Battelle

Pacific Northwest Laboratories
Richland, Washington 99352

AEC Research and Development Report

A TANDEM METHOD FOR FINDING
UNCONSTRAINED MINIMA IN
NONLINEAR LEAST-SQUARES
PROBLEMS: PROGRAM LEARN-1-C

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

PACIFIC NORTHWEST LABORATORY

operated by

BATTELLE

for the

U.S. ATOMIC ENERGY COMMISSION

Under Contract AT(45-1)-1830

3 3679 00062 0874

BNWL-1590

~~Special Distribution~~

A TANDEM METHOD FOR FINDING UNCONSTRAINED
MINIMA IN NONLINEAR LEAST-SQUARES PROBLEMS:
PROGRAM LEARN-I-C

By

D. A. Kottwitz

Reactor Physics Department

June 1971

BATTELLE
PACIFIC NORTHWEST LABORATORIES
RICHLAND, WASHINGTON 99352

A TANDEM METHOD FOR FINDING UNCONSTRAINED
MINIMA IN NONLINEAR LEAST-SQUARES PROBLEMS:
PROGRAM LEARN-I-C

D. A. Kottwitz

ABSTRACT

A new iterative method has been developed for finding minima in unconstrained, nonlinear least-squares problems with a moderate number of parameters (≤ 100). Based on the complete second-order Taylor expansion of the sum of squares, the scheme selectively uses two types of steps through the iteration. One type, found by Newton's method, has rapid terminal convergence. The other, based on a modified gradient method, efficiently avoids saddle points. The combination, together with careful attention to the control of step-size and to convergence tests, provides a high degree of reliability and stability. The algorithm has been adapted for use in the powerful least-squares program LEARN (written in FORTRAN V for the UNIVAC 1108). Information on programming and input-output details is included.

CONTENTS

	ABSTRACT	ii
I.	INTRODUCTION	1
II.	BACKGROUND	2
III.	GEOMETRY AND ANALYSIS OF NONLINEAR LEAST SQUARES	4
	1. Linear Case	4
	2. Intuitive Geometry of the Nonlinear Case	5
	3. Analysis and Algebra of the Nonlinear Case	6
	4. Modified Gradient Step	10
	5. Step Refinement	11
	6. Numerical Aspects	15
IV.	CONTROL OF ITERATIVE STEPS	15
V.	CONVERGENCE TESTS	16
VI.	ITERATIVE STRATEGY	19
	1. Initial Phase	19
	2. Terminal Phase	19
	3. Contingencies	20
VII.	IMPLEMENTATION OF ALGORITHM IN LEARN-I-C	20
	1. Subroutine MINMAX	20
	2. Other Modified Subroutines	21
	3. New Subprograms	23
VIII.	PROGRAM DETAILS	24
IX.	PRACTICAL REMARKS FOR USERS	27
	1. Need for Care	27
	2. Key Input	28
	3. Key Output	28
X.	MISCELLANEOUS TOPICS	29
	ACKNOWLEDGMENTS	30
	REFERENCES	31

A TANDEM METHOD FOR FINDING UNCONSTRAINED
MINIMA IN NONLINEAR LEAST-SQUARES PROBLEMS:
PROGRAM LEARN-I-C

D. A. Kottwitz

I. INTRODUCTION

The purpose of this report is twofold: (1) to present the basis of a new method for finding minima in nonlinear least-squares problems that arise in the fitting of experimental data in science and engineering, and (2) to describe the computer implementation of the method in sufficient detail to permit effective use.

In recent years the fitting of data by nonlinear least-squares methods has become an increasingly important method of experimental analysis. Sometimes a fairly large number of parameters (5-100) must be adjusted to fit a large number of data (100-1000) in highly nonlinear situations where it is difficult to provide close initial guesses for the parameters. Sometimes crude estimates are provided by an automated procedure. In such cases the reliability and stability of the iterative procedure for finding the least-squares minimum are of prime importance. At the same time the final convergence to a minimum should be rapid and decisive, not only to save computer time, but also to provide further assurance that a (local) minimum has in fact been found.

The scheme presented here has been designed to provide a high degree of reliability in difficult cases (parameters nonlinear and correlated) without an excessive loss of speed in simple cases (parameters almost linear). These goals are attained by means of three key elements. The first involves the selective use of two distinct types of steps: one based on Newton's method and the other based on a modified gradient method. Roughly speaking, the Newton step is included to provide rapid terminal convergence, while the gradient step is included to avoid saddle points. The second element is a sequence of step-size controls designed to make efficient use of the information in a second-order Taylor expansion and then if necessary to compensate for the approximate nature of the expansion. The third key element is a composite set of convergence tests designed to give prompt but not premature detection of solid convergence.

The basic iterative scheme has been adapted to fit into the pattern of computer program LEARN, a general maximum-likelihood nonlinear fitting program.⁽¹⁾ The present version of this offshoot is called LEARN-I-C. An earlier version, known as LEARN-I-A, was described briefly in a user's manual for programs LEARN and LIKELY.⁽²⁾

The version described here is compatible with an input deck prepared for the original program version LEARN-I.⁽²⁾ A deck prepared for the current standard version LEARN-II⁽²⁾ must be changed slightly to be used with LEARN-I-C.

II. BACKGROUND

There is a large variety of methods designed for the solution of non-linear least-squares problems.^(3,4,5,6) The present method is discussed relative to some of the known methods.

In the Gauss-Newton method and its variants the equations being fitted are first linearized in the parameters, assuming an initially guessed set of values.^(7,8) Thus the usual linear normal equations are brought into play, the only complication being the need to solve them iteratively. If the initial guesses are fairly close, rapid convergence is often obtained. However in difficult cases (parameters highly nonlinear and correlated) the method frequently has trouble in locating a satisfactory solution. It may stop near a saddle or produce an endless series of erratic steps.

In the gradient, or steepest-descent, methods steps of appropriate length are taken opposite to the direction of the formal gradient of the quantity being minimized.^(9,10) The process is reliable provided a sufficiently small step is taken at each iteration. However, in difficult cases the method is slow, because a great number of small steps are usually required.

Occupying a position somewhat between the previous basic methods are several elegant, formal methods.^(11,12,13) They have in common the characteristic that the iterative steps have an evolving nature, initially

being similar to gradient steps but gradually becoming similar to Gauss-Newton steps as the calculation proceeds to a minimum. These methods thus tend to combine the early reliability of the gradient methods with the rapid terminal convergence of the Gauss-Newton methods.

The present scheme, while sharing the spirit of these intermediate methods, differs in several respects. First, it is a tandem method, in that, at each iteration a pair of steps is calculated independently and compared for performance in decreasing the sum of squares. One step is a modified gradient type, and the other is a refinement of the Gauss-Newton step, being a result of the complete second-order Taylor expansion of the sum of squares.^(1,14) Another major difference is that the method is an intuitive one, based essentially on geometrical ideas. Furthermore, to obtain a wide range in the balance between speed and reliability, the present scheme uses a general and flexible method of controlling the step size. Finally a rather complicated set of convergence tests is used to get firm but fast convergence signals.

