Probability and Statistics Notes

Note 8

Classical Upper Confidence Limits for the Failure Probability of Systems

Ronald D. Halbgewachs, 2641
R. Curtis Mueller, 9525
Frank W. Müller, 9525
Sandia Laboratories, Albuquerque, NM 87115

September 1973

Abstract

The compute code C0NLIM evaluates classical upper confidence limits for failure probability of systems based on component test results. C0NLIM can accommodate systems ranging from the very simple to complex combinations utilizing several different components. Required input basically consists of component test data (number tested and number of failures) and the system reliability equation. This report (1) details the analysis for maximization of the nonlinear reliability function of many variables subject to a nonlinear constraint function, (2) develops the algorithms used in C0NLIM, (3) provides a users' manual for the program, and (4) presents program output and computation times for several hypothetical systems demonstrating the flexibility of the code.

Key Words: Probability, confidence limits, nonlinear function maximization, hypervolumes, mathematical analysis, computer program

TABLE OF CONTENTS

CLASSICAL UPPER CONFIDENCE LIMITS FOR THE FAILURE PROBABILITY OF SYSTEMS

I. Introduction

Most functional systems represent a combination of many different components or elements. The systems might be electrical, mechanical, or in general, any assemblage of small units which makes up a larger unit. We are concerned with determining a classical upper confidence limit for the probability of failure of such systems from test information about the components. To accomplish this task, we divide the problem into three major parts: (1) the construction of a set of "hypervolumes" to be used in repeated computations to evaluate a constraint equation, (2) the maximization of a nonlinear function of many variables subject to a constrant, and (3) the development of an algorithm to place the theory into application.

In this report we discuss in detail each of these parts of the problem, as well as provide an extensive set of appendices covering the use of a computer program CONLIM. Included in the appendices are summarized tabulations of results obtained from sample test cases, a listing of the program CONLIM, sample input data formats and computer output reports, and an explanation of the variety of options available to the user of CONLIM.

II. Basic Equations and Notation

Throughout this report several fundamental notations will be used which pertain to a system under study and each of the system components. These notations are

n = number of unique system components;

m_i = number of performance tests on component i, i=1,...,n;

x_i = number of failures occurring in the tests of component i, i=1,...,n;

α = confidence level at which the upper limit of system failure probability is to be computed.

(II-1)

The value of n is also defined to be the dimension of the system under consideration.

As described by Steck [1], the classical upper confidence limit on system failure probability is the maximum of the system failure probability function $f(p_1,p_2,\ldots,p_n)$ where p_i is the failure probability of the ith component in the system, subject to a constraint. Thus, we wish to find the values of the independent variables $(p_1^*,p_2^*,\ldots,p_n^*) = P^*$ such that $f(P^*)$ is the maximum function value attainable subject to the constraint that

$$H(p_1^*,p_2^*,\ldots,p_n^*) = 1 - \alpha \qquad \text{(II-2)}$$

where

$$H(p_1,p_2,\ldots,p_n) = \sum_{a\in\Psi} \prod_{i=1}^{n} \binom{m_i}{a_i} p_i^{a_i} (1-p_i)^{m_i-a_i} \qquad \text{(II-3)}$$

and Ψ is an appropriately defined index set of vectors $A = (a_1, a_2, \ldots, a_n)$. Adopting a proposal of Müller for specifying a reasonable set Ψ while avoiding the iterative computations required by Steck, we define the index set Ψ as all vectors A so that $f(\bar{A}) \leq f(\bar{P})$ where

$$\bar{A} = (\bar{a}_1, \ldots, \bar{a}_n) \; ; \; \bar{a}_i = \frac{a_i+1}{m_i+2} \; ; \; 0 \leq a_i \leq m_i \tag{II-4}$$

and $$\bar{P} = (\bar{p}_1, \ldots, \bar{p}_n) \; ; \; \bar{p}_i = \frac{x_i+1}{m_i+2} \; . \tag{II-5}$$

That is,

$$\Psi = \{A \mid f(\bar{A}) \leq f(\bar{P})\} \; . \tag{II-6}$$

For example, in two dimensions the index set Ψ would contain all pairs of integers (a_1, a_2) so that the function values obtained by evaluating f at each of these pairs would fall under the curve $f(\bar{p}_1, \bar{p}_2)$. In this sense, we sometimes speak of the pairs as points lying under the curve $f(\bar{p}_1, \bar{p}_2)$. In three dimensions we are dealing with surface functions and for problems of dimension $n > 3$ we write in terms of hypersurfaces.

III. Hypervolumes and the Index Set

Theoretically, there is no reason not to evaluate H directly from (II-3). Practically, the use of (II-3) in the many times that H must be evaluated results in an unacceptably long computer run time for the maximization search routine. It has proved possible to develop a more efficient method of evaluating H.

The discussion to follow has three segments. We first describe, in an intuitive manner, the various elemental ideas used in the construction of the set Ψ utilized for indexing in (II-3) and the development of hypervolumes. For those readers interested in hypervolumes only from the standpoint of how they relate to the problem of probability function maximization and not an in-depth analysis, this intuitive approach should suffice and not particularly detract from the discussions in later sections. Following this description are a rigorous development of hypervolumes and their properties, as well as a detailed algorithm for the index set construction based on that development.

From the simplified example given at the end of the previous section, we note that in two dimensions the index set Ψ would be composed of pairs of integers satisfying the set criteria of (II-6). Given a specific system failure probability function f, Figure 1 displays the single failure probability curve $f(\bar{p}_1,\bar{p}_2)$ that might be associated with the component failure probabilities of $\bar{p}_1$ and $\bar{p}_2$ described by (II-5). However, the values $\bar{p}_1$ and $\bar{p}_2$ are dependent upon the integer values x_1 and x_2 in the numerator of (II-5). Except in the "trivial" case where $x_1 = x_2 = 0$, other integer values substituted for x_1 and x_2, simultaneously, could produce a function value coinciding with this curve or located somewhere below the curve. For the trivial case, only one pair of integers will accomplish this fact; namely, the pair (0,0). To distinguish the specific test situation identified by $\bar{p}_1$ and $\bar{p}_2$, we make use of a similar notation (II-4) where these additional integer values may be considered.

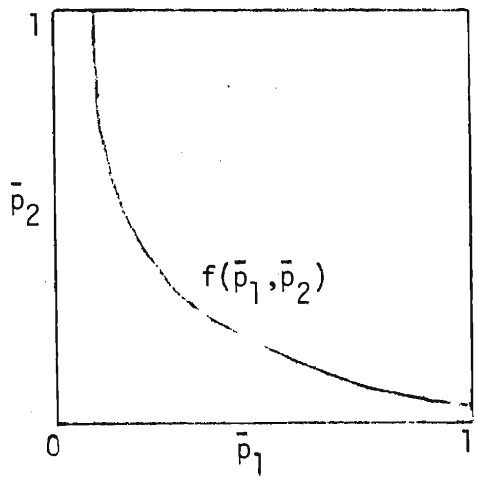


Figure 1.

In a sense, the p component values are scaled values of integers. From this standpoint we can envision the curve overlying a graph of integers or a lattice of integers. Figure 2 illustrates such a lattice possible for some particular two-dimensional problem.

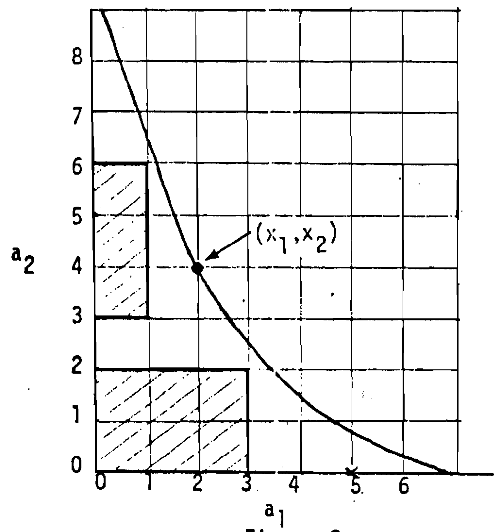


Figure 2

For the illustration, the pairs of integers (a_1,a_2) yielding function values, $f(\bar{a}_1,\bar{a}_2)$, below the curve would consist of

(0,0) (1,0) (2,0) (3,0) (4,0) (5,0) (6,0)

(0,1) (1,1) (2,1) (3,1) (4,1)

(0,2) (1,2) (2,2) (3,2) (4,2)

⋮ ⋮

(0,6) (1,6)

(0,7)

(0,8)

In order to express these pairs in a more compact notation, save storage space within a computer program, reduce the necessary time to determine all points satisfying (II-6), and develop the constraint equation (II-3) into a more efficient and usable form, we utilize the terminology of hypervolumes. A hypervolume, designated as HV, can be described simply as a rectangle, parallelpiped, etc., respectively, as the number of dimensions increase. Thus, in our example, we can describe all points ranging from 0 to 3 in a_1 and 0 to 2 in a_2 as belonging to the first hypervolume, $HV_1 = \{[0,3], [0,2]\}$. This rectangle is shown in Figure 2. A second hypervolume, $HV_2 = \{[0,1], [3,6]\}$ is also shown in Figure 2. Notice that we did not touch rectangles, since they would have overlapped on the two points (0,2) and (1,2). If one keeps in mind the idea that we want to express lattice points in a new terminology and that dimension only places an upper bound on the geometric form a hypervolume can assume, it will not be difficult to see that a third hypervolume might consist of a line, e.g., $HV_3 = \{[4,4], [0,1]\} = (4,0), (4,1)$, or a single point, $HV_4 = \{[5,5], [0,0]\} = (5,0)$.

In this manner, we can include all points of the underlying lattice in the hypervolume form of definition. Upon insuring that there are no overlapping points (i.e., maintaining unique hypervolumes), we then have the necessary index set Ψ of (II-3) whereby the total union of the hypervolumes compose Ψ. The discussion up to this point has been primarily concerned with two-dimensional situations, but the general idea of describing clusters of lattice points in a compact form can be carried through for higher dimensions.

To effectively define and describe a generalized approach of hypervolume construction and yet implement an algorithm in a practical and logical sequence, we must undertake a more rigorous analysis. We assume the notation of (II-1) throughout the remaining discussion.

Definition 1: Given a set of numbers $\{m_1, m_2, \ldots, m_n\}$ denoting the number of tests performed on each component, a choice space X of dimension n is defined to be the set of all points $A = (a_1, a_2, \ldots, a_n)$ so that $0 \le a_1 \le m_1$, $0 \le a_2 \le m_2$, ..., $0 \le a_n \le m_n$. With the dimensionality of a point $A \in X$ implied, the term point is taken to be synonymous with the term n-tuple.

From this definition and equation (II-4) we see that $\Psi \subset X$. As the dimension increases (i.e., larger systems) and each of the m_i also increases, the procedure to be used in determining the contents of the index set Ψ can become extremely time consuming. Furthermore, Ψ must be constructed so that repeated evaluations of equations similar to (II-3) can be made as efficiently as possible.

During these studies, several schemes were devised for the construction of Ψ to determine the elements $A \in \Psi$. Although one approach devised by Mueller was deemed more efficient than others, having several algorithms available proved beneficial in checking the contents of Ψ. We shall designate the method of construction utilized for Ψ as the "hypervolume algorithm." To lay the foundation for subsequent discussion and development, we begin the description of the method of construction in a formal manner.

Definition 2: If $(a_1, \ldots, a_k, \ldots, a_n)$ is an n-tuple, then the closed set of integers along the kth component coordinate for which a_k may range is designated as R_k. Thus, R_k is the set of integers ranging from a minimum value or lower limit, L_k, to a maximum value or upper limit, U_k, and $R_k = \left[L_k, U_k\right]$.

Definition 3: A hypervolume HV in the choice space $\mathbf{X}$ of dimension n is defined to be the set of points

$$HV = \left\{ (a_1, \ldots, a_k, \ldots, a_n) \,\middle|\, a_1 \varepsilon R_1, \ldots, a_k \varepsilon R_k, \ldots, a_n \varepsilon R_n \right\} .$$

A reduced form of notation to be used is

$$HV = \left\{ R_1, \ldots, R_k, \ldots, R_n \right\} .$$

From Definition 3, we see that each component a_k in an n-tuple has a definite range of values for a particular hypervolume. In a sense, this range represents an edge of the hypervolume being described. When taken together with the other components, the hypervolume then becomes filled with points. The next three definitions provide us with the means to distinguish between unique hypervolumes.

Definition 4: The hypervolume HV_i indicates the ith hypervolume in the space and is defined as

$$HV_i = \left\{ R_1(i), \ldots, R_n(i) \right\}$$

$$= \left\{ [L_1(i), U_1(i)] , \ldots, [L_n(i), U_n(i)] \right\} .$$

Let us consider an example where there are two unique components to a system under study. In examining the components, suppose that 50 observations were made of the first component and for the second component 47 observations were made. Hence, $n = 2$, $m_1 = 50$, $m_2 = 47$ and $\mathcal{X} = \{(a_1, a_2) \mid 0 \le a_1 \le 50,\ 0 \le a_2 \le 47\}$.

With these conditions, we might have hypervolumes such as

$$HV_1 = \{[0, 2], [0, 3]\} \text{ or } HV_2 = \{[1, 3], [2, 6]\} .$$

There would be 12 points in HV_1:

(0, 0) ,	(0, 1) ,	(0, 2) ,	(0, 3)
(1, 0) ,	(1, 1) ,	(1, 2) ,	(1, 3)
(2, 0) ,	(2, 1) ,	(2, 2) ,	(2, 3) .

Similarly, there would be 15 points in HV_2.

Definition 5: Two points A_1 and A_2 in the space $\mathcal{X}$ coincide, $A_1 \equiv A_2$, iff all components of the points are equal.

Definition 6: Hypervolumes HV_i and HV_j do not overlap in the k^{th} dimension when $U_k(i) < L_k(j)$ or $L_k(i) > U_k(j)$. We define the symbol $R_k(i)\ //\ R_k(j)$ to denote this relationship..

Theorem 1: If $R_k(i)$ // $R_k(j)$ for any k, then the two hypervolumes HV_i and HV_j have no common points,

$$HV_i \cap HV_j = \emptyset .$$

Proof: Consider some point $A \varepsilon HV_i$, $A = (a_1, \ldots, a_n)$. Then, by definition of hypervolumes, $a_k \notin [L_k(i), U_k(i)]$ for each k and we have

(1) $R_k(i) \, // \, R_k(j) \Rightarrow U_k(i) < L_k(j) \Rightarrow a_k < L_k(j) \Rightarrow a_k \notin [L_k(j), U_k(j)]$,

or

(2) $R_k(i) \, // \, R_k(j) \Rightarrow L_k(i) > U_k(j) \Rightarrow a_k > U_k(j) \Rightarrow a_k \notin [L_k(j), U_k(j)]$.

But $a_k \notin [L_k(j), U_k(j)] \Rightarrow A \notin HV_j$ and therefore

$HV_i \cap HV_j = \emptyset$. ▮

The theorem shows us that if one questions whether two hypervolumes have any points in common we need only find a single coordinate in which $R_k(i)$ // $R_k(j)$, and there will be no common points.

Corollary: Two hypervolumes overlap if there exists at least one coincident point of the hypervolumes.

Proof: Consider two points in the space X,

$A_1 = (a_1, \ldots, a_k, \ldots, a_n)$ and $A_2 = (a_1', \ldots, a_k', \ldots, a_n')$.

Then

$A_1 \varepsilon HV_i \cap HV_j$

$\Leftrightarrow A_1 \varepsilon HV_i$ and $A_1 \varepsilon HV_j$

$\Leftrightarrow a_k \varepsilon R_k(i)$ and $a_k \varepsilon R_k(j)$, $\forall k$

$\Leftrightarrow a_k$ = a component $a'_k \in R_k(j)$, $\forall k$

$\Leftrightarrow (a_1, \ldots, a_k, \ldots a_n) \equiv (a'_1, \ldots, a'_k, \ldots, a'_n)$

$\Leftrightarrow A_1 \equiv A_2$. ▌

<u>Definition 7</u>: A hypervolume HV_i is said to be a <u>subset</u> of another hypervolume HV_j iff

$$U_k(i) \leq U_k(j) \text{ and } L_k(i) \geq L_k(j), \quad \forall k.$$

We denote this relation by $HV_i \subset HV_j$. We also refer to containment within a single coordinate k as $R_k(i) \subset R_k(j)$ so that $\forall k$, $R_k(i) \subset R_k(j) \Leftrightarrow HV_i \subset HV_j$.

Note that when $HV_i \subset HV_j$, one hypervolume is completely contained within another. That is, $HV_i \cap HV_j = HV_i$. It may also be the case that the two hypervolumes are identical. By Theorem 1, $R_k(i) \mathbin{//} R_k(j)$ for any k $\Rightarrow$ $HV_i \not\subset HV_j$. Let us examine situations whereby there may be overlap between two hypervolumes, but not complete containment.

<u>Definition 8</u>: A hypervolume HV_i is said to be free from overlap at the lower end in the k-dimension relative to some other hypervolume HV_j iff

$$L_k(j) \leq U_k(i) \leq U_k(j) \text{ and } L_k(i) < L_k(j).$$

We define the symbol $R_k(i) \underline{O} R_k(j)$ to denote this relation.

<u>Theorem 2</u>: Let HV_i and HV_j be any two hypervolumes in X. If $R_m(i) \underline{O} R_m(j)$ for one or more m, then redefining HV_i so that $U_m(i) = L_m(j)-1$ for any single m yields $HV_i \cap HV_j = \emptyset$.

Proof: $U_m(i) = L_m(j)-1 \Rightarrow U_m(i) < L_m(j) \Rightarrow R_m(i) \mathbin{//} R_m(j)$

and from Theorem 1, $HV_i \cap HV_j = \emptyset$. ▌

Definition 9: A hypervolume HV_i is said to be free from overlap at the upper end in the k-dimension relative to some other hypervolume HV_j iff

$$U_k(i) > U_k(j) \text{ and } L_k(i) \leq U_k(j).$$

We define the symbol $R_k(i)\ \overline{O}\ R_k(j)$ to denote this relation.

Note that in the relation $R_k(i)\ \overline{O}\ R_k(j)$ there is no restriction on the location of the lower bound $L_k(j)$, and thus $R_k(i)\ \overline{O}\ R_k(j)$ cannot be considered the inverse of $R_k(i)\ \underline{O}\ R_k(j)$. That is, $R_k(i)\ \overline{O}\ R_k(j)$ allows the possibility of overlap at both ends.

Theorem 3: Let HV_i and HV_j be any two hypervolumes in $\mathbf{X}$. If $R_m(i)\ \overline{O}\ R_m(j)$ for one or more m, then redefining HV_i so that $L_m(i) = U_m(j) + 1$ for any single m yields $HV_i \cap HV_j = \emptyset$.

Proof: $L_m(i) = U_m(j) + 1 \Rightarrow L_m(i) > U_m(j) \Rightarrow R_m(i)\ //\ R_m(j)$ and from Theorem 1, $HV_i \cap HV_j = \emptyset$. ∎

The construction of the index set Ψ is based upon the preceding definitions and theorems. Acceptance of a hypervolume for possible inclusion in Ψ is governed by the criterion that the hypervolume bounds satisfy Equation (II-6). By determining all unique hypervolumes under the function surface, HV_j, j=1, ..., M we have

$$\Psi = \bigcup_{j=1}^{M} HV_j \text{ and } \bigcap_{j=1}^{M} HV_j = \emptyset .$$

Aligning the hypervolumes along one coordinate axis enables a uniform search procedure. That is, the minimum value of the first coordinate in each hypervolume remains fixed at zero, since all hypervolumes must fall under the function surface and extend from each axis out toward the function surface.

We begin the construction by defining a set of numbers $\{L_1^*, L_2^*, \ldots, L_n^*\}$ which contains the minimum values each hypervolume coordinate can respectively assume at the most current step in the search procedure. Essentially, these starred values maintain a log of where a new volume can be located. This process will become more apparent as we move through the method description. Initially we define $L_k^* = 0$, k=1, ..., n. The value L_1^* remains locked at zero as mentioned above, while all others vary as the construction proceeds.

For clarity in the following explanation, each step in the construction process is numbered. Branching is indicated at appropriate places for comparisons or tests, to take alternative action, or to begin new cycles. The bounds of a hypervolume are determined by starting at a single point in the space and then stepping out in each of the coordinate directions to encompass as much volume as possible and still satisfy the function surface criterion.

Let i indicate the index of the current hypervolume being constructed; initially i=1.

1. Set $U_1(i)$ to the largest a_1 for which

$$(a_1, L_2^*, L_3^*, \ldots, L_n^*) \in \Psi, \; U_1(i) \geq L_1^* \equiv 0.$$

2. Set $U_2(i)$ to the largest a_2 for which

$$(U_1(i),\ a_2,\ L_3^*,\ L_4^*,\ \ldots,\ L_n^*)\ \varepsilon\ \Psi,\ U_2(i) \geq L_2^* .$$

3. Set $U_k(i)$ to the largest a_k for which

$$(U_1(i),\ \ldots,\ U_{k-1}(i),\ a_k,\ L_{k+1}^*,\ L_{k+2}^*,\ \ldots,\ L_n^*)\ \varepsilon\ \Psi,$$

$U_k \geq L_k^*$ with repeated looping on k where k=3, ..., n.

4. Set $L_k(i) = L_k^*$, k=1, ..., n. Note that $L_1(i) \equiv L_1^* \equiv 0$.

At this point in the algorithm a hypervolume has been defined, but may not be uniquely different from previously defined hypervolumes. That is, this new hypervolume must be pairwise disjoint with all previously defined hypervolumes before accepted into the index set Ψ . For this reason, the hypervolume must be checked against all other hypervolumes constructed. Holding HV_i fixed $(i > 1)$, each HV_j, j=1, ..., i-1 is compared coordinate by coordinate. For any given coordinate k, one and only one of the following relations will hold:

(a). $R_k(i) \mathbin{//} R_k(j)$	(Definition 6)
(b). $R_k(i) \subset R_k(j)$	(Definition 7)
(c). $R_k(i)\ \underline{0}\ R_k(j)$	(Definition 8)
(d). $R_k(i)\ \overline{0}\ R_k(j)$	(Definition 9)

5. Looping on the coordinate index k, k=1, ..., n compare hypervolume HV_i with hypervolume HV_j for one of the possible conditions described above.

(a). If $R_k(i) // R_k(j)$ for any k, then by Theorem 1 $HV_i \cap HV_j = \emptyset$. Go to step 6.

(b). If $R_k(i) \subset R_k(j)$ for all k, then by Definition 7, $HV_i \subset HV_j$ and HV_i cannot be included in Ψ . Go to step 8.

(c). If $R_k(i)\ \underline{0}\ R_k(J)$ for one or more k, then by Theorem 2 we can redefine HV_i so that $HV_i \cap HV_j = \emptyset$. Before redefining HV_i we must insure that condition (a) does not already exist. Thus, all coordinates in HV_i must be compared with those of HV_j. From Theorem 2, only one coordinate in HV_i should be redefined. Let m be the last coordinate for which $R_m(i)\ \underline{0}\ R_m(j)$. Set $U_m(i) = L_m(j) - 1$ and go to step 6.

(d). If $R_k(i)\ \overline{0}\ R_k(j)$ for one or more k, then by Theorem 3 we can redefine HV_i so that $HV_i \cap HV_j = \emptyset$. All coordinates in HV_i must be compared with those of HV_j to insure that condition (a) does not already exist. We may also have the situation where $\overline{0}$ exists for one coordinate and $\underline{0}$ for another coordinate. If so, go to 5c. If not, then let m be the last coordinate for which $R_m(i)\ \overline{0}\ R_m(j)$ and set $L_m(i) = U_m(j) + 1$. Go on to step 6.

6. Hypervolume HV_i is disjoint with HV_j. Increment j to j+1. If $j+1 < i$ go to step 5 for comparison of HV_i with the next hypervolume. If $j+1 = i$,

all comparisons have been made and we have $HV_i \cap HV_j = \emptyset$, j=1, ..., i-1. Store the bounds of HV_i, $L_1(i)$, ..., $L_n(i)$ and $U_1(i)$, ..., $U_n(i)$ to represent a new and acceptable hypervolume. Proceed to step 7.

7. With a new hypervolume accepted, the starred variables must reflect the new minimums acceptable in locating the next hypervolume.

   Set $L_2^* = \text{Max.} \left\{L_2^*, U_2(i)\right\} + 1 = U_2(i) + 1$.

   Hold L_3^*, ..., L_n^* fixed. $L_1^* \equiv 0$. Increment i to i+1 and go to step 9.

8. The hypervolume being tested for possible inclusion was already covered. Set $L_2^* = \text{Max.} \left\{L_2^*, U_2(i)\right\} + 1 = U_2(i) + 1$. Do not increment i. Go to step 9.

9. If $(0, L_2^*, L_3^*, \ldots, L_n^*) \in \Psi$, go to step 1 and begin construction of the next hypervolume.

   If $(0, L_2^*, L_3^*, \ldots, L_n^*) \notin \Psi$, set $L_2^* = 0$ and $L_3^* = \text{Min.} \left\{U_3(j) \middle| U_3(j) \geq L_3^*, j=1, \ldots, i\right\} + 1$.

   Go to Step 10.

10. If $(0, L_2^* = 0, L_3^*, L_4^*, \ldots, L_n^*) \in \Psi$, go to 1. Otherwise, set $L_3^* = 0$ and $L_4^* = \text{Min.} \left\{U_4(j) \middle| U_4(j) \geq L_4^*, j=1, \ldots, i\right\} + 1$. Go on to step 11.

11. The procedure continues as described in steps 9 and 10. Looping on k, k=3, ..., n-1, if $(0, L_2^*, \ldots, L_n^*) \in \Psi$, go to 1. Otherwise, set $L_k^* = 0$ and $L_{k+1}^* = \text{Min.} \left\{U_{k+1}(j) \middle| U_{k+1}(j) \geq L_{k+1}^*, j=1, \ldots, i\right\} + 1$.

Note that if there exists a value $U_{k+1}(j) = L^*_{k+1}$ we can immediately define the new L^*_{k+1} without further testing in this coordinate. Check the new starred point for containment in Ψ .

12. A point will be reached when, through all the searching, $L^*_k = 0$, $k=1, \ldots, n-1$ and $(0, 0, \ldots, 0, L^*_k) \notin \Psi$. The next logical step would be to define $L^*_n = 0$, but this returns us to the very first starting point. Hence, the construction has been completed with i-1 hypervolumes in Ψ .

In the construction algorithm, we note in steps 9, 10, and 11 that there exists at least one value $U_k(j)$ so that $U_k(j) \geq L^*_k$; namely, the most recent upper bound determined in this dimension as so defined in steps 1, 2, or 3. Furthermore, since it may well be that this most recent upper bound is equal to L^*_k and the new L^*_k could be immediately defined without further checking, a descending search with $j=i, i-1, \ldots, 1$ is most expedient.

As remarked earlier, several different approaches were planned and written for the construction of Ψ . The advantages in doing so were twofold. First, insight and understanding of the problems of construction were gained which ultimately led to the method of hypervolumes. Questions such as how to present the index set most uniformly and concisely for explanation, how equations could most effectively be redesigned for efficient evaluation, and how systems of differing dimensions and sizes could be compared were more easily answered.

