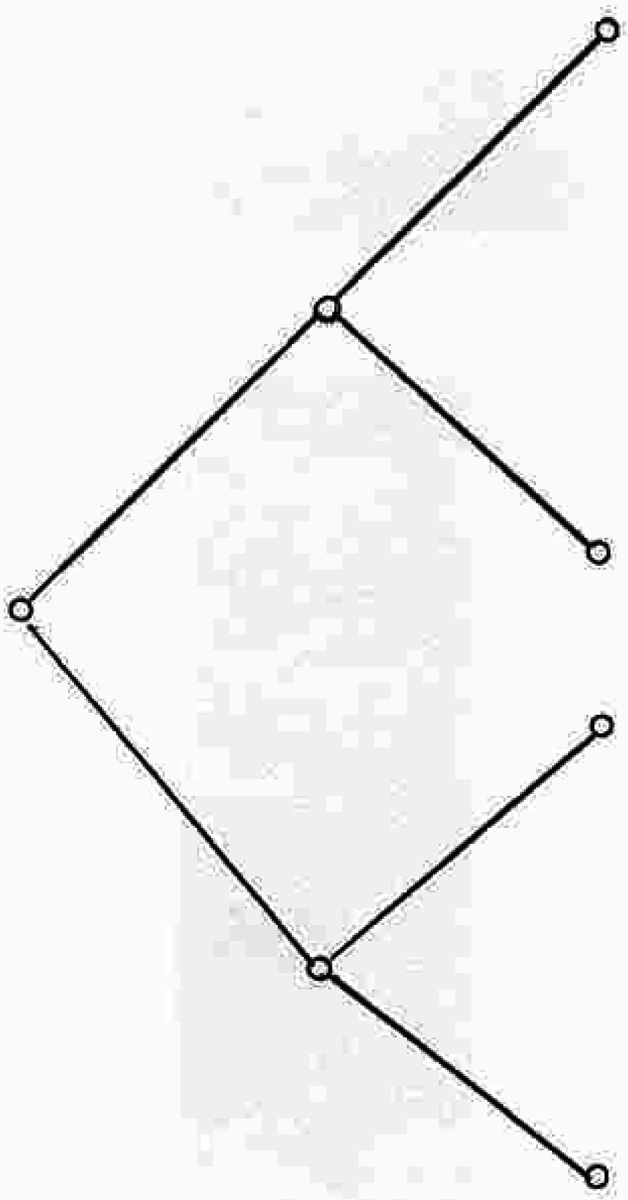


FIGURE A.2.4

TREE PRODUCED FOR $\langle(3,0,0); (4,4,0), (1,3,-1)\rangle$

$p = 2; n = 3$

7 points

(This is the minimal tree for $p = 2$.)

The parameter p in Algorithm 4.4.1 was chosen to be 2 in all cases. Execution times are given separately with each table.

The tables for $\langle(1,0),(1,1)\rangle$ and $\langle(1,1),(0,1)\rangle$ were identical to the above. Execution time corresponding to $\langle(1,0),(1,1)\rangle$ was .077 seconds while execution time for $\langle(1,1),(0,1)\rangle$ was .079 seconds. The table for $\langle(0,1),(0,0)\rangle$ follows.

A.4 Computing the Linking Number

Tables of tree information for Example 7.4.4 are given here. Note that there are only two tables, that for $\langle(0,0),(1,0),(1,1)\rangle$ and that for $\langle(0,0),(1,1),(0,1)\rangle$. Note that since F is independent of the third coordinant this coordinant can be chosen arbitrarily, e.g., it can be chosen to be 0. With this choice, Algorithm 4.4.1 will work on $\langle(0,0,0),(1,0,0),(1,1,0)\rangle$ and $\langle(0,0,0),(1,1,0),(0,1,0)\rangle$ with $n = 3$.

The parameter p of Algorithm 4.4.1 was taken to be 2 for each simplex. Execution times are given separately with each table.

A.5 A More Efficient Search Routine

A considerable fraction of the execution time for the degree calculation program is due to overhead in the routine which searches through the table of tree information and adds together the appropriate parity contributions. An improved method of search would make the algorithm considerably more efficient.