The least-squares problem under consideration has the following form. We assume a weighted sum of squares Q given by

$$Q = \sum_{m=1}^M W_m [y(\underline{x}_m, \underline{a}) - y_m]^2, \quad (1)$$

where m is a datum index, W_m is a constant weight, y_m is an observed datum, y is the theoretical function being fitted, $\underline{x}_m$ is a set of independent variables corresponding to the datum, and $\underline{a}$ is the unknown set of parameters to be determined. We assume that y is differentiable with respect to $\underline{a}$, and wish to find a minimum without any additional constraints. Thus we must solve the nonlinear equations given by the vanishing of the gradient

$$\partial Q / \partial \underline{a} = 0, \quad (2)$$

subject to the condition that the matrix of second derivatives $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ be positive definite. Equations (2) typically have more than one solution.

We shall use an iterative procedure that is designed to find a solution close (in an appropriate sense) to an initial guess for $\underline{a}$. In the process of finding the solution, we use the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$; thus we must assume that y is twice differentiable. In brief we consider the unconstrained nonlinear least-squares problem for which the first and second partial derivatives of the theoretical function exist and are available numerically. This is a problem which frequently arises in pure and applied science, and particularly in physics.

We note here, in passing, that the effect of inequality constraints can sometimes be obtained by proper definition of nonlinear parameters.⁽¹⁵⁾ For example, if a parameter $a_1 \geq 0$, it can be replaced by a_1^2 in the formula for $y(\underline{x}_m, \underline{a})$.

It must be emphasized that the present work is confined to the first half of the least-squares problem, that is, the process of finding a minimum. We completely ignore the second half, which is concerned with the statistical analysis and testing of the resulting fit.⁽¹⁾

III. GEOMETRY AND ANALYSIS OF NONLINEAR LEAST SQUARES

1. LINEAR CASE

Since the nonlinear problem includes the linear one as a special case, it is useful to review the simpler case. Equation (1) shows that Q is then a quadratic function of $\underline{a}$ and that the contours of constant Q are ellipses.* The set of Equations (2) becomes linear and has a unique solution, which corresponds to the true minimum (barring singular cases). The matrix of the system is positive definite (semidefinite if singular).

The shape of the contours depends on the degree of correlation between the parameters. If the parameters are relatively uncorrelated, the contours are approximately circular (or may be made so simply by rescaling the

* The geometrical language in this report is appropriate to the case of two parameters. For cases of higher dimensionality suitable generalizations, such as, hyperellipsoids, hyperplanes, hyperlines, etc., must be substituted.

parameters). If they are highly correlated, however, the contours become highly elongated. The limiting case occurs when the matrix of the linear case is singular. Then the contours become pairs of parallel straight lines and, instead of a unique minimum, there is a trough of infinite length. The parameters are then perfectly correlated.

Although the linear equations are exactly solvable in principle (barring singular cases), there are two possible sources of difficulty in the actual numerical solution. The first is that the equations may be so numerous that straightforward elimination methods are not practical. Some type of iterative scheme must then be used. This trouble does not arise in the present work, since the number of parameters is not greater than 100. The second difficulty is a high degree of correlation among the parameters. In such a case the set of equations, while not singular, is ill-conditioned. The solution is then extremely sensitive to errors both in the input data and in the numerical process of solution. It is surprisingly easy to generate least-squares cases in which essentially all numerical precision in the parameters is lost because of ill conditioning.⁽¹⁶⁾ In most cases the best remedy is to reduce the number of variable parameters or to increase the range and precision of the data. If the accuracy of the data and theory is good enough, the use of multiple-precision computation may be justified.

2. INTUITIVE GEOMETRY OF THE NONLINEAR CASE

In the general nonlinear case the sum of squares Q may be an extremely complicated function of the parameters $\underline{a}$. The Q -surface defined by Equation (1) then has a complicated shape, containing perhaps several stationary points defined by Equations (2). Furthermore these may include saddle points as well as true (local) minima. For brevity we class local maxima as saddle points, and for simplicity we exclude singular cases (in which the minimum is a trough, or in which the minimum occurs at $\underline{a} = \infty$). The possible presence of several minima and saddle points greatly complicates the solution of nonlinear least-squares problems.

The existence of multiple solutions has both practical and theoretical consequences. The practical effect is that a systematic method must be used in each case to minimize the probability that the correct solution has been overlooked. The theoretical consequence is that the problem as stated is not well posed. Let us first consider the latter question.

It would be highly desirable to modify the statement of the nonlinear least-squares problem in such a way as to ensure a unique solution. Since our method of solution is an iterative one, a natural way is to include the initial guess in the problem statement. For this purpose we could use the concept of a "watershed". We assume that each local minimum is surrounded by a watershed, that is, a region traversed by flow lines (gradients normal to contours) that descend to the minimum. Furthermore we assume that each point in parameter space (except those lying on watershed boundaries) belongs to some watershed. Thus corresponding to each interior point is a unique local minimum. In particular, the minimum corresponding to the initial iterative guess may be defined to be the solution of the problem. An ideal iterative algorithm would proceed from the initial guess to the minimum without ever leaving the original watershed. To what extent these intuitive notions can be made rigorous is unfortunately not known to the writer.

Assuming the validity of the watershed concept and the existence of an ideal iterative algorithm, we then see that the practical problem is solved if every watershed is represented in a set of initial parameter guesses. However, there is probably no completely general method of generating such a set. A systematic, fine-grained search is probably a reasonable procedure. In many practical cases, the parametric domain to be searched may be reduced by consideration of symmetry or periodicity in the theoretical function.

3. ANALYSIS AND ALGEBRA OF THE NONLINEAR CASE

At each point of the iterative process, the basis for the calculation of the next step is often some analytical approximation of the local

situation. In program LEARN this approximation is the full second-order Taylor expansion of Q relative to the current set of parameter values,⁽¹⁾

$$Q_{\underline{a}} = Q_0 + (\partial Q / \partial \underline{a})_0 \cdot \underline{da} + \frac{1}{2} \underline{da} \cdot (\partial^2 Q / \partial \underline{a} \partial \underline{a})_0 \cdot \underline{da} \quad (3)$$

where we use vector and dyad notation to avoid summation signs and indices. Note that this is the equation of a paraboloid that is a local approximation of the true Q -surface defined by Equation (1). The iterative stepping logic uses the fact that the accuracy improves as $\underline{da}$ decreases.

Differentiating Equation (1) twice with respect to $\underline{a}$, we get

$$\partial Q / \partial \underline{a} = 2 \sum_{m=1}^M W_m [y(\underline{x}_m, \underline{a}) - y_m] [\partial y / \partial \underline{a}] \quad (4)$$

$$\partial^2 Q / \partial \underline{a} \partial \underline{a} = 2 \sum_{m=1}^M W_m \left\{ \begin{array}{l} [\partial y / \partial \underline{a}] [\partial y / \partial \underline{a}] \\ + [y(\underline{x}_m, \underline{a}) - y_m] [\partial^2 y / \partial \underline{a} \partial \underline{a}] \end{array} \right\} \quad (5)$$

The key parts of Equation (3) are the quadratic form (the third term on the right) and the associated matrix of second derivatives $\partial^2 Q / \partial \underline{a} \partial \underline{a}$. In the immediate neighborhood of a local minimum the quadratic form is positive definite (in singular cases it may be semidefinite), and the corresponding approximate surface is a paraboloid having a true minimum. In such a case it is reasonable to consider an iterative step going to the approximate minimum. Analytically this consists of solving for $\underline{da}$ in the linear system obtained by applying Equation (2) to Equation (3),

$$(\partial^2 Q / \partial \underline{a} \partial \underline{a})_0 \cdot \underline{da}_M = - (\partial Q / \partial \underline{a})_0 \quad (6)$$

We refer to the parameter changes given by Equations (6) as the matrix step. This procedure is equivalent to solving Equation (2) by Newton's method.

