Title
THE OPTIME SYSTEM FOR FITTING THEORETICAL EXPRESSIONS

Authors
Eberhard, Philippe E.
Koellner, Werner O.

Publication Date
1970-10-01

UCRL-20159
Preprint

THE OPTIME SYSTEM FOR FITTING THEORETICAL EXPRESSIONS

Philippe H. Eberhard and Werner O. Koellner

October 1970

AEC Contract No. W-7405-eng-48

LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

UCRL-20159

DISCLAIMER

CONTENTS

THE OPTIME SYSTEM FOR FITTING THEORETICAL EXPRESSIONS*

Philippe H. Eberhard and Werner O. Koellner
Lawrence Radiation Laboratory
University of California
Berkeley, California 94720

October 1970

ABSTRACT

This paper describes the OPTIME System, designed to help a user fit theoretical expressions to statistical distributions of events. The system makes available several types of estimators relying on maximization techniques to adjust variable parameters. The maximizing processes involve special stepping procedures, some of them using approximations for the second derivative matrix.

The system also provides the possibilities to calculate error matrices, to perform integrations, and to make histograms. Much flexibility within the framework of FORTRAN allows an easy handling of all these features.

The mathematical expressions used and some justification for them are given.

I. INTRODUCTION

1. Goal of OPTIME

The estimation of parameters from statistical data is a common problem. Many techniques are used[1]. When the data are available as a large number of events, computer programs have been written to help solve the problem[2]. The usual approach relies on the use of a maximizing or minimizing program seeking the maximum or minimum of a function given by a subroutine programmed by the user. Parameters can be estimated by programming a likelihood or a χ^2 function[3].

The OPTIME System has also been designed to estimate parameters by maximizing a function w(a), however, it is especially intended to be used with data given as distribution of events to be "fitted" by a mathematical function. Advantage is taken of the structure of the function w(a) to provide efficient stepping procedures either by approximating second derivative matrices or by using the statistical character of the data.

Moreover, the system provides facilities to help solve problems often associated with fitting distributions: means for supplying integrations points and handling of integrals, for computing errors, and for displaying histograms with fitted curves superposed to event distributions. All are available as FORTRAN subroutines to provide much flexibility in their use. The method used to designate which

parameters have to be varied in a given fit while the others are kept constant was chosen to allow for successive fits with different variable parameters with little complication in programming. The user can choose from several functions and stepping procedures.

Fitting of statistical distributions is the only use of OPTIME described here, although all or part of OPTIME can help solve different problems.

2. Definition of the Problem

There are ν experimental events. Each event is given an ordinal number $k(1 \leq k \leq \nu)$. Its specification is represented by N_x quantities $\xi_{k,i}(1 \leq i \leq N_x)$ that we consider as components of a vector $|\xi|_k$. For instance, if event k is an interaction of particles, the N_x components of $|\xi|_k$ may contain the masses, azimuths, dips, lifetimes, etc. of all the particles involved. We suppose that the event k has been given a statistical weight η_k, introduced to correct for eventual inhomogeneities in detection efficiencies. Whenever weighting is not necessary, however, all the weights η_k are assumed to be equal to 1.0 in the formulae to come.

If $|x|$ is a particular value that any $|\xi|_k$ can take, the content of a hypervolume dx centered around $|x|$ (i.e., the sum of the weights of the events whose vector $|\xi|_k$ falls in that hypervolume) is a random variable $\gamma(|x|, dx)$. The contents $\gamma(|x|, dx)$ and $\gamma(|x|', dx)$ of

different hypervolumes are assumed to be independent random variables. The expectation value for the content $\gamma(|x|, dx)$ is

$$< \gamma (|x|, dx) > = \mathcal{Y}(|x|)\, dx. \qquad (I.2.1)$$

$\mathcal{Y}(|x|)$ is the true distribution, the one that represents the average of the distributions of an infinite number of identical experiments and is not known. However, there is a known function $y(|x|), |a|)$ of $|x|$ and of N_a parameters a_j that are components of a vector $|a|$, and there are assumed to be N_a unknown values $\mathcal{A}_j$ of the parameters $|a|$ such that

$$\mathcal{Y}(|x|) = y(|x|, |\mathcal{A}|), \qquad (I.2.2)$$

where the vector $|\mathcal{A}|$ with components $\mathcal{A}_j$ is called the true value of the parameters $|a|$. The purpose of the fit is to find an estimate $|\alpha|$ for $|\mathcal{A}|$, so that $y(|x|, |\alpha|)$ approximates the true distribution $\mathcal{Y}(|x|)$ of (I.2.2), i.e.

$$y(|x|, |\alpha|) \approx \mathcal{Y}(|x|). \qquad (I.2.3)$$

Even after the estimate has been made, Eq. (I.2.3) may not be satisfied either because the fitting routine has been unable to determine the set of parameters $|a|$ that make $y(x, a)$ fit best or because the model is wrong [i.e., $y(|x|, |a|)$ can never equal $\mathcal{Y}(x)$ for any values of the parameters $|a|$]. Regardless of what is considered suspect in the result of the fit, it is useful to check the result by histogramming the distribution $y(|x|, |\alpha|)$ and comparing it to the distribution of the experimental events.

3. General Description of the System

The OPTIME System consists of a set of subroutines that the user calls from a main program of his own. With appropriate calls to those subroutines, he can fit functions $y(|x|, |a|)$ to distributions of events, obtain an estimate of the errors, or perform checks by displaying histograms of the events along with the fitted curve.

The main subroutine MAXIME performs the fit of the mathematical distribution $y(|x|, |a|)$ to the experimental data. The function $y(|x|, |a|)$ is given to MAXIME by a secondary routine HUME written by the user. For a given set of parameters $|a|$ and some set of values of $|x|$, both given in the transfer vector, the secondary routine HUME must return the value of the function $y(|x|, |a|)$ that the user wants to fit. MAXIME is called from the main program with a transfer vector filled with some initial values of the parameters $|a|$ and it returns the same vector filled with values $|\alpha|$ corresponding to the fit. The values $|x|$, in a transfer vector during the call to HUME, are the values $|\xi|_k$ that represent the k^{th} experimental event or, eventually, some integration point $|x|_\ell$ whenever integrations are necessary for the fit.

Actually, the parameters $|a|$ in the transfer vector to MAXIME are contained in a vector $|p|$ longer than $|a|$. In addition to those variable parameters $|a|$ to be adjusted, $|p|$ may contain some or all

the constants that determine the function y (x,a). Another vector, |Loc|, is used for indicating to MAXIME the parameters to be varied. The j^{th} component, $(Loc)_j$, has the location of a_j in |p|:

$$p_{(Loc)_j} = a_j. \qquad (I.3.1)$$

A transfer vector to HUME is also |p|, containing the variable parameters a_j which are changed as the fit progresses, but with all the other components held constant at the values they had in the call to MAXIME. Different sets of components of |p| may be adjusted in different fits with the same HUME routine by just changing the vector |Loc| [4].

Prior to the fit, to introduce the experimental data into the system so that they can be handled by MAXIME and other main subroutines, the routine IMME can be called. IMME will write the vectors $|\xi|_k$ in the proper format for OPTIME. IMME calls two secondary routines. One of them is GETUM1, which reads the tape, where the user has his information stored, and returns after each event. The other one is DOME, which the user has to write, so that, for each event k, a transfer vector D is filled with the values $|\xi|_k$. Later on during the fit that $|\xi|_k$ is a transfer vector in calls to HUME, when y ($|\xi|_k$, |a|) has to be returned. DOME must also supply the weight η_k for the event k if the user wishes it to be different from 1.0.

For most types of fit, integrations must be performed over the

space of $|x|$. Prior to the fit the routine INTIME will introduce integration points into the system in the same manner as the routine IMME does for the experimental events. A secondary routine, GETUM2, is called to generate integration points with some preliminary weights $\Omega_\ell^{(o)}$. From the temporary weights $\Omega_\ell^{(o)}$, the definitive weights of the integration points are computed according to the procedure described in sections III.2 and III.3. The secondary routine DOME is also called and, for each integration point, it must fill a transfer vector D with the quantities $|x|_\ell$ that the user wants to have available in HUME to compute $y\ (|x|_\ell, |a|)$.

A call from the main program to AROME immediately after the corresponding call to MAXIME will print an estimate of the errors corresponding to the fit. This matrix (ERR) is an estimate of the error matrix and is defined as:

$$(\mathrm{ERR}) \approx < (|\alpha| - |\mathcal{A}|)\ \ (|\tilde{\alpha}| - |\tilde{\mathcal{A}}|)>. \qquad \text{(I.3.2)}$$

However, (I.3.2) is valid only if (I.2.3) is true.

Calling FAME with the parameters $|a|$ in the transfer vector followed by a call to COCHIS will cause histograms to be plotted of both the experimental points and the curve $y\ (x, a)$ for the parameters $|a|$ used for the call to FAME. The check of a fit requires calling FAME with $|\alpha|$ (the values for $|a|$ returned by MAXIME for that fit) in the transfer vector, followed by a call to COCHIS.

The primary subroutine, COCHIS, will display the histograms of the quantities that the user has decided to plot when he wrote DOME. There, he calls HIST from his subroutine HISTME with a histogram number and the quantity to be histogrammed in the transfer vector. The calls to HIST from DOME when it deals with the experimental events gives the content of the histograms. The calls to HIST when DOME deals with integration points will be used for plotting the curve. Integration points are, therefore, necessary for the display of the curves.

Integrals of the function y ($|x|$, $|a|$) for any value of $|a|$ can be obtained by calling NORME with the values of $|a|$ in a transfer vector. Of course this is possible only if integration points have been introduced into the system previously.

