

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



FACILITY FORM 602	<b>N70-19760</b>	
	(ACCESSION NUMBER)	(THRU)
	<b>83</b>	<b>1</b>
	(PAGES)	(CODE)
<b>CR-102478</b>	<b>08</b>	
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)	

BEHIND THE SCENES OF THE NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

CR-102478

RESEARCH DEPARTMENT,  
BELL AEROSYSTEMS COMPANY,  
P.O. BOX 1  
BUFFALO, N.Y. 14240

ADAPTIVE COMPUTING TECHNIQUES

DR. ERIC SCHMITT

REPORT ON OPTIMIZATION TECHNIQUES

FOR UNIMODAL FUNCTIONS

CONTRACT NAS8-21417

AUGUST 19, 1969

Prepared For George C. Marshall Space Flight Center

Marshall Space Flight Center

Alabama 35812

ABSTRACT  
INVESTIGATION OF ADAPTIVE  
COMPUTER TECHNIQUES

This report deals with a new approach to the problem of minimizing or maximizing an unknown function of many variables, given only the ability to evaluate the function at chosen points.

The best classical methods are reviewed, and it is noted that their strategy at a given point in a search depends very little, if at all, on the previous steps. Thus, valuable information which could lead to accelerated convergence is neglected.

An adaptive step size random search which makes use of past information is introduced. A theory for the rate of convergence is formulated and confirmed by simulation in a digital computer. The computer time is shown to be less than that needed by non-adaptive methods, particularly for functions of many variables.

The adaptive search method is shown to solve a trial problem, in which function measurements are corrupted by noise, twice as fast as the best classical method.

## CONTENTS

Section		Page
1.0	INTRODUCTION . . . . .	1-1
2.0	TECHNICAL DISCUSSION . . . . .	2-1
2.1	PROBLEM DEFINITION . . . . .	2-1
2.1.1	Direct Search Techniques . . . . .	2-1
2.1.2	Computation Models for Direct Search Techniques . . . . .	2-2
	a. Closed Loop System Configuration . . . . .	2-2
	b. Open Loop System Configuration . . . . .	2-6
2.1.3	Considerations and Assumptions . . . . .	2-8
2.2	BASIC ASPECTS OF AN ADAPTIVE SEARCH PROCEDURE . . . . .	2-9
2.3	UPPER BOUNDS FOR SPEED OF CONVERGENCE FOR HYPERSPHERICAL FUNCTIONS UNDER VARIOUS ASSUMPTIONS . . . . .	2-15
2.3.1	Assumption: Complete Information on Direction of Search Vector, No Information on Distance . . . . .	2-15
2.3.2	Assumption: Complete Information on Distance, No Information on Direction . . . . .	2-17
2.3.3	Assumption: Complete Information on Distance, Partial Information on Direction . . . . .	2-16
2.4	DERIVATION OF A QUASIOPTIMAL ADAPTIVE SEARCH ALGORITHM FOR UNDISTURBED HYPERSPHERICAL FUNCTIONS . . . . .	2-35
2.5	CONDITIONS AND LIMIT OF SPEED OF CONVERGENCE FOR HIGHLY DISTURBED HYPERSPHERICAL FUNCTIONS . . . . .	2-43
2.6	ADAPTIVE SEARCH ALGORITHMS FOR A LARGER CLASS OF DISTURBED QUALITY FUNCTIONS . . . . .	2-43
2.6.1	Adaptive Algorithms for Disturbed Hyperspherical Functions . . . . .	2-49
2.6.2	Adaptive Search Algorithm for Disturbed Quality Functions . . . . .	2-50
2.7	PROOF OF OPERATION, SOLUTION OF A TRIAL PROBLEM, COMPARISON WITH STOCHASTIC APPROXIMATION . . . . .	2-57

CONTENTS (CONT)

Section	Page
2.7.1 Problem Definition . . . . .	2-57
2.7.2 Solution using Adaptive Search Technique . . . . .	2-59
2.7.3 Optimization using Stochastic Approximation . . . . .	2-61
2.7.4 Comparison of Results . . . . .	2-66
3.0 CONCLUSION AND RECOMMENDATIONS . . . . .	3-1
4.0 REFERENCES . . . . .	4-1
APPENDIX A . . . . .	A-1

## ILLUSTRATIONS

Figure		Page
2-1	Mathematical Model of an Environmental Process . . . . .	2-3
2-2	Closed Loop Computational Model for Direct Search Techniques . . . . .	2-5
2-3	Open Loop Computational Model for Direct Search Techniques . . . . .	2-7
2-4	Basic Search Procedure, Search Parameter $r$ , $V$ . . . . .	2-11
2-5	Hyperspherical Function $Q(X)$ . . . . .	2-19
2-6	Probability of Success versus $\frac{r}{a}$ . . . . .	2-20
2-7	Expected Value of Fractional Distance Decrement . . . . .	2-24
2-8	Conditions for Search Vector Extension . . . . .	2-28
2-9	Conditions for Vector Inversion . . . . .	2-31
2-10	Probability of Success versus $\frac{r}{a}$ for Highly Disturbed Hyperspherical Functions . . . . .	2-47
2-11	Expected Value of Fractional Distance Decrement for Highly Disturbed Hyperspherical Functions . . . . .	2-51
2-12	Equiquality Contours of Function $Q(X)$ Given as Problem No. 1 . . . . .	2-58

TABLES

Number		Page
2-1	Sequence of Fibonacci Numbers . . . . .	2-16
2-2	Optimal Parameters for Assumption 2.3.2 . . . . .	2-23
2-3	Number of Search Steps Needed to Reduce the Initial Distance A Fraction $10^{-3}$ . . . . .	2-25
2-4	Optimal Limit Angles for Vector Extension and Inversion for Various Dimensions . . . . .	2-30
2-5	Expected Minimal Step Numbers for the Ideal Assumptions . . . . .	2-34
2-6	Possible Combinations for the Values of the Random Variables In (2-83) and (2-84) . . . . .	2-44
2-7	Optimal Parameters and Minimal Number of Search Steps for Highly Disturbed Hyperspherical Functions . . . . .	2-48
2-8	Average Step Number for Solution of Problem No. 1 Using Adaptive Search Techniques . . . . .	2-61
2-9	Number of Function Evaluations Needed by the Dupac Kiefer- Wolfowitz Rule to Solve Problem No. 1 . . . . .	2-66
2-10	Number of Function Evaluations Needed by the Kesten Kiefer- Wolfowitz Rule to Solve Problem No. 1 . . . . .	2-66

## 1.0 INTRODUCTION

In recent years, an increasing amount of research has been directed at the theory and development of adaptive systems, that is systems that improve their performance through experience. Although adaptive principles have been increasingly introduced for solving various specific problems, their application in the area of numerical mathematics occurred only recently more frequently, mostly in the Russian literature (1, 2, 3, 4).

Particular areas of interest for the application of adaptive principles is the analysis and optimization of multiparameter nonlinear systems. There are many nonadaptive methods that have been formulated for solving these problems and the variations or modifications to these methods are numerous. One explanation for the existence of so many methods and variations is simply that any one method is frequently not suited to a particular problem. It is for this reason one might ask if it is feasible to develop an adaptive method capable of solving a large variety of complicated optimization or root finding problems by automatic adaptation to each particular problem.

Fundamental questions on such an adaptive search technique are: convergence, speed of convergence, final accuracy, increasing complexity with increasing dimensionality, behavior when the systems parameters are ill-defined, or have noise of unknown characteristics superimposed upon them.

The various nonadaptive iterative approaches for solving systems of equations, such as Newton - Raphson (5) and Fletcher Powell (6) require an initial estimate sufficiently close to the root in question. They fail completely when the parameters are ill-defined or "noisy". The latter conditions can only be solved - up to now - by the well known method of stochastic approximation (7, 8, 9, 10, 11, 12), which has the disadvantage of converging very slowly. A remarkable disadvantage of all techniques (a discussion of the most important techniques can be found in 13) is that the complexity of computations increases quadratically with increasing numbers of parameters, or dimensions.

In the following technical discussion, general models for adaptive computation will be introduced. Subsequently, important basic aspects of adaptive search procedures will be discussed. In Chapter 2.3 we will derive some upper limits for the speed of convergence of the adaptive search procedure under the assumption that the information is given a priori and the functions are hyperspherical. Chapter 2.4 will be devoted to deriving an adaptive algorithm for the theoretical cases assumed under 2.3.

In Chapter 2.5 conditions and upper limits for the speed of convergence will be derived under the assumption that noise is superimposed. A convergent adaptive search algorithm based on these results applicable to a larger class of functions will be discussed in Chapter 2.6.

In Chapters 2.7 and 2.8 proof of operation of the derived adaptive search algorithm will be performed on the two problems formulated by NASA.

## 2.0 TECHNICAL DISCUSSION

### 2.1 PROBLEM DEFINITION

Recently, much attention has been given to the theory of optimization of systems. However, only simple, well formulated problems have been solved analytically using indirect methods because of the tedious computations which have to be carried out. Furthermore, efficient optimizing techniques are today of vital use for the solution of arbitrary systems of simultaneous algebraic and transcendental equations (5).

#### 2.1.1 Direct Search Techniques

The presence of high speed digital computers has stimulated the interest in a class of optimizing procedures which are called Direct Search Techniques. In accordance with Leon (14) a brief definition of direct search techniques is the following:

Direct search is a sequential examination of trial solutions which are obtained by direct numerical functional evaluations. Each trial solution might be compared with the "best" obtained in previous trials, and there is a strategy for determining the parameters for the next trial solution.

Direct search procedures may be classified as deterministic or nondeterministic, and further as adaptive and nonadaptive techniques. A search method is called deterministic if it does not contain any random element or random variable. Examples of deterministic methods include the factorial method, the universal method, the gradient method and some modifications and combination of these principles, for example, the powerful conjugate gradient procedure (6, 15).

Deterministic methods do not normally converge to a solution when the parameters are ill-defined due to erroneous measurements or noise. The only well known deterministic method, which converges under the presence of noise of a certain characteristic, is the method of stochastic approximation (7-12). The procedure is essentially a gradient method in which the partial derivatives are taken experimentally by measuring differences. The term 'stochastic' refers to the class of problems which can be solved; the search method is deterministic.

A search method is nondeterministic if the search algorithm itself contains a random element. The simplest method in this category is the global random search proposed by Brooks (16). Although, no assumption on the modality (number of extrema) of the function need be made, the global random search has a very slow rate of convergence. However, it has some merits as a starting procedure for the solution of multimodal problems. An improved nondeterministic procedure, but restricted to unimodal problems, was introduced by Box (17) who called the improved procedure "evolution strategy". However, this method does not converge with probability "one" to the optimal point.

### 2.1.2 Computation Models for Direct Search Techniques

It is convenient to divide the class of problems to be solved by direct search techniques into two categories. These are: (1) problems that are solvable by a closed loop system configuration; and (2) those solvable using an open loop configuration.

#### a. Closed Loop System Configuration

Let  $S_E$  be a physical system or a mathematical model of a process (Figure 2-1), and let

$U = (u_1, u_2 \dots u_r)$  be an independent input vector,

$Y = (y_1, y_2 \dots y_m)$  be its output vector,

and

$X = (x_1, x_2 \dots x_n)$  be the state vector

of the system  $S_E$ .  $X$  is the vector of state parameters which governs the behavior of system,  $S_E$ . Let be further defined a criterion,  $Q$ , of optimality or some general index of performance relating to system,  $S_E$ . In general, the criterion,  $Q$ , may be a functional on the vector,  $Y$ . Since the behavior of  $S_E$ , reflected in its output ( $Y$ ), depends upon the state vector,  $X$ , the criterion,  $Q$ , is an implicit function of the state vector,  $X$ .

The input vector, ( $U$ ), might be some external input to  $S_E$  which may be present or absent depending on the particular problem. Furthermore, there may be some noise superimposed with the effect that instead of observing  $Q(X)$  we observe

$$Z(X + X_N) = Q(X) + Q_N \quad (2-1)$$

where  $X_N$  indicates the "noisy" part of the state vector, and  $Q_N$  is the "noisy" part of the criterion.

The optimization problem consists of determining the vector,  $X$ , (i.e., the values of the parameters,  $x_i$ ) which represents the optimal state of system,  $S_E$ . The optimal state of  $S_E$  is the vector  $X = X_{opt}$ , where  $X_{opt}$  minimizes (maximizes) the function,  $Q$ , so that

$$Q(X_{opt}) \leq Q(X) \text{ for all } X \in \Omega, \quad (2-2)$$

where  $\Omega$  is the region in the state space,  $X$ , in which the function,  $Q(X)$ , is defined.

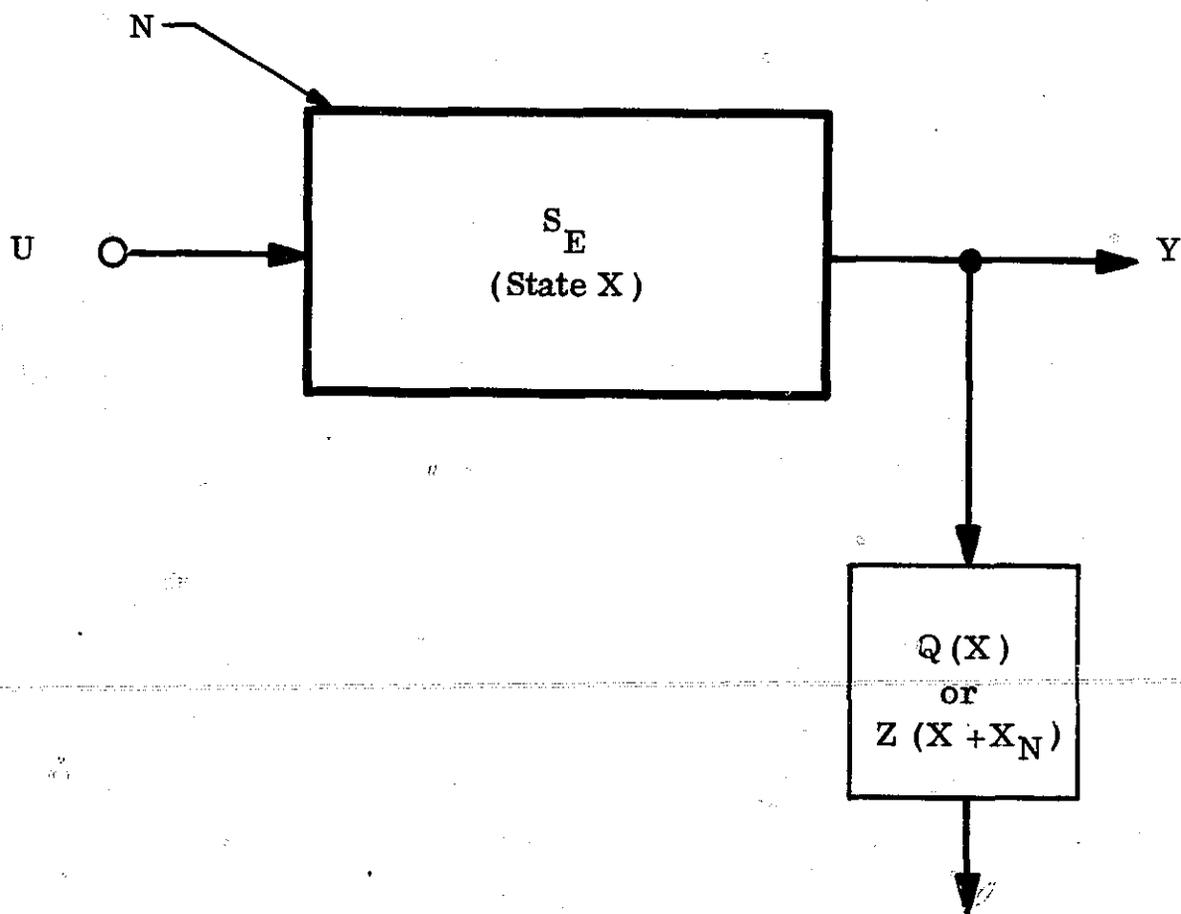


Figure 2-1. Mathematical Model of an Environmental Process

Throughout this paper we will assume minimization unless otherwise mentioned.

It is well known that minimization procedures may be used to solve systems of linear or nonlinear equations. Given the system of equations:

$$\begin{aligned} f_1(X) &= 0 \\ f_2(X) &= 0 \\ &\cdot \\ &\cdot \\ &\cdot \\ f_m(X) &= 0 \end{aligned} \tag{2-3}$$

where  $X$  can be thought as the state of system,  $S_E$ , in Figure 2-1, the input,  $U$ , for this case is not occurrent. The functions,  $f_1(X) \dots f_m(X)$ , are the components  $y_1 \dots y_m$  of the output vector,  $Y$ .

To find the roots of the system (2-3) the function  $Q(X)$  is formed, with

$$Q(X) = \sum_{j=1}^m f_j^2(X) \tag{2-4}$$

or more exact, if the  $f_j(X)$ 's are complex functions,

$$Q(X) = \sum_{j=1}^m f_j(X) \cdot f_j^*(X) \tag{2-5}$$

where  $f_j^*(X)$  indicates the conjugate complex functions.

For simplicity, we will restrict our discussion to real functions,  $f_j(X)$ . The function,  $Q(X)$ , is nonnegative and achieves the minimum value zero only when the system (2-3) is satisfied. The vector,  $X$ , which minimizes (2-4) therefore, satisfies (2-3), i.e., it is a root of the system.

The computational model for the solutions of optimization or root-finding problems is represented in Figure 2-2. The output vector,  $\xi$ , of system,  $S_A$ , produces changes in the state of system,  $S_E$ ; for instance, at the  $m^{\text{th}}$  trial point the state vector,  $X^{(m)}$  is given by

$$X^{(m)} = X^{(k)} + \xi^{(m)} \tag{2-6}$$

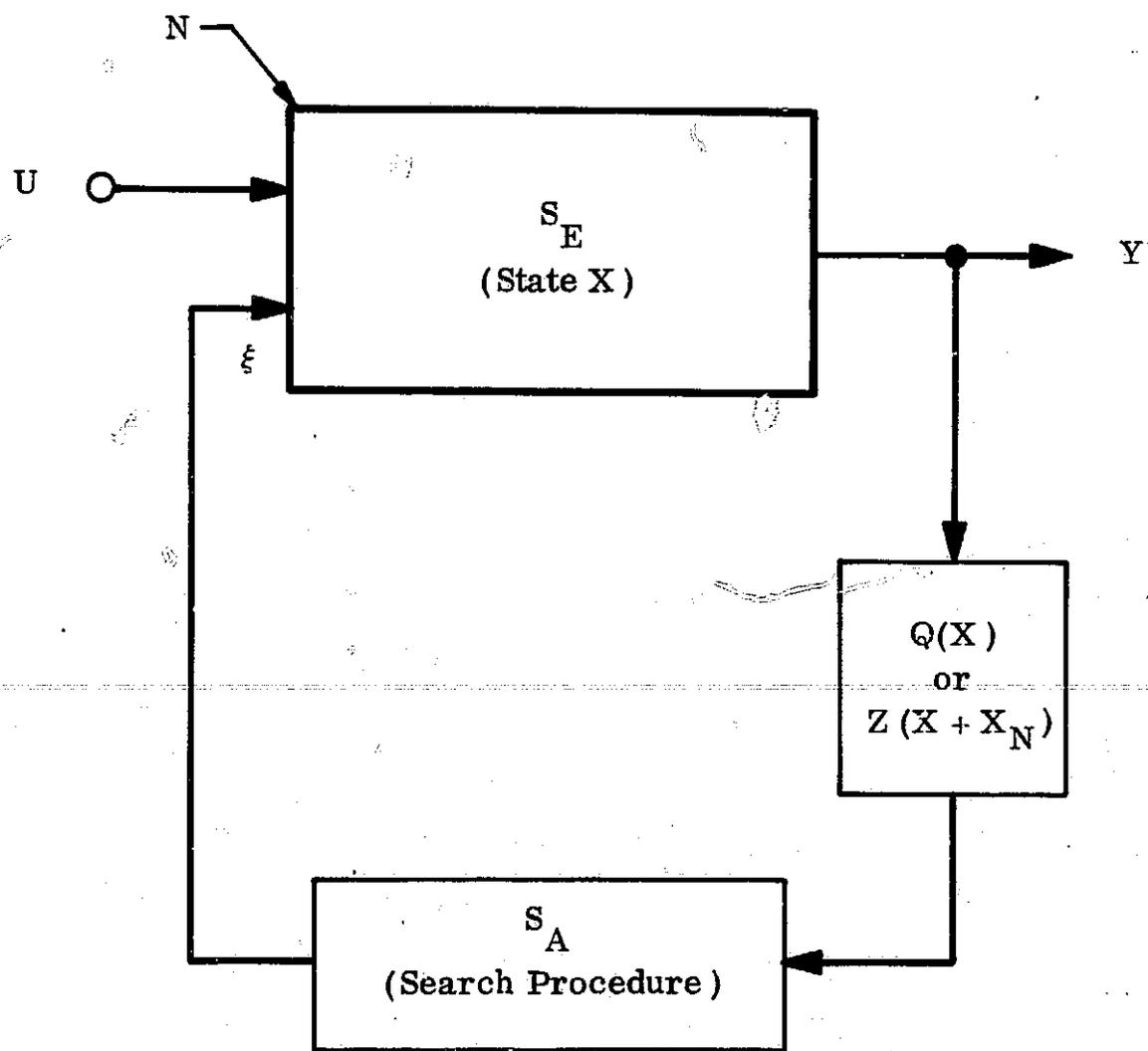


Figure 2-2. Closed Loop Computational Model for Direct Search Technique

where  $X^{(k)}$  is some state vector determined in previous trials and

$$\xi^{(m)} = (\xi_1^{(m)}, \xi_2^{(m)}, \dots, \xi_n^{(m)}) \quad (2-7)$$

is the output vector of  $S_A$  at the  $m^{\text{th}}$  trial.

After the evaluation of the function,  $Q(X^{(m)})$  the system,  $S_A$ , will produce, after corresponding computations, the output vector  $\xi^{(m+1)}$  which effects subsequently the state of  $S_E$  similar to that indicated by (2-6).

#### b. Open Loop System Configuration

There is a class of problems where the system input of  $S_E$  is not accessible. In other words,  $S_A$  cannot perform changes of the state vector,  $X$ , of system,  $S_E$ . The problem formulation is in general the design of a transformation system,  $T$ , in order to transform the output,  $U_T$ , of system,  $S_E$ , into the output,  $Y_T$ . All information filtering problems belong to this problem class.

The computational model for this class of problems is shown in Figure 2-3. Let the state of the transformation system,  $T$ , be

$$X_T = (x_{T1}, x_{T2}, \dots, x_{Tn}) \quad (2-8)$$

A system,  $S_A$ , representing a certain search procedure changes the state vector  $X_T$ , of system,  $T$ , by supplying a vector  $\xi_T$  at time instant  $m$ .

$$\xi_T^{(m)} = (\xi_{T1}^{(m)}, \xi_{T2}^{(m)}, \dots, \xi_{Tn}^{(m)}) \quad (2-9)$$

A quality criterion,  $Q(X)$ , evaluated at each time instant serves as input to system,  $S_A$ . The system,  $S_A$ , in Figure 2-3 performs the same search task as in the closed loop configuration. The only difference is that instead of experimenting with system,  $S_E$ , it experiments with the transformation system,  $T$ . It changes the parameters  $x_{Ti}$  step by step in such a way that the quality criterion,  $Q(X_T)$ , approximates the optimal value,  $Q(X_{T \text{opt}})$ .

The following discussion does not specifically deal with the details of the transformation system,  $T$ , which may be different for the various problems. Main interest is directed to system,  $S_A$ , representing the search procedure. In both the closed loop and the open loop configuration,  $S_A$  has the task of searching for the optimal conditions.