TABLE A.3.1

TREE INFORMATION FOR $\langle(0,0), (1,0)\rangle$

(Total Parity Contribution: 0; Execution Time: .075 seconds)

Number	n_1	n_2	Parity
1	0	2	0
2	2	2	0
3	0	1	0
4	1	2	0
5	3	2	0
6	1	1	0

TABLE A.4.1

TREE INFORMATION FOR $\langle(0,0), (1,0), (1,1)\rangle$

(Total Parity Contribution: 0; Execution Time: .474 seconds)

Number	n_1	n_2	Parity	Number	n_1	n_2	Parity
1	0	4	0	27	5	3	0
2	8	4	0	28	1	2	0
3	0	3	0	29	3	4	0
4	4	4	0	30	11	7	0
5	12	4	0	31	75	7	0
6	4	3	0	32	11	6	0
7	0	2	0	33	43	7	0
8	2	4	0	34	107	7	0
9	10	4	0	35	43	7	0
10	2	4	0	36	11	5	0
11	6	4	0	37	27	5	0
12	14	4	0	38	11	4	0
13	6	3	0	39	3	3	0
14	2	2	0	40	7	5	0
15	0	1	0	41	23	5	0
16	1	4	0	42	7	4	0
17	9	5	0	42	15	5	0
18	25	5	0	44	47	6	0
19	9	4	0	45	15	6	0
20	1	3	0	46	31	6	0
21	5	5	0	47	63	6	0
22	21	5	0	48	31	5	0
23	5	4	0	49	15	4	0
24	13	5	0	50	7	3	0
25	29	5	0	51	7	3	0
26	13	4	0	52	1	1	0

TABLE A.4.2

TREE INFORMATION FOR $\langle(0,0),\ (1,1),\ (0,1)\rangle$

(Total Parity Contribution: 1, Execution Time: .751 seconds)

Number	n_1	n_2	Parity	Number	n_1	n_2	Parity
1	0	4	0	53	89	9	0
2	8	4	0	54	345	9	0
3	0	3	0	55	89	8	0
4	4	4	0	56	217	9	0
5	12	4	0	57	473	9	0
6	4	3	0	58	217	8	0
7	0	2	0	59	89	7	0
8	2	4	0	60	25	6	0
9	10	4	0	61	57	6	1
10	2	3	0	62	25	5	1
11	6	4	0	63	9	4	1
12	14	4	0	64	1	3	1
13	6	3	0	65	5	4	0
14	2	2	0	66	13	6	0
15	0	1	0	67	45	6	0
16	1	7	0	68	13	5	0
17	65	7	0	69	29	6	0
18	1	6	0	70	61	6	0
19	33	7	0	71	29	5	0
20	97	7	0	72	13	4	0
21	33	6	0	73	5	3	0
22	1	5	0	74	1	2	1
23	17	7	0	75	3	6	0
24	81	7	0	76	35	6	0
25	17	6	0	77	3	5	0
26	49	7	0	78	19	6	0
27	113	7	0	79	51	6	0
28	49	6	0	80	19	5	0
29	17	5	0	81	3	4	0
30	1	4	0	82	11	4	0
31	9	8	0	83	3	3	0
32	137	8	0	84	7	7	0
33	9	7	0	85	71	7	0
34	73	8	0	86	7	6	0
35	201	8	0	87	39	7	0
36	73	7	0	88	103	7	0
37	9	6	0	89	39	6	0
38	41	8	0	90	7	5	0
39	169	8	0	91	23	6	0
40	41	7	0	92	55	6	0
41	105	8	0	93	23	5	0
42	233	8	0	94	7	4	0
43	105	7	0	95	15	6	0
44	41	6	0	96	47	6	0
45	9	5	0	97	15	5	0
46	25	9	0	98	31	6	0
47	281	9	0	99	63	6	0
48	25	8	0	100	31	5	0
49	153	9	0	101	15	4	0
50	409	9	0	102	7	3	0
51	153	8	0	103	3	2	0
52	25	7	0	104	1	1	1

It is seen that, when the table of tree information is rather long, the parity contributions to be added together appear near the end of the table. Indeed, very often, the points to be added together are the last two points in the table. Here, it is better to check the last points first, and search backwards. The search algorithm should then also stop if both parity contributions are formed.

The flowchart below takes this consideration into account. It also contains several other modifications which should also increase its efficiency.