4. Documentation on OPTIME

The mathematics used in the main routines are discussed in this paper as they apply to the problem described above, i.e., the fit of a random distribution of events. The significance of various options are explained whenever they affect the mathematical development. The object of this paper is to provide a general understanding of what is done inside the main subroutines of the system. Part II deals with the routine MAXIME and Part III with the other major subroutines. Some of this same information can also be found elsewhere [5].

Information concerning the programming is given in a separate

paper[6]. That paper contains a precise description of how to use OPTIME and should be considered the user's manual. It will also provide information useful in hunting for troubles generated by using a computer. Preliminary writeups exist also for this aspect of OPTIME[7].

Additional facilities have been provided for OPTIME. In particular, Jerry Friedman has written a quite efficient routine, SAGE, which generates fake events that simulate interaction and production of particles[8]. This facility can be used to provide integration points for MAXIME and COCHIS.

Much of COCHIS and associated subroutines was borrowed from the KIOWA System. More information about it may be found in Reference[9]. All available documentation mentioned here applies to the "official" version of OPTIME as of November 1970[10].

5. Notation

A symbol inserted between two vertical bars is a vector. Its components are represented by the same symbol affected by an index, e.g., $|x|$ and $|a|$ are vectors, x_i and a_j are their components. N_x is the number of components of $|x|$, N_a is the number of components of $|a|$. In order to lighten the notation, the two vertical bars may sometimes be omitted when we think it will not create confusion. For instance, $y(|x|,|a|)$ may sometimes be written just as $y(x, a)$.

A matrix will be represented by a capital letter, as for instance E, or by a group of capital letters in parentheses, as in (ERR). Its

components will be capital letters affected by two indices, as in $E_{j,j'}$ or $(ERR)_{j,j'}$. A tilde $\sim$ over a matrix or a vector represents the transposed matrix or the transposed vector.

Integrals will also be represented by a capital letter, as in $Y = \int y\, dx$. Whenever integrations are performed numerically, the integrals in the formulae will often represent a summation over the integration points such that

$$Y = \int y\,(x)\, dx = \sum_{\ell} \Omega_{\ell}\, y\,(|x|_{\ell}) \qquad \text{(I.5.1)}$$

where Ω_{ℓ} stands for the weight of the ℓ^{th} integration point and $|x|_{\ell}$ for its coordinates. An integral without limits implies that the integral is over all possible values of $|x|$ that $|\xi|_k$ can take.

As often as possible a random variable whose values depend on the random distribution of the experimental data will be represented by a Greek symbol, as in η_k, $|\xi|_k$, $\xi_{k,i}$. So does the estimation $|\alpha|$ of the parameters $|\mathcal{A}|$, whose value depends on the random distribution of the experimental events. That rule cannot be always respected, however.

Two pointed brackets $< >$ around a random variable will always represent the expectation value for the random variable inside, whether it is represented by a Greek letter or not. Thus $<\gamma(x), dx)>$ is the expectation value for $\gamma\ (x, dx)$ and $< w(a) >$ is the expectation for the function $w(a)$.

Whenever possible we will refer to a true but unknown quantity by a script symbol. For instance $\mathcal{y}(x)$ is the true distribution and $|\mathcal{A}|$ are the true values of the parameters.

Often, to shorten the terminology, we use the word transfer vector to designate the vector that is transferred by way of the calling sequence of a subroutine.

II. THE FITTING ROUTINE

1. The Function to be Maximized, w (a)

The fitting routine, MAXIME, operates by maximizing a function w ($|a|$) of the parameters $|a|$ and of the experimental data. The estimate $|\alpha|$ is the value of $|a|$ that corresponds to the maximum found for w (a).

There are different types of fit that the user may select by setting a flag KTYPE. Those types correspond to different functions w (a) or to different stepping procedures to get to the maximum. Seven types have been programmed so far; they are types 3, 4, 11, 12, 13, 14, and 20.

For types 3 and 4, the function w (a) that gets maximized is

$$w(a) = \sum_{k=1}^{\nu} \eta_k \ln y(\xi_k, a) - Y(a), \qquad \text{(II.1.1)}$$

where

$$Y(a) = \int y(x, a)\, dx. \qquad \text{(II.1.2)}$$

The estimation of $|\alpha|$ by maximizing w (a) of (II.1.1) can be justified by considering the expectation value of the function w:

$$< w\ (a) > = \int \ln y\ (x,\ a)\ \mathcal{Y}\ (x)\ dx - Y\ (a). \qquad \text{(II.1.3)}$$

Therefore,

$$< w\ |a| - w(\mathcal{A}) > = \qquad \text{(II.1.4)}$$
$$= \int \left[\ln \frac{y\ (x,\ a)}{y\ (x,\ \mathcal{A})} - \frac{y\ (x,\ a)}{y\ (x,\ \mathcal{A})} + 1 \right] y\ (x,\ \mathcal{A})\ dx.$$

The function $\ln z - z + 1$ is negative for all values of $z \neq 1$. Therefore (II.1.4) is negative for all values of $|a|$ except the ones that make $y\ (x,\ a) = y\ (x,\ \mathcal{A})$ everywhere. If the function w (a) were equal to its expectation value, its value for $|a| = |\mathcal{A}|$ would be maximum.

For types 11 and 12 the function to be maximized is

$$w\ (a) = \sum_{k=1}^{\nu} \eta_k \ln y\ (\xi_k,\ a). \qquad \text{(II.1.5)}$$

These types should be used only when the integral Y (a) of (II.1.2) is a constant, independent of $|a|$. Under this condition the maximum of w (a) of (II.1.5) occurs for the same values of $|a|$ as the one of w (a) of (II.1.1). Therefore, the estimation by (II.1.5) is valid also if there are $|\mathcal{A}|$ such that (I.2.2) is satisfied. However, the values of $|a|$ that maximize w (a) of (II.1.5) will not be modified if the function y (x, a) is multiplied by a constant, independent of $|a|$.

Therefore, the fit by (II.1.5) concerns the shape of y (x, a) only, independent of its normalization. At the fit, one may expect

$$y\ (x,\ a) \approx c_y\ \mathcal{Y}\ (x), \tag{II.1.6}$$

where c_y does not depend on $|x|$.

For types 13 and 14 the function w is given by

$$w\ (a) = \sum_{k=1}^{\nu} \eta_k\ \ln\ y\ (\xi_k,\ a) - W_t\ \ln\ Y\ (a), \tag{II.1.7}$$

where

$$W_t = \sum_{k=1}^{\nu} \eta_k, \tag{II.1.8}$$

and where Y (a) is again being given by the expression (II.1.2). Obviously, (II.1.7) is obtained from (II.1.5), with y (x, a) replaced by the normalized function $\left[y\ (x,\ a)\right]' = y\ (x,\ a)\ /\ Y\ (a)$ so that the normalization condition of $\left[y\ (x,\ a)\right]'$ is automatic. For types 13 and 14 the condition that the integral Y of y (x, a) be constant is not necessary. The fit with (II.1.7) will also be a fit of the shape only, expected to lead to equation (II.1.6) because (II.1.7) does not depend on a change in normalization.

For type 20 the function w (a) is

$$w\ (a) = \sum_{k} \eta_k\ y\ (\xi_k,\ a) - \tfrac{1}{2}\ Y_2\ (a), \tag{II.1.9}$$

where

$$Y_2\ (a) = \int y^2\ (x,\ a)\ dx. \tag{II.1.10}$$

To demonstrate its validity we consider the expectation value of w (a) of (II.1.9):

$$< w\ (a) > = \int \left[y\ (x,\ a)\ \mathcal{Y}\ (x) - \tfrac{1}{2}\ y^2\ (x,\ a) \right] dx$$

$$= \tfrac{1}{2} \int \mathcal{Y}^2\ (x)\ dx - \tfrac{1}{2} \int \left[\mathcal{Y}\ (x) - y\ (x,\ a) \right]^2 dx. \quad \text{(II.1.11)}$$

(II.1.11) is maximum when y (x, a) = $\mathcal{Y}$ (x) everywhere. Type 20 is a fit of the whole distribution, including normalization, just like type 3 or 4.

2. Basic Stepping Procedure -- the E Matrix

In MAXIME, the parameters are adjusted by successive approximations. At a point defined by some values $|a_o|$ of the parameters, the routine computes a better approximation $|a|$. If $w(a) > w(a_o)$, the new approximation is considered to be a good step toward the maximum and it is accepted. The values of $|a_o|$ are then replaced by the values stored in $|a|$ and a new approximation is computed. If $w(a) < w(a_o)$, the attempted step is not accepted and MAXIME computes a new one according to a given emergency procedure (see Section II.8).