The other advantage in having different approaches was the ability to check the components of index sets for various systems. The method of construction used in comparison testing was one in which all the n-tuple components were found one at a time. This method can be described somewhat analogously by the operation of an odometer. That is, each coordinate assumed a "register" position. All registers started at zero and operated from right to left (n^{th} coordinate to 1^{st} coordinate). A coordinate register value would increment until reaching the criterion function acceptance value, at which time a pointer moved to the next register in sequence and the previously used registers would restart at zero. This search method continued until all registers had moved to their maximum acceptable limit. During each step the current register readings were recorded and thus the total n-tuple readings made up the index set.

Tests were performed on various systems ranging from two to six dimensions, and one system of nine dimensions. In all instances, the index sets constructed by the hypervolume method were exactly the same as those constructed by the register method.

Apart from the comparison checks, other tests were performed. Defining the number of failures found in each system as zero (i.e., $x_i = 0$, i=1, ..., n,) several different systems were analyzed and each index set was determined. As desired, each set did consist of the single n-tuple (0, 0, ..., 0).

Another type of checking done was to consider subsets of index sets. Given a particular system with some number of failures input, x_i, i=1, ..., n, a corresponding index set was found. Then by reducing the number of failure values input, other index sets were determined. In each case tested, the smaller index set was completely contained by the larger set. This test was repeated several times for the systems ranging from two to six dimensions, as well as for the nine-dimensional system.

A slightly different approach was to select some n-tuple within a given index set as input to the system under consideration. As expected, the new resulting index set in each case was a subset of the original set and included all of the proper n-tuples.

The use of hypervolumes presents a concise and uniform method of describing the index set Ψ. As will be seen in the following section of this report, the most important feature of the method of hypervolumes is the very efficient manner in which complex equations such as (II-3) can be evaluated.

IV. Nonlinear Function Maximization

The problem of finding the maximum of a nonlinear function of several variables with nonlinear constraints is well known in numerical analysis. Considerable attention has been given to the minimization of nonlinear functions and several different approaches have been proposed [2]. Although normally posed in terms of minimization, by making a few minor changes the methods apply to maximization as well. In nearly all of the techniques proposed, the partial derivatives of the given function to be maximized, with respect to each independent variable, are necessary to provide input as to the direction of greatest change [3] and as a measure of the convergence process [4].

Considering the fact that we are concerned with probability functions involving several-to-many variables, $f(p_1, \ldots, p_n)$, the requirement for partial derivatives becomes rather undesirable. This undesirability is further magnified, since the user of the general CØNLIM computer program is responsible for supplying the function to be maximized. To circumvent these difficulties, we propose a modified univariate method; i.e., changing one variable at a time.

Since both f and the p values are probabilities, they must satisfy the conditions

$$0 \leq f(p_1, \ldots, p_n) \leq 1 \qquad \text{(IV-1)}$$

and

$$0 \leq p_i \leq 1 \quad , \qquad i = 1, \ldots, n \ .$$

For ease of illustration, diagrams supplied will be in terms of only two dimensions, but the theory and application hold for higher dimensions and have been used in this context. Hence, in two dimensions we can view the function f as a family of curves dependent upon the two variables p_1 and p_2. Figure 3 illustrates a possible configuration of the function curves and the constraint curve $H(p_1,p_2) = 1 - \sigma$. We search for the point at which the curve $H = 1 - \alpha$ is tangent to the curve f_k. The value of f_k is then the confidence limit we are seeking.

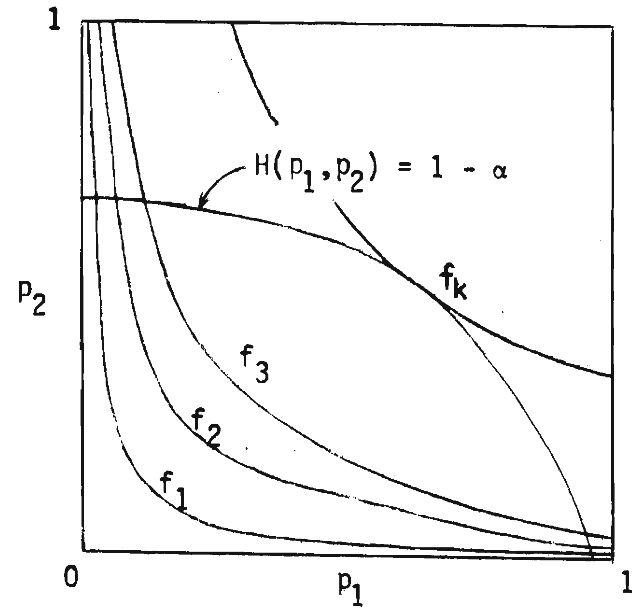


Figure 3

The search is initiated from the point

$$\overline{P} = (\overline{p}_1, \ldots, \overline{p}_n) \quad , \quad \overline{p}_i = \frac{x_i + 1}{m_i + 2} \quad , \quad x_i \leq m_i \quad ; \qquad \text{(IV-2)}$$
$$i = 1, \ldots, n \quad .$$

By using Steck's iterative procedure [1] for an index set, one obtains the global maximum function value. But we wish to avoid an impractically long iteration process to determine this set. Differing from Steck's approach, we admit tie points in the set ordering. The starting point $\overline{P}$ was chosen by Müller to direct the search toward a maximum judged to correspond most closely with the global. However, the search routine could conceivably find a local maximum rather than the global maximum. When this occurs, it is as though we are saying that the index set generated is not producing the ordering we want and we effectively reject (or ignore) those points in the ordering which would create a global maximum elsewhere. The contribution of those "global" points at the local maximum is very small, so the value of the local maximum is affected very little by whether they are, or are not, a part of the ordering. This has the unsettling aspect that we cannot precisely define in mathematical terms the ordering we are using, but the ordering is uniquely and precisely defined by the computer algorithm. The point is that <u>any</u> definite, repeatable ordering will produce a valid system of confidence limits, and we have such an ordering judged to be, if anything, somewhat better than the implied ordering identified by the index set.

With reference to Fig. 4, the procedure of maximization to be used will first be briefly described for an overview and then discussed in detail. Note that initially the $\overline{p}_i$ values are held away from the end (boundary) conditions so that $0 < \overline{p}_i < 1$.

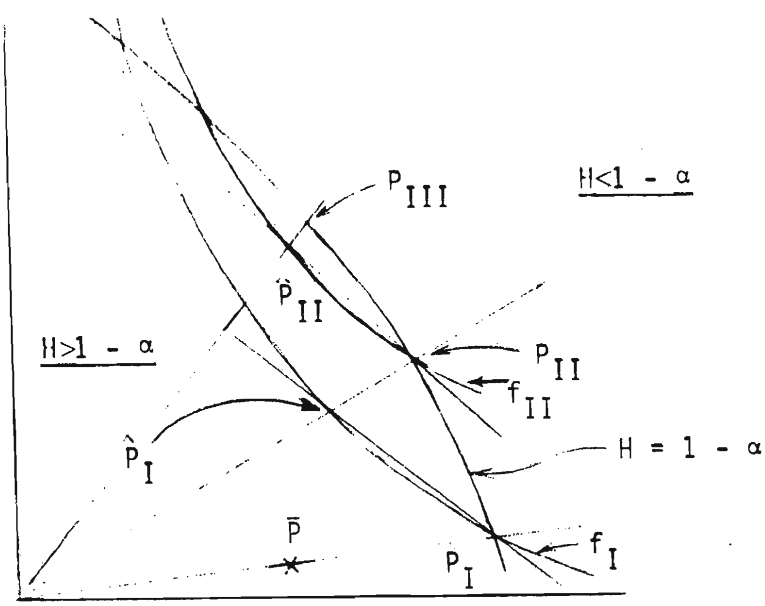


Figure 4

The computational complexity of the H function and the extreme amount of time that would be required for many evaluations of H preclude moving along the curve $H = 1 - \alpha$ to find the maximum of f. Consequently, we have followed the suggestion of Müller and "inverted" the maximization procedure. From the initial point $\overline{P}$ we move to the $1 - \alpha$ constraint curve along a line intersecting $\overline{P}$ and the origin. In so doing, the point $P_I = (p_1, \ldots, p_n)_I$ on the $1 - \alpha$ curve is obtained and we then calculate the corresponding function value $f(P_I)$. A step of appropriate length, δ_1, is taken to the point $\hat{P}_I$ along a constant f curve (i.e., $f_I = f(P_I) = C$), in the direction of increasing H values. From $\hat{P}_I$ we return to the $1 - \alpha$ curve at the point P_{II} directly away from the origin. Since H is monotonically decreasing away from the origin while F is monotonically increasing away from the origin, the new point P_{II} will have a larger f value than that for P_I; i.e., $f(P_{II}) > f(P_I)$. Thus the general direction of new point placement is toward the interior of the envelope formed between $H = 1 - \alpha$ and f_I. After determining f_{II}, the next function curve to be held constant, the same procedure is repeated. This iterative process is continued until a step with fixed δ is found to produce a corresponding H value less than the H value before the step was taken; i.e., until the step places the next point outside the current envelope and beyond the intersection of the constant $1 - \alpha$ and f curves. The step size is then reduced to $\delta_2 = \delta_1/2$ and the iteration continued. When the step size has been reduced to some value $\delta_k = \delta_{k-1}/2 < 10^{-d}$, where d is the number of digits at which the variables p_i are to be affected by the steps, the entire procedure is halted. Through this inverted method the function f evaluated at the last point, P^*, on the $H = 1 - \alpha$ curve produces the desired maximum of the function. The number of evaluations of H required by this approach is significantly reduced since H is only computed in each cycle to establish direction and to return to $1 - \alpha$ from the constant f curves.

Keeping in mind the procedure just described, we now detail the equations and formulation necessary to carry out these operations. During each part of the iteration, we select some point $\hat{P}_j = (\hat{p}_i, \ldots, \hat{p}_n)_j$ away from the $1 - \alpha$ constraint curve that increases the value of H and places us toward the center of the envelope. Each point $\hat{P}_j$ is forced to maintain a constant function value f_j; i.e., movement to the point $\hat{P}_j$ is along the f_j curve. To enable such a movement, we place a restriction on the system, and thus on the function f. Each component of the system can be described as connected in series or parallel with the other components. The system is restricted in that it must contain at least one series connected component. In Fig. 5, let Q represent the failure probability accumulated over n - 1 components, and p_n represent the failure probability of a single series component. We simply remark at this point in the discussion that when there exists more than one series component in the system, that p_n represents the series component probability of greatest value within the immediate step cycle. The subscript n is used to distinguish that component and the corresponding value from the remainder of the system, and does not necessarily indicate the final system component. For clarification and future reference, we shall henceforth refer to the series component considered separately in each cycle as the "slack" component since it is essentially used to take up the slack in holding to a constant f curve.

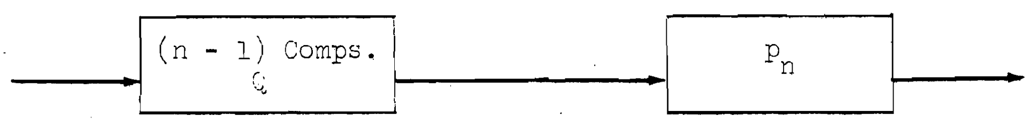


Figure 5

Then the failure probability of the system becomes

$$f = 1 - [(1 - Q)(1 - p_n)]$$
$$= Q + p_n (1 - Q) \qquad \text{(IV-3)}$$

Incrementing a parallel component by a relatively large step value produces approximately the same function incrementation as when a series component is only slightly modified. Hence, two separate step values, δ_s for series components and δ_p for parallel, are maintained. A too-small initial step would result in many cycles of maximization and increased computer time, and a very large start might overstep the envelope and possibly influence the direction of movement, initially, away from the maximum. In terms of an optimum first step size, experience has shown the initial step values to be best defined as

$$\delta_s = \left(\sum_{i=1}^{n} \bar{p}_i\right) \Big/ 10\,n, \quad \delta_p = \sqrt{\delta_s} \quad .$$

The series components are affected, at most, in the second decimal place and parallel components in the first or second place, depending on the average $\bar{p}$ values. In the following discussion, the step size will always be referred to in the generic sense, δ, with the actual value used assumed dependent upon the component designation.

Starting on the constraint curve in the jth iteration ($j = 1, \ldots, M$) with a given point $P_j = (p_1, \ldots, p_n)_j$, we calculate the intersecting function value $f(P_j)$ providing the "constant" curve along which a new point $\hat{P}_j = (\hat{p}_1, \ldots, \hat{p}_n)_j$ is to be found. The variable p_{n-1} is modified by some step value δ so that $\hat{p}_{n-1} = p_{n-1} + \delta$ and all other variables, p_i, $i \neq n - 1$, are left unchanged. From the series restriction on the slack component we can insure this step to fall on the "f-constant" curve by using (IV-3) to give

$$\hat{p}_n = \frac{f(P_j) - Q(p_1, p_2, \ldots, \hat{p}_{n-1})}{1 - Q(p_1, p_2, \ldots, \hat{p}_{n-1})} \quad . \qquad \text{(IV-4)}$$

Hence, the slack component takes up the difference necessary to maintain $f(P_j)$ and assumes the value $\hat{p}_n$.

The function H is evaluated for $(p_1, p_2, \ldots, \hat{p}_{n-1}, \hat{p}_n)$. Should it be the case that we have violated the boundary condition (IV-1) or that H has decreased, the value of the slack component is set back to p_n and we try to step in the opposite direction, $\hat{p}_{n-1} = p_{n-1} - \delta$. A new value for $\hat{p}_n$ is obtained from (IV-4) to hold to the curve and we again check this value. Provided that (IV-1) is satisfied and H has increased, p_{n-1} becomes fixed at the new value $\hat{p}_{n-1} = p_{n-1} \pm \delta$. If (IV-1) is not satisfied, $\hat{p}_{n-1} = p_{n-1}$.

The univariate process continues by adjusting the next component p_{n-2}, with δ. Modification of the slack component is again made to place the search point on the constant curve and (IV-1) is checked. However, we are also seeking to position the completely stepped point $\hat{P}_j$ at the greatest value of $H(\hat{P}_j)$ that can be obtained within the δ step limit. Before accepting the value $\hat{p}_{n-2} = p_{n-2} \pm \delta$ for this component, a comparison is made with the previously held position and we find that

$$\Delta H_{n-2} = H(p_1, p_2, \ldots, \hat{p}_{n-2}, \hat{p}_{n-1}, \hat{\hat{p}}_n) - H(p_1, p_2, \ldots, p_{n-2}, \hat{p}_{n-1}, \hat{p}_n) \quad .$$

Based on this calculation the following selection is made:

$$\hat{p}_{n-2} = \begin{cases} p_{n-2} \pm \delta \text{ if } \Delta H_{n-2} \geq 0 \\ p_{n-2} \text{ if } \Delta H_{n-2} < 0 \quad . \end{cases}$$

Taking one coordinate at a time, the iteration proceeds until all n - 1 components have been incremented, decremented, or left unchanged. At each step along the way the slack component is modified to hold the moving point on the constant curve, and also at each step the boundary condition (IV-1) is checked and a comparison made on the H function increase. In general, we can summarize the component values by

$$\hat{p}_i = \begin{cases} p_i \pm \delta \text{ if } H_i > 1 - \alpha \; ; \Delta H_i \geq 0 \; ; \; 0 \leq p_i \leq 1 \; ; 0 \leq \hat{p}_n^{\,i} \leq 1 \\ p_i \qquad \text{otherwise} \end{cases} \qquad \text{(IV-5)}$$

$$i = 1, \ldots, n - 1$$

and

$$\hat{p}_n = \hat{p}_n^{\,n-1} \; ,$$

where the designation $\wedge_i$ above the slack component denotes that p_n has been modified i times, as described in the text and as seen in (IV-4).

Although not obvious at first, one can see that if the same series component is held fixed as the slack component throughout the entire maximization procedure there exists the possibility of exhausting whatever magnitude and contribution that component may yield. In the situation where one component must continually be forced to maintain the difference to follow a constant curve, a resultant zero probability would halt the procedure since there would no longer be values to use as slack. Stopping at this point would quite possibly result in an erroneous solution. Also, concern that an undesirable bias in movement could result from a single slack component leads us to the position that all series components be considered for the slack role. The condition that a sizable adjustment might be necessary to remain on a constant f curve dictates the slack component to be the largest series component going into a step cycle. This component is held as slack through the complete step cycle, but then allowed to be reconsidered after a return is made to the $1 - \alpha$ curve.

Upon the completion of the univariate step procedure, a new point $\hat{P}_j = (\hat{p}_1, \ldots, \hat{p}_n)_j$ has been found so that

$$f(\hat{P}_j) = f(P_j) \text{ and } H(\hat{P}_j) > H(P_j) \quad . \qquad \text{(IV-6)}$$

With the point $\hat{P}_j$ fixed, we can determine another point, P_{j+1}, on the curve $1 - \alpha$ whose f value is greater (and therefore closer to the maximum) than $f(P_j)$. To do so, $\hat{P}_j$ and the origin are aligned as shown in Fig. 6. The next point, P_{j+1}, is calculated via a first-order approximation of the H function.

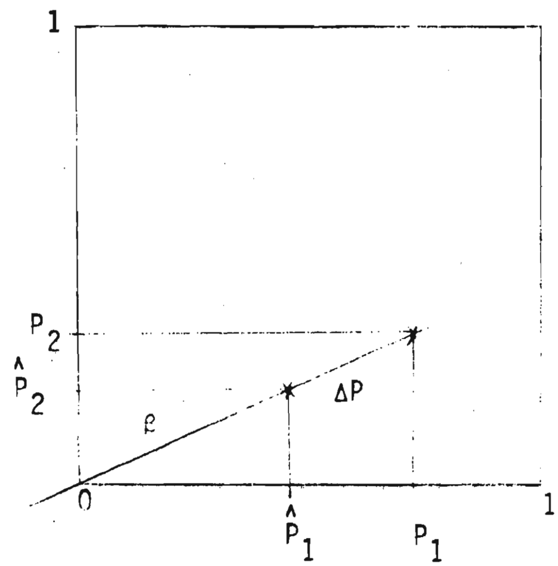


Figure 6.

Hence, for $H(\hat{P}) > 1 - \alpha$,

$$p_i = \hat{p}_i + (p_i - \hat{p}_i) = \hat{p}_i + \frac{\hat{p}_i}{\beta} \Delta p \quad , \qquad \text{(IV-7)}$$

where

$$\beta = \left\{ \sum_{k=1}^{n} \hat{p}_k^2 \right\}^{1/2} \quad . \qquad \text{(IV-8)}$$

From equation (II-3) and the form of H, we have

$$H(P) = \sum_{\psi} \prod_{i=1}^{n} \binom{m_i}{a_i} \left(\hat{p}_i + \hat{p}_i \frac{\Delta p}{\beta} \right)^{a_i} \left(1 - \hat{p}_i + \hat{p}_i \frac{\Delta p}{\beta} \right)^{m_i - a_i} \quad . \qquad \text{(IV-9)}$$

Expanding the first factor,

$$\left(\hat{p}_i + \hat{p}_i \frac{\Delta p}{\beta}\right)^{a_i} = \hat{p}_i^{\,a_i} + \binom{a_i}{1}\hat{p}_i^{\,a_i-1}\,\hat{p}_i \frac{\Delta p}{\beta} + \text{---} + \left(\hat{p}_i \frac{\Delta p}{\beta}\right)^{a_i} . \quad \text{(IV-10)}$$

Similarly, expanding the second factor of (IV-9), we have

$$\left(1 - \hat{p}_i + \hat{p}_i \frac{\Delta p}{-\beta}\right)^{m_i-a_i} = \left(1 - \hat{p}_i\right)^{m_i-a_i} + \binom{m_i-a_i}{1}\left(1 - \hat{p}_i\right)^{m_i-a_i-1}\,\hat{p}_i \frac{\Delta p}{-\beta}$$

$$+ \text{---} + \left(p_i \frac{\Delta p}{-\beta}\right)^{m_i-a_i} . \quad \text{(IV-11)}$$

By discarding all terms involving Δp^2 and higher orders in (IV-10) and (IV-11), the two factors become

$$\left[\hat{p}_i^{\,a_i} + a_i\hat{p}_i^{\,a_i-i}\,\hat{p}_i \frac{\Delta p}{\beta}\right]\left[\left(1 - \hat{p}_i\right)^{m_i-a_i} + (m_i - a_i)(1 - \hat{p}_i)^{m_i-a_i-1}\,\hat{p}_i \frac{\Delta p}{-\beta}\right]$$

$$= \hat{p}_i^{\,a_i}\,(1 - \hat{p}_i)^{m_i-a_i}\left\{1 + \frac{\Delta p}{\beta}\left(\frac{a_i - m_i\hat{p}_i}{1 - \hat{p}_i}\right)\right\} . \quad \text{(IV-12)}$$

Substituting (IV-12) into (IV-9), we approximate H(P) by

$$H(P) \approx \sum_{\Psi} \prod_{i=1}^{n} \binom{m_i}{a_i} \hat{p}_i^{\,a_i}\left(1 - \hat{p}_i\right)^{m_i-a_i}\left[1 + \frac{\Delta p}{\beta}\left(\frac{a_i - m_i\hat{p}_i}{1 - \hat{p}_i}\right)\right] \quad \text{(IV-13)}$$

$$= \sum_{\Psi}\left\{\prod_{i=1}^{n} \binom{m_i}{a_i} \hat{p}_i^{\,a_i}\left(1 - \hat{p}_i\right)^{m_i-a_i}\right\}\left\{\prod_{i=1}^{n}\left[1 + \frac{\Delta p}{\beta}\left(\frac{a_i - m_i\hat{p}_i}{1 - \hat{p}_i}\right)\right]\right\}.$$

Expanding the second product of (IV-13),

$$\prod_{i=1}^{n}\left[1+\frac{\Delta p}{\beta}\left(\frac{a_i - m_i\hat{p}_i}{1-\hat{p}}\right)\right] = 1 + \frac{\Delta p}{\beta}\sum_{i=1}^{n}\frac{a_i - m_i\hat{p}_i}{1-\hat{p}_i} + \sum \text{Cross Product Terms} \quad . \qquad \text{(IV-14)}$$

Ignoring the cross product terms and substituting (IV-14) into (IV-13) gives

$$H(P) \approx \sum_{\Psi}\left\{\prod_{i=1}^{n}\binom{m_i}{a_i}\hat{p}_i^{\,a_i}\left(1-\hat{p}_i\right)^{m_i-a_i}\right\}\left\{1+\frac{\Delta p}{\beta}\sum_{i=1}^{n}\frac{a_i - m_i\hat{p}_i}{1-\hat{p}_i}\right\}$$

$$= \sum_{\Psi}\prod_{i=1}^{n} b_i + \frac{\Delta p}{\beta}\sum_{\Psi}\left(\prod_{i=1}^{n} b_i\right)\left(\sum_{i=1}^{n} c_i\right) \qquad \text{(IV-15)}$$

$$= H(\hat{P}) + \frac{\Delta p}{\beta} B \, ,$$

where

$$B = \sum_{\Psi}\left(\prod_{i=1}^{n} b_i\right)\left(\sum_{i=1}^{n} c_i\right) \; ; \qquad \text{(IV-16)}$$

$$b_i = \binom{m_i}{a_i}\hat{p}_i^{\,a_i}\left(1-\hat{p}_i\right)^{m_i-a_i} \; ; \qquad \text{(IV-17)}$$

$$c_i = \frac{a_i - m_i\hat{p}_i}{1-\hat{p}_i} \; . \qquad \text{(IV-18)}$$

The goal of this approximation is to obtain $H(P) = 1 - \alpha$, so that

$$1 - \alpha = H(\hat{P}) + \frac{\Delta p}{\beta} B$$

yielding

$$\Delta p = \frac{(1-\alpha) - H(\hat{P})}{B} \beta . \qquad \text{(IV-19)}$$

Substituting (IV-19) into (IV-7) gives

$$p_i = \hat{p}_i + \hat{p}_i \left[\frac{(1-\alpha) - H(\hat{P})}{B} \right] . \qquad \text{(IV-20)}$$

Equation (IV-20) is also used to make the first step adjustment from the initial point estimate, $\bar{P} = (\bar{p}_1, \cdots, \bar{p}_n)$, to the constraint curve provided that $H(\bar{P}) > 1 - \alpha$. However, it may well be the case that this point lies above the constraint curve; i.e., $H(\bar{P}) < 1 - \alpha$. In this situation, equation (IV-7) should be modified to reflect the opposite direction:

$$p_i = \hat{p}_i - \frac{\hat{p}_i}{\beta} \Delta p = \hat{p}_i + \frac{\hat{p}_i}{-\beta} \Delta p .$$

But by substituting $-\beta$ throughout the equations derived, we again obtain equation (IV-20). Equation (IV-20) is valid for both situations, since the denominator term B is negative in each case.

As in many other methods of maximization (minimization) there may arise the problem of oscillation with no progress made toward the maximum sought. Simply stated, this situation would occur when the step process becomes trapped on a curve, f, and an oscillation is encountered with slack components alternating or "vying" for direction control. In this situation, there are more than one local maximum, but only one such maximum appropriate for the system under study.

For example, a family of f curves containing a ripple or wave effect could lead the maximization procedure to "stall out" in that each new try to step to a greater f function value via (IV-20) only results in positioning the new point on the opposite side of an f curve wave wall. As different slack components are utilized repeatedly to hold to this f curve it is evident that several paths to maximum values may exist. The immediate problem then becomes one of selecting the proper path to follow for the given system.

To recognize an oscillation, we maintain a survey of the pattern of slack components being used and, when a specific pattern is found to be repeating r times, a check on progress is made. If sufficient progress toward a maximum is taking place, then we do not want to interrupt the process but allow it to continue making progress. On the other hand, if very little or no progress is being made, then a junction exists and the different branches must be explored; i.e., we must force the maximization process to follow each branch path.

Let p_k , $k = 1, \ldots, K$ represent the components being used in the slack role at the time of oscillation detection. The criteria for sufficient progress is established by examining the probability values of the slack component p_1 held over this oscillation period, $p_1^{(i)}$, $i = 1, \ldots, r$. Remember that after any component values are modified by δ, an adjustment is made through (IV-20) back to the constraint curve. If each value held by p_1 during the period did not change more than one-half the current step size, δ, then we indicate that a nonprogressing oscillation has occurred. That is, when

$$\left| p_1^{(i+1)} - p_1^{(i)} \right| < \delta/2 \ , \ i = 1 \ , \ \ldots \ , \ r - 1 \ . \qquad \text{(IV-21)}$$

This situation dictates a slightly different approach to reach a global maximum. All p values, the current function value f, and the current H value are saved at the junction point and each path is then separately followed through the maximization procedure but with a fixed slack component for each path. The process is illustrated in Fig. 7.