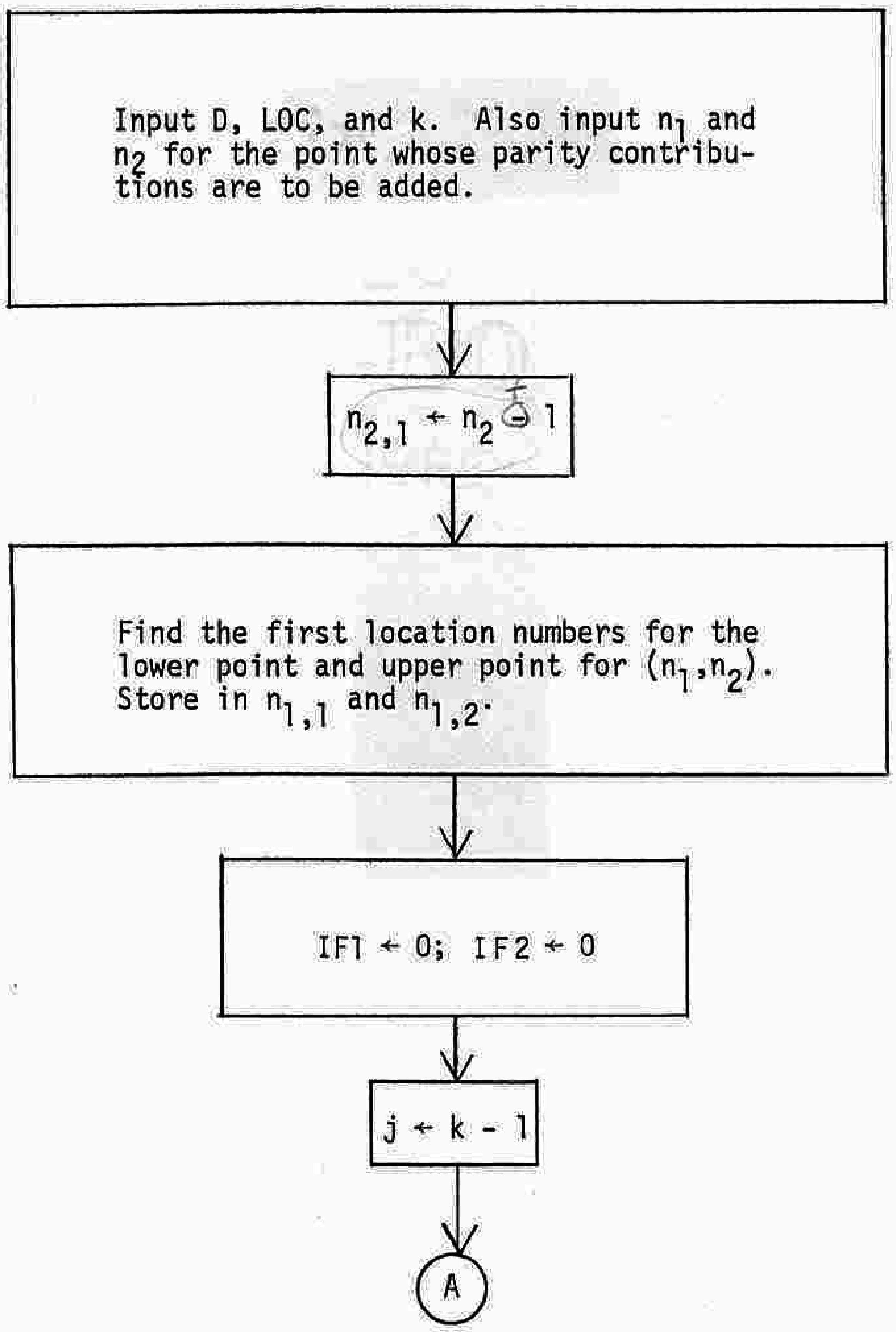

Input D, LOC, and k. Also input n_1 and n_2 for the point whose parity contributions are to be added.
$n_{2,1} \leftarrow n_2 \overset{+}{-} 1$
Find the first location numbers for the lower point and upper point for (n_1, n_2). Store in $n_{1,1}$ and $n_{1,2}$.
IF1 $\leftarrow$ 0; IF2 $\leftarrow$ 0
$j \leftarrow k - 1$
A