There are different modes of computation of the step. The basic mode is Mode 1. The corresponding approximation is given by:

$$|a| = |a_o| + |v|, \quad \text{(II.2.1)}$$

$$|v| = E^{-1} |u|, \quad \text{(II.2.2)}$$

$$u_j = \frac{\partial w\ (a)}{\partial a_j} \text{ at the point } |a_o|, \quad \text{(II.2.3)}$$

where u_j is the j^{th} component of the

vector $|u|$, gradient of the function w (a) and E is a positive definite covariant matrix computed from the gradient $|h|$ of the function y in the space of $|a|$. More precisely, $|h|$ is defined by

$$h_j(x) = \frac{\partial y(x,a)}{\partial a_j} \quad \text{at } |a| = |a_o|, \tag{II.2.4}$$

and its integral is defined as

$$H_j = \frac{\partial Y(a)}{\partial a_j} = \int h_j(x, a)\, dx. \tag{II.2.5}$$

The precise expression of E depends on the type of fit. Two types that make use of the same function w (a) in the preceding section will differ by their expression for E. For types 3 and 11, E is given by a summation over the experimental points:

$$E = \sum_{k=1}^{\nu} \frac{\eta_k}{y^2(\xi_k, a_o)} |h|(\xi_k) |\tilde{h}|(\xi_k). \tag{II.2.6}$$

For type 13,

$$E = \sum_{k=1}^{\nu} \frac{\eta_k}{y^2(\xi_k, a_o)} |h|(\xi_k) |\tilde{h}|(\xi_k) - \frac{|u|\,|\tilde{H}| + |H|\,|\tilde{u}|}{Y(a_o)} - W_t \frac{|H|\,|\tilde{H}|}{Y^2(a_o)}. \tag{II.2.7}$$

The E matrix of (II.2.6) or of (II.2.7) is positive definite if all the weights η_k are positive. It is equal and opposite to the second derivative matrix (DD) of the corresponding function w (a) for types 3 and 11 when y is a linear function of the parameters $|a|$ and,

for type 13, when y/Y is a linear function of the parameters $|a|$. Therefore, -E should be considered as an approximation for the matrix (DD) for types 3, 11, and 13. This stepping procedure resembles the Newton method for solving an equation [11].

For type 4, 12, 14, and 20, the expression for E involves integration over the space of $|x|$. For type 4,

$$E = \int \frac{|h|(x)\,|\tilde{h}|(x)}{y(x, a_o)}\,dx. \tag{II.2.8}$$

for type 12,

$$E = \frac{W_t}{Y(a_o)} \int \frac{|h|(x)\,|\tilde{h}|(x)}{y(x, a_o)}\,dx. \tag{II.2.9}$$

For type 14 we define

$$E_o = \frac{W_t}{Y(a_o)} \int \frac{|h|(x)\,|\tilde{h}|(x)}{y(x, a_o)}\,dx - \frac{W_t}{Y(a_o)}\,|H|(a_o)\,|\tilde{H}|(a_o), \tag{II.2.10}$$

$$C_{E1} = \frac{|\tilde{u}|\,E_o^{-1}\,|H|}{Y(a_o)}, \tag{II.2.11}$$

$$C_{E2} = \begin{cases} 1 - C_{E1} & \text{if } C_{E1} < \frac{1}{2} \\ \dfrac{1}{4\,C_{E1}} & \text{if } C_{E1} \geq \frac{1}{2} \end{cases}, \tag{II.2.12}$$

and finally

$$E = C_{E2}\,E_o. \tag{II.2.13}$$

For type 20, E is

$$E = \int |h|\ (x)\ |\tilde{h}|\ (x)\ dx. \qquad \text{(II.2.14)}$$

The E matrices of (II.2.8), (II.2.9), (II.2.13), and (II.2.14) are positive definite. If integrations are computed by a numerical approximation, the results are still positive definite as long as the weights of the integration points are all positive. For types 4, 12, and 20, if y is a linear function of $|a|$, we have, for $|u|$ and E computed at the point $|a_o|$,

$$<|u|> \ = E\ (|\mathcal{A}| - |a_o|)\ . \qquad \text{(II.2.15)}$$

Under such assumptions the step computed by (II.2.2) should then bring the parameters near the true values $|\mathcal{A}|$, if $|u|$ is not too different from $<|u|>$.

For type 14, (II.2.15) and its consequences are still true as long as the coefficient $C_{El} < 1/2$. The procedure introduced when $C_{El} \geq 1/2$ is a result of considerations about ensuring the positive definite property and the continuity of E.

Since, for every type, the matrix E is positive definite, there are matrices R such that

$$\tilde{R}\ E\ R = I. \qquad \text{(II.2.16)}$$

MAXIME computes a matrix R that satisfies (II.2.16) with the additional property that every element $R_{j',j}$ of R is zero for $j' > j$. R is a "triangular" matrix. The step (II.2.2) can be expressed as

$$|v| = R \tilde{R} |u|. \qquad \text{(II.2.17)}$$

The positive definite property of E ensures that $|v|$ defines a direction in which w (a) can be increased.

3. Derivatives

The user may choose not to program all derivatives h_j of y in the secondary routine HUME. Then, MAXIME gives increments to the parameters $|a_o|$ and calls HUME. The variations of y corresponding to the increments are linear combinations of the h_j's.

There are N_a linearly independent vectors $|DA|_j$ used to compute derivatives numerically. The j'^{th} component $(DA)_{j',j}$ of the vector $|DA|_j$ is not zero only if $j' \leqslant j$. Therefore, the matrix (DA) whose elements are $DA_{j',j}$ is triangular, but not diagonal in general. We define

$$|h|^* = (\tilde{DA}) |h|. \qquad \text{(II.3.1)}$$

If h^*_j is computed numerically, its computation is performed in general according to the following expression:

$$h^*_j (x) = \frac{y (x, |a_o| + |DA|_j) - y (x, |a_o| - |DA|_j)}{2}. \qquad \text{(II.3.2)}$$

If the user considers that increments in one direction only are good enough for the computation of some derivatives, he can give the corresponding parameters a_j a flag instructing MAXIME to call HUME with one sign for the increment only; then h^*_j is computed according to

$$h^*_j(x) = y(x, |a_o| + |DA|_j) - y(x, |a_o|). \qquad \text{(II.3.3)}$$

In some cases involving the definition of an unallowed region for the fit, the use of one-sided increments may be forced upon MAXIME (see Section II.9). Then, depending on which side is forbidden, (II.3.3) will be used or

$$h^*_j = y(x, |a_o|) - y(x, |a_o| - |DA|_j). \qquad \text{(II.3.4)}$$

The increments (DA) are computed in the secondary routine DAME from the matrix R computed at the previous step and a diagonal matrix (WGT) constant during the fit. The matrix (DA) is the product of the matrices R and (WGT):

$$(DA) = R\,(WGT). \qquad \text{(II.3.5)}$$

The diagonal elements of (WGT), called weights for derivative increments, are given by the user in his call to MAXIME, in the transfer vector |Wgt|. They may be set all equal to 1.0 in general. This procedure has been introduced to avoid too much correlation effect between parameters in the numerical computation of the derivatives. If the j^{th} derivative, however, is programmed in HUME, the j^{th} increment $|DA|_j$ has 1.0 for its j^{th} component and 0 everywhere else. For that parameter then, $h^*_j = h_j$.

Even if the j^{th} derivative is computed numerically, the increment $|DA|_j$ may differ from the expression derived from (II.3.5) by an overall factor 1/2, 1/4, or 1/8, in some cases involving the unallowed

region (as will be seen in Section II.9).

At step 0 there are no previous steps, therefore no R is available. The user has to provide a vector |Uda| in his call to MAXIME. Then the elements of |Uda| go into a diagonal matrix (UDA) and the increments for derivative computation at step 0 will be

$$(DA) = (UDA). \tag{II.3.6}$$

4. Computations for Mode 1

At a given iteration $|a_o|$, the function w, the vector $|u|^*$, and the matrix E^* are computed; where $|u|^*$ and E^* are defined, like |u| and E, by (II.2.3) and (II.2.6) through (II.2.14) but in the system of axes formed by the vectors $|Da|_j$:

$$|u|^* = (\tilde{DA})\ |u|, \tag{II.4.1}$$

$$E^* = (\tilde{DA})\ E\ (DA). \tag{II.4.2}$$

For each experimental point and for each integration point, the function HUME is called with $|a_o|$ in a transfer vector. From the returned value $y\ (x, |a_o|)$, the factors Wfac, Ufac and Matfac are computed. Table I shows their expressions for different types, with an index r or i to differentiate the expression to be used: r when dealing with the experimental events and i for the integration points. From those factors and from the vector $|h|^*(x)$, whose computation is described in the preceding section, MAXIME computes

$$w_r = \sum_{k=1}^{\nu} (\text{Wfac})_r \ , \qquad \text{(II.4.3)}$$

$$w_i = \int (\text{Wfac})_i \ dx \ , \qquad \text{(II.4.4)}$$

$$|u|_r^* = \sum_{k=1}^{\nu} (\text{Ufac})_r \ |h|^* \ (\xi_k), \qquad \text{(II.4.5)}$$

$$|u|_i^* = \int (\text{Ufac})_i \ |h|^* \ (x) \ dx. \qquad \text{(II.4.6)}$$

When one of those summations is not performed because it is not needed, the corresponding factor is indicated by a dash in Table I. For types 3, 11, and 13, the following additional summation is performed:

$$(\text{MAT}) = \sum_{k=1}^{\nu} (\text{Matfac})_r \ |h|^* \ (\xi_k) \ |\tilde{h}|^* \ (\xi_k). \qquad \text{(II.4.7)}$$

For types 4, 12, 14 and 20, the following integrations are performed instead:

$$(\text{MAT}) = \int (\text{Matfac})_i \ |h|^* \ (x) \ |\tilde{h}|^* \ (x) \ dx. \qquad \text{(II.4.8)}$$

From the result of these summations, the function w, the vector $|u|^*$, and the matrix E^* are computed according to their expression shown in Table I. The triangular matrices R^* and R, then $|v_1|^{**}$ and $|v_1|$, are computed:

$$\tilde{R}^{*} E^{*} R^{*} = I, \qquad \text{(II.4.9)}$$

with

$$R^{*}_{j',j} = 0 \text{ for } j' > j$$

$$R = (DA)\, R^{*}, \qquad \text{(II.4.10)}$$

$$|v_1|^{**} = \tilde{R}^{*} |u|^{*}, \qquad \text{(II.4.11)}$$

$$|v_1| = R \; |v_1|^{**}. \qquad \text{(II.4.12)}$$

If Mode 1 is used, then

$$|v| = |v_1| \qquad \text{(II.4.13)}$$

5. Computations for Mode 2 -- State of the Fit

When some or all derivatives are computed numerically, MAXIME may use the values y ($|x|$, $|a_o| \pm |DA|_j$) to compute the function $wp_j = w(|a_o| + |DA|_j)$ or $wm_j = w(|a_o| - |DA|_j)$ or both. The conditions that make MAXIME perform the computation of wp_j or wm_j are: (1) the j^{th} derivative is computed numerically, (2) the increment $+|DA|_j$ in the case of wp_j and $-|DA|_j$ in the case of wm_j is used for that numerical computation and does not drive the parameters into the unallowed region.