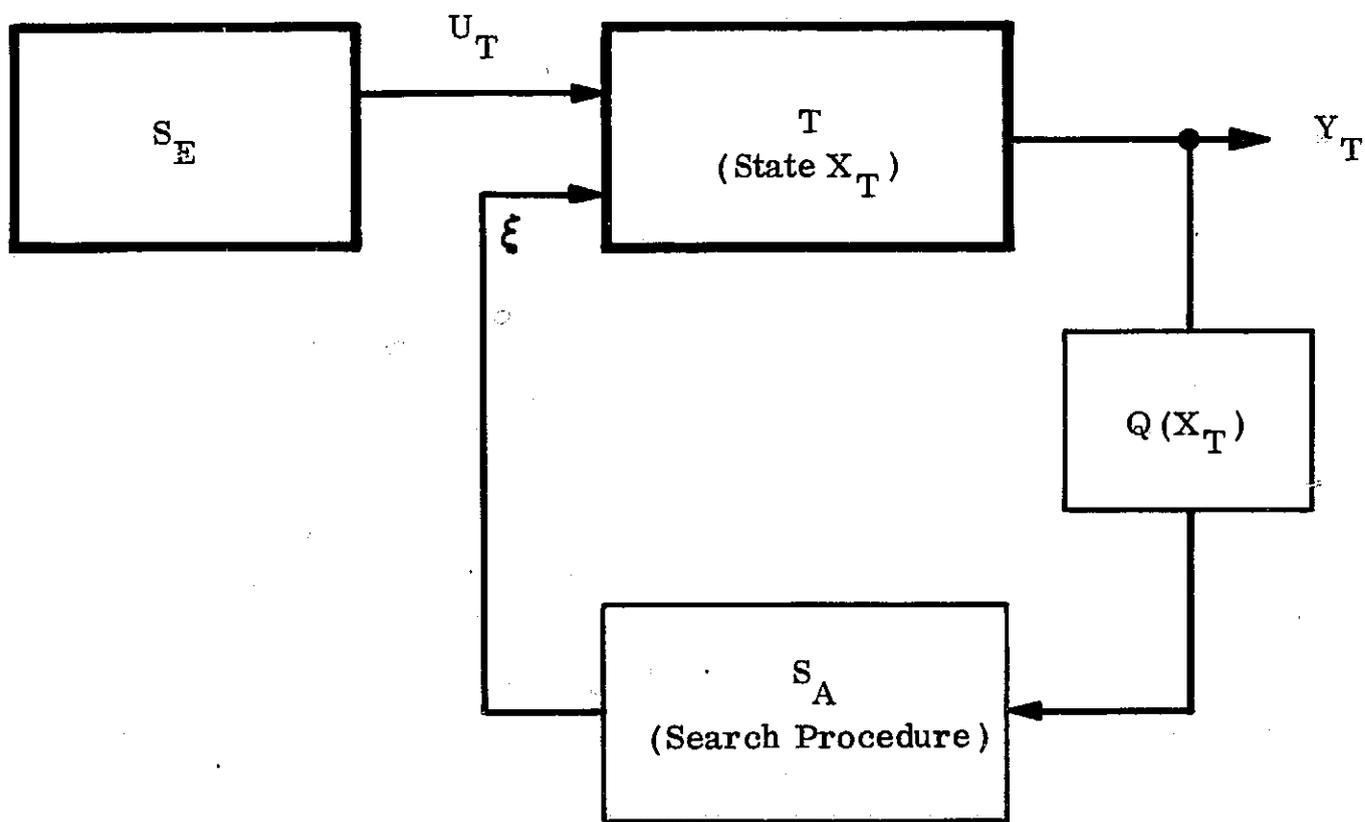


Figure 2-3. Open Loop Computational Model for Direct Search Techniques

### 2.1.3 Considerations and Assumptions

Depending on the specific problem, different possible conditions have to be considered. Some interesting items are briefly discussed.

- (a) The optimum quality value,  $Q_{opt}$ , may be a priori known or unknown.
- (b) The states of the systems,  $S_E$  or  $T$ , to be changed by the output of system,  $S_A$ , may be reversible or nonreversible.
- (c) The mathematical process governing system,  $S_E$  or  $T$ , may be stationary or nonstationary.
- (d) The system,  $S_E$  or  $T$ , may be:
  - (1) Undisturbed by noise
  - (2) Disturbed by noise with known characteristics
  - (3) Disturbed by noise with unknown characteristics
- (e) The Quality function,  $Q(X)$ , may be:
  - (1) Unimodal - there is a unique optimum
  - (2) Multimodal - there is a global optimum and some local optima or subextrema
- (f) The quality function,  $Q(X)$ , may be:
  - (1) A continuous function with existing derivatives for every  $X \in \Omega$
  - (2) A continuous function, but derivatives don't exist for every  $X \in \Omega$
  - (3) A discrete function
- (g) The parameter values,  $x_i$ , i.e., the domain of the function,  $Q(X)$ , may be:
  - (1) Continuous and unbounded in the real number space
  - (2) Continuous, but bounded to a certain region,  $\Omega$
  - (3) Discrete

The requirements on the algorithm or iterative search procedure of system,  $S_A$ , are different for the various conditions. From the point of view of application, it would be convenient to develop an algorithm which is appropriate for each of the conditions stated above. All known direct search procedures, with the exception of the simple global random search technique, impose a restricted number of the above conditions on the quality function,  $Q(X)$ .

One severe condition is the modality. If the problem to be solved is multimodal (this is, for instance, the case when the problem to be solved is the root-finding of a system of nonlinear equations), the various known search techniques require an initial estimate sufficiently close to the root in question. Generally, multimodal problems must be reduced a priori to a number, say  $K$ , of unimodal problems by partitioning the region,  $\Omega$ , in  $K$  subregions in which the local optima are located, and determining  $K$  initial vectors,  $X$ , such that each one is sufficiently close to the corresponding local optimal point  $X_{opt}$ .

Another severe condition is that of disturbance by noise or ill-defined parameters,  $x_i$ . All deterministic search techniques fail if noise is superimposed on the quality function,  $Q(X)$ . Besides simple random search procedures, the only method capable of solving problems with superimposed noise is stochastic approximation. However, the noise must be unbiased; i.e., the expected value of the noise is required to be zero.

Still another, perhaps most severe, property of practically all known search techniques is the so-called curse of dimensionality. For instance, all search techniques, which require measurements of the gradient and estimations of the location of the optimum, need for quadratic functions at least  $(n + 1)^2$  function evaluations,  $Q(X)$ , besides further computations for each of the  $(n + 1)^2$  search steps, where  $n$  is the number of dimensions. In the literature, gradient techniques are often said to be linearly dependent on the dimension,  $n$ . This accounts only for the measurements of the  $n$  partial derivatives. To locate the optimal point of a general quadratic at least  $n$  gradient measurements are needed, besides an additional function evaluation for each point at which the gradient is measured. Thus, referring to the number of function evaluations it can be stated that the computation time for all well known search techniques increases quadratically with the number of dimensions,  $n$ .

In the following we will investigate the general question: Whether adaptive principles can be introduced in system,  $S_A$ , (see Figures 2-2 and 2-3) with the property of having a faster speed of convergence than known search techniques and with the capability of solving problems with ill-defined parameters or superimposed noise, hopefully, with a faster speed of convergence than the method of stochastic approximation.

## 2.2 BASIC ASPECTS OF AN ADAPTIVE SEARCH PROCEDURE

We will begin our discussion with very simple and restricted conditions on the quality function  $Q(X)$  as follows:

- (a) The optimal value,  $Q_{opt}$ , is unknown.
- (b) The states of the system  $S_E$  or  $T$  to be optimized are reversible.
- (c) The process governing the behavior of  $S_E$  or  $T$  is stationary.
- (d) No noise is superimposed; all parameters are well-defined.

- (e) The function  $Q(X)$  is unimodal.
- (f) The function  $Q(X)$  may be continuous or discrete; no derivatives are required to exist.
- (g) The parameters  $x_i$  are not bounded.

These conditions are assumed for a first description and investigation of the underlying principles; they will be considerably relaxed, and most of them eventually removed as we progress with the exception of (e). Let us consider the projection of the domain of the  $n$ -dimensional quality function  $Q(X)$  in the two-dimensional plane of Figure 2-4. Let us denote the initial  $n$ -dimensional  $X$  vector by  $X^{(1)}$  and assume the search procedure starts at this point. The corresponding functional value  $Q$  at this point is observed as  $Q(X^{(1)}) = Q^{(1)}$ .

Let us denote the distance from the initial point  $X^{(1)}$  to the optimal point  $X_{opt}$  by  $a^{(1)}$

with

$$a^{(1)} = \sqrt{\sum_{i=1}^n (x_{i opt} - x_i^{(1)})^2}$$

or, generally, at the  $m^{th}$  search step

$$a^{(m)} = \sqrt{\sum_{i=1}^n (x_{i opt} - x_i^{(m)})^2} \quad (2-10)$$

where  $X^{(m)}$  is the point  $X$  at the  $m^{th}$  search step, and  $a^{(m)}$  the corresponding distance to  $X_{opt}$ .

Let us define a search hypersphere of radius  $r$  centered at the initial point  $X^{(1)}$ . Any point on the hypersphere of radius  $r$  can be described by vectors  $X$

$$X = X^{(1)} + r \cdot V \quad (2-11)$$

with  $V$  being an  $n$ -dimensional vector of unit length

$$V = (V_1, V_2, \dots, V_n) \quad (2-12)$$

$$|V| = \sqrt{\sum_{i=1}^n V_i^2} = 1 \quad (2-13)$$

and  $r > 0$ .

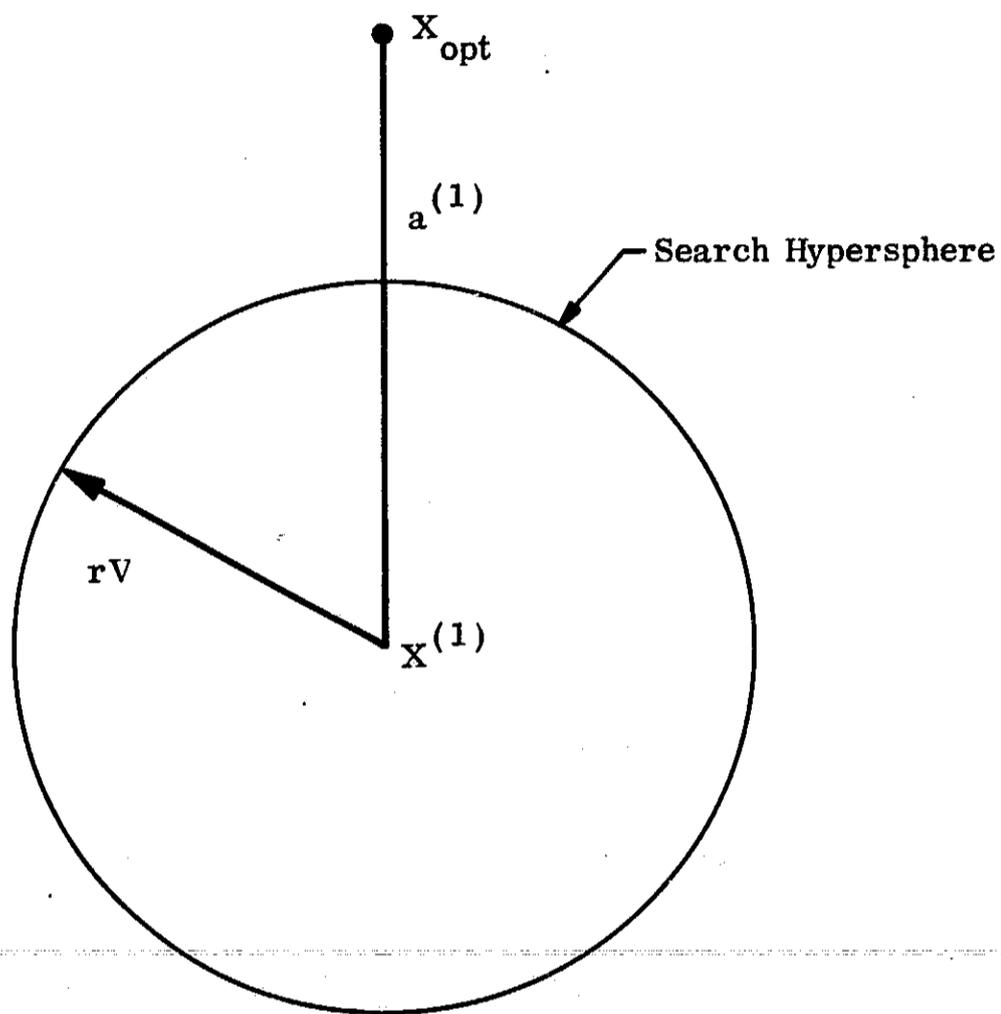


Figure 2-4. Basic Search Procedure Search Parameter  $r, V$

At the second search step let us evaluate the function  $Q(X)$  at the endpoint of the vector  $X$  in the  $n$ -dimensional space

$$\begin{aligned} X &= X^{(1)} + r^{(1)} \cdot V^{(1)} \\ &= X^{(1)} + \xi^{(1)} \end{aligned} \quad (2-14)$$

We assume the optimal point to be a minimum and call the search step a success if for a given  $\varepsilon > 0$

$$Q(X^{(1)} + \xi^{(1)}) < Q(X^{(1)}) \cdot \{1 - \varepsilon\} \quad (2-15)$$

We denote a success by  $S^{(1)} = 1$ .  $\varepsilon$  may be chosen to

$$\varepsilon = \frac{1}{2} \cdot 10^{-k} \quad (2-16)$$

where  $k$  is the number of digits in the mantissa of the floating point computation. Similarly, we call the search step a failure, denoted by  $S^{(1)} = 0$ , if

$$Q(X^{(1)} + \xi^{(1)}) \geq Q(X^{(1)}) \cdot \{1 - \varepsilon\} \quad (2-17)$$

Depending on the outcome of the second search step - success or failure - we determine the vector  $X^{(2)}$  after the second search step as follows:

$$X^{(2)} = X^{(1)}, \text{ if } S^{(1)} = 0 \quad (2-18)$$

or

$$X^{(2)} = X^{(1)} + \xi^{(1)}, \text{ if } S^{(1)} = 1 \quad (2-19)$$

At the third search step, we evaluate the quality function  $Q(X)$  for the endpoint of the vector

$$\begin{aligned} X &= X^{(2)} + r^{(2)} \cdot V^{(2)} \\ &= X^{(2)} + \xi^{(2)} \end{aligned} \quad (2-20)$$

and determine the vector  $X^{(3)}$  depending on the result of the evaluation,  $S^{(2)} = 0$  or  $S^{(2)} = 1$ .

In general, at the  $m^{\text{th}}$  search step we compute  $Q(X^{(m)} + \xi^{(m)})$  and determine  $S^{(m)}$  depending on the outcome of the test

$$Q(X^{(m)} + \xi^{(m)}) < Q(X^{(m)}) \cdot \{1 - \varepsilon\} \quad (2-21)$$

The vector  $X^{(m+1)}$  whose endpoint represents the relative optimal location of the search process, is then computed to

$$X^{(m+1)} = X^{(m)}, \text{ if } S^{(m)} = 0 \quad (2-22)$$

or

$$X^{(m+1)} = X^{(m)} + \xi^{(m)}, \text{ if } S^{(m)} = 1 \quad (2-23)$$

Out of  $m$  search steps designate the number of successes by  $m_s$  and the number of failures by  $m_f$ , where

$$m = m_s + m_f \quad (2-24)$$

Note that the number  $m$  of search steps equals the number of function evaluations. After each successful step a "move" to the new relative optimal point  $X^{(m+1)}$  is made. Clearly a fundamental question is that of the convergence of the sequence of "moves" to the optimal point  $X_{opt}$ . We realize, the convergence is completely governed by the sequence of the search vectors

$$\xi^{(m)} = r^{(m)} \cdot V^{(m)}, m = 1, 2, \dots \quad (2-25)$$

where  $r^{(m)}$  represents the length and  $V^{(m)}$  represents the direction of the search vector  $\xi^{(m)}$  at the  $m^{\text{th}}$  step with respect to the vector  $X^{(m)}$  representing the relative optimal location in the space at the  $m^{\text{th}}$  step.

At this point it is important to note that the search procedure depends on the two parameters  $r^{(m)}$ ,  $V^{(m)}$  only. From the point of view of function evaluations, the  $n$ -dimensional search problem is reduced to a two-dimensional one, provided that for the successive computations of  $r^{(m)}$ ,  $V^{(m)}$  no further function evaluations are required besides those determined by the number  $m$ . Assuming at the start no information is given on the type of the function  $Q(X)$ , in order that the search process converges, the system  $S_A$  (see Figures 2-2, and 2-3) must contain means to adapt the parameters  $r^{(m)}$ ,  $V^{(m)}$  at each step by using the information gained during the previous search steps.

Ideally, the adaptation procedure in  $S_A$  should be an optimal algorithm; i.e.,

- (a) It should assure convergence of the search.
- (b) It should maximize the speed of convergence, resulting in a minimal number of search steps.

The assumption of arbitrary functions  $Q(X)$  is surely too general to find an optimal adaptation algorithm in an easy way, if at all. One should remember that even for straightforward deterministic search procedures, for instance, such as the Fletcher-Powell method (6) the assumption that  $Q(X)$  is quadratic has to be made to prove convergence.

### 2.3 UPPER BOUNDS OF SPEED OF CONVERGENCE FOR HYPERSPHERICAL FUNCTIONS UNDER VARIOUS ASSUMPTIONS

Before we start with the development of adaptive algorithms, let us derive upper bounds for various assumed ideal cases. Although we will never achieve in reality the limits obtained under the ideal assumptions, the results will give some insight into the various corresponding ideal search processes.

#### 2.3.1 Assumption: Complete Information on Optimal Direction of Search Vector; No Information on Distance

We first assume the very unrealistic case that the directional information  $V^{(m)}$  is completely given a priori. Referring to Figure 2-4 this means the optimal direction of the search vector is given by

$$V_a^{(1)} = \frac{X_{opt} - X^{(1)}}{|X_{opt} - X^{(1)}|} \quad (2-26)$$

Let us assume  $V_a^{(1)} = V_a^{(2)} = \dots = V_a^{(m)}$  is given a priori; however, the distance  $a^{(1)}$  is unknown. For this case, our search problem is reduced to a one-parameter problem. For the optimal answer we refer to optimal single variable search procedures, exhaustively treated in (18, 19). The optimal sequence  $r^{(m)}$ ,  $m = 1, 2, \dots, M$  can be derived approximately from the Fibonacci sequence  $F_m$  (see also 20). The sequence  $F_m$  is represented for increasing  $M$  in Table 2-1. The number  $F_m$  is the ratio of the initial length of the uncertainty interval over the final length (at step  $m$ ) of the uncertainty interval. As the initial interval of uncertainty is unknown for our assumed ideal case, a slightly increased number of steps  $m$  can be expected for a desired ratio  $F_m$ . The sequence  $F_m$  represents the upper bound. For example, in order to reduce the distance  $a^{(1)}$  to  $a^{(M)}$  such that

$$\frac{a^{(M)}}{a^{(1)}} = \frac{1}{F_m} = \frac{1}{987} \approx 10^{-3}$$

TABLE 2-1  
SEQUENCE OF FIBONACCI NUMBERS

m	F <sub>m</sub>	m	F <sub>m</sub>
0	0	14	377
1	1	15	610
2	1	16	987
3	2	17	1,597
4	3	18	2,584
5	5	19	4,181
6	8	20	6,765
7	13	21	10,946
8	21	22	17,711
9	34	23	28,657
10	55	24	46,368
11	89	25	75,025
12	144	26	121,393
13	233		

at least  $M = 16$  search steps are needed, or in general, approximately five steps to reduce the distance by one order. For  $M = 50$  the initial distance  $a^{(1)}$  would be reduced to a fraction

$$\frac{a^{(M)}}{a^{(1)}} \approx 10^{-\frac{M}{5}} = 10^{-10}$$

for any dimension  $n$ .

### 2.3.2 Assumption: Complete Information on Distance, No Information on Direction

In order to come closer to the real case we derive the upper boundary for the speed of convergence for the assumptions stated in Chapter 2.2, (a) to (g) and under the following further assumptions:

- (h) The function  $Q(X)$  is spherical
- (i) Neither is any information on the optimal direction of the search vector given a priori, nor is an adaptive procedure used to determine any favorable direction during the search.
- (k) However, at each step the distance from the relative optimal point to the optimal point  $X_{opt}$  is known.

The derivation of the optimal conditions for the vector length  $r^{(m)}$  for the stated assumptions has already been given in the literature (4, 21, 22). However, no proof of convergence for the practical implementation of a corresponding algorithm could be achieved.

Based on the theoretical results in this chapter we will derive a convergent algorithm for the adaptation of  $r^{(m)}$  in Chapter 2.4.

Because of the assumption (i) that no information on the location of the optimum  $X_{opt}$  is known we select the parameter  $V^{(m)}$  of the search procedure at each step randomly with uniform density, i.e., each unit area on the surface of the search hypersphere is selected with equal probability.

The investigation in this section concentrates then on the derivation of the optimal condition for the parameter  $r^{(m)}$ , and the optimal speed of convergence which minimizes the number of search steps.

Suppose at some stage the distance from the optimum is  $a$ , and the vector  $\xi$  has length  $r$ , as indicated in Figure 2-5. Suppose the trial vector is at an angle  $\phi$  away from the direction of the optimum. Then from the figure, the trial vector leads to an improvement if  $-\alpha < \phi < \alpha$ . The angle  $\alpha$  is

$$\alpha = \arccos \left( \frac{r}{2a} \right) \quad (2-27)$$

or  $\frac{r}{a} = 2 \cos \alpha$

As shown in (1 and 23) the uniform density for the direction of the search vector can be expressed using the angle  $\phi$  defined in Figure 2-5. For the probability density as a function of the angle  $\phi$  one finds:

$$p(\phi) = \frac{\sin^{n-2} \phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi d\phi} \quad (2-28)$$

where  $\phi$  is considered in the interval  $(0, \pi)$ .

The probability that a search step leads to a success, denoted by  $P_S$  is

$$P_S = \int_0^\alpha p(\phi) d\phi$$

$$P_S = \frac{\int_0^\alpha \sin^{n-2} \phi d\phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi d\phi} \quad (2-29)$$

The behavior of  $P_S$  as a function of  $\alpha$  (or  $\frac{r}{a}$  respectively) and dimension  $n$  is an interesting pointer for the derivation of an adaptive algorithm (Chapter 2.4). For this reason it has been evaluated and represented in Figure 2-6 for various dimensions and the interval of interest.