It may happen that the approximate surface is such that its minimum point has a value $Q_a < 0$ (see Figure 2A). Since the true value of Q is never negative, it is obvious that the matrix step is then too long and must be reduced. If necessary, this is performed by a process, called "step refinement", to be described in a later section.

However, if the quadratic form is indefinite, the approximate surface is a paraboloid having a saddle shape. In such a case it is not appropriate to use the matrix step since that would take us to the saddle point. Two distinct possibilities must be considered, depending on whether the current iterative value of Q is (1) less, or (2) greater, than the value at the saddle point on the approximate surface. In the first case an increase in Q would be expected by stepping to the saddle point. To avoid this result we then reverse the matrix step in the step refinement process described later. In the second case a decrease in misfit would be expected by using Equations (6). However, such a step is in danger of leaving the local watershed, since saddle points frequently lie on watershed boundaries. To reduce this danger, while retaining the good directional properties of the matrix step, we then arbitrarily cut the step in half. However, the halving logic obviously breaks down as we approach an actual saddle point on the Q -surface. Thus we must consider an alternative downhill move, such as the modified gradient step, and select that which gives the larger reduction in Q . The decision as to whether the iterative value of Q is less or greater than the saddle value is made by examining the sign of the expected change in Q , which is given by

$$dQ_M = \frac{1}{2} \left(\frac{\partial Q}{\partial \underline{a}} \right)_0 \cdot d\underline{a}_M \quad . \quad (7)$$

It is clearly of decisive importance to know at each iteration whether the matrix of second derivatives is positive definite before using the solution of Equations (6). Fortunately it turns out that the Gaussian elimination scheme, restricted to the use of diagonal elements as pivots,

automatically computes criteria for this decision. It can be shown that the matrix is positive definite if and only if every diagonal element is positive at every stage of the elimination process. This result, which does not appear to be widely known, may be proved by a suitable modification of the discussion on pp. 81-85 of Reference 3.

To complete the discussion of Equations (6), it is necessary to consider cases in which the matrix is singular. Cases of exact singularity are relatively rare and usually are the result of a gross blunder. They are easily recognized by the exact vanishing of a pivot during elimination. More frequent and troublesome are cases of "near-singularity", for which the calculated parameter changes are often much too large. Two techniques for step reduction will be discussed subsequently.

It is appropriate to discuss here a key difference between program LEARN and some of the older nonlinear least-squares programs.^(7,8) These programs usually omit the term in Equation (5) that contains $\partial^2 y / \partial \underline{a} \partial \underline{a}$, the second derivatives of the theoretical function. This omission occurs because the theoretical function is linearized before substitution in Equation (1). Programming is simplified, and computing time for derivatives is reduced. As shown in Equation (5) this approximation is reasonable, if throughout the calculation the theory is nearly linear ($\partial^2 y / \partial \underline{a} \partial \underline{a}$ are small) or the fit is very good ($[y - y_m]$ are small). The resulting approximation to the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ is easily shown to be either positive definite or semidefinite. Thus the approximation is poor near an actual saddle point, and the resulting iterative step is unreliable. If the theory is highly nonlinear, and the initial parameter guess is crude, then it is not unlikely that the iterative solution will pass close to one or more saddle points. Use of the correct formula for $\partial^2 Q / \partial \underline{a} \partial \underline{a}$, Equation (5), and adequate handling of saddle points can give a reliable step in these unfavorable but uncommon cases.

Neglect of the second derivatives $\partial^2 y / \partial \underline{a} \partial \underline{a}$ also has undesirable effects in highly nonlinear cases with large systematic discrepancies between the theory and data. The neglected second term in Equation (5) may then be

large, and the calculated step unreliable. Convergence is thus delayed and in extreme cases sometimes prevented completely. In contrast, use of the complete formula for the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ provides the rapid (second-order) terminal convergence characteristic of Newton's method.⁽¹⁰⁾ A related additional benefit is that the variance-covariance matrix, which is proportional to the inverse of $\partial^2 Q / \partial \underline{a} \partial \underline{a}$, is more accurate if the complete formula is used.

4. MODIFIED GRADIENT STEP

In the previous subsection it was pointed out that the iterative step based on Equation (6) is not appropriate when the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ is indefinite. Instead it may be preferable to take a relatively short step in a "downhill" direction; this suggests some kind of gradient scheme. The simplest method of this type is to take a step in the opposite direction of the formal gradient,

$$\underline{da} = -k(\partial Q / \partial \underline{a})_0, \quad (8)$$

where k is a positive constant. A sufficiently small step in this direction would give a decrease in Q . The peculiar characteristic of the formal gradient is that it gives the direction of the greatest spatial rate of change of Q based on the usual Cartesian metric tensor. Geometrically it gives the direction of the local flow line. Thus a "small" step in this direction would cross few flow lines and would be in little danger of leaving a watershed.

However gradient methods are particularly slow when the parameters are poorly scaled, i.e., when Q has greatly different sensitivity to variations of each parameter. In an attempt to reduce this difficulty we have used a modified gradient step, in which the parameter sensitivities are somewhat equalized. In doing so we are probably exposed to a greater risk of leaving the initial watershed. It is intuitively clear that

a reasonable scaling of parameter changes may be obtained from the coefficients of the second-order Taylor expansion, Equation (3). This process divides naturally into two parts: finding first the direction of the step and then its length.

To specify the direction of the step, we wish to find a set of raw parameter changes, $\delta \underline{a}$, each of which acting alone would produce a significant decrease (say 50%) in Q . For this purpose, we use the parabolas that are obtained by successively setting all but one of the differentials $d\underline{a}$ to zero in Equation (3). As shown schematically in Figure 1, there are three alternative possibilities, depending on the signs and magnitudes of the Taylor coefficients. In parts A and B of the figure, the parabolas intersect the line $Q = \frac{1}{2}Q_0$; the raw parameter change δa is then chosen to be the value of da at the nearest intersection. In part C the parabola does not intersect the line $Q = \frac{1}{2}Q_0$; the raw parameter change then corresponds to the minimum of the parabola. The critical ratio Q/Q_0 to be used is a matter of somewhat arbitrary choice and probably should lie between 1/2 and 0. The latter limit is obvious, since Q is never negative. The value 1/2 is a rather conservative choice (biased in favor small changes).

Now that the direction of the gradient step is determined, we get its length by the step refinement process described in the next section. It should be noted that the gradient step usually is not a useful alternative to the matrix step in the final stages of iteration. Its convergence rate is only first-order, while the matrix step has much more rapid second-order convergence.⁽¹⁰⁾ Thus it is desirable to suppress the gradient step, if an acceptable matrix step is available in the final stages.

5. STEP REFINEMENT

It was mentioned earlier that the raw matrix step $\delta \underline{a}_M$ computed from Equation (6) may be too long (trying to make $Q_a < 0$) or even in the wrong direction (trying to climb up to a saddle point). Similarly the raw gradient step given by the parameter changes $\delta \underline{a}$, although always in a downhill direction, may be too long or too short. For each type of step we know a plausible direction but need to determine a suitable amplitude.