MAXIME will always remember the largest value of the function w found so far. As soon as new values of wp, wm, or w are computed, they are compared with the old best value. If one of the new values is found to be larger, that value will replace the old best and the

parameters that correspond to the new best value are stored in a vector $|b|$ called best values of the parameters.

Whenever a step $|v|$ turns out to be so small that

$$\tfrac{1}{2} |\tilde{u}||v| < w (|b|) - w (|a|), \qquad \text{(II.5.1)}$$

MAXIME uses $|b|$ as the new iteration point instead of $|a_o| + |v|$. This mode of stepping is called Mode 2. It can be justified by considering that the expected improvement of w, when taking step $|v|$, is $\tfrac{1}{2} |\tilde{u}| |v|$.

If, at a given step, there is a parameter a_j for which MAXIME has computed wp_j and wm_j and such that

$$w (a_o) < \tfrac{1}{2} (wp_j + wm_j), \qquad \text{(II.5.2)}$$

the fit is considered to be in State 1. This means that the current point is so far from the maximum that the second derivative matrix (DD) is not even negative definite. When the State is 1, it is not valid to approximate w by a quadratic function. It should be remarked that in State 1, Eq. (II.5.2) for any one of the parameters implies that there is a value wp or wm superior to $w (a_o)$.

When the condition for State 1 is not fulfilled, but a value of wp or wm computed at the current step or at any step before is greater than the value $w (a_o)$ of the current step, the state is called State 2.

State 3 is defined as the state where $w (a_o)$ is the best value found so far. Then $|b| = |a_o|$. State = 3 is one of the conditions

that the parameters $|a_o|$ have to fulfill so that the fit can stop at $|a_o|$ and be called a good fit.

Mode 2 can be used only in State 1 or 2.

6. Corrections to the Step -- Modes 3 and 4

To avoid having the parameters bounce back and forth when overshooting the maximum, the step computed with Mode 1 (II.4.12) may be submitted to corrections if there is a sympton of overshooting. That correction is actually an interpolation of the values determined at two successive steps.

We refer to the old values of $|a|$ as $|a_{oo}|$. They are the values of $|a|$ at the previous iteration, just before the last accepted step. We define

$$|v_o| = |a_o| - |a_{oo}| , \qquad \text{(II.6.1)}$$

$$(V_o tev_o) = |\tilde{v}_o| \; E \; |v_o| , \qquad \text{(II.6.2)}$$

$$(Utv_o) = |\hat{u}| \, |v_o| , \qquad \text{(II.6.3)}$$

$$(U_o t \; v_o) = |\tilde{u}|_o \, |v_o| , \qquad \text{(II.6.4)}$$

where $|u_o|$ is the gradient of w at $|a_{oo}|$ and $|u|$ its gradient at $|a_o|$. If (Utv_o) is negative, there must be a higher value of w (a) along the segment $|a_o|$ to $|a_{oo}|$; this is the sympton of overshooting.

Mode 3 and Mode 4 correspond to two different estimations of where the maximum of w (a) is on the segment $|a_o|$ to $|a_{oo}|$ in case of

overshooting. We define t such that $|a_o| - t|v_o|$ is the location of that maximum. Mode 3 is a computation of t, taking into account the derivatives at both ends of the segment and the difference $w(a_o) - w(a_{oo})$ and assuming w to be a cubic function of $|a|$. Mode 4 uses the derivatives at both ends only and assumes w quadratic. We define t', t'', and t_m:

$$t' = \frac{-(Utv_o)}{-(Utv_o) + (U_o tv_o)} , \qquad \text{(II.6.5)}$$

$$t'' = \frac{-\frac{1}{2}(Utv_o)}{w(a_o) - w(a_{oo}) - (Utv_o)} , \qquad \text{(II.6.6)}$$

$$\frac{1}{t_m} = \frac{3}{2t''} - \frac{1}{t'} + \left[\left(\frac{1}{t'} - \frac{1}{2t''}\right)^2 - \left(\frac{1}{t'} - \frac{1}{t''}\right)\left(\frac{2}{t''} - 3\right)\right]^{\frac{1}{2}} \qquad \text{(II.6.7)}$$

For Mode 3, $t=t_m$; for Mode 4, $t=t'$. However, in either mode, if t turns out to be < 0.1 by these formulae, it will be made 0.1 to prevent its being ridiculously small. It is reduced to 0.9 if it is larger than 0.9. Once t and v_1 of (II.4.12) have been determined, the step according to Mode 3 or Mode 4, is determined as

$$|v| = (1-t)|v_1| - \left[\frac{-(Utv_o)}{(V_o tev_o)}(1-t) - t\right]|v_o| . \qquad \text{(II.6.8)}$$

Using the E matrix as a metric, this step means that the part of $|v_1|$ that is orthogonal to $|v_o|$ is reduced by the factor (1-t) and its longitudinal part adjusted to be equal to $-t|v_o|$.

7. Choice of the Mode -- Different Cases

At each iteration the routine DECIME decides either to stop fitting or go on stepping. In the latter case, DECIME decides which mode to use.

Once the vector $|v_1|^{**}$ of (II.4.11) is computed, the first test is for adjusted parameters. There is a constant (Chilim), which presently is set arbitrarily to 10^{-2}. Parameter a_j is called adjusted if

$$(v_1^{**})_j^2 < (\text{Chilim}). \qquad \text{(II.7.1)}$$

If w is a quadratic function, with -E its second derivative matrix, and if R and $|v_1|^{**}$ are as defined above, then the difference between the value of w (a_o) and the maximum value attainable when varying only the j first parameters is

$$(\Delta w) = \frac{1}{2} \sum_{j'=1}^{j} (v_o^{**})^2_{j'} . \qquad \text{(II.7.2)}$$

The j^{th} parameter is therefore called adjusted if, when you vary it together with the j-1 preceding parameters or when you vary only the j-1 preceding parameters, it is not estimated to make a difference in the value of the function w greater than 0.5×10^{-2}.

If all parameters are adjusted and the State is 3, [i.e., w (a_o) larger than any other known value of w], the flag KEND is set equal to 1, meaning that the fit is good enough at $|a_o|$. An extra step is

going to be computed, however, to be added to $|a_o|$ for the parameters in the transfer vector to MAXIME when it returns. But the stepping process stops here, and a new value of w (a) for the last step is not even calculated.

Whether all parameters are adjusted or not, the next proposed step depends on different circumstances. Case 1 exists if no step has been accepted yet, or if the previous step or the present step is State 1, or if the mode used at the previous iteration was Mode 2, or if the new E matrix is so small that $(V_o tev_o)$ of (II.6.2) turns out to be smaller than the constant (Chilim). This means that the function w is still too wild to permit a meaningful interpolation like Mode 3 or 4. In Case 1, Mode 1 is used, unless the State is 1 or 2 and (II.5.1) is satisfied, then Mode 2 is used instead. Case 2 is defined by none of the conditions that satisfy Case 1 and by

$$(Utv_o) > -\left[(Chilim)\,(V_o tev_o)\right]^{\frac{1}{2}}. \qquad \text{(II.7.3)}$$

It means that the function has a good behavior and that there has been no overshooting. Either the derivative of w is negative in the direction or small enough that it cannot be increased by more than 1/2 of (Chilim) in that direction. Case 2 is treated like Case 1, i.e., Mode 1 is used unless (II.5.1) is true and forces Mode 2 to be used instead.

Cases 3 and 4 are the cases of overshooting defined by none of the conditions that satisfy Case 1 and by (II.7.3) not being satisfied. Then

t' and t'' are computed according to (II.6.5) and (II.6.6). Case 3 is the case where $t'' > t'$, then t is computed according to Mode 3. If $t'' \leqslant t'$ we have Case 4 and t is computed according to Mode 4. A subsequent test

$$t < \frac{-(Utv_o)}{(V_o tev_o)} \qquad \text{(II.7.4)}$$

has to be satisfied so that Mode 3 in Case 3 and Mode 4 in Case 4 can be used. If (II.7.4) is not satisfied, Mode 1 is preferred. In any case, if the step $|v|$ satisfies (II.5.1), Mode 2 is used instead.

There is a Case 5 defined, regardless of any other circumstances, by all parameters being adjusted, i.e., satisfying (II.7.1), but with the state not equal to 3. That case also triggers the use of Mode 1 unless (II.5.1) is satisfied, forcing use of Mode 2 instead.

8. Emergencies -- Modes 5 and 6

When a step $|v|$ has been computed according to Modes 1, 3, or 4 and when

$$w(|a_o| + |v|) < w(|a_o|) \qquad \text{(II.8.1)}$$

there is a case of emergency. The secondary routine CRIME adjusts the size of the step by using Mode 5. Computation of t is such that

$$t = \frac{\frac{1}{2}|\tilde{u}||v|}{w(a_o) - w(a_o + v) - |\tilde{u}||v|} \cdot \qquad \text{(II.8.2)}$$

When t is given by (II.8.2), $|a_o| + t|v|$ is the location of the maximum of w along the segment $|v|$ if w is quadratic; t is set to 0.1

if (II.8.2) gives it a value less than 0.1 to prevent its being ridiculously small. In any case Mode 5 consists of replacing $|v|$ by $t\,|v|$ as the new step to be attempted.