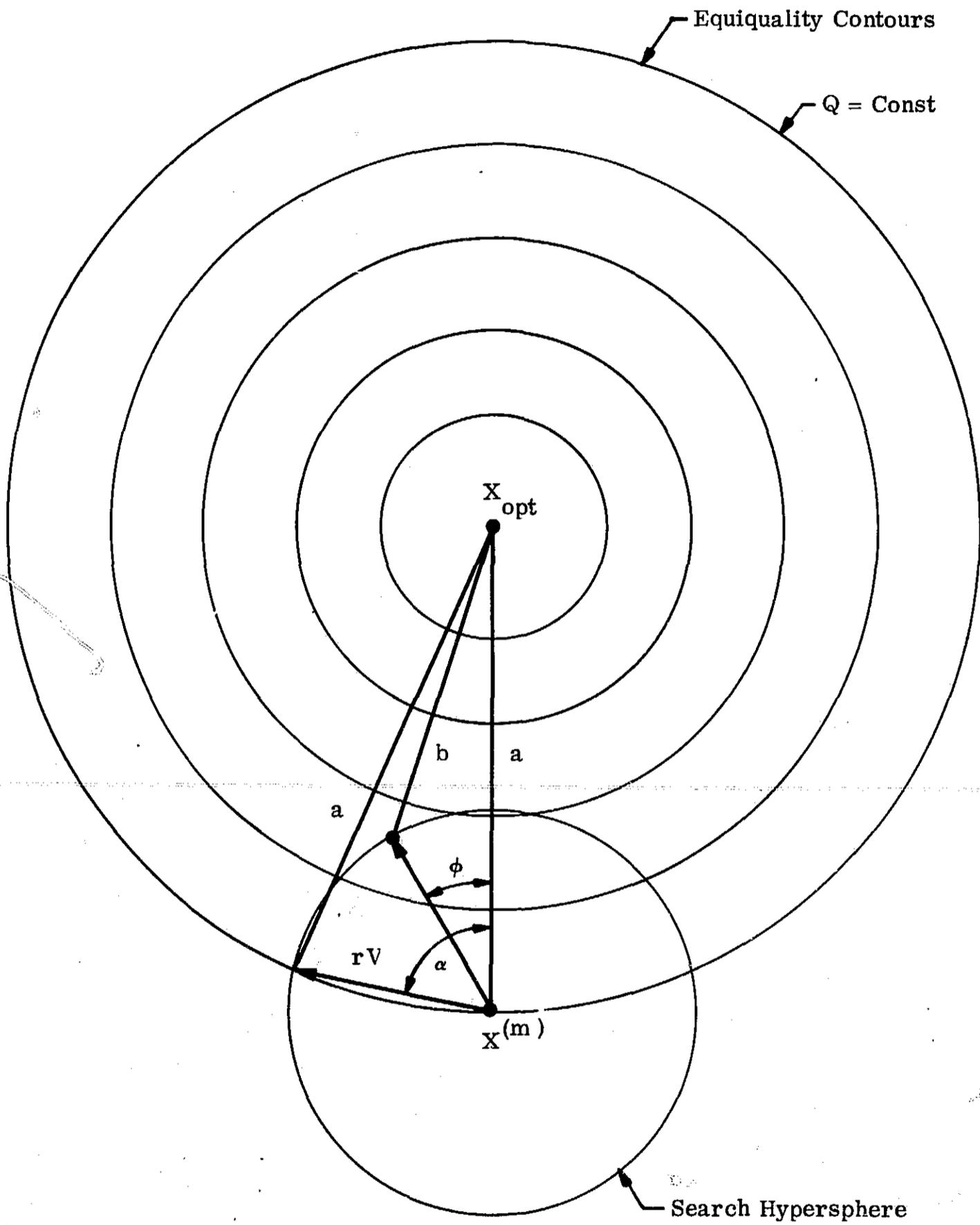


Figure 2-5. Hyperspherical Function  $Q(X)$

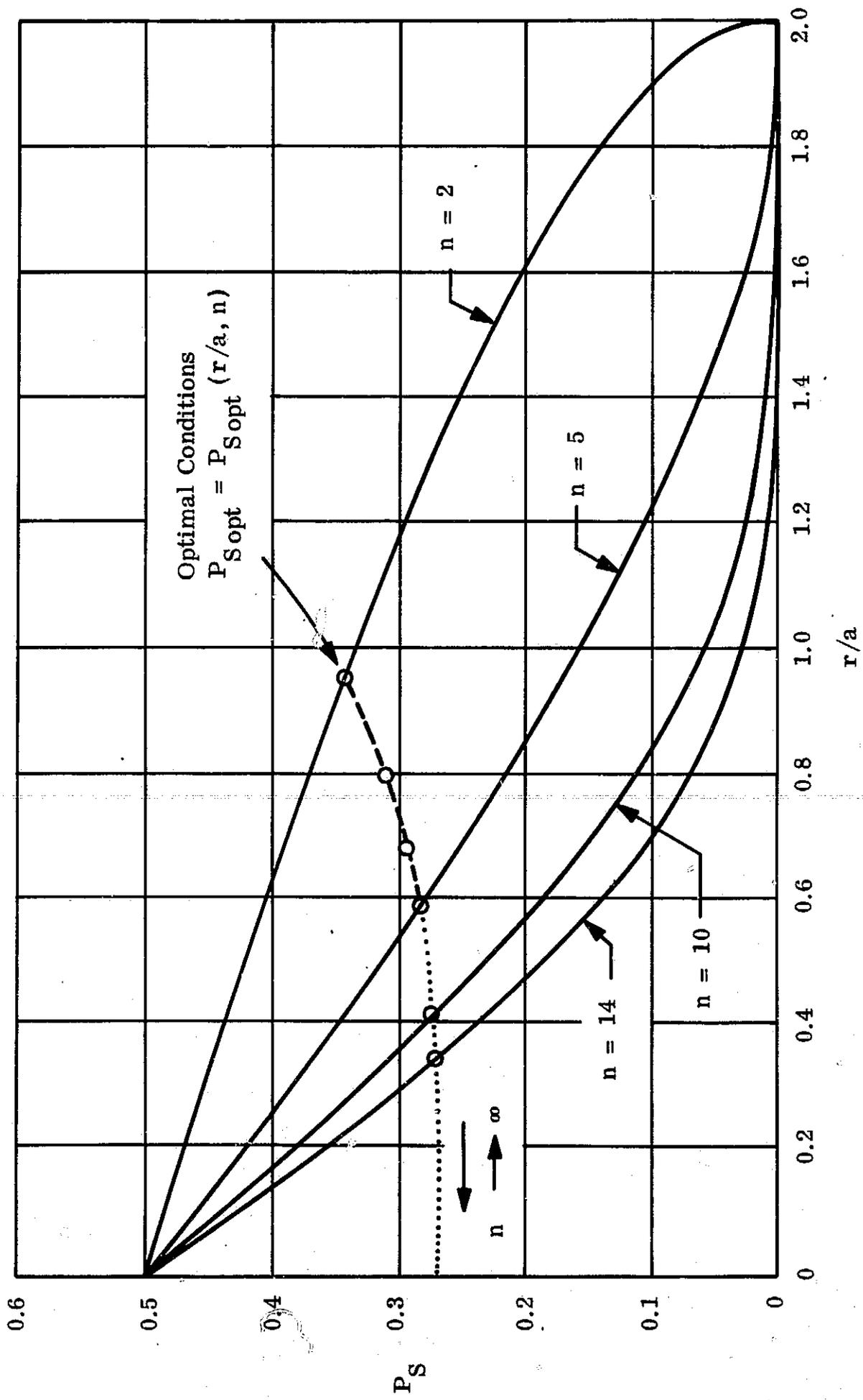


Figure 2-6. Probability of Success versus  $r/a$

$$0^\circ \leq \alpha \leq 90^\circ ,$$

or for  $\frac{r}{a}$  respectively

$$2 \geq \frac{r}{a} \geq 0$$

For  $\frac{r}{a} \rightarrow 0$  the probability  $P_S$  approaches  $P_S = 0.5$  for every dimension  $n$ . However, for

$$2 > \frac{r}{a} > 0$$

independently increasing the dimensionality  $n$  reduces  $P_S$ .

Referring to the search procedure described under 2.2 and from Figure 2-5 we see that if a search step leads to success,  $|\phi| < |\alpha|$ , then the search point is shifted to the new position nearer the optimum. The change in the distance is

$$\Delta a = \begin{cases} a - \sqrt{a^2 + r^2 - 2ar \cos \phi} & , \text{ if } -\alpha < \phi < \alpha \\ 0 & , \text{ otherwise} \end{cases} \quad (2-30)$$

and dividing through by "a" gives the fractional decrease in distance

$$\frac{\Delta a}{a} = 1 - \sqrt{1 + \left(\frac{r}{a}\right)^2 - 2\left(\frac{r}{a}\right) \cos \phi} \quad , \quad (2-31)$$

if  $-\alpha < \phi < \alpha$  , and 0 otherwise

Using (2-27) we obtain

$$\frac{\Delta a}{a} = 1 - \sqrt{1 + 4 \cos \alpha (\cos \alpha - \cos \phi)} \quad , \quad (2-32)$$

if  $-\alpha < \phi < \alpha$  , and 0 otherwise

The expected fractional decrement in distance per step is

$$E\left[\frac{\Delta a}{a}\right] = \int_0^\pi \frac{\Delta a}{a} p(\phi) d\phi \quad (2-33)$$

which substitutes using (2-28) and (2-32) to

$$E \left[ \frac{\Delta a}{a} \right] = \frac{\int_0^\alpha \left\{ 1 - \sqrt{1 + 4 \cos \alpha (\cos \alpha - \cos \phi)} \right\} \sin^{n-2} \phi \, d\phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi \, d\phi} \quad (2-34)$$

Maximizing the expected value  $E \left[ \frac{\Delta a}{a} \right]$  with respect to  $\alpha$ , or  $\left( \frac{r}{a} \right)$  respectively, will minimize the number of search steps and lead to the optimal conditions:

$$\alpha_{\text{opt}}, \left( \frac{r}{a} \right)_{\text{opt}}, E \left[ \frac{\Delta a}{a} \right]_{\text{opt}}, P_{S \text{ opt}}$$

The optimal parameters computed for increasing dimension  $n$  are listed in Table 2-2.

It is worthwhile to investigate further the behavior of the expected value  $E \left[ \frac{\Delta a}{a} \right]$  as a function of the ratio  $\frac{r}{a}$ . Figure 2-7 shows that the extrema of  $E \left[ \frac{\Delta a}{a} \right]$  for the various dimensions are relatively broad, i.e., small deviations from  $\left( \frac{r}{a} \right)_{\text{opt}}$  have little effect on  $E \left[ \frac{\Delta a}{a} \right]$ .

Using the optimal value  $E \left[ \frac{\Delta a}{a} \right]_{\text{opt}}$  we can now calculate the expected minimal number of search steps. Starting at an initial distance  $a^{(1)}$  from the optimum we can expect  $M$  search steps to reduce  $a^{(1)}$  to  $a^{(M)}$ , where  $M$  is

$$M = \frac{\log \frac{a^{(M)}}{a^{(1)}}}{\log \left( 1 - E \left[ \frac{\Delta a}{a} \right]_{\text{opt}} \right)} \quad (2-35)$$

Equation (2-35) has been evaluated for  $n = 2, \dots, 14$ , and a ratio  $\frac{a^{(M)}}{a^{(1)}} = 10^{-3}$ , the results are shown in Table 2-3. The step number  $M$  is approximately a linear function of the dimension  $n$ . One finds:

$$M = -47 + 34.03 n \quad (2-36)$$

for  $\frac{a^{(M)}}{a^{(1)}} = 10^{-3}$  and  $M = -15.66 + 11.34 n$  for  $\frac{a^{(M)}}{a^{(1)}} = 10^{-1}$

TABLE 2-2  
OPTIMAL PARAMETERS FOR ASSUMPTION 2.3.2

n	$\left(\frac{r}{a}\right)_{opt}$	$\alpha_{opt}$	$E \left[ \frac{\Delta a}{a} \right]_{opt}$	$p_S_{opt}$
2	0.909	62.98	0.1656	0.3434
3	0.750	67.99	0.0935	0.3024
4	0.644	71.20	0.0645	0.2891
5	0.572	73.37	0.0491	0.2834
6	0.520	74.92	0.0396	0.2794
7	0.479	76.14	0.0332	0.2784
8	0.447	77.09	0.0285	0.2772
9	0.418	77.95	0.0250	0.2766
10	0.397	78.55	0.0223	0.2751
11	0.377	79.13	0.0201	0.2753
12	0.361	79.60	0.0183	0.2743
13	0.345	80.06	0.0168	0.2740
14	0.332	80.45	0.0155	0.2735
⋮	⋮	⋮	⋮	⋮
n	$\frac{1.225}{\sqrt{n}}$	$\cos^{-1} \frac{0.6125}{\sqrt{n}}$	$1 - \sqrt{1 - \frac{0.406}{n}}$	0.27...
⋮	⋮	⋮	⋮	⋮
100	0.1225	86.5	0.00201	0.27...
⋮	⋮	⋮	⋮	⋮

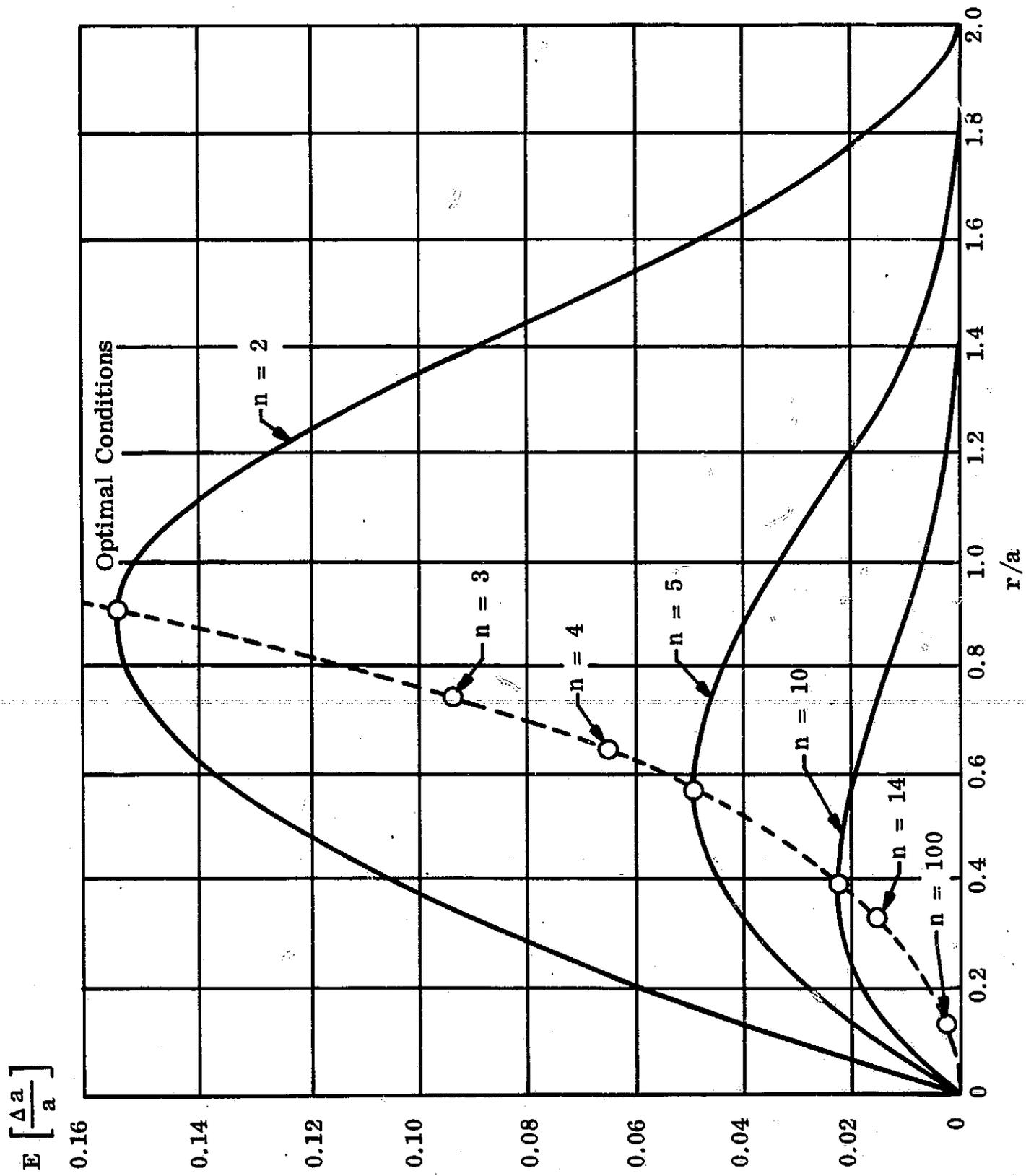


Figure 2-7. Expected Value of Fractional Distance Decrement

TABLE 2-3  
 NUMBER OF SEARCH STEPS NEEDED TO REDUCE THE  
 INITIAL DISTANCE TO A FRACTION  $10^{-3}$

n	M
2	31
3	62
4	95
5	128
6	162
7	196
8	230
9	264
10	298
11	332
12	366
13	400
14	434

Assuming spherical quality functions of the type

$$\begin{aligned}
 Q(X) &= Q_{\text{opt}} + \left( \sum_{i=1}^n (x_i - x_{i \text{opt}})^2 \right)^{\frac{\mu}{2}} \\
 &= Q_{\text{opt}} + a^{\mu}
 \end{aligned}
 \tag{2-37}$$

where  $a$  is the distance of any point  $X$  from the optimum  $X_{\text{opt}}$ , we find for the expected value of the fractional decrement of the  $Q$ -value

$$E \left[ \frac{\Delta Q}{|Q - Q_{opt}|} \right] = 1 - \left( 1 - E \left[ \frac{\Delta a}{a} \right] \right)^\mu \quad (2-38)$$

Schumer and Steiglitz (4) based their derivations on the function

$$Q(X) = a^2$$

that is, for the case  $\mu = 2$ . They derived the approximations of the optimal parameters for large dimensions  $n$ . Their results have been modified for the more general distance relation in (2-37) and are listed in Table 2-2.

With (2-38) and (2-85) we obtain for the minimal step number needed to reduce the initial difference  $Q$ -value  $|Q^{(1)} - Q_{opt}|$  to a value  $|Q^{(M)} - Q_{opt}|$

$$M_Q = \frac{1}{\mu} \cdot M_a \quad (2-39)$$

For instance for  $\mu = 2$  and

$$\frac{|Q^{(M)} - Q_{opt}|}{|Q^{(1)} - Q_{opt}|} = 10^{-8} \quad \text{(This case was assumed in (4) for a simulation)}$$

we find from the results in (2-36)

$$M_Q = -62.64 + 45.36 \cdot n \quad (2-40)$$

In (4) an adaptive procedure for the vector length  $r$  is described, however, without proof of convergence. The simulation of the assumed function corresponding to (2-40) shows that  $M_Q = 80 \cdot n$  steps, i.e., about twice the number of steps were needed for the real case. It will be shown in Chapter 2.4 that the speed of convergence for the real case can be brought very close to the theoretical results obtained in this section, if the adaptive algorithm is optimized.

### 2.3.3 Assumption: Complete Information on Distance, Partly Information on Direction

The two previous chapters showed the effect of complete information on the two parameters  $r$ ,  $V$ . The results for the two assumed cases show that information on the direction has a much higher effect on the speed of convergence. However, any adaptation of directional information can never result in a complete knowledge as

assumed under 2.3.1. For this reason we assume in this chapter that besides complete information on the distance, only some incomplete information on the direction is given.

The question arises what amount of incomplete information we should assume? For instance, we could change the uniform density in 2.3.2 to a shape which is maximal in the direction to the optimum ( $\phi = 0$ ), and which falls off towards  $\phi = \pi$ . A similar effect is obtained by the procedure discussed below.

Let's assume search vectors with uniform density for the angle  $\phi$  as under 2.3.2. If a random trial leads to a success we might ask whether it is worthwhile to use the same direction and length  $r$  deterministically at the next step, denoted by "vector extension."

If a random trial leads to a failure we ask whether it is worthwhile to use the opposite direction and same length  $r$  deterministically at the next step, called "vector inversion."

For the derivation of the optimal conditions for vector extension we refer to Figure 2-8. A successful search step centered around the initial point  $X^{(m)}$  results in a move to  $X^{(m+1)}$  along a line at an angle  $\phi$  from the direction of the optimum. It is profitable to extend to  $X^{(m+2)}$ , rather than make a new random trial at  $X^{(m+1)}$  if

$$\frac{c-b}{c} \geq E \left[ \frac{\Delta a}{a} \right] \quad (2-41)$$

Using the geometry of Figure 2-8 we find the equivalent conditions for vector extension if

$$\frac{\Delta a}{a} \geq 1 - \sqrt{\frac{1 - 2\left(\frac{r}{a}\right)^2}{2 - \left(1 - E \left[ \frac{\Delta a}{a} \right] \right)^2}} \quad (2-42)$$

and for the angle  $\phi$ , if

$$-\beta < \phi < \beta$$

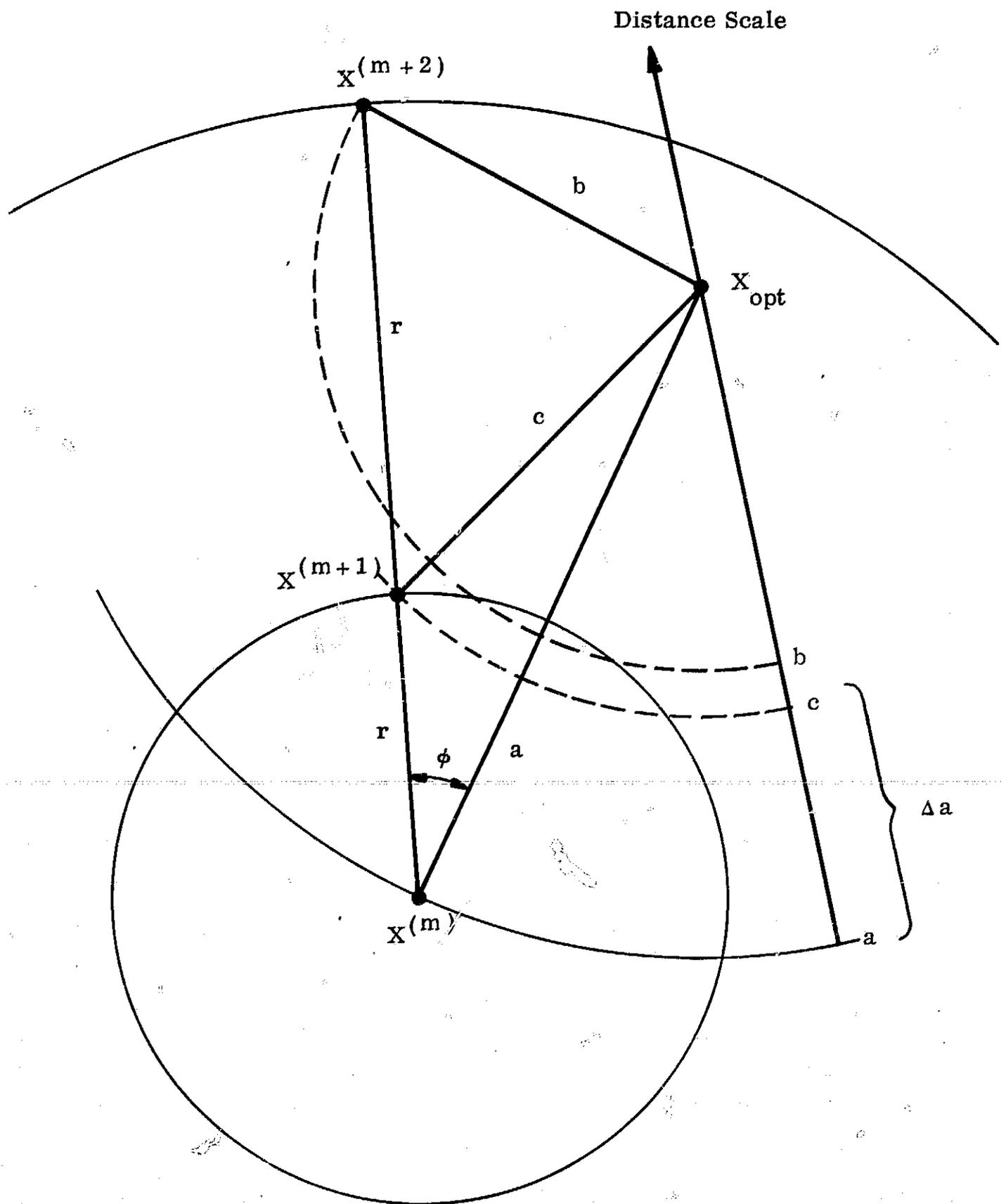


Figure 2-8. Conditions for Search Vector Extension

where  $\beta$  is a function of  $\alpha$

$$\beta = \arccos \left\{ \frac{1}{4 \cos \alpha} \left[ 1 + 4 \cos^2 \alpha - \frac{1 - 8 \cos^2 \alpha}{2 - \left( 1 - E \left[ \frac{\Delta a}{a} \right] \right)^2} \right] \right\} \quad (2-43)$$

$E \left[ \frac{\Delta a}{a} \right]$  is given by equation (2-34). Using the optimal values  $\alpha_{\text{opt}}$ ,  $E \left[ \frac{\Delta a}{a} \right]_{\text{opt}}$  derived under 2.3.2 the optimal limit angle  $\beta_{\text{opt}}$  can be obtained for the corresponding dimension. Some values are listed in Table 2-4. Note, the optimal criterion for extension cannot be satisfied for  $n = 2, 3, 4$ .