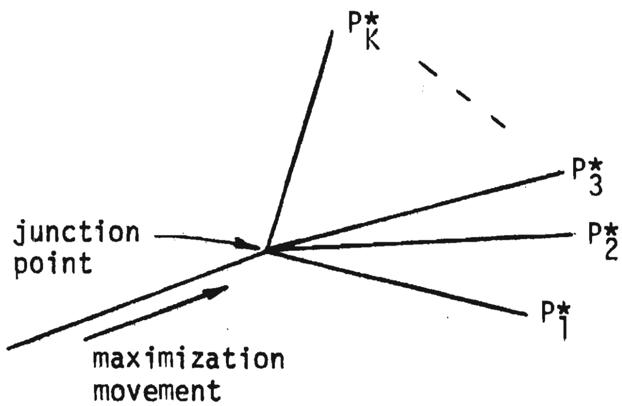


Figure 7.

Each point P_k^* , $k = 1, \ldots , K$ reached is the maximum point satisfying (II-2) along the respective K paths taken. For the first path, slack component p_1 is held as the only slack choice and no other components are considered for the slack role. The same procedure is followed for each kth path with the kth slack component fixed for that path. Each time a P_k^* maximum is obtained, the maximization procedure restarts at the junction point by picking up the saved junction values.

In the computer code, CONLIM, all P_k^* values are provided on output and the final selection of the "true" global maximum is left to the discretion of the code user. This method permits the user to consider the system structure, as well as any other exterior conditions, in making the selection. We note that in using CONLIM, the user has the option to not pursue any paths but rather to stop the problem at the junction point. Primary motivation for such a choice is the additional computer time necessary to complete the problem which would need to be weighed against the desirability of solution.

V. Hypervolumes and the H-Function for Maximization

In a previous section, hypervolumes were defined and developed to form the index set Ψ. As the size of Ψ increases, the time to evaluate equations (II-3) and (IV-16) also increases. This time increase becomes magnified by the fact that larger systems require more evaluations of these two equations. Aside from the desire to be able to express Ψ compactly by means of hypervolumes, the other primary motivation for using hypervolumes is to facilitate the most efficient and rapid evaluation of these two major equations.

Expressing equation (II-3) in terms of summation over hypervolumes gives

$$H(P) = \sum_{j=1}^{M} \left\{ \sum_{A \in HV_j} \prod_{i=1}^{n} \binom{m_i}{a_i} p_i^{a_i} \left(1 - p_i\right)^{m_i - a_i} \right\}. \quad \text{(V-1)}$$

Over each hypervolume HV_j we can factor the summation and product to obtain a reduced form so that

$$\sum_{A \in HV_j} \prod_{i=1}^{n} \binom{m_i}{a_i} p_i^{a_i} \left(1 - p_i\right)^{m_i - a_i}$$

$$= \prod_{i=1}^{n} \sum_{k=min_{ij}}^{max_{ij}} \binom{m_i}{k} p_i^{k} \left(1 - p_i\right)^{m_i - k}, \quad j=1, \ldots, M, \quad \text{(V-2)}$$

where min_{ij} = minimum value of ith coordinate, jth hypervolume;

max_{ij} = maximum value of ith coordinate, jth hypervolume.

Let us consider an example of application using equation (V-2). Suppose that we have a system with n = 3 and examine a single hypervolume with the following coordinate ranges:

coordinate	min value	max value
1	0	2
2	0	3
3	1	2

The hypervolume contains 24 points. If we use only the binomial coefficients as shorthand representation for complete terms, then by (V-1)

$$\sum_{A \epsilon HV_j} \prod_{i=1}^{3} \binom{m_i}{a_i} = \binom{m_1}{0}\binom{m_2}{0}\binom{m_3}{1} + \cdots + \binom{m_1}{2}\binom{m_2}{3}\binom{m_3}{2} , \qquad \text{(V-3)}$$

whereas the representation for (V-2) would be

$$\prod_{i=1}^{3} \sum_{k=min_{ij}}^{max_{ij}} \binom{m_i}{k} = \left[\binom{m_1}{0} + \binom{m_1}{1} + \binom{m_1}{2}\right]\left[\binom{m_2}{0} + \binom{m_2}{1} + \binom{m_2}{2} + \binom{m_2}{3}\right]\left[\binom{m_3}{1} + \binom{m_3}{2}\right].$$

(V-4)

The arithmetic operations to be made in each of these equations are summarized in **Table I.**

Table I

Type of Operation	Number of Operations Necessary	
	Equation (V-3)	Equation (V-4)
Exponentiation	144	18
Multiplication	192	20
Addition	23	6

A similar but somewhat more involved factorization of equation (IV-16) can also be made. In terms of summation over hypervolumes (IV-16) becomes

$$B = \sum_{j=1}^{M} \left\{ \sum_{A \epsilon HV_j} \left(\prod_{i=1}^{n} b_i \right) \left(\sum_{i=1}^{n} c_i \right) \right\}. \tag{V-5}$$

Over any single hypervolume, HV_j, we can rewrite the summations and products so that

$$\sum_{A \epsilon HV_j} \left\{ \prod_{i=1}^{n} \binom{m_i}{a_i} p_i^{a_i} \left(1 - p_i\right)^{m_i - a_i} \right\} \left\{ \sum_{i=1}^{n} \frac{a_i - m_i p_i}{1 - p_i} \right\}$$

$$= \sum_{\ell=1}^{n} \left\{ \prod_{i=1}^{n} \left[\sum_{k=min_{ij}}^{max_{ij}} \binom{m_i}{k} p_i^{k} \left(1 - p_i\right)^{m_i - k} \delta_{i\ell} \right] \right\} \qquad \text{(V-6)}$$

where min_{ij} = minimum value of ith coordinate, jth hypervolume;

max_{ij} = maximum value of ith coordinate, jth hypervolume;

and

$$\delta_{i\ell} = \begin{cases} 1 & \text{if } i \neq \ell \\ \dfrac{k - m_i p_i}{1 - p_i} & \text{if } i = \ell. \end{cases} \qquad \text{(V-7)}$$

Note that in (V-6) and (V-7) the function $\delta_{i\ell}$ has no relation whatsoever to the step size δ used earlier in the discussion. The step function $\delta_{i\ell}$ is necessary in (V-6), since we are concerned with a product/sum combination over hypervolumes rather than a product-only operation. Equation (V-6) has the feature of being able to use the hypervolume information available without intermediate steps of hypervolume component rearrangement necessary for equation (V-5).

The most illustrative, nontrivial example for comparing (V-6) with (V-5) is a two-dimensional system where the hypervolume coordinate minimums and maximum have the following values:

coordinate	min value	max value
1	0	2
2	0	1

Let $c_{ik} = \frac{k - m_i p_i}{1 - P_i}$ and

$\binom{m_i}{k}$ represent $\binom{m_i}{k} p_i^{\,k} \left(1 - p_i\right)^{m_i - k}$.

Then by (V-5),

$$\sum_{A \epsilon HV_j} \left(\prod_{i=1}^{2} b_i\right)\left(\sum_{i=1}^{2} c_i\right) =$$

$$\left\{\begin{array}{l} \binom{m_1}{0}\binom{m_2}{0}\left[c_{10} + c_{20}\right] + \binom{m_1}{0}\binom{m_2}{1}\left[c_{10} + c_{21}\right] + \\ \binom{m_1}{1}\binom{m_2}{0}\left[c_{11} + c_{20}\right] + \binom{m_1}{1}\binom{m_2}{1}\left[c_{11} + c_{21}\right] + \\ \binom{m_1}{2}\binom{m_2}{0}\left[c_{12} + c_{20}\right] + \binom{m_1}{2}\binom{m_2}{1}\left[c_{12} + c_{21}\right] \end{array}\right\} . \qquad \text{(V-8)}$$

From (V-6),

$$\sum_{\ell=1}^{2}\left\{\prod_{i=1}^{2}\left[\sum_{k=min_{ij}}^{max_{ij}}\binom{m_i}{k}\delta_{i\ell}\right]\right\} =$$

$$\left\{\begin{array}{l}\left[\binom{m_1}{0}c_{10}+\binom{m_1}{1}c_{11}+\binom{m_1}{2}c_{12}\right]\left[\binom{m_2}{0}+\binom{m_2}{1}\right]+\\ \left[\binom{m_1}{0}+\binom{m_1}{1}+\binom{m_1}{2}\right]\left[\binom{m_2}{0}c_{20}+\binom{m_2}{1}c_{21}\right]\end{array}\right\} . \qquad \text{(V-9)}$$

Equations (V-8) and (V-9) can be shown to be comparable by collecting like terms in each.

Assuming the c_{ik} terms can be evaluated once and then inserted in the equations wherever needed, we summarize the number of operations necessary to complete (V-8) and (V-9) in Table II.

Table II

Type of Operation	Number of Operations Necessary	
	Equation (V-8)	Equation (V-9)
Exponentiation	24	20
Multiplication	36	27
Addition	11	7

Without writing the expansions, we find the number of operations necessary to evaluate (V-5) and (V-6) for the first example summarized in Table III.

Table III

Type of Operation	Number of Operations Necessary	
	Equation (V-5)	Equation(V-6)
Exponentiation	144	54
Multiplication	216	69
Addition	71	20

In all of these equations, some of the operations can be avoided by storing similar terms for repeated use. This is particularly true for the exponentiations that must be performed. But clearly, for larger hypervolumes and larger systems the chasm between the number of operations necessary to evaluate (V-2) and (V-6) as opposed to (V-1) and (V-5) increases substantially.

Computationally, the evaluation of the binomial coefficients can be accelerated by

(1) selecting some integer N, computing n! once for each integer $n = 1, \cdots, N$, and storing these factorials in an array for ready access whenever needed;

(2) approximating the binomial coefficient involving integers greater than N.

The value of N chosen is dependent primarily on the exponent range of the computer being used. Of secondary consideration is the size of the integers to be encountered in most problems; i.e., the number of component tests that will normally be made and the size of N covering most values. Rewriting the binomial coefficient in terms of gamma functions, we have

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = \frac{\Gamma(n+1)}{\Gamma(r+1)\,\Gamma(n-r+1)} \quad . \qquad \text{(V-10)}$$

Stirling's formula [5] for an approximation to the logarithm of the gamma function is

$$\ell n\,\Gamma(n) \sim (n - \tfrac{1}{2})\,\ell n\, n - n + \tfrac{1}{2}\,\ell n\, 2\pi + \frac{1}{12n} \cdot \qquad \text{(V-11)}$$

Using the logarithmic form (V-11) in (V-10), we find

$$\ln\binom{n}{r} = n\ln\left(\frac{n+1}{n-r+1}\right) + r\ln\left(\frac{n-r+1}{r+1}\right) + \tfrac{1}{2}\ln\left\{\frac{n+1}{(r+1)(n-r+1)}\right\}$$

$$+ (1 - \tfrac{1}{2}\ln 2\pi) + \frac{r(n-r) - (n+1)^2}{12(n+1)(r+1)(n-r+1)} \quad . \qquad \text{(V-12)}$$

Combining the numerator and denominator terms of (V-10) into a single operation prevents exponent overflow that might occur should each term be evaluated separately and then a division be attempted. This is particularly true in the situation where the value of N has been selected to permit as great an exponent range as possible for calculating N!.

In some problems it may be that the initial starting point $\bar{P}$ is so far above the constraint curve H, $H(\bar{P}) << 1 - \alpha$, that the first-order approximation (IV-20) cannot provide a sufficient Δp correction to even place the next point in the vicinity of the constraint curve. This condition can be likened to one of trying to determine the root of an equation through the Newton-Raphson method. It is well known that should a poor initial guess be used in starting the iteration, the method can place the next step completely away from the root to be found. The first-order approximation in (IV-20) reacts in a very similar manner. Fig. 8 illustrates the situation that can arise.

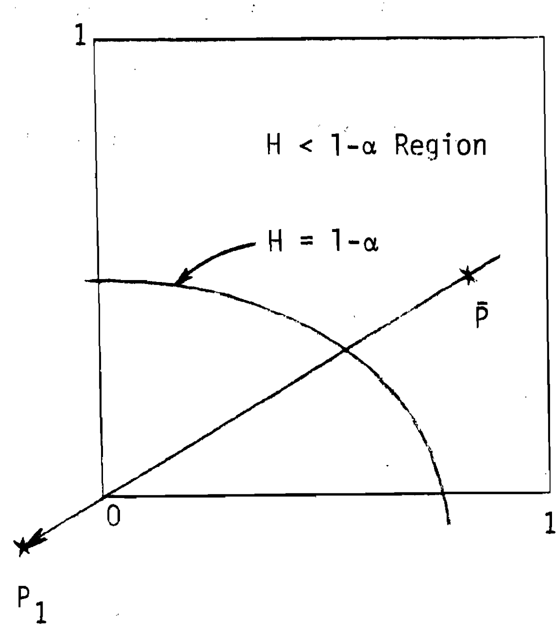


Figure 8

The proposed improvement places the point P_1 in violation of the boundary conditions imposed upon the variables.

By eliminating the higher orders from the approximation to simplify the resulting equation (IV-12), the contribution necessary from the higher ordered terms to prevent "overshoot" was also eliminated. Hence, approximations involving second- and third-ordered terms were derived to examine what improvement could be made. In light of the tremendous number of calculations needed to evaluate these approximations and after some preliminary trials of use, it was decided that although improvement of the approximation could be made the time involved to do so was prohibitive on any problems beyond the very simple.

Guided from experience in the problems of root-finding, we chose to use a method of bisection in moving to a more advantageous starting point. We know from the nature of this problem that the initial point falls in the vicinity of a maximum f value but for some other constraint curve. Because the constraint curves have relatively similar characteristics of behavior and do not intersect, we sought a more reasonable starting point along a line connecting the initial point supplied and the origin.

Let us define

$$\Delta_0 = H(0) - (1 - \alpha) > 0 \,,$$
$$\Delta_1 = H(\overline{P}) - (1 - \alpha) < 0 \,,$$
$$\Delta_j = H(P_j) - (1 - \alpha) \text{ for } j \geq 2 \,. \qquad \text{(V-13)}$$

We consider the sequence of points which lie halfway between the previous two points producing Δ values from (V-13) of opposite signs. The sequence is described by the equations

$$P_{j+1} = P_j - \tfrac{1}{2}(P_j - P_{j-1}) \text{ for } \Delta_{j-1} > 0 \text{ , } \Delta_j < 0 \text{ ;} \qquad \text{(V-14)}$$

$$P_{j+1} = P_j + \tfrac{1}{2}(P_{j-1} - P_j) \text{ for } \Delta_{j-1} < 0 \text{ , } \Delta_j > 0 \text{ .} \qquad \text{(V-15)}$$

Each step requires that a new H function be evaluated and in our previous discussion we have seen that this can become time consuming. Hence, bisection is continued via (V-14) and (V-15) only until a point is found within a region of the constraint curve for which the more expedient approximation can be used. The criterion established for choosing a stopping point P_{j+1} is when

$$\left| \Delta_{j+1} \right| \leq \tfrac{1}{2}(1 - \alpha) \text{ .} \qquad \text{(V-16)}$$

Since the method and the equations utilized to obtain a maximum value for function under the prescribed constraints have been described, the entire procedure is reduced to algorithmic form in the following section.

VI. Algorithm for Nonlinear Function Maximization

This algorithm assumes prior construction of the index set ψ in terms of hypervolumes, as detailed in Section III, necessary for evaluation of various equations presented in Section V.

1. Determine the initial starting point $\bar{P} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n)$ by Equation (IV-2) and calculate the corresponding function value $H(\bar{P})$ as expressed in Equation (V-2).
2. From Equation (V-5) calculate the value of B which, when substituted into (IV-20) together with $H(\bar{P})$, will provide the necessary Δp shift value to yield the first point P_I on or near the constraint curve (Ref. Figure 4). Determine $H(P_I)$ by Equation (V-2).
3. Test the validity of the values $(p_1, p_2, \ldots, p_n)_I$. If $0 < p_k < 1$ for all k, k=1, ..., n, we have an acceptable point on the constraint curve. Go to step 5.

   If $p_k \leq 0$ or $p_k \geq 1$ for any k, k=1, ..., n, then the initial starting point $\bar{P}$ was too far removed from the constraint curve and we would overshoot the problem boundary conditions. A more reasonable starting point must be found. Go to step 4.
4. For the method of bisection, determine a new starting point $\bar{P}$ by using Equations (V-13, 14, 15). Repeat the calculations of steps 2-4 until the criterion of acceptance for the bisection (V-16) has been met. Determine the point P_I on or near the constraint curve as described in step 2 and, upon completion, proceed on to step 5.
5. Calculate the intersecting function value f_I. Thus, the envelope of enclosure for the next point $\hat{P}_I$ has been formed. Also determine the initial increment values δ_s and δ_p for series and parallel components, respectively. (Let the generic term δ represent either δ_s or δ_p according to the component application). Go to step 6.

6. Determine the series component to be selected to serve as the slack component for the current step cycle. If specific branch paths are dictated as a result of a detected oscillation, slack component has been fixed so go to 7. For no oscillation yet detected, go on to 6a.

(a). Check step movement. If oscillation is taking place, go to 6b. If no oscillation, go to 7.

(b). For oscillation between components, test sufficiency condition (IV-21). Go to 7 if sufficient progress. Otherwise, a junction exists and if user indicates a continuation of maximization, go to 6c. If user indicates a halt at a junction, print out diagnostic message with all pertinent information.

(c). For maximization along several branches, flag all slack components involved in the oscillation as separate branch paths to be followed. Also save junction point data for the restart of each path.

7. Define $\hat{p}_{n-1} = p_{n-1} + \delta$.

(a). If $\hat{p}_{n-1} < 1$, continue to 7b. Otherwise, $\hat{p}_{n-1} \geq 1$, which violates the boundary restrictions. In this case, go to step 8.

(b). The slack component, $\hat{p}_n$, is defined by Equation (IV-4) to maintain a position on the f_I curve. Thus, $\hat{P}_I = (p_1, \ldots, p_{n-2}, \hat{p}_{n-1}, \hat{p}_n)$. If $H(\hat{P}_I) \geq H(P_I)$, go to step 10. Otherwise, the point $\hat{P}_I$ has been moved beyond the envelope and we go to step 8.

8. Try the opposite direction. Define $\hat{p}_{n-1} = p_{n-1} - \delta$.

(a). If $\hat{p}_{n-1} > 0$, continue to 8b. Otherwise, $\hat{p}_{n-1} \leq 0$, which also violates the boundary restrictions. In this case, go to step 9.

(b). Follow the same procedure as step 7b to determine $\hat{p}_n$. If $H(\hat{P}_I) \geq H(P_I)$, go to step 10. If not, then $\hat{P}_I$ is outside the envelope and we go to step 9.

9. The component p_{n-1} could not be incremented or decremented with the currently fixed step size δ. Define $\hat{p}_{n-1} = p_{n-1}$ and $\hat{p}_n = p_n$. Proceed to step 10.

10. Looping on i, i=n-2,..., 1, modify each component p_i by δ in the manner described in steps 7 and 8 and in Equation (IV-5). Upon completion, the final point position $\hat{P}_I$ has been determined.

    (a). If any of the components, p_i, were modified by δ then calculate $H(\hat{P}_I)$ and go to 12.

    (b). If it was not possible to modify any component p_i, i=1, ..., n-1. by δ and stay within the criteria of movement, a smaller step size must be used. Go to step 11.

11. Reduce the step size, $\delta_s^1 = \delta_s/2$, $\delta_p^1 = \sqrt{\delta_s^1}$. Return to step 7.

12. As described in step 2, use Equations (IV-20) and (V-5) to calculate the necessary values to produce the next point on the constraint curve. Go on to 13.

13. Follow the same procedure as detailed in steps 6 through 10 to determine the next point $\hat{P}_j$. Provided that the criteria placed upon $\hat{P}_j$ are satisfied ($H(\hat{P}_j) \geq H(P_j)$ and boundary conditions met), return to step 12 to determine P_{j+1}. Thus begins an iterative procedure, j=1, 2, ... , M, in locating the maximum value for the function f. If it was not possible to modify any of the components of P_j without violating a part of the criteria, go to step 14.

14. The δ step size was too large for the current envelope of containment and must be reduced, $\delta_s^{k+1} = \delta_s^k/2$ and $\delta_p^{k+1} = \sqrt{\delta_s^{k+1}}$.

With some number of decimal places, d, of accuracy prescribed for the series components, check δ_s^{k+1}.

(a). If $\delta_s^{k+1} \geq 10^{-d}$, go to step 13.

(b). If $\delta_s^{k+1} < 10^{-d}$, go to step 15.

15. Convergence criterion has been met and the final point $P^* = (p_1^*, \ldots, p_n^*) = (p_1, \ldots, p_n)_M$ established so that

$$f(P^*) = \underset{j=1}{\overset{M}{\text{Max.}}}\ f(P_j)\ ;\ H(P^*) = 1-\alpha.$$

16. Test on branch paths.

(a). If no oscillation is indicated, maximization procedure has been completed as described in step 15.

(b). If oscillation is indicated, branch paths are being taken. If all branch paths are exhausted, maximization procedure is completed. Otherwise, return to step 7 with a new fixed slack component for a new path to be followed.

VII. Conclusions

The material presented in this report is not the last word in efforts to arrive at new ideas and methods of solution for the problem posed: confidence limits for system failure probability. Certainly, major progress has been made by taking a theoretical equation with seemingly unfathomable computational roadblocks into the realm of real-life application.

Methods of checking the hypervolume construction have been described in Section III. Summaries of the various systems considered in testing the methods proposed are presented in the appendices, along with system diagrams, initial parameters, resultant failure probabilities at different confidence levels, the size of related index sets, and the amount of computer time needed for solutions. These test systems not only serve as the "shake-down" of the computer code, CONLIM, but also indicate the nature of problems that can be analyzed and the directions for usage of CONLIM.

As described by Wilde [6], one of the principal indicators of preferred optimization occurs in the greatest possible interval of uncertainty to obtain the maximum value sought. Furthermore, a sequential search using previously discovered information can greatly assist in the search process of maximization. By enclosing the interval of search within the boundary envelopes described in Section IV, we actually utilize a sequential process in each successive step, and as a result, the interval of uncertainty for maximization is narrowed with each step.

Several factors appear to be critical when a system for analysis is considered. None can be considered independently, but must be viewed in conjunction with each of the other factors. All are an integral part of the system complexity. The first and most obvious factor is the size of the system and the number of unique components making up the system. Not only does the index set increase in size and scope, thus requiring a longer construction period, but the maximization procedure also must take into consideration more variables to be examined. The conditions of iteration and convergence apply to all components and thus the time for solution to the problem becomes lengthened.

Secondly, the arrangement of components can be a factor in the amount of time required for a solution. We have observed that systems containing many series-connected components require more time for solution, as opposed to systems made up of several parallel combinations.

Another factor must be considered in relation to the arrangement scheme and the dimension of the system. Parameters input to the problem include the number of failures x_i, $i=1, \ldots, n$, that were found in m_i, $i=1, \ldots, n$, tests performed on each component. In the case of a series component and, to a lesser degree, of a parallel component, the higher the failure rate the more critical that component becomes to the system. The function criterion for hypervolume acceptance enlarges as the ratio x_i/m_i increases for any ith component. This means a larger space under the surface to be filled by hypervolumes; the greater the number of hypervolumes, the greater the amount of computation to be done in evaluating the various maximization equations.

Of course the amount of time required for the computation of predicted failure probability should be considered. Only in special cases would an expected poor system even be considered for use. We must hold as the final criteria for using any technique the importance of the system and the accuracy of prediction desired. In those instances where a question of the global maximum arises as a result of branch paths, we clearly indicate the possible choices available without hesitation.

The use of hypervolume construction for the index sets and the method presented for function maximization open the door for analysis of systems ranging from the most simple composition to those of a complex, many-component structure. We cannot overemphasize the possibilities of application to the myriad of differing types of systems. Aside from the application to new systems under consideration, this approach permits a standard of comparison for previously used or proposed approximational methods of failure probability analysis.

REFERENCES

1. G. P. Steck, Upper Confidence Limits for the Failure Probability of Complex Networks, Sandia Laboratories, SC-4133(TR), December 1957.

2. H. A. Spang, III, "A Review of Minimization Techniques for Nonlinear Functions," SIAM Review, v. 4, 1962, pp. 343-365.

3. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," The Computer Journal, v. 6, 1963, pp. 163-168.

4. A. M. Ostrowski, "Contributions to the Theory of the Method of Steepest Descent," Archive for Rational Mechanics and Analysis, v. 26, 1967, pp. 257-280.

5. M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, National Bureau of Standards, AMS55, 1965, p. 257.

6. D. J. Wilde, Optimum Seeking Methods, Prentice-Hall, Inc., 1964.

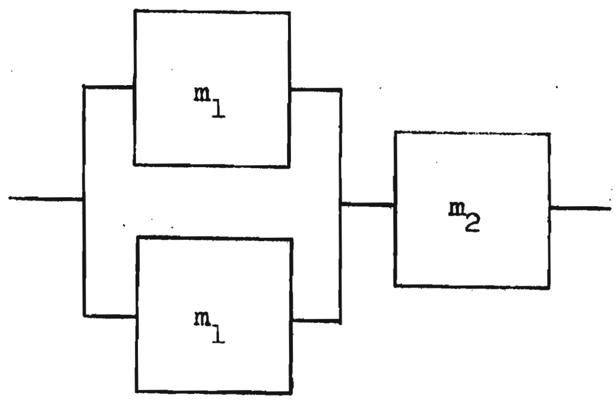
APPENDIX A

SYSTEMS UTILIZED TO STUDY AND TEST CONLIM

Diagrams for each of the test systems are given in this appendix. In many of the tests, several different input parameters are provided for comparison checks on resultant values. The probability functions, f, describing each system are also provided.

The appendices following are related to this appendix via the identification of system number and test number subordinate to each system.