FIGURE 1. Schematic Geometry for One Component of the Modified Gradient Step (raw)

Since the refinement of the gradient step is somewhat simpler, let us consider it first. We now set

$$\underline{da} = z\delta\underline{a}, \quad (9)$$

where the refinement factor z is to be determined. Combining Equations (3) and (9) we get

$$Q_a = Q_o + [\partial Q / \partial \underline{a}]_o \cdot \delta \underline{a} z + \frac{1}{2} [\delta \underline{a} \cdot (\partial^2 Q / \partial \underline{a} \partial \underline{a})_o \cdot \delta \underline{a}] z^2. \quad (10)$$

Depending on the values of the bracketted coefficients of this expression, we may have one of the three parabolas shown in parts A, B and C of Figure 2. Remember that the coefficient of z is negative. In cases A and B the parabola intersects the abscissa $Q_a = 0$, and we select the smallest root z_r of $Q_a(z) = 0$, corresponding to the nearest intersection. If, as in case C, the parabola does not intersect the abscissa, we select the root z_r of $dQ_a(z)/dz = 0$, corresponding to the minimum.

For the refinement of the matrix step, we let

$$\underline{da} = z \underline{da}_M. \quad (11)$$

Combining Equations (3) and (11) and simplifying with the aid of Equations (6) and (7), we get

$$Q_a = Q_o + (2dQ_M)z - (dQ_M)z^2. \quad (12)$$

Noting that the extremum of this parabola is at $z = 1$, we see that the three possible cases are shown in parts A, C and D of Figure 2. In cases A and D the parabola intersects the abscissa, and as before we select the nearest intersection. Note that the direction of the matrix step is reversed in case D. In case C the parabola does not intersect the

FIGURE 2. Schematic Geometry for Refinement of Modified Gradient Step or Matrix Step

abscissa, and we select the minimum at $z_r = 1$. In this last case the raw matrix step is not changed. However, if the $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ matrix is indefinite and $dQ_M < 0$, the refined matrix step is later cut in half.

The step refinement process is basically similar to that described in Section 4 for the determination of the components of the raw gradient step $\delta \underline{a}$. In fact the same subroutine is used for these two parts of the calculation. In passing it should be mentioned that this subroutine has provision for degenerate forms of the parabolas, that is, straight lines. It is interesting to note that the off-diagonal elements of the $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ matrix are used in the determination of the length of the gradient step but not its direction.

6. NUMERICAL ASPECTS

The nonlinear problem is subject to the same numerical difficulties as the linear case discussed earlier. If the parameters are highly correlated, Equations (6) are ill-conditioned and the resulting solution may not be very accurate. Unless there is a complete loss of significance, however, the net effect usually is merely a slight increase in the number of iterations. When all significance is lost, the iterations may wander irregularly and aimlessly. A special convergence test is used to detect this condition.

IV. CONTROL OF ITERATIVE STEPS

Experience has shown that the step refinement scheme alone is not enough to provide reliable, rapid progress in reducing Q in highly nonlinear problems. The difficulty shows up as steps, mostly matrix steps, that result in unexpectedly large increases in Q . It is clear that such steps are too long, in that they have gone beyond the range of applicability of the second-order Taylor expansion, Equation (3). Such steps may ricochet wildly from watershed to watershed without any semblance of convergence.

Three additional types of tests are thus made before a matrix or gradient step is considered to be acceptable. Unless all tests are satisfied, the step is reduced by $1/2$ and tested again, up to a maximum of 19 times. If steps of each type are found acceptable, that giving the smaller actual value of Q is adopted. If only one type is acceptable, it is immediately adopted. If neither type is acceptable, a message is printed, and iteration is stopped.

The first test for the acceptability of a (reduced) step requires a certain degree of accuracy in the prediction of Q or in the change in Q . It requires Q to be within 1% or ΔQ within 10% of the values predicted by the second-order approximation. The test is a composite one, since conditions are quite different at the beginning and end of iteration. At first ΔQ is rather large and predictable, while near convergence it is small and greatly affected by rounding errors.

The second test requires that Q cannot increase by more than 0.01% of its current value. The idea here is to permit fluctuations of Q due to rounding errors, while preventing large systematic increases.

The third test requires that all parameters lie within the ranges specified by the user. The primary purpose of this test is to prevent illegal function calls.

These simple controls of the step size produce a remarkable steadying of the iteration process in difficult cases having correlated, nonlinear parameters without slowing the simple cases having almost linear parameters.

V. CONVERGENCE TESTS

The convergence tests are an important part of an iterative scheme because they interact so strongly with the main computational steps. The tests must fit the scheme, and vice versa. This is especially true when we are trying to iterate to "effective" convergence, that is, when we are

trying to get a solution as accurate as the number of significant figures and the condition of the equations allow. Such strict convergence is sometimes needed to prevent false indications of convergence near a saddle point, for example.

A scheme of two-level convergence testing has been adopted to get increased reliability. The detailed convergence tests are made only if it is first determined that the iteration process is in its terminal phase. An iterative step is considered to be within the terminal phase if all of the following requirements are met: (1) the matrix of second derivatives $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ is positive definite, (2) the current step does not change the determinant of this matrix more than 1% of its previous value, and (3) the current step is a full, refined step.

The detailed convergence test is also a composite. Since it is difficult to develop a single test that reliably gives a prompt but not premature convergence signal, we have adopted three rather strict tests that are used in parallel. That is, convergence is signalled if any of the three tests is satisfied. The following quantities are tested for smallness: (1) the partial derivatives $\partial Q / \partial \underline{a}$, (2) the actual parameter changes and (3) the predicted change in Q .

Let us briefly describe the tests used. The test for $\partial Q / \partial \underline{a}$ is based on the decomposition of the right-hand side of Equation (4) into two sums that balance when $\partial Q / \partial \underline{a} = 0$. The test is that each component satisfy

$$|\partial Q / \partial \underline{a}| \leq f_g \left| 2 \sum_m w_m y_m \partial y / \partial \underline{a} \right|, \quad (13)$$

where f_g is preset to 10^{-8} .

A simple test for the parameter changes is that each component satisfy

$$|\Delta \underline{a}| \leq f_a |\underline{a}|, \quad (14)$$

where $\Delta \underline{a}$ is the difference between two successive parameter sets, and f_a is preset to 10^{-8} . However, a more complicated test of the parameter changes is needed to stop the noisy, oscillatory movement of the iterative point near a minimum in ill-conditioned cases. Such a test is supplied by the ratio of the net movement of each parameter to the gross movement in the previous n steps (all in the terminal phase)

$$R_n = \left| \sum_n \Delta \underline{a}_n \right| / \sum_n \left| \Delta \underline{a}_n \right| \quad . \quad (15)$$

A parameter is considered to be converged if this trend ratio

$$R_n \leq C_n \quad , \quad (16)$$

where tests are currently made only for $n = 4$ and $n = 10$, and C_4 and C_{10} are preset to 0.5 and 0.3, respectively. To satisfy the composite test on parameter changes, each parameter has to satisfy inequality (14), or (16) with $n = 4$, or (16) with $n = 10$ at the same step.

The test for the predicted change in Q for the refined full step is

$$|dQ_R| \leq f_Q Q, \quad (17)$$

where f_Q is preset to 10^{-8} . Experience shows that this test is usually the first to signal convergence.

All of the numerical values of the convergence criteria can be reset by the user simply by optional data input.

VI. ITERATIVE STRATEGY

We are now in a position to give a brief summary of the iterative strategy together with an outline of the iterative process as it systematically reduces the sum of squares Q . There are two phases.