The conditions for vector inversion are derived using Figure 2-9. The random trial centered around  $X^{(m)}$  leads to a failure at  $X^{(m+1)}$ . Therefore, no move to  $X^{(m+1)}$  will be made. It is profitable to invert the search vector to  $X^{(m+2)}$ , rather than make a new random trial at  $X^{(m)}$  if

$$\frac{a-b}{a} \geq E \left[ \frac{\Delta a}{a} \right] \quad (2-44)$$

which leads from the geometry of Figure 2-9 to the equivalent inequality for vector inversion if

$$\frac{\Delta a}{a} \geq \sqrt{2 \left[ \left( \frac{r}{a} \right)^2 + 1 \right] - \left( 1 - E \left[ \frac{\Delta a}{a} \right] \right)^2} - 1 \quad (2-45)$$

and for the angle  $\phi$ , if

$$-\gamma < \phi < \gamma$$

where

$$\gamma = \arccos \left\{ \frac{1}{4 \cos \alpha} \left[ 4 \cos^2 \alpha + 1 - \left( 1 - E \left[ \frac{\Delta a}{a} \right] \right)^2 \right] \right\} \quad (2-46)$$

The optimal limit angle  $\gamma_{\text{opt}}$  can be obtained by inserting the optimal values  $\alpha_{\text{opt}}$ , and  $E \left[ \frac{\Delta a}{a} \right]_{\text{opt}}$  in (2-46). Table 2-4 contains values of  $\gamma_{\text{opt}}$  for  $n = 2, 3, \dots, 14$ . The values in Table 2-4 show that  $\gamma_{\text{opt}} > \beta_{\text{opt}}$ , i.e., the condition for vector inversion is satisfied more often than the condition for extension.

TABLE 2-4  
 OPTIMAL LIMIT ANGLES FOR VECTOR EXTENSION AND  
 INVERSION FOR VARIOUS DIMENSIONS

n	$\beta_{opt}$ (degrees)	$\gamma_{opt}$ (degrees)
2	0	50.72
3	0	59.47
4	0	64.52
5	23.92	67.72
6	33.21	70.03
7	39.59	71.79
8	44.01	73.14
9	47.46	74.24
10	50.00	75.11
11	52.38	75.90
12	54.19	76.54
13	55.84	77.12
14	57.24	77.62

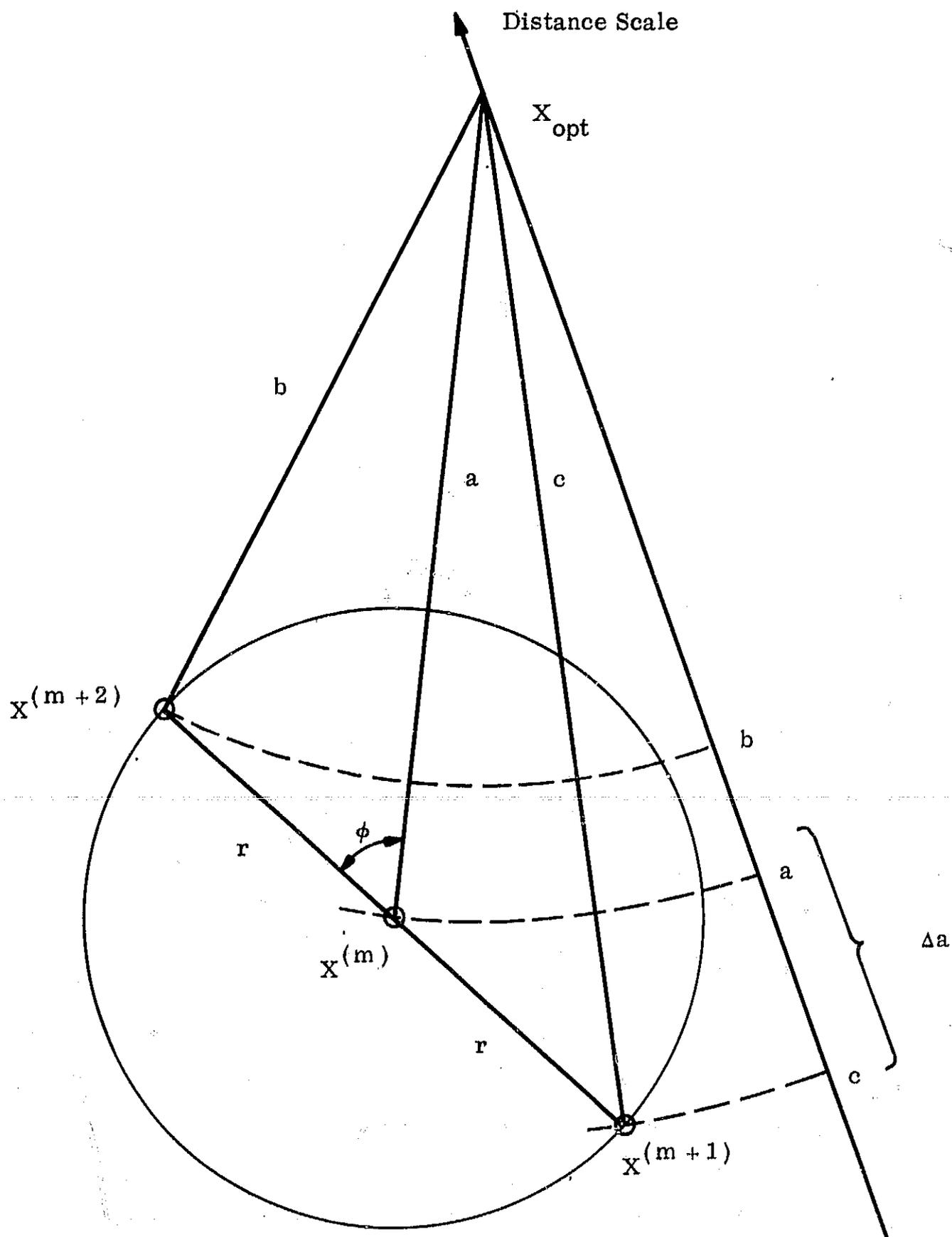


Figure 2-9. Condition for Vector Inversion

Finally we are interested in the expected value of the fractional distance decrement per step including extension and inversion. The search steps can be divided in independent random steps and completely dependent steps for extension and inversion. The corresponding density function for the angle  $\phi$  can be expressed by

$$p'(\phi) = \frac{\sin^{n-2} \phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi d\phi + \int_0^{\beta} \sin^{n-2} \phi d\phi + \int_0^{\gamma} \sin^{n-2} \phi d\phi} \quad (2-47)$$

The expected displacement for a random trial is obtained by, using (2-32) and (2-47) to:

$$E \left[ \frac{\Delta a}{a} \right]_{\text{rand.}} = \int_0^{\alpha} \left( 1 - \sqrt{1 + 4 \cos \alpha (\cos \alpha - \cos \phi)} \right) p'(\phi) d\phi \quad (2-48)$$

Referring to Figure 2-8 the expected displacement due to extension is obtained to

$$E \left[ \frac{\Delta a}{a} \right]_{\text{ext.}} = \int_0^{\beta} \left( 1 - \sqrt{\frac{1 + 8 \cos \alpha (2 \cos \alpha - \cos \phi)}{1 + 4 \cos \alpha (\cos \alpha - \cos \phi)}} \right) p'(\phi) d\phi \quad (2-49)$$

With the geometry in Figure 2-9 the expected displacement due to vector inversion is

$$E \left[ \frac{\Delta a}{a} \right]_{\text{inv.}} = \int_0^{\gamma} \left( 1 - \sqrt{1 + 4 \cos \alpha (\cos \alpha - \cos \phi)} \right) p'(\phi) d\phi \quad (2-50)$$

The expected value of the displacement per search step is obtained to

$$E' \left[ \frac{\Delta a}{a} \right] = E \left[ \frac{\Delta a}{a} \right]_{\text{rand.}} + E \left[ \frac{\Delta a}{a} \right]_{\text{ext.}} + E \left[ \frac{\Delta a}{a} \right]_{\text{inv.}} \quad (2-51)$$

(rand. + ext. + inv.)

Using formula (2-35) and the optimal conditions  $\alpha_{\text{opt}}$ ,  $\beta_{\text{opt}}$ ,  $\gamma_{\text{opt}}$  for the computation of  $E' \left[ \frac{\Delta a}{a} \right]$ , the expected minimal step number results. Table 2-5 shows the results for a ratio  $\frac{a(M)}{a(1)} = 10^{-3}$  for the various assumed cases. As under 2.3.2 the minimal expected step number for the case 2.3.3 is approximately a linear function of the dimension  $n$

$$M = -25.9 + 20.5 n \quad (2-52)$$

for a ratio  $\frac{a^{(M)}}{a^{(1)}} = 10^{-3}$ ,

and

$$M = -8.63 + 6.83 n \quad (2-53)$$

for a ratio  $\frac{a^{(M)}}{a^{(1)}} = 10^{-1}$

Although not much directional information was assumed in using vector extension and inversion the results show that the number of steps needed for a certain distance reduction is only about half the step number needed for the case 2.3.2, without any directional information. We conclude: directional information is very important to consider in the search process.

Unfortunately, directional information is only effective when it is applied together with the appropriate information on the distance. Although directional information is relatively easy to obtain using adaptive principles it is not worthwhile to consider them until the much harder part is solved, namely the adaptation of the search vector length  $r$  in relation to the distance with proof of convergence.

TABLE 2-5

EXPECTED MINIMAL STEP NUMBERS FOR THE IDEAL ASSUMPTIONS

n	Expected Minimal Step Number for the Assumed Cases		
	(Fibonacci) 2.3.1	(simple random) 2.3.2	(extension, inversion) 2.3.3
2	 16	31	20
3		62	39
4		95	59
5		128	79
6		162	98
7		196	119
8		230	139
9		264	159
10		298	180
11		332	200
12		366	221
13		400	241
14		434	262

## 2.4 DERIVATION OF A QUASIOPTIMAL ADAPTIVE SEARCH ALGORITHM FOR UNDISTURBED HYPERSPHERICAL FUNCTIONS

Based on the results obtained for the hypothetical case under 2.3.2 we now develop an adaptive algorithm for the adaption of the optimal search vector length  $r$ . We consider the same assumptions as in 2.3.2 except that we remove condition (k) and assume that no information on the distance to the optimal point is given other than by the search algorithm.

Let's first assume we start our search with an initial vector length  $r^{(1)}$  with  $\frac{r^{(1)}}{a^{(1)}} \gg 2$ . From Figure 2-6 we realize no success can be obtained. In order to make successes possible  $r$  has to be reduced in successive failure trials, i.e., the sequence  $r^{(1)}, r^{(2)}, \dots, r^{(u)}, \dots$ , where the superscript  $u$  indicates the number of successive unsuccesses must be a decreasing sequence. Consider the sequence corresponding to the recursive formula

$$r^{(m+1)} = r^{(m)} \cdot (1 - A) \quad (2-54)$$

where  $0 < A < 1$

The sequence in (2-54) has the property

$$\lim_{u \rightarrow \infty} r^{(u)} = 0 \quad (2-55)$$

$$\sum_{u=1}^{\infty} r^{(u)} < \infty \quad (2-56)$$

The property in (2-56) indicates that the sequence converges relatively fast towards  $r = 0$ . The speed of convergence is governed by the parameter  $A$ . Obviously the sequence has the property to converge with a constant rate. The unrestricted use of this sequence for the change of  $r$  after an unsuccessful search step however, bears the danger that  $r^{(u)}$  diminishes too fast toward zero before a success is obtained due to the statistical nature of the process.

It is well known that the fastest diverging sequence is the harmonic sequence

$$r^{(u)} = \frac{r^{(1)}}{u} \quad u = 1, 2, \dots \quad (2-57)$$

It has the property

$$\lim_{u \rightarrow \infty} \frac{r^{(1)}}{u} = 0 \quad (2-58)$$

$$\sum_{u=1}^{\infty} \frac{r^{(1)}}{u} = \infty \quad (2-59)$$

Comparing the two sequences we find that for  $0 < A < 0.5$  the sequence in (2-54) stays first above the harmonic sequence and falls below at a certain finite  $u$ .

The combination of the two sequences by imposing the harmonic sequence as the lower boundary satisfies our general requirements that at successive failures the vector length  $r$  is reduced such that in a finite number of steps it is  $\frac{r^{(u)}}{a^{(m)}} < 2$ , however,  $\frac{r^{(u)}}{a^{(m)}}$  approaching zero with a decreasing rate of convergence as  $u$  increases.

We defer the discussion of the parameter  $A$  for the moment and assume the other extreme of an initial condition that the length  $r$  is much smaller than the unknown distance, i.e.,  $\frac{r^{(1)}}{a^{(1)}}$  is close to zero. From Figure 2-6 we see that in this case the probability of success approaches  $P_S \rightarrow 0.5$ , however, Figure 2-7 shows that the expected amount of success is much smaller than the maximal obtainable for  $P_{Sopt}$ . In order to increase the length  $r$ , we must have an adaptation rule for successful steps

$$r^{(m+1)} = r^{(m)} \cdot C, \quad C > 1 \quad (2-60)$$

After the general discussion of the two rules, namely to shorten the length  $r$  in case of unsuccessful and to enlarge  $r$  in case of success we now state our objectives more precisely. Since  $r$  is changed after each search step and we never can adapt complete information on the distance we cannot achieve the optimal speed of convergence obtained for the ideal hypothetical case under 2.3.2. However, as Figure 2-7 indicates, by proper choice of the parameters  $A$  in (2-54) and  $C$  in (2-60), it is possible to achieve in the real case an average amount of displacement  $E \left[ \frac{\Delta a}{a} \right]$  real close to  $E \left[ \frac{\Delta a}{a} \right]_{opt}$  in Figure 2-7. Let's denote the vector length after the  $m_s$ -th success as  $r_S$  and define

$$\frac{r_S}{a_S} = \left(\frac{r}{a}\right)_{\text{opt}} + \Delta_1, \quad \Delta_1 > 0 \quad (2-61)$$

where  $\left(\frac{r}{a}\right)_{\text{opt}}$  depends on the dimension and is given in Table 2-2. Let's assume that the following number of expected successive unsuccesses and the parameter A is such that the next success can be expected at

$$\frac{r_S \cdot (1-A)^{(u)}}{a_S} = \left(\frac{r}{a}\right)_{\text{opt}} - \Delta_2, \quad \Delta_2 > 0 \quad (2-62)$$

where  $r_S \cdot (1-A)^{(u)}$  represents the value of the length r after formula (2-54) is applied u times. From Figure 2-7 we see, since the functions  $E \left[ \frac{\Delta a}{a} \right] = E \left( \frac{r}{a}, n \right)$  are for small  $\Delta_1, \Delta_2$ , approximately symmetric with respect to  $\left(\frac{r}{a}\right)_{\text{opt}}$ , that we would have optimal conditions for the real case and given  $\Delta_1, \Delta_2$ . Moreover for  $\Delta_1, \Delta_2 \rightarrow 0$  we would approach the theoretical value  $E \left[ \frac{\Delta a}{a} \right]_{\text{opt}}$ . Obviously the fractions  $\Delta_1, \Delta_2$  depend on A, and the expected number of successive unsuccesses.  $\Delta_1, \Delta_2 = 0$  would require  $A = 0$ , that is, no adaptation is possible anymore. Thus, A must be greater than zero. What is an optimal value for A? The exact solution of this problem can only be obtained by very tedious and costly computations. We shall come close to the optimal conditions by assuming A as a function of the theoretical optimal parameters obtained in 2.3.2. We found that the optimal probability of success approaches the value  $P_{\text{Sopt}} = 0.27$  for large n, where a significant deviation occurs practically only for  $n < 5$  (or even  $n < 4$ , as seen in Table 2-2). That means that about every fourth step is successful, or more exact in the average out of  $\frac{1}{P_{\text{Sopt}}}$  steps u =

$\frac{1-P_{\text{Sopt}}}{P_{\text{Sopt}}}$  successive unsuccesses occur. The expected displacement at a successful step

is given by

$$E_{\text{Sopt}} = \frac{E \left[ \frac{\Delta a}{a} \right]_{\text{opt}}}{P_{\text{Sopt}}} \quad (2-63)$$

In the real case we can only expect  $E_{\text{Sreal}}$  which is dependent on the search parameters. For instance, with the very unfavorable initial condition  $\frac{r^{(1)}}{a^{(1)}}$  close to zero the expected value  $E_{\text{Sreal}}$  for the first success is very small. For strong proof of con-

vergence we will not take into account the amount of success at each successful step. The loss resulting from this assumption should be compensated at least in one unsuccessful step, that is the quasi-optimal conditions are restored after one step. Referring to the formulas (2-61) and (2-62) the value  $\left(\frac{r}{a}\right)_{opt}$  is shifted towards  $\left(\frac{r}{a}\right)_{opt} + \varepsilon$  ( $\varepsilon > 0$ ) by neglecting  $E_{Sreal}$ . If we assign the value

$$A = E_{Sopt} \quad (2-64)$$

the unsymmetry  $\varepsilon$  can be reduced by at most one step. Using the relations for large  $n$  in Table 2-2 we find the relation

$$A = \frac{1}{0.27} \cdot \left(1 - \sqrt{1 - \frac{0.406}{n}}\right) \quad (2-65)$$

For small  $n$  the relation (2-63) using the values in Table 2-2 can be used for a more exact computation.

The parameter  $A$  is a function of the dimension  $n$  and has a characteristic proportional to  $E \left[ \frac{\Delta a}{a} \right]_{opt} = E \left[ \frac{\Delta a}{a} \right]_{opt}(n)$ . It remains to derive the factor  $C$  such that the assumed conditions in (2-61) and 2-62) with the correction

$\left(\frac{r}{a}\right)_{opt} \rightarrow \left(\frac{r}{a}\right)_{opt} + \varepsilon$  are obtained. We require that the real expected value of the ratio  $\left(\frac{r}{a}\right)_{real}$  for all possible sequences of successive unsuccesses is identical to the initial value  $\frac{r_S}{a_S}$  where the sequences of unsuccesses started, that is

$$E \left[ \frac{r}{a} \right]_{real} \stackrel{!}{=} \frac{r_S}{a_S} \quad (2-66)$$

According to the previous discussion we may select as appropriate value for  $\frac{r_S}{a_S}$

$$\frac{r_S}{a_S} = \left(\frac{r}{a}\right)_{opt} \cdot \frac{1}{(1-A) \left( \frac{1}{2} \cdot \frac{1 - P_{opt}}{P_{opt}} \right)} \quad (2-67)$$

The considered adaptation rules are in case of failure or unsuccess:

$r = r(1-A)$  or equivalent  $r = r_S \cdot (1-A)^u$  with the boundary condition

$$r \stackrel{!}{=} \frac{r_S}{u+1} \quad (2-68)$$

where  $r_s$  is the initial condition after the  $s$ -th unsuccess, and  $u$  the number of successive unsuccesses

In case of success:

$$r = r \cdot C \quad (2-69)$$

The expected value  $E \left[ \frac{r}{a} \right]_{\text{real}}$  is obtained to:

$$E \left[ \frac{r}{a} \right]_{\text{real}} = C \cdot \frac{r_s}{a_s} \left\{ \sum_{i=1}^K (1-A)^i W_i + \sum_{i=K+1}^{\infty} \frac{1}{i} W_i \right\} \quad (2-70)$$

with the condition for  $K$ :

$$r_s (1-A)^K \geq \frac{r_s}{K+1}$$

but

$$r_s (1-A)^{K+1} < \frac{r_s}{K+2} \quad (2-71)$$

$W_i$  is the probability for the occurrence of a success after  $i-1$  unsuccesses. As the random trials are assumed to be independent it is

$$W_1 = P_{S1}$$

$$W_2 = P_{S2} q_1 \text{ with } q_1 = 1 - P_{S1}$$

...

$$W_i = P_{Si} q_1 q_2 \dots q_{i-1} \quad (2-72)$$

where  $q_0 = 1$ , and  $q_{i-1} = 1 - P_{Si-1}$

Further it is:

$$P_{Si} = \frac{\int_0^{\alpha_i} \sin^{n-2} \phi d\phi}{2 \int_0^{\pi} \sin^{n-2} \phi d\phi} \quad (2-73)$$

where

$$\alpha_i = \arccos \left\{ \frac{1}{2} \cdot \left( \frac{r}{a} \right)_i \right\} \quad (2-74)$$

with

$$\left(\frac{r}{a}\right)_i = \frac{r_S}{a_S} (1 - A)^{i-1} \quad (2-75)$$

Substituting (2-67) in (2-75) we obtain

$$\left(\frac{r}{a}\right)_i = \left(\frac{r}{a}\right)_{opt} (1 - A)^{\left(i - \frac{1-P_{opt}}{2P_{opt}} - 1\right)} \quad (2-76)$$

Combining (2-66) and (2-70) we find for the parameter C

$$C = \frac{1}{\sum_{i=1}^K (1 - A)^i \cdot W_i + \sum_{i=K+1}^{\infty} \frac{1}{i} W_i} \quad (2-77)$$

The adaptive search process converges with certainty to the final solution since C is determined such that (2-66) is satisfied. The adaptation parameters A, C are, clearly, only quasi-optimal. The exact optimization process would be to optimize  $E \left[ \frac{\Delta a}{a} \right]_{real}$  subject to the variation of A and  $\frac{r_S}{a_S}$  in equation (2-67) and repetitive computation of the corresponding C from (2-77) which assures convergence.

Note: For the computation of C the infinite series can be approximated by terminating the computing process when most of the possible sequences of successive unsuccesses are included, precisely when

$$\sum_{i=1}^T W_i + \epsilon = 1, \epsilon > 0 \quad (2-78)$$

where  $\epsilon$  can be predetermined, for instance  $\epsilon = 10^{-3}$ .

The computation of the parameter  $C = C(n, A)$  corresponding to the parameter A in (2-64) has been delayed until needed, since the computation of C is costly. To prove the operation a simplified version of (2-77) has been assumed for  $n = 10$ , where  $P_{Si}$  in (2-73) was approximated as a linear function of  $\left(\frac{r}{a}\right)_i$  in (2-74). As a result  $C = 1.28$  was obtained.

Since exhaustive simulations were made in an earlier investigation for assumed values  $A = 0.1$ ,  $C = 1.3$ , no further simulations were performed with the computed value  $C = 1.28$  for  $A = 0.081$ . For  $A = 0.1$ ,  $C = 1.3$  in the average  $M = 338$  search steps were needed for  $\frac{a^{(n)}}{a^{(1)}} = 10^{-3}$ . The theoretical limit under 2.3.2 shows  $M_{min} = 298$ .

With the quasi-optimal value  $C$  based on equation (2-77) better results will be obtained, since the assumed values for  $A$ ,  $C$  ( $A = 0.1$ ,  $C = 1.3$ ) are less optimal. The reason for this is, that  $\frac{r_S}{a_S}$  is shifted towards a higher ratio which satisfies the convergence condition because the expected value  $E \left[ \frac{r}{a} \right]_{\text{sim}} > E \left[ \frac{r}{a} \right]$  in (2-66). However,  $E \left[ \frac{\Delta a}{a} \right]_{\text{sim}}$  is smaller than the quasi-optimal value due to the unsymmetric shift.