SYSTEM 1 (2-D)



$$f = p_1^2 + p_2 (1 - p_1^2)$$

Test 1

$m_1 = 20$, $x_1 = 1$

$m_2 = 20$, $x_2 = 1$

Test 2

$m_1 = 40$, $x_1 = 1$

$m_2 = 40$, $x_2 = 1$

Test 3

$m_1 = 40$, $x_1 = 2$

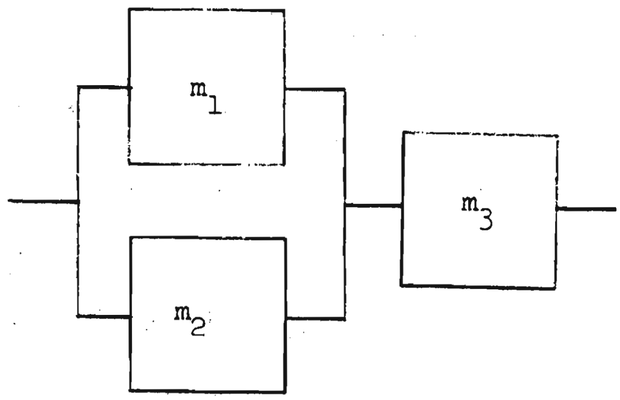
$m_2 = 40$, $x_2 = 2$

Test 4

$m_1 = 40$, $x_1 = 3$

$m_2 = 40$, $x_2 = 3$

SYSTEM 2 (3-D)



$$f = p_1p_2 + p_3\,(1 - p_1p_2)$$

Test 1

$m_1 = 20$, $x_1 = 1$

$m_2 = 20$, $x_2 = 1$

$m_3 = 20$, $x_3 = 1$

Test 2

$m_1 = 40$, $x_1 = 2$

$m_2 = 40$, $x_2 = 2$

$m_3 = 40$, $x_3 = 2$

Test 3

$m_1 = 40$, $x_1 = 4$

$m_2 = 40$, $x_2 = 9$

$m_3 = 40$, $x_3 = 1$

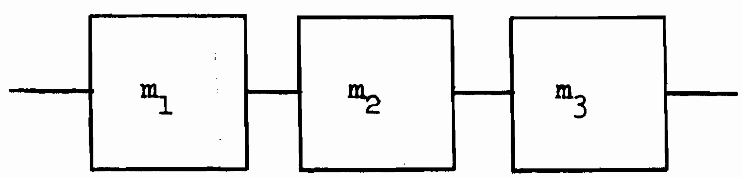
Test 4

$m_1 = 40$, $x_1 = 2$

$m_2 = 40$, $x_2 = 20$

$m_3 = 40$, $x_3 = 0$

SYSTEM 3 (3-D)



$$f = 1 - (1 - p_1)(1 - p_2)(1 - p_3)$$

$$\sim p_1 + p_2 + p_3$$

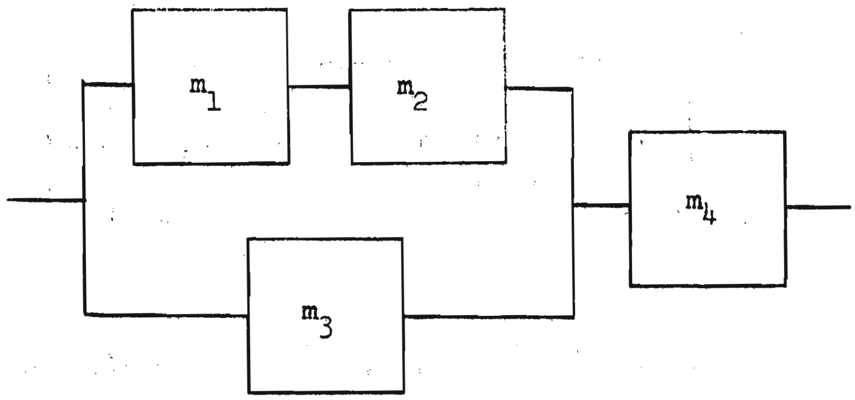
Test 1

$m_1 = 20$, $x_1 = 1$

$m_2 = 15$, $x_2 = 0$

$m_3 = 10$, $x_3 = 0$

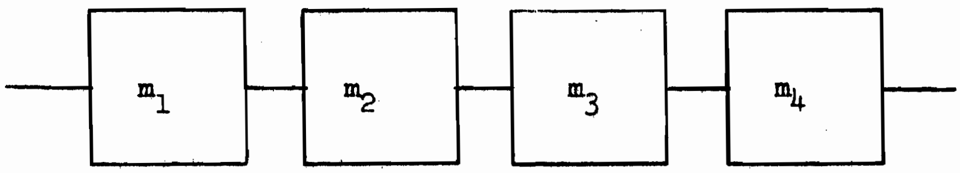
SYSTEM 4 (4-D)



$$f = 1 - (1 - p_4)\left[1 - p_3\,(p_1 + p_2 - p_1 p_2)\right]$$

Test 1	Test 2	Test 3
$m_1 = 30$, $x_1 = 1$	$m_1 = 20$, $x_1 = 1$	$m_1 = 40$, $x_1 = 2$
$m_2 = 20$, $x_2 = 1$	$m_2 = 30$, $x_2 = 1$	$m_2 = 60$, $x_2 = 2$
$m_3 = 25$, $x_3 = 1$	$m_3 = 25$, $x_3 = 1$	$m_3 = 50$, $x_3 = 2$
$m_4 = 20$, $x_4 = 1$	$m_4 = 20$, $x_4 = 1$	$m_4 = 40$, $x_4 = 2$

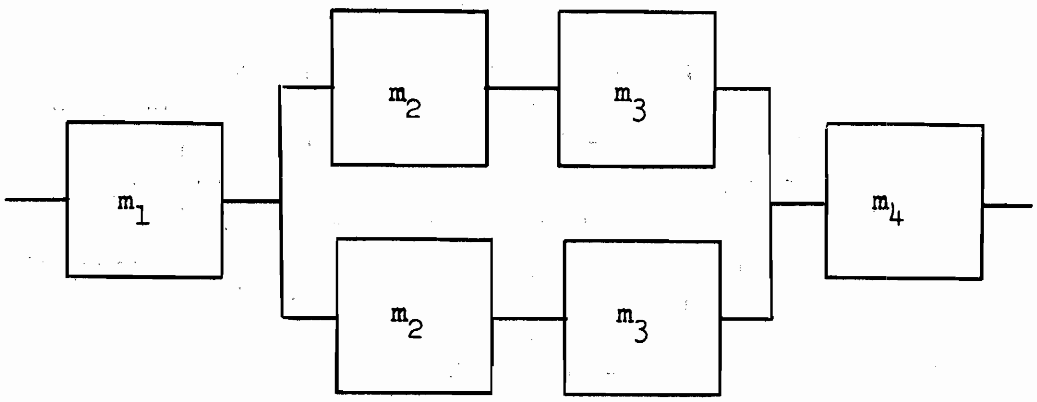
SYSTEM 5 (4-D)



$$f = 1 - \left[(1 - p_1)(1 - p_2)(1 - p_3)(1 - p_4)\right]$$

Test 1	Test 2
$m_1 = 49$, $x_1 = 0$	$m_1 = 48$, $x_1 = 5$
$m_2 = 41$, $x_2 = 1$	$m_2 = 41$, $x_2 = 1$
$m_3 = 23$, $x_3 = 0$	$m_3 = 23$, $x_3 = 0$
$m_4 = 48$, $x_4 = 5$	$m_4 = 49$, $x_4 = 0$

SYSTEM 6 (4-D)

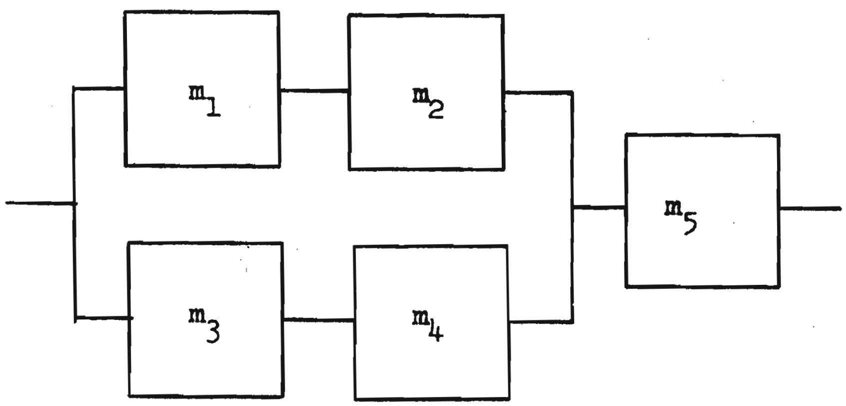


$$f = p_4 + (1 - p_4)\, f'$$

$$f' = p_1 + (1 - p_1) \left[p_2 + (1 - p_2)\, p_3 \right]^2$$

Test 1	Test 2	Test 3
$m_1 = 40$, $x_1 = 1$	$m_1 = 40$, $x_1 = 1$	$m_1 = 40$, $x_1 = 0$
$m_2 = 50$, $x_2 = 2$	$m_2 = 50$, $x_2 = 1$	$m_2 = 50$, $x_2 = 0$
$m_3 = 50$, $x_3 = 1$	$m_3 = 50$, $x_3 = 2$	$m_3 = 50$, $x_3 = 0$
$m_4 = 20$, $x_4 = 0$	$m_4 = 20$, $x_4 = 0$	$m_4 = 20$, $x_4 = 0$

SYSTEM 7 (5-D)



$$f = p_5 + (1 - p_5)\left[p_1 + (1 - p_1)\,p_2\right]\left[p_3 + (1 - p_3)\,p_4\right]$$

<u>Test 1</u>

$m_1 = 30\ ,\ x_1 = 2$

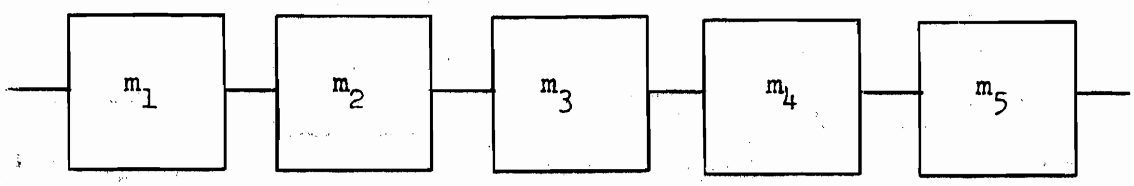
$m_2 = 20\ ,\ x_2 = 1$

$m_3 = 40\ ,\ x_3 = 2$

$m_4 = 30\ ,\ x_4 = 1$

$m_5 = 20\ ,\ x_5 = 1$

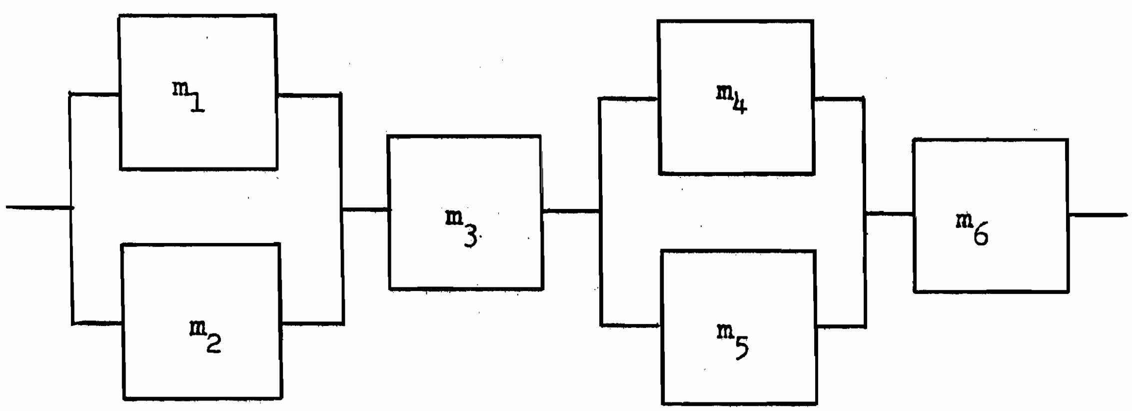
SYSTEM 8 (5-D)



$$f = 1 - (1 - p_1)(1 - p_2)(1 - p_3)(1 - p_4)(1 - p_5)$$

Test 1	Test 2
$m_1 = 50 \ , \ x_1 = 0$	$m_1 = 50 \ , \ x_1 = 1$
$m_2 = 50 \ , \ x_2 = 0$	$m_2 = 50 \ , \ x_2 = 0$
$m_3 = 50 \ , \ x_3 = 0$	$m_3 = 50 \ , \ x_3 = 0$
$m_4 = 50 \ , \ x_4 = 0$	$m_4 = 50 \ , \ x_4 = 0$
$m_5 = 50 \ , \ x_5 = 1$	$m_5 = 50 \ , \ x_5 = 0$

SYSTEM 9 (6-D)



$$f = 1 - \left[(1 - p_1p_2)(1 - p_4p_5)(1 - p_3)(1 - p_6)\right]$$

<u>Test 1</u>

$m_1 = 20 \;,\; x_1 = 1$

$m_2 = 25 \;,\; x_2 = 0$

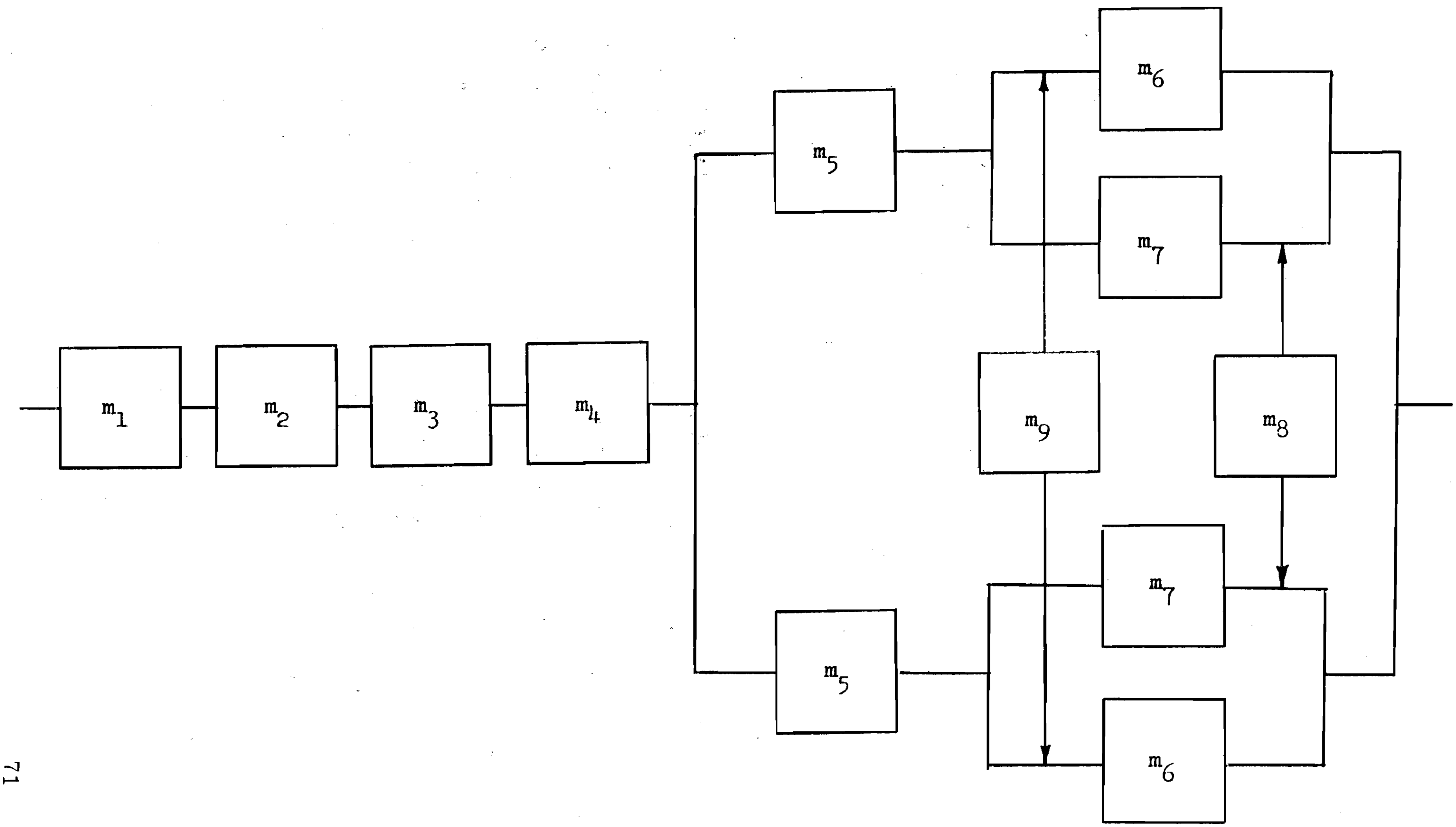
$m_3 = 30 \;,\; x_3 = 1$

$m_4 = 25 \;,\; x_4 = 1$

$m_5 = 20 \;,\; x_5 = 1$

$m_6 = 40 \;,\; x_6 = 0$

SYSTEM 10 (9-D)



SYSTEM 10
(CONTINUED)

$$f = 1 - \prod_{i=1}^{4}(1 - p_i)\left[(1 - p_9)(1 - p_8)\left\{1 - \left(1 - \left[1 - p_5\right]\left[1 - p_6 p_7\right]\right)^2\right\}\right.$$

$$+ p_9 (1 - p_8)\left\{1 - \left(p_5 + \left[1 - p_5\right] p_7\right)^2\right\}$$

$$\left. + p_8 (1 - p_9)\left\{1 - \left(p_5 + \left[1 - p_5\right] p_6\right)^2\right\}\right]$$

Test 1	Test 2	Test 3	Test 4
$m_1 = 44$, $x_1 = 0$	$m_1 = 44$, $x_1 = 0$	$m_1 = 44$, $x_1 = 0$	$m_1 = 44$, $x_1 = 0$
$m_2 = 54$, $x_2 = 0$	$m_2 = 54$, $x_2 = 0$	$m_2 = 54$, $x_2 = 0$	$m_2 = 54$, $x_2 = 1$
$m_3 = 30$, $x_3 = 0$	$m_3 = 30$, $x_3 = 0$	$m_3 = 30$, $x_3 = 0$	$m_3 = 30$, $x_3 = 0$
$m_4 = 101$, $x_4 = 0$	$m_4 = 101$, $x_4 = 1$	$m_4 = 101$, $x_4 = 0$	$m_4 = 101$, $x_4 = 0$
$m_5 = 32$, $x_5 = 1$	$m_5 = 32$, $x_5 = 1$	$m_5 = 32$, $x_5 = 2$	$m_5 = 32$, $x_5 = 1$
$m_6 = 23$, $x_6 = 1$	$m_6 = 23$, $x_6 = 1$	$m_6 = 23$, $x_6 = 3$	$m_6 = 23$, $x_6 = 1$
$m_7 = 32$, $x_7 = 1$	$m_7 = 32$, $x_7 = 1$	$m_7 = 32$, $x_7 = 2$	$m_7 = 32$, $x_7 = 1$
$m_8 = 43$, $x_8 = 1$	$m_8 = 43$, $x_8 = 1$	$m_8 = 43$, $x_8 = 3$	$m_8 = 43$, $x_8 = 1$
$m_9 = 17$, $x_9 = 0$	$m_9 = 17$, $x_9 = 0$	$m_9 = 17$, $x_9 = 0$	$m_9 = 17$, $x_9 = 0$

APPENDIX B

INDEX SET COMPOSITION AND CONSTRUCTION TIME OF CONLIM TEST SYSTEMS

Index Set Composition and Construction Time

System	Test	Index Set Ψ		
		No. of HV	No. of n-tuples	time (secs.)
1	1	2	7	0.001
	2	2	8	0.001
	3	3	19	0.002
	4	4	32	0.003
2	1	12	89	0.009
	2	34	543	0.041
	3	33	524	0.040
	4	22	295	0.023
3	1	1	2	0.001
4	1	65	1243	0.107
	2	69	1243	0.115
	3	414	15404	2.42
5	1	39	109	0.045
	2	39	109	0.043
6	1	10	44	0.010
	2	10	44	0.009
	3	1	1	0.001
7	1	583	6061	6.88
8	1	5	6	0.004
	2	5	6	0.004
9	1	373	2848	2.52
10	1	44	183	0.38
	2	433	2451	11.5
	3	675	3726	23.2
	4	1665	8912	119.3

APPENDIX C

CONLIM RESULTS OF SYSTEM TESTS

System	Test	α=0.80		α=0.90		α=0.95		α=0.98		α=0.99	
		f value	time	f value	time	f value	time	f value	time	f value	time
1	1	0.14226	0.02	0.18095	0.03	0.21606	0.03	0.25878	0.03	0.28820	0.03
	2	0.07288	0.02	0.09378	0.03	0.11311	0.03	0.13733	0.03	0.15104	0.02
	3	0.10382	0.05	0.12763	0.06	0.14915	0.07	0.17540	0.06	0.19401	0.06
	4	0.13372	0.12	0.15960	0.11	0.18273	0.12	0.21054	0.15	0.23004	0.14
2	1	0.14243	0.33	0.18095	0.34	0.21602	0.34	0.25177	0.32	0.28877	0.39
	2	0.10402	1.68	0.12769	1.82	0.14921	1.95	0.17546	2.36	0.19406	2.18
	3	0.10397	2.36	0.12764	1.97	0.14916	2.15	0.17541	2.20	0.19402	2.36
	4	0.07427	2.11	0.09503	2.09	0.11439	2.12	0.13851	2.31	0.15588	2.33
3	1	0.17026	0.03	0.22809	0.03	0.26451	0.03	0.33026	0.03	0.37584	0.03
4	1	0.14243	3.80	0.18096	6.47	0.21611	6.27	0.25879	4.66	0.28879	4.84
	2	0.14243	3.64	0.18096	6.41	0.21611	6.26	0.25879	4.27	0.28879	4.66
	3	0.10412	63.5	0.12779	44.1	0.14931	60.7	0.17555	61.7	0.19416	63.2
5	1	0.19772	6.32	0.23480	6.26	0.26793	6.77	0.30750	6.92	0.33500	8.18
	2	0.19772	5.79	0.23480	5.97	0.26793	6.32	0.30750	6.50	0.33500	7.84
6	1	0.08120	1.37	0.11760	2.58	0.14272	0.94	0.18111	1.01	0.21357	1.45
	2	0.08120	1.40	0.11760	2.56	0.14272	1.08	0.18111	1.06	0.21357	1.46
	3	0.07419	0.07	0.10719	0.08	0.13668	0.08	0.17280	0.07	0.20488	0.09
7	1	0.14316	89.9	0.18165	87.7	0.21677	81.5	0.25941	78.5	0.28939	72.7
8	1	0.05870	0.83	0.07558	0.79	0.09140	0.93	0.11119	0.88	0.12552	0.90
	2	0.05870	0.83	0.07558	0.79	0.09140	0.92	0.11119	0.87	0.12552	0.90
9	1	0.09655	64.3	0.12357	63.1	0.14860	64.4	0.17950	64.4	0.20159	81.4
10	1	0.05221	12.1	0.07388	13.1	0.09503	13.4	0.12226	14.8	0.14230	13.9
	2	0.05400	197.5	0.07561	236.9	0.09672	210.0	0.12390	194.8	0.14391	224.3
	3	0.05418	348.6	0.07578	365.1	0.09689	399.1	0.12407	639.4	0.14407	728.5
	4	0.05972	1789								

NOTE: All times shown are in seconds

APPENDIX D

EXAMPLES OF INPUT DATA AND CONTROL CARDS FOR CONLIM

The specific data input structure for CONLIM is detailed on the first two pages of Appendix F in the computer code listing of CONLIM. Each data card is described, and the particular format arrangement for each card is also provided. Users of CONLIM are directed to those two pages for references, as well as to the sample decks listed in this appendix.

We note that for systems of dimension greater than 8, the user must perform a minor modification to CONLIM. Two cards, CONLIM 117 and CONLIM 118, must be replaced since the limiting dimension automatically accommodated by CONLIM is set at 8. The reason for this dimension value is only to maintain CONLIM at a computer memory core requirement less than $100{,}000_8$ words and in no way reflects upon the capabilities of CONLIM. The card deck structure from one of the sample tests described in the preceding appendices (System 10) is presented in this appendix to demonstrate the necessary modifications.

In addition to the numerical input data, the user also has the responsibility of supplying the function QTHETA (PROB), which describes the reliability function of the system under study. PROB is an array which must be dimensioned N, the size of the system. Examples of such functions are included on the following pages. Both QTHETA and PROB are probabilities in terms of system failure and component failure, respectively.

```
SYSTEM1,T---,CM100000,EC---.                    IDENTIFICATION
ACCOUNT,S---------,D----,G----,A--------,R-,K---.
ATTACH,CONTAP,CONLIM.
COPYBF,CONTAP,LGO.
FTN,B=LGO1,L,OPT=1.
REWIND,LGO1.
PREP,LGO1,QTHE.
COLLFCT,LGO,FTNLIB,QTHE.
LGO.
7/8/9
      FUNCTION QTHETA(PROB)
C
C     PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE
C
      DIMENSION PROB(2)
      A = PROB(1)*PROB(1)
      QTHETA = A + PROB(2)*(1.0-A)
      RETURN
      END
7/8/9
```

In this example, the program CONLIM resides in a permanent file in binary form. The function deck is compiled and collected to the program for execution.

```
7/8/9   <------------------------------blank option:  standard output

    PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE (EXAMPLE 1)
  2
          20          20
           1           1
           1           0
0.8
0.9
0.95
0.98
0.99
0.999
7/8/9   <------------------------------option 1:  intermediate printout
1
    PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE (EXAMPLE 2)
  2
          40          40
           1           1
           1           0
0.8
0.9
0.95
0.98
0.99
0.999
7/8/9   <------------------------------option 3:  list hypervolumes
 1
    PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE (EXAMPLE 3)
  2
          40          40
           2           2
           1           0
0.8
0.9
0.95
0.98
0.99
0.999
7/8/9
6/7/8/9
```

NOTE: 7/8/9 represents standard CDC 6600 EOF card

6/7/8/9 represents standard CDC 6600 EOI card

```
SYSTEM7,T---,CM100000,EC---,MT1.         IDENTIFICATION
ACCOUNT,S---------,D----,G----,A-------,R-,K---.
REQUEST,TAPE3,HI.            VRN =
REWIND,TAPE3.
ATTACH,CONTAP,CONLIM.
COPYBF,CONTAP,LGO.
FTN,B=LGO1,L,OPT=1.
REWIND,LGO1.
PREP,LGO1,QTHE.
COLLECT,LGO,FTNLIB,QTHE.
LGO.
UNLOAD,TAPE3.
EXIT.
UNLOAD,TAPE3.
7/8/9
      FUNCTION QTHETA(PROB)
      DIMENSION PROB(5)
      P1 = (1.0-PROB(1))*(1.0-PROB(2))
      P2 = (1.0-PROB(3))*(1.0-PROB(4))
      QTHETA = PROB(5) + (1.0-PROB(5))*(1.0-P1)*(1.0-P2)
      RETURN
      END
7/8/9
  1
        PARALLEL SYSTEM-FIVE COMPONENTS
 5
      30        20        40        30        20
       2         1         2         1         1
       1         1         1         1         0
0.8
0.9
0.95
0.98
0.99
0.999
7/8/9
6/7/8/9
```

CONLIM is maintained in a permanent file. In this example, a magnetic tape is requested to save the hypervolume structure for subsequent runs. Note that permanent files, rather than tape, can also be used for this purpose. The manual detailing the use of permanent files should be consulted for creating/cataloging a permanent file.