Of course the new step computed by Mode 5 may also satisfy (II.8.1). If it does and if it satisfies (II.5.1), Mode 2 is used then. If not, CRIME tests for the condition

$$|\tilde{v}|^* \, |v|^* \leqslant 1, \tag{II.8.3}$$

where

$$|v|^* = (DA)^{-1} \, |v|. \tag{II.8.4}$$

If (II.8.3) is satisfied and some derivatives are computed numerically, CRIME considers that the step $|v|$ is smaller than the increments to compute the derivatives. The numerical estimate of the derivative is considered suspect and the step is then completely recalculated with all the increments $|DA|_j$ cut by one-half. This is called Mode 6.

If (II.8.3) is not satisfied, Mode 5 is used again to cut the step some more. Mode 5 and Mode 6 can be used several times in the same iteration in some pathological cases.

Earlier in the step calculation, another kind of emergency may occur. The solution R^* of (II.4.9) does not exist if E^* has zero eigenvalues. This fact is recognized by the secondary routine DIAME which uses a Gram-Schmidt orthonormalization method [12] to compute

the triangular matrix R^* that satisfies (II.4.9). If a diagonal element $E^*_{j,j}$ is zero or near zero ($< 10^{-200}$) the parameter j is called insensitive, because that element corresponds to a sum of squares of a derivative element and it is zero only if the function is not sensitive to the parameter in each of the terms of the sum. If $E^*_{j,j}$ is not zero or near zero, but if the singularity appears when the j^{th} parameter is added to the j-1 previous parameters, the j^{th} parameter is called correlated.

Whether the j^{th} parameter is insensitive or correlated, DIAME makes $R_{j',j} = 0$ for $j' \neq j$ and $R_{jj} = 1$. If all parameters are insensitive, MAXIME exits with an emergency comment and the flag KEND is set to 8.

There are other cases of emergencies that may interrupt the fit. A comment gets printed to indicate the cause of the emergency.

9. The Unallowed Region -- Derivative Indices

When the user wants to restrict the possible values of the parameters to a domain inside of some boundaries, he should express each of his conditions by a relation of the type

$$g_m(a) \geqslant 0. \qquad \text{(II.9.1)}$$

Then, $g_m(a) < 0$ when the m^{th} condition is not satisfied. The user should write a routine ALARME that returns the values $|g|(a)$ in a transfer vector when it is called with the values $|a|$ in another

transfer vector, and he should indicate the number N_g of conditions in the calling sequence of MAXIME. Then MAXIME will refrain from stepping into the unallowed region.

In the unallowed region there might be values of $|a|$ for which the analytic expression of $y(x, a)$ programmed in HUME would be mathematical nonsense (square root of a negative number, poles, etc.). For such values of $|a|$, ALARME should return the flag Talarme = true in addition to some negative boundary functions. A value of $|a|$ corresponding to Talarme = true will not be used for the numerical computation of the derivatives, while it is used if Talarme is false regardless of the sign of the boundary function. The secondary routine DAME tests if the increment $|DA|_j$ computed according to (II.3.5) would generate values of $|a|$ in the region where Talarme is true. If an increment $|DA|_j$ does it, DAME will cut it by 1/2 and test again. After three unsuccessful cutbacks by a factor 1/2 each time, DAME tries a one-sided increment only, by setting the flag Tdp_j or Tdm_j false if increment $+|DA|_j$ or $-|DA|_j$ is the cause of the trouble. Then those flags will be recognized when h^*_j has to be computed. In such a case (II.3.3) or (II.3.4) will be used instead of (II.3.2).

If both increments $\pm|DA|_j$ still make Talarme = true, Tdp_j and Tdm_j are both set false and the j^{th} derivative won't be computed at that step. Then h^*_j will be equal to 0, therefore a_j will be considered as insensitive. However, at the following step, DAME will try

to give it some increments again so a derivative can be computed. Tdp_j and Tdm_j are false also if $Wgt_j = 0$; that is, if h_j is programmed in HUME. If none of the N_a derivatives can be computed numerically and none are programmed, MAXIME quits with an error comment.

When an increment $+|DA|_j$ or $-|DA|_j$ is accepted for numerical computation of the derivative, the flag Twp_j or Twm_j is set true if that increment leads to values of $|a|$ such that the routine ALARME returns with all the boundary functions $|g|$ positive. If not, the corresponding flag Twp_j or Twm_j is set false and the function wp_j or wm_j will not be computed. That procedure prevents Mode 2 from ever stepping into the unallowed region.

10. Fits Against Boundary -- Modes 7, 8, and 9

CORME is the routine in charge of correcting steps into the unallowed region. It first tries to cut the step so that it fits inside the allowed region, but if the parameters are already at the limit it tries to maximize w (a) while staying against the boundary.

CORME makes use of a subroutine LIME several times. Given two vectors $|a|'$ and $|\delta a|$ such that the point $|a|'$ is inside the boundaries but $|a|' + |\delta a|$ is outside of them, LIME finds a point $|a|' + t_1 |\delta a|$ on the segment $|\delta a|$ just at the boundary. Actually, LIME finds two coefficients t_1 and t_2 such that $|a|' + t_1 |\delta a|$ is inside the boundary,

$|a|' + t_2 |\delta a|$ is outside of it, and such that $t_2 - t_1$ is very small:

$$t_2 - t_1 < \frac{Chilim}{|\tilde{u}| |\delta a|} \qquad (II.10.1)$$

If LIME is used MAXIME will consider the next approximation

$$|a| = |a|' + t_1 |\delta a| . \qquad (II.10.2)$$

Whatever mode has been used to compute a step $|v|$ according to the prescription of Sections (II.7) or (II.8), a check is made to see if $|a_o| + |v|$ is in the unallowed region. If it is, CORME will first use the routine LIME to cut it so that the next approximation $|a_o| + t_1|v|$ is just inside the boundary. That stepping procedure is called Mode 7; it is illustrated in Figure 1.

If Mode 7 does not result in a step of appreciable size (i.e., if LIME finds a solution of (II.10.1) with $t_1 = 0$, then CORME will try a step $|v_1|$ according Mode 1 unless such a step has already been tried before from the same iteration point. If Mode 1 has been tried already or if $|v_1|$ still leads into the unallowed region, CORME then embarks upon a search for the largest value of w (a) against the boundary (Modes 8 and 9).

A list is made of the relevant boundary functions $g_m(a)$ that are not satisfied at the point $|a_o| + |v_1|$. Those functions are placed in the vector $|grel|$ of length N_{rel}. The derivatives of $|grel|$ with respect to $|a|$ form a matrix S with $N_{rel} \times N_a$ elements:

$$S_{n,j} = \frac{\partial (grel)_n}{\partial a_j} . \qquad (II.10.3)$$

Additionally, at the point $|a_o|$, CORME computes $|grel|(a_o)$ and the $N_{rel} \times N_a$ matrix,

$$S^{**} = SR, \qquad (II.10.4)$$

where R is defined by (II.4.10).

The derivatives S^{**} are always computed numerically. The increment $|R|_j$ used to compute $S^{**}_{n,j}$ is the vector that forms the j^{th} column of the triangular matrix R. This increment is always used in both directions, and $S^{**}_{n,j}$ is defined as:

$$S^{**}_{n,j} = \tfrac{1}{2} \left[grel_n(|a_o| + |R|_j) - grel_n(|a_o| - |R|_j) \right] . \qquad (II.10.5)$$

Then CORME computes

$$Q = S^{**} \tilde{S}^{**} , \qquad (II.10.6)$$

$$|Lambda| = - Q^{-1} (S^{**} R^{-1} |v_1| + |grel|), \qquad (II.10.7)$$

$$v_2 = R \tilde{S}^{**} |Lambda| . \qquad (II.10.8)$$

If $|grel|$ were composed of linear functions of $|a|$, if $w(a)$ were a quadratic function of $|a|$, and if E were its second derivative matrix, the point

$$|a_c| = |a_o| + |v_1| + |v_2| \qquad (II.10.9)$$

would be the location of the maximum of w (a) with the constraints

$$|grel| = 0. \qquad (II.10.10)$$

However, our constraints are expressed by an inequality of the type (II.10.1) not by an equality of type (II.10.10). Among the relevant inequalities |grel| violated at $|a_o| + |v_1|$ there might be some that would become satisfied as soon as the others are satisfied. To take that phenomenon into account, CORME inspects the vector |Lambda| for negative components. If $(\text{Lambda})_n$ is negative, the n^{th} boundary condition is dropped from the list of relevant conditions. Then (II.10.6) and (II.10.7) are recomputed and |Lambda| reinspected till all the components of |Lambda| are positive. Only then (II.10.8) is calculated.

A trial step is made, equal to

$$|v_3| = |v_1| + C_{v2} |v_2| , \qquad \text{(II.10.11)}$$

where

$$C_{v2} = 0.9 - 0.1 \frac{|\tilde{u}| \, |v_1|}{|\tilde{u}| \, |v_2|} . \qquad \text{(II.10.12)}$$

C_{v2} from (II.10.12) is such that $|\tilde{u}| \, |v_3|$ is positive and therefore defines a direction of improvement for w(a). It is near 1.0, so $|v_3|$ takes the parameters close to the point $|a_c|$ of (II.10.9). See Figure 2. It is greater than 1.0 in order to be on the safe side of the inequality if there are small non linearities in the constraints.