With proper computation of  $C$  in (2-77) one can expect results for the average number of search steps which are less than 10% higher than the theoretical values in Table 2-3. This is true for dimensions  $n \geq 4$ . For  $n = 2, 3$  the step number deviates more from the theoretical values due to the fact recognizable in Figure 2-7 that the shape of the characteristic for  $E \left[ \frac{\Delta a}{a} \right]$ ,  $n = 2, 3$  is more sensitive to the unsymmetric shift by an amount  $\epsilon$  to

$$\left( \frac{r}{a} \right)_{\text{opt}} + \epsilon \quad (2-79)$$

#### 2.4.1 Adaptive Search Algorithm Including Directional Information

The previously derived search algorithm does not require or assume any directional information. In Chapter 2.3 we concluded, however, that directional information is very effective in accelerating the speed of convergence. Given a certain algorithm for the adaptation of directional information one can derive, similar to the cases in Chapter 2.3, the theoretical limits, and can furthermore derive the corresponding quasi-optimal adaptive search algorithm using the principles applied for the non-directional search algorithm in the previous section. Clearly, since the random trials become dependent, because the information of the outcome of previous trials is used to control the probability density of the angle  $\phi$ , the computations will become much more complex. Again, derivations can only be performed for a specific class of functions, say hyperspherical functions. No such derivations have been performed during this work.

However, directional information has been used in simulations for which the adaptive search algorithm derived above has been applied. The adaptation of the direction was based on the adaptation of the mean value vector similar to the method described by Matyas in [2]. However, besides the normal distribution, applied in

[2] , some other distributions like the triangle distribution and similar ones, have been used.

Since the directional information increases the probability of success, i.e., all curves in Figure 2-6 are shifted towards higher values, the convergence of the adaptive search algorithm (based on non-directional information) is assured. However, the process stabilizes in a region for  $\frac{r}{a}$  which is off the optimal conditions. Still, the results obtained for the step number were below the theoretical limits obtained in 2.3.2, but higher than the minimal step number obtained in 2.3.3. It can be concluded that an optimal adaptation algorithm based on the adaptation of the search vector length and search vector direction will result in a speed of convergence which is at least as good or even better than the values obtained in 2.3.3.

The reason why the derivation of an adaptation algorithm has been delayed is the fact that the imposed assumptions on the derivations are still too restrictive and as we have now gained insight into the behavior of the adaptation process, we can study the more complex problem of allowing disturbances in the evaluation of the quality function.

## 2.5 CONDITIONS AND LIMIT OF SPEED OF CONVERGENCE FOR HIGHLY DISTURBED HYPERSPHERICAL FUNCTIONS

Up to now we assumed that the components of the state vector,  $X$ , can be exactly assigned and that the corresponding function value,  $Q(X)$ , can be exactly computed. In the following we shall investigate the case that the functional values,  $Q$ , are replaced by disturbed measurements. Again, in order to get an answer to this problem we consider hyperspherical functions.

We impose the same assumptions as under 2.3.2 except we assume that all function evaluations are disturbed by Bernoulli noise; i.e., instead of observing  $Q(X^{(m)})$  we observe

$$Z(X^{(m)}) = Q(X^{(m)}) + \beta^{(m)} \cdot \delta_N \quad (2-80)$$

where  $\beta^{(m)}$  are independent random variables

$$\beta^{(m)} = \begin{cases} +1 \\ -1 \end{cases} \text{ with probability } P_{\beta} = \begin{cases} 0.5 \\ 0.5 \end{cases} \quad (2-81)$$

and  $\delta_N > 0$  is the amount of disturbance. We apply the same basic search and evaluation procedure as described under 2.2; however, in the test function in (2-21)  $Q$  has to be replaced by  $Z$ ; i.e., at each search step we test

$$Z(X^{(m)} + \xi^{(m)}) < Z(X^{(m)}) \cdot \{1 - \xi\} \quad (2-82)$$

where

$$Z(X^{(m)} + \xi^{(m)}) = Q(X^{(m)} + \xi^{(m)}) + \beta^{(m+1)} \cdot \delta_N \quad (2-83)$$

$$\text{and } Z(X^{(m)}) = Q(X^{(m)}) + \beta^{(m)} \cdot \delta_N \quad (2-84)$$

The two random variables,  $\beta^{(m+1)}$ ,  $\beta^{(m)}$ , are independent at each search step; i.e., in case of successive unsuccesses the relative optimal value,  $Z(X^{(m)})$ , is also affected randomly at each search step. This is the worst type of noise which can be considered.

We assume that the amount,  $\delta_N$ , of the noise is very much greater than the difference,  $|Q^{(1)} - Q_{opt}|$ , where  $Q^{(1)}$  is the undisturbed initial value. Since each random variable in (2-83) and (2-84) can assume the binary values, +, -, there are four possible combinations (Table 2-6) each occurring with equal probability.

TABLE 2-6  
POSSIBLE COMBINATIONS FOR THE VALUES OF THE  
RANDOM VARIABLES IN (2-83) AND (2-84)

	$\beta^{(m)}$	$\beta^{(m+1)}$	Probability of Occurrence
(a)	+	+	0.25
(b)	-	-	0.25
(c)	-	+	0.25
(d)	+	-	0.25

We discuss the four cases further:

$$\begin{aligned} \text{(a)} \quad Z(X^{(m)}) &= Q(X^{(m)}) + \delta_N \\ Z(X^{(m)} + \xi^{(m)}) &= Q(X^{(m)} + \xi^{(m)}) + \delta_N \end{aligned} \quad (2-85)$$

Since both values, Z, are disturbed by the same sign and amount, the noise has no effect. The situation is identical to the case in 2.3.2.

$$\begin{aligned} \text{(b)} \quad Z(X^{(m)}) &= Q(X^{(m)}) - \delta_N \\ Z(X^{(m)} + \xi^{(m)}) &= Q(X^{(m)} + \xi^{(m)}) - \delta_N \end{aligned} \quad (2-86)$$

Here the same applies as discussed under (a), above. The noise has no effect.

$$\begin{aligned} \text{(c)} \quad Z(X^{(m)}) &= Q(X^{(m)}) - \delta_N \\ Z(X^{(m)} + \xi^{(m)}) &= Q(X^{(m)} + \xi^{(m)}) + \delta_N \end{aligned} \quad (2-87)$$

As we base our discussion on minimization and  $\delta_N \gg |Q(X^{(m)}) - Q_{opt}|$  (2-88)

no success is possible. The noise disturbs the observation completely. However, no move away from the optimum can occur.

$$\begin{aligned}
 \text{(d) } Z(X^{(m)}) &= Q(X^{(m)}) + \delta_N \\
 Z(X^{(m)} + \xi^{(m)}) &= Q(X^{(m)} + \xi^{(m)}) - \delta_N
 \end{aligned}
 \tag{2-89}$$

For this case the probability of success is

$$P_S^{(d)} = P_{SD}^{(d)} + P_{SN}^{(d)} = 1
 \tag{2-90}$$

where  $P_{SD}$  is the part which corresponds to real successes and  $P_{SN}$  the part which leads to "noisy" successes causing moves away from the optimum.

The question to be answered in the following is:

- (a) Under what conditions can the search process converge?
- (b) What is the theoretical limit for the speed of convergence?

Note that we assume complete information on the distance,  $a^{(m)}$ , but no directional information at all. Using the nomenclature as under 2.3.2 and the relations derived from Figure 2-5 we find the following.

The probability that a search step leads to a real success resulting in a move towards the optimum is obtained to

$$P_{SD} = 0.75 \frac{\int_0^{\alpha} \sin^{n-2} \phi \, d\phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi \, d\phi}
 \tag{2-91}$$

where the cases, (a), (b) and (d), contribute the same fraction, 0.25.

For the probability of "noisy" successes we find from (d) and relation (2-90)

$$P_{SN} = 0.25 \frac{\int_{\alpha}^{\pi} \sin^{n-2} \phi \, d\phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi \, d\phi}
 \tag{2-92}$$

The behavior of  $P_{SD}$  and  $P_{SN}$  as a function of  $\frac{r}{a}$  has been evaluated for the interesting interval and various dimensions. Figure 2-10 shows the functional behavior for two different dimensions,  $n = 6$  and  $n = 10$ . For  $\frac{r}{a} \rightarrow 0$  the probability,  $P_{SD}$ , approaches the value,  $P_{SD} = 0.375$ , and the "noisy" part,  $P_{SN}$ , goes towards  $P_{SN} = 0.125$  for every dimension. For an increasing ratio,  $\frac{r}{a}$ , the probability,  $P_{SD}$ , is monotonically decreasing and reaches the value  $P_{SD} = 0$  for  $\frac{r}{a} = 2$  for every dimension. However, independently increasing  $n$  reduces  $P_{SD}$ ; i.e., with increasing dimensionality the function,  $P_{SD} = P_{SD}(\frac{r}{a})$  decreases faster.

Contrary to this the "noisy" part,  $P_{SN}$ , increases with increasing  $\frac{r}{a}$  and approaches the value,  $P_{SN} = 0.25$ , for every dimension. Because of the assumption  $\delta_N \gg |Q(X^{(m)}) - Q_{opt}|$  the probability  $P_{SN}$  is

$$P_{SN} = 0.25 \text{ for } \frac{r}{a} \geq 2 \quad (2-93)$$

as long as the above assumption for  $\delta_N$  holds.

For a certain ratio,  $\frac{r}{a}$ , and dimension,  $n$ , the two probabilities,  $P_{SD}$  and  $P_{SN}$ , become equal. Below that value,  $\frac{r}{a}$ ,  $P_{SD}$  is superior and above that value  $P_{SN}$  is greater than  $P_{SD}$ . Clearly, only the overall probability of success,  $P_S$ , is observable as

$$P_S = P_{SD} + P_{SN} \quad (2-94)$$

The behavior of  $P_S$  can be seen in Figure 2-10. Of specific interest is the expected value for the fractional distance decrement per step,  $E\left[\frac{\Delta u}{a}\right]$ . The noise causes the relative optimum to move away from the true optimum under certain conditions. If the "noisy" moves are overwhelming the expected value  $E\left[\frac{\Delta u}{a}\right]$  will be negative and the search process does not converge.

Corresponding to the theory under 2.3.2 we formulate  $E\left[\frac{\Delta u}{a}\right]$  to

$$E\left[\frac{\Delta u}{a}\right] = 0.5 \int_0^{\alpha} \frac{\Delta u}{a} p(\phi) d\phi + 0.25 \int_0^{\pi} \frac{\Delta u}{a} p(\phi) d\phi \quad (2-95)$$

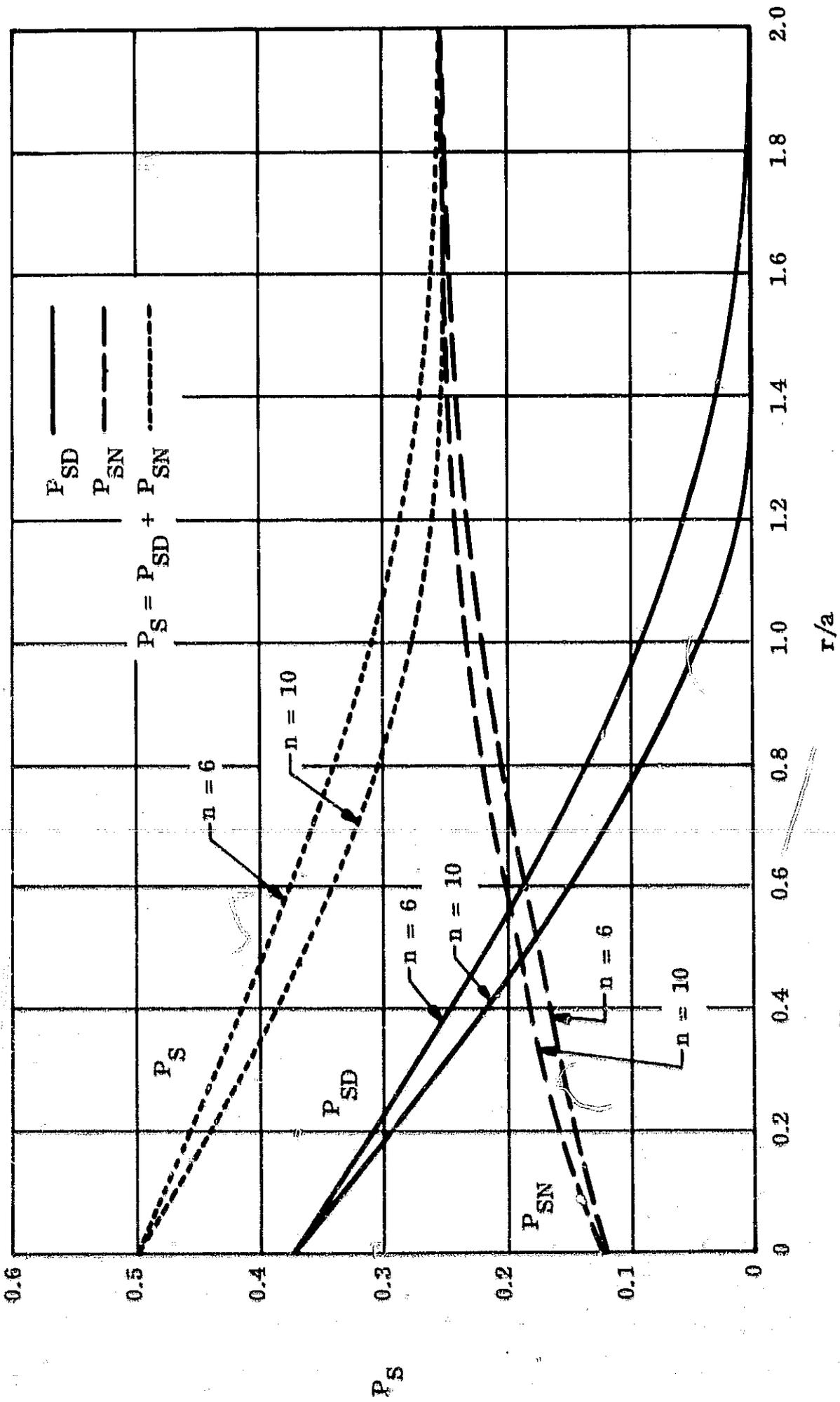


Figure 2-10. Probability of Success versus  $r/a$  for Highly Disturbed Hyperspherical Functions

where  $\frac{\Delta a}{a}$  and  $p(\phi)$  are defined under 2.2.2. The first integral in (2-95) relates to the cases in (2-85), and (2-86). The second integral results from case (2-89). Of particular interest is the behavior of  $E\left[\frac{\Delta a}{a}\right]$  as a function of the ratio  $\frac{r}{a}$ . Figure 2-11 shows the dependency for the two dimensions,  $n = 6$  and  $n = 10$ . It is seen that after a maximal value for a relatively small  $\left(\frac{r}{a}\right)_{opt}$  the expected fractional decrement is continuously decreasing, passes the value, 0, and decreases steadily with increasing  $\frac{r}{a}$ . The search process can only converge for  $E\left[\frac{\Delta a}{a}\right] > 0$ . As in 2.3.2, optimal conditions exist for the speed of convergence. The optimal conditions for some dimensions,  $n$ , are listed in Table 2-7.

TABLE 2-7  
OPTIMAL PARAMETERS AND MINIMAL NUMBER OF  
SEARCH STEPS FOR HIGHLY DISTURBED  
HYPERSPHERICAL FUNCTIONS

$n$	$\left(\frac{r}{a}\right)_{opt}$	$\alpha_{opt}$ (deg)	$E\left[\frac{\Delta a}{a}\right]_{opt}$	$P_{Sopt}$	$M_{opt}$ $\frac{a^{(m)}}{a^{(1)}} = 10^{-3}$
2	0.560	73.2	0.0478	0.4533	141
4	0.289	81.7	0.0153	0.4539	450
6	0.213	83.9	0.0080	0.4551	774
8	0.174	85.0	0.0062	0.4556	1103
10	0.153	85.6	0.0048	0.4561	1433

Besides the optimal parameters the theoretical minimal step number,  $M$ , is listed.  $M$  can be computed with formula (2-35). As for the results under 2.3.2 and shown in Table 2-3  $M$  was calculated for the ratio  $\frac{a^{(m)}}{a^{(1)}} = 10^{-3}$ . The relationship  $M = M(u)$  again is linear like under 2.3.2; however, the minimal step number is about 4.8 times greater than for the undisturbed case.

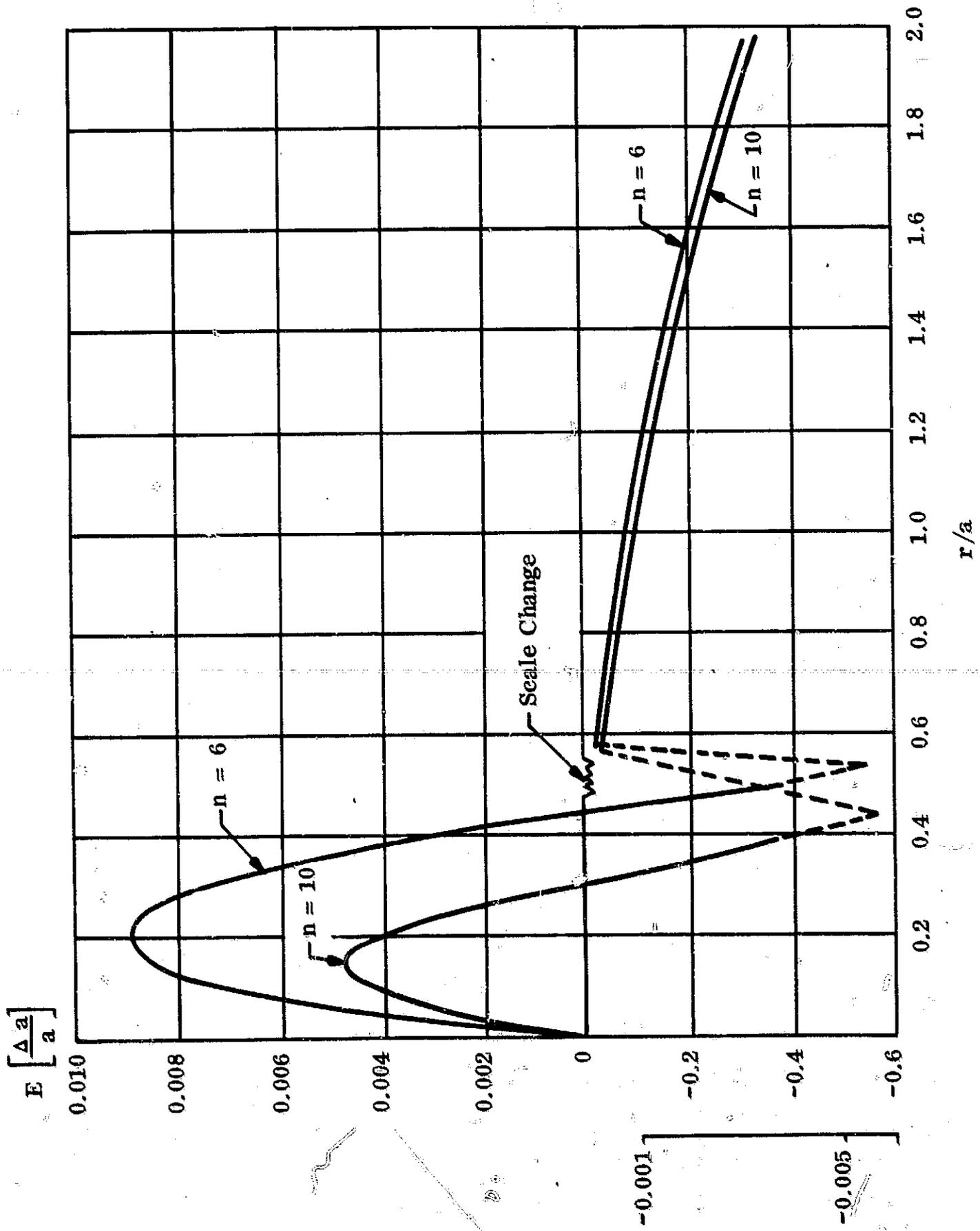


Figure 2-11. Expected Value of Fractional Distance Decrement for Highly Disturbed Hyperspherical Functions

## 2.6 ADAPTIVE SEARCH ALGORITHMS FOR A LARGER CLASS OF DISTURBED QUALITY FUNCTIONS

### 2.6.1 Adaptive Algorithms for Disturbed Hyperspherical Functions

The results of the previous chapter show that certain similarities exist for the disturbed hyperspherical functions with the results obtained in 2.3.2 for the undisturbed case. Because the probability of success,  $P_S$ , (see Figure 2-10) is also for the "noisy" case a decreasing function with respect to  $\frac{r}{a}$  in the interesting interval for which  $E \left[ \frac{\Delta a}{a} \right]$  is positive around the maximum, we can use the theory in 2.4 to establish an adaptive algorithm for highly disturbed hyperspherical functions. The condition is again that, for a given parameter,  $A$ , the parameter,  $C$ , has to be determined so that the search process stabilizes around  $\left(\frac{r}{a}\right)_{opt}$  after the beginning transient period. Obviously, the formulas for  $A$ ,  $\frac{r_S}{a_S}$ , and  $C$  in Chapter 2.4, which were applied for the disturbed case, satisfy the requirement for convergence if we use the corresponding parameters,

$$E \left[ \frac{\Delta a}{a} \right]_{opt}, E_{S_{opt}}, \left(\frac{r}{a}\right)_{opt}, P_{S_{opt}} \text{ and } P_{S_i}$$

The formula for  $P_{S_i}$  in (2-73) changes however to

$$P_{S_i} = 0.5 \frac{\int_0^{\alpha_i} \sin^{n-2} \phi d\phi}{2 \int_0^{\pi/2} \sin^{n-2} \phi d\phi} + 0.25 \quad (2-96)$$

where (2-96) is obtained by combining (2-91), (2-92), and (2-94).

At this point, we could derive an adaptive algorithm based on the adaptation of the search vector length and search vector direction. The general steps of such a derivation were outlined in the last part of Chapter 2.4. However, no convergence could be assured if the algorithm were applied to problems other than the hyperspherical type. One can even show the divergence for certain problems. The reason is because the algorithm would have been approximately optimized for hyperspherical contours using the functional relationships  $p_S = p_S \left(\frac{r}{a}, n\right)$  and  $E \left[ \frac{\Delta a}{a} \right] = E \left[ \frac{\Delta a}{a} \right] \left(\frac{r}{a}, n\right)$ .

### 2.6.2 Adaptive Search Algorithm for Disturbed Quality Functions

The objective is the derivation of an adaptive search algorithm applicable for disturbed arbitrary quality functions. In the following we will loose the restriction of the function type and shall derive a convergent adaptive search algorithm for a larger class of quality functions.

We refer to the derivations under 2.5 and Figure 2-11 and impose the following conditions. Assume a class of ill-defined or disturbed functions,  $F$ . Every member of  $F$  has the general property that there exists an expected value for the fractional decrement  $E \left[ \frac{\Delta a}{a} \right] > 0$  in the interval,  $0 < \frac{r}{a} < b$ ,  $b > 0$ , where  $r$  is the length of the search vector length of a search procedure. For  $\frac{r}{a} \geq b$   $E \left[ \frac{\Delta a}{a} \right]$  might be zero or negative. Clearly, the class of functions,  $F$ , is now completely dependent on the search procedure. A simpler algorithm will restrict  $F$  to a smaller number. It should be noted that we do not impose any specific conditions on the noise. For instance, we allow biased noise also, as long as the required condition on  $E \left[ \frac{\Delta a}{a} \right]$  is satisfied. Furthermore, directional information may increase the number of elements in the allowed class,  $F$ .

For simplicity we will first derive the algorithm without directional information because the latter can be added later without modifications.

Assume the search process starts at an initial condition such that  $E \left[ \frac{\Delta a}{a} \right]$  is negative (see Figure 2-11); i.e., the noise causes moves away from the true optimum. In order to reduce the ratio  $\frac{r}{a}$  to the region where  $E \left[ \frac{\Delta a}{a} \right]$  becomes positive, we must have an algorithm which 1) performs the reduction of  $r$  in a finite number of steps and 2) prevents  $r$  from going to zero with a rate higher than the rate the distance,  $a$ , approaches zero. The contradiction in this requirement can be removed by splitting off the algorithm into two adaptation rules, namely a rule for successful steps and a rule for the case of failures.