Option 4=1; Save hypervolume structure (more than one set can be submitted and more than one structure saved)

NOTE: 7/8/9 represents standard CDC 6600 EOF card

6/7/8/9 represents standard CDC 6600 EOI card

```
SYSTM10,T---,CM120000,EC---.                 IDENTIFICATION
ACCOUNT,S---------,D----,G----,A-------,R-,K---.
ATTACH,OLDPL,CONLIM,CY=1.
UPDATE,P=OLDPL,F.
REWIND,COMPILE.
FTN,I=COMPILE,L,OPT=1.
COLLECT,LGO,FTNLIB.
LGO.
7/8/9
```

The dimensions of the program CONLIM must be expanded for this run and two cards of CONLIM are modified. (Note: increase in core memory required) This information and the function deck are submitted via the UPDATE system of the CDC 6600. UPDATE source of CONLIM is taken from cycle 1 of the permanent file as shown.

```
*IDENT   CHANL
*DELETE  CONLIM.116,CONLIM.117
      DIMENSION NFMAX(9,500),NFMIN(9,500),NFMAXX(9,500),NFMINX(9,500)
      DATA  IVSIZE/500/,IVCOL/9/
*INSERT  CONLIM.1600
      FUNCTION QTHETA(PROB)
C
C           TWO CHANNEL - TWO OPTION SYSTEM
C
      DIMENSION PROB(9)
      PART1 = (1.0-PROB(1))*(1.0-PROB(2))*(1.0-PROB(3))
      P8 = 1.0-PROB(8)
      P7 = 1.0-PROB(7)
      P4 = 1.0-PROB(4)
      P56 = (1.0 - PROB(5)*PROB(6))
      P456 = P4*P56
      PART2 = 1.0 - (1.0 - P456)*(1.0 - P456)
      PART3 = P8*P7*PART2
      PART4 = 1.0 - (PROB(4) + P4*PROB(6))**2
      PART5 = PROB(8)*P7*PART4
      PART6 = 1.0 - (PROB(4) + P4*PROB(5))**2
      PART7 = PROB(7)*P8*PART6
      PARTS = PART3 + PART5 + PART7
      THETA = 1.0 - PART1*PARTS
      QTHETA = PROB(9) + (1.0-PROB(9))*THETA
      RETURN
      END
7/8/9
```

*IDENT identifies this modification to UPDATE

*DELETE deletes two cards and replaces them with the cards following

*INSERT places the cards (function deck) immediately following the CONLIM routine proper

← blank option: standard output

```
            TWO CHANNEL - TWO OPTION SYSTEM
  9
      54        30        101       32        23        32        43        17
      44
       1         0          0        1         1         1         1         0
       0
       0         0          0        1         1         1         1         1
       0
0.8
0.90
0.95
0.98
0.99
0.999
7/8/9
6/7/8/9
```

NOTES: The data cards containing the number of tests, failures, and classification are each continued to second cards.

7/8/9 represents standard CDC 6600 EOF card.

6/7/8/9 represents standard CDC 6600 EOI card.

APPENDIX E

EXAMPLES OF OUTPUT RESULTS
USING VARIOUS CONLIM OPTIONS

Each of the input data options detailed on the first two pages of the CONLIM listing in Appendix F results in various forms of output and information. This appendix demonstrates the multitude of output forms resulting from these option selections by the user. Should CONLIM prematurely stop as a result of exceeding the designated time limit for the computer run, a brief printout of pertinent information is supplied from CONLIM prior to the run exit.

TIME REQUIRED TO DETERMINE INDEX SET AND HYPERVOLUMES =   .001 SECONDS

HYPERVOLUME CROSSCHECK BYPASSED

TIME REQUIRED FOR FIT PROCEDURE =   .019 SECONDS

(Standard first page of results which appears at the start of each new set. For multiple α values only the time for the fit procedure is supplied after the first since the hypervolumes will have been determined previously.)

PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE

SYSTEM COMPONENTS
NUMBER OF TESTS AND FAILURES

1	2
20	20
1	1

ALPHA UPPER CONFIDENCE LIMIT = .8000

SYSTEM FAILURE PROBABILITY Q = .1422641E+00

NUMBER OF HYPERVOLUMES IN INDEX SET = 2
TOTAL NUMBER OF NTUPLES IN SET = 7

INDIVIDUAL COMPONENT P VALUES ARE AS FOLLOWS

I = 1 P(I) = .7096878E-04
I = 2 P(I) = .1422641E+00

(Standard output of results - for the standard (blank option card) run this will be the only page printed in addition to the timing pages.)

(Option 1 selected - intermediate output)

```
I =   1      PHAT(I) =     .4761905E-01
I =   2      PHAT(I) =     .4761905E-01
HVALUE STARTING OFF DELTAP =     .26153023E+00
I =  1  PHAT(I) =      .5284574E-01  QHAT(I) =     .9471543E+00
I =  2  PHAT(I) =      .5284574E-01  QHAT(I) =     .9471543E+00
HVALUE AFTER DELTAP SHIFT =     .205653418552E+00
QP VALUE AFTER DELTAP SHIFT =     .554908261854E-01
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .5284574E-02
I =  1  PHAT(I) =      .5284574E-01  QHAT(I) =     .9471543E+00
I =  2  PHAT(I) =      .5284574E-01  QHAT(I) =     .9471543E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .2642287E-02
I =  1  PHAT(I) =      .1442557E-02  QHAT(I) =     .9985574E+00
I =  2  PHAT(I) =      .5548886E-01  QHAT(I) =     .9445111E+00
HVALUE STARTING OFF DELTAP =     .34107094E+00
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
HVALUE AFTER DELTAP SHIFT =     .221418001683E+00
QP VALUE AFTER DELTAP SHIFT =     .697492598601E-01
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .2642287E-02
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .1321143E-02
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .6605717E-03
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .3302858E-03
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .1651429E-03
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .8257146E-04
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .4128573E-04
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .2064287E-04
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .1032143E-04
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =      .5160716E-05
I =  1  PHAT(I) =      .1813208E-02  QHAT(I) =     .9981868E+00
I =  2  PHAT(I) =      .6974620E-01  QHAT(I) =     .9302538E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
```

```
DEL VALUE TRIED =     .2580358E-05
I =  1  PHAT(I) =     .2068588E-03  QHAT(I) =     .9997931E+00
I =  2  PHAT(I) =     .6974922E-01  QHAT(I) =     .9302508E+00
HVALUE STARTING OFF DELTAP =    .22179892E+00
I =  1  PHAT(I) =     .2161293E-03  QHAT(I) =     .9997839E+00
I =  2  PHAT(I) =     .7287508E-01  QHAT(I) =     .9271249E+00
HVALUE AFTER DELTAP SHIFT =     .200887018676E+00
QP VALUE AFTER DELTAP SHIFT =     .728751201759E-01
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =     .2580358E-05
I =  1  PHAT(I) =     .2161293E-03  QHAT(I) =     .9997839E+00
I =  2  PHAT(I) =     .7287508E-01  QHAT(I) =     .9271249E+00
     COMPONENT   2 USED TO MAINTAIN CONSTANT Q
DEL VALUE TRIED =     .1290179E-05
I =  1  PHAT(I) =     .2161293E-03  QHAT(I) =     .9997839E+00
I =  2  PHAT(I) =     .7287508E-01  QHAT(I) =     .9271249E+00
               TIME REQUIRED FOR FIT PROCEDURE =      .114 SECONDS
```

(Option 1 printout continued with the time requirement printed after the intermediate output.)

(Option 2 selected - determine hypervolumes only)

PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE

SYSTEM COMPONENTS
NUMBER OF TESTS AND FAILURES

1	2
40	40
1	1

ALPHA UPPER CONFIDENCE LIMIT = .8000

HYPERVOLUME ONLY OPTION SELECTED

INDEX SET DETERMINED VIA OPTION 2, ALPHA = .9000 BYPASSED

INDEX SET DETERMINED VIA OPTION 2, ALPHA = .9500 BYPASSED

(The option is indicated and since the fit procedure was to be bypassed, so also are any other α values in the set. Hypervolumes need only be calculated once for the set.)

(Option 3 selected - list hypervolume structure)

```
                         HYPERVOLUME STRUCTURE WITHIN INDEX SET PSI

                         NUMBER OF HYPERVOLUMES IN INDEX SET =      2
                           TOTAL NUMBER OF NTUPLES IN SET =       8


***VOLUME     1

          J =    1  MIN(J) =     0  MAX(J) =     5
          J =    2  MIN(J) =     0  MAX(J) =     0

***VOLUME     2

          J =    1  MIN(J) =     0  MAX(J) =     1
          J =    2  MIN(J) =     1  MAX(J) =     1
```

(The structure is listed prior to any other action - standard output and/or intermediate output would follow.)

(Options 2 and 3 selected - determine hypervolumes and list)

PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE

SYSTEM COMPONENTS
NUMBER OF TESTS AND FAILURES

1	2
40	40
1	1

ALPHA UPPER CONFIDENCE LIMIT = .8000

HYPERVOLUME ONLY OPTION SELECTED

INDEX SET COMPOSITION PREVIOUSLY LISTED VIA OPTION 3, ALPHA = .9000 BYPASSED

INDEX SET COMPOSITION PREVIOUSLY LISTED VIA OPTION 3, ALPHA = .9500 BYPASSED

(All hypervolumes would be printed as shown on the previous page. However, processing of the set will stop without performing any further calculations. Multiple α values in the set are bypassed.)

(Option 4 = 1 selected - save index set for subsequent runs)

```
TIME REQUIRED TO DETERMINE INDEX SET AND HYPERVOLUMES =       .001 SECONDS

HYPERVOLUME STRUCTURE SAVED ON FILE 1 OF TAPE3

HYPERVOLUME CROSSCHECK BYPASSED

TIME REQUIRED FOR FIT PROCEDURE =      .020 SECONDS
```

(This information is supplied along with any other options selected. If only option 4 is selected CONLIM will complete the fit procedure and supply a standard report. Other options selected will produce additional output as described in this appendix.)

(Option 4 = 2 selected - read hypervolumes from tape)

```
INDEX SET AND HYPERVOLUMES INPUT FROM TAPE4

HYPERVOLUME CROSSCHECK BYPASSED

TIME REQUIRED FOR FIT PROCEDURE =      .119 SECONDS
```

(This message is supplied if the system described on tape matches the system described via the data set. Notice that no time is given for construction of the index set since it is taken from tape.)

(Option 4 = 2 Continued)

```
PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE

                    SYSTEM COMPONENTS
                 NUMBER OF TESTS AND FAILURES

                          1   2
                         40  40
                          3   3

          ALPHA UPPER CONFIDENCE LIMIT =      .8000

          SYSTEM FAILURE PROBABILITY Q =  .1337244E+00

          NUMBER OF HYPERVOLUMES IN INDEX SET =      4
          TOTAL NUMBER OF NTUPLES IN SET =     32

          INDIVIDUAL COMPONENT P VALUES ARE AS FOLLOWS

              I =  1      P(I) =  .2020070E-01
              I =  2      P(I) =  .1333707E+00
```

(This report can be compared to the situation when the systems did not agree.)

(Option 4 = 2 selected - read hypervolume structure from tape)

PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE

SYSTEM COMPONENTS
NUMBER OF TESTS AND FAILURES

1	2
20	20
1	1

DATA DESCRIBED AS INPUT TO CONLIM DID NOT MATCH
REQUESTED DATA FROM TAPE4 - SET BYPASSED

INFORMATION FROM TAPE4 LISTED BELOW

I = 1	N(I) = 40	NX(I) = 3	NTYPE(I) = 1
I = 2	N(I) = 40	NX(I) = 3	NTYPE(I) = 0

(In this situation, the information supplied via the data set did not agree with that on tape and thus would not have produced the proper hypervolume structure. A diagnostic message is supplied, multiple α values in the set bypassed, and the next set is processed.)

(Option 5 selected - check the hypervolumes criteria)

COMPARISON CHECKS ON Q VALUES FOR MINMAX LIMITS

Q VALUE USED FOR INDEX SET CRITERIA, QLIM = .761661807580E-01

KVOL	QMIN	QLIM-QMIN	QMAX	QLIM-QMAX
1	.243629197711E-01	.518E-01	.686345966958E-01	.753E-02
2	.481589461181E-01	.280E-01	.740740740741E-01	.209E-02
3	.719549724652E-01	.421E-02	.761661807580E-01	0.

HYPERVOLUME CHECK INDICATES GOOD SET

(To be included in the index set all n-tuples must satisfy the set criteria, and hence the upper and lower bounds of hypervolumes must also meet that criteria. This crosscheck indicates if this condition exists. Note that we must always have)

QLIM-QMIN ≥ 0
QLIM-QMAX ≥ 0

(Option 5 selected - continued)

TIME REQUIRED TO DETERMINE INDEX SET AND HYPERVOLUMES = .002 SECONDS

TIME REQUIRED TO PERFORM HYPERVOLUME CROSSCHECK = .014 SECONDS

TIME REQUIRED FOR FIT PROCEDURE = .045 SECONDS

(Timing for the crosscheck is also supplied when this option is selected.)

(Option 6 = 0 selected - do not take branches if encountered)

```
SERIES SYSTEM GIVING BRANCHES FOR OPTION 6

                           SYSTEM COMPONENTS
                      NUMBER OF TESTS AND FAILURES

                                 1    2    3
                                20   15   10
                                 1    0    0

   ALPHA UPPER CONFIDENCE LIMIT =      .6000

   NUMBER OF HYPERVOLUMES IN INDEX SET =      1
   TOTAL NUMBER OF NTUPLES IN SET =      2

*************************************************
*                                               *
*  OSCILLATION BETWEEN COMPONENT P VALUES       *
*  ENCOUNTERED - EXECUTION FOR THIS ALPHA VALUE *
*  IS TERMINATED                                *
*  BELOW ARE LISTED THE INDIVIDUAL P VALUES UPON *
*  TERMINATION WITH OSCILLATING COMPONENTS MARKED *
*         I =  1     P(I) =  .5158699E-01 ***   *
*         I =  2     P(I) =  .1522537E-03       *
*         I =  3     P(I) =  .5737241E-01 ***   *
*                                               *
*************************************************
```

(Should the situation arise when two or more components oscillate without advancing significantly toward a maximum and option 6 is not set, a message is supplied to the user. Further processing on this α is halted. CONLIM will proceed to perform calculations for additional α values and sets.)

(Option 6 = 1 selected - take individual branches upon oscillation)

SERIES SYSTEM GIVING BRANCHES FOR OPTION 6

SYSTEM COMPONENTS
NUMBER OF TESTS AND FAILURES

1	2	3
20	15	10
1	0	0

ALPHA UPPER CONFIDENCE LIMIT = .6000

SYSTEM FAILURE PROBABILITY Q = .1089992E+00

NUMBER OF HYPERVOLUMES IN INDEX SET = 1
TOTAL NUMBER OF NTUPLES IN SET = 2

INDIVIDUAL COMPONENT P VALUES ARE AS FOLLOWS

I = 1 P(I) = .4735771E-01
I = 2 P(I) = .1189675E-05
I = 3 P(I) = .6164032E-01

RESULTS SHOWN ABOVE WERE OBTAINED BY HOLDING
COMPONENT 1 FIXED AS THE SLACK COMPONENT VIA OPTION 6

(If option 6 is selected and oscillation occurs, CONLIM will hold each of the oscillating components fixed and proceed to finish the calculations. We previously noted that components 1 and 3 were oscillating. The next page shows 3 being held fixed.)

(Option 6 = 1 continued)

SERIES SYSTEM GIVING BRANCHES FOR OPTION 6

SYSTEM COMPONENTS
NUMBER OF TESTS AND FAILURES

1	2	3
20	15	10
1	0	0

ALPHA UPPER CONFIDENCE LIMIT = .6000

SYSTEM FAILURE PROBABILITY Q = .1089217E+00

NUMBER OF HYPERVOLUMES IN INDEX SET = 1
TOTAL NUMBER OF NTUPLES IN SET = 2

INDIVIDUAL COMPONENT P VALUES ARE AS FOLLOWS

I = 1 P(I) = .5984061E-01
I = 2 P(I) = .8799240E-06
I = 3 P(I) = .4908024E-01

RESULTS SHOWN ABOVE WERE OBTAINED BY HOLDING
COMPONENT 3 FIXED AS THE SLACK COMPONENT VIA OPTION 6

(As on the preceding page, component 3 is now fixed. CONLIM saves all information at the point of the branch and proceeds from there without retracing the early portion of the path.)

END OF ANALYSIS - PARALLEL SYSTEM - LIKE COMPONENTS - TREATED AS 2-D CASE

(Standard last page of a data set. The title used for the set is repeated at the end of the set analysis.)