FIGURE A.5.1

AN EFFICIENT SEARCH ROUTINE

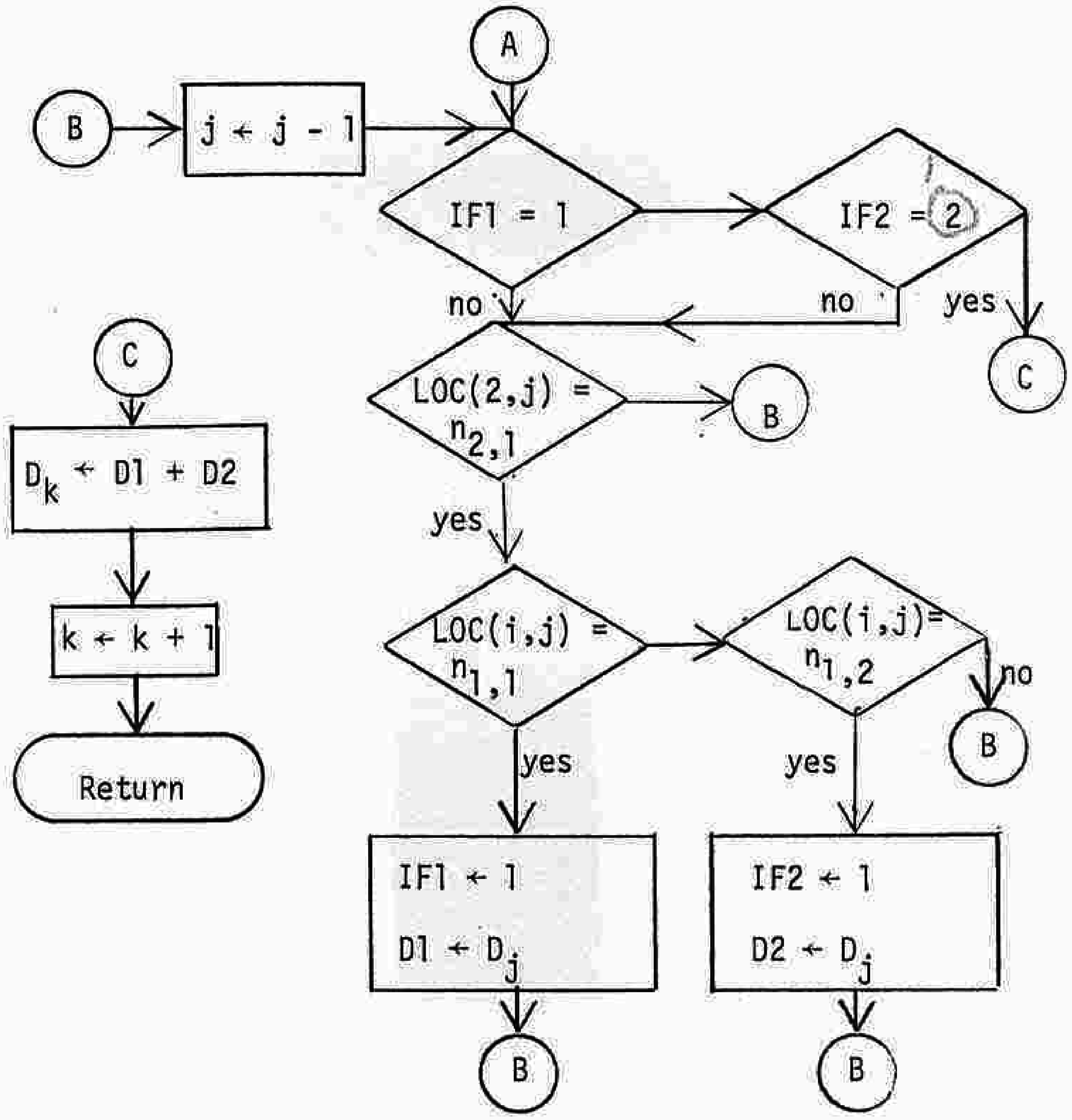


FIGURE A.5.2

AN EFFICIENT SEARCH ROUTINE (Continued)

Please note that LOC(1,j) is the first location number and LOC(2,j) is the second location number for the jth point in storage.

REFERENCES

[1] P. Alexandroff and H. Hopf, Topologie, Chelsea, New York (1973: originally 1935).

[2] E. L. Allgower, Numerische Approximation von Losungen Nicht-linearer Ranswerts-Aufgaben mit meheren Losungen, ZAMM, Tagung, 54, Munchen (1974).

[3] E. L. Allgower and M. M. Jeppson, The Approximation of Solutions of Nonlinear Elliptic Boundary Value Problems Having Several Solutions, Springer Lecture Notes 333, pp. 1-20 (1973).

[4] E. L. Allgower and K. L. Keller, "A Search Routine for a Sperner Simplex," Computing 8, pp. 157-165 (1971).