We derived in Chapter 2.4 equation (2-68) for hyperspherical functions the rule in case of failure:

$$r = r (1 - A),$$

or equivalent  $r = r_S (1 - A)^u$  (2-97)

with the boundary

$$r \geq \frac{r_S}{u + 1}$$

This rule satisfies a part of our requirements. The harmonic sequence

- a. Prevents  $r$  from going to zero too fast, if we are in the interval  $0 < \frac{r}{a} < b$  and we are in a cycle of successive failures.
- b. Reduces  $r$  in case of failures in a finite number of steps, if we are in the interval  $\frac{r}{a} \geq b < \infty$

What remains to be found is the rule for successes which satisfies also the two requirements stated. As the probability of success  $P_S$  can be greater than zero for  $\frac{r}{a} \geq b$  the simple rule in (2-69)  $r = r \cdot C$  is not satisfactory, if  $P_S$  is unknown and arbitrary  $0 \leq P_S \leq 1$ , which we will assume.

We may further consider the two cases that the shape of the function  $E \left[ \frac{\Delta a}{a} \right] = E \left[ \frac{\Delta a}{a} \right] \left( \frac{r}{a}, n \right)$  is known or unknown. If the shape is known we can calculate the real value,  $E \left[ \frac{\Delta a}{a} \right]_{\text{real}}$ , the following rule

$$r = r^{(1)} \cdot (1 - B)^{m_S} \quad \text{if success, } (m_S \text{ is the number of } \quad (2-98)$$

successful steps)

$$\text{where } 0 < B \leq E_{s \text{ real}} \quad (2-99)$$

would then satisfy the requirements, since

- a. it reduces  $r$  with a rate not faster than the distance,  $a$ , is reduced, and
- b. reduces  $r$  fast enough so that the region  $0 < \frac{r}{a} < b$  can be reached in a finite number of steps.

As it is seen from (2-99) the values  $E \left[ \frac{\Delta a}{a} \right] P_S$  have not to be known exactly since any  $B$  in the stated boundaries is satisfactory. Optimal speed of convergence is obtained for  $B = E_{s \text{ real}}$ . For  $B > E_{s \text{ real}}$  the

search process stops before the optimum is reached, since  $r$  approaches zero with a faster rate than  $a$ .

In case that no information on  $E \left[ \frac{\Delta a}{a} \right]$  and  $P_S$  is available, the fastest sequence satisfying the requirements is the harmonic sequence resulting in the rule

$$r = \frac{r^{(1)}}{m_S} \quad \text{if success and no information on } E_{S \text{ real}} \quad (2-100)$$

The requirements are satisfied because

The harmonic sequence satisfies the case  $E_{S \text{ real}} \rightarrow 0$ , since

$$\sum_{m_S=1}^{\infty} \frac{r^{(1)}}{m_S} = \infty$$

We can further refine the rule for the case of success in (2-100) based on the discussion in 2.4 and equation (2-57). Using (2-100) as boundary rule, any sequence is allowed which has the property of staying above the sequence in (2-100) and converging; i.e.,  $\sum_{i=1}^{\infty} e_i < \infty$  where  $e_i$  are the elements of the sequence. We

found that the significant property of the sequences stated in (2-54) and (2-98) is the constant rate of convergence. A combination with the harmonic sequence as it is done in (2-97) for the case of failure results in a sequence which normally behaves better than the harmonic sequence in the interval after the starting point.

For the case that nothing is known about  $E_{S \text{ real}}$  we may state with convenience the combined rule for successes similar to (2-97):

$$r_S = r^{(1)} (1 - A')^{m_S} \quad (2-101)$$

with the boundary

$$r_S = \frac{r^{(1)}}{m_S + 1}$$

where  $A'$  may be selected independently of  $A$  in (2-97). Useful values for  $A'$  (as with  $A$ , if not determined by (2-64)) are in the interval  $0 < A < 0.5$ . For instance, for  $A = 0.1$  the initial value of the first sequence in (2-101) is reduced to  $1/10$  in about  $m_S = 20$  steps and still satisfies for this  $m_S$  the boundary condition

$$r \geq \frac{r^{(1)}}{21}.$$

To summarize: if no information on  $E_{S \text{ real}}$  is a priori given, the simplest rules for the adaptation of the search vector length,  $r$ , are the boundary sequences in (2-97) and (2-101), denoted as ASR - 1 (Adaptive Search Rule No. 1)

ASR - 1:

$$r = \frac{r_S}{u + 1} \quad \text{if failure} \quad (2-102)$$

$$r_S = \frac{r^{(1)}}{m_S + 1} \quad \text{if success}$$

where  $u$  is the number of successive failures and  $m_S$  the number of successes.

A refined set of rules under the same assumption that no a priori information is given is

ASR - 2:

$$\left. \begin{aligned} r &= r_S (1 - A)^u \\ r &\geq \frac{r_S}{U + 1} \end{aligned} \right\} \text{if failure} \quad (2-103)$$

$$\left. \begin{aligned} r_S &= r^{(1)} (1 - A')^{m_S} \\ r_S &\geq \frac{r^{(1)}}{m_S + 1} \end{aligned} \right\} \text{if success}$$

where  $A, A'$  are constants discussed above.

In case some a priori information was available on  $F_S$  real we found previously the set of rules

ASR-3:

$$\left. \begin{aligned} r &= r_S (1 - A)^u \\ r &\geq \frac{r_S}{u + 1} \end{aligned} \right\} \text{if failure} \quad (2-104)$$

$$r_S = r^{(1)} (1 - B)^{m_S} \text{ if success}$$

where B is defined in (2-99).

The "payment" for needing no a priori information at all in ASR-1 and ASR-2 is a slower rate of convergence compared with the more or less optimal convergence at a constant rate for ASR-3.

Theoretically, complete convergence for all algorithms is achieved for the limit  $m \rightarrow \infty$ . However, in practice, since we have to assume  $\epsilon > 0$  in (2-15) for the function evaluations because of the limited computer accuracy, the search process terminates at a finite step number. More exactly, the search terminates when a distance,  $a$ , is reached which corresponds to the value,  $\epsilon$ , in (2-15).

The adaptation rules above can be further extended including adaptation of the vector length by the principles discussed under 2.4, if more a priori information is given. The result would be a further increased speed of convergence.

Finally, we note that the introduction of the adaptation of the direction of the search vector in addition to the vector length will result in a set of rules in addition to those stated, without modifying the structure of the formulas. For derivation of the rules for direction adaptation the same general requirements as stated in this chapter for the length adaptation have to be imposed. It is clear that the directional information will increase the speed of convergence. Moreover, even more important, it will increase the number of solvable problems in the class,  $F$ . Some preliminary investigations using converging directional adaptation rules show promising results for highly disturbed quality functions. A detailed study is outstanding because of the complexity

of the problem already mentioned for the noiseless case under Chapter 2.4. Continuation of the study in this direction can be considered very vital.

## 2.7 PROOF OF OPERATION, SOLUTION OF PROBLEM NO. 1, COMPARISON WITH STOCHASTIC APPROXIMATION

### 2.7.1 Problem Definition

The following problem set by NASA, and denoted as Problem No. 1, will be solved using the developed adaptive search technique.

Given the two-dimensional unimodal function

$$Q = 2 - 0.2 |x_1| - 0.4 |x_2| = Q(X) \quad (2-105)$$

defined in the region  $\Omega$ : 
$$\begin{aligned} -10 \leq x_1 \leq +10 \\ -5 \leq x_2 \leq +5 \end{aligned}$$

The function defines a pyramid in the three-dimensional space  $(x_1, x_2, Q)$ . The peak of the pyramid is located at  $(0, 0, 2)$ .

The problem to be solved consists of finding the coordinates  $x_{1 \text{ opt}}, x_{2 \text{ opt}}$  that correspond to the maximum value of  $Q$ . However,  $Q$  in (2-105) cannot be observed directly; instead only a noisy functional value,  $Z$ , is obtained for each two-dimensional vector (or its end point),  $X$ .

$$Z(X) = Q(X) + \beta \delta_N \quad (2-106)$$

where the random variable,  $\beta$ , has the property

$$\beta = \begin{cases} +1 \\ -1 \end{cases} \text{ with probability } p_\beta = \begin{cases} 0.5 \\ 0.5 \end{cases}$$

and  $\delta_N \approx 0.2$

A search is to be started at two different initial points

$$1. \quad X^{(1)} = (x_1^{(1)}, x_2^{(1)}) = (8, 0)$$

$$2. \quad X^{(1)} = (x_1^{(1)}, x_2^{(1)}) = (8, 1)$$

For the comparison of the two methods, adaptive search - stochastic approximation, we will use for the end test the fact that we know the optimal point,  $X_{\text{opt}} = (0, 0)$ . The search process is terminated when the distance  $a^{(m)} = \sqrt{x_1^2 + x_2^2}$  reaches a value so that  $\frac{a^{(m)}}{a^{(1)}} \leq 10^{-3}$ . That is, we end the search when a search point falls in the circle with radius,  $a^{(m)}$ , and the center in the origin.

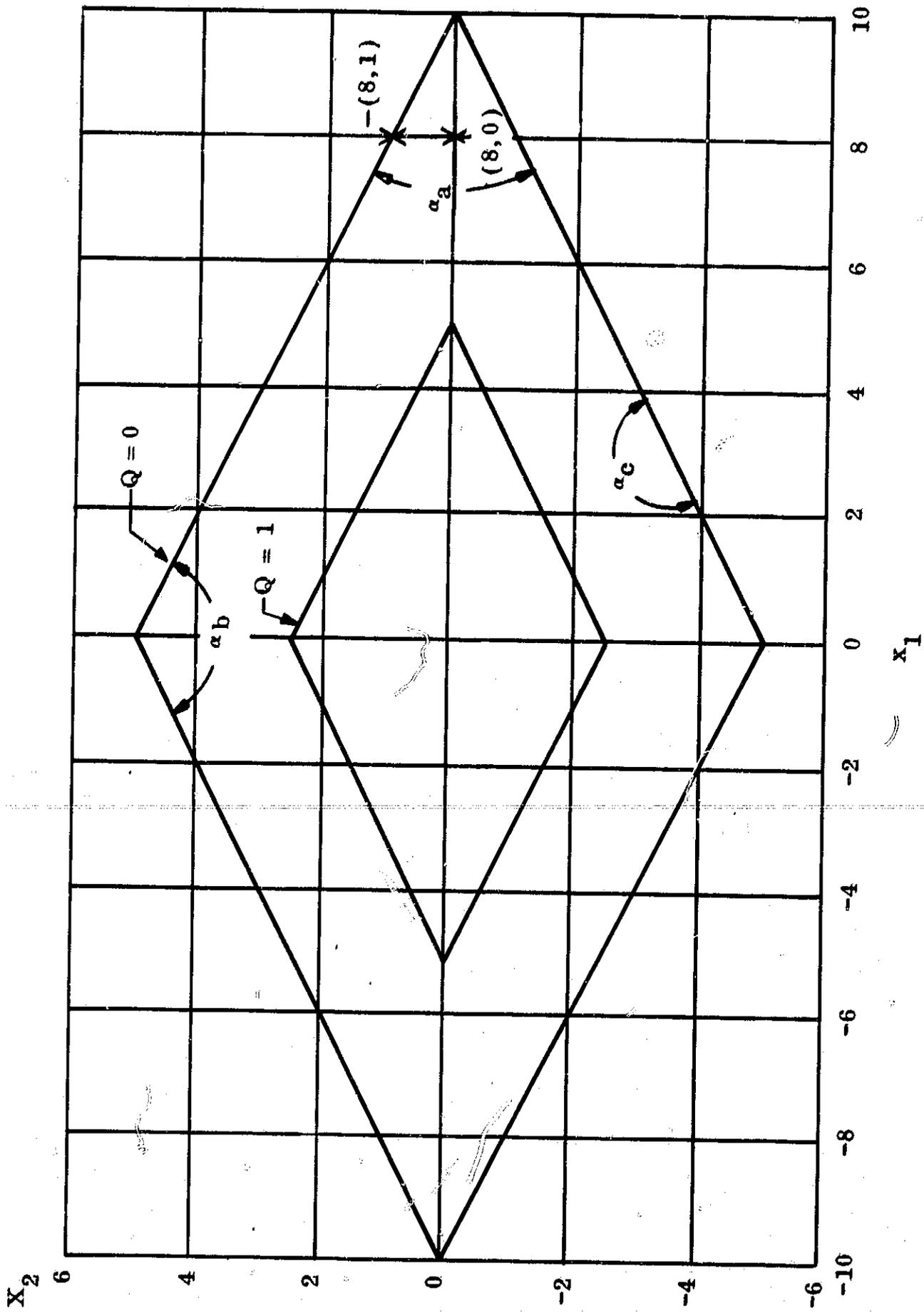


Figure 2-12. Equiquality Contours of Function  $Q(X)$  Given as Problem No. 1

It should be noted that in a general, real problem we would not have the information on the location of the peak and would have to use another criterion for the stop of the search.

Since each measurement  $Z(x_1, x_2)$  is disturbed by a relatively large amount of additive noise, no two searches from the same initial point to the same final region will be the same. In general, the number of steps needed is itself a statistical quantity and its distribution function is unknown. Therefore, it is necessary to run a number of searches to derive a firm estimate on which a comparison may be based. It is accepted practice in dealing with Gaussian distributions to take at least 30 sample searches before computing confidence level. In solving Problem No. 1 we have found that the step number is not normally distributed, so we have chosen to make 100 runs of each search in order to estimate the average number of steps.

#### 2.7.2 Solution Using the Adaptive Search Technique

For the solution of Problem No. 1 using the adaptive search technique we will assume that no a priori information is given. Then from chapter 2.6 we can use either of the two algorithms: ASR-1 or ASR-2.

Before we discuss the results obtained we will investigate if the stated Problem No. 1, belongs to the class of functions,  $F$ , solvable by ASR-1 or ASR-2. Figure 2-12 shows two equiquality contours of the function defined in (2-105). Because the contours are not circular the probability of success depends on the location of the center of the search circle. We will consider the three extreme cases when:

- (a) Moves occur only along the  $x_1$  - axes
- (b) Moves occur only along the  $x_2$  - axes
- (c) Moves occur only perpendicular to the equiquality contours

Assume that the amount of noise,  $\delta_N$ , is much greater than the distance

$$a^{(1)} = \sqrt{x_1^2 + x_2^2}.$$

Then, for assumption, (a), using the angle,  $\alpha_a$ , the probability of success for every ratio,  $\frac{r}{a}$ , is

$$P_S = P_{SD} + P_{SN}$$

$$P_S = 0.1105 + 0.213$$

That is, the noise causes more often move away from the optimum and, clearly,  $E \left[ \frac{\Delta a}{a} \right] < 0$  for all  $r$  or, rather, case (a) does not satisfy the condition of being in the class of functions, F.

Considering case (b) we find with  $\alpha_b$

$$P_S = P_{SD} + P_{SN}$$

$$P_S = 0.265 + 0.162$$

Since  $P_{SD} > P_{SN}$  is will result in  $E \left[ \frac{\Delta a}{a} \right] > 0$  for every  $\frac{r}{a}$ . It follows that case (b), belongs to the class, F.

Finally, for the hypothetical case, (c), we obtain with  $\alpha_c$  and  $0 < \frac{r}{a} < b$ ,

$$b > 0$$

$$P_S = P_{SD} + P_{SN}$$

$$P_S = 0.375 + 0.125$$

for which  $E \left[ \frac{\Delta a}{a} \right] > 0$  with the conclusion that (c) satisfies the condition to be in class F.

That is, only when we would stay on the unfavorable edge at every search step the defined function would not belong to the solvable problems in class F. However, the probability of staying on the edge at each move is zero. We conclude from (a), (b) and (c) that the problem function is in F.

The problem has been solved by using the adaptive search technique, ASR-2, defined in (2-103) with the fixed, arbitrarily chosen parameters,  $A' = 0.1$ ,  $A = 0.2$  and various initial conditions for the vector length,  $r^{(1)}$ . Table 2-8 shows the average number of search steps needed to reach a point in the circle around the origin with radius  $a^{(m)} \leq 10^{-3}$ . The average was taken from 100 runs for each case.

TABLE 2-8

AVERAGE STEP NUMBER FOR SOLUTION OF PROBLEM NO. 1  
USING ADAPTIVE SEARCH TECHNIQUE

$r^{(1)}$	$\bar{M}$ from $X^{(1)} = (8, 0)$	$\bar{M}$ from $X^{(1)} = (8, 1)$
10	616	590
12	511 (best)	477 (best)
14	563	548
16	594	544
18	588	632

The simpler rules in ASR-1 would give slightly better results. With ASR-1 it was found, for example, for  $r^{(1)} = 12$  and  $X^{(1)} = (8, 0)$  that  $\bar{M} = 485$  compared with  $\bar{M} = 511$  obtained with ASR-2. The reason for this is neither of the parameters  $A'$ ,  $A$  has been optimized.

The printout of a single run using ASR-1 is shown in Appendix A-1.

### 2.7.3 Optimization Using Stochastic Approximation

The only well known method suited for the optimization of noisy problem functions is the stochastic approximation. In the following we discuss the Kiefer-Wolfowitz (K-W) rule, which uses stochastic approximation to find the maximum of a function, and some of the practical considerations encountered in programming.

First we define some of the notations which will be used in the discussion of stochastic approximation.

- $x$  a single independent variable
- $y(x)$  the underlying noiseless function whose maximum is to be found
- $\hat{x}$  the location of the maximum of  $y(x)$
- $\delta$  an additive noise term
- $z(x) = y(x) + \delta$ , a noisy measurement of  $y(x)$
- $i$  iteration index or, rather, running step number
- $m$  total step number

- $a_i$  step size at the  $i^{\text{th}}$  step
- $a_0$  initial step size
- $c_i$  span over which finite difference approximation to gradient is measured at  $i^{\text{th}}$  step

The actual problem we are dealing with is two-dimensional so the subscript is used to denote dimension and a bracketed superscript denotes step number. For instance,  $x_2^{(1)}$  is the value of the  $x_2$  component at the  $i^{\text{th}}$  step.

The K-W rule is stated for a function of one variable,  $y(x)$ , as follows: it is assumed that only noisy measurements,  $z(x) = y(x) + \delta$ , are available, and that the noise  $\delta$  has zero mean and finite variance. To find a maximum of  $y(x)$ ,  $x$  is stepped according to

$$x_{i+1} = x_i + a_i \frac{z(x_i + c_i) - z(x_i - c_i)}{c_i}$$

The fraction amounts to a finite difference approximation to the gradient and it will usually be in error at any given step because different values of noise will enter into the two measurements of  $z$ . The convergence conditions consist of a bound on  $y(x)$ , which is nearly always satisfied in practical problems, and some stipulations which limit the area of choice of  $a_i$  and  $c_i$ . The bound on  $y(x)$  is that, for all  $x_1$  and  $x_2$ ,  $x_1 \neq x_2$ , we must have

$$|y(x_2) - y(x_1)| < A|x_2 - \hat{x}| + B < \infty$$

where  $\hat{x}$  is the true location of the maximum of  $y(x)$ , and  $A$  and  $B$  are constants. The stipulations about  $a_i$  and  $c_i$  are

$$\begin{aligned} \lim_{i \rightarrow \infty} a_i &= 0 & \lim_{i \rightarrow \infty} c_i &= 0 \\ \sum_{i=1}^{\infty} a_i &= \infty & \sum_{i=1}^{\infty} (a_i/c_i)^2 &< \infty, \end{aligned}$$

which completes the description of the rule. Note that the conditions imply ultimate convergence in mean square

$$\lim_{m \rightarrow \infty} E \left[ (x_m - \hat{x})^2 \right] = 0$$

and with probability one

$$\text{Probability} \quad \left\{ \lim_{m \rightarrow \infty} x_m = \hat{x} \right\} = 1$$

However, in practice one would like to know more about convergence in a reasonable amount of computer time, although this is strongly problem-dependent. The sequence  $a_i$  is usually chosen to be harmonic,  $a_i = 1/i$ , because this is the fastest diminishing series which satisfies the convergence conditions.

It is perhaps worthwhile to see qualitatively why such a sequence is useful when applied to noisy measurements. Suppose we wish to estimate the value of a constant,  $k$ , given only a series of noisy measurements

$$z_i = k + \delta_i \quad i = 1, 2, \dots, (m-1)$$

where  $\delta_i$  is random with zero mean. Then an estimate is

$$k_{m-1} = z_{m-1} = \frac{1}{m-1} \sum_{i=1}^{m-1} z_i$$

If we are given one more measurement, the new estimate is

$$\begin{aligned} k_m &= z_m = \frac{1}{m} \sum_{i=1}^m z_i \\ &= \frac{1}{m} \left\{ z_m + \sum_{i=1}^{m-1} z_i \right\} = \frac{1}{m} z_m + \frac{m-1}{m} k_{m-1} \end{aligned}$$

that is,  $k_m = \left(\frac{1}{m}\right) z_m + (1 - 1/m) k_{m-1}$

The new measurement is accepted with harmonic weight  $1/m$ .

This leaves a certain available range for  $c_i$ , which is found as follows. It is well known fact that

$$\sum_{j=1}^{\infty} 1/j^p < \infty \text{ for } p > 1.$$

Now suppose  $a_i = i^{-1}$  and  $c_i = i^{-\gamma}$ . What is the permissible range of  $\gamma$ ? We have

$$\left( \frac{a_i}{c_i} \right)^2 = \left( \frac{i^{-1}}{i^{-\gamma}} \right)^2 = 1/i^{2(1-\gamma)}$$

so the condition is  $p = 2(1 - \gamma) > 1$ , that is  $\gamma < 1/2$ . Also,  $\gamma$  must be greater than zero, so

$$x_{i+1} = x_i + \frac{1}{i} \frac{z(x_i + c_i) - z(x_i - c_i)}{c_i}$$

with  $c_i = i^{-\gamma}$  and  $0 < \gamma < 1/2$  as a convergent class of K-W rules. Dupac (10) has shown that for certain general assumptions about  $y(x)$ , the best practical choice to minimize  $E \left[ (x_i - \hat{x})^2 \right]$  is  $\gamma = 1/4$ . There still remains a selection of the initial step size, so the full statement of this rule is

$$x_{i+1} = x_i + \frac{a_0}{i} \frac{z(x_i + c_i) - z(x_i - c_i)}{c_i}$$

$$c_i = \left( \frac{a_0}{i} \right)^{1/4} \quad a_0 \text{ free.} \quad (\text{DK-W Rule})$$

Kesten (11) has proposed an acceleration scheme for the K-W search which should mitigate two well known disadvantages: firstly, that the search moves very slowly in regions of slight gradient, and secondly that high gradient regions may cause repetitive overshooting. The Kesten modification is based on the reasoning that there should be few changes in the sign of the gradient when  $x_i$  is far from  $\hat{x}$ , so the step size should not be reduced much, but that when  $x_i$  is near the maximum, the sign of the gradient changes often, and the step size should shrink quite fast. The statement of the Kesten K-W rule is

$$x_{i+1} = x_i + a_i \frac{z(x_i + c) - z(x_i - c)}{c} \quad (\text{KK-W Rule})$$

The  $a_i$  are still based on the harmonic sequence, but they are only stepped along that sequence when the fraction changes sign. Thus a typical Kesten sequence might be

1 1 1 1 1 1/2 1/2 1/2 1/3 1/3 1/3 1/3 1/4 1/5 . . . . .

Note that  $c = \text{constant, all } i$ , is a condition for convergence of the accelerated rule.

Also, as before, the initial step size,  $a_0$ , is free, so in a computer program there are two parameters,  $a_0$  and  $c$ , to be chosen. Although this is called an acceleration, there is no known proof that it is always faster than the basic rule.