APPENDIX F

LISTING OF CONLIM

```
      PROGRAM CONLIM(INPUT,OUTPUT,TAPE3,TAPE4,TAPE5=INPUT)               CONLIM
C                                                                        CONLIM
C     THIS ROUTINE PERFORMS THE CALCULATIONS NECESSARY TO ESTABLISH AN   CONLIM
C     UPPER CONFIDENCE LIMIT FOR SYSTEM FAILURE PROBABILITY.             CONLIM
C                                                                        CONLIM
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CONLIM
C   *                ISSUED BY SANDIA LABORATORIES,                      CONLIM
C   *                  A PRIME CONTRACTOR TO THE                         CONLIM
C   *             UNITED STATES ATOMIC ENERGY COMMISSION                 CONLIM
C   * * * * * * * * * * * * * *  NOTICE   * * * * * * * * * * * * * * *  CONLIM
C   * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE    CONLIM
C   * UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE       CONLIM
C   * UNITED STATES ATOMIC ENERGY COMMISSION, NOR ANY OF THEIR           CONLIM
C   * EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR  CONLIM
C   * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY  CONLIM
C   * LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS   CONLIM
C   * OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT OR PROCESS    CONLIM
C   * DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE           CONLIM
C   * PRIVATELY OWNED RIGHTS.                                            CONLIM
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CONLIM
C   * THE BASIC REFERENCE DOCUMENT FOR THIS CODE IS SLA-73-0563,         CONLIM
C   * SEPTEMBER 1973.                                                    CONLIM
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CONLIM
C   * THIS CODE HAS BEEN APPROVED FOR PUBLIC RELEASE WITHIN THE          CONLIM
C   * UNITED STATES.  NO FOREIGN DISSEMINATION IS PERMITTED WITHOUT      CONLIM
C   * SPECIFIC APPROVAL FROM THE U.S. ATOMIC ENERGY COMMISSION.          CONLIM
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CONLIM
C     WRITTEN BY    RONALD D. HALBGEWACHS                                CONLIM
C                   SYSTEMS SOFTWARE DIVISION 2641                       CONLIM
C                   SANDIA LABORATORIES                                  CONLIM
C                                                                        CONLIM
C                                                                        CONLIM
C     RELEASE DATE   MAY,1973.                                           CONLIM
C                                                                        CONLIM
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CONLIM
C                                                                        CONLIM
C     INPUT DATA FORMAT                                                  CONLIM
C                                                                        CONLIM
C         CARD 1  - OPTION INDICATORS                                    CONLIM
C                                                                        CONLIM
C             BLANK CARD INDICATES NORMAL EXECUTION WITH STANDARD        CONLIM
C                   OUTPUT RETURNED TO THE USER                          CONLIM
C                                                                        CONLIM
C             NON-BLANK CARD INDICATES SPECIAL OPTIONS TO BE TAKEN       CONLIM
C                   AND SPECIAL OUTPUT IN ADDITION TO STANDARD OUTPUT    CONLIM
C                                                                        CONLIM
C                   COL. 1 = 1, PRINT INTERMEDIATE STEP VALUES OF        CONLIM
C                               P TERMS, H VALUES, AND F VALUES FOR      CONLIM
C                               EACH STEP IN THE FIT PROCESS             CONLIM
C                                                                        CONLIM
C                   COL. 2 = 1, DETERMINE HYPERVOLUMES ONLY, DO NOT      CONLIM
C                               CALCULATE FAILURE PROBABILITY            CONLIM
C                                                                        CONLIM
C                   COL. 3 = 1, LIST COMPLETE SET OF HYPERVOLUMES        CONLIM
```

```
C                                                                   CONLIM
C                   COL. 4 = 1, SAVE HYPERVOLUME STRUCTURE ON TAPE3    CONLIM
C                                FOR LATER RUNS - IF MORE THAN ONE     CONLIM
C                                INDEX SET TO BE SAVED (SEVERAL PROBLEMS CONLIM
C                                IN THE DATA SET) SEPARATE FILES WILL  CONLIM
C                                BE SAVED                              CONLIM
C                          = 2, HYPERVOLUMES PREVIOUSLY SAVED ON       CONLIM
C                                MAGNETIC TAPE - READ FROM TAPE4       CONLIM
C                                (TAPE4 MUST BE POSITIONED PRIOR TO    CONLIM
C                                 EXECUTION)                           CONLIM
C                                                                   CONLIM
C                   COL. 5 = 1, PERFORM CROSSCHECK ON HYPERVOLUMES     CONLIM
C                                FOR MATCHING INDEX SET CRITERIA       CONLIM
C                                                                   CONLIM
C                   COL. 6 = 1, IF COMPONENT OSCILLATION OCCURS,       CONLIM
C                                TAKE EACH OF THE SEPARATE BRANCHES    CONLIM
C                                BEFORE CONTINUING                     CONLIM
C                                                                   CONLIM
C         CARD 2  - TITLE OF SYSTEM UNDER STUDY  (8A10)                CONLIM
C                                                                   CONLIM
C         CARD 3  - NUMBER OF INDIVIDUAL COMPONENTS IN SYSTEM  (I3)    CONLIM
C                                                                   CONLIM
C         CARD 4  - NUMBER OF TESTS PERFORMED ON EACH COMPONENT  (8I10) CONLIM
C                   (NOTE - DATA CAN BE CONTINUED TO AS MANY CARDS     CONLIM
C                    AS NECESSARY)                                     CONLIM
C                                                                   CONLIM
C         CARD 5  - NUMBER OF FAILURES ON EACH COMPONENT DURING        CONLIM
C                   TESTING  (8I10)                                    CONLIM
C                   (NOTE - DATA CAN BE CONTINUED TO AS MANY CARDS     CONLIM
C                    AS NECESSARY CONFORMING WITH PREVIOUS CARDS)      CONLIM
C                                                                   CONLIM
C         CARD 6  - INDICATORS OF COMPONENT IN PARALLEL OR             CONLIM
C                   SERIES  (8I10)   (NOTE - CONTINUE DATA AS PER      CONLIM
C                   THE PRECEDING COMPONENT DATA)                      CONLIM
C                                                                   CONLIM
C                        0 -- INDICATES SERIES LINKAGE                 CONLIM
C                        1 -- INDICATES PARALLEL LINKAGE               CONLIM
C                                                                   CONLIM
C         CARD 7  - ALPHA UPPER CONFIDENCE CRITERIA  (F10.5)           CONLIM
C                   (NOTE - AS MANY ALPHA VALUES AS DESIRED CAN BE     CONLIM
C                    SUPPLIED, ONE PER CARD)                           CONLIM
C                                                                   CONLIM
C         CARD 8  - END OF FILE CARD  (7-8-9 IN FIRST COLUMN)          CONLIM
C                                                                   CONLIM
C         AS MANY SETS OF DATA AS DESIRED CAN BE INPUT PROVIDED EACH   CONLIM
C         SET (CARD 1 - CARD 7 INCLUSIVE) IS SEPARATED BY CARD 8 AND   CONLIM
C         THE FINAL SET ALSO CONTAINS CARD 8.                          CONLIM
C                                                                   CONLIM
      INTEGER  OPT                                                  CONLIM
      EXTERNAL SUB                                                  CONLIM
      DIMENSION N(50),NX(50),FN(50),FNX(50),PHAT(50),QHAT(50),        CONLIM
     1FACT(50),M(50),MX(50),MTYPE(50),NTYPE(50),NSTAR(50),OPT(6),     CONLIM
     2TITLE(8),ID(10),VAL(11),SAVAL(10),IDINV(10)                     CONLIM
      COMMON /QVAR/N,FN,FNX,PHAT,QHAT,IFLAG,IVOL,IVSIZE,QLIM,IVCOL,OPT, CONLIM
```

```
     1ITOTAL,NSTAR
      COMMON /PSIFLG/IPSIFG
      COMMON /FACTR/ FACT
      COMMON /TYPE/NTYPE
      COMMON /FIRST/ IFIRST,DEL,IFREEZ,NIDS,ID,NLAST,IVAL,VAL,
     1SAVAL,IDINV
      EQUIVALENCE (PHAT(1),M(1)),(GHAT(1),MX(1)),(NSTAR(1),MTYPE(1))
      DIMENSION NFMAX(8,500),NFMIN(8,500),NFMAXX(8,500),NFMINX(8,500)
      DATA  IVSIZE/500/,IVCOL/8/
C
C     RECOVERY FLAGS SET FOR ABNORMAL TERMINATION
C
      CALL RECOVR(SUB,63,0)
C
C     BUILD SMALL FACTORIAL ARRAY TO AVOID SOME UNNECESSARY CALCULATION
C
      FACT(1) = 1.0
      DO 20 I=1,49
      XI = I + 1
      FACT(I+1) = XI*FACT(I)
   20 CONTINUE
      IVS = IVSIZE
      IVCOLS = IVCOL
      IFILE = 0
      IFREEZ = 0
C
C     READ INPUT DATA
C
      READ 30, OPT
   30 FORMAT (6I1)
   35 IALP = 1
      IFLAG = 0
      NINBAD = 0
      READ 40, TITLE
   40 FORMAT (8A10)
      READ 60, NCOM
   60 FORMAT (I3)
      READ 80, (N(I),I=1,NCOM)
   80 FORMAT (8I10)
      READ 80, (NX(I),I=1,NCOM)
      READ 80, (NTYPE(I),I=1,NCOM)
      DO 100 I=1,NCOM
      FN(I) = N(I)
      FNX(I) = NX(I)
  100 CONTINUE
  110 READ (5,120) ALPHA
  120 FORMAT (F10.5)
      IF (EOF(5)) 300,121
  121 IF (IALP.GT.1) GO TO 1470
      IF (OPT(4).LE.1) GO TO 131
      READ (4,80) MCOM
      READ (4,80) (M(I),I=1,MCOM),(MX(I),I=1,MCOM),(MTYPE(I),I=1,MCOM)
      IF (MCOM.NE.NCOM) GO TO 123
      DO 122 I=1,NCOM
```

```
      IF (N(I).NE.M(I)) GO TO 123                                       CONLIM
      IF (NX(I).NE.MX(I)) GO TO 123                                     CONLIM
      IF (NTYPE(I).NE.MTYPE(I)) GO TO 123                               CONLIM
  122 CONTINUE                                                          CONLIM
      GO TO 125                                                         CONLIM
  123 NINBAD = 1                                                        CONLIM
      GO TO 150                                                         CONLIM
  125 NVOL = 0                                                          CONLIM
      ITOTAL = 0                                                        CONLIM
      IECS = 1                                                          CONLIM
      NWORDS = IVSIZE*IVCOL                                             CONLIM
      THYP = 0.0                                                        CONLIM
  126 IN = 1                                                            CONLIM
  127 READ (4,128) (NFMIN(IJ,IN),NFMAX(IJ,IN),IJ=1,NCOM)                CONLIM
  128 FORMAT (2I4)                                                      CONLIM
      IF (EOF(4)) 130,129                                               CONLIM
  129 ITUP = 1                                                          CONLIM
      DO 1290 IT=1,NCOM                                                 CONLIM
      ITUP = ITUP*(NFMAX(IT,IN) - NFMIN(IT,IN) + 1)                     CONLIM
 1290 CONTINUE                                                          CONLIM
      ITOTAL = ITOTAL + ITUP                                            CONLIM
      IN = IN + 1                                                       CONLIM
      NVOL = NVOL + 1                                                   CONLIM
      IF (IN.LE.IVSIZE) GO TO 127                                       CONLIM
      CALL WRITEC(NFMIN,IECS,NWORDS)                                    CONLIM
      IECS = IECS + NWORDS                                              CONLIM
      CALL WRITEC(NFMAX,IECS,NWORDS)                                    CONLIM
      IECS = IECS + NWORDS                                              CONLIM
      IFLAG = IFLAG + 1                                                 CONLIM
      GO TO 126                                                         CONLIM
  130 IVOL = IN - 1                                                     CONLIM
      IF (IVOL.EQ.0) GO TO 137                                          CONLIM
      NWORDS = IVOL*IVCOL                                               CONLIM
      CALL WRITEC(NFMIN,IECS,NWORDS)                                    CONLIM
      IECS = IECS + NWORDS                                              CONLIM
      CALL WRITEC(NFMAX,IECS,NWORDS)                                    CONLIM
      IF (OPT(3).EQ.1)  CALL PSI(NCOM,NVOL,NFMIN,NFMAX,NFMINX,NFMAXX,   CONLIM
     1                           IVS,IVCOLS,IER)                        CONLIM
      GO TO 137                                                         CONLIM
C                                                                       CONLIM
C     DETERMINE THE INDEX SET AND INITIAL H FUNCTION VALUE FOR THE      CONLIM
C     BEGINNING PHAT PROBABILITIES - START THE BALL ROLLING.            CONLIM
C                                                                       CONLIM
  131 CALL SECOND(THYP1)                                                CONLIM
      CALL PSI(NCOM,NVOL,NFMIN,NFMAX,NFMINX,NFMAXX,IVS,IVCOLS,IER)      CONLIM
      CALL SECOND(THYP2)                                                CONLIM
      THYP = THYP2 - THYP1                                              CONLIM
      IF (OPT (4).NE.1) GO TO 137                                       CONLIM
      WRITE (3,80) NCOM                                                 CONLIM
      WRITE (3,80) (N(I),I=1,NCOM),(NX(I),I=1,NCOM),(NTYPE(I),I=1,NCOM) CONLIM
      MFLAG = 0                                                         CONLIM
      IECS = 1                                                          CONLIM
      NWORDS = IVSIZE*IVCOL                                             CONLIM
      IFLG = 0                                                          CONLIM
```

```
 132 IF (IFLAG.EQ.MFLAG) GO TO 135                                          CONLIM
     IVEND = IVSIZE                                                     CONLIM
 133 CALL READEC(NFMIN,IECS,NWORDS)                                     CONLIM
     IECS = IECS + NWORDS                                               CONLIM
     CALL READEC(NFMAX,IECS,NWORDS)                                     CONLIM
     IECS = IECS + NWORDS                                               CONLIM
1330 DO 134 IK=1,IVEND                                                  CONLIM
     WRITE (3,128) (NFMIN(IJ,IK),NFMAX(IJ,IK), IJ=1,NCOM)               CONLIM
 134 CONTINUE                                                           CONLIM
     MFLAG = MFLAG + 1                                                  CONLIM
     IF (IFLG.EQ.1) GO TO 135                                           CONLIM
     GO TO 132                                                          CONLIM
 135 IF (IVOL.EQ.0) GO TO 136                                           CONLIM
     IVEND = IVOL                                                       CONLIM
     NWORDS = IVOL*IVCOL                                                CONLIM
     IFLG = 1                                                           CONLIM
     IF (IFLAG.EQ.0) GO TO 1330                                         CONLIM
     GO TO 133                                                          CONLIM
 136 ENDFILE 3                                                          CONLIM
     IFILE = IFILE + 1                                                  CONLIM
 137 TCOMP = 0.0                                                        CONLIM
     IF (OPT(5).EQ.0) GO TO 138                                         CONLIM
     CALL SECOND(TCOMP1)                                                CONLIM
     CALL COMPAQ(NCOM,NFMIN,NFMAX,IVS,IVCOLS)                           CONLIM
     CALL SECOND(TCOMP2)                                                CONLIM
     TCOMP = TCOMP2 - TCOMP1                                            CONLIM
 138 PRINT 139                                                          CONLIM
 139 FORMAT (1H1///)                                                    CONLIM
     IF (OPT(4).EQ.2) GO TO 141                                         CONLIM
     PRINT 140, THYP                                                    CONLIM
 140 FORMAT (20X,56HTIME REQUIRED TO DETERMINE INDEX SET AND HYPERVOLUM CONLIM
    1ES = ,F9.3, 8H SECONDS/)                                           CONLIM
     IF (OPT(4).NE.1) GO TO 143                                         CONLIM
     PRINT 1400, IFILE,NVOL                                             CONLIM
1400 FORMAT (/20X,35HHYPERVOLUME STRUCTURE SAVED ON FILE,I2,14H OF TAPE CONLIM
    13    (,I6,15H HYPERVOLUMES )/)                                     CONLIM
     GO TO 143                                                          CONLIM
 141 PRINT 142                                                          CONLIM
 142 FORMAT (20X,43HINDEX SET AND HYPERVOLUMES INPUT FROM TAPE4/)       CONLIM
 143 IF (OPT(5).EQ.0) GO TO 145                                         CONLIM
     PRINT 144, TCOMP                                                   CONLIM
 144 FORMAT (20X,50HTIME REQUIRED TO PERFORM HYPERVOLUME CROSSCHECK = , CONLIM
    1F9.3,8H SECONDS/)                                                  CONLIM
     GO TO 147                                                          CONLIM
 145 PRINT 146                                                          CONLIM
 146 FORMAT (20X,31HHYPERVOLUME CROSSCHECK BYPASSED/)                   CONLIM
 147 IF (OPT(2).EQ.1) GO TO 150                                         CONLIM
1470 DO 148 I=1,NCOM                                                    CONLIM
     PHAT(I) = (FNX(I)+1.0)/(FN(I)+2.0)                                 CONLIM
 148 CONTINUE                                                           CONLIM
     CALL SECOND(TFIT1)                                                 CONLIM
     CALL FIT(NCOM,ALPHA,GOFP,NFMIN,NFMAX,IVS,IVCOLS,IERR)              CONLIM
     IF (IERR.EQ.3) GO TO 150                                           CONLIM
     CALL SECOND(TFIT2)                                                 CONLIM
```

```
      TFIT = TFIT2 - TFIT1                                              CONLIM
      PRINT 149, TFIT                                                   CONLIM
  149 FORMAT (20X,34HTIME REQUIRED FOR FIT PROCEDURE = ,F9.3,8H SECONDS) CONLIM
  150 PRINT 151, TITLE                                                  CONLIM
  151 FORMAT (1H1///27X,8A10//)                                         CONLIM
      PRINT 152                                                         CONLIM
  152 FORMAT (58X,17HSYSTEM COMPONENTS/53X,28HNUMBER OF TESTS AND FAILUR CONLIM
     1ES/)                                                              CONLIM
      ILIST = NCOM                                                      CONLIM
      ISTRT = 1                                                         CONLIM
      IEND = 0                                                          CONLIM
  153 IF (ILIST.LE.20) GO TO 156                                        CONLIM
      IEND = IEND + 20                                                  CONLIM
      PRINT 154, (I,I=ISTRT,IEND)                                       CONLIM
  154 FORMAT (27X,20I4)                                                 CONLIM
      PRINT 154, (N(I),I=ISTRT,IEND)                                    CONLIM
      PRINT 154, (NX(I),I=ISTRT,IEND)                                   CONLIM
      PRINT 155                                                         CONLIM
  155 FORMAT (///)                                                      CONLIM
      ILIST = ILIST - 20                                                CONLIM
      ISTRT = ISTRT + 20                                                CONLIM
      GO TO 153                                                         CONLIM
  156 IF (ILIST.LT.16) GO TO 158                                        CONLIM
      PRINT 157, (I,I=ISTRT,NCOM)                                       CONLIM
  157 FORMAT (34X,20I4)                                                 CONLIM
      PRINT 157, (N(I),I=ISTRT,NCOM)                                    CONLIM
      PRINT 157, (NX(I),I=ISTRT,NCOM)                                   CONLIM
      GO TO 170                                                         CONLIM
  158 IF (ILIST.LT.12) GO TO 160                                        CONLIM
      PRINT 159, (I,I=ISTRT,NCOM)                                       CONLIM
  159 FORMAT (42X,16I4)                                                 CONLIM
      PRINT 159, (N(I),I=ISTRT,NCOM)                                    CONLIM
      PRINT 159, (NX(I),I=ISTRT,NCOM)                                   CONLIM
      GO TO 170                                                         CONLIM
  160 IF (ILIST.LT.8) GO TO 162                                         CONLIM
      PRINT 161, (I,I=ISTRT,NCOM)                                       CONLIM
  161 FORMAT (48X,12I4)                                                 CONLIM
      PRINT 161, (N(I),I=ISTRT,NCOM)                                    CONLIM
      PRINT 161, (NX(I),I=ISTRT,NCOM)                                   CONLIM
      GO TO 170                                                         CONLIM
  162 IF (ILIST.LT.4) GO TO 164                                         CONLIM
      PRINT 163, (I,I=ISTRT,NCOM)                                       CONLIM
  163 FORMAT (56X,8I4)                                                  CONLIM
      PRINT 163, (N(I),I=ISTRT,NCOM)                                    CONLIM
      PRINT 163, (NX(I),I=ISTRT,NCOM)                                   CONLIM
      GO TO 170                                                         CONLIM
  164 PRINT 165, (I,I=ISTRT,NCOM)                                       CONLIM
  165 FORMAT (63X,4I4)                                                  CONLIM
      PRINT 165, (N(I),I=ISTRT,NCOM)                                    CONLIM
      PRINT 165, (NX(I),I=ISTRT,NCOM)                                   CONLIM
  170 IF (NINBAD.EQ.0) GO TO 171                                        CONLIM
      PRINT 1700                                                        CONLIM
 1700 FORMAT (///40X,47HDATA DESCRIBED AS INPUT TO CONLIM DID NOT MATCH/ CONLIM
     140X,40HREQUESTED DATA FROM TAPE4 - SET BYPASSED//40X,35HINFORMATIO CONLIM
```

```
     2N FROM TAPE4 LISTED BELOW//)                                      CONLIM
      PRINT 1710, (I,M(I),MX(I),MTYPE(I),I=1,MCOM)                       CONLIM
 1710 FORMAT (35X,4HI = ,I2,5X,7HN(I) = ,I4,5X,8HNX(I) = ,I4,5X,         CONLIM
     111HNTYPE(I) = ,I1)                                                 CONLIM
      GO TO 1730                                                         CONLIM
  171 IF (OPT(2).EQ.0) GO TO 174                                         CONLIM
      PRINT 175, ALPHA                                                   CONLIM
      PRINT 172                                                          CONLIM
  172 FORMAT (///20X,32HHYPERVOLUME ONLY OPTION SELECTED//)              CONLIM
 1730 READ (5,120) ALPHA                                                 CONLIM
      IF  (EOF(5)) 300,1732                                              CONLIM
 1732 IF (NINBAD.EQ.1) GO TO 1730                                        CONLIM
      IF (OPT(3).NE.1) GO TO 1736                                        CONLIM
      PRINT 1734, ALPHA                                                  CONLIM
 1734 FORMAT (/20X,62HINDEX SET COMPOSITION PREVIOUSLY LISTED VIA OPTION CONLIM
     1 3, ALPHA = ,F9.4,9H BYPASSED/)                                    CONLIM
      GO TO 1730                                                         CONLIM
 1736 PRINT 1738, ALPHA                                                  CONLIM
 1738 FORMAT (/20X,43HINDEX SET DETERMINED VIA OPTION 2, ALPHA = ,F9.4,  CONLIM
     19H BYPASSED/)                                                      CONLIM
      GO TO 1730                                                         CONLIM
  174 PRINT 175, ALPHA                                                   CONLIM
  175 FORMAT (//43X,31HALPHA UPPER CONFIDENCE LIMIT = ,F9.4)             CONLIM
      IF (IFREEZ .GT. 0) GO TO 200                                       CONLIM
      PRINT 178, QOFP                                                    CONLIM
  178 FORMAT (//43X,30HSYSTEM FAILURE PROBABILITY Q =,E14.7)             CONLIM
      PRINT 180, NVOL,ITOTAL                                             CONLIM
  180 FORMAT (//43X,38HNUMBER OF HYPERVOLUMES IN INDEX SET = ,I5/        CONLIM
     143X,33HTOTAL NUMBER OF NTUPLES IN SET = ,I6)                       CONLIM
      PRINT 182                                                          CONLIM
  182 FORMAT (//43X,45HINDIVIDUAL COMPONENT P VALUES ARE AS FOLLOWS //)  CONLIM
      PRINT 184, (I,PHAT(I), I=1,NCOM)                                   CONLIM
  184 FORMAT (50X,4HI = ,I2,5X,6HP(I) =,E14.7)                           CONLIM
      IALP = 2                                                           CONLIM
      IF (IERR.NE.2) GO TO 195                                           CONLIM
      PRINT 190                                                          CONLIM
  190 FORMAT (//40X,                                                     CONLIM
     1 51H***************************************************/40X,       CONLIM
     2 51H*                                                 */40X,       CONLIM
     3 51H*  NEAR-ZERO SERIES COMPONENTS MAY INDICATE THAT  */40X,       CONLIM
     4 51H*  CONLIM RESULTS HAVE NOT REFLECTED TRUE SYSTEM  */40X,       CONLIM
     5 51H*  VALUES                                         */40X,       CONLIM
     6 51H*                                                 */40X,       CONLIM
     7 51H***************************************************)          CONLIM
  195 PRINT 139                                                          CONLIM
      GO TO 110                                                          CONLIM
C                                                                        CONLIM
C     OSCILLATION BETWEEN COMPONENTS WITHOUT CONVERGENCE                 CONLIM
C                                                                        CONLIM
  200 IF (OPT(6).EQ.1)GO TO 240                                          CONLIM
      PRINT 180, NVOL,ITOTAL                                             CONLIM
      PRINT 205                                                          CONLIM
  205 FORMAT (//40X,                                                     CONLIM
     1 51H***************************************************/40X,       CONLIM
```

```
     2 51H*                                                 */40X,       CONLIM
     3 51H*  OSCILLATION BETWEEN COMPONENT P VALUES           */40X,       CONLIM
     4 51H*  ENCOUNTERED - EXECUTICN FOR THIS ALPHA VALUE     */40X,       CONLIM
     5 51H*  IS TERMINATED                                    */40X,       CONLIM
     6 51H*  BELOW ARE LISTED THE INDIVIDUAL P VALUES UPON    */40X,       CONLIM
     7 51H*  TERMINATION WITH OSCILLATING COMPONENTS MARKED *)             CONLIM
      DO 230 I=1,NCOM                                                      CONLIM
      DO 210 J=1,NIDS                                                      CONLIM
      IF (I.EQ.ID(J)) GO TO 220                                            CONLIM
  210 CONTINUE                                                             CONLIM
      PRINT 215, I,PHAT(I)                                                 CONLIM
  215 FORMAT (40X,14H*        I = ,I2,5X,6HP(I) = ,E14.7,10H          *)  CONLIM
      GO TO 230                                                            CONLIM
  220 PRINT 225, I,PHAT(I)                                                 CONLIM
  225 FORMAT (40X,14H*        I = ,I2,5X,6HP(I) = ,E14.7,10H ***      *)  CONLIM
  230 CONTINUE                                                             CONLIM
      PRINT 235                                                            CONLIM
  235 FORMAT (40X,                                                         CONLIM
     1 51H*                                                 */40X,         CONLIM
     2 51H***************************************************)            CONLIM
      IFREEZ = 0                                                           CONLIM
      GO TO 195                                                            CONLIM
C                                                                          CONLIM
C     TAKE BRANCHES FOR USER INFORMATION                                   CONLIM
C                                                                          CONLIM
  240 PRINT 178, QOFP                                                      CONLIM
      PRINT 180, NVOL,ITOTAL                                               CONLIM
      PRINT 182                                                            CONLIM
      PRINT 184, (I,PHAT(I), I=1,NCOM)                                     CONLIM
      PRINT 250, ID(IFREEZ)                                                CONLIM
  250 FORMAT (//43X,44HRESULTS SHOWN ABOVE WERE OBTAINED BY HOLDING/43X,   CONLIM
     110HCOMPONENT ,I2,42H FIXED AS THE SLACK COMPONENT VIA OPTION 6//)   CONLIM
      PRINT 139                                                            CONLIM
      CALL FIT(NCOM,ALPHA,QOFP,NFMIN,NFMAX,IVS,IVCOLS,IERR)                CONLIM
      IF (IFREEZ.EQ.0) GO TO 110                                           CONLIM
      GO TO 150                                                            CONLIM
C                                                                          CONLIM