1. INITIAL PHASE

In this, the most complicated part of the iterative calculation, the parameters may be far from a minimum and may in fact be in a saddle-like region of the Q -surface. That is, the approximating paraboloid may have a saddle point. To move efficiently past such a saddle point both matrix and gradient steps must be calculated. The raw steps of both types are refined to a length appropriate to the second-order Taylor approximation. This may involve the halving or reversal of a matrix step. Because of the limited accuracy of the second-order approximation, even the refined steps may have to be halved one or more times to provide reliable progress based on accurate predictions. After the better of the two steps is selected, we determine whether or not the process has reached the terminal phase by checking three preliminary criteria. If the terminal phase has not been reached, the detailed convergence tests are skipped, and calculation of the next step is begun. If the terminal phase has been reached, we proceed as follows.

2. TERMINAL PHASE

As soon as the terminal phase is entered, the detailed convergence tests are made. If any of three convergence tests is satisfied, the iteration process is stopped. If not, we begin calculation of the next step. Now, however, we are probably close to a minimum. Thus if the full-length refined matrix step is acceptable, we select it immediately by suppressing competition with the gradient step. If not, we go through the usual halving procedure and step competition. After every step selection, we determine whether or not the iterative process is still in the terminal phase. If so, we go on to the detailed convergence tests. If not, the process reverts to the initial phase and, skipping the convergence tests, goes to the beginning of the next step.

3. CONTINGENCIES

If at any stage neither the (reduced) matrix step nor the (reduced) gradient step is acceptable, iteration is stopped. If convergence is not obtained within the number of iterations specified by the user, the process is stopped.

VII. IMPLEMENTATION OF ALGORITHM IN LEARN-I-C

The iterative algorithm outlined in Section V has been incorporated into the FORTRAN V program LEARN by extensive revision of subroutine MINMAX and its associated subroutines. A few new subroutines have also been written. This new version is designated LEARN-I-C.

1. SUBROUTINE MINMAX

Various features of the modified version of MINMAX not already mentioned will be described briefly here. The user has the option of suppressing through data input an arbitrary fraction of the written iterative output; this may be useful in large jobs transmitted by remote terminals. However, the zeroth, last and best (smallest Q) iteration are always written. If iterative output is partially suppressed, a table of the last 100 values of Q is printed at the end of iteration.

If convergence is detected, a message is written indicating which of the three tests has been satisfied along with the numerical value of the test criterion.

After iteration has been completed, two types of summary information are written. One is a table of values for the last ten iterations of the trend ratio R_n defined by Equations (15). This table, which provides at a glance a picture of recent iterative progress, is particularly useful for slowly convergent, ill-conditioned cases. The second is a short table giving the elapsed execution times accumulated in various important segments of the iterative process. Such information is valuable to the user in judging the costs of choosing various options. It is also useful to the programmer in judging the worth and cost of program modifications.

The LAG feature has been removed from MINMAX. A flow diagram of MINMAX is given in Figure 3. Note that the convergence tests are made early in the iterative loop, and that iterative output is written midway. Note also that the crude gradient step and the related parabolic coefficients are computed before the Gaussian elimination is begun, since the elimination process destroys the original matrix elements. It must also be mentioned that the block indicating the retrieval and writing of the best iteration is oversimplified. The program actually stores and retrieves only part of the needed data; it generates the omitted portion by an extra passage through the main loop.

Several options related to the iterative process are available to the user at execution time via optional NAMELIST input. The maximum number of matrix and gradient (sub) steps may be changed independently. For example, matrix steps may be completely suppressed. The user may also specify a minimum number of iterations. The user may elect to find the nearest extremum of Q instead of a minimum. In such a case gradient steps are automatically suppressed. The user may also elect to avoid the programming effort and computation time connected with the matrix of second derivatives $\partial^2 Q / \partial a \partial a$. However, this last option must be exercised with great care, since it may reduce both the reliability of the iterative process and the accuracy of parameter error estimates. If either of the last two options is elected, it may be desirable to change the values of some of the iterative control criteria.

Other changes have been made to increase the speed of execution. All references to the rounding functions (FADR, FMPR and FDPR) have been eliminated from the iterative loop. Furthermore, the function previously performed by subroutine CHANCE has been incorporated directly into MINMAX. Note, however, that other subroutines in the package still call CHANCE. The cost of these changes is a (usually) negligible loss of numerical accuracy and a tolerable loss of statistical versatility.

2. OTHER MODIFIED SUBROUTINES

Nine other original LEARN subroutines have been modified in various degrees. The matrix division subroutine FDPM has been completely rewritten

FIGURE 3. Schematic Flow Diagram of Subroutine MINMAX

to use only diagonal pivots in Gaussian elimination of a symmetric matrix. If the matrix is found to be singular, the process is aborted, and the solution is set to zero.

The output routine PATH has been changed to increase the amount of written iterative information. The following items have been added: the actual parameter changes $\Delta \underline{a}$, the formal gradient components $\partial Q / \partial \underline{a}$, the diagonal elements of the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$, the modified gradient DQN, the type and fraction of the step used, the refinement factor z , the predicted change in Q for the refined step dQ_R , the first and second order coefficients of the gradient parabola QS and QSS , and the natural logarithm of the determinant of the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$ (if positive definite only) DLN.

New common blocks NEW and NEWB have been added to the iterative subroutines. The BLOCK DATA subroutine FIT has been changed to preset the new control parameters, and subroutine LOAD has been changed to provide for optional resetting by NAMELIST data input. Subroutine LIST also writes the new control parameters, while the main program LIKELY contains modified input instructions.

Several minor output improvements have been made. PAGE has been modified to write the case number on each printed page. The plotting subroutines CALCOM and MAPS have been changed to provide greater flexibility in plotting graphs. The new options are: the number of times each graph is traced, the width of the X-axis margin, and the cycle length along the X-axis of the graph paper.

3. NEW SUBPROGRAMS

Two new subprograms have been written for the iterative process. Subroutine QCALC is used to calculate the sum of squares Q for the full and fractional refined steps. Function PRBOLA (C, B, A, F, Z) carries out the step-refinement operations on the parabola defined by

$$Q = C + Bz + \frac{1}{2}Az^2. \quad (18)$$

The critical ratio Q/C is specified by F .

VIII. PROGRAM DETAILS

We now present programming, input and output details particularly pertinent to LEARN-I-C. It is assumed that the reader is familiar with the standard LEARN package.^(1,2) LEARN-I-C is compatible with the original LEARN-I package. That is to say, it is possible to use this version without any changes in the user's input data, THEORY subroutine or auxiliary subroutines (PUTIN, PUTOUT, PLOT, etc.). However, in many cases a few relatively simple changes by the user can greatly increase the effectiveness of the program.

LEARN-I-C is not compatible with LEARN-II. The principal differences are in the names of the NAMELIST input lists and in the contents of the named common blocks. However, it is easy to make the necessary input deck changes.

THEORY Subroutine - It may not be worthwhile to rewrite a small THEORY subroutine that is used only occasionally. However, execution time may be substantially reduced for large or frequently used THEORY subroutines by making two kinds of changes. First, replace any references to the floating arithmetic rounding functions (FADR, FMPR, FDPR) with ordinary FORTRAN operations (+, *, /). Secondly, use the parameter MAX to bypass the first partial derivatives of y, as well as the second. If MAX = 0, calculate y only. If MAX = 1, calculate y and $\partial y / \partial \underline{a}$. If MAX = 2, calculate y, $\partial y / \partial \underline{a}$ and $\partial^2 y / \partial \underline{a} \partial \underline{a}$.