If the point $|a_o| + |v_3|$ is still in the allowed region, the boundary is approached again on the segment $|a_o| + |v_3|$ to $|a_o| + |v_1|$, using LIME. The resulting point is the point proposed for the next iteration.

This mode of computing a step is called Mode 8.

If the point $|a_o| + |v_3|$ is still in the unallowed region, it means that either |grel| is not linear enough or that a new boundary function not in the list of relevant boundary functions has become relevant now. In such a case, LIME is used to cut the step so that $|a_o| + t_1|v_3|$ falls inside the allowed region. This mode of step computation is called Mode 9.

If Mode 8 produces such a small step that $|\tilde{u}||v| <$ Chilim, the routine MAXIME declares $|a_o|$ a good fit against the boundary. If Mode 9 produces too small a step, the case is too complicated for MAXIME to handle and the iteration process is stopped. The comment FIT MAY BE GOOD is printed. It is necessary for the user to understand the mathematical complexity of the situation to appreciate the value of his fit.

III. OTHER ROUTINES

1. The Error Routine AROME

AROME computes an estimation of (ERR), the error matrix of (I.3.2), for the parameters fitted in the previous call to MAXIME. AROME uses some quantities computed by MAXIME in its last step. It uses the matrices R, R^* of (II.4.10) and (II.4.9), and sometimes the vector $|u_i|^*$ of (II.4.6) and w_i of (II.4.4). Therefore AROME should be called just after MAXIME, at least before those quantities get over-

written in the computer.

For AROME, the values $|a_o|$ are those for which $|u_i|^*$, R, and R^* have been computed, the ones for which all parameters have been declared adjusted before the last computed step $|v|$ is added. MAXIME returns $|a_o|+|v|$ as the fitted parameters, but AROME uses $|a_o|$ for its own computation unless, of course, special intervention of the user occurred between his call to MAXIME and his call to AROME.

For all types, AROME performs summations over the experimental events. For types 3, 4, 11, and 12, the matrix $(U2)^*$ is defined as

$$(U2)^* = \sum_{k=1}^{\nu} \frac{\eta_k^2}{y^2(\xi_k, a_o)} |h|^*(\xi_k)\ \tilde{|h|}^*(\xi_k), \qquad \text{(III.1.1)}$$

where $|h|^*(\xi_k)$ is the gradient of $y(\xi_k,a)$ in the space of $|a|$ at $|a_o|$, as in (II.3.1).

For types 13 and 14,

$$(U2)^* = \sum_{k=1}^{\nu} \frac{\eta_k^2}{y^2(\xi_k,a_o)} |h|^*(\xi_k)\tilde{|h|}^*(\xi_k) - \frac{|u_i|^*\,\tilde{|Vec|}+|Vec|\,\tilde{|u_i|}^*}{w_i} + \frac{Wt2}{W_i^2}\,|u_i|^*\,\tilde{|u_i|}^* \qquad \text{(III.1.2)}$$

where

$$|Vec| = \sum_{k=1}^{\nu} \frac{\eta_k^2}{y(\xi_k,a_o)} |h|^*(\xi_k)\ , \qquad \text{(III.1.3)}$$

and where

$$(Wt2) = \sum_{k=1}^{\nu} \eta_k^2 \,. \qquad \text{(III.1.4)}$$

$|u_i|^*$ and w_i of (II.4.6) and (II.4.4) still represent the integral $|H|^*$ of $|h|^*(x)$ and $Y(a_o)$ of $y(x,a_o)$.

For type 20,

$$(U2)^* = \sum_{k=1}^{\nu} \eta_k^2 \, |h|^* (\xi_k) \, \tilde{|h|}^* (\xi_k). \qquad \text{(III.1.5)}$$

Then, for all types, the error matrix (ERR) of (I.3.1) is given by

$$(ERR) = R \, \tilde{R}^* \, (U2)^* \, R^* \, \tilde{R} \,. \qquad \text{(III.1.6)}$$

These formulae can be justified if the values $|\alpha|$ of $|a|$ that maximize $w(a)$ and the true values $|\mathcal{A}|$ are not too far apart. Then, $|\alpha|$ could be reached in one step in the maximization process, using Mode 1, if one would happen to start the iterations in $|\mathcal{A}|$. From (II.2.2) or (II.4.13), using $|u|^*$ computed in $|\mathcal{A}|$, we have the relation

$$|v| = |\alpha| - |\mathcal{A}| = E^{-1} \, |u| = R \, \tilde{R}^* \, |u|^* \qquad \text{(III.1.7)}$$

For $|u|^*$ computed in $|\mathcal{A}|$, $< |u|^* > = 0$ but the random variables $|u|^*$ are not zero in general. There is a correlation matrix $(U2)^*$:

$$(U2)^* = < |u|^* \tilde{|u|}^* > \qquad \text{(III.1.8)}$$

If $(U2)^*$ is an estimation for $< |u|^* |u|^* >$ then (III.1.7) justifies (III.1.6), but we still have to justify (III.1.1), (III.1.2), and (III.1.5).

To justify (III.1.1) for types 3, 4, 11, and 12, it should be noticed that the random part of $|u|^*$ is $|u_r|^*$ of (II.4.5); $|u_r|^*$ is a summation over the contents $\gamma(x, dx)$ of different hypervolumes dx (see Section I.2):

$$|u_r|^* = \int \gamma(x, dx) \frac{1}{y(x, \mathcal{A})} |h|^* (x) \qquad \text{(III.1.9)}$$

The different hypervolumes have uncorrelated contents; therefore, the correlation matrix for $|u_r|^*$ is the sum of the contribution of each hypervolume.

$$(U2)^* = <|u_r|^* |\tilde{u_r}|^* > - <|u_r|^* > <|\tilde{u_r}|^* > \qquad \text{(III.1.10)}$$
$$= \int \left[< \gamma^2(x,dx) > - < \gamma(x,dx) >^2 \right] \frac{1}{y^2(x, \mathcal{A})} |h|^* (x) \; |\tilde{h}|^* (x).$$

However $< \gamma(x, dx) >^2 = y^2(x)\, dx^2 \to 0$, as $(dx)^2$ when $dx \to 0$. Therefore

$$(U2)^* = \int < \gamma^2(x, dx) > \frac{1}{y^2(x, \mathcal{A})} |h|^* (x) \; |\tilde{h}|^* (x)$$
$$\approx \int \gamma^2 (x, dx) \frac{1}{y^2(x, \mathcal{A})} |h|^* (x) \; |\tilde{h}|^* (x)$$
$$= \sum_k \frac{\eta_k^2}{y^2(\xi_k, \mathcal{A})} |h|^* (\xi_k) \; |\tilde{h}|^* (\xi_k). \qquad \text{(III.1.11)}$$

(III.1.1) is an approximation for (III.1.11), where the values of y (x, a) and $|h|^*$ are taken for $|a| = |\alpha|$ instead of $|a| = |\mathcal{A}|$.

The justification of (III.1.2) for types 13 and 14 is straight-

forward once (III.1.1) is justified for types 11 and 12, and once $y(x, \alpha)$ and $\left[1/y(x, \alpha)\right] |h|^*(x)$ are replaced by

$$y'(x, \alpha) = \frac{y(x, \alpha)}{Y(x, \alpha)}, \text{ and } \frac{1}{y'(x, \alpha)} |h'|^*(x) = \frac{1}{y(x,\alpha)} |h|^*(x) \frac{1}{Y(\alpha)} |H|^*.$$

(III.1.5) for type 20 can be justified just as (III.1.1) can be for types 3 and 4 by replacing the factor $\left[1/y(x,a)\right]$ by 1.0 in (III.1.9) and developing the same arguments as above.

Warning! If integrals are performed numerically, the errors printed by AROME and described here do not include the uncertainties associated with the evaluation of integrals.

2. Integrations in OPTIME -- NORME

Fits with type 11 do not require any integration. For the other types, however, the fits in MAXIME involve expressions with integrals over the space of $|x|$. The structure of the program is such that each integral of the type

$$F = \int f(x)\, dx \qquad \text{(III.2.1)}$$

is actually computed as a summation over Nint integration points. Let Ω_ℓ be the weight and $|x|_\ell$ the coordinates of the ℓ^{th} integration point. For F of (III.2.1), the program uses the quantity Φ instead:

$$\Phi = \sum_\ell \Omega_\ell f(x_\ell). \qquad \text{(III.2.2)}$$

For types 3 and 13, exact expressions can be used for integrals. The user has to make Nint = 1, give the single integration point the

weight $\Omega_\ell = 1.0$, and, when HUME is called with that integration point, return for y (x_ℓ, a) the value of the integral Y(a). If some derivatives $h_j(x)$ are not computed numerically but are programmed in HUME, the user should also return for $h_j(x_\ell)$, the value of the integral H_j. Then all integrals are exact in MAXIME because, for types 3 and 13, the only integrals used are the integrals of y and $|h|^*$. Of course, this technique of using a single point of integration may also be useful with type 3 or 13 if integrals cannot be programmed exactly for y (x, a) but only approximated with sufficient accuracy.

For types 4, 12, and 14, the matrix E^* also is determined by using integrations, but its expression (II.4.8 and Table I) is unlikely to be a linear combination of simple "reference functions" because y (x,a) appears in a denominator. Therefore, there is need for many integration points to get a reasonable estimate of E^*. However, E^* is used only to compute the step. For type 12 the location of the maximum of w (a) does not depend on any integrated value. For types 4 and 14 that location depends on the integrated value of y and |h| only. If y and |h| are always linear combinations of some functions $(fr)_n$ (x) whose integrals $(Fr)_n$ are known, the integrals Y and |H| will be computed exactly if the $(fr)_n(x)$ are used as reference functions in INTIME (see Section III.3). The use of reference functions can also reduce the error on all integrals anyway.