In the pyramid problem,  $y$  is a function of two variables,  $x_1$  and  $x_2$ , so the univariate convergence proofs do not apply directly. Fortunately, there is a general theorem due to Dvoretzky (12) which contains both the K-W and KK-W proofs as special cases, and which applies to multivariate functions. Thus the two methods may be formulated for Problem No. 1 as follows:

Dupac Kiefer-Wolfowitz Rule

$$x_1^{(i+1)} = x_1^{(i)} + a^{(i)} \frac{z(x_1^{(i)} + c^{(i)}, x_2^{(i)}) - z(x_1^{(i)} - c^{(i)}, x_2^{(i)})}{c^{(i)}}$$

$$x_2^{(i+1)} = x_2^{(i)} + a^{(i)} \frac{z(x_1^{(i)}, x_2^{(i)} + c^{(i)}) - z(x_1^{(i)}, x_2^{(i)} - c^{(i)})}{c^{(i)}}$$

$$a^{(i)} = a_0 i^{-1} \qquad c^{(i)} = (a^{(i)})^{1/4}$$

empirical choice of  $a_0$

Kesten Kiefer-Wolfowitz Rule

$$x_j^{(i+1)} = x_j^{(i)} + a_j^{(i)} s_j^{(i)} \qquad j = 1, 2$$

$$s_1^{(i)} = \frac{z(x_1^{(i)} + c_1, x_2^{(i)}) - z(x_1^{(i)} - c_1, x_2^{(i)})}{c_1}$$

$$s_2^{(i)} = \frac{z(x_1^{(i)}, x_2^{(i)} + c_2) - z(x_1^{(i)}, x_2^{(i)} - c_2)}{c_2}$$

$$a_j^{(i)} = a_0 / k_j \qquad k_j = 1, 2, 3, \dots$$

$k_j$  is stepped only when the sign of  $s_j$  changes. Empirical choice of  $a_0$ ,  $c_1$  and  $c_2$ .

To keep the investigation reasonably simple, we have set  $c_1 = c_2 = c$  and explored the performance of the rule for various values of  $a_0$  and  $c$ . The following tables show the average number of function evaluations needed for a search which starts at  $(8, 0)$  and terminates within a radius of 0.008 of  $(0, 0)$ .

TABLE 2-9  
 NUMBER OF FUNCTION EVALUATIONS NEEDED BY THE DUPAC  
 KIEFER-WOLFOWITZ RULE TO SOLVE PROBLEM NO. 1

<u>a<sub>o</sub></u>	<u>Average Number of Function Evaluations</u>
3	3,800
4	1,552
5	1,280 (best)
10	1,628
50	2,456

TABLE 2-10  
 NUMBER OF FUNCTION EVALUATIONS NEEDED BY THE KESTEN  
 KIEFER-WOLFOWITZ RULE TO SOLVE PROBLEM NO. 1

<u>a<sub>o</sub></u> / <u>c</u>	<u>0.8</u>	<u>1</u>	<u>1.2</u>
4	1,372	1,360	1,368
5	1,412	1,140 (best)	1,328
7	1,504	1,352	1,368

Thus, the Kesten acceleration leads to a slightly better result, given an optimum choice of parameters in both methods:

#### 2.7.4 Comparison of Results

The results in Tables 2-7, 2-8 and 2-9 show that the adaptive search technique under the worst condition (i.e., no a priori information is given) converges better than twice as fast to a solution of a predetermined accuracy compared with the solution using stochastic approximation. Clearly, in both methods the limiting factor of the speed of convergence is the harmonic sequence. However, if a little information on the type of the quality function were to be used in order to be able to apply ASR-3, the speed of convergence would be highly increased since the search would converge with a constant rate.

Further, an important fact not discussed so far is that the convergence of the stochastic approximation requires that the noise be unbiased; i.e., its expected value is zero. This condition is satisfied for the considered problem function No. 1. No specific assumption on the noise was made in defining the class of functions,  $F$ , tractable by the adaptive search method. Thus, even if the noise is biased, the adaptive search will converge to the optimal point whenever the problem function is in class  $F$ . From a theoretical aspect of the information one can even expect that the obtainable speed of convergence is faster for the biased case, since the entropy of the noise source is smaller for the unsymmetric case. But the adaptive search is based on the results (or information) gained in the past. Although in the discussed adaptation rules only the information "success" or "failure" is used, the vector length,  $r^{(m)}$  at a certain step,  $m$ , is a function of the whole previous history.

### 3.0 CONCLUSION AND RECOMMENDATION

It has been shown that existing search procedures are greatly improved by a new adaptive non-deterministic search technique. The technique is especially applicable to problems operating in the presence of noise and having ill-defined parameters and/or inputs. Further advantages and features of the procedure are:

- (1) It converges for a large class of functions.
- (2) The speed of convergence is high and does not affect the final accuracy.
- (3) The accuracy is only restricted by instrumentation of the computing device.
- (4) No strong conditions are imposed on the functions, for example, no derivatives are required to exist.
- (5) Superimposed noise does not prevent the search process from converging; it only reduces the speed of convergence.
- (6) No conditions have to be imposed on the unknown characteristic of the noise.
- (7) The search algorithm can be easily implemented as computer program.
- (8) Assuming undisturbed functions it is superior to all known direct search techniques above a certain dimension  $n$ .
- (9) Assuming disturbed functions it is superior to the method of stochastic approximation with respect to speed of convergence and requirements on the characteristic of the noise.

Clearly, a number of investigations and improvements remain to be solved. Some vital problems are stated below:

- (a) Investigations of "noisy" problems assuming other characteristics of noise for the theoretical case, and corresponding simulations with the adaptive search technique
- (b) Studies concerning the size of the class of functions  $F$  for which the search converges under the consideration of various adaptation rules
- (c) Development of adaptation rules for the direction of the search vector for disturbed and undisturbed functions
- (d) Development of adaptive search techniques for multimodal undisturbed and disturbed functions.

#### 4.0 REFERENCES

1. Rastrigin, L.A., "The Convergence of the Random Search Method in the External Control of a Many Parameter System," Automation and Remote Control, Vol. 24, Nr. 11, 1963, pp. 1337-1342.
2. Matyas, T., "Random Optimization," Automation and Remote Control, Vol. 26, Nr. 2, 1965, pp. 244-251.
3. Tsytkin, Ya. Z., "Adaptation, Training and Self-Organization in Automatic Systems," Automation and Remote Control, Vol. 27, Nr. 1 1966, pp. 23-61.
4. Schumer, M.A. and Steiglitz, K., "Adaptive Step Size Random Search," IEEE Trans. AC-13, Nr. 3, June 1968, pp. 270-276.
5. Remmler, K.L. et. al. "Solutions of Systems of Nonlinear Equations," Final Report, Lockheed Missiles and Space Co., Huntsville Research and Engineering Center, October 1966.
6. Fletcher, R., Powell, M.J.D. "A Rapidly Convergent Descent Method for Minimization," Computer J., Vol. 6, 1963, p. 163.
7. Robbins, H. and Monro, S. "A Stochastic Approximation Method," Annals of Math. Stat., 22, 1957, pp. 400-407.
8. Kiefer, T. and Wolfowitz, T. "Stochastic Estimation of the Maximum of a Regression Function," Annals of Math. Stat., 23, 1952, pp. 462-466.
9. Blum, T.R., "Multidimensional Stochastic Approximation Methods," Annals of Math. Stat., 25, 1954, pp. 737-744.
10. Dupac, M. "On the Kiefer - Wolfowitz Approximation Method," Translated by M.D. Friedmann, MIT Lincoln Lab. Report 22G-0008, 1960.
11. Kesten, H. "Accelerated Stochastic Approximation," Annals of Math. Stat., 29, 1958, pp. 41-59.
12. Dvoretzky, A. "On Stochastic Approximation," Proc. 3rd Berkeley Symposium on Math. Stat. and Prob., T. Neyman (ed.), Berkeley: University of California Press, 1956, pp. 39-55.
13. Lavi, A. and Vogl, T.P., "Recent Advances in Optimization Techniques," T. Wiley and Sons Inc., New York, 1966.

#### 4.0 REFERENCES (CONT)

14. Leon, A., "A Comparison Among Eight Known Optimization Procedures," in (13), pp. 23-46.
15. Powell, M.J.D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives," Computer J. Vol. 7, 1964, pp. 155-162.
16. Brooks, S.H., "A Discussion of Random Methods for Seeking Maxima," Operations Research 6, 2, March 1958, pp. 244-251.
17. Box, G.E.P., "Evolutionary Operation: A Method for Increasing Industrial Productivity," Appl. Statist. 6, 1957, pp. 81-101.
18. Wilde, D.F., "Optimum Seeking Methods," Prentice-Hall Inc., Englewood Cliffs, N.J., 1964.
19. Wilde, D.F. and Beightler, C.S., "Foundations of Optimization," Prentice-Hall Inc. Englewood Cliffs, N.J., 1967.
20. Knuth, D.E., "The Art of Computer Programming," Volume 2, "Seminumerical Algorithms", Addison-Wesley Publishing Company Reading, Massachusetts, 1969.
21. Avenhaus, E., "Adaptive Methods for the Optimization of n-dimensional Functions," Diploma Thesis under E. Schmitt, Institute for Information Processing and Transmission, University of Karlsruhe, West Germany, 1966.
22. Schmitt, E., "Adaptive Methods for Optimization," Annual Report 1966, Institute for Information Processing and Transmission, University of Karlsruhe, West Germany, 1966.
23. Rastrigin, L.A., "External Control of Continuous Multiparameter Systems by the Method of Random Search," Engrg. Cybernetics, Vol. 2, January/February 1964, pp. 82-90

APPENDIX

ASR-2 X(1) = 12  
 A = 0.2  
 A' = 0.1  
 X(1) = (8.0)

NO. TRIALS=	1	NO. S. TRIALS=	0	R=	7.68000000	Q=	2.00000000	SQRT(X)=	8.00000000	X(1)=	8.00000000	X(2)=	0.00000000
NO. TRIALS=	2	NO. S. TRIALS=	1	R=	12.00000000	Q=	2.00000000	SQRT(X)=	8.8184610	X(1)=	.36220328	X(2)=	-.80402819
NO. TRIALS=	3	NO. S. TRIALS=	1	R=	7.68000000	Q=	1.80594807	SQRT(X)=	.88184610	X(1)=	.36220328	X(2)=	.80402819
NO. TRIALS=	4	NO. S. TRIALS=	1	R=	6.14400000	Q=	1.80594807	SQRT(X)=	.88184610	X(1)=	.36220328	X(2)=	.80402819
NO. TRIALS=	5	NO. S. TRIALS=	1	R=	4.91520000	Q=	1.40594807	SQRT(X)=	.88184610	X(1)=	.36220328	X(2)=	.80402819
NO. TRIALS=	6	NO. S. TRIALS=	1	R=	3.93216000	Q=	1.80594807	SQRT(X)=	.88184610	X(1)=	.36220328	X(2)=	.80402819
NO. TRIALS=	7	NO. S. TRIALS=	1	R=	3.14572800	Q=	1.80594807	SQRT(X)=	.88184610	X(1)=	.36220328	X(2)=	.80402819
NO. TRIALS=	8	NO. S. TRIALS=	1	R=	2.51658240	Q=	1.40594807	SQRT(X)=	.88184610	X(1)=	.36220328	X(2)=	.80402819
NO. TRIALS=	9	NO. S. TRIALS=	2	R=	9.72000000	Q=	1.40594807	SQRT(X)=	1.64670412	X(1)=	-.96684863	X(2)=	-.33298094
NO. TRIALS=	10	NO. S. TRIALS=	2	R=	4.91520000	Q=	1.47343790	SQRT(X)=	1.64670412	X(1)=	-.96684863	X(2)=	-.33298094
NO. TRIALS=	11	NO. S. TRIALS=	2	R=	3.14572800	Q=	1.47343790	SQRT(X)=	1.64670412	X(1)=	-.96684863	X(2)=	-.33298094
NO. TRIALS=	12	NO. S. TRIALS=	2	R=	2.01326592	Q=	1.47343790	SQRT(X)=	1.64670412	X(1)=	-.96684863	X(2)=	-.33298094
NO. TRIALS=	13	NO. S. TRIALS=	3	R=	8.74800000	Q=	1.07343790	SQRT(X)=	1.31012280	X(1)=	.98969322	X(2)=	-.85844573
NO. TRIALS=	14	NO. S. TRIALS=	3	R=	3.14572800	Q=	1.25868306	SQRT(X)=	1.31012280	X(1)=	.98969322	X(2)=	-.85844573
NO. TRIALS=	15	NO. S. TRIALS=	3	R=	1.61061274	Q=	1.65868306	SQRT(X)=	1.31012280	X(1)=	.98969322	X(2)=	-.85844573
NO. TRIALS=	16	NO. S. TRIALS=	4	R=	7.87320000	Q=	1.25868306	SQRT(X)=	.32355149	X(1)=	-.30992760	X(2)=	.09290022
NO. TRIALS=	17	NO. S. TRIALS=	4	R=	2.01326592	Q=	1.70085439	SQRT(X)=	.32355149	X(1)=	-.30992760	X(2)=	.09290022
NO. TRIALS=	18	NO. S. TRIALS=	4	R=	1.00000000	Q=	1.70085439	SQRT(X)=	.32355149	X(1)=	-.30992760	X(2)=	.09290022
NO. TRIALS=	19	NO. S. TRIALS=	5	R=	7.08588000	Q=	1.70085439	SQRT(X)=	1.20783368	X(1)=	-1.07450938	X(2)=	-.55162650
NO. TRIALS=	20	NO. S. TRIALS=	5	R=	1.28849019	Q=	1.76444752	SQRT(X)=	1.20783368	X(1)=	-1.07450938	X(2)=	-.55162650
NO. TRIALS=	21	NO. S. TRIALS=	5	R=	.80000000	Q=	1.76444752	SQRT(X)=	1.20783368	X(1)=	-1.07450938	X(2)=	-.55162650
NO. TRIALS=	22	NO. S. TRIALS=	6	R=	6.37729200	Q=	1.36444752	SQRT(X)=	1.32210079	X(1)=	-1.30457126	X(2)=	.21457943
NO. TRIALS=	23	NO. S. TRIALS=	6	R=	1.00000000	Q=	1.85325398	SQRT(X)=	1.32210079	X(1)=	-1.30457126	X(2)=	.21457943
NO. TRIALS=	24	NO. S. TRIALS=	6	R=	.66666667	Q=	1.85325398	SQRT(X)=	1.32210079	X(1)=	-1.30457126	X(2)=	.21457943
NO. TRIALS=	25	NO. S. TRIALS=	6	R=	.50000000	Q=	1.85325398	SQRT(X)=	1.32210079	X(1)=	-1.30457126	X(2)=	.21457943
NO. TRIALS=	26	NO. S. TRIALS=	7	R=	5.73956280	Q=	1.45325398	SQRT(X)=	1.76137719	X(1)=	-1.67213656	X(2)=	.55354218
NO. TRIALS=	27	NO. S. TRIALS=	7	R=	.85714286	Q=	1.64415582	SQRT(X)=	1.76137719	X(1)=	-1.67213656	X(2)=	.55354218
NO. TRIALS=	28	NO. S. TRIALS=	8	R=	5.16560652	Q=	1.64415582	SQRT(X)=	2.07417763	X(1)=	-2.06362363	X(2)=	-.20897453
NO. TRIALS=	29	NO. S. TRIALS=	8	R=	.75000000	Q=	1.70368546	SQRT(X)=	2.07417763	X(1)=	-2.06362363	X(2)=	-.20897453
NO. TRIALS=	30	NO. S. TRIALS=	9	R=	4.64904587	Q=	1.30368546	SQRT(X)=	2.70521171	X(1)=	-2.60557642	X(2)=	-.72742143
NO. TRIALS=	31	NO. S. TRIALS=	9	R=	.66666667	Q=	1.38791615	SQRT(X)=	2.70521171	X(1)=	-2.60557642	X(2)=	-.72742143
NO. TRIALS=	32	NO. S. TRIALS=	10	R=	4.18414128	Q=	.98791615	SQRT(X)=	2.95484479	X(1)=	-2.60530733	X(2)=	-1.39408804
NO. TRIALS=	33	NO. S. TRIALS=	10	R=	.60000000	Q=	1.12130332	SQRT(X)=	2.95484479	X(1)=	-2.60530733	X(2)=	-1.39408804
NO. TRIALS=	34	NO. S. TRIALS=	10	R=	.40000000	Q=	1.12130332	SQRT(X)=	2.95484479	X(1)=	-2.60530733	X(2)=	-1.39408804
NO. TRIALS=	35	NO. S. TRIALS=	11	R=	3.76572715	Q=	.72130332	SQRT(X)=	2.56949532	X(1)=	-2.21664212	X(2)=	-1.29953989
NO. TRIALS=	36	NO. S. TRIALS=	11	R=	.54545455	Q=	.83685562	SQRT(X)=	2.56949532	X(1)=	-2.21664212	X(2)=	-1.29953989
NO. TRIALS=	37	NO. S. TRIALS=	12	R=	3.38915444	Q=	.83685562	SQRT(X)=	3.10160332	X(1)=	-2.73988686	X(2)=	-1.45360350
NO. TRIALS=	38	NO. S. TRIALS=	12	R=	.50000000	Q=	1.07058123	SQRT(X)=	3.10160332	X(1)=	-2.73988686	X(2)=	-1.45360350
NO. TRIALS=	39	NO. S. TRIALS=	13	R=	3.05023899	Q=	.67058123	SQRT(X)=	3.55802153	X(1)=	-3.03470709	X(2)=	-1.85743643
NO. TRIALS=	40	NO. S. TRIALS=	13	R=	.46153846	Q=	.85008401	SQRT(X)=	3.55802153	X(1)=	-3.03470709	X(2)=	-1.85743643

NO. TRIALS=	41	NO. S. TRIALS=	13	R=	.30764231	0=	.85008401	SORT(X)=	3.55802153	X(1)=	-3.03470709	X(2)=	-1.85743643
NO. TRIALS=	42	NO. S. TRIALS=	14	R=	2.74521509	0=	.45008401	SORT(X)=	3.52257570	X(1)=	-3.15189179	X(2)=	-1.57293282
NO. TRIALS=	43	NO. S. TRIALS=	15	R=	2.47069359	0=	.94044851	SORT(X)=	3.21792318	X(1)=	-2.99847040	X(2)=	1.16799180
NO. TRIALS=	44	NO. S. TRIALS=	16	R=	2.22362423	0=	1.13310920	SORT(X)=	1.41679469	X(1)=	-1.28006860	X(2)=	-6.0723271
NO. TRIALS=	45	NO. S. TRIALS=	16	R=	.37500000	0=	1.70109320	SORT(X)=	1.41679469	X(1)=	-1.28006860	X(2)=	-6.0723271
NO. TRIALS=	46	NO. S. TRIALS=	17	R=	2.00126180	0=	1.30193320	SORT(X)=	1.15057969	X(1)=	-1.11876511	X(2)=	-2.6869736
NO. TRIALS=	47	NO. S. TRIALS=	17	R=	.35294118	0=	1.86876803	SORT(X)=	1.15057969	X(1)=	-1.11876511	X(2)=	-2.6869736
NO. TRIALS=	48	NO. S. TRIALS=	17	R=	.23529412	0=	1.46876803	SORT(X)=	1.15057969	X(1)=	-1.11876511	X(2)=	-2.6869736
NO. TRIALS=	49	NO. S. TRIALS=	18	R=	1.80113562	0=	1.46876803	SORT(X)=	.96085855	X(1)=	-.95574160	X(2)=	-.09903105
NO. TRIALS=	50	NO. S. TRIALS=	18	R=	.33333333	0=	1.56923926	SORT(X)=	.96085855	X(1)=	-.95574160	X(2)=	-.09903105
NO. TRIALS=	51	NO. S. TRIALS=	18	R=	.22222222	0=	1.96923926	SORT(X)=	.96085855	X(1)=	-.95574160	X(2)=	-.09903105
NO. TRIALS=	52	NO. S. TRIALS=	18	R=	.16666667	0=	1.96923926	SORT(X)=	.96085855	X(1)=	-.95574160	X(2)=	-.09903105
NO. TRIALS=	53	NO. S. TRIALS=	18	R=	.13333333	0=	1.96923926	SORT(X)=	.96085855	X(1)=	-.95574160	X(2)=	-.09903105
NO. TRIALS=	54	NO. S. TRIALS=	19	R=	1.62102206	0=	1.56923926	SORT(X)=	1.05332300	X(1)=	-1.05332300	X(2)=	-.00813671
NO. TRIALS=	55	NO. S. TRIALS=	19	R=	.31578947	0=	1.58608700	SORT(X)=	1.05332300	X(1)=	-1.05332300	X(2)=	-.00813671
NO. TRIALS=	56	NO. S. TRIALS=	20	R=	1.45891986	0=	1.58608700	SORT(X)=	1.19799536	X(1)=	-1.16285303	X(2)=	.28803769
NO. TRIALS=	57	NO. S. TRIALS=	20	R=	.30000000	0=	1.85221432	SORT(X)=	1.19799536	X(1)=	-1.16285303	X(2)=	.28803769
NO. TRIALS=	58	NO. S. TRIALS=	21	R=	1.31302787	0=	1.45221432	SORT(X)=	.93743936	X(1)=	-.86943517	X(2)=	.35053537
NO. TRIALS=	59	NO. S. TRIALS=	21	R=	.28571429	0=	1.48589882	SORT(X)=	.93743936	X(1)=	-.86943517	X(2)=	.35053537
NO. TRIALS=	60	NO. S. TRIALS=	22	R=	1.18172508	0=	1.48589882	SORT(X)=	.93223221	X(1)=	-.93223221	X(2)=	.07278137
NO. TRIALS=	61	NO. S. TRIALS=	23	R=	1.06355257	0=	1.98360584	SORT(X)=	.48765676	X(1)=	.11073027	X(2)=	-.47491886
NO. TRIALS=	62	NO. S. TRIALS=	23	R=	.26086957	0=	1.98360584	SORT(X)=	.48765676	X(1)=	.11073027	X(2)=	-.47491886
NO. TRIALS=	63	NO. S. TRIALS=	24	R=	.95719732	0=	1.58788640	SORT(X)=	.36264296	X(1)=	-.11488788	X(2)=	-.34396321
NO. TRIALS=	64	NO. S. TRIALS=	24	R=	.25000000	0=	1.58788640	SORT(X)=	.36264296	X(1)=	-.11488788	X(2)=	-.34396321
NO. TRIALS=	65	NO. S. TRIALS=	25	R=	.86147759	0=	1.63943714	SORT(X)=	.43237372	X(1)=	-.12513974	X(2)=	-.41386844
NO. TRIALS=	66	NO. S. TRIALS=	26	R=	.77532983	0=	2.00942468	SORT(X)=	.67064220	X(1)=	-.66652952	X(2)=	-.07415766
NO. TRIALS=	67	NO. S. TRIALS=	26	R=	.23076923	0=	2.03703103	SORT(X)=	.67064220	X(1)=	-.66652952	X(2)=	-.07415766
NO. TRIALS=	68	NO. S. TRIALS=	27	R=	.69779684	0=	2.03703103	SORT(X)=	.51705643	X(1)=	-.50844706	X(2)=	.09396246
NO. TRIALS=	69	NO. S. TRIALS=	27	R=	.22222222	0=	2.06072560	SORT(X)=	.51705643	X(1)=	-.50844706	X(2)=	.09396246
NO. TRIALS=	70	NO. S. TRIALS=	27	R=	.14814815	0=	1.66072560	SORT(X)=	.51705643	X(1)=	-.50844706	X(2)=	.09396246
NO. TRIALS=	71	NO. S. TRIALS=	28	R=	.62801716	0=	1.66072560	SORT(X)=	.58820822	X(1)=	-.58736877	X(2)=	-.03141392
NO. TRIALS=	72	NO. S. TRIALS=	28	R=	.21428571	0=	1.66996068	SORT(X)=	.58820822	X(1)=	-.58736877	X(2)=	-.03141392
NO. TRIALS=	73	NO. S. TRIALS=	28	R=	.14285714	0=	1.66996068	SORT(X)=	.58820822	X(1)=	-.58736877	X(2)=	-.03141392
NO. TRIALS=	74	NO. S. TRIALS=	28	R=	.10714286	0=	1.66996068	SORT(X)=	.58820822	X(1)=	-.58736877	X(2)=	-.03141392
NO. TRIALS=	75	NO. S. TRIALS=	29	R=	.56521544	0=	1.66996068	SORT(X)=	.48466637	X(1)=	-.48466637	X(2)=	-.00089115
NO. TRIALS=	76	NO. S. TRIALS=	29	R=	.20689655	0=	1.70271043	SORT(X)=	.48466637	X(1)=	-.48466637	X(2)=	-.00089115
NO. TRIALS=	77	NO. S. TRIALS=	30	R=	.50863390	0=	1.70271043	SORT(X)=	.36364252	X(1)=	-.33485352	X(2)=	.14188622
NO. TRIALS=	78	NO. S. TRIALS=	30	R=	.20000000	0=	2.07630681	SORT(X)=	.36364252	X(1)=	-.33485352	X(2)=	.14188622
NO. TRIALS=	79	NO. S. TRIALS=	31	R=	.45782451	0=	1.67630681	SORT(X)=	.38889486	X(1)=	-.23109390	X(2)=	.31278558
NO. TRIALS=	80	NO. S. TRIALS=	31	R=	.19354839	0=	2.02886699	SORT(X)=	.38889486	X(1)=	-.23109390	X(2)=	.31278558
NO. TRIALS=	81	NO. S. TRIALS=	32	R=	.41204206	0=	1.62866699	SORT(X)=	.40239646	X(1)=	-.36382328	X(2)=	.17191723
NO. TRIALS=	82	NO. S. TRIALS=	33	R=	.37083785	0=	1.65846845	SORT(X)=	.76446936	X(1)=	-.76171330	X(2)=	.06485563
NO. TRIALS=	83	NO. S. TRIALS=	33	R=	.18181818	0=	2.02171509	SORT(X)=	.76446936	X(1)=	-.76171330	X(2)=	.06485563
NO. TRIALS=	84	NO. S. TRIALS=	33	R=	.12121212	0=	2.02171509	SORT(X)=	.76446936	X(1)=	-.76171330	X(2)=	.06485563
NO. TRIALS=	85	NO. S. TRIALS=	33	R=	.09090909	0=	2.02171509	SORT(X)=	.76446936	X(1)=	-.76171330	X(2)=	.06485563
NO. TRIALS=	86	NO. S. TRIALS=	33	R=	.07272727	0=	2.02171509	SORT(X)=	.74415196	X(1)=	-.73245239	X(2)=	.13143682
NO. TRIALS=	87	NO. S. TRIALS=	34	R=	.35294118	0=	1.62171509	SORT(X)=	.74415196	X(1)=	-.73245239	X(2)=	.13143682
NO. TRIALS=	88	NO. S. TRIALS=	34	R=	.1764059	0=	2.00093479	SORT(X)=	.90079241	X(1)=	-.89803616	X(2)=	.07041320
NO. TRIALS=	89	NO. S. TRIALS=	35	R=	.34285714	0=	1.60093479	SORT(X)=	.56338295	X(1)=	-.55584857	X(2)=	.09182980
NO. TRIALS=	90	NO. S. TRIALS=	36	R=	.33333333	0=	1.99222749	SORT(X)=	.56338295	X(1)=	-.55584857	X(2)=	.09182980
NO. TRIALS=	91	NO. S. TRIALS=	36	R=	.16666667	0=	2.95209837	SORT(X)=	.56338295	X(1)=	-.55584857	X(2)=	.09182980
NO. TRIALS=	92	NO. S. TRIALS=	36	R=	.11111111	0=	1.65209837	SORT(X)=	.45229610	X(1)=	-.44658176	X(2)=	.07166933
NO. TRIALS=	93	NO. S. TRIALS=	37	R=	.32432432	0=	1.65209837	SORT(X)=	.45229610	X(1)=	-.44658176	X(2)=	.07166933
NO. TRIALS=	94	NO. S. TRIALS=	37	R=	.16216216	0=	2.08201591	SORT(X)=	.45229610	X(1)=	-.44658176	X(2)=	.07166933
NO. TRIALS=	95	NO. S. TRIALS=	38	R=	.31578947	0=	1.68201591	SORT(X)=	.49536625	X(1)=	-.43686024	X(2)=	.23353983
NO. TRIALS=	96	NO. S. TRIALS=	39	R=	.30769231	0=	2.01921202	SORT(X)=	.62111281	X(1)=	-.62067801	X(2)=	-.02323637
NO. TRIALS=	97	NO. S. TRIALS=	39	R=	.15384615	0=	2.06656985	SORT(X)=	.62111281	X(1)=	-.62067801	X(2)=	-.02323637
NO. TRIALS=	98	NO. S. TRIALS=	39	R=	.10256410	0=	1.66656985	SORT(X)=	.62111281	X(1)=	-.62067801	X(2)=	-.02323637
NO. TRIALS=	99	NO. S. TRIALS=	39	R=	.07692308	0=	1.66656985	SORT(X)=	.62111281	X(1)=	-.62067801	X(2)=	-.02323637
NO. TRIALS=	100	NO. S. TRIALS=	39	R=	.06153846	0=	1.66656985	SORT(X)=	.62111281	X(1)=	-.62067801	X(2)=	-.02323637
NO. TRIALS=	101	NO. S. TRIALS=	39	R=	.05128205	0=	1.66656985	SORT(X)=	.62111281	X(1)=	-.62067801	X(2)=	-.02323637