New Input Data (Optional) - The NAMELIST input list \$INPUT has been augmented to include several new control parameters important to the iterative process. Thus, the preset values may be optionally overloaded for flexible control. These parameters are as follows:

- (1) NP is the reciprocal of the fraction of iterative output to be written. (However, the zeroth, last and lowest iterations are always written.) It is particularly useful for reducing output at remote terminals. (Preset to 1; minimum 1; no maximum.)

- (2) NM is the maximum number of matrix steps (full and fractional) to be considered at each stage in the iterative process. Increasing NM tends to increase reliability and running time. (Preset to 20; minimum 0; no maximum.)
- (3) NG is the maximum number of gradient steps (full and fractional) to be considered. Increasing NG tends to increase reliability and running time. (Preset to 20; minimum 0; no maximum.)
- (4) IMIN is the minimum number of iterations. (Preset to 0; minimum 0; no maximum.)
- (5) ISAD is the flag for type of solution to be sought. 0 for a minimum only, 1 for any extremum. (Preset to 0.)
- (6) IDDY is the flag for status of $\partial^2 y / \partial \underline{a} \partial \underline{a}$ matrix. 0 for normal use, 1 for bypass. (Preset to 0.)
- (7) R is the critical (floor) ratio used in the step-refinement procedure. Increasing R tends to increase reliability and running time. (Preset to 0.0; minimum 0.0; maximum <1.0.)
- (8) DTEST is the criterion for testing the constancy (in successive iterations) of the determinant of the matrix $\partial^2 Q / \partial \underline{a} \partial \underline{a}$. Decreasing DTEST makes the test more stringent. (Preset to 1.0×10^{-2} .)
- (9) GCRIT is the convergence criterion for the gradient components $\partial Q / \partial \underline{a}$. Decreasing GCRIT (as well as any of the following convergence criteria) makes the convergence test more stringent. (Preset to 1.0×10^{-8} .)
- (10) ACRIT is the convergence criterion for the parameter changes $\Delta \underline{a}$. (Preset to 1.0×10^{-8} .)
- (11) TCRT4 is the convergence criterion for the trend ratio R_4 . (Preset to 0.5.)
- (12) TCRT10 is the convergence criterion for the trend ratio R_{10} . (Preset to 0.3.)
- (13) QCRIT is the convergence criterion for the predicted change dQ_R . (Preset to 1.0×10^{-8} .)

- (14) QACC is the accuracy criterion for Q in the iterative stepping. (Preset to 0.01.)
- (15) DQACC is the accuracy criterion for ΔQ in the iterative stepping. (Preset to 0.1.)
- (16) QINC is the maximum fractional increase in Q in the iterative stepping. (Preset to 1.0×10^{-4} .)

Optional input related to plotting is as follows:

- (17) NTR is the number of times each graph is to be traced. (Preset to 2; minimum 1; no maximum.)
- (18) MAR is the width in mils of the margin along the X-axis. (Preset to 500; minimum 0; maximum 30000.)
- (19) CYCLE is the length in inches of the cycle along the X-axis on gridded paper. (Preset to 1.)

Written Output - Except for listing the new control parameters at the start of the case, only the output for the iterative part of the program has been changed. The following items are written for each iteration (unless suppressed):

- (1) A(J) are the current parameter values.
- (2) DELA(J) are the previous parameter changes (by subtraction).
- (3) DQM/DA(J) are the components of the gradient partition.
- (4) DQ/DA(J) are the formal gradient components.
- (5) DIA(J) are the original diagonal elements of the matrix $\partial^2 Q / \partial a \partial a$.
- (6) DDQ/DA/DA(J,J) are the pivot elements of the matrix.
- (7) DDQ(JJ+1,J) are the components of the solution vector.
- (8) DQN(J) are the components of the modified gradient.
- (9) I = iteration number.
- (10) Class of previous step. (M = matrix, G = gradient, N = none.)
- (11) Length of previous step. (1 = full step, 2 = half step, 3 = quarter step, etc.)

- (12) Z = refinement factor for previous step.
- (13) DQR = predicted change in misfit for refined previous step.
- (14) Q = current value of sum of squares.
- (15) QS = linear coefficient for parabola for modified-gradient direction.
- (16) QSS = quadratic coefficient for above parabola.
- (17) PDQ = predicted change in Q for next full matrix step.
- (18) DLN = natural logarithm of the determinant of the matrix $\partial^2 Q / \partial a \partial a$.

After iteration has been completed several types of summary output are written:

- (1) The index of the iteration having the smallest Q.
- (2) The most recent 100 values of Q (written only when the detailed output is partially suppressed).
- (3) The trend ratios for each parameter for the previous 10 steps.
- (4) Cumulative intervals of time spent in execution of various parts of the iterative scheme.
- (5) A warning message indicating the first instance of a parameter's reaching its minimum or maximum value.

IX. PRACTICAL REMARKS FOR USERS

It is probably presumptuous to offer generalized advice to users of non-linear least-squares methods. The entire field seems to be covered with destroyed misconceptions. However, the following comments are offered in the hope that they may be useful.

1. NEED FOR CARE

Since nonlinear least-squares methods are mostly applied to complicated problems, it is usually necessary to do a relatively large amount of thinking and analysis both before and after the calculations. A stable, reliable least-squares program is a powerful tool, which can mangle the scientific content of a problem without any sign of difficulty unless the operator is alert and well prepared.

One of the worst sources of difficulty, particularly if the parameters have interpretable physical significance, is correlation between the parameters. It is very easy to ask that a set of data determine more parameters than it has leverage for. Invariably then the parameters are highly correlated and poorly determined. However, this difficulty is not usually important if the main purpose is to fit data for purposes of smoothing or interpolation.

Another possible source of difficulty is a substantial systematic discrepancy between data and theory, particularly when the parameters are correlated. The parameter values and error estimates are then highly suspect. The possibility of a spurious solution due to multiple minima must be considered. No important least-squares fit should be accepted without a look at a graphical comparison of data and theory.

2. KEY INPUT

It is strongly recommended, in general, that the parameter limits AMIN and AMAX be set to the smallest and largest values available ($\approx \pm 1 \times 10^{38}$ for single precision in the UNIVAC 1108), since the process of constraining a parameter in this way frequently prevents convergence. The limiting feature was retained primarily for cases in which certain ranges of parameter values are not permitted in the theoretical function, for example, the logarithm.

3. KEY OUTPUT

We confine our attention here to the output that helps to judge the quality of the solution and the iterative steps leading to it. Experience shows that the most sensitive indicator of solid convergence is dQ_M , the change in Q predicted by the raw matrix step (defined in Equation (7) and designated PDQ in the program output). Just before convergence this quantity usually decreases by several orders of magnitude in one iteration. After convergence it remains at a fairly constant order of magnitude governed by rounding errors. The size of the formal gradient components relative to the partition components is a rough measure of the degree to which the fundamental

Equations (2) are satisfied. In reasonably well-conditioned problems, the former are several orders of magnitude smaller than the latter.

The matrix pivots are significant indicators of the Q-surface. Negative values of pivots indicate a saddle-like region. In addition, the size of the pivots should be compared with the diagonal matrix elements. Highly correlated parameters give pivots of very small relative magnitude.

Additional checks on positive definiteness are provided by QSS, which should be positive, and PDQ, which should be negative.