C     COMPLETED ONE SET OF DATA                                            CONLIM
C                                                                          CONLIM
  300 PRINT 320, TITLE                                                     CONLIM
  320 FORMAT (1H1///20X,17HEND OF ANALYSIS -,8A10)                         CONLIM
      READ (5,30) OPT                                                      CONLIM
      IF (EOF(5)) 340,35                                                   CONLIM
  340 CALL EXIT                                                            CONLIM
      END                                                                  CONLIM
```

```
      SUBROUTINE PSI(NCOMD,NVOL,NFMIN,NFMAX,NFMINX,NFMAXX,IVSIZE,IVCOL,
     1                 IERR)
C
C     CONSTRUCTION OF THE INDEX SET PSI BY USE OF HYPERVOLUMES
C
      INTEGER  OPT
      DIMENSION N(50),FN(50),FNX(50),PHAT(50),QHAT(50),FN2(50),
     1NSTAR(50),PSAVE(50),OPT(6),JFLAG(5)
      DIMENSION  NFMIN(IVCOL,IVSIZE),NFMAX(IVCOL,IVSIZE),
     1           NFMINX(IVCOL,IVSIZE),NFMAXX(IVCOL,IVSIZE)
      COMMON /QVAR/N,FN,FNX,PHAT,QHAT,IFLAG,IVOL,IVS,QLIM,IVC,OPT,
     1ITOTAL,NSTAR
      COMMON /PSIFLG/IPSIFG
      DATA  NFAULT/1000000/
C
      IERR = 0
      IPSIFG = 0
      NCOMP = NCOMD
      IF (OPT(4).EQ.2) GO TO 900
      IFLAG = 0
      IDM = 0
      NWORDS = IVCOL*IVSIZE
      MWORDS = NWORDS
      ITOTAL = 0
C
C     INITIALIZATION OF CRITERION FOR HYPERVOLUME ACCEPTANCE INTO PSI
C
      DO 20 I=1,NCOMP
      FN2(I) = 1.0/(FN(I)+2.0)
      PHAT(I) = (FNX(I)+1.0)*FN2(I)
      PSAVE(I) = PHAT(I)
   20 CONTINUE
      QLIM = QTHETA(PHAT)
C
C     INITIALIZE NSTAR
C
      DO 40 ICOMP=1,NCOMP
      NSTAR(ICOMP) = 0
   40 CONTINUE
      IVOL = 1
      IVCHK = 1
      IECS = 1
C
C     START NEW HYPERVOLUME UPPER LIMITS AT STAR VALUES AND THEN WORK
C     UPWARD TO MAXIMUM
C
  100 DO 120 ICOMP=2,NCOMP
      NFMAX(ICOMP,IVOL) = NSTAR(ICOMP)
      XSTAR = NSTAR(ICOMP) + 1
      PHAT(ICOMP) = XSTAR*FN2(ICOMP)
  120 CONTINUE
C
C     DETERMINE UPPER LIMITS ON EACH DIMENSION
C
```

```
      DO 180 ICOMP=1,NCOMP
      IF (NSTAR(ICOMP).EQ.N(ICOMP)) GO TO 180
      NICOMP = N(ICOMP) + 1
      ISTRT = NSTAR(ICOMP) + 2
      DO 140 I=ISTRT,NICOMP
      IA = I-1
      A = I
      PHAT(ICOMP) = A*FN2(ICOMP)
      IF (QTHETA(PHAT) - QLIM) 140,140,160
  140 CONTINUE
      NFMAX(ICOMP,IVOL) = IA
      GO TO 180
  160 IF (IA.EQ.0) GO TO 170
      NFMAX(ICOMP,IVOL) = IA - 1
      A = IA
      PHAT(ICOMP) = A*FN2(ICOMP)
      GO TO 180
  170 NFMAX(ICOMP,IVOL) = 0
  180 CONTINUE
C
C     SET UP LOWER LIMITS ON EACH COMPONENT
C     COMPLETES DEFINITION OF A HYPERVOLUME
C
      DO 200 ICOMP=1,NCOMP
      NFMIN(ICOMP,IVOL) = NSTAR(ICOMP)
  200 CONTINUE
C
C     CHECK ON OVERLAP AS DETAILED IN ALGORITHM STEP 5
C
      IF (IVCHK.EQ.1) GO TO 430
      IECI = 1
      MFLAG = 0
      IF (IVOL.EQ.1) GO TO 402
      IEND = IVOL - 1
      DO 400 ITRY=1,IEND
      ICHK = IEND - ITRY + 1
      JF = 0
      JCOMP = 0
      DO 300 ICOMP=2,NCOMP
      IF (NFMAX(ICOMP,IVOL) - NFMIN(ICOMP,ICHK)) 400,210,210
  210 IF (NFMAX(ICOMP,IVOL) - NFMAX(ICOMP,ICHK)) 240,240,220
  220 IF (NFMIN(ICOMP,IVOL) - NFMAX(ICOMP,ICHK)) 260,260,400
  240 IF (NFMIN(ICOMP,IVOL) - NFMIN(ICOMP,ICHK)) 250,300,300
  250 JCOMP = ICOMP
      GO TO 300
  260 IF (JF.EQ.5) GO TO 300
      JF = JF+1
      JFLAG(JF) = ICOMP
  300 CONTINUE
C
C     SCANNED ALL DIMENSIONS IN THE VOLUME
C     IF OVERLAP EXISTS, USE THEOREMS 2 AND 3 TO SEPARATE VOLUMES.
C
      IF (JCOMP.GT.0) GO TO 340
```

```
      IF (NFMAX(1,IVOL) - NFMAX(1,ICHK)) 302,302,304
  302 IF (JF.EQ.0) GO TO 500
      GO TO 305
  304 IF (JF.EQ.5) GO TO 305
      JF = JF+1
      JFLAG(JF) = 1
  305 DO 310 JI =1,JF
      JFLG = JFLAG(JI)
      IF (NFMAX(JFLG,ICHK).LT.N(JFLG)) GO TO 315
  310 CONTINUE
      GO TO 500
  315 NFMIN(JFLG,IVOL) = NFMAX(JFLG,ICHK) + 1
      LAP = ICHK + IFLAG*IVSIZE
      IERR = 1
      GO TO 400
  340 NFMAX(JCOMP,IVOL) = NFMIN(JCOMP,ICHK) - 1
  400 CONTINUE
  402 IF (MFLAG.EQ.IFLAG) GO TO 430
      CALL READEC(NFMINX,IECI,MWORDS)
      IECI = IECI + MWORDS
      CALL READEC(NFMAXX,IECI,MWORDS)
      IECI = IECI + MWORDS
      MFLAG = MFLAG + 1
      DO 420 ITRY=1,IVSIZE
      ICHK = IVSIZE - ITRY + 1
      JF = 0
      JCOMP = 0
      DO 415 ICOMP=2,NCOMP
      IF (NFMAX(ICOMP,IVOL) - NFMINX(ICOMP,ICHK)) 420,405,405
  405 IF (NFMAX(ICOMP,IVOL) - NFMAXX(ICOMP,ICHK)) 408,408,407
  407 IF (NFMIN(ICOMP,IVOL) - NFMAXX(ICOMP,ICHK)) 414,414,420
  408 IF (NFMIN(ICOMP,IVOL) - NFMINX(ICOMP,ICHK)) 410,415,415
  410 JCOMP = ICOMP
      GO TO 415
  414 IF (JF.EQ.5) GO TO 415
      JF = JF + 1
      JFLAG(JF) = ICOMP
  415 CONTINUE
      IF (JCOMP.GT.0) GO TO 419
      IF (NFMAX(1,IVOL) - NFMAXX(1,ICHK)) 416,416,417
  416 IF (JF.EQ.0) GO TO 500
      GO TO 418
  417 IF (JF.EQ.5) GO TO 418
      JF = JF + 1
      JFLAG(JF) = 1
  418 DO 4180 JI=1,JF
      JFLG = JFLAG(JI)
      IF (NFMAXX(JFLG,ICHK).LT.N(JFLG)) GO TO 4190
 4180 CONTINUE
      GO TO 500
 4190 NFMIN(JFLG,IVOL) = NFMAXX(JFLG,ICHK) + 1
      LAP = ICHK + (MFLAG-1)*IVSIZE
      IERR = 1
      GO TO 420
```

```
  419 NFMAX(JCOMP,IVOL) = NFMINX(JCOMP,ICHK) - 1                          CONLIM
  420 CONTINUE                                                            CONLIM
      GO TO 402                                                           CONLIM
C                                                                         CONLIM
C     LATEST HYPERVOLUME PROPERLY DESCRIBED - DETERMINE NEXT STARRED      CONLIM
C     VALUE IN THE SECOND DIMENSION                                       CONLIM
C                                                                         CONLIM
  430 NSTAR(2) = NFMAX(2,IVOL) + 1                                        CONLIM
      ITUPLE = 1                                                          CONLIM
      DO 440 I=1,NCOMP                                                    CONLIM
      ITUPLE = ITUPLE*(NFMAX(I,IVOL)-NFMIN(I,IVOL)+1)                     CONLIM
  440 CONTINUE                                                            CONLIM
      ITOTAL = ITOTAL + ITUPLE                                            CONLIM
      IVCHK = IVCHK + 1                                                   CONLIM
      IF (IVOL.LT.IVSIZE) GO TO 460                                       CONLIM
      CALL WRITEC(NFMIN,IECS,NWORDS)                                      CONLIM
      IECS = IECS + NWORDS                                                CONLIM
      CALL WRITEC(NFMAX,IECS,NWORDS)                                      CONLIM
      IFLAG = IFLAG + 1                                                   CONLIM
      IECS = IECS + NWORDS                                                CONLIM
      IVOL = 1                                                            CONLIM
      GO TO 600                                                           CONLIM
  460 IVOL = IVOL + 1                                                     CONLIM
      GO TO 600                                                           CONLIM
C                                                                         CONLIM
C     HYPERVOLUME DID NOT CONTRIBUTE - REDUNDANT WITH VOLUMES             CONLIM
C     PREVIOUSLY DEFINED                                                  CONLIM
C                                                                         CONLIM
  500 NSTAR(2) = MAX0(NSTAR(2),NFMAX(2,IVOL)) + 1                         CONLIM
C                                                                         CONLIM
C     STEPS 7 THRU 11 - CALCULATE ALL NEW STARRED VALUES FOR EACH OF      CONLIM
C     THE DIMENSIONS - WHEN FOUND, GO BACK THROUGH ENTIRE VOLUME          CONLIM
C     DETERMINATION AND CHECKING                                          CONLIM
C                                                                         CONLIM
  600 DO 620 I=1,NCOMP                                                    CONLIM
      XSTAR = NSTAR(I) + 1                                                CONLIM
      PHAT(I) = XSTAR*FN2(I)                                              CONLIM
  620 CONTINUE                                                            CONLIM
      ICOMP = 2                                                           CONLIM
      IF (NSTAR(2).GT.N(2)) GO TO 700                                     CONLIM
  640 XSTAR = NSTAR(ICOMP) + 1                                            CONLIM
      PHAT(ICOMP) = XSTAR*FN2(ICOMP)                                      CONLIM
C                                                                         CONLIM
C     SEE IF READY TO GO FIND A NEW HYPERVOLUME                           CONLIM
C                                                                         CONLIM
      IF (QTHETA(PHAT) - QLIM) 100,100,700                                CONLIM
C                                                                         CONLIM
C     STARRED VALUES REPRESENT CURRENT MINIMUMS - VALUE THAT MUST BE      CONLIM
C     USED IN THIS DIMENSION, BY DEFAULT, IS ZERO                         CONLIM
C                                                                         CONLIM
  700 NSTAR(ICOMP) = 0                                                    CONLIM
      PHAT(ICOMP) = FN2(ICOMP)                                            CONLIM
      ICOMP = ICOMP + 1                                                   CONLIM
      IF (ICOMP.GT.NCOMP) GO TO 800                                       CONLIM
```

```
C
C     SET UP THE VALUE FOR THE NEWEST STARRED DIMENSION - USE THE
C     MINIMUM OF THE UPPER BOUNDS THAT ARE .GE. THE CURRENT STARRED
C     VALUE IN THIS DIMENSION.  THAT IS, NEW L* = MIN( U(K,J).GE. L*).
C     NOTE THAT IF U(K,J) = L*, THIS IS THE DEFINED MINIMUM.
C
      NEXT = NFAULT
      IF (IVCHK.EQ.1) GO TO 760
      IF (IVOL.EQ.1) GO TO 749
      IEND = IVOL-1
      DO 750 ITRY =1,IEND
      ICHK = IEND - ITRY + 1
      IF (NFMAX(ICOMP,ICHK) - NSTAR(ICOMP)) 750,760,740
  740 NEXT = MIN0(NEXT,NFMAX(ICOMP,ICHK))
  750 CONTINUE
  749 IECI = 1
      MFLAG = 0
  751 IF (MFLAG.EQ.IFLAG) GO TO 758
      CALL READEC(NFMINX,IECI,MWORDS)
      IECI = IECI + MWORDS
      CALL READEC(NFMAXX,IECI,MWORDS)
      IECI = IECI + MWORDS
      MFLAG = MFLAG +1
      DO 756 ITRY=1,IVSIZE
      ICHK = IVSIZE - ITRY + 1
      IF (NFMAXX(ICOMP,ICHK) - NSTAR(ICOMP)) 756,760,754
  754 NEXT = MIN0(NEXT,NFMAXX(ICOMP,ICHK))
  756 CONTINUE
      GO TO 751
C
C     NOTE - SINCE L* INDICATES THE MINIMUM FOR THE PREVIOUS SET,
C            THERE MUST BE AT LEAST ONE U(K,J).GE.L* PROVIDED THERE
C            EXISTS MORE THAN ONE HYPERVOLUME
C
  758 IF (NEXT.EQ.NFAULT) GO TO 760
C
C     MINIMUM READY FOR THIS DIMENSION
C
      NSTAR(ICOMP) = NEXT + 1
      GO TO 780
  760 NSTAR(ICOMP) = NSTAR(ICOMP) + 1
  780 IF (NSTAR(ICOMP).GT.N(ICOMP)) GO TO 700
      GO TO 640
C
C     FINI - ALL HYPERVOLUMES DESCRIBED
C
  800 NVOL = IVCHK-1
      IVOL = IVOL-1
      IPSIFG = 1
      IF (IFLAG.EQ.0) GO TO 900
      IF (IVOL.EQ.0) GO TO 900
      NWORDS = IVCOL*IVOL
      CALL WRITEC(NFMIN,IECS,NWORDS)
      IECS = IECS + NWORDS
```

```
      CALL WRITEC(NFMAX,IECS,NWORDS)
C
C     CHECK OPTIONS - SEE IF HYPERVOLUME PRINTOUT DESIRED
C
  900 IF (IFLAG.GT.0) GO TO 1000
      IF (OPT(3).EQ.0) RETURN
      PRINT 905
  905 FORMAT (1H1///45X,42HHYPERVOLUME STRUCTURE WITHIN INDEX SET PSI//)
      PRINT 910, NVOL,ITOTAL
  910 FORMAT (45X,38HNUMBER OF HYPERVOLUMES IN INDEX SET = ,I5/
     147X,33HTOTAL NUMBER OF NTUPLES IN SET = ,I6//)
      DO 980 I=1,NVOL
      PRINT 920, I
  920 FORMAT (/20X,10H***VOLUME ,I4/)
      DO 960 J=1,NCOMP
      PRINT 940, J,NFMIN(J,I),NFMAX(J,I)
  940 FORMAT (30X,4HJ = ,I3,2X,9HMIN(J) = ,I4,2X,9HMAX(J) = ,I4)
  960 CONTINUE
  980 CONTINUE
      RETURN
 1000 NWORDS = IVCOL*IVSIZE
      IECS = 1
      KVOL = 0
      IF (OPT(3).EQ.0) RETURN
      PRINT 905
      PRINT 910, NVOL,ITOTAL
      DO 1060 I=1,IFLAG
      CALL READEC(NFMIN,IECS,NWORDS)
      IECS = IECS + NWORDS
      CALL READEC(NFMAX,IECS,NWORDS)
      IECS = IECS + NWORDS
      DO 1040 J=1,IVSIZE
      KVOL = KVOL + 1
      PRINT 920, KVOL
      DO 1020 ICOMP=1,NCOMP
      PRINT 940, ICOMP,NFMIN(ICOMP,J),NFMAX(ICOMP,J)
 1020 CONTINUE
 1040 CONTINUE
 1060 CONTINUE
      IF (IVOL.EQ.0) RETURN
      NWORDS = IVCOL*IVOL
      CALL READEC(NFMIN,IECS,NWORDS)
      IECS = IECS + NWORDS
      CALL READEC(NFMAX,IECS,NWORDS)
      DO 1100 J=1,IVOL
      KVOL = KVOL + 1
      PRINT 920, KVOL
      DO 1080 ICOMP=1,NCOMP
      PRINT 940, ICOMP,NFMIN(ICOMP,J),NFMAX(ICOMP,J)
 1080 CONTINUE
 1100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COMPAQ(NCOMD,NFMIN,NFMAX,IVSIZE,IVCOL)
C
C     CROSSCHECK ON HYPERVOLUMES TO INSURE THE MINIMUM AND MAXIMUM
C     LIMITS ON VOLUMES SATISFY BOUND CRITERIA DESCRIBING PSI
C
      DIMENSION N(50),FN(50),FNX(50),PHAT(50),QHAT(50),FN2(50)
      DIMENSION  NFMIN(IVCOL,IVSIZE),NFMAX(IVCOL,IVSIZE)
      COMMON /QVAR/N,FN,FNX,PHAT,QHAT,IFLAG,NVOL,IVS,QDMM,IVC
C
      IERR = 0
      IERR1 = 0
      IERRT = 0
      NCOMP = NCOMD
      PRINT 10
   10 FORMAT (1H1//39X,47HCOMPARISON CHECKS ON Q VALUES FOR MINMAX LIMIT
     1S//)
      DO 20 I=1,NCOMP
      FN2(I) = 1.0/(FN(I)+2.0)
      PHAT(I) = (FNX(I)+1.0)*FN2(I)
   20 CONTINUE
      QLIM = QTHETA(PHAT)
      PRINT 15, QLIM
   15 FORMAT (30X,44HQ VALUE USED FOR INDEX SET CRITERIA, QLIM = ,
     1E20.12//21X,4HKVOL,12X,4HQMIN,14X,9HQLIM-QMIN,13X,4HQMAX,14X,
     29HQLIM-QMAX/)
      KVOL = 0
      MFLAG = 0
      IECI = 1
      NWORDS = IVCOL*IVSIZE
   30 IF (MFLAG.EQ.IFLAG) GO TO 50
      IVEND = IVSIZE
   40 CALL READEC(NFMIN,IECI,NWORDS)
      IECI = IECI + NWORDS
      CALL READEC(NFMAX,IECI,NWORDS)
      IECI = IECI + NWORDS
      MFLAG = MFLAG + 1
      GO TO 60
   50 IF (NVOL.EQ.0) RETURN
      IVEND = NVOL
      IF (IFLAG.EQ.0) GO TO 60
      NWORDS = IVCOL*NVOL
      GO TO 40
   60 DO 100 IVOL=1,IVEND
      KVOL = KVOL + 1
      DO 70 I=1,NCOMP
      IF (NFMIN(I,IVOL).GT.N(I)) IERR1 = 1
      AP1 = NFMIN(I,IVOL) + 1
      PHAT(I) = AP1*FN2(I)
   70 CONTINUE
      QMIN = QTHETA(PHAT)
      DMIN = QLIM - QMIN
      IF (DMIN.GE. 0.0) GO TO 75
      IERR = 1
      IVR = KVOL
```

```
 75 DO 80 I=1,NCOMP                                                      CONLIM
    IF (NFMAX(I,IVOL).GT.N(I)) IERR1 = 1                                 CONLIM
    AP1 = NFMAX(I,IVOL) + 1                                              CONLIM
    PHAT(I) = AP1*FN2(I)                                                 CONLIM
 80 CONTINUE                                                             CONLIM
    QMAX = QTHETA(PHAT)                                                  CONLIM
    DMAX = QLIM - QMAX                                                   CONLIM
    IF (DMAX.GE. 0.0) GO TO 85                                           CONLIM
    IERR = 1                                                             CONLIM
    IVR = KVOL                                                           CONLIM
 85 PRINT 90, KVOL,QMIN,DMIN,QMAX,DMAX                                   CONLIM
 90 FORMAT (20X,I4,5X,E20.12,5X,E10.3,5X,E20.12,5X,E10.3)                CONLIM
    IF (IERR1.EQ.0) GO TO 100                                            CONLIM
    PRINT 92                                                             CONLIM
 92 FORMAT (15X,10H**********)                                           CONLIM
    DO 96 I=1,NCOMP                                                      CONLIM
    PRINT 94, I,NFMIN(I,IVOL),NFMAX(I,IVOL)                              CONLIM
 94 FORMAT (20X,4HI = ,I2,5X,6HMIN = ,I4,5X,6HMAX = ,I4)                 CONLIM
 96 CONTINUE                                                             CONLIM
    IERR1 = 0                                                            CONLIM
    IERRT = IERRT + 1                                                    CONLIM
    PRINT 92                                                             CONLIM
100 CONTINUE                                                             CONLIM
    IF (IFLAG.EQ.0) GO TO 110                                            CONLIM
    IF (IVEND.LT.IVSIZE) GO TO 110                                       CONLIM
    GO TO 30                                                             CONLIM
110 IF (IERR.EQ.1 .OR. IERRT.GT.0) GO TO 130                             CONLIM
    PRINT 120                                                            CONLIM
120 FORMAT (//30X,36HHYPERVOLUME CHECK INDICATES GOOD SET)               CONLIM
    RETURN                                                               CONLIM
130 PRINT 140, IVR                                                       CONLIM
140 FORMAT (//20X,40HHYPERVOLUME CHECK INDICATES THAT VOLUME   ,I4,23H   CONLIM
   1IMPROPERLY CONSTRUCTED)                                              CONLIM
    RETURN                                                               CONLIM
    END                                                                  CONLIM
```

```
      SUBROUTINE FIT(NCOMD,ALPHA,QOFP,NFMIN,NFMAX,IVSIZE,IVCOL,IERR)      CONLIM
C                                                                         CONLIM
C     ROUTINE MATCHES THE Q CURVES ALONG THE H CURVE SO THAT THE          CONLIM
C     MAXIMUM Q VALUE CAN BE OBTAINED SATISFYING THE ALPHA-UPPER          CONLIM
C     CONFIDENCE LIMIT.                                                   CONLIM
C                                                                         CONLIM
C                                                                         CONLIM
C     DELTA SHIFT VALUES ARE WEIGHTED ACCORDING TO COMPONENT POSITION     CONLIM
C              NTYPE = 0 , SERIES CONNECTION (DELSER)                     CONLIM
C                    = 1 , PARALLEL CONNECTION (DELPAR = SQRT(DELSER) )   CONLIM
C                                                                         CONLIM
      INTEGER  OPT                                                        CONLIM
      DIMENSION N(50),FN(50),FNX(50),PHAT(50),QHAT(50),FACT(50),          CONLIM
     1PSAVE(50),DELH(50),NTYPE(50),PORIG(50),QORIG(50),QM(50),OPT(6),     CONLIM
     2PTIL(50),PLAST(50),ID(10),PFRZ(50),VAL(11),SAVAL(10),IDINV(10)      CONLIM
      DIMENSION  NFMIN(IVCOL,IVSIZE),NFMAX(IVCOL,IVSIZE)                  CONLIM
      COMMON /QVAR/N,FN,FNX,PHAT,QHAT,IFLAG,NVOL,IVS,QP,IVC,OPT           CONLIM
      COMMON /FACTR/ FACT                                                 CONLIM
      COMMON /TYPE/NTYPE                                                  CONLIM
      COMMON /FIRST/ IFIRST,DELSER,IFREEZ,NIDS,ID,NLAST,IVAL,VAL,         CONLIM
     1SAVAL,IDINV                                                         CONLIM
      DATA  DELMIN/1.0E-06/                                               CONLIM
C                                                                         CONLIM
      ONEMAL = 1.0 - ALPHA                                                CONLIM
      NCOMP = NCOMD                                                       CONLIM
      IERR = 1                                                            CONLIM
      ITHRU = 1                                                           CONLIM
      ITOTM = 0                                                           CONLIM
      NSER = 0                                                            CONLIM
      DO 10 I=1,NCOMP                                                     CONLIM
      IF (NTYPE(I).EQ.1) GO TO 10                                         CONLIM
      NSER = NSER + 1                                                     CONLIM
   10 CONTINUE                                                            CONLIM
      NSER = MIN0(50,10*NSER)                                             CONLIM
      DELSER = 0.0                                                        CONLIM
      IFIRST = 1                                                          CONLIM
      IF (IFREEZ.GT.0) GO TO 760                                          CONLIM
      DO 20 I=1,NCOMP                                                     CONLIM
      QHAT(I) = 1.0- PHAT(I)                                              CONLIM
      PORIG(I) = PHAT(I)                                                  CONLIM
      QORIG(I) = QHAT(I)                                                  CONLIM
   20 CONTINUE                                                            CONLIM
      IF (OPT(1).EQ.0) GO TO 38                                           CONLIM
      PRINT 25                                                            CONLIM
   25 FORMAT (1H1//)                                                      CONLIM
      PRINT 30, (I,PHAT(I), I=1,NCOMP)                                    CONLIM
   30 FORMAT (5X,4HI = ,I3,5X,10HPHAT(I) = ,E15.7)                        CONLIM
      GO TO 38                                                            CONLIM
   32 ITHRU = 1                                                           CONLIM
      ITOTM = 0                                                           CONLIM
      DELSER = 0.5*DELSER                                                 CONLIM
      DEL2 = 0.5*DELSER                                                   CONLIM
      DELSAV = 0.5*DELSAV                                                 CONLIM
      DELPAR = SQRT(DELSER)                                               CONLIM
```

```
      DO 35 I=1,NCOMP                                                   CONLIM
      PHAT(I) = PORIG(I)                                                CONLIM
      QHAT(I) = QORIG(I)                                                CONLIM
   35 CONTINUE                                                          CONLIM
   38 HVALUE = HFUN(NCOMP,NFMIN,NFMAX,IVSIZE,IVCOL)                     CONLIM
      IF (HVALUE - ONEMAL) 36,37,37                                     CONLIM
   36 ISTAT = -1                                                        CONLIM
      GO TO 39                                                          CONLIM
   37 ISTAT = 1                                                         CONLIM
   39 ISTATC = 1                                                        CONLIM
   40 IF (OPT(1).EQ.0) GO TO 45                                         CONLIM
      PRINT 9010, HVALUE                                                CONLIM
 9010 FORMAT (5X,29HHVALUE STARTING OFF DELTAP = ,E15.8)                CONLIM
   45 DO 50 I=1,NCOMP                                                   CONLIM
      DELH(I) = 0.0                                                     CONLIM
      QHAT(I) = 1.0 - PHAT(I)                                           CONLIM
   50 CONTINUE                                                          CONLIM
      XNUM = ONEMAL - HVALUE                                            CONLIM
C                                                                       CONLIM
C     CHECK VERY FIRST STARTING VALUE TO SEE IF TOO FAR REMOVED         CONLIM
C     FROM CURVE FOR ANY CONVERGENCE POSSIBILITY                        CONLIM
C                                                                       CONLIM
      IF (ITHRU.GT.1) GO TO 103                                         CONLIM
      IF (ABS(XNUM) - 0.5*ONEMAL) 100,100,80                            CONLIM
   80 IF (XNUM) 187,103,189                                             CONLIM
  100 IF (XNUM) 103,103,105                                             CONLIM
  103 IF (ISTAT.EQ.1) GO TO 108                                         CONLIM
      ISTATC = ISTATC + 1                                               CONLIM
      ISTAT = 1                                                         CONLIM
      IF (ISTATC .LT. 5) GO TO 108                                      CONLIM
      DELSER = 0.5*DELSER                                               CONLIM
      DELSAV = 0.5*DELSAV                                               CONLIM
      DELPAR = SQRT(DELSER)                                             CONLIM
      ISTATC = 1                                                        CONLIM
      GO TO 108                                                         CONLIM
  105 IF (ISTAT.EQ.-1) GO TO 110                                        CONLIM
      ISTATC = ISTATC + 1                                               CONLIM
      ISTAT = -1                                                        CONLIM
      GO TO 110                                                         CONLIM
  108 IF (XNUM) 110,180,110                                             CONLIM
  110 CALL DELTA2(NCOMP,A2,NFMIN,NFMAX,IVSIZE,IVCOL)                    CONLIM
      DO 120 I=1,NCOMP                                                  CONLIM
      PSAVE(I) = PHAT(I)*(1.0 - XNUM/A2)                                CONLIM
  120 CONTINUE                                                          CONLIM
  140 DO 160 I=1,NCOMP                                                  CONLIM
      PHAT(I) = PSAVE(I)                                                CONLIM
      QHAT(I) = 1.0 - PSAVE(I)                                          CONLIM
  160 CONTINUE                                                          CONLIM
  180 IF (OPT(1).EQ.0) GO TO 185                                        CONLIM
      PRINT 9030, (I,PHAT(I),QHAT(I),I=1,NCOMP)                         CONLIM
 9030 FORMAT (5X,4HI = ,I2,2X,10HPHAT(I) = ,E15.7,2X,10HQHAT(I) = ,E15.7 CONLIM
     1)                                                                 CONLIM
  185 DO 190 I=1,NCOMP                                                  CONLIM
      IF (PHAT(I).LE. 0.0) GO TO 191                                    CONLIM
```

```
      IF (PHAT(I).GE. 1.0) GO TO 191                                   CONLIM   945
  190 CONTINUE                                                         CONLIM   946
      HVALUE = HFUN(NCOMP,NFMIN,NFMAX,IVSIZE,IVCOL)                    CONLIM   947
      HSAVE = HVALUE                                                   CONLIM   948
      QP = QTHETA(PSAVE)                                               CONLIM   949
      ITOTM = ITOTM + 1                                                CONLIM   950
      IF (ITOTM.GT.NSER) GO TO 1900                                    CONLIM   951
      QM(ITOTM) = QP                                                   CONLIM   952
      GO TO 1940                                                       CONLIM   953
C                                                                      CONLIM   954
C     LONG PERIOD FOR THIS STEPSIZE - ARE WE PROGRESSING TO MAXIMUM    CONLIM   955
C                                                                      CONLIM   956
 1900 IF ( ABS(QM(NSER)-QM(1)) - DEL2) 530,530,1920                    CONLIM   957
 1920 ISTOP = NSER-1                                                   CONLIM   958
      DO 1930 I=1,ISTOP                                                CONLIM   959
 1930 QM(I) = QM(I+1)                                                  CONLIM   960
      QM(NSER) = QP                                                    CONLIM   961
 1940 IF (OPT(1).EQ.0) GO TO 213                                       CONLIM   962
      PRINT 9020, HVALUE                                               CONLIM   963
 9020 FORMAT (5X,28HHVALUE AFTER DELTAP SHIFT = ,E20.12)               CONLIM   964
      PRINT 9040, QP                                                   CONLIM   965
 9040 FORMAT (5X,30HQP VALUE AFTER DELTAP SHIFT = ,E20.12)             CONLIM   966
      GO TO 213                                                        CONLIM   967
  191 IF (ITHRU.GT.1) GO TO 32                                         CONLIM   968
      IF (XNUM) 187,213,189                                            CONLIM   969
C                                                                      CONLIM   970
C     STARTING HVALUE TOO FAR REMOVED FROM 1-ALPHA CURVE -             CONLIM   971
C     BISECTION USED TO IMPROVE INITIAL GUESS FOR FIT.                 CONLIM   972
C                                                                      CONLIM   973
  187 DO 188 I=1,NCOMP                                                 CONLIM   974
      PLAST(I) = 1.0                                                   CONLIM   975
  188 PHAT(I) = PORIG(I)                                               CONLIM   976
      ICHKCT = 0                                                       CONLIM   977
      IDIR = 1                                                         CONLIM   978
      GO TO 201                                                        CONLIM   979
  189 DO 192 I=1,NCOMP                                                 CONLIM   980
      PLAST(I) = 0.0                                                   CONLIM   981
  192 PHAT(I) = PORIG(I)                                               CONLIM   982
      ICHKCT = 0                                                       CONLIM   983
      IDIR = -1                                                        CONLIM   984
C                                                                      CONLIM   985
C     MOVE TOWARD ZERO                                                 CONLIM   986
C                                                                      CONLIM   987
  193 ICHKCT = ICHKCT + 1                                              CONLIM   988
      IF (IDIR.EQ.1) GO TO 195                                         CONLIM   989
      DO 194 I=1,NCOMP                                                 CONLIM   990
  194 PTIL(I) = PHAT(I)                                                CONLIM   991
      GO TO 197                                                        CONLIM   992
  195 DO 196 I=1,NCOMP                                                 CONLIM   993
      PLAST(I) = PTIL(I)                                               CONLIM   994
  196 PTIL(I) = PHAT(I)                                                CONLIM   995