NO. TRIALS= 102	NO. S. TRIALS=	40	R=	30000000	0=	1.66656985	SOFT(X)=	58527211	X(1)=	-.58244882	X(2)=	-.05741791
NO. TRIALS= 103	NO. S. TRIALS=	41	R=	15000000	0=	2.06054307	SOFT(X)=	58527211	X(1)=	-.58244882	X(2)=	-.05741791
NO. TRIALS= 104	NO. S. TRIALS=	41	R=	29268293	0=	1.66054307	SOFT(X)=	44715589	X(1)=	-.44709735	X(2)=	-.00723485
NO. TRIALS= 105	NO. S. TRIALS=	41	R=	14634146	0=	2.10768659	SOFT(X)=	44715589	X(1)=	-.44709735	X(2)=	-.00723485
NO. TRIALS= 106	NO. S. TRIALS=	42	R=	28571429	0=	1.70768659	SOFT(X)=	55686906	X(1)=	-.54456996	X(2)=	-.11639031
NO. TRIALS= 107	NO. S. TRIALS=	43	R=	27906977	0=	2.04521662	SOFT(X)=	48646585	X(1)=	-.46054087	X(2)=	-.15668800
NO. TRIALS= 108	NO. S. TRIALS=	43	R=	13953488	0=	2.04521662	SOFT(X)=	48646585	X(1)=	-.46054087	X(2)=	-.15668800
NO. TRIALS= 109	NO. S. TRIALS=	43	R=	09302326	0=	2.04521662	SOFT(X)=	39912356	X(1)=	-.36752344	X(2)=	-.15664749
NO. TRIALS= 110	NO. S. TRIALS=	44	R=	27272727	0=	1.64521662	SOFT(X)=	39912356	X(1)=	-.36752344	X(2)=	-.15664749
NO. TRIALS= 111	NO. S. TRIALS=	44	R=	13636364	0=	2.06423632	SOFT(X)=	46471549	X(1)=	-.46125519	X(2)=	-.05660507
NO. TRIALS= 112	NO. S. TRIALS=	45	R=	26666667	0=	2.06423632	SOFT(X)=	46471549	X(1)=	-.46125519	X(2)=	-.05660507
NO. TRIALS= 113	NO. S. TRIALS=	45	R=	13333333	0=	2.08510693	SOFT(X)=	46471549	X(1)=	-.46125519	X(2)=	-.05660507
NO. TRIALS= 114	NO. S. TRIALS=	45	R=	08888889	0=	2.08510693	SOFT(X)=	46471549	X(1)=	-.46125519	X(2)=	-.05660507
NO. TRIALS= 115	NO. S. TRIALS=	45	R=	06666667	0=	2.08510693	SOFT(X)=	46471549	X(1)=	-.46125519	X(2)=	-.05660507
NO. TRIALS= 116	NO. S. TRIALS=	46	R=	26086957	0=	2.08510693	SOFT(X)=	40163672	X(1)=	-.39570655	X(2)=	-.06876321
NO. TRIALS= 117	NO. S. TRIALS=	46	R=	13043478	0=	2.09335340	SOFT(X)=	40163672	X(1)=	-.39570655	X(2)=	-.06876321
NO. TRIALS= 118	NO. S. TRIALS=	46	R=	08695652	0=	1.69335340	SOFT(X)=	40163672	X(1)=	-.39570655	X(2)=	-.06876321
NO. TRIALS= 119	NO. S. TRIALS=	47	R=	25531915	0=	1.69335340	SOFT(X)=	36997573	X(1)=	-.34323389	X(2)=	-.13810336
NO. TRIALS= 120	NO. S. TRIALS=	47	R=	12765957	0=	2.07611188	SOFT(X)=	36997573	X(1)=	-.34323389	X(2)=	-.13810336
NO. TRIALS= 121	NO. S. TRIALS=	48	R=	25000000	0=	1.67611188	SOFT(X)=	48891269	X(1)=	-.47074878	X(2)=	-.13202726
NO. TRIALS= 122	NO. S. TRIALS=	48	R=	12500000	0=	2.05303934	SOFT(X)=	48891269	X(1)=	-.47074878	X(2)=	-.13202726
NO. TRIALS= 123	NO. S. TRIALS=	48	R=	08333333	0=	1.65303934	SOFT(X)=	4891269	X(1)=	-.47074878	X(2)=	-.13202726
NO. TRIALS= 124	NO. S. TRIALS=	49	R=	24489796	0=	1.65303934	SOFT(X)=	44476381	X(1)=	-.44147280	X(2)=	-.05400572
NO. TRIALS= 125	NO. S. TRIALS=	49	R=	12244898	0=	2.09010315	SOFT(X)=	44476381	X(1)=	-.44147280	X(2)=	-.05400572
NO. TRIALS= 126	NO. S. TRIALS=	50	R=	24000000	0=	1.69010315	SOFT(X)=	34488847	X(1)=	-.344428018	X(2)=	-.02047480
NO. TRIALS= 127	NO. S. TRIALS=	51	R=	23529412	0=	2.12295404	SOFT(X)=	13783840	X(1)=	-.11190278	X(2)=	-.08048101
NO. TRIALS= 128	NO. S. TRIALS=	51	R=	11764706	0=	2.14542704	SOFT(X)=	13783840	X(1)=	-.11190278	X(2)=	-.08048101
NO. TRIALS= 129	NO. S. TRIALS=	52	R=	23076923	0=	1.74542704	SOFT(X)=	03003595	X(1)=	-.02979743	X(2)=	-.00377778
NO. TRIALS= 130	NO. S. TRIALS=	53	R=	22641509	0=	1.79252940	SOFT(X)=	24646719	X(1)=	-.16415583	X(2)=	-.16384488
NO. TRIALS= 131	NO. S. TRIALS=	53	R=	11320755	0=	2.09363088	SOFT(X)=	24646719	X(1)=	-.16415583	X(2)=	-.16384488
NO. TRIALS= 132	NO. S. TRIALS=	54	R=	22222222	0=	1.69363088	SOFT(X)=	16770337	X(1)=	-.15178453	X(2)=	-.07131533
NO. TRIALS= 133	NO. S. TRIALS=	55	R=	21818182	0=	1.74111696	SOFT(X)=	29548405	X(1)=	-.08456098	X(2)=	-.28312588
NO. TRIALS= 134	NO. S. TRIALS=	55	R=	11764706	0=	2.06983745	SOFT(X)=	19852051	X(1)=	-.17876665	X(2)=	-.08633005
NO. TRIALS= 135	NO. S. TRIALS=	56	R=	10714286	0=	2.12971465	SOFT(X)=	19852051	X(1)=	-.17876665	X(2)=	-.08633005
NO. TRIALS= 136	NO. S. TRIALS=	56	R=	07142857	0=	2.12971465	SOFT(X)=	19852051	X(1)=	-.17876665	X(2)=	-.08633005
NO. TRIALS= 137	NO. S. TRIALS=	56	R=	05357143	0=	2.12971465	SOFT(X)=	19852051	X(1)=	-.17876665	X(2)=	-.08633005
NO. TRIALS= 138	NO. S. TRIALS=	57	R=	21052632	0=	1.72971465	SOFT(X)=	14722216	X(1)=	-.13780815	X(2)=	-.05180037
NO. TRIALS= 139	NO. S. TRIALS=	57	R=	10526316	0=	2.15171822	SOFT(X)=	14722216	X(1)=	-.13780815	X(2)=	-.05180037
NO. TRIALS= 140	NO. S. TRIALS=	57	R=	07017544	0=	1.75171822	SOFT(X)=	14722216	X(1)=	-.13780815	X(2)=	-.05180037
NO. TRIALS= 141	NO. S. TRIALS=	58	R=	20689655	0=	1.75171822	SOFT(X)=	08049404	X(1)=	-.07947035	X(2)=	-.01279661
NO. TRIALS= 142	NO. S. TRIALS=	58	R=	10344828	0=	1.77898729	SOFT(X)=	08049404	X(1)=	-.07947035	X(2)=	-.01279661
NO. TRIALS= 143	NO. S. TRIALS=	58	R=	06896552	0=	1.77898729	SOFT(X)=	08049404	X(1)=	-.07947035	X(2)=	-.01279661
NO. TRIALS= 144	NO. S. TRIALS=	59	R=	20338983	0=	1.77898729	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 145	NO. S. TRIALS=	59	R=	10169492	0=	2.18896590	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 146	NO. S. TRIALS=	59	R=	06779661	0=	2.18896590	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 147	NO. S. TRIALS=	59	R=	05084746	0=	2.18896590	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 148	NO. S. TRIALS=	59	R=	04067797	0=	2.18896590	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 149	NO. S. TRIALS=	59	R=	03389831	0=	2.18896590	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 150	NO. S. TRIALS=	59	R=	02905569	0=	2.18896590	SOFT(X)=	02467321	X(1)=	-.01112477	X(2)=	-.02202287
NO. TRIALS= 151	NO. S. TRIALS=	60	R=	20000000	0=	2.18896590	SOFT(X)=	00772775	X(1)=	-.00451561	X(2)=	-.00627116

NO. TRIALS	NO. S. TRIALS	R	SORT(X)
151	60	.200000000000000	.00772775
632	262	.04580152671756	.00776629
211	77	.15584415584416	.00272247
780	337	.03560830860534	.00749595
425	152	.07894734842105	.00760194
575	234	.05128205128205	.00633784
399	164	.07317073170732	.00391067
245	96	.125000000000000	.00411181
1786	758	.01583113456464	.00631063
410	161	.07453416149068	.00345643
603	239	.05020920502092	.00506702
440	189	.06349206349206	.00299138
525	207	.05797101449275	.00790072
369	146	.08219178082192	.00313363
764	313	.03833865814696	.00512655
341	127	.09448818897638	.00767627
527	212	.05660377358491	.00436385
460	168	.07142857142857	.00750085
530	214	.05607476635514	.00750020
698	289	.04152249134948	.00745453
687	261	.04597701149425	.00474418
235	90	.133333333333333	.00747651
250	86	.13953488372093	.00297464
451	181	.06629834254144	.00703239
517	205	.05853658536585	.00232722
640	275	.04363636363636	.00498837
351	143	.08391608391608	.00446382
589	223	.05381165919283	.00564704
354	135	.088888888888889	.00760111
341	148	.08108108108108	.00440387
621	256	.046875000000000	.00431742
726	296	.04054054054054	.00686534
494	197	.06091370558376	.00656558
679	272	.04411764705882	.00722349
518	201	.05970149253731	.00450263
819	337	.03560830860534	.00689824
562	233	.05150214592275	.00497682
331	126	.09523809523809	.00533745
734	291	.04123711340206	.00233088
536	209	.05741626794258	.00784408
789	332	.03614457831325	.00781867
464	178	.06741573033708	.00505206
588	226	.05309734513274	.00327099
362	144	.083333333333333	.00789466
623	234	.05128205128205	.00767193
294	117	.10256410256410	.00716033
341	140	.08571428571429	.00186170
518	196	.06122448979592	.00174291
461	171	.07017543859649	.00348155
397	167	.07185628742515	.00712129
863	359	.03342618384401	.00611104
355	136	.08823529411765	.00636933
654	279	.04301075268817	.00590083
433	172	.06976744186046	.00654262
909	366	.03278688524590	.00438354
397	148	.08108108108108	.00126318
592	249	.04819277108434	.00443198
425	166	.07228915662651	.00672921
441	176	.06818181818182	.00438309
396	154	.67792207792208	.00784640

466	185	.06486486486486	.00763900
642	274	.04379562043796	.00438190
308	122	.09836065573770	.00210322
159	45	.26666666666667	.00678168
241	94	.12765957446809	.00429932
274	103	.11650485436893	.00190792
217	79	.15189873417722	.00620806
336	125	.09600000000000	.00456259
612	247	.04858209595142	.00040683
1412	621	.01932367149758	.00714633
438	162	.07407407407407	.00489252
414	159	.07547169811321	.00753369
252	94	.12765957446809	.00740840
325	128	.09375000000000	.00757333
440	167	.07185628742515	.00573656
430	174	.06896551724138	.00206262
381	154	.07792207792208	.00245130
925	352	.03409090909091	.00511222
551	205	.05853658536585	.00336453
456	176	.06818181818182	.00354296
248	94	.12765957446809	.00449101
758	297	.04040404040404	.00352301
1008	424	.02830188679245	.00435706
391	148	.08108108108108	.00705001
465	189	.06349206349206	.00625828
258	98	.12244897959184	.00553724
882	377	.03183023872679	.00458381
711	296	.04054054054054	.00204200
752	292	.04109589041096	.00765943
550	211	.05687203791469	.00608629
403	143	.08391608391608	.00680047
192	76	.15789473684210	.00736862
134	49	.24489795918367	.00609813
330	132	.09090909090909	.00449721
270	95	.12631578947368	.00570007
495	206	.05825242718447	.00673206
837	348	.03448275862069	.00570013
946	392	.03061224489796	.00603292
284	109	.11009174311927	.00740068
357	135	.08888888888889	.00251151

AVERAGE NUMBER OF TRIALS= 511.860

```

PROGRAM PRONAME(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
000003 DIMENSION X(10),V(10),VV(10),XMIN(10),KKZ(105),D(10)
000003 N=2
000004 IX=33333
000005 IZ=33333
000006 IW=33333
000007 IXX=0
000010 UOLD=1.0
000011 SOLD=1.0
000012 ROLD=12.0
000014 RINT=ROLD
000015 X(1)=8.0
000017 DO 10 I=2,N
000021 10 X(I)=0.0
000025 KZ=1
000026 8 FKZ=KZ
000030 CALL RANDU(IZ,IY,YEL)
000032 IZ=IY
000033 IF(YEL-0.5) 2,2,3
000036 2 QNEW=+2.0-0.2*ABS(X(1))-0.4*ABS(X(2))-0.2
000047 GO TO 4
000050 3 QNEW=+2.0-0.2*ABS(X(1))-0.4*ABS(X(2))+0.2
000061 4 IF(KZ.GT.1) GO TO 24
000065 QRMS=QNEW
000066 DO 28 I=1,N
000067 28 XMIN(I)=X(I)
000073 24 DO 104 I=1,N
000075 CALL RANDU(IX,IY,YFL)
000077 IX=IY
000100 104 VV(I)=2.0*YFL-1.0
000106 CALL SNORM(V,VV,N)
000111 IF(KZ.EQ.1) GO TO 43
000111 CALL RANDU(IW,IY,YEL)
000116 IW=IY
000117 IF(YEL-0.5) 41,41,42
000122 41 QRMS=2.0-0.2*ABS(XMIN(1))-0.4*ABS(XMIN(2))-0.2
000133 GO TO 43
000134 42 QRMS=2.0-0.2*ABS(XMIN(1))-0.4*ABS(XMIN(2))+0.2
000145 43 IF(QNEW-QRMS) 103,103,102
000150 102 IXX=IXX+1
000152 FIXX=IXX
000153 DO 1 I=1,N
000155 1 XMIN(I)=X(I)
000161 RUPP=RINT*(1.0-0.10)**FIXX
000166 RLOW=RINT/SOLD
000170 IF(RUPP.LT.RLOW) GO TO 44
000173 RHOND=RUPP
000174 GO TO 45
000174 44 RBOND=RLOW
000176 45 ROLD=RINT/SOLD
000200 IF(ROLD.GE.RBOND) GO TO 46
000203 ROLD=RBOND
000203 46 SOLD=SOLD+1.0
000205 UOLD=1.0
000206 GO TO 108
000206 103 UOLD=UOLD+1.0
000210 IF(IXX.EQ.0) GO TO 31

```

```

000212      UOLD=1.0
000212      XRMS=0.0
000214      DO 11 I=1,N
000215      11 XRMS=XRMS+XMIN(I)**2
000222      XRMS=SQRT(XRMS)
000223      KKZ(JJ)=KZ
000226      IF(XRMS=0.0001) 13,13,108
000231      103 UOLD=UOLD+1.0
000233      IF(IXX.F0.0) GO TO 31
000234      F1=(SOLD-1.0)*UOLD
000236      GO TO 32
000237      31 F1=SOLD*UOLD
000241      32 RUPP=RINT*(1.0-0.2)**F1
000246      RLOW=RINT/F1
000250      IF(RUPP.LT.RLOW) GO TO 47
000253      RROND=RUPP
000254      GO TO 48
000254      47 RROND=RLOW
000256      48 ROLD=RINT/F1
000260      IF(ROLD.GE.RROND) GO TO 108
000263      ROLD=RROND
000263      108 DO 107 I=1,N
000265      107 X(I)=XMIN(I)+ROLD*V(I)
000272      KZ=KZ+1
000274      GO TO 8
000274      13 WRITE(6,301) KZ,IXX,ROLD,XRMS
000310      301 FORMAT(4X,I4,9X,I4,9X,F17,14,2X,F11.8)
000310      JJ=JJ+1
000312      IF(JJ.LF.100) GO TO 14
000314      KSUM=0
000315      DO 15 I=1,100
000317      15 KSUM=KSUM+KKZ(I)
000323      FSUM=KSUM
000324      FSUM=FSUM/100.0
000326      WRITE(6,303) FSUM
000333      303 FORMAT(///5X,25HAVERAGE NUMBER OF TRIALS=,F8.3)
000333      CALL EXIT
000334      FND

```

```

SUBROUTINE SNORM(U,R,N2)
000006      DIMENSION U(10),R(10)
000006      VR=0.0
000007      DO 1 I=1,N2
000010      1 VR=VR+R(I)**2
000014      VR=SQRT(VR)
000016      DO 2 I=1,N2
000021      2 U(I)=R(I)/VR
000026      RETURN
000026      END

```

SUBROUTINE RANDU

USAGE

CALL RANDU (IX, IY, YFL)  
IX = IY

Produces uniformly distributed random numbers between 0. and 1. 0 in location YFL.

LISTING

```
SUBROUTINE RANDU (IX, IY, YFL)
  IY = IX * 65539
  IF (IY) 5, 6, 6
5  IY = IY + 2147483647
6  YFL = IY
  YFL = YFL * .4656613E-9
  RETURN
  END
```

WARNING

THIS SUBROUTINE IS SPECIFIC TO THE IBM SYSTEM/360 COMPUTER. IT  
MAY NOT BE SATISFACTORY ON OTHER MACHINES.