The behavior of Q, of course, is an important indication of possible trouble.

After the completion of iteration, the table of trend ratios for the ten most recent steps provides a quick indication of the parameters that have been changing steadily and those which have been oscillating back and forth. In extremely ill-conditioned cases, all parameters tend to oscillate in the terminal phase of iteration.

X. MISCELLANEOUS TOPICS

It is sometimes desirable to constrain the parameters in some manner when fitting data that are not accurate or extensive enough to determine all of the parameters satisfactorily. The simplest method is to hold one or more parameters fixed. Somewhat more general is a set of explicit linear (equality) constraints. An example of the latter is

$$a_3 = 3a_1 - 5a_2 + 9, \quad (19)$$

where a_1 and a_2 are free parameters. Easily generalized methods have been developed to handle both of these types of constraints by incorporating standard sets of instructions in subroutines PUTIN and THEORY. The iterative process and statistical analysis for the free parameters are performed in the same manner as if the constrained parameters had been completely replaced from the beginning.

One of the most tedious parts of getting a new least-squares analysis into operation is checking the calculation of the partial derivatives $\partial y / \partial a$ and $\partial^2 y / \partial a \partial a$. A special subroutine DIFFER has been developed, which performs differencing checks of the formulas programmed in the THEORY subroutine. By performing numerical tests it is thus possible to locate analytical, programming and compiling errors.

ACKNOWLEDGMENTS

The author wishes to thank B. H. Duane, the originator of the LEARN program, for generous cooperation and many stimulating discussions. Thanks are also due to R. P. Matsen for several useful discussions and to J. R. Morrey, G. D. Seybold and W. P. Stinson for providing instructive test cases.

REFERENCES

1. B. H. Duane. Maximum Likelihood Nonlinear Correlated Fields (BNW Program LIKELY), BNWL-390, Pacific Northwest Laboratory, Richland, WA, September 1967.
2. G. D. Seybold, Users Aid: Programs LEARN and LIKELY, BNWL-1057, Pacific Northwest Laboratory, Richland, WA, May 1969.
3. D. J. Wilde and C. S. Beightler. Foundations of Optimization, Prentice-Hall, Englewood Cliffs, NJ, 1967.
4. A. Lavi and T. P. Vogl, Ed. Recent Advances in Optimization Techniques, Wiley, NY, 1966.
5. N. R. Draper and H. Smith. Applied Regression Analysis, Wiley, NY, 1966.
6. H. A. Spang. "A Review of Minimization Techniques for Nonlinear Functions." SIAM Review, vol. 4, p. 343. 1962.
7. H. O. Hartley. "The Modified Gauss-Newton Method for the Fitting of Non-linear Regression Functions by Least Squares." Technometrics, vol. 3, p. 269. 1961.
8. R. H. Moore and R. K. Zeigler. The Solution of the General Least Squares Problem with Reference to High-Speed Computers, LA-2367. Los Alamos Scientific Laboratory, March 1960.
9. H. B. Curry. "The Method of Steepest Descent for Nonlinear Minimization Problems." Quart. Appl. Math., vol. 2, p. 258. 1944.
10. J. B. Crockett and H. Chernoff. "Gradient Methods of Maximization." Pacific J. Math., Vol, 5, p. 33. 1955.
11. K. Levenberg. "A Method for the Solution of Certain Nonlinear Problems in Least Squares." Quart. Appl. Math., vol. 2, p. 164. 1944.
12. W. C. Davidon. Variable Metric Method for Minimization, ANL-5990 (Rev.). Argonne National Laboratory, November 1959.
13. D. W. Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." J. Soc. Ind. Appl. Math., vol. 2, p. 431. 1963.
14. J. C. Grout. A General Computer Program for Solving Nonlinear Regression Problems, IC-8423. Bureau of Mines, 1969.
15. M. J. Box. "A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems." The Computer Journal, vol. 9, p. 67. 1966.
16. R. H. Wampler. "An Evaluation of Linear Least Squares Computer Programs." J. Res. NBS, vol. 73B, p. 59. 1969.

DISTRIBUTION

No. of
Copies

OFFSITE

2	<u>AEC Chicago Patent Group</u> G. H. Lee
8	<u>AEC Division of Reactor Development and Technology</u> Asst. Dir. for Project Management Chief, Water Projects Branch (2) Chief, Gas Cooled Projects Branch Asst. Dir. for Reactor Technology Chief, Reactor Physics Branch (2) Chief, Core Design Branch
1	<u>AEC Division of Nuclear Materials Safeguards</u> H. Werner
1	<u>AEC Division of International Affairs</u> M. B. Kratzer
1	<u>AEC Division of Production</u> F. P. Baranowski
2	<u>AEC Division of Licensing and Regulations</u> C. D. Luke R. J. Odegaarden
1	<u>AEC Division of Research</u> G. A. Kolstad
3	<u>AEC Division of Technical Information Extension</u>
1	<u>AEC Savannah River Operations Office</u> R. Thorne
4	<u>Argonne National Laboratory</u> Reactor Physics Constants Center (2) R. Avery P. Gast

No. of
Copies

- 3 Atomic Energy of Canada Limited
Chalk River Nuclear Laboratory
Reactor Physics Branch
Chalk River, Ontario, Canada
 M. Duret
 C. Miller
 L. Pease
- 3 Atomics International (LMEC)
Post Office Box 309
Canoga Park, California 91305
 H. Alter
 N. Ketzlach
- 1 Australian Atomic Energy Commission
AAEC Research Establishment
Private Mail Bag, Sutherland 2232
N. S. W., Australia
 Dr. J. L. Symonds
 Chief, Physics Division
- 3 Babcock and Wilcox
P. O. Box 1260
Lynchburg, VA 24505
 H. Jones
 D. H. Roy
 W. A. Wittkopf
- 1 Bechtel Corporation
Vernon, California
 M. Aronchick
- 1 Bettis Laboratory, Westinghouse Electric Company
 J. J. Taylor
- 1 Bhabha Atomic Research Centre
Trombay, Bombay-85, India
 S. R. Dwivedi, Theoretical Physics Section/RED
 Central Complex Bldg.
- 3 Brookhaven National Laboratory
 J. Chernick
 H. Kouts
 S. Pearlstein

No. of
Copies

- 1 California Institute of Technology
Pasadena, California 91109
H. Lurie, Engineering Division
- 1 Catholic University of America
Dept. of Nuclear Science and Engineering
Washington, D.C.
G. L. Simmons
- 10 C.E.N. - Saclay
Boite Postale 2
Gif-Sur-Yvette (S et O), France
B. LaPonche
P. Lecorche
G. Vendryes
- 1 CNEN - Casaccia
00060 - S. Maria Di Galeria
Rome, Italy
Paolo Loizzo
- 2 CNEN-Centro Studi-Nucleaire
Casaccia, Rome, Italy
Ugo Farinelli
Augusto Gandini
- 1 Combustion Engineering, Nuclear Division
P. O. Box 500
Windsor, Connecticut 06095
R. Harding
- 2 Cornell University, Ithaca, N. Y.
R. T. Cuykendall, Engineering Physics
M. Nelkin
- 2 Assoc. C.E.N. Belgo Nucleaire
35 Rue Des Colonies, Belgium
H. Bairiot
L. Bindler