For type 20, if y^2, y, $|h|$, and $|h|\,|\tilde{h}|$ are always the linear combination of the functions $(fr)_n(x)$ used as reference functions in INTIME, all integrals performed in MAXIME will be exact. If not, errors on integrations are just reduced by the use of reference functions.

To obtain the integral $Y(a)$ of the function $y(x,a)$, the routine NORME can be called with the parameters in a transfer vector. NORME will perform summations of the type given by (III.2.2). The result will be exact for all linear combinations of the reference functions, i.e.,

$$Y(a) = \int y(x,a)\,dx \simeq \sum_{\ell=1}^{Nint} \Omega_\ell\, y\,(x_\ell,a) \qquad \text{(III.2.3)}$$

if $|a|$ happens to be such that for all $|x|$'s

$$y(x,a) = \sum_n c_n (fr)_n(x). \qquad \text{(III.2.4)}$$

If (III.2.4) is not satisfied, the output of NORME by (III.2.3) is still an estimation for the integral Y.

It should be noticed that NORME, just like MAXIME, uses the vector $|p|$ of (I.3.1) containing variable and constant parameters and not $|a|$. Therefore, to get integrals of y when the constant parameters are changed, one just changes the constant parameter in $|p|$ before calling NORME.

Sometimes not only the integral $Y(a)$ is wanted but also its

derivatives H(a) with respect to |a|, as defined by (II.2.5). For this purpose, the routine ENORME may be called, which also performs summations of the type (III.2.2) by using for $f(x_\ell)$ the values $y(x_\ell,a)$ and the values $h^*_j(x_\ell)$ as defined in Section (II.3). The summation with y(x,a) gives Y(a) and the summation over $h^*_j(x_\ell)$ gives

$$|H|^* = (\tilde{DA})\,|H|. \qquad \text{(III.2.5)}$$

After the summation is finished, the routine performs the division of $|H|^*$ by the matrix $\tilde{DA}$ and returns

$$|H| = (\tilde{DA})^{-1}\,|H|^*. \qquad \text{(III.2.6)}$$

3. Generation of Integration Points by INTIME

The coordinates of every integration point are generated by the secondary routine GETUM2. Versions of GETUM2 exist that generate points representing interacting particles by using a Monte Carlo technique. Other versions of GETUM2 can be programmed for other applications or to replace the Monte Carlo technique. We describe here the action of INTIME only when supplied with a GETUM2 using Monte Carlo technique.

In addition to the coordinates, GETUM2 provides a temporary weight $\Omega_\ell^{(o)}$ that is inversely proportional to the local density of points. DOME is then called. There, the user can reject the point to restrict the domain of integration. If DOME keeps the point, it must also return the vector $|x|_\ell$ in a convenient format for future use in HUME to com-

pute $y(x_\ell, a)$.

If no special secondary routine FLAME is programmed, all INTIME does is to store the vector $|x|_\ell$ and give the event a definitive weight Ω_ℓ equal to

$$\Omega_\ell = \frac{\Omega_\ell^{(o)}}{\sum_{\ell=1}^{Nint} \Omega_\ell^{(o)}}, \qquad \text{(III.3.1)}$$

where Nint is the number of points retained in DOME. The normalization of the weights expressed by (III.3.1) will make the functions w independent of the number of fake events produced, except for the fluctuations in the Monte Carlo generation.

If reference functions are to be used, however, a routine FLAME has to be written that returns the value of the N_{ref} functions $(fr)_n(x_\ell)$ for each point $|x|_\ell$. After the generation is over, FLAME must return the N_{ref} values of their integral Fr_n. Using vectorial notation,

$$|Fr| = \int |fr|\,(x)\,dx. \qquad \text{(III.3.2)}$$

INTIME performs the following computation:

$$|\Phi r| = \sum_{\ell=1}^{Nint} \Omega_\ell^{(o)}\ |fr|\,(x_\ell), \qquad \text{(III.3.3)}$$

$$D = \sum_{\ell=1}^{Nint} (\Omega_\ell^{(o)})^2\ |fr|\,(x_\ell)\ |\tilde{fr}|\,(x_\ell), \qquad \text{(III.3.4)}$$

$$c_{fr} = \frac{|\tilde{Fr}|\ D^{-1}\ |Fr|}{|\tilde{\Phi r}|\ D^{-1}\ |Fr|}\ , \qquad \text{(III.3.5)}$$

$$|q| = D^{-1}\ |Fr| - c_{fr}\ D^{-1}\ |\Phi r|. \qquad \text{(III.3.6)}$$

Then INTIME goes over the integration points again and corrects their weights, giving the ℓ^{th} point a new weight Ω_ℓ that will be used in the future integrations:

$$\Omega_\ell = c_{fr}\ \Omega_\ell^{(o)} + (\Omega_\ell^{(o)})^2\ |\tilde{q}|\ |fr|\ (x_\ell). \qquad \text{(III.3.7)}$$

If a function f(x) is a linear combination of the functions $(fr)_n(x)$,

$$f(x) = |\tilde{fr}|(x)\ |c|\ , \qquad \text{(III.3.8)}$$

where the c_n's are coefficients independent of x, the result of the summation (III.2.2) is the exact integral F of f:

$$\Phi = \sum_{\ell=1}^{Nint} \Omega_\ell^{(o)}\ c_{fr}\ |\tilde{fr}|\ (x_\ell)\ |c| +$$

$$+|\tilde{q}| \sum_{\ell=1}^{Nint} (\Omega_\ell^{(o)})^2\ |fr|\ (x_\ell)\ |\tilde{fr}|(x_\ell)|c| = |\tilde{Fr}|\ |c| = F \qquad \text{(III.3.9)}$$

If f(x) is not a linear combination of the $(fr)_n$'s, the result of (III.2.2) depends on the randomness of the Monte Carlo generation. However, there is an expectation value $<\Phi>$ for the quantity Φ and it is the same whether Ω_ℓ is computed with (III.3.7) or with (III.3.1) because

$$\langle \Phi r \rangle = |Fr| . \qquad \text{(III.3.10)}$$

4. Histograms -- FAME and COCHIS

Histograms involve different categories of events. Among the experimental events, there are the events of category 1 whose specifications fall into the domain of the fit and there are the events of category 2 which fall outside but that the user wants to include in some histograms though not in the fit. Those histograms can be obtained by calling COCHIS if proper preparation has been provided during the call to IMME [6].

If displayed curves are wanted on the histograms, the same Monte Carlo generation that provides integration points can be used to construct fake events. Actually, the Nint integration points are used as fake events for the histograms. They are the events of category 3, all falling inside the domain accepted for the fit. If the user wants more fake events for the histograms then he has integration points, he can trigger the generation of events of category 4: fake events falling in the domain of the fit but not accepted as integration points.

There is a fifth category of events, the fake events that fall outside the domain of the fit. The events of category 5 are the fake events that correspond to the experimental events of category 2.

We define N_p to be the number of events of categories 3 and 4 only. They will be used with their preliminary weight $\Omega_\ell^{(o)}$ and not their

corrected weight Ω_ℓ (see Section III.3). For types 3, 4, and 20, the normalization factor C_p is computed from a summation over the Nint events of category 3 and another summation over the N_p events of categories 3 and 4.

$$C_p = \frac{\sum_{\ell=1}^{Nint} \Omega_\ell}{\sum_{\ell=1}^{Np} \Omega_\ell^{(o)}} \qquad \text{(III.4.1)}$$

To get the fitted curve displayed on the histograms, FAME must be called with the parameters $|\alpha|$ in the transfer vector. Then, FAME will give each event of categories 3, 4, and 5 a new weight

$$\Omega'_\ell = \Omega_\ell^{(o)} \; y\,(|x|_\ell, |\alpha|)\; C_p, \qquad \text{(III.4.2)}$$

where the value for $y\,(|x|_\ell, |\alpha|)$ will be obtained by calling HUME with $|x|_\ell$ and $|\alpha|$ in the transfer vector. If COCHIS is called next, it will histogram the points $|x|_\ell$ with the weights Ω'_ℓ to represent the fitted curve along with the experimental events.

For types 11, 12, 13, and 14, $y\,(|x|, |\alpha|)$ represents only the shape of the distribution and was fitted independently of the normalization. Therefore, the factor C_p used by FAME in (III.4.2) is not (III.4.1) but, instead,

$$C_p = \frac{\sum_{\ell=1}^{Nint} \Omega_\ell}{\sum_{\ell=1}^{Np} \Omega_\ell^{(o)}} \quad \frac{W_t}{Y(\alpha)} \quad , \qquad \text{(III.4.3)}$$

where W_t is the sum of the weights of the experimental events as in (II.1.8) and $Y(\alpha)$ is the integral of $y(x,\alpha)$. For types 12, 13, and 14, w_i at the last step of the fit is $Y(\alpha)$. FAME will use the value w_i stored there. For type 11 the value of w_i must be filled by the user with the proper value of Y. If the user does not know it he can call NORME to find out.

The user may want to display histograms with curves corresponding to parameters $|a|$ different from the value $|\alpha|$ of the last fit. Then he can call FAME with those parameters $|a|$ in the transfer vector. If only the shape of $y(x,a)$, not its normalization, is meaningful, he should set the type to 11, 12, 13, or 14 and set w_i to the value that will give him the proper normalization. However, if the normalization of y is meaningful, he should set the type to 3, 4, or 20. The coefficient then is going to be computed according to (III.4.1), and w_i will be irrelevant.

ACKNOWLEDGMENTS

The authors are indebted to Dr. Ronald Ross and Dr. Gerald Lynch for helpful discussions, Dr. Jerry Friedman for his contribution to many parts of the programming, and to Dr. Barrie Pardoe for reviewing our manuscript.