  197 DO 198 I=1,NCOMP                                                 CONLIM   996
      PHAT(I) = PTIL(I) - 0.5*(PTIL(I)-PLAST(I))                       CONLIM   997
  198 QHAT(I) = 1.0 - PHAT(I)                                          CONLIM   998
```

```
      IDIR = -1
      GO TO 207
C
C     MOVE TOWARD ONE
C
  201 ICHKCT = ICHKCT + 1
      IF (IDIR.EQ.1) GO TO 203
      DO 202 I=1,NCOMP
      PLAST(I) = PTIL(I)
  202 PTIL(I) = PHAT(I)
      GO TO 205
  203 DO 204 I=1,NCOMP
  204 PTIL(I) = PHAT(I)
  205 DO 206 I=1,NCOMP
      PHAT(I) = PTIL(I) + 0.5*(PLAST(I)-PTIL(I))
  206 QHAT(I) = 1.0 - PHAT(I)
      IDIR = 1
  207 IF (OPT(1).EQ.0) GO TO 2070
      DO 9207 I=1,NCOMP
      PRINT 9206, I,PHAT(I)
 9206 FORMAT (20X,I2,5X,E15.7)
 9207 CONTINUE
 2070 DO 208 I=1,NCOMP
      IF (PHAT(I).LE. 0.0) GO TO 201
  208 CONTINUE
      DO 209 I=1,NCOMP
      IF (PHAT(I).GE. 1.0) GO TO 193
  209 CONTINUE
      HCHECK = HFUN(NCOMP,NFMIN,NFMAX,IVSIZE,IVCOL) - ONEMAL
      IF (OPT(1).EQ.0) GO TO 2090
      PRINT 9210, HCHECK
 9210 FORMAT (20X,9HHCHECK = ,E15.7/)
 2090 IF (ABS(HCHECK) - 0.5*ONEMAL) 211,211,210
  210 IF (HCHECK) 193,211,201
  211 IF (OPT(1).EQ.0) GO TO 38
      PRINT 212, ICHKCT
  212 FORMAT (10X,I3,36H MOVES WITH BISECTION WERE PERFORMED/)
      GO TO 38
  213 IF (IFREEZ.GT.0) GO TO 2110
      CALL SLACK(NCOMP,PHAT,ISLACK)
      IF (IFIRST.EQ.2) GO TO 700
 2110 PHATSV = PSAVE(ISLACK)
      IF (OPT(1).EQ.0) GO TO 2130
      PRINT 2120,ISLACK
 2120 FORMAT (10X,9HCOMPONENT,I4,28H USED TO MAINTAIN CONSTANT Q)
 2130 IREG = NCOMP + 1
      IF (ITHRU.GT.1) GO TO 214
      CALL SMDELT(NCOMP,PSAVE,DELSER)
      DEL2 = 0.5*DELSER
      DELSAV = DELSER
      DELPAR = SQRT(DELSER)
      ITHRU = 2
      GO TO 215
  214 IF (ITHRU.EQ.3) GO TO 215
```

```
      DELSER = DELSAV
      DEL2 = 0.5*DELSER
      DELPAR = SQRT(DELSER)
  215 IF (DELSER - DELMIN) 600,220,220
  220 IREG = IREG - 1
      IF (IREG.EQ.0) GO TO 500
      IF (IREG.EQ.ISLACK) GO TO 220
      IF (NTYPE(IREG).EQ.1) GO TO 225
      DEL = DELSER
      GO TO 228
  225 DEL = DELPAR
  228 PHAT(IREG) = PSAVE(IREG) + DEL
      QHAT(IREG) = 1.0 - PHAT(IREG)
      NEG = 0
      IF (PHAT(IREG).GE. 1.0) GO TO 270
  230 THETA = (QTHETA(PHAT)-PHAT(ISLACK))/QHAT(ISLACK)
      PTRY = (QP - THETA)/(1.0 - THETA)
      IF(PTRY.LE. 0.0) GO TO 260
      IF (PTRY .GE. 1.0) GO TO 260
      PHAT(ISLACK) = PTRY
      QHAT(ISLACK) = 1.0 - PTRY
      H = HFUN(NCOMP,NFMIN,NFMAX,IVSIZE,IVCOL)
      IF (H - HSAVE) 260,400,400
  260 IF (NEG.EQ.1) GO TO 300
C
C     MAY HAVE MOVED IN WRONG DIRECTION - TRY OTHER WAY
C
  270 PHAT(IREG) = PSAVE(IREG) - DEL
      QHAT(IREG) = 1.0 - PHAT(IREG)
      NEG = 1
      IF (PHAT(IREG)) 300,300,230
C
C     CANNOT ACCEPT NEW P(IREG) VALUE - EITHER DEL STEP WAS TOO LARGE
C     TO MAINTAIN A CONSTANT Q VALUE OR H VALUE FOR NEW P(IREG) WAS
C     ON WRONG SIDE OF CURVE (MAINTAIN VALUES OF H .GT. 1-ALPHA).
C     IN EITHER CASE, TRY A SMALLER DEL.
C
  300 DELH(IREG) = 0.0
      PHAT(IREG) = PSAVE(IREG)
      PHAT(ISLACK) = PHATSV
      QHAT(IREG) = 1.0 - PHAT(IREG)
      QHAT(ISLACK) = 1.0 - PHAT(ISLACK)
      GO TO 220
C
C     ACCEPTABLE P(IREG) VALUE - MAINTAINED CONSTANT Q AND VALUE OF H
C     ON PROPER SIDE OF CURVE.
C
C     P VALUES ARE HELD BACK AWAY FROM ABSOLUTE BOUNDARY - WILL ASSUME
C     VALUES VERY CLOSE HOWEVER.  PREVENTS H FUNCTION FROM BECOMING
C     INDEFINITE OR INFINITE.
C
  400 DELH(IREG) = H - HSAVE
      IF (PHAT(IREG) - 1.0) 420,300,300
  420 IF (PHAT(IREG).LE.0.0) GO TO 300
```

```
      HSAVE = H                                                         CONLIM
      PHATSV = PHAT(ISLACK)                                             CONLIM
      GO TO 220                                                         CONLIM
C                                                                       CONLIM
C     COMPLETED DEL ADJUSTMENT OF ALL P VALUES - NOW DETERMINE WHAT     CONLIM
C     FURTHER ADJUSTMENTS SHOULD BE MADE.                               CONLIM
C                                                                       CONLIM
  500 IF (OPT(1).EQ.0) GO TO 510                                        CONLIM
      PRINT 9060, DELSER                                                CONLIM
 9060 FORMAT (5X,18HDEL VALUE TRIED = ,E15.7)                           CONLIM
      PRINT 9030, (I,PHAT(I),QHAT(I),I=1,NCOMP)                         CONLIM
  510 DO 520 I=1,NCOMP                                                  CONLIM
      IF (DELH(I)) 550,520,550                                          CONLIM
  520 CONTINUE                                                          CONLIM
  530 ITHRU = 3                                                         CONLIM
      PHAT(ISLACK) = PSAVE(ISLACK)                                      CONLIM
      QHAT(ISLACK) = 1.0 - PSAVE(ISLACK)                                CONLIM
      DELSER = 0.5*DELSER                                               CONLIM
      DELPAR=SQRT(DELSER)                                               CONLIM
      ITOTM = 0                                                         CONLIM
      DEL2 = 0.5*DELSER                                                 CONLIM
      IF (DELSER - DELMIN) 600,213,213                                  CONLIM
  550 HVALUE = HSAVE                                                    CONLIM
      GO TO 40                                                          CONLIM
C                                                                       CONLIM
C     COMPLETED MATCHING OF THE Q AND H CURVES - WE HAVE REACHED THE    CONLIM
C     MINIMUM DEL SET FOR CONVERGENCE CRITERION                         CONLIM
C                                                                       CONLIM
  600 DO 620 I=1,NCOMP                                                  CONLIM
      PHAT(I) = PSAVE(I)                                                CONLIM
      QHAT(I) = 1.0- PSAVE(I)                                           CONLIM
  620 CONTINUE                                                          CONLIM
      QOFP = QTHETA(PHAT)                                               CONLIM
      IF (PHAT(ISLACK) .LE. (2.0*DELMIN))  IERR = 2                     CONLIM
      RETURN                                                            CONLIM
C                                                                       CONLIM
C     OSCILLATION IN FIT - CHECK FOR FREEZE OPTION ON COMPONENTS        CONLIM
C                                                                       CONLIM
  700 IF (OPT(6).NE.0) GO TO 720                                        CONLIM
      IFREEZ = NIDS                                                     CONLIM
      IERR = 3                                                          CONLIM
      RETURN                                                            CONLIM
  720 IFREEZ = NIDS                                                     CONLIM
      DO 740 I=1,NCOMP                                                  CONLIM
      PFRZ(I) = PSAVE(I)                                                CONLIM
  740 CONTINUE                                                          CONLIM
      HFRZ = HSAVE                                                      CONLIM
      DELFRZ = DELSER                                                   CONLIM
      QPFRZ = QP                                                        CONLIM
      ITHUFZ = ITHRU                                                    CONLIM
      ISLACK = ID(IFREEZ)                                               CONLIM
      GO TO 2110                                                        CONLIM
  760 IFREEZ = IFREEZ-1                                                 CONLIM
      IF (IFREEZ.EQ.0) RETURN                                           CONLIM
```

```
      ISLACK = ID(IFREEZ)
      DO 780 I=1,NCOMP
      PSAVE(I) = PFRZ(I)
  780 CONTINUE
      HSAVE = HFRZ
      DELSER = DELFRZ
      DELSAV = DELFRZ
      QP = QPFRZ
      ITHRU = ITHUFZ
      GO TO 2110
      END
```

```
      SUBROUTINE SLACK(NCOMP,PHAT,ISLACK)
C
C     DETERMINE WHICH SERIES COMPONENT CURRENTLY HAS THE LARGEST PHAT
C     VALUE AND SET ISLACK EQUAL TO THE COMPONENT INDEX FOR USE IN
C     THE FIT ROUTINE.  MAINTAIN A SURVEILANCE OVER THE INDICES CHOSEN
C     TO DIAGNOSE CYCLING.  TWO CHECKS ARE MADE, THE FIRST TO CATCH
C     REPETITION AND THE SECOND TO DETERMINE IF ACTUAL STEP IMPROVE-
C     MENT IS STILL PROGRESSING OR HALTED.
C
      DIMENSION PHAT(NCOMP),NTYPE(50),ID(10),VAL(11),SAVAL(10),SAVT(10),
     1IDINV(10)
      COMMON  /FIRST/ IFIRST,DEL,IFREEZ,N,ID,NLAST,IVAL,VAL,SAVAL,IDINV
      COMMON /TYPE/ NTYPE
C
C     FIND MAXIMUM COMPONENT (SERIES ONLY)
C
      PMAX = 0.0
      DO 20 I=1,NCOMP
      IF (NTYPE(I) .EQ. 1) GO TO 20
      IF (PHAT(I) .LT. PMAX) GO TO 20
      ISLACK = I
      PMAX = PHAT(I)
   20 CONTINUE
      IF (IFIRST.EQ.1) GO TO 400
C
C     BEGIN PRIMARY CYCLE CHECK
C
      IF (ISLACK.NE.IDINV(NLAST)) GO TO 100
      IF (N.EQ.1) GO TO 400
      IDEX = N-NLAST+1
      SAVAL(IDEX) = PMAX
      IF (NLAST.EQ.N) GO TO 30
      IF (NLAST.NE.1) GO TO 35
      NLAST = N
      GO TO 40
   30 IVAL = IVAL+1
      VAL(IVAL) = PMAX
   35 NLAST = NLAST-1
   40 IFLAG = IFLAG+1
      IF ( (IFLAG/N) .LT.5 ) RETURN
C
C     COMPLETE CYCLE PERIOD ENCOUNTERED - PERFORM SECONDARY CHECK
C     ON PROGRESS IN CONVERGENCE
C
      IVEND = IVAL-1
      DO 50 J=1,IVEND
      IF (ABS(VAL(J+1)-VAL(J)) - 0.5*DEL) 50,50,60
   50 CONTINUE
C
C     NO PROGRESS MADE - WARN FIT ROUTINE AND USER
C
      IFIRST = 2
      RETURN
C
```

```
C     SOME PROGRESS MADE - KNOCK OFF FIRST CYCLE AND CONTINUE
C     WITHIN THE PERIOD
C
   60 NLAST = N
      IFLAG = IFLAG-N
      IVAL = IVAL-1
      DO 70 I=1,IVAL
      VAL(I) = VAL(I+1)
   70 CONTINUE
      RETURN
C
C     SLACK COMPONENT DID NOT MATCH THE EXPECTED COMPONENT OF THE CYCLE.
C     IF ONLY ONE COMPONENT IN CYCLE THEN WE ARE BUILDING CYCLE
C     AND WE HAVE A NEW COMPONENT.
C     IF MORE THAN ONE COMPONENT THEN WE MAY HAVE INTERRUPTED THE CYCLE
C     AND THE PERIOD, AND A NEW CYCLE WILL BE FORMED.
C
  100 IF (N.EQ.1) GO TO 300
      INEW = 0
      NSTOP = N-1
      DO 120 II=1,NSTOP
      I = II
      IF (ISLACK.EQ.ID(I)) GO TO 160
  120 CONTINUE
      IF (ISLACK.EQ.ID(N)) GO TO 400
C
C     ISLACK AN ENTIRELY NEW COMPONENT - NOW CHECK IF FIRST CYCLE
C     STILL BEING FORMED
C
      IF (IVAL.EQ.1) GO TO 300
C
C     NEW COMPONENT FOR NEW CYCLE - INTERRUPTED OSCILLATION -- REFRESH
C
      I = N-NLAST
      NNEW = N+1
      IVAL = 1
      VAL(1) = SAVAL(NNEW-NLAST)
      GO TO 200
C
C     CYCLE COMPONENT OUT OF PHASE - INTERRUPTED OSCILLATION
C     PICK UP NEW CYCLE WITH COMPONENT IMMEDIATELY FOLLOWING MATCH
C
  160 NNEW = (N-I) + (N-NLAST+1)
      IVAL = 1
      VAL(1) = SAVAL(I+1)
      IF (NNEW.LT.N) GO TO 200
C
C     SPLIT IN OLD CYCLE INTERRUPTED IN MIDDLE WITH DUPLICATE
C     COMPONENT IN SAME CYCLE - NO PARTS IN PREVIOUS CYCLE
C
      NNEW = NNEW-N
      IF (NNEW.EQ.1) GO TO 290
      INEW = 1
C
```

```
C     NEW CYCLE CREATION USING PARTS OF PREVIOUS CYCLES -- NEW          CONLIM
C     REGENERATE BOOKKEEPING                                            CONLIM
C                                                                       CONLIM
  200 DO 220 J=1,N                                                      CONLIM
      JJ = N-J+1                                                        CONLIM
  220 SAVT(J) = SAVAL(JJ)                                               CONLIM
      IF (INEW.EQ.0) GO TO 230                                          CONLIM
      JSTOP = NNEW-1                                                    CONLIM
      GO TO 235                                                         CONLIM
  230 JSTOP = N-NLAST                                                   CONLIM
      IF (JSTOP.EQ.0) GO TO 250                                         CONLIM
C                                                                       CONLIM
C     PICK UP COMPONENTS FROM FIRST PART OF INTERRUPTED CYCLE           CONLIM
C                                                                       CONLIM
  235 DO 240 J=1,JSTOP                                                  CONLIM
      JNEW = NNEW-J                                                     CONLIM
      JJ = NLAST+J                                                      CONLIM
      ID(JNEW) = IDINV(JJ)                                              CONLIM
  240 SAVAL(JNEW) = SAVT(JJ)                                            CONLIM
      IF (INEW.EQ.1) GO TO 290                                          CONLIM
      GO TO 260                                                         CONLIM
C                                                                       CONLIM
C     EVERYTHING FOR NEW CYCLE FROM PREVIOUS COMPLETED CYCLE            CONLIM
C                                                                       CONLIM
  250 JNEW = NNEW                                                       CONLIM
C                                                                       CONLIM
C     PICK UP REMAINDER OF NEW CYCLE FROM PREVIOUS COMPLETED CYCLE      CONLIM
C                                                                       CONLIM
  260 JSTOP = N-I                                                       CONLIM
      DO 280 J=1,JSTOP                                                  CONLIM
      JNEW = JNEW-1                                                     CONLIM
      ID(JNEW) = IDINV(J)                                               CONLIM
  280 SAVAL(JNEW) = SAVT(J)                                             CONLIM
  290 N = NNEW                                                          CONLIM
      IFLAG = N                                                         CONLIM
      NLAST = N                                                         CONLIM
      SAVAL(N) = PMAX                                                   CONLIM
      ID(N) = ISLACK                                                    CONLIM
      DO 295 J=1,N                                                      CONLIM
      JJ = N-J+1                                                        CONLIM
  295 IDINV(J) = ID(JJ)                                                 CONLIM
      RETURN                                                            CONLIM
C                                                                       CONLIM
C     NEW MEMBER OF CYCLE -- CON                                        CONLIM
C                                                                       CONLIM
  300 N = N+1                                                           CONLIM
      ID(N) = ISLACK                                                    CONLIM
      SAVAL(N) = PMAX                                                   CONLIM
      IFLAG = IFLAG+1                                                   CONLIM
      DO 320 II=1,NLAST                                                 CONLIM
      I = N-II+1                                                        CONLIM
  320 IDINV(I) = IDINV(I-1)                                             CONLIM
      IDINV(1) = ISLACK                                                 CONLIM
      NLAST = N                                                         CONLIM
```

```
      RETURN
C
C     RESET ALL VALUES TO INITIAL - NEW CYCLE OF A NEW PERIOD
C
  400 IFIRST = 0
      N = 1
      IFLAG = 1
      NLAST = 1
      ID(1) = ISLACK
      IDINV(1) = ISLACK
      IVAL = 1
      VAL(1) = PMAX
      SAVAL(1) = PMAX
      RETURN
      END
```

```
      SUBROUTINE SUB(EX,X,Y)                                            CONLIM
C                                                                       CONLIM
C     THIS SUBROUTINE IS CALLED VIA THE CALL TO RECOVR IN THE MAINLINE  CONLIM
C     WHEN A TIME LIMIT OCCURS.  INFORMATION AVAILABLE TO THE USER      CONLIM
C     AT THAT TIME IS PRINTED.                                          CONLIM
C                                                                       CONLIM
      DIMENSION N(50),FN(50),FNX(50),PHAT(50),QHAT(50)                  CONLIM
      COMMON /QVAR/ N,FN,FNX,PHAT,QHAT,IFLAG,IVOL,IVS,QDMM,IVCOL        CONLIM
      COMMON /PSIFLG/ IPSIFG                                            CONLIM
      ITOT = IFLAG*IVS + IVOL                                           CONLIM
      PRINT 20, IFLAG,IVOL,ITOT                                         CONLIM
   20 FORMAT (1H1///10X,21HRUN WAS NOT COMPLETED//10X,8HIFLAG = ,I10,   CONLIM
     15X,7HIVOL = ,I10//10X,57HTOTAL NUMBER OF HYPERVOLUMES SELECTED AT CONLIM
     2TIME OF ABORT = ,I6)                                              CONLIM
      IF (IPSIFG.EQ.1) GO TO 30                                         CONLIM
      PRINT 25                                                          CONLIM
   25 FORMAT (//10X,61HSUBROUTINE PSI HAD NOT COMPLETED HYPERVOLUME SET CONLIM
     1CONSTRUCTION)                                                     CONLIM
      GO TO 38                                                          CONLIM
   30 PRINT 35                                                          CONLIM
   35 FORMAT (//10X,57HHYPERVOLUME SET CONSTRUCTION WAS COMPLETED PRIOR CONLIM
     1TO ABORT)                                                         CONLIM
   38 PRINT 40, QDMM                                                    CONLIM
   40 FORMAT (///10X,37HVALUE OF SYSTEM Q AT TIME OF ABORT = ,E15.7)    CONLIM
      STOP                                                              CONLIM
      END                                                               CONLIM
```

```
      FUNCTION HFUN(NCOMD,NFMIN,NFMAX,IVSIZE,IVCOL)
      DIMENSION N(50),FN(50),FNX(50),PHAT(50),QHAT(50),FACT(50)
      DIMENSION  NFMIN(IVCOL,IVSIZE),NFMAX(IVCOL,IVSIZE)
      COMMON /QVAR/N,FN,FNX,PHAT,QHAT,IFLAG,NVOL,IVS,QDMM,IVC
      COMMON / FACTR/ FACT
      DATA CONST/ 8.1061466795328E-02/
C
C     CALCULATE THE ENTIRE H FUNCTION USING HYPERVOLUMES.
C     THE PROBABILITY VALUES P OF THE FUNCTION ARE OBTAINED FROM
C     INCREMENTAL MOVES (SUBROUTINE FIT).
C
      NCOMP = NCOMD
      HFUN = 0.0
      MFLAG = 0
      IECI = 1
      NWORDS = IVCOL*IVSIZE
    5 IF (MFLAG.EQ.IFLAG) GO TO 10
      IVEND = IVSIZE
    8 CALL READEC(NFMIN,IECI,NWORDS)
      IECI = IECI + NWORDS
      CALL READEC(NFMAX,IECI,NWORDS)
      IECI = IECI + NWORDS
      MFLAG = MFLAG + 1
      GO TO 15
   10 IF (NVOL.EQ.0) RETURN
      IVEND = NVOL
      IF (IFLAG.EQ.0) GO TO 15
      NWORDS = IVCOL*NVOL
      GO TO 8
   15 DO 300 IVOL=1,IVEND
      HPROD = 1.0
      DO 200 ICOMP=1,NCOMP
      NI = N(ICOMP)
      HSUM = 0.0
      ISTRT = NFMIN(ICOMP,IVOL) + 1
      IEND = NFMAX(ICOMP,IVOL) + 1
      RECFAC = PHAT(ICOMP)/QHAT(ICOMP)
      ISTRM = ISTRT-1
      DO 100 I=ISTRT,IEND
      IA = I-1
      AI = IA
      IF (IA.EQ.NI) GO TO 80
      IF (IA.EQ.0) GO TO 60
      IF (NI - 50) 20,20,40
C
C     UTILIZE STORED FACTORIALS
C
   20 NMIA = NI - IA
      H1 = FACT(NI)/(FACT(IA)*FACT(NMIA))
      IF (IA.GT.ISTRM) GO TO 16
      H2 = (PHAT(ICOMP)**IA)*(QHAT(ICOMP)**NMIA)
      GO TO 18
   16 H2 = H2*RECFAC
   18 HF = H1*H2
```

```
      GO TO 90                                                          CONLIM
C                                                                       CONLIM
C     USE STIRLING APPROXIMATION FOR LOGARITHM OF BINOMIAL COEFFICIENT  CONLIM
C                                                                       CONLIM
   40 FNI = FN(ICOMP)                                                   CONLIM
      RM1 = FNI - AI + 1                                                CONLIM
      IF (IA.GT.ISTRM) GO TO 50                                         CONLIM
      RM = RM1 - 1.0                                                    CONLIM
      XN1 = FNI + 1.0                                                   CONLIM
      R1 = 1.0/(AI+1.0)                                                 CONLIM
      ORM1 = 1.0/RM1                                                    CONLIM
      TN = 12.0*XN1                                                     CONLIM
      H2 = FNI*ALOG(XN1*QHAT(ICOMP)*ORM1)                               CONLIM
      HF = H2 + AI*ALOG(RM1*RECFAC*R1)                                  CONLIM
      H2 = HF + 0.5*ALOG(XN1*R1*ORM1) + CONST                           CONLIM
      HF = EXP(H2 + (AI*RM - XN1*XN1)*R1*ORM1/TN)                       CONLIM
      GO TO 90                                                          CONLIM
   50 HF = HF*RECFAC*RM1/AI                                             CONLIM
      GO TO 90                                                          CONLIM
C                                                                       CONLIM
C     AI PARAMETER EQUAL ZERO -- PHAT TERM AND B.C. OUT                 CONLIM
C                                                                       CONLIM
   60 HF = QHAT(ICOMP)**NI                                              CONLIM
      H2 = HF                                                           CONLIM
      GO TO 90                                                          CONLIM
C                                                                       CONLIM
C     AI PARAMETER EQUAL MAXIMUM N(I) -- QHAT TERM AND B.C. OUT         CONLIM
C                                                                       CONLIM
   80 HF = PHAT(ICOMP)**IA                                              CONLIM
      H2 = HF                                                           CONLIM
C                                                                       CONLIM
C     ADD VALUE INTO SUM FOR FOR THIS DIMENSION OF VOLUME               CONLIM
C                                                                       CONLIM
   90 HSUM = HSUM + HF                                                  CONLIM
  100 CONTINUE                                                          CONLIM
      HPROD = HPROD*HSUM                                                CONLIM
  200 CONTINUE                                                          CONLIM
C                                                                       CONLIM
C     SUM UP CONTRIBUTION FROM ALL VOLUMES                              CONLIM
C                                                                       CONLIM
      HFUN = HFUN + HPROD                                               CONLIM
  300 CONTINUE                                                          CONLIM
      IF (IFLAG.EQ.0) RETURN                                            CONLIM
      IF (IVEND.LT.IVSIZE) RETURN                                       CONLIM
      GO TO 5                                                           CONLIM
      END                                                               CONLIM
```

```
      SUBROUTINE SMDELT(NCOMD,P,DEL)                                    CONLIM
C                                                                       CONLIM
C     DETERMINE THE INITIAL STEPSIZE DELTA                              CONLIM
C                                                                       CONLIM
      DIMENSION P(NCOMD)                                                CONLIM
      NCOM = NCOMD                                                      CONLIM
      XNCOM = NCOM                                                      CONLIM
      PSUM = 0.0                                                        CONLIM
      DO 20 I=1,NCOM                                                    CONLIM
      PSUM = PSUM + P(I)                                                CONLIM
   20 CONTINUE                                                          CONLIM
      AVE = PSUM/XNCOM                                                  CONLIM
      DEL = 0.1*AVE                                                     CONLIM
      RETURN                                                            CONLIM
      END                                                               CONLIM
```

```
      SUBROUTINE DELTA2(NCOMD,A2,NFMIN,NFMAX,IVSIZE,IVCOL)
C
C     ADJUSTMENT FACTOR FOR HYPERVOLUME SYSTEM
C
C     COMPUTE NECESSARY DELTA VALUE TO MOVE TO THE H CURVE FROM THE
C     CURRENT COORDINATES.  MOVE TOWARD CORNER (0,0,...,0).
C
      DIMENSION N(50),FN(50),FNX(50),PHAT(50),QHAT(50),FACT(50),XNP(50)
      DIMENSION  NFMIN(IVCOL,IVSIZE),NFMAX(IVCOL,IVSIZE)
      COMMON /QVAR/ N,FN,FNX,PHAT,QHAT,IFLAG,NVOL,IVS,QDMM,IVC
      COMMON /FACTR/ FACT
      DATA CONST/8.1061466795328E-02/
C
      A2 = 0.0
      NCOMP = NCOMD
      DO 20 I=1,NCOMP
      XNP(I) = FN(I)*PHAT(I)
   20 CONTINUE
      MFLAG = 0
      IECI = 1
      NWORDS = IVCOL*IVSIZE
    5 IF (MFLAG.EQ.IFLAG) GO TO 10
      IVEND = IVSIZE
    8 CALL READEC(NFMIN,IECI,NWORDS)
      IECI = IECI + NWORDS
      CALL READEC(NFMAX,IECI,NWORDS)
      IECI = IECI + NWORDS
      MFLAG = MFLAG + 1
      GO TO 15
   10 IF (NVOL.EQ.0) RETURN
      IVEND = NVOL
      IF (IFLAG.EQ.0) GO TO 15
      NWORDS = IVCOL*NVOL
      GO TO 8
   15 DO 500 IVOL=1,IVEND
      ASUM = 0.0
      DO 400 IM =1,NCOMP
      APROD = 1.0
      DO 300 IN=1,NCOMP
      AISUM = 0.0
      NI = N(IN)
      ISTRT = NFMIN(IN,IVOL) + 1
      IEND = NFMAX(IN,IVOL) + 1
      RECFAC = PHAT(IN)/QHAT(IN)
      ISTRM = ISTRT - 1
      DO 200 I= ISTRT,IEND
      IA = I-1
      AI = IA
      IF (IA.EQ.NI) GO TO 100
      IF (IA.EQ.0) GO TO 80
      IF (NI - 50) 40,40,60
C
C     UTILIZE STORED FACTORIALS
C
```

```
   40 NMIA = NI - IA                                                      CONLIM
      H1 = FACT(NI)/(FACT(IA)*FACT(NMIA))                                 CONLIM
      IF (IA.GT.ISTRM) GO TO 50                                           CONLIM
      H2 = (PHAT(IN)**IA)*(QHAT(IN)**NMIA)                                CONLIM
      GO TO 55                                                            CONLIM
   50 H2 = H2*RECFAC                                                      CONLIM
   55 HF = H1*H2                                                          CONLIM
      GO TO 120                                                           CONLIM
C                                                                         CONLIM
C     USE STIRLING APPROXIMATION FOR LOGARITHM OF BINOMIAL COEFFICIENT    CONLIM
C                                                                         CONLIM
   60 FNI = FN(IN)                                                        CONLIM
      RM1 = FNI - AI + 1                                                  CONLIM
      IF (IA.GT.ISTRM) GO TO 70                                           CONLIM
      RM = RM1 - 1.0                                                      CONLIM
      XN1 = FNI + 1.0                                                     CONLIM
      R1 = 1.0/(AI+1.0)                                                   CONLIM
      ORM1 = 1.0/RM1                                                      CONLIM
      TN = 12.0*XN1                                                       CONLIM
      H2 = FN1*ALOG(XN1*QHAT(IN)*ORM1)                                    CONLIM
      HF = H2 + AI*ALOG(RM1*RECFAC*R1)                                    CONLIM
      H2 = HF + 0.5*ALOG(XN1*R1*ORM1) + CONST                             CONLIM
      HF = EXP(H2 + (AI*RM - XN1*XN1)*R1*ORM1/TN)                         CONLIM
      GO TO 120                                                           CONLIM
   70 HF = HF*RECFAC*RM1/AI                                               CONLIM
      GO TO 120                                                           CONLIM
C                                                                         CONLIM
C     AI PARAMETER EQUAL ZERO -- PHAT TERM AND B.C. OUT                   CONLIM
C                                                                         CONLIM
   80 HF = QHAT(IN)**NI                                                   CONLIM
      H2 = HF                                                             CONLIM
      GO TO 120                                                           CONLIM
C                                                                         CONLIM
C     AI PARAMETER EQUAL MAXIMUM N(I) -- QHAT TERM AND B.C. OUT           CONLIM
C                                                                         CONLIM
  100 HF =PHAT(IN)**IA                                                    CONLIM
      H2 = HF                                                             CONLIM
C                                                                         CONLIM
C     CHECK TO SEE IF WE WANT DELTA(M,N) FACTOR                           CONLIM
C                                                                         CONLIM
  120 IF (IN.NE.IM) GO TO 140                                             CONLIM
      HF = HF*(XNP(IN) - AI)/QHAT(IN)                                     CONLIM
  140 AISUM = AISUM + HF                                                  CONLIM
  200 CONTINUE                                                            CONLIM
      APROD = APROD*AISUM                                                 CONLIM
  300 CONTINUE                                                            CONLIM
      ASUM = ASUM + APROD                                                 CONLIM
  400 CONTINUE                                                            CONLIM
C                                                                         CONLIM
C     SUM UP VALUES FOR ALL VOLUMES                                       CONLIM
C                                                                         CONLIM
      A2 = A2 + ASUM                                                      CONLIM
  500 CONTINUE                                                            CONLIM
      IF (IFLAG.EQ.0) RETURN                                              CONLIM
```

```
      IF (IVEND.LT.IVSIZE) RETURN
      GO TO 5
      END
```