[5] H. Amann and S. Weiss, On the Uniqueness of the Topological Degree, Math Z. 130, pp. 39-54 (1973).

[6] E. F. Beckenbach, Applied Combinational Mathematics, John Wiley and Sons, New York (1964).

[7] Box, Davies, and Swan, Nonlinear Optimization Techniques, Imperial Chemical Industries Monograph No. 5, Oliver and Bord, Edinburgh (1969).

[8] J. Cronin, Fixed Points and Topological Degree in Nonlinear Analysis, the American Mathematical Society, Providence, R.I. (1964).

[9] B. C. Eaves, "An Odd Theorem," Proceedings of the A.M.S. 26, 3 (1970), pp. 509-513.

[10] B. C. Eaves and R. Saigal, "Homotopies for the Computation of Fixed Points on Unbounded Regions," Mathematical Programming 13, nos. 1 and 2 (1972).

[11] A. Eiger and F. Stenger, "A Program of Bisections for Solving Two Nonlinear Equations," to appear in Comm. A.C.M.

[12] Geoffrion, Elements of Large-Scale Mathematical Programming, the Rand Corporation for USAF Project Rand, Santa Monica (1969).

[13] Marvin Greenberg, Lectures on Algebraic Topology, W. A. Benjamin, New York (1967).

[14] J. Hadamard, Sur Quelques Applications de l'Indice de Kronecker, Hermann, Paris (1910).

[15] Communicated by G. Hamrick, The University of Texas at Austin.

[16] E. Heinz, "An Elementary Analytic Theory of the Degree of a Mapping in n-Dimensional Space," Journal of Mathematics and Mechanics 8, no. 2 (1959).

[17] I. N. Herstein, Topics in Algebra, Ginn and Co., Waltham, Mass. (1964).

[18] M. Jeppson, "A Search for the Fixed Points of a Continuous Mapping," Mathematical Topics in Economic Theory and Computation, 122-128, SIAM, Philadelphia (1972).

[19] J. Keesee, Introduction to Algebraic Topology, Wadsworth Publishing Company, Belmont, Ca. (1970).

[20] M. A. Krasnosel'skii, Topological Methods in the Theory of Nonlinear Integral Equations, the Macmillan Company, New York (1964).

[21] H. W. Kuhn, Some Combinatorial Lemmas in Topology, IBM Journal, Nov., 1960, pp. 518-524.

[22] J. McNamee and F. Stenger, Numerical Analysis, to be published.

[23] Milnor, Topology from the Differentiable Viewpoint, The Univ. of Virginia, Charlottesville (1965).

[24] M. Nagumo, "A Theory of Degree of Mapping Based on Infinitesimal Analysis," Amer. J. Math 73, pp. 485-492 (1951).

[25] T. O'Neil and J. W. Thomas, "The Calculation of the Topological Degree by Quadrature," SIAM J. Numer. Anal. 12, pp. 637-680 (1975).

[26] J. M. Rheinbolt and W. C. Ortega, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York (1970).

[27] H. Scarf, "The Approximation of Fixed Points of a Continuous Mapping," SIAM J. Appl. Math 15, No. 5, September, 1967.

[28] H. Scarf, The Computation of Economic Equilibria, London Univ. Press, New Haven (1973).

[29] M. Steins, Univ. of Oregon at Corvallis.

[30] F. Stenger, "An Algorithm for the Topological Degree of a Mapping in R^n," Numerische Mathematik 25, pp. 23-28 (1976).

[31] F. Stenger and C. Harvey, "A Two-Dimensional Analogue to the Method of Bisections for Solving Nonlinear Equations," Quarterly Journal of Applied Mathematics, January, 1976.

VITA

Name	Ralph Baker Kearfott
Birthdate	January 27, 1954
Birthplace	Salt Lake City, Utah
High School	Colegio Loyola-Gumilla Pto. Ordaz, Venezuela
Universities 1970-1976	University of Utah Salt Lake City, Utah
1976-1977	The University of Texas at Austin Austin, Texas
Degrees 1972	B.A., University of Utah Salt Lake City, Utah
1974	M.A., University of Utah Salt Lake City, Utah
Honorary Societies	Phi Beta Kappa, Phi Kappa Phi, Pi Mu Epsilon
Professional Organizations	The American Mathematical Society, Society of Industrial and Applied Mathematics
Professional Positions	Teaching Fellow, University of Utah, 1970-1976; Teaching Assistant, The University of Texas at Austin, 1976-1977