FOOTNOTES AND REFERENCES

*Work done under the auspices of the U.S. Atomic Energy Commission.

1. F.T. SOLMITZ, Ann. Ev. Nucl. Sci. 14 (1964), 375.

2. F. JAMES and M. ROOS, "Minuit", CERN Comp. 6000 Ser. Prog. Lib., D-506, Geneva, Switzerland, 1969. W. E. HUMPHREY and B. J. COTTRELL, "Minfun", Group A Prog. Note, P-6, Lawrence Radiation Laboratory, 1966.

   W. C. DAVIDON, The Computer Journal, 10 (1968), 406.

NOTE: This is a partial list. We apologize for not being able to quote all existing programs.

3. In some versions the subroutine that the user has to provide gives the theoretical prediction of the data, and the least-square or likelihood function is already built into the standard program; see for instance:

   E. R. BEALS, "Variable Metric Minimization, VARMIT", Math. and Comp. Dept., E4 BKY VARMIT, Lawrence Radiation Laboratory, Berkeley, 1967.

4. In OPTIME the rule is to specify which parameters are variable whereas in MINUIT one specifies which ones are to remain constant.

5. P. H. EBERHARD, "Introduction to MAXIME", Group A Phys. Notes, 708, Lawrence Radiation Laboratory, Berkeley, 1970. P. H. EBERHARD, "Types of Fit," Group A Phys. Notes, 709, Lawrence Radiation Laboratory, Berkeley, 1970.

6. P. H. EBERHARD and W. O. KOELLNER, "Users Manual For the OPTIME System", Lawrence Radiation Laboratory Report, UCRL-20160, Lawrence Radiation Laboratory, Berkeley, 1970.

7. W. O. KOELLNER, "A Users Guide to OPTIME", Group A Prog. Note, P-183, Lawrence Radiation Laboratory, Berkeley, 1969.

8. J. FRIEDMAN, "Sage", Group A Prog. Note, P-189, Lawrence Radiation Laboratory, Berkeley, 1969. J. FRIEDMAN, "Random Event Generation With Nonconstant Frequency Distributions", Lawrence Radiation Laboratory Report, UCRL-19206, Lawrence Radiation Laboratory, Berkeley, 1969 (to be published in J. Comp. Phys.). J. FRIEDMAN, "Sage 3", Group A Phys. Note, Lawrence Radiation Laboratory, Berkeley, 1970 (received from the author).

9. J. FRIEDMAN and A RITTENBERG, "Kiowa, A General Description", Group A Prog. Note, P-171, Lawrence Radiation Laboratory, Berkeley, 1968.

10. The current official version of OPTIME can always be obtained from W. O. Koellner, Building 50, Room 228, Lawrence Radiation Laboratory, Berkeley, CA. 94720.

11. F. B. HILDEBRAND, "Advanced Calculus for Application", 362, Prentice-Hall, Inc. Englewood Cliffs, N.J., 1963.

12. F. E. HOHN, "Elementary Matrix Algebra", 264, Macmillan, New York, 1965.

FIGURE CAPTIONS

Fig. 1 Cutting the Step According to Mode 7.

Fig. 2 Determination of $|v|_3$.

Table I. Coefficients Computed in MAXIME

Type	3	4	11	12	13	14	20
$wfac_r$	$\eta_k \ln y$		$\eta_k \ln y$		$\eta_k \ln y$		$\eta_k y$
$wfac_i$	y		----	y	y		y^2
w	$w_r - w_i$		w_r		$w_r - W_t \ln w_i$		$W_r - \frac{w_i}{2}$
$Ufac_r$	η_k/y		η_k/y		η_k/y		η_k
$Ufac_i$	1		-----		1		y
$\lvert U\rvert^*$	$\lvert U_r\rvert^* - \lvert U_i\rvert^*$		$\lvert U_r\rvert^*$		$\lvert U_r\rvert^* - \frac{W_t}{w_i}\lvert U_i\rvert^*$		$\lvert U_r\rvert^* - \lvert U_i\rvert^*$
$Matfac_r$	η_k/y^2	----	η_k/y^2	----	η_k/y^2	----	----
$Matfac_i$	----	$1/y$	----	$1/y$	----	$1/y$	1
E^*	(MAT) (MAT) is given by (II.4.7) and (II.4.8)		(MAT)	$\frac{W_t}{w_i}$ (MAT)	(MAT) $- \frac{\lvert U\rvert^*\lvert\tilde{U}_i\rvert^* + \lvert U_i\rvert\lvert\tilde{U}\rvert^*}{w_i} - \frac{W_t}{w_i^2}\lvert U_i\rvert^*\lvert\tilde{U}_i\rvert^*$	$C_{E2}\frac{W_t}{w_i}\left[(MAT) - \frac{U_i^*\tilde{U}_i^*}{w_i}\right]$ C_{E2} given by (II.2.12)	(MAT)

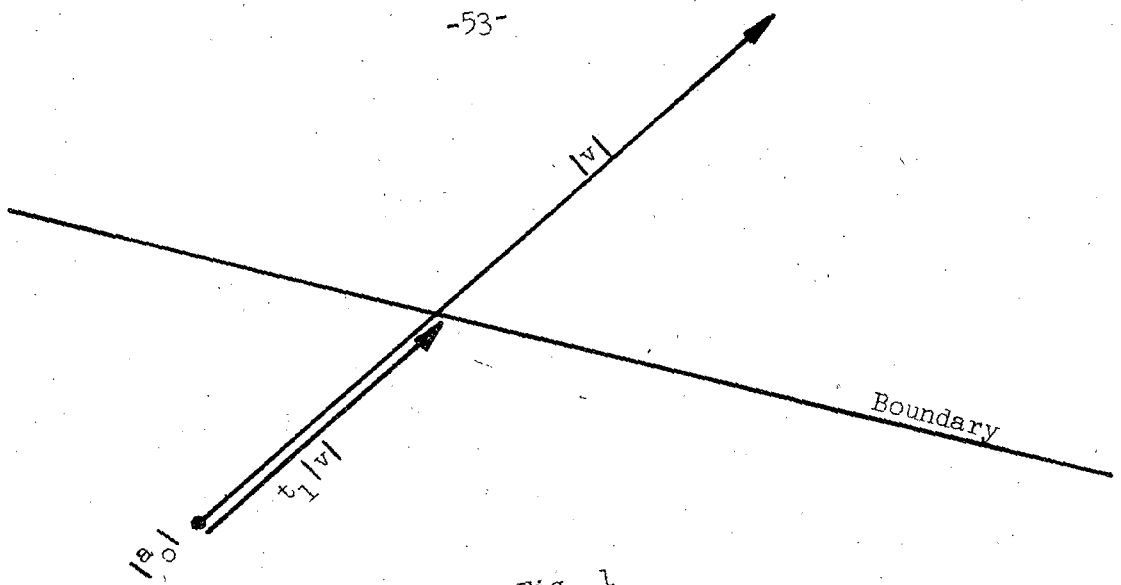

Fig. 1

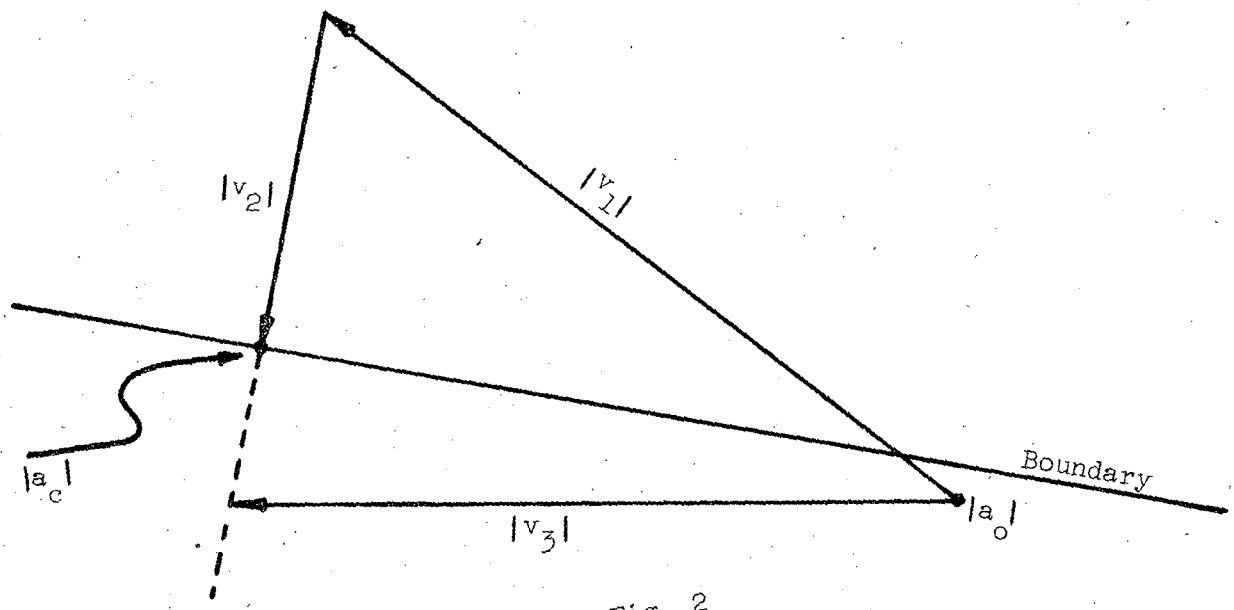

Fig. 2

LEGAL NOTICE

TECHNICAL INFORMATION DIVISION
LAWRENCE RADIATION LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720