No. of
Copies

- 2 Duke University
Durham, N. C.
H. W. Newson, Physics Department
W. J. Seeley, School of Engineering
- 6 E. I. du Pont de Nemours & Co., Inc.,
Savannah River Laboratory
H. K. Clark
J. L. Crandall
G. Dessauer
E. J. Hennelly
H. Honeck
J. Suich
- 10 EURATOM
53, Rue Billiard
Brussels 4, Belgium
A. de Stordeur
- 1 FFR - AB Afomenergi
Studsvik, Pa NYKOPING
Sweden
Evelyn Sokolowski
- 4 Gulf General Atomic, Inc.
P. O. Box 608
San Diego, California 92112
J. M. Neill
L. W. Nordheim
H. B. Stewart
G. D. Trimble
- 3 General Electric Company
Knolls Atomic Power Laboratory
R. Ehrlich
C. Lubitz
K. W. Seeman
- 5 General Electric Company
175 Curtner Avenue
San Jose, California 95112
R. L. Crowther
M. R. Egan
D. L. Fischer
P. Greebler
S. Levy

No. of
Copies

- 1 General Electric Company
R&D Center
Schenectady, New York
W. R. Kanne
- 1 General Electric Company
Nucleonics Laboratory
Dr. H. W. Alter, Manager
P. O. Box 846
Pleasanton, California 94566
- 2 Idaho Nuclear Corporation
R. A. Grimesey
- 1 Istanbul Technical University
Giimiis, Suyer, Istanbul, Turkey
Director, Nuclear Energy Institute
- 1 Japan Atomic Energy Research Institute (JAERI)
Tokaimura, Naka-gun, Ibarakiken, Japan
Jhime Sakata
- 1 Japan Atomic Energy Institute (JPDR-TCA)
Tokaimura, Ibarakiken, Japan
Shojiro Matsuura
- 1 Kansas State University
Manhattan, Kansas
W. R. Kimel, Nuclear Engineering
- 1 Kernforschungszentrum Karlsruhe
7500 Karlsruhe, Germany
Professor W. Haefele
- 1 Los Alamos Scientific Laboratory
G. E. Hansen
- 1 Manhattan College
Riverdale, New York, New York
Brother Gabriel Kane

No. of
Copies

- 2 Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Professor Irving Kaplan
D. D. Lanning
- 1 North American Rockwell Corp.
Sci. Center
1049 Camino Dos Rios
Thousand Oaks, Cal. 91360
E. R. Cohen
- 1 North Carolina State Univ.
Raleigh, N. C. 27607
R. L. Murray
- 1 Nuclear Materials and Equipment Corp.
Penn Center
Apollo, Pennsylvania 15613
Karl Puechl
- 2 Oak Ridge National Laboratory
F. C. Maienschein
A. M. Perry
- 1 Pakistan Institute of Nuclear Science & Technology
P. O. Nilore, Rawalpindi, Pakistan
M. A. Mannan
- 1 Penn. State University
University Park, Pa. 16802
W. F. Witzig
- 1 Phillips Petroleum Company
Idaho Falls, Idaho
W. B. Lewis
- 1 Power Reactor and Nuclear Fuel Development Corp.
9-13, 1-chome, Akasaka
Minato-ku, Tokyo, Japan
Setsuo Kobayashi

No. of
Copies

- 1 Purdue University
West Lafayette, Indiana 47907
P. N. Powers, Nuc. Eng. Dept.
- 1 Rensselaer Polytechnic Institute
E. R. Gaerttner
- 1 S.C.K. - C.E.N.
Brl,
Mol-Donk, Belgium
Dr. H. Vanden Broeck
- 1 Tennessee Valley Authority
310 Power Bldg., Chattanooga, Tenn.
H. B. Brooks
- 1 Union Carbide Corporation (ORNL)
E. B. Johnson
- 2 United Kingdom Atomic Energy Agency
Atomic Weapons Research Establishment
Aldermaston, Berkshire, UK
R. C. Lane (1)
Authority Health and Safety Branch,
Safeguards Division, Risley,
Warrington, UK
J. H. Chalmers (1)
- 1 United Kingdom Atomic Energy Authority
General Reactor Physics Division
Winfrith, England
C. G. Campbell
Dr. J. G. Tyror
- 1 United Nuclear Corporation
White Plains, N.Y.
J. R. Tomonto
- 1 University of Arizona
Tucson, Arizona
Monte V. Davis, Nuclear Engineering Department

No. of
Copies

- 1 University of Florida
Gainesville, Florida
- 1 University of Illinois
Urbana, Illinois
Frederick Seitz, Physics Department
- 1 University of Minnesota
Minneapolis, Minnesota
H. S. Isben, Chemical Engineering Department
- 1 University of Nevada
Reno, Nevada
T. V. Frazier, Physics Department
- 1 University of Notre Dame
Notre Dame, Indiana
E. W. Jerger, Department of Mechanical Engineering
- 1 University of Oregon
Eugene, Oregon
J. L. Powell, Physics Department
- 2 University of Tennessee
Knoxville, Tennessee
A. H. Nielsen, Physics Department
P. F. Pasqua, Nuclear Engineering Department
- 1 University of Toledo
Toledo, Ohio
J. J. Turin
- 3 University of Washington
Seattle, Washington
A. L. Babb, Department of Nuclear Engineering
K. L. Garlid
M. A. Robkin
- 1 University of Wisconsin
Madison, Wisconsin
M. W. Carbon, Nucl. Eng. Com.

No. of
Copies

1	<u>U.S. Atomic Energy Commission DNR</u>
Division of Naval Reactors
Washington, D.C. 20545

Dr. A. Radkowsky |
| 1 | <u>Virginia Polytechnic Institute</u>
Blacksburg, Virginia

A. Robeson, Physics Department |
| 1 | <u>Washington State University</u>
Pullman, Washington

J. P. Spielman, College of Engineering |
| 7 | <u>Westinghouse Electric</u>
Atomic Power Division
P. O. Box 355
Pittsburgh, Pennsylvania 15230

C. A. Anderson
R. J. French
W. D. Leggett
R. J. Nodvik
W. L. Orr
J. B. Roll
J. R. Worden |

ONSITE-HANFORD

1	<u>AEC Chicago Patent Group</u>

R. M. Poteat (Richland) |
| 2 | <u>RDT Assistant Director for Pacific Northwest Programs</u>

T. Nemzek (Richland)
W. E. Fry |
| 2 | <u>AEC Richland Operations Office</u>

M. R. Schneller |
| 4 | <u>Atlantic Richfield Hanford Company</u>

R. E. Isaacson
A. E. Smith
D. P. Brown
ARHCO File |

No. of
Copies

2	<u>Computer Sciences Corporation</u> E. Z. Block J. L. Spurgeon
1	<u>Donald W. Douglas Laboratories</u> J. Greenborg
3	<u>Douglas United Nuclear</u> G. F. Bailey G. F. Owsley DUN File
6	<u>WADCO (Westinghouse)</u> W. L. Bunch P. L. Hofmann W. W. Little K. R. Merckx J. C. Sonnichsen H. A. Treibs
40	<u>Battelle-Northwest</u> J. L. Carter N. E. Carter D. G. Daniels B. H. Duane A. G. Gibbs C. M. Heeb R. T. Jaske U. P. Jenquin D. A. Kottwitz (15) R. C. Liikala C. W. Lindenmeier J. R. Morrey W. L. Nicholson D. R. Oden, Jr. W. W. Porath D. L. Prezbindowski W. A. Reardon L. C. Schmid G. D. Seybold R. I. Smith Technical Publications Technical Information Files (5)